Copyright © 2012 NTT DATA INTRAMART CORPORATION

目次

- 1. 改訂情報
- 2. はじめに
  - 2.1. 本書の目的
  - 2.2. 前提条件
  - 2.3. 対象読者
  - 2.4. 用語解説
- 3. セットアップの流れ
- 4. ミドルウェアのセットアップ
  - 4.1. JDK
  - 4.2. データベース
  - 4.3. Web Application Server
  - 4.4. Web Server
  - 4.5. Apache Cassandra
  - 4.6. Apache Solr
- 5. WARファイルの作成
  - 5.1. プロジェクトの作成とモジュールの選択
  - 5.2. ユーザモジュール
  - 5.3. intra-mart Accel Platform の設定ファイル
  - 5.4. WARファイルの出力
  - 5.5. 静的ファイルの出力
- 6. Web Application Server の起動・停止
  - 6.1. Windows
  - 6.2. Linux
- 7. デプロイ
  - 7.1. WAR ファイルのデプロイ
  - 7.2. 静的ファイルの配置
- 8. テナント環境セットアップ
  - 8.1. システム管理者情報
  - 8.2. テナント情報
  - 8.3. テナント環境情報
  - 8.4. パスワード保存方式設定
  - 8.5. テナント管理者情報
  - 8.6. LDAP連携・設定
  - 8.7. ログインセッション管理
  - 8.8. Apache Cassandra接続情報
  - 8.9. Apache Solr接続情報
  - 8.10. 多要素認証機能
  - 8.11. ベクトルデータベース接続情報
  - 8.12. 登録
- 9. ライセンスの登録
  - 9.1. ライセンスについて
  - 9.2. ライセンスの登録
- 10. アップデート・パッチの適用・モジュール構成の変更
  - 10.1. アップデート
  - 10.2. パッチ
  - 10.3. モジュール構成の変更
- 11. 付録
  - 11.1. クラスタリング
  - 11.2. 二重ログイン防止機能
  - 11.3. 統合Windows認証
  - 11.4. SMTP認証で OAuth2.0 アクセストークン を使用する
  - 11.5. IM-LogicDesigner のメール受信タスクで OAuth2.0 アクセストークン を使用する
  - 11.6. IM-Notice
  - 11.7. Accel Platform Mobile
  - 11.8. セッション管理モジュール

- 11.9. スマートメニュー
- 11.10. WARファイルによる複数テナント
- 11.11. テナント解決機能
- 11.12. ポート一覧
- 11.13. IM-Juggling の応用
- 11.14. intra-mart Accel Platform のチューニング
- 11.15. バックアップ・リストア（復元）
- 11.16. アンインストール
- 11.17. サンプルデータの投入
- 11.18. セットアップで困ったら・・・
- 11.19. DocuWorks Content Filter のインストール方法
- 11.20. intra-mart Accel Platform のヘルスチェック

改訂情報

変更年月日	変更内容
2012-10-01	初版
2012-11-01	第2版 下記を追加・変更しました ■ 「*データベースサーバ*」を追加 ■ 「*セットアップで困ったら・・・*」を追加
2012-12-21	第3版 下記を追加・変更しました ■ 「*ベースURL*」を追加 ■ 「*認可ポリシー設定キャッシュ*」を追加 ■ 「*IM-Juggling を利用中にエラーが発生してしまう場合*」を追加
2013-04-01	第4版 下記を追加・変更しました ■ intra-mart Accel Platform 2013 Spring(Climbing) のシステム要件に合わせて内容を変更 ■ テナント環境 の構築で「ポートレットの初期化」を不要のため削除 ■ 「*LDAP認証設定ファイル（アカウントの認証にLDAP認証を利用する場合）*」の説明を変更 ■ 「*テナント環境セットアップ・サンプルデータセットアップに失敗した場合*」の説明を変更 ■ 「*外部メニュー連携*」の説明を追加 ■ 「*Resin でWAR ファイルのデプロイ中にエラーが発生する場合*」の説明を変更 ■ 「*データベースサーバ*」にデータベースの権限に関するコラムを追加 ■ 「*WAR ファイルのアンデプロイ*」の説明を変更 ■ 「*デプロイ直後の Web Application Server 起動時にエラーが発生する場合*」を追加 ■ 「*認可リソースグループ設定キャッシュ*」の説明を追加 ■ 「*メニュー側ルーティング設定キャッシュ*」の説明を追加 ■ 「*認可IPv4 サブジェクト設定キャッシュ*」の説明を追加 ■ 「*グローバルナビキャッシュ*」の説明を追加 ■ 「*個人設定メニューキャッシュ*」の説明を追加 ■ 「*Windowsサービスへの登録・削除*」の説明を追加 ■ 「*Linuxデーモンへの登録、削除*」の説明を追加 ■ 「*WAR ファイルの出力*」にWARファイルに関するコラムを追加 ■ 「*Apache HTTP Server*」の設定方法を追加 ■ 「*Apache Cassandra*」の設定方法を追加 ■ 「*プロジェクトの作成とモジュールの選択*」にプロキシ設定に関するコラムを追加 ■ 「*データベースサーバ*」に WARファイルによる複数テナント および バーチャルテナントによる複数テナント に関するコラムを追加 ■ 「*データベースサーバ*」にMicrosoft SQL Serverに関するコラムを追加 ■ 「*モジュールのアップデート*」追加しました。
2013-04-30	第5版 下記を追加・変更しました ■ 「*アップデート・パッチの適用・モジュール構成の変更*」を追加 ■ 「*プロジェクトの作成とモジュールの選択*」にプロキシ設定に関する説明を変更

変更年月日	変更内容
2013-07-01	第6版 下記を追加・変更しました ■ 「*Oracle Database*」スキーマに付与する権限についてわかりづらい表現を改善 ■ 「*テナント環境セットアップ*」にコンテキストパスに関するコラムを追加 ■ 「*コンソール起動・停止*」に Windows Server 2012 利用時における注意点を追加 ■ 「Internet Information Services 7.5 が使用する .NET Framework バージョンの確認・変更方法」を追加 ■ 「*.NET Framework のセットアップ*」を追加 ■ 「*統合Windows認証*」に設定時における注意点を追加 ■ 「*Resinの設定*」にタイムゾーンの設定における注意点を追加 ■ 「*Network*」の設定方法を改善しました。 ■ 「*WAR ファイルによる複数テナント*」の設定方法を追加 ■ 「*Linuxデーモンへの登録、削除*」における実行ユーザの変更方法を追加
2013-10-01	第7版 下記を追加・変更しました ■ 「*SAStruts版ポートレットが404エラーでアクセスできない場合*」を追加 ■ 「*Linux環境でWAR ファイルのデプロイ中にファイル入出力エラーが発生する場合*」を追加 ■ 「*Oracle WebLogic Server 12c R2(12.2.1) でテナント環境セットアップに失敗した場合*」を追加 ■ 「*プロジェクトの作成とモジュールの選択*」の説明を変更 ■ 「*DataSource*」の説明を変更 ■ 「*SAStruts用設定ファイル（SAStruts版ポートレットを利用する場合）*」を追加 ■ 「*ファイルのアップロードを制限する方法*」を追加 ■ 「*認可ポリシー設定キャッシュ*」の説明を変更 ■ 「*認可リソースグループ設定キャッシュ*」の説明を変更 ■ 「*メニュー側ルーティング設定キャッシュ*」の説明を変更 ■ 「*認可IPv4サブジェクト設定キャッシュ*」の説明を変更 ■ 「*グローバルナビキャッシュ*」の説明を変更 ■ 「*個人設定メニューキャッシュ*」の説明を変更 ■ 「DataSourceマッピングの設定」に注意点を追加 ■ 「*外部メニュー連携*」に「メニュー設定画面での表示順序設定」に関する説明を追加 ■ 「*RHEL6の場合*」に「複数のIPアドレスが設定されている場合」に関する説明を追加 ■ 「*Resinの設定*」に作業ディレクトリ、変更検出に関する注意点を追加 ■ 「*プロジェクトの作成とモジュールの選択*」にTERASOLUNA Global Frameworkに関する説明を追加 ■ 「*intra-mart Accel Platform の設定ファイル*」にTERASOLUNA Global Frameworkに関する説明を追加 ■ 「*TERASOLUNA Server Framework for Java (5.x) 用設定ファイル（シェアードデータベースを利用する場合）*」を追加 ■ 「TERASOLUNA Global Framework用設定ファイル（リポジトリ層にJPAを利用する場合）」を追加 ■ 「TERASOLUNA Global Framework用設定ファイル（リポジトリ層にMyBatisを利用する場合）」を追加 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」を追加

変更年月日	変更内容
2014-01-01	第8版 下記を追加・変更しました ■ 目次構成を改善しました。 ■ intra-mart Accel Platform 2013 Winter(Felicia) のシステム要件に合わせて内容を変更 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」に「*2013 Summer(Damask) からアップデート*」に関する説明を追加 ■ 「*インポート処理結果ログの確認*」にエラー原因の追跡についての説明を追記 ■ 「*テキスト抽出設定（solr-extractor-config.xml）*」の説明を追加 ■ 「*テナント環境セットアップ中にタイムアウトが発生した場合*」を追加 ■ 「*IM-Workflow 用設定ファイル （シーケンスオブジェクト採番機能を利用する場合）*」を追加 ■ 「*アップデート・パッチの適用・モジュール構成の変更*」にCassandraのバージョンアップについての記載を追加 ■ 「*Resin でWAR ファイルの再デプロイが正常にできない場合*」を追加 ■ 「*IMBox モジュールを外す方法*」を追加 ■ 「*iPadからアクセス時にクライアントタイプをPCとして扱う場合*」を追加 ■ 「*外部メニュー連携サービス for intra-mart WebPlatform v7.2*」にIM-SecureSignOn for Accel Platform利用時の設定についての記載を追加 ■ 「*アップデート・パッチの適用・モジュール構成の変更*」に静的コンテンツ、テナント環境セットアップについて説明を追加 ■ 非推奨のため「Apache HTTP Server （mod_caucho.dll による設定）」を削除しました。 ■ 「*Apache Solr*」に「利用するJavaのバージョンについて」の説明を追加
2014-02-14	第9版 下記を追加・変更しました ■ 「*Internet Information Services（IIS）*」の設定方法を改善 ■ 「*統合Windows認証*」の設定方法を変更
2014-04-01	第10版 下記を追加・変更しました ■ 目次構成を改善しました。 ■ 「*外部メニュー連携*」のメニュープロバイダ情報の説明を変更、および バーチャルテナントによる複数テナント 運用時のメニュープロバイダの設定についての説明を追加 ■ 「*自動ログイン機能を利用する場合*」を追加 ■ 「*統合Windows認証*」の設定方法を変更 ■ 「*intra-mart Accel Platform の設定ファイル*」にWARファイル展開時の注意点を追加 ■ 「*intra-mart Accel Platform の設定ファイル*」に「*IM-Workflow システム設定*」を追加 ■ 「*プロジェクトの新規作成*」に エディション毎のモジュールの選択方法を追加 ■ 「*セットアップ実行結果ログの確認*」のログ出力先情報を変更 ■ 「*IMBox*」に「*intra-mart Accel Platform 2014 Spring(Granada)からの変更点*」を追加 ■ 「*統合Windows認証環境でWebサービスを利用する*」を追加 ■ 「*統合Windows認証環境で外部ソフトウェア連携機能を利用する*」を追加 ■ 「*認可IPv4サブジェクト設定キャッシュ*」にキャッシュサイズの計算式を追加 ■ 「*付録*」に「*テナント解決機能*」を追加 ■ 「*付録*」に「*Resin をクラスタリングしての分散環境*」を追加 ■ 「*付録*」に「*ポート一覧*」を追加 ■ 「*付録*」に「*IM-Juggling の応用*」を追加 ■ 「*付録*」に「*intra-mart Accel Platform のチューニング*」を追加 ■ 「*認可リソース閉塞情報キャッシュ*」を追加
2014-05-01	第11版 下記を追加・変更しました ■ 「*統合Windows認証*」に「*統合Windows認証機能を無効化するには*」を追加

変更年月日	変更内容
2014-08-01	第12版 下記を追加・変更しました ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」の「2013 Autumn(Eden) からアップデート」に「データベース検索時のエスケープフラグ」に関する説明を追加 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」に「2013 Winter(Felicia) からアップデート」に関する説明を追加 ■ 「*付録*」に「*WAR ファイルに含まれるモジュール情報・ショートモジュールIDの一覧を確認する方法*」を追加 ■ 「*付録*」に「*初回アクセス時に「[E.IWP.ADMIN.CONTEXT.10004] Tenant ID cannot be resolved.」が発生します。*」を追加 ■ 「*Apache Solr*」の記載を「Solr管理者ガイド」-「Solrのセットアップ」に移動 ■ 「*iAP-iWP間SSO連携（IM-HybridSSO）*」の説明を追加 ■ 「*外部メニュー連携*」に IM-HybridSSO についての説明を追加
2014-09-01	第13版 下記を追加・変更しました。 ■ 「*iAP-iWP間SSO連携（IM-HybridSSO）*」に iWP patch8 適用時の注意喚起を追加。
2014-12-01	第14版 下記を追加・変更しました。 ■ 「*ログインセッション管理*」の設定方法を追加 ■ 「*テナント環境情報*」に「グローバルナビ最大表示数」を追加 ■ 「*統合Windows認証*」に「*統合Windows認証機能の認証失敗時に通常のログイン機能を利用するには*」を追加 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」に「2014 Summer(Honoka) からアップデート」に関する説明を追加 ■ 「*ヘルプドロップダウンキャッシュ*」の説明を追加 ■ 「*WAR ファイル作成時の実行環境の変更*」の説明を追加 ■ 「*IM-Notice*」の設定方法を追加 ■ 「*Internet Information Services（IIS）*」に「OAuth認証モジュールを利用する場合の追加設定」を追加

変更年月日	変更内容
2015-04-01	第15版 下記を追加・変更しました。 ■ 「*ライセンスの登録*」を追加 ■ 「*統合Windows認証*」に Internet Information Services（IIS） が必須である旨の注意を追加 ■ 「*統合Windows認証*」に 利用できるブラウザの説明を追加 ■ 「*インターネットに接続できない環境で IM-Juggling を利用する場合*」を追加 ■ 「*プロキシ設定が必要な環境で IM-Juggling を利用する場合*」を追加 ■ 「*アップデート・パッチの適用・モジュール構成の変更*」に「*モジュール構成の変更*」を追加 ■ 「*バックアップ・リストア（復元）*」 に各対象ファイルの詳細と、復元手順の注意を追加 ■ 「*Resin でWAR ファイルのデプロイ中にエラーが発生する場合*」 に 「dependency-check-interval」の値の単位について説明を追加 ■ 「*PostgreSQL*」に PostgreSQL 9.4 を追加 ■ 「*ベースURL*」の説明を変更 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」に「*2014 Winter(Iceberg) からアップデート*」に関する説明を追加 ■ 「*IM-Juggling を利用せず、CUI でWAR ファイルを作成する方法*」に静的ファイルの出力方法を追記 ■ 「*IM-ContentsSearch*」の「*Solrサーバ接続設定（solr-config.xml）*」にポート番号に関する説明を追加 ■ TERASOLUNA Global FrameworkをTERASOLUNA Server Framework for Java (5.x)に変更 ■ 「TERASOLUNA Global Framework用設定ファイル（リポジトリ層にJPAを利用する場合）」を削除 ■ 「TERASOLUNA Global Framework用設定ファイル（リポジトリ層にMyBatisを利用する場合）」を削除 ■ 「*TERASOLUNA Server Framework for Java (5.x) 用設定ファイル（リポジトリ層にMyBatis3を利用する場合）*」を追加 ■ 「*TERASOLUNA Server Framework for Java (5.x) for Accel Platform を使用する場合の設定*」を追加 ■ 「*Locale*」 の システムロケール変更に関する説明を変更
2015-08-01	第16版 下記を追加・変更しました。 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」の内容を変更 ■ 「*モジュールのアップデート*」に 「*テナント環境セットアップ後の各種メンテナンス*」の説明を追加 ■ 「*テナント環境セットアップ後の各種メンテナンス（アップデートによるメンテナンス）*」を追加 ■ 「*テナント環境セットアップ*」に 2014 Spring(Granada) 以降のバージョンでLDAP認証モジュール、またはIMBoxモジュールを追加する場合の記載を追加 ■ 「*統合Windows認証*」の設定方法を変更 ■ 「Baidu+Amazon SNSを使用する場合」を追加 ■ 「*プロジェクトの作成とモジュールの選択*」に移行ツールおよび互換機能に関する注意書きを追加 ■ 「*セットアップで困ったら・・・*」に「*Resin で PreparedStatement の キャッシュサイズに大きな値を指定している場合にテナント環境セットアップが失敗する*」を追加 ■ 「*バックアップ・リストア（復元）*」内の分かりづらい表現を削除 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」の「*2014 Winter(Iceberg) からアップデート*」に 「【 TERASOLUNA Global Framework を利用している場合】」の説明を追加 ■ 「*プロジェクトの新規作成*」の TERASOLUNA Server Framework for Java (5.x) に関する説明を変更 ■ 「*ライセンスの登録*」に「テナント毎のライセンス設定」、「アプリケーションライセンス設定」へのリンクを追加 ■ 「*Office 365 連携*」を追加 ■ 「*WAR ファイルのアンデプロイ*」にファイルの削除の際 Resin の停止が必要な旨を追加 ■ 「*分散構成の場合*」にダイナミックサーバをWindowsサービスに登録する方法を追加

変更年月日	変更内容
2015-12-01	第17版 下記を追加・変更しました。 ■ 「*Apache HTTP Server*」にProxyTimeoutの設定方法を追加 ■ 「*DataSourceマッピングの設定*」の説明を変更 ■ 「*ライセンスの登録*」の説明を追加 ■ 「*Internet Information Services（IIS）*」の「OAuth認証モジュール、または統合Windows認証モジュールのリダイレクト機能を利用する場合の追加設定」を変更 ■ 「*統合Windows認証機能の認証失敗時にリダイレクトさせるには*」を追加 ■ 「*Resinの設定*」と「*負荷試験を実施する際の各種設定*」に、Resin の jvm_args の指定内容を変更 ■ 「*P12証明書ファイルの更新*」の説明を追加 ■ 「*WAR ファイル作成時の実行環境の変更*」に「*エラーページにエラーの内容を表示します。*」を追加 ■ 「*DataSource*」にPostgreSQL JDBCを使用する際の説明を追加 ■ 「*Resin をクラスタリングしての分散環境*」にスティッキーセッションに関するコラムを追加 ■ 「*LDAP認証設定ファイル（アカウントの認証にLDAP認証を利用する場合）*」にバージョンによってファイルの扱いが異なることを追記 ■ 「*LDAP認証設定ファイル（アカウントの認証にLDAP認証を利用する場合）*」に「*LDAP認証でSSL接続(LDAPS)を利用するための環境設定*」を追加 ■ 「*LDAP認証設定ファイル*」を変更 ■ 「*テナント環境セットアップ後の各種メンテナンス（アップデートによるメンテナンス）*」に「2015 Summer(8.0.11) からアップデート」を追加 ■ 「*ポート一覧*」に「モジュール開発支援ライブラリ」で使用されるポート番号の情報を追加
2016-04-01	第18版 下記を追加・変更しました。 ■ 「*テナント環境情報*」に日付・時刻の入力形式の変更を追加 ■ 「*パスワード保存方式設定*」を追加 ■ 「*Apache HTTP Server*」の「apache設定ファイル」の設定方法を変更 ■ 「*Internet Information Services（IIS）*」の「web.config」の設定方法を変更 ■ 「*IM-ContentsSearch*」に「*intra-mart Accel Platform 2016 Spring(Maxima)からの変更点*」を追加 ■ 「*レスポンスヘッダ設定*」を追加 ■ 「*DocuWorks Content Filter のインストール方法*」を追加
2016-08-01	第19版 下記を追加・変更しました。 ■ 「*付録*」に「*intra-mart Accel Platform のヘルスチェック*」を追加 ■ 「*Resinの設定*」に設定ファイルリファレンスのResinの設定へのリンクを追加 ■ 「*Resinの設定*」に「app_servers」プロパティに関するコラムを追加 ■ 「*Network*」にサーバ間通信で利用する最大スレッド数の設定を追加 ■ 「*Network*」にアドレス指定に関するコラムおよび通信プロトコルに関するコラムを追加 ■ 「*Resin をクラスタリングしての分散環境*」の説明を変更
2016-12-01	第20版 下記を追加・変更しました。 ■ 「*プロジェクトの新規作成*」のTERASOLUNA Server Framework for Java (5.x)に関する注意を変更 ■ 「*JDBC ドライバ*」に「*SAP HANA*」に関する記載を追加 ■ 「*DataSource*」に「*SAP HANA*」「*PostgreSQL*」に関する記載を追加 ■ 「*TERASOLUNA Server Framework for Java (5.x) for Accel Platform を使用する場合の設定*」の説明を変更 ■ 「*WAR ファイルの作成*」の「*intra-mart Accel Platform の設定ファイル*」に「*imuiPictureの暗号化キーの設定*」に関する記載を追加 ■ 「*テナント環境セットアップ後の各種メンテナンス（アップデートによるメンテナンス）*」に「2016 Summer(8.0.14) からアップデート」を追加 ■ DB2に関する記述を削除

変更年月日	変更内容
2017-04-01	第21版 下記を追加しました。 ■ 「*プロジェクトの新規作成*」に外部ソフトウェア接続モジュールに関する注意を追加 ■ 「*インターネットに接続できない環境で IM-Juggling を利用する場合*」にプロダクトファイルダウンロードに関する記載を追加 ■ 「*モジュールのアップデート*」にプロダクトファイルダウンロードに関する記載を追加 ■ 「*IM-Notice*」のブローカーサービス実行ファイルの取得の説明を変更 ■ 「*iAP-iWP間SSO連携（IM-HybridSSO）*」の個別パッチモジュールのダウンロード方法を変更 ■ 「*セッション管理モジュール*」の説明を追加 ■ 「*ポート一覧*」に「*Hazelcast*」に関する記載を追加
2017-08-01	第22版 下記を追加しました。 ■ 「*Apache HTTP Server*」にKeepAliveに関する設定を追加 ■ 「*Internet Information Services（IIS）*」にKeepAliveに関する設定を追加 ■ 「*統合Windows認証*」にKeepAliveに関する設定を追加 ■ 「*データベースサーバ*」にSQL Server 2012 以降での推奨する照合順序について追記 ■ 「*Resinの設定*」に `*.https` プロパティが非サポートである旨を追記 ■ 「*テナント環境セットアップ後の各種メンテナンス（アップデートによるメンテナンス）*」に「*2017 Spring(8.0.16) からアップデート*」を追加
2017-12-01	第23版 下記を追加・変更しました。 ■ 「*テナント環境セットアップ後の各種メンテナンス（アップデートによるメンテナンス）*」に「*2017 Summer(Quadra) からアップデート*」を追加 ■ 「*IM-Juggling を利用せず、CUI でWAR ファイルを作成する方法*」に製品版の出力方法を追加 ■ 「*アイコンキャッシュ*」を追加 ■ 「*Linuxデーモンへの登録、削除*」に「*RHEL7の場合*」を追加 ■ 「*Resin をクラスタリングしての分散環境*」を変更 ■ 「*intra-mart Accel Platform の分散環境（Resin をクラスタリングせずに構築）*」を追加 ■ 「*IM-Notice*」の「*ブローカーサービスの実行*」に停止方法のコラムを追加 ■ 「*JDBC ドライバ*」の各データベースのバージョン情報、および、URLを変更 ■ 「*DataSource*」のデータベースURLのフォーマットに関する記載を追加 ■ 「*スマートメニュー*」を追加 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」に「【 IM共通マスタ-設定ファイル (im-master-config.xml) を出力している場合】」の説明を追加
2018-04-01	第24版 下記を追加・変更しました。 ■ 「*IM-Workflow システム設定*」の設定ファイルの出力に関する説明を変更 ■ 「*IM-ContentsSearch*」に「*検索画面設定（contentssearch-display-config.xml）*」および「*intra-mart Accel Platform 2018 Spring(Skylark)からの変更点*」を追加 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」に「*2017 Winter(Rebecca) からアップデート*」の説明を追加 ■ 「*テナント環境セットアップ*」に「*多要素認証機能*」を追加 ■ 「*負荷試験を実施する際の各種設定*」に「*多要素認証機能*」に関する記載を追加 ■ 「*Resin （Windows版）のインストール*」、「*Resin （Linux版）のインストール*」にある Resin の入手先に関する記載を変更 ■ 「*JDBC ドライバ*」へJDBC 4.2に関する記載を追加、および、URLを変更 ■ 「*DataSource*」のJDBC 4に限定している記載を変更 ■ 「*セットアップの流れ*」のWeb Serverのステップを必須に変更 ■ 「*Web Server*」にWeb Serverに関する注意事項を追加 ■ 「静的ファイルの配置」の記述を「*デプロイ*」に変更

変更年月日	変更内容
2018-08-01	第25版 下記を追加・変更しました。 ■ 「*DataSourceの設定*」にプリペアドステートメントキャッシュに関する注意事項を追記 ■ 「*デプロイ*」に「*webapps ディレクトリに WAR ファイルを直接配置してデプロイ*」を追記 ■ 「*デプロイ*」の「deploy コマンド利用してデプロイ」に注意事項を追記 ■ 「*アンインストール*」に「*webapps ディレクトリに WAR ファイルを直接配置した場合のアンデプロイ*」を追記 ■ 「*intra-mart Accel Platform の分散環境（Resin をクラスタリングせずに構築）*」にデプロイを行う方法と注意事項を追記 ■ 「*Resin をクラスタリングしての分散環境*」のデプロイ時の注意事項を追記 ■ 「*テナント環境セットアップ後の各種メンテナンス（アップデートによるメンテナンス）*」に「*2018 Spring(Skylark) からアップデート*」を追加 ■ 「*アプリケーションの追加*」にモジュールID重複時に関するコラムを追加 ■ 「*WAR ファイルに含まれるモジュール情報・ショートモジュールIDの一覧を確認する方法*」にモジュールID重複時に関するコラムを追加 ■ 「*セッション管理モジュール*」の「*セッション情報の永続化を行う場合*」のデータソースの設定例を変更 ■ 「*DataSource*」の「*DataSourceの設定*」に ConnectionPoolDataSource に関する説明を追加 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」に「2018 Spring(Skylark) からアップデート」の説明を追加 ■ 「*モジュールのアップデート*」の「*WAR ファイルのアンデプロイ*」に注意事項を追加
2018-12-01	第26版 下記を追加・変更しました。 ■ 「*JDK*」の説明を、リリースノートに記載されているシステム要件を満たす JDK を入手するよう変更 ■ 「*IM-Juggling を利用中にエラーが発生してしまう場合*」の「*IM-Juggling が最新版になっている必要があります*」の説明を変更 ■ 「*テナント環境情報*」のリソース参照名に関するコラムの説明を変更
2018-12-27	第27版 下記を追加・変更しました。 ■ 「*JDBC ドライバ*」の Microsoft SQL Server のJDBCドライバの入手に関する説明を変更
2019-04-01	第28版 下記を追加・変更しました。 ■ 「*intra-mart Accel Platform の分散環境（Resin をクラスタリングせずに構築）*」にデプロイの説明を変更 ■ 「*Resin をクラスタリングしての分散環境*」の説明を変更 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」に「2018 Winter(Urara) からアップデート」の説明を追加 ■ 「*セッション管理モジュール*」の「*セッション情報の永続化を行う場合*」に関する注意事項を追加 ■ 「*セッション管理モジュール*」に「*セッションストアメモリサイズの計算式*」の説明を追加 ■ 「*Resin （Linux版）のインストール*」の「*フォントの設定*」の説明を変更 ■ 「*負荷試験を実施する際の各種設定*」の「*クラスタリング環境を想定した負荷試験を行うための設定*」に関する注意事項を追加 ■ 「*Microsoft SQL Server*」の Microsoft SQL Server のJDBCドライバの入手に関する説明を変更 ■ 「*プロジェクトの作成とモジュールの選択*」の「*プロジェクトの新規作成*」に記載されている Metro と OpenPortal WSRP の選択に関する注意事項を変更 ■ 「*個人設定メニューキャッシュ*」の説明を変更 ■ 「*IM-Workflow 一覧画面を利用する場合の遷移先プラグインに関する設定*」を追加 ■ 「*テナント環境セットアップ後の各種メンテナンス（アップデートによるメンテナンス）*」に「*2018 Winter(Urara) からアップデート*」を追加

変更年月日	変更内容
2019-08-01	第29版 下記を追加・変更しました。 ■ 「*クロスオリジンリソース共有のキャッシュ*」の説明を追加 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」で不要な手順を削除 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」に「*2019 Spring(Violette) からアップデート*」を追加 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」の「*2014 Winter(Iceberg) からアップデート*」 からライブラリ修正に関する記述を削除 ■ 「*アップデート・パッチの適用・モジュール構成の変更*」に「*Resin のアップデート後に必要なメンテナンス作業*」のリンクを追加
2019-12-01	第30版 下記を追加・変更しました。 ■ 「*Web Application Server 起動後にログインができない（データベースに接続できない）場合*」の「*Web Application Server 起動時のコンソール情報の確認*」に関する説明を変更 ■ 「*プロジェクトの新規作成*」に dicon ファイルのエラー出力に関する注意事項を追加 ■ 「*Apache HTTP Server*」の「*apache設定ファイルの編集*」の説明を変更 ■ 「*Internet Information Services（IIS）*」の「*前提条件*」に条件を追加 ■ 「*テナント環境セットアップ後の各種メンテナンス（アップデートによるメンテナンス）*」に「*2019 Summer(Waltz) からアップデート*」を追加 ■ 「*統合Windows認証*」にMicrosoft Edgeの設定を追加 ■ 「*JDBC ドライバ*」の JDBCダウンロードリンクを2019/12/01現在の最新版に更新しました。 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」に「*2019 Summer(Waltz) からアップデート*」を追加
2020-04-01	第31版 下記を追加・変更しました。 ■ 「*統合Windows認証*」の「*「インターネット オプション」の設定*」に関する説明を変更 ■ 「*統合Windows認証*」の制限事項「統合Windows認証のユーザ認証に失敗した際にリダイレクトされない場合があります。」を削除 ■ 「*Internet Information Services（IIS）*」の説明を変更 ■ 「*IM-Juggling を利用せず、CUI でWAR ファイルを作成する方法*」のディレクトリ内の構造を変更 ■ 「*デプロイ*」の「deploy コマンド利用してデプロイ」を削除 ■ 「*WAR ファイルのアンデプロイ*」の「undeploy コマンド利用してアンデプロイ」を削除 ■ 「*Resin をクラスタリングしての分散環境*」の「クラスタへのデプロイ」を削除 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」に「*2019 Winter(Xanadu) からアップデート*」を追加 ■ 「*テナント環境セットアップ後の各種メンテナンス（アップデートによるメンテナンス）*」に「*2019 Winter(Xanadu) からアップデート*」を追加 ■ 「*Apache HTTP Server*」の「*apache設定ファイルの編集*」に条件を追加 ■ 「*Internet Information Services（IIS）*」の「*設定*」に条件を追加
2020-08-01	第32版 下記を追加・変更しました。 ■ 「*IM-Notice*」の「GCMを使用する場合」を「*FCMを使用する場合*」に変更 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」に「*2020 Spring(Yorkshire) からアップデート*」を追加 ■ 「*プロジェクトの新規作成*」の注意書きを変更
2020-09-01	第33版 下記を追加・変更しました。 ■ 「*プロジェクトの作成とモジュールの選択*」にIM-RPAに関する説明を追加
2020-12-01	第34版 下記を追加・変更しました。 ■ 「*プロジェクトの作成とモジュールの選択*」にアプリケーション別の参照先を追加 ■ 「*プロジェクトの作成とモジュールの選択*」に IM-Juggling は *プロダクトファイルダウンロード* よりダウンロード可能であることを追加 ■ 「*テナント環境セットアップ後の各種メンテナンス（アップデートによるメンテナンス）*」に「*2020 Summer(Zephirine) からアップデート*」を追加

変更年月日	変更内容
2021-04-01	第35版 下記を追加・変更しました。 ■ 「*IM-Notice*」にブローカー利用時の im-notice-mq-config.xml の設定方法に関する説明を追加 ■ 「*IM-Juggling を利用せず、CUI でWAR ファイルを作成する方法*」の「ディレクトリ内の構造」として記載されている im_juggling_ant-1.0.X.jar のバージョンを変更 ■ 「*テナント環境セットアップ後の各種メンテナンス（アップデートによるメンテナンス）*」に「*2020 Winter(Azalea) からアップデート*」を追加 ■ 「*Resin で PreparedStatement の キャッシュサイズに大きな値を指定している場合にテナント環境セットアップが失敗する*」にエラー発生時の具体的なエラーメッセージを追記
2021-08-01	第36版 下記を追加・変更しました。 ■ 「*.NET Framework のセットアップ*」に各OS別インストール方法のコラムを追加 ■ 「*テナント環境セットアップ後の各種メンテナンス（アップデートによるメンテナンス）*」に「*2021 Spring(Bergamot) からアップデート*」を追加
2022-06-01	第37版 下記を追加・変更しました。 ■ 「*SMTP認証で OAuth2.0 アクセストークン を使用する*」を追加 ■ 「*IM-LogicDesigner のメール受信タスクで OAuth2.0 アクセストークン を使用する*」を追加 ■ 「*JavaMail*」に画面操作についてのコラムを追加 ■ 「*Internet Information Services（IIS）*」の「*設定*」に、リダイレクトに対応するための手順を追加 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」に「*2021 Winter(Dandelion) からアップデート*」を追加 ■ 「*テナント環境セットアップ後の各種メンテナンス（アップデートによるメンテナンス）*」に「*2021 Winter(Dandelion) からアップデート*」を追加 ■ 「*IM-ContentsSearch*」の設定の有効化方法の記述を修正 ■ 「*DocuWorks Content Filter のインストール方法*」のDocuWorks商品情報ページのURLを修正 ■ 「*Resin （Windows版）のインストール*」にアップデートを行った場合の注意事項を追加 ■ 「*Resin （Linux版）のインストール*」にアップデートを行った場合の注意事項を追加
2022-07-29	第38版 下記を追加・変更しました。 ■ 「*2021 Winter(Dandelion) からアップデート*」に「*【ViewCreator（クロス集計）を利用している場合】*」を追加
2022-12-01	第39版 下記を追加・変更しました。 ■ 「*DataSourceの設定*」の Microsoft SQL Server の設定を修正 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」に「*2022 Spring(Eustoma) からアップデート*」を追加
2023-04-01	第40版 下記を追加・変更しました。 ■ 「*前提条件*」の Application Request Routing のインストール方法を修正 ■ 「*テナント環境セットアップ後の各種メンテナンス（アップデートによるメンテナンス）*」に「*2022 Winter(Freesia) からアップデート*」を追加 ■ 「*ライセンスの登録*」にカスタマーサクセスライセンスの説明を追加 ■ 「*プロジェクトの新規作成*」の注意書きにベーシックとプロフェッショナルに関する説明を追加 ■ 「*プロジェクトの新規作成*」のコラムにベーシックとプロフェッショナルに関する説明を追加 ■ 「*IM-Notice*」に利用時の注意事項を追加 ■ 「*Resinの設定*」のスタンドアローン／分散構成共通のコラムにカスタマーサクセスライセンスとプロキシ利用時に設定する JVM 引数を追加 ■ 「*ライセンスの登録*」のコラムにカスタマーサクセスライセンスとプロキシ利用時に設定する JVM 引数を追加 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」に「*2022 Winter(Freesia) からアップデート*」を追加
2023-05-31	第41版 下記を追加・変更しました。 ■ 「*IM-Notice*」の「Baidu+Amazon SNSを使用する場合」を削除

変更年月日	変更内容
2023-06-30	第42版 下記を追加・変更しました。 ■ 「*IM-Notice*」から Accel Platform Mobile に関しての記述を削除 ■ 「*Accel Platform Mobile*」ページを追加
2023-10-01	第43版 下記を追加・変更しました。 ■ 「*テナント環境セットアップ後の各種メンテナンス（アップデートによるメンテナンス）*」に「*2023 Spring(Gerbera) からアップデート*」を追加 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」に「*2023 Spring(Gerbera) からアップデート*」を追加 ■ 「*IM-Juggling を利用せず、CUI でWAR ファイルを作成する方法*」に Java 17 で運用する場合の注意事項を追加 ■ 「*IM-Notice*」の「*ブローカーサービスの実行*」に Java 17 で運用する場合のコラムを追加 ■ 「*Resinの設定*」のスタンドアローン／分散構成共通の注意書きに、Java 8、Java 11 を利用する場合の JVM 引数に関する説明を追加 ■ 「*Web Application Server の起動・停止*」に分散環境で複数のサーバを同時起動する場合の注意事項を追加 ■ 「*Storage*」に注意書きを追加 ■ 「*データベースサーバ*」で使用するpgAdminをⅢから4に変更 ■ 「*セットアップで困ったら・・・*」に「*Apache POI 5.2.3 を利用している機能で Microsoft Office ファイル読み込み時にエラーとなる場合*」を追加 ■ 「*DocuWorks Content Filter のインストール方法*」を Docuworks 8 から Docuworks 9.1 に更新 ■ 「*IM-ContentsSearch*」に「*intra-mart Accel Platform 2023 Autumn(Hollyhock)からの変更点*」を追加 ■ 「*IM-Notice*」の「*モバイル通知機能（iOS版）*」および「*モバイル通知機能（Android版）*」に以下を追加 　■ 「*ポリシーの作成*」、「*アクセスキーとシークレットキーの作成*」、および、「*ロールの作成*」を追加 　■ 「*IM-Notice Mobile設定ファイルの編集*」にロールについての記載を追加 ■ 「*Accel Platform Mobile*」の「*モバイル通知機能（iOS版）*」および「*モバイル通知機能（Android版）*」に以下を追加 　■ 「*ポリシーの作成*」、「*アクセスキーとシークレットキーの作成*」、および、「*ロールの作成*」を追加 　■ 「*Accel Platform Mobile 設定ファイルの編集*」にロールについての記載を追加
2023-10-31	第44版 下記を追加・変更しました。 ■ 「*プロジェクトの新規作成*」の注意書きにカスタマーサクセスライセンス向けのアドバンスエディションに関する説明を追加 ■ 「*プロジェクトの新規作成*」のコラムにカスタマーサクセスライセンス向けのアドバンスエディションに関する説明を追加
2024-01-31	第45版 下記を追加・変更しました。 ■ 「*モジュール別の設定*」に「*データベース出力用ログ情報設定*」を追加 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」にデータベース出力用ログ情報に関する説明を追加 ■ 「*テナント環境セットアップ後の各種メンテナンス（アップデートによるメンテナンス）*」にデータベース出力用ログ情報に関する説明を追加

変更年月日	変更内容
2024-04-01	第46版 下記を追加・変更しました。 ■ 「*任意のユーザでログインを行うための設定*」に任意ユーザでログインを行う際の注意事項を追加 ■ 「*Microsoft SQL Server*」の READ_COMMITTED_SNAPSHOT に関する注意事項を変更 ■ 「*ベースURL*」と「*Apache HTTP Server*」にベースURLの設定に関する注意事項を追加 ■ 「*ベースURLが未設定の場合に警告ログを出力します。* 」の説明を追加 ■ 「*IM-Notice*」の「*FCMの設定*」および「*IM-Notice Mobile設定ファイルの編集*」および「*Amazon SNS の設定*」の説明を変更 ■ 「*IM-Notice*」に「*サービスアカウントの認証情報ファイルの更新*」および「*FCM HTTP v1 API への移行*」を追加 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」に「*2023 Autumn(Hollyhock) からアップデート*」を追加 ■ 「*テナント環境セットアップ後の各種メンテナンス（アップデートによるメンテナンス）*」に「*2023 Autumn(Hollyhock) からアップデート*」を追加 ■ 「*テナント環境セットアップ後の各種メンテナンス（アップデートによるメンテナンス）*」からデータベース出力用ログ情報に関する説明を削除 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」にデータベース出力用ログ情報に関する説明を追加 ■ 「*データベース出力用ログ情報設定*」の説明を追加 ■ 名称変更のため、Office 365 を Microsoft 365、 Azure Active Directory（AzureAD） を Microsoft Entra ID に修正 ■ 「*Accel Platform Mobile*」の「*FCMの設定*」および「*Accel Platform Mobile 設定ファイルの編集*」の説明を変更 ■ 「*Accel Platform Mobile*」に「*サービスアカウントの認証情報ファイルの更新*」および「*FCM HTTP v1 API への移行*」を追加 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」の「*2023 Autumn(Hollyhock) からアップデート*」に Accel Platform Mobile に関しての記述を追加
2024-08-30	第47版 下記を追加・変更しました。 ■ 「*プロジェクトの新規作成*」の注意書きにASEAN地域向けのカスタマーサクセスライセンスに関する説明を追加 ■ 「*プロジェクトの新規作成*」のコラムにASEAN地域向けのカスタマーサクセスライセンスに関する説明を追加 ■ 「*ライセンスの登録*」の「*カスタマーサクセスライセンスの場合*」にASEAN地域向けのカスタマーサクセスライセンスに関する注意書きを追加 ■ 「*Resinの設定*」のコラムにASEAN地域向けのカスタマーサクセスライセンスに関する説明を追加

変更年月日	変更内容
2024-10-01	第48版 下記を追加・変更しました。 ■ 「*intra-mart Accel Platform のヘルスチェック*」のコラムに intra-mart Accel Platform 各種サービスの起動確認を行わないパターンのURLを追加 ■ 「*テナント環境セットアップ後の各種メンテナンス（アップデートによるメンテナンス）*」に「*2024 Spring(Iris) からアップデート*」を追加 ■ 「*Resinの設定*」のタイムゾーンの設定に関する注意事項を変更 ■ 「*Resinの設定*」のヒープの最大サイズに関する注意事項を変更 ■ 「*テナント環境セットアップ*」に「*ベクトルデータベース接続情報*」を追加 ■ 「*PostgreSQL*」に pgvector のインストール手順を追加 ■ 「*IM-ContentsSearch*」に「*intra-mart Accel Platform 2024 Autumn(Jasmine)からの変更点*」を追加 ■ 「*IM-ContentsSearch*」に「*intra-mart Accel Platform 2024 Autumn(Jasmine)からの変更点*」を追加 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」に「*2024 Spring(Iris) からアップデート*」を追加 ■ 「*IM-Copilot*」を追加 ■ 「*ベースURL*」に置換文字列「IM_BaseURL」に関する注意事項を追加 ■ 「*IM-Notice*」の英語版ドキュメントにおいて、英語版向けの画像を表示するように変更しました。 ■ 「*Accel Platform Mobile*」の英語版ドキュメントにおいて、英語版向けの画像を表示するように変更しました。
2025-04-01	第49版 下記を追加・変更しました。 ■ 「*テナント環境セットアップ後の各種メンテナンス（アップデートによるメンテナンス）*」に「*2024 Autumn(Jasmine) からアップデート*」を追加 ■ 「*アップデート時に IM-Juggling で必要なメンテナンス作業*」に「*2024 Autumn(Jasmine) からアップデート*」を追加 ■ 「*ベクトルデータベース接続情報*」にベクトルデータベース接続情報の設定が必要な状況に関する説明を追加 ■ 「*ベクトルデータベース接続情報*」にシェアードデータベースの使用に関する説明および注釈を追加 ■ 「*ベクトルデータベース接続情報*」にベクトルデータベース種別の選択候補に pgvector が表示されない場合の確認事項に関する説明を追加

はじめに

本書の目的

本書では intra-mart Accel Platform のセットアップ手順について説明します。

前提条件

- リリースノートに記載されているシステム要件を満たしている必要があります。
  詳細は「リリースノート 」を参照してください。
- 本書は Resin に関するセットアップ内容です。

> **コラム**
>
> WebSphere Application Server 9.0.5 については、「セットアップガイド for WebSphere 」を参照してください。
> Oracle WebLogic Server 12c R2(12.2.1) については、「セットアップガイド for WebLogic 」を参照してください。

- intra-mart Accel Platform で利用するポート番号については、「*付録*」 -「*ポート一覧*」を参照してください。

対象読者

本書は、運用環境を想定したセットアップについて説明しています。
intra-mart Accel Platform を体験されたい方、はじめて intra-mart Accel Platform をセットアップされる方は、 intra-mart Accel Platform の最小構成（スタンドアローン）での簡易的なセットアップについて説明している
「クイック セットアップガイド 」を参照してください。

用語解説

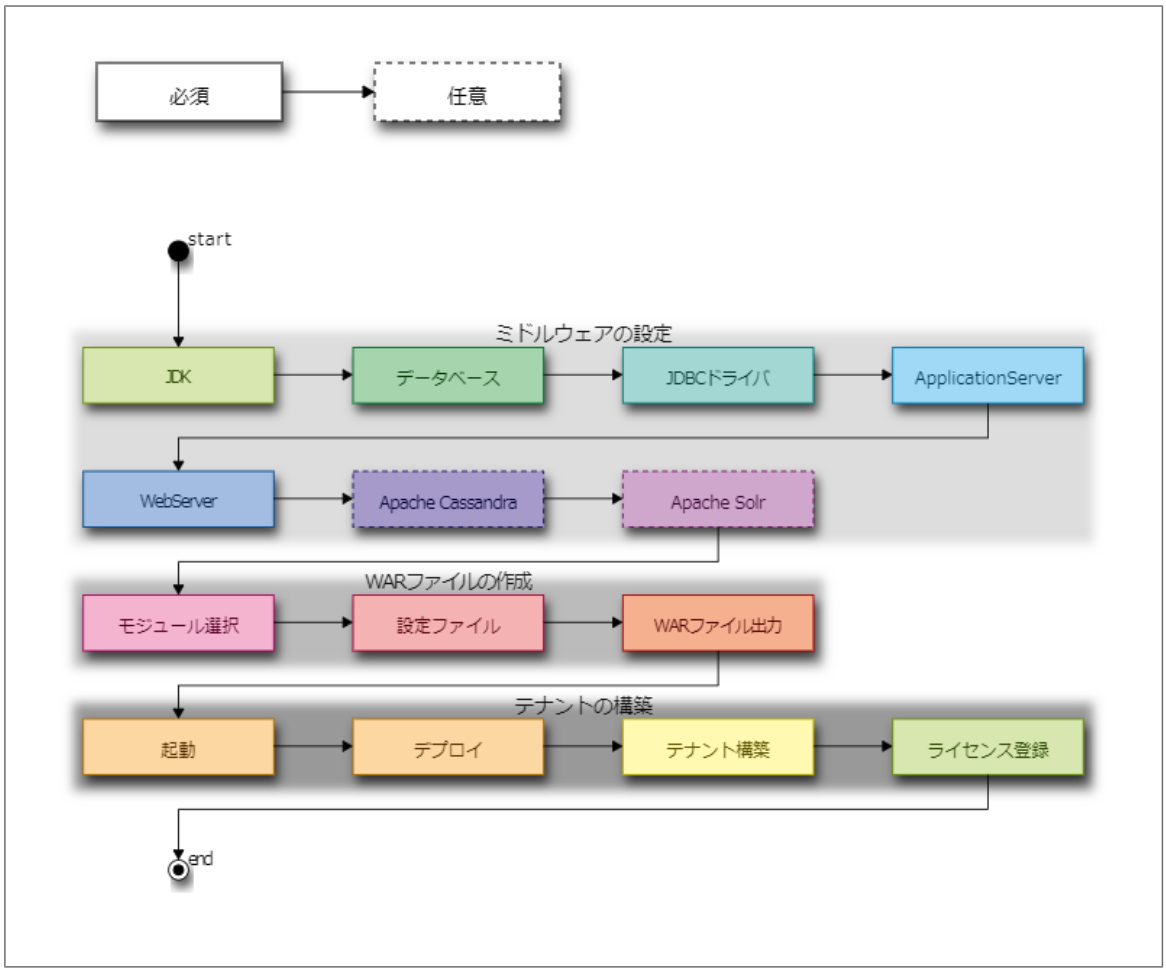
Resin をインストールしたディレクトリを %RESIN_HOME% と略します。
Apache HTTP Server をインストールしたディレクトリを %APACHE_HOME% と略します。
Storage として使用するディレクトリを %STORAGE_PATH% と略します。
PublicStorage として使用するディレクトリを %PUBLIC_STORAGE_PATH% と略します。
Webサーバ利用時の静的コンテンツを配置するディレクトリを %WEB_PATH% と略します。

セットアップの流れ

セットアップの流れは次の通りです。
各ステップごとのセットアップ手順は一覧のリンク先を参照してください。

- 凡例

  必須・・・セットアップが必要な項目です。
  任意・・・セットアップをスキップする事ができる項目です。



Name	Description
start	
JDK	*JDK*
データベース	*データベースサーバ*
JDBCドライバ	*JDBC ドライバ*
ApplicationServer	*Web Application Server*
WebServer	*Web Server*
Apache Cassandra	*Apache Cassandra*
Apache Solr	*Apache Solr*
モジュール選択	*WAR ファイルの作成* - *プロジェクトの新規作成*
設定ファイル	*WAR ファイルの作成* - *intra-mart Accel Platform の設定ファイル*
WARファイル出力	*WAR ファイルの作成* - *WAR ファイルの出力*
起動	*Web Application Server の起動・停止*
デプロイ	*デプロイ*
テナント構築	*テナント環境セットアップ*
ライセンス登録	*ライセンスの登録*

Name	Description
end	

ミドルウェアのセットアップ

intra-mart Accel Platform のセットアップに必要な以下のインストールおよび設定を行います。

JDK

intra-mart Accel Platform を利用するには、JDK が必要です。
Resin の場合、リリースノートに記載されているシステム要件を満たす JDK を入手し、セットアップを行います。
詳細は「 リリースノート 」-「 システム要件 」を参照してください。
Resin 以外の Web Application Server の場合、各製品のドキュメントを参照してください。

Oracle Java（JDK）

Oracle Java（JDK）を入手する場合は以下から入手できます。
https://www.oracle.com/java/technologies/downloads/ (English)

データベース

データベース のセットアップに必要な以下のインストールおよび設定を行います。

データベースサーバ

項目
- データベース のインストール
  - 各 データベース 作成時の注意点
    - Oracle Database
    - Microsoft SQL Server
  - PostgreSQL
    - PostgreSQL のインストール
    - pgvector のインストール
    - データベースとログインロールの作成

データベース のインストール

- intra-mart Accel Platform は、データベースが必要です。
  ご利用になる データベース をインストールしてください。インストールの詳細は各製品元のドキュメントを参照してください。
  本書では、PostgreSQL （Windows版）を一例として説明します。

> **注意**
> データベース のエンコーディングは、*UTF-8* で作成する必要があります。

> **注意**
> データベース に対してデータの登録・更新・削除を行う権限に加え、テーブルやVIEWを作成する権限も必要です。
> 具体的な設定方法については、各データベースベンダから提供されているドキュメントを参照してください。

> **コラム**
> WARファイルによる複数テナント 、 バーチャルテナントによる複数テナント を構築する場合、接続先のデータベースはインスタンス単位で分ける事を推奨します。

各 データベース 作成時の注意点

Oracle Database

- intra-mart Accel Platform を利用するには、最低限「CONNECT」・「RESOURCE」・「CREATE VIEW」・「UNLIMITED TABLESPACE」権限が
  付与されている必要があります。
  その他利用するアプリケーション、エクステンションシリーズによって必要な権限があります。
  詳細については、各製品のリリースノートおよび、セットアップガイドを参照してください。

> **注意**
>
> IM-Workflow を利用する場合、テナントデータベースとして接続するユーザのデフォルト表領域は、自動セグメント領域管理(ASSM)を有効にする必要があります。
> IM-Workflow モジュールでは、 2013 Summer(Damask) にて トランザクションデータの保存先としてデータベースを選択できるようになりました。
> この対応で、テナント環境セットアップ時にBLOB データ型カラムを持つテーブルを作成しています。
> 対象の BLOB データ型カラムには、パフォーマンスの改善を目的として、SECUREFILE パラメータを指定しています。
> この場合、対象のカラムは自動セグメント領域管理(ASSM)で管理される表領域でのみ作成できます。

Microsoft SQL Server

- 作成するデータベースの既定の照合順序は、 データベース をインストールしたプラットフォーム（OS）の
  コントロールパネル- [地域と言語のオプション] - [詳細設定] タブで Windows システム ロケール名を確認します。
  既存の Windowsロケールでの照合順序設定に対応する照合順序指定子を指定します。
  各Windows 照合順序は大文字小文字、アクセント、文字幅、かなの区別を定義する一連のサフィックスとして組み合わせることが可能です。
  運用される環境において、サフィックスの組み合わせを設定する必要があります。

> **コラム**
>
> intra-mart Accel Platform では、大文字・小文字を区別する設定でデータベースを作成することを推奨します。
>
> - SQL Server 2008 R2 で推奨する設定値
>   - Japanese_90_CS_AS_KS_WS
> - SQL Server 2012 以降のバージョンで推奨する設定値
>   - Japanese_XJIS_100_CS_AS_KS_WS

> **注意**
>
> intra-mart Accel Platform では、「READ_COMMITTED_SNAPSHOT」をONにする必要があります。
> 「READ_COMMITTED_SNAPSHOT」をONにしない場合、「テーブル スキャンによるロック待ち」 が発生する可能性があります。
> ※「テーブル スキャンによるロック待ち」および 「READ_COMMITTED_SNAPSHOT」について、詳しくは Microsoft SQL Server のドキュメントを参照してください。
>
> [READ_COMMITTED_SNAPSHOT の利用方法]
> 利用するには、以下のようにデータベースに対して READ_COMMITTED_SNAPSHOT を ON に設定する必要があります。
>
> - ALTER DATABASE {data_base_name} SET READ_COMMITTED_SNAPSHOT ON;
>
> ※ データベースへの接続コネクションがすべて終了している状態で実行してください。

PostgreSQL

PostgreSQL のインストール

- PostgreSQL は次のURLよりダウンロードできます。
  環境に適したものをダウンロード後、インストールを行ってください。
  https://www.enterprisedb.com/downloads/postgres-postgresql-downloads

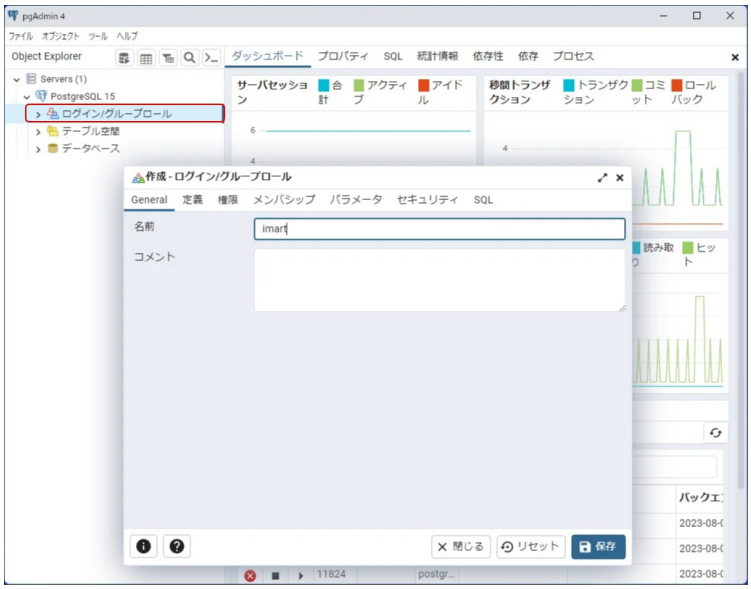
pgvector のインストール

- intra-mart Accel Platform では、IM-Copilot for Accel Platform が提供する生成AI活用基盤を使用して PostgreSQL をベクトルデータベースとして利用できます。
  PostgreSQL をベクトルデータベースとして利用する場合は、以下のサイトを参考に pgvector のインストールを行ってください。
  https://github.com/pgvector/pgvector

> **コラム**
>
> ベクトルデータベースは 2024 Autumn(Jasmine) から利用可能です。

データベースとログインロールの作成

1. pgAdmin ツールを起動します。
2. ログインロールの作成
   「Object Explorer」の「ログイン/グループロール」を右クリック、「作成」→「ログイン/グループロール」をクリックします。
   サブウィンドウが表示されます。



次の項目を入力し「保存」をクリックします。

Generalタブ
　「名前」（任意）

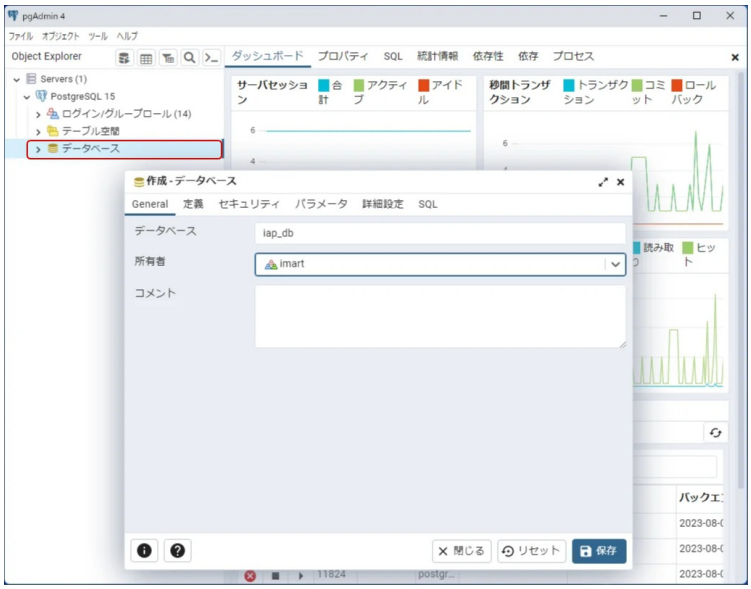
定義タブ
　「パスワード」（任意）

> **コラム**
>
> 本書では、例として次の内容を指定します。
>
> 名前「imart」
>
> パスワード「imart」

3. データベースの作成

「Object Explorer」の「データベース」を右クリック、「作成」→「データベース」をクリックします。

サブウィンドウが表示されます。



次の項目を入力、選択し「保存」をクリックします。

Generalタブ
　[データベース]（任意）
　[所有者]（上記「2. ログインロールの作成」で作成したログインロール）

コラム

本書では、例として次の内容を指定します。

データベース「iap_db」

所有者「imart」

コラム

作成したデータベースをベクトルデータベースとして利用する場合は、拡張機能を有効化する必要があります。
以下の SQL を作成したデータベースで実行して、拡張機能を有効にしてください。

```
CREATE EXTENSION vector;
```

JDBCドライバ

項目

- JDBCドライバの入手
  - PostgreSQL
  - Oracle Database
  - Microsoft SQL Server
  - SAP HANA

JDBCドライバの入手

以下の条件に該当するJDBCドライバを入手します。

- 利用するデータベースがサポートしている最新JDBCドライバ
- **JDBC 4.2** または **JDBC 4.3** のJDBCドライバ

コラム

本ページの外部URLは、2019-12-01時点のものです。
なお、掲載されているURLは変更になる場合があります。
その場合は条件に合ったJDBCドライバを検索して入手してください。

注意

Resinは、**JDBC 4.2** または **JDBC 4.3** を利用する場合4.0.56以上が必要です。

PostgreSQL

1. PostgreSQL のJDBCドライバは以下のURLよりダウンロードできます。
   https://jdbc.postgresql.org/download （English）
2. 利用するデータベースがサポートしている JDBC 4.2 の最新ファイルを入手してください。

Oracle Database

1. Oracle のJDBCドライバは以下のURLよりダウンロードできます。
   https://www.oracle.com/database/technologies/appdev/jdbc-downloads.html （English）
2. 利用するデータベースがサポートしている JDBC 4.2 または JDBC 4.3 の各バージョンの最新ファイルを入手してください。
   Oracle Database 12c Release 2 の場合は、Oracle Database 12c Release 2 Build の最新
   Oracle Database 18c の場合は、Oracle Database 18c Build の最新
   Oracle Database 19c の場合は、Oracle Database 19c Build の最新

   JDKのバージョンがJDK 8 の場合は、JDBC 4.2 のJDBCドライバを使用してください。
   JDKのバージョンがJDK 11 の場合は、JDBC 4.3 のJDBCドライバを使用してください。

注意

Oracle Database 12c Release 2 および Oracle Database 18c は JDBC 4.2 のみ対象です。

Microsoft SQL Server

1. Microsoft SQL Server の JDBCドライバは以下のURLよりダウンロードできます。
   https://docs.microsoft.com/ja-jp/sql/connect/jdbc/download-microsoft-jdbc-driver-for-sql-server （日本語）
   https://docs.microsoft.com/en-us/sql/connect/jdbc/download-microsoft-jdbc-driver-for-sql-server （English）
   https://docs.microsoft.com/zh-CN/sql/connect/jdbc/download-microsoft-jdbc-driver-for-sql-server （中文）

2. 利用するデータベースがサポートしている JDBC 4.2 または JDBC 4.3 の各バージョンの最新ファイルを入手してください。
   JDKのバージョンがJDK 8 の場合は、JDBC 4.2 のJDBCドライバを使用してください。
   JDKのバージョンがJDK 11 の場合は、JDBC 4.3 のJDBCドライバを使用してください。

SAP HANA

1. SAP HANA のJDBCドライバをインストールしてください。
   詳しくは、SAP HANA のドキュメントを参照してください。

2. <ngdbc.jar> ファイルを入手します。

> **コラム**
> 弊社ではデータベース接続に利用するJDBCドライバは、Type4（非XA）にて検証を実施しています。

Web Application Server

Resin （Windows版）のインストール

項目
- インストール
- JDBCドライバの配置

インストール

「プロダクトファイルダウンロード 」サイトより最新のResin<resin-pro-4.0.xx.zip>を入手して任意のパスに展開します。

> **注意**
> Resin を起動するためには 「.NET Framework 3.5」 が必要です。
> Windows Server 2012 環境では、「.NET Framework 4.5」のみインストールされているため、「.NET Framework 3.5」 のセットアップを行ってください。
> 詳細は、*.NET Framework のセットアップ*を参照してください。

> **注意**
> intra-mart Accel Platformのアップデートを行った場合においても、最新のResinを利用してください。

JDBCドライバの配置

1. JDBCドライバをダウンロードします。

   JDBCドライバの入手先は以下を参照してください。
   JDBC ドライバ

2. ダウンロードした、JDBCドライバを以下のディレクトリにコピーします。

```
%RESIN_HOME%/lib
```

Resin （Linux版）のインストール

項目
- インストール
- フォントの設定
- JDBCドライバの配置

インストール

1. 「プロダクトファイルダウンロード」サイトより最新のResin<resin-pro-4.0.xx.tar.gz>を入手して任意のパスに配置します。
2. ファイルを展開します。

```
# tar xzvf resin-pro-4.0.xx.tar.gz
```

3. Resin のソースを展開したディレクトリから、以下を実行します。

```
# ./configure --prefix=<%RESIN_HOME%>
# make
# make install
```

> **注意**
> ディストリビューションによってコンパイルオプションを指定する必要があります。
> 具体的にどのオプションが必要になるかは、ディストリビューションによって異なります。
> ./configure コマンドの実行結果を確認し、必要なオプションを指定してください。
>
> 64bit環境上で Resin のコンパイルを行う際、 ./configureを呼び出す際に以下のオプションがLinux環境によって必要です。
>
> ```
> # ./configure --prefix=<%RESIN_HOME%> --enable-64bit
> ```

> **注意**
> intra-mart Accel Platformのアップデートを行った場合においても、最新のResinを利用してください。

> **コラム**
> make installを実行する際の実行ユーザがroot権限を持っている場合、/etc/init.dディレクトリ配下に起動用スクリプトresinが作成されます。

フォントの設定

intra-mart Accel Platform の一部の機能では、アプリケーションサーバで画像を生成します。
画像に文字が含まれる場合、OS にインストールされているフォントを利用して画像を生成します。
OS にフォントがインストールされていない場合、画像内の文字が正しく表示されません。
画像中の文字を正常に表示するために、アプリケーションサーバを起動する OS にフォントをインストールしてください。
複数の言語を扱う運用を行う場合は、それぞれの言語の文字が含まれたフォントをインストールしてください。

> **コラム**
> ベースモジュールにおいて、アプリケーションサーバで画像を生成する機能は以下のとおりです。
>
> - グラフ描画モジュール - 「chart」タグ利用時
> - ViewCreator - グラフ表示（グラフ描画形式として「JFreeChart」を利用している場合）
> - IM-Workflow - フロー図の画像出力（ 2019 Summer(Waltz) 以前の場合）

JDBCドライバの配置

1. JDBCドライバをダウンロードします。

   JDBCドライバの入手先は以下を参照してください。
   JDBC ドライバ

2. ダウンロードした、JDBCドライバを以下のディレクトリにコピーします。

```
%RESIN_HOME%/lib
```

Resinの設定

項目

- スタンドアローン／分散構成共通
- 分散構成
- TERASOLUNA Server Framework for Java (5.x) for Accel Platform を使用する場合の設定

スタンドアローン／分散構成共通

1. <%RESIN_HOME%/conf/resin.properties> ファイルを開きます。
2. 「jvm_args」プロパティに、インストール環境に応じたメモリ値、ヒープの最大サイズを設定します。
   本書では、例として次の値を設定します。

```
# Arg passed directly to the JVM
jvm_args  : -Xmx4096m -Dfile.encoding=UTF-8 --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/sun.util.locale=ALL-UNNAMED --add-opens=java.xml/com.sun.org.apache.xerces.internal.jaxp=ALL-UNNAMED --add-opens=java.base/sun.util.calendar=ALL-UNNAMED --add-opens=java.xml/com.sun.org.apache.xerces.internal.jaxp.datatype=ALL-UNNAMED --add-opens=java.base/java.text=ALL-UNNAMED --add-opens=java.desktop/java.awt.font=ALL-UNNAMED --add-opens=java.base/java.math=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAMED
```

> **注意**
>
> 設定する値は「4096m」以上にしてください。
> 「4096m」より小さい値を設定した場合、正常に動作しない恐れがあります。

> **注意**
>
> Java 8、または Java 11 を利用する場合は以下のコメントアウトを外して利用してください。
> （メモリ値、ヒープの最大サイズはインストール環境に応じた値に読み替えてください。）
>
> ```
> # When using Java 8 or 11
> # jvm_args : -Dfile.encoding=UTF-8 -Djava.io.tmpdir=tmp -Xmx4096m -Xms4096m -XX:+UseG1GC -XX:MaxGCPauseMillis=200 -XX:InitiatingHeapOccupancyPercent=30 -XX:-OmitStackTraceInFastThrow -XX:+HeapDumpOnOutOfMemoryError
> ```
>
> その際、以下はコメントアウトしてください。
>
> ```
> jvm_args : -Dfile.encoding=UTF-8 -Djava.io.tmpdir=tmp -Xmx4096m -Xms4096m -XX:+UseG1GC -XX:MaxGCPauseMillis=200 -XX:InitiatingHeapOccupancyPercent=30 -XX:-OmitStackTraceInFastThrow -XX:+HeapDumpOnOutOfMemoryError --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/sun.util.locale=ALL-UNNAMED --add-opens=java.xml/com.sun.org.apache.xerces.internal.jaxp=ALL-UNNAMED --add-opens=java.base/sun.util.calendar=ALL-UNNAMED --add-opens=java.xml/com.sun.org.apache.xerces.internal.jaxp.datatype=ALL-UNNAMED --add-opens=java.base/java.text=ALL-UNNAMED --add-opens=java.desktop/java.awt.font=ALL-UNNAMED --add-opens=java.base/java.math=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAMED
> ```

> **コラム**
>
> 「jvm_args」プロパティに「-Djava.io.tmpdir」オプションを追加し、 Resin が利用する作業ディレクトリを変更できます。
> このオプションが指定されていない場合、 Resin が利用する作業ディレクトリはJVMのデフォルトの設定が利用されます。
> 以下の例では、作業ディレクトリを /var/resin-tmp に変更する場合の設定例です。
>
> ```
> # Arg passed directly to the JVM
> jvm_args  : -Xmx4096m -Dfile.encoding=UTF-8 -Djava.io.tmpdir=/var/resin-tmp
> ```
>
> > **注意**
> >
> > ・「-Djava.io.tmpdir」オプションにより指定されたディレクトリは事前に作成しておく必要があります。
> > ・Resin 実行ユーザが読み込み、書き込みを行うことができる権限を設定しておく必要があります。
> > ・Linux系の環境では、このオプションが未指定の場合 /tmp が利用されます。
> > cron等の設定により定期的に/tmp配下の内容が削除される設定が標準で組み込まれている場合があります。
> > ・Resin では、作業用ディレクトリに展開した設定ファイル等の変更を検出した場合、自動的に再起動が行われる場合があります。
> > その為、 Resin が利用する作業ディレクトリを変更しておくことを推奨します。

コラム

「jvm_args」プロパティに「-Duser.timezone」オプションを追加し、システムタイムゾーンを変更できます。
例えば、”UTC” を設定する場合は、以下のように指定します。

```
# Arg passed directly to the JVM
jvm_args  : -Xmx4096m -Dfile.encoding=UTF-8 -Duser.timezone=UTC
```

システムタイムゾーンの説明については、「intra-mart Accel Platform タイムゾーン仕様書」-「3 種類のタイムゾーンを設定可能」を参照してください。

注意

intra-mart Accel Platform を様々な国のユーザが利用する場合には、以下の点を考慮してシステムタイムゾーンを適切に設定する必要があります。

- 夏時間がないタイムゾーンを指定する必要があります。
- Etc/GMT+1 等の + という文字を含むタイムゾーンを指定できません。
- 運用開始後にシステムタイムゾーンを変更できません。
- タイムゾーンマスタに定義されているタイムゾーンを設定してください。
- ログには、システムタイムゾーンの日時が出力されます。
- データベースには、システムタイムゾーンの日時データが保存されます。

詳しくは、「intra-mart Accel Platform タイムゾーン仕様書」-「環境」を参照してください。

注意

タイムゾーン設定の初期設定は、インストールを行った JDK に依存します。

コラム

Resin は、起動後に設定ファイルやプログラムの変更を検出した場合、 Resin 自身の再起動を行います。
開発中、運用中等により用途は異なりますが、Resin に変更の検出を行わせたくない場合は、「dependency_check_interval」項目の設定を行います。
下記は、変更の検出を行わせない設定例です。

```
  dependency_check_interval : -1

「dependency_check_interval」は、2s （2秒毎）, 5m （5分毎）等の値が設定可能です。
```

変更が完了したら、Resin を再起動してください。

コラム

Resin の起動時にポートが重複している旨のエラーが発生する場合、Resin の停止後、ポート番号の設定を変更してください。

```
 java.lang.RuntimeException: java.net.BindException: Address already in use: JVM_Bind
        Can't bind to *:8080.
        Check for another server listening to that port.

「app.http」プロパティに設定されている「8080」を、「8081」などの使用されていないポート番号に変更してください。
```

```
# Set HTTP and HTTPS ports.
# Use overrides for individual server control, for example: app-0.http : 8081
app.http        : 8081
```

変更が完了したら、Resin を再起動してください。

注意

intra-mart Accel Platform ではHTTPSによるResinへの接続をサポートしていません。*.https プロパティを有効にしないでください。
セキュアな通信を行う場合は、*Web Server* を介してResinへ接続を行ってください。

コラム
その他の設定については、「設定ファイルリファレンス 」 - 「Resin 」を参照してください。

コラム
カスタマーサクセスライセンスをご契約中の場合には、ライセンスポータルと通信してご契約内容の変更が自動反映されます。
Resin をインストールした環境からライセンスポータルへの通信にプロキシサーバを利用する場合は、次の JVM 引数の設定が必要です。

-Dhttps.proxyHost	プロキシサーバのホストURL
-Dhttps.proxyPort	プロキシサーバのポート番号
-Dhttps.proxyUser	プロキシサーバへの接続ユーザ
-Dhttps.proxyPassword	接続ユーザのパスワード

ただし、ASEAN地域向けのカスタマーサクセスライセンスに関しては、自動反映する機能は提供されていません。
カスタマーサクセスライセンスについての詳細は、「*ライセンスの登録*」を参照してください。

分散構成

1. <%RESIN_HOME%/conf/resin.properties> ファイルを開きます。
2. 「app_servers」プロパティに分散構成として構築したResinサーバの情報を設定します。
   例としてResinを3台構成で運用する場合、次の値を設定します。

```
# app-tier Triad servers: app-0 app-1 app-2
app_servers     : 192.168.100.1:6800 192.168.100.2:6800 192.168.100.3:6800
```

「app_servers」プロパティに設定した「IPアドレス:ポート番号」は、記述した順に app-0, app-1, app-2 とIDが振られます。

コラム
- 「app_servers」プロパティにはホスト名を指定することも可能です。
- ホスト名に紐づくIPアドレスが変わった場合は Resin サーバを再起動してください。

```
# app-tier Triad servers: app-0 app-1 app-2
app_servers     : app0-host:6800 app1-host:6800 app2-host:6800
```

注意
分散構成に含まれるResinの<%RESIN_HOME%/conf/resin.properties>ファイルは、すべて同じ内容にしてください。
「app_servers」プロパティに指定するIPアドレスは、ローカルアドレスのみ指定が可能であり、グローバルアドレスの指定を行うことはできません。
また、分散環境を構築する場合、127.0.0.1等のループバックアドレスが指定された場合正常に動作しません。

TERASOLUNA Server Framework for Java (5.x) for Accel Platform を使用する場合の設定

- Resin 4.0.49 以降の場合
  1. <%RESIN_HOME%/lib>から以下のjarファイルを退避します。
     - validation-api-1.0.0.GA.jar
  2. <%RESIN_HOME%/webapp-jars>から以下のjarファイルを退避します。
     - hibernate-validator-4.3.0.Final.jar
- Resin 4.0.48 以前の場合
  1. <%RESIN_HOME%/lib>から以下のjarファイルを退避します。
     - hibernate-validator-4.3.0.Final.jar
     - validation-api-1.0.0.GA.jar

  コラム
  TERASOLUNA Server Framework for Java (5.x)では、Hibernate Validatorのバージョンが Resinが提供するものと異なるため、競合するバージョンのjarを退避します。

注意

モジュール構成の変更などにより、TERASOLUNA Server Framework for Java (5.x) for Accel Platform の利用をやめる場合には、必ず退避していたjarファイルを元に戻してください。

- Resin 4.0.49 以降の場合
  1. <%RESIN_HOME%/lib>に以下のjarファイルを配置します。
     - validation-api-1.0.0.GA.jar
  2. <%RESIN_HOME%/webapp-jars>に以下のjarファイルを配置します。
     - hibernate-validator-4.3.0.Final.jar
- Resin 4.0.48 以前の場合
  1. <%RESIN_HOME%/lib>に以下のjarファイルを配置します。
     - hibernate-validator-4.3.0.Final.jar
     - validation-api-1.0.0.GA.jar

コラム

Resin のクラスタリングについては、「*Resin をクラスタリングしての分散環境*」を参照してください。

注意

Resin の分散環境を構築する場合には製品版に同梱されている Resin が必要です。
検証目的などで分散環境を構築されたい場合は、弊社営業までお問い合わせください。

Web Server

intra-mart Accel Platform は、 Web Server を利用します。
ご利用になる Web Server をインストールしてください。インストールの詳細は各 Web Server のドキュメントを参照してください。

Apache HTTP Server

項目

- 前提条件
- mod_proxy/mod_rewriteの利用
- mod_proxy/mod_rewriteの取得
- apache設定ファイルの編集
- アクセスログの編集
- 制限事項

前提条件

- ベースURLの設定
  Apache を使用する場合ベースURLの設定が必要です。

  コラム
  - ベースURLには *<httpまたはhttps>://<ApacheのIPアドレスまたはドメイン名>:<Apacheのポート番号>/<コンテキストパス>* を設定します。
    - 構築するサーバ環境に応じて適切に設定してください。
  - ベースURLについての詳細は、*ベースURL*を参照してください。

mod_proxy/mod_rewriteの利用

ここではResinの組み込みモジュールを利用せずApacheのモジュール（mod_proxy/mod_rewrite）を利用して、webサーバとintra-mart Accel Platformの連携を行う場合の設定を行います。

mod_proxy/mod_rewriteの取得

製品元のマニュアルを参照してください。

apache設定ファイルの編集

1. <%APACHE_HOME%/conf/httpd.conf> ファイルを開きます。

2. Dynamic Shared Object (DSO) Supportエリアに以下の設定を追加またはコメントアウトを外してください。

```
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule proxy_wstunnel_module modules/mod_proxy_wstunnel.so
```

3. 末尾にAllowEncodedSlashesディレクティブ、mod_proxyおよびmod_rewriteの設定を追加してください。

コンテキストパス	imart
apacheのドキュメントルート	/usr/local/apache/htdocs
静的ファイルの展開フォルダ	/usr/local/apache/htdocs/imart
Web Application ServerのIPアドレス	192.168.1.1
Web Application Serverのポート番号	8080

上記の例である場合の設定は以下の通りです。

```
SetEnv proxy-nokeepalive 1
AllowEncodedSlashes On
RewriteEngine On
RewriteCond %{REQUEST_URI} !^/imart/reverse_proxy/
RewriteCond %{REQUEST_URI} ^/imart/(.*\.gif|.*\.GIF)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.png|.*\.PNG)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.jpg|.*\.JPG)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.jpeg|.*\.JPEG)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.css|.*\.CSS)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.js|.*\.JS)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.swf|.*\.SWF)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.ico|.*\.ICO)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.svg|.*\.SVG)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.json|.*\.JSON)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.jar|.*\.JAR)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.xml|.*\.XML)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.yaml|.*\.YAML)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.txt|.*\.TXT)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.html|.*\.HTML|.*\.htm|.*\.HTM)$
RewriteRule ^/imart/(.*)?$ /imart/$1

RewriteCond %{HTTP:Connection} Upgrade [NC]
RewriteCond %{HTTP:Upgrade} websocket [NC]
RewriteRule ^/imart/(.*)?$ ws://192.168.1.1:8080/imart/$1 [P,L]

ProxyPreserveHost On
ProxyPass /imart/ http://192.168.1.1:8080/imart/ nocanon
ProxyPassReverse /imart/ http://192.168.1.1:8080/imart/ nocanon
ProxyTimeout 1200
```

> **コラム**
>
> 上記に指定した拡張子は、intra-mart Accel Platform Advanced版＋intra-mart Accel Collaboration＋IM-FormaDesigner for Accel Platformで静的ファイルとして扱う必要がある拡張子です。
> 上記RewriteCondにない静的ファイルを取り扱いたい場合には、RewriteCondの記述を追加してください。

> **コラム**
>
> 「ProxyTimeout」にはプロキシサーバ が Resin への接続を切断するまでのタイムアウト時間を設定します。
> テナント環境セットアップを完了するのに十分に大きい値を設定してください。
> ここでは例として 1200 秒を設定しています。

> **注意**
>
> APサーバのコンテキストルートを必ずリクエストに含むようにしてください。
> APサーバ側 *http://myapp/imart* であれば、 リクエストのURLは *http://myweb/imart* にしてください。

4. Apacheを再起動してください。

アクセスログの編集

リバースProxy の場合、Resinが出力するaccess.logのソースIPがすべて127.0.0.1になってしまいます。 これを回避するためにaccess.logの出力内容を編集します。

1. <%RESIN_HOME%/conf/resin.xml> ファイルを開きます。
2. <host id="" root-directory="."> ディレクティブ内に以下の設定を追加してResinの再起動を行ってください。

```
<access-log path='log/access.log'>
  <rollover-period>1D</rollover-period>
  <format>%{X-Forwarded-For}i %h %l %u %t "%r" %>s %b "%{Referer}i" "%{User-Agent}i"</format>
</access-log>
```

> **コラム**
> access-logについての詳細な設定内容については「Resin Documentation 」を参照してください。

制限事項

制限事項については「リリースノート 」を参照してください。

Internet Information Services（IIS）

項目

- 前提条件
  - URL Rewrite のセットアップ
  - Application Request Routing のセットアップ
- 設定
- アクセスログの編集

前提条件

- 「システム要件」に記載の Windows Server を使用していること。
  - 以降の手順では Windows Server 2022 のキャプチャ画像を使用しますが、上記のどのバージョンの Windows Server でも同じ手順でセットアップできます。
- ベースURLの設定
  Internet Information Services（IIS） を使用する場合ベースURLの設定が必要です。

  > **コラム**
  > - ベースURLには *<httpまたはhttps>://<IISのIPアドレスまたはドメイン名>:<IISのポート番号>/<コンテキストパス>* を設定します。
  >   - 構築するサーバ環境に応じて適切に設定してください。
  > - ベースURLについての詳細は、*ベースURL*を参照してください。

- Internet Information Services（IIS） に **WebSocket プロトコル をインストール** してください。

コラム

- 以下の手順で Internet Information Services（IIS） に WebSocket プロトコル をインストールできます。

  1. **サーバマネージャー を開きます。**
  2. **管理メニュー の「役割と機能の追加」をクリックします。**
  3. **「役割ベースまたは機能のベースのインストール」を選択し、「次へ」をクリックします。**
  4. **「サーバーの選択」にて対象のサーバーを選択し、「次へ」をクリックします。**
  5. **「サーバーの役割」の「役割」ツリーで「Webサーバー（IIS）」の配下にある「アプリケーション開発」を展開します。**
  6. **「WebSocket プロトコル」を選択し、「次へ」をクリックします。**
  7. **「機能」で、「次へ」をクリックします。**
  8. **「確認」で、「インストール」をクリックします。**

- Websocket プロトコルに関する情報は、以下のドキュメントを参考にしてください。
  - https://docs.microsoft.com/en-us/iis/get-started/whats-new-in-iis-8/iis-80-websocket-protocol-support（English）

- Internet Information Services（IIS） を稼働させるには「URL Rewrite」と「Application Request Routing」が必要です。
  下記のホームページより実行環境に合わせたインストーラをダウンロードし、以下の手順でセットアップを行ってください。

コラム

URL（2023年4月現在）

- URL Rewrite ダウンロードページ（English）
- Application Request Routing ダウンロードページ（English）

URL Rewrite のセットアップ

1. ダウンロードしたインストーラを実行します。
2. 「インストール」をクリックします。

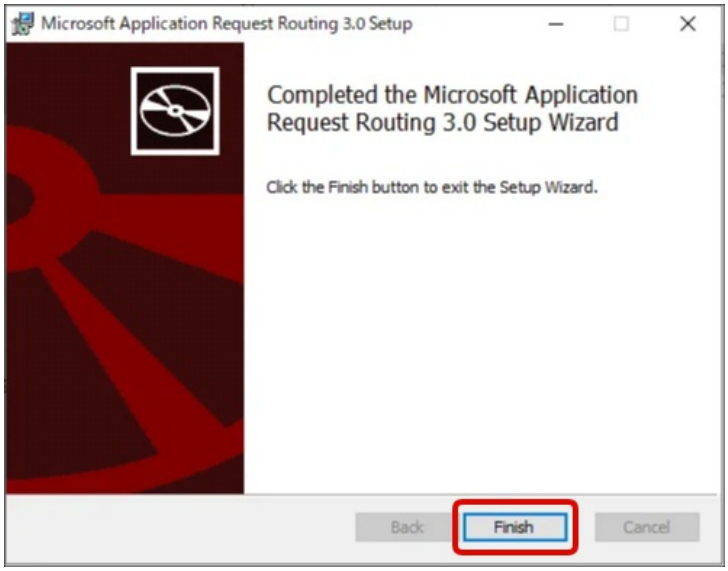
3. 「完了」をクリックします。

以上で URL Rewrite のセットアップは終了です。

Application Request Routing のセットアップ

1. ダウンロードしたインストーラを実行します。
2. 「Install」をクリックします。

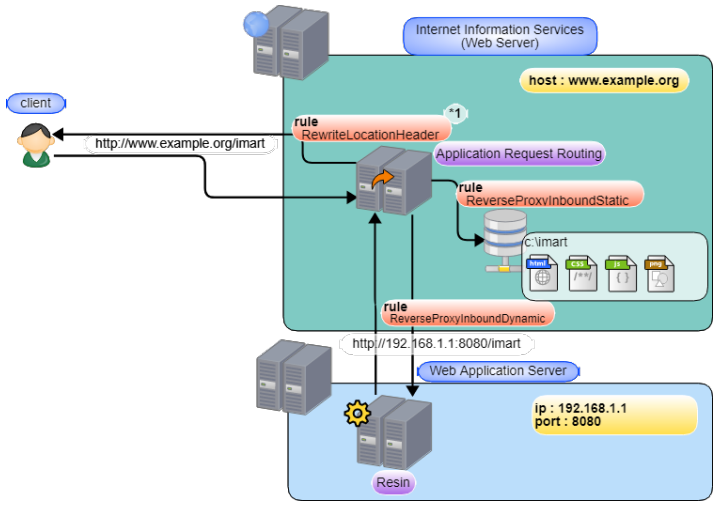
3. 「Finish」をクリックします。



以上で Application Request Routing のセットアップは終了です。

設定

- Resin へ接続するための設定を行います。
  以降の手順では下記の環境を想定した手順を記載します。



*1：RewriteLocationHeader はレスポンスヘッダのリダイレクト先の URL（Locationヘッダ）を書き換えます。

intra-mart Accel Platform のコンテキストパス	imart
IIS のルート ディレクトリ	C:\inetpub\wwwroot
静的ファイルの展開フォルダ	C:\imart
Web Application ServerのIPアドレス	192.168.1.1
Web Application Serverのポート番号	8080
ベースURL	*http://www.example.org/imart*

1. Internet Information Services（IIS） と Resin を接続するための設定ファイル <web.config> を <C:/inetpub/wwwroot> 直下に作成します。

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <system.web>
    <httpRuntime maxRequestLength="102400" requestPathInvalidCharacters="" />
    <customErrors mode="Off" />
  </system.web>
  <system.webServer>
    <security>
      <requestFiltering>
        <requestLimits maxAllowedContentLength="104857600" />
      </requestFiltering>
    </security>
    <httpErrors errorMode="Detailed" />
    <rewrite>
      <rules>
        <clear />
        <rule name="ReverseProxyInboundStatic" stopProcessing="true">
          <match url="^imart/(?!reverse_proxy/)
(.*\.gif|.*\.png|.*\.jpg|.*\.jpeg|.*\.css|.*\.js|.*\.swf|.*\.ico|.*\.svg|.*\.json|.*\.jar|.*\.xml|.*\.yaml|.*\.txt|.*\.html|.*\.htm)$" />
          <action type="None" />
        </rule>
        <rule name="ReverseProxyInboundDynamic" stopProcessing="true">
          <match url="^imart/(.*)" />
          <action type="Rewrite" url="http://192.168.1.1:8080/imart/{R:1}" />
        </rule>
      </rules>
      <outboundRules>
        <rule name="RewriteLocationHeader" preCondition="Redirect Response">
          <match serverVariable="RESPONSE_Location" pattern="^http://192.168.1.1:8080/imart/(.*)" />
          <action type="Rewrite" value="http://www.example.org/imart/{R:1}" />
        </rule>
        <preConditions>
          <preCondition name="Redirect Response">
            <add input="{RESPONSE_STATUS}" pattern="3\d\d" />
          </preCondition>
        </preConditions>
      </outboundRules>
    </rewrite>
  </system.webServer>
</configuration>
```

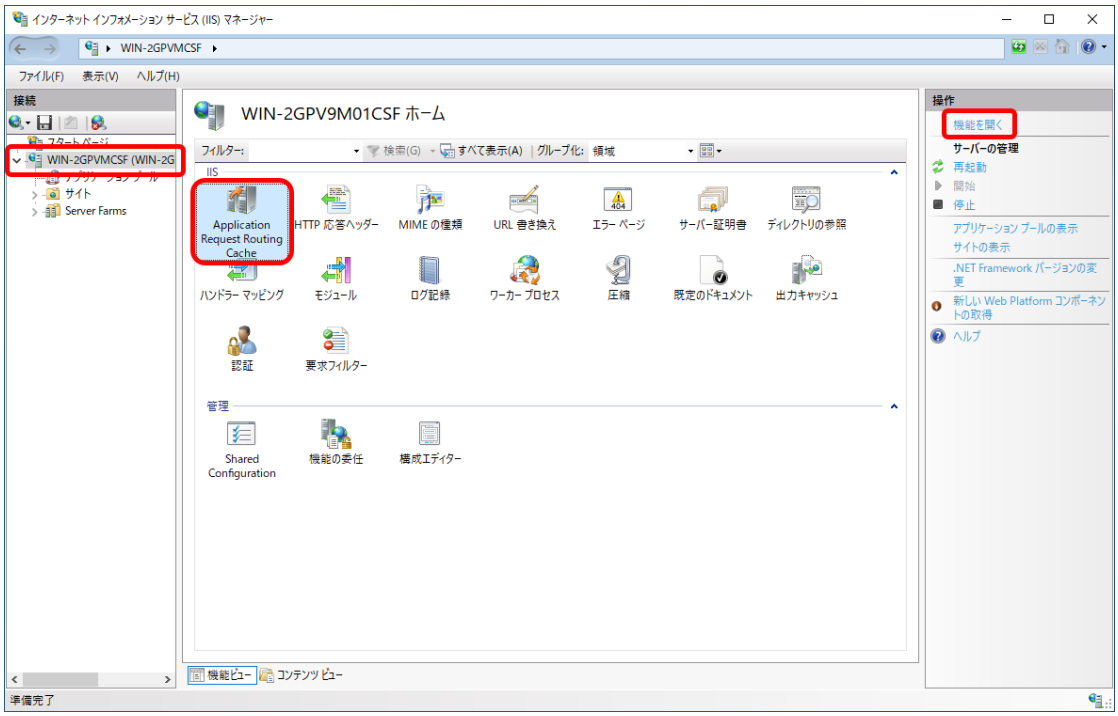
> **コラム**
>
> web.config とは Internet Information Services（IIS） の構成ファイルです。

> **注意**
>
> <web.config> ファイルは IM-Juggling 上では編集できません。

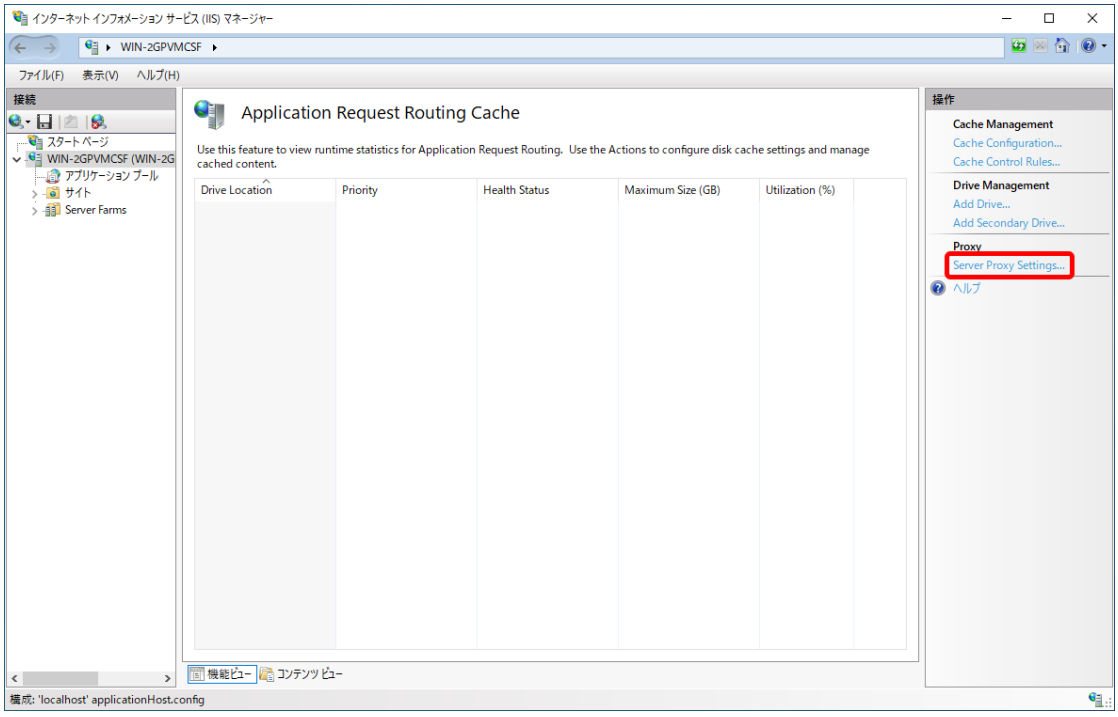
2. サーバ環境に合わせて web.config 内の下記の要素等を適宜変更します。
   - maxRequestLength, maxAllowedContentLength 属性
     - アップロードのファイルサイズの制限を変更する場合には、maxRequestLength, maxAllowedContentLength 属性の設定を変更してください。（上記の例では 100 MB）

> **コラム**
>
> web.config の詳細は以下のリファレンスを参照してください。
> https://learn.microsoft.com/ja-jp/aspnet/core/host-and-deploy/iis/web-config （日本語）

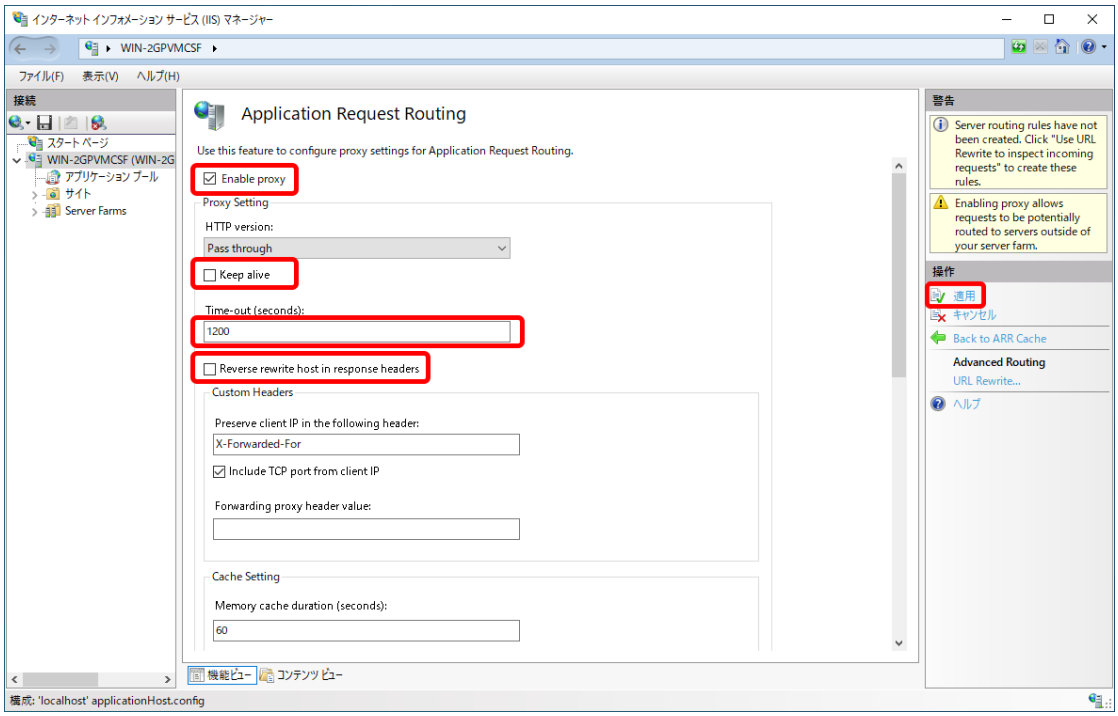
3. インターネット インフォメーション サービス (IIS) マネージャー より「IIS サーバ」「Application Request Routing Cache」「機能を開く」の順にクリックします。



4. 「Server Proxy Settings...」をクリックし、「Application Request Routing」画面を表示します。



5. 「Enable Proxy」をチェックし、「Time-out (seconds)」にタイムアウト値を秒単位で設定します。
「Reverse rewrite host in response headers」のチェックを外します。
統合Windows認証機能を利用していない場合、「Keep alive」のチェックを外します。
さらに、「適用」をクリックし、設定を反映します。

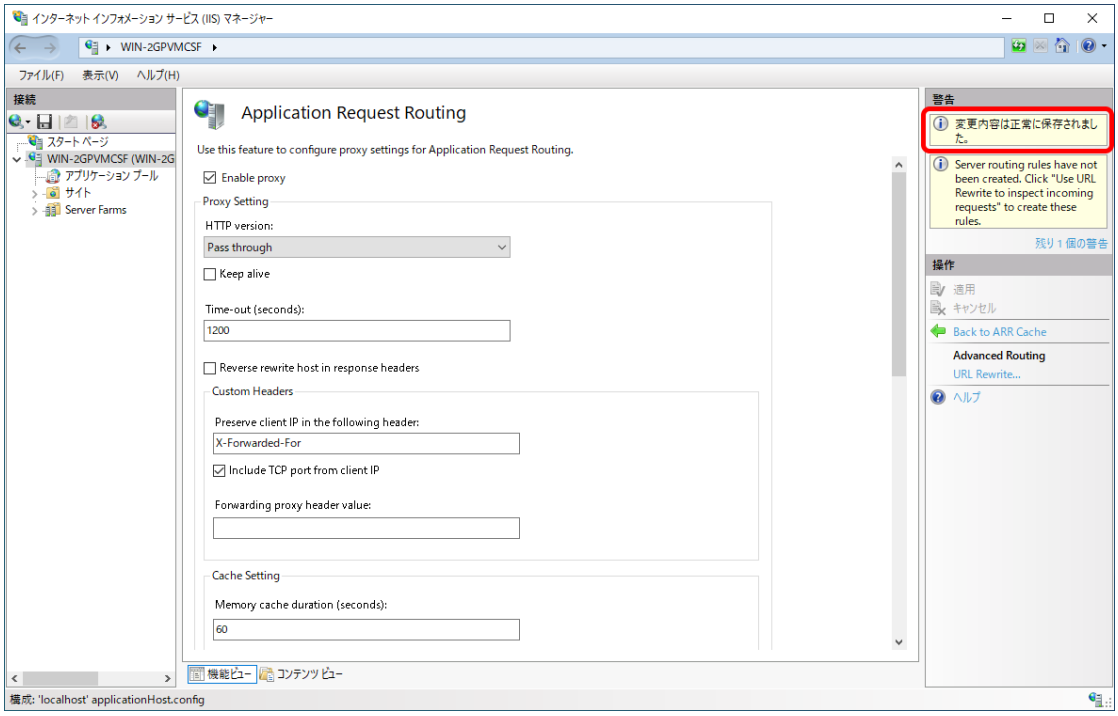


> **注意**
>
> 統合Windows認証機能を利用している場合「Keep alive」は必ず有効にしてください。

> **コラム**
>
> 「Keep alive」をオフにすることで、Bad Gateway エラーを減らすことができます。
> ただし、統合Windows認証機能を利用している場合には「Keep alive」はオフにせず必ず有効にしてください。

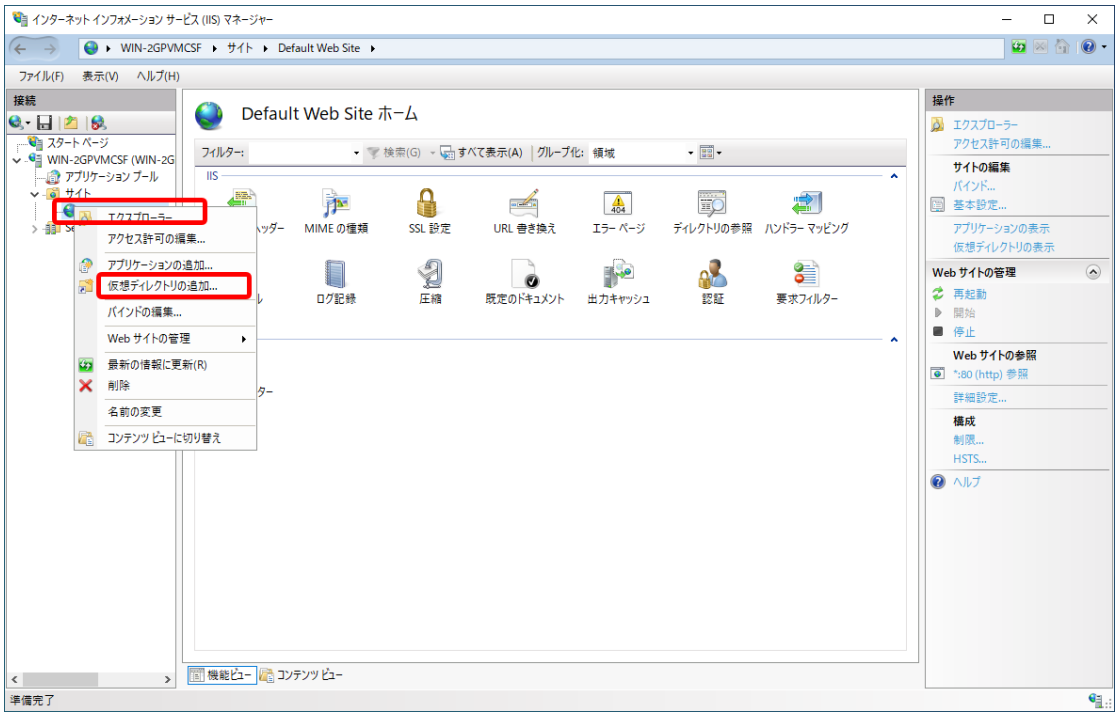
> **コラム**
>
> 「Time-out (seconds)」には Application Request Routing が Resin への接続を切断するまでのタイムアウト時間を設定します。
> テナント環境セットアップを完了するのに十分に大きい値を設定してください。
> ここでは例として 1200 秒を設定しています。

6. 「変更内容は正常に保存されました。」と表示されていることを確認してください。



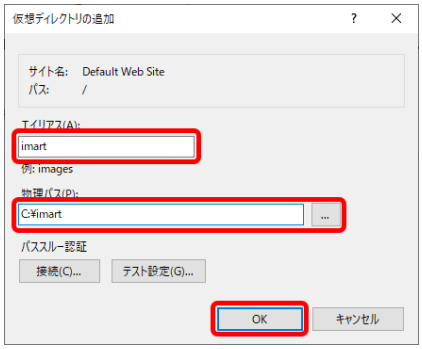
7. 続いて、静的ファイル用の仮想ディレクトリを作成します。

左ペインの「Default Web Site」のサブメニューより「仮想ディレクトリの追加...」をクリックします。

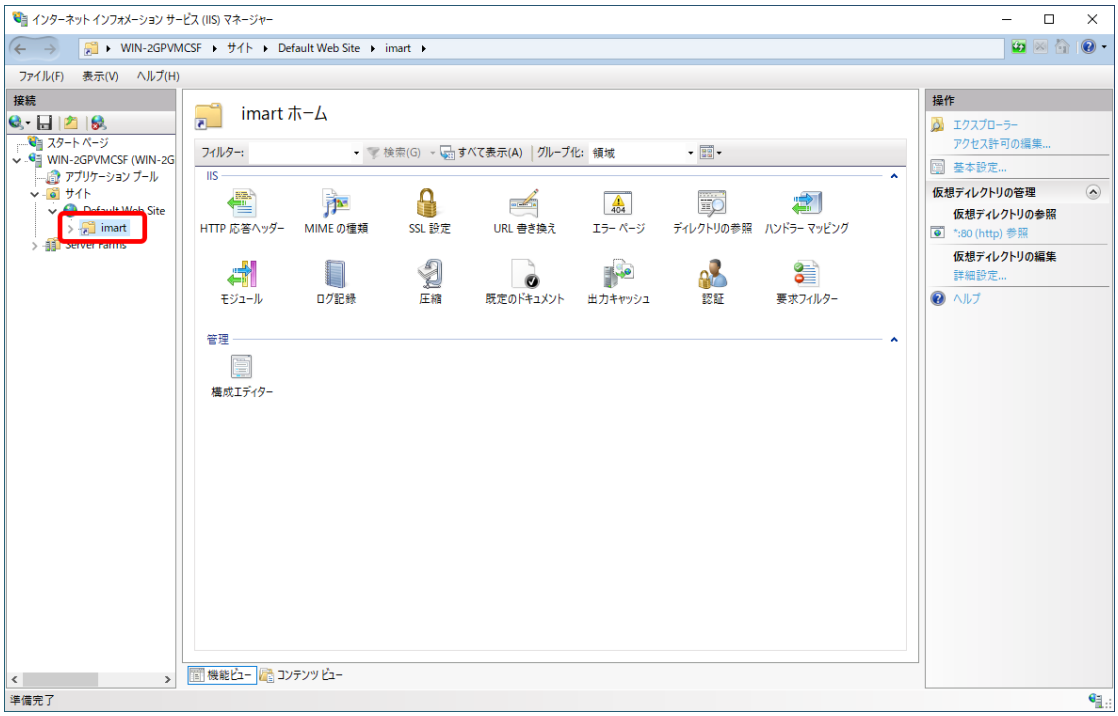


8. 新規ウィンドウ内で「エイリアス（A）」、「物理パス（P）」を次のように設定し、「OK」をクリックします。

エイリアス	imart
物理パス	C:\imart

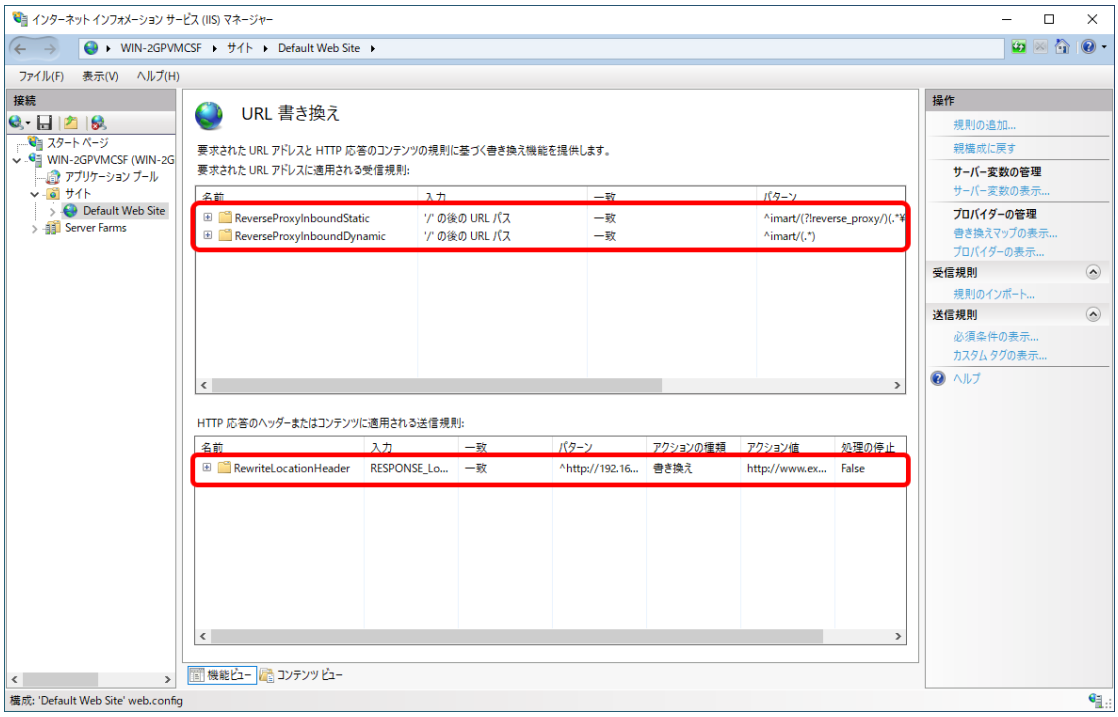


9. 「imart」ディレクトリが作成されていることを確認してください。



10. インターネット インフォメーション サービス (IIS) マネージャー より「Default Web Site」「URL 書き換え」「機能を開く」の順にクリックします。

11. 「要求されたURLアドレスに適用される受信規則」と「HTTP 応答のヘッダーまたはコンテンツに適用される送信規則」に下図のように追加されていることを確認します。



アクセスログの編集

リバース Proxy の場合、Resin が出力する access.log のソース IP がすべて IIS の IP に変わります。
回避するためには access.log の出力内容を編集します。

1. <%RESIN_HOME%/conf/resin.xml> ファイルを開きます。
2. <host id="" root-directory="."> ディレクティブ内に以下の設定を追加して Resin の再起動を行ってください。

```
<access-log path='log/access.log'>
  <rollover-period>1D</rollover-period>
  <format>%{X-Forwarded-For}i %h %l %u %t "%r" %>s %b "%{Referer}i" "%{User-Agent}i"</format>
</access-log>
```

コラム
access.log についての詳細な設定内容については「Resin Documentation 」を参照してください。

警告
運用環境でのご利用は、必ずWeb Serverを経由してWeb Application Serverへ接続してください。
静的コンテンツは、必ずWeb Serverに配置することが条件です。
これに該当しない接続の場合は、動作保証外（サポートの対象外）です。
Resin の httpd機能は、開発（e Builder）用です。

Apache Cassandra

Apache Cassandra は、「IMBox」を利用するために必要です。
詳細は「 IMBox Cassandra管理者ガイド 」を参照してください。

コラム
*プロジェクトの作成とモジュールの選択*で IMBox アプリケーションを選択しない場合、Apache Cassandra のセットアップは不要です。

Apache Solr

Apache Solr は、「IM-ContentsSearch」を利用するために必要です。
セットアップ方法の詳細は「 Solr管理者ガイド 」-「 Solrのセットアップ 」を参照してください。

コラム
*プロジェクトの作成とモジュールの選択*で IM-ContentsSearch アプリケーションを選択しない場合、Apache Solr のセットアップは不要です。

WARファイルの作成

プロジェクトの作成とモジュールの選択

項目

- IM-Juggling とは
- IM-Juggling のインストール
  - インターネットに接続できない環境で IM-Juggling を利用する場合
  - プロキシ設定が必要な環境で IM-Juggling を利用する場合
- プロジェクトの新規作成
- モジュールの選択
- アプリケーションの追加

IM-Juggling とは

IM-Juggling はモジュールの管理、WARファイルを出力できる環境構築ツールです。
このツールを使用し、新しい機能の導入や不具合修正の適用をします。

IM-Juggling のインストール

製品に同梱されているディレクトリより、環境に合わせた IM-Juggling を任意のパスに展開します。

> **コラム**
>
> IM-Juggling は、プロダクトファイルダウンロード よりダウンロードすることも可能です。
> intra-mart Accel Platform のライセンスキーを入力してダウンロードしてください。

> **注意**
>
> IM-Juggling を利用してプロジェクトの作成や、WARファイルの作成を行っている際にリポジトリ情報の取得等の
> エラーが発生した場合、下記のディレクトリにあるデータを削除して再度、 IM-Juggling を起動して試行してください。
>
> %OSユーザディレクトリ%/juggling/workspace/.repository ディレクトリ
>
> 古い情報が残っているためにエラーとなる場合があります。
> この古いファイルを削除する事で、最新のデータが再取得され問題を回避します。

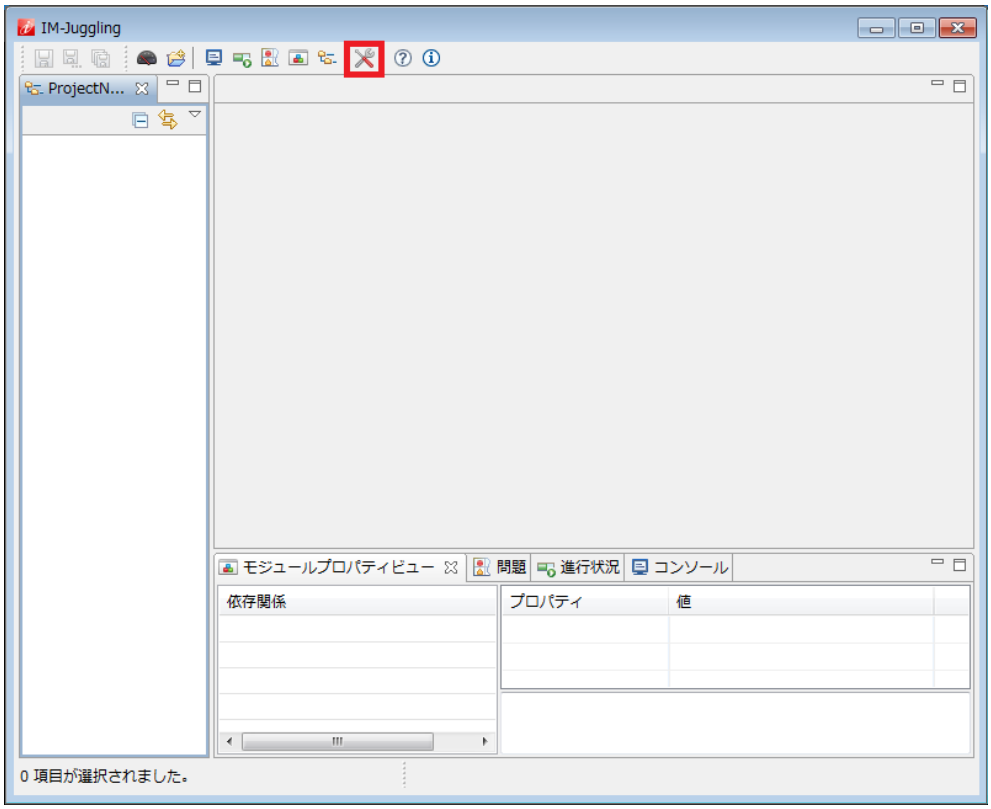
インターネットに接続できない環境で IM-Juggling を利用する場合

IM-Juggling で利用するリポジトリデータは以下の方法で取得できます。

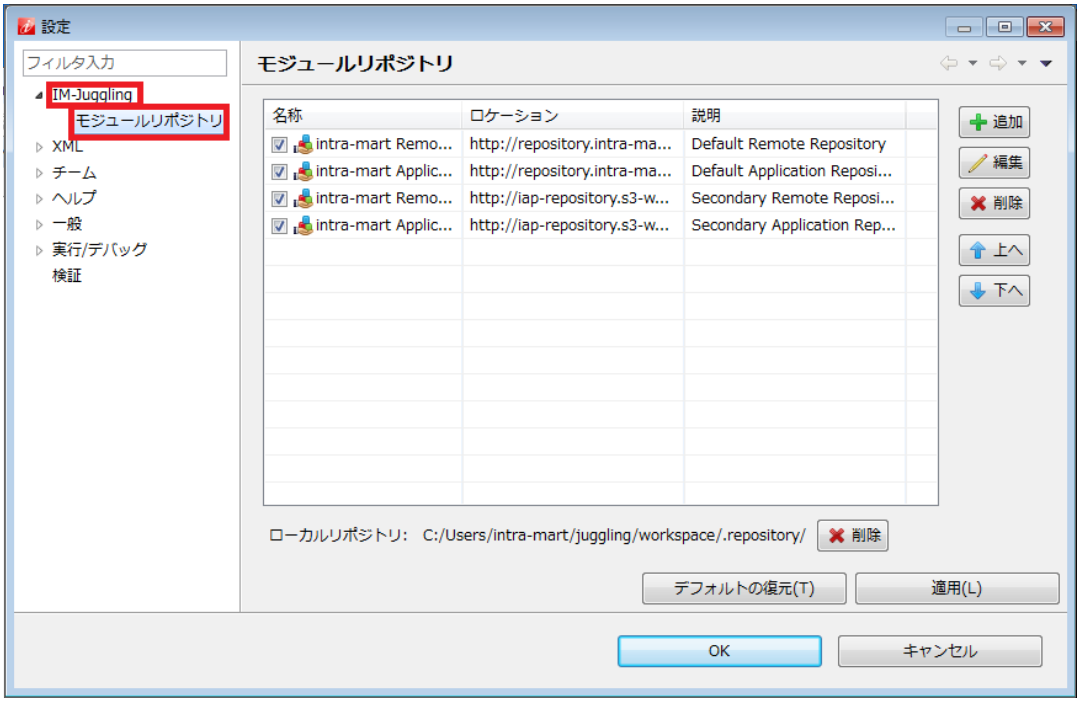
- プロダクトファイルダウンロード より intra-mart Accel Platform のライセンスキーを入力してダウンロード

リポジトリデータをコピーし、 IM-Juggling のモジュールリポジトリとして設定することで、インターネットに接続できない環境でも IM-Juggling を利用できます。
以下の方法で、「platform」「products」のそれぞれのリポジトリデータをモジュールリポジトリとして設定してください。

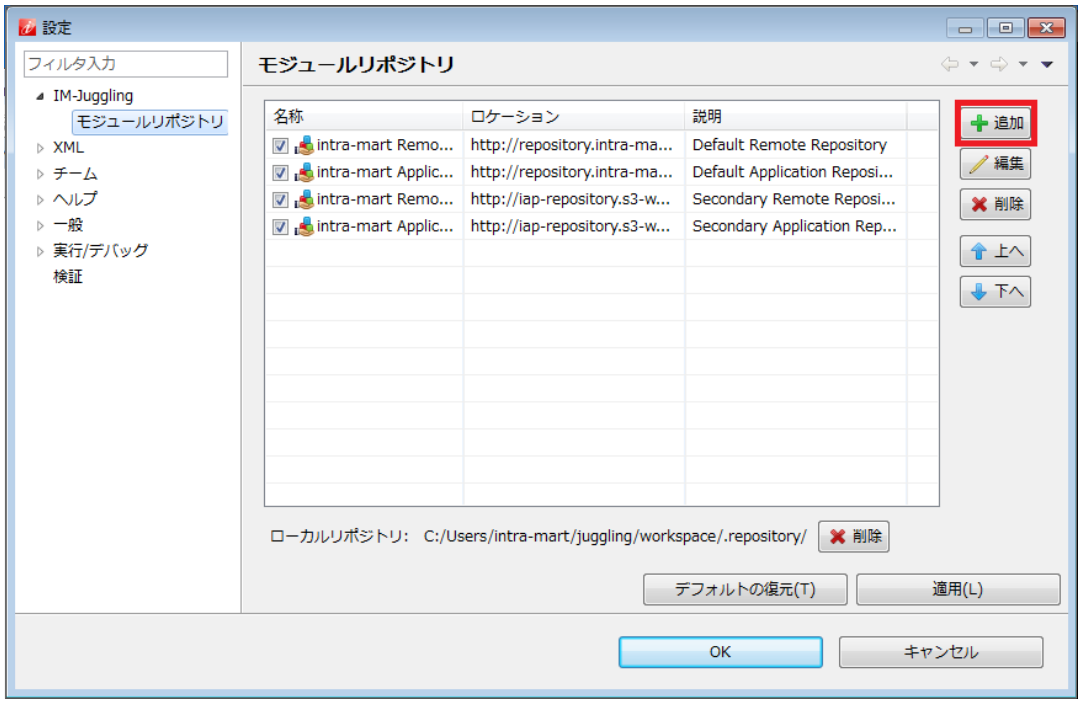
1. IM-Juggling ウィンドウ内-ツールバー右端にある「設定」アイコンをクリックします。



2. 「IM-Juggling」-「モジュールリポジトリ」を選択します。

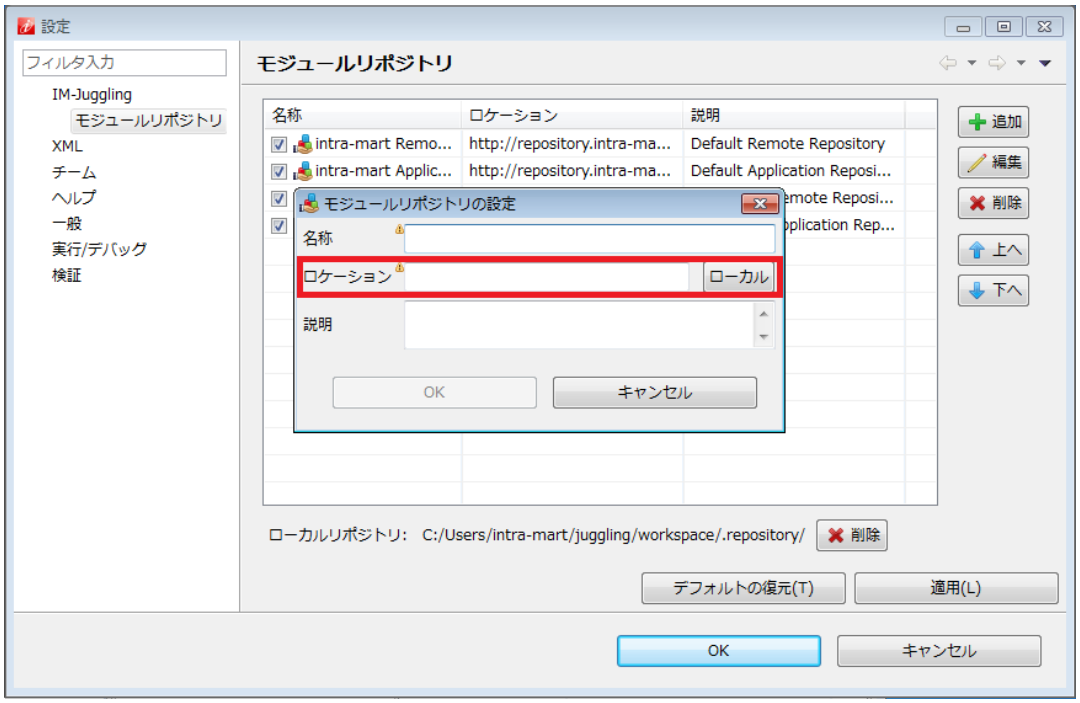


3. 「追加」をクリックします。

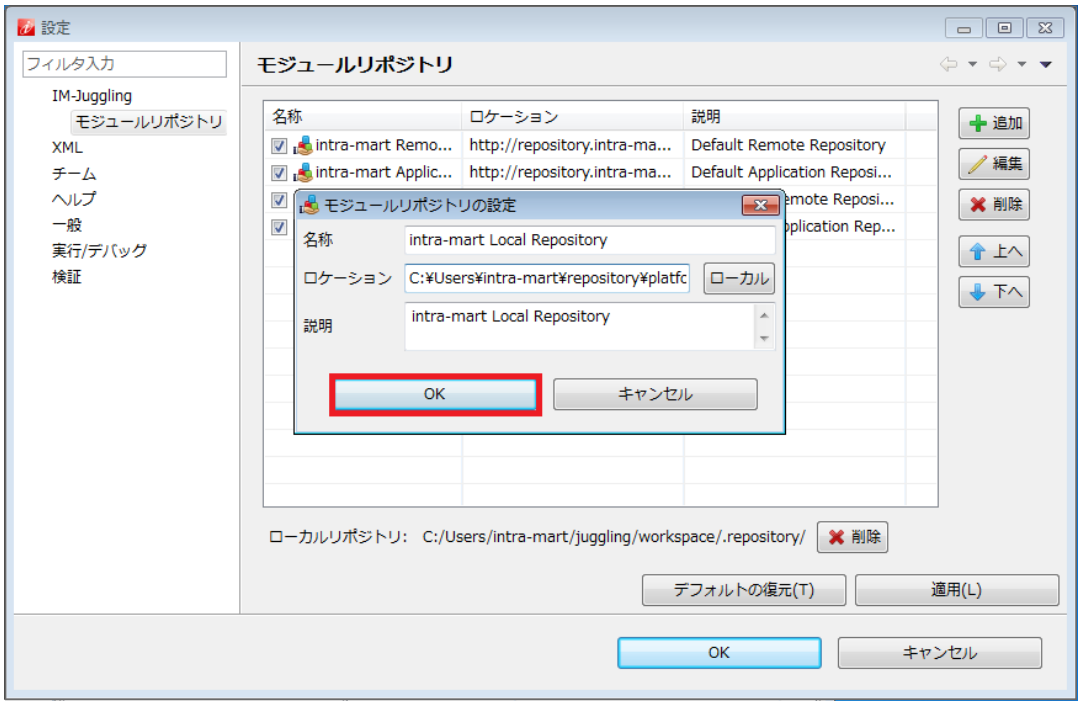


4. モジュールリポジトリの追加画面が表示されます。

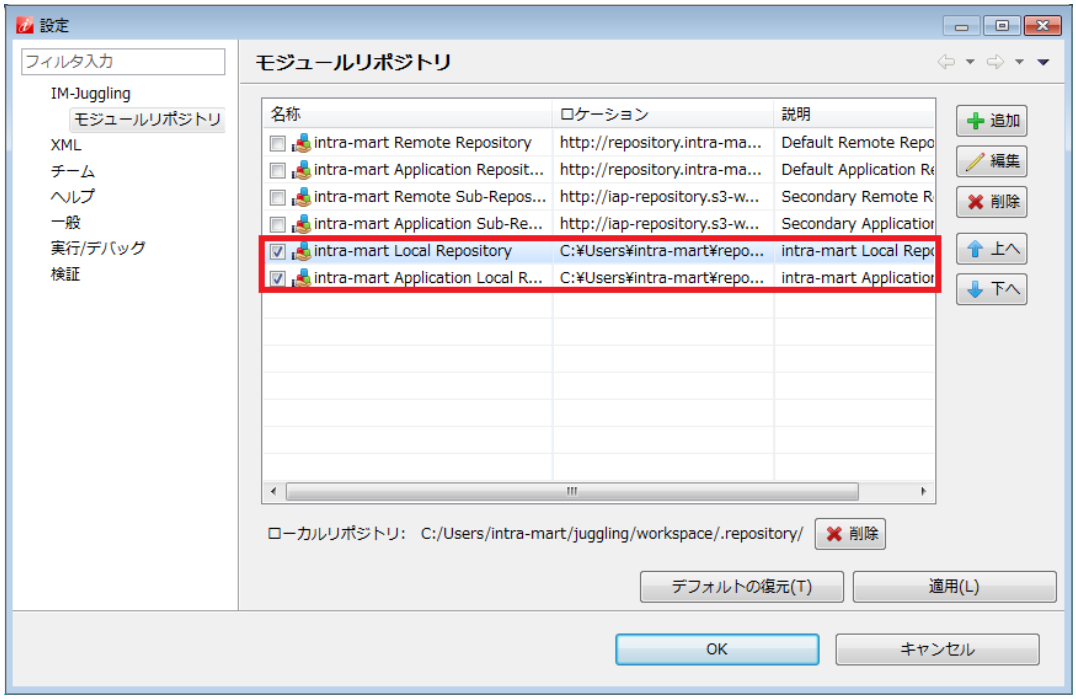
5. 「ロケーション」にコピーしたリポジトリデータの場所を設定します。



6. 「名称」「説明」を記入し、「OK」をクリックします。

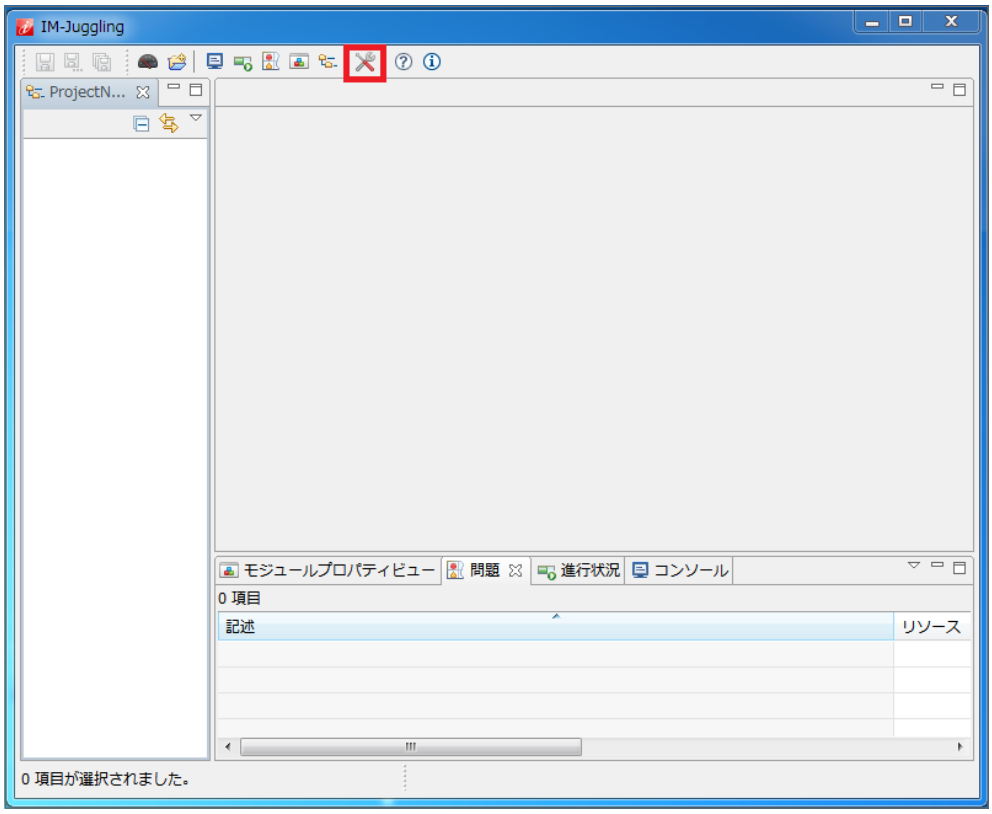


7. 追加したモジュールリポジトリにチェックがついていることを確認し、「OK」をクリックします。

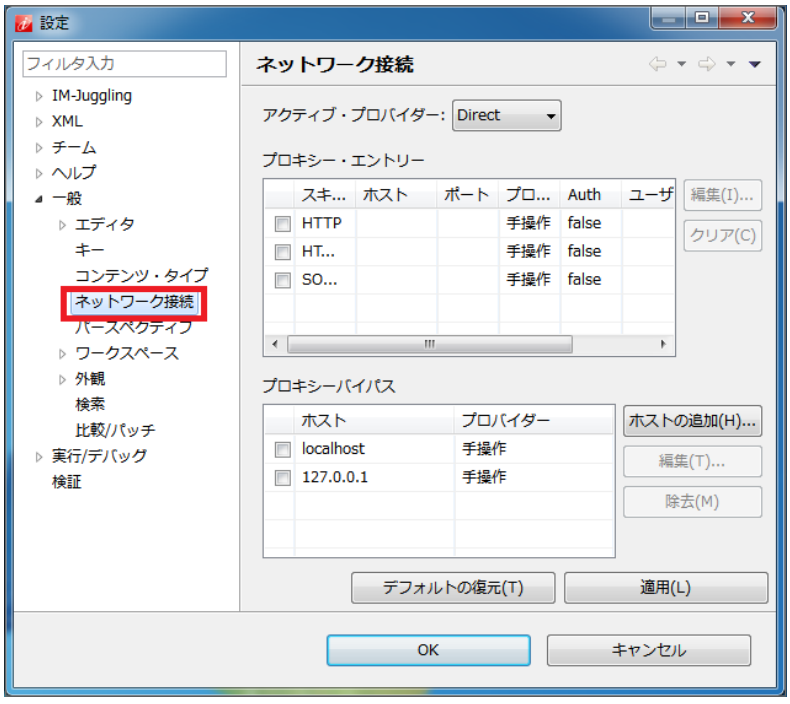


プロキシ設定が必要な環境で IM-Juggling を利用する場合

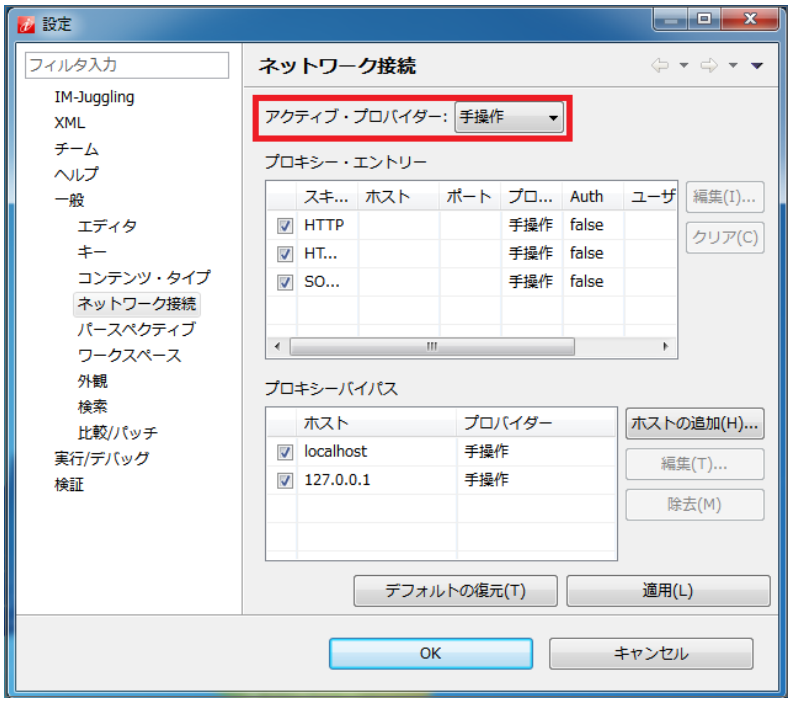
1. IM-Juggling ウィンドウ内-ツールバー右端にある「設定」アイコンをクリックします。



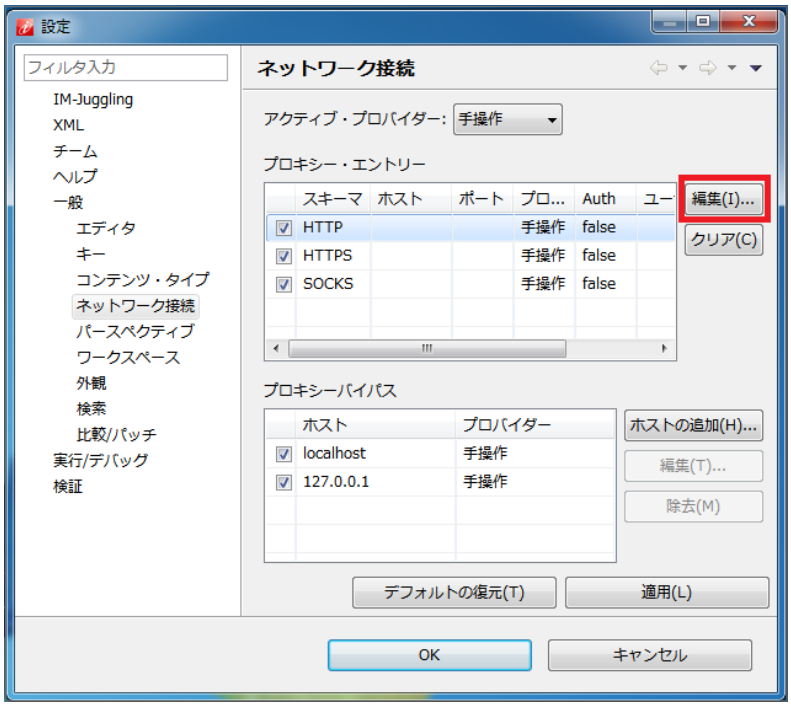
2. 「一般」-「ネットワーク接続」を選択します。



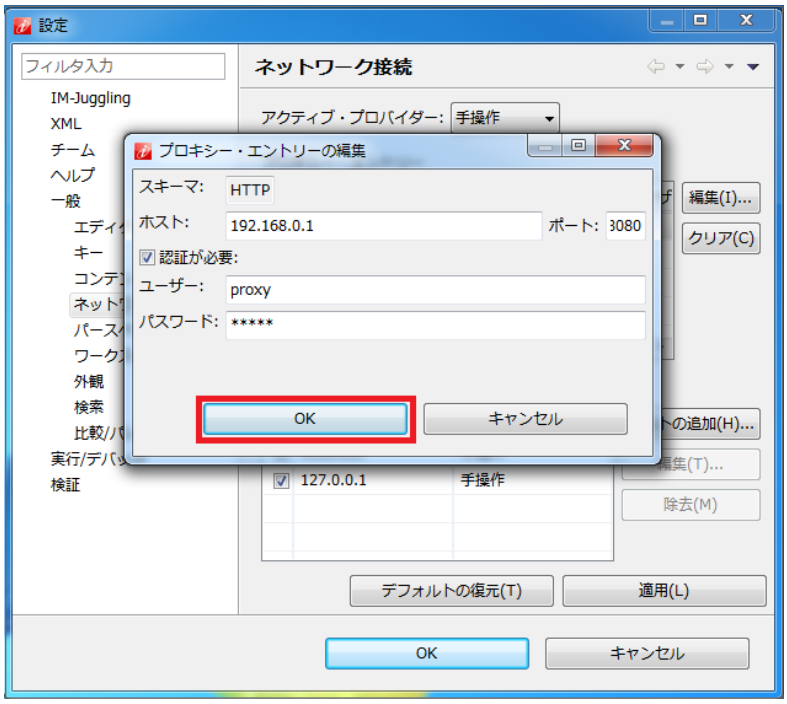
3. アクティブ・プロバイダに「手操作（マニュアル）」を選択します。



4. プロキシー・エントリーの「HTTP」の「編集」をクリックします。



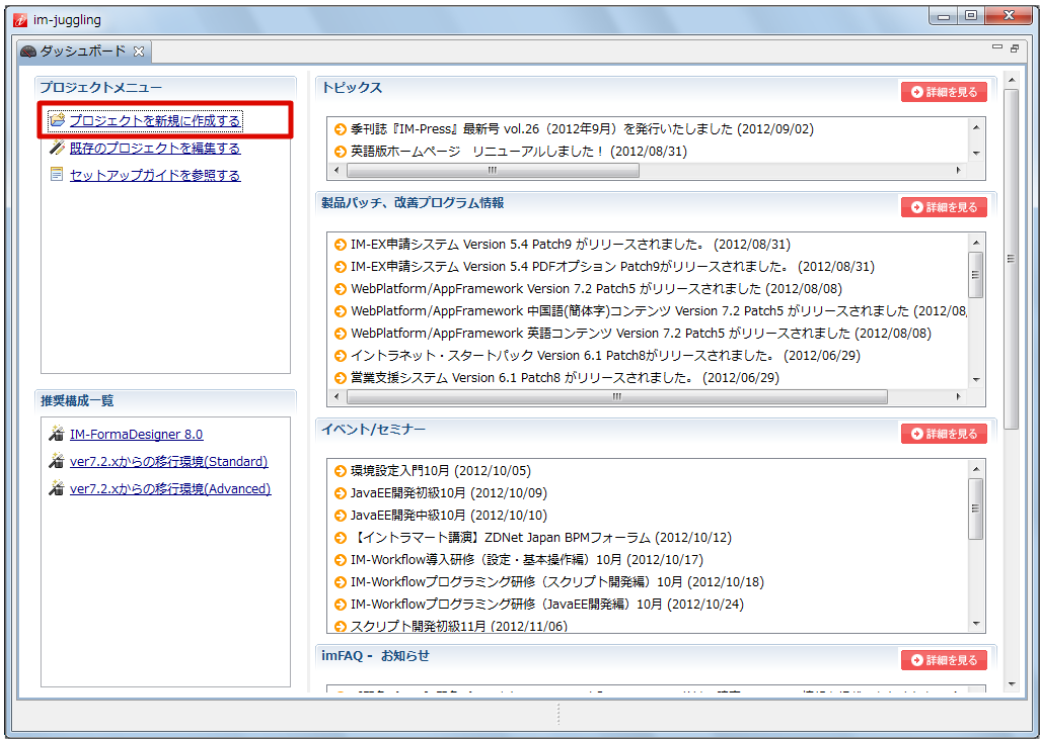
5. プロキシー・エントリーの編集」画面で「ホスト」、「ポート」、「認証情報」を設定して「OK」をクリックします。



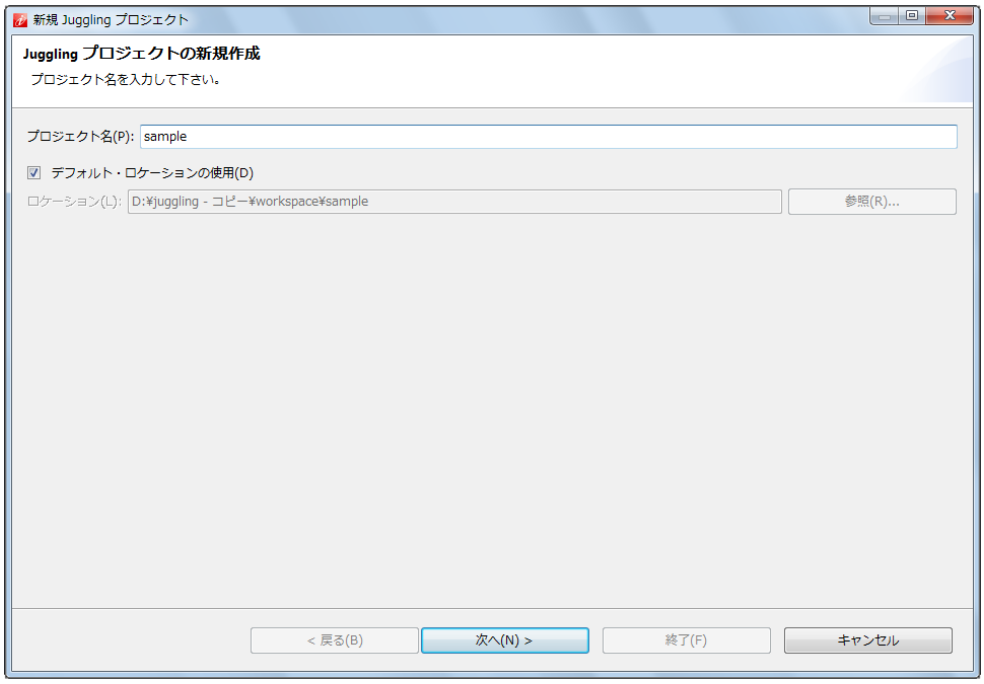
> **注意**
>
> アクティブ・プロバイダとして「ネイティブ」を選択した場合、Internet Explorer のプロキシサーバの設定情報を引き継ぎます。
> （Internet Explorerで「Alt」キーを押してメニューを表示し、「ツール」→「インターネットオプション」→「接続」タブ→「LANの設定」で表示されるプロキシサーバの設定です）

プロジェクトの新規作成

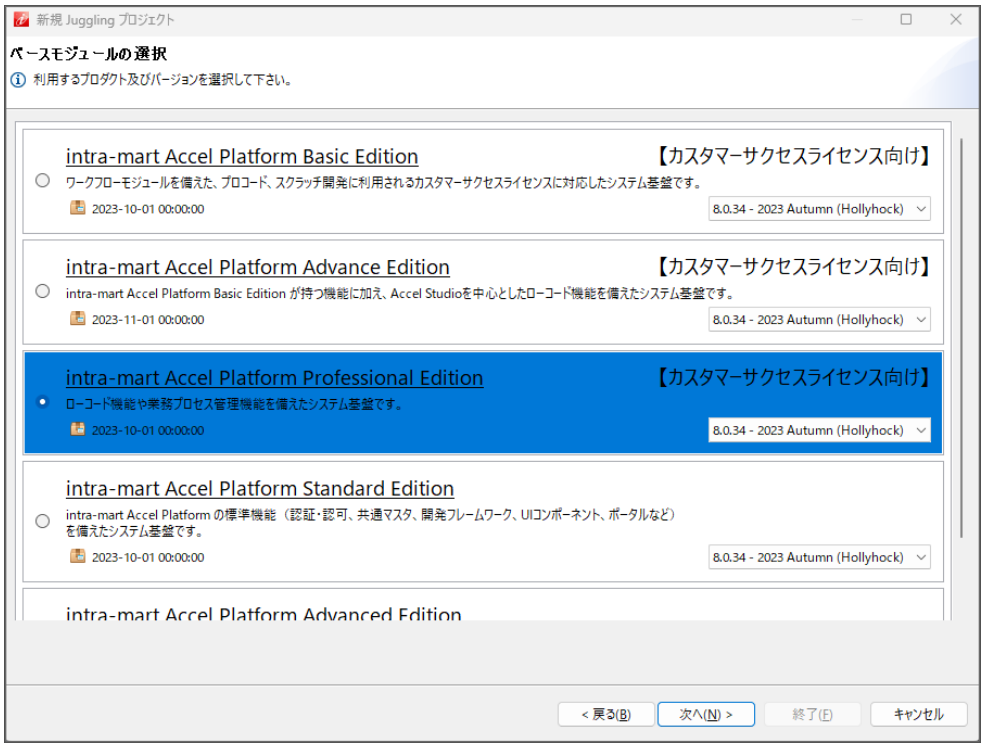
1. 展開したディレクトリ直下にある、 **juggling.exe** を起動します。
2. 初回起動時の場合、「使用許諾契約書の同意」が表示されます。
   「同意する」を選択し「OK」をクリックします。
3. ダッシュボードが表示されます。
   「プロジェクトを新規に作成する」を選択します。



4. ポップアップ表示されたウィザード画面内の「プロジェクト名」に半角英数字の任意のプロジェクト名を入力します。
   入力後、「次へ（N）」 をクリックします。



5. 利用するプロダクトおよびバージョンを選択します。
   選択後、「次へ（N）」をクリックします。



注意

ライセンスの種別によって、選択できるintra-mart Accel Platformの構成が異なります。

- **カスタマーサクセスライセンスの場合**

  intra-mart Accel Platformには、以下の3つの構成が用意されています。
  各構成の選択方法は次の通りです。

  - **ベーシック**

    ＜ intra-mart Accel Platform Basic Edition ＞のツリーから選択します。

  - **アドバンス**

    ＜ intra-mart Accel Platform Advance Edition ＞のツリーから選択します。

  - **プロフェッショナル**

    ＜ intra-mart Accel Platform Professional Edition ＞のツリーから選択します。

- **パッケージライセンスの場合**

  intra-mart Accel Platformには、以下の3つの構成が用意されています。
  各構成の選択方法は次の通りです。

  - **スタンダード**

    ＜ intra-mart Accel Platform Standard Edition ＞のツリーから選択します。

  - **アドバンスト**

    ＜ intra-mart Accel Platform Advanced Edition ＞のツリーから選択します。

  - **エンタープライズ**

    ＜ intra-mart Accel Platform Advanced Edition ＞のツリーから選択します。
    次の「アプリケーションの選択」より
    IM-BloomMaker for Accel Platform 、 IM-FormaDesigner for Accel Platform 、 IM-BIS for Accel Platform 、
    IM-RPA for Accel Platform を選択します。

    また、IM-BPM for Accel Platform を購入されている場合は IM-BPM for Accel Platform も選択します。

- **ASEAN地域向けカスタマーサクセスライセンスの場合**

  intra-mart Accel Platformには、以下の2つの構成が用意されています。
  各構成の選択方法は次の通りです。

  - **プロコード**

    ＜ intra-mart Accel Platform Pro-Code Edition ＞のツリーから選択します。

  - **ローコード**

    ＜ intra-mart Accel Platform Low-Code Edition ＞のツリーから選択します。

ライセンスの種別の詳細は、「 *ライセンスについて* 」を参照してください。
ご購入頂いたライセンスは、全ての製品構成とも、「 *ライセンスの登録* 」にて登録を行います。
製品構成の詳細については、「Introduction 」を参照してください。

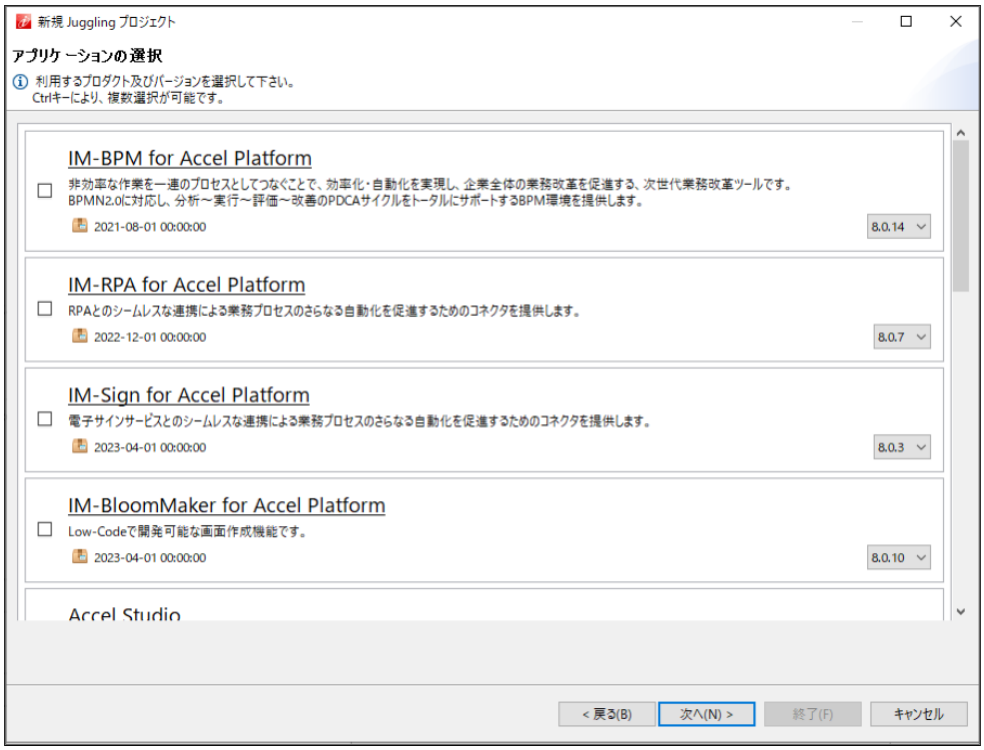
コラム

同一プロダクトにおいて複数のバージョンが表示されている場合は最新のものを選択する事を推奨します。

コラム

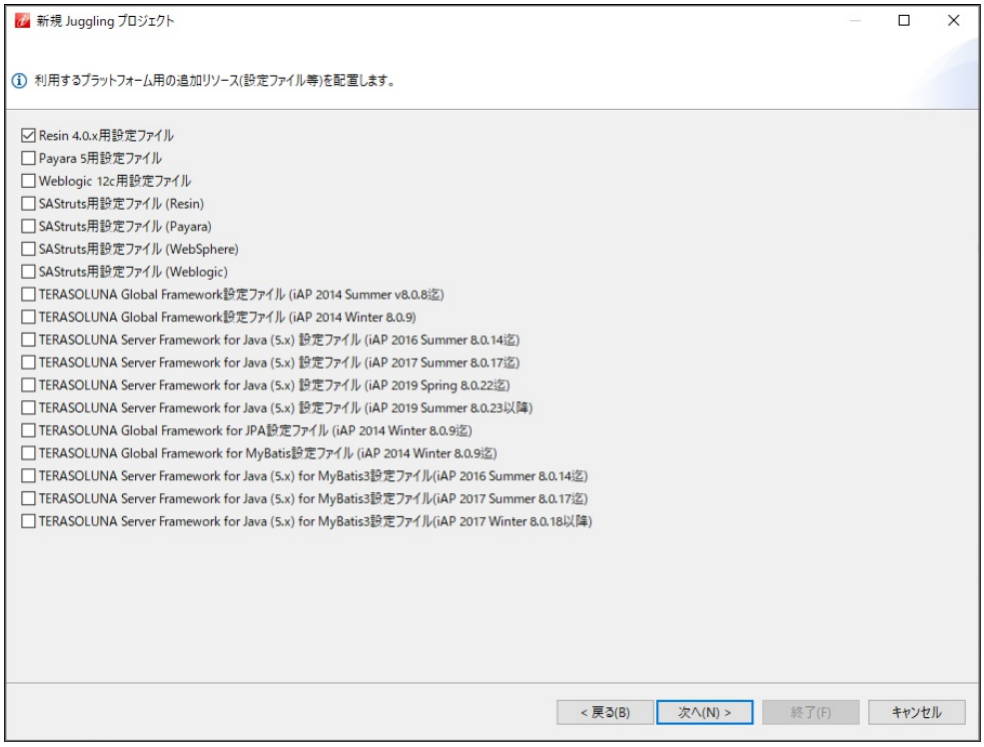
カスタマーサクセスライセンス向けのベーシックとプロフェッショナルは、intra-mart Accel Platform 2023 Spring(Gerbera) 以降より、ご利用いただけます。
カスタマーサクセスライセンス向けのアドバンスは、intra-mart Accel Platform 2023 Autumn(Hollyhock) 以降より、ご利用いただけます。
ASEAN地域向けのプロコードとローコードは、intra-mart Accel Platform 2024 Spring(Iris) 以降より、ご利用いただけます。

6. 利用するアプリケーションを選択します。
   選択後、「次へ（N）」をクリックします。



> **注意**
>
> 「移行ツール」は intra-mart WebPlatform v7.2 からの移行を行う場合に選択します。
> 新規で intra-mart Accel Platform を構築する場合は、「移行ツール」は 選択しないようにしてください。
> 移行を行う場合は、別途公開されている「移行ガイド」を参照してください。
>
> 「互換機能」は 互換APIや互換機能を利用する場合に選択します。
> 新規または移行によって選択するモジュールが異なりますので、 「互換ガイド」を参照して必要なものだけ選択してください。
> 互換APIや互換機能の利用についても「互換ガイド」を参照してください。

7. 利用するプラットフォーム用の追加リソース（設定ファイル等）を選択します。
   選択後、「次へ（N）」をクリックします。

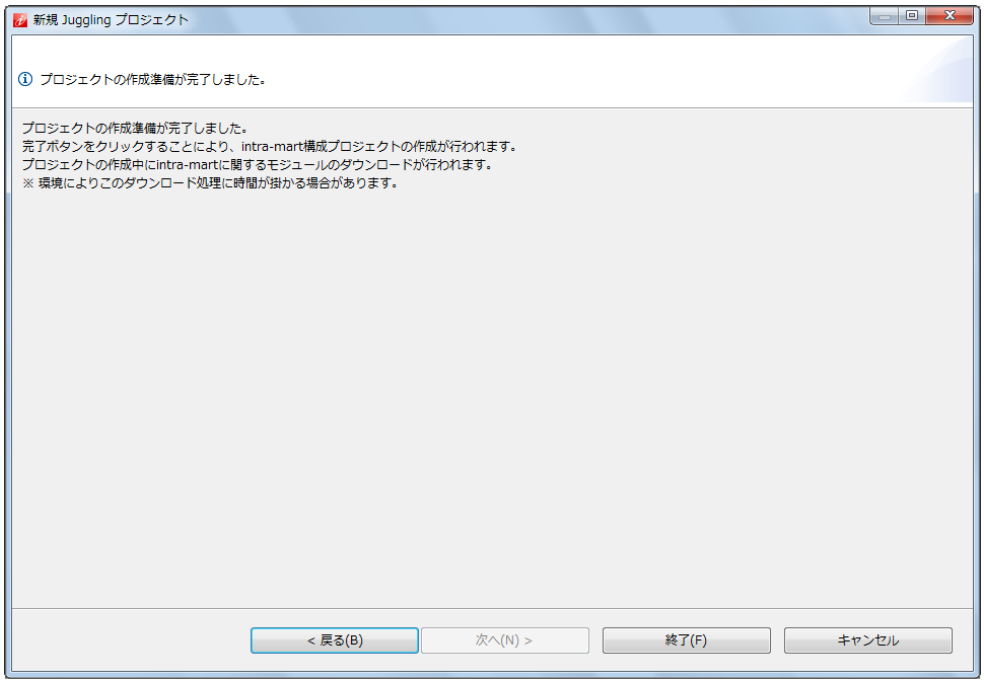


> **注意**
>
> WARファイルを Oracle WebLogic Server 12c R2(12.2.1) へデプロイする場合は、必ず「**Weblogic 12c用設定ファイル**」を選択してください。
> この設定ファイルを配置していない場合、WebLogicへデプロイする際にエラーが発生します。

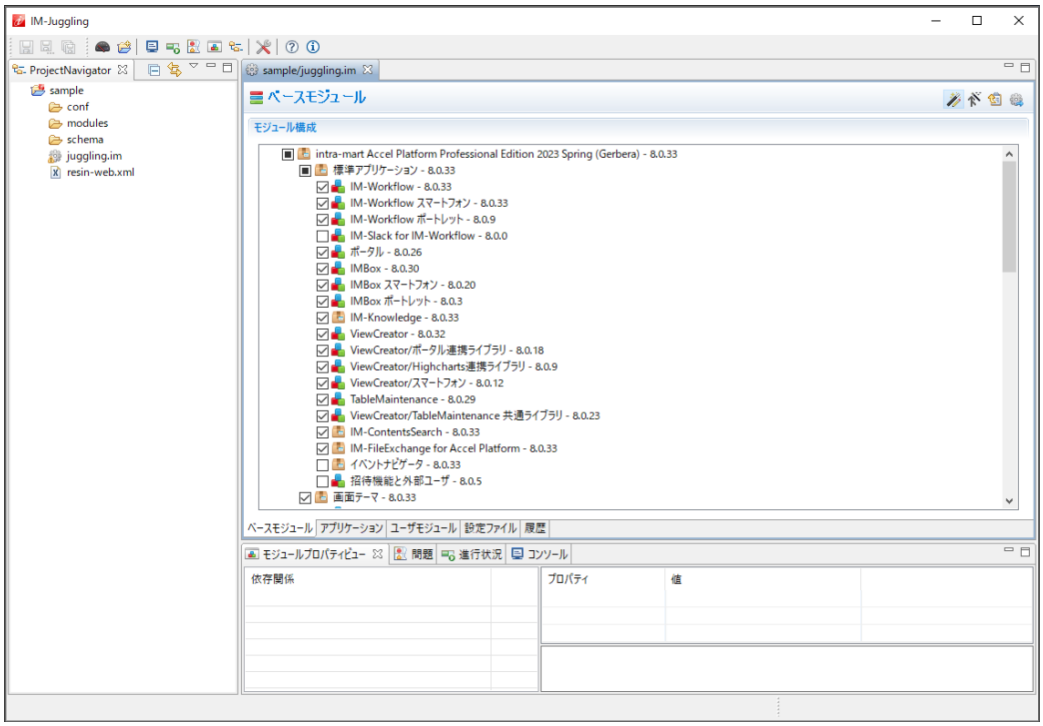
> **コラム**
>
> SAStruts+S2JDBCにてデータベースを利用する場合は、 使用するWebアプリケーションサーバに応じた「**SAStruts用設定ファイル**」にチェックをつけてください。
> 設定変更については、 *SAStruts用設定ファイル（SAStruts+S2JDBCにてデータベースを利用する場合）* で説明します。

> **コラム**
>
> TERASOLUNA Server Framework for Java (5.x) を利用する場合は、「**TERASOLUNA Server Framework for Java (5.x) 用設定ファイル**」にチェックをつけてください。
> 設定変更については、 *TERASOLUNA Server Framework for Java (5.x) 用設定ファイル（シェアードデータベースを利用する場合）* で説明します。
>
> リポジトリ層にMyBatis3を利用する場合は、「**TERASOLUNA Server Framework for Java (5.x) for MyBatis3用設定ファイル**」にチェックをつけてください。
> 設定変更については、 *TERASOLUNA Server Framework for Java (5.x) 用設定ファイル（リポジトリ層にMyBatis3を利用する場合）* で説明します。

8. 最後に「終了（F）」をクリックします。



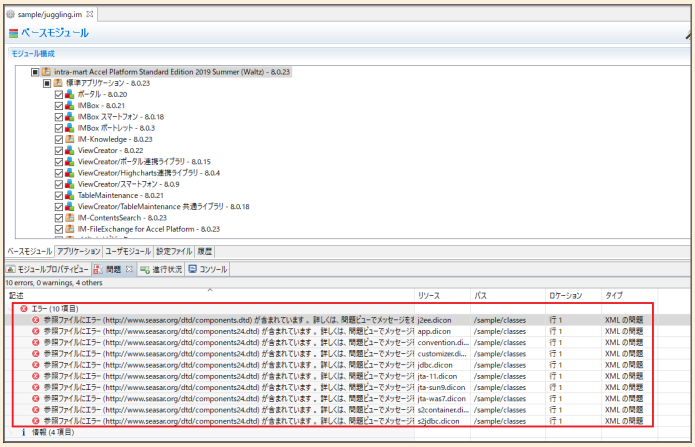
9. プロジェクトの作成処理が行われ、完了すると以下の画面が表示されます。



注意

SAStruts 用設定ファイルを出力した場合、各dicon ファイルに関して以下のようなエラーが出力されます。

- 参照ファイルにエラー (http://www.seasar.org/dtd/components24.dtd) が含まれています 。 詳しくは、問題ビューでメッセージを右クリックし、「詳細表示...」を選択してください



上記のエラーは war ファイルの作成、intra-mart Accel Platform の動作に影響はありません。
このエラーを解消する場合は、出力された dicon ファイルの DTD の参照先を http から https に変更してください。
2019 Winter(Xanadu) 以降の Juggling では、デフォルトで出力する dicon ファイルの DTD の参照先が https のため、Juggling プロジェクトを新規作成する場合はエラーは出力されません。

注意

WAR ファイルを Resin 以外のサーバへデプロイする場合は、下記のモジュールを選択しないでください。
これらのライブラリは Resin 以外のサーバで利用できないためデプロイまたは起動時にエラーが発生します。

- ライブラリ > サードパーティ製ライブラリ > Metro
- ライブラリ > サードパーティ製ライブラリ > OpenPortal WSRP

注意

開発フレームワーク「 TERASOLUNA Server Framework for Java (5.x) for Accel Platform 」モジュールは WebSphere Application Server 9.0.5 では利用できません。
そのため、WARファイルを WebSphere Application Server 9.0.5 へデプロイする場合は、開発フレームワーク「 TERASOLUNA Server Framework for Java (5.x) for Accel Platform 」モジュールを選択しないでください。

> **注意**
>
> 外部ソフトウェアとの連携は Web API Maker または IM-LogicDesigner を利用してください。
>
> - 標準機能 > 基盤機能 > Web API Maker
> - 標準機能 > 基盤機能 > IM-LogicDesigner
>
> 外部ソフトウェア接続モジュールはセキュリティ面での懸念があるため非推奨です。
> 下記のモジュールを利用する場合は十分な注意が必要です。
>
> - 追加機能 > 外部連携クライアント
> - 追加機能 > 外部連携 認証・認可

モジュールの選択

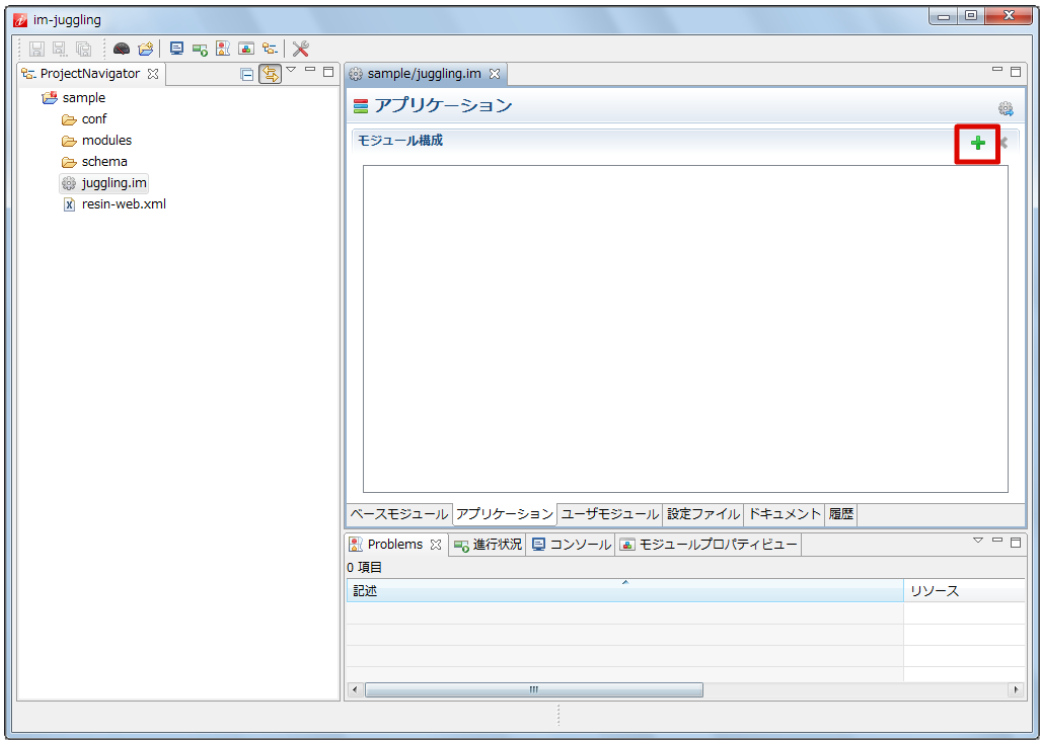
1. プロジェクト内に表示されているモジュールの右クリックメニューで表示されるサブメニューより、
   一括で関連するモジュールの選択、選択解除を行う事ができます。

> **コラム**
>
> 例として、 IMBox を構成に含めない場合、 IMBox モジュールを外す必要があります。
> 詳細は、「*IMBox モジュールを外す方法*」を参照してください。

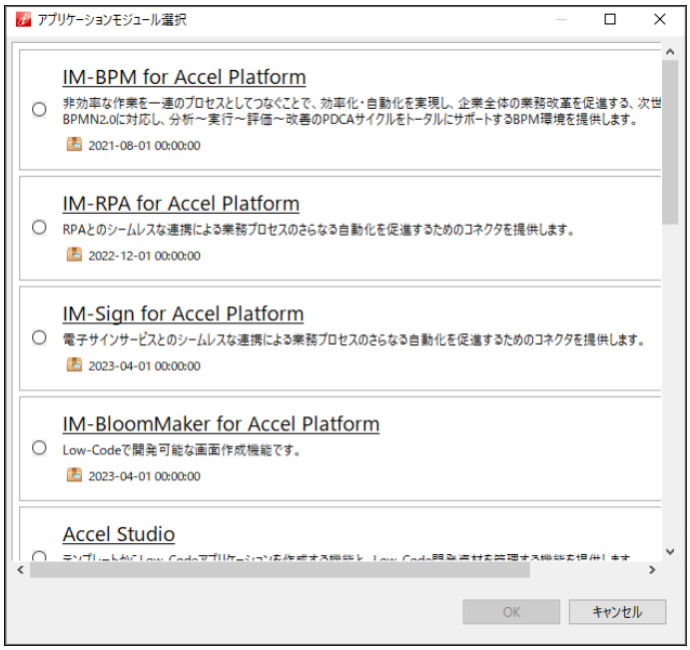
アプリケーションの追加

プロジェクト作成後、アプリケーションを追加できます。

1. 「アプリケーション」タブをクリックし、「+」をクリックします。



2. 追加したいアプリケーションをクリックし、「OK」をクリックします。



注意

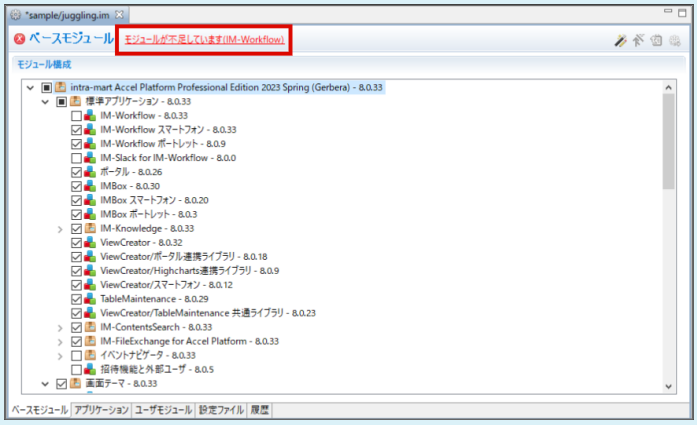
「移行ツール」は intra-mart WebPlatform v7.2 からの移行を行う場合に選択します。
新規で intra-mart Accel Platform を構築する場合は、「移行ツール」は 選択しないようにしてください。
移行を行う場合は、別途公開されている「移行ガイド」を参照してください。

「互換機能」は 互換APIや互換機能を利用する場合に選択します。
新規または移行によって選択するモジュールが異なりますので、 「互換ガイド」を参照して必要なものだけ選択してください。
互換APIや互換機能の利用についても「互換ガイド」を参照してください。

コラム

画面上部にエラーメッセージが表示される場合

モジュール単位・アプリケーション単位で依存関係を持っています。



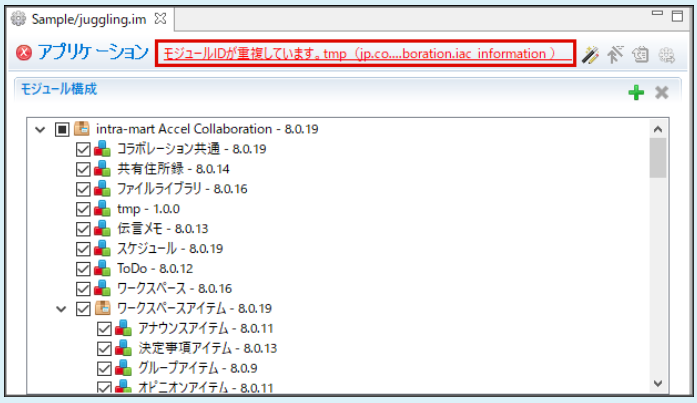
以下の手順で必要な設定を追加できます。

1. メッセージをクリックします。
2. 「依存関係の解決」画面で「OK」をクリックします。

コラム

モジュールIDが重複した場合

以下のようなエラーメッセージが表示されます。



モジュールID が重複しているモジュールを削除してください。
なお、削除実施後は<juggling.im>ファイルを開きなおしてください。

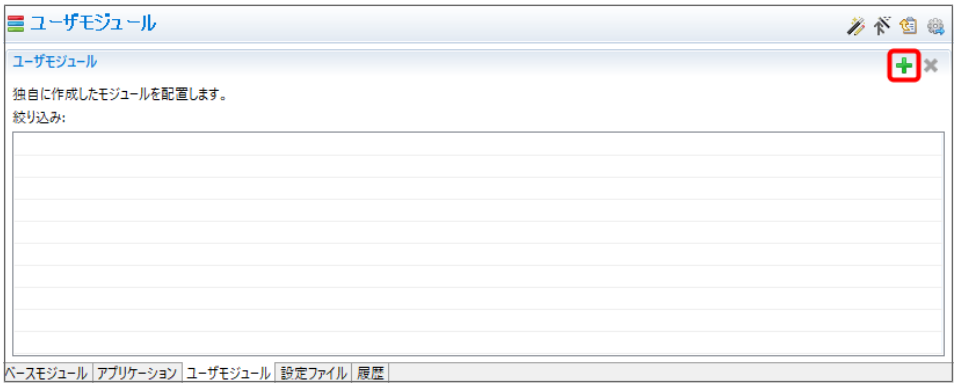
コラム

アプリケーション別の情報については、各アプリケーションのセットアップガイドを参照してください。
IM-BIS for Accel Platform を追加した場合は、「IM-BIS セットアップガイド」を参照してください。
IM-FormaDesigner for Accel Platform を追加した場合は、「IM-FormaDesigner セットアップガイド」を参照してください。

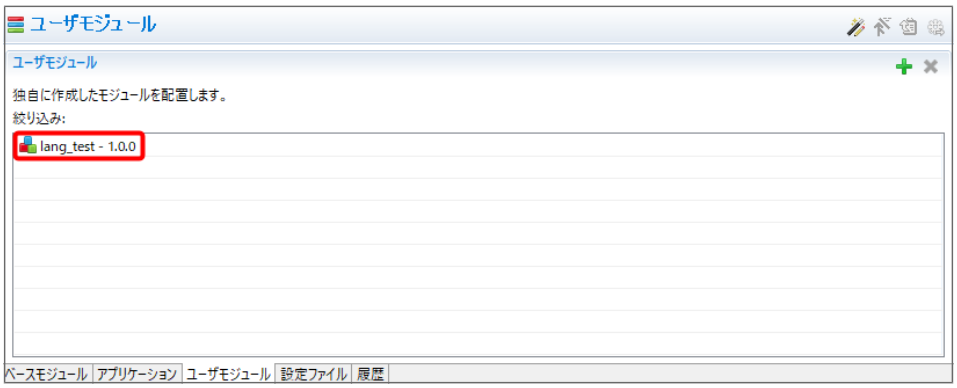
ユーザモジュール

プロジェクト作成後、独自で作成したアプリケーションをユーザモジュールとしてWARファイルに追加できます。

1. 「ユーザモジュール」タブをクリックし、「+」をクリックします。



2. 追加したいimmファイルを選択し、「OK」をクリックします。



intra-mart Accel Platform の設定ファイル

intra-mart Accel Platform を稼働させるために以下の設定ファイルの編集を行います。

> **注意**
>
> 自作したユーザモジュール内に含まれる設定ファイルと、同一の設定ファイルが既に IM-Juggling プロジェクト上に存在する場合、このプロジェクトのWARファイルを出力すると、 IM-Juggling プロジェクト上に配置されている設定ファイルが反映されます。
> IM-Juggling プロジェクトの設定ファイルが反映される動作仕様です。
> ユーザモジュールを動作させるための設定は、 IM-Juggling プロジェクト上に配置されている設定ファイル側に書き写す必要があります。

基本

DataSource

intra-mart Accel Platform で利用するデータベースの設定を行います。

項目

- 接続先の種類
- DataSourceの設定
  - PostgreSQL
  - Oracle
  - Microsoft SQL Server
  - SAP HANA
- DataSourceマッピングの設定

接続先の種類

intra-mart Accel Platform では、以下の3種類のデータベースに接続できます。

- システムデータベース

  システムのデータを保存するデータベースです。
  アプリケーション起動時にあらかじめ接続可能となっている必要があります。
  システムデータベースはシステム内部で利用されるため、本番環境等ではアプリケーションの
  接続先としては推奨しません。

- テナントデータベース

  テナント内で利用するデータを保存するデータベースです。

- シェアードデータベース

  intra-mart Accel Platform 外のデータを保存するデータベースです。
  外部システムと連携したい場合等に利用します。

> **注意**
>
> ViewCreator/TableMaintenance等で利用できる データベース はテナントデータベースとシェアードデータベースが対象です。
>
> システムデータベースとテナントデータベースを同じ接続先として設定することは可能ですが、 システムデータベースとして作成されるテーブルの利用はサポート対象外です。

DataSourceの設定

1. 「ProjectNavigator」内の <（プロジェクト名）/resin-web.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <web-app>/<database>/<driver> に接続先のデータベース接続情報を設定します。
   以下は各データベース毎の接続例です。

> **コラム**
>
> **バーチャルテナントによる複数テナント の場合**
> テナント数分、DataSourceを準備してください。
> 各テナント毎に利用するDataSourceを設定してください。

コラム

弊社では、Resin 4.0.56 以降で PostgreSQL, Oracle, Microsoft SQL Server の設定として**ConnectionPoolDataSource** の利用を推奨します。

Resin 4.0.55 以前では、**java.sql.Driver** を利用してください。
Resin 4.0.56 以降で、**java.sql.Driver** を利用することも可能です。

注意

resin-web.xml ではプリペアドステートメントキャッシュに関する以下の設定の初期値として 0 が設定されています。

- database/prepared-statement-cache-size
- database/driver/preparedStatementCacheQueries

これは、以下の事象を回避するための設定値です。

Resin で PreparedStatement の キャッシュサイズに大きな値を指定している場合にテナント環境セットアップが失敗する

プリペアドステートメントキャッシュ設定を 0 に指定している場合、データベースに問い合わせを行う際のパフォーマンスが低下します。
テナント環境セットアップ後は、適切にプリペアドステートメントキャッシュ設定を行ったデータソースを利用することを推奨します。

PostgreSQL

次のフォーマットを利用して、データベースURLを指定します。

```
jdbc:postgresql://<host>:<port>/<databasename>
```

- Resin 4.0.56 以降をご利用の場合の設定例は、以下のとおりです。

```
<driver>
  <type>org.postgresql.ds.PGConnectionPoolDataSource</type>
  <url>jdbc:postgresql://localhost:5432/iap_db</url>
  <user>imart</user>
  <password>imart</password>
  <preparedStatementCacheQueries>0</preparedStatementCacheQueries>
</driver>
```

- Resin 4.0.55 以前をご利用の場合の設定例は、以下のとおりです。
  - Version 9.4-1202 以降

```
<driver>
  <type>org.postgresql.Driver</type>
  <url>jdbc:postgresql://localhost:5432/dbname</url>
  <user>username</user>
  <password>password</password>
  <init-param>
    <param-name>preparedStatementCacheQueries</param-name>
    <param-value>20</param-value>
  </init-param>
</driver>
```

  - Version 9.4-1201 以前

```
<driver>
  <type>org.postgresql.Driver</type>
  <url>jdbc:postgresql://localhost:5432/dbname</url>
  <user>username</user>
  <password>password</password>
</driver>
```

コラム

java.sql.Driver を利用する場合 JDBC ドライバのバージョンによって設定が異なります。
Version 9.4-1202 以降では <init-param> に **preparedStatementCacheQueries** を設定する必要があります。
詳しくは、「設定ファイルリファレンス」の「プリペアドステートメントキャッシュ設定」を参照してください。

Oracle

システム識別子（SID）で接続する場合は次のフォーマットを利用して、データベースURLを指定します。

```
jdbc:oracle:thin:@<host>:<port>:<SID>
```

以下は、記述例です。

```
<driver>
  <type>oracle.jdbc.pool.OracleConnectionPoolDataSource</type>
  <url>jdbc:oracle:thin:@localhost:1521:orcl</url>
  <user>username</user>
  <password>password</password>
</driver>
```

サービス名で接続する場合は次のフォーマットを利用して、データベースURLを指定します。

```
jdbc:oracle:thin:@//<host>:<port>/<service>
```

- Resin 4.0.56 以降をご利用の場合の設定例は、以下のとおりです。

```
<driver>
  <type>oracle.jdbc.pool.OracleConnectionPoolDataSource</type>
  <url>jdbc:oracle:thin:@//localhost:1521/orcl</url>
  <user>username</user>
  <password>password</password>
</driver>
```

- Resin 4.0.55 以前をご利用の場合の設定例は、以下のとおりです。

```
<driver>
  <type>oracle.jdbc.driver.OracleDriver</type>
  <url>jdbc:oracle:thin:@//localhost:1521/orcl</url>
  <user>username</user>
  <password>password</password>
</driver>
```

Microsoft SQL Server

次のフォーマットを利用して、データベースURLを指定します。

```
jdbc:sqlserver://<host>:<port>;DatabaseName=<databasename>
```

- Resin 4.0.56 以降をご利用の場合の設定例は、以下のとおりです。

```
<driver>
  <type>com.microsoft.sqlserver.jdbc.SQLServerConnectionPoolDataSource</type>
  <url>jdbc:sqlserver://localhost:1433;DatabaseName=databasename</url>
  <user>username</user>
  <password>password</password>
  <selectMethod>cursor</selectMethod>
  <sendTimeAsDatetime>false</sendTimeAsDatetime>
</driver>
```

- Resin 4.0.55 以前をご利用の場合の設定例は、以下のとおりです。

```
<driver>
  <type>com.microsoft.sqlserver.jdbc.SQLServerDriver</type>
  <url>jdbc:sqlserver://localhost:1433;DatabaseName=databasename</url>
  <user>username</user>
  <password>password</password>
  <init-param>
    <param-name>SelectMethod</param-name>
    <param-value>cursor</param-value>
  </init-param>
</driver>
```

SAP HANA

次のフォーマットを利用して、データベースURLを指定します。

```
jdbc:sap://<host>:<port>?currentschema=<schemaname>
```

以下は、記述例です。

```
<driver>
 <type>com.sap.db.jdbc.Driver</type>
 <url>jdbc:sap://localhost:30015?currentschema=schemaname</url>
 <user>username</user>
 <password>password</password>
</driver>
```

> **コラム**
>
> システムデータベースとテナントデータベースを別々の接続先として設定する場合、JNDI名は違うものを指定してください。

DataSourceマッピングの設定

1. 「ProjectNavigator」内の <（プロジェクト名）/conf/data-source-mapping-config.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。

2. DataSourceとして設定したJNDI名を指定します。
   システムデータベースとテナントデータベース毎に設定したDataSourceを指定します。
   - WebSphere Application Server 9.0.5 の場合

     <data-source-mapping-config>/<system-data-source>/<resource-ref-name> にシステムデータベースのJNDI名を設定します。

     ```
     <system-data-source>
      <resource-ref-name>jdbc/default</resource-ref-name>
     </system-data-source>
     ```

     <data-source-mapping-config>/<tenant-data-source>/<resource-ref-name> にテナントデータベースのJNDI名を設定します。

     ```
     <tenant-data-source>
      <tenant-id>default</tenant-id>
      <resource-ref-name>jdbc/default</resource-ref-name>
     </tenant-data-source>
     ```

     > **注意**
     >
     > WebSphere Application Server 9.0.5 の場合、<resource-ref-name> には「java:comp/env/」を除去したJNDI名を指定する必要があります。

   - WebSphere Application Server 9.0.5 以外の場合

     <data-source-mapping-config>/<system-data-source>/<resource-ref-name> にシステムデータベースのJNDI名を設定します。

     ```
     <system-data-source>
      <resource-ref-name>java:comp/env/jdbc/system</resource-ref-name>
     </system-data-source>
     ```

     <data-source-mapping-config>/<tenant-data-source>/<resource-ref-name> にテナントデータベースのJNDI名を設定します。

     ```
     <tenant-data-source>
      <tenant-id>default</tenant-id>
      <resource-ref-name>java:comp/env/jdbc/tenant</resource-ref-name>
     </tenant-data-source>
     ```

> **注意**
>
> <tenant-id> には、初回で作成するテナントIDを設定してください。

注意

Oracle WebLogic Server 12c R2(12.2.1), WebSphere Application Server 9.0.5 ではシステムデータベースとテナントデータベースは同一のデータベースを指定してください。
詳細は以下を参照してください。

- Oracle WebLogic Server 12c R2(12.2.1) では、システムデータベースとテナントデータベースは同一のデータベースを指定してください。
- WebSphere Application Server 9.0.5 では、システムデータベースとテナントデータベースは同一のデータベースを指定してください。

Storage

Storage 領域として利用するパスを指定します。

1. 「ProjectNavigator」内の 「（プロジェクト名）/conf/storage-config.xml」 ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <storage-config>/<storage-info>/<root-path-name> に Storage 領域のパスを設定します。

```
<root-path-name>${resin.home}/storage</root-path-name>
```

注意

必ず適切なパスに変更してください。
Storage 領域には、システムを運用するために必要不可欠な情報が保存されます。
/tmp 等、OS により意図せず削除される可能性のあるパスは指定しないでください。
また、デフォルト値として指定されている ${resin.home}/storage は、Resin のバージョンアップ時に誤って削除してしまう可能性が考えられます。
そのため、適切なパスに変更していただくことを推奨します。

- テナント毎に利用するストレージ（パブリックストレージ）については設定ファイルでの指定と画面からの指定で、パスが異なります。
設定ファイルで指定した場合は 「<root-path-name>/<public-directory-name>/<storage-directory-name>/テナントID」です。
後述する画面から指定する場合は 「画面から指定したストレージパス/<storage-directory-name>」です。

コラム

WARファイルによる複数テナントの場合
テナント数分、Storage領域を準備してください。
各テナント毎に利用するStorage領域を設定してください。

コラム

分散環境を構築する場合
Storageのルートディレクトリに指定するパスは全て同じ共有ディレクトリを参照するように設定してください。
サーバ毎に参照先が異なる場合、Storageに配置したファイルが共有できません。

コラム

分散環境のサーバをWindowsサービスに登録する場合
共有ディレクトリのパスはUNC形式で指定してください。
ドライブレターで指定した場合、Windowsサービスから起動を行った際に共有ディレクトリにアクセスできません。

SessionTimeOut

最終操作からタイムアウト時間（分）の設定を行います。

1. 「ProjectNavigator」内の <（プロジェクト名）/resin-web.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <web-app>/<session-config>/<session-timeout> にタイムアウト時間（分）を指定します。

```
<session-config>
  <session-timeout>30</session-timeout>
</session-config>
```

注意

WARファイルによる複数テナント の場合
　各テナント毎にタイムアウト時間（分）を設定してください。

注意

バーチャルテナントによる複数テナント の場合
　各テナント毎にタイムアウト時間（分）を設定する事はできません。
　セッションタイムアウトは、Webアプリケーション （WARファイル）単位による機能です。

Locale

利用する言語を設定します。
intra-mart Accel Platform は日本語（ja）、英語（en）、中国語（簡体字）（zh_CN）の3言語が利用できます。

1. 「ProjectNavigator」内の「juggling.im」をダブルクリックします。
2. 「設定ファイル」タブより、「国際化機能」-「ロケールマスタ」を選択し、右側にある「出力」をクリックします。
   出力されると、「ProjectNavigator」内に <locale-config/im-locale-default.xml> ファイルが表示されます。
3. 「ProjectNavigator」内の <locale-config/im-locale-default.xml> ファイルをダブルクリックで開き、
   「ソース」タブを選択してください。
4. <locale-config>/<locale> 内を編集します。

```
<locale name="en" default="true">
 <encoding-name>UTF-8</encoding-name>
</locale>
<locale name="ja">
 <encoding-name>UTF-8</encoding-name>
</locale>
<locale name="zh_CN">
 <encoding-name>UTF-8</encoding-name>
</locale>
```

コラム

例として日本語（ja）のみで運用する場合の設定内容を説明します。

```
<?xml version="1.0" encoding="UTF-8"?>
<locale-config
   xmlns="http://intra_mart.co.jp/system/i18n/locale/config/locale-config"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://intra_mart.co.jp/system/i18n/locale/config/locale-config locale-config.xsd ">
 <locale name="ja">
  <encoding-name>UTF-8</encoding-name>
 </locale>
</locale-config>
```

注意

intra-mart Accel Platform ではこの設定変更を行っても、後述の「テナント環境セットアップ」実行時では、3言語分のデータがインポートされます。
ただし、画面上では選択した言語のみが利用できる仕組みです。

注意

運用開始後にシステムロケールを **追加する** 場合には制約があります。

運用開始後にシステムロケールを追加した場合、そのまま運用を再開すると、マスタデータ・トランザクションデータともに不整合が発生します。以下のドキュメントを参照し、追加したロケール分のデータを補完してください。

- 言語追加ガイド
- 国際化支援機能仕様書

なお、運用開始後にシステムロケールを **削除する** 事は推奨していません。

JavaMail

intra-mart Accel Platform のメール送信に関する設定です。

「ProjectNavigator」内の <（プロジェクト名）/conf/javamail-config/javamail-config.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。

「javamail-config.xml」の設定内容については「設定ファイルリファレンス」 - 「メール設定」を参照してください。

> **コラム**
>
> 2022 Spring(Eustoma) 以降のバージョンでは、セットアップ完了後、システム管理者画面から、メール送信に関する設定を変更、追加できます。
>
> - 「システム管理者操作ガイド」 - 「SMTPサーバ設定」
> - 「*SMTP認証で OAuth2.0 アクセストークン を使用する*」

分散・Webサーバ構成時の設定

Network

intra-mart Accel Platform のネットワーク構成を設定します。

1. 「ProjectNavigator」内の <conf/network-agent-config.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <network-agent-config>/<TCP>/<initial-hosts>/<host> に Resin のIPアドレスを指定します。
   分散構成の場合は、対象の Resin の数分追加してください。
3. <network-agent-config>/<id> にクラスタリングIDを指定します。
   マルチテナント環境 を構築する場合はテナント単位でこのクラスタリングIDを指定する必要があります。
   分散構成の場合は同一のクラスタリングIDを指定します。
4. <network-agent-config>/<bind-port> にクラスタリングを行う際の通信用ポート番号を指定します。
   マルチテナント環境 を構築する場合はポート番号が衝突しないようポート番号を分ける必要があります。
5. <network-agent-config>/<port-range> にポートレンジを指定します。
   このポートレンジは、<network-agent-config>/<bind-port>により指定されたポート番号が既に利用されていた場合
   設定値のレンジだけ代替のポート番号を検索、利用します。
6. <network-agent-config>/<max-threads> および <network-agent-config>/<max-oob-threads> にサーバ間通信で利用するスレッドの最大数を指定します。
   分散環境を構成するサーバ台数が多い場合、サーバ間通信に利用するスレッド数が足りないと正常にクラスタリングできない可能性があります。
   <max-threads> および <max-oob-threads> の値は「サーバ台数 * 2」を目安に設定してください。

```
<id>prototype</id>
<bind-port>5200</bind-port>
<port-range>2</port-range>
<preferIPv4Stack>true</preferIPv4Stack>
<!--
<max-threads>10</max-threads>
<max-oob-threads>10</max-oob-threads>
 -->

<TCP>
 <initial-hosts>
   <host address="127.248.100.1"/>
   <host address="127.248.100.2"/>
   <host address="127.248.100.3"/>
 </initial-hosts>
</TCP>
```

コラム

- <conf/network-agent-config.xml> ファイルのアドレス指定にはホスト名を設定することも可能です。
- ホスト名に紐づくIPアドレスが変わった場合は Resin サーバを再起動してください。

```
<TCP>
 <initial-hosts>
  <host address="app0-host"/>
  <host address="app1-host"/>
  <host address="app2-host"/>
 </initial-hosts>
</TCP>
```

コラム

- <conf/network-agent-config.xml> ファイルの通信プロトコルには UDP を指定することも可能です。
  マルチキャストが利用可能な環境では UDP を利用することでサーバ間通信を高速に行えます。
- UDP を指定する場合は <TCP> の設定を削除し以下のように設定してください。

```
<UDP>
 <mcast-address>228.10.10.10</mcast-address>
 <mcast-port>45588</mcast-port>
 <initial-members>3</initial-members>
</UDP>
```

ベースURL

サーバコンテキスト設定の ベースURL を設定します。

1. 「ProjectNavigator」内の <conf/server-context-config.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <server-context-config>/<base-url> に intra-mart Accel Platform の ベースURL を指定します。

コラム

ベースURLとは？

intra-mart Accel Platform のシステムを外部から参照する際に利用される基底のURLです。
一般的には、 *http://example.org/imart* 等に設定されています。

このベースURLは、あくまでも外部から参照される際に利用するURLです。
従って、localhostや、127.0.0.1（ループバックアドレス）等をURLとして指定した場合、外部からの接続時に適切にサーバに接続されなくなります。

注意

ベースURL が設定されていない場合、次の制限が発生します。

- IM-Workflow を利用する場合

  メール定義 および IMBox定義 の置換文字列 「IM_BaseURL」 / 「IM_URL」 / 「Matter_Detail_URL」 の置換が行われません。

注意

運用環境で intra-mart Accel Platform を利用する場合、必ず Web Server 経由で Web Application Server へ接続する必要があります。
そのため、ベースURL の設定は必須です。

モジュール別の設定

レスポンスヘッダ設定

Webモジュールが提供するレスポンスヘッダ設定です。
intra-mart Accel Platform に対してリクエストを送信した際のレスポンスに任意のヘッダを追加する設定です。
レスポンスヘッダ設定の詳細は、「 設定ファイルリファレンス 」-「 レスポンスヘッダ設定 」を参照してください。

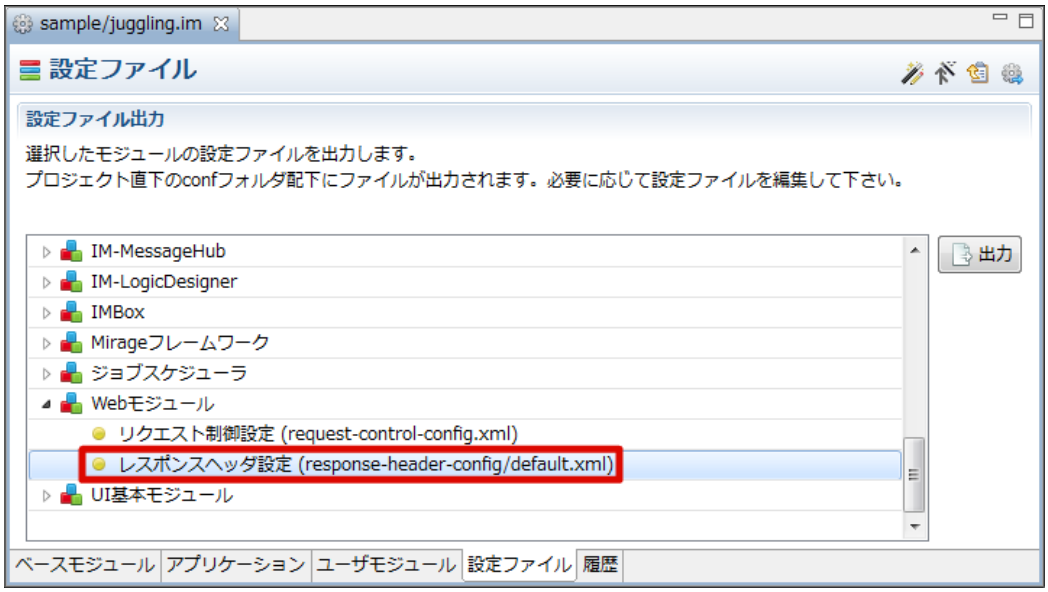
> **注意**
> レスポンスヘッダに依存する処理が存在する場合、設定を追加することでアプリケーションが正常に動作しない可能性があります。
> レスポンスヘッダを変更する際には、十分な検証の上で適用してください。

項目

- 設定ファイルの変更方法
- intra-mart Accel Platform が推奨する設定
  - X-Content-Type-Options

設定ファイルの変更方法

レスポンスヘッダ設定は、初期状態では、IM-Juggling プロジェクトに存在しません。
レスポンスヘッダ設定を変更するには IM-Juggling の「設定ファイル」タブから設定ファイルの出力を実施してください。
「Webモジュール」-「レスポンスヘッダ設定 (response-header-config/default.xml)」を選択して「出力」をクリックすることで
「ProjectNavigator」内の <（プロジェクト名）/conf/response-header-config/default.xml> ファイルが出力されます。



intra-mart Accel Platform が推奨する設定

X-Content-Type-Options

一部のブラウザでは、Webサーバが返すレスポンスの内容からMIMEタイプを自動判別を行います。
自動判別が行われることによりHTMLではない内容がHTMLとして扱われてしまい、XSS（Cross Site Scripting）を誘発してしまうことがあります。
以下のレスポンスヘッダを付与することにより、ブラウザによるMIMEタイプの自動判別を抑制することが可能です。

```
X-Content-Type-Options: nosniff
```

このレスポンスヘッダを常に付与するために「レスポンスヘッダ設定」に以下の「静的なヘッダ指定」設定を追加します。
この設定は初期状態でコメントアウトされています。設定を有効にするためにコメントを解除してください。

```
<static-response-hader name="X-Content-Type-Options" value="nosniff" />
```

> **コラム**
> 2016 Spring(Maxima) より、上記レスポンスヘッダ「X-Content-Type-Options: nosniff」を設定した状態で各機能の検証を行っています。

> **注意**
> 2015 Winter(Lydia) 以前のバージョンをご利用の場合、以下の不具合により本設定を行うと一部の機能が動作しなくなることが確認されています。
> https://issue.intra-mart.jp/issues/22558
> そのため 2015 Winter(Lydia) 以前のバージョンをご利用の場合は本設定を行わないでください。

LDAP認証設定ファイル（アカウントの認証にLDAP認証を利用する場合）

intra-mart Accel Platform でアカウントの認証にLDAP認証を利用する方法を解説します。

項目

- LDAP認証モジュールの機能
- LDAP認証モジュールの利用
- LDAP認証設定ファイル
- LDAP認証でSSL接続(LDAPS)を利用するための環境設定

LDAP認証モジュールの機能

アカウントの認証時に入力されたユーザIDおよびパスワードを利用してLDAPサーバに対して認証を行うモジュールです。

LDAP認証モジュールでは、以下の機能を提供します。

- 設定したLDAPサーバのディレクトリ配下に存在するユーザを検索して認証ができます。（複数 OU 対応）
- 複数のLDAPサーバを設定できます。
  - LDAPサーバがダウンしてる場合に、順次設定されたLDAPサーバに問合せ先を切り替えます。
  - LDAPサーバで認証が失敗した場合、順次設定されたLDAPサーバに問合せ先を切り替えることも可能です。

注意

LDAPサーバのユーザIDと同じユーザIDのアカウントが intra-mart Accel Platform にも必要です。

- パスワード以外の情報は intra-mart Accel Platform のアカウント情報を利用します。

注意

LDAPサーバに Active Directory を利用する場合は、LDAPサーバでユーザの検索を行うための検索用LDAPユーザの設定が必要です。

注意

intra-mart Accel Platform 2013 Winter(Felicia) までのバージョン

- LDAP設定の内容はこのファイルを参照し、利用します。

intra-mart Accel Platform 2014 Spring(Granada) 以降のバージョン

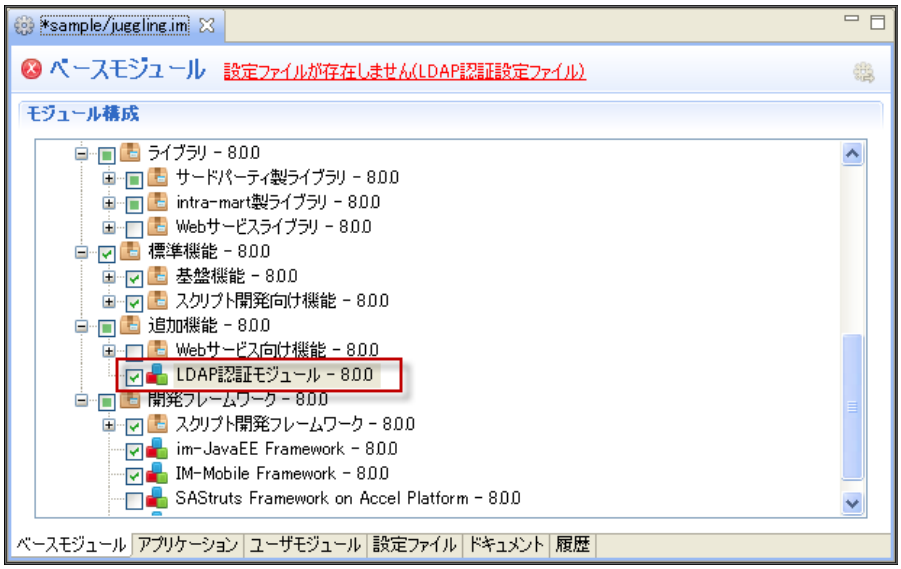
- LDAP設定の内容は、テナント毎に管理されます。
  このLDAP設定ファイルの内容は、テナントの作成時にテナント毎の設定のひな形として利用されます。

- テナントの作成時、画面に表示される設定内容はこの設定ファイルをひな形として表示し、かつ、LDAP認証機能利用設定の値は必ずfalseです。
  テナントにおいてLDAP認証を有効にする場合は、画面に表示される設定内容のLDAP認証機能利用設定の値をtrueにしてください。
  また、画面に表示される設定内容を必要に応じて、編集してください。

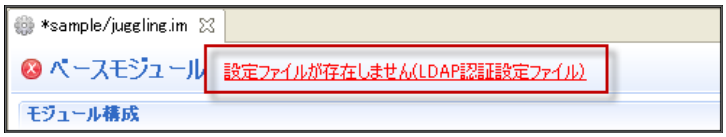
LDAP認証モジュールの利用

LDAP認証モジュールを利用する場合は、以下の手順でモジュールを追加します。

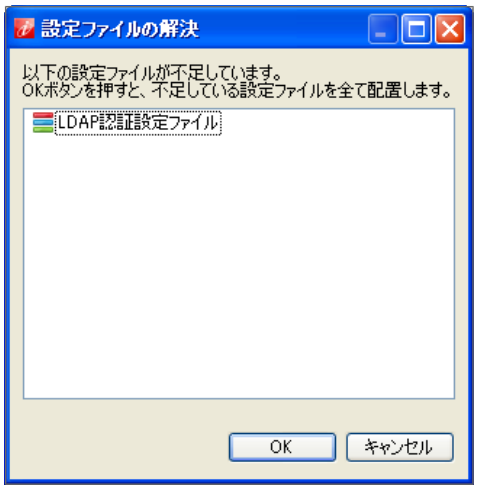
1. IM-Juggling で LDAP認証モジュールを選択してください。



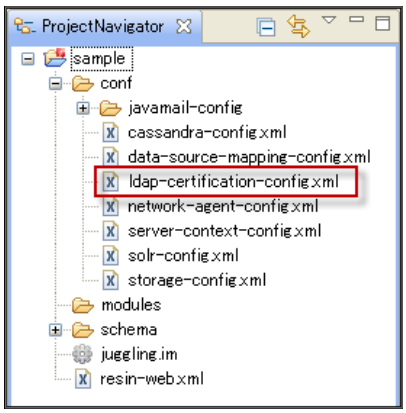
2. LDAP認証モジュールを選択すると、エディタ上部に警告が表示されます。
   [設定ファイルが存在しません（LDAP認証設定ファイル）]のリンクをクリックします。



3. 「設定ファイル解決」ダイアログが表示されますので、[OK]をクリックします。
   LDAP認証設定ファイル(ldap-certification-config.xml)がプロジェクトのconfに追加されます。



4. 追加されたldap-certification-config.xmlをエディタで編集します。



LDAP認証設定ファイル

内容については、「 LDAP認証設定ファイル 」の説明を参照してください。

LDAP認証でSSL接続(LDAPS)を利用するための環境設定

LDAP認証にSSL接続(LDAPS)を利用する場合は、 intra-mart Accel Platform にLDAPサーバのSSL証明書を登録し、利用出来るようにする設定が必要です。
intra-mart Accel Platform が LDAPサーバのクライアントとして接続を可能とする設定です。

1. LDAPサーバへ設定したSSL証明書ファイルを取得します。
2. 取得したファイルを %WORK_FOLDER%にコピーします。
   - %WORK_FOLDER%は作業用の任意のフォルダです。
3. javaのデフォルトキーストアを %WORK_FOLDER% に コピーします。
   - %JAVA_HOME%/jre/lib/security/cacerts が デフォルトキーストアファイルです。
4. コンソール（Windowsならばコマンドプロンプト）で %WORK_FOLDER% に移動します。
5. keytoolを利用して、SSL証明書をキーストアに登録します。

```
keytool -import -storepass changeit -keystore cacerts -alias {任意の名前} -file {SSL証明書ファイル}
```

> **コラム**
> キーストアファイルのデフォルトパスワードはchangeitです。
> 変更されている場合は、変更したパスワードを指定してください。

6. 証明書が追加されたキーストアファイルを intra-mart Accel Platform のサーバの任意フォルダにコピーします。
   - コピーしたフォルダを %CACERTS_FOLDER% とします。
7. Web Application Server の起動時にキーストアファイルのパスを登録する設定を行います。
   - Web Application Server の JavaVM引数に以下を追加します。

【追加するJavaVM引数】

```
-Djavax.net.ssl.trustStore=%CACERTS_FOLDER%/cacerts
```

【JavaVM引数の例】

- %CACERTS_FOLDER% が C:/trust_store

```
-Djavax.net.ssl.trustStore=C:/trust_store/cacerts
```

> **注意**
> 各Web Application Server によりJavaVM引数の設定方法は異なりますので、それぞれの設定方法に従って設定してください。

SAStruts

SAStruts用設定ファイル（SAStruts+S2JDBCにてデータベースを利用する場合）

SAStruts+S2JDBCフレームワークを利用する場合の設定の変更、確認を行います。

- app.diconの編集を行います。
  1. s2jdbc.dicon のコメントアウトをはずして、include を有効にします。

```
<include path="convention.dicon"/>
<include path="aop.dicon"/>
<include path="j2ee.dicon"/>
<include path="s2jdbc.dicon"/>
```

> **注意**
> s2jdbc.diconにてどのデータベースを利用するかを定義する dialect の設定が必要です。
> 接続したいデータベースに合わせて <property name=”dialect”>の定義をコメントの外に出して有効にしてください。
> 利用するデータベースに対してどのdialectを設定すれば良いかは、 http://s2container.seasar.org/2.4/ja/s2jdbc_setup.html を参照してください。

SAStruts用設定ファイル（SAStruts版ポートレットを利用する場合）

SAStruts版のポートレットを利用する場合、設定ファイルの変更が必要です。

- ＜struts-config.xml＞ファイルの編集を行います。
  1. 設定ファイルを「ProjectNavigator」上に追加します。
     - 次の方法があります。
       ・ IM-Juggling プロジェクトを作成するウィザード中の追加リソースの配置より、「SAStruts用設定ファイル」を追加します。
       ・ IM-Juggling プロジェクト作成後に「追加リソースの選択」より「SAStruts用設定ファイル」を追加します。
  2. 「ProjectNavigator」内の <struts-config.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
  3. 次の設定を変更します。

     変更前 ： processorClass=”jp.co.intra_mart.framework.extension.seasar.struts.action.IMS2RequestProcessor”
     変更後 ： processorClass=”jp.co.intra_mart.framework.extension.seasar.struts.portlet.action.IMS2RequestProcessor”

TERASOLUNA Server Framework for Java (5.x)

TERASOLUNA Server Framework for Java (5.x) 用設定ファイル（シェアードデータベースを利用する場合）

TERASOLUNA Server Framework for Java (5.x) より intra-mart Accel Platform のシェアードデータベースを利用する場合、下記設定ファイルにシェアードデータベースの登録が必要です。

- ＜classes/META-INF/spring/applicationContext-im_tgfw_common.xml＞ファイルの編集を行います。
  1. applicationContext-im_tgfw_common.xml の下記コメントアウトをはずして、connectIdパラメータのvalue値にシェアードデータベースでユニークなIDとなる「接続ID」を指定してください。

```
<bean id="sharedDataSource" class="jp.co.intra_mart.framework.extension.spring.datasource.SharedDataSource">
  <constructor-arg name="connectId" value="＜接続ID＞" />
</bean>
```

> **コラム**
>
> 「接続ID」は、テナントにシェアードデータベースを登録する際に指定します。
> テナントにシェアードデータベースを登録する方法については、システム管理者操作ガイドのシェアードデータベース設定を参照してください。

> **コラム**
>
> 複数のシェアードデータベースを利用する場合は、beanタグをそれぞれ定義し、beanのid属性値がそれぞれ一意になるように設定してください。

TERASOLUNA Server Framework for Java (5.x) 用設定ファイル（リポジトリ層にMyBatis3を利用する場合）

TERASOLUNA Server Framework for Java (5.x) のリポジトリ層にMyBatis3を利用する場合、下記設定ファイルの確認および変更が必要です。
テナントデータベース用の設定

テナントデータベース用の設定は、＜classes/META-INF/spring/applicationContext-mybatis3.xml＞ファイルに設定されております。
セットアップ方法の詳細は「TERASOLUNA Server Framework for Java (5.x) プログラミングガイド」-「MyBatis-Springの設定」を参照してください。

シェアードデータベースを利用する場合の設定

シェアードデータベースを利用する場合は、applicationContext-im_tgfw_common.xmlおよびMyBatis-Springの設定を編集する必要があります。
セットアップ方法の詳細は「TERASOLUNA Server Framework for Java (5.x) プログラミングガイド」-「シェアードデータベースの利用」を参照してください。

IMBox

IMBox を利用するための IMBox モジュールを説明します。
IMBox で利用する Apache Cassandra の設定方法を解説します。

項目

- IMBox モジュールの機能
- IMBox モジュールの利用
- Cassandraサーバ接続設定ファイル（cassandra-config.xml）
- intra-mart Accel Platform 2014 Spring(Granada)からの変更点

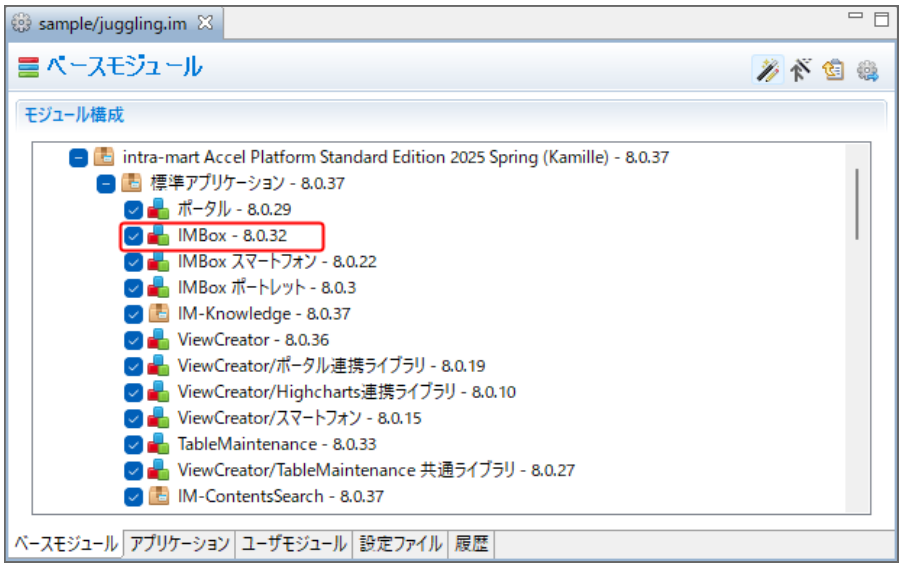
IMBox モジュールの機能

IMBox を利用するためのモジュールです。
IMBox モジュールでは、以下の機能を提供します。

- IMBox を利用することできます。

IMBox モジュールの利用

IMBox モジュールを利用する場合は、IM-Juggling で IMBox モジュールを選択してください。
IM-Juggling のプロジェクト作成時の初期設定では、IMBox モジュールは選択済みです。



> **コラム**
>
> IMBox をスマートフォンで利用する場合は、IMBox スマートフォンモジュールが必要です。
> IMBox をポートレットで利用する場合は、IMBox ポートレットモジュールが必要です。

Cassandraサーバ接続設定ファイル（cassandra-config.xml）

設定内容

Cassandraサーバ接続に関する設定情報を保持するファイルです。

> **注意**
>
> cassandra-configへの設定値は、intra-mart Accel Platform 2014 Spring(Granada)から使用用途が変更されました。
> 詳細は *intra-mart Accel Platform 2014 Spring(Granada) からの変更点* を参照してください。

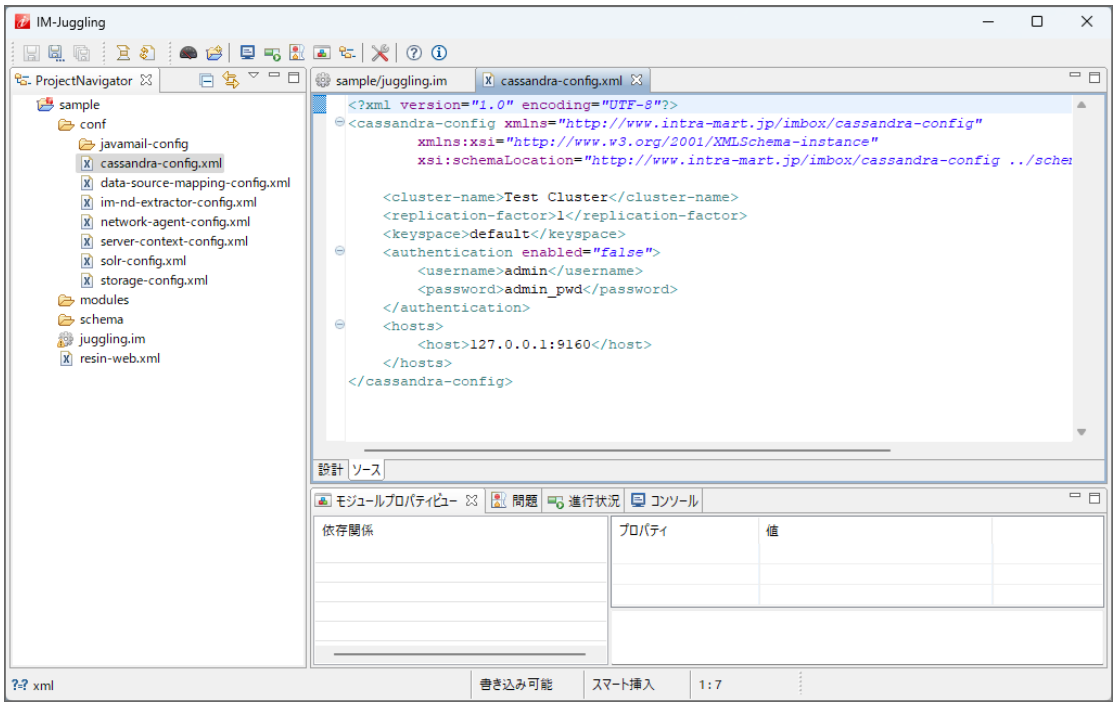
> **注意**
>
> cassandra-config.xmlには、Cassandraサーバの構築時に設定した<%CASSANDRA_HOME%/conf/cassandra.yaml>ファイルの接続情報を指定してください。
> Apache Cassandraの設定に関しての詳細は「 IMBox Cassandra管理者ガイド 」を参照してください。

設定方法

以下は標準で用意されているcassandra-config.xmlの一部です。

```
<cluster-name>Test Cluster</cluster-name>
<replication-factor>1</replication-factor>
<keyspace>default</keyspace>
<authentication enabled="false">
 <username>admin</username>
 <password>admin_pwd</password>
</authentication>
<hosts>
 <host>127.0.0.1:9160</host>
</hosts>
```

- IM-Juggling で設定する場合



設定項目

Cassandraサーバ接続設定ファイルの項目を説明します。

cassandra-config.xmlの要素

タグ	説明	必須	設定値	デフォルト値	複数設定
<cluster-name>	Cassandraサーバのクラスタ名を設定します。 初期値に設定してある “Test Cluster” は <%CASSANDRA_HOME%/conf/cassandra.yaml> ファイルの初期値です。	○	クラスタ名	Test Cluster	×
<replication-factor>	分散構成とするクラスタ内部のデータのレプリカ数を設定します。 レプリカ数の設定は、「Cassandra管理者ガイド」-「Cassandraのクラスタ構築」を参照してください。	○	数値	1	×
<keyspace>	Cassandraサーバのキースペースを設定します。 新規に intra-mart Accel Platform を構築する場合は、初期値の “default” のままで問題ありません。 複数の intra-mart Accel Platform から同じCassandraクラスタを利用する場合は、 intra-mart Accel Platform のテナント毎に異なるキースペースを指定してください。	○	キースペース	default	×
<authentication>	Cassandraへの接続に対する認証情報を設定します。 認証情報の設定はenabled属性の値に “true” を設定した場合のみ有効です。 認証情報を使用する場合、Cassandra自体への接続認証の設定を行う必要があります。 接続認証に関しては、「Cassandra管理者ガイド」-「Cassandraへの接続認証設定」を参照してください	○	なし	なし	×
<username>	接続ユーザ名を設定します。 認証情報を使用する場合のみ設定が適用されます。	認証設定を使用する場合のみ○	接続ユーザ名	admin	×
<password>	接続パスワードを設定します。 認証情報を使用する場合のみ設定が適用されます。	認証設定を使用する場合のみ○	接続パスワード	admin_pwd	×
<hosts>	Cassandraが稼働しているサーバの設定を行います。	○	なし	なし	×

タグ	説明	必須	設定値	デフォルト値	複数設定
<host>	Cassandraが稼働しているサーバのIPアドレスとポート番号の設定を行います。 分散構成で複数のCassandraが稼働している場合、すべてのCassandraサーバのIPアドレスとポート番号を設定してください。	○	IPアドレス:ポート番号	127.0.0.1:9160	○

> **注意**
>
> Cassandraサーバが複数存在する場合、クラスタ名で同一クラスタのCassandraであるか判断され、クラスタが組まれてしまう場合があります。
> 新規に intra-mart Accel Platform を構築する場合、初期値を変更することを推奨します。

intra-mart Accel Platform 2014 Spring(Granada)からの変更点

cassandra-configの設定値は以下のように使用用途が変更されました。

- intra-mart Accel Platform 2013 Winter(Felicia)までの場合
  - Cassandraへの接続の設定値
- intra-mart Accel Platform 2014 Spring(Granada)以降の場合
  - テナント環境セットアップのCassandra接続情報の初期値

> **コラム**
>
> intra-mart Accel Platform 2014 Spring(Granada)からは、テナント管理 - Cassandra接続情報 で登録したCassandra接続情報が接続時に使用されます。

Cassandra接続情報画面の初期値と設定値

Cassandra接続情報の項目名	cassandra-configの設定値	備考
クラスタ名	cluster-name	
キースペース	keyspace	
接続先	host	hostが複数設定されている場合、すべて反映されます。
レプリケーションファクタ	replication-factor	
認証情報設定	authentication	
認証ユーザ名	username	authenticationのenabled属性がtrueの場合のみ表示されます。
認証パスワード	password	authenticationのenabled属性がtrueの場合のみ表示されます。

- Cassandra接続情報画面

IM-ContentsSearch

IM-ContentsSearch for Accel Platform を利用するために必要な設定を行います。

項目

- IM-ContentsSearch の機能
- IM-ContentsSearch モジュールの利用
- IM-ContentsSearch 検索対象追加モジュールの利用
- Solrサーバ接続設定（solr-config.xml）
  - intra-mart Accel Platform 2014 Spring(Granada)からの変更点
  - intra-mart Accel Platform 2024 Autumn(Jasmine)からの変更点
- テキスト抽出設定（solr-extractor-config.xml）
  - intra-mart Accel Platform 2016 Spring(Maxima)からの変更点
  - intra-mart Accel Platform 2023 Autumn(Hollyhock)からの変更点
  - intra-mart Accel Platform 2024 Autumn(Jasmine)からの変更点
- 検索画面設定（contentssearch-display-config.xml）
  - intra-mart Accel Platform 2018 Spring(Skylark)からの変更点

IM-ContentsSearch の機能

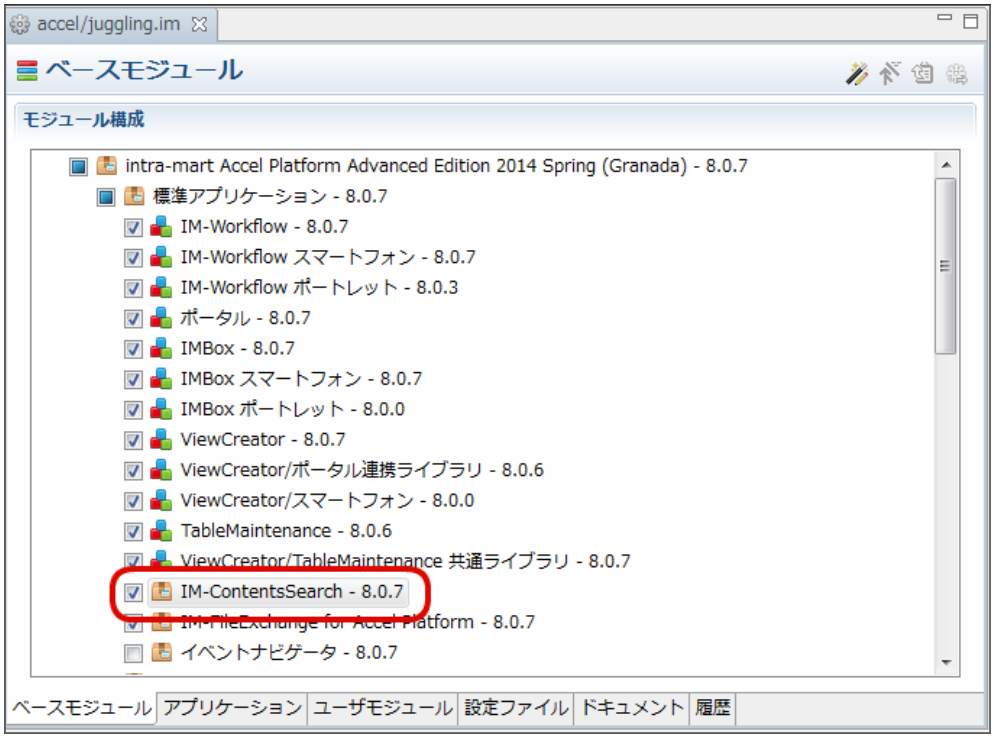
intra-mart Accel Platform で全文検索機能を実現するためのモジュールです。
全文検索機能を利用するためには「Apache Solr」サーバを構築する必要があります。
検索対象を追加するには検索対象追加モジュールを追加する必要があります。

コラム

- Apache Solr のセットアップについては、「Solr管理者ガイド」を参照してください。
- 追加した検索対象のコンテンツを全文検索機能で検索できるようにするためには、検索対象追加モジュールの機能に含まれているジョブをジョブスケジューラ機能で実行する必要があります。
  IM-ContentsSearch のジョブ・ジョブネットについては「ジョブ・ジョブネットリファレンス」を参照してください。

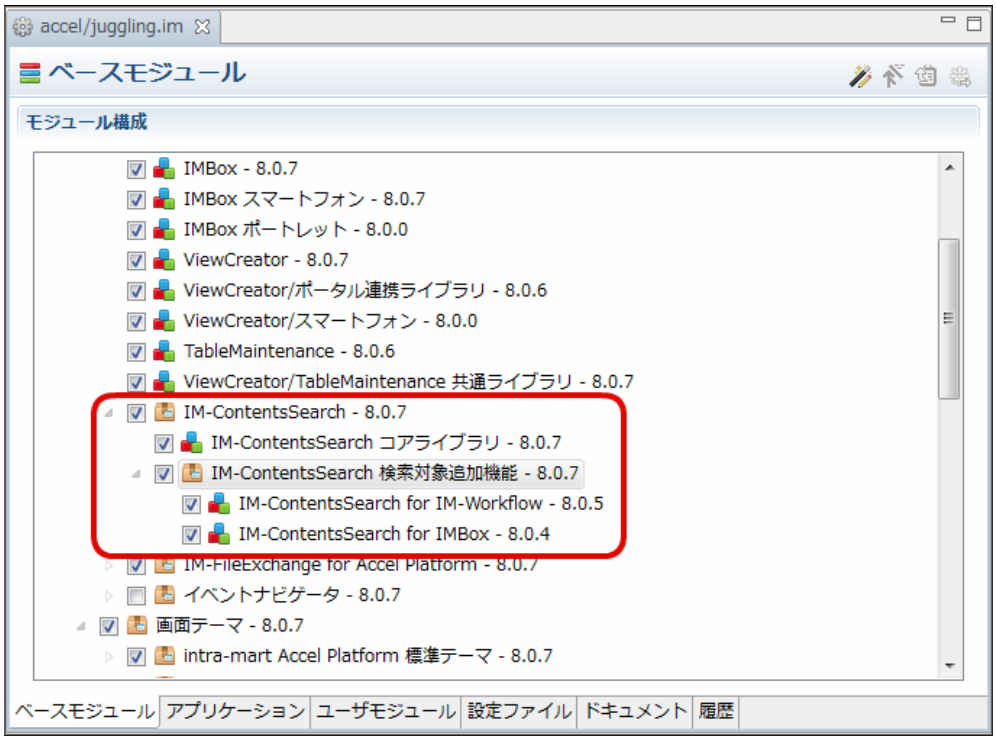
IM-ContentsSearch モジュールの利用

IM-ContentsSearch モジュールを利用する場合は、IM-Juggling で IM-ContentsSearch モジュールを選択してください。
IM-Juggling のプロジェクト作成時の初期設定では、IM-ContentsSearch モジュールは選択済みです。



IM-ContentsSearch 検索対象追加モジュールの利用

検索対象追加モジュールを変更する場合、IM-ContentsSearch > IM-ContentsSearch 検索対象追加機能 のモジュールのツリー配下のモジュールを変更してください。



Solrサーバ接続設定（solr-config.xml）

設定内容

Solrサーバへの接続設定情報を保持するファイルです。

> **注意**
>
> solr-config.xmlの設定値は、intra-mart Accel Platform 2014 Spring(Granada)から使用用途が変更されました。
> 詳細は *intra-mart Accel Platform 2014 Spring(Granada) からの変更点* を参照してください。

設定方法

以下は標準で用意されているsolr-config.xmlの一部です。

```
<group name="default">
  <searcher>
    <method>POST</method>
    <distribution-policy>FIRST</distribution-policy>
    <servers>
      <url>http://localhost:8983/solr/default</url>
    </servers>
  </searcher>
  <indexer>
    <distribution-policy>IDHASH</distribution-policy>
    <servers>
      <url>http://localhost:8983/solr/default</url>
    </servers>
  </indexer>
  <extractor ref="" />
</group>
<group name="default_vector">
  <searcher>
    <method>POST</method>
    <distribution-policy>FIRST</distribution-policy>
    <servers>
      <url>http://localhost:8983/solr/default_vector</url>
    </servers>
  </searcher>
  <indexer>
    <distribution-policy>IDHASH</distribution-policy>
    <servers>
      <url>http://localhost:8983/solr/default_vector</url>
    </servers>
  </indexer>
  <extractor ref="" />
</group>
```

設定項目

solr-config.xmlの要素

タグ	説明	必須	設定値	デフォルト値	複数設定
<group>	Solrサーバグループ名を設定します。 <group>のnameにSolrサーバグループ名を設定します。 初期値の “default” のままで問題ありません。	○	Solrサーバグループの設定	なし	○
<searcher>	IM-ContentsSearchが検索時に接続するSolrサーバを設定します。	○	検索用Solrサーバの設定	なし	×
<method>	IM-ContentsSearchが検索時にSolrサーバへ送るリクエストのメソッドを設定します。 通常はPOSTから変更する必要はありません。	×	検索時に使用するリクエストメソッドの設定	POST	×
<distribution-policy>	IM-ContentsSearchが検索リクエストを送るSolrサーバを選択するポリシーを設定します。 検索用と索引作成用で利用可能なポリシーが異なります。	×	検索用Solrサーバの選択ポリシー設定	FIRST	×
<servers>	IM-ContentsSearchが検索時に接続するSolrサーバ一覧を設定します。 Solrサーバ設定は配下のurlタグで行います。	○	検索用Solrサーバ一覧の設定	なし	×

タグ	説明	必須	設定値	デフォルト値	複数設定
<url>	IM-ContentsSearchが検索時に接続するSolrサーバのURLを設定します。 Solrサーバを構築しているホストのアドレス、および、ポート番号を指定してください。 URL内の「default」は Apache Solr core名であり、Solrサーバグループ名とは別の設定値であるため、変更しないでください。 参考： 各Web Application Serverのデフォルトのポート番号 Resin 8080 Tomcat 8080 Jetty 8983	○	検索用Solrサーバ のURLの設定	なし	○
<indexer>	IM-ContentsSearchが索引作成時に接続するSolrサーバを設定します。	○	索引作成用Solr サーバの設定	なし	○
<distribution-policy>	索引作成用サーバを選択するポリシーを設定します。 検索用と索引作成用で利用可能なポリシーが異なります。	×	索引作成用Solr サーバの選択ポリシー設定	IDHASH	×
<servers>	IM-ContentsSearchが索引作成時に接続するSolrサーバ一覧を設定します。 Solrサーバ設定は配下のurlタグで行います。	○	索引作成用Solr サーバ一覧の設定	なし	×
<url>	IM-ContentsSearchが索引作成時に接続するSolrサーバのURLを設定します。 defaultは Apache Solr core名であり、Solrサーバグループ名とは別の設定値であるため、変更しないでください。	○	索引作成用Solr サーバのURLの設定	なし	○
<extractor>	IM-ContentsSearchが索引作成時にテキスト抽出するファイルの設定グループを指定します。 ref属性にsolr-extractor-config.xmlに定義されたテキスト抽出設定グループIDを指定します。 ※指定されたテキスト抽出設定グループが見つからない場合や未指定の場合は、テキスト抽出機能 (ND Universal Extractor) の共通設定 (im-nd-extractor-config.xml) が使用されます。	○	テキスト抽出設定 グループの指定	なし	×

intra-mart Accel Platform 2014 Spring(Granada)からの変更点

intra-mart Accel Platform 2014 Spring(Granada)以降のバージョンをご利用の場合、solr-config.xmlの設定値は以下のように使用用途が変更されました。

- 初回テナント環境セットアップ時、または、Solr接続設定が1件も登録されていない場合のテナント環境セットアップ時に登録されるSolr接続設定
- Solr接続設定新規作成画面の初期値

> **コラム**
> <group>のnameが”default”である設定のみ使用されます。

intra-mart Accel Platform 2024 Autumn(Jasmine)からの変更点

intra-mart Accel Platform 2024 Autumn(Jasmine)以降のバージョンをご利用の場合、標準で用意されているsolr-config.xmlの設定値は以下のように変更されました。

- テキスト抽出設定グループの指定（`extractor` タグの `ref` 属性値）の必須設定が解除されました。
- テキスト抽出設定グループの指定が空値（テキスト抽出機能 (ND Universal Extractor) の共通設定 (im-nd-extractor-config.xml) を利用する設定）に変更されました。
- ベクトルデータベース用のSolrサーバグループの設定が追加されました。

> **コラム**
>
> IM-Copilot for Accel Platform が提供する生成AI活用基盤を使用して Apache Solr をベクトルデータベースとして利用する機能が追加されました。
> Apache Solr をベクトルデータベースとして利用する場合は IM-ContentsSearch のSolrサーバ接続情報が使用されるため、標準で用意されているsolr-config.xmlの設定にベクトルデータベース用のSolrサーバグループの設定が追加されました。

テキスト抽出設定（solr-extractor-config.xml）

> **注意**
>
> intra-mart Accel Platform 2024 Autumn(Jasmine)以降のバージョンでは、標準で用意されているテキスト抽出設定は、テキスト抽出機能 (ND Universal Extractor) の共通設定 (im-nd-extractor-config.xml) に設定されています。
> Solrサーバグループごとに個別のテキスト抽出設定を使用する場合のみ、solr-extractor-config.xmlをご利用ください。

設定内容

以下の内容に関する設定情報を保持するファイルです。

- 共通パラメータの設定
- テキスト抽出対象ファイルの設定
- テキスト抽出方法の設定

設定方法

以下はsolr-extractor-config.xmlの設定例です。

```
<extractor name="im_default">
 <default>
  <min-size>0</min-size>
  <max-size>100M</max-size>
  <properties>
   <!-- temporary directory sample for Linux -->
   <property name="tmpDir" type="string">/tmp</property>
   <!-- temporary directory sample for Windows -->
   <!--
   <property name="tmpDir" type="string">C:\temp</property>
   -->
  </properties>
 </default>

 <mapping>
  <!-- Entry for text files -->
  <entry type="text/plain" extension="txt" max-size="1M"
class="jp.co.nttdata.intra_mart.contentssearch.text.extraction.extractor.PlainTextExtractor" />
  ...
 </mapping>
</extractor>
```

設定項目

solr-extractor-config.xmlに設定する項目を記載します。
一部の要素のみ記載しているため、詳細は 設定ファイルリファレンス テキスト抽出設定 を参照してください。

solr-extractor-config.xmlの要素（一部）

タグ	説明	必須	設定値	デフォルト値	複数設定
<extractor>	一つのテキスト抽出設定グループを表します。 name属性にはテキスト抽出設定グループの名称を指定します。	○	テキスト抽出設定グループ	なし	○
<min-size>	テキスト抽出対象ファイルサイズ最小値の既定値を設定します。	×	テキスト抽出対象ファイルサイズ最小値の既定値	0（制限なし）	×
<max-size>	テキスト抽出対象ファイルサイズ最大値の既定値を設定します。	×	テキスト抽出対象ファイルサイズ最大値の既定値	0（制限なし）	×

タグ	説明	必須	設定値	デフォルト値	複数設定
<property>	プロパティの既定値を設定します。	×	プロパティ既定値の設定	なし	○

> **注意**
>
> 一時ディスク領域 (propertyタグのname属性”tmpDir”) は必ず設定してください。
> 指定するディレクトリは、 intra-mart Accel Platform の実行ユーザからアクセス可能である必要があります。

intra-mart Accel Platform 2016 Spring(Maxima)からの変更点

intra-mart Accel Platform 2016 Spring(Maxima)以降のバージョンをご利用の場合、 **DocuWorks文書** および **DocuWorksバインダー** からテキストを抽出して全文検索の対象にすることが可能になりました。
機能を有効にするためには、以下の設定を行う必要があります。

- *solr-extractor-config.xml* の設定を有効化する。
- 富士フイルムビジネスイノベーション株式会社より提供されている、DocuWorks Content Filter を intra-mart Accel Platform が動作するサーバにインストールする。

> **コラム**
>
> DocuWorks は富士フイルムビジネスイノベーション株式会社の商標です。

設定の有効化方法

2015 Winter(Lydia)以前のバージョンからアップデートした場合、 2016 Spring(Maxima)から追加されたDocuWorks文書に対応するための *solr-extractor-config.xml* の記述方法については、「Accel Documents / IM-ContentsSearch for Accel Documents 仕様書」を参照してください。

DocuWorks Content Filterのインストール方法

DocuWorks Content Filter のインストール方法については、「*DocuWorks Content Filter のインストール方法*」を参照してください。

intra-mart Accel Platform 2023 Autumn(Hollyhock)からの変更点

intra-mart Accel Platform 2023 Autumn(Hollyhock)以降のバージョンをご利用の場合、Microsoft Office ファイルのテキスト抽出設定にシステムプロパティの設定が追加されました。
Microsoft Office ファイルのテキスト抽出に標準のテキスト抽出クラスを使用している場合はシステムプロパティの設定が必要です。
詳細は 「 設定ファイルリファレンス 」 - 「 テキスト抽出設定 」 - 「 システムプロパティで Apache POI の上限値を調整する 」を参照してください。

> **コラム**
>
> intra-mart Accel Platform 2022 Winter(Freesia)または、 intra-mart Accel Platform 2023 Spring(Gerbera)で以下のパッチが適用されている場合も対象です。
> * IM-ContentsSearch コアライブラリ 8.0.22-PATCH_001
> * Apache POI 5.2.3-PATCH_002
> * Solr Client 3.0.2-PATCH_001

intra-mart Accel Platform 2024 Autumn(Jasmine)からの変更点

intra-mart Accel Platform 2024 Autumn(Jasmine)以降のバージョンでは、solr-extractor-config.xmlは、Solrサーバグループごとに個別のテキスト抽出設定を使用するためのオプション設定に変更されました。

> **コラム**
>
> 標準で用意されているテキスト抽出設定は、テキスト抽出機能 (ND Universal Extractor) の共通設定 (im-nd-extractor-config.xml) に移動されました。

検索画面設定（contentssearch-display-config.xml）

設定内容

以下の内容に関する設定情報を保持するファイルです。

- 検索クエリに関する設定
- 検索オプションに関する設定
- 検索結果表示に関する設定

設定方法

以下は標準で用意されているcontentssearch-display-config.xmlの一部です。

```
<?xml version="1.0" encoding="UTF-8"?>
<contentssearch-display-config
xmlns="http://intra-mart.co.jp/system/contentssearch/web/config/contentssearch-display-config"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://intra-mart.co.jp/system/contentssearch/web/config/contentssearch-display-config ../schema/contentssearch-display-config.xsd ">

 <query-settings>
  <max-length>200</max-length>
  <default-match-type>partial</default-match-type>
 </query-settings>

 <search-options>
  ...
 </search-options>

 <view-options>
  ...
 </view-options>

</contentssearch-display-config>
```

設定項目

contentssearch-display-config.xmlに設定する項目を記載します。
一部の要素のみ記載しているため、詳細は 「 設定ファイルリファレンス 」 - 「 検索画面設定 」 を参照してください。

contentssearch-display-config.xmlの要素（一部）

タグ	説明	必須	設定値	デフォルト値	複数設定
<query-settings>	検索クエリに関する設定です。 詳細な設定は配下のタグで行います。	○	検索クエリに関する設定値	なし	×
<max-length>	検索キーワードの最大文字数を設定します。 検索キーワードがこの値を超えた場合、検索処理は実行されずに警告メッセージが表示されます。	×	検索キーワードの最大文字数の設定値	200	×
<default-match-type>	検索方法の初期値の設定です。	×	検索方法の初期値の設定値	partial	×
<search-options>	検索オプションに関する設定です。 詳細な設定は配下のタグで行います。	○	検索オプションに関する設定値	なし	×
<view-options>	検索結果の表示に関するオプションの設定です。 詳細な設定は配下のタグで行います。	○	検索結果の表示に関するオプションの設定値	なし	×

intra-mart Accel Platform 2018 Spring(Skylark)からの変更点

intra-mart Accel Platform 2018 Spring(Skylark)以降のバージョンをご利用の場合、 検索方法の初期値が変更可能になりました。
設定できる検索方法は、以下です。

- 部分一致検索
- 完全一致検索

設定の有効化方法

以下は2018 Spring(Skylark)から追加された検索方法の初期値の設定を対応するための *contentssearch-display-config.xml* の記述です。
2017 Winter(Rebecca)以前のバージョンからアップデートした場合は、<query-settings>タグ配下に下記の内容を追記してください。

```
<?xml version="1.0" encoding="UTF-8"?>
<contentssearch-display-config
xmlns="http://intra-mart.co.jp/system/contentssearch/web/config/contentssearch-display-config"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://intra-mart.co.jp/system/contentssearch/web/config/contentssearch-display-config ../schema/contentssearch-display-config.xsd ">

 <query-settings>
  ...
  <!-- 検索方法の初期値の設定値 -->
  <default-match-type>partial</default-match-type>
 </query-settings>

 <search-options>
  ...
 </search-options>

 <view-options>
  ...
 </view-options>

</contentssearch-display-config>
```

検索方法の初期値を「部分一致検索」に設定する場合は、<default-match-type>を”partial” に変更してください。
検索方法の初期値を「完全一致検索」に設定する場合は、<default-match-type>を”perfect” に変更してください。
詳細は、「 設定ファイルリファレンス 」 - 「 検索画面設定 」-「 検索方法の初期値の設定 」を参照してください。

IM-Workflow

IM-Workflow システム設定

IM-Workflow のシステム共通設定について設定します。
IM-Workflow システム設定

1. 「ProjectNavigator」内の <conf/im-workflow-system-config.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. 設定の内容を運用に応じて変更します。

> **コラム**
>
> IM-Juggling 上に、設定ファイル「IM-Workflow システム設定」がない場合は、以下の手順で設定ファイルを出力してください。
>
> 1. <（プロジェクト名）/juggling.im> ファイルをダブルクリックします。
> 2. 「設定ファイル」タブをクリックします。
> 3. IM-Workflow モジュールをクリックして展開します。
> 4. 展開結果から対象の設定ファイル「IM-Workflow システム設定（im-workflow-system-config.xml）」を選択し、右側にある「出力」をクリックします。
>
> 詳細については「 IM-Workflow 仕様書 」の「5.1.1.1 システム設定」を参照してください。

IM-Workflow デザイナ設定

1. 「ProjectNavigator」内の <conf/im-workflow-designer-config.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. 設定の内容を運用に応じて変更します。

コラム

IM-Juggling 上に、設定ファイル「IM-Workflow デザイナ設定」がない場合は、以下の手順で設定ファイルを出力してください。

1. <（プロジェクト名）/juggling.im> ファイルをダブルクリックします。
2. 「設定ファイル」タブをクリックします。
3. IM-Workflow モジュールをクリックして展開します。
4. 展開結果から対象の設定ファイル「IM-Workflow デザイナ設定（im-workflow-designer-config.xml）」を選択し、右側にある「出力」をクリックします。

注意

<im-workflow-designer-config>/<icon> のアイコンパス定義を標準から変更する場合、または、ノードアイコンの差し替えを行う場合は、ユーザモジュールによるアイコン配置・差し替えが必要です。
定義したアイコンパスが示す場所（システムストレージ配下）に、差し替え後のアイコンがデプロイされるようユーザモジュールを作成し、warにユーザモジュールを含めてください。

ユーザモジュールの作成方法については「 intra-mart e Builder for Accel Platform アプリケーション開発ガイド」を参照してください。

なお、ユーザモジュールではmodule.xmlを編集し、IM-Workflow（モジュールID：jp.co.intra_mart.im_workflow）モジュールを「依存するモジュール」として必ず指定してください。
指定がない場合、アイコンの差し替えが正しく行われない可能性があります。

コラム

モジュール・プロジェクトの作成例を示します。

モジュール・プロジェクト作成例の前提

プロジェクト名（アーティファクトID）	workflow_user_icon_replace
グループID	mypackage

- module.xml の例

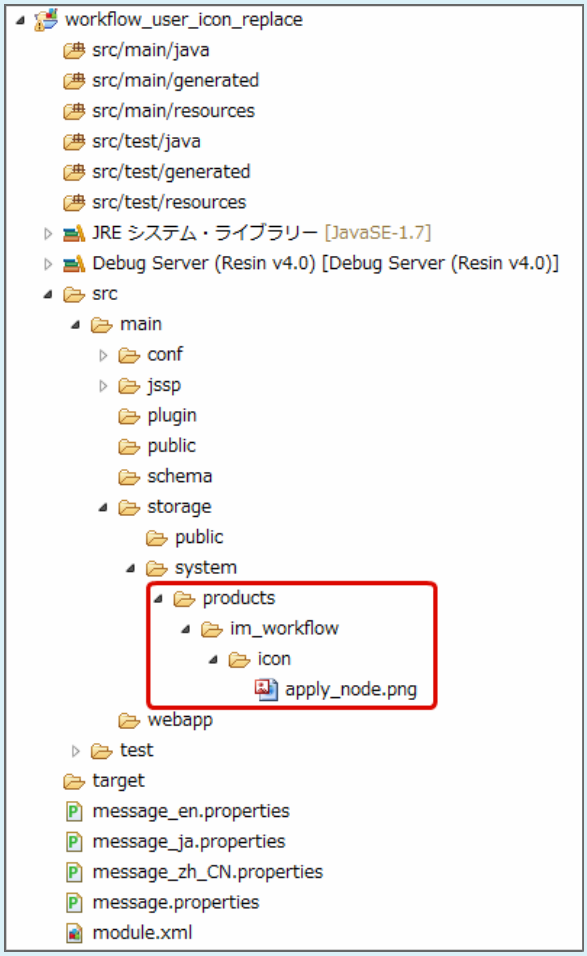
  IM-Workflow（モジュールID：jp.co.intra_mart.im_workflow）モジュールを「依存するモジュール」として指定します。

```
<module xmlns="urn:intramart:jackling:module" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:conf="urn:intramart:jackling:toolkit:configurations"
  xsi:schemeLocation="urn:intramart:jackling:module module.xsd"
  conf:schemaLocation="urn:intramart:jackling:toolkit:configurations configurations.xsd">

  <id>mypackage.workflow_user_icon_replace</id>
  <version>1.0.0</version>
  <type>module</type>

  <name>${module.name}</name>
  <vendor>${module.vendor}</vendor>
  <description>${module.description}</description>

  <!-- 変更を不可とする場合やサードパーティモジュールの場合のみ以下を指定する -->
  <tags>
    <tag>immutable</tag>
    <tag>3rd-party</tag>
  </tags>

  <dependencies>
      <dependency>
          <module-id>jp.co.intra_mart.im_workflow</module-id>
          <verified-version min="8.0.7">8.0.7</verified-version>
      </dependency>
  </dependencies>

</module>
```

- モジュール・プロジェクトのリソース配置例 1

  <im-workflow-designer-config>/<icon> のアイコンパス定義は標準のままとし、申請ノードアイコンを差し替える場合

  差し替えるアイコンのみを配置します。



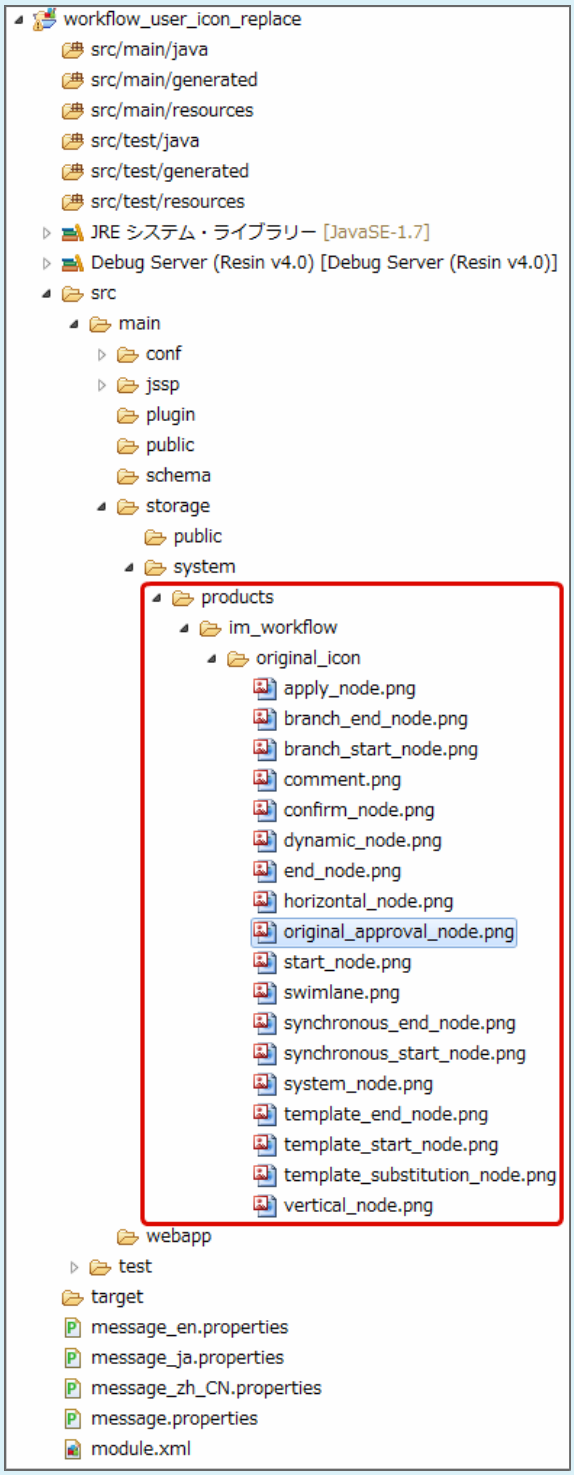
- モジュール・プロジェクトのリソース配置例 2

  <im-workflow-designer-config>/<icon> のアイコンパス定義の変更を以下の通りに指定した場合
  ※記載のないパラメータについては、標準の値から変更なしの状態とする。

アイコンパス定義の設定

node-icon-dir	products/im_workflow/original_icon/
approval-node-icon	original_approval_node.png

  ノードアイコン保存ディレクトリを変更しているため、変更後のディレクトリにすべてのノードアイコンを配置します。
  また、承認ノードについては、変更後のアイコンファイル名と合致する承認ノードアイコンファイルを配置します。



IM-Workflow 用設定ファイル （シーケンスオブジェクト採番機能を利用する場合）

シーケンスオブジェクト採番機能を利用する場合、プラグイン設定の変更が必要です。
シーケンスオブジェクト採番機能

IM-Workflow の WorkflowNumberingManager を利用して連続した番号を取得する際にデータベースのシーケンスオブジェクトを利用して採番処理を行う機能です。

> **注意**
> シーケンスオブジェクト採番機能は、 2013 Winter(Felicia) から利用可能です。

シーケンスオブジェクト採番機能の利用

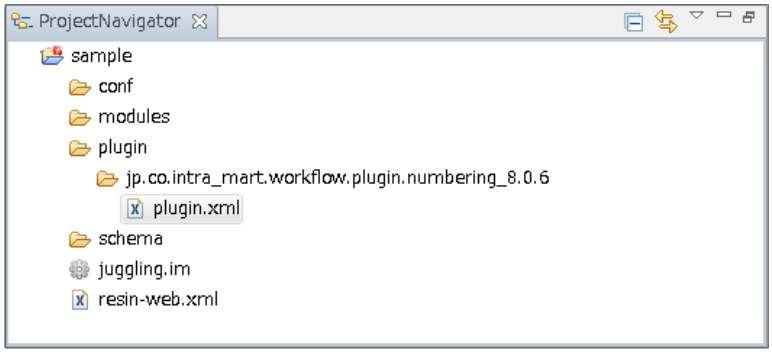
シーケンスオブジェクト採番機能を利用する場合は、以下の手順でプラグインを作成します。

1. IM-Juggling でプロジェクトを選択し、フォルダを作成します。
   フォルダは「plugin/jp.co.intra_mart.workflow.plugin.numbering_8.0.6」にしてください。

2. 作成されたフォルダに「plugin.xml」を作成します。
   「plugin.xml」に以下の内容をコピーしてください。

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin>
 <extension point="jp.co.intra_mart.workflow.plugin.numbering" >
  <numbering
   name="simpleNumberCounter"
   id="jp.co.intra_mart.workflow.plugin.numbering.simpleNumberCounter"
   version="8.0.6"
   rank="0"
   enable="false">
   <extend>
    <java class="jp.co.intra_mart.system.workflow.plugin.numbering.SimpleNumberCounterEvent" />
   </extend>
  </numbering>
  <numbering
   name="databaseSequence"
   id="jp.co.intra_mart.workflow.plugin.numbering.databaseSequence"
   version="8.0.6"
   rank="0"
   enable="true">
   <extend>
    <java class="jp.co.intra_mart.system.workflow.plugin.numbering.DatabaseSequenceEvent" />
   </extend>
  </numbering>
 </extension>
</plugin>
```

- プラグインを作成した後のプロジェクト構成



IM-Workflow 一覧画面を利用する場合の遷移先プラグインに関する設定

IM-Workflow には戻り先を指定できる遷移先プラグイン機能があります。
基本的には、一覧から遷移した処理画面において「戻る」や「申請等の処理」を行った場合は、それぞれ遷移元の一覧へ遷移します。
ただし、遷移元の一覧を特定できないような次のケースにおいては、遷移先プラグイン機能で戻り先を判定します。

- 申請ポートレットの「申請／処理開始画面へ」リンクを押下した際の遷移先
- 参照依頼等のショートカットURLを押下した際の遷移先

使用する一覧を変更する場合は、下記を参照し、遷移先プラグインを変更してください。

- 「 遷移先プラグインの設定 」

> **コラム**
>
> 2019 Winter(Xanadu) 以降のアップデートでは、「申請一覧」および「案件一覧」がデフォルトで設定されています。
> 2019 Summer(Waltz) 以前の一覧画面を利用する場合は上記に従って変更してください。

iPadからアクセス時にクライアントタイプをPCとして扱う場合

intra-mart Accel Platform ではアクセスする端末によりクライアントタイプが割り当てられます。
intra-mart Accel Platform で標準で提供しているクライアントタイプは以下の通りです。

クライアントタイプID	提供バージョン	提供モジュール	備考
pc	2012 Autumn	マルチデバイス	PC端末からの利用を想定しています。（標準）

sp	2012 Autumn	IM-Mobile Framework	スマートフォン、タブレットなどのモバイル端末からの利用を想定しています。

intra-mart Accel Platform では標準設定では iPad がスマートフォン端末として扱われます。
この章では、iPad をPC端末として扱う（スマートフォン端末として扱わない）設定方法を紹介します。

iPadプラグインの無効化

クライアントタイプの判別を行う設定方法としてpluginを採用しています。
pluginの設定仕様については、「PluginManagerのJavaDoc」を参照してください。
以下ではiPadのクライアントタイプを `sp` と判別している設定を無効化します。

1. IM-Juggling でプロジェクトを選択し、フォルダを作成します。
   フォルダは「plugin/ignore_sp_for_ipad」にしてください。
2. 作成されたフォルダに「plugin.xml」を作成します。
   「plugin.xml」に以下の内容をコピーしてください。

```
<?xml version="1.0" encoding="utf-8"?>
<plugin>
  <extension point="jp.co.intra_mart.foundation.multi_device.client_type.matcher">
    <client-type-config id="ipad" version="8.0.999" enable="false" />
  </extension>
</plugin>
```

> **コラム**
> pluginのバージョン機能を利用して、`ipad` に対しての設定を優先的に無効化します。

自動ログイン機能を利用する場合

intra-mart Accel Platform は、ログイン実行ページURLに対してリクエストパラメータを付与することにより、自動的にログインすることが可能です。

例えば、以下のURLでアクセスすると、ユーザ「ueda」でログインし「/imbox/mybox」に遷移します。

```
http://<HOST>:<PORT>/<CONTEXT_PATH>/certification?im_user=ueda&im_password=（パスワード）&im_url=/imbox/mybox
```

自動ログインの詳細は「設定ファイルリファレンス - 認証機能 - 認証設定（一般ユーザ用） - 自動ログインについて」を参照してください。

この自動ログイン機能は、セキュリティの観点からデフォルトでは無効化されています。
この章では、自動ログイン機能を有効にする方法を紹介します。

自動ログイン機能の有効化

- 「conf/token-filtering-target-config/im_certification.xml」ファイルの編集を行います。
  1. 設定ファイルを「ProjectNavigator」上に追加します。
     1. <（プロジェクト名）/juggling.im> ファイルをダブルクリックします。
     2. 「設定ファイル」タブをクリックします。
     3. 「認証機能」の「セキュア・トークンフィルタ設定（認証機能用）」を選択し、右側にある「出力」をクリックします。
  2. 「ProjectNavigator」内の <conf/token-filtering-target-config/im_certification.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
  3. 以下の設定をコメントアウトします。
     - 変更前

```
<p:entry url="/certification" />
```

     - 変更後

```
<!-- <p:entry url="/certification" /> -->
```

外部メニュー連携

intra-mart Accel Platform の外部メニュー連携を行うための接続先情報を設定します。

> **注意**
> 外部メニュー連携は、2013 Spring(Climbing) から利用することが可能です。

項目

- 外部メニュー連携とは
- メニュークライアント環境構築手順
- メニュープロバイダから取得できる情報
- Webサービスモジュール

外部メニュー連携とは

外部メニュー連携とは、シングルサインオン（以下 SSO と記述します）を利用してシステムを構築している場合に、複数の *メニュープロバイダ* のメニュー情報を intra-mart Accel Platform 上で 1 つのメニューとして表示し、利用することを可能とするための機能です。

取得したメニュー情報は、テーマのグローバルナビ、ツリーメニューおよびサイトマップに表示されます。

intra-mart Accel Platform のログインユーザのユーザコードと、メニュープロバイダのユーザコードは一致している必要があります。

メニュープロバイダとは

メニュー情報を取得するための接続先の Webアプリケーションを、メニュープロバイダと呼びます。

外部メニュー連携機能では、メニュープロバイダから SOAP Webサービスを利用してメニュー情報を取得します。

メニュープロバイダには、あらかじめ Webサービスのモジュールをインストールする必要があります。
Webサービスモジュールに関しては、「 *Webサービスモジュール* 」 を参照してください。

メニュークライアントとは

メニュープロバイダからメニュー情報を取得し、 1 つのメニュー情報に統合して表示する側の Webアプリケーションを、メニュークライアントと呼びます。

ユーザは、メニュークライアントとして稼働する intra-mart Accel Platform にアクセスすることで、複数の Webアプリケーションのサービスを意識することなく利用することが可能です。

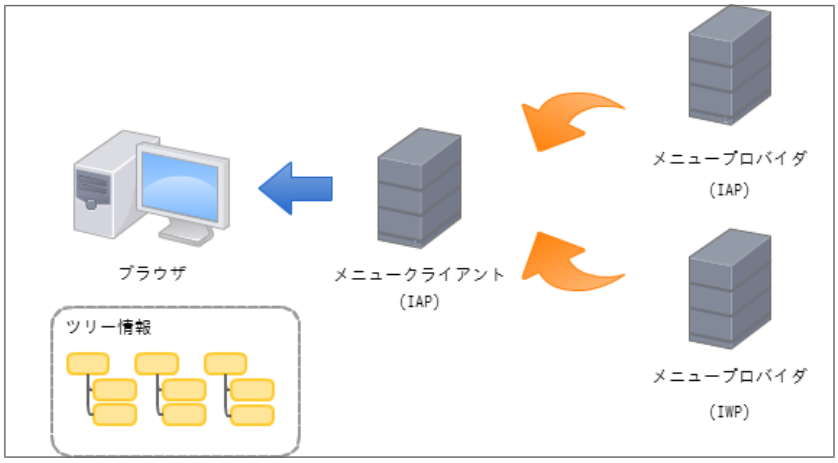


図 外部メニュー連携

注意

外部メニュー連携機能を利用する場合は、連携する各アプリケーションサーバで、SSO による認証を行うことを推奨します。

SSOを利用せずに、外部メニュー連携機能を利用した場合、以下の様な制限事項が発生します。

- メニュープロバイダから取得したメニューにアクセスしても、自動的にログインされません。
  該当のページに権限設定がある場合、エラーページが表示されます。
  エラーページからログイン画面へ遷移してログインすることも可能ですが、ブラウザの設定や環境によっては正常にログインできない場合があります。
  また、メニュープロバイダが intra-mart Accel Platform の場合、グローバルナビを表示しない制御を行っていますが、ログインしていない状態でアクセスした場合は制御が行われず、メニュークライアントのグローバルナビとメニュープロバイダのグローバルナビが２重に表示されます。
- メニュークライアントとメニュープロバイダに別々にログインすることになるため、同じユーザであることが保証されません。
  外部メニュー連携で取得するメニュー情報は、メニュークライアントにログインしているユーザにひもづく情報です。
  メニュープロバイダに別のユーザでログインしていても画面上では判断できないため、誤ったオペレーションを行う可能性があります。
- メニュークライアントからログアウトしても、メニュープロバイダ側は自動ログアウトされません。
  メニュープロバイダに直接アクセスすることで、ログイン状態のままアクセスが可能です。

IM-HybridSSO

2014 Summer(Honoka) から、intra-mart WebPlatform を対象とした簡易SSO機能として IM-HybridSSO が利用可能です。
IM-HybridSSO については、「*iAP-iWP間SSO連携（IM-HybridSSO）*」を参照してください。

IM-HybridSSO では、 intra-mart WebPlatform v7.2 patch7 で利用するための個別パッチモジュールが提供されています。
個別パッチモジュールには IM-HybridSSO の機能以外に、patch8 で提供される予定の外部メニュー連携に関する不具合の修正が含まれます。
不具合の内容については、個別パッチモジュールに添付する readme.txt を参照してください。

メニュークライアント環境構築手順

外部メニュー連携モジュールの選択

IM-Juggling で、「追加機能」-「外部メニュー連携」モジュールを選択します。

メニュープロバイダの設定

「設定ファイル」タブから、「外部メニュー連携」-「外部メニュー連携接続先設定」を出力します。

「外部メニュー連携接続先設定（menu-provider-config.xml）」に、以下のようにメニュープロバイダの設定を記述します。

```
<?xml version="1.0" encoding="UTF-8"?>
 ...
<menu-provider id="sample">
  <end-point> http://<HOST>:<PORT>/<CONTEXT_PATH>/services/MenuService </end-point>
  <login-group> <接続先のログイングループID> </login-group>
  <user> <接続先ユーザのユーザコード> </user>
  <password> <接続先ユーザパスワード> </password>
</menu-provider>
 ...
```

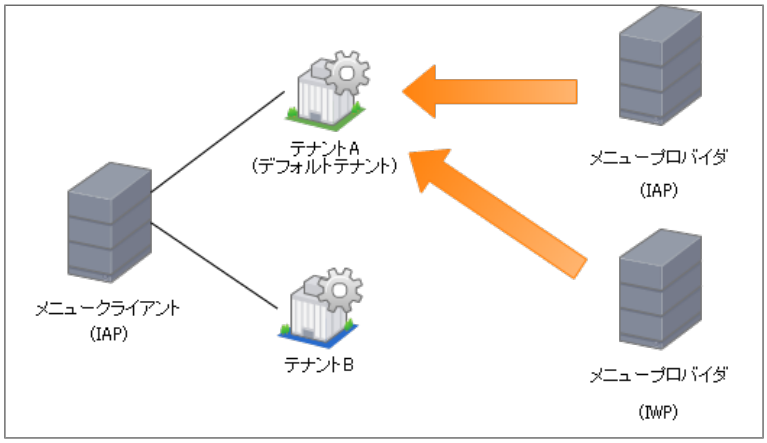
メニュープロバイダ情報について

設定する情報はあらかじめメニュープロバイダのシステム管理者から提供されている必要があります。
以下の情報をメニュープロバイダのシステム管理者に問い合わせてください。

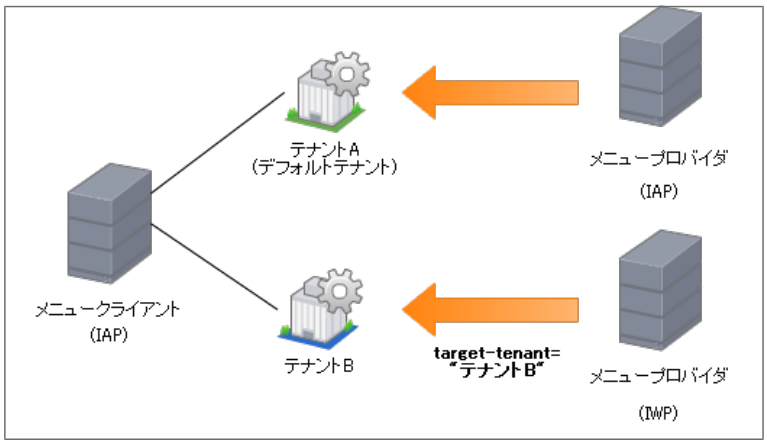
接続先エンドポイント	メニュープロバイダの接続先URLです。 以下のフォーマットです。 *http://<HOST>:<PORT>/<CONTEXT_PATH>/services/MenuService*
接続先ログイングループID	メニュープロバイダが intra-mart WebPlatform の場合のログイングループIDです。 メニュープロバイダが intra-mart Accel Platform の場合はテナントIDです。
接続先ユーザコード	Webサービスの権限を付与された Webサービス接続用ユーザのユーザコードです。
接続先ユーザパスワード	Webサービスの権限を付与された Webサービス接続用ユーザのパスワードです。

バーチャルテナントによる複数テナント運用時のメニュープロバイダの設定について

バーチャルテナントによる複数テナントを運用する場合、標準ではメニュープロバイダの設定はデフォルトテナントに対して有効です。



デフォルトテナント以外のテナントにメニュープロバイダを設定したい場合、「target-tenant」オプションを設定することで任意のテナントへメニュープロバイダを設定できます。



メニュープロバイダの設定順序について

メニュープロバイダは複数設定することが可能です。

標準では、グローバルナビ、ツリーメニューおよびサイトマップには、設定された順に取得したメニュー情報が表示されます。

グローバルナビの場合は、「editable」オプションを設定することで、以下の様な動作です。

- メニュー設定画面で表示対象を選択し、任意の位置に表示するように設定可能です。
- メニュー設定画面で選択しなかったメニュー情報は表示されません。

設定ファイルの詳細な設定内容については「 設定ファイルリファレンス 」-「 外部メニュー連携接続先設定 」 を参照してください。
メニュー設定画面での表示順序の設定については「 テナント管理者操作ガイド 」-「 外部サイトのメニューリンクを登録する 」 を参照してください。

> **注意**
> 「editable」オプションは、2013 Autumn(Eden) から利用することが可能です。

メニュープロバイダから取得できる情報

メニュープロバイダからは、以下の情報が取得できます。

- *一般ユーザが、メニューを表示する場合のメニュー情報*
- *メニュー管理者が、メニューを設定する場合のメニュー情報*

一般ユーザが、メニューを表示する場合のメニュー情報

一般ユーザが、グローバルナビやサイトマップにメニューを表示する場合に取得できるメニュー情報です。

メニュープロバイダからは、メニュークライアントにログインしているユーザが表示権限を持っているメニュー情報を取得します。
intra-mart Accel Platform のログインユーザのユーザコードと、メニュープロバイダのユーザコードは一致している必要があります。

取得できる主なメニュー情報は、以下の情報です。

- メニューの構成
- メニューの表示名
- メニュー情報の URL

「メニューの構成」は、メニュープロバイダで設定されているメニューのフォルダとアイテムの表示順、階層の情報です。

「メニューの表示名」は、メニュープロバイダのユーザロケールの表示名です。
メニュープロバイダとメニュークライアントのユーザロケールが一致しない場合、表示名のロケールは統一されません。

「メニュー情報の URL」は、メニュープロバイダにアクセスする場合の絶対 URLです。
メニュープロバイダが、プロキシサーバ等を利用してアクセスする必要がある場合は、メニュープロバイダ側で適切なベースURLを設定する必要があります。
次項「 *Webサービスモジュール* 」で提供される、各 Webサービスモジュールごとのベースの設定方法を確認し、設定を行ってください。

メニュー管理者が、メニューを設定する場合のメニュー情報

メニュー管理者が、メニュー設定画面で表示対象と表示先を設定する場合に、設定元として取得できるメニュー情報です。

メニュープロバイダからは、メニュークライアントのメニュー設定画面を操作しているユーザが、管理権限を持っているメニュー情報を取得します。
intra-mart Accel Platform のログインユーザのユーザコードと、メニュープロバイダのユーザコードは一致している必要があります。

取得できる主なメニュー情報は、以下の情報です。

- メニューの構成
- メニューの表示名
- メニュー情報の ID

「メニューの構成」は、メニュープロバイダで設定されているメニューのフォルダとアイテムの表示順、階層の情報です。

「メニューの表示名」は、メニュープロバイダのユーザロケールの表示名です。
ここで取得した表示名は、メニュー設定画面でのみ利用されます。
グローバルナビなどに表示する場合は、「 *一般ユーザが、メニューを表示する場合のメニュー情報* 」で取得した表示名が利用されます。

「メニュー情報の ID」は、グローバルナビなどに表示する場合の対象のメニュー IDです。
メニュー設定画面で表示先を設定した場合、「プロバイダの ID」と「メニュー情報の ID」が保存されます。

> **注意**
> *外部メニュー連携サービス for intra-mart WebPlatform v7.2* で提供されるモジュールでは、一般ユーザは管理権限を持たないため、常に全てのメニュー情報を取得します。

Webサービスモジュール

イントラマートより提供される Webサービスは、以下のモジュールです。

- 外部メニュー連携サービス for intra-mart Accel Platform
- 外部メニュー連携サービス for intra-mart WebPlatform v7.2

外部メニュー連携サービス for intra-mart Accel Platform

外部メニュー連携サービス for intra-mart Accel Platform は、オプションモジュールです。

IM-Juggling で、「追加機能」-「Webサービス向け機能」-「外部メニュー連携 Webサービス」モジュールを選択して War を作成します。

プロキシサーバ等を利用している場合、メニュー情報生成のため、ベースURLの設定が必要です。
ベースURL設定は intra-mart Accel Platform 共通の設定を利用しています。 「*ベースURL*」の章を参照してください。

Webサービスモジュールを有効にするためには、認可の設定を行う必要があります。
Webサービス接続用のユーザを新規に作成し、認可設定により以下のリソースに実行許可を与えてください。 （既存のユーザに権限を与えることも可能です。）
以下の２つのリソースに同じ権限を設定してください。

- リソースの種類 ： Webサービス
- リソースグループ／リソース名 ：
  1. 外部メニュー連携／外部メニュー情報取得
  2. 外部メニュー連携／外部メニュー情報取得（管理）

メニュークライアントのシステム管理者に以下の情報を提供します。

Webサービス接続用エンドポイント	以下のフォーマットです。 *http://<HOST>:<PORT>/<CONTEXT_PATH>/services/MenuService*
Webサービス接続用ユーザコード	権限を与えられたユーザのユーザコードです。
Webサービス接続用ユーザパスワード	権限を与えられたユーザのパスワードです。

外部メニュー連携サービス for intra-mart WebPlatform v7.2

外部メニュー連携サービス for WebPlatform v7.2 は、patch6 で提供されました。
また、メニュークライアントで外部メニューの表示順序を設定するための機能は、patch7 で提供されました。
外部メニュー連携を利用するためには、最新のパッチを利用するようにしてください。

プロキシサーバ等を利用している場合、メニュー情報生成のため、ベースURLの設定が必要です。
このモジュールでは、以下の設定を利用しています。設定がない場合は、追加するようにしてください。

- 設定ファイル ： <%ServerManager のインストールパス% /conf/parameter.xml>
- 設定内容 ：

```
<param>
  <param-name>im.web_server.url</param-name>
  <param-value>http://<アクセス可能なホスト名>:<PORT></param-value>
</param>
```

Webサービスモジュールを有効にするためには、ログイングループ管理者により、Webサービスアクセス設定を行う必要があります。
Webサービス接続用のロールとそのロールを付与したユーザを新規に作成し、Webサービスアクセス設定により以下のWebサービスに実行許可を与えてください。（既存のユーザに権限を与えることも可能です。）
以下の２つのオペレーションに同じ権限を設定してください。

- Webサービス名 ： MenuService
- オペレーション名 ：
  1. getAvailableMenuTree
  2. getManagedMenuTree （patch7 以降で提供されます。）

メニュークライアントのシステム管理者に以下の情報を提供します。

Webサービス接続用エンドポイント	以下のフォーマットです。 *http://<HOST>:<PORT>/<CONTEXT_PATH>/services/MenuService*
Webサービス接続先ログイングループID	メニュー情報を取得するログイングループのIDです。
Webサービス接続用ユーザコード	権限を与えられたユーザのユーザコードです。
Webサービス接続用ユーザパスワード	権限を与えられたユーザのパスワードです。

注意

intra-mart WebPlatform v7.2 patch6 では、外部メニューの表示順序を設定できません。
intra-mart WebPlatform v7.2 patch6 を利用する場合は、メニュークライアントの設定ファイルに「editable」オプションを設定しないようにしてください。

コラム

intra-mart Accel Platform のメニューから intra-mart WebPlatform のリンクをクリックした場合、intra-mart WebPlatform へのログインは自動的には行われません。
自動的にログインが行われるようにするためには、別途 SSO 製品の対応が必要です。

2014 Summer(Honoka) から、簡易的にSSOを実現するための機能（IM-HybridSSO）が追加されました。
IM-HybridSSO を利用する場合は、「*iAP-iWP間SSO連携（IM-HybridSSO）*」を参照してください。

IM-SecureSignOn for Accel Platform（以下 IM-SSO と記述します） は、intra-mart Accel Platform に対応した SSO 製品です。
intra-mart WebPlatform の patch6 以降では、IM-SSO を利用した場合に、外部メニュー連携のメニューから自動的にログインが行われるようにするための機能が提供されています。 この機能を有効にするためには、IM-SSO 導入後、以下の設定を行ってください。

<%ApplicationRuntime のインストールパス% /doc/imart/WEB-INF/web.xml> を修正します。
（ `HTTPContextHandlingFilter` の設定の直後に正しく設定を追加してください。設定箇所が間違っている場合、正常に動作しません。）

```
<filter>
   <filter-name>HTTPContextHandlingFilter</filter-name>
   <filter-class>jp.co.intra_mart.common.aid.jsdk.javax.servlet.filter.HTTPContextHandlingFilter</filter-class>
</filter>
<!-- ↓↓↓ 以下を追加 -->
<filter>
   <filter-name>AutoLoginFilter</filter-name>
   <filter-class>jp.co.intra_mart.foundation.security.filter.AutoLoginFilter</filter-class>
</filter>
<!-- ↑↑↑ ここまで -->

  :

<filter-mapping>
   <filter-name>HTTPContextHandlingFilter</filter-name>
   <servlet-name>MenuServlet</servlet-name>
</filter-mapping>
<!-- ↓↓↓ 以下を追加 -->
<filter-mapping>
   <filter-name>AutoLoginFilter</filter-name>
   <servlet-name>MenuServlet</servlet-name>
</filter-mapping>
<!-- ↑↑↑ ここまで -->
```

以上の設定により、メニューURL（～.menu）へのアクセス時に、VANADIS® Login Server で設定された Cookie 情報を参照して自動ログインを行います。

Cookie 情報が設定されている場合、 intra-mart WebPlatform で一般ユーザのログイン・ログアウトは実行できません。
必ず VANADIS® Login Server におけるログイン・ログアウトを実行してください。

iAP-iWP間SSO連携（IM-HybridSSO）

intra-mart Accel Platform（以下 iAP と記述します）と iWP / iAF v7.2 （以下 iWP と記述します）の間で、 シングルサインオン（以下 SSO と記述します）連携するための機能（IM-HybridSSO）の設定方法について説明します。

注意

IM-HybridSSO は、2014 Summer(Honoka) から利用することが可能です。

注意

IM-HybridSSO の iWP用のモジュールは、patch8 で提供されました。

iWP の patch7 で IM-HybridSSO を利用するためには、別途個別パッチモジュールをインストールする必要があります。
個別パッチモジュールは、以下の URL からダウンロード可能です。
※ダウンロードには製品のライセンスキーが必要です。

プロダクトファイルダウンロード

該当のファイルを解凍し、展開された readme.txt を参照してインストールおよび設定を行ってください。

なお、上記個別パッチモジュールを適用し、その後 patch8 の適用を行う場合、patch8 の適用によって一部設定ファイルが初期化されるためSSO連携が行えなくなります。
必ず patch8 の readme.txt を参照した上でパッチの適用を行ってください。

注意

IM-HybridSSO を利用する上での制限事項については、「 リリースノート 」-「 制限事項 」-「 HybridSSO・外部メニュー連携 」を参照してください。

項目

- 概要
  - IM-HybridSSO とは
  - IM-HybridSSO の構成
  - IM-HybridSSO の機能
- IM-HybridSSO の設定
  - iWP の設定
  - iAP の設定
  - 同一ホストで運用する場合の設定

概要

IM-HybridSSO とは

IM-HybridSSO とは、iAP と iWP の間で、SSO 連携するための機能です。
iWP の認証情報の管理と、外部メニュー連携を利用して取得したメニューからのアクセス時に iWP へ自動的にログインすることにより実現されます。
これにより、iAP と iWP のサービスをシームレスに利用することが可能です。

外部メニュー連携については、「 *外部メニュー連携とは* 」を参照してください。

IM-HybridSSO を利用するためには、外部メニュー連携の設定が必要です。
「 *IM-HybridSSO の設定* 」では、IM-HybridSSO と外部メニュー連携の設定を併せて行いますので、手順に沿って設定を行ってください。

IM-HybridSSO の構成

IM-HybridSSO では、接続元と接続先を以下のように定義します。

SSO認証プロバイダ	接続元の iAP のログイン対象のテナント 外部メニュー連携機能のメニュークライアントとしての役割も担います。
SSOサービスプロバイダ	接続先の iWP のログイングループ 外部メニュー連携機能のメニュープロバイダとしての役割も担います。

IM-HybridSSO の基本構成は以下の通りです。

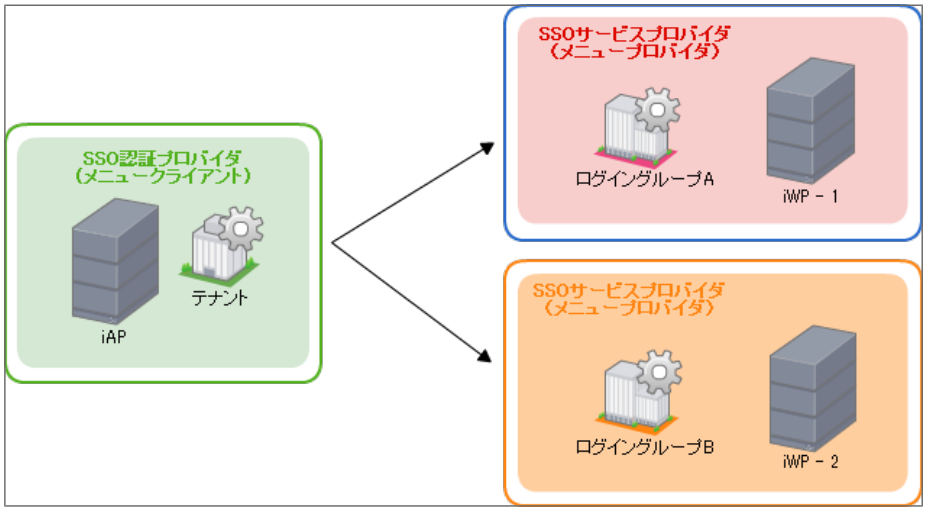


図 IM-HybridSSO の構成

設定はテナントごとに連携先のSSOサービスプロバイダを定義することで行います。

IM-HybridSSO の機能

IM-HybridSSO は、以下の機能により実現されます。

- 認証情報の管理

  iAP にログインすることで、iWP にログイン確認を行い、 ログイン可能な場合は、ブラウザの Cookie に認証情報を保存します。

- SSO自動ログイン

  外部メニュー連携で取得した外部メニューからアクセスすることで、Cookie の認証情報を利用して iWP に自動的にログインし、該当のページを表示します。

- ログアウト連携

  iAP からログアウトすることで、連携している全ての iWP からログアウトし、Cookie から認証情報を削除します。

IM-HybridSSO では、iAP にログインしたユーザのユーザコードを利用して iWP にログインします。
iWP に同一のユーザコードのユーザが存在しない場合は、SSO連携できません。

iAP と iWP のアカウント情報を同期するためには、マスタ情報の同期機能の BackwardSync が利用可能です。
BackwardSync については「BackwardSync(version 7.2へのマスタ同期)仕様書」、および、「BackwardSync（version 7.2へのマスタ同期）操作ガイド」を参照してください。

IM-HybridSSO の設定

iWP の設定

SSO自動ログインの設定

IM-HybridSSO によるSSO自動ログインを有効にするため、以下の設定を行います。

> **注意**
>
> 修正内容の位置に正しく設定を追加してください。
> 設定箇所が間違っている場合、正常に動作しません。

1. メニューサーブレットにフィルタを追加します。
   - 設定ファイル

     <%ApplicationRuntime のインストールパス% /doc/imart/WEB-INF/web.xml>

   - 修正内容

```
  :

<filter>
  <filter-name>HTTPContextHandlingFilter</filter-name>
  <filter-class>jp.co.intra_mart.common.aid.jsdk.javax.servlet.filter.HTTPContextHandlingFilter</filter-class>
</filter>
<!-- ↓↓↓ 以下を追加 -->
<filter>
  <filter-name>AutoLoginFilter</filter-name>
  <filter-class>jp.co.intra_mart.foundation.security.filter.AutoLoginFilter</filter-class>
</filter>
<!-- ↑↑↑ ここまで -->

  :

<filter-mapping>
  <filter-name>HTTPContextHandlingFilter</filter-name>
  <servlet-name>MenuServlet</servlet-name>
</filter-mapping>
<!-- ↓↓↓ 以下を追加 -->
<filter-mapping>
  <filter-name>AutoLoginFilter</filter-name>
  <servlet-name>MenuServlet</servlet-name>
</filter-mapping>
<!-- ↑↑↑ ここまで -->
```

2. SSO自動ログイン用のリクエスト解析モジュールを追加します。
   - 設定ファイル

     <%ServerManager のインストールパス% /conf/access-security.xml>

   - 修正内容

```
<user-security>

    :

   <initial-request-analyzer>
     <request-analyzer-class>jp.co.intra_mart.foundation.security.certification.ShortCutInitialRequestAnalyzer</request-analyzer-class>
   </initial-request-analyzer>
   <!-- ↓↓↓ 以下を追加 -->
   <initial-request-analyzer>
     <request-analyzer-class>jp.co.intra_mart.system.hybrid_sso_provider.HybridSSORequestAnalyzer</request-analyzer-class>
   </initial-request-analyzer>
   <!-- ↑↑↑ ここまで -->

    :
 </user-security>
```

SSOエラーページの設定

以下のエラーページを IM-HybridSSO 用に差し替えます。

- セッションタイムアウトエラーページ
- アクティブセッション無効エラーページ
- ログインブロックエラーページ

差し替えには以下のファイルをそれぞれ修正します。

1. セッションタイムアウトエラーページ
   - 設定ファイル

     <%ServerManager のインストールパス% /conf/access-security.xml>

   - 修正内容

```
  :

<security-config>
 <error-page-provider>
  <page-provider-class>jp.co.intra_mart.foundation.security.certification.StandardErrorPageProvider</page-provider-
class>

    :

  <init-param>
   <!-- このパスを修正してください -->
   <param-name>session-timeout-page</param-name>
   <param-value>system/security/error/im_hybrid_sso/session_timeout_page.jssp</param-value>
  </init-param>

    :

  <init-param>
   <!-- スマートフォンを利用している場合は以下も修正してください -->
   <param-name>session-timeout-page.sp</param-name>
   <param-value>system/security/error/im_hybrid_sso/session_timeout_page_smartphone.jssp</param-value>
  </init-param>

 </error-page-provider>
</security-config>
```

2. アクティブセッション無効エラーページ
   - 設定ファイル

     <%ServerManager のインストールパス% /conf/active-session-config.xml>

   - 修正内容

```
  :

<access-security-page-provider provider-class="jp.co.intra_mart.system.security.AccessSecurityPageProviderImpl">
 <access-security-page page-id="active-session-invalidated">
  <location request-
header="maskat_layoutid">system/security/error/active_session_invalidated_page_maskat.jssp</location>
  <location client-type="mobile">system/security/error/active_session_invalidated_page_mobile.jssp</location>

  <!-- このパスを修正してください -->
  <location>system/security/error/im_hybrid_sso/active_session_invalidated_page.jssp</location>

  <!-- スマートフォンを利用している場合は以下も修正してください -->
  <location client-
type="sp">system/security/error/im_hybrid_sso/active_session_invalidated_page_smartphone.jssp</location>
 </access-security-page>
</access-security-page-provider>
```

3. ログインブロックエラーページ
   - 設定ファイル

     <%ApplicationRuntime のインストールパス% /doc/imart/WEB-INF/web.xml>

   - 修正内容

```
  :

<filter>
 <filter-name>LoginBlockFilter</filter-name>
 <filter-class>jp.co.intra_mart.foundation.security.filter.LoginBlockFilter</filter-class>
 <init-param>
  <param-name>pagePath</param-name>
  <!-- このパスを修正してください -->
  <param-value>/system/security/error/im_hybrid_sso/login_block_page.jssp</param-value>
 </init-param>
 <init-param>
  <param-name>mobilePagePath</param-name>
  <param-value>/system/security/error/login_block_page_mobile.jssp</param-value>
 </init-param>
 <init-param>
  <param-name>pagePath.sp</param-name>
  <!-- スマートフォンを利用している場合は以下も修正してください -->
  <param-value>/system/security/error/im_hybrid_sso/login_block_page_smartphone.jssp</param-value>
 </init-param>
</filter>
```

SSOサービスプロバイダの設定

SSOサービスプロバイダのIDと接続先のSSO連携するログイングループの設定を行います。

- 設定ファイル

  <%ServerManager のインストールパス% /conf/hybrid-sso-provider-config.xml>

- 設定例

```
<hybrid-sso-provider-config
   xmlns="http://www.intra-mart.jp/hybrid_sso_provider/hybrid-sso-provider-config"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/hybrid_sso_provider/hybrid-sso-provider-config"
  provider-id="sample">

 <mappings>
  <mapping login-group="default" encrypt-key="default-enc-key" />
  <mapping login-group="other" encrypt-key="other-enc-key" />
 </mappings>

</hybrid-sso-provider-config>
```

SSOサービスプロバイダ設定ファイルには、以下の設定を行います。

- プロバイダIDの設定

  <hybrid-sso-provider-config> の provider-id 属性にプロバイダIDを設定します。
  プロバイダIDは、同一のSSOを構成するドメイン上の iWP に対して、ユニークなIDとなるように設定してください。

- ログイングループの設定

  <mappings> 内にSSO連携に利用するログイングループの数だけ <mapping> タグを記述します。
  login-group 属性には、ログイングループIDを設定します。
  encrypt-key 属性には、ブラウザの Cookie に設定される認証情報を暗号化するためのキーを設定します。

> **注意**
>
> 認証情報の暗号化方式は、Blowfish を利用しています。
> そのため暗号化のキーには、Blowfish でキーとして設定可能な文字列長の文字列を設定する必要があります。
>
> また、環境によりキーの文字列長が制限されている場合があります。
> Windows 用の Oracle JDK の暗号化プロバイダの実装では、キーの文字列長はデフォルトでは128ビット（半角16文字）に制限されています。（日本で利用する場合）
>
> encrypt-key 属性には、利用している環境で設定可能な文字列長の文字列を設定してください。
>
> Oracle JDK の暗号化プロバイダについては、以下を参照してください。
>
> http://docs.oracle.com/javase/jp/7/technotes/guides/security/SunProviders.html （日本語）
> http://docs.oracle.com/javase/7/docs/technotes/guides/security/SunProviders.html （English）

ベースURLの設定

Webサーバを利用するなどして、ブラウザからのリクエストURLと iWP が受け付けるリクエストURLが異なる場合は、ベースURLの設定が必要です。

ベースURLを設定するために、以下の設定が必要です。
設定がない場合は、追加するようにしてください。

- 設定ファイル

  <%ServerManager のインストールパス% /conf/parameter.xml>

- 修正内容

```
 :

<param>
 <param-name>im.web_server.url</param-name>
 <param-value><http または https>://<アクセス可能なホスト名>:<PORT></param-value>
</param>
```

以下の構成の場合は、<param-value> に http://www.intra-mart.jp を設定します。

WebサーバリクエストURL	*http://www.intra-mart.jp/imart/～*
iWP が受け付けるリクエストURL	*http://iap.intra-mart.jp:8080/imart/～*

外部メニュー連携用モジュールでは、この設定を利用してメニュー情報の URL を生成しています。
この設定がない場合は、iWP が受け付けるリクエストURL を基にメニュー情報の URL が生成されるため、Webサーバ経由のアクセスができなくなります。

Webサービス実行ユーザ設定

Webサービスを実行する特定のユーザを定義します。

1. Webサービス実行ユーザを定義します。
   Webサービス実行用に新しくユーザを作成し、特定のロールを付与してください。
   新しくユーザが作成できない場合は、既存のユーザにロールを付与して、実行用ユーザとして利用することも可能です。
2. 「ログイングループ管理画面」の「Webサービスアクセスメニュー」より、以下のWebサービスにWebサービス実行ユーザの権限（1.で設定したロール）を付与します。
   IM-HybridSSO 用と外部メニュー連携用のWebサービスがありますが、それぞれのオペレーションに同じ権限を設定してください。

Webサービス	**オペレーション**	**説明**
AdmissionService	publishKey	IM-HybridSSO のログイン確認と認証情報の作成。
MenuService	getAvailableMenuTree	表示用の外部メニューツリーの取得。 グローバルナビやサイトマップに表示されるメニューツリーを取得します。
MenuService	getManagedMenuTree	管理用の外部メニューツリーの取得。 メニュー設定画面でメニューの表示位置を設定するためのメニューツリーを取得します。

各画面の操作手順については、iWP の「 グループ管理者操作ガイド 」を参照してください。

- ロールの作成について・・・「1.5.2 ロールの設定」
- ユーザの作成とロールの付与について・・・「1.5.3 ユーザの登録と削除（アカウント情報とプロファイル情報）」
- Webサービス実行ユーザの権限について・・・「1.14 Webサービスアクセス設定」

また、iWP の Webサービスについては詳しく知りたい場合は、 iWP の「Web サービス プログラミングガイド 」を参照してください。

SSOサービスプロバイダ情報の提供

SSO 連携する iAP のシステム管理者に以下の情報を提供します。

iAP のシステム管理者は、この情報を参照して、iAP の設定を行います。
iAP の設定については、「 *iAP の設定* 」を参照してください。

プロバイダID	iWP の設定ファイル「hybrid-sso-provider-config.xml」に設定したプロバイダ ID

認証情報用エンドポイント	以下のフォーマットです。 *http://<HOST>:<PORT>/<CONTEXT_PATH>/services/AdmissionService*
外部メニュー連携用エンドポイント	以下がフォーマットです。 *http://<HOST>:<PORT>/<CONTEXT_PATH>/services/MenuService*
Webサービス接続先 ログイングループID	SSOサービスプロバイダのログイングループIDです。
Webサービス接続用 ユーザコード	権限を与えられたユーザのユーザコードです。
Webサービス接続用 ユーザパスワード	権限を与えられたユーザのパスワードです。
ログアウトURL	以下のフォーマットです。 *http://<HOST>:<PORT>/<CONTEXT_PATH>/user.logout*

iAP の設定

IM-HybridSSO モジュールの選択

IM-Juggling で、「追加機能」-「認証拡張機能」-「iAP-iWP間SSO連携モジュール(IM-HybridSSO)」を選択します。

「iAP-iWP間SSO連携モジュール(IM-HybridSSO)」を利用するには、「外部メニュー連携」モジュールが必要です。
「外部メニュー連携」は、「iAP-iWP間SSO連携モジュール(IM-HybridSSO)」に対して「依存関係も含めて選択」を行うことで自動的に選択されます。

SSOサービスプロバイダの設定

利用するSSOサービスプロバイダを設定ファイルに記述します。

「設定ファイル」タブから、「iAP-iWP間SSO連携モジュール(IM-HybridSSO)」-「SSO連携用マッピング設定（hybrid-sso-mapping-config）」を出力します。

「SSO連携用マッピング設定（hybrid-sso-mapping-config）」に、SSOサービスプロバイダの設定を記述します。
連携するSSOサービスプロバイダの数だけ `<mapping>` タグにSSOサービスプロバイダの情報を記述してください。
プロバイダIDには、「 *SSO サービスプロバイダの設定* 」で iWP に設定した値を記述してください。

IM-HybridSSO を構成するサーバは、全て同一のドメイン上に構築する必要があります。
以下の構成の場合、設定ファイルのサイトドメインに設定する値は「intra-mart.jp」です。

- SSO認証プロバイダ： iap.intra-mart.jp
- SSOサービスプロバイダ１： iwp1.intra-mart.jp
- SSOサービスプロバイダ２： iwp2.intra-mart.jp

設定ファイルの詳細な設定内容については「 設定ファイルリファレンス 」-「 SSO連携用マッピング設定 」を参照してください。

メニュープロバイダの設定

利用する外部メニュー連携用のメニュープロバイダを設定ファイルに記述します。

IM-HybridSSO を利用する場合、メニュープロバイダの接続先情報には、「 *SSO サービスプロバイダの設定* 」で設定した情報と同じ iWP の接続先情報を設定します。
SSOサービスプロバイダと同じ接続先（URLのホスト）、ログイングループ、Webサービス実行ユーザを設定してください。

設定方法については、「 *メニュープロバイダの設定* 」、および、「 設定ファイルリファレンス 」-「 外部メニュー連携接続先設定 」を参照してください。

同一ホストで運用する場合の設定

iAP と iWP を同一ホストで運用する場合、一部の `Cookie` 名が重複するため、SSO を利用するためには、`Cookie` 名を変更する必要があります。
以下の設定を変更してください。

1. セッション `Cookie` 名の変更
   セッションの `Cookie` 名は、アプリケーションサーバにより変更が可能です。
   Resin の場合の設定方法は、以下の通りです。
   - 設定ファイル

     <%RESIN_HOME%/conf/resin.xml>

- 修正内容

```
<!--
  - Resin 4.0 configuration file.
 -->
<resin xmlns="http://caucho.com/ns/resin"
       xmlns:resin="urn:java:com.caucho.resin">

  :

 <cluster id="app">

   :

   <session-cookie>IAP_SESSIONID</session-cookie>

  :

 </cluster>

</resin>
```

Resin 以外の設定方法は、アプリケーションサーバのドキュメントを参照してください。

2. セッション情報チェック用 Cookie 名の変更
   セッション情報の有効性を確認するための Cookie 名を変更します。
   セッション情報チェック用 Cookie 名を変更するためには、「 設定ファイルリファレンス 」-「 セッション情報チェック設定 」-「 セッション情報 Cookie 設定名 」を参照してください。
   Cookie 名は、デフォルトでは「jp.co.intra_mart.session.cookie」が設定されていますので、この値を任意の値に変更してください。
   - 設定例

```
  :

 <!-- セッション情報管理で使用する設定 -->
 <category name="im_session">
  <!-- セッション管理情報を保存するクッキー名 -->
  <param>
   <param-name>cookie_name</param-name>
   <param-value>jp.co.intra_mart.session.cookie.iap</param-value>
  </param>
 </category>
```

Office 365 連携

Office 365 連携 は OAuth2.0 を利用し、 intra-mart Accel Platform 上で Microsoft 365 のリソースの利用を可能にする機能です。

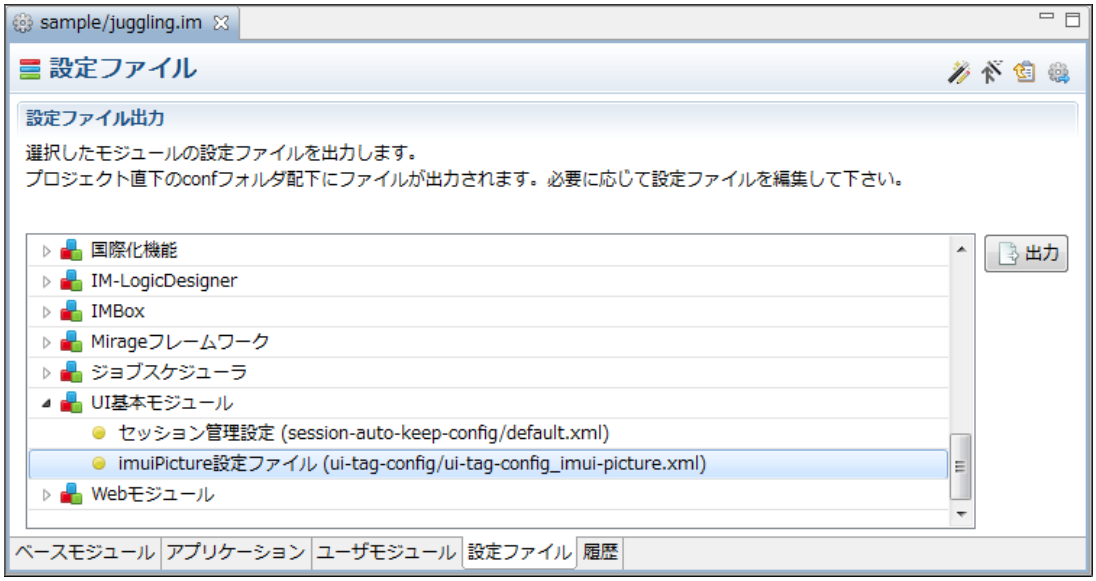
Office 365 連携 のセットアップの詳細は、 Office 365 連携 セットアップガイド を参照してください。

imuiPictureの暗号化キーの設定

imuiPictureの暗号化キーの設定を変更してください。
imuiPictureの暗号化キーの設定の詳細は、「 設定ファイルリファレンス 」-「 UI タグコンポーネント設定 」を参照してください。

設定ファイルの変更方法

imuiPicture設定ファイルは、初期状態では、IM-Juggling プロジェクトに存在しません。
imuiPicture設定ファイルを変更するには IM-Juggling の「設定ファイル」タブから設定ファイルの出力を実施してください。
「UI基本モジュール」-「imuiPicture設定ファイル (ui-tag-config/ui-tag-config_imui-picture.xml)」を選択して「出力」をクリックすることで「ProjectNavigator」内の <（プロジェクト名）/conf/ui-tag-config/ui-tag-config_imui-picture.xml> ファイルが出力されます。



データベース出力用ログ情報設定

テナント管理機能が提供するデータベース出力用ログ情報設定です。
詳細は、「設定ファイルリファレンス」-「データベース出力用ログ情報設定」を参照してください。
本設定ファイルは、2023 Autumn(Hollyhock) 以降のバージョンでご利用いただけます。

注意

データベースの容量圧迫
各種製品でデータベースへ出力されたログ情報が多くなると、データベースの容量を圧迫します。
これに対する対策として、ジョブを使用してログ情報を適宜削除する方法と、ログ情報を出力しないように設定する方法を提供しています。
ジョブの詳細は「ジョブ・ジョブネットリファレンス」の「データベースのシステムログ削除」および「データベースのリクエストログ削除」を参照してください。
ログ情報を出力しないように設定する方法は、intra-mart Accel Platform のバージョンによって異なります。

- 2023 Spring(Gerbera) のバージョンをご利用の場合
  1. IM-Jugglingプロジェクトにシステムログの設定ファイルを追加します。
     IM-Jugglingの「設定ファイル」タブから「コアモジュール」-「システムログ設定 (log/im_logger.xml)」を選択して「出力」をクリックすることで「ProjectNavigator」内の <（プロジェクト名）/conf/log/im_logger.xml> ファイルが出力されます。
  2. 「ログ仕様書」-「システムログ」を参考にして、「ProjectNavigator」内の <（プロジェクト名）/conf/log/im_logger.xml> ファイルで、以下のように configuration タグの末尾へ logger タグを追加します。

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
 ...
 <logger name="jp.co.intra_mart.system.workflow.util.WorkflowDatabaseLogUtil">
  <level value="off" />
 </logger>
</configuration>
```

- 2023 Autumn(Hollyhock) 以降のバージョンをご利用の場合
  「設定ファイルリファレンス」-「データベース出力用ログ情報設定」を参照してください。

ログ情報を削除、または、出力しない場合は、該当のログ情報が「案件ログ情報参照」や「ログ管理機能」で参照できなくなる点に注意してください。

項目
- 設定ファイルの編集方法

設定ファイルの編集方法

1. IM-Jugglingプロジェクトに設定ファイルを追加します。
   データベース出力用ログ情報設定は、初期状態では、IM-Jugglingプロジェクトに存在しません。
   IM-Jugglingの「設定ファイル」タブから「テナント管理機能」-「データベース出力用ログ情報設定 (database-log-config.xml)」を選択して「出力」をクリックすることで、「ProjectNavigator」内の <（プロジェクト名）/conf/database-log-config.xml> ファイルが出力されま

す。

2. 追加された設定ファイルを開き、「設定ファイルリファレンス」-「データベース出力用ログ情報設定」を参考にして設定を変更します。

IM-Copilot

IM-Copilot は intra-mart Accel Platform 上で生成AIを利用した業務アプリケーション開発、および
intra-mart Accel Platform の各種製品で生成AIを活用するための基盤機能です。

IM-Copilot のセットアップの詳細は、「IM-Copilot 利用ガイド」を参照してください。

- 生成AIサービスに関する詳細は、「セットアップ（ 生成AI ）」を参照してください。
- intra-mart Accel Platform に関する詳細は、「セットアップ（ iAP ）」を参照してください。
- 各種製品アシスタントに関する詳細は、「各種製品アシスタント」を参照してください。

キャッシュ設定

認可ポリシー設定キャッシュ

intra-mart Accel Platform の認可ポリシー設定のキャッシュについて設定します。

> **コラム**
> 認可ポリシー設定キャッシュ設定は テナント管理機能 に含まれています。

1. 「ProjectNavigator」内の <conf/im-ehcache-config/authz-policy.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <im-ehcache-config>/<cache> の内容を運用規模に応じて変更します。

```
<cache
 name="IM_TENANT-AUTHZPOLICY"
 enable="true"
 max-bytes-memory="160M"
 time-to-live-seconds="3600"
 time-to-idle-seconds="1800"
/>
```

> **コラム**
> 詳細については「認可仕様書 認可のキャッシュ設定」を参照してください。

認可リソースグループ設定キャッシュ

intra-mart Accel Platform の認可設定のリソースグループに関するキャッシュについて設定します。

> **コラム**
> 認可リソースグループ設定キャッシュ設定は テナント管理機能 に含まれています。

1. 「ProjectNavigator」内の <conf/im-ehcache-config/authz-resourcetype-service.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <im-ehcache-config>/<cache> の内容を運用規模に応じて変更します。

```
<cache
 name="IM_TENANT-AUTHZ_RESOURCETYPE_GENERALSERVICE"
 enable="true"
 max-bytes-memory="128k"
 time-to-live-seconds="86400"
 time-to-idle-seconds="86400"
/>
```

> **コラム**
> 詳細については「認可仕様書 認可のキャッシュ設定」を参照してください。

認可リソース閉塞情報キャッシュ

intra-mart Accel Platform の認可リソースの閉塞情報に関するキャッシュについて設定します。

> コラム
> 認可リソース閉塞情報キャッシュ設定は テナント管理機能 に含まれています。

1. 「ProjectNavigator」内の <conf/im-ehcache-config/authz-resource-block.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <im-ehcache-config>/<cache> の内容を運用規模に応じて変更します。

```
<cache
 name="IM_TENANT-AUTHZ_RESOURCE_BLOCK"
 enable="true"
 max-bytes-memory="320k"
 time-to-live-seconds="86400"
 time-to-idle-seconds="86400"
/>
```

> コラム
> 詳細については「認可仕様書 認可のキャッシュ設定」を参照してください。

メニュー側ルーティング設定キャッシュ

intra-mart Accel Platform のメニュー設定のルーティングに関するキャッシュについて設定します。

> コラム
> メニュー側ルーティング設定キャッシュ設定は テナント管理機能 に含まれています。

1. 「ProjectNavigator」内の <conf/im-ehcache-config/authz-mapped-entry-url.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <im-ehcache-config>/<cache> の内容を運用規模に応じて変更します。

```
<cache
 name="IM_TENANT-AUTHZ_MAPPED_ENTRY_URL"
 enable="true"
 max-bytes-memory="128k"
 time-to-live-seconds="86400"
 time-to-idle-seconds="86400"
/>
```

> コラム
> 詳細については「認可仕様書 認可のキャッシュ設定」を参照してください。

認可IPv4サブジェクト設定キャッシュ

intra-mart Accel Platform の認可設定のIPv4サブジェクトに関するキャッシュについて設定します。

> コラム
> 認可IPv4サブジェクト設定キャッシュ設定は テナント管理機能 に含まれています。

1. 「ProjectNavigator」内の <conf/im-ehcache-config/authz-subjecttype-ipv4.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <im-ehcache-config>/<cache> の内容を運用規模に応じて変更します。

```
<cache
 name="IM_TENANT-AUTHZ_SUBJECTTYPE_IPV4"
 enable="true"
 max-bytes-memory="64k"
 time-to-live-seconds="3600"
 time-to-idle-seconds="1800"
/>
```

コラム

「max-bytes-memory」及び、「max-bytes-disk」属性が設定されている場合、Cacheにオブジェクトを登録する際に、そのオブジェクトのサイズの計算処理が行われます。
この際、登録するオブジェクトが、別のオブジェクトの参照を大量に持つ場合、計算処理に時間がかかりパフォーマンスの低下の原因となる可能性があります。
登録するオブジェクトが1000以上の参照を持つ場合、下記のようなメッセージがログに出力されます。

```
The configured limit of 1,000 object references was reached while attempting to calculate the size of the object graph.
Severe performance degradation could occur if the sizing operation continues.
This can be avoided by setting the CacheManger or Cache <sizeOfPolicy> elements maxDepthExceededBehavior to "abort" or adding stop points with @IgnoreSizeOf annotations.
If performance degradation is NOT an issue at the configured limit, raise the limit value using the CacheManager or Cache <sizeOfPolicy> elements maxDepth attribute.
For more information, see the Ehcache configuration documentation.
```

このログが出力される場合は、キャッシュに格納するオブジェクトの構成を見直すか、「max-bytes-memory」または、「max-bytes-disk」の代わりに、「max-elements-on-memory」または「max-elements-on-disk」の利用を検討して下さい。

キャッシュするオブジェクトの単位

認可IPv4サブジェクト設定は、テナント単位でキャッシュされます。
そのため、キャッシュするオブジェクト数は 1 つです。

キャッシュサイズの計算式

キャッシュが行われる対象データのおおまかなサイズは以下の計算式で求めることができます。

キャッシュサイズ ＝ （（1） ＋ （2））× （3）

（1） ... IPv4 に関するサブジェクトID のバイト数 （64byte）
（2） ... アドレスパターン文字列 （平均 100byte）
（3） ... 登録されている IPv4 に関するサブジェクトの数

intra-mart Accel Platform のデフォルト値は以下の計算をもとに設定されています。

（64 + 100） × 400 ＝ 65600
⇒ 64KB

コラム

上記を算出するために利用している各要素の想定数については以下の通りです

- 登録されている IPv4 に関するサブジェクトの数 : 400

グローバルナビキャッシュ

intra-mart Accel Platform のグローバルナビキャッシュについて設定します。
グローバルナビキャッシュは intra-mart Accel Platform 標準テーマを使用した場合に利用されます。

コラム

グローバルナビキャッシュ設定は テナント管理機能 に含まれています。

1. 「ProjectNavigator」内の <conf/im-ehcache-config/menu-dropdown.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <im-ehcache-config>/<cache> の内容を運用規模に応じて変更します。

```
<cache
  name="IM_TENANT-MENU_DROPDOWN"
  enable="true"
  max-bytes-memory="62M"
  time-to-live-seconds="259200"
  time-to-idle-seconds="259200"
/>
```

コラム

「max-bytes-memory」及び、「max-bytes-disk」属性が設定されている場合、Cacheにオブジェクトを登録する際に、そのオブジェクトのサイズの計算処理が行われます。
この際、登録するオブジェクトが、別のオブジェクトの参照を大量に持つ場合、計算処理に時間がかかりパフォーマンスの低下の原因となる可能性があります。
登録するオブジェクトが1000以上の参照を持つ場合、下記のようなメッセージがログに出力されます。

```
The configured limit of 1,000 object references was reached while attempting to calculate the size of the object graph.
Severe performance degradation could occur if the sizing operation continues.
This can be avoided by setting the CacheManger or Cache <sizeOfPolicy> elements maxDepthExceededBehavior to "abort" or adding stop points with @IgnoreSizeOf annotations.
If performance degradation is NOT an issue at the configured limit, raise the limit value using the CacheManager or Cache <sizeOfPolicy> elements maxDepth attribute.
For more information, see the Ehcache configuration documentation.
```

このログが出力される場合は、キャッシュに格納するオブジェクトの構成を見直すか、「max-bytes-memory」または、「max-bytes-disk」の代わりに、「max-elements-on-memory」または「max-elements-on-disk」の利用を検討して下さい。

キャッシュするオブジェクトの単位

グローバルナビキャッシュは、ユーザ単位でキャッシュされます。
そのため、キャッシュするオブジェクト数は、利用ユーザ数によって見積もります。

キャッシュサイズの計算式

キャッシュが行われる対象データのおおまかなサイズは以下の計算式で求めることができます。

キャッシュサイズ ＝ （（1） × （2） ＋ （3） × （4） ＋ （5）） × （6）

（1） ... 各メニューアイテムに設定されている内容サイズ（平均600byte）
　（1） ＝ （1ａ） ＋ （1ｂ） ＋ （1ｃ） ＋ （1ｄ） ＋ （1ｅ） ＋ （1ｆ） ＋ （（1ｇ） × （1ｈ））
　（1ａ） ... メニューアイテム毎に出力されるHTMLのバイト数
　（1ｂ） ... メニューアイテムIDのバイト数
　（1ｃ） ... URLのバイト数
　（1ｄ） ... メニューアイテム表示名
　（1ｅ） ... メニューアイテム表示オプション（ポップアップ表示、iframe表示など）
　（1ｆ） ... 画像情報
　（1ｇ） ... 引数（キー+値）のバイト数
　（1ｈ） ... 引数の数
（2） ... 表示されるメニューアイテムの数
（3） ... 各メニューフォルダに設定されている内容サイズ（平均150byte）
　（3） ＝ （3ａ） ＋ （3ｂ） ＋ （3ｃ）
　（3ａ） ... メニューフォルダ毎に出力されるHTMLのバイト数
　（3ｂ） ... メニューフォルダ表示名
　（3ｃ） ... 画像情報
（4） ... 表示されるメニューフォルダの数
（5） ... グローバルナビが常に出力するHTML（700byte）
（6） ... 利用ユーザ数

intra-mart Accel Platform のデフォルト値は以下の計算をもとに設定されています。

（600 × 50 ＋ 150 × 10 ＋ 700） × 2000 ＝ 64400000
⇒ 62MB

コラム

上記を算出するために利用している各要素の想定数については以下の通りです

- 表示されるメニューアイテムの数：50
- 表示されるメニューフォルダの数：10
- 利用ユーザ数：2000

ヘルプドロップダウンキャッシュ

intra-mart Accel Platform のヘルプドロップダウンのキャッシュについて設定します。

コラム
ヘルプドロップダウンキャッシュ設定は テナント管理機能 に含まれています。

1. 「ProjectNavigator」内の <conf/im-ehcache-config/help-dropdown.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <im-ehcache-config>/<cache> の内容を運用規模に応じて変更します。

```
<cache
 name="IM_TENANT-MENU_SITE_HELP_DROPDOWN"
 enable="true"
 max-bytes-memory="3700K"
 time-to-live-seconds="259200"
 time-to-idle-seconds="259200"
/>
```

コラム
「max-bytes-memory」及び、「max-bytes-disk」属性が設定されている場合、Cacheにオブジェクトを登録する際に、そのオブジェクトのサイズの計算処理が行われます。
この際、登録するオブジェクトが、別のオブジェクトの参照を大量に持つ場合、計算処理に時間がかかりパフォーマンスの低下の原因となる可能性があります。
登録するオブジェクトが1000以上の参照を持つ場合、下記のようなメッセージがログに出力されます。

```
The configured limit of 1,000 object references was reached while attempting to calculate the size of the object graph.
Severe performance degradation could occur if the sizing operation continues.
This can be avoided by setting the CacheManger or Cache <sizeOfPolicy> elements maxDepthExceededBehavior to "abort" or adding stop points with @IgnoreSizeOf annotations.
If performance degradation is NOT an issue at the configured limit, raise the limit value using the CacheManager or Cache <sizeOfPolicy> elements maxDepth attribute.
For more information, see the Ehcache configuration documentation.
```

このログが出力される場合は、キャッシュに格納するオブジェクトの構成を見直すか、「max-bytes-memory」または、「max-bytes-disk」の代わりに、「max-elements-on-memory」または「max-elements-on-disk」の利用を検討して下さい。

キャッシュするオブジェクトの単位

ヘルプドロップダウンは、ユーザ単位でキャッシュされます。
そのため、キャッシュするオブジェクト数は、利用ユーザ数によって見積もります。

キャッシュサイズの計算式

キャッシュが行われる対象データのおおまかなサイズは以下の計算式で求めることができます。

キャッシュサイズ ＝ （（1） × （2）） × （3）

（1） ... 各メニューアイテムに設定されている内容サイズ（平均630byte）
　（1） ＝ （1 a） ＋ （1 b） ＋ （1 c）
　（1 a） ... メニューアイテムIDのバイト数
　（1 b） ... URLのバイト数
　（1 c） ... メニューアイテム表示名
（2） ... 表示されるメニューアイテムの数
（3） ... 利用ユーザ数

intra-mart Accel Platform のデフォルト値は以下の計算をもとに設定されています。

（630 × 3） × 2000 ＝ 3780000
⇒ 3,700KB

コラム
上記を算出するために利用している各要素の想定数については以下の通りです

- 表示されるメニューアイテムの数：3
- 利用ユーザ数：2000

個人設定メニューキャッシュ

intra-mart Accel Platform の個人設定メニューのキャッシュについて設定します。

> **コラム**
> 個人設定メニューキャッシュ設定は テナント管理機能 に含まれています。

1. 「ProjectNavigator」内の <conf/im-ehcache-config/menu-personal-settings-item.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <im-ehcache-config>/<cache> の内容を運用規模に応じて変更します。

```
<cache
 name="IM_TENANT-MENU_PERSONAL_SETTINGS_ITEM"
 enable="true"
 max-bytes-memory="80M"
 time-to-live-seconds="259200"
 time-to-idle-seconds="259200"
/>
```

> **コラム**
> 「max-bytes-memory」及び、「max-bytes-disk」属性が設定されている場合、Cacheにオブジェクトを登録する際に、そのオブジェクトのサイズの計算処理が行われます。
> この際、登録するオブジェクトが、別のオブジェクトの参照を大量に持つ場合、計算処理に時間がかかりパフォーマンスの低下の原因となる可能性があります。
> 登録するオブジェクトが1000以上の参照を持つ場合、下記のようなメッセージがログに出力されます。
>
> ```
> The configured limit of 1,000 object references was reached while attempting to calculate the size of the object graph.
> Severe performance degradation could occur if the sizing operation continues.
> This can be avoided by setting the CacheManger or Cache <sizeOfPolicy> elements maxDepthExceededBehavior to "abort" or
> adding stop points with @IgnoreSizeOf annotations.
> If performance degradation is NOT an issue at the configured limit, raise the limit value using the CacheManager or Cache
> <sizeOfPolicy> elements maxDepth attribute.
> For more information, see the Ehcache configuration documentation.
> ```
>
> このログが出力される場合は、キャッシュに格納するオブジェクトの構成を見直すか、「max-bytes-memory」または、「max-bytes-disk」の代わりに、「max-elements-on-memory」または「max-elements-on-disk」の利用を検討して下さい。

キャッシュするオブジェクトの単位

個人設定メニューは、ユーザ単位でキャッシュされます。
そのため、キャッシュするオブジェクト数は、利用ユーザ数によって見積もります。

キャッシュサイズの計算式

キャッシュが行われる対象データのおおまかなサイズは以下の計算式で求めることができます。

キャッシュサイズ ＝ （（１） ＋ （２） ＋ （３） ＋ （４） × （５）） × （６）

（１） ... キャッシュの管理情報（平均 200byte）
（２） ... キャッシュのキー情報（平均 1000byte）
（３） ... 個人設定のメニュー情報（平均 200byte）
（４） ... 個人設定配下のメニュー情報（平均 2000byte）
　（４） ＝ （４ａ） ＋ （４ｂ）
　（４ａ） ... メニュー基本情報
　（４ｂ） ... メニュー多言語情報
（５） ... 個人設定配下のメニュー数
（６） ... 利用ユーザ数

intra-mart Accel Platform のデフォルト値は以下の計算をもとに設定されています。

（200 ＋ 1000 ＋ 200 ＋ 2000 × 20） × 2000 ＝ 82800000
⇒ 80MB

コラム

上記を算出するために利用している各要素の想定数については以下の通りです

- 個人設定配下のメニュー数：20
- 利用ユーザ数：2000

ポータルの認可サブジェクト情報キャッシュ

intra-mart Accel Platform のポータルの認可サブジェクト情報のキャッシュについて設定します。

コラム

ポータルの認可サブジェクト情報キャッシュ設定は ポータル機能 に含まれています。

1. 「ProjectNavigator」内の <conf/im-ehcache-config/im_portal-subjecttype.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <im-ehcache-config>/<cache> の内容を運用規模に応じて変更します。

```
<cache
 name="IM_PORTAL-SUBJECTTYPE"
 enable="true"
 max-elements-on-memory="200"
 time-to-live-seconds="3600"
 time-to-idle-seconds="1800"
/>
```

コラム

「max-bytes-memory」及び、「max-bytes-disk」属性が設定されている場合、Cacheにオブジェクトを登録する際に、そのオブジェクトのサイズの計算処理が行われます。
この際、登録するオブジェクトが、別のオブジェクトの参照を大量に持つ場合、計算処理に時間がかかりパフォーマンスの低下の原因となる可能性があります。
登録するオブジェクトが1000以上の参照を持つ場合、下記のようなメッセージがログに出力されます。

```
The configured limit of 1,000 object references was reached while attempting to calculate the size of the object graph.
Severe performance degradation could occur if the sizing operation continues.
This can be avoided by setting the CacheManger or Cache <sizeOfPolicy> elements maxDepthExceededBehavior to "abort" or
adding stop points with @IgnoreSizeOf annotations.
If performance degradation is NOT an issue at the configured limit, raise the limit value using the CacheManager or Cache
<sizeOfPolicy> elements maxDepth attribute.
For more information, see the Ehcache configuration documentation.
```

このログが出力される場合は、キャッシュに格納するオブジェクトの構成を見直すか、「max-bytes-memory」または、「max-bytes-disk」の代わりに、「max-elements-on-memory」または「max-elements-on-disk」の利用を検討して下さい。

キャッシュするオブジェクトの単位

ポータルの認可サブジェクト情報は、HTTPセッション単位でキャッシュされます。
そのため、キャッシュするオブジェクト数は、同時アクセスの最大HTTPセッション数によって見積もります。

キャッシュサイズの計算式

キャッシュが行われる対象データのおおまかなサイズは以下の計算式で求めることができます。

キャッシュサイズ（オブジェクトの数） ＝ （1）

（1） ... HTTPセッション数

intra-mart Accel Platform のデフォルト値は以下の計算をもとに設定されています。

同時アクセスの最大HTTPセッション数を200と仮定しています。

アイコンキャッシュ

intra-mart Accel Platform のアイコンのキャッシュについて設定します。
アイコンキャッシュで対象とするアイコンは、アイコン管理で登録されたアイコンです。

アイコンキャッシュは、登録されたアイコンを表示する際に利用されます。

> **コラム**
>
> アイコンキャッシュ設定は テナント管理機能 に含まれています。

> **コラム**
>
> アイコン管理についての詳細は、「 intra-mart 要件情報公開サイト 」-「 要件 #25242 アイコン管理機能を提供します。 」 を参照してください。

> **注意**
>
> アイコンキャッシュは、intra-mart Accel Platform 2017 Winter(Rebecca) 以降で利用できます。

1. 「ProjectNavigator」内の <conf/im-ehcache-config/icon-picker.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <im-ehcache-config>/<cache> の内容を運用規模に応じて変更します。

```
<cache
 name="IM_TENANT-ICON-PICKER"
 enable="true"
 max-elements-on-memory="100"
 max-bytes-memory="500k"
 overflow-to-disk="false"
 max-elements-on-disk="100"
 max-bytes-disk="500k"
 time-to-live-seconds="3600"
 time-to-idle-seconds="1800"
/>
```

> **注意**
>
> デフォルトは、以下の属性が無効です。
>
> - max-elements-on-memory
> - max-elements-on-disk
> - max-bytes-disk
>
> 各属性の詳細については、「 設定ファイルリファレンス 」-「 キャッシュ設定 」を参照してください。

> **コラム**
>
> 「max-bytes-memory」及び、「max-bytes-disk」属性が設定されている場合、Cacheにオブジェクトを登録する際に、そのオブジェクトのサイズの計算処理が行われます。
> この際、登録するオブジェクトが、別のオブジェクトの参照を大量に持つ場合、計算処理に時間がかかりパフォーマンスの低下の原因となる可能性があります。
> 登録するオブジェクトが1000以上の参照を持つ場合、下記のようなメッセージがログに出力されます。
>
> ```
> The configured limit of 1,000 object references was reached while attempting to calculate the size of the object graph.
> Severe performance degradation could occur if the sizing operation continues.
> This can be avoided by setting the CacheManger or Cache <sizeOfPolicy> elements maxDepthExceededBehavior to "abort" or adding stop points with @IgnoreSizeOf annotations.
> If performance degradation is NOT an issue at the configured limit, raise the limit value using the CacheManager or Cache <sizeOfPolicy> elements maxDepth attribute.
> For more information, see the Ehcache configuration documentation.
> ```
>
> このログが出力される場合は、キャッシュに格納するオブジェクトの構成を見直すか、「max-bytes-memory」または、「max-bytes-disk」の代わりに、「max-elements-on-memory」または「max-elements-on-disk」の利用を検討して下さい。

キャッシュするオブジェクトの単位

アイコンは、テナント単位でキャッシュされます。
デフォルトでは、1テナントあたり100個のアイコンをキャッシュします。

キャッシュサイズの計算式

キャッシュが行われる対象データのおおまかなサイズは以下の計算式で求めることができます。

キャッシュサイズ ＝ （1） × （2）

（1） ... アイコンの中で一番大きいバイト数
（2） ... キャッシュするオブジェクトの数

intra-mart Accel Platform のデフォルト値は以下の計算をもとに設定されています。

5000 × 100 ＝ 500000
⇒ 500KB

クロスオリジンリソース共有のキャッシュ

クロスオリジンリソース共有のキャッシュについて設定します。
クロスオリジンリソース共有のキャッシュはアプリケーションサーバにつき 1 つだけ保持されます。

> **コラム**
> クロスオリジンリソース共有のキャッシュ設定は テナント管理機能 に含まれています。

> **注意**
> クロスオリジンリソース共有のキャッシュは、intra-mart Accel Platform 2019 Summer(Waltz) 以降で利用できます。

1. 「ProjectNavigator」内の <conf/im-ehcache-config/im_tenant_cors.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <im-ehcache-config>/<cache> の内容を運用規模に応じて変更します。

```
<cache
 name="IM_TENANT-SYSTEM-CORS-CONFIG"
 enable="true"
 max-bytes-memory="110k"
 time-to-live-seconds="259200"
 time-to-idle-seconds="259200"
/>
```

> **コラム**
> 「max-bytes-memory」及び、「max-bytes-disk」属性が設定されている場合、Cacheにオブジェクトを登録する際に、そのオブジェクトのサイズの計算処理が行われます。
> この際、登録するオブジェクトが、別のオブジェクトの参照を大量に持つ場合、計算処理に時間がかかりパフォーマンスの低下の原因となる可能性があります。
> 登録するオブジェクトが1000以上の参照を持つ場合、下記のようなメッセージがログに出力されます。
>
> ```
> The configured limit of 1,000 object references was reached while attempting to calculate the size of the object graph.
> Severe performance degradation could occur if the sizing operation continues.
> This can be avoided by setting the CacheManger or Cache <sizeOfPolicy> elements maxDepthExceededBehavior to "abort" or
> adding stop points with @IgnoreSizeOf annotations.
> If performance degradation is NOT an issue at the configured limit, raise the limit value using the CacheManager or Cache
> <sizeOfPolicy> elements maxDepth attribute.
> For more information, see the Ehcache configuration documentation.
> ```
>
> このログが出力される場合は、キャッシュに格納するオブジェクトの構成を見直すか、「max-bytes-memory」または、「max-bytes-disk」の代わりに、「max-elements-on-memory」または「max-elements-on-disk」の利用を検討して下さい。

キャッシュするオブジェクトの単位

クロスオリジンリソース共有設定情報は、クロスオリジンリクエストによるアクセスがあったパス単位でキャッシュされます。
そのため、キャッシュするオブジェクト数は、クロスオリジンリクエストによるアクセスが想定されるパス数によって見積もります。

キャッシュサイズの計算式

キャッシュが行われる対象データのおおまかなサイズは以下の計算式で求めることができます。

キャッシュサイズ ＝ （1） × （2）

（1） ... クロスオリジンリソース共有設定情報（平均 1100byte）
　（1） ＝ （1 a） ＋ （1 b） ＋ （1 c）
　（1 a） ... パス情報

（１ｂ） ... オリジン情報
（１ｃ） ... その他ヘッダ情報
（２） ... クロスオリジンリクエストによるアクセスが想定されるパス数

intra-mart Accel Platform のデフォルト値は以下の計算をもとに設定されています。

1100 × 100 ＝ 100000
⇒ 110KB

> **コラム**
> 上記を算出するために利用している各要素の想定数については以下の通りです
>
> - クロスオリジンリクエストによるアクセスが想定されるパス数 : 100

機能制御の設定

ファイルのアップロードを制限する方法

intra-mart Accel Platform を稼働させるために以下の設定ファイルの編集を行います。

リクエスト制御設定

> **コラム**
> リクエスト制御設定は Webモジュール機能 に含まれています。

1. 「ProjectNavigator」内の <（プロジェクト名）/conf/request-control-config.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. 「request-control-config.xml」の設定内容については「設定ファイルリファレンス 」を参照してください。

SAStrutsを利用している場合

SAStruts でファイルのアップロードを行うには、「struts-config.xml」の設定が必要です。

- <struts-config.xml>ファイルの編集を行います。

1. 設定ファイルを「ProjectNavigator」上に追加します。
   - 次の方法があります。
     ・ IM-Juggling プロジェクトを作成するウィザード中の追加リソースの配置より、「SAStruts用設定ファイル」を追加します。
     ・ IM-Juggling プロジェクト作成後に「追加リソースの選択」より「SAStruts用設定ファイル」を追加します。
2. 「ProjectNavigator」内の <（プロジェクト名）/struts-config.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
3. 「ソース」タブをクリックし、Controller Configurationが定義されている行に移動します。

```
<!-- ==================================== Controller Configuration -->

   <controller
      maxFileSize="1024K"
      bufferSize="1024"
      processorClass="org.seasar.struts.action.S2RequestProcessor"
      multipartClass="org.seasar.struts.upload.S2MultipartRequestHandler"/>
```

4. 「maxFileSize」属性に定義されている値を変更します。

> **コラム**
> **「ProjectNavigator」内のツリー上に設定ファイルがない場合**
>
> 1. <（プロジェクト名）/juggling.im> ファイルをダブルクリックします。
> 2. 「設定ファイル」タブをクリックします。
> 3. 対象の設定ファイルを選択し、右側にある「出力」をクリックします。
> 4. 「ProjectNavigator」内のツリー上に表示されたファイルをダブルクリックして編集を行います。

コラム

「WARファイルによる複数テナント」環境の場合、*WARファイルによる複数テナント* を参考にしてください。

コラム

各アプリケーションで選択したモジュール内の設定ファイルは、各アプリケーションのドキュメントを参照してください。

WARファイルの出力

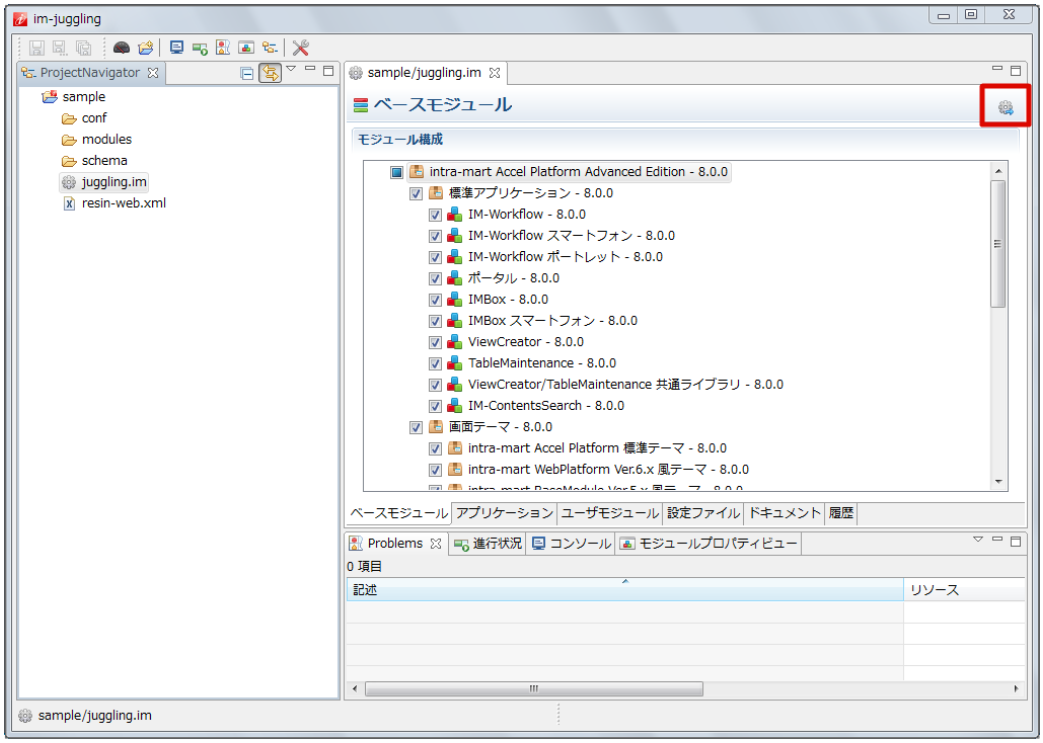
コラム

WARファイルとは？

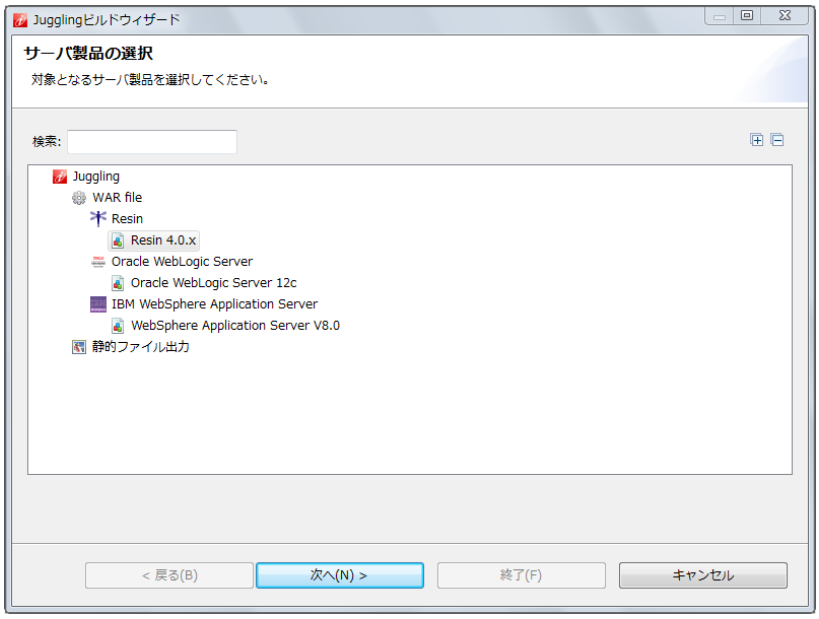
JavaEEで定義されている Web Application Archive ファイルです。
intra-mart Accel Platform の動作に必要な各種ファイルが格納され、intra-mart Accel Platform 用のWARファイルの作成は IM-Juggling にて行います。
IM-Juggling にて作成されたWARファイルを Resin などの Web Application Server に
配置（デプロイ）して、 Web Application Server を起動することにより、WARファイルに格納された intra-mart Accel Platform 自体や intra-mart Accel Platform に対応した
各種アプリケーションが起動され、利用できます。

このことから intra-mart Accel Platform の製品自体の最小単位になるため、intra-mart Accel Platform のライセンスはWARファイル単位でカウントします。

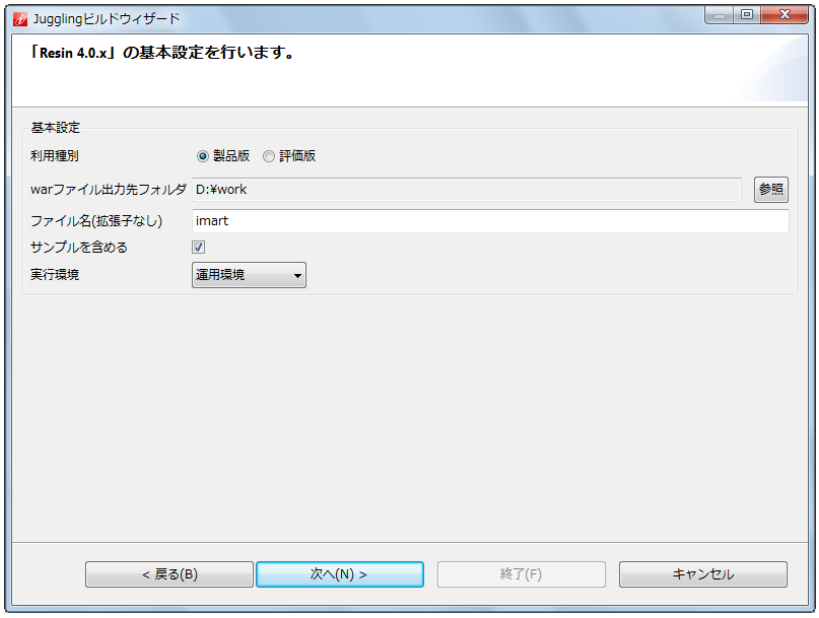
1. <juggling.im> ファイルを開き、右上にある「ビルドウィザード」アイコンをクリックします。



2. 「サーバ製品の選択」画面で対象となるサーバを選択し、「次へ」をクリックします。



3. 「基本設定」画面で設定を行い、「次へ」をクリックします。



利用種別

- 製品版
  ライセンスを購入されたお客様は製品版を選択してください。
- 評価版
  評価版です。
  正規に使用する場合はライセンスを購入してください。

サンプルデータを含める

チェックを入れた場合、サンプルデータを投入します。

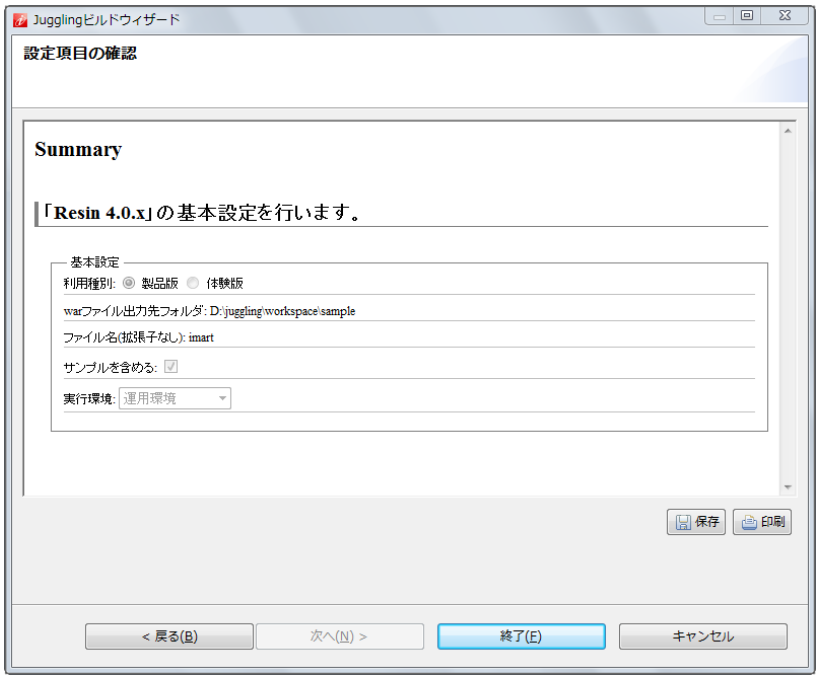
実行環境

- intra-mart Accel Platform を稼働する環境に応じて、実行環境を選択します。
  詳細は、「*WAR ファイル作成時の実行環境の変更*」を参照してください。

> **注意**
> 「WARファイル出力先フォルダ」には、実行するユーザの権限があるフォルダを指定してください。

4. 「ライセンスのレビュー」画面で内容を確認し、同意頂ける場合は「使用条件の条項に同意します。」を選択し、「次へ」をクリックします。

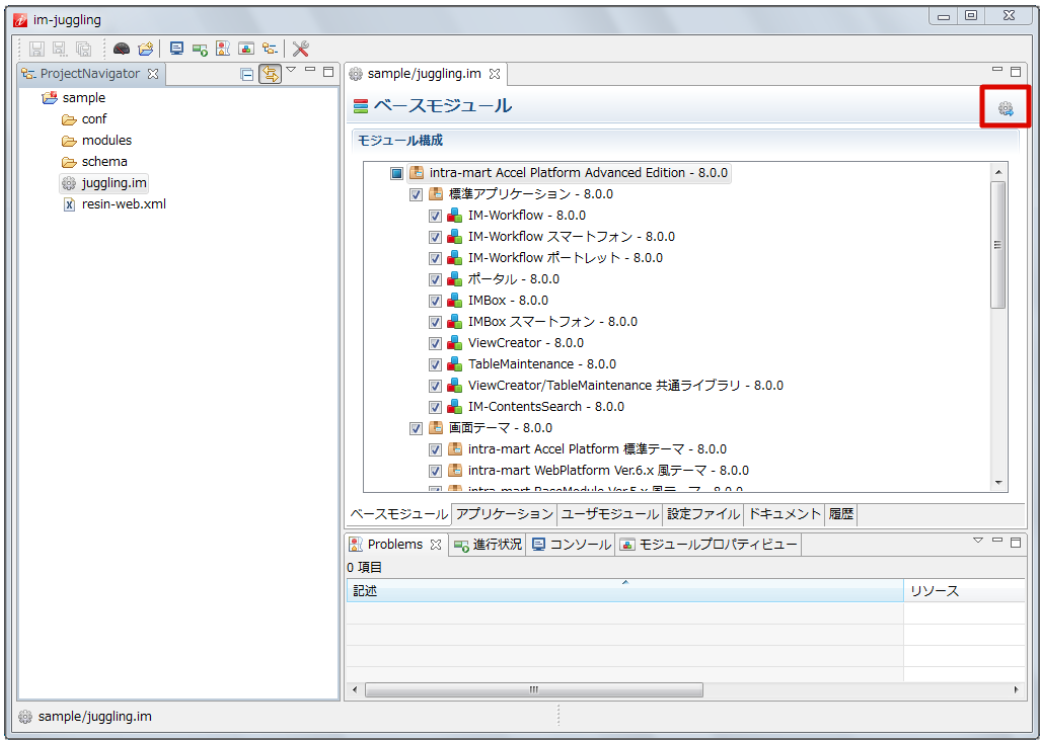
5. 「設定項目の確認」画面で内容を確認し、「終了」をクリックします。
   Jugglingビルドウィザード画面が立ち上がり、ダウンロードが開始されます。
   ダウンロードに時間がかかりますので、完了するまで操作を行わないでください。



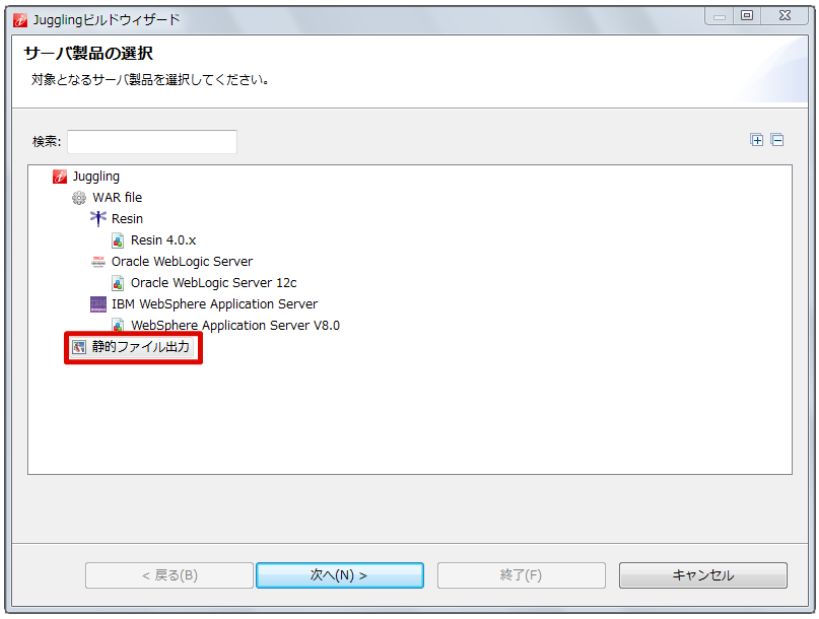
6. WARファイルが指定の場所に出力されると、ビルドウィザード画面が閉じます。
   指定したディレクトリにWARファイルが出力されます。

静的ファイルの出力

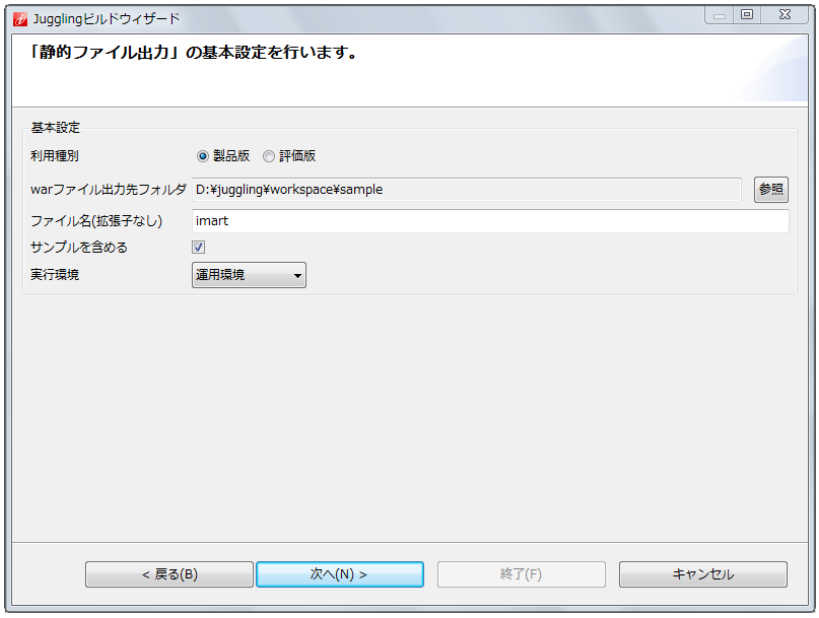
1. <juggling.im> ファイルを開き、右上にある「ビルドウィザード」アイコンをクリックします。



2. 「サーバ製品の選択」画面で「静的ファイル」を選択し、「次へ」をクリックします。



3. 「基本設定」画面で設定を行い、「次へ」をクリックします。



利用種別

- 製品版
  ライセンスを購入されたお客様は製品版を選択してください。
- 評価版
  評価版です。
  正規に使用する場合はライセンスを購入してください。

サンプルデータを含める

チェックを入れた場合、サンプルデータを投入します。

実行環境

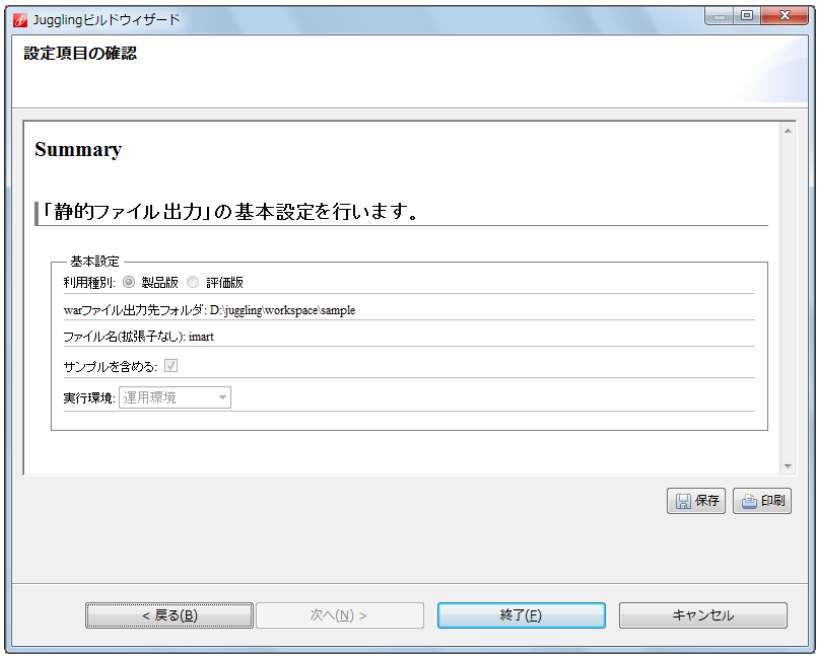
- 単体テスト環境
  単体テスト環境を表します。
  JUnitなどを利用する場合を想定しています。
  パフォーマンスに関しては考慮しません。
- 結合テスト環境
  結合テスト環境を表します。
  IDEを使用したテスト環境などを想定しています。
  一般的に最も利用される環境を想定しています。
  必要最低限の動作環境です。
- 統合テスト環境
  統合テスト環境を表します。
  本来の実行環境（APサーバ）を使用した統合テスト環境を想定しています。
  カットオーバー直後等でもこの環境を利用することを想定しています。

- 運用環境
  運用環境を表します。
  パフォーマンスを考慮した動作環境です。

> **注意**
> 「warファイル出力先フォルダ」には、実行するユーザの権限があるフォルダを指定してください。

4. 「ライセンスのレビュー」画面で内容を確認し、同意頂ける場合は「使用条件の条項に同意します。」を選択し、「終了」をクリックします。

5. 「設定項目の確認」画面で内容を確認し、「終了」をクリックします。
   Jugglingビルドウィザード画面が立ち上がり、ダウンロードが開始されます。
   ダウンロードに時間がかかりますので、完了するまで操作を行わないでください。



6. zipファイルが指定の場所に出力されると、ビルドウィザード画面が閉じます。

7. 出力されたzipファイルを、 Web Server の仮想ディレクトリで設定したディレクトリにzipファイルを展開して配置します。

Web Application Server の起動・停止

> **注意**
>
> 分散環境で複数のサーバを起動する場合、各サーバの起動時刻をずらすことを推奨します。
> 詳細は 「分散環境で複数のサーバを同時起動しようとすると、起動に失敗する場合があります。」 を参照してください。

Windows

コンソール起動・停止

項目

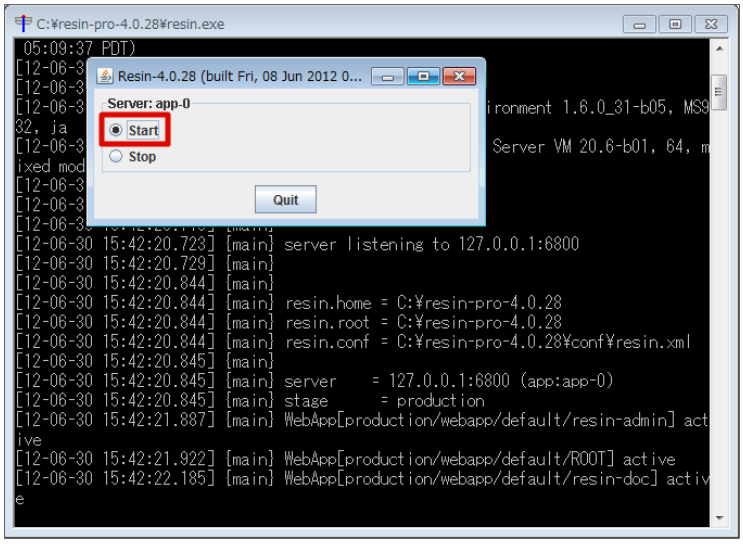
- Resin の起動
  - コンソールから起動する
  - 複数のIPアドレスが設定されている場合
- Resin の停止
  - コンソールから停止する

Resin の起動

> **注意**
>
> Apache Cassandra をご利用の場合は、 Resin の起動前に、Apache Cassandra を起動する必要があります。

> **注意**
>
> Resin を起動するためには 「.NET Framework 3.5」 が必要です。
> Windows Server 2022 環境では、「.NET Framework 4.5」のみインストールされているため、「.NET Framework 3.5」 のセットアップを行ってください。
> 詳細は、*.NET Framework のセットアップ*を参照してください。

1. <%RESIN_HOME%/resin.exe> ファイルをダブルクリックします。
   「起動停止」画面とコンソールが表示されます。



2. コンソールに以下のメッセージが表示されたら起動は完了です。

```
[12-07-02 18:30:00.000] {main} http listening to *:8080
[12-07-02 18:30:00.000] {main}
[12-07-02 18:30:00.000] {main} Resin[id=app-0] started in 55480ms
```

コンソールから起動する

resin.exeはコンソールから実行することが可能です。

フォアグラウンドで起動する場合、引数に「console」を指定します。

```
> cd <%RESIN_HOME%>
> resin.exe console
```

バックグラウンドで起動する場合、引数に「start」を指定します。

```
> cd <%RESIN_HOME%>
> resin.exe start
```

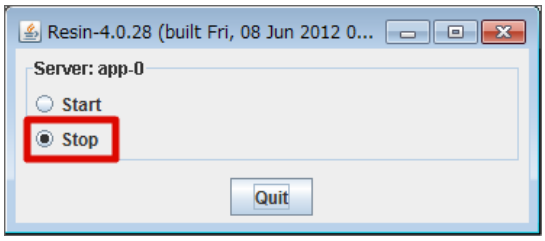
複数のIPアドレスが設定されている場合

起動する環境に複数のIPアドレスが設定されている場合、「resin.exe」をダブルクリックではなく、コンソールより「jgroups.bind_addr」プロパティを指定して起動する必要があります。

```
> cd <%RESIN_HOME%>
> resin.exe console -Djgroups.bind_addr=192.168.1.1
```

Resin の停止

1. 起動時に立ち上がった起動停止画面にて「Stop」をクリックします。



2. コンソールに以下のメッセージが表示されたら停止した状態です。

```
{resin-shutdown} Shutdown Resin reason: OK
```

コンソールから停止する

コンソールからバックグラウンドで起動している場合、停止するには引数に「stop」を指定します。
これは、引数「console」により起動した場合、または「resin.exe」をダブルクリックで起動した場合には利用できません。

```
> cd <%RESIN_HOME%>
> resin.exe stop
```

Windowsサービスへの登録・削除

項目

- Windowsサービスへの登録
  - 分散構成の場合
  - 複数のIPアドレスが設定されている場合
- Windowsサービスの削除

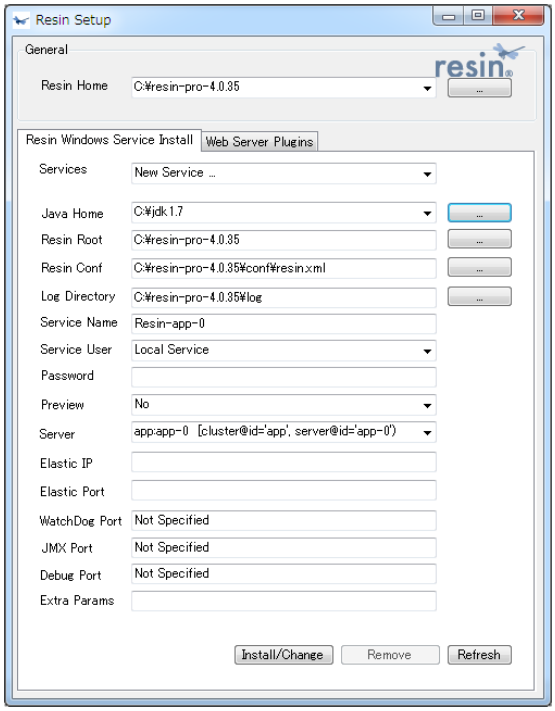
Windowsサービスへの登録

- Resin をインストールした直下にある「setup.exe」を起動します。

```
%RESIN_HOME%/setup.exe
```

> **注意**
> setup.exe を実行する場合は、.NET Framework 3.5 が必要です。

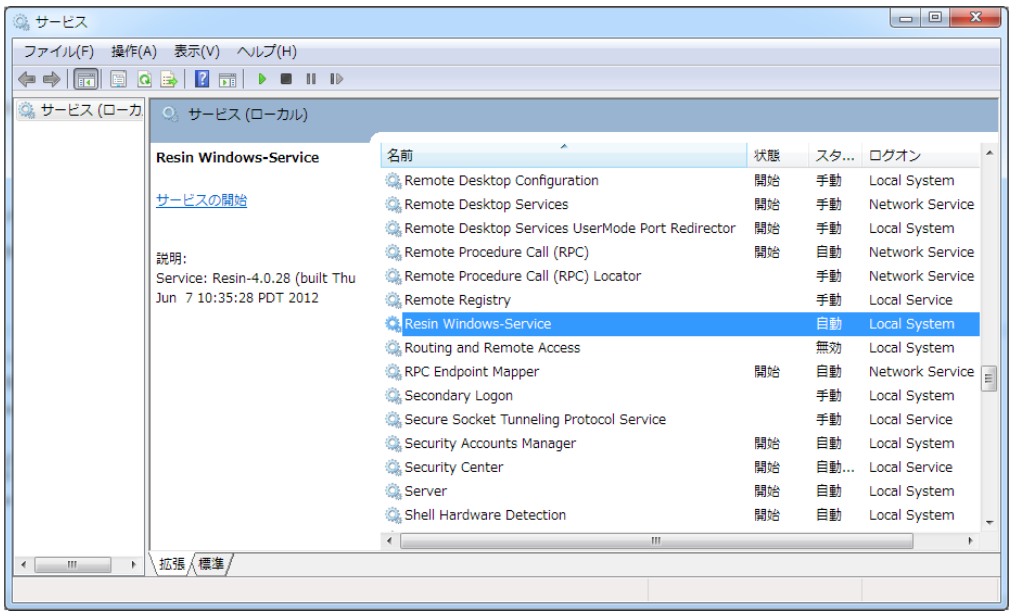
- 起動すると次の画面が表示されます。



- 「Resin Windows Service Install」タブ内の項目を編集します。

> **コラム**
>
> 本書ではサービス名を変更します。
> 変更前：「Resin-app-0」
> 変更後：「Resin Windows-Service」

- 下部にある「Install/Change」ボタンをクリックします。
- 次の画面が表示されたらWindowsサービスへの登録は完了です。
  必要に応じて、スタートアップの種類を「自動」等に変更してください。



分散構成の場合

- 分散構成により複数の Resin を使用している場合、それぞれの Resin に対してWindowsサービスへの登録を行う必要があります。
  「Server」項目に、各サーバに応じて自身のIDを指定してください。
- ダイナミックサーバをWindowsサービスに登録するには一度コマンドプロンプトからダイナミックサーバを起動する必要があります。
  コマンドプロンプトからダイナミックサーバを起動後にsetup.exeを実行し、「Server」項目に表示される「dyn-app-0」を選択してWindowsサービスに登録してください。
  ダイナミックサーバについては、「セットアップガイド - Resinのクラスタリング」を参照してください。
- 「Service User」項目に、共有ディレクトリに対するアクセス権を持つユーザを指定してください。
  Storageの設定については、「*Storage*」を参照してください。

> **コラム**
>
> 同一のOS上に複数の Resin を起動する場合、watchdogのポートが重複しないように設定する必要があります。
> 「WatchDog Port」項目に、ポート番号を指定してください。
> ポート番号を指定しない場合、「6600」が適用されます。
>
> watchdogについては、「設定ファイルリファレンス - watchdog 」を参照してください。

複数のIPアドレスが設定されている場合

起動する環境に複数のIPアドレスが設定されている場合、「Extra Params」項目に、「jgroups.bind_addr」プロパティを指定する必要があります。

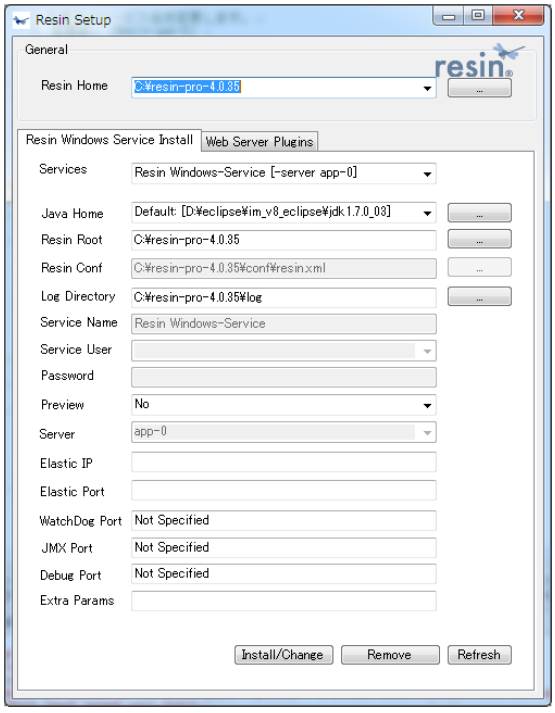
```
-Djgroups.bind_addr=192.168.1.1
```

Windowsサービスの削除

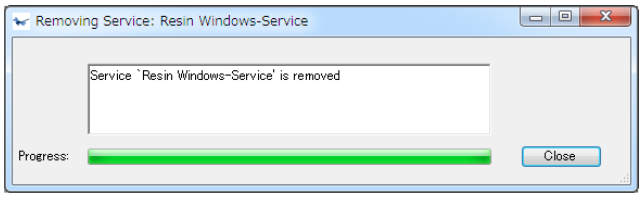
- Resin をインストールした直下にある「setup.exe」を起動します。

  ```
  %RESIN_HOME%/setup.exe
  ```

- 起動すると次の画面が表示されます。



- 下部にある「Remove」ボタンをクリックします。
- 次の画面が表示されたらWindowsサービスへの削除は完了です。



Linux

コンソール起動・停止

項目
- Resin の起動
  - 複数のIPアドレスが設定されている場合
- Resin の停止

Resin の起動

> **注意**
> Apache Cassandra をご利用の場合は、 Resin の起動前に、Apache Cassandra を起動してください。

1. 以下のコマンドを実行します。

```
# <%RESIN_HOME%>/bin/resinctl -server app-0 start
```

> **注意**
> 分散構成の場合、各 Resin がインストールされている環境より起動してください。
> 「app-0」の部分は <%RESIN_HOME%/conf/resin.properties> ファイルの「app_servers」プロパティに設定したIPアドレスに対応するIDを指定してください。

2. コンソールに以下のメッセージが表示されたら起動は完了です。

```
[12-07-02 18:30:00.000] {main} http listening to *:8080
[12-07-02 18:30:00.000] {main}
[12-07-02 18:30:00.000] {main} Resin[id=app-0] started in 55480ms
```

複数のIPアドレスが設定されている場合

起動する環境に複数のIPアドレスが設定されている場合、「jgroups.bind_addr」システムプロパティを指定して起動する必要があります。

```
# <%RESIN_HOME%>/bin/resinctl -server app-0 start -Djgroups.bind_addr=192.168.1.1
```

Resin の停止

1. 以下のコマンドを実行します。

```
# %RESIN_HOME%/bin/resinctl -server app-0 stop
```

2. コンソールに以下のメッセージが表示されたら停止した状態です。

```
{resin-shutdown} Shutdown Resin reason: OK
```

Linuxデーモンへの登録、削除

項目
- RHEL6の場合
  - Linuxデーモンへの登録
    - 複数のIPアドレスが設定されている場合
  - Linuxデーモンからの削除
  - 実行ユーザの変更
- RHEL7の場合
  - Linuxデーモンへの登録
    - 複数のIPアドレスが設定されている場合
  - Linuxデーモンからの削除
  - 実行ユーザの変更

RHEL6の場合

Linuxデーモンへの登録

> **注意**
> *intra-mart Accel Platform （Linux編）のインストールと設定*で説明している方法でインストールされている必要があります。
> また、インストール時にroot権限を持つユーザにて実行されている必要があります。

- chkconfig コマンドを実行して、起動時に自動起動するようにしてください。

```
# chkconfig --add resin
```

- chkconfig コマンドを実行して、以下のように表示されれば成功です。

```
# chkconfig --list resin
resin      0:off   1:off   2:off   3:on    4:on    5:on    6:off
```

複数のIPアドレスが設定されている場合

起動する環境に複数のIPアドレスが設定されている場合、/etc/init.dディレクトリに配置されているresinスクリプトを編集し、システムプロパティ「jgroups.bind_addr」を指定する必要があります。
25行目前後に、環境変数が指定されている部分が存在します。そこに、環境変数「ARGS」を追加し、システムプロパティを追加します。

```
JAVA_HOME="/usr/lib/jvm/java-7-sun"
RESIN_HOME="/usr/local/resin"
CONSOLE="/var/log/resin/console.log"
# システムプロパティ追加
ARGS="-Djgroups.bind_addr=192.168.1.1"
```

Linuxデーモンからの削除

- chkconfig コマンドを実行して、削除してください。

```
# chkconfig --del resin
```

- chkconfig コマンドを実行して、以下のように表示されれば成功です。

```
# chkconfig --list resin
 サービス resin は chkconfig をサポートしますが実行レベルで参照されていません (run 'chkconfig --add resin')
```

- 必要に応じて、/etc/init.d/resin を削除してください。

実行ユーザの変更

/etc/init.dディレクトリに配置されているresinスクリプトを編集することにより、Linuxデーモンとして起動するResinの実行ユーザを変更することが可能です。
25行目前後に、環境変数が指定されている部分が存在します。そこに、環境変数「USER」を追加し、Resinの実行ユーザを指定します。
実行ユーザは事前に作成されている必要があります。

```
JAVA_HOME="/usr/lib/jvm/java-7-sun"
RESIN_HOME="/usr/local/resin"
CONSOLE="/var/log/resin/console.log"
# 実行ユーザ追加
USER=imart
```

RHEL7の場合

以下の説明では下記の環境を想定します。

RESIN_HONE	/usr/local/resin
起動するサーバID	app-0

Linuxデーモンへの登録

> **注意**
> *intra-mart Accel Platform （Linux編）のインストールと設定*で説明している方法でインストールされている必要があります。
> また、インストール時にroot権限を持つユーザにて実行されている必要があります。

- 以下のようなファイルを作成します。
  /usr/lib/systemd/system/resin.service

```
[Unit]
Description=Resin Server

[Service]
Type=forking
User=root
ExecStart=/usr/local/resin/bin/resinctl -server app-0 start
ExecReload=/usr/local/resin/bin/resinctl -server app-0 restart
ExecStop=/usr/local/resin/bin/resinctl -server app-0 stop

[Install]
WantedBy=multi-user.target
```

- systemctl コマンドを実行して、起動時に自動起動するようにしてください。

```
# systemctl enable resin
```

複数のIPアドレスが設定されている場合

起動する環境に複数のIPアドレスが設定されている場合、上記ファイルを編集し、システムプロパティ「jgroups.bind_addr」を指定する必要があります。

```
ExecStart=/usr/local/resin/bin/resinctl -server app-0 start -Djgroups.bind_addr=192.168.1.1
ExecReload=/usr/local/resin/bin/resinctl -server app-0 restart -Djgroups.bind_addr=192.168.1.1
```

Linuxデーモンからの削除

- systemctl コマンドを実行して、削除してください。

```
# systemctl disable resin
```

- 必要に応じて、/usr/lib/systemd/system/resin.service を削除してください。

実行ユーザの変更

上記ファイルを編集することにより、Linuxデーモンとして起動するResinの実行ユーザを変更することが可能です。
実行ユーザは事前に作成されている必要があります。

```
# 実行ユーザを imart に変更
User=imart
```

デプロイ

WAR ファイルのデプロイ

以下の方法でデプロイを行います。

- *webapps ディレクトリに WAR ファイルを直接配置してデプロイ*

> **コラム**
> WARファイルによる複数テナントをご利用の場合は、各テナントにデプロイを実施してください。

> **コラム**
> 既にデプロイ済みである場合は、*WAR ファイルのアンデプロイ*を行ってからデプロイを行ってください。

webapps ディレクトリに WAR ファイルを直接配置してデプロイ

> **注意**
> 再デプロイを繰り返すと <%RESIN_HOME%/resin-data/.git> 配下のディレクトリが肥大化します。 <.git> ディレクトリの使用量を確認し、適時 <.git> ディレクトリを削除するようにしてください。

> **注意**
> WAR ファイルのデプロイには、Resin が停止している必要があります。

> **注意**
> 分散構成でこのデプロイ方法を採用する場合、デプロイ作業を各サーバ上で実施する必要があります。

1. IM-Juggling で作成した WAR ファイルを任意のディレクトリに配置します。
2. WAR ファイルを <%RESIN_HOME%/webapps> 配下に配置します。
3. Resin を起動します。

静的ファイルの配置

IM-Juggling から取得した「静的ファイル」を Web Server のドキュメントルートに解凍します。
静的ファイルを解凍する際には、「*apache設定ファイルの編集*」・「*Internet Information Services（IIS）設定ファイルの編集*」で行った設定時の「静的ファイルの展開フォルダ」を指定してください。

静的ファイルの出力は、*静的ファイルの出力*を参照してください。

テナント環境セットアップ

- Webブラウザを利用して、**テナント環境セットアップ** を行います。
  Webブラウザより 以下のURLへアクセスします。

  システム管理者ログイン画面： *http://<HOST>:<PORT>/<CONTEXT_PATH>/system/login*

  ※ Web ServerとWeb Application Server をローカル環境にセットアップ、かつ、本書の例に合わせてセットアップした場合、次のアドレスで接続できます。
  システム管理者ログイン画面： http://localhost/imart/system/login

  > **コラム**
  >
  > コンテキストパスはデプロイ時に設定したものを指定します。
  > Resin の場合、デプロイ時のWARファイル名です。
  > *デプロイ* を参照してください。
  > Resin 以外の Web Application Server については、デプロイ時に指定した「コンテキスト・ルート」です。

- intra-mart Accel Platform の**テナント環境セットアップ** 手順は次の通りです。

各ウィザードの詳細については下記を参照してください。

システム管理者情報

1. **システム全体の管理者** を設定します。

項目	必須/任意	説明
ユーザコード	必須	システム管理者のユーザコードを入力します。 （例：system）
パスワード	任意	システム管理者のパスワードを入力します。
パスワード(確認)	任意	システム管理者のパスワードを入力します。
ロケール	必須	システム管理者のロケールを選択します。 初期表示されているロケールは、アクセスしているブラウザのロケール設定です。

コラム

“任意”項目は、必要に応じて入力してください。必要がなければ入力の必要はありません。

テナント情報

1. **テナントの基本的な情報** を設定します。

項目	必須/任意	説明
テナントID	必須	テナントのIDを入力します。 「DataSourceマッピングの設定」の <tenant-id> で設定した値を入力します。 DataSourceマッピングの設定を行っていない場合は、「**default**」を入力します。
デフォルトロケール	必須	テナントのデフォルトロケールを選択します。 初期表示されているロケールは、アクセスしているブラウザのロケール設定です。 **このロケールは運用中に変更することは推奨していません。** **運用に応じたロケールを設定してください。**
タイムゾーン	必須	テナントのタイムゾーンを選択します。
アカウントライセンス数	必須	テナントのアカウントライセンス数を入力します。 **テナント管理者を登録するため「1」以上を入力してください。**

注意

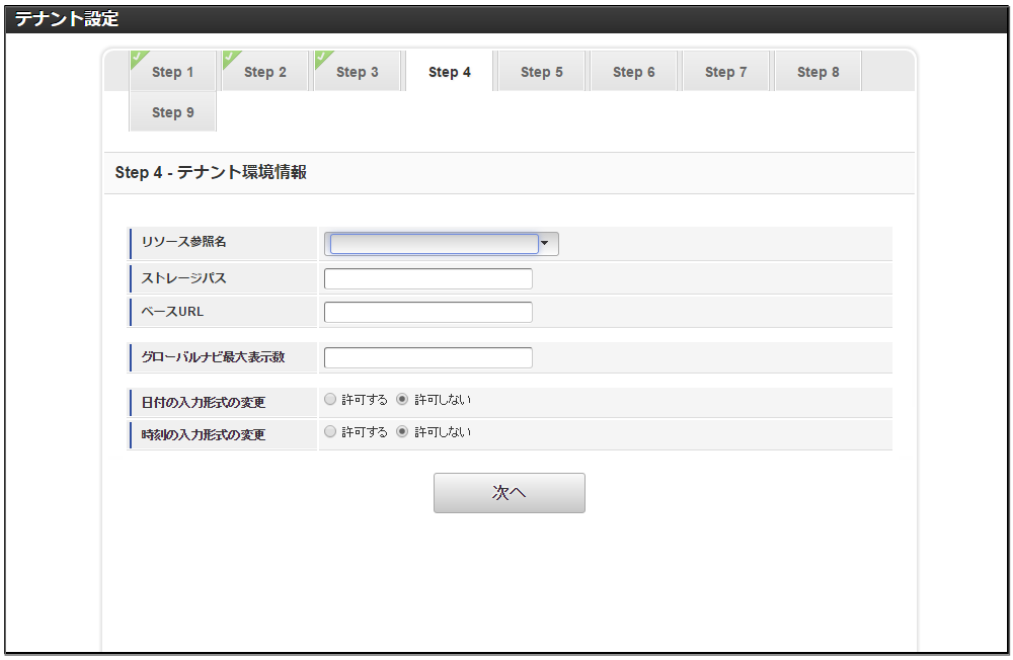
試用版利用などにおいてサンプルデータセットアップを行う場合は、「アカウントライセンス数」は **無制限** を選択する事を推奨します。

テナント環境情報

1. **テナントの環境情報** を設定します。

コラム

「Resinデータソース設定」モジュールまたは「Payaraデータソース設定」モジュールを適用している場合、リソース参照名にセレクトボックスが表示されます。
モジュールが適用されていない場合はリソース参照名は表示されません。



項目	必須/任意	説明
リソース参照名	任意	リソース参照名を選択します。
ストレージパス	任意	ストレージパスを入力します。
ベースURL	任意	ベースURLを入力します。
グローバルナビ最大表示数	任意	グローバルナビの最大表示件数を入力します。

項目	必須/任意	説明
日付の入力形式の変更	必須	「許可する」を選択した場合、「日付と時刻の形式」にて日付の入力形式の変更が可能です。 初期値は「許可しない」です。
時刻の入力形式の変更	必須	「許可する」を選択した場合、「日付と時刻の形式」にて時刻の入力形式の変更が可能です。 初期値は「許可しない」です。

コラム

ストレージパスに storage-config.xml の <storage-directory-name> を付加したパスが、パブリックストレージパス %PUBLIC_STORAGE_PATH% です。
（例）
ストレージパスに /var/imart と入力し、設定ファイル storage-config.xml の <storage-directory-name> に storage と設定した場合、%PUBLIC_STORAGE_PATH% は /var/imart/storage です。

コラム

未指定の場合は、それぞれの設定ファイルの内容が有効です。

パスワード保存方式設定

1. **アカウントパスワードの保存方式** を設定します。

コラム

パスワード保存方式設定は 2016 Spring(Maxima) から利用可能です。
2016 Spring(Maxima) より前のバージョンでの保存方式は「暗号化」です。

項目	必須/任意	説明

項目	必須/任意	説明
パスワード保存方式	必須	データベース内で保持するアカウントパスワードの保存方式です。 以下の２つの方式を選択可能です。 ▪ 暗号化：アカウントパスワードを暗号化して保存します。キーを用いる事でパスワードの復号化(平文パスワードの取得)が可能です。 ▪ ハッシュ化：アカウントパスワードのハッシュ値を保存します。復号化(平文パスワードの取得)はできません。

注意

「ハッシュ化」を選択し登録した場合、以下の制限があります。

- 設定値を基にアカウントパスワードのハッシュ化を行うため一度設定した「ハッシュ化」の設定値を変更することはできません
- 平文パスワードを復元する方法が失われるため「暗号化」に戻すことは出来ません
- **平文パスワードを復元する方法が失われるため一部の機能がご利用いただけなくなります**
  - 詳細は 要件情報公開サイトを参照してください

以下の項目はパスワード保存方式で「ハッシュ化」を選択した場合のみ入力必須です。

項目	必須/任意	説明
ハッシュアルゴリズム	必須	ハッシュ文字列を生成する計算式（関数）です。 intra-mart Accel Platform では以下の値が利用可能です。 ▪ SHA-256 ▪ SHA-384 ▪ SHA-512
ソルト値	必須	ハッシュ文字列を生成するためのパラメータの1つです。 パスワードのハッシュ化において、パスワード値にソルト値を付与した値に対してハッシュアルゴリズムによる計算を行った値がハッシュ文字列（パスワードとして保存される値）となります。 ソルト値はユーザ毎に異なる値を利用するため、異なるユーザが同じパスワードを利用していた場合も保存されている値は異なる値となります。 ソルト値の生成式は以下の通りです。 `対象ユーザのユーザコード + "|" + テナントID + "|" + テナント属性で永続化されているソルトサフィックス`
ストレッチング回数	必須	ハッシュ文字列を生成するためのパラメータの1つです。 パスワードのハッシュ化において、ハッシュアルゴリズムによる計算をストレッチング数の回数繰り返して生成された値がハッシュ文字列（パスワードとして保存される値）となります。 ストレッチング回数が多いほどパスワードの特定が難しくなりますが、ハッシュ化するための負荷が高くなります。

テナント管理者情報

1. **テナントの管理者** を設定します。

項目	必須/任意	説明
ユーザコード	必須	テナント管理者のユーザコードを入力します。 （例：tenant）
パスワード	任意	テナント管理者のパスワードを入力します。
パスワード(確認)	任意	テナント管理者のパスワードを再入力します。

コラム

“任意”項目は、必要に応じて入力してください。必要がなければ入力の必要はありません。

コラム

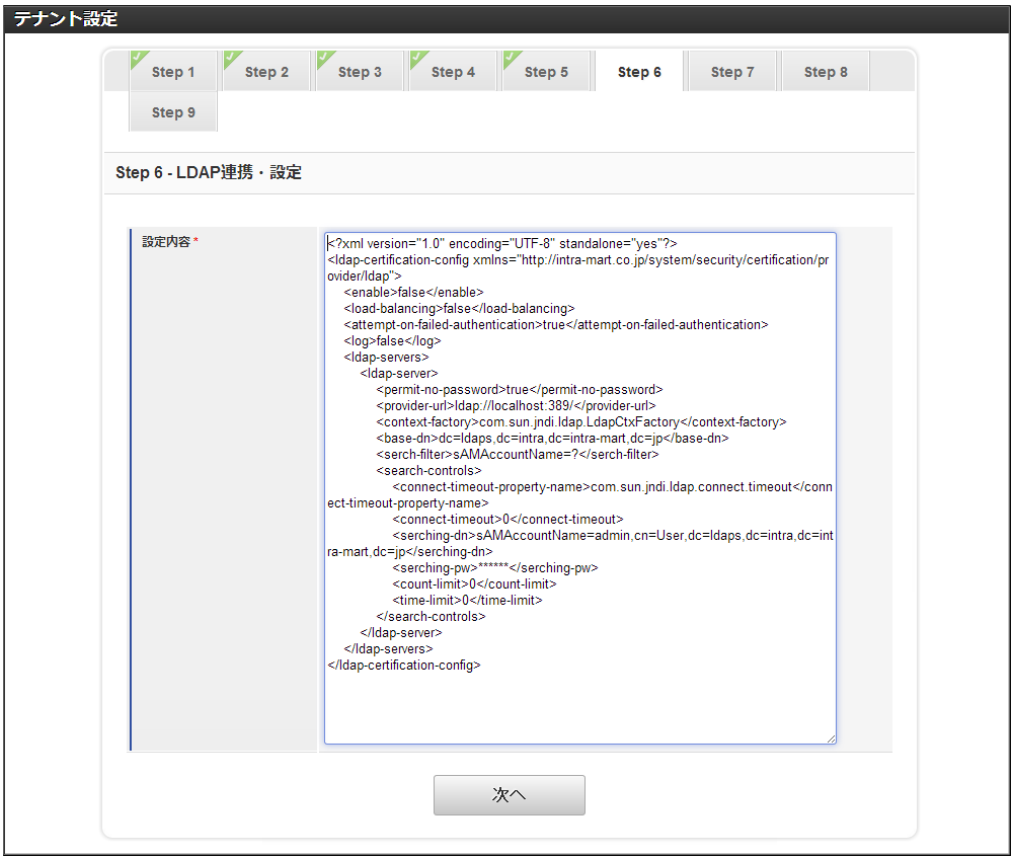
テナント環境セットアップ中にエラーが発生してセットアップが中断してしまった場合、 再度セットアップを実施してもテナント管理者は登録されません。
このように正常にテナント管理者を登録できなかった場合、テナント管理画面から改めてテナント管理者を登録することができます。
詳しくは「システム管理者操作ガイド」-「テナント管理」の「テナント管理者を新規に作成する」の項を参照してください。

LDAP連携・設定

注意

IM-Jugglingにおいて、 **LDAP認証モジュール** を選択した場合のみ、この画面が表示されます。

1. **LDAP連携・設定** 情報を設定します。



コラム

初期表示は IM-Juggling のプロジェクト/conf/ldap-certification-config.xmlの内容が表示されます。
ただし、<enable>タグの値はfalseとなっています。

注意

<enable>タグの内容がtrueである場合のみLDAP認証が有効となります。
LDAP認証を有効とする場合、認証先であるLDAPの設定を正しく行ってください。

項目	必須/任意	説明
設定内容	必須	LDAP認証の有効/無効、および、認証先であるLDAPの情報を入力します。 入力内容については「intra-mart Accel Platform セットアップガイド」-「LDAP認証設定ファイル」の説明を参照してください。

ログインセッション管理

注意

IM-Jugglingにおいて、 **ログインセッション管理モジュール** を選択した場合のみ、この画面が表示されます。

1. 一般ユーザのログイン時に二重ログインを検出した場合の動作を設定します。

ログインセッション管理の設定内容は以下のとおりです。

- 標準の認証エラーページを表示する

  標準の認証エラーページを表示します。

- 二重ログインの検出を表示する

  二重ログインを検出したことを一般ユーザに通知します。
  一般ユーザは通知された画面からログインを再試行することができます。

- ログインユーザによるセッションの無効化を許可する

  二重ログインを検出したことを一般ユーザに通知します。
  一般ユーザは通知された画面からログイン中のセッションを強制的に無効化しログインをすることができます。

Apache Cassandra接続情報

> **注意**
>
> IM-Jugglingにおいて、 **IMBoxモジュール** を選択した場合のみ、この画面が表示されます。

1. Apache Cassandra接続情報を入力します。

注意

Apache Cassandra接続情報はIMBox利用時のみ表示されます。
テナント環境セットアップを実行する前にApache Cassandraの設定、起動が行われている必要があります。
Apache Cassandraの設定に関しての詳細は「IMBox Cassandra管理者ガイド」を参照してください。

項目	必須/任意	説明
クラスタ名	必須	Cassandraサーバのクラスタ名を入力します。 （例：IMBox Cluster）
キースペース	必須	Cassandraサーバのキースペースを入力します。 （例：default）
接続先	必須	Cassandraが稼働しているサーバのIPアドレスとポート番号を入力します。 接続先は「IPアドレス」または、「IPアドレス:ポート番号」の形式で入力します。（ポート番号を省略した場合、9160を利用します。） 9160以外のポート番号を指定した場合、新規ノードの検出機能にてエラーが発生します。 分散構成で複数のCassandraが稼働している場合、すべての接続先を1行ずつ入力してください。 （例：127.0.0.1:9160）
レプリケーションファクタ	必須	クラスタ内部のデータのレプリカ数を入力します。 レプリケーションファクタは、キースペース作成時のみ使用されます。 （例：1）
認証情報設定	任意	Cassandraへの接続における認証の利用を選択します。 認証情報を設定する場合には書込み権限の確認を行うため、事前にキースペースを作成しておく必要があります。
認証ユーザ名	認証設定利用時のみ必須	Cassandraへの認証接続における接続ユーザ名を入力します。 認証設定利用時のみ表示されます。 （例：admin）
認証パスワード	認証設定利用時のみ必須	Cassandraへの認証接続におけるパスワードを入力します。 認証設定利用時のみ表示されます。 （例：admin）
テスト接続		入力した内容でCassandraが接続可能であるかのテストが行えます。 テナント作成時には、必ず行うことを推奨します。

コラム

“任意”項目は、必要に応じて入力してください。必要がなければ入力の必要はありません。

コラム

Cassandra接続情報の登録時の初期値は、Cassandraサーバ接続設定（cassandra-config.xml）の設定値となります。
cassandra-config.xmlに関する詳細は「intra-mart Accel Platform セットアップガイド」-「IMBox」を参照してください。

Apache Solr接続情報

注意

- IM-Jugglingにおいて、 **IM-ContentsSearchモジュール** を選択した場合のみ、この画面が表示されます。
- IM-ContentsSearch for Accel Platformを利用する場合は、Apache Solrのセットアップが必要となります。
  Apache Solrのセットアップに関しての詳細は「Apache Solr」の「セットアップ」を参照してください。

1. Apache Solr接続情報を入力します。

項目	必須/任意	説明
Solr接続情報を設定する	任意	テナント環境セットアップ時にApache Solr接続情報を設定するかどうかをこのボタンで切り替えることができます。 「Solr接続情報」ボタンを未選択状態にすることで、Apache Solr接続情報を設定せずにテナント環境セットアップを実行することができます。
グループID	固定(“default”)	標準接続先のグループIDは”default”から変更することはできません。
標準接続先	必須（「Solr接続情報」ボタンが有効になっている場合のみ）	接続先を選択します。 「Solr接続情報」ボタンが選択状態になっている場合のみ選択することができます。 他のテナントの標準接続先に設定されている接続先は表示されません。

Solr接続情報の設定方法

テナント環境セットアップ時に、システムデータベースへ登録するSolr接続情報を設定する方法は以下の通りです。

- **設定ファイルから変更する方法**

  初回のテナント環境セットアップ前にSolrサーバ接続設定ファイル（solr-config.xml）を設定しておくことで、solr-config.xmlの設定（<group name=”default”>）で定義されている設定値がSolr接続情報としてシステムデータベースへ登録されます。
  Solrサーバ接続設定ファイル（solr-config.xml）は初回のテナント環境セットアップ時、または、Solr接続設定が1件も登録されていない場合に標準接続先の項目として利用します。
  solr-config.xmlに関する詳細は「intra-mart Accel Platform セットアップガイド」-「IM-ContentsSearch」を参照してください。

- **画面から変更する方法**

  別のテナントを新規作成するときは「Solr接続設定」画面で登録したSolr接続情報が標準接続先の選択項目として表示されます。
  システム管理者メニューのテナント管理画面で、作成したテナントにApache Solr接続情報を設定することができます。
  「Solr接続設定」画面に関する詳細は「 システム管理者操作ガイド 」-「 Solr接続設定 」を参照してください。

多要素認証機能

多要素認証とは、「知る要素」「持つ要素」「備える要素」の複数の要素を組み合わせて行う認証です。
intra-mart Accel Platform では、ユーザコード・パスワード（「知る要素」）に加え、認証アプリにより発行される確認コード（「持つ要素」）を利用してログインを行う機能を提供します。
多要素認証機能の設定は以下の通りです。

> **注意**
>
> IM-Juggling において、**多要素認証機能モジュール**を選択した場合のみ、この画面が表示されます。

1. 多要素認証機能の動作を設定します。

項目	必須/任意	説明
多要素認証適用ポリシー	必須	多要素認証機能の運用に関するポリシーを以下から選択できます。 適用しない　多要素による認証を行いません。 ユーザが任意で適用　ユーザが個人設定の多要素認証設定を行うことにより、ログイン時にユーザコード・パスワードと多要素による認証が行われます。 強制的に適用　多要素による認証を行うことをユーザに強制します。 多要素認証の設定が行われていないユーザに対して、ログイン時に設定を要求します。
バックアップコードの生成数	必須	バックアップコードを生成する数を設定できます。
ブラウザ情報管理機能	必須	ブラウザ情報管理を有効にすることで、ユーザが多要素による認証時に認証を行うブラウザ情報を信頼済みのブラウザとして記憶できます。 信頼済みのブラウザからログインを行う場合、次回のログインから多要素による認証を省略できます。
ブラウザ情報の信頼期間(日)	必須	ブラウザ情報管理機能を有効にした場合にブラウザ情報を信頼する期間を日数で設定します。 設定されている日数を過ぎたブラウザでは、再び多要素による認証を要求します。
認証アプリ発行者情報	必須	認証アプリにアカウントを登録する際に付与する発行者情報を設定します。

ベクトルデータベース接続情報

注意
- IM-Juggling において、**IM-Copilot モジュール** を選択した場合のみ、この画面が表示されます。
- intra-mart Accel Platform で標準で利用可能なベクトルデータベースは Apache Solr または PostgreSQL です。
- Apache Solr をベクトルデータベースとして利用する場合は **IM-ContentsSearch モジュール** が必要です。
  IM-ContentsSearch for Accel Platform を利用する場合は、Apache Solrのセットアップが必要です。
  Apache Solr のセットアップに関しての詳細は「Solr管理者ガイド」の「Solrのベクトル検索機能の利用」を参照してください。
- PostgreSQL を利用する場合は、pgvector のインストールと拡張機能の有効化が必要です。
  インストール手順に関しての詳細は「*pgvector のインストール*」を参照してください。

コラム

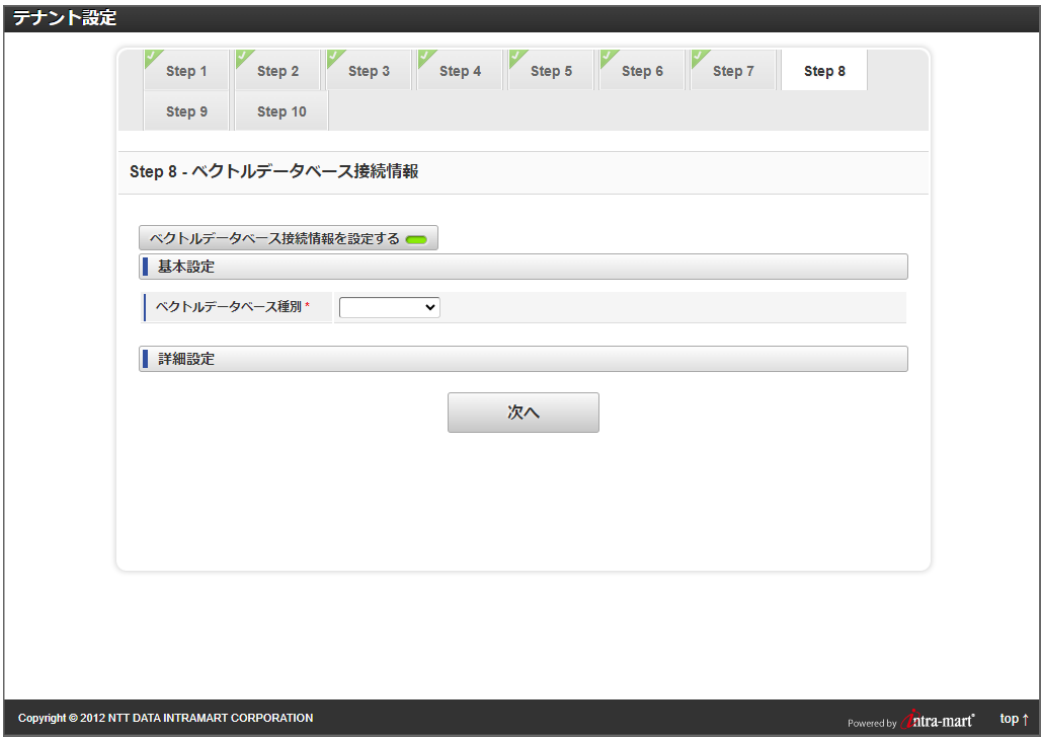
ローコードまたはプロコードでベクトルデータベースを利用する場合や以下のアシスタント機能を利用する場合は、ベクトルデータベース接続情報を設定してください。

- Wiki アシスタント
  詳細は「IM-Knowledge管理者操作ガイド」の「Wiki アシスタント」を参照してください。
- SQL ビルダアシスタント
  詳細は「ViewCreator 管理者操作ガイド」の「SQLビルダ アシスタント」を参照してください。

コラム

ベクトルデータベース接続情報は 2024 Autumn(Jasmine) から利用可能です。

1. ベクトルデータベース接続情報を入力します。



項目	必須/任意	説明
ベクトルデータベース接続情報を設定する	任意	テナント環境セットアップ時にベクトルデータベース接続情報を設定するかどうかをこのボタンで切り替えることができます。 ボタンを未選択状態にすることで、ベクトルデータベース接続情報を設定せずにテナント環境セットアップを実行することができます。
ベクトルデータベース種別	必須	ベクトルデータベースの種別を選択します。

Apache Solr をベクトルデータベースとして使用する場合の設定方法

Solrサーバ接続設定ファイル（solr-config.xml）や「Solr接続設定」画面で登録したSolr接続情報を接続先として選択できます。
solr-config.xmlに関する詳細は「intra-mart Accel Platform セットアップガイド」の「IM-ContentsSearch」を参照してください。
「Solr接続設定」画面に関する詳細は「システム管理者操作ガイド」の「Solr接続設定」を参照してください。

項目	必須/任意	説明
Solr 接続先	必須	ベクトルデータベースとして使用するSolr接続情報を選択します。

PostgreSQL をベクトルデータベースとして使用する場合の設定方法

pgvector がインストールされた PostgreSQL を接続先に設定しているテナントデータベースまたはシェアードデータベースを接続先として選択できます。

> **コラム**
>
> PostgreSQL 以外をテナントデータベースに利用している場合も、シェアードデータベースとして PostgreSQL を用意することで使用できます。

シェアードデータベースはデータソースマッピング設定ファイル（data-source-mapping-config.xml）や「シェアードデータベース設定」画面で接続先を設定できます。
data-source-mapping-config.xmlに関する詳細は「設定ファイルリファレンス」の「データソースマッピング設定」を参照してください。
「シェアードデータベース設定」画面に関する詳細は「システム管理者操作ガイド」の「シェアードデータベース設定」を参照してください。

項目	必須/任意	説明
接続先データベース	必須	ベクトルデータベースとして使用するデータベースを選択します。

コラム

ベクトルデータベース種別の選択候補に pgvector が表示されない場合について

ベクトルデータベース種別の選択候補に pgvector が表示されない場合は、以下の点について確認してください。

- PostgreSQL に pgvector がインストールされていること。
- ベクトルデータベースとして使用するデータベースの拡張機能 *vector* が有効になっていること。

```
CREATE EXTENSION vector;
```

- テナントデータベースをベクトルデータベースとして使用する場合、テナント環境情報のリソース参照名に該当データベースへの接続情報が設定されているリソース参照名が指定されていること。
- シェアードデータベースをベクトルデータベースとして使用する場合、シェアードデータベースの接続設定に該当データベースへの接続情報が登録されていること。

コラム

PostgreSQL によるベクトルデータベース構築について

PostgreSQL をベクトルデータベースとして構築する場合、通信はベクトルデータの更新・検索時に限定されるため、常時高負荷な通信が発生するわけではありません。
ただし、データ量や更新・検索の頻度によっては、適切なハードウェアリソースの調整が必要になる場合があります。
実際の運用状況に合わせてスペックを検討してください。

注意

シェアードデータベースをベクトルデータベースとして使用する場合、他のテナントで使用しているシェアードデータベースを接続先データベースに設定しないように注意してください。
ベクトルデータの格納先が重複すると、テナント固有の情報を他のテナントから更新・参照される可能性があります。

> **注意**
>
> Web Application Server に Oracle WebLogic Server または WebSphere Application Server を使用している環境では、シェアードデータベースをベクトルデータベースとして使用することはできません。
> 詳細については以下の制限事項を参照してください。
>
> - Oracle WebLogic Server 12c R2(12.2.1) では、ベクトルデータベースにシェアードデータベースを利用できません。
> - WebSphere Application Server 9.0.5 では、ベクトルデータベースにシェアードデータベースを利用できません。

登録

1. 各ステップの設定が完了したら、「登録」をクリックします。

> **注意**
> セットアップでエラーが発生した場合、「 セットアップで困ったら・・・ 」を参照してください。

セットアップが完了すると、下記の画面が表示されます。

注意

「 *プロジェクトの作成とモジュールの選択* 」で選択したモジュールによりウィザードのステップ数が異なります。

コラム

旧アップデート版における **テナント環境セットアップ** については、*intra-mart Accel Platform 2013 Winter までのテナント環境セットアップ* を参照してください。

ライセンスの登録

ライセンスについて

ライセンスには、カスタマーサクセスライセンスとパッケージライセンスが存在します。
カスタマーサクセスライセンスは、intra-mart Accel Platform 2023 Spring(Gerbera) 以降より、ご利用いただけます。

> **注意**
> ライセンスには、対象となる製品のバージョン情報が含まれています。
> 製品バージョンと一致していないライセンスはご利用できません。

カスタマーサクセスライセンス

カスタマーサクセスライセンスは、 intra-mart Accel Platform ライセンスとアプリケーションライセンスが存在します。
このライセンスは、intra-mart Accel Platform 2023 Spring(Gerbera) 以降より、ご利用いただけます。

intra-mart Accel Platform ライセンス

ユーザ数の制限はありませんが、サーバ台数（コア数）に制限があります。
環境に合わせ必要な数量をご購入頂く必要があります。
有効期限が存在します。

アプリケーションライセンス

ユーザ数に制限がありますが、サーバの台数に制限はありません。
有効期限が存在します。

パッケージライセンス

パッケージライセンスは、 intra-mart Accel Platform ライセンスとアプリケーションライセンスが存在します。
パッケージライセンスは、製品ごとに登録が必要です。

intra-mart Accel Platform ライセンス

intra-mart Accel Platform 本体に適用されるライセンスです。
intra-mart Accel Platform ライセンスには通常ライセンス、時限ライセンス、試用版ライセンスの3つの種別が存在します。

- **通常ライセンス**
  通常ライセンスはお客様向けに発行する正式ライセンスです。
  ユーザ数の制限はありませんが、サーバ台数に制限があります。
  1サーバ(war)あたり、1ライセンスキーが必要であり、分散構成時には台数分のライセンスキーが必要です。
  通常ライセンスには有効期限はありません。
- **時限ライセンス**
  時限ライセンスは年月日で指定された有効期限が存在するライセンスです。
  ユーザ数やサーバ台数に制限はありません。
- **試用版ライセンス**
  試用版ライセンスはライセンス登録がされていない場合に、自動で適用されるライセンスです。
  ユーザ数に制限はありません。
  インストールしてから60日間の有効期限があります。
  試用版ライセンスを用いて分散環境を構築することはできません。

アプリケーションライセンス

intra-mart Accel Applications に適用されるライセンスです。
アプリケーションライセンスには通常ライセンス、時限ライセンス、試用版ライセンスの3つの種別が存在します。

- **通常ライセンス**
  通常ライセンスはお客様向けに発行する正式ライセンスです。
  ユーザ数に制限がありますが、サーバの台数に制限はありません。
  通常ライセンスには有効期限はありません。
- **時限ライセンス**
  時限ライセンスは年月日で指定された有効期限が存在するライセンスです。

ユーザ数やサーバ台数に制限はありません。

- **試用版ライセンス**
  試用版ライセンスはライセンス登録がされていない場合に、自動で適用されるライセンスです。
  ユーザ数に制限はありません。
  インストールしてから60日間の有効期限があります。

ライセンスの登録

ライセンスによって、登録手順が異なります。

カスタマーサクセスライセンスの場合

> **注意**
>
> ASEAN地域向けのローコードとプロコードについては、「*カスタマーサクセスライセンスの場合*」の手順ではなく、「*パッケージライセンスの場合*」を参照してください。

1. システム管理者の「メニュー」画面を表示します。
   メニューから「ライセンス管理」をクリックします。
2. 環境をアクティベートして、ライセンスを登録します。

   詳細は「システム管理者操作ガイド」-「ライセンス管理」を参照してください。

> **コラム**
>
> 環境のアクティベートが終わると自動的に環境がライセンスポータルと通信し、ご契約内容の変更が自動反映されます。
> 環境からライセンスポータルへの通信にプロキシサーバを利用する場合は、「*Resinの設定*」に次の JVM 引数の設定が必要です。

-Dhttps.proxyHost	プロキシサーバのホストURL
-Dhttps.proxyPort	プロキシサーバのポート番号
-Dhttps.proxyUser	プロキシサーバへの接続ユーザ
-Dhttps.proxyPassword	接続ユーザのパスワード

> **注意**
>
> intra-mart Accel Applications のライセンスの場合は、次の設定が必要です。
>
> **テナント毎のライセンス設定**
> バーチャルテナントによる複数テナント が存在する場合には、システム管理者がそれぞれのテナントに対して、ライセンス数を割り当てる必要があります。
>
> 詳細は「システム管理者操作ガイド」-「ライセンス設定」を参照してください。
>
> **アプリケーションライセンス設定**
> アプリケーションライセンスはアプリケーションを利用するユーザごとに付与する必要があります。
> アプリケーションライセンスが付与されていないユーザは、そのアプリケーションの画面を開くことができません。
> アプリケーションによってはAPIを実行できないものもあります。
>
> 詳細は「テナント管理者操作ガイド」-「アプリケーションライセンス一覧を使用する」を参照してください。

> **注意**
>
> **Resin の分散環境を構築する場合**
> Resin の分散環境構築には、Resin Proのライセンスが必要です。
> Caucho社のWebサイトよりダウンロードした Resin Proにはライセンスキーが含まれていません。
> Resin Proのライセンスキーは intra-mart Accel Platform 本体に同梱している扱いとなるので、サーバ数分の intra-mart Accel Platform ライセンスを入手して頂く必要があります。

パッケージライセンスの場合

1. システム管理者の「メニュー」画面を表示します。
   メニューから「ライセンス管理」をクリックします。

2. 「ライセンスキー登録」からライセンスの登録を行ってください。

詳細は「システム管理者操作ガイド」-「ライセンス管理」を参照してください。

注意

intra-mart Accel Applications のライセンスの場合は、次の設定が必要です。

テナント毎のライセンス設定

バーチャルテナントによる複数テナント が存在する場合には、システム管理者がそれぞれのテナントに対して、ライセンス数を割り当てる必要があります。

詳細は「システム管理者操作ガイド」-「ライセンス設定」を参照してください。

アプリケーションライセンス設定

アプリケーションライセンスはアプリケーションを利用するユーザごとに付与する必要があります。
アプリケーションライセンスが付与されていないユーザは、そのアプリケーションの画面を開くことができません。
アプリケーションによってはAPIを実行できないものもあります。

詳細は「テナント管理者操作ガイド」-「アプリケーションライセンス一覧を使用する」を参照してください。

注意

Resin の分散環境を構築する場合

Resin の分散環境構築には、Resin Proのライセンスが必要です。
Caucho社のWebサイトよりダウンロードした Resin Proにはライセンスキーが含まれていません。
Resin Proのライセンスキーは intra-mart Accel Platform 本体に同梱している扱いとなるので、サーバ数分の intra-mart Accel Platform ライセンスを入手して頂く必要があります。

アップデート・パッチの適用・モジュール構成の変更

- IM-Juggling を利用して最新モジュールの適用や、モジュール構成を変更する事ができます。

アップデート

- アップデートとは、一定の期間内でリリースする機能追加、不具合修正を含めた形で提供するリリース形態です。
  この他に弊社から提供している、Resin の更新、システム要件（サポート環境・検証済み環境）の改変も含まれます。

Resin のアップデート後に必要なメンテナンス作業

【 TERASOLUNA Global Framework を利用している場合】

- ライブラリ修正
  1. Resin のライブラリからjarを削除
     warファイルをデプロイする前に、<%RESIN_HOME%/lib>からHibernate Validatorのライブラリを削除します。
     詳細は、「*TERASOLUNA Server Framework for Java (5.x) for Accel Platform を使用する場合の設定*」を参照してください。

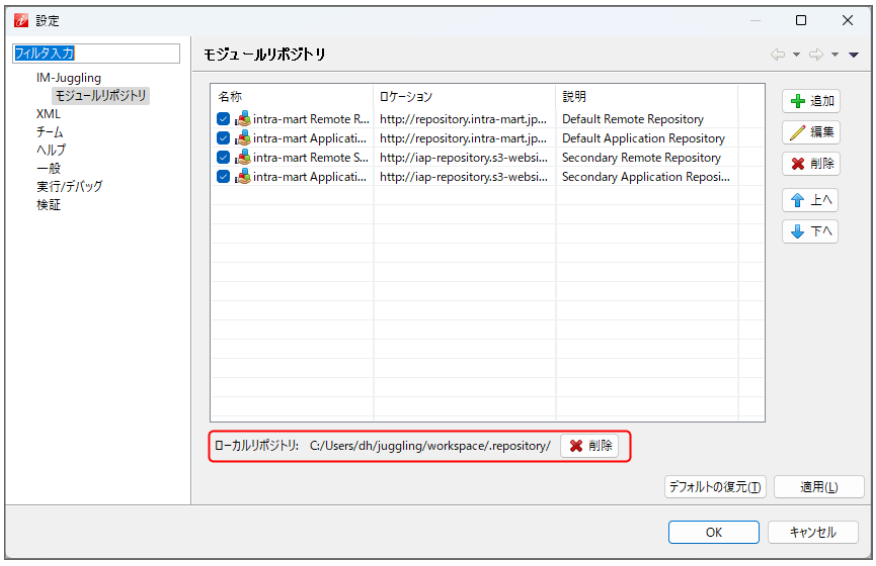
モジュールのアップデート

項目

- 古いリポジトリ情報の削除
- アップデート対象の選択
- IM-Juggling で必要なメンテナンス作業（アップデートによる設定項目のメンテナンス等）
- WARファイルの出力
- WARファイルのアンデプロイ
- Web Application Server の更新
- WARファイルの再デプロイ
- 静的ファイルの出力と再配置
- テナント環境セットアップ
- テナント環境セットアップ後の各種メンテナンス

古いリポジトリ情報の削除

- 前回使用した古いリポジトリ情報を削除します。
  1. IM-Juggling を起動し、ウィンドウ内-ツールバー右端にある「設定」-「IM-Juggling」-「モジュールリポジトリ」を開きます。
  2. 下部にある「ローカルリポジトリ」の削除をクリックしてください。



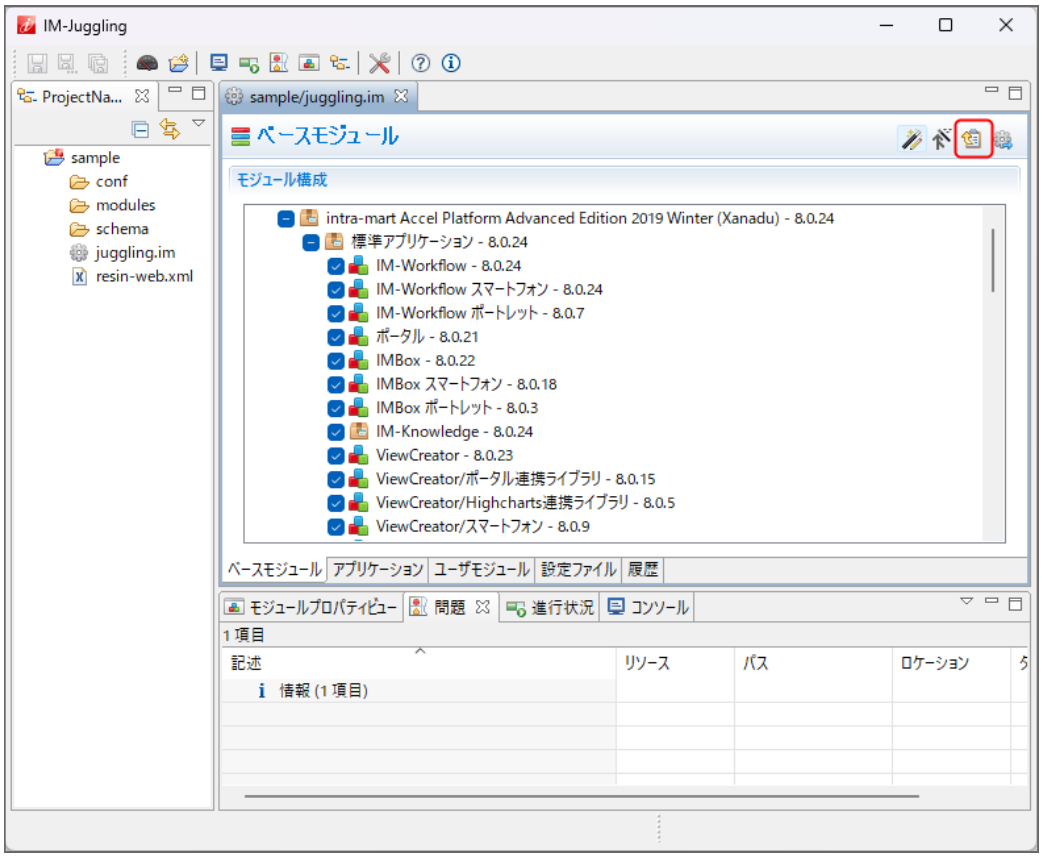
> **注意**
>
> ファイルの削除処理が行われている間は IM-Juggling の操作が行えません。

3. 削除が完了したらOKをクリックします。

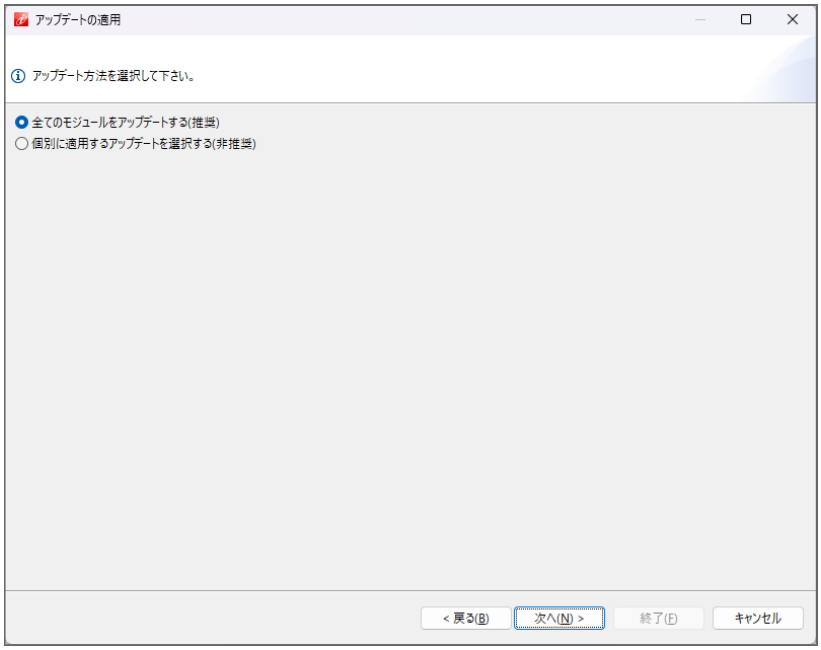
アップデート対象の選択

- 前述の「古いリポジトリ情報の削除」を実施している事が前提です。

1. 画面右上の「アップデート」アイコンをクリックします。

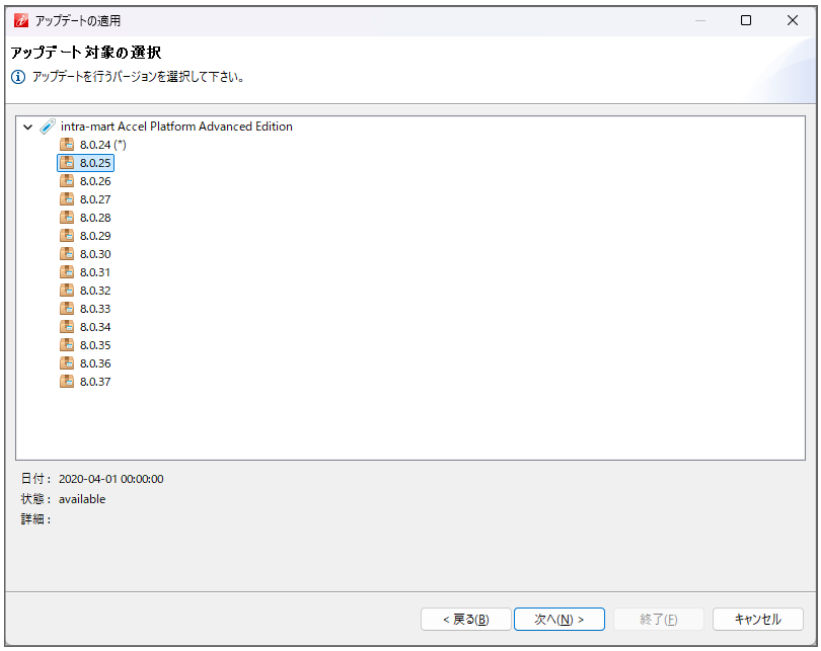


2. アップデート方法を選択します。
   通常は「全てのモジュールをアップデートする（推奨）」を選択してください。
   選択後、「次へ（N）」をクリックします。



> **注意**
>
> 「個別に適用するアップデートを選択する（非推奨）」を選択した場合は、今後のアップデートが正常に行えない可能性があります。

3. ベースモジュールのアップデート対象を選択します。
   選択後、「次へ（N）」をクリックします。



注意

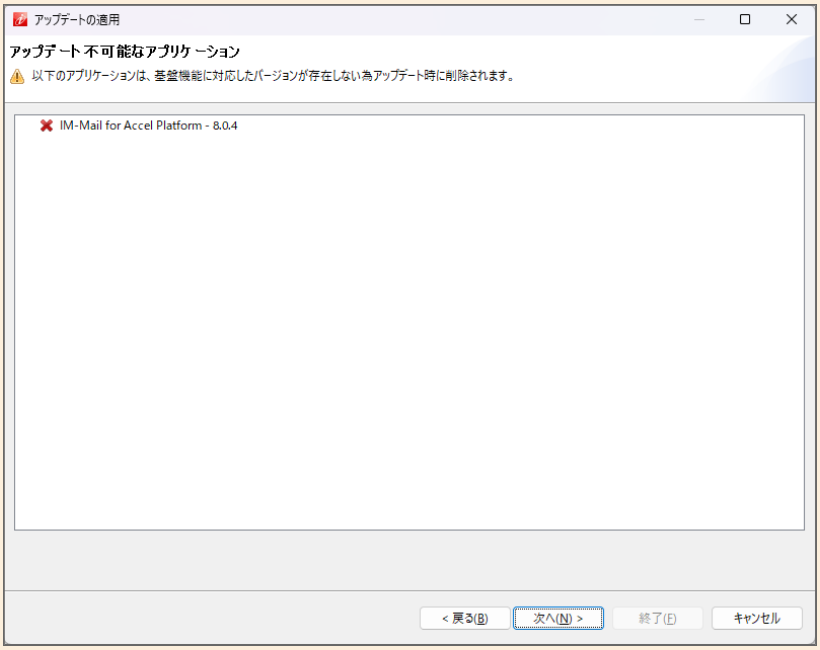
既存モジュールのバージョンがすべて同一の場合（上図では「8.0.24」を指します）、同一バージョン（「8.0.24（*）」）に対して適用できるモジュールは存在しません。
同一バージョン（「8.0.24（*）」）を選択しても、一覧には表示されません。

注意

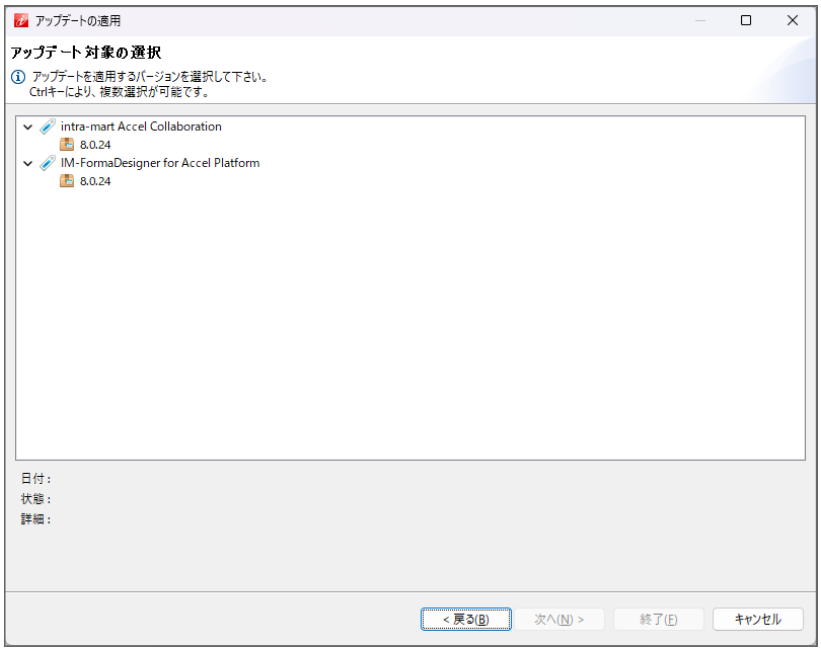
ベースモジュールのアップデート内容によっては、既存のアプリケーションが利用できない場合があります。

例：既存のプロジェクトにおいて次の構成だった場合
　ベースモジュール：8.0.24
　アプリケーション（IM-Mail）：8.0.4

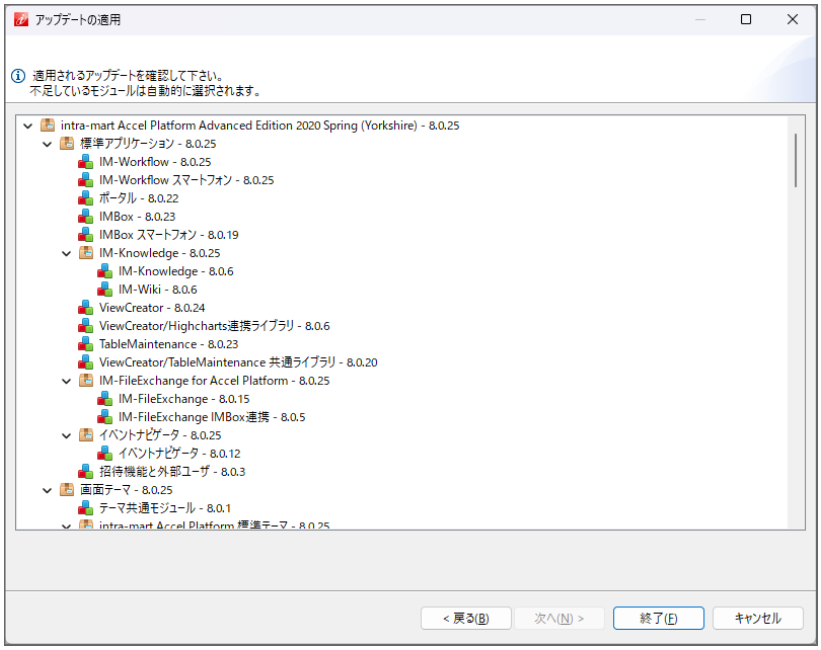
　ベースモジュールのアップデートで「8.0.25」を選択すると、アプリケーション（IM-Mail）は、ベースモジュール「8.0.25」には対応していないため、下記の警告画面が表示されます。



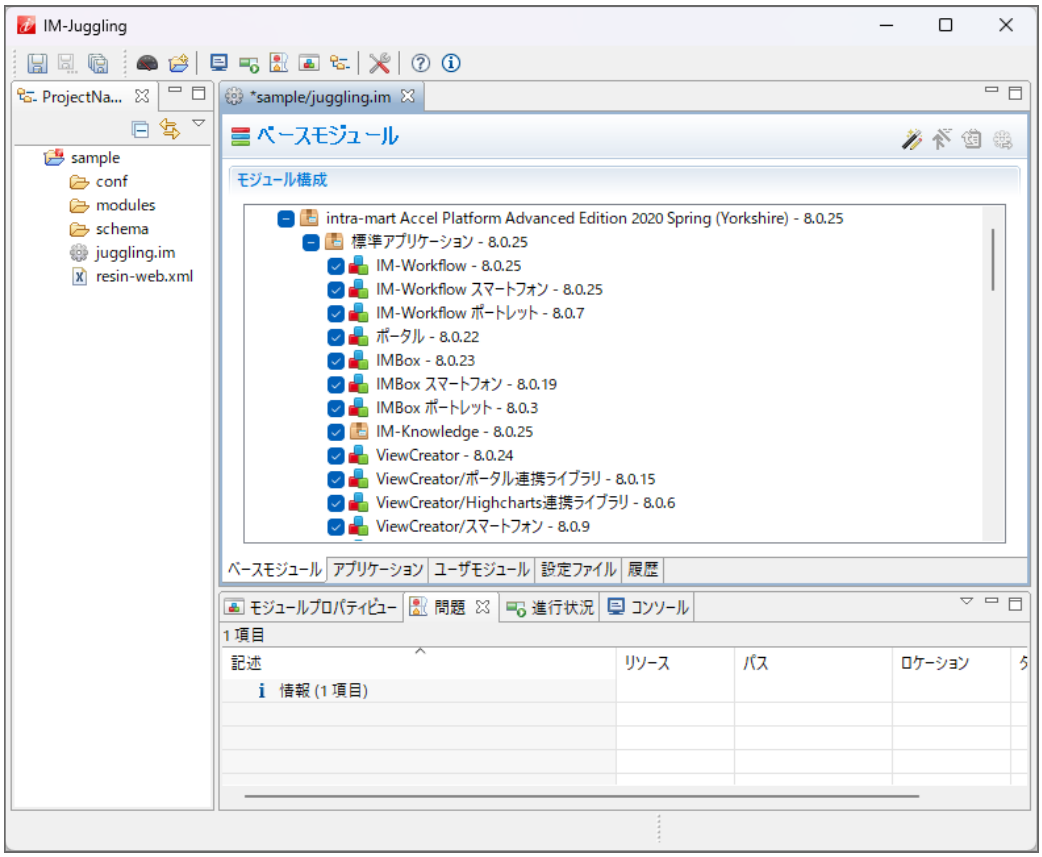
4. アプリケーションのアップデート対象を選択します。
   ベースモジュールのアップデート対象で選択したバージョンに基づくアプリケーションとそのバージョンが選択できます。
   選択後、「次へ（N）」をクリックします。



5. アップデート内容を確認し「終了（E）」をクリックします。



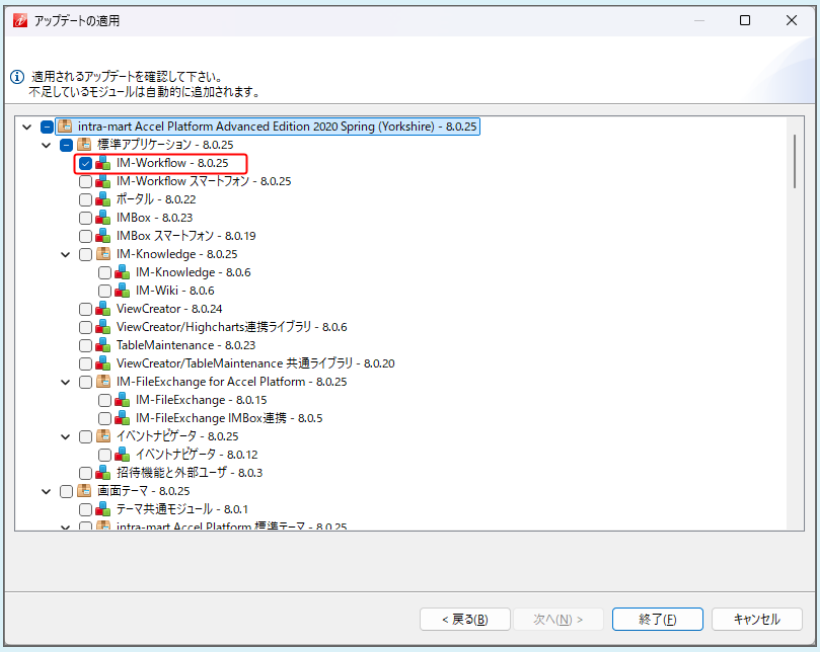
アップデート情報を取得し、既存プロジェクトのモジュール情報が変更されます。


IM-Juggling
ProjectNa...
*sample/juggling.im
sample
conf
modules
schema
juggling.im
resin-web.xml
ベースモジュール
モジュール構成
intra-mart Accel Platform Advanced Edition 2020 Spring (Yorkshire) - 8.0.25
標準アプリケーション - 8.0.25
IM-Workflow - 8.0.25
IM-Workflow スマートフォン - 8.0.25
IM-Workflow ポートレット - 8.0.7
ポータル - 8.0.22
IMBox - 8.0.23
IMBox スマートフォン - 8.0.19
IMBox ポートレット - 8.0.3
IM-Knowledge - 8.0.25
ViewCreator - 8.0.24
ViewCreator/ポータル連携ライブラリ - 8.0.15
ViewCreator/Highcharts連携ライブラリ - 8.0.6
ViewCreator/スマートフォン - 8.0.9
ベースモジュール
アプリケーション
ユーザモジュール
設定ファイル
履歴
モジュールプロパティビュー
問題
進行状況
コンソール
1 項目
記述
リソース
パス
ロケーション
情報 (1 項目)

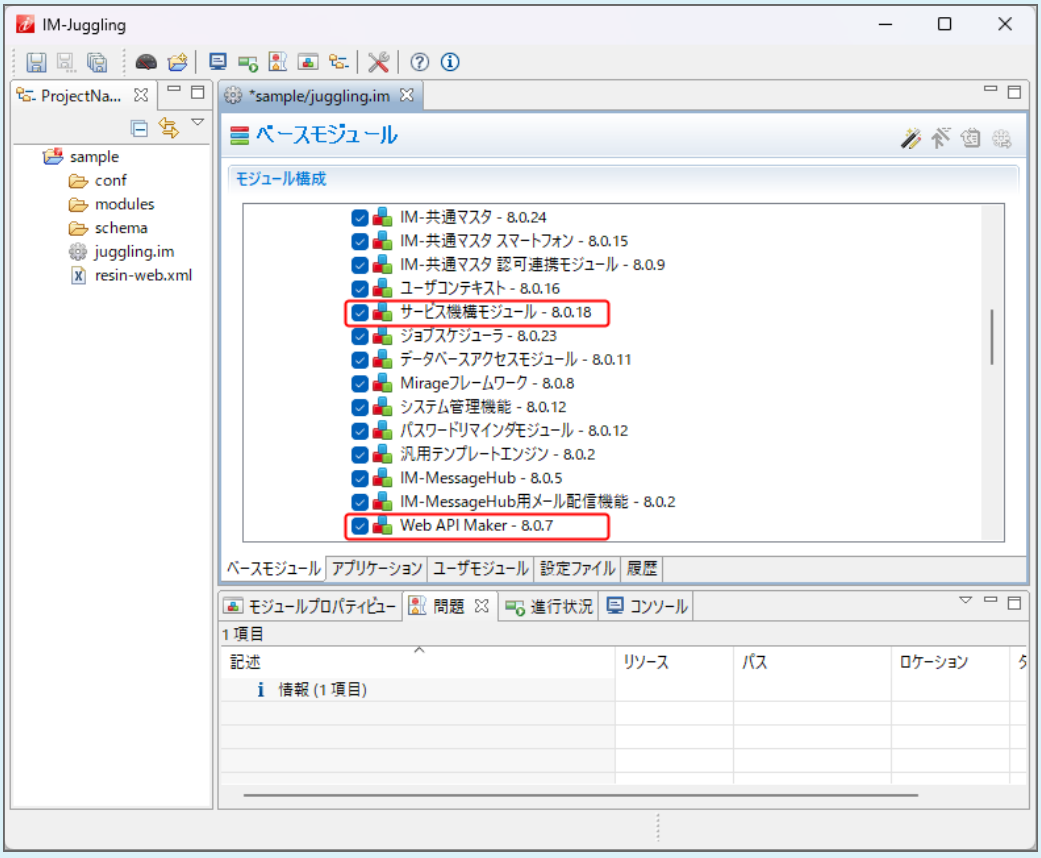
コラム

「個別に適用するアップデートを選択する（非推奨）」を選択した場合、既存プロジェクトのモジュール情報は、選択したモジュールと、そのモジュールが依存するモジュールのバージョンが更新されます。

例：「IM-Workflow」モジュールのみを選択します。



アップデート適用後のモジュール構成では、「IM-Workflow」と「IM-Workflow」が依存する「サービス機構モジュール」、「Web API Maker」などのモジュールも自動的に更新されます。



アップデート対象がプロジェクトに反映されます。

IM-Juggling で必要なメンテナンス作業（アップデートによる設定項目のメンテナンス等）

アップデート版を適用前に環境を構築している場合、個別作業が必要です。
個別作業の詳細は、

- *アップデート時に IM-Juggling で必要なメンテナンス作業*

を参照してください。

> **コラム**
> ※ intra-mart Accel Platform 以外のアプリケーションについては、各セットアップガイドを参照してください。

WARファイルの出力

1. 前述の IM-Juggling 上の設定が完了したら *WAR ファイルの出力* を行います。

WARファイルのアンデプロイ

1. 現在稼働している環境のアンデプロイを行います。
   Resin の場合は、 *WAR ファイルのアンデプロイ* を参照してください。
   Resin 以外の Web Application Server については、各製品ベンダから提供されているドキュメントを参照してください。

> **注意**
> セッション情報の永続化を行っている場合に、アップデート後にアクセスすると例外が出力されることがあります。
> 詳しくは以下を参照ください。
>
> - FAQ: アップデート後にアクセスすると例外が出力されてセッションタイムアウトまたはエラー画面に遷移します。
>
> 再デプロイ前に永続化したセッション情報を削除しておくことで回避できます。
>
> - Resin の session_store を true に設定してセッション情報の永続化を行っている場合
>   - <%RESIN_HOME%/resin-data/app-0/distcache> を削除することで永続化したセッション情報を削除できます。
> - セッション管理モジュールを利用していて、セッション情報の永続化を設定している場合
>   - 永続化先のデータベースにて im_http_session テーブルを truncate することで永続化したセッション情報を削除できます。
>
> また、アップデート後の初回アクセス時にブラウザに保存されているクッキーを削除することで回避することも可能です。

Web Application Server の更新

- アップデートではシステム要件の改変により、 Web Application Server の更新が必要となる場合があります。
  Resin の場合は次の通りです。

> **コラム**
> 最新版の Resin は製品メディアイメージ/Products/tools/iAP/Resinディレクトリまたは、次のサイトよりダウンロードして利用する事ができます。
> プロダクトファイルダウンロード
> ※ダウンロードには製品のライセンスキーが必要です。

1. 既存の Resin 環境を停止します。

   > **コラム**
   > 必要に応じて、Windowsサービスの解除、Linuxデーモンの解除が必要です。
   > 詳細は、 *分散構成の場合* や *複数のIPアドレスが設定されている場合* を参照してください。

   > **コラム**
   > 既存の Resin 環境は以後利用する事はないため、必要に応じて退避等を行ってください。

2. アップデート版で提供される最新版の Resin をインストールします。
   詳細は、 *Web Application Server（Resin）* を参照してください。

   > **コラム**
   > 必要に応じて、Windowsサービスの登録、Linuxデーモンの登録が必要です。
   > 詳細は、 *Windowsサービスへの登録* や *RHEL6の場合* を参照してください。

3. 既存の Resin 環境で設定していた内容を新しい Resin 環境へ反映します。

例：<%RESIN_HOME%/conf/resin.properties>ファイル内のJVMの設定等

> **注意**
>
> Resin 以外の Web Application Server においてシステム要件の改変が発生した場合、各製品の更新方法については、各製品ベンダから提供されているドキュメントを参照してください。
> データベース製品についても同様です。

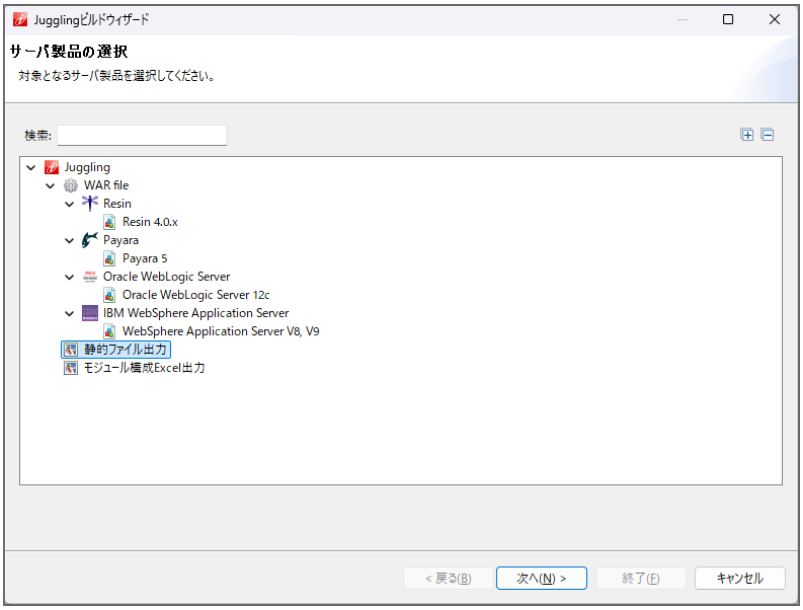
WARファイルの再デプロイ

1. アップデート版を反映したWARファイルにてデプロイを行います。
   Resin は、 *デプロイ* を参照してください。
   WebSphere Application Server 9.0.5 は、セットアップガイド for WebSphere 」を参照してください。
   Oracle WebLogic Server 12c R2(12.2.1) は、セットアップガイド for WebLogic 」を参照してください。

> **注意**
>
> 再デプロイを行ったが、追加したモジュール（資材）がデプロイ先に反映されないといった事象が発生した場合は、「 *Resin で WAR ファイルの再デプロイが正常にできない場合* 」を参照してください。

静的ファイルの出力と再配置

1. Web Server を利用している場合、IM-Juggling より静的ファイルを出力します。

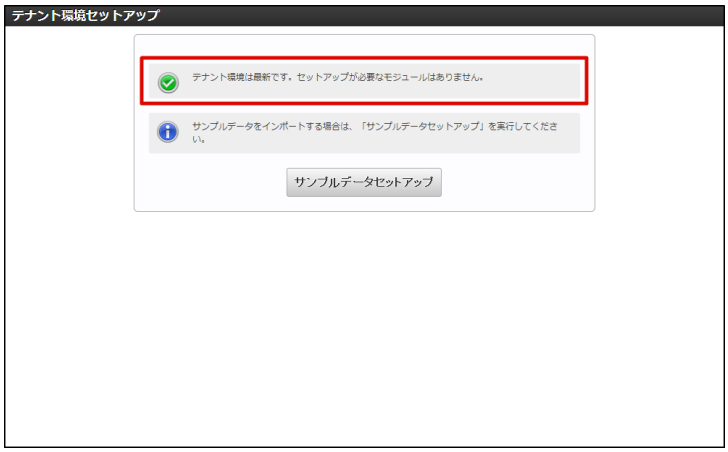


2. 出力した静的ファイルは、 Web Server でエイリアスとして設定しているディレクトリ配下に配置します。
   ※既存のファイルとは差し替えてください。

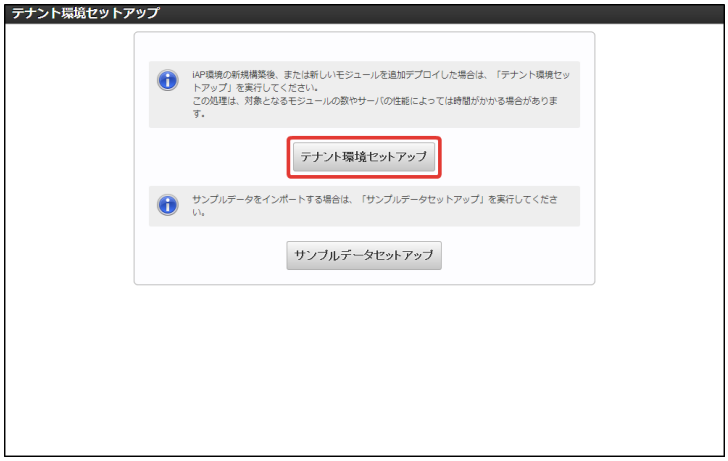
テナント環境セットアップ

- Web Application Server を起動後、テナント環境セットアップを実施します。
  システム管理者でログインします。
  メニューから「テナント環境セットアップ」をクリックします。

下図のように、「テナント環境は最新です。セットアップが必要なモジュールはありません。」という旨のメッセージが表示されていれば、テナント環境セットアップは正常に完了しています。



下図のように、「テナント環境セットアップ」ボタンが表示されている場合、テナント環境セットアップは未完了です。「テナント環境セットアップ」ボタンをクリックしセットアップを再度実施します。



> **注意**
>
> Web Server を経由してテナント環境セットアップを行う場合、Web Server のタイムアウト設置値を変更します。
> または、Web Application Server 経由でセットアップを実施する事を推奨します。

> **コラム**
>
> サンプルをセットアップするボタンは処理結果に関わらず、常に表示される仕様です。
> このため、サンプルデータ投入中にエラーが発生した場合、再度セットアップを実施すると
> データベース で一意制約違反が発生します。
> この場合は、「*アンインストール*」を行い、改めてセットアップからやり直す事を推奨します。

テナント環境セットアップ後の各種メンテナンス

- アップデート版を適用前に環境を構築している場合、個別作業が必要です。
  詳細は、 *テナント環境セットアップ後の各種メンテナンス（アップデートによるメンテナンス）* を参照してください。

- IMBoxをご利用のお客様は、既知の不具合が改修された Apache Cassandra へのバージョンアップを実施することを強く推奨しています。
  - 「 Cassandra管理者ガイド 」-「 Cassandraのバージョンアップ 」

パッチ

- パッチとは、アップデートとアップデートの間に発生した緊急度が高い修正等をモジュール単位で提供するリリース形態です。
  このパッチで提供される内容（不具合を修正した内容）は、今後リリース予定のアップデート版に含まれます。

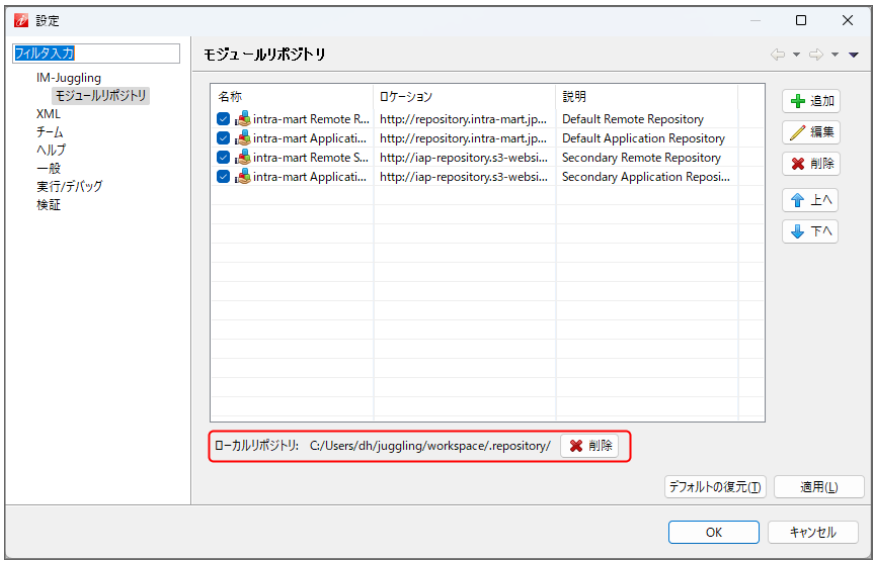
モジュールのパッチ適用

項目

- 古いリポジトリ情報の削除
- パッチを適用するモジュールの選択とWARファイルの作成
- WARファイルのアンデプロイ
- WARファイルの再デプロイ
- 静的ファイルの出力と再配置
- テナント環境セットアップ

古いリポジトリ情報の削除

- 前回使用した古いリポジトリ情報を削除します。
  1. IM-Juggling を起動し、ウィンドウ内-ツールバー右端にある「設定」-「IM-Juggling」-「モジュールリポジトリ」を開きます。
  2. 下部にある「ローカルリポジトリ」の削除をクリックしてください。

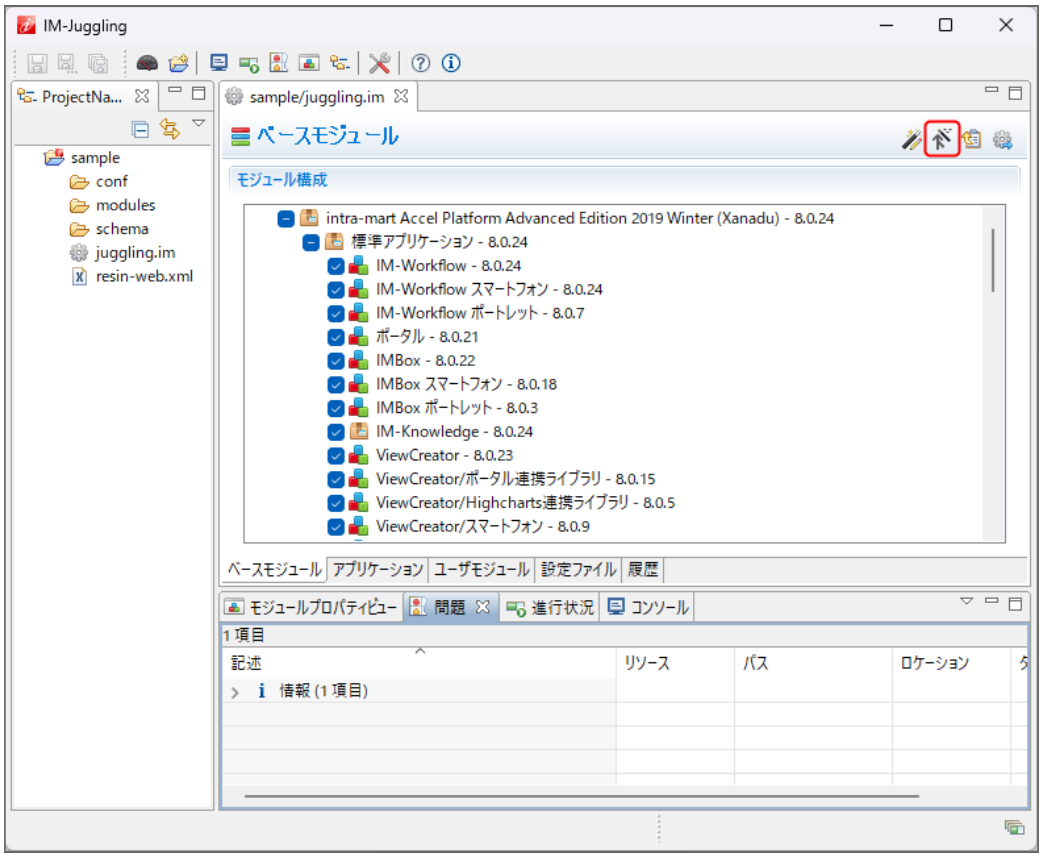


> **注意**
>
> ファイルの削除処理が行われている間は IM-Juggling の操作が行えない状態です。

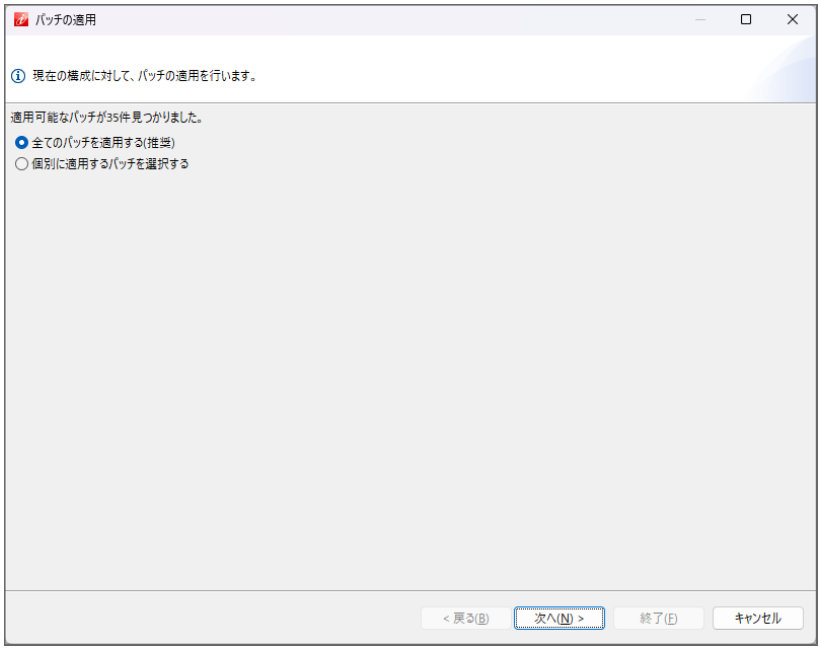
  3. 削除が完了したらOKをクリックします。

パッチを適用するモジュールの選択とWARファイルの作成

- 前述の「古いリポジトリ情報の削除」を実施している事が前提です。
  1. 画面右上の「パッチ」アイコンをクリックします。

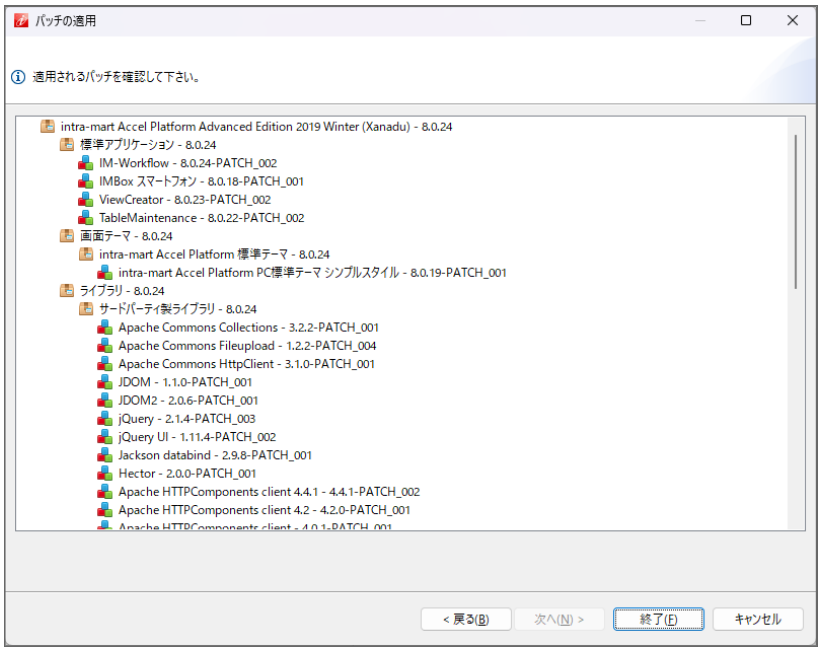


2. パッチ適用方法を選択します。
   「全てのパッチを適用する（推奨）」
   「個別に適用するパッチを選択する」
   のどちらかを選択してください。選択後、「次へ（N）」をクリックします。

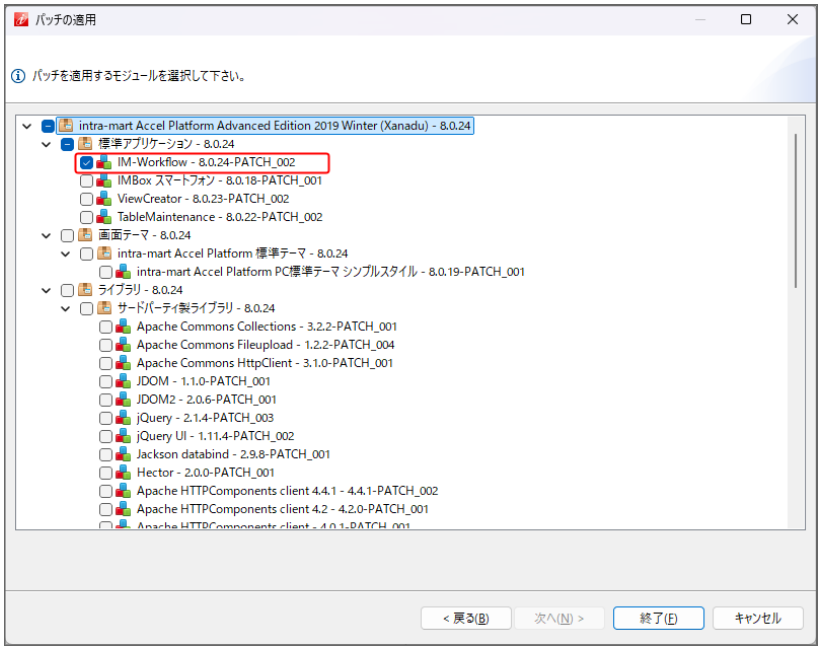


> **コラム**
>
> パッチとして提供される内容（不具合を修正した内容）は、今後リリース予定のアップデート版に含まれます。

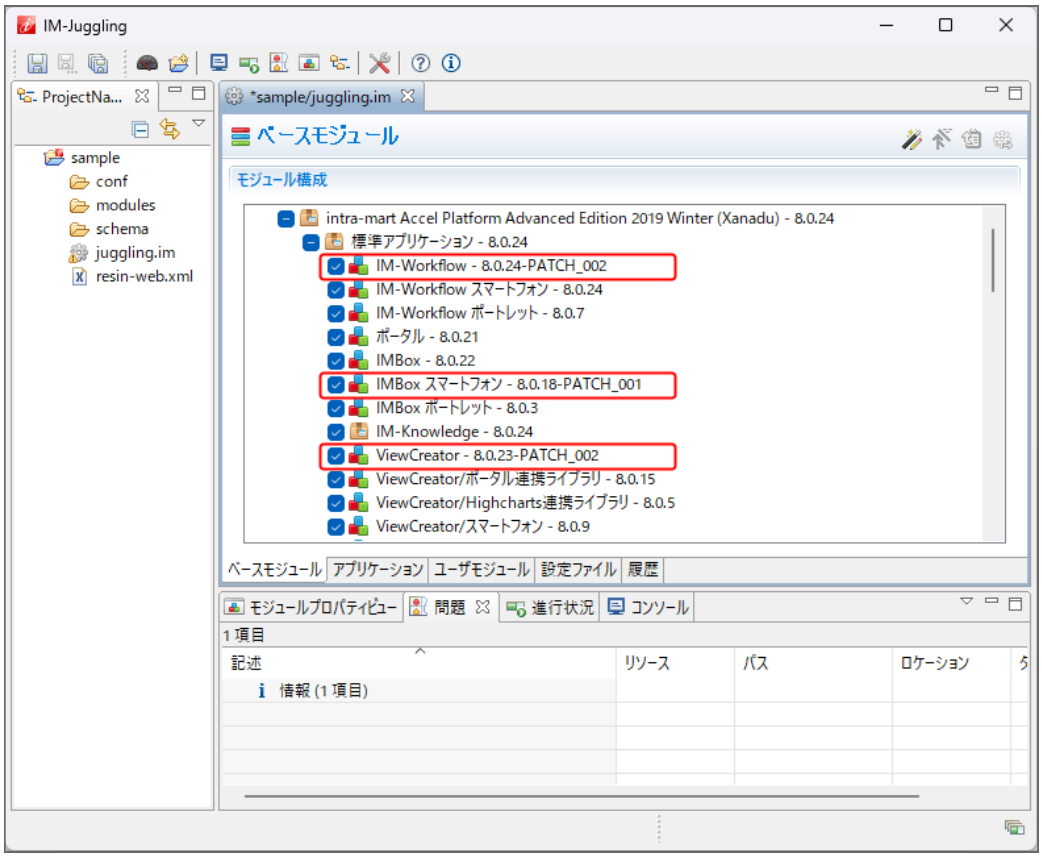
3. 前画面で「全てのパッチを適用する（推奨）」を選択していた場合、
   適用するパッチ内容が表示されます。
   適用するパッチ内容を確認し「終了（E）」をクリックします。



4. 前画面で「個別に適用するパッチを選択する」を選択していた場合、
パッチを適用するモジュールを選択します（チェックボックスにチェックをいれます）。
選択後、「終了（E）」をクリックします。



パッチ情報を取得し、既存プロジェクトのモジュール情報が変更されます。



5. パッチがプロジェクトに反映されたら、 *WAR ファイルの出力* を行います。

WARファイルのアンデプロイ

1. 現在稼働している環境のアンデプロイを行います。
   Resin の場合は、 *WAR ファイルのアンデプロイ* を参照してください。
   Resin 以外の Web Application Server については、各製品ベンダから提供されているドキュメントを参照してください。

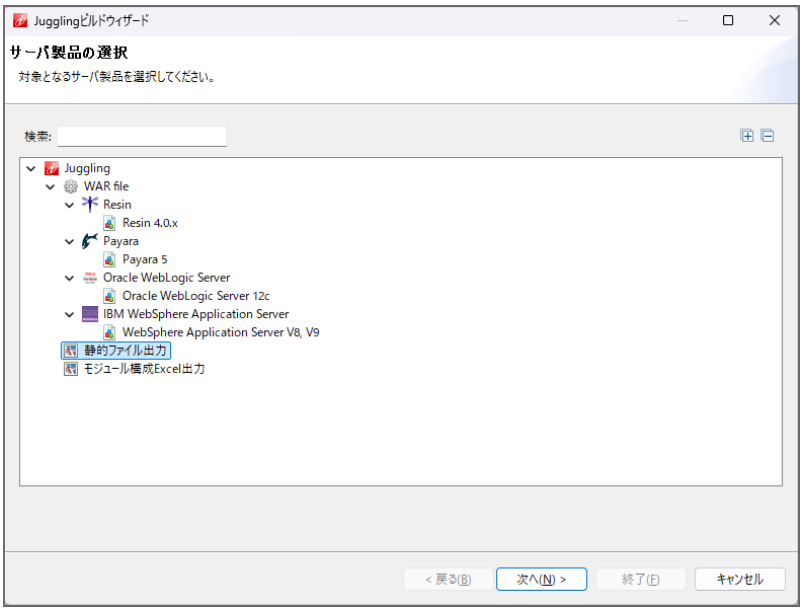
WARファイルの再デプロイ

1. パッチを反映したWARファイルにてデプロイを行います。
   Resin は、 *デプロイ* を参照してください。
   WebSphere Application Server 9.0.5 は、セットアップガイド for WebSphere 」を参照してください。
   Oracle WebLogic Server 12c R2(12.2.1) は、セットアップガイド for WebLogic 」を参照してください。

> **注意**
>
> 再デプロイを行ったが、追加したモジュール（資材）がデプロイ先に反映されないといった事象が発生した場合は、「 *Resin で WAR ファイルの再デプロイが正常にできない場合* 」を参照してください。

静的ファイルの出力と再配置

1. Web Server を利用している場合、IM-Juggling より静的ファイルを出力します。



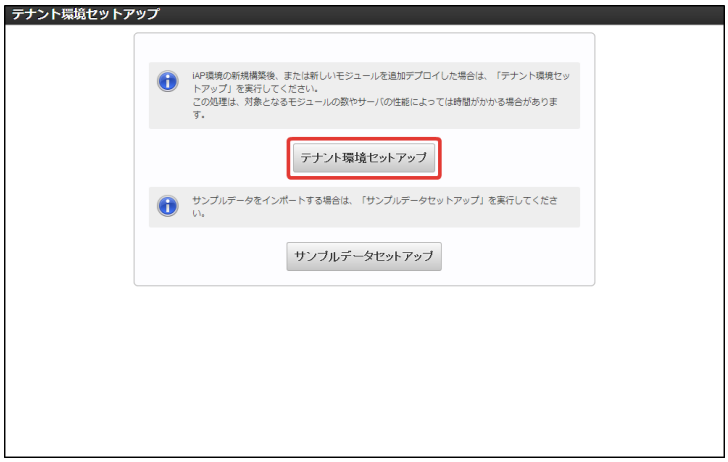
2. 出力した静的ファイルは、 Web Server でエイリアスとして設定しているディレクトリ配下に配置します。
※既存のファイルとは差し替えてください。

テナント環境セットアップ

- Web Application Server を起動後、テナント環境セットアップを実施します。
システム管理者の「メニュー」画面を表示します。
メニューから「テナント環境セットアップ」をクリックします。

下図のように、「テナント環境は最新です。セットアップが必要なモジュールはありません。」という旨のメッセージが表示されていれば、テナント環境セットアップは正常に完了しています。

下図のように、「テナント環境セットアップ」ボタンが表示されている場合、テナント環境セットアップは未完了です。「テナント環境セットアップ」ボタンをクリックしセットアップを再度実施します。



> **注意**
>
> Web Server を経由してテナント環境セットアップを行う場合、Web Server のタイムアウト設置値を変更します。
> または、Web Application Server 経由でセットアップを実施する事を推奨します。

> **コラム**
>
> サンプルをセットアップするボタンは処理結果に関わらず、常に表示される仕様です。
> このため、サンプルデータ投入中にエラーが発生した場合、再度セットアップを実施すると
> データベース で一意制約違反が発生します。
> この場合は、「*アンインストール*」を行い、改めてセットアップからやり直す事を推奨します。

モジュール構成の変更

- IM-Juggling を利用し、モジュールやアプリケーションの追加を行うことができます。

モジュール構成の変更

項目

- ミドルウェアのセットアップ
- 古いリポジトリ情報の削除
- モジュール構成の変更
- アプリケーションの追加
- 設定ファイルの編集
- WARファイルの出力
- WARファイルのアンデプロイ
- WARファイルの再デプロイ
- 静的ファイルの出力と再配置
- テナント環境セットアップ
  - テナント環境セットアップの実施

ミドルウェアのセットアップ

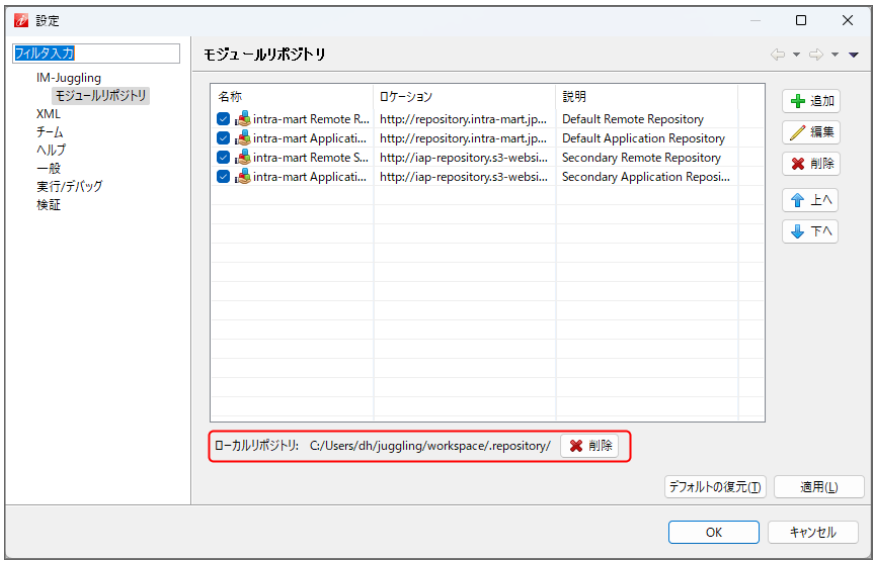
- モジュールやアプリケーションを追加する場合、新たにミドルウェアのセットアップが必要になる場合があります。
  例えば、IMBox を追加する場合、 Apache Cassandra のインストールおよび設定が必要です。
  詳細は、 *ミドルウェアのセットアップ* を参照してください。

> **コラム**
>
> ※ intra-mart Accel Platform 以外のアプリケーションについては、各セットアップガイドを参照してください。

古いリポジトリ情報の削除

- 前回使用した古いリポジトリ情報を削除します。
  1. IM-Juggling を起動し、ウィンドウ内-ツールバー右端にある「設定」-「IM-Juggling」-「モジュールリポジトリ」を開きます。

2. 下部にある「ローカルリポジトリ」の削除をクリックしてください。



> **注意**
>
> ファイルの削除処理が行われている間は IM-Juggling の操作が行えない状態です。

3. 削除が完了したらOKをクリックします。

モジュール構成の変更

1. モジュール構成の変更を行います。
   IM-Juggling プロジェクト内に表示されているモジュールの右クリックメニューで表示されるサブメニューより、一括で関連するモジュールの選択、選択解除を行う事ができます。

> **注意**
>
> intra-mart Accel Platform の Edition の変更は IM-Juggling のプロジェクトの再作成が必要です。

アプリケーションの追加

1. アプリケーションの追加を行います。
   詳細は、 *アプリケーションの追加* を参照してください。

設定ファイルの編集

- モジュールやアプリケーションを追加する場合、新たに設定ファイルの編集が必要になる場合があります。
  すでに出力して編集している設定ファイルは上書きされないため、再度編集する必要はありません。
  詳細は、*intra-mart Accel Platform の設定ファイル* を参照してください。
  また、新たに設定ファイルが必要となった場合は「設定ファイルリファレンス」を参照のうえ編集してください。

> **コラム**
>
> ※ intra-mart Accel Platform 以外のアプリケーションについては、各セットアップガイドを参照してください。

> **コラム**
>
> ※ 不要となったモジュールが出力したファイルが存在する場合は削除してください。

WARファイルの出力

1. 前述の IM-Juggling 上の設定が完了したら *WAR ファイルの出力* を行います。

WARファイルのアンデプロイ

1. 現在稼働している環境のアンデプロイを行います。
   Resin の場合は、 *WAR ファイルのアンデプロイ* を参照してください。
   Resin 以外の Web Application Server については、各製品ベンダから提供されているドキュメントを参照してください。

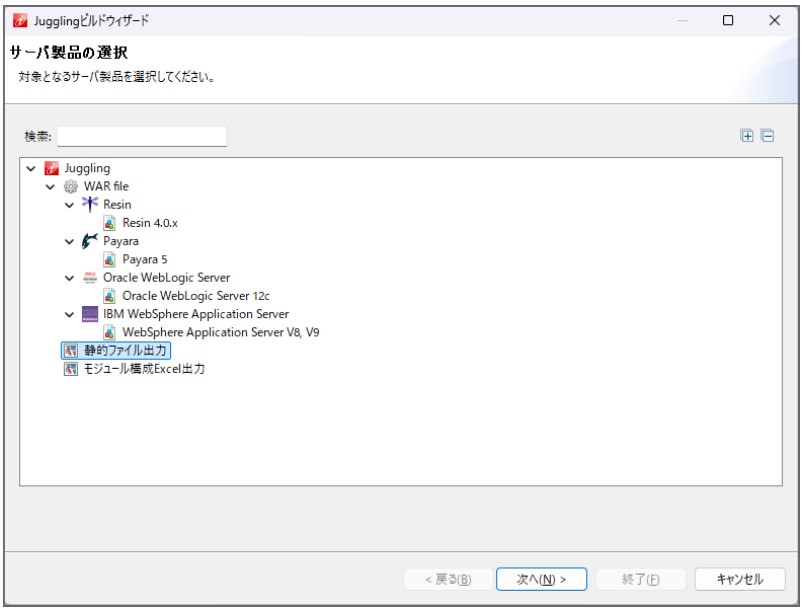
WARファイルの再デプロイ

1. アップデート版を反映したWARファイルにてデプロイを行います。
   Resin は、 *デプロイ* を参照してください。
   WebSphere Application Server 9.0.5 は、「intra-mart Accel Platform セットアップガイド (WebSphere編)」-「テナント環境セットアップ」を参照してください。
   Oracle WebLogic Server 12c R2(12.2.1) は、「intra-mart Accel Platform セットアップガイド (WebLogic編)」-「テナント環境セットアップ」を参照してください。

> **注意**
>
> 再デプロイを行ったが、追加したモジュール（資材）がデプロイ先に反映されないといった事象が発生した場合は、「*Resin で WAR ファイルの再デプロイが正常にできない場合*」を参照してください。

静的ファイルの出力と再配置

1. Web Server を利用している場合、IM-Juggling より静的ファイルを出力します。



2. 出力した静的ファイルは、 Web Server でエイリアスとして設定しているディレクトリ配下に配置します。
   ※既存のファイルとは差し替えてください。

テナント環境セットアップ

> **注意**
>
> 2014 Spring(Granada) 以降のバージョンでLDAP認証モジュール、またはIMBoxモジュールを**追加**する場合は、テナント環境セットアップを実行する前に以下の操作を行う必要があります。
>
> - LDAP認証モジュールを追加する場合
>
>   テナント環境セットアップを行う前に、「テナント管理」画面にてLDAP連携・設定情報を更新する必要があります。
>   LDAP連携・設定情報の更新に関しては、「システム管理者操作ガイド」-「LDAP連携・設定」を参照してください。
>
> - IMBoxモジュールを追加する場合
>
>   テナント環境セットアップを行う前に、「テナント管理」画面にてCassandra接続情報を更新する必要があります。
>   その際にCassandra接続情報の初期値は、Cassandraサーバ接続設定（cassandra-config.xml）の設定値です。
>   Cassandra接続情報の更新に関しては、「システム管理者操作ガイド」-「Apache Cassandra接続情報」を参照してください。

テナント環境セットアップの実施

Web Application Server を起動後、テナント環境セットアップを実施します。

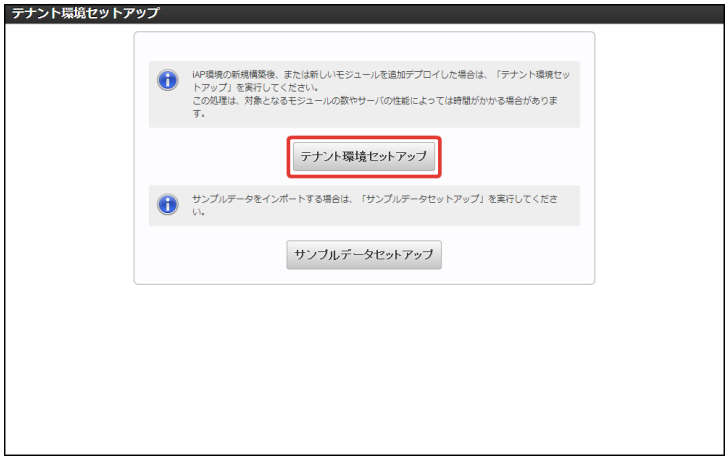
1. システム管理者の「メニュー」画面を表示します。
2. メニューから「テナント環境セットアップ」をクリックします。

- テナント環境が最新の場合

  下図のように、「テナント環境は最新です。セットアップが必要なモジュールはありません。」という旨のメッセージが表示されていれば、テナント環境セットアップの実施は必要ありません。そのまま利用できます。

- テナント環境が最新ではない場合

  下図のように、「テナント環境セットアップ」ボタンが表示されている場合、テナント環境セットアップが必要です。「テナント環境セットアップ」ボタンをクリックし、セットアップを再度実施します。



> **注意**
>
> Web Server を経由してテナント環境セットアップを行う場合、Web Server のタイムアウト設置値を変更します。
> または、Web Application Server 経由でセットアップを実施する事を推奨します。

> **コラム**
>
> サンプルをセットアップするボタンは処理結果に関わらず、常に表示される仕様です。
> このため、サンプルデータ投入中にエラーが発生した場合、再度セットアップを実施すると
> データベース で一意制約違反が発生します。
> この場合は、「*アンインストール*」を行い、改めてセットアップからやり直す事を推奨します。

> **コラム**
>
> アップデート、パッチの適用およびモジュール構成の変更を行う前にバックアップを行う事を推奨します。
> 詳細については、「*バックアップ・リストア（復元）*」を参照してください。

コラム

インターネットに接続できない環境で IM-Juggling を利用する場合については以下を参照してください。

- *インターネットに接続できない環境で IM-Juggling を利用する場合*

付録

クラスタリング

intra-mart Accel Platform では分散環境を構築する方法は以下の２通りあります。

intra-mart Accel Platform の分散環境（Resin をクラスタリングせずに構築）

項目
- 概要
- 手順
  - <%Jugglingプロジェクト%/conf/storage-config.xml> の編集
  - <%Jugglingプロジェクト%/resin-web.xml> の編集
  - <%Jugglingプロジェクト%/conf/data-source-mapping-config.xml> の編集
  - <%Jugglingプロジェクト%/conf/network-agent-config.xml> の編集
  - セッション管理モジュールのインストール
  - conf/hazelcast-config.xml の設定
  - <%RESIN_HOME%/conf/resin.properties> ファイルの編集
  - Web Application Serverの起動
  - アプリケーションのデプロイ

概要

intra-mart Accel Platform の分散環境を、Resin のクラスタリング機能を利用せずに構築する手順について説明します。

手順

<%Jugglingプロジェクト%/conf/storage-config.xml> の編集

<root-path-name> にストレージのルートパスを設定してください。
このパスは、すべての intra-mart Accel Platform からアクセス可能なディレクトリを指定してください。
詳細は「*Storage*」を参照してください。

<%Jugglingプロジェクト%/resin-web.xml> の編集

使用するデータベースを設定してください。
すべての intra-mart Accel Platform からアクセス可能なデータベースを指定してください。
詳細は「*DataSource*」を参照してください。

<%Jugglingプロジェクト%/conf/data-source-mapping-config.xml> の編集

システムデータベースとテナントデータベースを設定してください。
すべての intra-mart Accel Platform からアクセス可能なデータベースを指定してください。
詳細は「*DataSource マッピングの設定*」を参照してください。

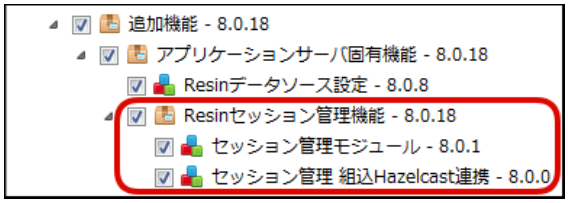
<%Jugglingプロジェクト%/conf/network-agent-config.xml> の編集

「サービス仕様書 分散環境の構築」を参照し、<%Jugglingプロジェクト%/conf/network-agent-config.xml> を編集してください。

セッション管理モジュールのインストール

Resin のクラスタリング機能を利用しない場合、Resin によるセッションフェイルオーバーができません。 そのため、「セッション管理モジュール」を利用しセッションフェイルオーバーを実現します。

IM-Juggling より「セッション管理モジュール（モジュールID: `jp.co.intra_mart.im_session_store`）」「セッション管理 組込Hazelcast連携（モジュールID: `jp.co.intra_mart.im_session_store_hazelcast_em`）」を選択します。



> **コラム**
>
> 「セッション管理モジュール」、「セッション管理 組込Hazelcast連携」は intra-mart Accel Platform 2017 Spring(Portland) を含む以降のバージョンで利用可能です。

> **コラム**
>
> intra-mart Accel Platform 2014 Spring(Granada) から intra-mart Accel Platform 2016 Winter(Olga) までのバージョンをご利用の場合、「セッション管理モジュールのご紹介」内に添付してある「session_store_module.zip」内のユーザモジュールを利用することで、「セッション管理モジュール」「セッション管理 組込Hazelcast連携」機能を代替できます。

conf/hazelcast-config.xml の設定

「intra-mart Accel Platform セットアップガイド セッション管理モジュール」を参照し、<%Jugglingプロジェクト%/conf/hazelcast-config.xml> を編集します。

<%RESIN_HOME%/conf/resin.properties> ファイルの編集

以下の1, 2を修正します。

1. app_servers に自身のIPアドレスのみを設定
2. elastic_cloud_enable をコメントアウト

例えば「192.168.0.1」「192.168.0.2」「192.168.0.3」の三台構成の場合、以下のように設定します。

- 「192.168.0.1」上の <%RESIN_HOME%/conf/resin.properties> ファイル

```
app_servers     : 192.168.0.1 # 「192.168.0.1」のみを設定します
# elastic_cloud_enable : true  # コメントアウトします
```

- 「192.168.0.2」上の <%RESIN_HOME%/conf/resin.properties> ファイル

```
app_servers     : 192.168.0.2 # 「192.168.0.2」のみを設定します
# elastic_cloud_enable : true  # コメントアウトします
```

- 「192.168.0.3」上の <%RESIN_HOME%/conf/resin.properties> ファイル

```
app_servers     : 192.168.0.3 # 「192.168.0.3」のみを設定します
# elastic_cloud_enable : true  # コメントアウトします
```

Web Application Serverの起動

以下の方法で起動を行ってください。

Web Application Server の起動・停止

アプリケーションのデプロイ

以下の方法で、すべてのサーバにてデプロイを行ってください。

webapps ディレクトリに WAR ファイルを直接配置してデプロイ

> **注意**
>
> すべてのサーバに対して同じ WAR ファイルをデプロイしてください。

Resin をクラスタリングしての分散環境

項目

- 概要
- トライアドサーバ
- ダイナミックサーバ

注意

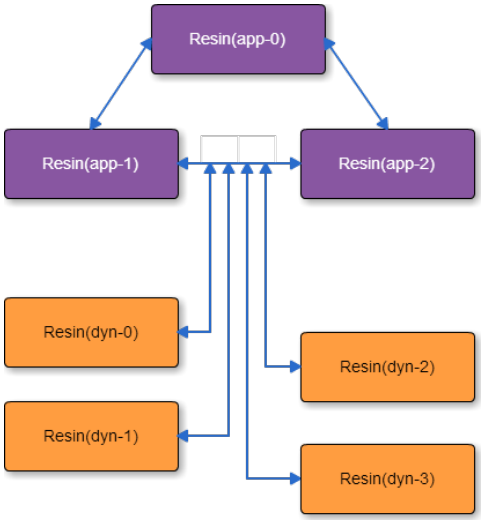
Resin のクラスタリングによる intra-mart Accel Platform の分散環境の構築は以下の理由により非推奨です。

- 「*intra-mart Accel Platform の分散環境（Resin をクラスタリングせずに構築）*」と比べ、内部データベースが破損する可能性が高い

代わりに、「*intra-mart Accel Platform の分散環境（Resin をクラスタリングせずに構築）*」を基に分散環境を構築してください。

概要

Resin のクラスタリングは静的な3台のサーバ（トライアドサーバ）と、負荷に応じて自由に手動で追加・削除が可能なサーバ（ダイナミックサーバ）により構成されます。
クラスタ構成にすることで負荷分散、デプロイとアンデプロイの伝播、セッションフェールオーバーが可能です。



トライアドサーバ

トライアドサーバは常に起動している静的なサーバです。クラスタの中心部でハブの役割となり互いに活性監視を行います。
また、3 重の冗長性により1台がメンテナンスを行っている場合でも残り 2 台が処理を継続することにより信頼性を保つことができます。

トライアドサーバを設定するには **resin.properties** ファイルを編集してください。
トライアドサーバとなる3台のIPアドレスまたはホスト名を「**app_servers**」に記述します。
同様の設定を 3 台すべてに記述します。

```
# app-tier Triad servers: app-0 app-1 app-2
app_servers     : 192.168.100.100 192.168.100.101 192.168.100.102
```

コラム

app_servers に IP アドレスのみを記述した場合、ポート番号は <resin.xml> ファイルの <server-multi port=”6800”/> が使用されます。
任意のポート番号を使用する場合は「[IP アドレス]:[ポート番号]」と記述してください。

```
# app-tier Triad servers: app-0 app-1 app-2
app_servers     : 192.168.100.100:6801 192.168.100.101:6802 192.168.100.102:6803
```

IPアドレスを記述した順に自動的に ID が割り振られます。 ID はプレフィックスが「**app-**」となり 0 から始まる数字が割り当てられます。
この記述例では下記のように ID が割り当てられます。

```
192.168.100.100   app-0
192.168.100.101   app-1
192.168.100.102   app-2
```

各サーバ上からコマンドを実行してトライアドサーバを起動してください。コマンドに含まれる ID は各サーバに応じて自身の ID を指定してください。
下記は IP アドレスが 192.168.100.100 のサーバから実行するコマンドの例です。
192.168.100.101 のサーバでは ID を「app-1」に、192.168.100.102 のサーバでは「app-2」に変更してコマンドを実行してください。

- Windows の場合

```
> resin.exe -server app-0 console
```

- Linux の場合

```
> resinctl -server app-0 console
```

以上でトライアドサーバを使用する手順は終了です。

> **コラム**
>
> トライアドサーバは 3 重の冗長性を意味する名称です。
> 1 台または 2 台でも役割を行いますが、1 台に障害が発生したりメンテナンスの場合でも、トライアドサーバは常に起動していなければならないため、最大で3台が起動可能な仕組みです。
> トライアドサーバが起動していないシステムではダイナミックサーバがクラスタに参加できません。
> 想定するシステム負荷からダイナミックサーバも必要なく 2 台で十分な場合は、 2 台のみのクラスタリング構成も可能です。

> **コラム**
>
> Windows サービスへの登録については、「 *Windows サービスへの登録* 」を参照してください。

ダイナミックサーバ

ダイナミックサーバを使用することでシステムの利用状況に応じて、自由に手動でサーバの追加・削除を行うことができます。
新規にダイナミックサーバを追加すると自動的にクラスタリングの調整が行われ活性監視の対象として扱われます。
また、トライアドサーバから Web アプリケーションの配信とセッションのフェールオーバーが自動的に行われます。

ダイナミックサーバを設定するには **resin.properties** ファイルを編集してください。下記の3つの設定が必要です。

- **app_servers**
  トライアドサーバとなる3台の IP アドレスまたはホスト名を記述します。
- **elastic_cloud_enable**
  コメントアウトされている属性を有効化し値を「**true**」とします。
- **home_cluster**
  ダイナミックサーバを追加するクラスタの ID を指定します。**resin.xml** ファイルがデフォルト設定の場合は「**app**」としてください。

下記はダイナミックサーバを使用する場合の記述例です。
この設定はトライアドサーバとダイナミックサーバの両方に同じ設定を行ってください。

```
# app-tier Triad servers: app-0 app-1 app-2
app_servers     : 192.168.100.100 192.168.100.101 192.168.100.102

. . .

# Allow elastic nodes to join the cluster (enable for cloud mode)
elastic_cloud_enable : true

. . .

# The cluster that elastic nodes should join - each will contact a Triad server
# Use a separate resin.properties file for each cluster
home_cluster : app
```

設定が記述された **resin.properties** ファイルは、サーバを容易に複製するためにコピーを保存しておくことをお勧めします。

ダイナミックサーバを実行します。サーバ上からコマンドを実行してください。
全てのトライアドサーバは予め起動してください。

- Windows の場合

```
> resin.exe --elastic-server --cluster app console
```

- Linux の場合

```
> resinctl --elastic-server --cluster app console
```

以上でダイナミックサーバを使用する手順は終了です。

> **コラム**
>
> 弊社では resin-data に関する障害を回避するために Resin をクラスタリングせずに分散環境を構築することを推奨しています。
> Resin 運用時の注意点については開発者ブログ「Resin利用時の安定運用について」を参照してください。

Resin をクラスタリングした際との比較

「*Resin をクラスタリングしての分散環境*」の手順を基にして分散環境を構築した場合と、「*intra-mart Accel Platform の分散環境（Resin をクラスタリングせずに構築）*」の手順を基にして分散環境を構築した場合の違いについて説明します。

resin-admin

- **Resin のクラスタリング機能を利用する場合**
  何処か 1 つの resin-admin にログインする事で、すべてのサーバ情報を参照できます。
- **Resin のクラスタリング機能を利用しない場合**
  それぞれのサーバで resin-admin にログインする必要があります。
  また、ログインしたサーバの情報のみが表示されます。

セッションフェイルオーバー

- **Resin のクラスタリング機能を利用する場合**
  Resin の機能にてセッションフェイルオーバーができます。
- **Resin のクラスタリング機能を利用しない場合**
  「セッション管理モジュール」機能にてセッションフェイルオーバーができます。

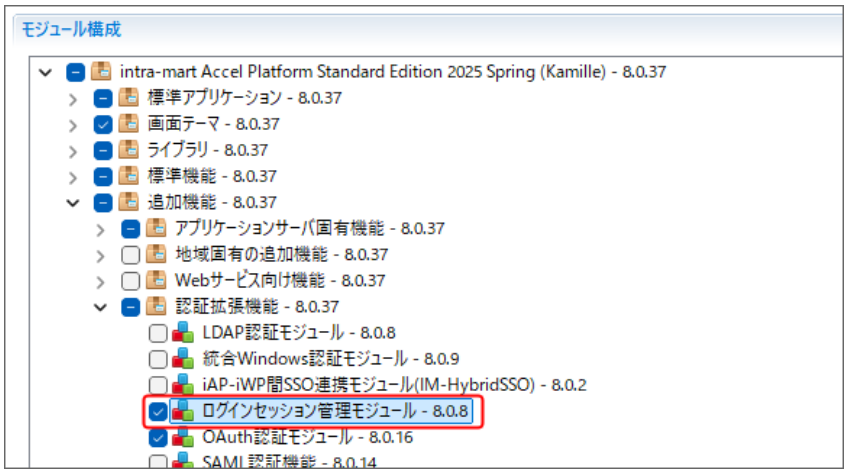
> **コラム**
>
> Resin のクラスタリング機能を利用しない場合ではセッション情報を内部データベースに保持しなくなるため、内部データベースが壊れる可能性を大幅に減らすことができます。
> 参考：Resin が「exit reason: HEALTH (exit code=9)」で再起動してしまいます。

ログ・PDFレポートファイル名

- **Resin のクラスタリング機能を利用する場合**
  ログファイル名は「<jvm-app-N.log>」（N は 0 からサーバ台数 - 1 まで）で出力されます。
  PDFレポートファイル名は「<app-N-${event}-${timestamp}.pdf>」（N は 0 からサーバ台数 - 1 まで）で出力されます。
- **Resin のクラスタリング機能を利用しない場合**
  ログファイル名は「jvm-app-0.log」で出力されます。
  PDFレポートファイル名は「<app-0-${event}-${timestamp}.pdf>」で出力されます。

二重ログイン防止機能

二重ログイン防止機能を利用する場合、IM-Juggling よりログインセッション管理モジュールをチェックしWARを作成してください。



統合Windows認証

項目

- 概要
- 前提条件
- セットアップ
  - 統合Windows認証機能の設定
    - テナント解決プラグインの設定
  - 統合Windows認証環境でWebサービスを利用する
  - 統合Windows認証環境で外部ソフトウェア連携機能を利用する
  - 「インターネット オプション」の設定
  - Keep-Alive の設定
- 統合Windows認証機能を無効化するには
- 統合Windows認証機能をリクエストに応じて無効化するには
  - Apache を使用して無効化する場合
  - Internet Information Services（IIS） を使用して無効化する場合
- 統合Windows認証機能の認証失敗時に通常のログイン機能を利用するには
- ケルベロス認証を無効化しNTLM認証を強制するには
- 統合Windows認証機能の認証失敗時にリダイレクトさせるには

概要

統合Windows認証機能は、ドメインコントローラ上の認証済みユーザと同じユーザコードをもつユーザで intra-mart Accel Platform アクセス時に、統合Windows認証済みユーザ情報を取得して自動ログインを行うことができる機能です。
これにより、シングルサインオンを実現できます。

> **注意**
>
> 統合Windows認証は、 Internet Information Services（IIS） が必須です。

> **注意**
>
> ログイン画面からのログインはサポートしません。詳しくは以下の制限を参照してください。
> SSO（SingleSignOn）環境での注意点があります。

前提条件

- アプリケーションサーバには Resin を使用してください。
- Resin を Windows Server 上で動作させる必要があります。
- Resin を実行する Windows Server がドメインに参加している必要があります。
- ブラウザは 「リリースノート」に記した Microsoft Edge を使用してください。
- 統合Windows認証モジュールが必要です。
  - IM-Juggling 上で「追加機能」-「認証拡張機能」-「統合Windows認証モジュール」を選択し intra-mart Accel Platform を構築する必要があります。
  - 統合Windows認証モジュールは、8.0.0-PATCH_001 以降のバージョンを使用してください。

> **注意**
>
> シングルサインオンを実現するためには、ドメインコントローラ上のユーザコードと intra-mart Accel Platform 上のユーザコードが一致している必要があります。

> **注意**
>
> 統合Windows認証機能には、ドメインコントローラ、統合Windows認証に対応したブラウザが必要です。

> **注意**
>
> スマートフォンでの統合Windows認証機能の利用はサポートしておりません。

> **注意**
>
> 統合Windows認証は intra-mart Accel Platform の機能として Resin 上で実行されます。そのため、IIS の Windows 認証を無効化してください。
> その他認証は必要に応じて設定してください。

セットアップ

統合Windows認証機能の設定

IM-Juggling で次の設定を行いwarファイルを作成してください。

- 機能の有効化の設定
  1. <（プロジェクト名）/juggling.im>の「ベースモジュール」タブで統合Windows認証モジュール(im_sso_windows)を選択します。「設定ファイル」タブの統合Windows認証モジュールを選択しim-sso-windows-config.xmlを出力します。
  2. <（プロジェクト名）/conf> 配下に出力されたim-sso-windows-config.xmlを開き<im-sso-windows-config>/<authentication>/<@enable> を true に設定します。
- テナント解決の設定（intra-mart Accel Platform 2014 Spring(Granada) 以降のみ）

  認証対象となるテナントを解決する方法として plugin を採用しています。
  plugin の設定仕様については、「PluginManagerのAPIドキュメント」を参照してください。

  - intra-mart Accel Platform 2015 Spring(Juno) まで（統合Windows認証モジュールバージョン 8.0.3 まで）をご利用の場合
    1. 「plugin/jp.co.intra_mart.foundation.admin.tenant.context.tenant.resolver.windows_8.0.1」フォルダを作成し、そのフォルダ内に以下のようなplugin.xmlを配置してください。

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin>
 <extension point="jp.co.intra_mart.foundation.admin.tenant.context.tenant.resolvers">
  <tenant-id-resolvers
   id="jp.co.intra_mart.foundation.admin.tenant.context.tenant.resolver.windows"
   name="Windows Tenant Id Resolver"
   version="8.0.1"
   rank="90"
   enable="true">

   <!-- ドメインをテナントIDとするリゾルバー
        参加しているドメインをテナントIDとして解決します。 -->
   <!--tenant-id-resolver
class="jp.co.intra_mart.foundation.admin.tenant.context.DomainBasedWindowsTenantIdResolver"/-->

   <!-- 設定された固定値をテナントIDとするリゾルバー
        tenant_idパラメータに設定した値をテナントIDとして解決します。 -->
   <tenant-id-resolver
class="jp.co.intra_mart.foundation.admin.tenant.context.FixedConfigBasedWindowsTenantIdResolver">
    <init-param>
     <param-name>tenant_id</param-name>
     <param-value>default</param-value>
    </init-param>
   </tenant-id-resolver>
  </tenant-id-resolvers>
 </extension>
</plugin>
```

  - intra-mart Accel Platform 2015 Summer(Karen) 以降（統合Windows認証モジュールバージョン 8.0.4 以降）をご利用の場合
    1. 「plugin/jp.co.intra_mart.foundation.admin.tenant.context.tenant.resolver.windows_8.0.4」フォルダを作成し、そのフォルダ内に以下のようなplugin.xmlを配置してください。

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin>
 <extension point="jp.co.intra_mart.foundation.admin.tenant.context.tenant.resolvers">
  <tenant-id-resolvers
   id="jp.co.intra_mart.foundation.admin.tenant.context.tenant.resolver.windows"
   name="Windows Tenant Id Resolver"
   version="8.0.4"
   rank="90"
   enable="true">

   <!-- ドメインをテナントIDとするリゾルバー
        参加しているドメインをテナントIDとして解決します。 -->
   <!--tenant-id-resolver
class="jp.co.intra_mart.foundation.admin.tenant.context.DomainBasedWindowsTenantIdResolver"/-->

   <!-- 設定された固定値をテナントIDとするリゾルバー
        tenant_idパラメータに設定した値をテナントIDとして解決します。 -->
   <!--tenant-id-resolver
class="jp.co.intra_mart.foundation.admin.tenant.context.FixedConfigBasedWindowsTenantIdResolver">
     <init-param>
      <param-name>tenant_id</param-name>
      <param-value>default</param-value>
     </init-param>
   </tenant-id-resolver-->

   <!-- 設定された固定値をテナントIDとするリゾルバー
        tenant_idパラメータに設定した値をテナントIDとして解決します。
        FixedConfigBasedWindowsTenantIdResolver と違い、どのようなリクエストに対しても必ず設定値を返却します。 -->
   <tenant-id-resolver
class="jp.co.intra_mart.foundation.admin.tenant.context.ConstantConfigBasedWindowsTenantIdResolver">
     <init-param>
      <param-name>tenant_id</param-name>
      <param-value>default</param-value>
     </init-param>
   </tenant-id-resolver>
  </tenant-id-resolvers>
 </extension>
</plugin>
```

1. <plugin>/<extension>/<tenant-id-resolvers>/<tenant-id-resolver>/<@class> に実装クラスを指定し、必要に応じて実装クラスにパラメータを設定します。
   - 使用可能な実装クラス、および設定するパラメータについては*テナント解決プラグインの設定*を参照してください。

以上で統合Windows認証機能の設定は終了です。

テナント解決プラグインの設定

テナント解決を行う仕組みとして、以下の三つの実装クラスを用意しています。

実装クラス	概要
jp.co.intra_mart.foundation.admin.tenant.context.DomainBasedWindowsTenantIdResolver	ユーザが参加しているドメインをテナントIDとして解決します。
jp.co.intra_mart.foundation.admin.tenant.context.FixedConfigBasedWindowsTenantIdResolver	設定された固定値をテナントIDとして解決します。
jp.co.intra_mart.foundation.admin.tenant.context.ConstantConfigBasedWindowsTenantIdResolver	どのようなリクエストに対しても、設定された固定値をテナントIDとして解決します。intra-mart Accel Platform 2015 Summer(Karen) 以降（統合Windows認証モジュールバージョン 8.0.4 以降）でのみ利用可能です。

- jp.co.intra_mart.foundation.admin.tenant.context.DomainBasedWindowsTenantIdResolver

  ユーザが参加しているドメインをテナントIDとして解決します。
  本実装クラスに必要なパラメータはありません。

- jp.co.intra_mart.foundation.admin.tenant.context.FixedConfigBasedWindowsTenantIdResolver

tenant_id パラメータを受け取り、その値をテナントIDとして解決します。
下記の設定例では、「default」がテナントIDとして解決されます。
*統合Windows認証が無効化されたリクエスト*に対しては null を返却します

```
<tenant-id-resolver class="jp.co.intra_mart.foundation.admin.tenant.context.FixedConfigBasedWindowsTenantIdResolver">
 <init-param>
  <param-name>tenant_id</param-name>
  <param-value>default</param-value>
 </init-param>
</tenant-id-resolver>
```

- jp.co.intra_mart.foundation.admin.tenant.context.ConstantConfigBasedWindowsTenantIdResolver

  tenant_id パラメータを受け取り、その値をテナントIDとして解決します。
  下記の設定例では、「default」がテナントIDとして解決されます。
  FixedConfigBasedWindowsTenantIdResolver と違い、どのようなリクエストに対しても設定値を返却します。
  intra-mart Accel Platform 2015 Summer(Karen) 以降（統合Windows認証モジュールバージョン 8.0.4 以降）でのみ利用可能です。

```
<tenant-id-resolver class="jp.co.intra_mart.foundation.admin.tenant.context.ConstantConfigBasedWindowsTenantIdResolver">
 <init-param>
  <param-name>tenant_id</param-name>
  <param-value>default</param-value>
 </init-param>
</tenant-id-resolver>
```

> **コラム**
>
> intra-mart Accel Platform 2015 Summer(Karen) 以前（統合Windows認証モジュールバージョン 8.0.4 以前）では FixedConfigBasedWindowsTenantIdResolver, 以降では ConstantConfigBasedWindowsTenantIdResolver をご利用ください。

統合Windows認証環境でWebサービスを利用する

統合Windows認証環境でWebサービスを利用するためには、以下の設定が必要です。

IM-Juggling で次の設定を行いwarファイルを作成してください。

- 機能の有効化の設定
  1. <（プロジェクト名）/juggling.im>の「ベースモジュール」タブでWebサービス 認証・認可(im_ws_auth)、Webサービス認証・認可クライアント(im_ws_auth_client)、Apache Axis2(axis2) を選択します。
     「設定ファイル」タブのwebサービス 認証・認可を選択しim-sso-windows-path-config/im-sso-windows-path-config_ws_auth.xmlを出力します。
  2. <（プロジェクト名）/conf/im-sso-windows-path-config> 配下に出力されたim-sso-windows-path-config_ws_auth.xmlを開き以下のようにコメントを外して、設定を有効化します。

     ```
     <no-authentication>
      <path regex="true">/services/.*</path>
      <path regex="true">/axis2-admin/.*</path>
     </no-authentication>
     ```

> **コラム**
>
> ベースモジュール選択時にWebサービスライブラリをチェックし依存関係を解消することでモジュールを探すことなく設定ファイルを出力可能です。

統合Windows認証環境で外部ソフトウェア連携機能を利用する

統合Windows認証環境で外部ソフトウェア連携機能を利用するためには、以下の設定が必要です。

IM-Juggling で次の設定を行いwarファイルを作成してください。

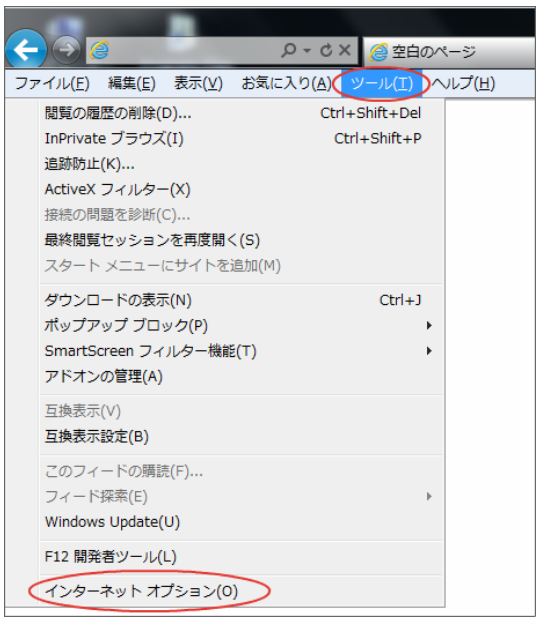
- 機能の有効化の設定
  1. <（プロジェクト名）/juggling.im>の「ベースモジュール」タブで 外部連携認証 認可(imaca_provider)、外部連携クライアント(imaca_client)を選択します。
     「設定ファイル」タブの外部連携 認証・認可を選択しconf/im-sso-windows-path-config/im-sso-windows-path-config_imaca_provider.xmlを出力します。
  2. <（プロジェクト名）/conf/im-sso-windows-path-config> 配下に出力されたim-sso-windows-path-config_imaca_provider.xmlを開き以下のようにコメントを外して、設定を有効化します。

```
<no-authentication>
  <path>/HTTPActionEventListener</path>
</no-authentication>
```

「インターネット オプション」の設定

Microsoft Edge は「インターネット オプション」の設定を変更することにより、Windows にログインしているユーザで自動的に intra-mart Accel Platform にログインできます。

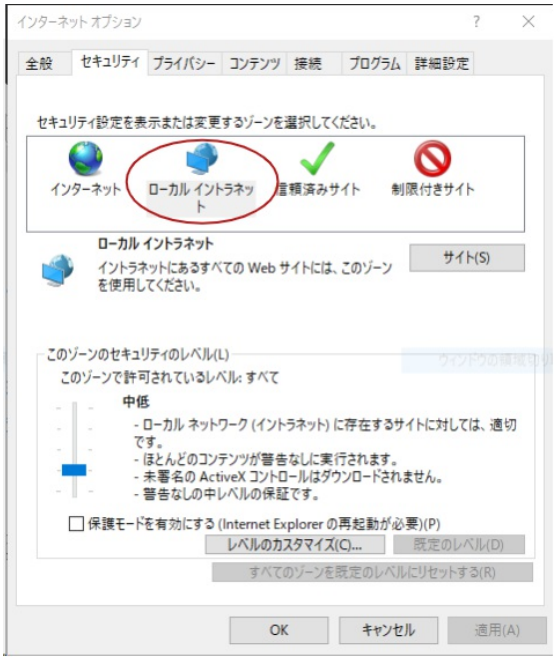
1. インターネット オプションを表示します。



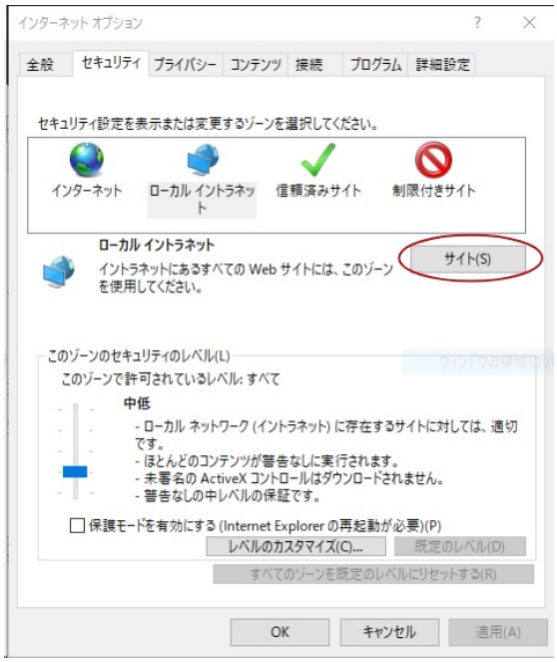
> **コラム**
>
> ツールメニューが表示されていない場合、Alt キーを押すことで表示できます。

> **コラム**
>
> インターネット オプションはコントロールパネルからも表示できます。

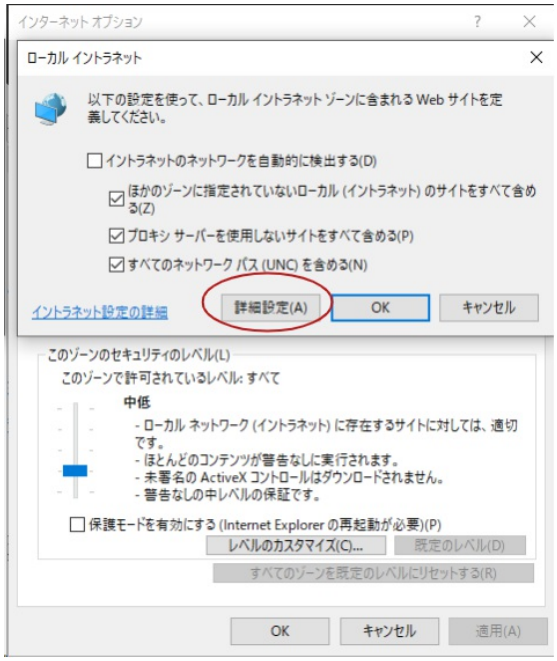
2. 「セキュリティ設定」画面を表示し「ローカル イントラネット」を選択します。



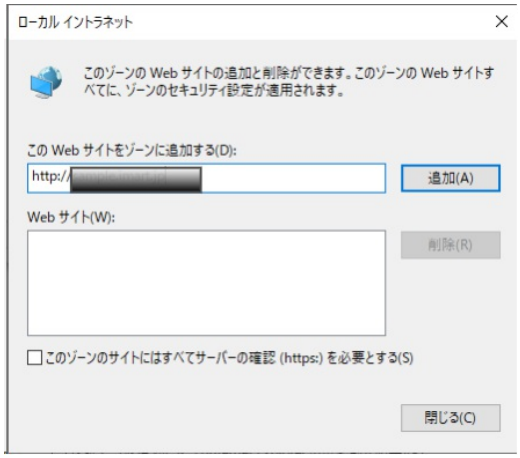
3. 「ローカル イントラネット」ゾーンの「サイト」画面を表示します。



4. 「詳細設定ボタン」を押下して「詳細設定」画面を表示します。

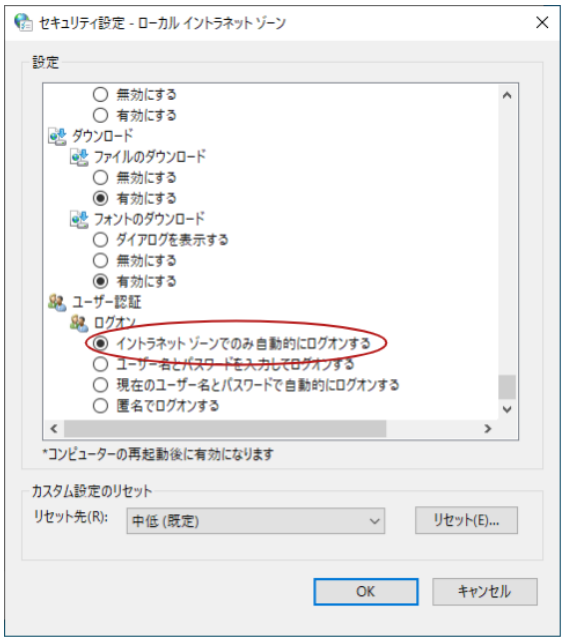


5. intra-mart Accel Platform に該当するURLを追加します。



6. 「ローカル イントラネット」ゾーンのレベルの「カスタマイズ」画面を表示します。

7. ユーザ認証の設定を変更します。
「イントラネットゾーンでのみ自動的にログオンする」を選択します。

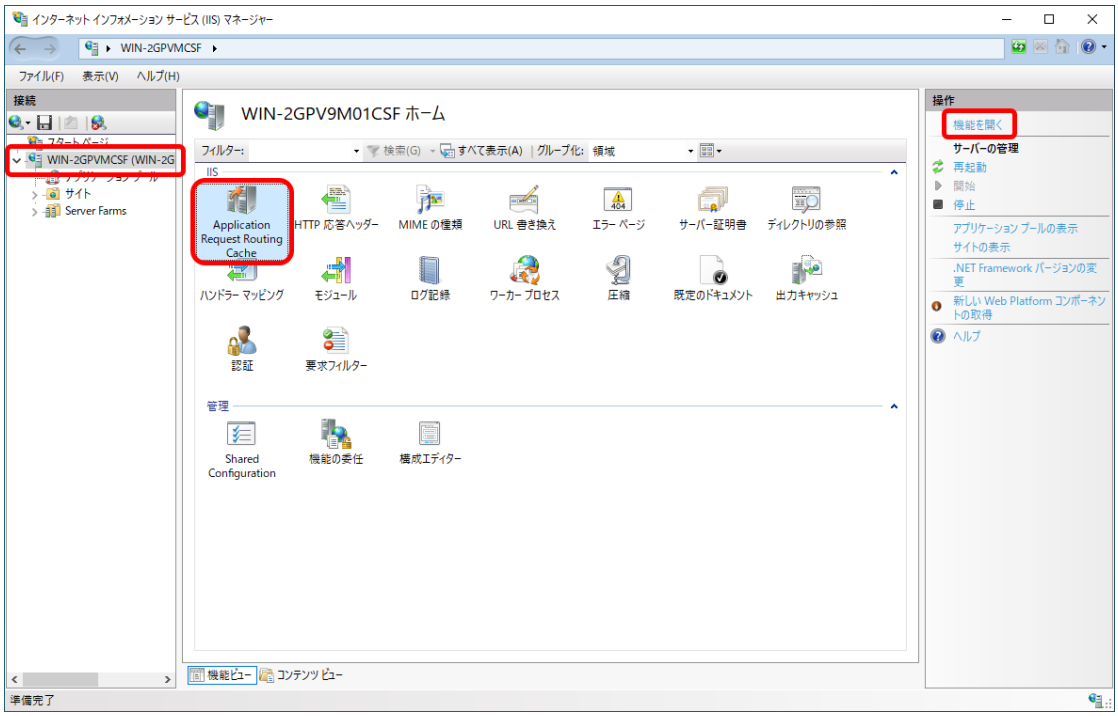


> **コラム**
>
> 上記の設定を行うことで、Microsoft Edge 利用時に自動的にユーザ認証を行うことができます。

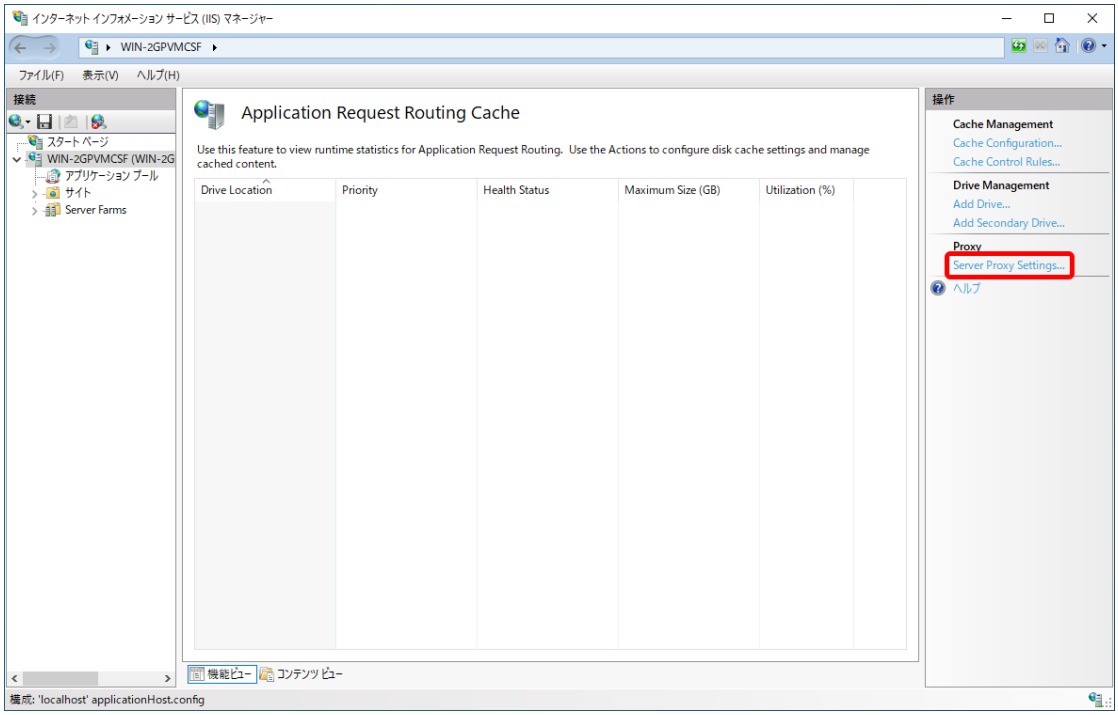
Keep-Alive の設定

統合Windows認証機能を利用するためには Keep-Alive を有効化する必要があります。

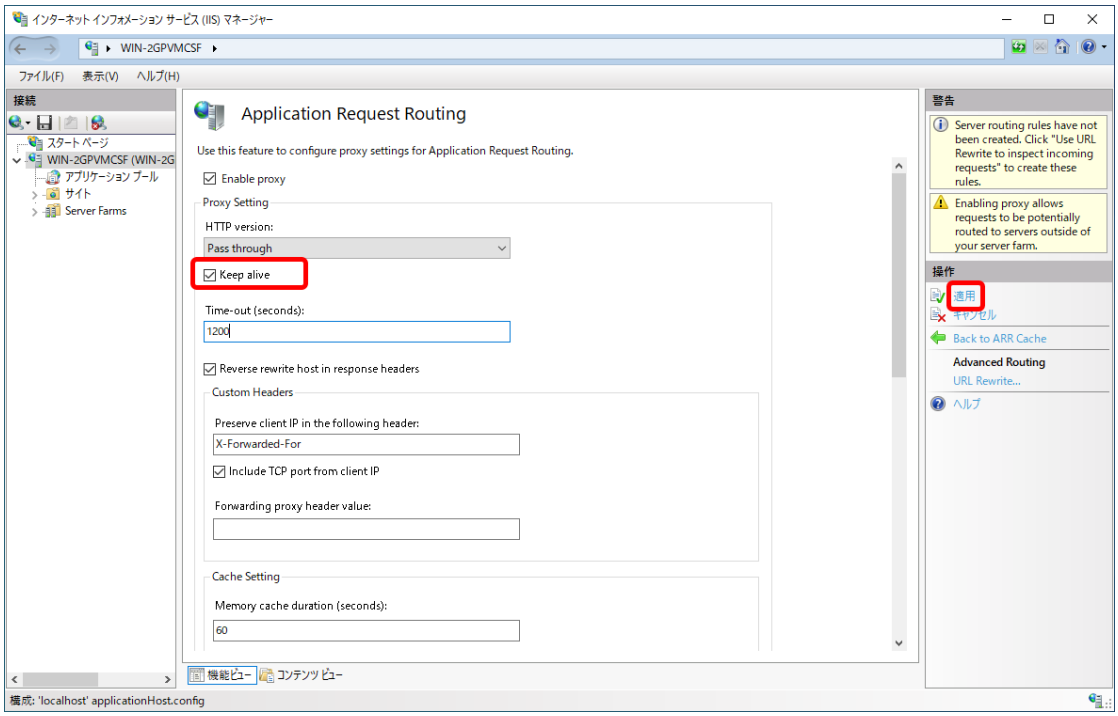
1. インターネット インフォメーション サービス (IIS) マネージャー より「IIS サーバ」「Application Request Routing Cache」「機能を開く」の順にクリックします。



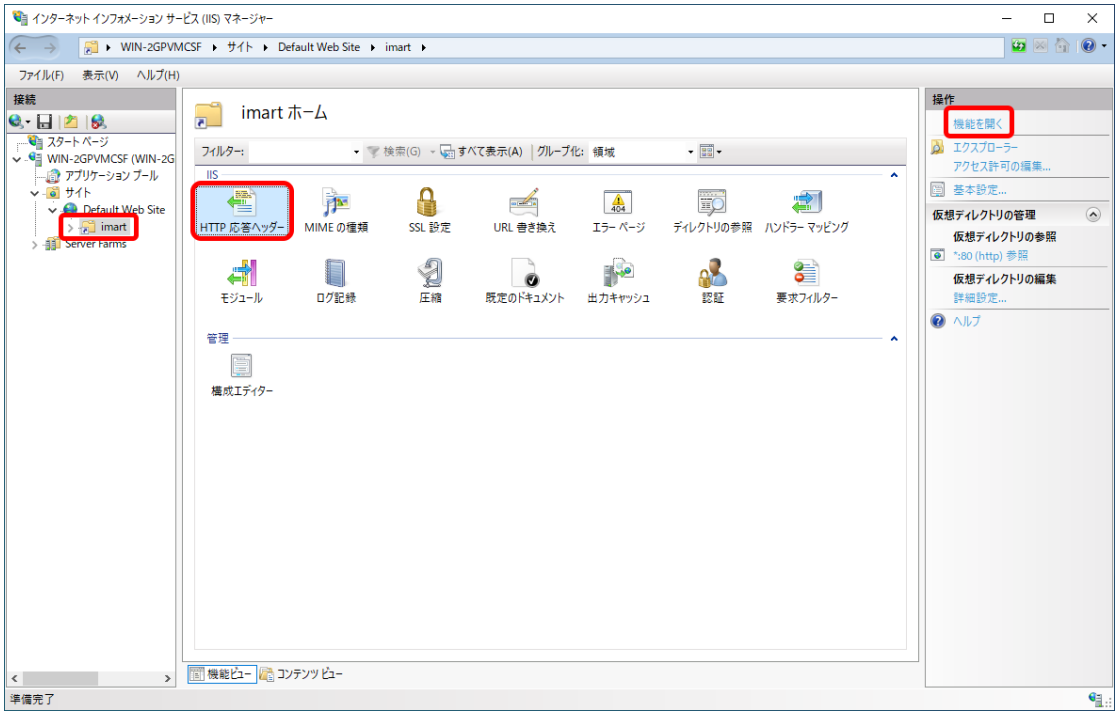
2. 「Server Proxy Settings...」をクリックし、「Application Request Routing」画面を表示します。



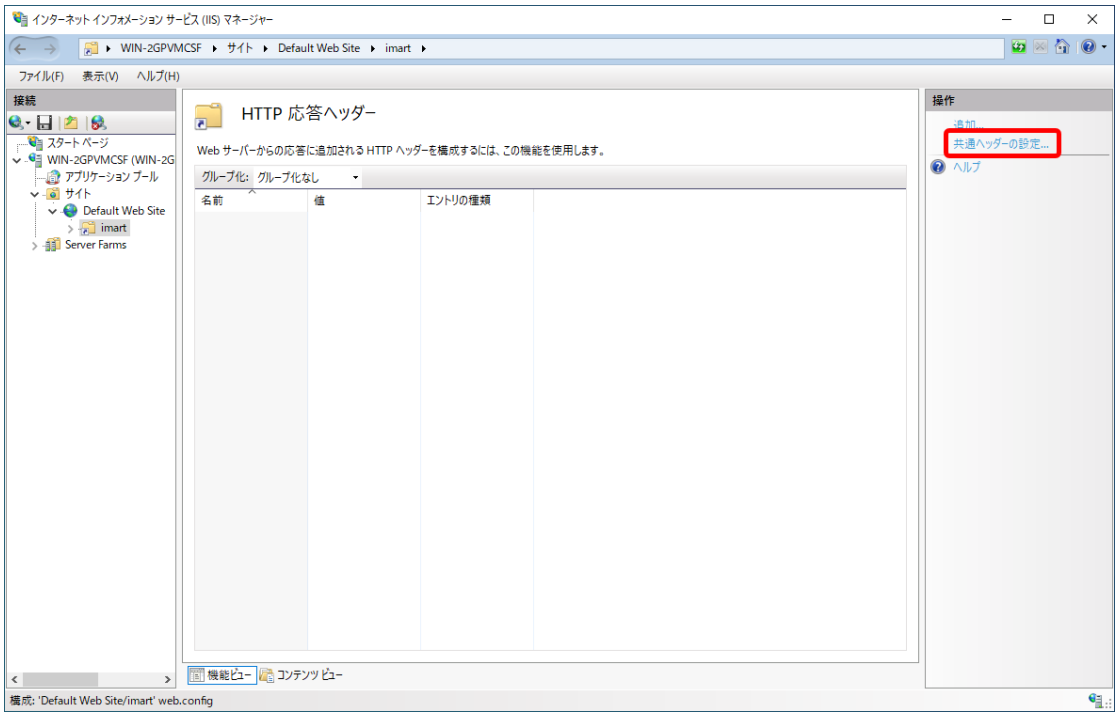
3. 「Keep alive」チェックボックスをオンにします。
   さらに、「適用」をクリックし、設定を反映します。



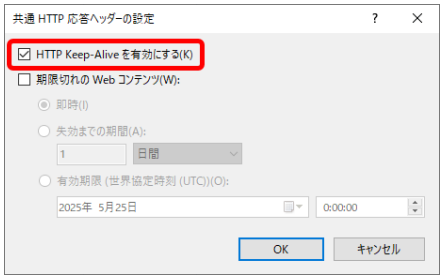
4. 「imart」ディレクトリを選択後、「HTTP 応答ヘッダー」、「機能を開く」をクリックします。



5. 「共通ヘッダーの設定」をクリックします。



6. 「HTTP Keep-Alive を有効にする」チェックボックスをオンにします。



統合Windows認証機能を無効化するには

統合Windows認証機能を無効化したい場合、以下の手順を実施するか、または単に統合Windows認証モジュールを含めずに war を作成し、再デプロイを行ってください。

IM-Juggling で次の設定を行いwarファイルを作成してください。

- 機能の無効化の設定
  1. <（プロジェクト名）/conf> 配下に出力されているim-sso-windows-config.xmlを開き<im-sso-windows-config>/<authentication>/<@enable> を false に設定します。

```
<?xml version="1.0" encoding="UTF-8"?>
<im-sso-windows-config
   xmlns="http://www.intra-mart.jp/sso-windows/config/im-sso-windows-config"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/sso-windows/config/im-sso-windows-config ../schema/im-sso-windows-config.xsd">
  <authentication enable="false">
    <parameter>
      <param-name>allow-guest-login</param-name>
      <param-value>false</param-value>
    </parameter>
  </authentication>
</im-sso-windows-config>
```

> **コラム**
>
> デフォルトではfalseが指定されています。

  2. plugin配下にフォルダを作成（例:「jp.co.intra_mart.foundation.security.certification.sso.user.provider.windows_8.0.0.disable」）し、そのフォルダ内に以下のようなplugin.xmlを配置してください。

version 属性にはすでに差し込まれているプラグイン ID の version 属性より大きい値を設定してください。

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin>
 <extension point="jp.co.intra_mart.foundation.security.certification.sso.user.providers">
  <sso-user-providers
   id="jp.co.intra_mart.foundation.security.certification.sso.user.provider.windows"
   name="Windows SSO User Provider"
   version="8.0.1"
   rank="90"
   enable="false">
   <sso-user-provider class="jp.co.intra_mart.foundation.security.certification.sso.WindowsSSOUserProvider"/>
  </sso-user-providers>
 </extension>
</plugin>
```

3. plugin配下にフォルダを作成（例:「jp.co.intra_mart.foundation.admin.tenant.context.tenant.resolver.windows_8.0.4.disable」）し、そのフォルダ内に以下のようなplugin.xmlを配置してください。
   version 属性にはすでに差し込まれているプラグイン ID の version 属性より大きい値を設定してください。

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin>
 <extension point="jp.co.intra_mart.foundation.admin.tenant.context.tenant.resolvers">
  <tenant-id-resolvers
   id="jp.co.intra_mart.foundation.admin.tenant.context.tenant.resolver.windows"
   name="Windows Tenant Id Resolver"
   version="8.0.5"
   rank="90"
   enable="false">

   <!-- ドメインをテナントIDとするリゾルバー
        参加しているドメインをテナントIDとして解決します。 -->
   <!--tenant-id-resolver class="jp.co.intra_mart.foundation.admin.tenant.context.DomainBasedWindowsTenantIdResolver"/-
->

   <!-- 設定された固定値をテナントIDとするリゾルバー
        tenant_idパラメータに設定した値をテナントIDとして解決します。 -->
   <!--tenant-id-resolver
class="jp.co.intra_mart.foundation.admin.tenant.context.FixedConfigBasedWindowsTenantIdResolver">
     <init-param>
      <param-name>tenant_id</param-name>
      <param-value>default</param-value>
     </init-param>
   </tenant-id-resolver-->

   <!-- 設定された固定値をテナントIDとするリゾルバー
        tenant_idパラメータに設定した値をテナントIDとして解決します。
        FixedConfigBasedWindowsTenantIdResolver と違い、どのようなリクエストに対しても必ず設定値を返却します。 -->
   <!--tenant-id-resolver
class="jp.co.intra_mart.foundation.admin.tenant.context.ConstantConfigBasedWindowsTenantIdResolver">
     <init-param>
      <param-name>tenant_id</param-name>
      <param-value>default</param-value>
     </init-param>
   </tenant-id-resolver-->
  </tenant-id-resolvers>
 </extension>
</plugin>
```

4. plugin配下にフォルダを作成（例:「jp.co.intra_mart.foundation.admin.tenant.context.tenant.validator.windows_8.0.1.disable」）し、そのフォルダ内に以下のようなplugin.xmlを配置してください。
   version 属性にはすでに差し込まれているプラグイン ID の version 属性より大きい値を設定してください。

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin>
 <extension point="jp.co.intra_mart.foundation.admin.tenant.context.tenant.validators">
  <tenant-id-validators
   id="jp.co.intra_mart.foundation.admin.tenant.context.tenant.validator.windows"
   name="Windows TenantIdValidator"
   version="8.0.2"
   rank="100"
   enable="false">
   <tenant-id-validator class="jp.co.intra_mart.system.sso_windows.context.WindowsSSOTenantIdValidator">
    <!-- テナントID解決必須チェック -->
    <init-param>
     <param-name>required_tenant_id</param-name>
     <param-value>true</param-value>
    </init-param>
    <!-- テナントID存在チェック -->
    <init-param>
     <param-name>valid_tenant_id</param-name>
     <param-value>true</param-value>
    </init-param>
   </tenant-id-validator>
  </tenant-id-validators>
 </extension>
</plugin>
```

> **コラム**
>
> プラグイン配下に作成するフォルダ名は任意です。
> 他のフォルダ名と重複することの無いようユニークな名称を設定してください。

統合Windows認証機能をリクエストに応じて無効化するには

intra-mart Accel Platform 2014 Summer(Honoka) 以降では、以下のリクエストヘッダを設定することで、統合Windows認証機能、テナントバリデーション機能を無効化できます。

リクエストヘッダ	値	動作
x-jp-co-intra-mart-sso-windows-disable-authentication	true	該当のリクエストにおいて統合Windows認証を行いません。
x-jp-co-intra-mart-sso-windows-disable-tenant-validation	true	該当のリクエストにおいてテナントバリデーションを行いません。

> **注意**
>
> 値には “true” を指定してください。その他の値を指定した場合の動作は保証しません。

Apache を使用して無効化する場合

Apache を利用して上記ヘッダを設定するには、以下の手順を行ってください。

1. <%APACHE_HOME%/conf/httpd.conf> ファイルを開きます。
2. Dynamic Shared Object (DSO) Supportエリアに以下の設定を追加またはコメントアウトを外してください。

```
LoadModule headers_module modules/mod_headers.so
```

3. 下記設定を追加してください。

```
RequestHeader append x-jp-co-intra-mart-sso-windows-disable-authentication "true"
RequestHeader append x-jp-co-intra-mart-sso-windows-disable-tenant-validation "true"
```

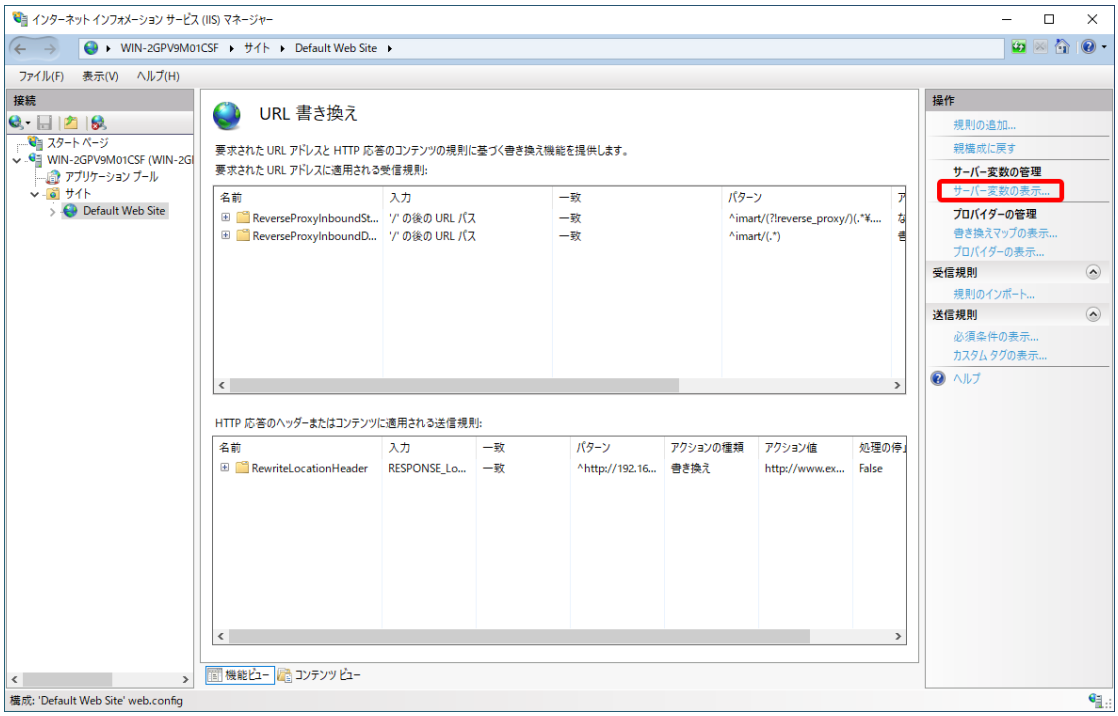
4. Apache を再起動してください。

Internet Information Services（IIS） を使用して無効化する場合

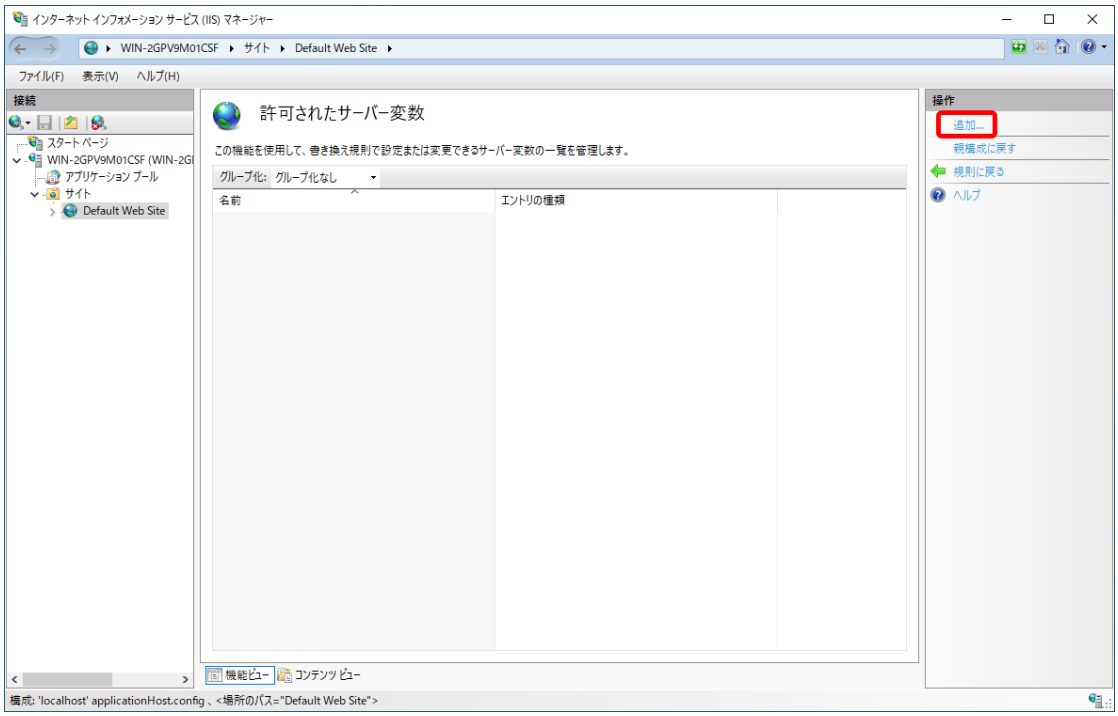
IIS を利用して上記ヘッダを設定するには、以下の手順を行ってください。

1. 統合Windows認証機能を無効化したいサイト、「URL 書き換え」、「機能を開く」の順にクリックします。

2. 「サーバ変数の表示」をクリックします。



3. 「追加」をクリックします。

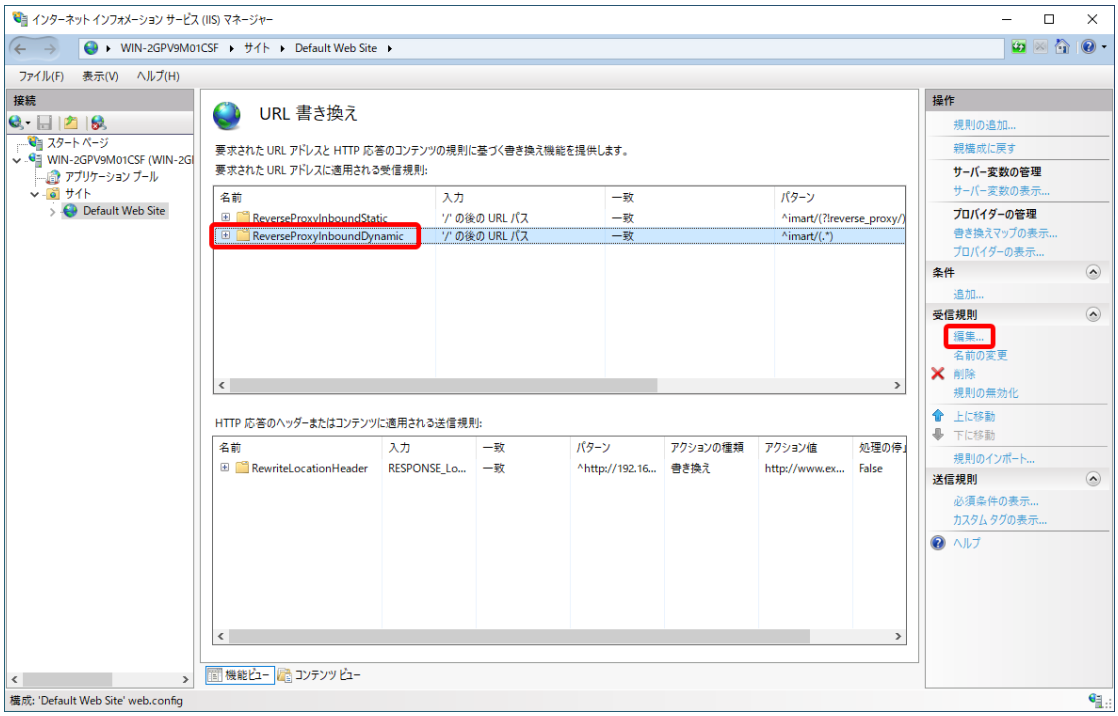


4. 「サーバー変数名」に「HTTP_X_JP_CO_INTRA_MART_SSO_WINDOWS_DISABLE_AUTHENTICATION」を入力し、「OK」をクリックします。

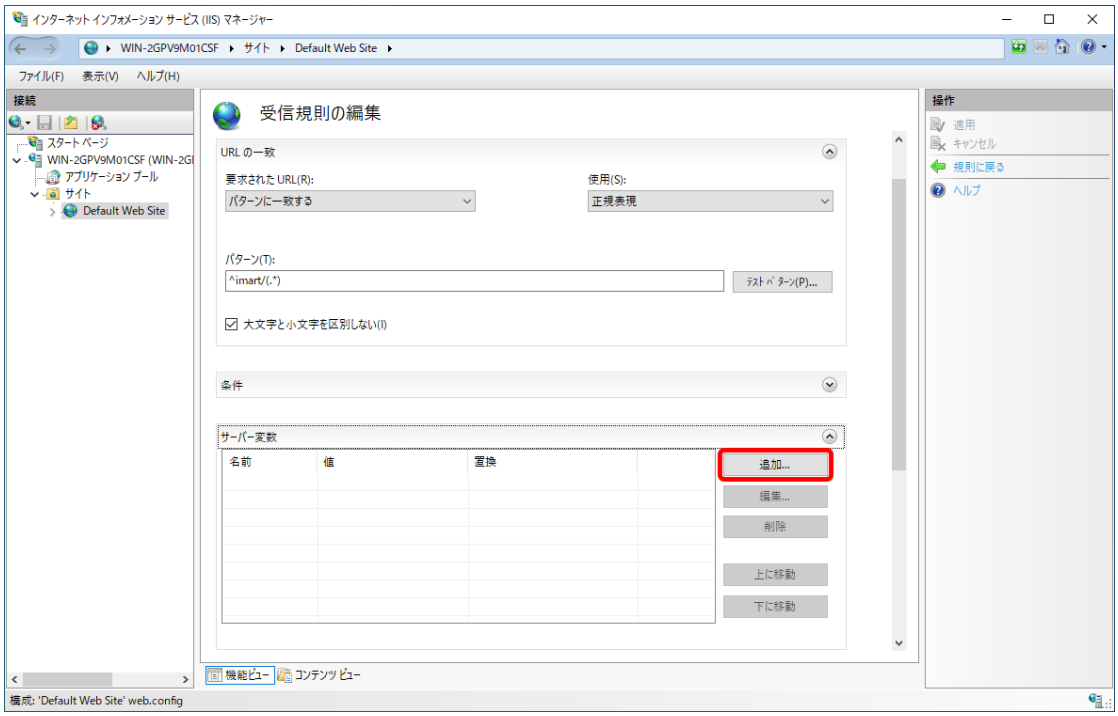
5. 同様に、サーバー変数「HTTP_X_JP_CO_INTRA_MART_SSO_WINDOWS_DISABLE_TENANT_VALIDATION」を追加します。

6. 統合Windows認証機能を無効化したいサイト、「URL 書き換え」、「機能を開く」の順にクリックします。

7. Resin への振り分けルール（ここでは ReverseProxyInboundDynamic）を選択し、編集をクリックします。

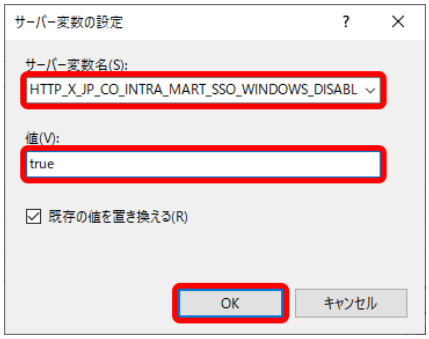


8. 「サーバー変数」の「追加」をクリックします。



9. 以下の内容を入力し、「OK」をクリックします。

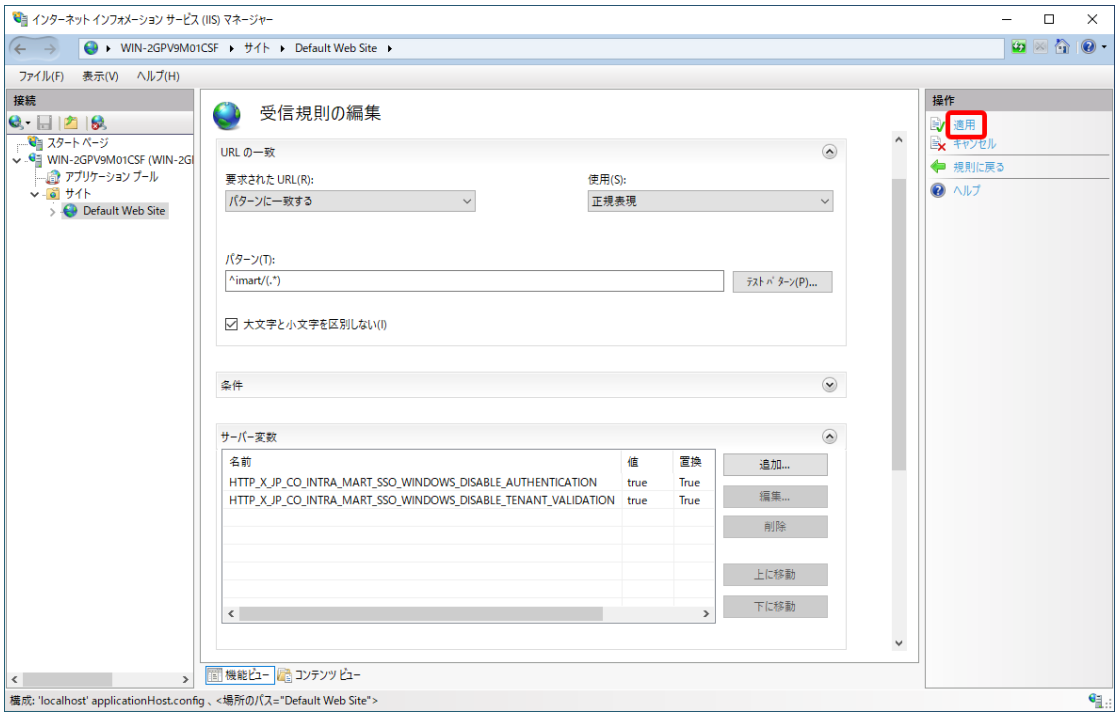
サーバー変数名	HTTP_X_JP_CO_INTRA_MART_SSO_WINDOWS_DISABLE_AUTHENTICATION
値	true



10. 同様に、以下のサーバ変数を追加します。

サーバー変数名	HTTP_X_JP_CO_INTRA_MART_SSO_WINDOWS_DISABLE_TENANT_VALIDATION
値	true

11. 「適用」をクリックします。



12. Internet Information Services（IIS） を再起動します。

統合Windows認証機能の認証失敗時に通常のログイン機能を利用するには

intra-mart Accel Platform 2014 Winter(Iceberg) 以降では、<（プロジェクト名）/conf> 配下に出力されている im-sso-windows-config.xml の allow-fallback-login パラメータに true を設定することで、統合Windows認証機能による認証に失敗したユーザは通常のログイン機能を使用できます。

```
<?xml version="1.0" encoding="UTF-8"?>
<im-sso-windows-config
   xmlns="http://www.intra-mart.jp/sso-windows/config/im-sso-windows-config"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/sso-windows/config/im-sso-windows-config ../schema/im-sso-windows-config.xsd">
 <authentication enable="true">
   <parameter>
    <param-name>allow-guest-login</param-name>
    <param-value>false</param-value>
   </parameter>
   <parameter>
    <param-name>allow-fallback-login</param-name>
    <param-value>true</param-value>
   </parameter>
 </authentication>
</im-sso-windows-config>
```

注意

本機能における、 統合Windows認証 機能の認証失敗の条件は以下のような場合です。

- ユーザ認証に使用されるユーザ名、パスワードがドメインコントローラー上のものと一致しない場合。
- 認証ダイアログによる認証要求をキャンセルした場合。
- ゲストログインが無効の場合に、ドメインコントローラー上のゲストユーザで認証をした場合。

注意

本機能有効時に 統合Windows認証 に失敗したユーザに対しては、以降セッションが破棄されるまで 統合Windows認証は行われません。

ケルベロス認証を無効化しNTLM認証を強制するには

NTLM認証を強制するには、以下の設定が必要です。

IM-Juggling で次の設定を行いwarファイルを作成してください。

- NTLM認証強制化の設定
  1. 「設定ファイル」タブの統合Windows認証モジュールを選択しim-sso-windows-config.xmlを出力します。
  2. <（プロジェクト名）/conf配下に出力されたim-sso-windows-config.xmlを開き以下のようにコメントを外して、設定を有効化します。

```
<parameter>
 <param-name>waffle.servlet.spi.NegotiateSecurityFilterProvider/protocols</param-name>
 <param-value>NTLM</param-value>
</parameter>
```

注意

本設定は以下のバージョンでのみ利用可能です。

- 統合Windows認証モジュールバージョン 8.0.1-PATCH_001 以降
- 統合Windows認証モジュールバージョン 8.0.2-PATCH_001 以降
- 統合Windows認証モジュールバージョン 8.0.3-PATCH_001 以降
- 統合Windows認証モジュールバージョン 8.0.4 以降

統合Windows認証機能の認証失敗時にリダイレクトさせるには

intra-mart Accel Platform 2015 Winter(Lydia) 以降では、<（プロジェクト名）/conf> 配下に出力されている im-sso-windows-config.xml の redirect-on-authentication-failure パラメータに URL を設定することで、統合Windows認証機能による認証に失敗したユーザに指定した URL にリダイレクトさせることができます。

```
<?xml version="1.0" encoding="UTF-8"?>
<im-sso-windows-config
   xmlns="http://www.intra-mart.jp/sso-windows/config/im-sso-windows-config"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/sso-windows/config/im-sso-windows-config ../schema/im-sso-windows-config.xsd">
 <authentication enable="true">
   <parameter>
    <param-name>allow-guest-login</param-name>
    <param-value>false</param-value>
   </parameter>
   <parameter>
    <param-name>allow-fallback-login</param-name>
    <param-value>false</param-value>
   </parameter>
   <!--parameter>
    <param-name>waffle.servlet.spi.NegotiateSecurityFilterProvider/protocols</param-name>
    <param-value>NTLM</param-value>
   </parameter-->
   <parameter>
    <param-name>redirect-on-authentication-failure</param-name>
    <param-value></param-value>
   </parameter>
 </authentication>
</im-sso-windows-config>
```

注意

本機能における、統合Windows認証機能の認証失敗の条件は以下のような場合です。

- ユーザ認証に使用されるユーザ名、パスワードがドメインコントローラー上のものと一致しない場合。
- ゲストログインが無効の場合に、ドメインコントローラー上のゲストユーザで認証をした場合。

> **注意**
>
> 本機能を利用して intra-mart Accel Platform へリダイレクトを行う場合、リダイレクト先の URL に対し統合Windows認証が行われます。
> これを避けたい場合、以下の機能等を用いて、リダイレクト先 URL に対する統合Windows認証を無効化してください。
>
> - *統合Windows認証機能をリクエストに応じて無効化するには*
> - *統合Windows認証機能の認証失敗時に通常のログイン機能を利用するには*
> - 統合Windows認証パス設定

SMTP認証で OAuth2.0 アクセストークン を使用する

項目

- 概要
- アクセストークンの発効
  - サービスの設定
  - プロバイダ情報を設定する
  - 連携許可を行う
  - SMTPサーバ設定を行う

概要

SMTPサーバの設定を行うことで、メール通知などが使用できます。
2022 Spring(Eustoma) 以降のバージョンではSMTP認証に、OAuth2.0認可におけるアクセストークンを使用できます。
また、 2022 Spring(Eustoma) 以降のバージョンでは、SMTPサーバ情報をシステム管理画面から設定できます。

- 画面でSMTPサーバ設定を行う
  「システム管理者操作ガイド」 - 「SMTPサーバ設定」
- 設定ファイルでSMTPサーバ設定を行う
  「設定ファイルリファレンス」 - 「メール設定」

ここではOAuth2.0 アクセストークンを使用したSMTP認証の手順について説明します。

アクセストークンの発効

アクセストークンをSMTP認証で利用するためには、システム用OAuthプロバイダ設定でプロバイダ情報を登録し、システム用外部連携アプリケーションで連携の許可を行って、アクセストークンを発効する必要があります。

- 「システム管理者操作ガイド」 - 「システム用OAuthプロバイダ設定」
- 「システム管理者操作ガイド」 - 「システム用外部連携アプリケーション」

サービスの設定

メールサーバに Exchange Online を利用する場合の設定

OAuth認証に必要な関連サービスの準備を行います。
本項の内容は Microsoft Azure 管理者 向けの作業です。

アプリケーションを登録する

Microsoft Azure の管理ポータルから OAuth認証 に必要な情報をアプリケーションとして登録します。

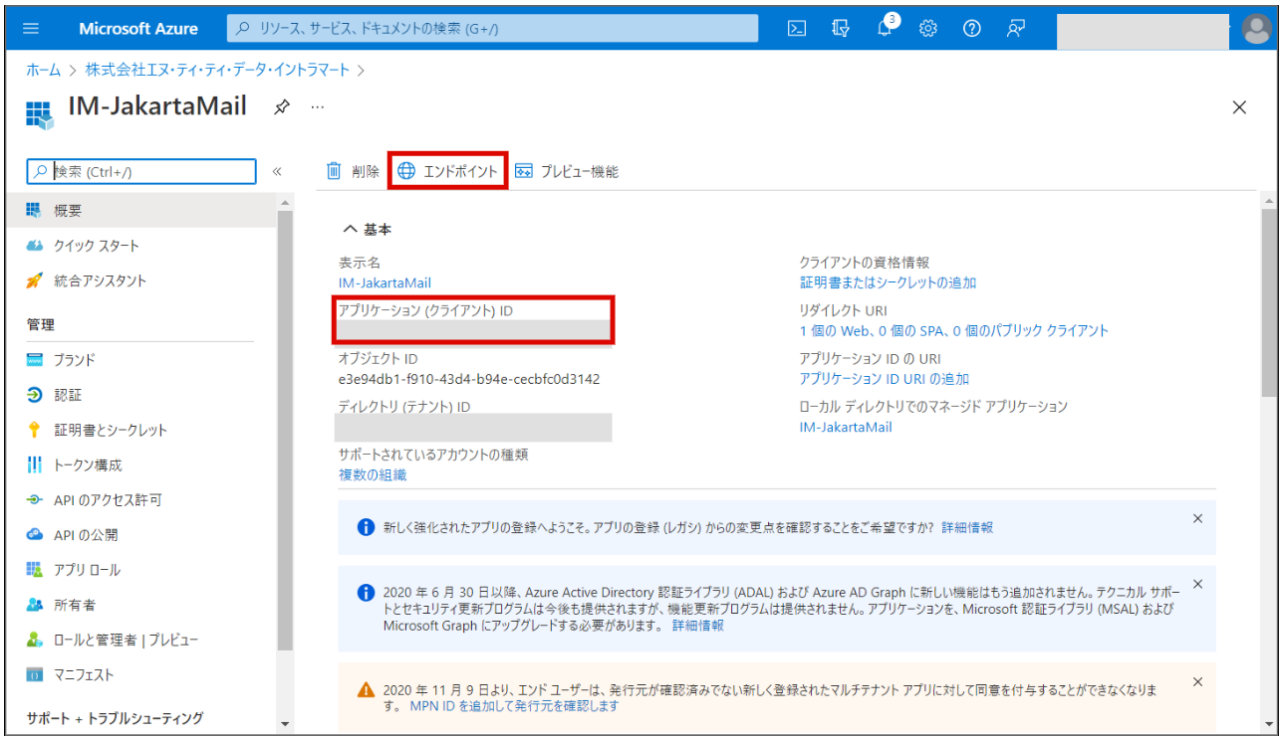
1. 以下のURLから Microsoft Azure の管理ポータルに **Microsoft Azure 管理者ユーザ** でサインインします。
   - https://portal.azure.com/
2. サイドメニューから「Microsoft Entra ID」をクリックします。
3. 現在のテナントが「 OAuth認証を利用する組織のテナント」ではない場合は「テナントの切り替え」を行います。
4. 「概要」のサイドメニュー「管理」の「アプリの登録」をクリックします。
5. 「新規登録」をクリックします。
6. 以下を入力または選択して「登録」をクリックします。
   - 名前に任意の名称を入力

- サポートされているアカウントの種類に「任意の組織ディレクトリ内のアカウント（任意の Microsoft Entra ID テナント - マルチテナント ）」を選択
- リダイレクト URI に「Web」を選択し、 intra-mart Accel Platform のベース**URL + /system/oauth/redirect**を入力

7. 以上でアプリケーションの登録は完了です。

   以下の内容は intra-mart Accel Platform システム管理者 がプロバイダ情報を設定する際に利用します。
   エンドポイントはアプリケーション概要の「エンドポイント」より値を取得してください。

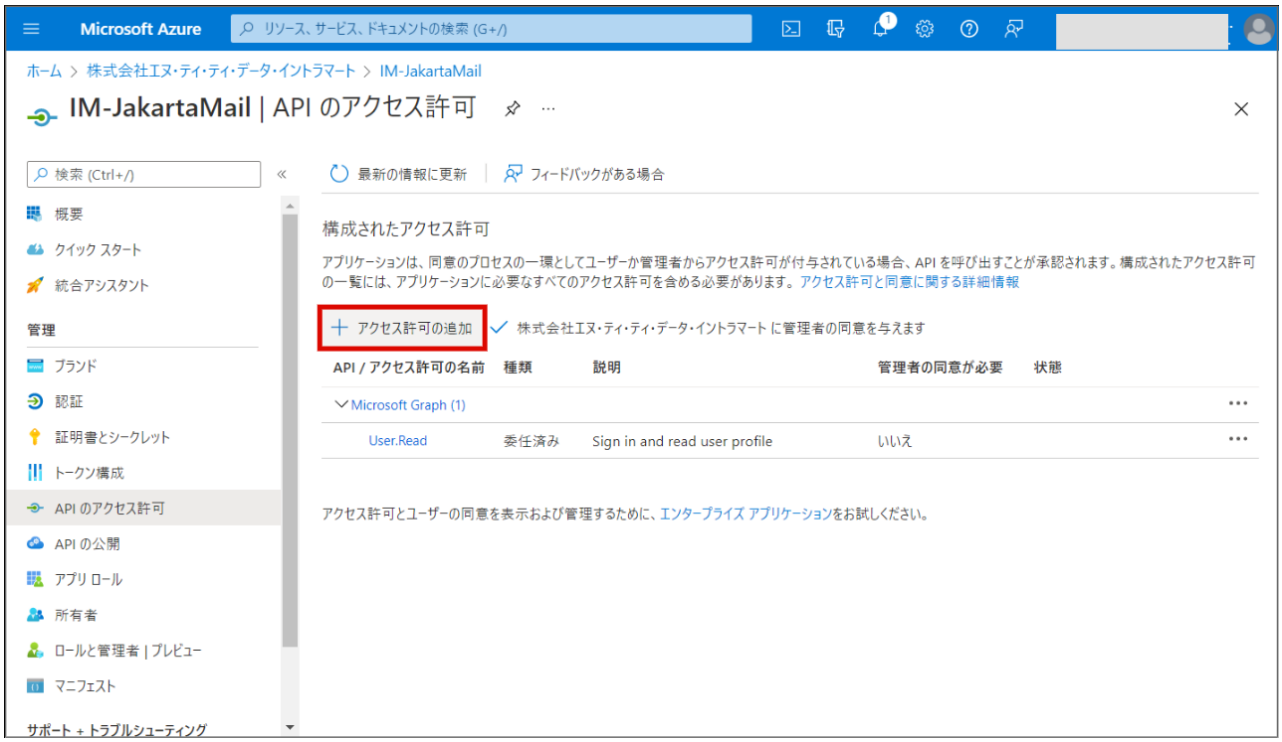
   - アプリケーションID （クライアントIDとして利用します）
   - OAuth 2.0 承認エンドポイント (v2) （認可エンドポイントとして利用します）
   - OAuth 2.0 トークン エンドポイント (v2)（トークンエンドポイントとして利用します ）



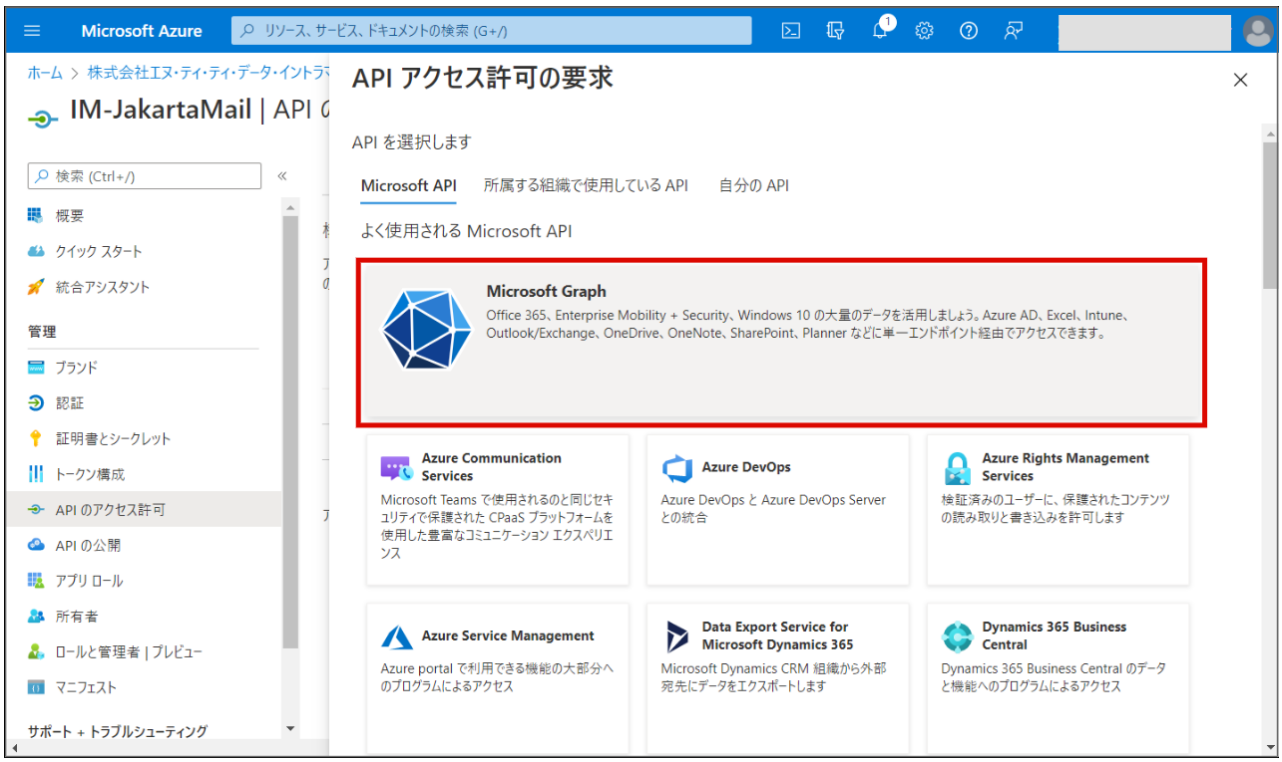
アプリケーションを設定する

Microsoft Azure の管理ポータルから登録したアプリケーションの構成を変更します。

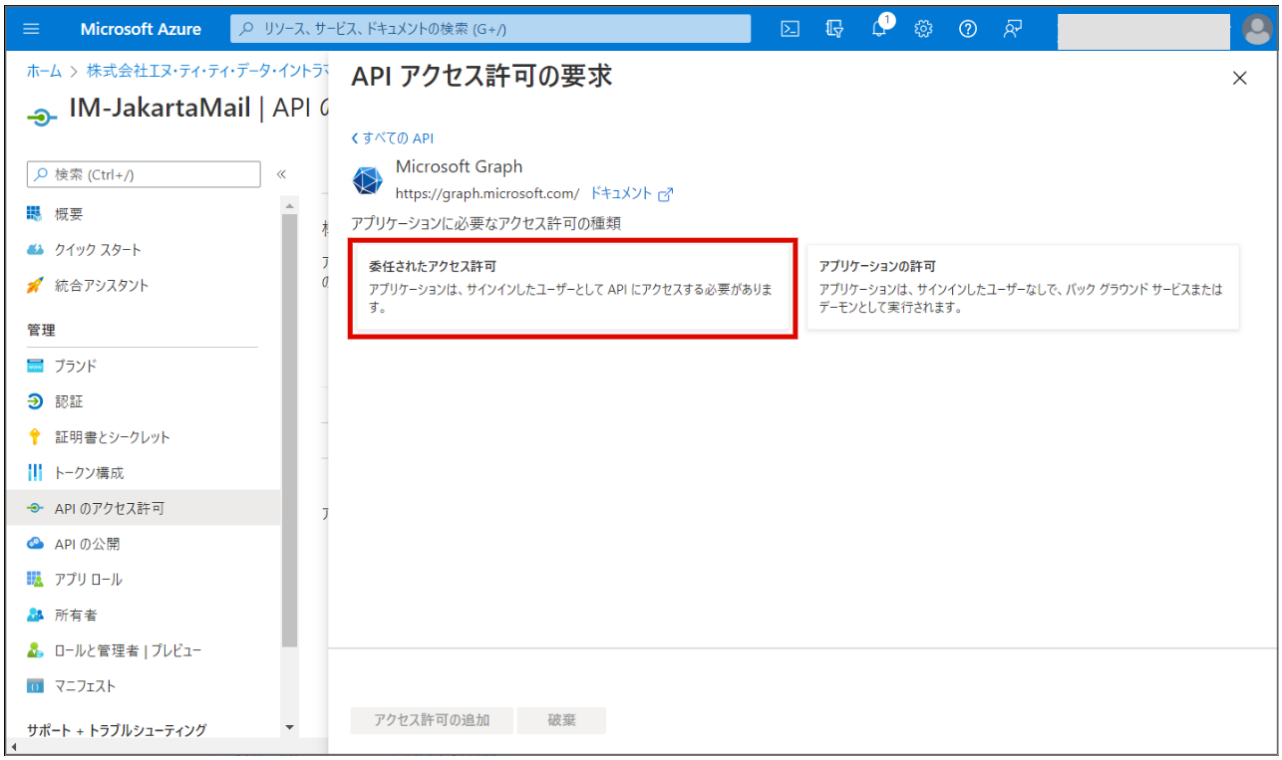
1. 先程登録したアプリの「管理」の「APIのアクセス許可」をクリックします。
2. 「アクセス許可の追加」をクリックします。



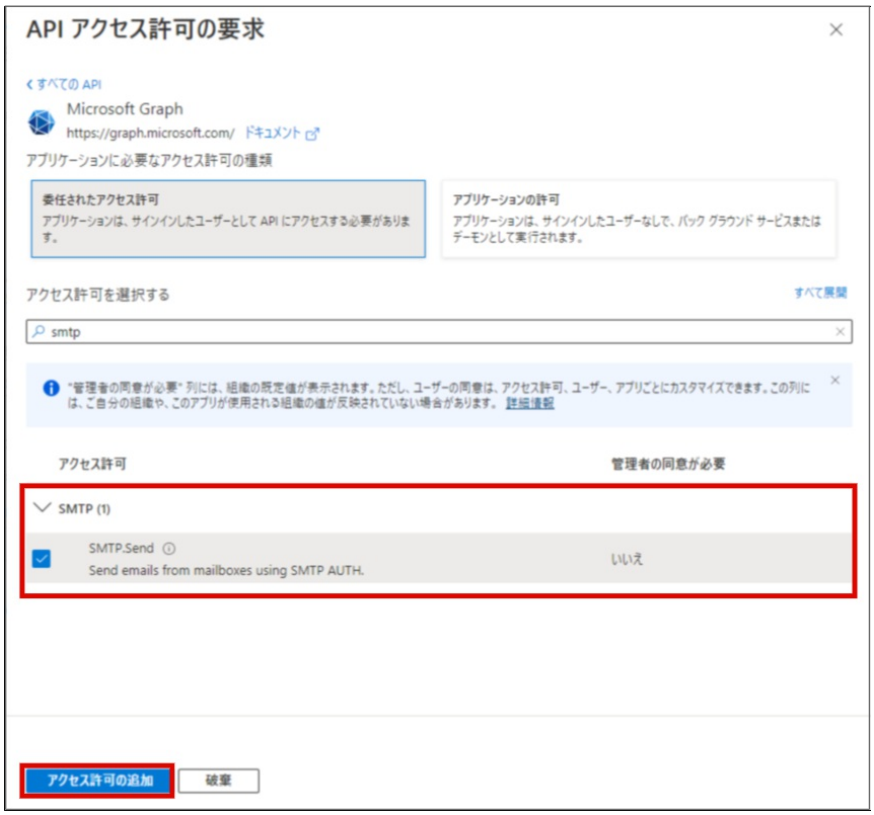
3. 「 Microsoft Graph 」をクリックします。



4. 「アプリケーションに必要なアクセス許可の種類」の「委任されたアクセス許可」をクリックします。

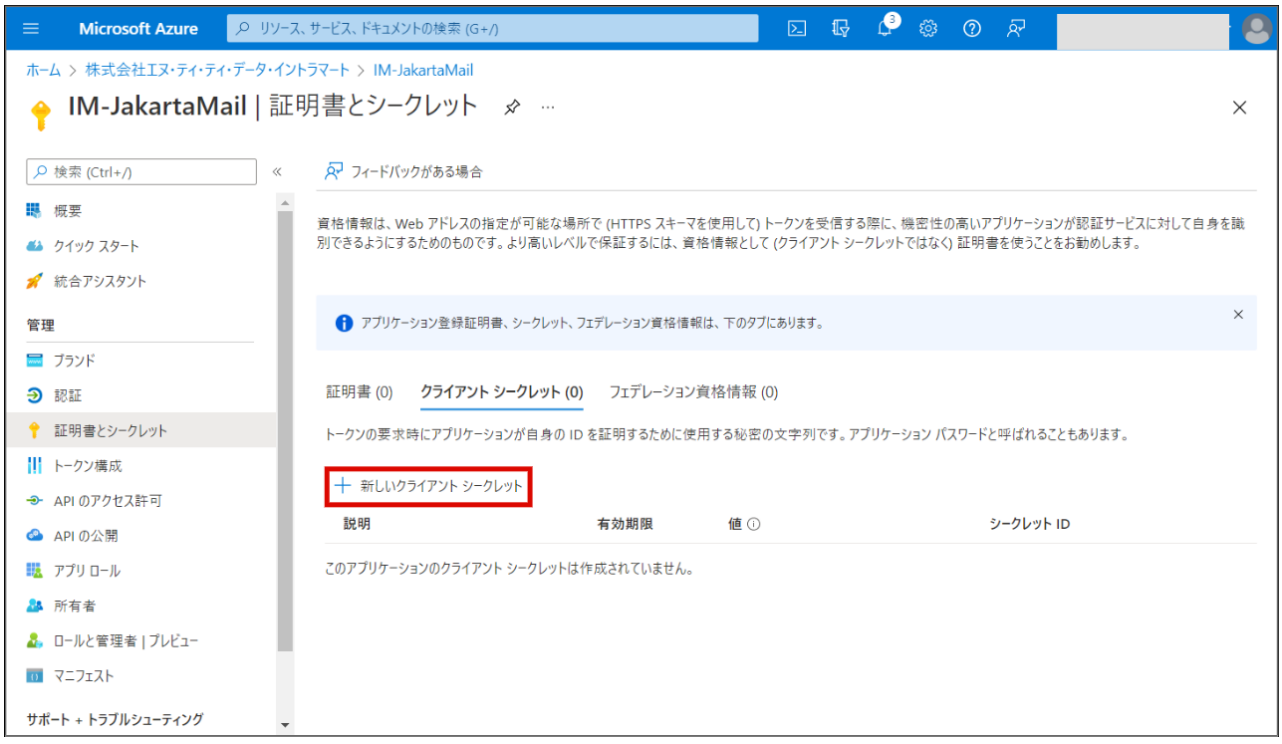


5. 「SMTP.Send」、「offline_access」を選択し、「アクセス許可の追加」をクリックします。



> **コラム**
>
> 「APIアクセス」の「アクセスの有効化」にて「委任されたアクセス許可」におけるメール操作の許可設定についての詳細は Microsoft社 の以下のドキュメントを参照してください。
>
> - Microsoft Graph permissions reference :
>   - https://docs.microsoft.com/en-us/graph/permissions-reference （English）
>   - https://docs.microsoft.com/ja-jp/graph/permissions-reference （日本語）
>   - https://docs.microsoft.com/zh-cn/graph/permissions-reference （中文）

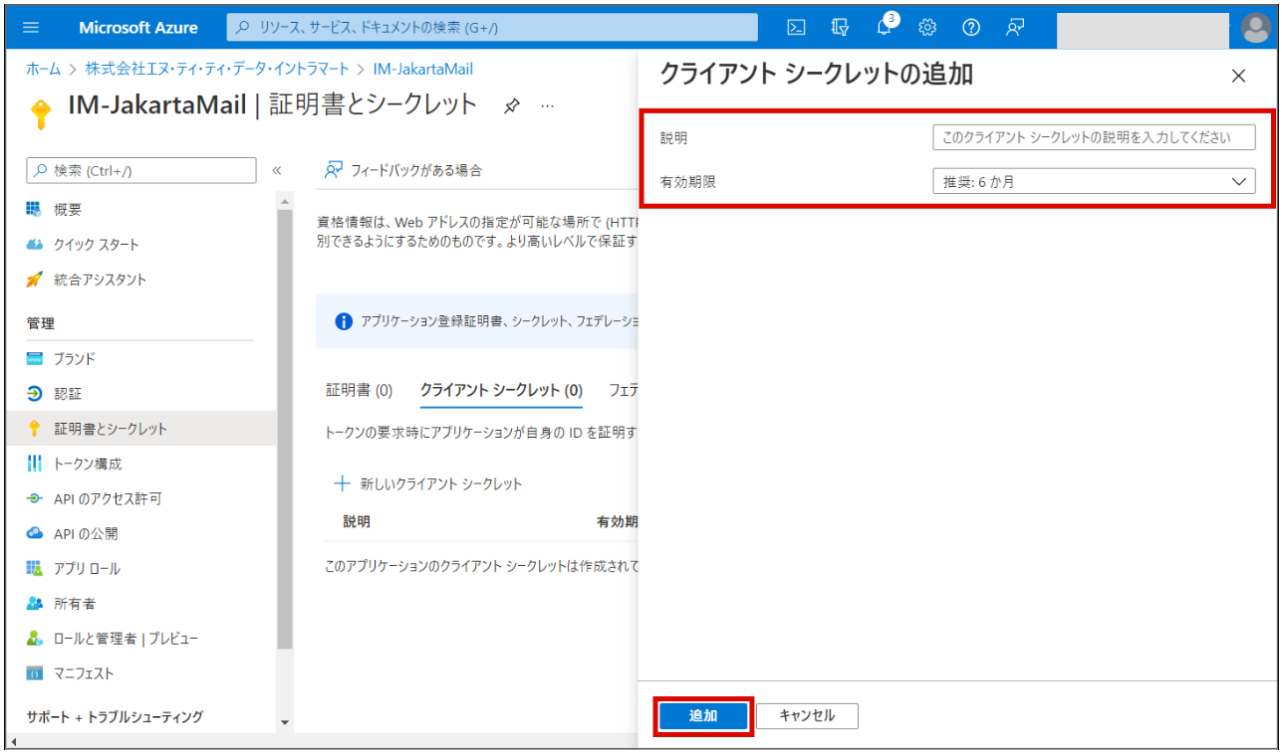
6. 「管理」の「証明書とシークレット」をクリックします。
7. 「クライアント シークレット」の「新しいクライアント シークレット」をクリックします。



8. intra-mart Accel Platform からアクセスする際に必要なキーを生成します。

以下を入力または選択して「追加」をクリックします。

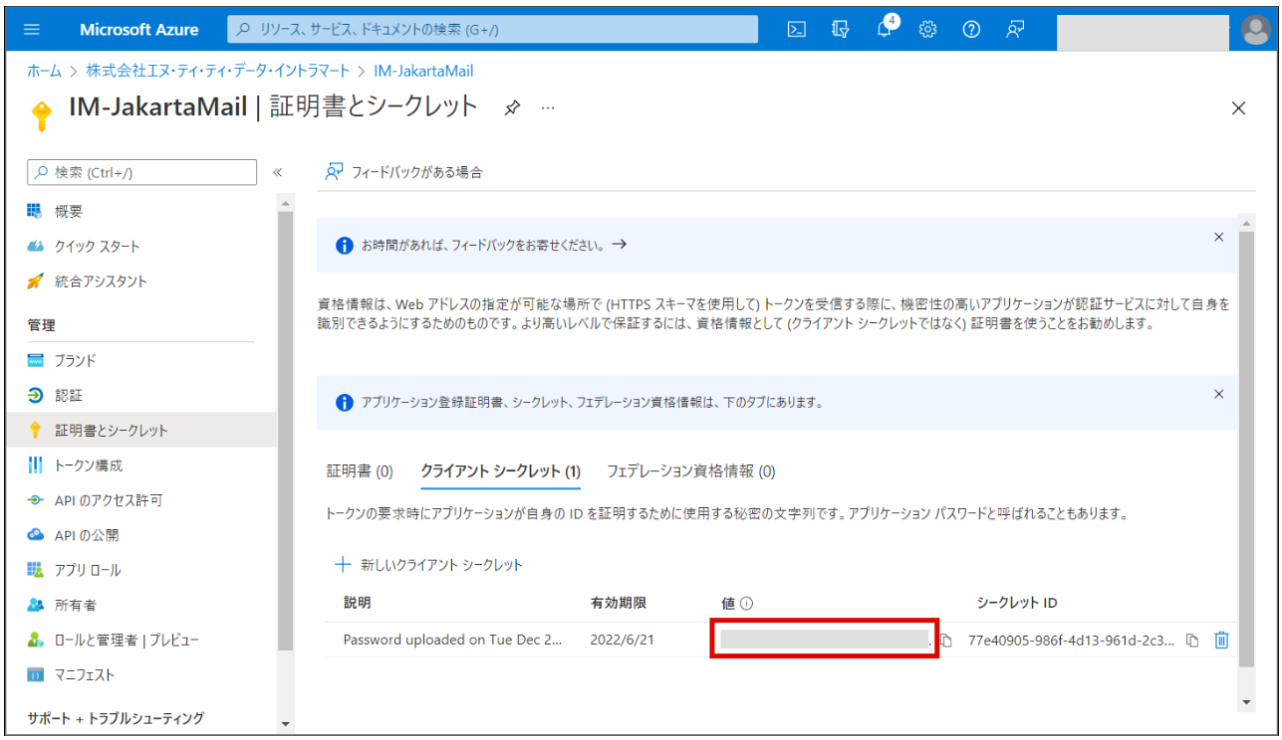
- 説明に任意のキーの説明を入力
- 有効期間に任意のキーの有効期限を選択



以下の内容は intra-mart Accel Platform システム管理者 がプロバイダ情報を設定する際に利用します。

- クライアントシークレット値（クライアントシークレットとして利用します）

> **注意**
> キーは設定の保存後に一度のみ表示されます。移動前にキーの表示内容を退避させてください。



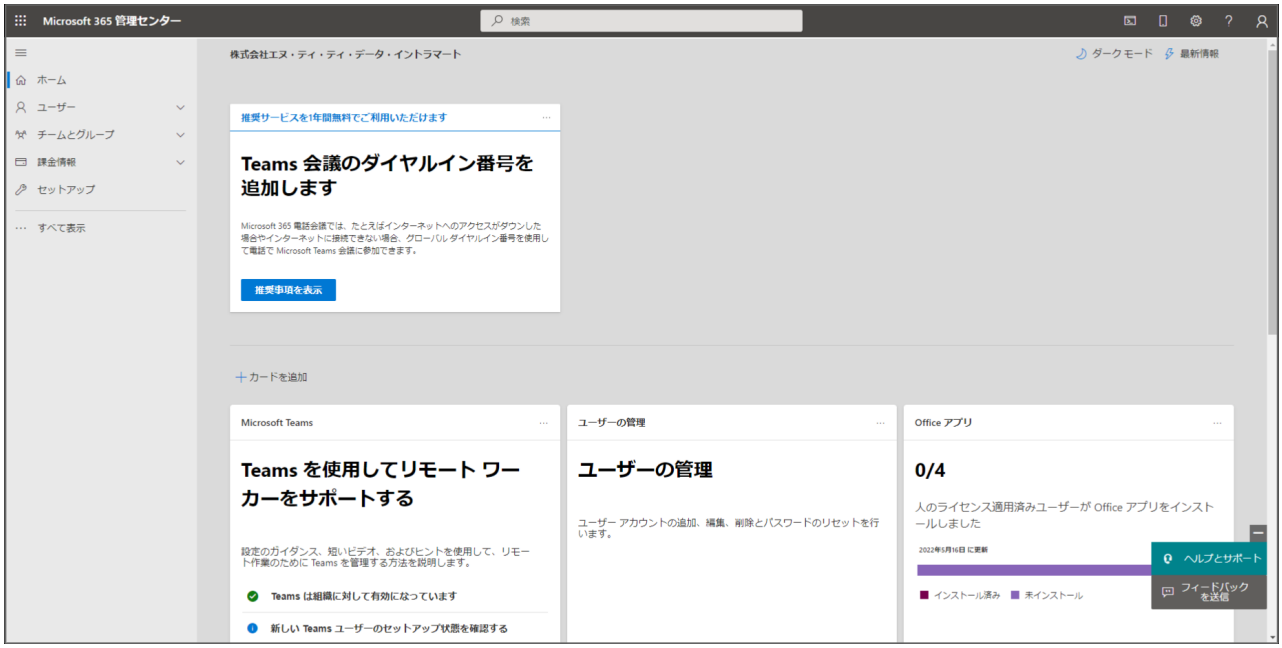
> **コラム**
> 有効期限が切れた場合は、上記の手順でキーを再発行する必要があります。

9. 以上でアプリケーションの設定は完了です。

送信アクセスの許可を行う

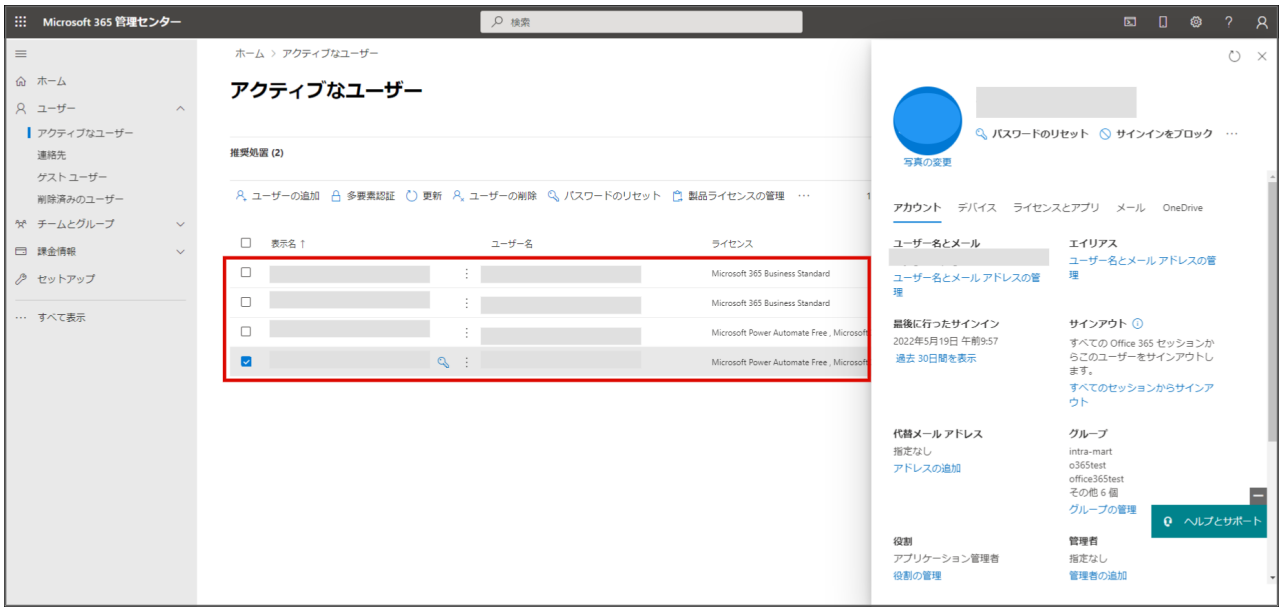
SMTPサーバの認証ユーザで、別のユーザ名義を送信者にするための設定を行います。

1. 以下のURLから *Microsoft 365 管理センター* に **Microsoft 365 管理者ユーザ** でサインインします。
   - https://admin.microsoft.com/



2. サイドメニュー「ユーザ」 - 「アクティブなユーザ」をクリックします。

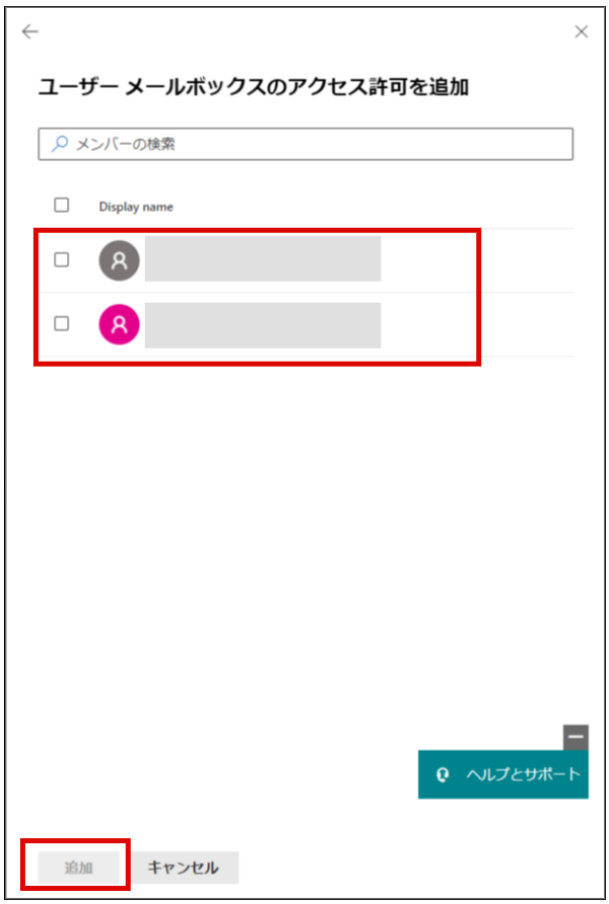
3. 送信者として指定したいユーザの表示名をクリックします。



4. メールタブの「メールボックス所有者として送信するアクセス許可」をクリックします。

5. アクセス許可を追加をクリックします。

6. SMTP サーバの認証ユーザをアクセス許可されるユーザとして選択し、「追加」をクリックします。

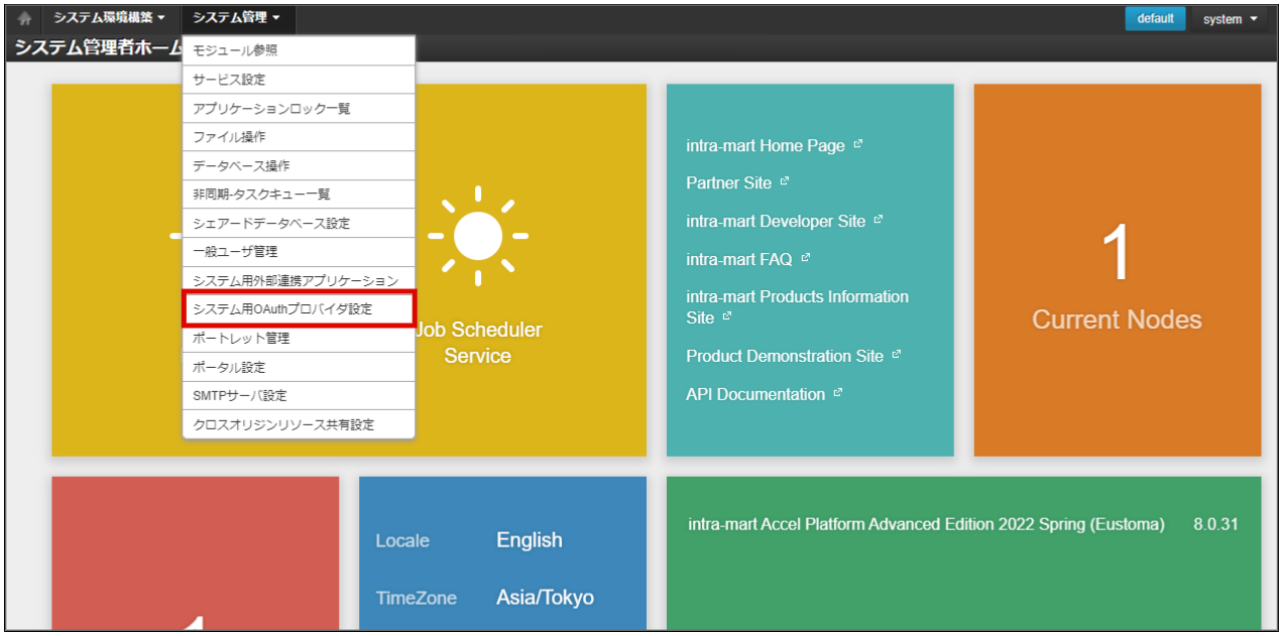


> **コラム**
>
> 「*送信アクセスの許可を行う*」でアクセス許可されるユーザ、「*連携許可を行う*」でサインインするユーザ、「*SMTPサーバ設定を行う*」で設定するユーザは、すべて同一ユーザを設定してください。

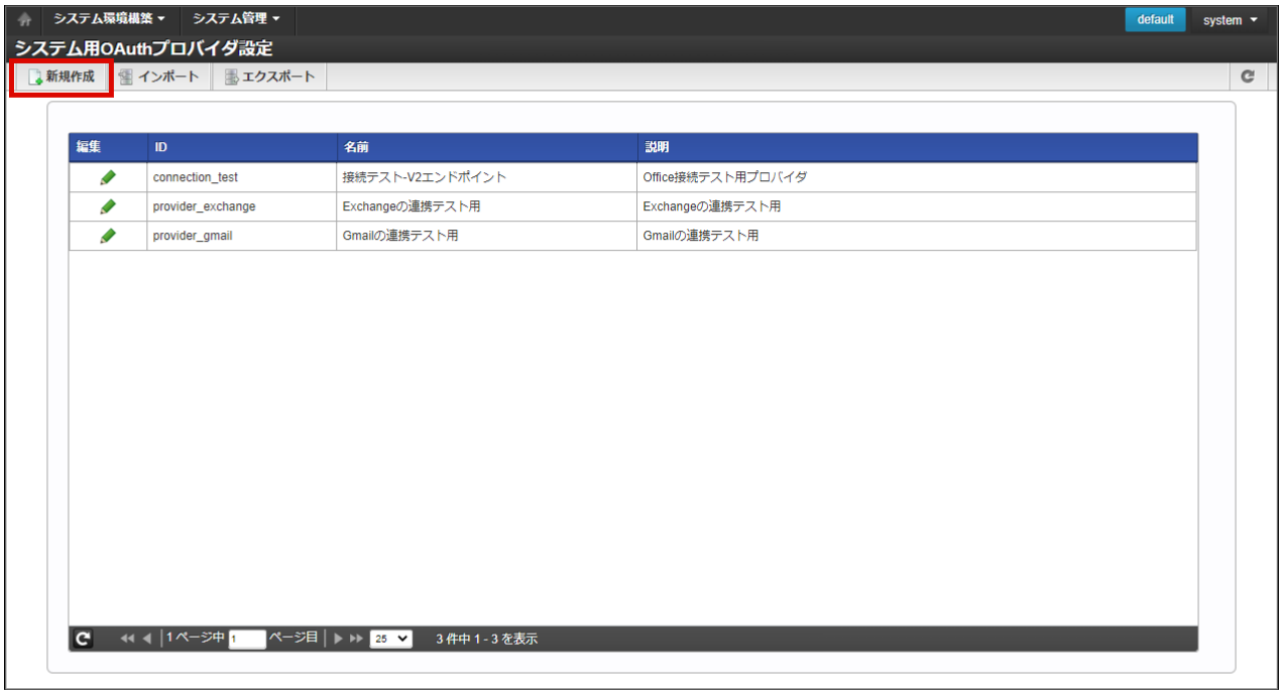
7. 送信者として指定したいユーザすべてに対して行えば完了です。

プロバイダ情報を設定する

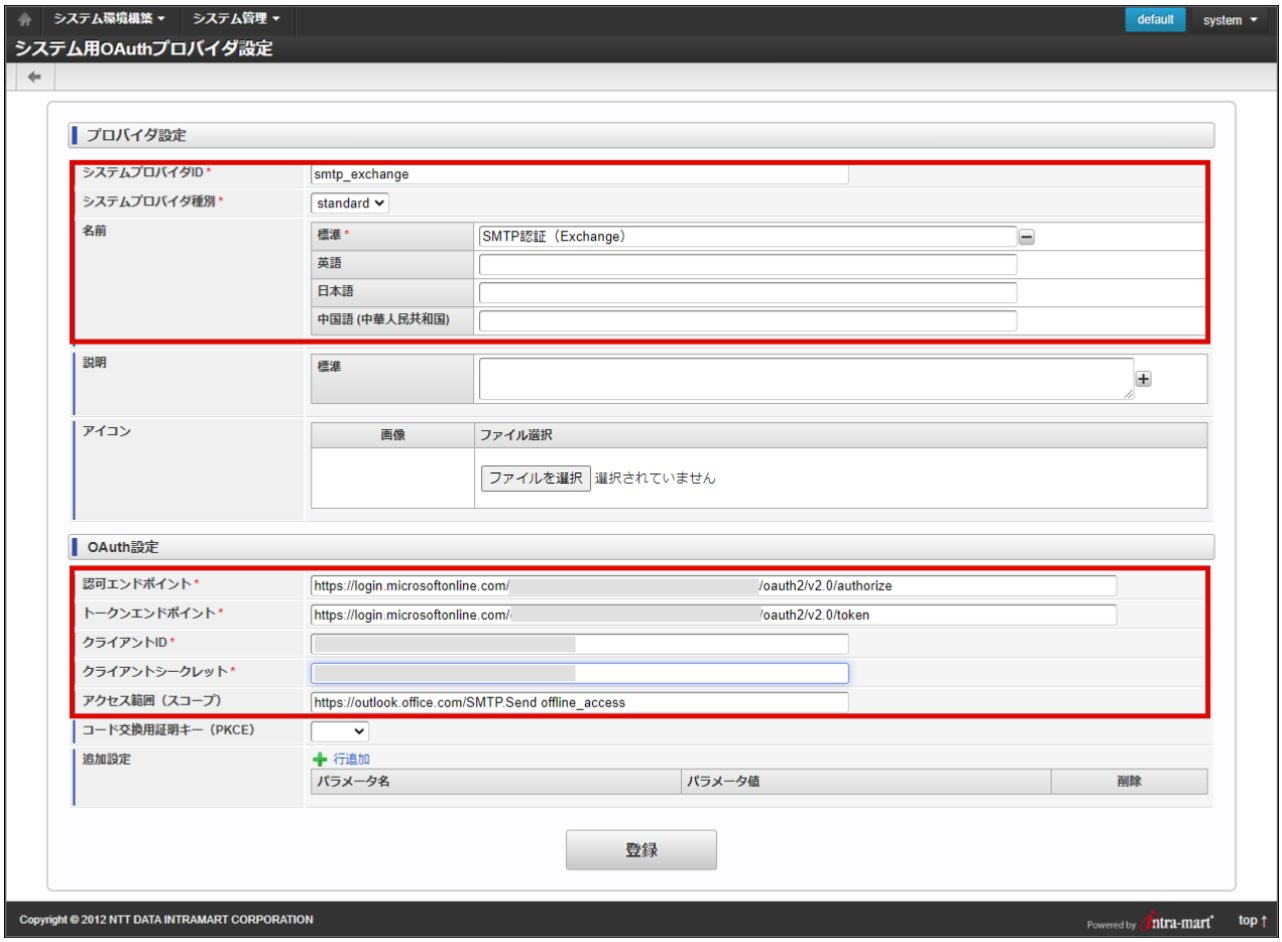
1. 「システム管理者画面」 にログインし、「システム管理」→「システム用OAuthプロバイダ設定」に遷移します。



2. 「新規作成」をクリックして 登録画面 に遷移します。



3. 作成したアプリケーション情報を設定して、「登録」をクリックします。



システムプロバイダID	任意のIDを設定します。
システムプロバイダ種別	*standard* を設定します。
名前	任意の名前を設定します。
認可エンドポイント	Microsoft Azure で作成したアプリケーションのOAuth 2.0 承認エンドポイント (v2)を設定します。
トークンエンドポイント	Microsoft Azure で作成したアプリケーションのOAuth 2.0 トークン エンドポイント (v2)を設定します。
クライアントID	Microsoft Azure で作成したアプリケーションのアプリケーションIDを設定します。
クライアントシークレット	Microsoft Azure で作成したアプリケーションの認証情報のクライアントシークレット値を設定します。
アクセス範囲	SMTP AUTHスコープと *offline_access* を半角スペース区切りで設定します。 詳しくは アクセス許可のスコープの文字列 を参照してください。

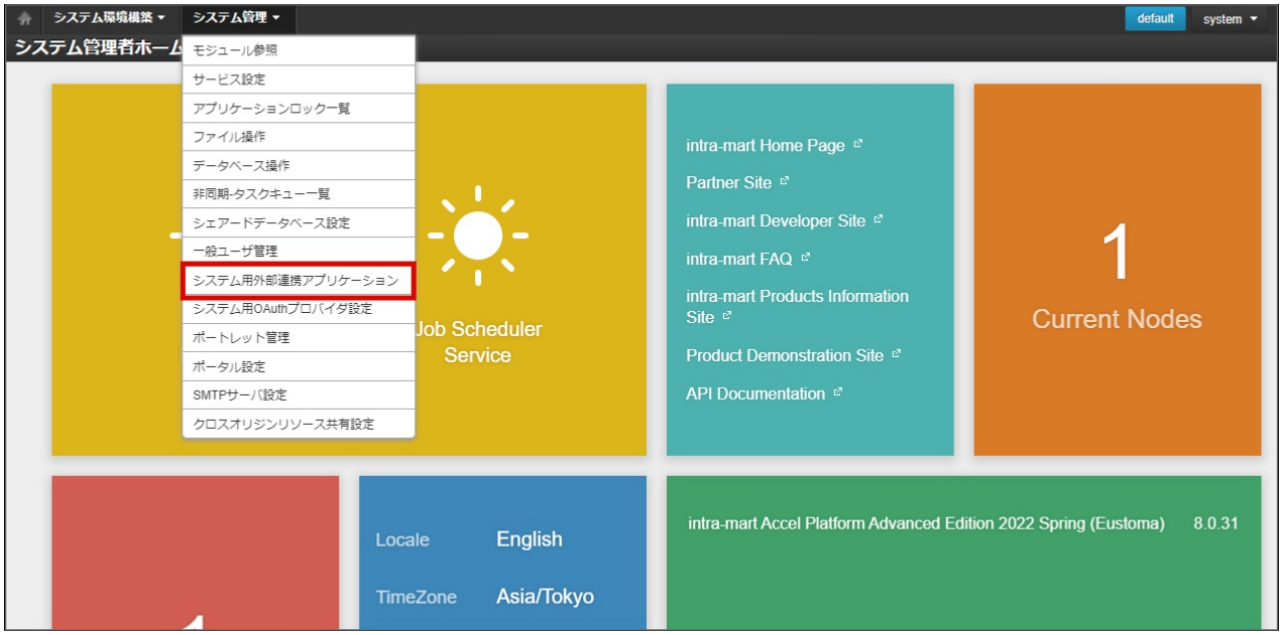
4. 以上でプロバイダ情報の設定が完了です。

> **コラム**
>
> OAuth2.0の仕様や用語については、以下を参照してください。リンク先は 2015年12月1日 時点で情報を確認しています。
>
> 「The OAuth 2.0 Authorization Framework」
>
> - https://tools.ietf.org/html/rfc6749 （English）
> - http://openid-foundation-japan.github.io/rfc6749.ja.html （日本語）

連携許可を行う

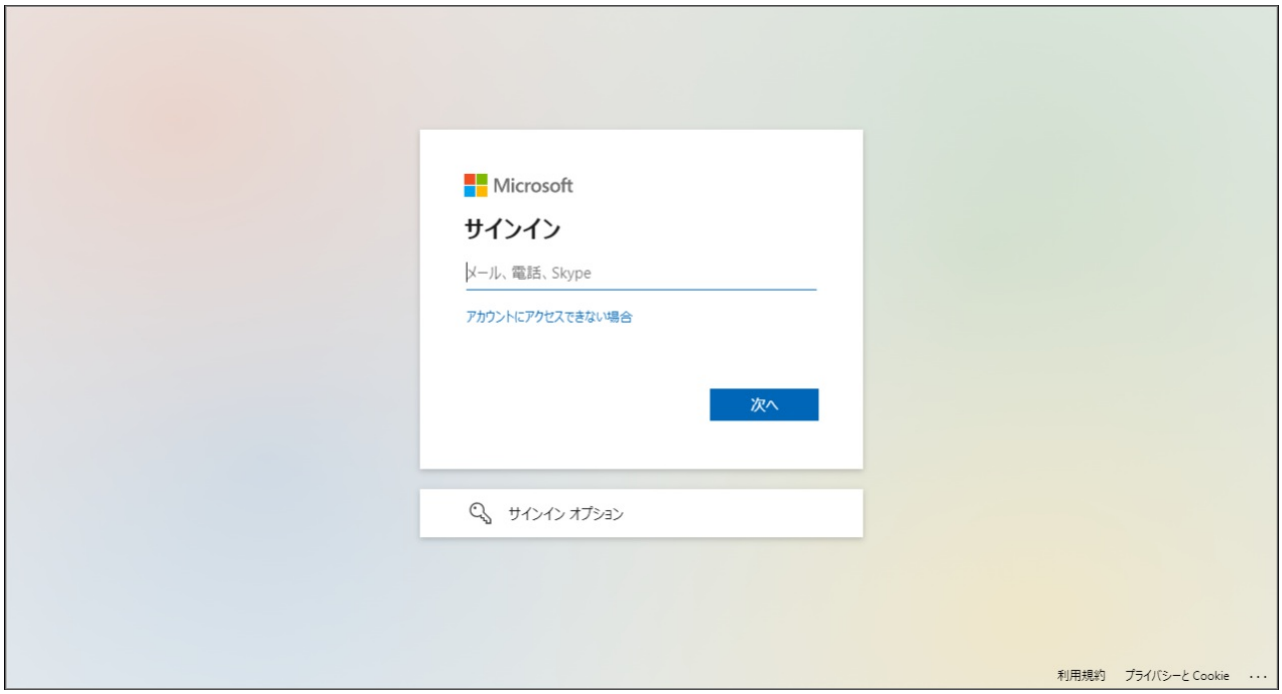
SMTP認証でアクセストークンを利用するには、アクセストークンを発行する必要があります。
「システム用外部連携アプリケーション」より、連携の許可を行うことで、アクセストークンが発行されます。

1. 「システム管理」→「システム用外部連携アプリケーション」に遷移します。



2. 「*プロバイダ情報を設定する*」で作成したプロバイダ情報の「許可する」をクリックします。

3. アプリケーションの認証画面に遷移します。「*送信アクセスの許可を行う*」でアクセス許可されたユーザでサインイン、および許可を行ってください。

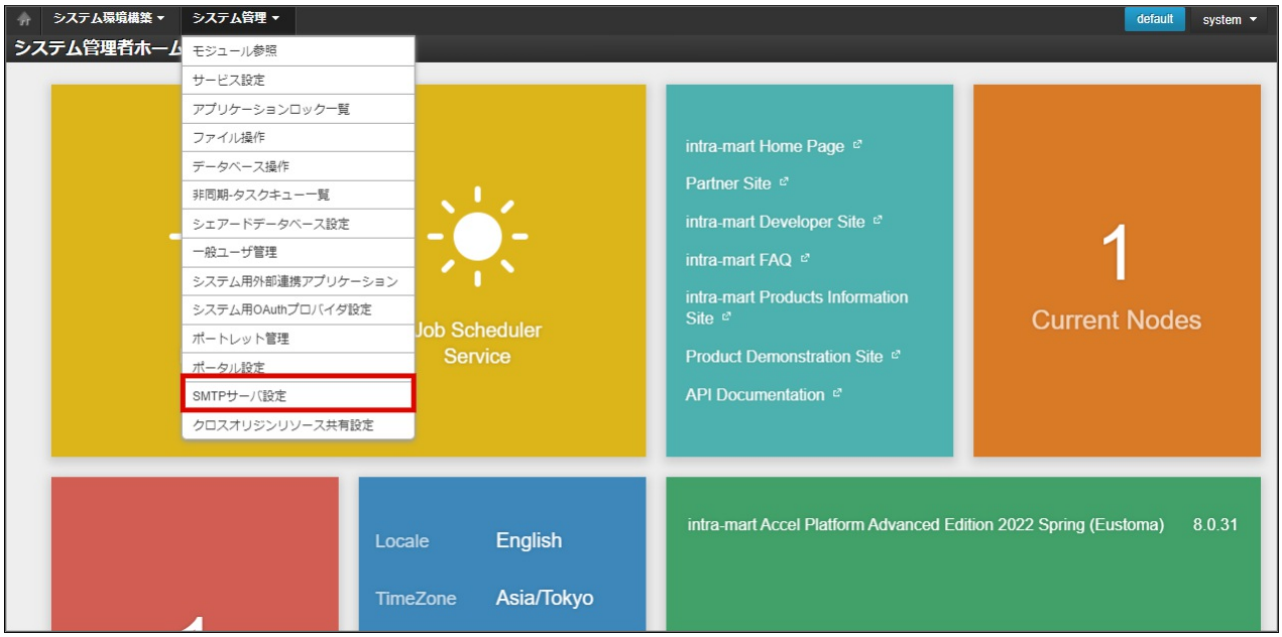


4. 以下の画面が表示されれば連携が完了です。

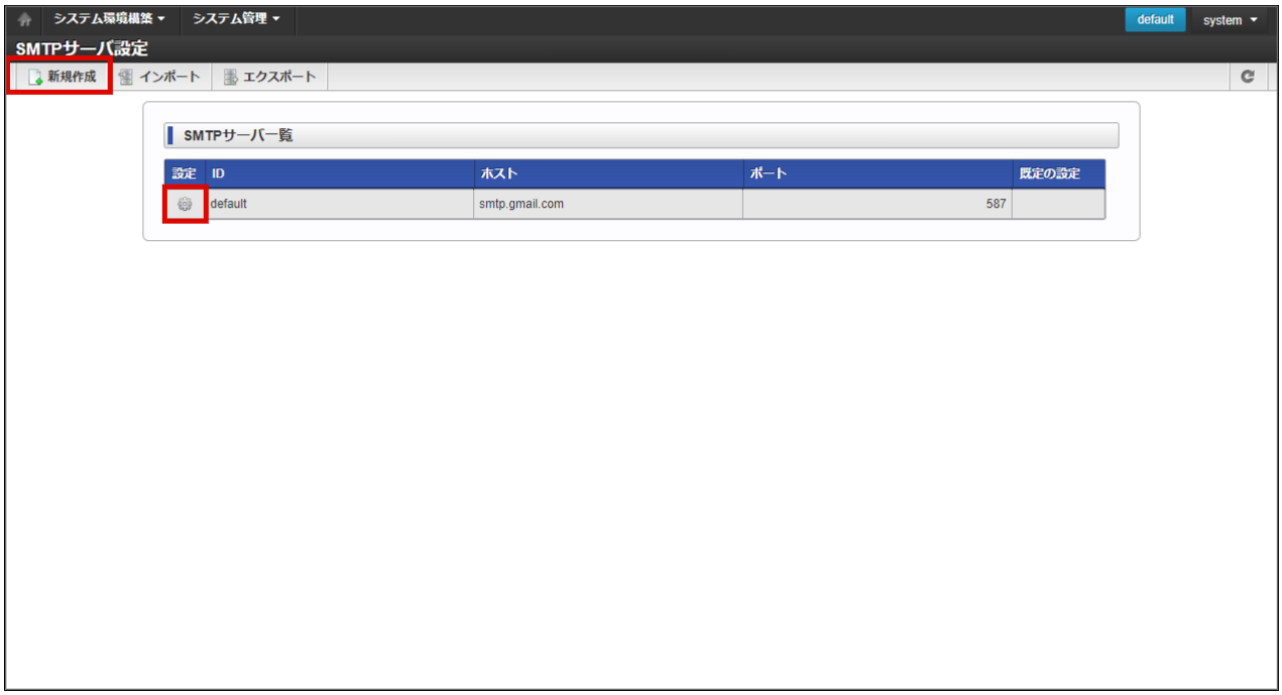
> **コラム**
>
> アクセストークンの期限切れなどで再発行を行う場合は、「解除する」で一度連携を解除すると、「許可する」がクリックできます。

SMTPサーバ設定を行う

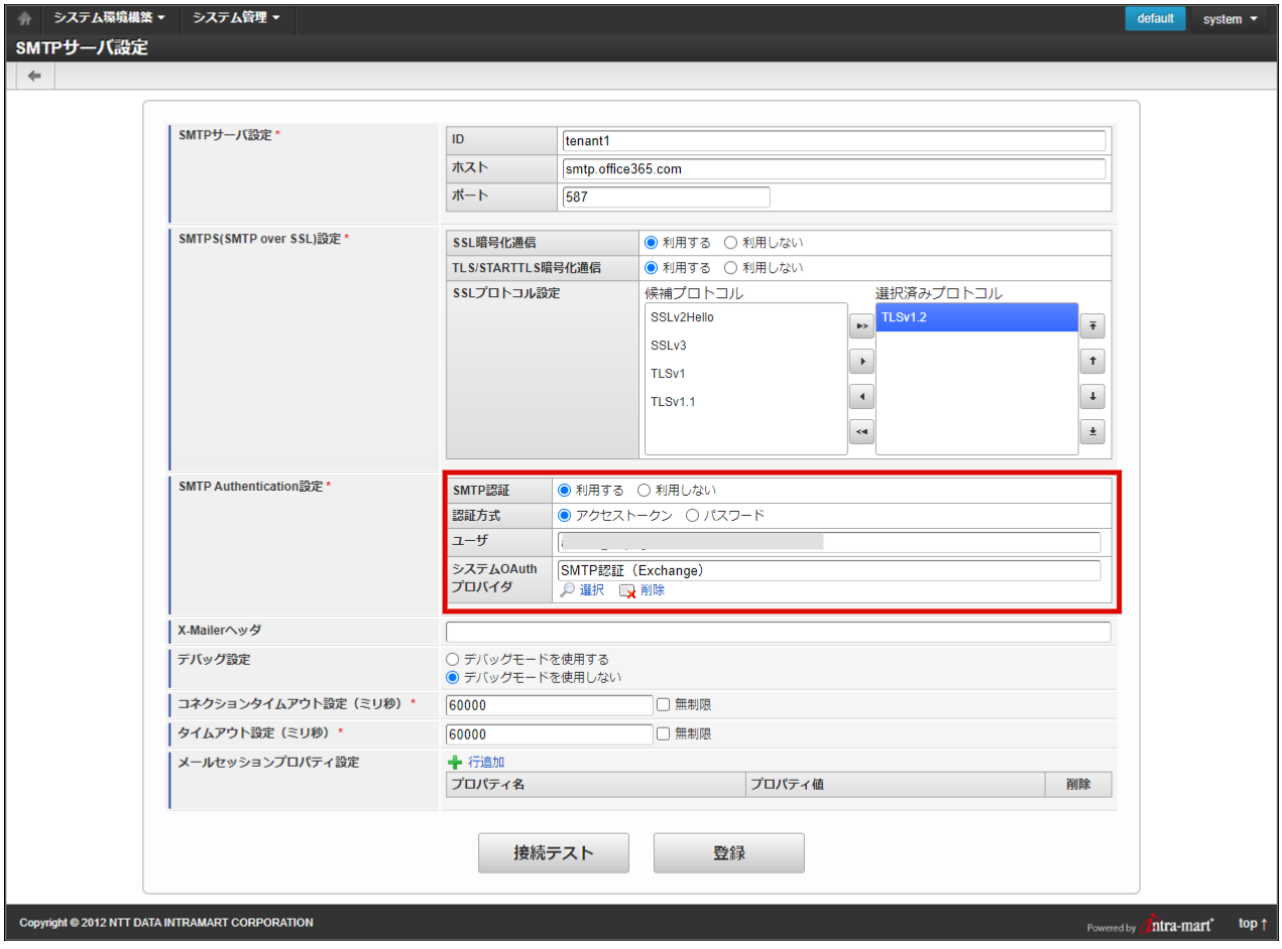
1. 「システム管理」→「SMTPサーバ設定」に遷移します。



2. SMTPサーバを設定する対象のテナントIDの「設定」をクリックします
   対象のテナントIDがない場合は、IDが *default* のSMTPサーバ設定が適用されています。 「新規作成」をクリックして対象のテナントIDで新規作成するか、 *default* の「設定」をクリックして編集画面に遷移してください。

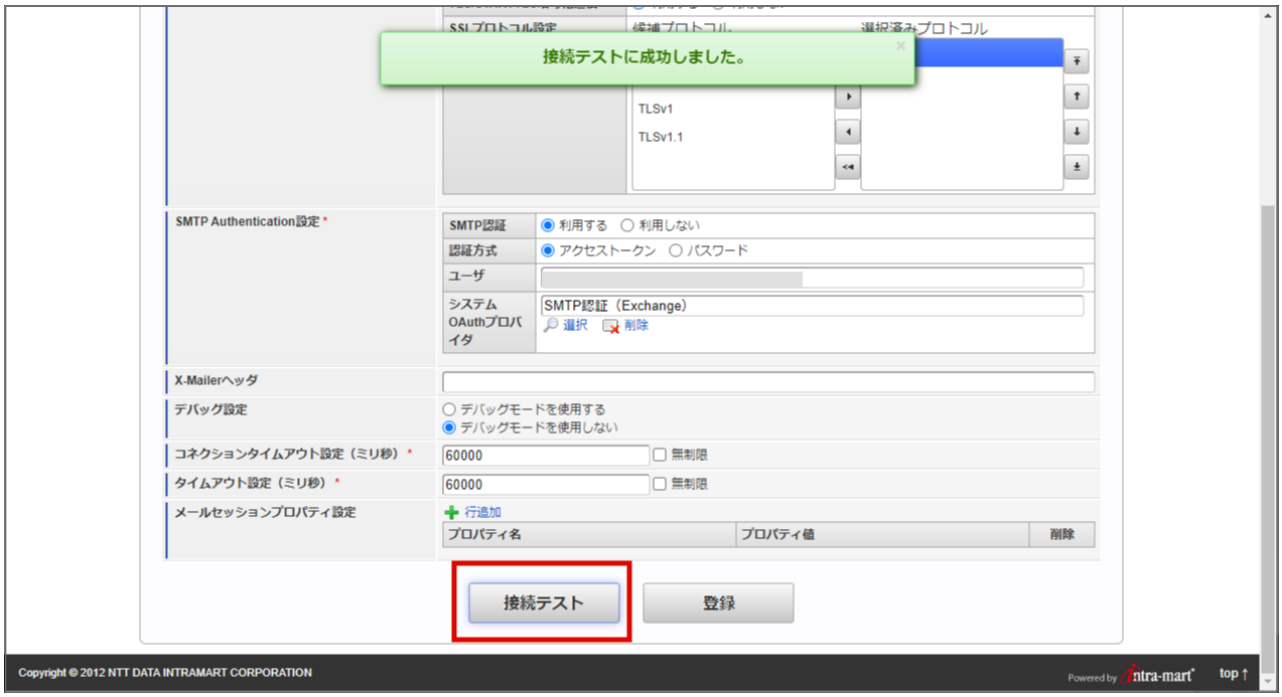


3. 「SMTP Authentication設定」 の設定を以下の通りに変更します。
   - 「SMTP認証」で「利用する」を選択
   - 「認証方式」で「アクセストークン」を選択
   - 「ユーザ」に「*連携許可を行う*」でサインインしたユーザを入力
   - 「システムOAuthプロバイダ」に「*連携許可を行う*」で許可したプロバイダを選択

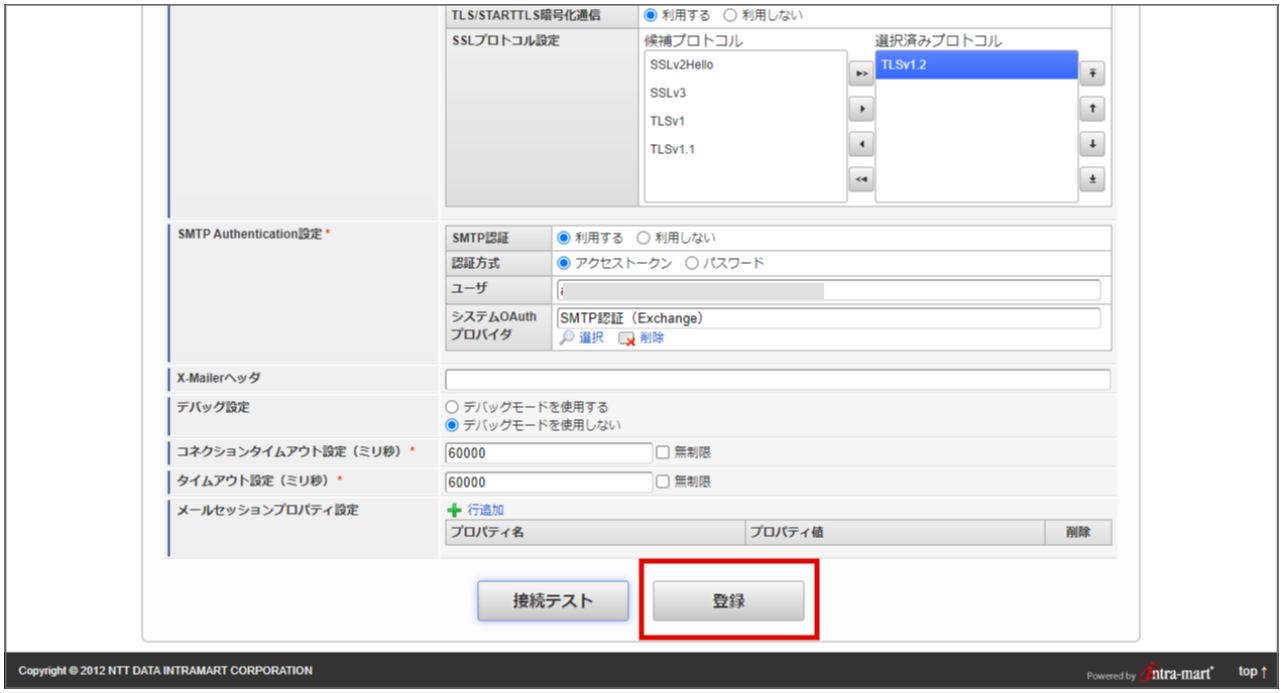


> **コラム**
> **新規作成の場合は、連携先のSMTPサーバ情報を設定する必要があります。**
>
> - Exchange Online の設定値 （リンク先は2022年6月1日 時点で情報を確認しています。）
>   https://docs.microsoft.com/ja-jp/exchange/mail-flow-best-practices/how-to-set-up-a-multifunction-device-or-application-to-send-email-using-microsoft-365-or-office-365#how-to-set-up-smtp-auth-client-submission

4. 「接続テスト」をクリックし、以下の画面が表示されればSMTPサーバとの接続が成功です。



5. 「更新」または「登録」をクリックし、設定を保存して完了です。



IM-LogicDesigner のメール受信タスクで OAuth2.0 アクセストークン を使用する

IM-LogicDesigner のタスクを利用して、メールサーバからメールを受信できます。
IM-LogicDesigner ではIMAP、POP3を利用したメッセージ操作が可能です。
タスクは以下を参照してください。
「IM-LogicDesigner仕様書」 - 「タスク一覧」

アクセストークンの発効

アクセストークンをIMAP、POP3認証で利用するためには、OAuthプロバイダ設定でプロバイダ情報を登録し、外部連携アプリケーションで連携の許可を行って、アクセストークンを発効する必要があります。

- 「設定ファイルリファレンス」 - 「プロバイダ設定」
- 「一般ユーザ操作ガイド」 - 「外部連携アプリケーション」

サービスの設定

メールサーバに Gmail を利用する場合の設定

OAuth認証に必要な関連サービスの準備を行います。
本項の内容は Google Cloud Platform 管理者 向けの作業です。

プロジェクトを作成する

Google Cloud Platform の管理コンソールから OAuth認証 に必要な情報をプロジェクトとして登録します。

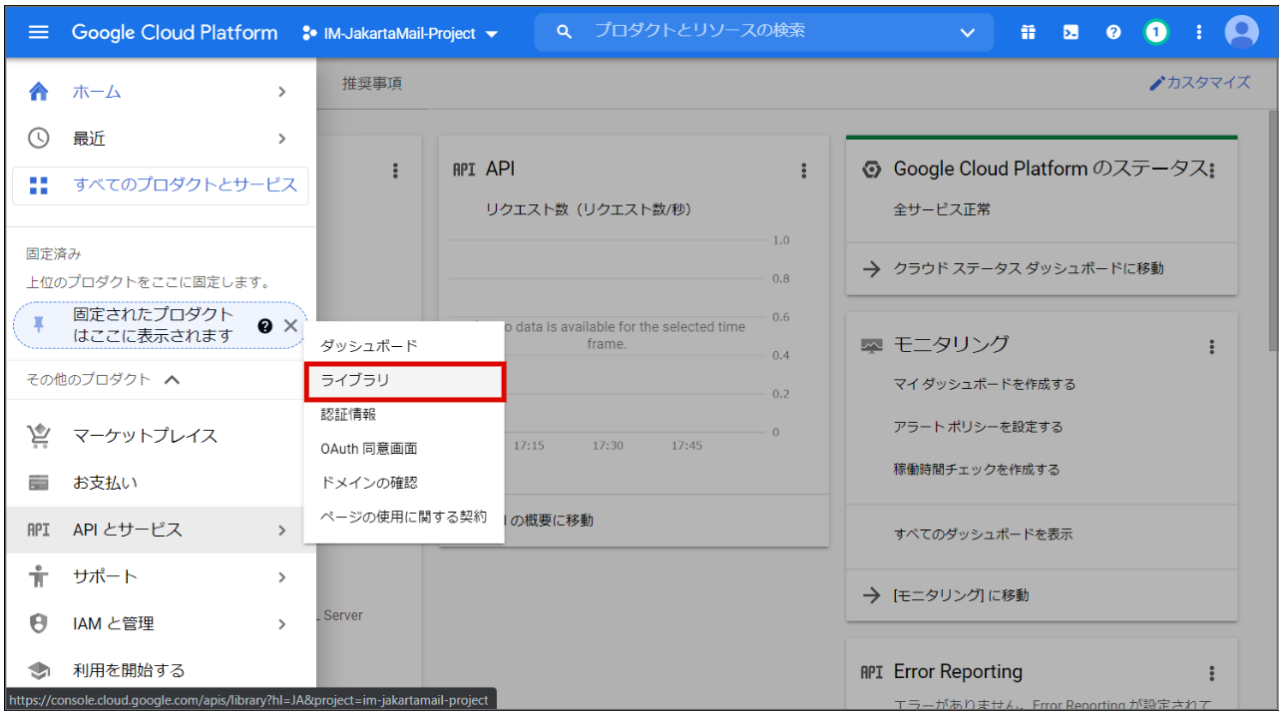
1. 以下のURLから Google Cloud Platform の管理コンソールに **Google Cloud Platform 管理者ユーザ** でサインインします。
   - https://console.cloud.google.com/
2. サイドメニューから「IAM と管理」 - 「プロジェクトを作成」をクリックします。
3. 以下を入力または選択して「作成」をクリックします。
   - プロジェクト名に任意の名称を入力
   - 場所に任意のフォルダを選択

1. 以上でプロジェクトの作成は完了です。

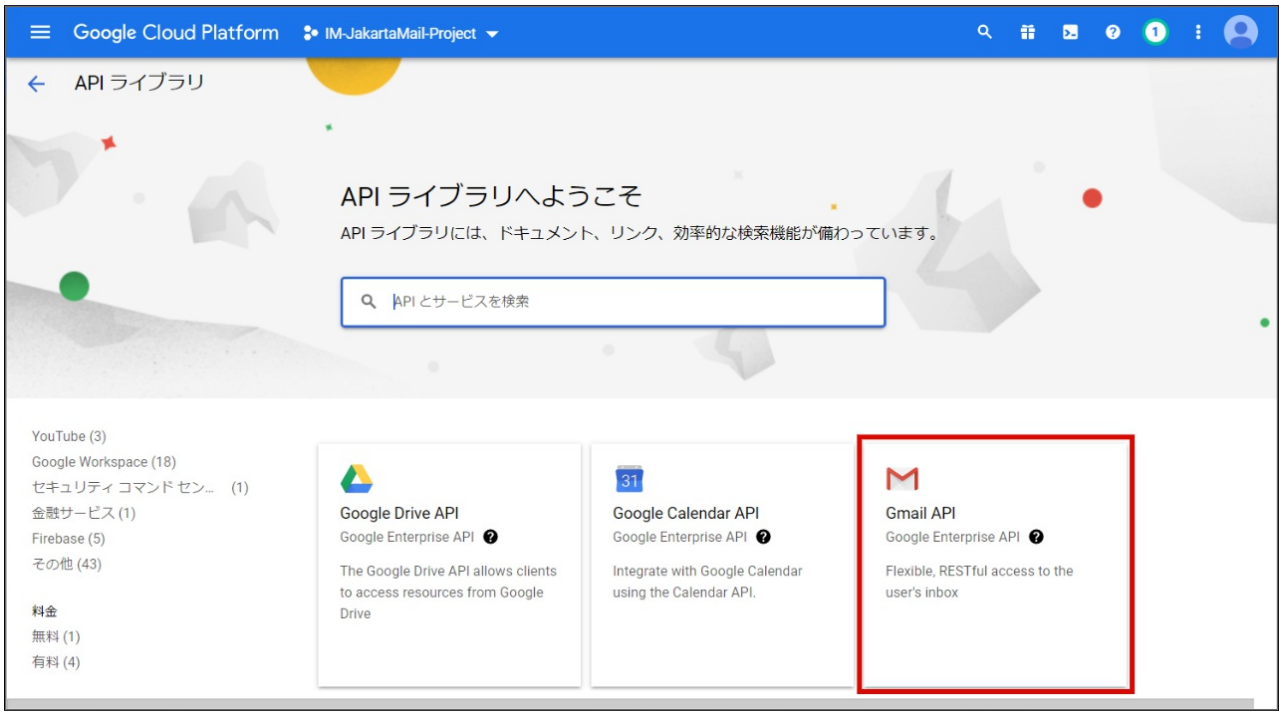
Gmail API を有効にする

作成したプロジェクトで *Gmail API* を利用できるようにします。

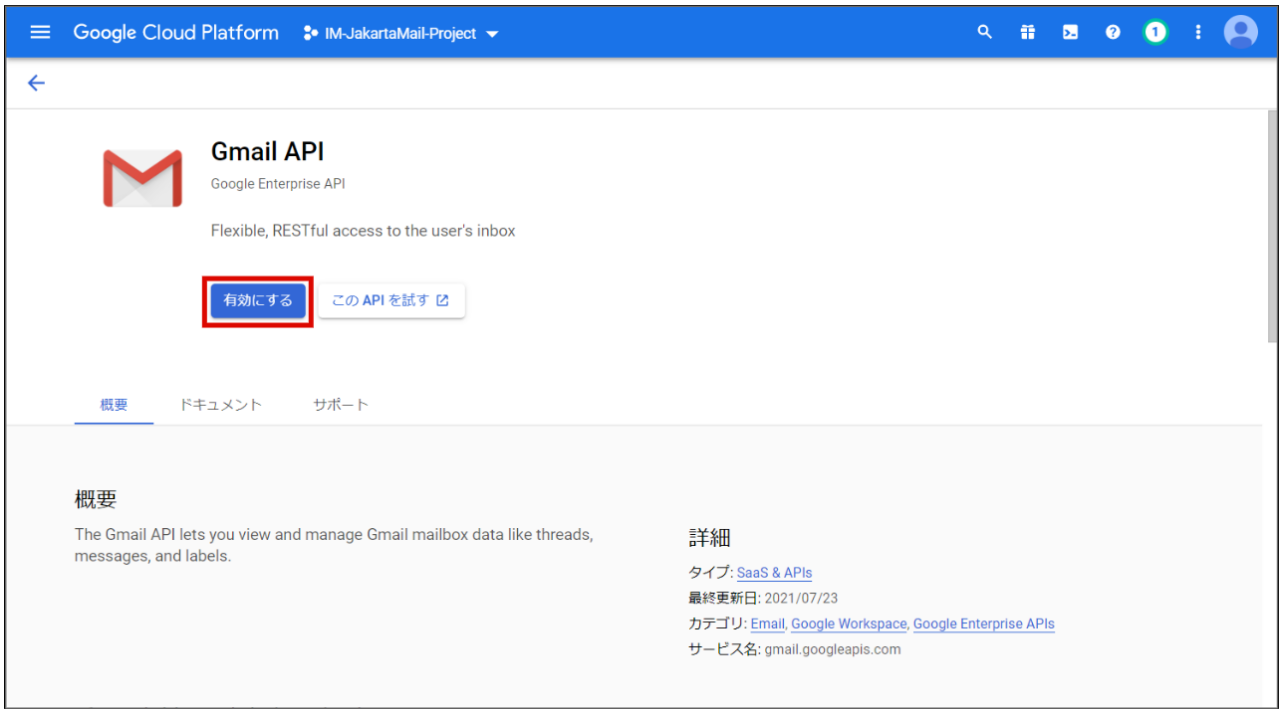
1. 作成したプロジェクトが選択されていることを確認し、サイドメニューから「API とサービス」 - 「ライブラリ」をクリックします。



2. API ライブラリから *Gmail API* をクリックします。



3. 「有効にする」ボタンをクリックし、 *Gmail API* を有効にします。



4. 以上で作成したプロジェクトで *Gmail API* が利用できます。

認証情報を作成する

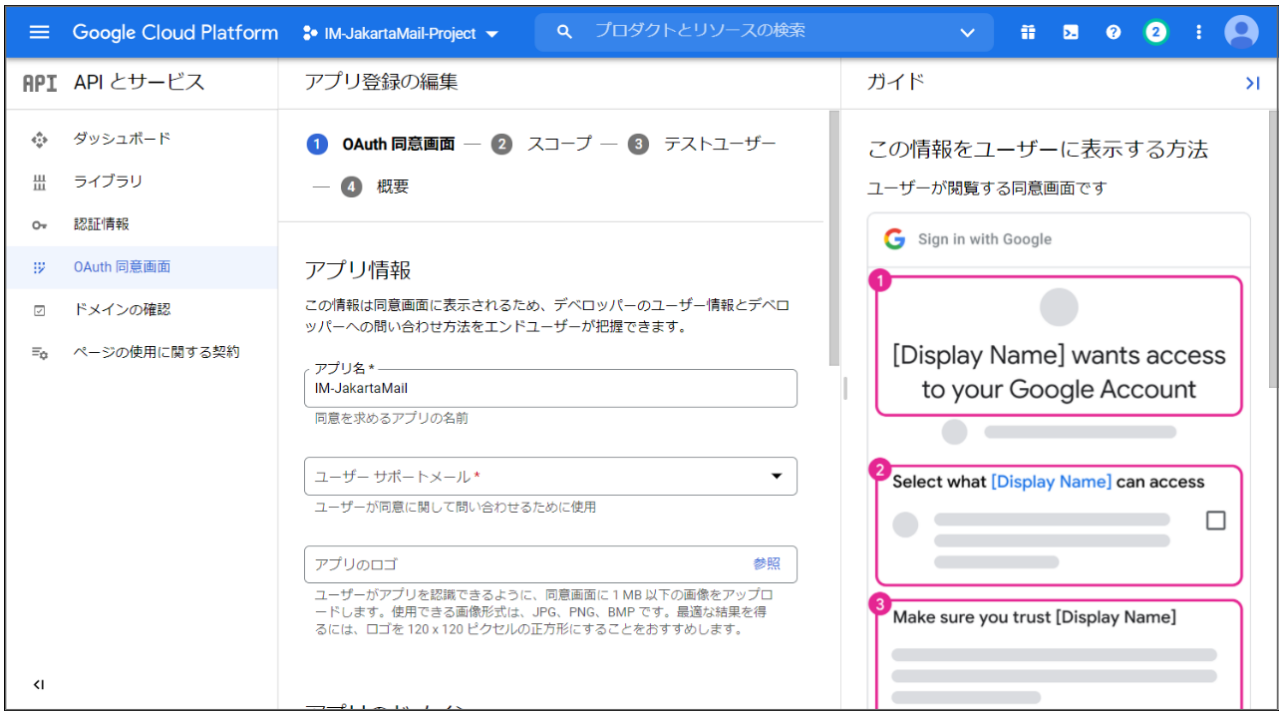
OAuth認証に必要な「OAuth 同意画面」の設定と「OAuth クライアントID」の作成を行います。

1. サイドメニューから「認証情報」をクリックします。

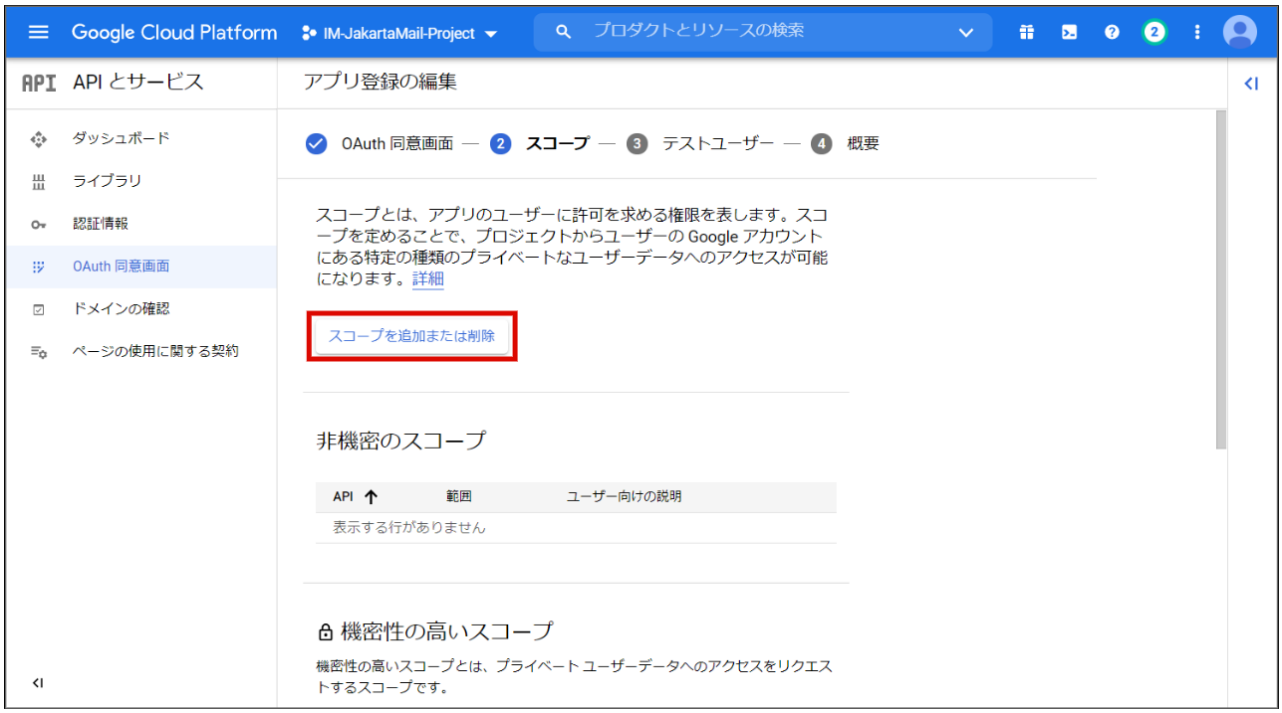
2. 「同意画面を構成」をクリックします。

3. 任意の「User Type」を選択し、「作成」をクリックします。

4. 以下を入力して「保存して次へ」をクリックします。
   - アプリ情報に任意の名称、メールアドレス、ロゴを入力
   - アプリのドメインに任意のドメイン情報を入力
   - デベロッパーの連絡先情報に任意のメールアドレスを入力



5. 「スコープを追加または削除」をクリックし、 *Gmail API* をスコープに追加します。
利用用途にあわせてスコープを選択してください。



6. スコープを追加したら、「保存して次へ」をクリックします。

7. 「ADD USERS」をクリックし接続テストを行うユーザを追加し、「保存して次へ」をクリックします。

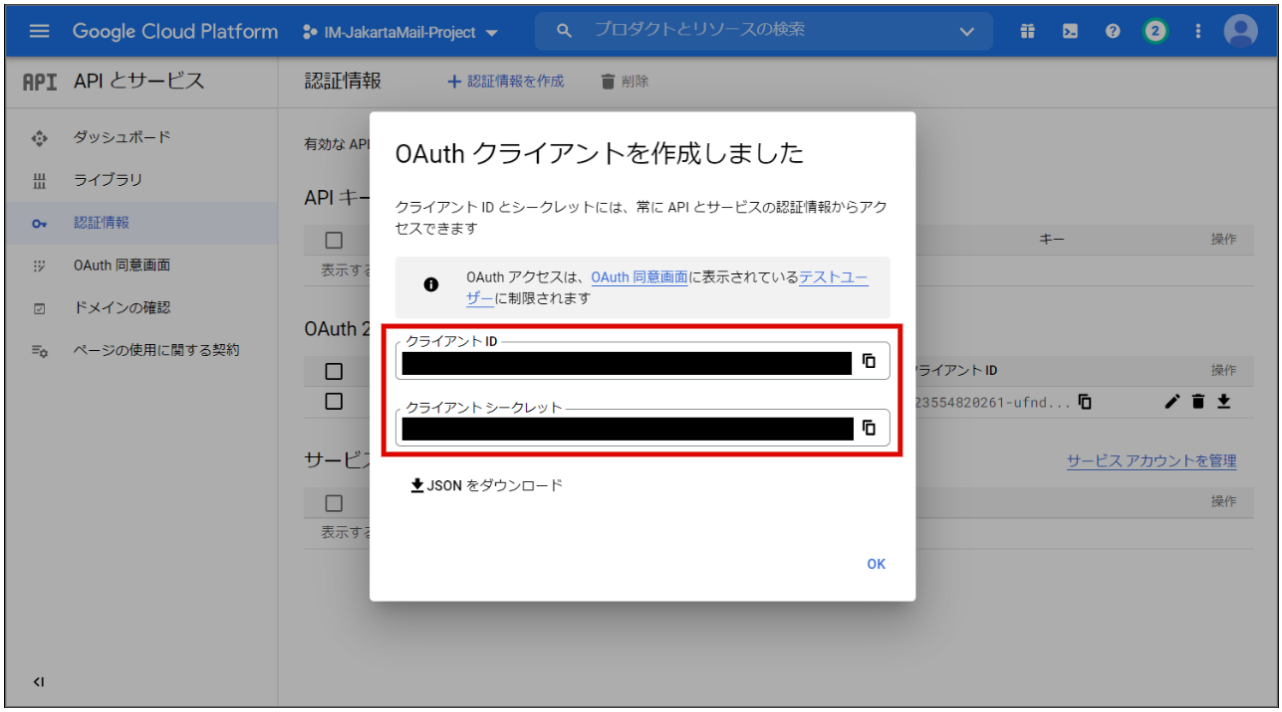
8. 以上で、OAuth 同意画面の設定は完了です。次に OAuth クライアントID を作成します。
9. サイドメニューから「認証情報」をクリックします。
10. 「認証情報を作成」をクリックし、「OAuth クライアント ID」を選択します。

11. 以下を入力または選択して「作成」をクリックします。
    - アプリケーションの種類に「ウェブアプリケーション」を選択
    - 名前に任意の名称を入力
    - 承認済みの JavaScript 生成元に intra-mart Accel Platform のベースURL を入力
    - 承認済みのリダイレクト URI に **ベースURL + /oauth/redirect** を入力

12. 以上で認証情報の作成が完了です。

    以下の内容は intra-mart Accel Platform システム管理者 が環境構築を行う際に利用します。 「JSONをダウンロード」で情報をダウンロードできます。

    - クライアントID （ *client-id* として利用します）
    - クライアントシークレット （ *client-secret* として利用します）



メールサーバに Exchange Online を利用する場合の設定

OAuth認証に必要な関連サービスの準備を行います。
本項の内容は Microsoft Azure 管理者 向けの作業です。

アプリケーションを登録する

Microsoft Azure の管理ポータルから OAuth認証 に必要な情報をアプリケーションとして登録します。

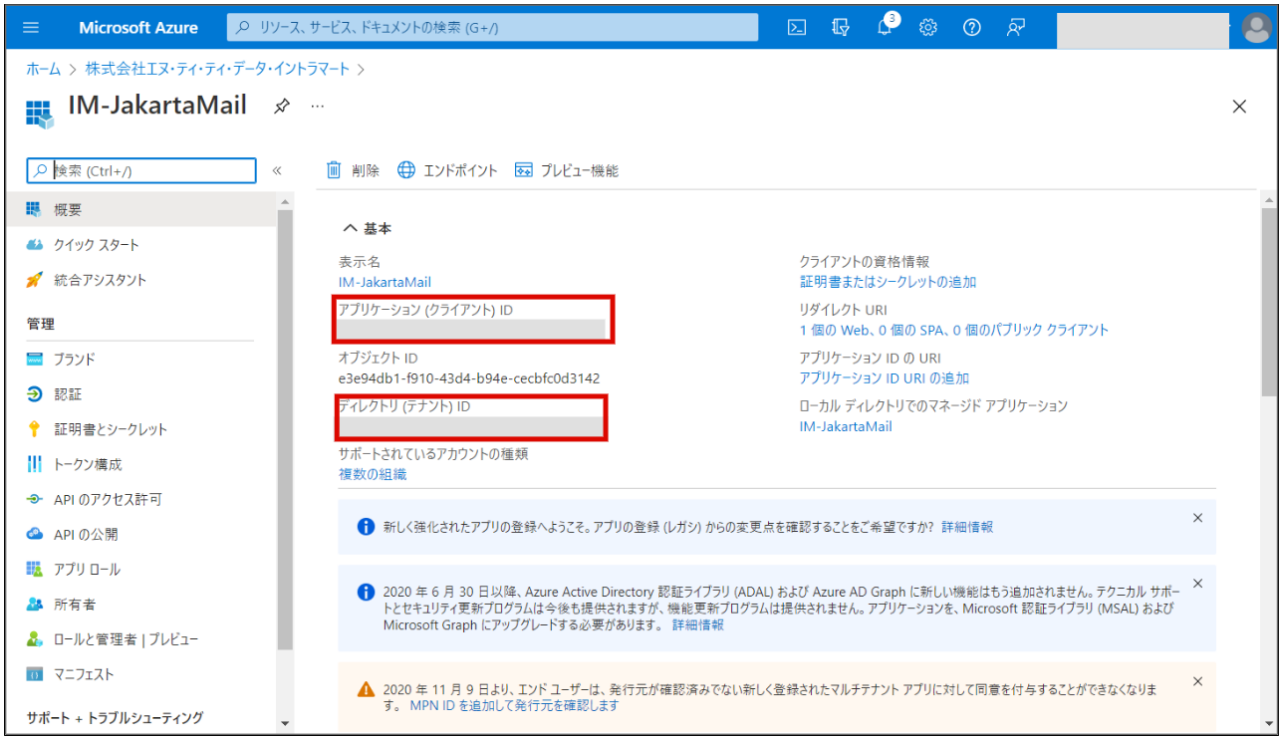
1. 以下のURLから Microsoft Azure の管理ポータルに **Microsoft Azure 管理者ユーザ** でサインインします。
   - https://portal.azure.com/
2. サイドメニューから「Microsoft Entra ID」をクリックします。

3. 現在のテナントが「 OAuth認証を利用する組織のテナント」ではない場合は「テナントの切り替え」を行います。
4. 「概要」のサイドメニュー「管理」の「アプリの登録」をクリックします。
5. 「新規登録」をクリックします。
6. 以下を入力または選択して「登録」をクリックします。
   - 名前に任意の名称を入力
   - サポートされているアカウントの種類に「任意の組織ディレクトリ内のアカウント（任意の Microsoft Entra ID テナント - マルチテナント ）」を選択
   - リダイレクト URI に「Web」を選択し、 intra-mart Accel Platform の**ベースURL + /oauth/redirect**を入力

7. 以上でアプリケーションの登録は完了です。

   以下の内容は intra-mart Accel Platform システム管理者 がプロバイダ情報を設定する際に利用します。

   - アプリケーションID （ *client-id* として利用します）
   - ディレクトリID （ *authz-end-point* 、 *token-end-point* のテナントとして利用します）

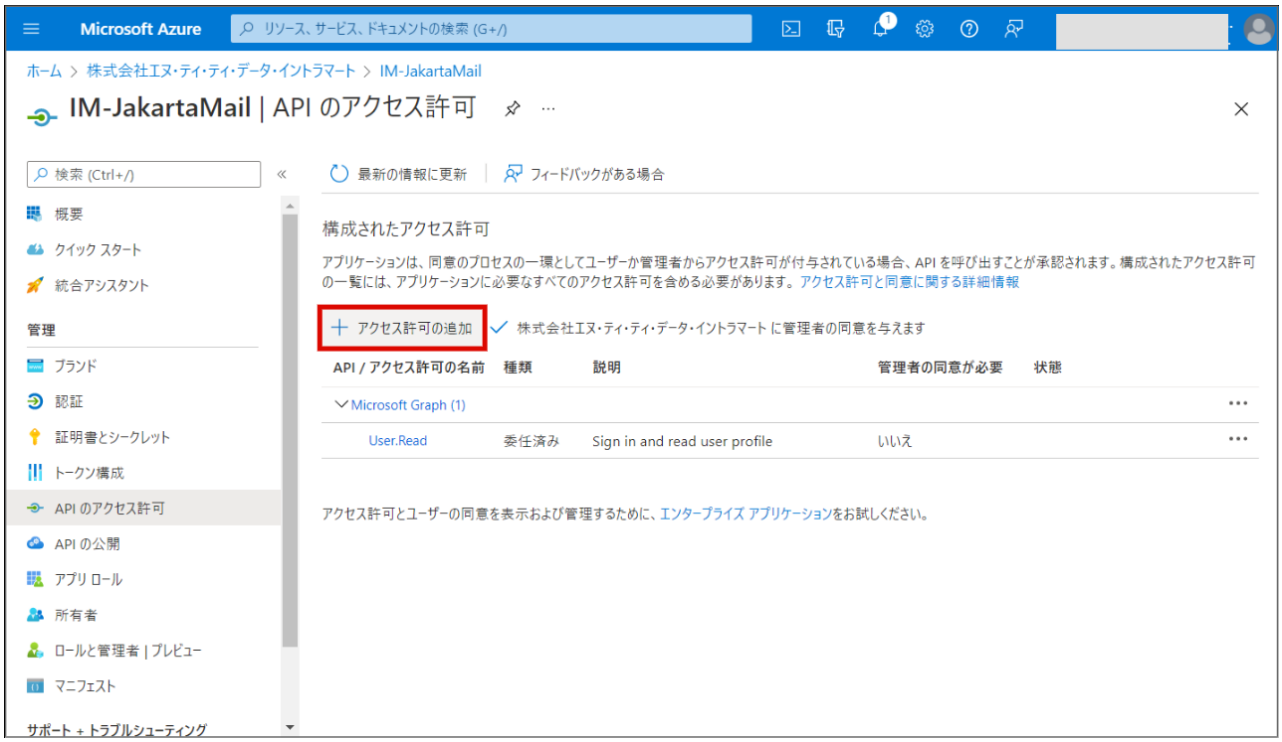


アプリケーションを設定する

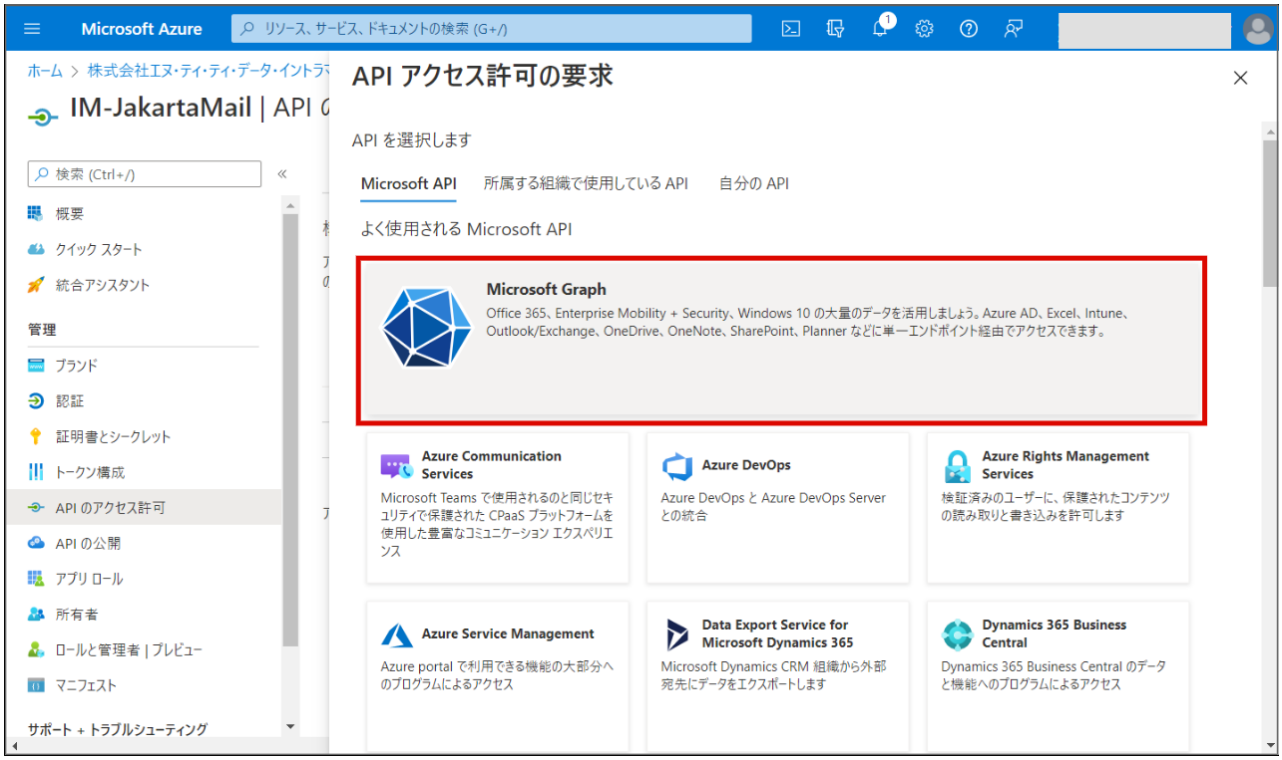
Microsoft Azure の管理ポータルから登録したアプリケーションの構成を変更します。

1. 先程登録したアプリの「管理」の「APIのアクセス許可」をクリックします。

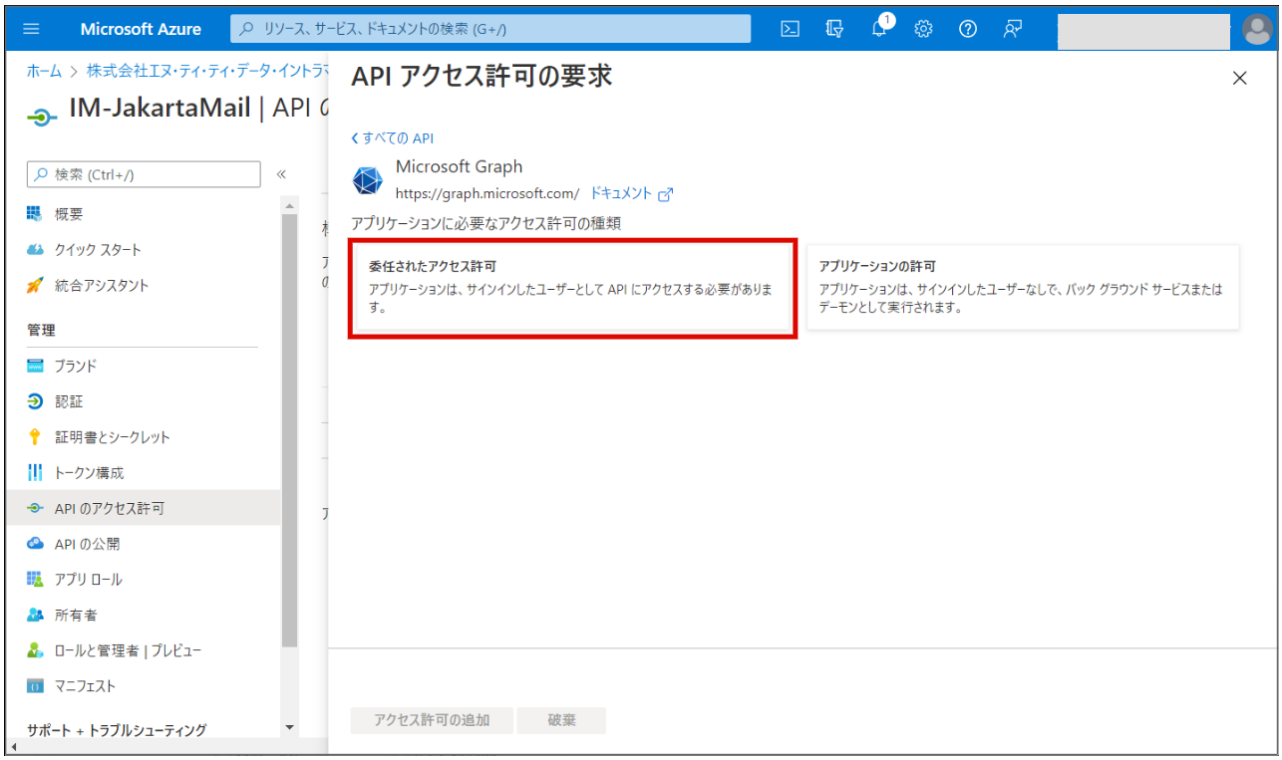
2. 「アクセス許可の追加」をクリックします。



3. 「 Microsoft Graph 」をクリックします。



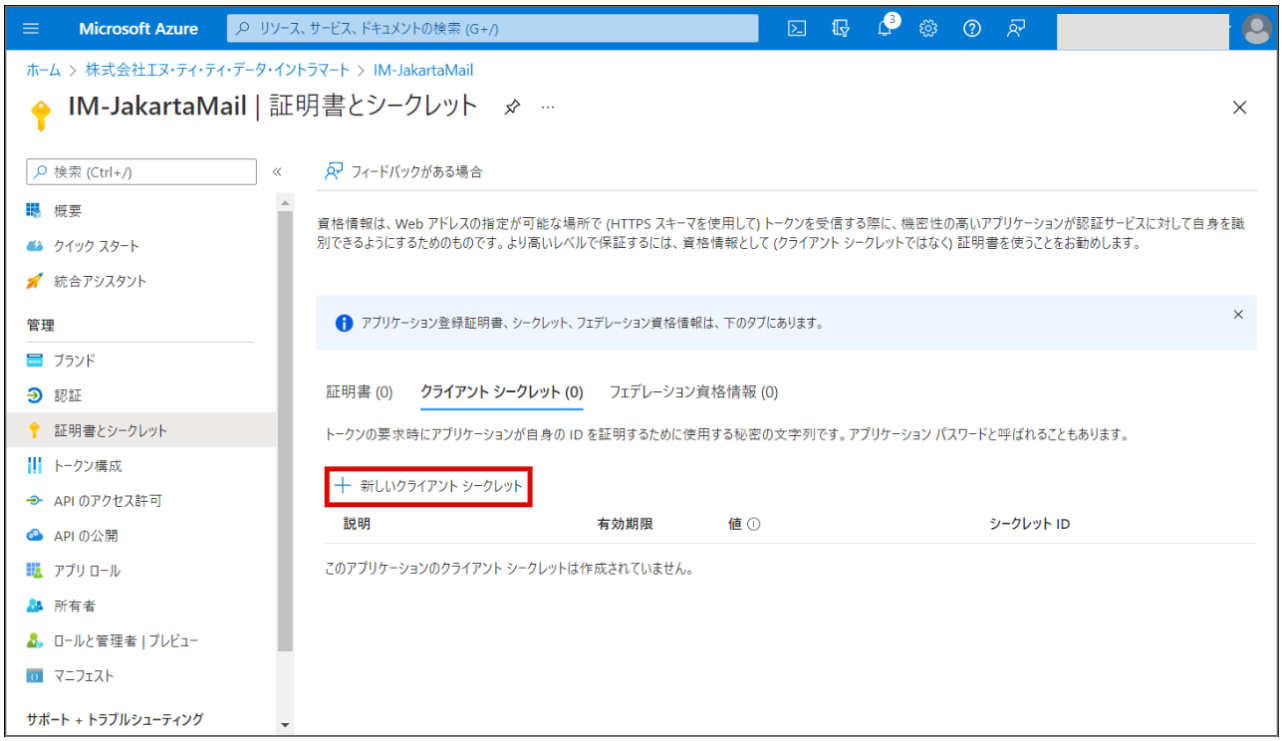
4. 「アプリケーションに必要なアクセス許可の種類」の「委任されたアクセス許可」をクリックします。



5. *offline_access* と、利用用途にあわせ *IMAP.AccessAsUser.All* または *POP.AccessAsUser.All* を選択し、「アクセス許可の追加」をクリックします。

> **コラム**
>
> 「APIアクセス」の「アクセスの有効化」にて「委任されたアクセス許可」におけるメール操作の許可設定についての詳細はMicrosoft社 の以下のドキュメントを参照してください。
>
> - Microsoft Graph permissions reference :
>   - https://docs.microsoft.com/en-us/graph/permissions-reference （English）
>   - https://docs.microsoft.com/ja-jp/graph/permissions-reference （日本語）
>   - https://docs.microsoft.com/zh-cn/graph/permissions-reference （中文）

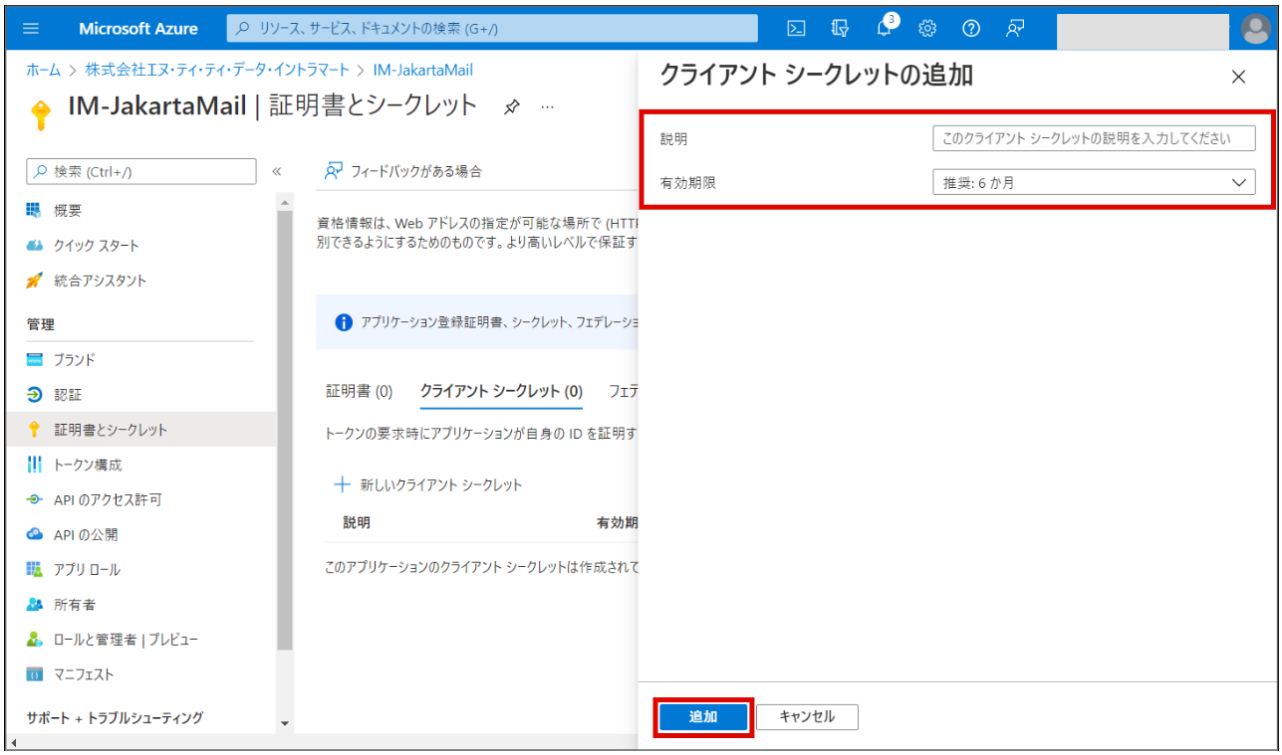
6. 「管理」の「証明書とシークレット」をクリックします。
7. 「クライアント シークレット」の「新しいクライアント シークレット」をクリックします。



8. intra-mart Accel Platform からアクセスする際に必要なキーを生成します。

   以下を入力または選択して「追加」をクリックします。

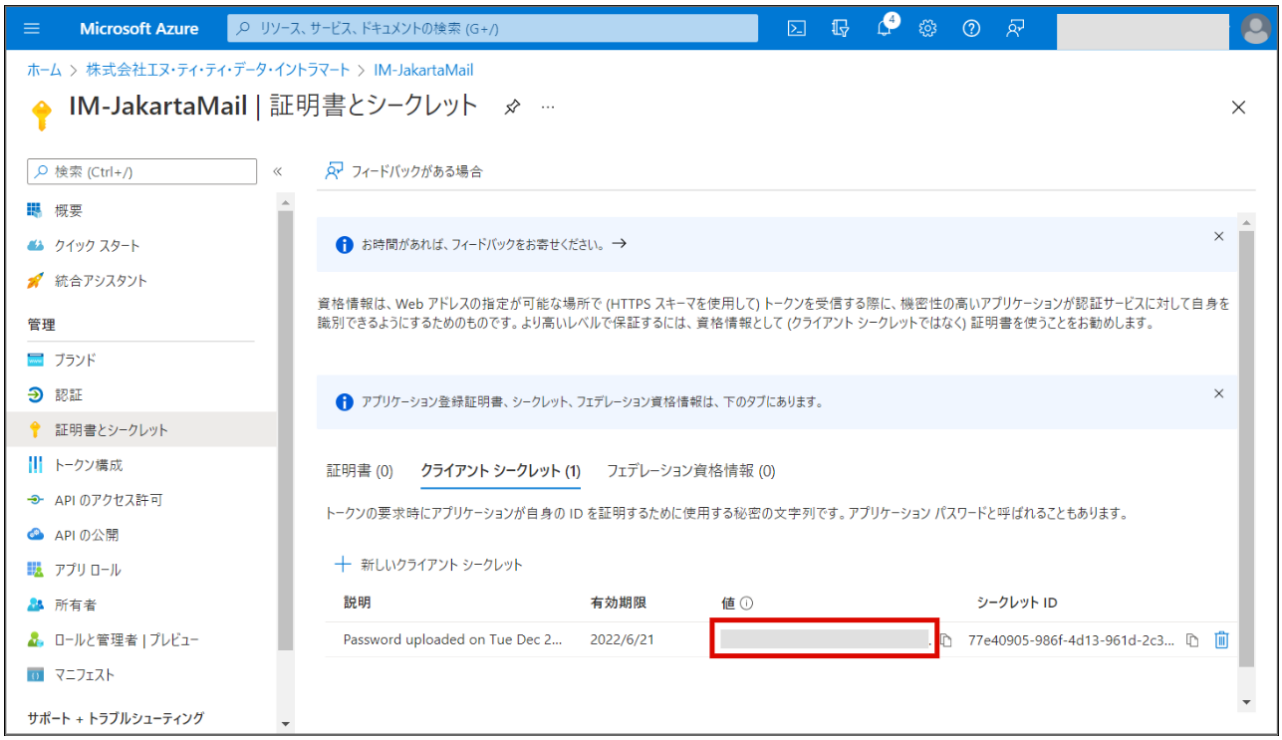
   - 説明に任意のキーの説明を入力
   - 有効期間に任意のキーの有効期限を選択



以下の内容は intra-mart Accel Platform システム管理者 がプロバイダ情報を設定する際に利用します。

- クライアントシークレット値（ *client-secret* として利用します）

> **注意**
> キーは設定の保存後に一度のみ表示されます。移動前にキーの表示内容を退避させてください。



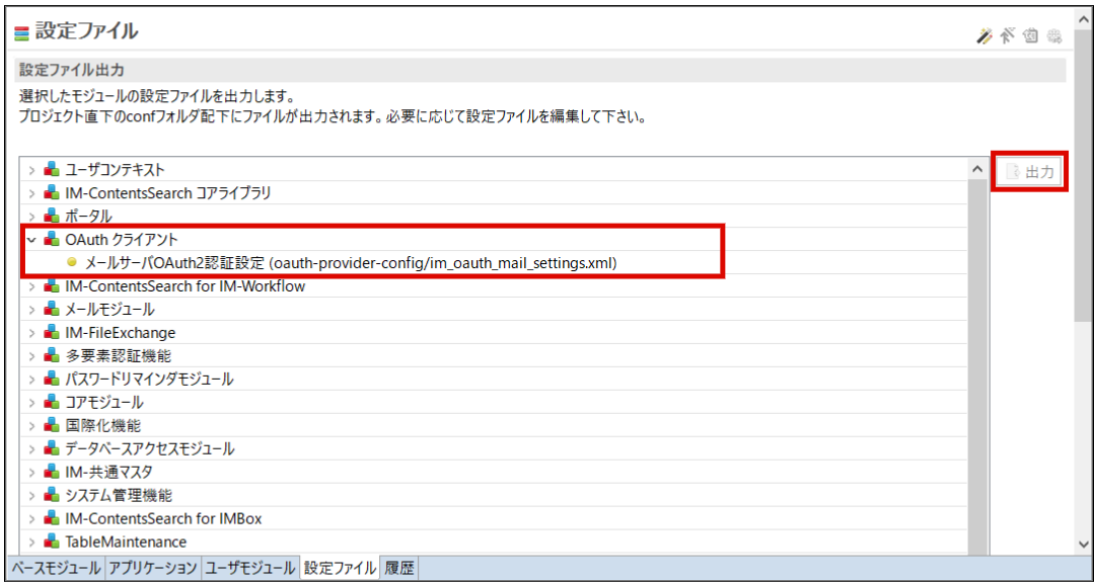
> **コラム**
>
> 有効期限が切れた場合は、上記の手順でキーを再発行する必要があります。

9. 以上でアプリケーションの設定は完了です。

プロバイダ情報を設定する

1. IM-Juggling プロジェクトの「追加機能」 -「外部システム連携機能」 -「 OAuth クライアント 」モジュールを選択します。

2. 「設定ファイル」タブを開き、「 OAuth クライアント 」 -「メールサーバOAuth2認証設定」を選択して「出力」ボタンをクリックします。



3. 「メールサーバOAuth2認証設定」を利用するサービスにあわせて設定します。
   - Gmail API を利用する場合
     Gmail API を利用する場合は設定ファイルの以下の部分のコメントアウトを外し、各項目の設定を行って保存します。

```
<oauth-provider id="Please_input_your_ID">
  <provider-type>extra</provider-type>
  <name message-cd="CAP.Z.IWP.OAUTHCLIENT.GMAIL.XOAUTH.NAME">Gmail API</name>
  <description message-cd="CAP.Z.IWP.OAUTHCLIENT.GMAIL.XOAUTH.DESCRIPTION">It is allow the use of the Gmail API.</description>
  <icon-path>im_oauth_client/images/gmail.png</icon-path>
  <oauth-config>
    <authz-end-point>https://accounts.google.com/o/oauth2/auth</authz-end-point>
    <token-end-point>https://oauth2.googleapis.com/token</token-end-point>
    <client-id>[Please input your application's Client ID]</client-id>
    <client-secret>[Please input your application's Client Secret]</client-secret>
    <scope>https://mail.google.com/</scope>
  </oauth-config>
  <extra-config>
    <parameter name="access_type">offline</parameter>
  </extra-config>
</oauth-provider>
```

oauth-provider@id	任意のIDを設定します。
client-id	Google Cloud Platform で作成したプロジェクトの認証情報のクライアントIDを設定します。
client-secret	Google Cloud Platform で作成したプロジェクトの認証情報のクライアントシークレットを設定します。
scope	利用用途にあわせて必要に応じて変更します。 詳しくは Gmail scopes を参照してください。

   - Exchange Online を利用する場合
     Exchange Online を利用する場合は設定ファイルの以下の部分のコメントアウトを外し、各項目の設定を行って保存します。

```
<oauth-provider id="Please_input_your_ID">
  <provider-type>standard</provider-type>
  <name message-cd="CAP.Z.IWP.OAUTHCLIENT.EXCHANGE.NAME">Exchange Online Provider</name>
  <description message-cd="CAP.Z.IWP.OAUTHCLIENT.EXCHANGE.DESCRIPTION">It is allow the use of Exchange Online.</description>
  <icon-path>im_oauth_client/images/microsoft_exchange_48.png</icon-path>
  <oauth-config>
   <authz-end-point>https://login.microsoftonline.com/[Please input your tenant]/oauth2/v2.0/authorize</authz-end-point>
   <token-end-point>https://login.microsoftonline.com/[Please input your tenant]/oauth2/v2.0/token</token-end-point>
   <client-id>[Please input your application's Client ID]</client-id>
   <client-secret>[Please input your application's Client Secret]</client-secret>
   <scope>offline_access [Please input your application's scope. See : https://docs.microsoft.com/ja-jp/exchange/client-developer/legacy-protocols/how-to-authenticate-an-imap-pop-smtp-application-by-using-oauth#get-an-access-token]</scope>
  </oauth-config>
 </oauth-provider>
```

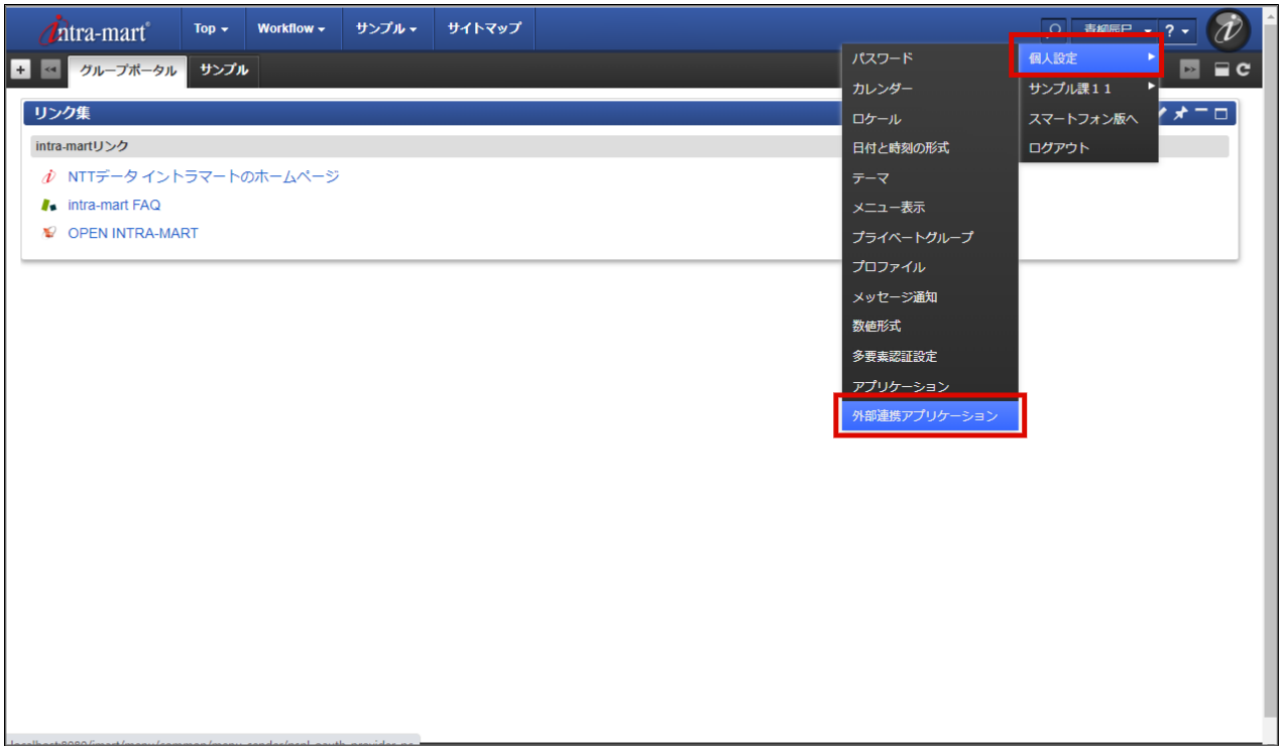
oauth-provider@id	任意のIDを設定します。
authz-end-point	Microsoft Azure で作成したアプリケーションのディレクトリIDを設定します。
token-end-point	Microsoft Azure で作成したアプリケーションのディレクトリIDを設定します。
client-id	Microsoft Azure で作成したアプリケーションのアプリケーションIDを設定します。
client-secret	Microsoft Azure で作成したアプリケーションの認証情報のクライアントシークレット値を設定します。
scope	利用用途にあわせて設定します。 詳しくは アクセス許可のスコープの文字列 を参照してください。

セットアップ時の設定は以上です。

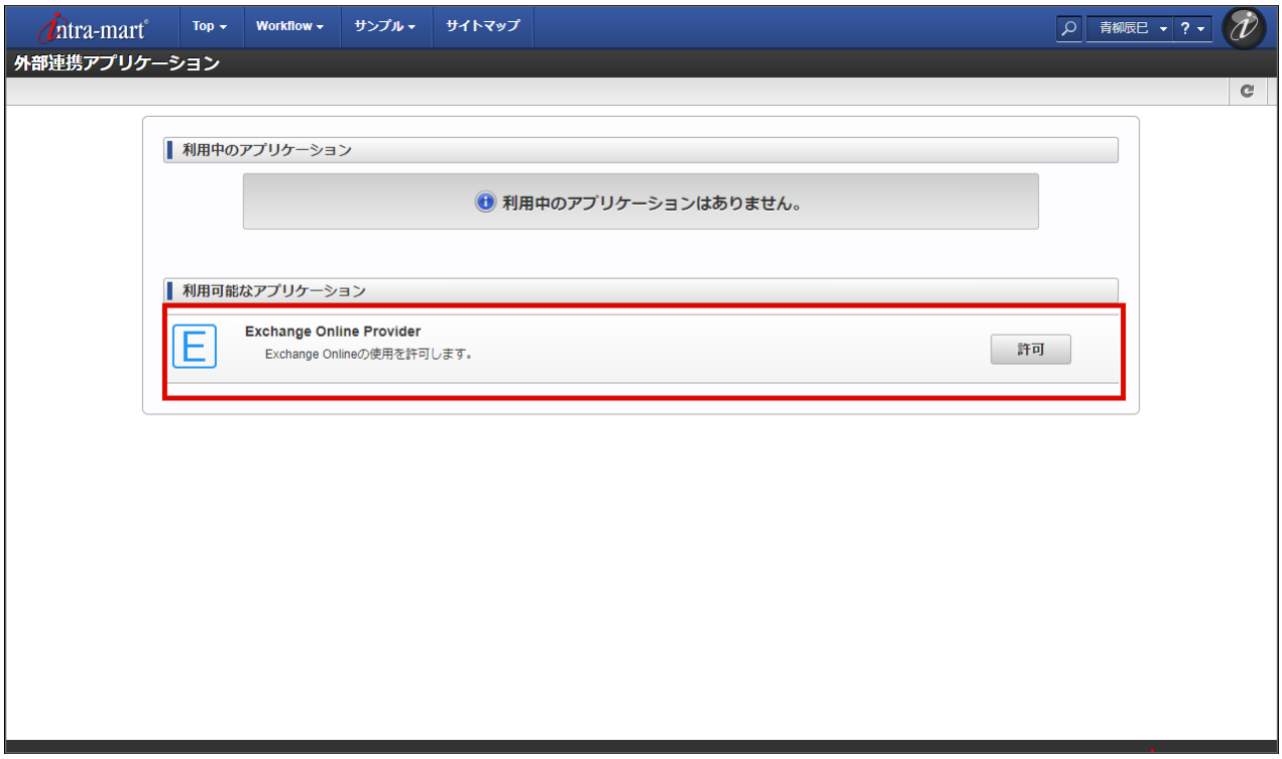
連携許可を行う

メール操作タスクでアクセストークンを利用するには、操作対象となるユーザ毎に、セットアップ後アクセストークンを発行する必要があります。
「外部連携アプリケーション」より、連携の許可を行うことで、アクセストークンが発行されます。

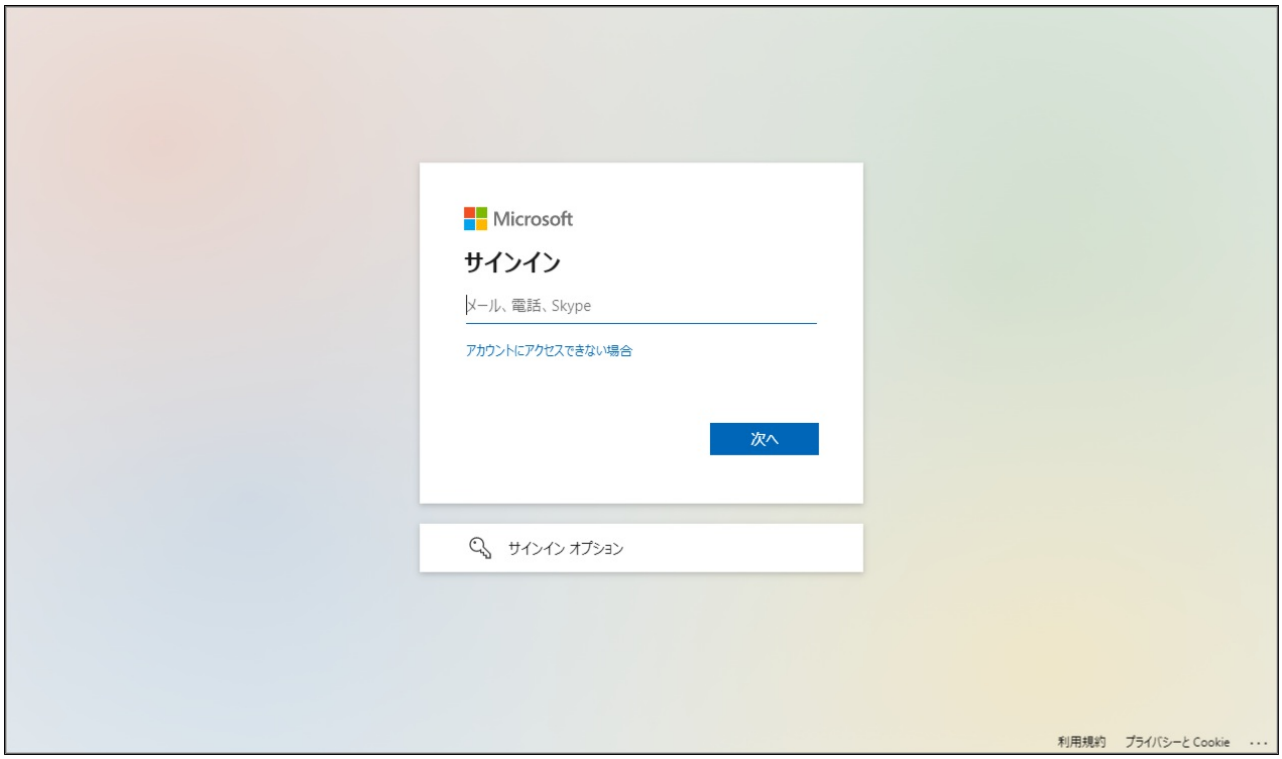
1. メール操作の対象ユーザで intra-mart Accel Platform にログインしてください。
2. ユーティリティメニューより、「個人設定」-「外部連携アプリケーション」を選択します。



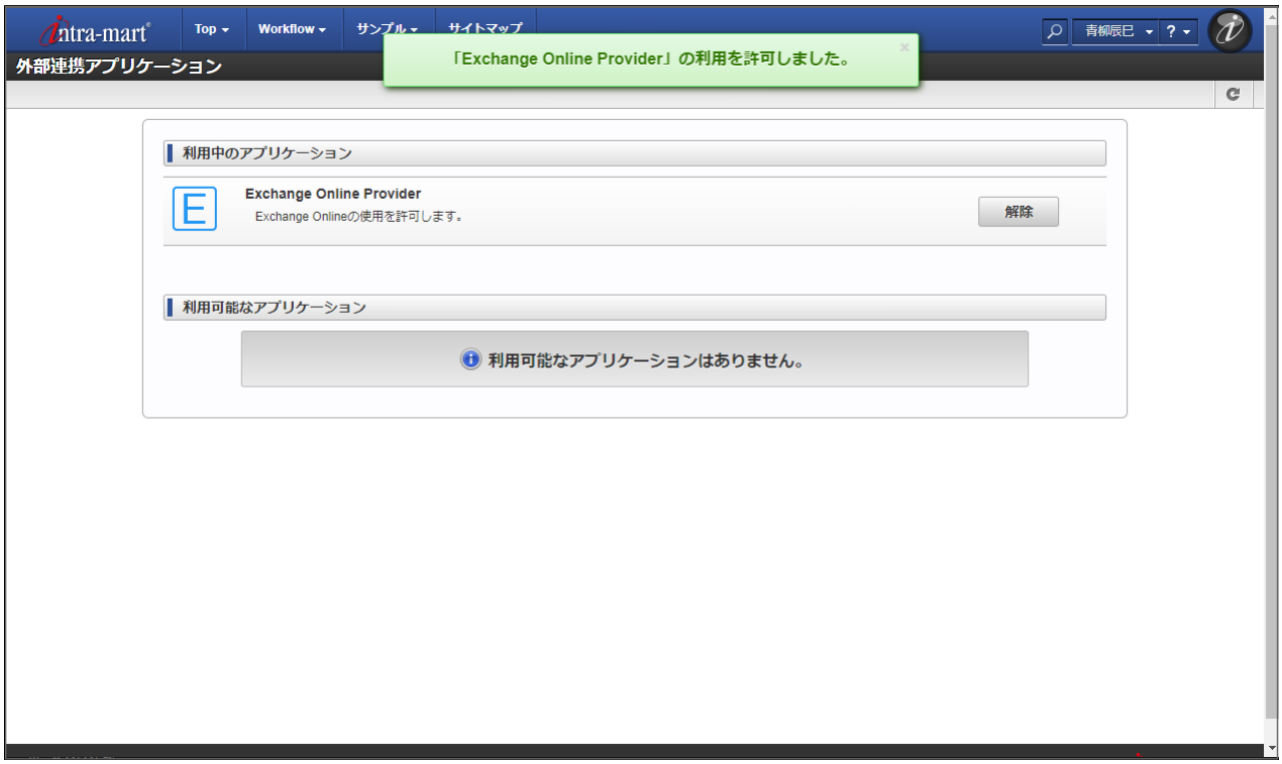
3. 「*プロバイダ情報を設定する*」で作成したプロバイダ情報の「許可」をクリックします。



4. アプリケーションの認証画面に遷移します。サインイン、および許可を行ってください。



5. 以下の画面が表示されれば連携が完了です。

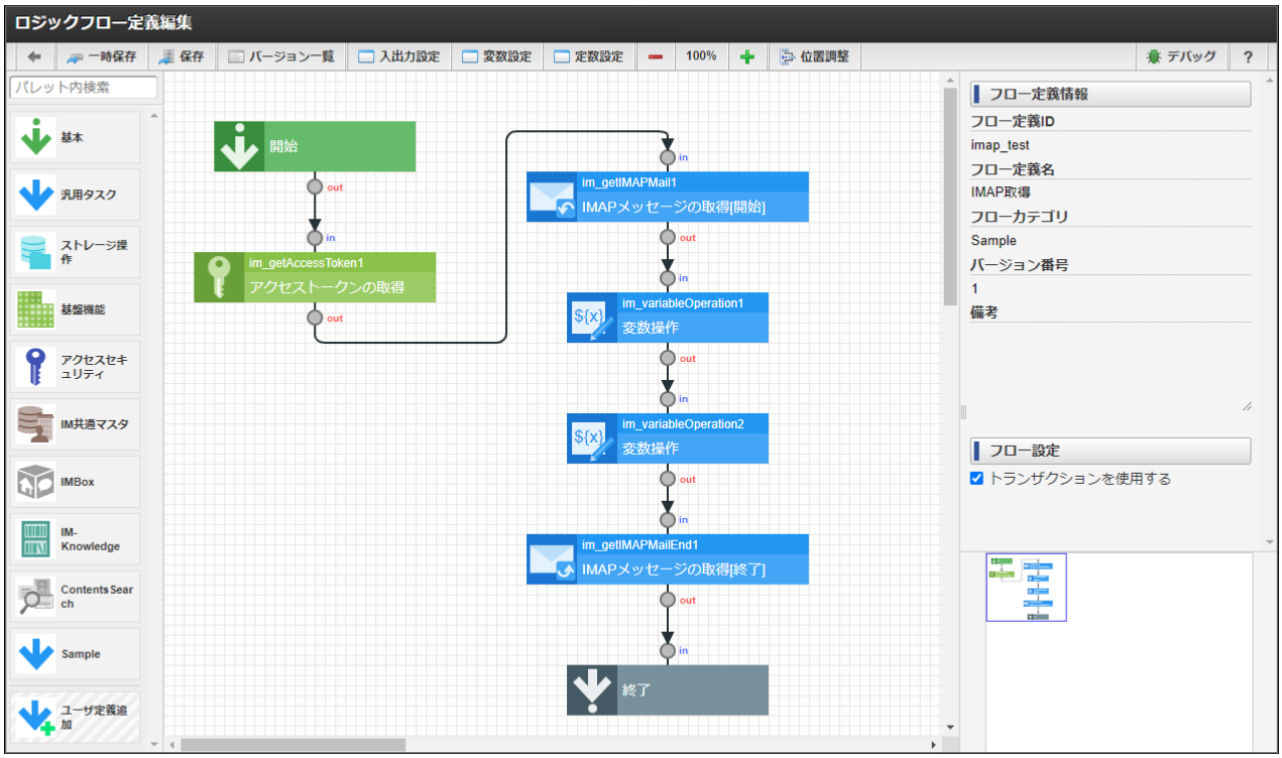


> **コラム**
>
> アクセストークンの期限切れなどで再発行を行う場合は、「解除」で一度連携を解除すると、「許可」がクリックできます。

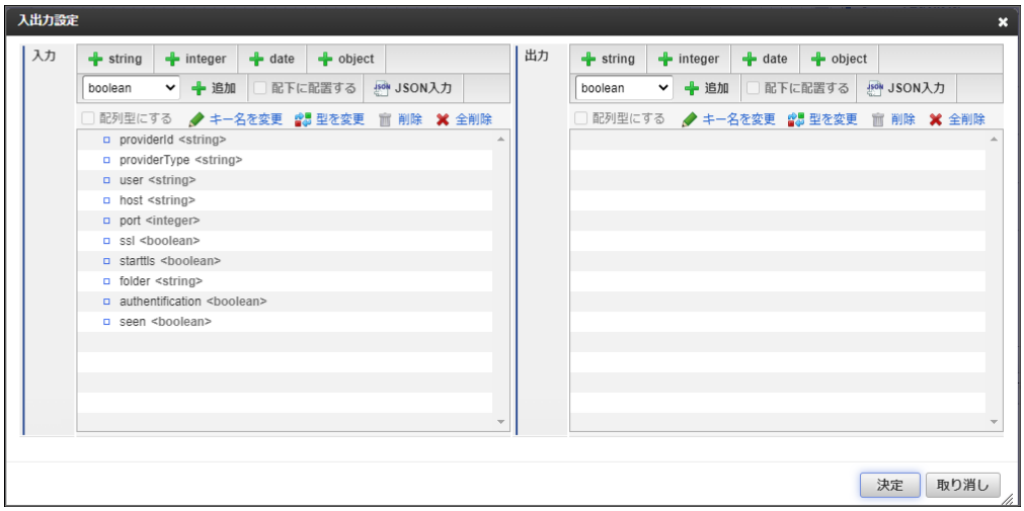
IM-LogicDesigner メール操作タスク（IMAPメッセージの取得）の設定例

以下はIMAPを利用したメール取得の設定例です。

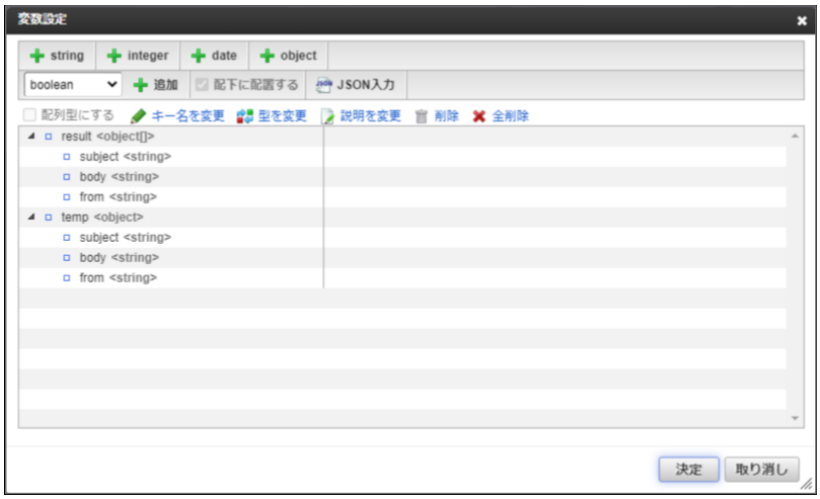
1. 連携許可を行ったユーザで 「サイトマップ」→「LogicDesigner」→「フロー定義一覧」に遷移します。
   「LogicDesigner」の操作には「LogicDesigner管理者」のロールが必要です。
2. 「ロジックフロー新規作成」をクリックし、以下のタスクを使用したフローを作成します。
   取得したメッセージを格納する変数も用意します。
   - アクセストークンの取得タスク
   - IMAPメッセージの取得タスク



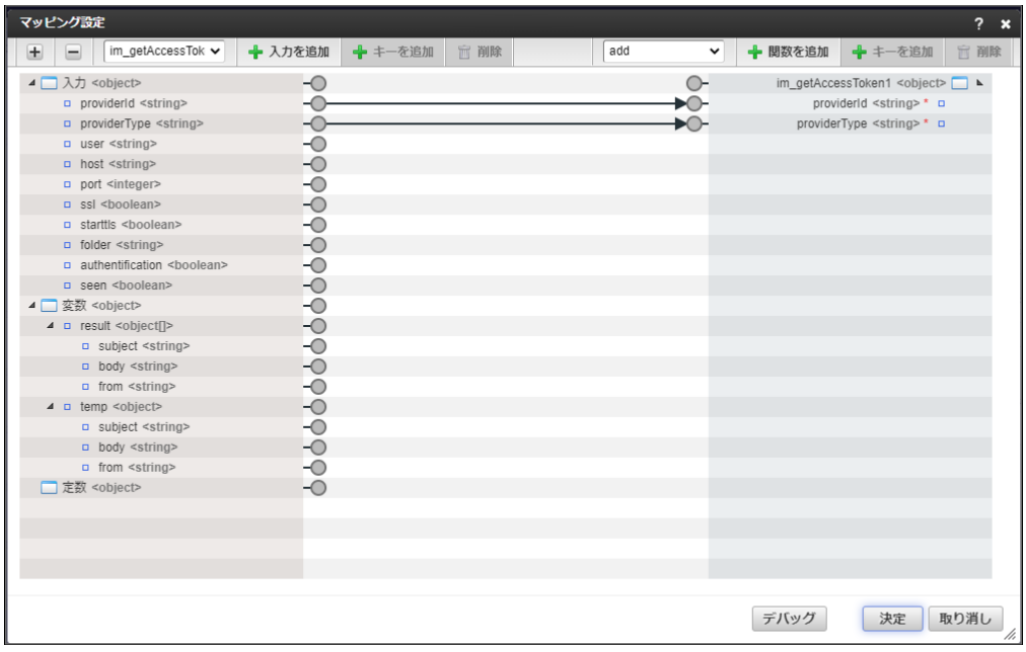
- アクセストークンの取得タスクと、IMAPメッセージの取得タスクに渡す値を入出力設定で定義しています。



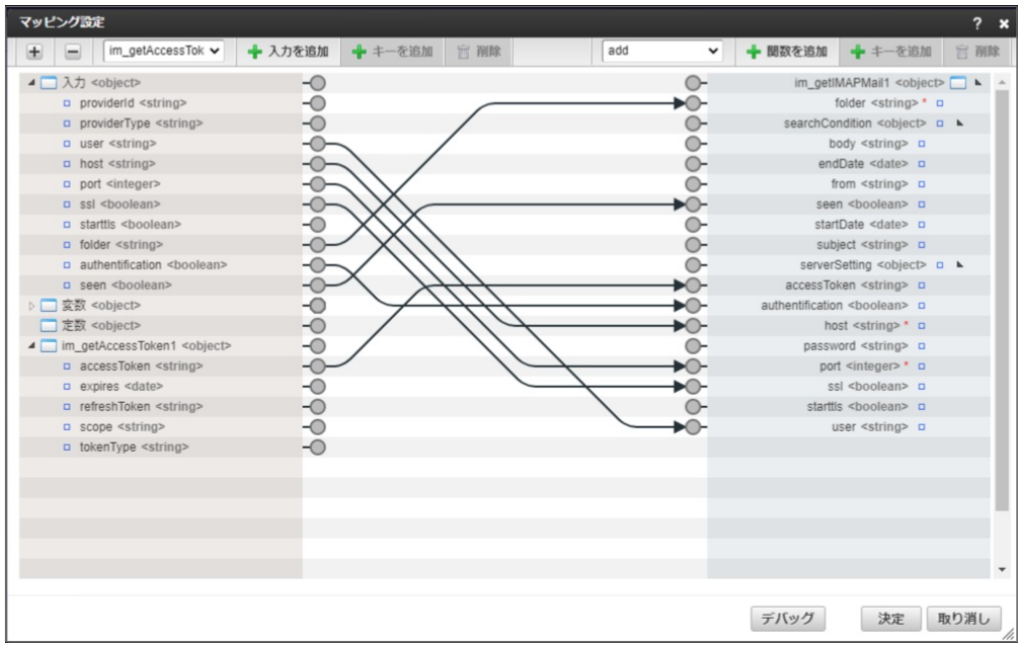
- IMAPメッセージの取得タスクの返却値を格納する変数を変数設定で定義しています。



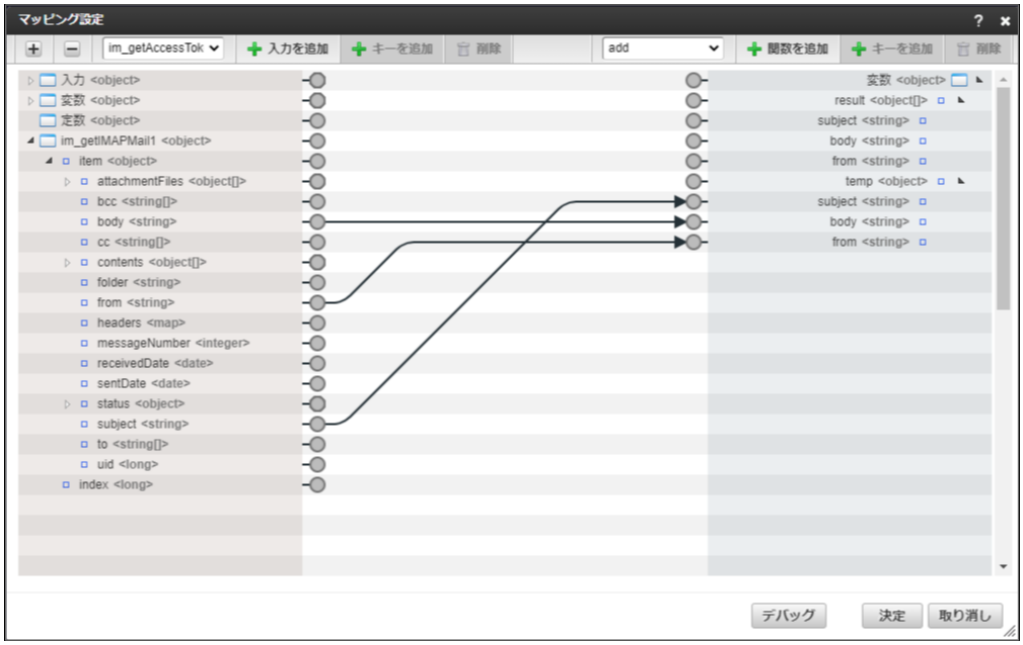
- アクセストークンの取得タスクのマッピングで入力値を渡しています。



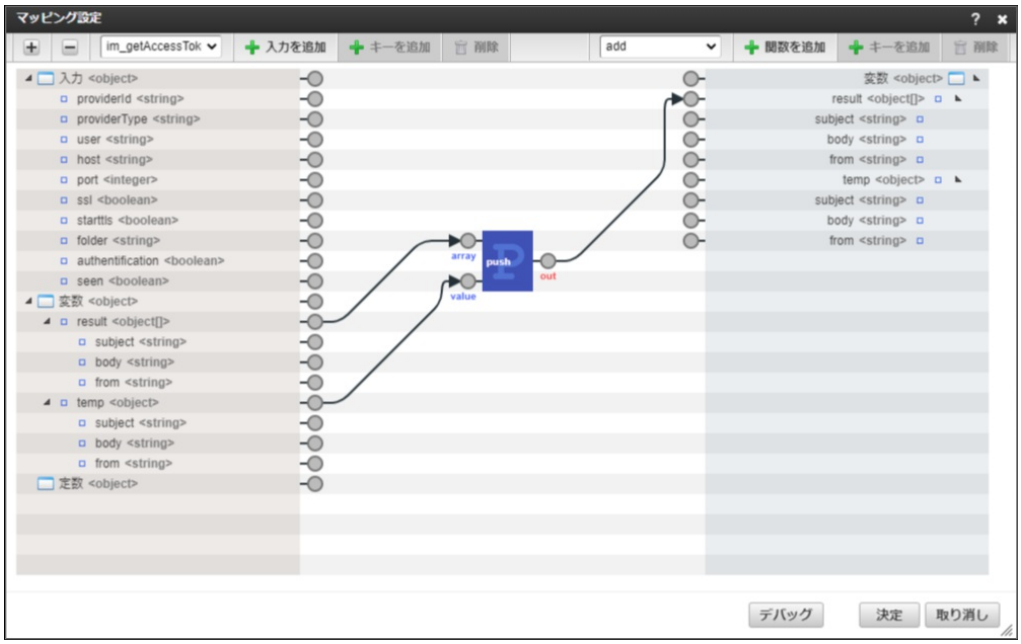
- IMAPメッセージの取得タスクのマッピングで、アクセストークンの取得タスクの返却値と、入力値を渡しています。



- IMAPメッセージの取得タスクからの返却値を、変数操作で変数「temp」にセットしています。

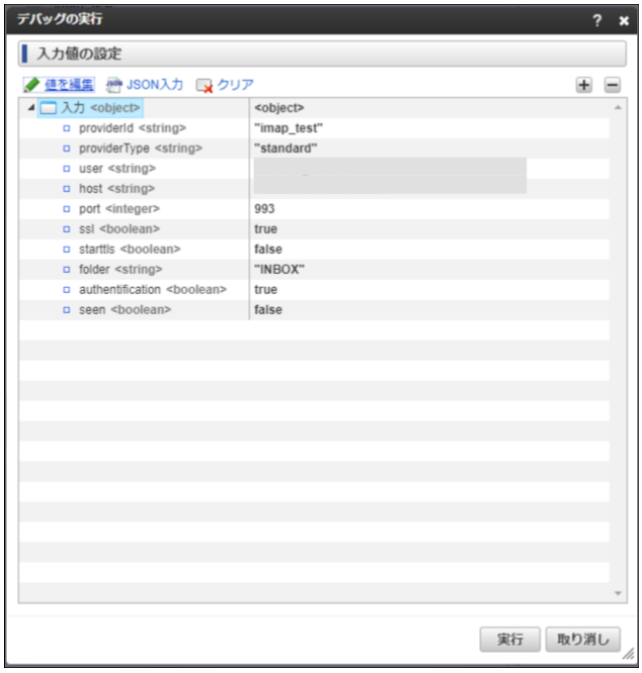


- 変数「temp」を、変数操作で変数「result」にプッシュしています。



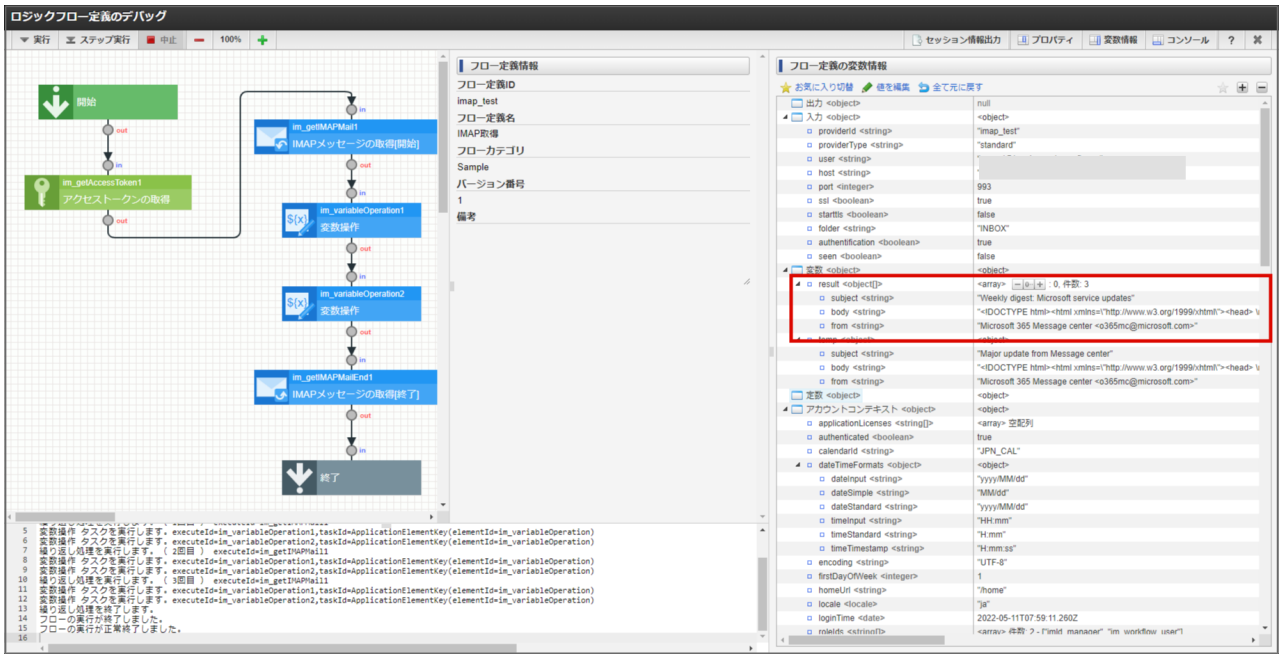
3. 「デバッグ」をクリックしてデバッグ画面を呼び出します。

4. 「実行」をクリックして、入力値を設定画面を呼び出し、「実行」します。



providerId	「*プロバイダ情報を設定する*」で設定した *id* を設定します。
providerType	「*プロバイダ情報を設定する*」で設定した *provider-type* を設定します。
authentification	*true* を設定します。
user	「*連携許可を行う*」でサインインしたユーザを設定します 。
folder	IMAPメッセージの取得タスク の *folder* から目的に合った値を設定します。
seen	IMAPメッセージの取得タスク の *seen* から目的に合った値を設定します。
host	サービスのIMAP設定に合わせて設定します。 Gmail Check Gmail through other email platforms を参照してください。 Exchange Online POP3 と IMAP4 を参照してください。
port	サービスのIMAP設定に合わせて設定します。 Gmail Check Gmail through other email platforms を参照してください。 Exchange Online POP3 と IMAP4 を参照してください。
ssl	サービスのIMAP設定に合わせて設定します。 Gmail Check Gmail through other email platforms を参照してください。 Exchange Online POP3 と IMAP4 を参照してください。
starttls	サービスのIMAP設定に合わせて設定します。 Gmail Check Gmail through other email platforms を参照してください。 Exchange Online POP3 と IMAP4 を参照してください。

5. 終了後、変数「result」にメッセージが格納できれば成功です。



IM-Notice

項目

- 概要
- 通知機能全般
  - モジュールの選択
  - IM-Notice 設定ファイルの編集
  - ベースURLの設定
- デスクトップ通知機能
  - スタンドアローン構成の場合
    - IM-Notice MQ設定ファイルの編集
  - 分散構成やWeb Serverを利用している構成の場合
    - IM-Notice MQ設定ファイルの編集
    - ブローカーサービス実行ファイルの取得
    - IM-Notice MQブローカー設定ファイルの編集
    - ブローカーサービスの実行
- モバイル通知機能（iOS版）
  - P12証明書ファイルの取得
  - Amazon SNSの設定
  - ポリシーの作成
  - アクセスキーとシークレットキーの作成
  - ロールの作成
  - IM-Notice Mobile設定ファイルの編集
  - P12証明書ファイルの更新
- モバイル通知機能（Android版）
  - FCMを使用する場合
    - FCMの設定
    - IM-Notice Mobile設定ファイルの編集
    - サービスアカウントの認証情報ファイルの更新
    - FCM HTTP v1 API への移行
  - FCM+Amazon SNSを使用する場合
    - FCM の設定
    - Amazon SNS の設定
    - ポリシーの作成
    - アクセスキーとシークレットキーの作成
    - ロールの作成
    - IM-Notice Mobile設定ファイルの編集

概要

IM-Noticeは、 intra-mart Accel Platform 上で動作しているさまざまなアプリケーションからの通知をデスクトップやスマートフォンに配信するための通知機能です。
Amazon Simple Notification Service（以下、Amazon SNS） や Firebase Cloud Messaging（以下、FCM）のサービスを使用し、スマートフォンアプリへ通知を配信することが可能です。

> **注意**
>
> IM-Notice を利用するには、OAuth認証モジュール が必要です。

> **注意**
>
> IM-Notice は統合Windows認証環境では利用できません。

> **注意**
>
> IM-Notice と Accel Platform Mobile のモバイル通知の同時利用は非対応です。

> **コラム**
>
> 本ページで紹介しているAWSやFCM資材の作成方法は一例です。お客様の環境に合わせて対応してください。

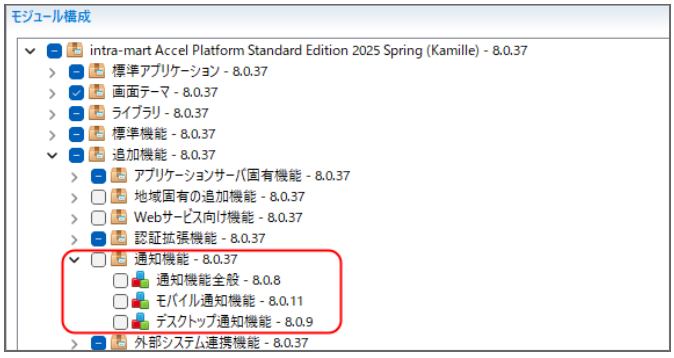
通知機能全般

通知機能全般を使用するための設定を行います。

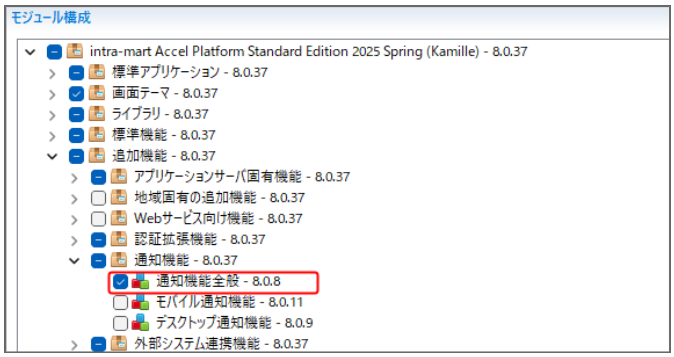
モジュールの選択

IM-Juggling でモジュールを選択します。

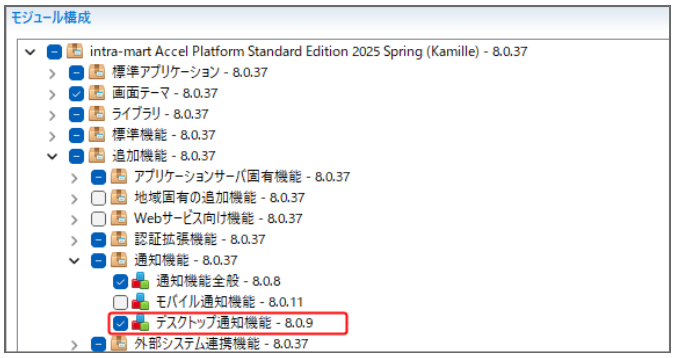
1. <（プロジェクト名）/juggling.im>の「ベースモジュール」タブから「通知機能」を開いてください。



2. 通知機能全般を選択します。

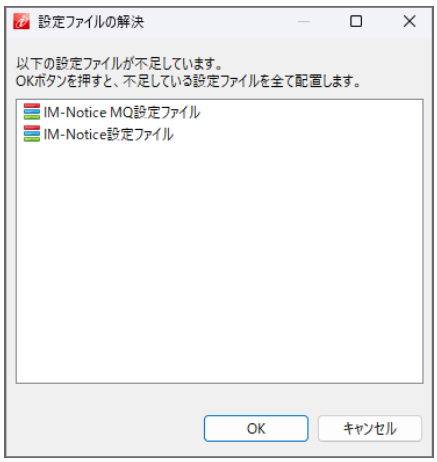


3. デスクトップへの通知を行う場合、デスクトップ通知機能を選択してください。



4. メッセージをクリックし、依存関係を解決してください。

5. 設定ファイルの解決を行うと、<（プロジェクト名）/conf> 配下に設定ファイルが出力されます。



IM-Notice 設定ファイルの編集

IM-Juggling の <（プロジェクト名）/conf> 配下に出力されたim-notice-config.xmlを開き、以下の設定を行ってください。

1. <base-url>
   - ベースURLを設定してください。
     この設定はショートカットURLの作成に使用されます。
     未設定の場合、server-context-config.xml で指定したベースURLが使用されます。

     ```
     <base-url>https://example.org/imart</base-url>
     ```

2. <short-cut-duration-minutes>
   - ショートカットURLの有効期間を設定してください。単位は分です。
     「0」を設定した場合、ショートカットURLは作成されません。
     デフォルトでは43200分=30日が設定されています。

     ```
     <short-cut-duration-minutes>43200</short-cut-duration-minutes>
     ```

ベースURLの設定

IM-Noticeへのメッセージ配信処理をジョブ経由で行う場合は、ベースURLの設定が必要です。 「*ベースURL*」または、「*テナント環境情報*」から

設定を行ってください。

> **注意**
>
> ベースURLを指定しない場合、以下のエラーが発生します。
> jp.co.intra_mart.system.notice.exception.NoticeRuntimeException: [E.NOTICE.CORE.00016] ベースURLを解決できませんでした。

デスクトップ通知機能

デスクトップ通知機能を使用するための設定を行います。

スタンドアローン構成の場合

スタンドアローン構成の場合は、Web Application Server とデスクトップ通知のクライアントが直接通信を行います。
IM-Notice MQ設定ファイル（im-notice-mq-config）を設定する必要があります。

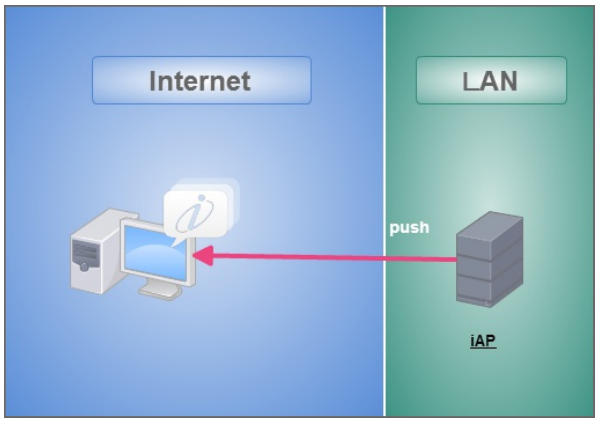


図 スタンドアローンのサーバ構成例（デスクトップ通知）

IM-Notice MQ設定ファイルの編集

IM-Juggling の <（プロジェクト名）/conf> 配下に出力されたim-notice-mq-config.xmlを開き、以下の設定を行ってください。

1. <address>
   - デスクトップ通知で使用するポート番号を設定してください。
     デフォルトでは40608ポートを使用する設定です。
     クライアントからサーバまでのネットワーク機器などに対して、ここで指定したポートを開放してください。

   ```
   <address bind="true">tcp://*:40608</address>
   ```

2. <ping>
   - ping機能の設定をします。
     有効にするには、enable属性に「true」を設定してください。
   - interval-seconds属性で、クライアントにpingを流す間隔を設定します。単位は秒です。
   - time-to-live-seconds属性で、クライアントがサーバに再接続を行うまでの時間を設定します。単位は秒です。
     設定した秒数だけサーバから応答がなければ、クライアントはサーバに再接続を行います。

   ```
   <ping enable="true" interval-seconds="270" time-to-live-seconds="300" />
   ```

   > **コラム**
   >
   > 環境により、一定時間データが流れないソケットは切断されてしまう場合があります。
   > そのような環境では、数分間隔でpingを流し続けることで自動切断を避けることができます。
   > （Microsoft AzureやAmazon Elastic Compute Cloudなど）

3. <endpoint>
   - クライアントの接続先を設定します。
     <address>で設定したポート番号を指定してください。

```
<endpoints>
  <endpoint>tcp://example.org:40608</endpoint>
</endpoints>
```

分散構成やWeb Serverを利用している構成の場合

分散構成やWeb Serverを利用している構成の場合は、ブローカーのサービスを実行するサーバの起動、および設定ファイルの編集が必要です。

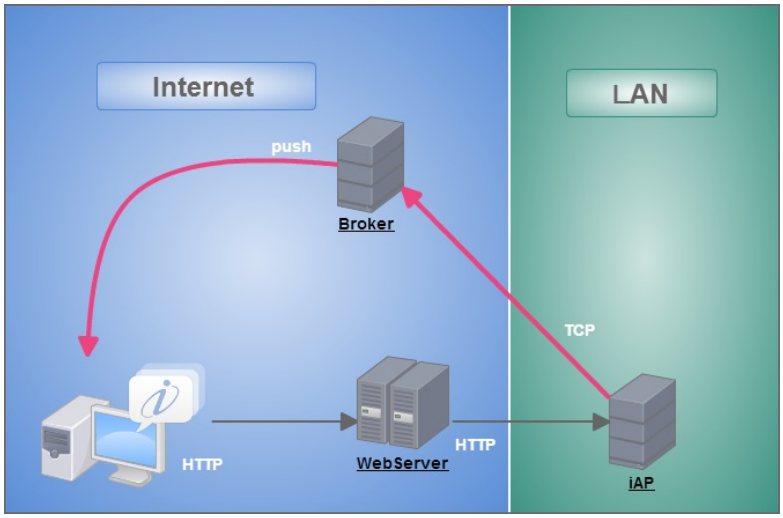


図 WebServerなどを含む分散構成のサーバ構成例（デスクトップ通知）

> **コラム**
> ブローカーとは、ネットワーク中継を行うためのプロキシです。

> **コラム**
> ブローカーサービスを実行するサーバのシステム要件は「リリースノート」-「システム要件」を参照してください。

IM-Notice MQ設定ファイルの編集

IM-Juggling で出力されたim-notice-mq-config.xmlを編集してください。

1. <address> および <endpoint>
   <address> の bind 属性に false を指定し、ブローカーのバックエンドポートの接続情報を設定してください。
   <endpoint> にブローカーのフロントエンドポートの接続情報を設定してください。

```
<address bind="false">tcp://{ブローカーを実行しているサーバのホストまたはIPアドレス}:{ブローカーのバックエンドポート番号}</address>
<ping enable="true" interval-seconds="270" time-to-live-seconds="300" />
<endpoints>
 <endpoint>tcp://{ブローカーを実行しているサーバのホストまたはIPアドレス}:{ブローカーのフロントエンドポート番号}</endpoint>
</endpoints>
```

> **コラム**
> IM-Notice MQ設定ファイルのその他の項目については *IM-Notice MQ設定ファイルの編集* を参照してください。

ブローカーサービス実行ファイルの取得

ブローカーサービスを実行する実行ファイル（im_notice_mq_broker）を取得します。
弊社サイトの プロダクトファイルダウンロード よりライセンスキーを入力して取得してください。

IM-Notice MQブローカー設定ファイルの編集

取得したブローカーサービス実行ファイルを展開したconfディレクトリ内のim-notice-mq-broker-config.xmlを編集します。

1. <frontend>
   - デスクトップ通知アプリケーションとの通信を行うポート番号を設定してください。
     デフォルトでは40608ポートを使用する設定です。
     クライアントからブローカーまでのネットワーク機器などに対して、ここで指定したポートを開放してください。

```
<frontend>tcp://*:40608</frontend>
```

2. <backend>
   - Web Application Server との通信を行うポート番号を設定してください。
     デフォルトでは40609ポートを使用する設定です。
     ブローカーからサーバまでのネットワーク機器などに対して、ここで指定したポートを開放してください。

     ```
     <backend>tcp://*:40609</backend>
     ```

> **コラム**
>
> IM-Notice MQブローカー設定ファイルのその他の項目は IM-Notice ブローカー設定ファイルと同様です。
> IM-Notice ブローカー設定ファイルについては *IM-Notice MQ設定ファイルの編集* を参照してください。

ブローカーサービスの実行

取得したブローカーサービス実行ファイルを展開したディレクトリにある im_notice_mq_broker-8.0.x.jar をjavaコマンドで実行してください。

フォアグラウンドで動作させる場合（例）

```
java -jar im_notice_mq_broker-8.0.x.jar
```

バックグラウンドで動作させる場合（例）

```
nohup java -jar im_notice_mq_broker-8.0.x.jar > stdout.log 2>stderr.log &
```

> **コラム**
>
> 実行したブローカーサービスを停止する場合は、実行方法に応じた停止を行ってください。
> 上記実行例に対する停止方法は以下です。
>
> - フォアグラウンドでjavaコマンドを実行した場合、Ctrl-C で終了。
> - バックグラウンドでjavaコマンドを実行した場合、プロセスIDを調べてkillコマンドで終了。

> **コラム**
>
> Java 17 以降で運用する場合、javaコマンドの直後に `--add-opens java.base/java.lang=ALL-UNNAMED` を追加して実行してください。
>
> - フォアグラウンドで動作させる場合（例）
>
>   ```
>   java --add-opens java.base/java.lang=ALL-UNNAMED -jar im_notice_mq_broker-8.0.x.jar
>   ```
>
> - バックグラウンドで動作させる場合（例）
>
>   ```
>   nohup java --add-opens java.base/java.lang=ALL-UNNAMED -jar im_notice_mq_broker-8.0.x.jar > stdout.log 2>stderr.log &
>   ```

モバイル通知機能（iOS版）

モバイル通知機能を使用するための設定を行います。 iOSへの通知を行うには、Amazon SNSを使用します。

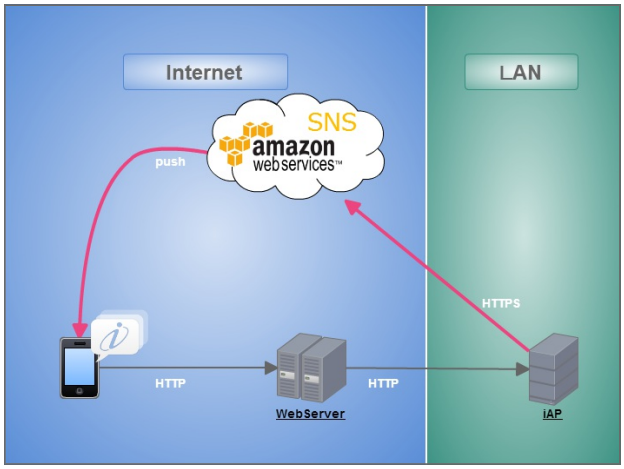


図 Amazon SNSを利用したサーバ構成例

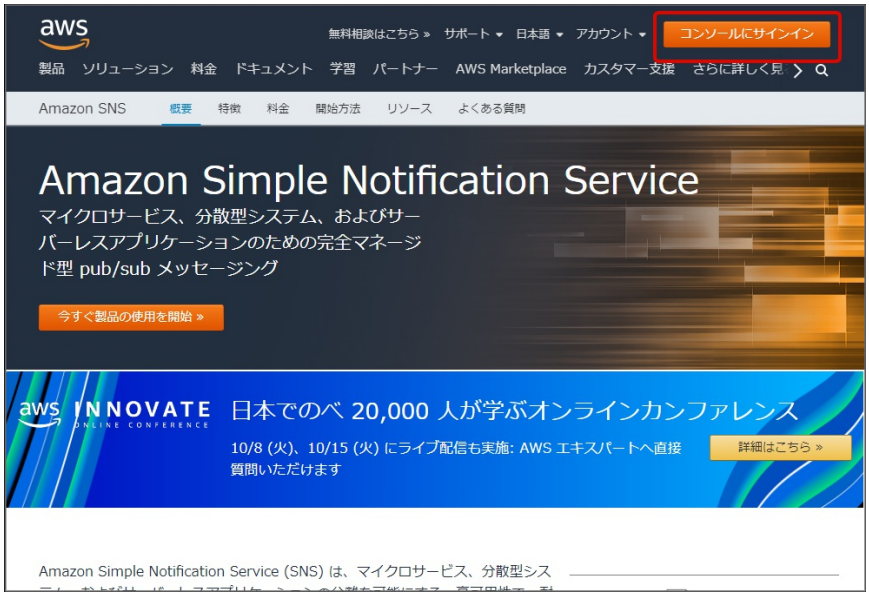
P12証明書ファイルの取得

P12証明書ファイル（IM-Notice Push Notifications Production.p12）を取得します。
弊社サイトの プロダクトファイルダウンロード よりライセンスキーを入力して取得してください。

Amazon SNSの設定

Amazon SNSの設定を行います。

1. Amazon SNS にサインインしてください。



2. リージョンを選択してください。

3. サービスを選択してください。

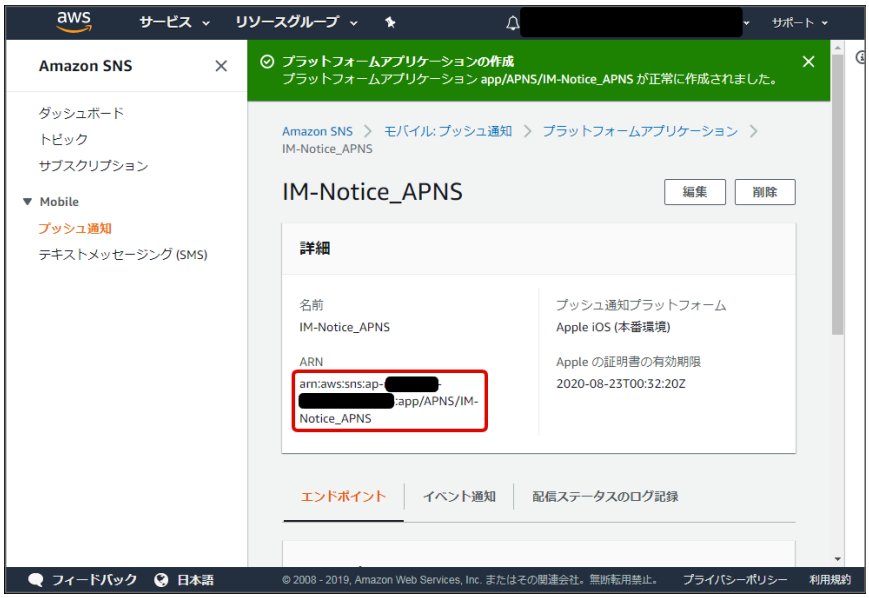
4. 「Mobile」-「プッシュ通知」から「プラットフォームアプリケーションの作成」をクリックしてください。

5. フォームに以下の内容を入力し、「プラットフォームアプリケーションの作成」をクリックしてください。

項目	説明
アプリケーション名	任意の名前を入力します。

プッシュ通知プラットフォーム	「Apple iOS/VoIP/Mac」を選択します。
プッシュ証明書タイプ	「iOS プッシュ証明書」を選択します。
ファイルの選択	「IM-Notice Push Notifications Production.p12」を添付します。
証明書のパスワード	「intra-mart」と入力し、「認証情報をファイルから読み込み」をクリックします。

6. アプリケーションが追加されました。
   「ARN」を確認します。
   設定ファイルを編集する際に必要な文字列ですので、控えておいてください。



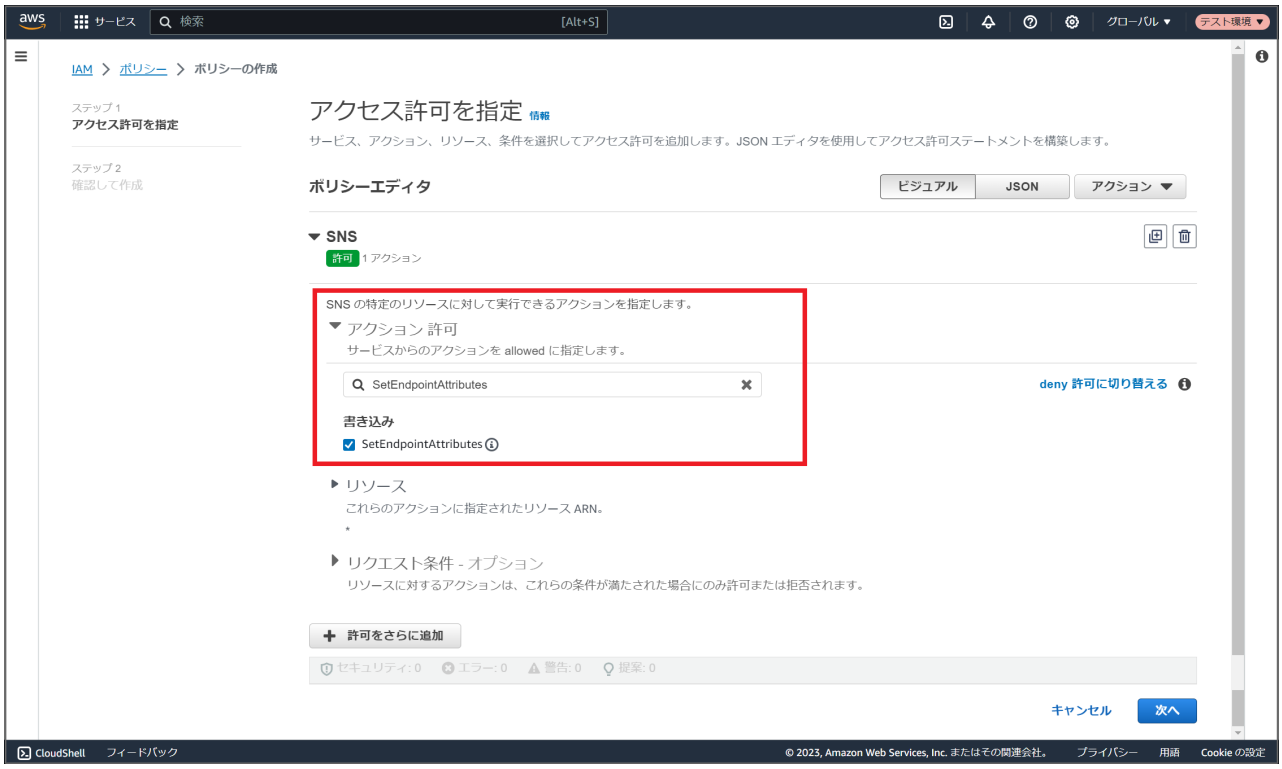
ポリシーの作成

Amazon SNSに接続するために必要な権限を付与するための、IAMポリシーの作成を行います。
作成したポリシーは、 *アクセスキーとシークレットキーの作成* または *ロールの作成* に使用します。

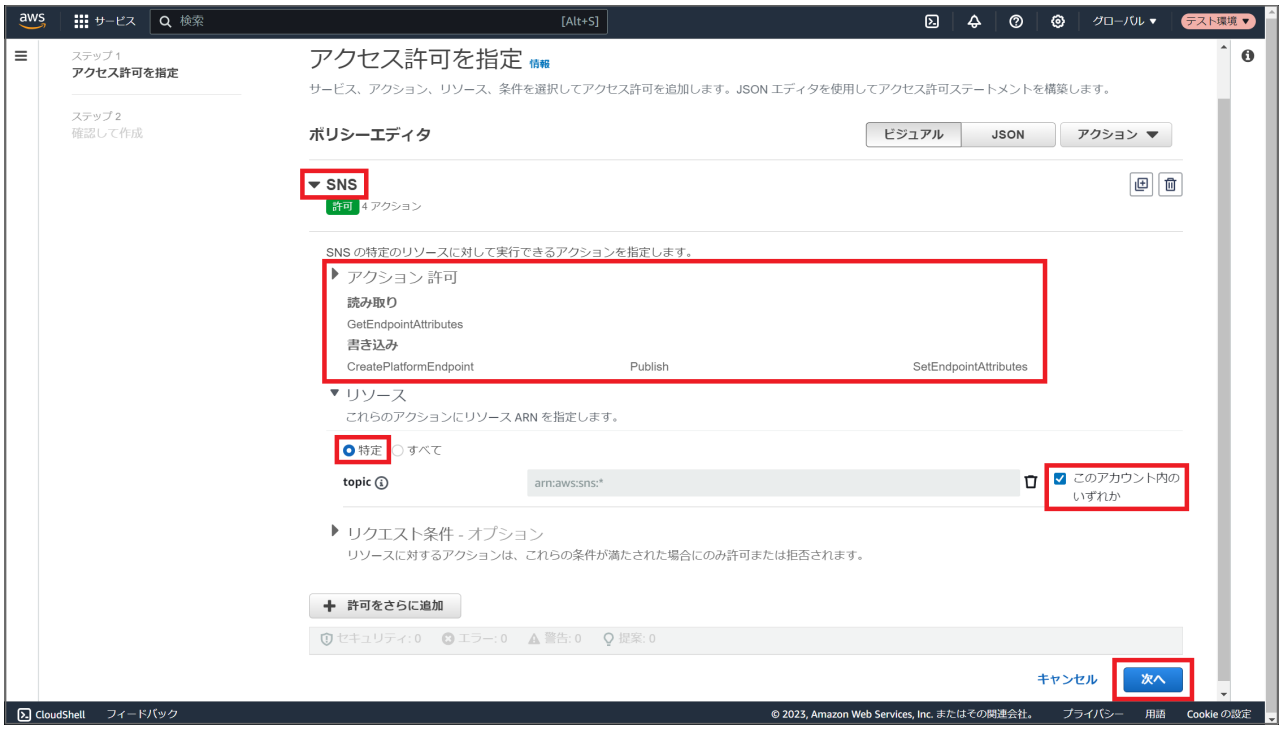
1. AWS Identity and Access Management (IAM) にサインインしてください。
2. 「ポリシー」を選択し、「ポリシーを作成」を押下してください。

3. 「サービスを選択」にて「SNS」を検索し、選択してください。

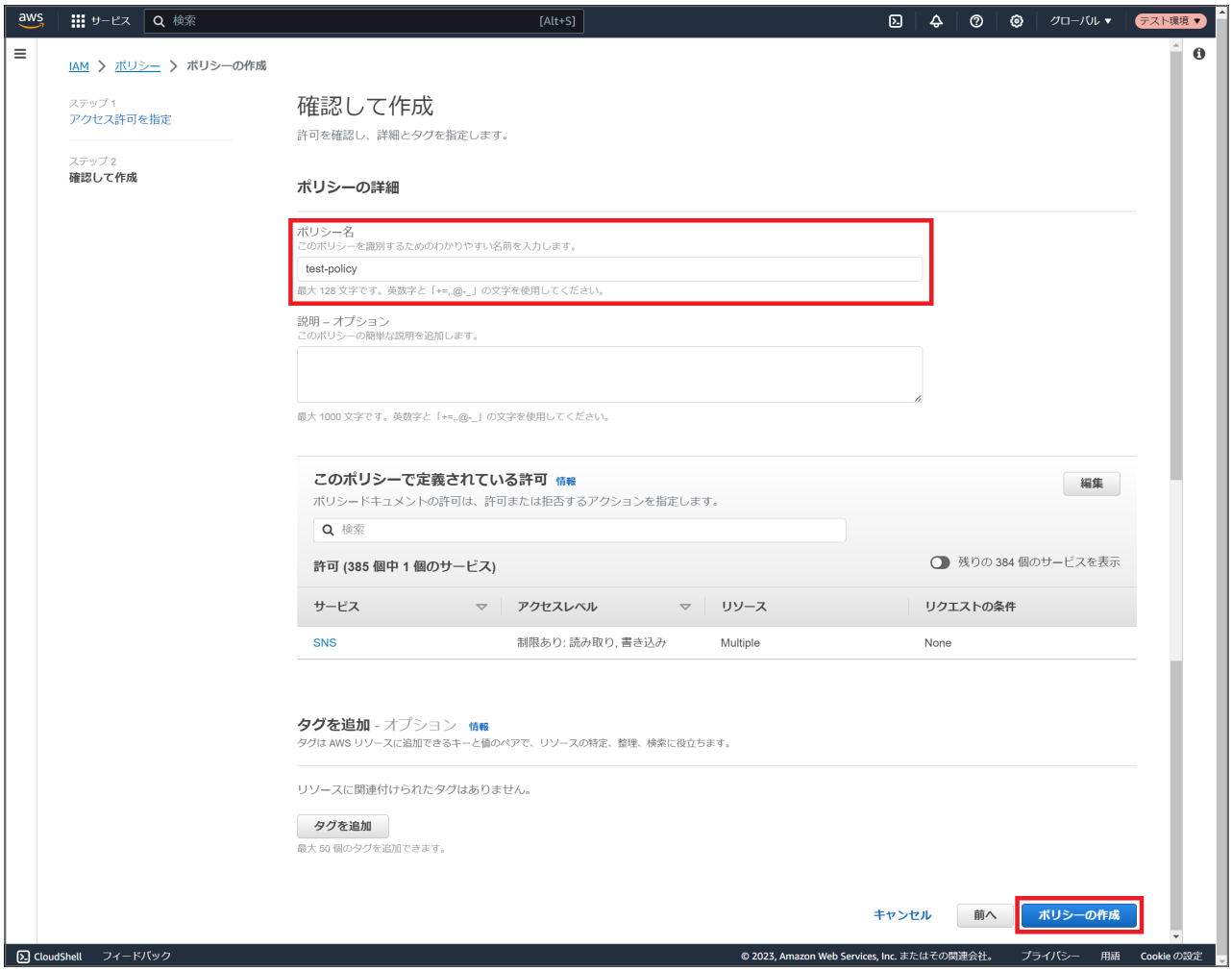
4. サービスを選択すると表示される「アクション許可」にて以下4つのアクションをそれぞれ検索し、チェックを入れてください。
   - SetEndpointAttributes
   - GetEndpointAttributes
   - Publish
   - CreatePlatformEndpoint



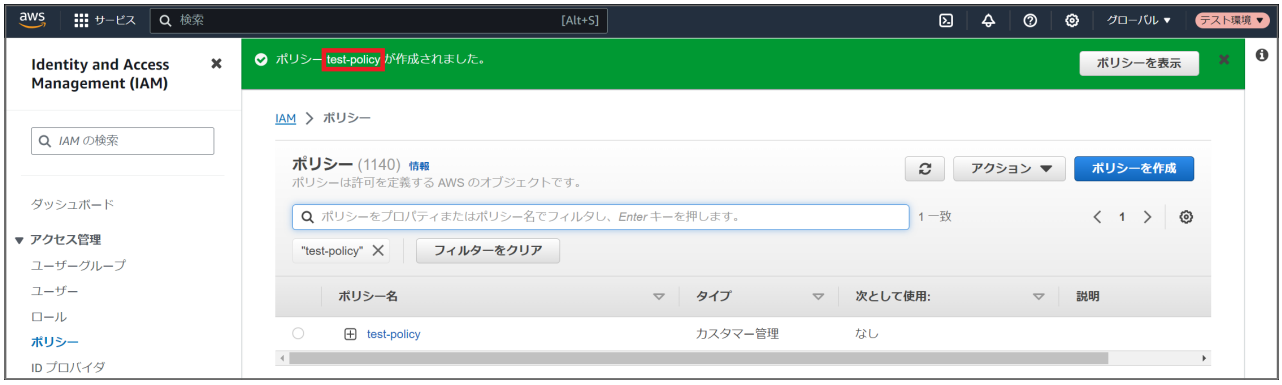
5. 選択したサービス、および、アクションが表示されていることを確認してください。
   その後、「リソース」にて「特定」を選択し、「このアカウント内のいずれか」にチェックを入れ、「次へ」を押下してください。



6. 「ポリシー名」に任意のポリシー名を入力し、「ポリシーの作成」を押下してください。



7. ポリシーが作成されました。
   作成されたポリシー名は *アクセスキーとシークレットキーの作成* や *ロールの作成* に使用しますので、控えておいてください。



アクセスキーとシークレットキーの作成

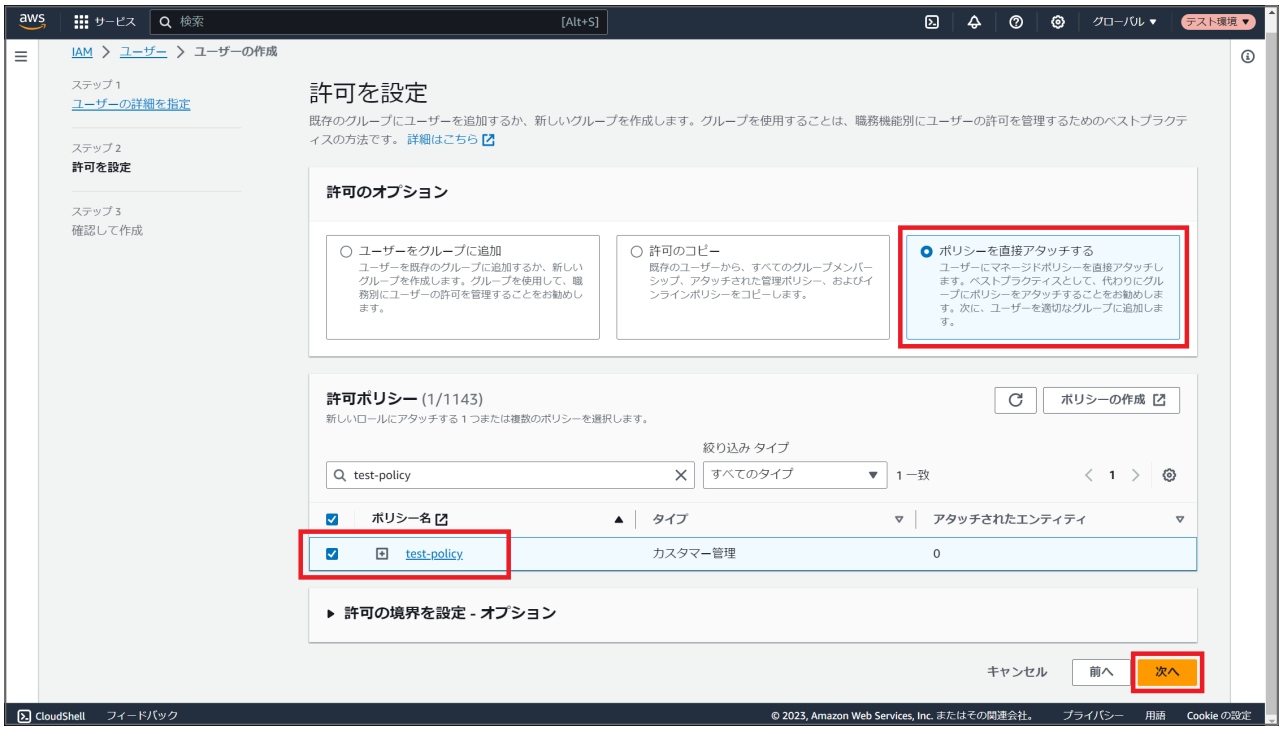
Amazon SNSへの接続に必要な、アクセスキーとシークレットキーの作成を行います。

> **コラム**
>
> アクセスキーとシークレットキー、または、ロールのうち、いずれか一方の作成が必要です。
> なお、ロールは、お客様環境がEC2インスタンス上で稼働している場合のみ利用できます。

1. AWS Identity and Access Management (IAM) にサインインしてください。
2. 「ユーザー」を選択し、「ユーザーの作成」を押下してください。

3. 「ユーザー名」に任意のユーザー名を入力し、「次へ」を押下してください。

4. 「許可のオプション」にて「ポリシーを直接アタッチする」を選択してください。
その後、 *ポリシーの作成* で作成したポリシー名にチェックを入れ、「次へ」を押下してください。

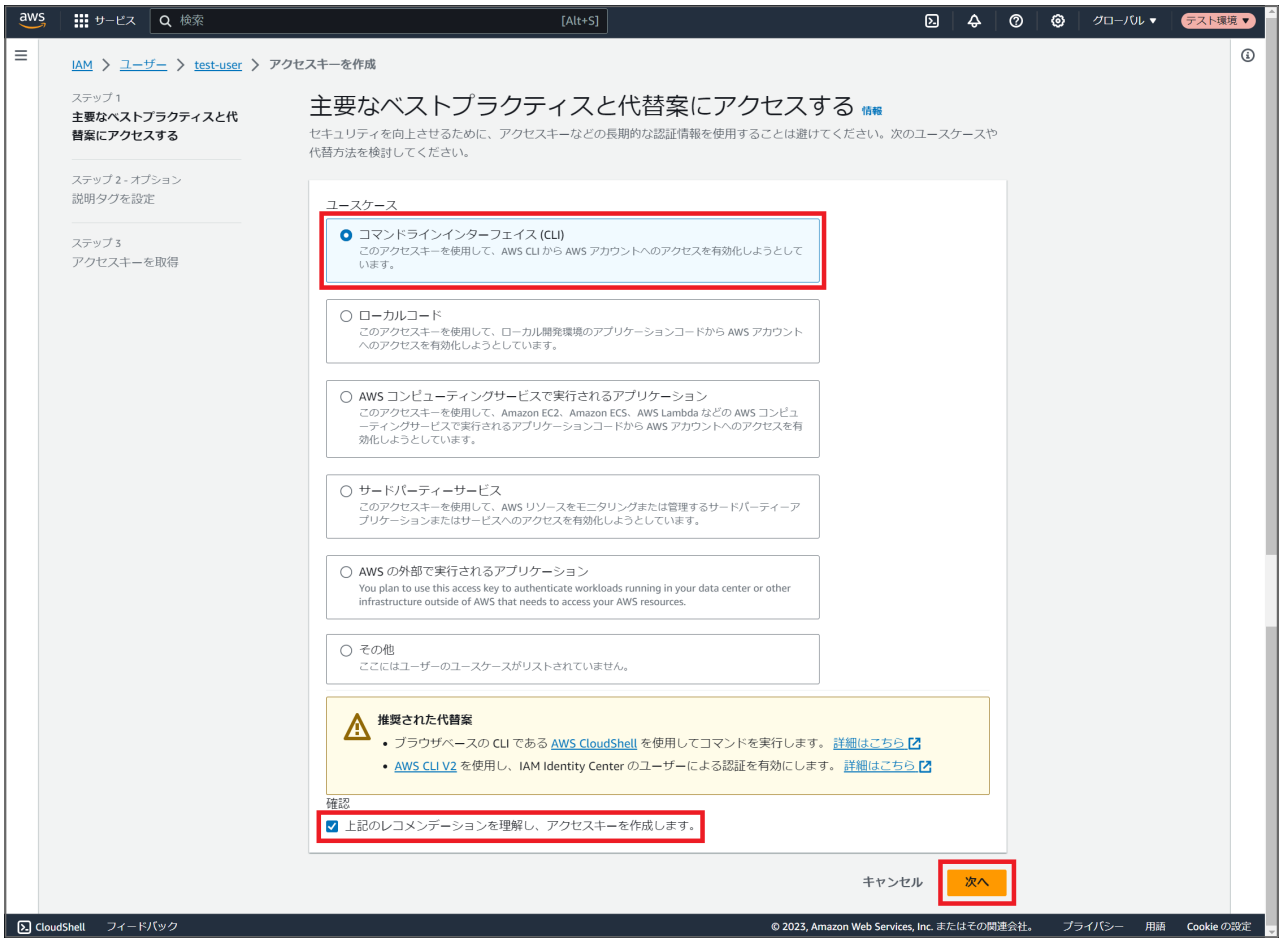


5. 「ユーザーの作成」を押下してください。

6. 作成したユーザー名を押下してください。

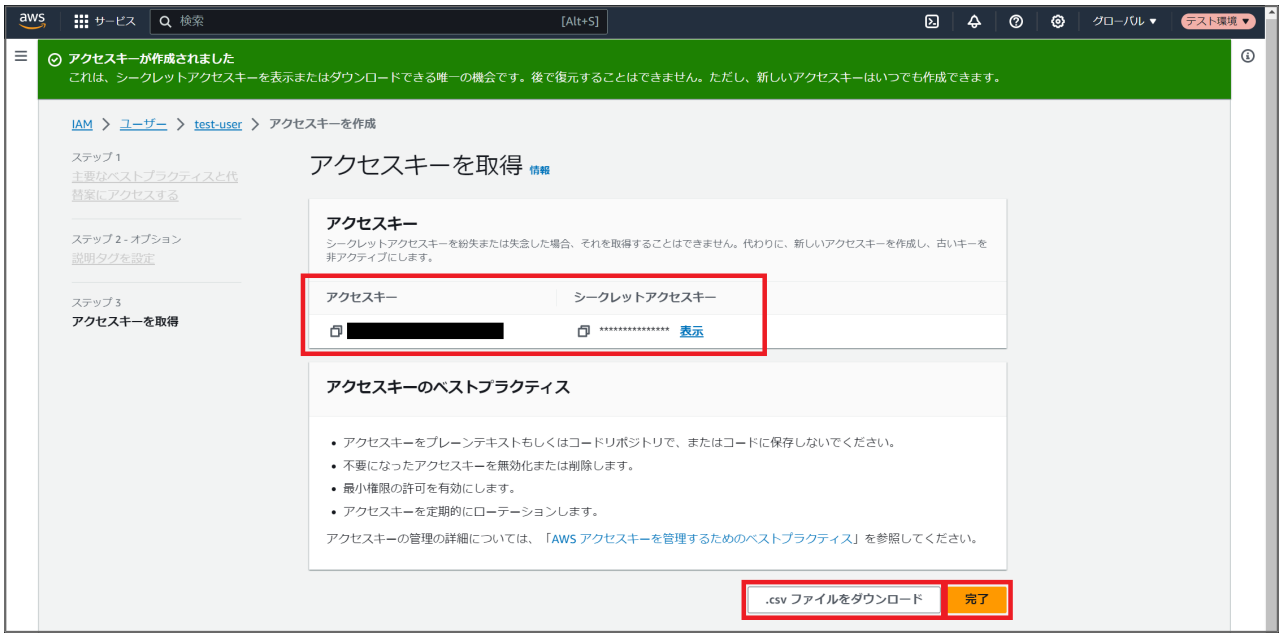
7. 「セキュリティ認証情報」タブを押下し、「アクセスキー」にて「アクセスキーを作成」を押下してください。

8. 「コマンドラインインターフェイス (CLI)」を選択し、「上記のレコメンデーションを理解し、アクセスキーを作成します。」にチェックを入れ、「次へ」を押下してください。



9. 「アクセスキーを作成」を押下してください。

10. 「アクセスキーを取得」画面でアクセスキーとシークレットアクセスキーを控えるか、「.csv ファイルをダウンロード」を押下してください。その後、「完了」を押下してください。

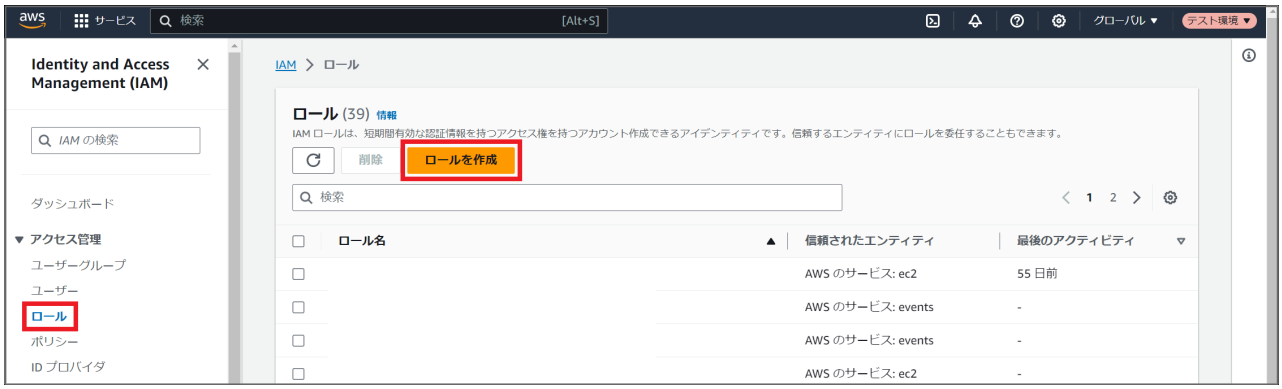


ロールの作成

Amazon SNSへの接続に必要な、IAMロールの作成を行います。

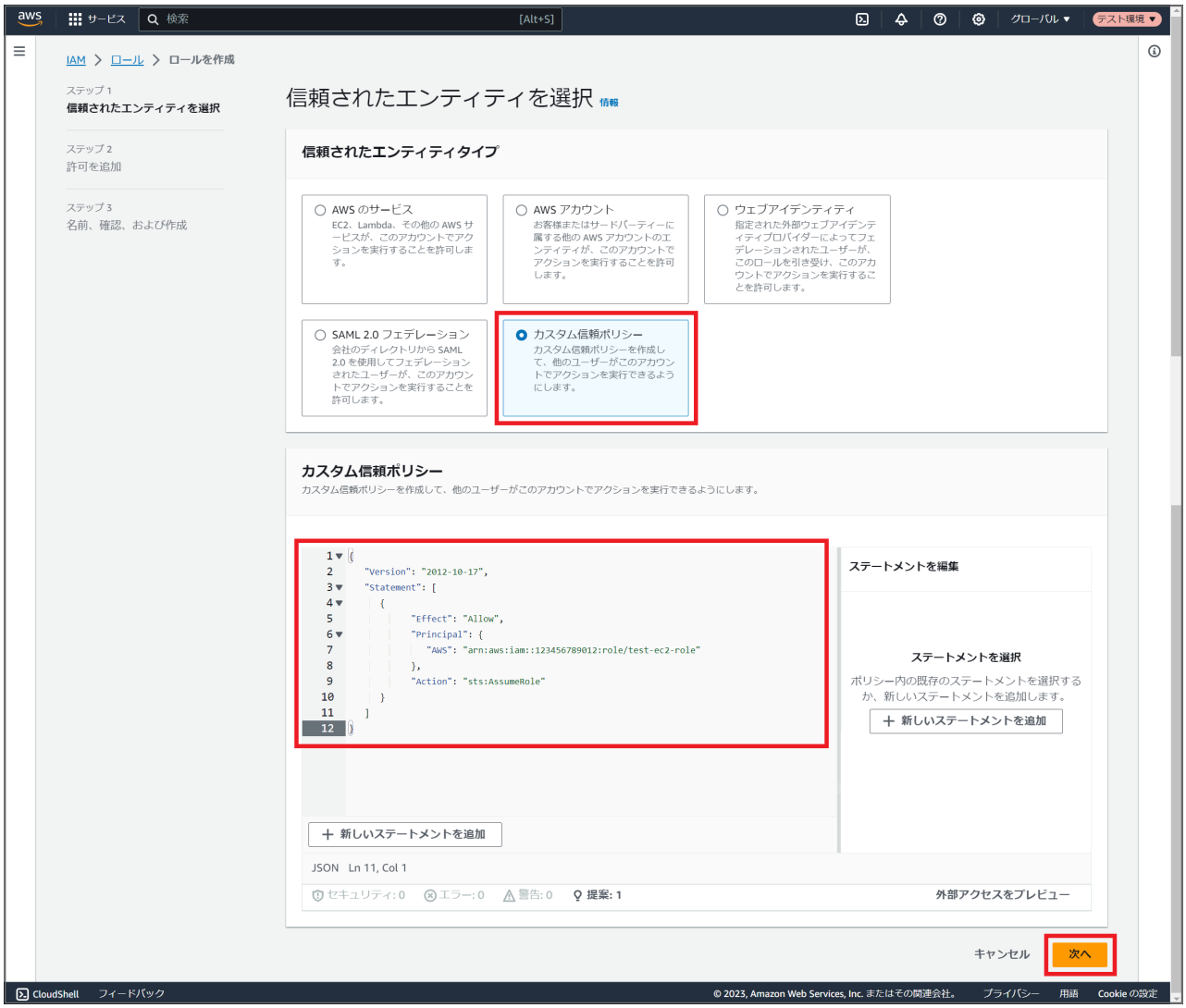
> **コラム**
>
> アクセスキーとシークレットキー、または、ロールのうち、いずれか一方の作成が必要です。
> なお、ロールは、お客様環境がEC2インスタンス上で稼働している場合のみ利用できます。

1. AWS Identity and Access Management (IAM) にサインインしてください。
2. 「ロール」を選択し、「ロールを作成」を押下してください。



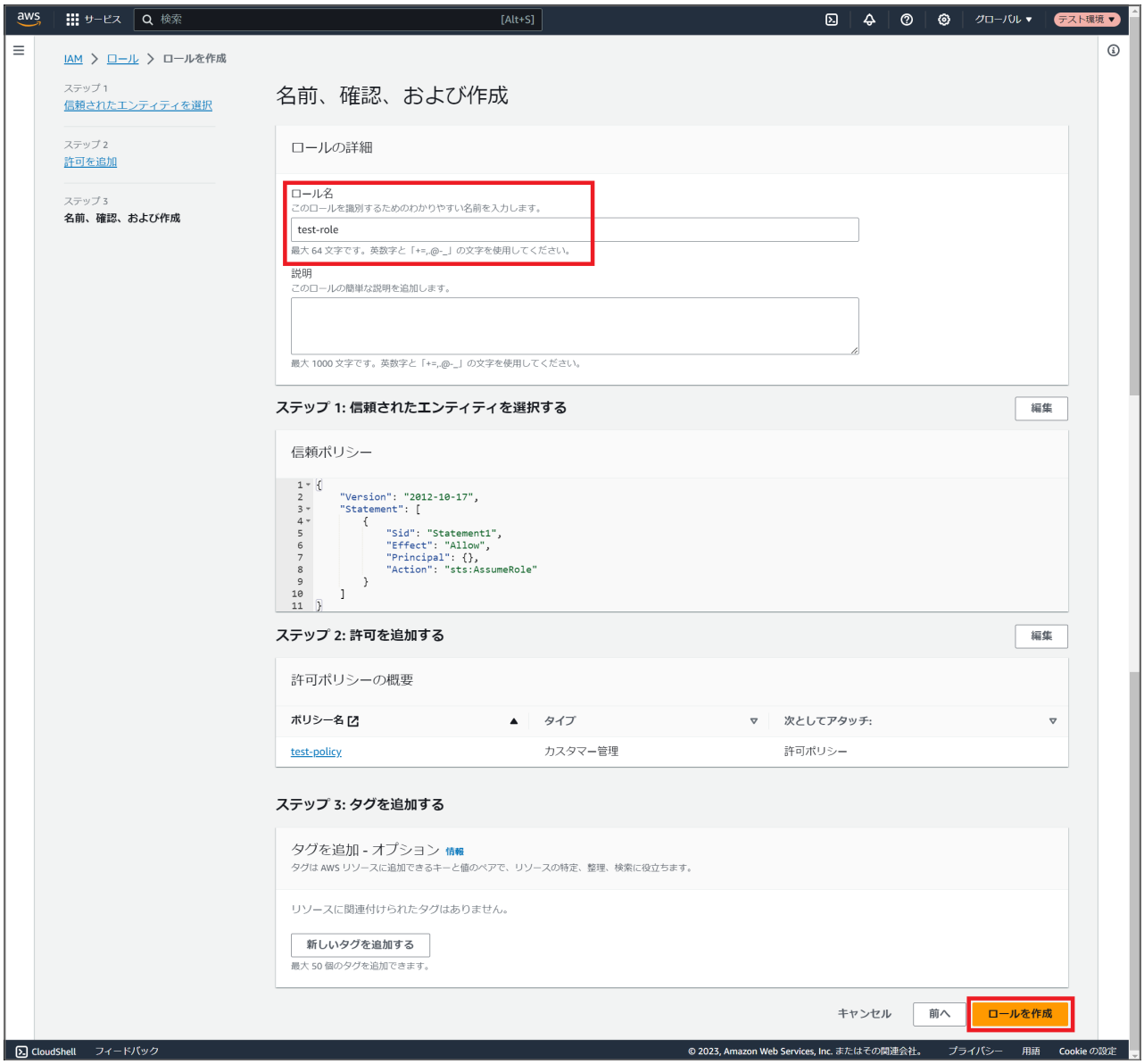
3. 「信頼されたエンティティタイプ」にて「カスタム信頼ポリシー」を選択し、「カスタム信頼ポリシー」に以下を入力してください。
`${AWSアカウントID}` と `${EC2インスタンスにアタッチされたロール名}` の箇所は、お客様の環境に合わせて変更してください。
その後、「次へ」を押下してください。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
        "Effect": "Allow",
        "Principal": {
          "AWS": "arn:aws:iam::${AWSアカウントID}:role/${EC2インスタンスにアタッチされたロール名}"
        },
        "Action": "sts:AssumeRole"
    }
  ]
}
```

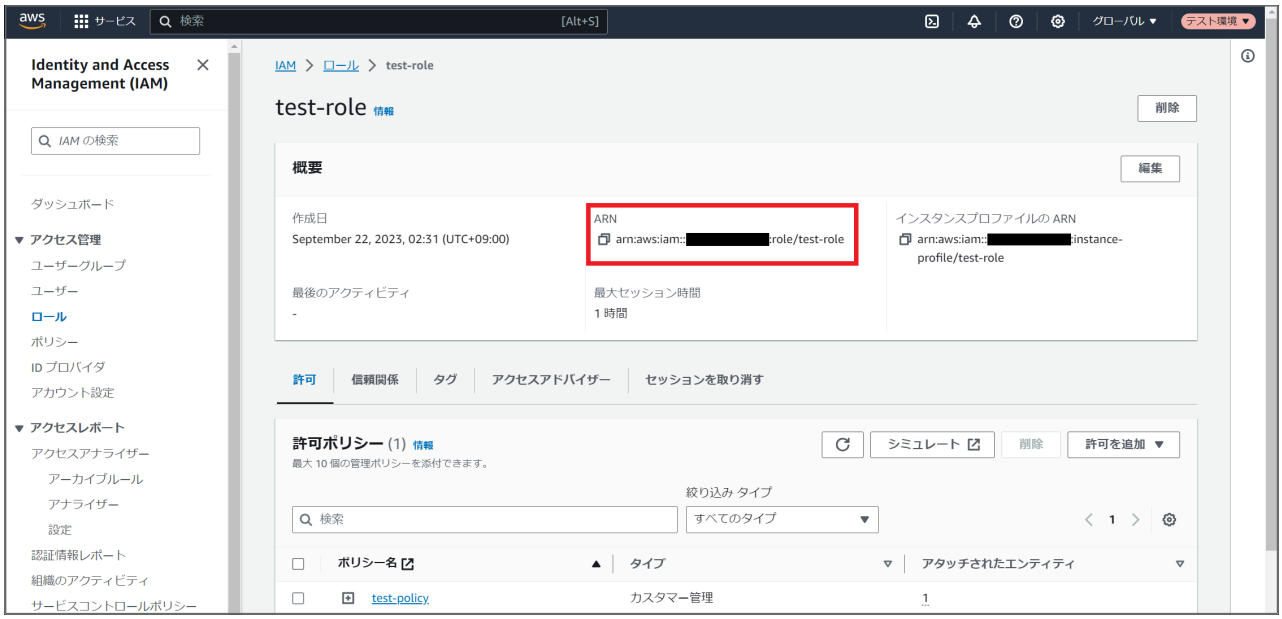


4. 「許可ポリシー」にて、 *ポリシーの作成* で作成したポリシー名にチェックを入れ、「次へ」を押下してください。

5. 「ロール名」に任意のロール名を入力し、「ロールを作成」を押下してください。



6. ロールが作成されました。
   作成されたロールのARNは設定ファイルの編集に使用しますので、控えておいてください。



IM-Notice Mobile設定ファイルの編集

IM-Juggling の <（プロジェクト名）/conf> 配下に出力されたim-notice-mobile-config.xmlを開き、以下の設定を行ってください。

- <proxy>

1. Amazon SNSへ接続するためにプロキシを使用する場合、設定を行ってください。

```
<proxy enable="true">
  <host></host>
  <port xsi:nil="true"></port> <!-- ポート番号を指定する場合は xsi:nil="true" を削除してください。 -->
  <username></username>
  <password></password>
  <workstation></workstation>
  <domain></domain>
</proxy>
```

- <asns>
  1. <asns>のname属性に任意の名前を設定してください。

```
<asns name="asns-apns">
  ....
</asns>
```

  2. <endpoint>
     - Amazon SNSのリージョンに対応するEndpointを設定してください。

```
<endpoint>sns.ap-northeast-1.amazonaws.com</endpoint>
```

> **コラム**
>
> Endpointは AWS documentation - Regions and Endpoints で確認できます。
> リージョンが「Asia Pacific (Tokyo)」の場合、Endpoint は「sns.ap-northeast-1.amazonaws.com」です。

  3. <platform-application-arn>
     - *Amazon SNSの設定* で確認したApplication ARNを設定してください。

```
<platform-application-arn>arn:aws:sns:ap-northeast-1:XXXXXXXXXXXX:app/APNS/IM-Notice_APNS</platform-application-arn>
```

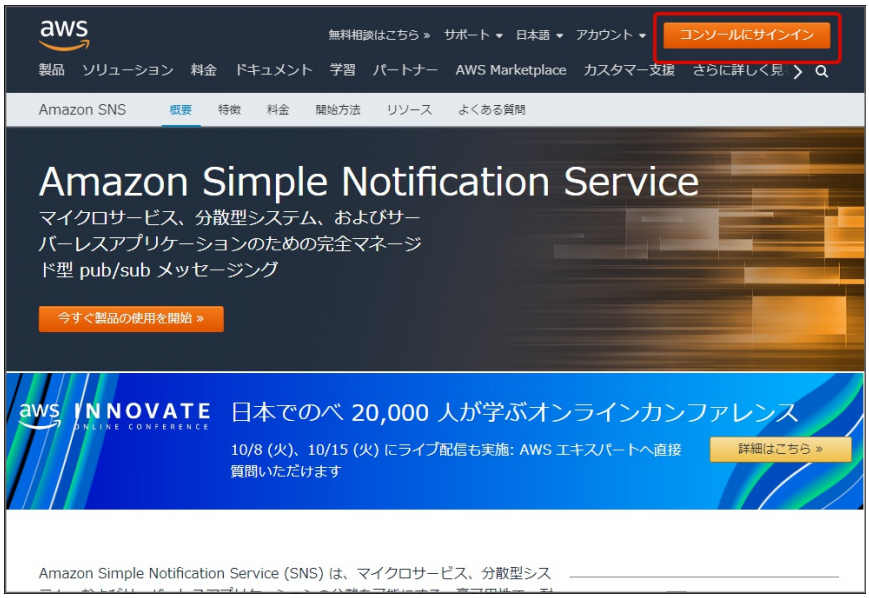
  4. <access-key>
     - 通知にアクセスキーとシークレットキーを使う場合、 *アクセスキーとシークレットキーの作成* で作成したアクセスキーを設定してください。
  5. <secret-key>
     - 通知にアクセスキーとシークレットキーを使う場合、 *アクセスキーとシークレットキーの作成* で作成したシークレットキーを設定してください。
  6. <iam-role-arn>
     - 通知にロールを使う場合、 *ロールの作成* で作成したロールのARNを設定してください。
- <push type=”ios” />
  1. engine属性に、<asns>のname属性に設定した名前を指定してください。

```
<push type="ios" engine="asns-apns" />
```

P12証明書ファイルの更新

P12証明書ファイル（IM-Notice Push Notifications Production.p12）の有効期限が切れてしまった、または新しいP12証明書ファイルが公開された場合、更新を行います。 有効期限が過ぎてしまったP12証明書ファイルを利用した場合、通知が届かなくなるため、有効期限が切れる前に新しいP12証明書ファイルに更新してください。

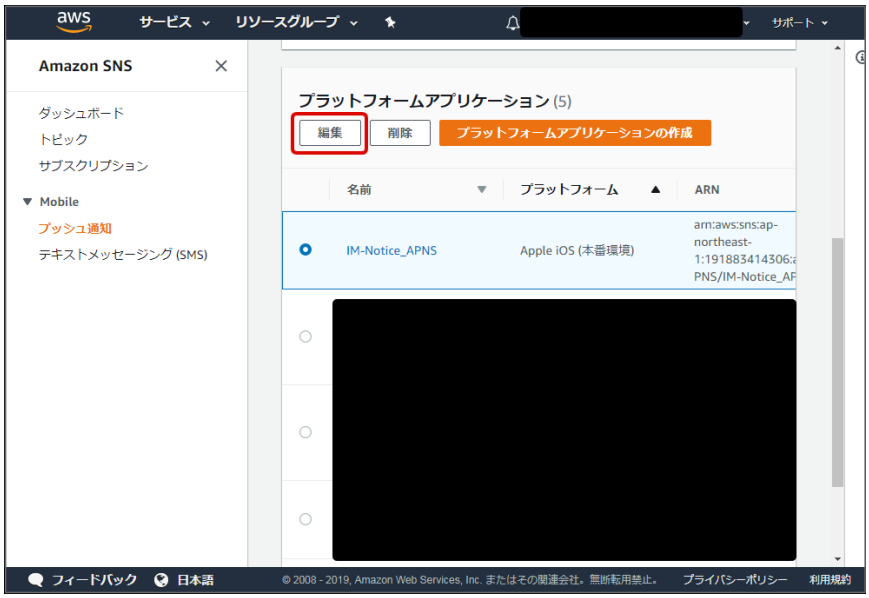
1. 弊社サイトのダウンロード ライブラリから最新のP12証明書ファイルを取得してください。
2. Amazon SNSにサインインしてください。



3. リージョンを選択してください。

4. サービスを選択してください。

5. 「Mobile」ー「プッシュ通知」からアプリケーションを選択し、「編集」をクリックします。



6. フォームに以下の内容を入力し、「変更の保存」をクリックしてください。

プッシュ証明書タイプ	「iOS プッシュ証明書」を選択します。
ファイルの選択	「IM-Notice Push Notifications Production.p12」を添付します。
証明書のパスワード	「intra-mart」と入力し、「認証情報をファイルから読み込み」をクリックします。

モバイル通知機能（Android版）

モバイル通知機能を使用するための設定を行います。 Androidへの通知を行うには、以下の2種類の構成が利用できます。

- FCM を使用する場合の設定方法は、 *FCMを使用する場合* を参照してください。

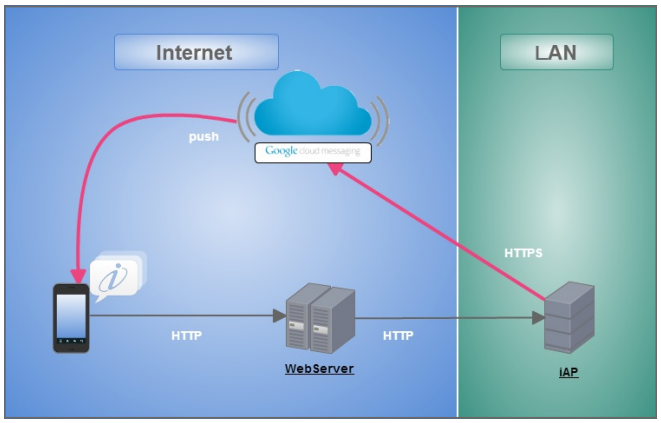


図 FCM を使用する場合のサーバ構成例

- FCM＋Amazon SNSを使用する場合の設定方法は、 *FCM+Amazon SNSを使用する場合* を参照してください。

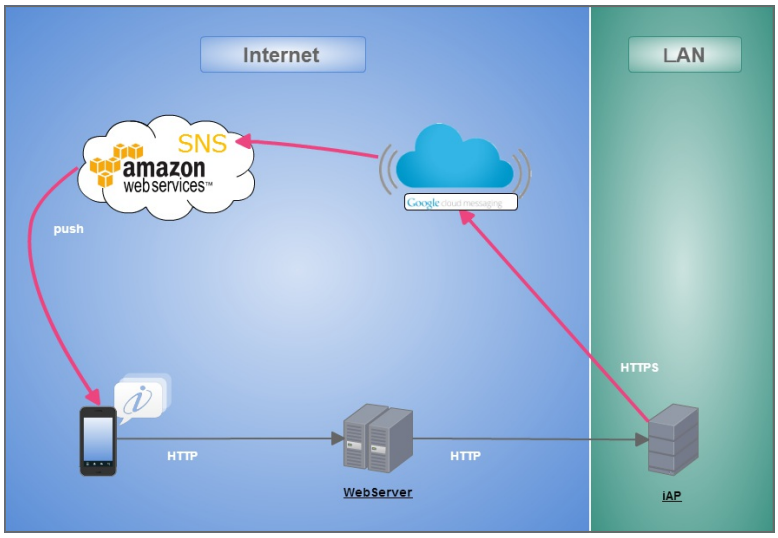


図 FCM＋Amazon SNS を使用する場合のサーバ構成例

コラム

Amazon SNSを組み合わせて使用することで、Push通知対象となっているデバイスのトークンリストや、その状態（有効または無効）をAmazon SNSで一覧できます。
また、AndroidとiOSを併用する場合、Amazon SNSで一元管理が可能です。

注意

中国国内ではFCM(Firebase Cloud Messaging)を利用できません。

FCMを使用する場合

FCMを使用しAndroidへの通知を行う場合、以下の設定を行ってください。

FCMの設定

FCMの設定を行います。

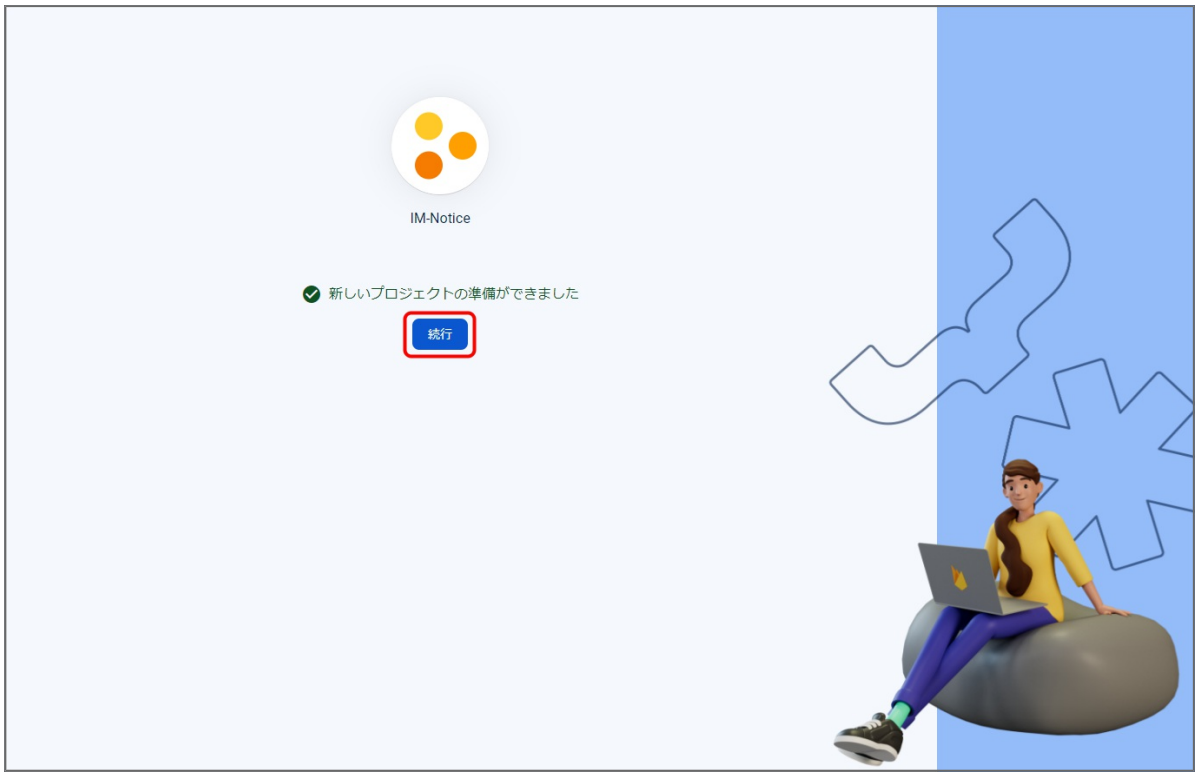
1. Firebase Console にログインしてください。
2. プロジェクトを作成します。

3. フォームに以下の内容を入力し、「続行」をクリックしてください。

項目	説明
プロジェクト名	任意の名前を入力してください。
プロジェクトID	任意のIDを入力してください。

4. 「Googleアナリティクス」を無効にし、「プロジェクトの作成」をクリックしてください。
（Googleアナリティクスを利用する場合は有効にして必要な設定をおこなってください。）

5. プロジェクトの準備ができたら、「続行」をクリックしてください。



6. 作成したプロジェクトをクリックしてください。

7. プロジェクトへAndroidアプリを追加します。「Androidアプリ追加」アイコンをクリックしてください。

8. フォームに以下の内容を入力し、「アプリを登録」をクリックしてください。

項目	説明
Android パッケージ名	jp.co.intra_mart.system.notice.android

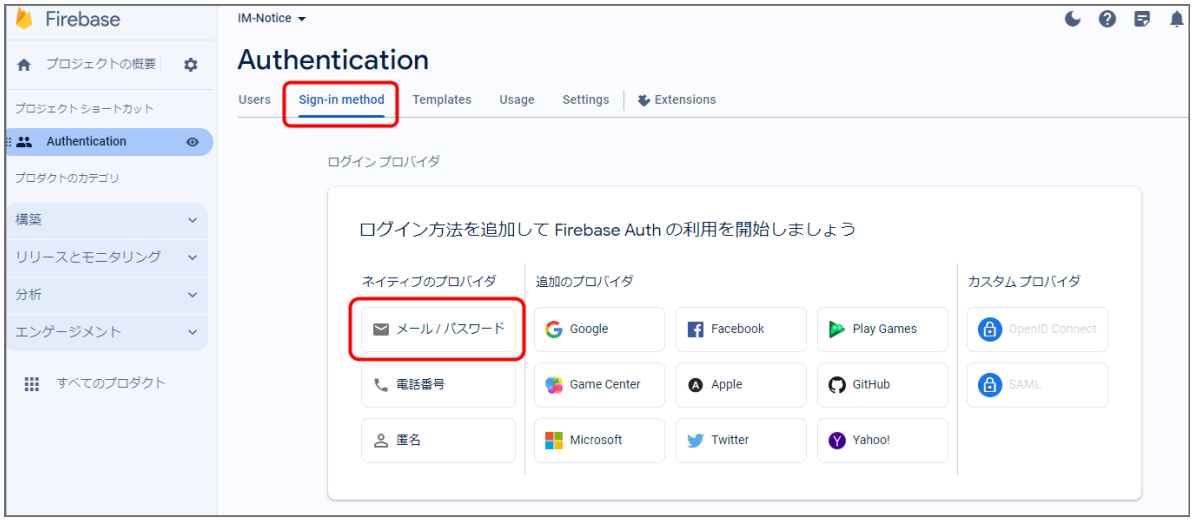
9. 「次へ」をクリックし、４まで進めてください。

10. 「コンソールに進む」をクリックしてください。

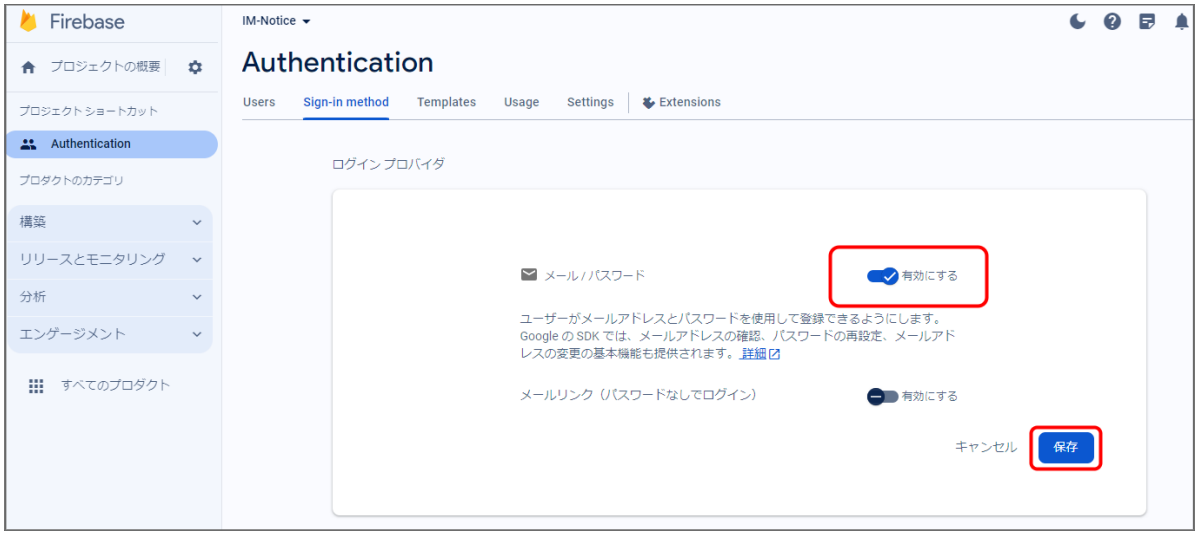
11. ウェブAPIキーを発行するために、Androidアプリにプロダクトを追加します。「Authentication」をクリックしてください。

12. 「始める」をクリックしてください。

13. 「Sign-in method」タブを選択し、ログイン プロバイダ「メール/パスワード」をクリックしてください。



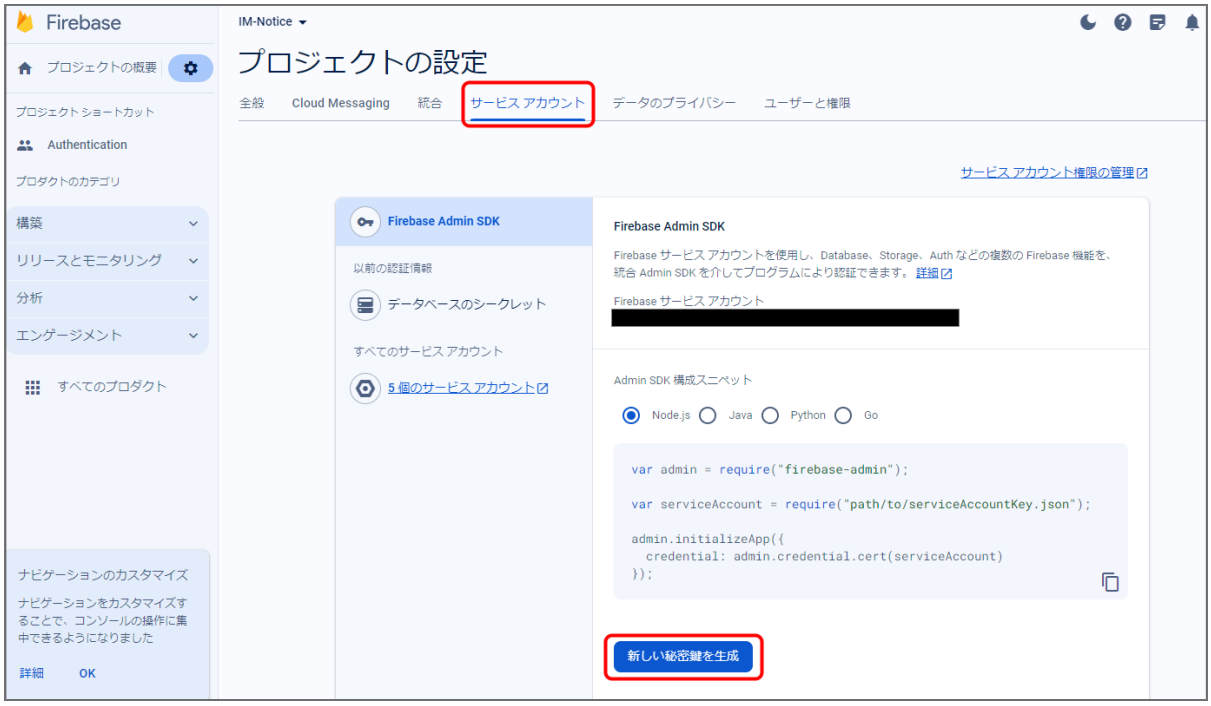
14. 「有効にする」をONに変更し、「保存」をクリックしてください。



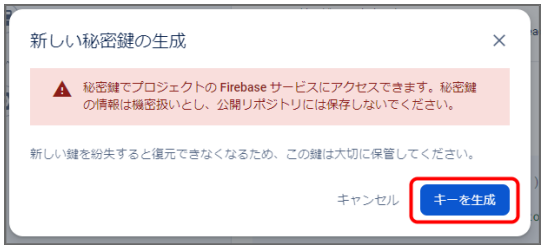
> **コラム**
>
> この項目の操作はウェブAPIキー発行のために行います。「メール/パスワード」の機能を利用しない場合は、ウェブAPIキー発行の発行が確認でき次第無効化してください。

15. サービスアカウントの秘密鍵を生成します。
プロジェクト概要の歯車アイコンをクリックし、「プロジェクトを設定」をクリックしてください。

16. 「サービスアカウント」タブを選択し、「新しい秘密鍵を生成」をクリックしてください。

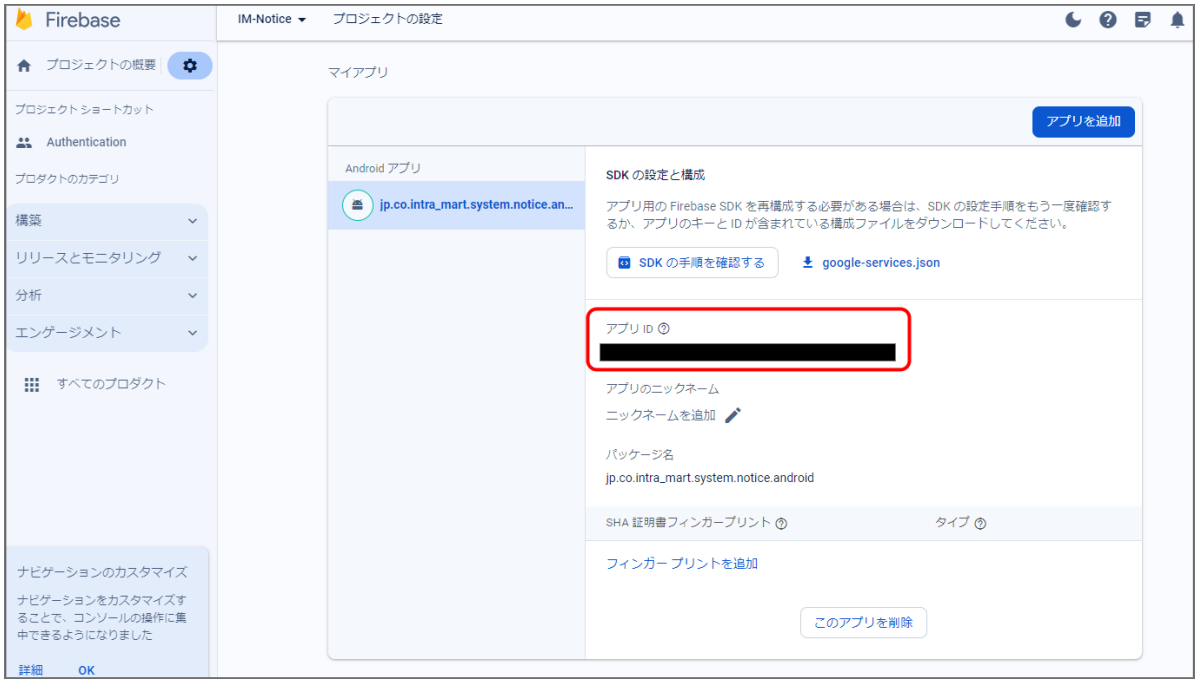


17. 表示されたメッセージを確認し、「キー生成」をクリックしてください。
秘密鍵が生成されると、JSON 形式のサービスアカウントの認証情報ファイルがダウンロードされます。
設定ファイルを編集する際に、設定ファイルと同じディレクトリに配置する必要がありますので、保持しておいてください。



18. 全般タブを選択し、プロジェクトのプロジェクトID、ウェブAPIキーを確認します。
設定ファイルを編集する際に必要な文字列ですので、控えておいてください。

19. マイアプリ - Androidアプリのアプリ IDを確認します。
設定ファイルを編集する際に必要な文字列ですので、控えておいてください。



20. クラウドメッセージングタブへ移動し、送信者IDを確認します。
設定ファイルを編集する際に必要な文字列ですので、控えておいてください。

IM-Notice Mobile設定ファイルの編集

IM-Juggling の <（プロジェクト名）/conf> 配下に出力されたim-notice-mobile-config.xmlを開き、以下の設定を行ってください。

- <proxy>
  1. Firebase Cloud Messagingへ接続するためにプロキシを使用する場合、設定を行ってください。

```
<proxy enable="true">
  <host></host>
  <port xsi:nil="true"></port> <!-- ポート番号を指定する場合は xsi:nil="true" を削除してください。 -->
  <username></username>
  <password></password>
  <workstation></workstation>
  <domain></domain>
</proxy>
```

- <fcm>
  1. <fcm>のname属性に任意の名前を設定してください。

```
<fcm name="fcm">
  ....
</fcm>
```

  2. <url>
     - https://fcm.googleapis.com/v1/projects/%REPLACE_WITH_PROJECT_ID%/messages:send を指定してください。
       ただし、%REPLACE_WITH_PROJECT_ID% はFirebaseプロジェクトのプロジェクトIDに置き換えて指定してください。
  3. <project-id>
     - Firebaseプロジェクトのプロジェクトを設定してください。
  4. <application-id>
     - Androidアプリのアプリを設定してください。
  5. <api-key>
     - Firebaseプロジェクトのウェブを設定してください。
  6. <credentials-file>
     - ダウンロードした JSON 形式のサービスアカウントの認証情報ファイルを im-notice-mobile-config.xml と同じディレクトリに配置してください。
       credentials-file には、認証情報ファイルのファイル名を設定してください。
  7. <sender-id>
     - クラウドメッセージングの送信者IDを設定してください。

> **コラム**
>
> 各項目の確認方法は、 *FCMの設定* を参照してください。

- <push type=”android” />
  1. engine属性に、<fcm>のname属性に設定した名前を指定してください。

```
<push type="android" engine="fcm" />
```

- <max-push-subject-length>
  1. Push通知受信時に表示するタイトルの最大文字数を設定してください。

```
<max-push-subject-length>30</max-push-subject-length>
```

- <max-push-body-length>
  1. Push通知受信時に表示する本文の最長文字数を設定してください。

```
<max-push-body-length>70</max-push-body-length>
```

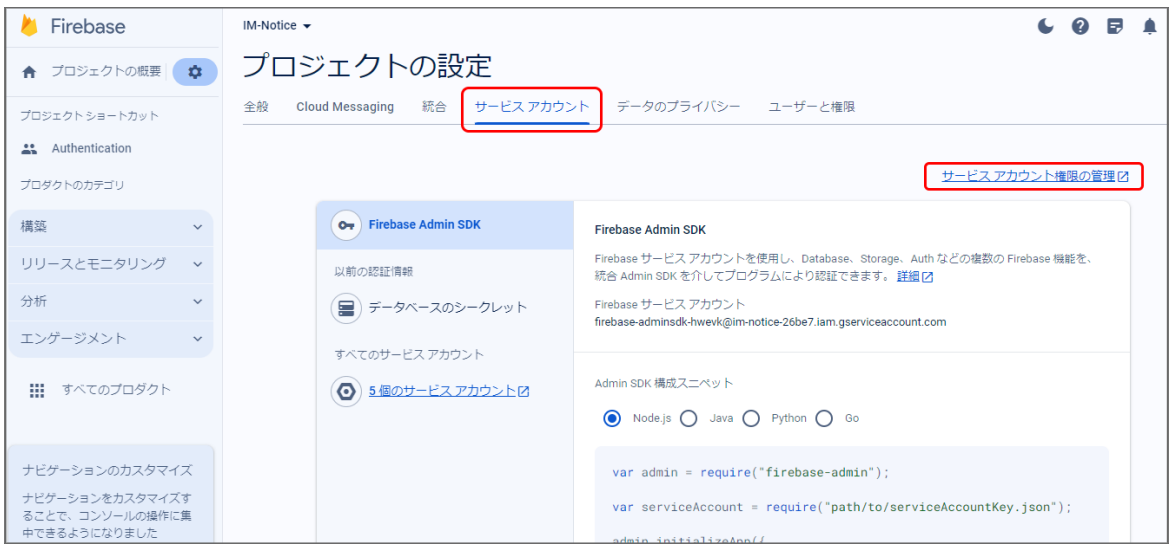
サービスアカウントの認証情報ファイルの更新

サービスアカウントの秘密鍵は、定期的にローテーションすることが推奨されています。
詳しくは「サービスアカウントキーのローテーション」を参照してください。
本項では、サービスアカウントの秘密鍵を新しく生成し、通知の送信に使用する認証情報ファイルを更新する手順を説明します。

1. サービスアカウントの新しい秘密鍵を作成します。
   Firebase Console にログインしてください。
2. 対象のプロジェクトをクリックしてください。

3. プロジェクト概要の歯車アイコンをクリックし、「プロジェクトを設定」をクリックしてください。

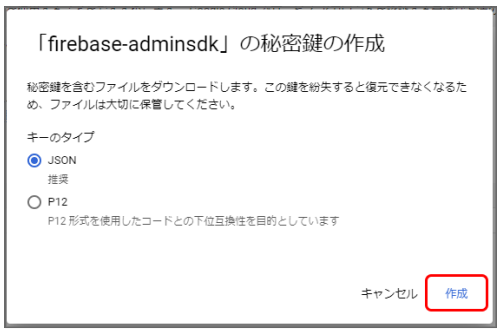
4. 「サービスアカウント」タブを選択し、「サービスアカウント権限の管理」をクリックしてください。



5. 「firebase-adminsdk」という名前のアカウントの行の縦三点リーダー「⋮」をクリックし、「鍵を管理」をクリックしてください。

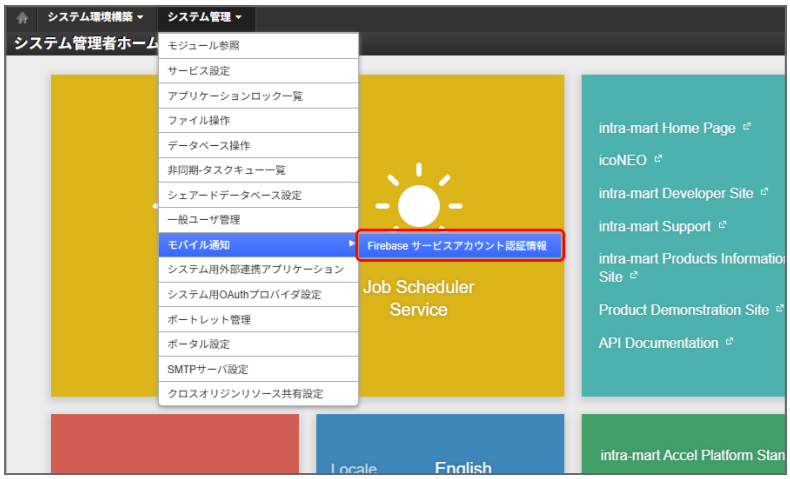
6. 「鍵を追加」をクリックし、「新しい鍵を作成」をクリックしてください。

7. 「作成」をクリックしてください。
秘密鍵が生成されると、JSON 形式のサービスアカウントの認証情報ファイルがダウンロードされます。
このファイルは、intra-mart Accel Platform にアップロードする必要がありますので、保持しておいてください。



8. サービスアカウントの認証情報ファイルを intra-mart Accel Platform にアップロードすることで、認証情報を更新できます。
intra-mart Accel Platform にシステム管理者でログインしてください。

9. 「システム管理」→「モバイル通知」→「Firebase サービスアカウント認証情報」をクリックしてください。



10. 認証情報ファイルアップロードをクリックしてください。

11. サービスアカウントの認証情報ファイルを選択し、更新をクリックしてください。

12. 更新確認ダイアログで決定をクリックすると更新できます。

FCM HTTP v1 API への移行

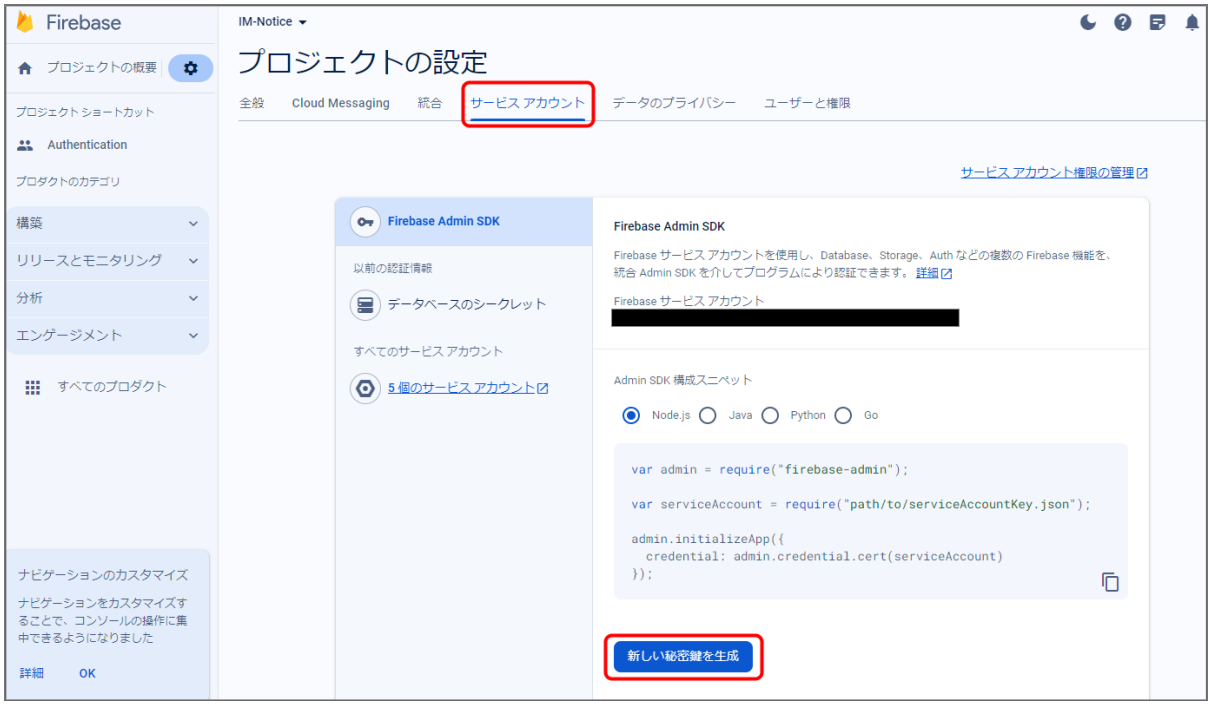
intra-mart Accel Platform 2023 Autumn(Hollyhock) 以前のバージョンでは、IM-Notice で FCM を使用して通知を送信する際に、FCM HTTP API (Legacy HTTP API) が使用されていました。
FCM HTTP API (Legacy HTTP API) は、2024年6月20日に廃止される予定です。
intra-mart Accel Platform 2024 Spring(Iris) 以降のバージョンから、FCM HTTP v1 API を使用して通知を送信可能です。
本項では、intra-mart Accel Platform 2023 Autumn(Hollyhock) 以前のバージョンで FCM を使用して通知を送信していた場合に、intra-mart Accel Platform 2024 Spring(Iris) 以降にバージョンアップ後、FCM HTTP v1 API が使用されるように設定する手順を説明します。

1. Firebase Console にログインしてください。
2. 対象のプロジェクトをクリックしてください。

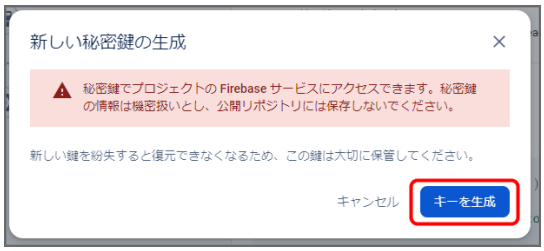
3. プロジェクト概要の歯車アイコンをクリックし、「プロジェクトを設定」をクリックしてください。

4. 「Cloud Messaging」タブをクリックし、Firebase Cloud Messaging API（V1）が有効であることを確認してください。
無効だった場合は有効にしてください。

5. 「サービスアカウント」タブを選択し、「新しい秘密鍵を生成」をクリックしてください。



6. 表示されたメッセージを確認し、「キー生成」をクリックしてください。
秘密鍵が生成されると、JSON 形式のサービスアカウントの認証情報ファイルがダウンロードされます。
設定ファイルを編集する際に、設定ファイルと同じディレクトリに配置する必要がありますので、保持しておいてください。



7. IM-Juggling の <（プロジェクト名）/conf> 配下に出力された im-notice-mobile-config.xml を開き、<server-key> を削除して <credentials-file> を追加してください。
<credentials-file> には、ダウンロードした JSON 形式のサービスアカウント認証情報ファイルのファイル名を設定してください。
認証情報ファイルを im-notice-mobile-config.xml と同じディレクトリに配置してください。
また、<url> には https://fcm.googleapis.com/v1/projects/%REPLACE_WITH_PROJECT_ID%/messages:send を設定してください。
ただし、%REPLACE_WITH_PROJECT_ID% はFirebaseプロジェクトのプロジェクトIDに置き換えて設定してください。

```
<fcm name="fcm">
    <url>https://fcm.googleapis.com/v1/projects/xxxxxxxxxxxx/messages:send</url>
    <project-id>xxxxxxxxxxxx</project-id>
    <application-id>xxxxxxxxxxxx</application-id>
    <api-key>xxxxxxxxxxxx</api-key>
    <!-- <server-key>xxxxxxxxxxxx</server-key> -->
    <credentials-file>xxxxxxxxxxxx.json</credentials-file>
    <sender-id>xxxxxxxxxxxx</sender-id>
</fcm>
```

8. WAR ファイルを出力して再デプロイすることで、FCM HTTP v1 API への移行が完了します。
   WAR ファイルの出力に関しては「*WAR ファイルの出力*」を、再デプロイに関しては「*WAR ファイルの再デプロイ*」を参照してください。

FCM+Amazon SNSを使用する場合

FCMとAmazon SNSを使用しAndroidへの通知を行う場合、以下の設定を行ってください。

FCM の設定

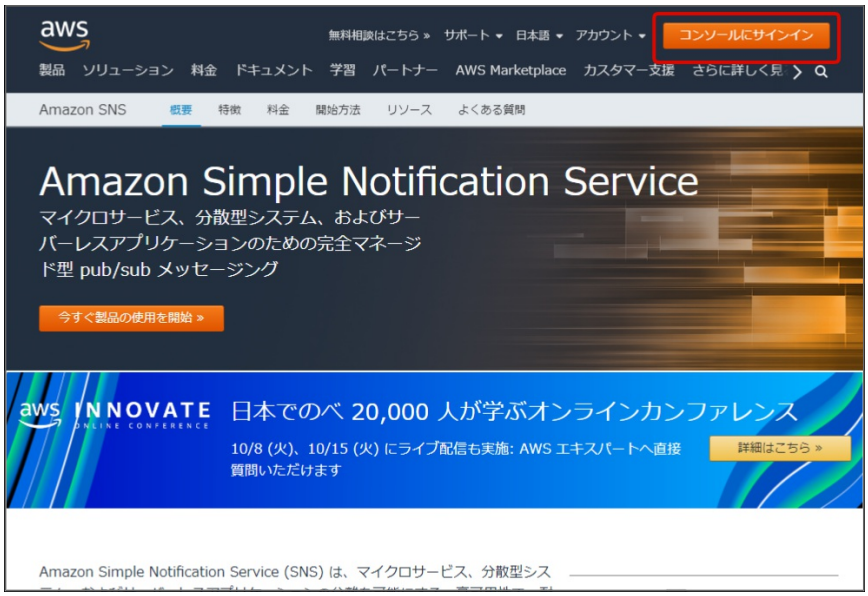
FCMの設定を行います。

> **コラム**
> FCMの設定方法は、 *FCMの設定* を参照してください。

Amazon SNS の設定

Amazon SNSの設定を行います。

1. Amazon SNS にサインインしてください。



2. リージョンを選択してください。

3. サービスを選択してください。

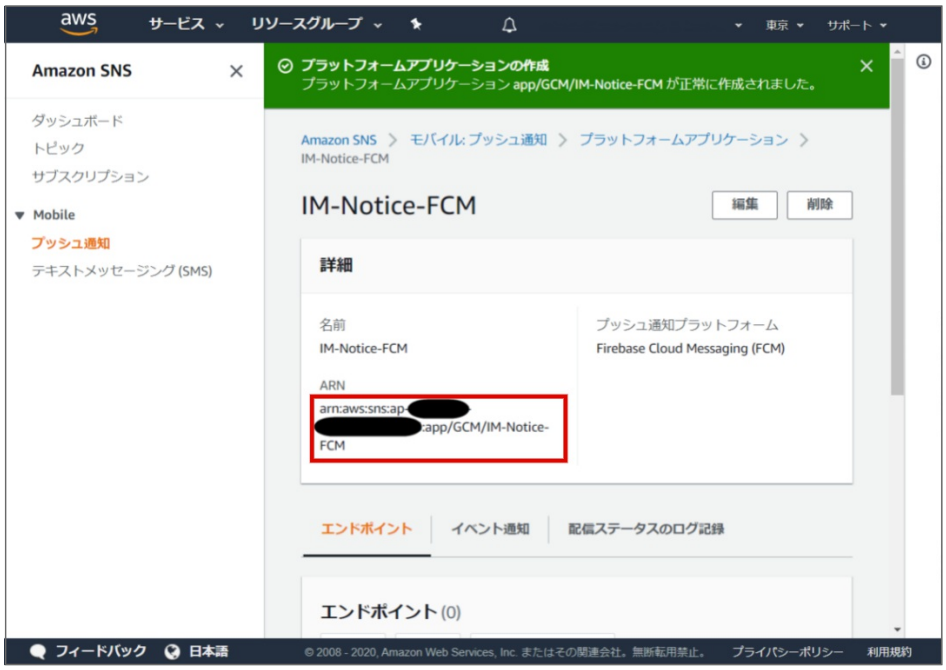
4. 「Mobile」-「プッシュ通知」から「プラットフォームアプリケーションの作成」をクリックしてください。

5. フォームに以下の内容を入力し、「プラットフォームアプリケーションの作成」をクリックしてください。

項目	説明
アプリケーション名	任意の名前を入力します。

プッシュ通知プラットフォーム	「Firebase Cloud Messaging(FCM)」を選択します。
認証方法	「トークン (推奨)」を選択します。
Service Json	*FCM の設定* でダウンロードした JSON 形式のサービスアカウントの認証情報ファイルを選択します。

6. アプリケーションが追加されました。
   「ARN」を確認します。
   設定ファイルを編集する際に必要な文字列ですので、控えておいてください。



ポリシーの作成

Amazon SNSに接続するために必要な権限を付与するための、IAMポリシーの作成を行います。
作成したポリシーは、 *アクセスキーとシークレットキーの作成* または *ロールの作成* に使用します。

> **コラム**
> ポリシーの作成方法は、 *ポリシーの作成* を参照してください。

アクセスキーとシークレットキーの作成

Amazon SNSへの接続に必要な、アクセスキーとシークレットキーの作成を行います。

> **コラム**
> アクセスキーとシークレットキーの作成方法は、 *アクセスキーとシークレットキーの作成* を参照してください。

ロールの作成

Amazon SNSへの接続に必要な、IAMロールの作成を行います。

> **コラム**
> ロールの作成方法は、 *ロールの作成* を参照してください。

IM-Notice Mobile設定ファイルの編集

IM-Juggling の <（プロジェクト名）/conf> 配下に出力されたim-notice-mobile-config.xmlを開き、以下の設定を行ってください。

- <proxy>
  1. Firebase Cloud MessagingやAmazon SNSへ接続するためにプロキシを使用する場合、設定を行ってください。

```
<proxy enable="true">
  <host></host>
  <port xsi:nil="true"></port> <!-- ポート番号を指定する場合は xsi:nil="true" を削除してください。 -->
  <username></username>
  <password></password>
  <workstation></workstation>
  <domain></domain>
</proxy>
```

- <asns>
  1. <asns>のname属性に任意の名前を設定してください。

```
<asns name="asns-fcm">
  ....
</asns>
```

  2. <endpoint>
     - Amazon SNSのリージョンに対応するEndpointを設定してください。

```
<endpoint>sns.ap-northeast-1.amazonaws.com</endpoint>
```

> **コラム**
> Endpointは AWS documentation - Regions and Endpoints で確認できます。
> リージョンが「Asia Pacific (Tokyo)」の場合、Endpointは「sns.ap-northeast-1.amazonaws.com」です。

  3. <platform-application-arn>
     - Application ARNを設定してください。

> **コラム**
> Application ARNの確認方法は、 *Amazon SNS の設定* を参照してください。

  4. <access-key>
     - 通知にアクセスキーとシークレットキーを使う場合、 *アクセスキーとシークレットキーの作成* で作成したアクセスキーを設定してください。
  5. <secret-key>
     - 通知にアクセスキーとシークレットキーを使う場合、 *アクセスキーとシークレットキーの作成* で作成したシークレットキーを設定してください。
  6. <iam-role-arn>

- 通知にロールを使う場合、 *ロールの作成* で作成したロールのARNを設定してください。

7. <fcm-project-id>
   - Firebaseプロジェクトのプロジェクトを設定してください。
8. <fcm-application-id>
   - Androidアプリのアプリを設定してください。
9. <fcm-api-key>
   - Firebaseプロジェクトのウェブを設定してください。
10. <fcm-sender-id>
    - Firebase クラウドメッセージングの送信者を設定してください。

> **コラム**
>
> プロジェクトID等Firebase設定情報の確認方法は、 *FCMの設定* を参照してください。

- <push type=”android” />
  1. engine属性に、<asns>のname属性に設定した名前を指定してください。

```
<push type="android" engine="asns-fcm" />
```

- <max-push-subject-length>
  1. Push通知受信時に表示するタイトルの最大文字数を設定してください。

```
<max-push-subject-length>30</max-push-subject-length>
```

- <max-push-body-length>
  1. Push通知受信時に表示する本文の最長文字数を設定してください。

```
<max-push-body-length>70</max-push-body-length>
```

Accel Platform Mobile

項目

- 概要
- 通知機能全般
  - モジュールの選択
- モバイル通知機能（iOS版）
  - P12証明書ファイルの取得
  - Amazon SNSの設定
  - ポリシーの作成
  - アクセスキーとシークレットキーの作成
  - ロールの作成
  - Accel Platform Mobile 設定ファイルの編集
  - P12証明書ファイルの更新
- モバイル通知機能（Android版）
  - FCMを使用する場合
    - FCMの設定
    - Accel Platform Mobile 設定ファイルの編集
    - サービスアカウントの認証情報ファイルの更新
    - FCM HTTP v1 API への移行
  - FCM+Amazon SNSを使用する場合
    - FCM の設定
    - Amazon SNS の設定
    - ポリシーの作成
    - アクセスキーとシークレットキーの作成
    - ロールの作成
    - Accel Platform Mobile 設定ファイルの編集

概要

Accel Platform Mobileは、 スマートフォンで intra-mart Accel Platform の機能を利用できるiOS/Android 対応スマートフォンアプリケーションです。
Amazon Simple Notification Service（以下、Amazon SNS） や Firebase Cloud Messaging（以下、FCM）のサービスを使用し、スマートフォンアプリへ通知を配信することが可能です。

> **注意**
> Accel Platform Mobile を利用するには、モバイルアプリケーション連携モジュールが必要です。

> **注意**
> Accel Platform Mobile は統合Windows認証環境では利用できません。

> **注意**
> IM-Notice と Accel Platform Mobile のモバイル通知の同時利用は非対応です。

> **コラム**
> 本ページで紹介しているAWSやFCM資材の作成方法は一例です。お客様の環境に合わせて対応してください。

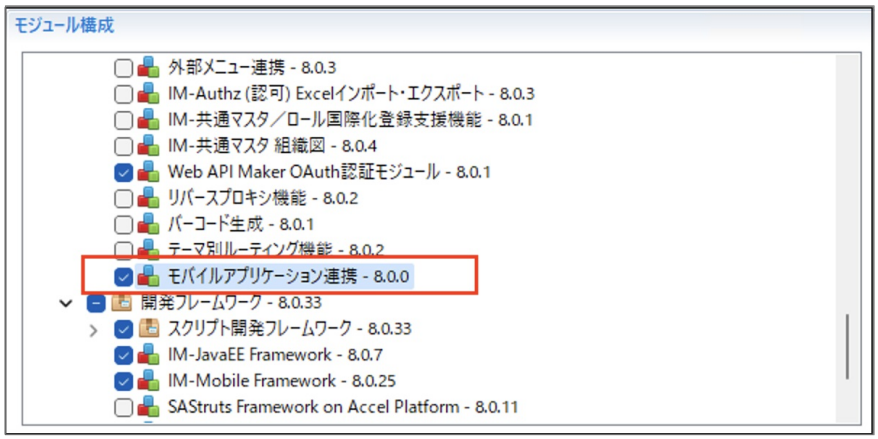
通知機能全般

通知機能全般を使用するための設定を行います。

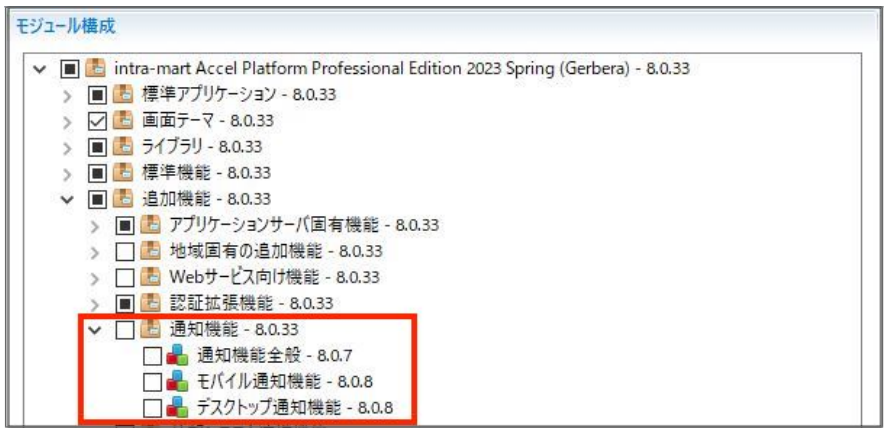
モジュールの選択

IM-Juggling でモジュールを選択します。

1. <（プロジェクト名）/juggling.im>の「ベースモジュール」タブから、モバイルアプリケーション連携を選択してください。



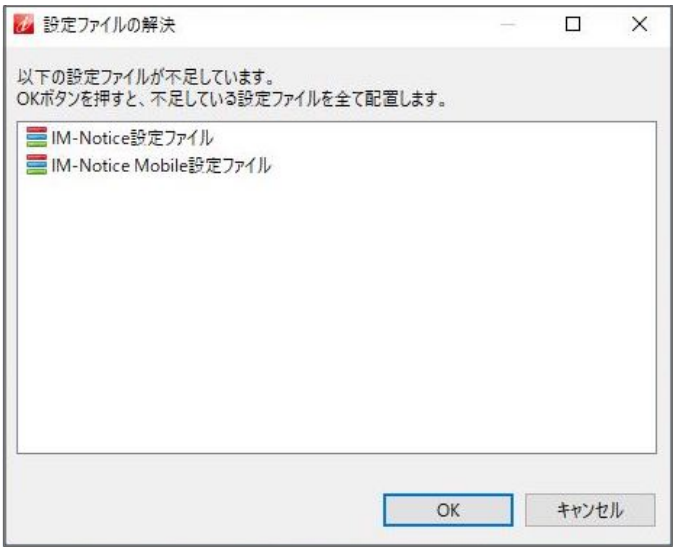
2. 「通知機能」を開いてください。



3. 通知機能全般とモバイル通知機能を選択します。

4. メッセージをクリックし、依存関係を解決してください。

5. 設定ファイルの解決を行うと、<（プロジェクト名）/conf> 配下に設定ファイルが出力されます。



モバイル通知機能（iOS版）

モバイル通知機能を使用するための設定を行います。 iOSへの通知を行うには、Amazon SNSを使用します。

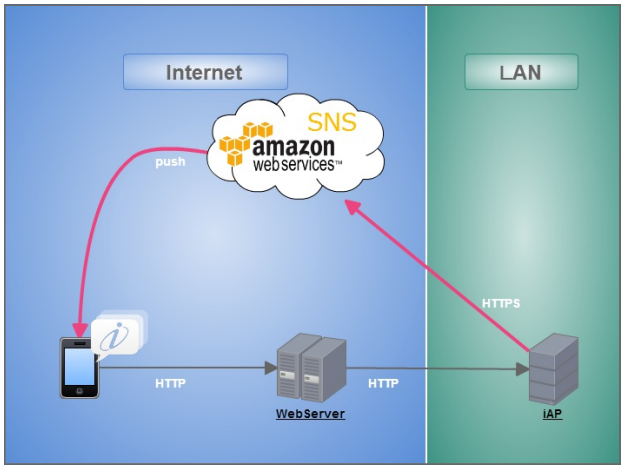


図 Amazon SNSを利用したサーバ構成例

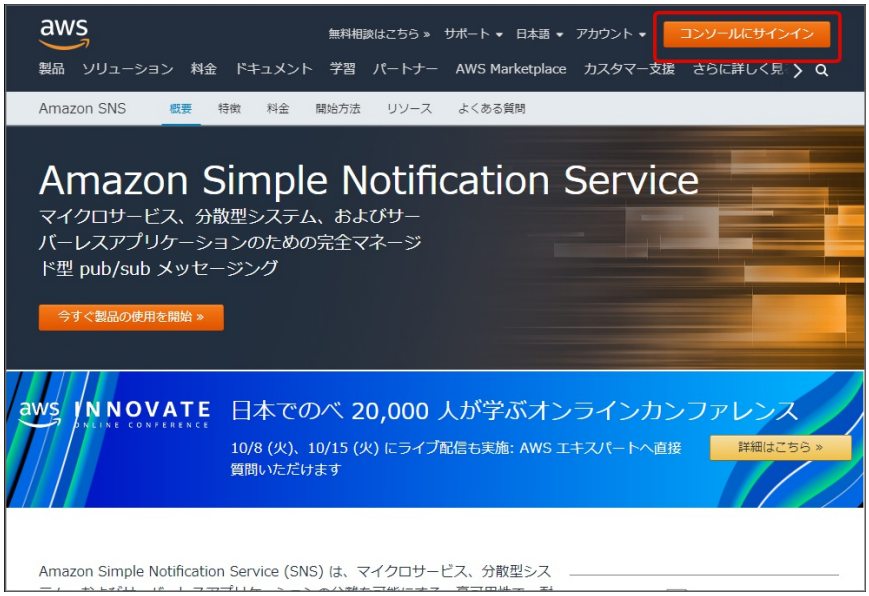
P12証明書ファイルの取得

P12証明書ファイル（Accel Platform Mobile Push Notifications Production.p12）を取得します。
弊社サイトの プロダクトファイルダウンロード よりライセンスキーを入力して取得してください。

Amazon SNSの設定

Amazon SNSの設定を行います。

1. Amazon SNS にサインインしてください。



2. リージョンを選択してください。

3. サービスを選択してください。

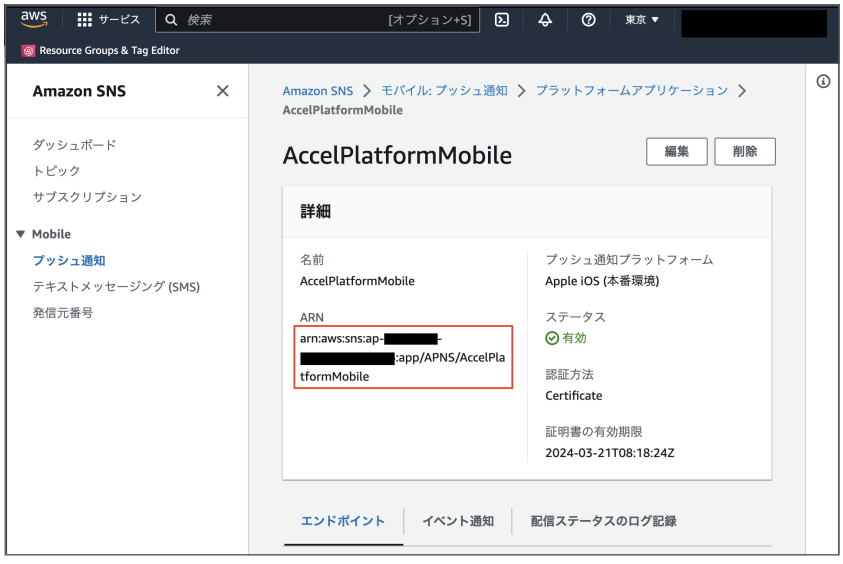
4. 「Mobile」-「プッシュ通知」から「プラットフォームアプリケーションの作成」をクリックしてください。

5. フォームに以下の内容を入力し、「プラットフォームアプリケーションの作成」をクリックしてください。

項目	説明
アプリケーション名	任意の名前を入力します。

プッシュ通知プラットフォーム	「Apple iOS/VoIP/Mac」を選択します。
プッシュ証明書タイプ	「iOS プッシュ証明書」を選択します。
ファイルの選択	「Accel Platform Mobile Push Notifications Production.p12」を添付します。
証明書のパスワード	「intra-mart」と入力し、「認証情報をファイルから読み込み」をクリックします。

6. アプリケーションが追加されました。
「ARN」を確認します。
設定ファイルを編集する際に必要な文字列ですので、控えておいてください。



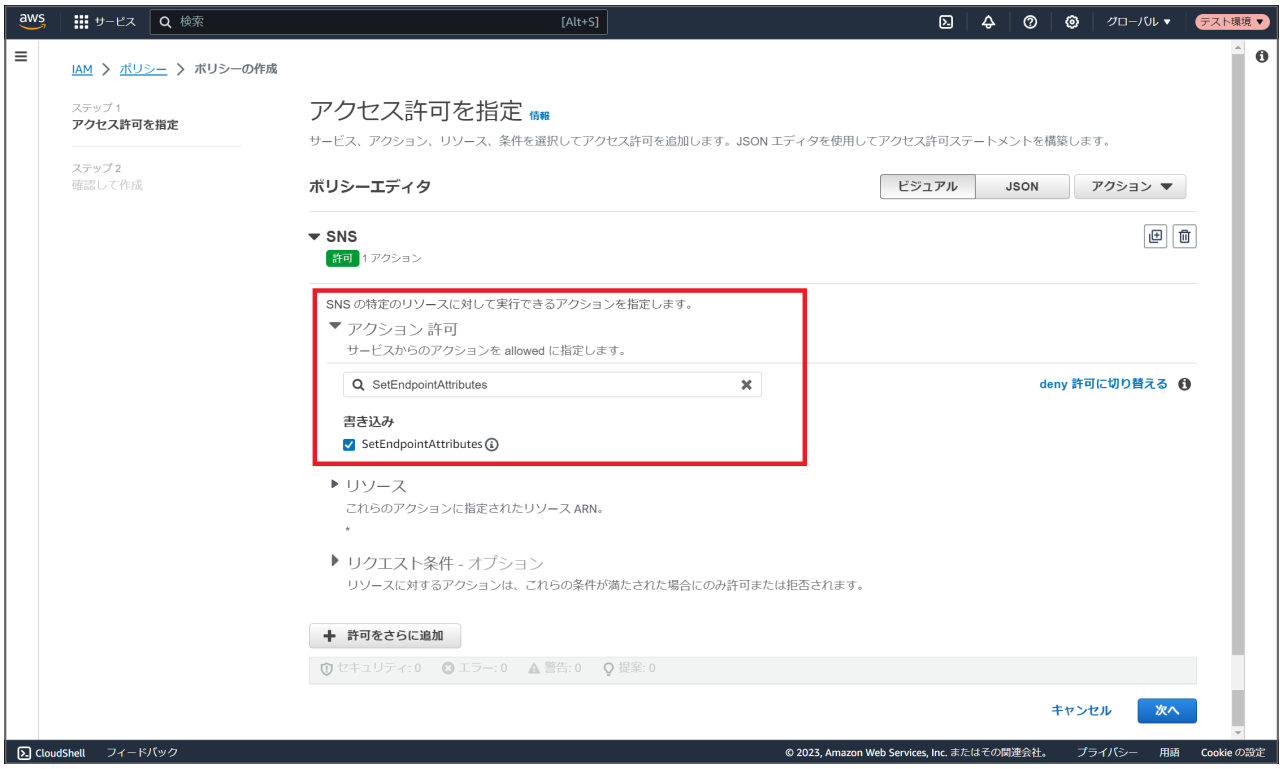
ポリシーの作成

Amazon SNSに接続するために必要な権限を付与するための、IAMポリシーの作成を行います。
作成したポリシーは、 *アクセスキーとシークレットキーの作成* または *ロールの作成* に使用します。

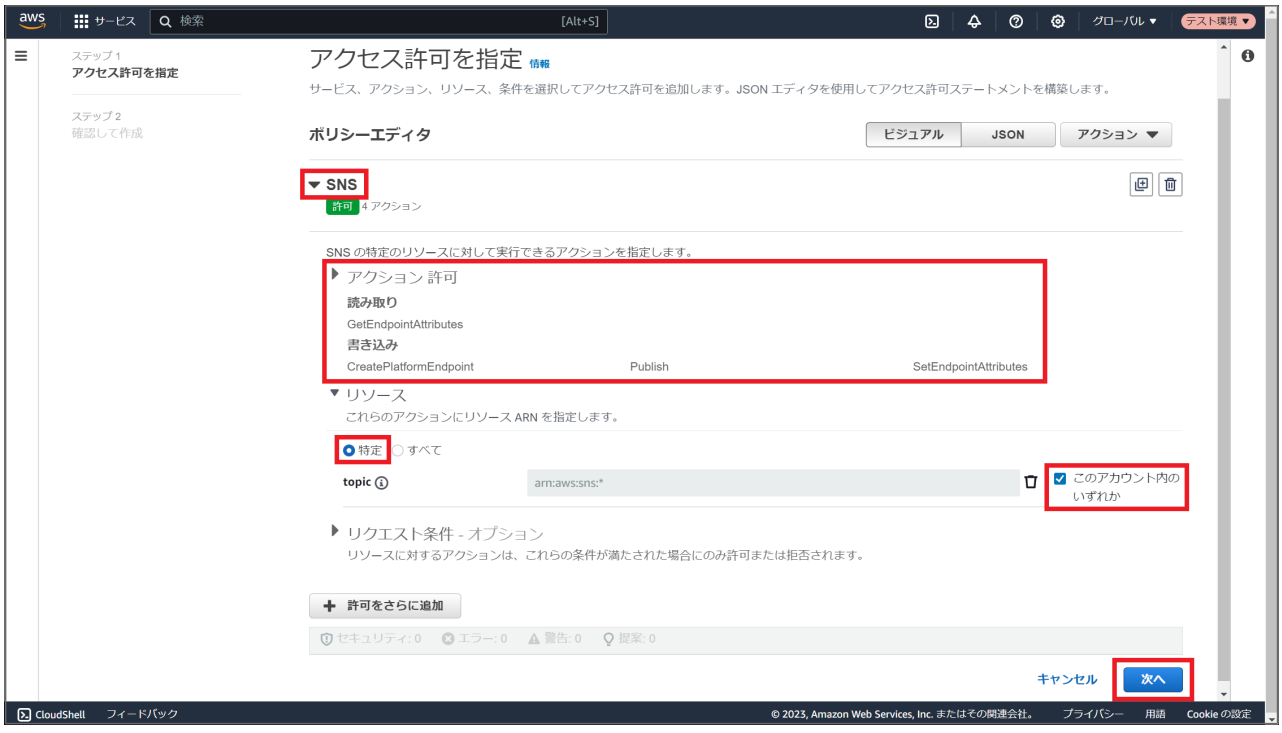
1. AWS Identity and Access Management (IAM) にサインインしてください。
2. 「ポリシー」を選択し、「ポリシーを作成」を押下してください。

3. 「サービスを選択」にて「SNS」を検索し、選択してください。

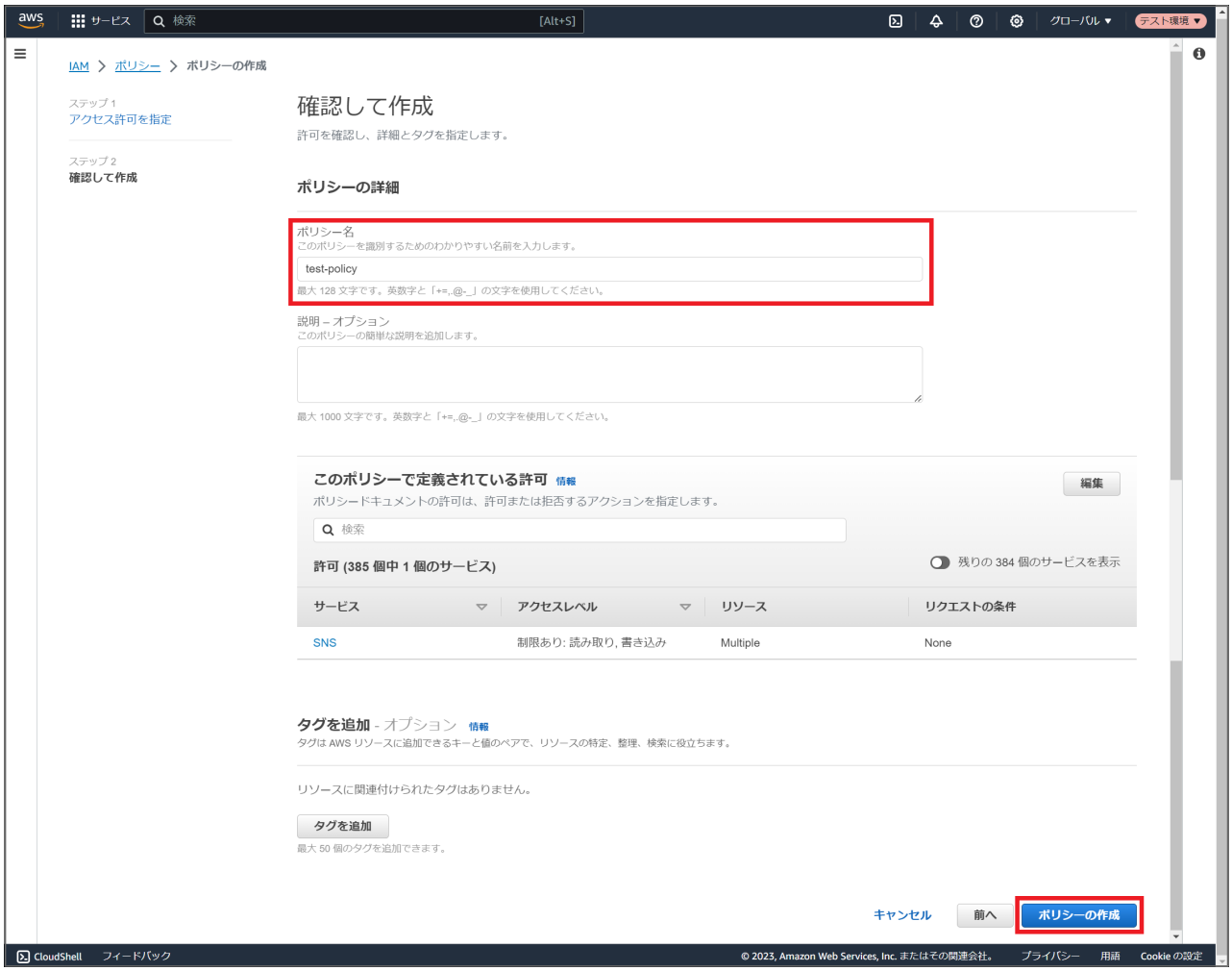
4. サービスを選択すると表示される「アクション許可」にて以下4つのアクションをそれぞれ検索し、チェックを入れてください。
   - SetEndpointAttributes
   - GetEndpointAttributes
   - Publish
   - CreatePlatformEndpoint



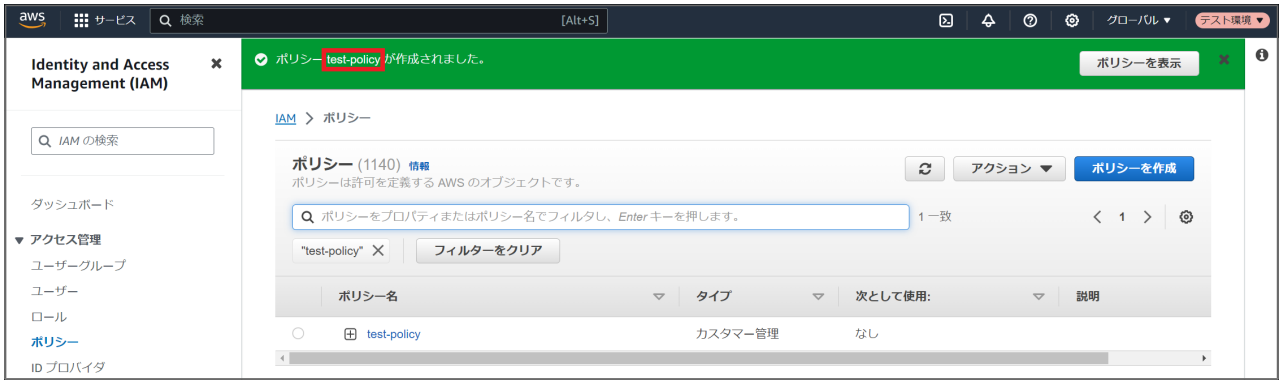
5. 選択したサービス、および、アクションが表示されていることを確認してください。
   その後、「リソース」にて「特定」を選択し、「このアカウント内のいずれか」にチェックを入れ、「次へ」を押下してください。



6. 「ポリシー名」に任意のポリシー名を入力し、「ポリシーの作成」を押下してください。



7. ポリシーが作成されました。
   作成されたポリシー名は *アクセスキーとシークレットキーの作成* や *ロールの作成* に使用しますので、控えておいてください。



アクセスキーとシークレットキーの作成

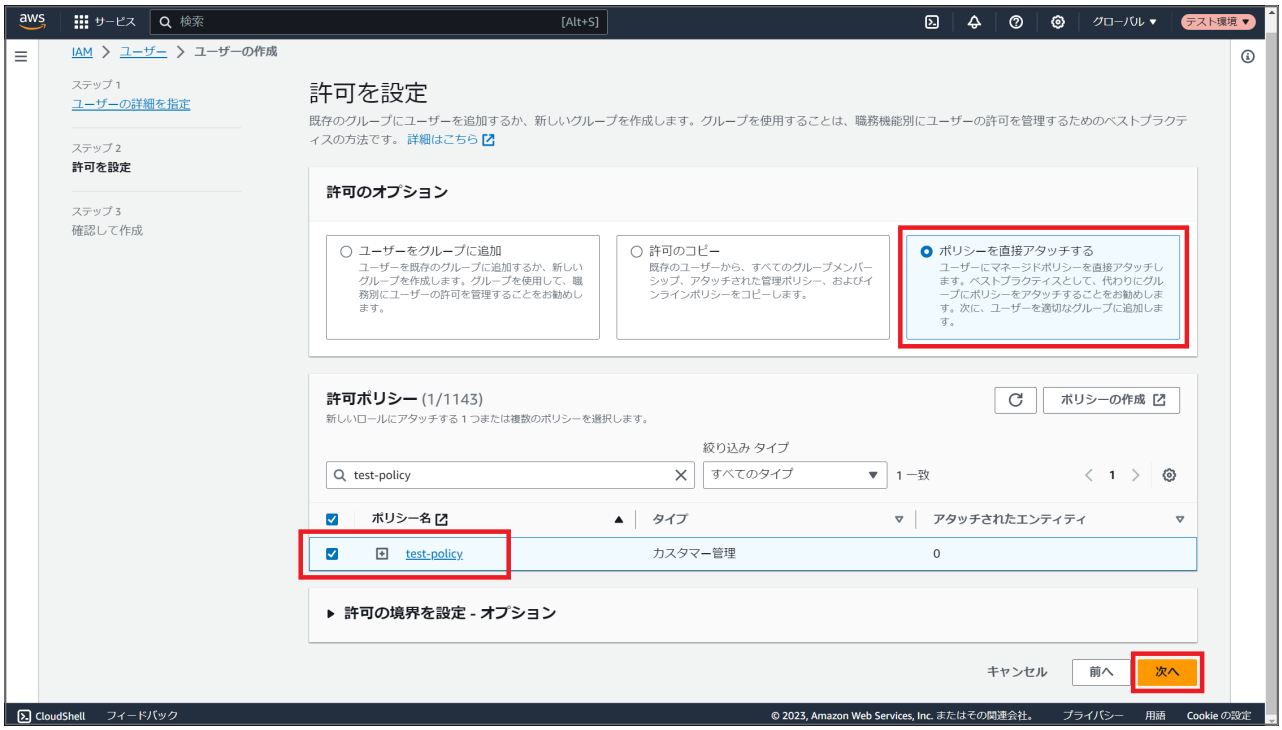
Amazon SNSへの接続に必要な、アクセスキーとシークレットキーの作成を行います。

> **コラム**
>
> アクセスキーとシークレットキー、または、ロールのうち、いずれか一方の作成が必要です。
> なお、ロールは、お客様環境がEC2インスタンス上で稼働している場合のみ利用できます。

1. AWS Identity and Access Management (IAM) にサインインしてください。
2. 「ユーザー」を選択し、「ユーザーの作成」を押下してください。

3. 「ユーザー名」に任意のユーザー名を入力し、「次へ」を押下してください。

4. 「許可のオプション」にて「ポリシーを直接アタッチする」を選択してください。
その後、 *ポリシーの作成* で作成したポリシー名にチェックを入れ、「次へ」を押下してください。

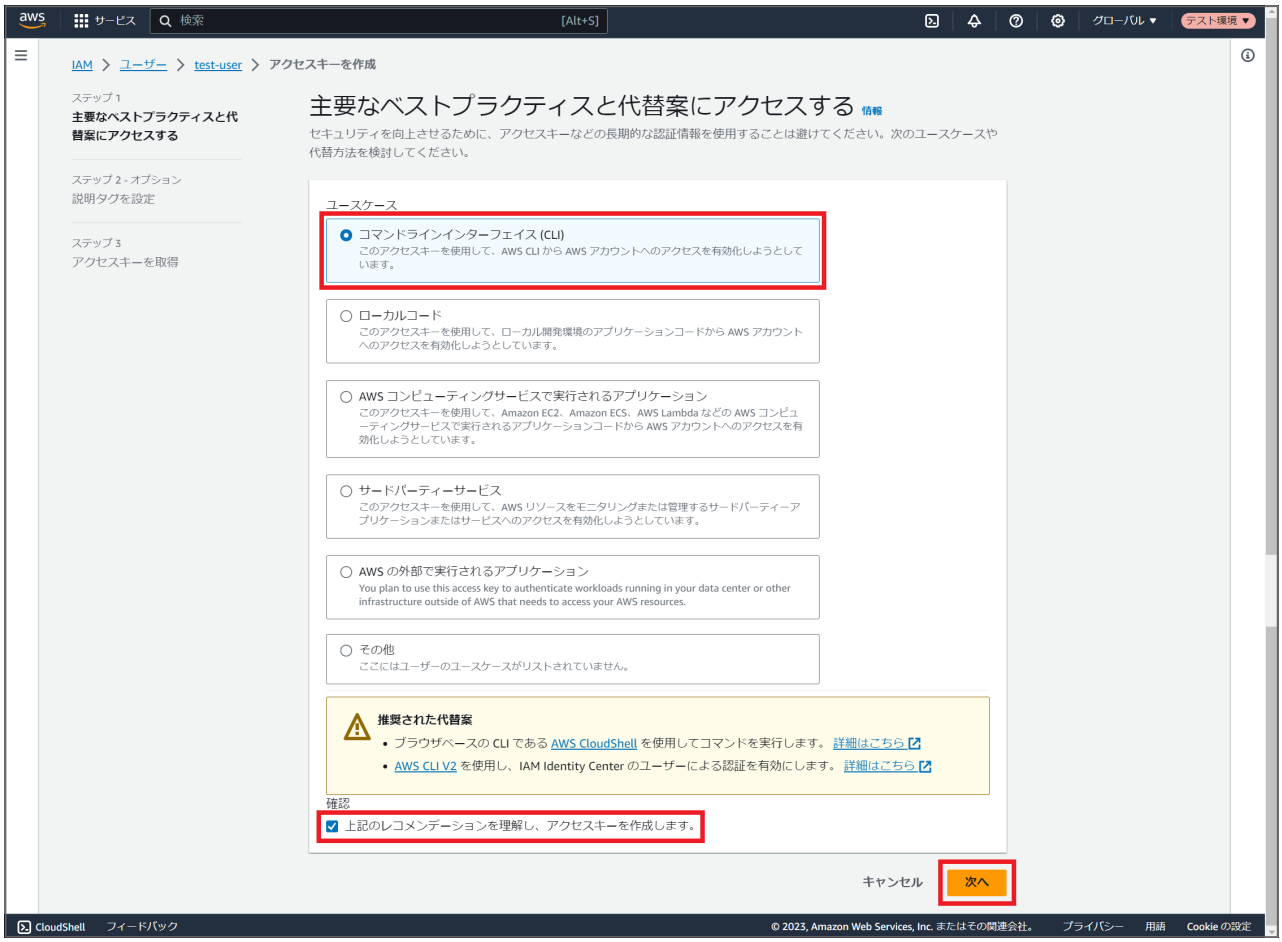


5. 「ユーザーの作成」を押下してください。

6. 作成したユーザー名を押下してください。

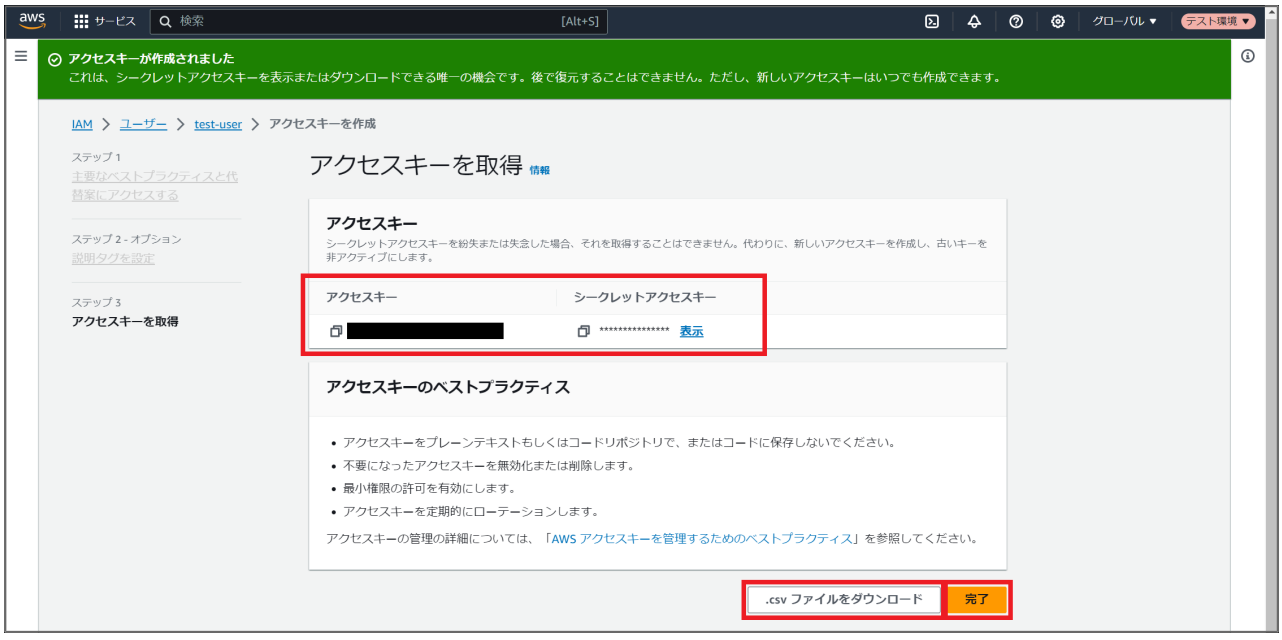
7. 「セキュリティ認証情報」タブを押下し、「アクセスキー」にて「アクセスキーを作成」を押下してください。

8. 「コマンドラインインターフェイス (CLI)」を選択し、「上記のレコメンデーションを理解し、アクセスキーを作成します。」にチェックを入れ、「次へ」を押下してください。



9. 「アクセスキーを作成」を押下してください。

10. 「アクセスキーを取得」画面でアクセスキーとシークレットアクセスキーを控えるか、「.csv ファイルをダウンロード」を押下してください。その後、「完了」を押下してください。

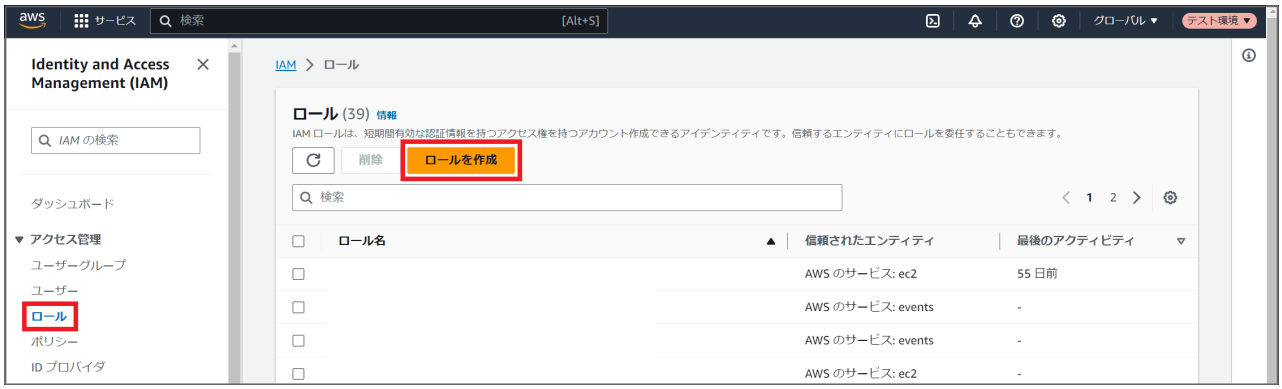


ロールの作成

Amazon SNSへの接続に必要な、IAMロールの作成を行います。

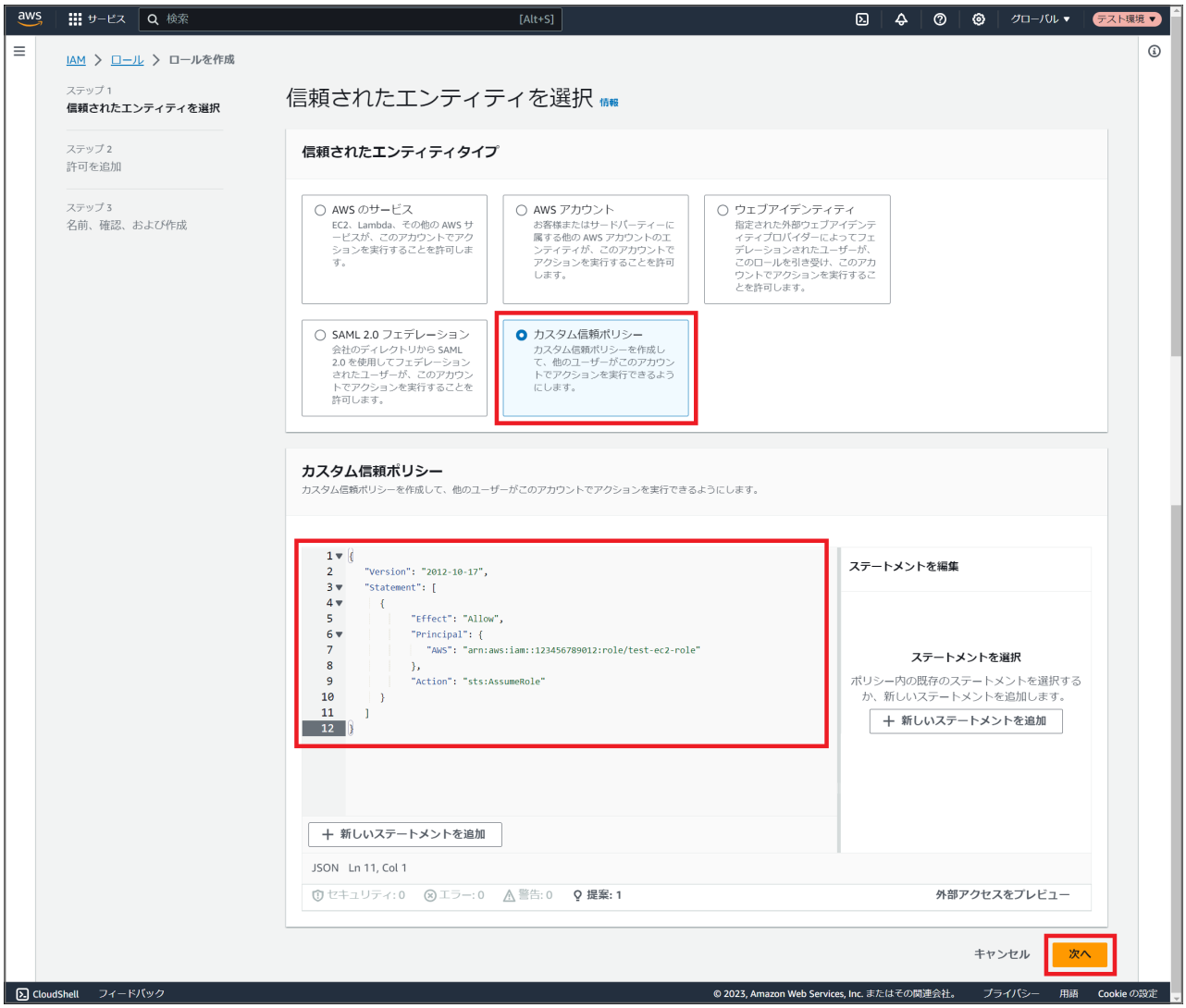
> **コラム**
>
> アクセスキーとシークレットキー、または、ロールのうち、いずれか一方の作成が必要です。
> なお、ロールは、お客様環境がEC2インスタンス上で稼働している場合のみ利用できます。

1. AWS Identity and Access Management (IAM) にサインインしてください。
2. 「ロール」を選択し、「ロールを作成」を押下してください。



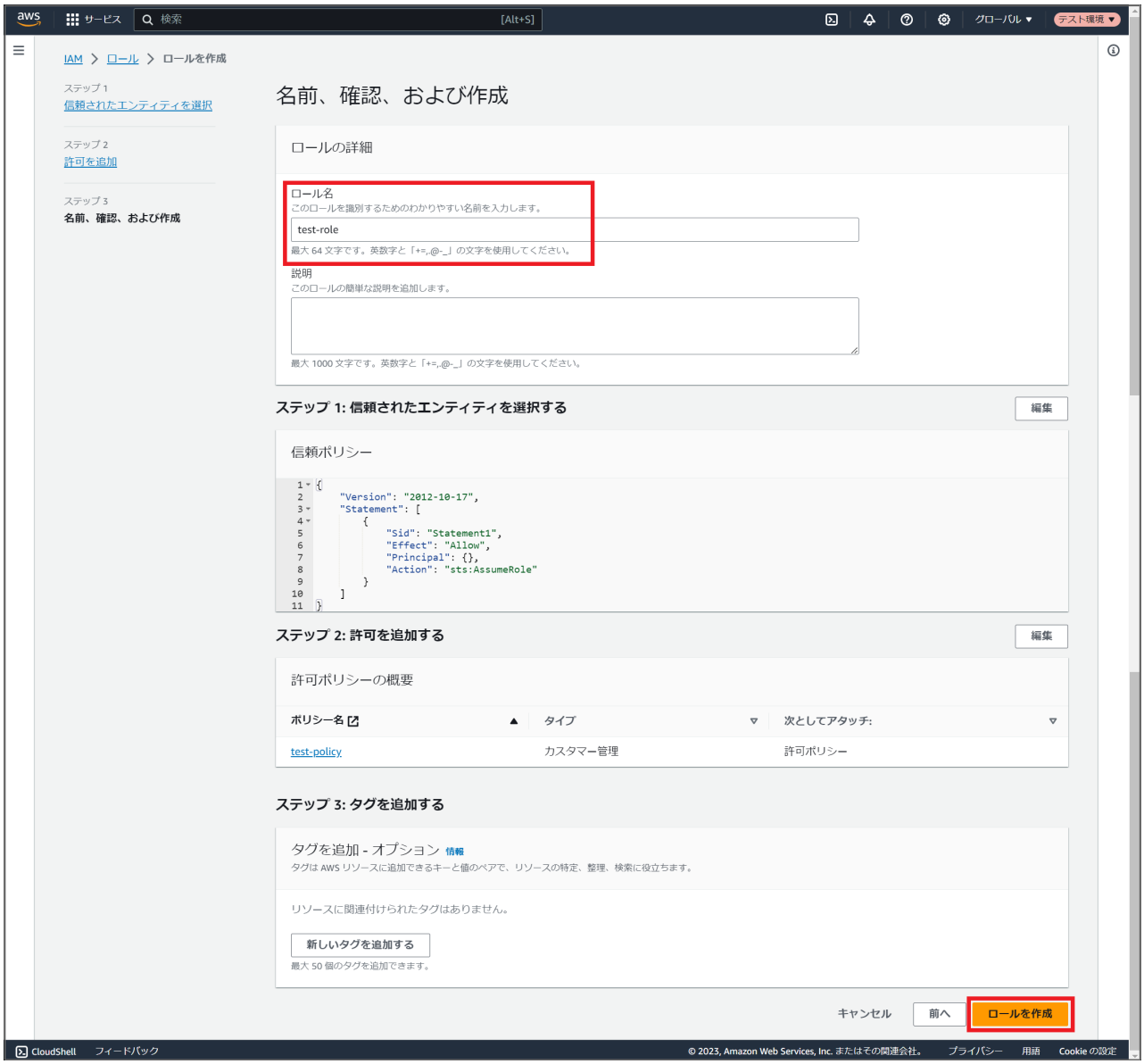
3. 「信頼されたエンティティタイプ」にて「カスタム信頼ポリシー」を選択し、「カスタム信頼ポリシー」に以下を入力してください。
`${AWSアカウントID}` と `${EC2インスタンスにアタッチされたロール名}` の箇所は、お客様の環境に合わせて変更してください。
その後、「次へ」を押下してください。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
        "Effect": "Allow",
        "Principal": {
          "AWS": "arn:aws:iam::${AWSアカウントID}:role/${EC2インスタンスにアタッチされたロール名}"
        },
        "Action": "sts:AssumeRole"
    }
  ]
}
```

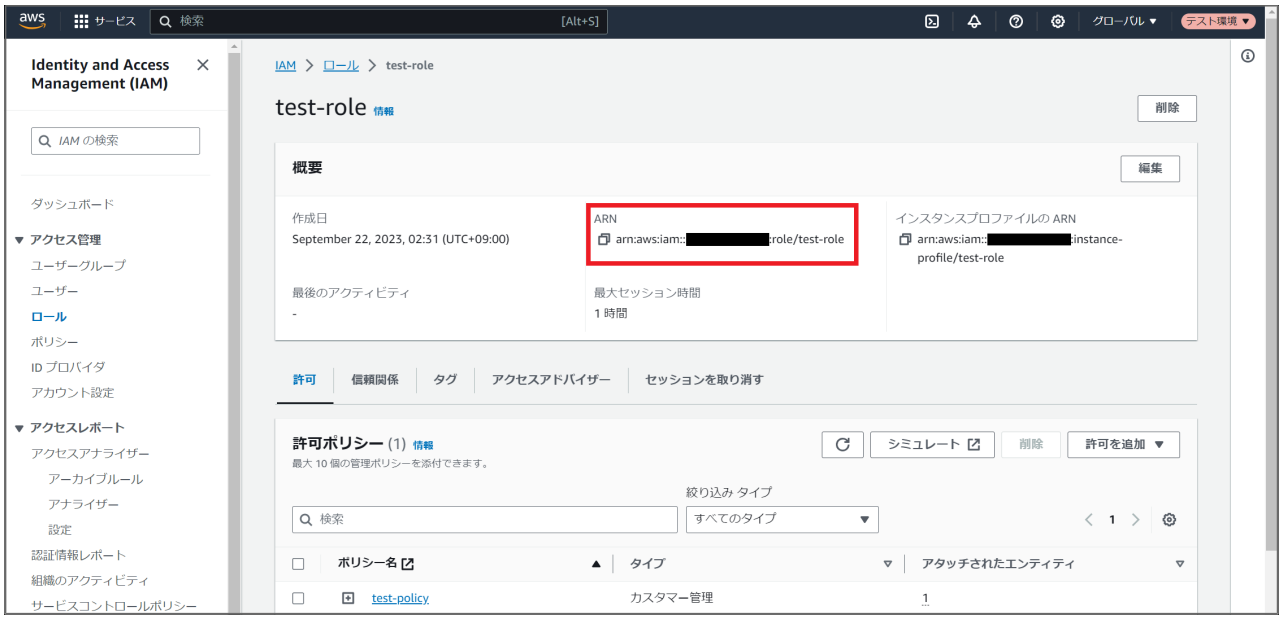


4. 「許可ポリシー」にて、 *ポリシーの作成* で作成したポリシー名にチェックを入れ、「次へ」を押下してください。

5. 「ロール名」に任意のロール名を入力し、「ロールを作成」を押下してください。



6. ロールが作成されました。
作成されたロールのARNは設定ファイルの編集に使用しますので、控えておいてください。



Accel Platform Mobile 設定ファイルの編集

IM-Juggling の <（プロジェクト名）/conf> 配下に出力されたim-notice-mobile-config.xmlを開き、以下の設定を行ってください。

- <proxy>

1. Amazon SNSへ接続するためにプロキシを使用する場合、設定を行ってください。

```
<proxy enable="true">
  <host></host>
  <port xsi:nil="true"></port> <!-- ポート番号を指定する場合は xsi:nil="true" を削除してください。 -->
  <username></username>
  <password></password>
  <workstation></workstation>
  <domain></domain>
</proxy>
```

- <asns>
  1. <asns>のname属性に任意の名前を設定してください。

```
<asns name="asns-apns">
  ....
</asns>
```

  2. <endpoint>
     - Amazon SNSのリージョンに対応するEndpointを設定してください。

```
<endpoint>sns.ap-northeast-1.amazonaws.com</endpoint>
```

> **コラム**
>
> Endpointは AWS documentation - Regions and Endpoints で確認できます。
> リージョンが「Asia Pacific (Tokyo)」の場合、Endpoint は「sns.ap-northeast-1.amazonaws.com」です。

  3. <platform-application-arn>
     - *Amazon SNSの設定* で確認したApplication ARNを設定してください。

```
<platform-application-arn>arn:aws:sns:ap-northeast-1:XXXXXXXXXXXX:app/APNS/AccelPlatformMobile</platform-application-arn>
```

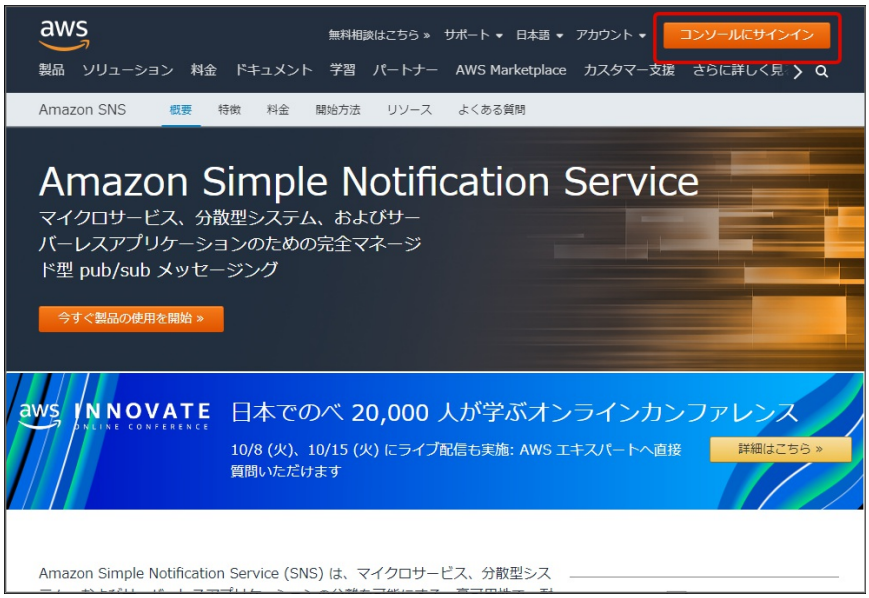
  4. <access-key>
     - 通知にアクセスキーとシークレットキーを使う場合、 *アクセスキーとシークレットキーの作成* で作成したアクセスキーを設定してください。
  5. <secret-key>
     - 通知にアクセスキーとシークレットキーを使う場合、 *アクセスキーとシークレットキーの作成* で作成したシークレットキーを設定してください。
  6. <iam-role-arn>
     - 通知にロールを使う場合、 *ロールの作成* で作成したロールのARNを設定してください。
- <push type=”ios” />
  1. engine属性に、<asns>のname属性に設定した名前を指定してください。

```
<push type="ios" engine="asns-apns" />
```

P12証明書ファイルの更新

P12証明書ファイル（Accel Platform Mobile Push Notifications Production.p12）の有効期限が切れてしまった、または新しいP12証明書ファイルが公開された場合、更新を行います。 有効期限が過ぎてしまったP12証明書ファイルを利用した場合、通知が届かなくなるため、有効期限が切れる前に新しいP12証明書ファイルに更新してください。

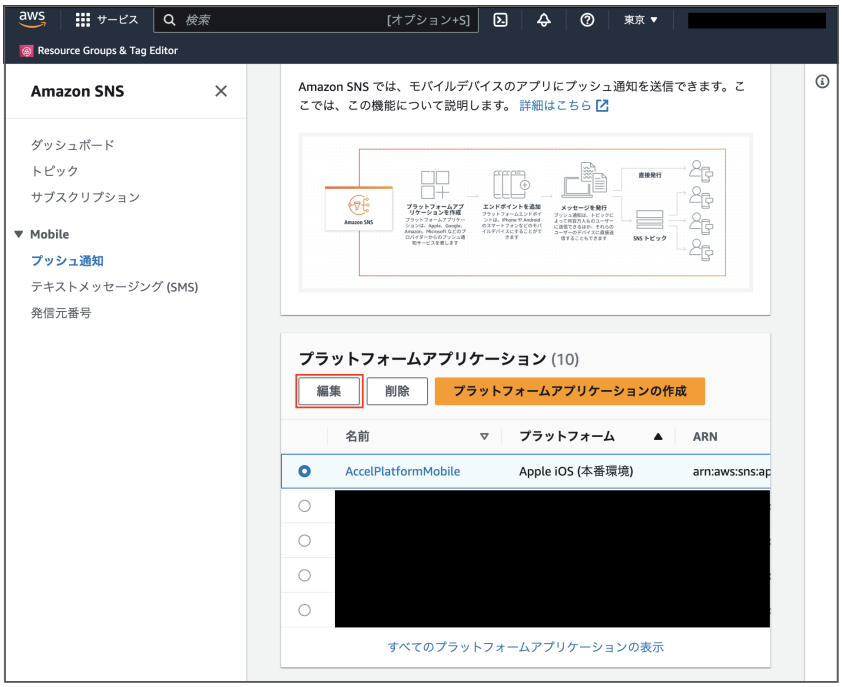
1. 弊社サイトのダウンロード ライブラリから最新のP12証明書ファイルを取得してください。
2. Amazon SNSにサインインしてください。



3. リージョンを選択してください。

4. サービスを選択してください。

5. 「Mobile」ー「プッシュ通知」からアプリケーションを選択し、「編集」をクリックします。



6. フォームに以下の内容を入力し、「変更の保存」をクリックしてください。

プッシュ証明書タイプ	「iOS プッシュ証明書」を選択します。
ファイルの選択	「Accel Platform Mobile Push Notifications Production.p12」を添付します。
証明書のパスワード	「intra-mart」と入力し、「認証情報をファイルから読み込み」をクリックします。

モバイル通知機能（Android版）

モバイル通知機能を使用するための設定を行います。 Androidへの通知を行うには、以下の2種類の構成が利用できます。

- FCM を使用する場合の設定方法は、 *FCMを使用する場合* を参照してください。

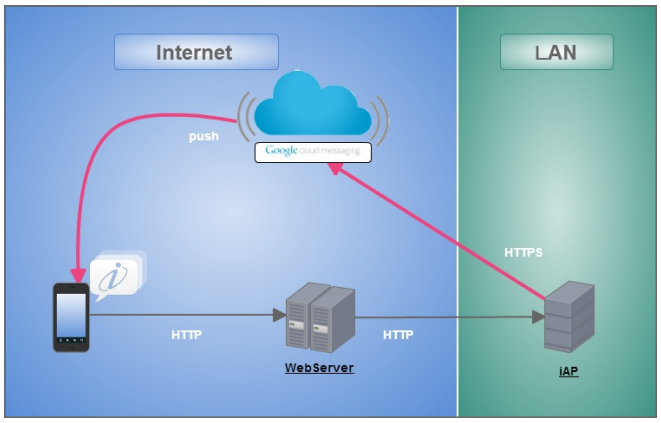


図 FCM を使用する場合のサーバ構成例

- FCM＋Amazon SNSを使用する場合の設定方法は、 *FCM+Amazon SNSを使用する場合* を参照してください。

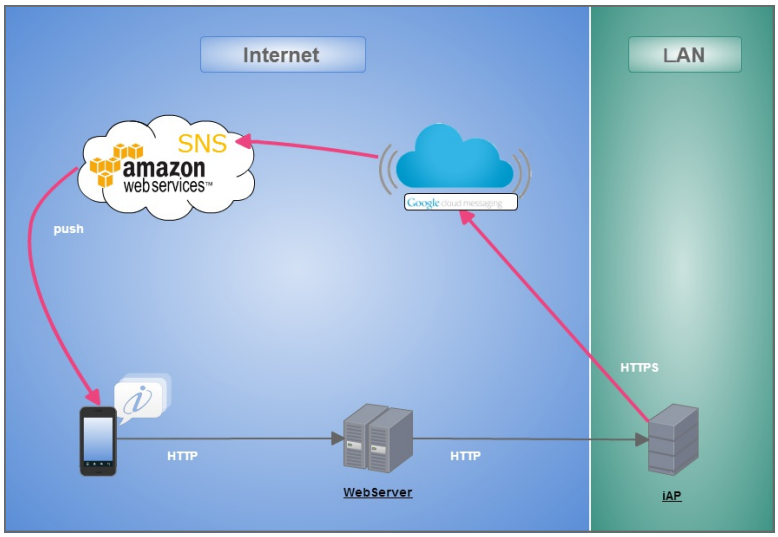


図 FCM＋Amazon SNS を使用する場合のサーバ構成例

コラム

Amazon SNSを組み合わせて使用することで、Push通知対象となっているデバイスのトークンリストや、その状態（有効または無効）をAmazon SNSで一覧できます。
また、AndroidとiOSを併用する場合、Amazon SNSで一元管理が可能です。

注意

中国国内ではFCM(Firebase Cloud Messaging)を利用できません。

FCMを使用する場合

FCMを使用しAndroidへの通知を行う場合、以下の設定を行ってください。

FCMの設定

FCMの設定を行います。

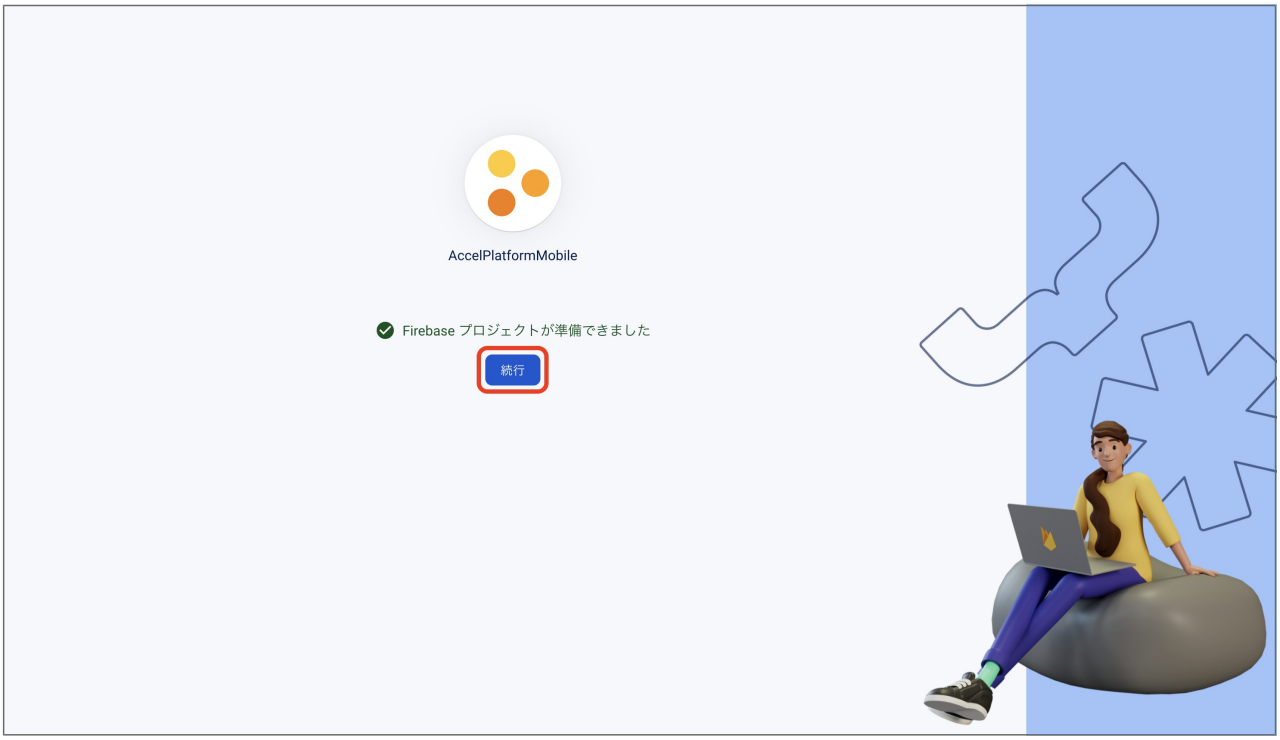
1. Firebase Console にログインしてください。
2. プロジェクトを作成します。

3. フォームに以下の内容を入力し、「続行」をクリックしてください。

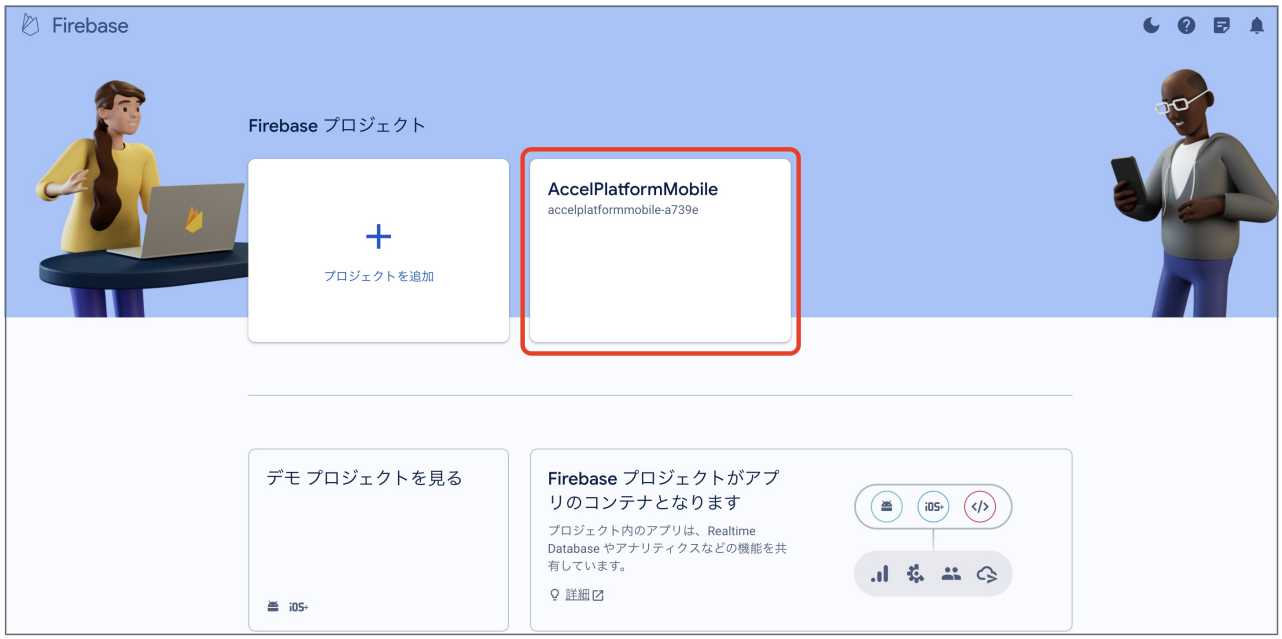
項目	説明
プロジェクト名	任意の名前を入力してください。
プロジェクトID	任意のIDを入力してください。

4. 「Googleアナリティクス」を無効にし、「プロジェクトの作成」をクリックしてください。
（Googleアナリティクスを利用する場合は有効にして必要な設定をおこなってください。）

5. プロジェクトの準備ができたら、「続行」をクリックしてください。



6. 作成したプロジェクトをクリックしてください。



7. プロジェクトへAndroidアプリを追加します。「Androidアプリ追加」アイコンをクリックしてください。

8. フォームに以下の内容を入力し、「アプリを登録」をクリックしてください。

項目	説明
Android パッケージ名	jp.co.intra_mart.AccelPlatformMobile

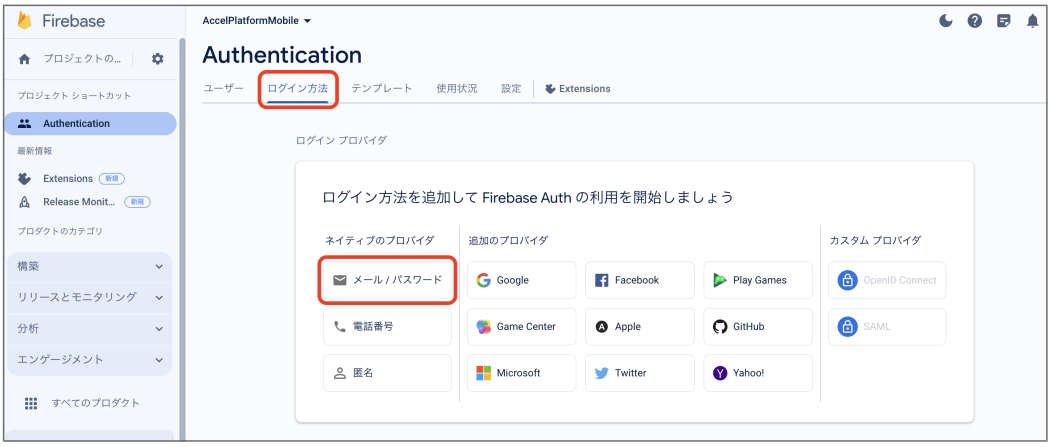
9. 「次へ」をクリックし、４まで進めてください。

10. 「コンソールに進む」をクリックしてください。

11. ウェブAPIキーを発行するために、Androidアプリにプロダクトを追加します。「Authentication」をクリックしてください。

12. 「始める」をクリックしてください。

13. 「ログイン方法」タブを選択し、ログイン プロバイダ「メール/パスワード」をクリックしてください。

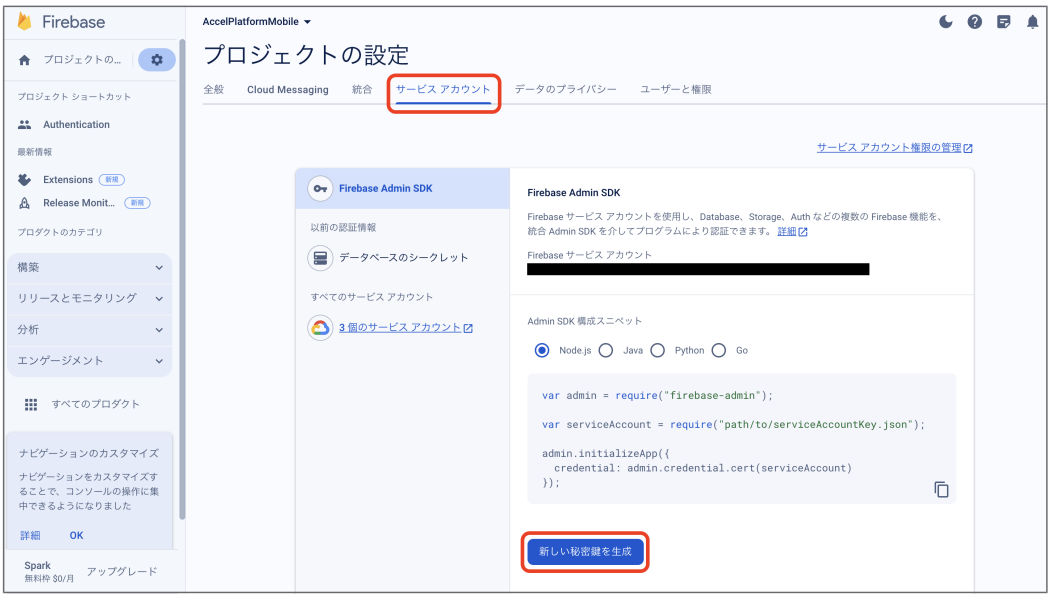


14. 「有効にする」をONに変更し、「保存」をクリックしてください。

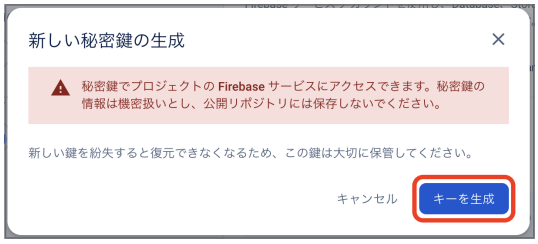
> **コラム**
>
> この項目の操作はウェブAPIキー発行のために行います。「メール/パスワード」の機能を利用しない場合は、ウェブAPIキー発行の発行が確認でき次第無効化してください。

15. サービスアカウントの秘密鍵を生成します。
プロジェクト概要の歯車アイコンをクリックし、「プロジェクトを設定」をクリックしてください。

16. 「サービスアカウント」タブを選択し、「新しい秘密鍵を生成」をクリックしてください。

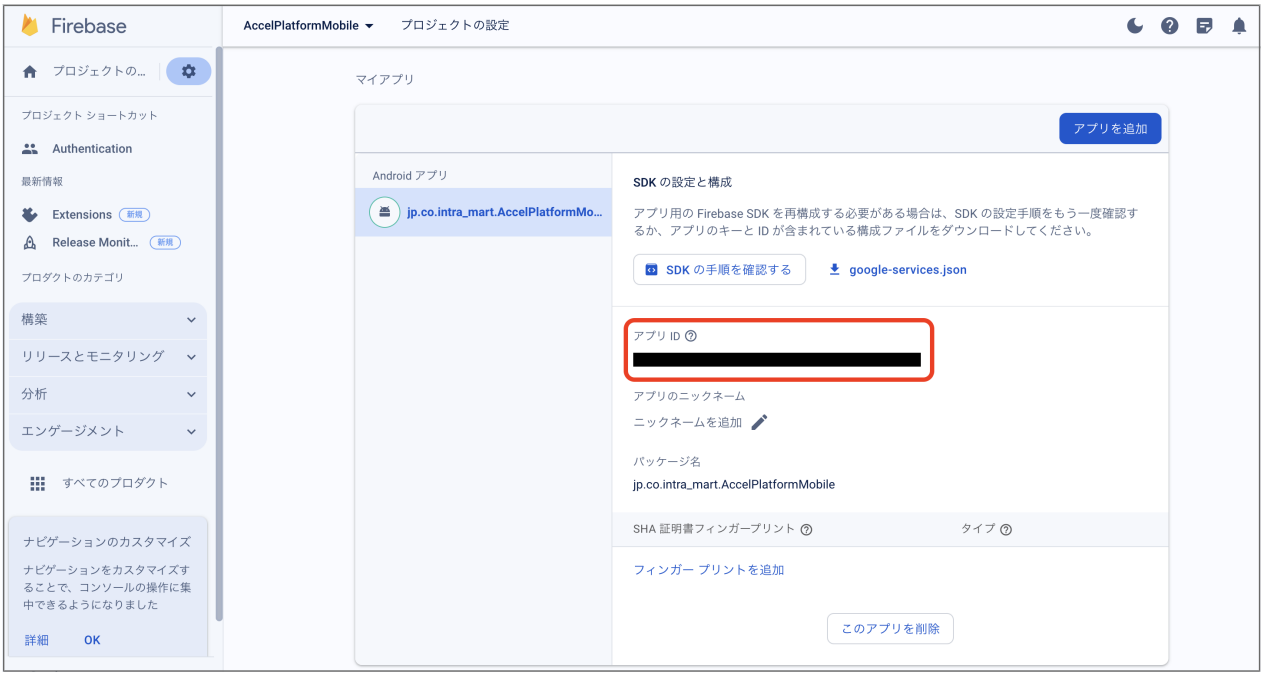


17. 表示されたメッセージを確認し、「キー生成」をクリックしてください。
秘密鍵が生成されると、JSON 形式のサービスアカウントの認証情報ファイルがダウンロードされます。
設定ファイルを編集する際に、設定ファイルと同じディレクトリに配置する必要がありますので、保持しておいてください。



18. 全般タブを選択し、プロジェクトのプロジェクトID、ウェブAPIキーを確認します。
設定ファイルを編集する際に必要な文字列ですので、控えておいてください。

19. マイアプリ - Androidアプリのアプリ IDを確認します。
設定ファイルを編集する際に必要な文字列ですので、控えておいてください。



20. 「Cloud Messaging」タブへ移動し、送信者IDを確認します。
設定ファイルを編集する際に必要な文字列ですので、控えておいてください。

Accel Platform Mobile 設定ファイルの編集

IM-Juggling の <（プロジェクト名）/conf> 配下に出力されたim-notice-mobile-config.xmlを開き、以下の設定を行ってください。

- <proxy>
  1. Firebase Cloud Messagingへ接続するためにプロキシを使用する場合、設定を行ってください。

```
<proxy enable="true">
  <host></host>
  <port xsi:nil="true"></port> <!-- ポート番号を指定する場合は xsi:nil="true" を削除してください。 -->
  <username></username>
  <password></password>
  <workstation></workstation>
  <domain></domain>
</proxy>
```

- <fcm>
  1. <fcm>のname属性に任意の名前を設定してください。

```
<fcm name="fcm">
  ....
</fcm>
```

2. <url>
   - https://fcm.googleapis.com/v1/projects/%REPLACE_WITH_PROJECT_ID%/messages:send を指定してください。
     ただし、%REPLACE_WITH_PROJECT_ID% はFirebaseプロジェクトのプロジェクトIDに置き換えて指定してください。
3. <project-id>
   - Firebaseプロジェクトのプロジェクトを設定してください。
4. <application-id>
   - Androidアプリのアプリを設定してください。
5. <api-key>
   - Firebaseプロジェクトのウェブを設定してください。
6. <credentials-file>
   - ダウンロードした JSON 形式のサービスアカウントの認証情報ファイルを im-notice-mobile-config.xml と同じディレクトリに配置してください。
     credentials-file には、認証情報ファイルのファイル名を設定してください。
7. <sender-id>
   - クラウドメッセージングの送信者IDを設定してください。

> **コラム**
> 各項目の確認方法は、 *FCMの設定* を参照してください。

- <push type=”android” />
  1. engine属性に、<fcm>のname属性に設定した名前を指定してください。

```
<push type="android" engine="fcm" />
```

- <max-push-subject-length>
  1. Push通知受信時に表示するタイトルの最大文字数を設定してください。

```
<max-push-subject-length>30</max-push-subject-length>
```

- <max-push-body-length>
  1. Push通知受信時に表示する本文の最長文字数を設定してください。

```
<max-push-body-length>70</max-push-body-length>
```

サービスアカウントの認証情報ファイルの更新

サービスアカウントの認証情報ファイルの更新を行います。

> **コラム**
> サービスアカウントの認証情報ファイルの更新方法は、 *サービスアカウントの認証情報ファイルの更新* を参照してください。

FCM HTTP v1 API への移行

intra-mart Accel Platform 2023 Autumn(Hollyhock) 以前のバージョンでは、Accel Platform Mobile で FCM を使用して通知を送信する際に、FCM HTTP API (Legacy HTTP API) が使用されていました。
FCM HTTP API (Legacy HTTP API) は、2024年6月20日に廃止される予定です。
intra-mart Accel Platform 2024 Spring(Iris) 以降のバージョンから、FCM HTTP v1 API を使用して通知を送信可能です。
intra-mart Accel Platform 2023 Autumn(Hollyhock) 以前のバージョンで FCM を使用して通知を送信していた場合に、intra-mart Accel Platform 2024 Spring(Iris) 以降にバージョンアップ後、FCM HTTP v1 API が使用されるように設定する手順を説明します。

> **コラム**
> FCM HTTP v1 API への移行方法は、 *FCM HTTP v1 API への移行* を参照してください。

FCM+Amazon SNSを使用する場合

FCMとAmazon SNSを使用しAndroidへの通知を行う場合、以下の設定を行ってください。

FCM の設定

FCMの設定を行います。

> **コラム**
> FCMの設定方法は、 *FCMの設定* を参照してください。

Amazon SNS の設定

Amazon SNSの設定を行います。

> **コラム**
> Amazon SNSの設定方法は、 *Amazon SNS の設定* を参照してください。

ポリシーの作成

Amazon SNSに接続するために必要な権限を付与するための、IAMポリシーの作成を行います。
作成したポリシーは、 *アクセスキーとシークレットキーの作成* または *ロールの作成* に使用します。

> **コラム**
> ポリシーの作成方法は、 *ポリシーの作成* を参照してください。

アクセスキーとシークレットキーの作成

Amazon SNSへの接続に必要な、アクセスキーとシークレットキーの作成を行います。

> **コラム**
> アクセスキーとシークレットキーの作成方法は、 *アクセスキーとシークレットキーの作成* を参照してください。

ロールの作成

Amazon SNSへの接続に必要な、IAMロールの作成を行います。

> **コラム**
> ロールの作成方法は、 *ロールの作成* を参照してください。

Accel Platform Mobile 設定ファイルの編集

IM-Juggling の <（プロジェクト名）/conf> 配下に出力されたim-notice-mobile-config.xmlを開き、以下の設定を行ってください。

- <proxy>
  1. Firebase Cloud MessagingやAmazon SNSへ接続するためにプロキシを使用する場合、設定を行ってください。

```
<proxy enable="true">
  <host></host>
  <port xsi:nil="true"></port> <!-- ポート番号を指定する場合は xsi:nil="true" を削除してください。 -->
  <username></username>
  <password></password>
  <workstation></workstation>
  <domain></domain>
</proxy>
```

- <asns>
  1. <asns>のname属性に任意の名前を設定してください。

```
<asns name="asns-fcm">
  ....
</asns>
```

  2. <endpoint>
     - Amazon SNSのリージョンに対応するEndpointを設定してください。

```
<endpoint>sns.ap-northeast-1.amazonaws.com</endpoint>
```

> **コラム**
>
> Endpointは AWS documentation - Regions and Endpoints で確認できます。
> リージョンが「Asia Pacific (Tokyo)」の場合、Endpointは「sns.ap-northeast-1.amazonaws.com」です。

3. <platform-application-arn>
   - Application ARNを設定してください。

> **コラム**
>
> Application ARNの確認方法は、 *Amazon SNS の設定* を参照してください。

4. <access-key>
   - 通知にアクセスキーとシークレットキーを使う場合、 *アクセスキーとシークレットキーの作成* で作成したアクセスキーを設定してください。
5. <secret-key>
   - 通知にアクセスキーとシークレットキーを使う場合、 *アクセスキーとシークレットキーの作成* で作成したシークレットキーを設定してください。
6. <iam-role-arn>
   - 通知にロールを使う場合、 *ロールの作成* で作成したロールのARNを設定してください。
7. <fcm-project-id>
   - FirebaseプロジェクトのプロジェクトIDを設定してください。
8. <fcm-application-id>
   - Androidアプリのアプリ IDを設定してください。
9. <fcm-api-key>
   - FirebaseプロジェクトのウェブAPIキーを設定してください。
10. <fcm-sender-id>
    - Firebase クラウドメッセージングの送信者IDを設定してください。

> **コラム**
>
> プロジェクトID等Firebase設定情報の確認方法は、 *FCMの設定* を参照してください。

- <push type=”android” />
  1. engine属性に、<asns>のname属性に設定した名前を指定してください。

```
<push type="android" engine="asns-fcm" />
```

- <max-push-subject-length>
  1. Push通知受信時に表示するタイトルの最大文字数を設定してください。

```
<max-push-subject-length>30</max-push-subject-length>
```

- <max-push-body-length>
  1. Push通知受信時に表示する本文の最長文字数を設定してください。

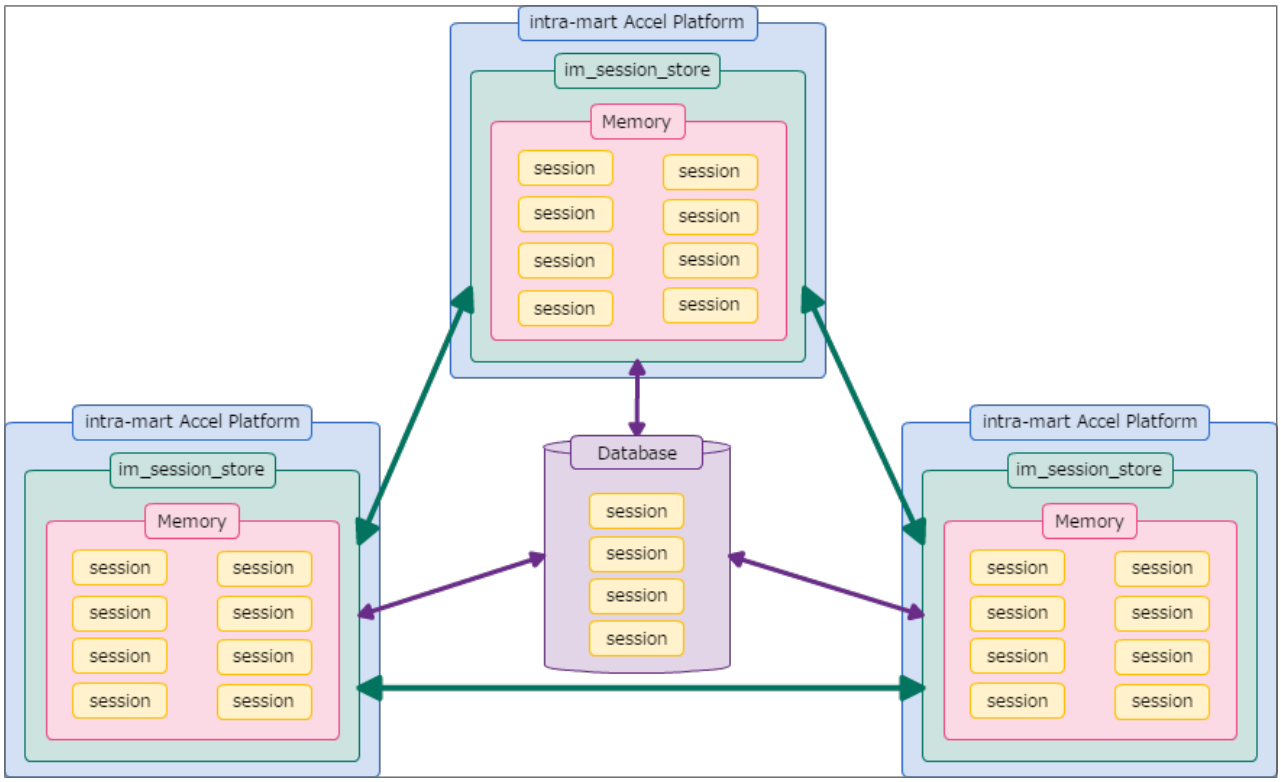
```
<max-push-body-length>70</max-push-body-length>
```

セッション管理モジュール

項目

- 概要
  - クラスタリング
- 前提条件
- セットアップ
  - セッション管理設定
  - Hazelcast設定
- クラスタ構成場合
- セッション情報の永続化を行う場合
- セッションストアメモリサイズの計算式

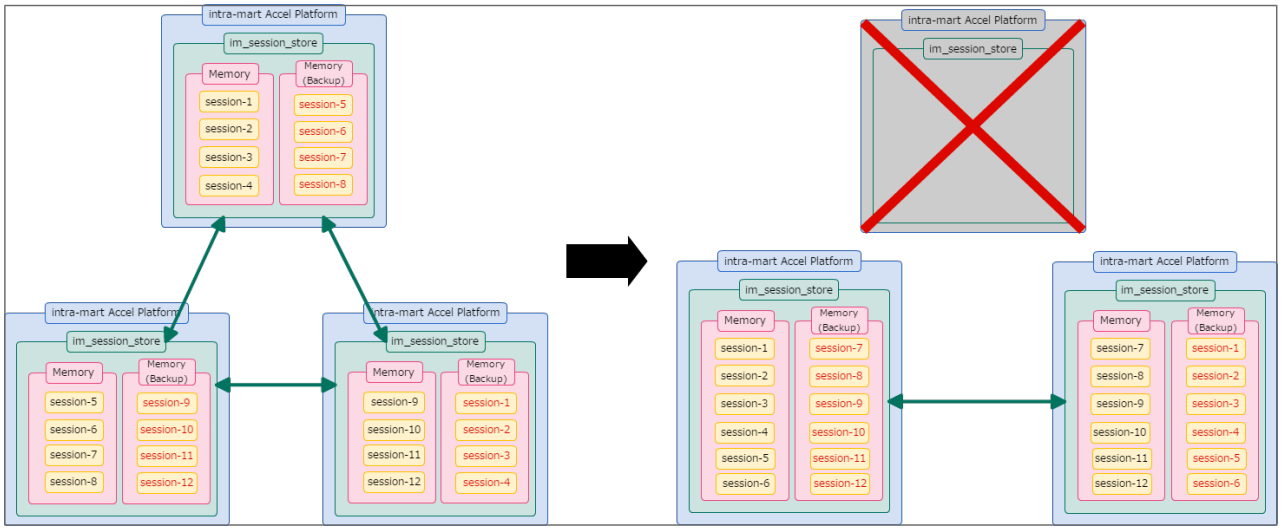
概要



セッション管理モジュールは、アプリケーションサーバのセッション管理機能を利用せずに intra-mart Accel Platform にてセッション管理を行う機能です。
セッション管理モジュールでは、セッション情報をオンメモリで管理します。永続化設定を行うことでデータベースにセッション情報を永続化することも可能です。永続化設定を行うことにより、アプリケーションサーバの再起動を行った場合でも再起動前のセッション情報が参照可能です。

> **コラム**
> セッション管理モジュールを利用した場合、アプリケーションサーバのセッションに関する設定が無視されます。

クラスタリング



クラスタ構成を採用している場合セッション情報は各ノードに分散して保持します。また、１つのセッション情報はメイン／バックアップの２つが異なるノードにてそれぞれ保持されます。このため永続化設定を行わない環境においても、単一のノードが停止した場合にクラスタの再構築により停止したノードが保持していた情報が復元されます。

前提条件

- アプリケーションサーバには Resin を使用してください。
- 以下のモジュールが必要です。
  - セッション管理モジュール
  - セッション管理 組み込みHazelcast連携

セットアップ

セッション管理設定

IM-Juggling で以下の設定を行います。

- セッション管理モジュール - セッション管理設定 session-store-config.xml

詳細は設定ファイルリファレンスの「セッション管理設定」を参照してください。

Hazelcast設定

IM-Juggling で以下の設定を行います。

- セッション管理 組み込みHazelcast連携 - Hazelcast設定 hazelcast-config.xml

詳細は設定ファイルリファレンスの「Hazelcast設定」を参照してください。

クラスタ構成場合

1. TCPを採用している場合、Hazelcast設定の以下の要素をノードの台数分指定します。
   hazelcast-config/network/join/TCP/member-list/member

```
<network>
 <join>
  <TCP>
   <member-list>
    <member>127.248.100.1</member>
    <member>127.248.100.2</member>
    <member>127.248.100.3</member>
   </member-list>
  </TCP>
 </join>
</network>
```

2. Hazelcast設定の以下の要素にノード間通信に利用するスレッド数を指定します。
   hazelcast-config/base/io-thread-count

```
<base>
 <id>im_session_store</id>
 <io-thread-count>3</io-thread-count>
</base>
```

セッション情報の永続化を行う場合

> **注意**
>
> セッションの永続化を有効にしていない場合でも、フェイルオーバーを行うことはできます。
> フェイルオーバーに加えて永続化を行いたい場合のみ、本項記載の設定を行ってください。
>
> 永続化を有効にした場合は、永続化先としてデータベースが利用されます。
> データベースへのアクセスがリクエストの都度発生するため、有効にしない場合に比べてパフォーマンスが劣ります。
>
> データベースへのアクセスが遅延すると、リクエスト処理のパフォーマンス劣化が発生します。
> **セッションの永続化は、必ず以下のような検証を行った上でご利用ください。**
>
> - 負荷試験
> - 運用環境に応じたデータベースのチューニング

1. セッション情報の永続化に利用するデータソースを用意します。
   セッション情報の永続化先用にアプリケーションサーバのデータソース設定を追加してください。
   以下は resin-web.xml の <database> でデータソース設定を行う例です。

```
<database jndi-name="jdbc/session-store">
  <driver>
    <type>org.postgresql.ds.PGConnectionPoolDataSource</type>
    <url>jdbc:postgresql://domain:5432/session_store</url>
    <user>session_store</user>
    <password>session_store</password>
  </driver>
  <max-connections>1024</max-connections>
</database>
```

詳細は設定ファイルリファレンスの「resin-web 設定」を参照してください。

> **注意**
> セッション管理機能で利用するデータソースと他の機能で利用するデータソースが同時に利用された場合、永続化するセッション情報に不整合が発生する可能性があります。 セッション情報の永続化に利用するデータソースはセッション情報管理専用のデータソースを用意してください。

> **注意**
> データソースへの書き込み／読み込みはアプリケーションサーバへのリクエストの都度発生する可能性があります。このため、データベース製品のアーカイブログが大量に出力される可能性があります。

2. データベースにセッション管理を行うためのテーブルを作成します。
   永続化先データソースに対して以下のSQLを実行してください。SQLの実行は各データベース製品が提供しているSQL実行ツールを利用してください。
   - 「**PostgreSQL用DDL**」
   - 「**Oracle用DDL**」
   - 「**SQLServer用DDL**」
3. Hazelcast設定の以下の要素を `true` に設定します。
   `hazelcast-config/store/persistence@enabled`
4. Hazelcast設定の以下の要素に用意したデータソースのJNDI名を指定します。
   `hazelcast-config/store/persistence/jndi-name`

```
<store>
  <persistence enabled="true">
    <jndi-name>java:comp/env/jdbc/session-store</jndi-name>
  </persistence>
</store>
```

セッションストアメモリサイズの計算式

セッション管理 組込Hazelcast連携では、各ノードのメモリサイズを設定する必要があります。この設定により、各ノードでセッション情報をどの程度保持できるのかが決まります。

> **コラム**
> メモリサイズの設定に関しては、「設定ファイルリファレンス」-「セッションストアメモリサイズ設定」を参照してください。

各ノードのメモリサイズについては、以下の計算式で見積もることができます。

- クラスタ構成を採用している場合（ノード数が２以上の場合）

  メモリサイズ ＝（１）×（２）× ２ ÷（３）

- スタンドアローンの場合（ノード数が１の場合）

  メモリサイズ ＝（１）×（２）

（１）... １つのセッションで使用するメモリサイズ
（２）... ピーク時に保持するセッション数
（３）... ノード数

コラム

クラスタ構成を採用している場合、システム全体でセッションを保持するために利用可能なメモリサイズは以下のよう求められます。

システム全体でセッションを保持するために利用可能なメモリサイズ =（1）×（2）÷ 2

（1）... 設定した各ノードのメモリサイズ
（2）... ノード数

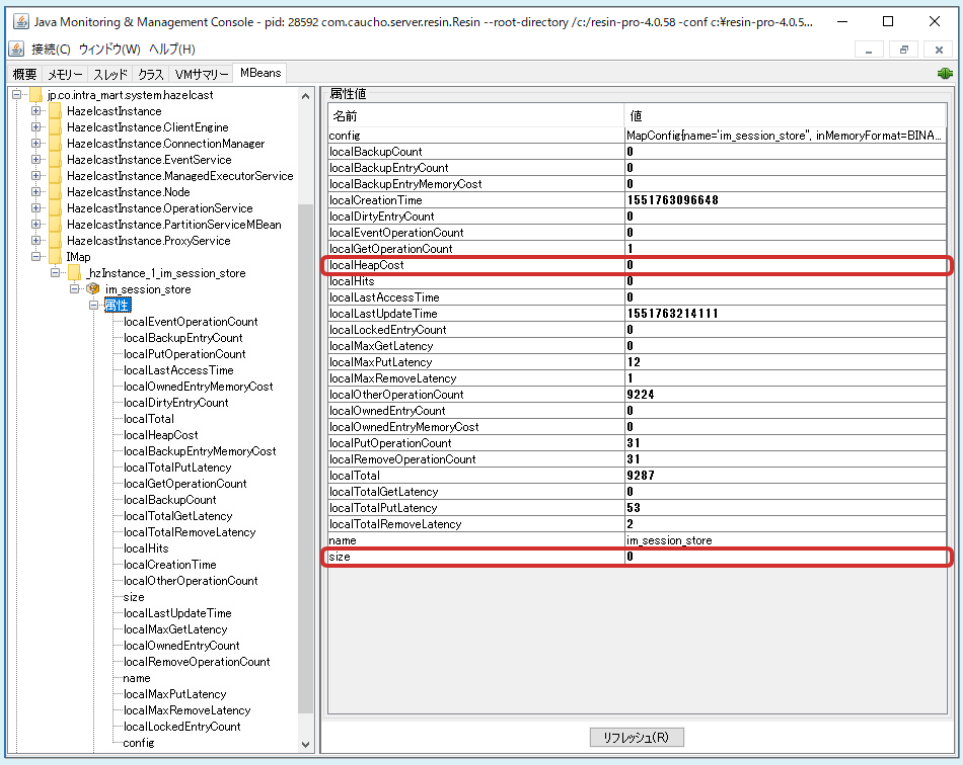
クラスタ構成を採用した場合、バックアップが保持されます。そのため、計算式において 2 で割っています。

コラム

1 つのセッションで使用するメモリサイズについては、以下の方法で確認できます。

1. ＜%RESIN_HOME%/conf/resin.properties＞ ファイルを開きます。
2. 「jvm_args」プロパティに「-Dhazelcast.jmx=true」を追加して JMX 経由で内部情報を確認できるようにします。
3. アプリケーションサーバを起動します。
4. 業務でよく利用する機能を一通り操作します。
5. JConsole を起動し、MBeans タブで「jp.co.intra-mart.system.hazelcast」の以下を展開します。
   - 「IMap」-「_hzInstance_1_im_session_store」-「im_session_store」-「属性」
6. 「localHeapCost」と「size」を確認します。

localHeapCost	そのノードで保持しているすべてのデータのメモリサイズです。単位はバイトです。
size	データのエントリ数です。



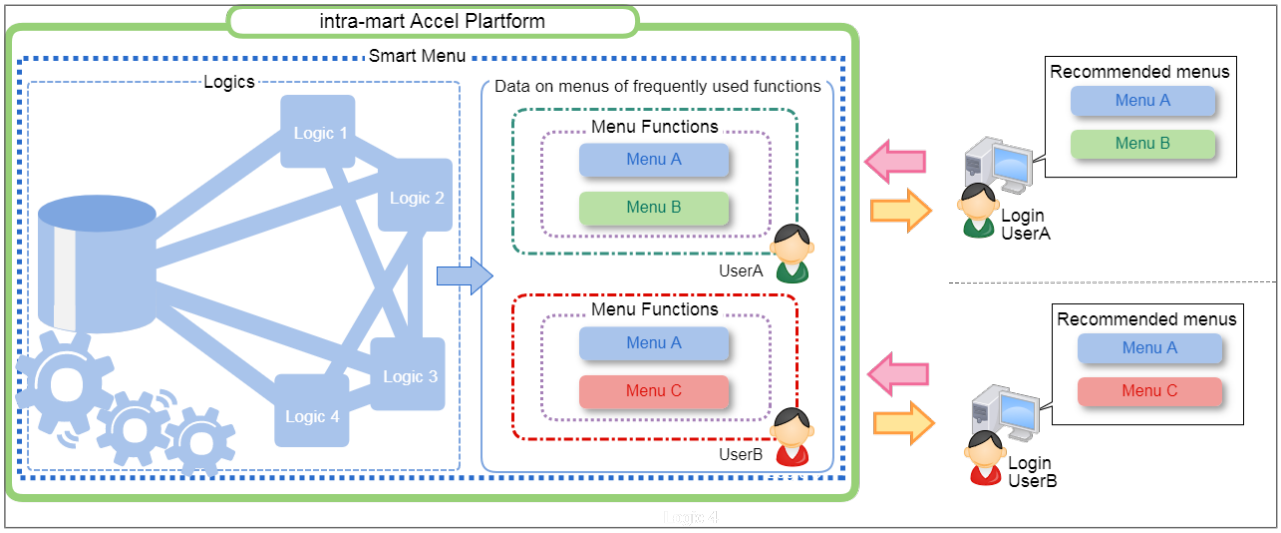
7. 確認した localHeapCost を size で割ることで、1 つのセッションで使用するメモリサイズが確認できます。

スマートメニュー

項目

- 概要
- スマートメニューに関するモジュール群
  - スマートメニューランキング
- セットアップ
  - モジュールの選択
  - 設定ファイル編集
  - デプロイとテナント環境セットアップ
  - ジョブスケジューラトリガの設定
  - スマートメニューポートレットの設定

概要



スマートメニューは、ログインユーザごとに利用を推奨するメニューを提供する機能です。
intra-mart Accel Platform では、グローバルナビやサイトマップなど様々なメニューを提供しています。
これらは、ログインユーザが intra-mart Accel Platform において利用可能なメニューがすべて列挙されています。
スマートメニューは、ログインユーザが利用したいメニューを管理すること無く表示する機能です。

スマートメニューに関するモジュール群

提供されているモジュールは以下の通りです。

- スマートメニュー

モジュールID	jp.co.intra_mart.im_smart_menu
説明	スマートメニューとしての基盤機能を提供するモジュールです。

- *スマートメニューランキング*

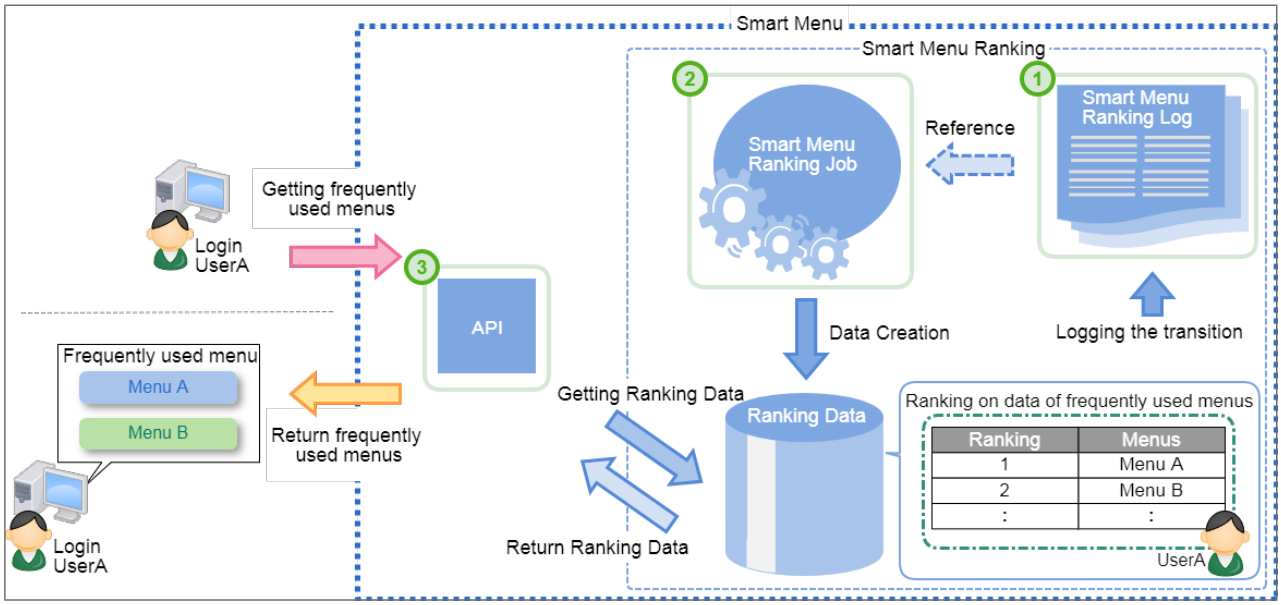
モジュールID	jp.co.intra_mart.im_smart_menu_ranking
説明	ログインユーザがよく利用するメニューを提供するスマートメニュー用のモジュールです。

- スマートメニューポートレット

モジュールID	jp.co.intra_mart.im_smart_menu_potlet
説明	スマートメニューを表示するポートレット（スマートメニューポートレット）を提供するモジュールです。

スマートメニューランキング

スマートメニューランキングは、ログインユーザのアクセス回数が多いメニューを「よく使うメニュー」として提供するモジュールです。
スマートメニューランキングモジュールでは以下の手順で「よく使うメニュー」を表示します。



1. ログインユーザの遷移情報を「スマートメニューランキングログ」に記録します
2. ジョブ「ランキングデータ集計」を実行し、「スマートメニューランキングログ」からランキングデータを集計します
3. ランキングデータからメニューを取得します

セットアップ

以下の手順に従いセットアップを行ってください。

モジュールの選択

IM-Juggling のjugglingプロジェクトで以下のモジュールを選択してください。

- スマートメニュー

  intra-mart Accel Platform > 追加機能 > スマートメニュー > スマートメニュー
- スマートメニューランキング

  intra-mart Accel Platform > 追加機能 > スマートメニュー > スマートメニューランキング
- スマートメニューポートレット

  intra-mart Accel Platform > 追加機能 > ポートレット > スマートメニューポートレット

設定ファイル編集

必要に応じて、以下の設定ファイルの内容を変更してください。すべて「スマートメニューランキング」モジュールの設定ファイルです。

- スマートメニューランキングログ

  ログの出力先やローテート方式を変更したい場合に設定します。
- スマートメニューランキング設定

  「スマートメニューランキングログ」設定を変更した場合や、メニューの最大表示数を変更したい場合に設定します。

デプロイとテナント環境セットアップ

IM-Juggling でWARを作成し、テナント環境セットアップを行ってください。

ジョブスケジューラトリガの設定

スマートメニューランキングの「よく使うメニュー」のメニューを表示するには、ランキングデータを集計する必要があります。
テナント管理者でログインしてジョブネット「ランキングデータ集計」に対してトリガを設定してください。
トリガの設定方法についてはテナント管理者操作ガイドの「ジョブネットを設定する」を参照してください。

> **コラム**
>
> 「よく使うメニュー」に表示されるメニュー情報は「ランキングデータ集計」ジョブを実行することにより生成されます。
> 定期的にジョブネットを実行することで、ログインユーザの最新の遷移情報をメニューに反映できます。

> **注意**
>
> ジョブ「ランキングデータ集計」実行中は「よく使うメニュー」のメニューが表示されないことがあります。
> ユーザのアクセスが少ない時間帯にジョブネットを実行するように設定してください。

スマートメニューポートレットの設定

ポートレットにスマートメニューポートレットを設定します。設定方法については以下のドキュメントを参照してください。

- ポータル 管理者操作ガイドの「スマートメニューポートレットを設定する」

WARファイルによる複数テナント

intra-mart Accel Platform では、複数のwarファイルをResin上にデプロイすることにより マルチテナント環境 を実現します。
マルチテナント環境 は、それぞれのテナント環境とリソースを共有しません。
その為、 intra-mart Accel Platform の設定はテナント単位でそれぞれ分ける必要があります。
設定を分ける必要がある設定は以下の通りです。

1. クラスタリングID、クラスタリング用ポート番号、およびポートレンジ
   Network ネットワーク設定を参照してください。
   クラスタリングIDおよび、クラスタリング用ポート番号、ポートレンジがテナント間で重複しないよう設定する必要があります。
2. DataSource
   DataSource を参照してください。
   テナント毎にデータベース接続先をそれぞれ設定する必要があります。
3. Storage領域
   Storage を参照してください。
   テナント毎に利用するStorage領域をそれぞれ設定する必要があります。
4. Apache Cassandra の接続先
   IMBox を参照してください。
   テナント毎に接続する Apache Cassandra の接続先、またはkeyspaceをそれぞれ設定する必要があります。
5. Apache Solr の接続先
   IM-ContentsSearch を参照してください。
   テナント毎に接続する Apache Solr を設定する必要があります。
6. warファイルの名称
   Resinでは、warファイルの名称を元にコンテキストパスが決定されます。
   マルチテナント環境 の場合、warファイルの名称を分ける必要があります。

> **注意**
>
> WARファイルによる複数テナント 環境を構築する場合、単体の場合と比べメモリを消費しますので注意してください。
> メモリ使用量の設定に関しては *Resinの設定* を参照してください。

テナント解決機能

概要

この章では、バーチャルテナント機能利用時における、操作対象のテナントがどのように解決されるかを説明します。

intra-mart Accel Platform では、アカウントコンテキストのテナントIDプロパティが、操作対象のテナントを意味しています。
アカウントコンテキストのテナントIDプロパティは、テナント解決機能を利用して解決されます。

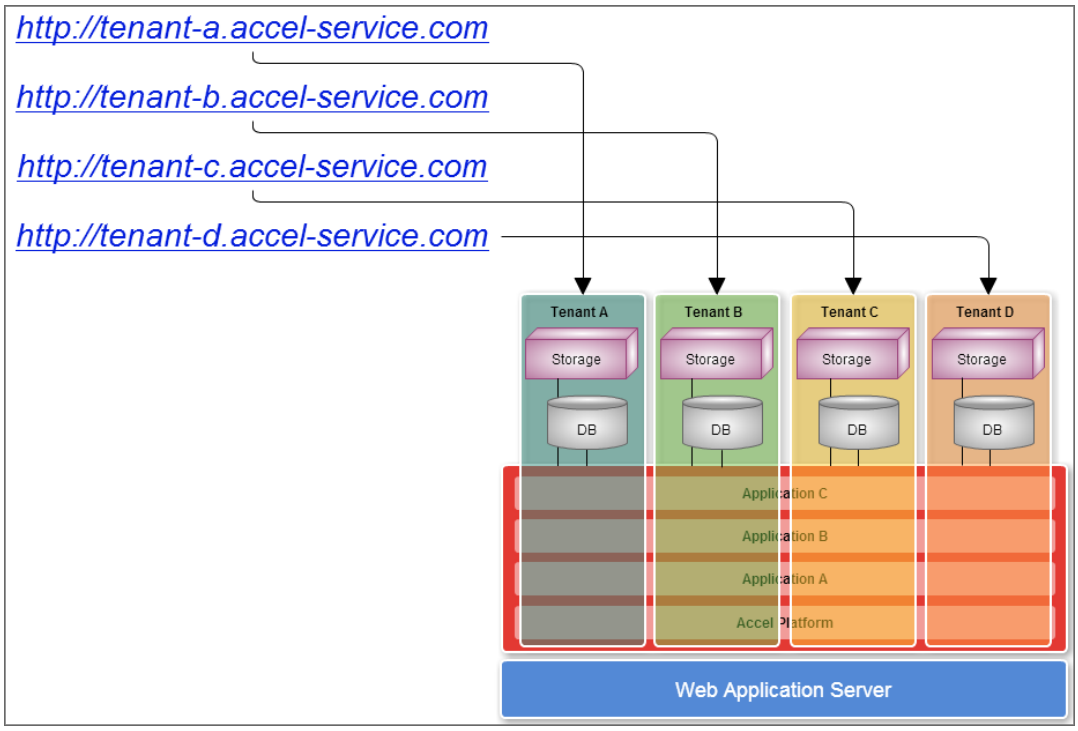
以降では、テナント解決機能の仕様について説明します。

なお、intra-mart Accel Platform 上に作成するプログラムは、操作対象のテナントを意識する必要はありません。
（操作対象のテナントは、このページで説明するテナント解決の仕組みによって自動的に解決されます。）

リクエスト情報を利用したテナント自動解決機能について

intra-mart Accel Platform では、リクエスト情報を利用して操作対象となるテナントを決定することが可能です。
この機能を利用することにより、利用者の認証状態を問わず特定のテナントが操作対象に指定され、ログイン時にテナントIDを指定する必要がありません。

例えば、URLのサブドメインを利用してテナントの自動解決を行うことで、以下のように、URLごとに操作対象となるテナントを決定することが可能です。



> **コラム**
>
> 標準では、リクエスト情報を利用したテナント自動解決機能は利用しません。
> リクエスト情報を利用したテナント自動解決機能は、「IM-SecureSignOn for Accel Platform」アプリケーションや、機能を提供するモジュールを利用することで有効に設定できます。

> **コラム**
>
> リクエスト情報を利用したテナント自動解決機能を実装する方法については、「*リクエスト情報を利用したテナント自動解決機能を提供する*」を参照してください。

テナント解決のパターン

Webアクセス時にテナントを解決するパターンは、以下の通りです。

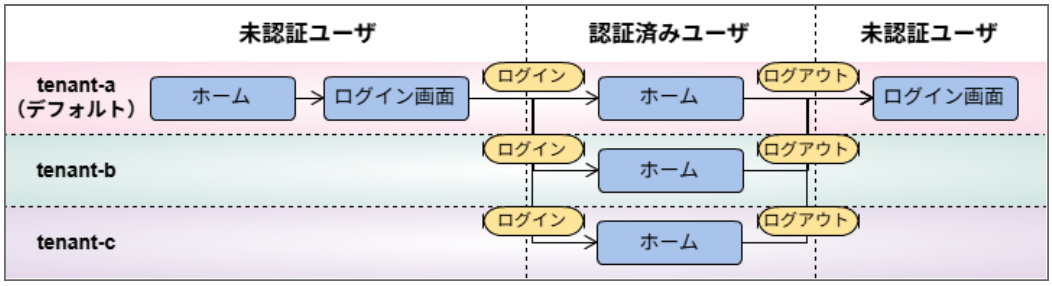
- 一般ユーザ
  - ログイン時にテナントを指定する場合
  - リクエスト情報を利用したテナント自動解決機能を利用する場合
- システム管理者

一般ユーザ

ログイン時にテナントを指定する場合

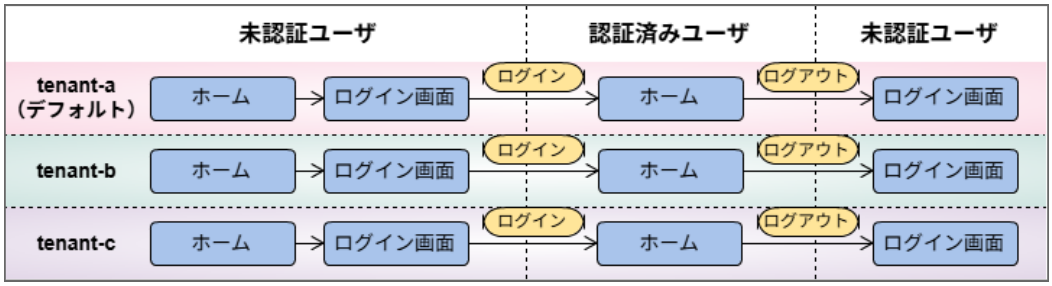
テナントが複数存在する場合は、一般ユーザのログイン画面でテナントを指定してログインを行います。

ログインに成功すると、ログイン画面で指定したテナントが操作対象のテナントに指定されます。
ログアウトを行うことで、未認証状態となり、デフォルトテナントが操作対象に切り替わります。
未認証状態では、デフォルトテナントが操作対象のテナントに指定されています。



リクエスト情報を利用したテナント自動解決機能を利用する場合

リクエスト情報を利用したテナント自動解決機能を利用することで、リクエストの任意の情報からテナントの解決を行うことが可能です。
この機能を利用している状態では、ログイン／ログアウトにより操作対象テナントが切り替わることは無く、ログイン画面でテナントを指定してログインを行うことはできません。
未認証状態では、自動解決されたテナントが操作対象のテナントに指定されています。



システム管理者

システム管理者の場合、操作対象のテナントはデフォルトテナントに指定されています。
システム管理者は、テナントの切り替え機能を利用して操作対象のテナントを切り替えることが可能です。詳細は「システム管理者操作ガイド」の「テナントの切り替え」を参照してください。

リクエスト情報を利用したテナント自動解決機能を提供する

概要

リクエスト情報を利用したテナント自動解決機能が有効な場合、操作対象のテナントは、以下の流れで解決されます。

1. 提供されているテナント自動解決機能にて、テナントIDの解決を行います。
2. 提供されているテナント検証機能にて、テナントIDの検証を行います。
3. テナントIDをアカウントコンテキストのテナントIDプロパティに設定します。

以下では、*テナント自動解決機能*、*テナント検証機能* とその提供方法について説明します。

テナント自動解決機能

リクエスト情報から、テナントIDを解決する機能です。

`RequestBasedTenantIdResolver` インタフェースを実装したクラスを作成し、プラグイン設定ファイルを設定することで、リクエスト情報を利用したテナント自動解決機能を利用することが可能です。

テナント解決機能は、複数設定して提供することが可能です。
複数のテナント解決機能が提供されている場合は、プラグインの仕様に従い、優先度の高いテナント解決機能から順番に実行されます。
テナントの解決が行われた（`getTenantId(HttpServletRequest)` メソッドで `null` 以外を返した）時点でテナント解決処理が終了します。後続のテナント解決機能は実行されません。
テナントが解決できない（`getTenantId(HttpServletRequest)` メソッドで `null` を返した）場合は、後続のテナント解決機能にてテナントの解決を行います。
全てのテナント解決機能がテナントを解決できなかった（全て `null` を返した）場合、次項の *テナント検証機能* の検証で問題がなかった場合は、 デフォルトテナントと解決されます。

テナント自動解決機能の作成

以下のインタフェースを実装したクラスを作成します。

- `jp.co.intra_mart.foundation.admin.tenant.context.RequestBasedTenantIdResolver`

以下はCookieを利用してテナントを解決するサンプル実装です。

```
package sample.context;

import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServletRequest;

import jp.co.intra_mart.foundation.admin.tenant.context.RequestBasedTenantIdResolver;

public class CookieTenantIdResolver implements RequestBasedTenantIdResolver {

   private static final String COOKIE_KEY = "X-TENANT-ID";

   @Override
   public String getTenantId(final HttpServletRequest request) {
      return getCookieValue(request, COOKIE_KEY);
   }

   private Cookie getCookie(final HttpServletRequest request, final String name) {
      final Cookie[] cookies = request.getCookies();

      if (cookies == null) {
         return null;
      }

      for (final Cookie cookie : cookies) {
         if (name.equals(cookie.getName())) {
            return cookie;
         }
      }

      return null;
   }

   private String getCookieValue(final HttpServletRequest request, final String name) {
      final Cookie cookie = getCookie(request, name);

      if (cookie == null) {
         return null;
      }

      return cookie.getValue();
   }

}
```

> **コラム**
> RequestBasedTenantIdResolver インタフェースの詳細は「RequestBasedTenantIdResolverインタフェースのAPIリスト」を参照してください。

テナント自動解決機能の設定

テナント自動解決機能はプラグインファイルで設定を行います。従って、プラグインの機能を利用した設定が可能です。
リクエスト情報を利用したテナント自動解決機能を設定するための拡張ポイントは
jp.co.intra_mart.foundation.admin.tenant.context.tenant.resolvers です。
作成したプラグインファイルを WEB-INF/plugin/sample_tenant_resolver/plugin.xml に配置します。（plugin直下のフォルダ名は、他のプラグインと重複しない名前で任意に変更することが可能です。）
上記のCookieを利用したテナント自動解決機能の実装を動作させるプラグイン設定のサンプルは以下の通りです。

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin>
  <extension point="jp.co.intra_mart.foundation.admin.tenant.context.tenant.resolvers">
    <tenant-id-resolvers
      id="sample.context.cookie_resolver"
      version="1.0.0"
      rank="1">

      <!-- tenant-id-resolver タグは複数指定可能です。 -->
      <!-- RequestBasedTenantIdResolverインタフェースを実装したクラスの完全修飾クラス名をclass属性に指定してください。 -->
      <tenant-id-resolver class="sample.context.CookieTenantIdResolver" />

    </tenant-id-resolvers>
  </extension>
</plugin>
```

> **コラム**
> プラグインの設定仕様については、「PluginManagerのAPIリスト」を参照してください。

上記の資材をデプロイすると、Cookieのキー `X-TENANT-ID` の値をテナントIDとして扱います。

テナント検証機能

テナント自動解決機能により解決したテナントIDが妥当な値であるかを判別する機能です。

テナント自動解決機能により解決したテナントIDは検証機能にてエラーとならない場合、アカウントコンテキストのテナントIDプロパティに設定されます。例えば、存在しないテナントIDがアカウントコンテキストのテナントIDプロパティに設定されてしまうと intra-mart Accel Platform が提供する機能が正常に動作しなくなります。このような状態を回避するために、テナント検証機能にて妥当性チェックを行います。

テナント検証機能は、複数設定して提供することが可能です。
複数のテナント検証機能が提供されている場合は、プラグインの仕様に従い、優先度の高いテナント検証機能から順番に実行されます。
テナントの検証にてエラー（`validate(String, Resource)` メソッドで `InvalidTenantIdException` ）が発生した時点でテナント検証処理が終了します。後続のテナント検証機能は実行されません。

テナント検証機能の作成

以下のインタフェースを実装したクラスを作成します。

- `jp.co.intra_mart.foundation.admin.tenant.context.TenantIdValidator`

以下にサンプル実装を例示します。

```
package jp.co.intra_mart.foundation.admin.tenant;

import jp.co.intra_mart.foundation.admin.tenant.context.TenantIdValidator;
import jp.co.intra_mart.foundation.context.core.Resource;
import jp.co.intra_mart.foundation.admin.tenant.InvalidTenantIdException;

public class SampleTenantIdValidator implements TenantIdValidator {

  @Override
  public void validate(final String tenantId, final Resource resource) throws InvalidTenantIdException {

    if (tenantId != null && !tenantId.startsWith("TENANT_")) {
      // テナントID書式チェック
      // テナントIDが "TENANT_" で始まっていることをチェックします。
      throw new InvalidTenantIdException("テナントIDの書式が正しくありません。");
    }

  }

}
```

> **コラム**
> `TenantIdValidator` インタフェースの詳細は「TenantIdValidatorインタフェースのAPIリスト」を参照してください。

テナント検証機能の設定

テナント検証機能も、テナント自動解決機能と同様にプラグイン設定ファイルで設定を行います。

テナント検証機能を設定するための拡張ポイントは jp.co.intra_mart.foundation.admin.tenant.context.tenant.validators です。

以下にサンプルのプラグイン設定ファイルを例示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin>
  <extension point="jp.co.intra_mart.foundation.admin.tenant.context.tenant.validators">
    <tenant-id-validators
      id="sample.context.sample_tenant_id_validator"
      version="1.0.0"
      rank="1">

      <!-- tenant-id-validator タグは複数指定可能です。 -->
      <!-- TenantIdValidatorインタフェースを実装したクラスの完全修飾クラス名をclass属性に指定してください。 -->
      <tenant-id-validator class="sample.context.SampleTenantIdValidator" />

    </tenant-id-validators>
  </extension>
</plugin>
```

StandardTenantIdValidator

標準で提供されている TenantIdValidator の実装クラスです。
テナント自動解決機能によるテナント解決チェック、テナントの存在チェックなどを行います。

完全クラス修飾子
jp.co.intra_mart.system.admin.context.StandardTenantIdValidator
プラグインの初期化パラメータ

StandardTenantIdValidator は、PropertyInitParamable インタフェースを実装しているため、 プラグイン設定ファイルにて初期化パラメータを受け渡すことが可能です。
初期化パラメータは <init-param> タグで指定します。

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin>
  <extension point="jp.co.intra_mart.foundation.admin.tenant.context.tenant.validators">
    <tenant-id-validators
      id="jp.co.intra_mart.tenant_id.validator.standard"
      version="8.0.7"
      rank="100">
      <tenant-id-validator class="jp.co.intra_mart.system.admin.context.StandardTenantIdValidator">

        <init-param>
          <param-name>required_tenant_id</param-name>
          <param-value>true</param-value>
        </init-param>
        <init-param>
          <param-name>valid_tenant_id</param-name>
          <param-value>true</param-value>
        </init-param>

      </tenant-id-validator>
    </tenant-id-validators>
  </extension>
</plugin>
```

param-name	説明
required_tenant_id	テナントIDが解決されているかを検証します。true / false で指定し、true の場合は検証を行います。 テナントIDが解決されていない場合は、例外 InvalidTenantIdException が発生し、システムにアクセスできません。 省略した場合は、検証を行いません。
valid_tenant_id	解決されたテナントIDが存在するテナントのテナントIDであるかを検証します。true / false で指定し、true の場合は検証を行います。 解決されたテナントIDが存在するテナントのテナントIDではない場合は、例外 InvalidTenantIdException が発生し、システムにアクセスできません。 省略した場合は、検証を行いません。

コラム
PropertyInitParamable インタフェースの詳細は「PropertyInitParamableインタフェースのAPIリスト」を参照してください。

ポート一覧

- intra-mart Accel Platform で利用するポート番号について説明します。

項目
- Resin
  - HTTP通信ポート
  - クラスタポート（トライアドサーバ）
  - クラスタポート（ダイナミックサーバ）
  - watchdogポート
- JGroup
- Hazelcast
- Apache Cassandra
  - クライアント接続ポート
  - ノード間通信ポート
  - ノード間通信ポート（ノード間通信にSSLを利用した時）
  - JMX接続ポート
- Apache Solr
- IM-Notice
- モジュール開発支援ライブラリ

Resin

- Resin は以下の機能のポートを使用します。
  - HTTP（HTTPS）ポート
  - クラスタポート

> **注意**
>
> クラスタポートはトライアドサーバ、ダイナミックサーバで設定方法が異なります。
>
> - トライアドサーバ
>   <resin.properties>ファイルを編集
> - ダイナミックサーバ
>   コマンドライン引数を指定

  - watchdogポート

HTTP通信ポート

- ResinがHTTP通信を行うためのポートの設定です。
  - 設定項目

デフォルトのポート番号	8080
デフォルトの有効/無効	有効
設定ファイル	resin.properties
プロパティ	app.http

  - 設定例

全てのサーバで同じポートを使用する場合は「**app.http**」をひとつ記述してください。

```
app.http        : 8080
```

サーバ毎に使用するポートを指定する場合は「**[サーバID].http**」を記述します。
下記の記述例では「**app-0**」のResinは「8081」を使用しますが、それ以外のResinは「8080」を使用します。

```
app.http        : 8080
app-0.http      : 8081
```

クラスタポート（トライアドサーバ）

- Resinがクラスタリング行うためのポートの設定です。
  - 設定項目

デフォルトのポート番号	6800
デフォルトの有効/無効	有効
設定ファイル	resin.properties
プロパティ	app_servers

  - 設定例

    デフォルトのポート番号を指定する場合は「app_servers」にIPアドレスのみを記述してください。

```
app_servers     : 192.168.100.100 192.168.100.101 192.168.100.102
```

    任意のポート番号を使用する場合は「[IPアドレス]:[ポート番号]」と記述してください。

```
app_servers     : 192.168.100.100:6801 192.168.100.101:6802 192.168.100.102:6803
```

クラスタポート（ダイナミックサーバ）

- Resinがクラスタリング行うためのポートの設定です。
  - 設定項目

デフォルトのポート番号	6830
コマンドライン引数	–elastic-server-port

  - 設定例

    ポート番号を指定しない起動コマンドではデフォルトのポート番号を使用します。

```
> resin.exe --elastic-server --cluster app console
```

    任意のポート番号を使用する場合は「–elastic-server-port」を起動コマンドに指定してください。

```
> resin.exe --elastic-server --cluster app --elastic-server-port 6831 console
```

watchdogポート

- 設定項目

デフォルトのポート番号	6600
デフォルトの有効/無効	有効
設定ファイル	resin.xml
タグ	<watchdog-port>

- 設定例

```
<watchdog-port>6700</watchdog-port>
```

JGroup

- JGroups はサーバ間通信として利用します。

デフォルトのポート番号	5200-5202

Hazelcast

セッション管理 組み込みHazelcast連携を導入している場合に動作します。

- Hazelcast はサーバ間通信として利用します。

デフォルトのポート番号	5701-5702

> **コラム**
> 詳細は設定ファイルリファレンスの「Hazelcast設定」を参照してください。

Apache Cassandra

- Apache Cassandra は以下の機能のポートを使用します。
  - クライアント接続ポート
  - ノード間通信ポート
  - ノード間通信ポート（ノード間通信にSSLを利用した時）
  - JMX接続ポート

> **コラム**
> 詳細は、「 Cassandra管理者ガイド 」を参照してください。

クライアント接続ポート

デフォルトのポート番号	9160

ノード間通信ポート

デフォルトのポート番号	7000

ノード間通信ポート（ノード間通信にSSLを利用した時）

デフォルトのポート番号	7001

JMX接続ポート

デフォルトのポート番号	7199

Apache Solr

- 「 *Apache Solr* 」にて構築したアプリケーションサーバによって異なります。

デフォルトのポート番号	アプリケーションサーバ「Jetty」	8983
	アプリケーションサーバ「Resin」	8080
	アプリケーションサーバ「Tomcat」	8080

IM-Notice

- IM-Noticeが通知を行うためのポートの設定です。

デフォルトのポート番号	40608
ブローカー利用時	40608, 40609

モジュール開発支援ライブラリ

- Jugglingプロジェクト新規作成時に「アプリケーションの選択」で「モジュール開発支援ライブラリ」を選択し、「スクリプト開発のためのデバッグ機能」を含むように選択した場合、
  スクリプトのデバッグのために標準で下記のポートを使用します。

デフォルトのポート番号	9000

> コラム
> 詳細は「intra-mart e Builder for Accel Platform アプリケーション開発ガイド / デバッグ（スクリプト開発）」を参照してください。

IM-Juggling の応用

IM-Juggling を利用せず、CUIでWARファイルを作成する方法

intra-mart Accel Platform は IM-Juggling を利用してWARファイルを作成しますが、このWARファイルの作成をCUIで行うことが可能です。

IM-Juggling をインストールしたディレクトリのpluginsディレクトリ配下に下記形式のディレクトリが存在します。

```
jp.co.intra_mart.flashcat.ant_1.0.0.XXXXXXXXXXXX/ant

※XXXXXXXXXXXX はタイムスタンプが入ります。
※自動更新適用後の場合、複数ディレクトリが存在することがあります。
複数存在する場合は最新のタイムスタンプのディレクトリをご利用ください。
```

ディレクトリ内の構造は下記の通りです。

```
jp.co.intra_mart.flashcat.ant_1.0.0.XXXXXXXXXXXX/ant
├── build.xml  // WAR作成用のantスクリプト
└── lib  // WAR作成に必要なライブラリ群
    ├── FastInfoset-1.2.15.jar
    ├── commons-compress-1.3.jar
    ├── commons-io-1.4.jar
    ├── im_commons-8.0.6.jar
    ├── im_core_base-8.0.6.jar
    ├── im_j2ee_assist_base-8.0.5.jar
    ├── im_j2ee_assist_impl-8.0.5.jar
    ├── im_jackling_core-0.0.3.jar
    ├── im_jackling_toolkit-0.0.4.jar
    ├── im_jdk_assist-8.0.4.jar
    ├── im_juggling_ant-1.0.6.jar
    ├── im_modules-8.0.12.jar
    ├── istack-commons-runtime-3.0.7.jar
    ├── javassist-3.7.ga.jar
    ├── javax.activation-1.2.0.jar
    ├── jaxb-api-2.3.1.jar
    ├── jaxb-runtime-2.3.1.jar
    ├── ognl-2.7.3.jar
    ├── slf4j-api-1.7.5.jar
    ├── stax-ex-1.8.jar
    ├── txw2-2.3.1.jar
    └── xalan-2.7.0.jar
```

このantディレクトリ直下に、projectという名称のディレクトリを作成し、その中に IM-Juggling プロジェクトのファイル群を配置します。 その後、コマンドラインから 以下のantコマンドを実行することによりWARファイルが作成されます。

```
ant clean make
```

また、コマンドラインから 以下のantコマンドを実行することによりWARファイルとあわせて静的ファイルも作成できます。

```
ant clean make static
```

デフォルトの設定では試用版のWARファイル、静的ファイルの作成が可能です。 製品版を作成する際には、 antディレクトリ直下にある <build.xml> ファイルに追記する必要があります。

以下のように <build.xml> の juggling タグ内に trial=”false” を追記してから ant コマンドを実行することで、製品版のWARファイル、静的ファイルの作成が可能です。

```
1  <!-- warファイルの内容を展開します -->
2  <juggling trial="false" type="war" sample="${juggling.sample}" runtime="${juggling.runtime}" project="${juggling.project}"
3  env="${juggling.env}" dest="${juggling.dest}" work="${juggling.work}">
4      <repository name="base" path="${juggling.repository.base}" />
5      <repository name="app" path="${juggling.repository.app}" />
   </juggling>
```

```
<!-- 静的ファイルの内容を展開します -->
<juggling trial="false" type="static" sample="${juggling.sample}" runtime="${juggling.runtime}" project="${juggling.project}"
env="${juggling.env}" dest="${juggling.dest}" work="${juggling.work}">
   <repository name="base" path="${juggling.repository.base}" />
   <repository name="app" path="${juggling.repository.app}" />
</juggling>
```

> **注意**
>
> CUIでWARファイルを作成する際、使用許諾の同意確認が表示されません。
> WARファイルを「製品版」で作成した場合、使用許諾に同意したものと見なします。予めご了承ください。

> **注意**
>
> Java 17 以降で運用を行う場合、CUIでWARファイルの作成はできません。
> IM-Juggling のディレクトリ配下のjreディレクトリをJAVA_HOMEに設定してWARファイルの作成を行ってください。

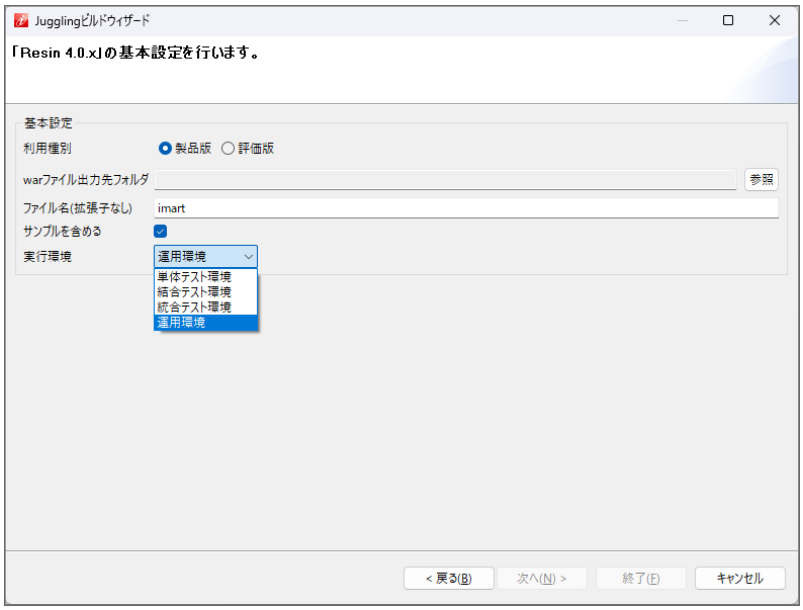
WARファイル作成時の実行環境の変更

IM-Juggling でWARファイルを作成する際、以下の画面で「実行環境」項目を選択する事ができます。

項目

- 概要
- 環境種別毎における各モジュールの動作仕様
  - アクセスコンテキスト情報をログに出力します。
  - ベースURLが未設定の場合に警告ログを出力します。
  - Debug#browse 画面にコールスタック情報を表示します。
  - すべてのスクリプト開発モデルのプログラムをインタプリタモードで実行します。
  - 同一パスに対して異なる認可設定が存在している場合に警告ログを出力します。
  - デバッグ情報を付与したキークラスを生成します。
  - CertificationServletUtil#validateCertificationConfirm()がtrueを返却します。
  - 検索実行エラー発生時に、開発者向けの詳細メッセージを含めて出力します。
  - 設定ファイル読み込み時に、Instance.PROTOTYPE を渡します。
  - ルーティング実行時に、ログにデバッグ情報を出力します。
  - script type=“text/javascript” タグのラッパータグファイルを最小化します。
  - link rel=“stylesheet” タグのラッパータグファイルを最小化します。
  - エラーページにエラーの内容を表示します。

概要



「実行環境」項目の主な違いは次の通りです。

種別	説明
単体テスト環境	JUnitなどを利用する場合を想定しています。パフォーマンスに関しては考慮しません。
結合テスト環境	IDEを使用したテスト環境などを想定しています。一般的に最も利用される環境を想定しています。必要最低限の動作環境です。
統合テスト環境	本来の実行環境（APサーバ）を使用した統合テスト環境を想定しています。カットオーバー直後等でもこの環境を利用することを想定しています。
運用環境	パフォーマンスを考慮した動作環境です。

注意

デバッグ情報の出力処理やキャッシュで生成される情報と違い、レスポンス面やエラー画面の表示などで大きな違いが発生します。
本番の商用環境や運用環境では必ず、実行環境を「運用環境」に選択して、WARファイルを作成することを推奨します。

コラム

「実行環境」項目の設定情報は、WEB-INF/classes/im_env.txtに以下のような文字列で定義されています。

```
単体テスト環境：ut
結合テスト環境：si
総合テスト環境：pt
運用環境：production
```

WARファイルをデプロイした後に一時的に実行環境を変更したい場合は、 上記ファイルの実行環境に関する文字列に変更して再起動することで、 WARファイル作成時と違う実行環境で動作させることもできますが、非推奨です。

環境種別毎における各モジュールの動作仕様

環境種別毎のモジュールの動作仕様については次の通りです。

環境種別	対象モジュール	概要
単体テスト環境	コアモジュール	「*アクセスコンテキスト情報をログに出力します。* 」
	スクリプト開発モデル	「*Debug#browse 画面にコールスタック情報を表示します。* 」
	スクリプト開発モデル	「*すべてのスクリプト開発モデルのプログラムをインタプリタモードで実行します。* 」
	テナント管理機能	「*同一パスに対して異なる認可設定が存在している場合に警告ログを出力します。* 」
	テナント管理機能	「*デバッグ情報を付与したキークラスを生成します。* 」
	認証機能	「*CertificationServletUtil#validateCertificationConfirm() がtrueを返却します。* 」
	IM Contents Search	「*検索実行エラー発生時に、開発者向けの詳細メッセージを含めて出力します。* 」
	IM Contents Search	「*設定ファイル読み込み時に、Instance.PROTOTYPE を渡します。* 」
	UI基本モジュール	「*エラーページにエラーの内容を表示します。* 」
結合テスト環境	コアモジュール	「*アクセスコンテキスト情報をログに出力します。* 」
	スクリプト開発モデル	「*Debug#browse 画面にコールスタック情報を表示します。* 」
	スクリプト開発モデル	「*すべてのスクリプト開発モデルのプログラムをインタプリタモードで実行します。* 」
	ルーティング機能	「*ルーティング実行時に、ログにデバッグ情報を出力します。* 」
	UI基本モジュール	「*エラーページにエラーの内容を表示します。* 」
総合テスト環境	コアモジュール	「*アクセスコンテキスト情報をログに出力します。* 」
	コアモジュール	「*ベースURLが未設定の場合に警告ログを出力します。* 」
	ルーティング機能	「*ルーティング実行時に、ログにデバッグ情報を出力します。* 」
	UI基本モジュール	「*エラーページにエラーの内容を表示します。* 」

環境種別	対象モジュール	概要
運用環境	コアモジュール	「*ベースURLが未設定の場合に警告ログを出力します。*」
	UI基本モジュール	「*script type="text/javascript" タグのラッパータグファイルを最小化します。*」
	UI基本モジュール	「*link rel="stylesheet" タグのラッパータグファイルを最小化します。*」

アクセスコンテキスト情報をログに出力します。

キャッシュされたアクセスコンテキストの取得、および、アクセスコンテキストをキャッシュに格納する際、アクセスコンテキスト情報を ライフサイクル共有情報に格納し、ログを出力します。
アクセスコンテキスト情報とは、以下の２点です。
・アクセスコンテキスト種別
・有効判定の条件（実装ごとに出力される内容は異なります）
　ログ出力するには、jp.co.intra_mart.system.context.impl.cache.SessionContextCachePolicy のログレベルを TRACE にする必要があります。

http://localhost:8080/imart/sample/tutorial/edit にアクセスした際のログ

```
[TRACE] j.c.i.s.c.i.c.SessionContextCachePolicy - [T.IWP.NO_CODE] Use Cach for interface
jp.co.intra_mart.foundation.context.model.ClientContext (Condition = Use Cache: false)
[TRACE] j.c.i.s.c.i.c.SessionContextCachePolicy - [T.IWP.NO_CODE] Use Cach for interface
jp.co.intra_mart.foundation.context.model.AccountContext (Condition = Use Cache: Thu Jul 11 23:59:59 JST 2013)
[TRACE] j.c.i.s.c.i.c.SessionContextCachePolicy - [T.IWP.NO_CODE] Use Cach for interface
jp.co.intra_mart.foundation.user_context.model.UserContext (Condition = Use Cache: Thu Jul 11 23:59:59 JST 2013)
[TRACE] j.c.i.s.c.i.c.SessionContextCachePolicy - [T.IWP.NO_CODE] Use Cach for interface
jp.co.intra_mart.foundation.authz.context.AuthzSubjectContext (Condition = Use Cache: Thu Jul 11 23:59:59 JST 2013)
[TRACE] j.c.i.s.c.i.c.SessionContextCachePolicy - [T.IWP.NO_CODE] Use Cach for interface
jp.co.intra_mart.foundation.context.model.ClientContext (Condition = Use Cache: false)
[TRACE] j.c.i.s.c.i.c.SessionContextCachePolicy - [T.IWP.NO_CODE] Use Cach for interface
jp.co.intra_mart.foundation.context.model.AccountContext (Condition = Use Cache: Thu Jul 11 23:59:59 JST 2013)
[TRACE] j.c.i.s.c.i.c.SessionContextCachePolicy - [T.IWP.NO_CODE] Use Cach for interface
jp.co.intra_mart.foundation.user_context.model.UserContext (Condition = Use Cache: Thu Jul 11 23:59:59 JST 2013)
[TRACE] j.c.i.s.c.i.c.SessionContextCachePolicy - [T.IWP.NO_CODE] Use Cach for interface
jp.co.intra_mart.foundation.authz.context.AuthzSubjectContext (Condition = Use Cache: Thu Jul 11 23:59:59 JST 2013)
```

ベースURLが未設定の場合に警告ログを出力します。

「*ベースURL*」が未設定の場合に警告ログを出力します。
BaseUrlProvider の実装が提供されている場合は出力されません。

Debug#browse 画面にコールスタック情報を表示します。

スクリプト開発モデルの Debug#browse 画面にコールスタック情報を表示します。
また、Debug#printStackTrace メソッドを利用すると、コンソール上にコールスタック情報を出力します。
このコールスタックの取得機能は、im-jssp-config.xml の trace 設定が有効の場合のみ利用可能です。

すべてのスクリプト開発モデルのプログラムをインタプリタモードで実行します。

すべてのスクリプト開発モデルの HTML と JavaScript をインタプリタモードで実行します。
インタプリタモードではソースコードの変更がサーバを再起動することなく反映されます。
「単体テスト環境」と「結合テスト環境」以外を選択した場合のモードは source-config.xml の設定値よって決まります。

同一パスに対して異なる認可設定が存在している場合に警告ログを出力します。

同一パスに対して異なる認可設定をされているかどうかを確認し、異なる認可設定がされているパスが存在した場合、警告ログを出力します。
具体的には、以下の警告ログが出力されます。

```
[W_IWP_ROUTER_AUTHZ_00003] 認可設定が取得できませんでした。 path = {0}
[W_IWP_ROUTER_AUTHZ_00004] 同一パスに対して異なる認可URIが設定されています。 path = {0} uri1 = {1}, uri2 = {2}
[W_IWP_ROUTER_AUTHZ_00005] 同一パスに対して異なる認可アクションが設定されています。 path = {0} action1 = {1}, action2 = {2}
```

デバッグ情報を付与したキークラスを生成します。

認可設定をキャッシュする際のキークラスに対し、デバッグ情報を付与したキークラスを生成します。
その他の場合は必要最低限の情報のみ保持するキークラスが生成されます。

CertificationServletUtil#validateCertificationConfirm()がtrueを返却します。

CertificationServletUtil#validateCertificationConfirm(HttpServletRequest, HttpServletResponse)は、常にtrueを返却します。
これは、認証確認時、遷移先情報（リクエストパラメータ名：im_page_key）は、常に設定されているものとして動作することを意味します。
これにより、認証確認画面に直接アクセスすることを可能とし、認証確認画面自体の動作を確認できます。

検索実行エラー発生時に、開発者向けの詳細メッセージを含めて出力します。

検索実行エラー発生時に出力するメッセージに、開発者向けの詳細メッセージを含めて出力します。

設定ファイル読み込み時に、Instance.PROTOTYPE を渡します。

ContentsSearch系の設定ファイル読み込み時に、ConfigurationLoader#load or #loadAll に渡す Instance(enum) を判別し渡します。
環境種別が、 **単体テスト** の場合：Instance.PROTOTYPE を渡します。
環境種別が、 **単体テスト以外** の場合：Instance.SINGLETON を渡します。

ルーティング実行時に、ログにデバッグ情報を出力します。

ルーティング実行時に、ルーティング処理に要した時間をログに出力します。
ログ出力するには、jp.co.intra_mart.system.router.DebugRouter のログレベルを DEBUG にする必要があります。

```
[DEBUG] j.c.i.s.r.DebugRouter - [D.IWP.NO_CODE] dispatch: 148 [msec]
[DEBUG] j.c.i.s.r.DebugRouter - [D.IWP.NO_CODE] dispatch: 7 [msec]
[DEBUG] j.c.i.s.r.DebugRouter - [D.IWP.NO_CODE] no route: /imart/ui/libs/jquery-validation-1.9.0/jquery.validate.js 0 [msec]
[DEBUG] j.c.i.s.r.DebugRouter - [D.IWP.NO_CODE] no route: /imart/ui/theme/im_theme_dropdown_blue/css/images/ui-
icons_004276_256x240.png 0 [msec]
```

script type=“text/javascript” タグのラッパータグファイルを最小化します。

src に指定された文字列の最後の .js を .min.js に置換して href の文字列として使用します。
「運用環境」以外を選択した場合、 src に指定された文字列をそのまま src の文字列として使用します。
※ min.jsは、同名のjsファイルを圧縮した内容です。

link rel=“stylesheet” タグのラッパータグファイルを最小化します。

href に指定された文字列の最後の .css を .min.css に置換して href の文字列として使用します。
「運用環境」以外を選択した場合、 href に指定された文字列をそのまま href の文字列として使用します。
※ min.cssは、同名のcssファイルを圧縮した内容です。

エラーページにエラーの内容を表示します。

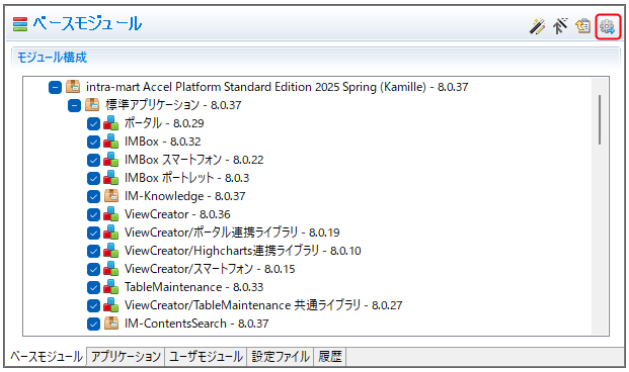
HTTPステータスコードが400番台と500番台のエラー画面で、発生した Exception のスタックトレースを toggle 形式で表示します。

WARファイルに含まれるモジュール情報・ショートモジュールIDの一覧を確認する方法

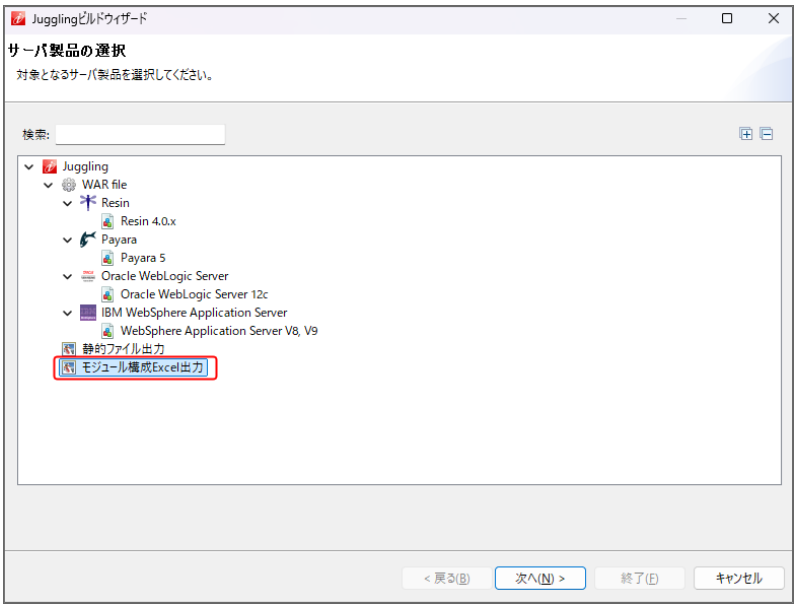
ここでは、既存のモジュール内で定義されているショートモジュールIDの一覧を確認する方法を説明します。

- モジュール構成Excel出力手順

1. IM-Juggling の「juggling.im」を開き、 **「ベースモジュール」** タブを表示します。

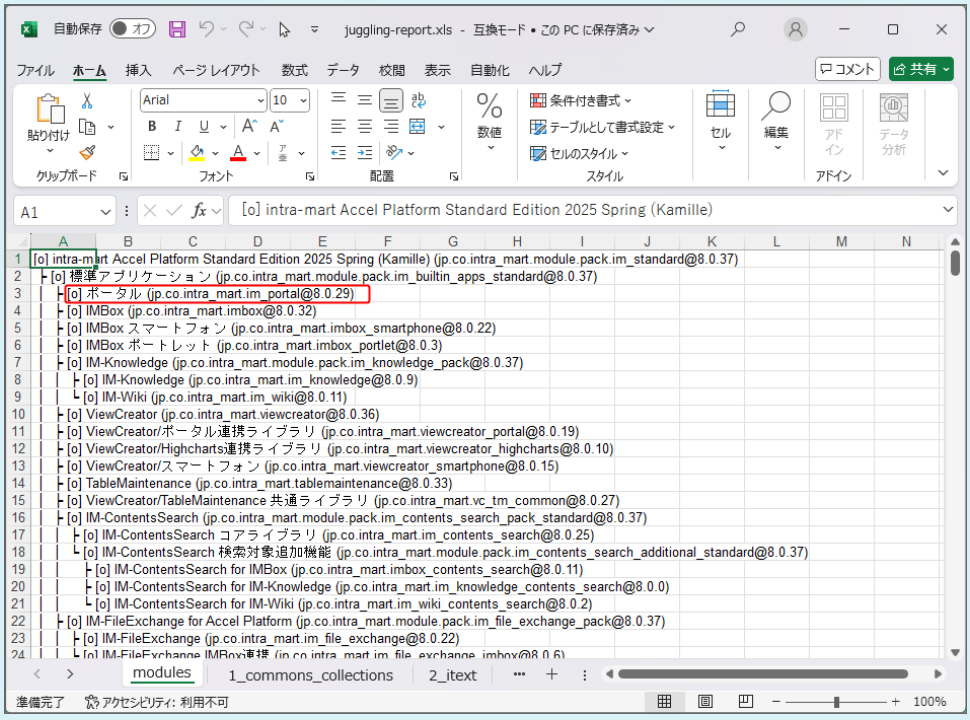


2. 画面右上のビルドウィザードを起動し、 **「モジュール構成Excel出力」** を選択しウィザードに沿ってファイルを出力します。



3. 出力されたExcelファイルを開くと、次のようにモジュール情報を確認する事ができます。

コラム



ポータルモジュールの場合、次の内容を確認する事ができます。

・ポータル (*jp.co.intra_mart.im_portal@8.0.29*)
　モジュール名：ポータル
　モジュールID：jp.co.intra_mart.im_portal
　ショートモジュールID：im_portal
　モジュールバージョン：8.0.29

注意

ユーザアプリケーションとして作成したプログラムをユーザモジュール（immファイル）として作成する場合、**モジュールを識別する、モジュールIDやショートモジュールIDが重複しないようにする必要があります。**

ユーザモジュール作成時にモジュールIDやショートモジュールIDが重複しないことを確認するには、本項記載の方法で一覧を出力して確認してください。

なお、IM-Juggling（1.0-20180801）以降のバージョンでは、ユーザモジュール（immファイル）を追加する場合、モジュールIDが重複した際に以下のようなエラーメッセージが表示されます。

ショートモジュールIDが重複していた場合にも同様のエラーメッセージが表示されるため、ユーザモジュール追加時にも重複しているかどうかの確認が可能です。

intra-mart Accel Platform のチューニング

- intra-mart Accel Platform を最適な運用環境にチューニングをするための負荷試験のポイント、ノウハウについて説明します。

負荷試験実施の際の注意点

- intra-mart Accel Platform で行う負荷試験についての観点、注意点が以下の通りです。

項目

- 負荷試験の目的・性能の目標値の設定
- 負荷試験を実施する観点
- 多重アクセスのシナリオ
- スレッド数等の各設定値
- GCのチューニング
- キャッシュの設定
- Resin のnative機能のコンパイル
- GZip圧縮の検討
- 静的ファイルはAPサーバではなくWEBサーバに配置
- データベース 側のチューニング
- PreparedStatementCache
- VisualVM等を利用してのプロファイリング
- 負荷試験の繰り返し実施

負荷試験の目的・性能の目標値の設定

負荷試験の目的や性能の目標値（どのような環境でどのくらいの性能を出したいのか）をはじめに設定します。
本番環境と同等の検証環境で負荷試験を行う場合や、本番環境の何分の1かでの結果から本番環境での性能が類推比較できる環境であれば問題ありませんが、例えとして、ノートPCに環境を構築して負荷試験を行った場合、その負荷試験の結果を確認しても本番環境での性能値の参考にはなりません。
必ず負荷試験の目的や性能の目標値を明確にしてから、負荷試験環境やその実施内容を検討する必要があります。

負荷試験を実施する観点

どのような観点で何を調査したいのかを明確にする必要があります。観点の例としては以下があります。

- サーバサイドでのレスポンス
- クライアントサイドでのレスポンス
- サーバサイドでのメモリ消費状況
- 自作アプリケーションのパフォーマンス
- データベースにかかる負荷状況

また、一度の負荷検証で抽出した観点をすべて確認するということは、データを取得するだけでも時間を要します。

特にJMeterなどのオープンソース系の負荷試験ツールで負荷試験と、同時にサーバリソース状況を取得するのは困難です。
データ取得後のデータ解析には工数が掛かりますので、予め注意してください。

多重アクセスのシナリオ

負荷試験のシナリオとしてよく見受けられるケースが、全リクエストを同時に実行させるようなシナリオを作成・実施するケースです。
このような負荷は、実際の運用ではまず発生しません。
上記シナリオで取得した数値をもとにサイジング等を実施したとしても、実際の運用ではオーバースペックのサーバ構成となったり、反対にレスポンスが出ないという問題が発生したりします。
シナリオを作成する場合は、多重アクセスの観点以外にも実際の運用に近い負荷を再現できるように、シンクタイム（思考時間遅延）等をシナリオに組み込んで作成・実施を行う検討が必要です。

スレッド数等の各設定値

Resin やintra-martのスレッド数等の各設定値を極端に大きく設定すれば良いという事はありません。
想定される利用者数等を考慮した上で値を設定することが重要です。
設定値は1回設定したら完了ということはなく、幾度も設定値を変更しながら負荷試験を実施し環境に応じた最適値を見つける必要があります。

GCのチューニング

G1GCを使用する、ConcurrentGCを使用する等、どの領域に何を置きたいのかを考えると、適切な値を見つけることは非常に難しい作業です。
特に、WebSphereならGCポリシーを変更するだけで大幅にメモリの使い方が変わります。
VisualVM等を利用して、プロファイリングやGCログでの精査が必要です。

キャッシュの設定

「 *intra-mart Accel Platform の設定ファイル* 」-「 *キャッシュ設定* 」、および、 「 認可仕様書 」を基に適切な値を算出して設定してください。
メモリを考慮しない場合、キャッシュ容量ベースのLRUキャッシュより、数ベースのキャッシュのほうが効果的です。

Resin のnative機能のコンパイル

Resin はFile/IO、Socket/IOの最適化の為に一部nativeな機能を利用しています。
Linux環境の場合、 「./configure –prefix=`pwd` && make && sudo make install」でのインストールを行ってください。
resin.properties にある「sendfile」や「tcp_cork」の設定は native 向けの設定です。

GZip圧縮の検討

WEBサーバを構築して運用を行う場合、GZip圧縮の検討を行います。
なお負荷試験でGZip圧縮を行う場合、負荷をかけるクライアント側が送信するリクエストのヘッダに「Accept-Encoding:gzip, deflate」を入れてください。

静的ファイルはAPサーバではなくWEBサーバに配置

Resin で静的ファイルの処理をさせることは可能ですが、処理を行わせると静的ファイル処理分のスレッドを消費し、本来必要となる処理が回せなくなる可能性もあります。
小規模サイトで無い限り、静的ファイルはWEBサーバに配置する事を推奨します。
WEBサーバは、Apache HTTP Server を推奨します。

データベース 側のチューニング

データベース のチューニングは非常に重要です。必ず負荷試験を兼ねてチューニングを行う必要があります。
チューニングを実施しない場合（例. PostgreSQL を入れただけで shared_buffers がデフォルト設定のままである等）、月末等の処理集中時にパフォーマンス問題を引き起こす事例があります。
また、 **Oracle Database Appliance(ODA)** や **SQL Server SSD Appliance** などのアプライアンスサーバを導入する案もあります。

PreparedStatementCache

テナント環境セットアップ完了後であれば、適切な値を設定してください。
Resin のprepared-statement-cacheはLRUで管理していますので、ある程度大きく設定を行います。

VisualVM等を利用してのプロファイリング

GCのチューニング以外でも、CPUの使用時間、メソッドの呼び出し回数等について、VisualVM等を利用してのプロファイリングを行って確認を行います。

負荷試験の繰り返し実施

1回負荷試験を行っただけでは、レスポンスが期待する結果になる事は極めて低いと考えられます。
結果を基にレスポンス等を改善する方法を検討し、設定と負荷試験の再試行を繰り返し行う必要があります。
CPUの処理速度に依存したり、メモリやサイズだけではなくネットワークトラフィック等の影響も発生する事があります。
さらにディスクIOの影響などあらゆる要因があるため、さまざまな視点で検証・分析を実施することが重要です。

> **コラム**
>
> 以上に関しては、弊社のコンサルティングサービスをご利用頂く事も可能です。詳細は担当営業までお問い合わせください。

負荷試験を実施する際の各種設定

- intra-mart Accel Platform 上で負荷試験を実施する際に、設定の変更が必要となる項目について説明します。

項目

- Resin のJVMオプション設定
- 任意のユーザでログインを行うための設定
  - 多要素認証の確認コード検証を無効化するための設定
- jsspRpc、非推奨であるformタグを利用した業務画面の試験を行うための設定
- 多重アクセスを想定した試験を行うための設定
- クラスタリング環境を想定した負荷試験を行うための設定

Resin のJVMオプション設定

Javaのヒープサイズ等のJVMのオプション設定を変更してください。

```
設定例（設定およびオプション設定）
jvm_args ：-Xmx4096m -Dfile.encoding=UTF-8
```

> **注意**
>
> Linux系OSで、JVMオプションでtempディレクトリの指定を行わない場合、「/tmp」が利用され、cron等の設定により、定期的に「/tmp」配下の内容が削除される設定が標準で組み込まれている場合があります。
> Resin では、作業用ディレクトリに展開した設定ファイル等の変更を検出した際に自動的に再起動が行われる場合があり、回避するためには下記のような設定を行います。
>
> ```
> jvm_args ：-Xmx4096m -Dfile.encoding=UTF-8 -Djava.io.tmpdir=/var/resin-tmp
> ```
>
> ※-Djava.io.tmpdirに指定したディレクトリは事前に作成し、権限設定を行う必要があります。

任意のユーザでログインを行うための設定

セキュア・トークン設定を無効化した上で、「*http://ホスト名:ポート番号/コンテキストパス/certification*」 にパラメータを指定してアクセスすることで任意のユーザで認証することが可能です。
設定方法はWEB-INF/conf/token-filtering-target-config/im_certification.xml の 「/certification」をコメントアウトします。
詳細は以下の通りです。

WEB-INF/conf/token-filtering-target-config/im_certification.xml

```
<p:entry url="/certification">
↓
<!--    <p:entry url="/certification" /> -->
```

これで以下のURLでログイン可能です。

```
http://ホスト名:ポート番号/コンテキストパス/certification?im_user=（ユーザ名）&im_password=（パスワード）
```

また、「&im_url」属性を付加することにより自動ログイン後に任意のアプリケーションの画面に遷移させることも可能です。

```
http://ホスト名:ポート番号/コンテキストパス/certification?im_user=（ユーザ名）&im_password=（パスワード）&im_url=（ログイン後に表示したいパス）
```

> **注意**
>
> デフォルトで無効化されているとおり、この設定を変更することでURL引数やPOSTリクエストでログイン可能でかつ、任意のアプリを実行させることが可能ですので、運用上で利用する場合はセキュリティ観点で問題が無いかを十分に確認の上、ご利用ください。
> （例：リクエストログにURLパラメータが表示されるため、URLパラメータを外す設定を加える等。）

多要素認証の確認コード検証を無効化するための設定

多要素認証機能を有効にしている場合、多要素認証が有効になっているユーザに対してはユーザコード・パスワードでの認証後に確認コードの入力を求められます。
この確認コードの検証機能を無効化し、どのような確認コードが入力された場合でも認証を OK とするためには以下のシステムプロパティを設定します。

```
-Djp.co.intra_mart.system.mfa.extension.app_auth.service.impl.TOTPAuthCodeVerifier.ignore_token_check=true
```

jsspRpc、非推奨であるformタグを利用した業務画面の試験を行うための設定

jsspRpcは、通常セッションを利用したトークンチェックが都度行われるため、ログインを行うごとにトークンチェックに必要なパラメータの値が変更します。このため、負荷試験のシナリオ生成時と同一のリクエストのパラメータを引き渡してもエラーとなってしまいます。
回避するにはWEB-INF/web.xmlを編集し、SecureJSSPServlet、および、jsspRpcServletのサーブレットパラメータ”security”の値を”false”に変更します。

WEB-INF/web.xml

```
<servlet>
  <servlet-name>SecureJSSPServlet</servlet-name>
  <servlet-class>jp.co.intra_mart.system.servlet.jssp.SecureJSSPServlet</servlet-class>
  <init-param>
    <param-name>security</param-name>
    <param-value>false</param-value>
  </init-param>
</servlet>
<servlet>
  <servlet-name>JsspRpcServlet</servlet-name>
  <servlet-class>jp.co.intra_mart.system.servlet.jssp.JsspRpcServlet</servlet-class>
  <init-param>
    <param-name>security</param-name>
    <param-value>false</param-value>
  </init-param>
</servlet>
```

多重アクセスを想定した試験を行うための設定

多重アクセスを想定した試験を行うために、以下の値を設定します。

- リクエスト制御
  intra-mart Accel Platform にはリクエストに対して同時に実行可能な数を制限する機能があります。複数のユーザが同時アクセスを行うような試験を行う場合は、この制限に該当し、エラーとなってしまう可能性あるため、制限数を調整します。サーバの同時処理の制限を解除するには リクエスト制御設定のthreadタグのmax属性に”0”を指定します。
  WEB-INF/conf/request-control-config.xml

  ```
  <request-control-config>
    <thread max="0">
      ...
    </thread>
  </request-control-config>
  ```

  リクエスト制御設定については、「設定ファイルリファレンス」-「サーバの同時処理の制限」を参照してください。
- データソースへの最大コネクション数
  アプリケーションサーバにより、データベースに同時に接続できる数が制限がかかる可能性があります。アプリケーションサーバのデータソース設定にて、同時に接続可能な数が十分な値となることを確認してください。
  アプリケーションサーバに Resin を利用する場合は、resin-web.xmlを編集することにより変更可能です。
  設定方法・詳細は、「設定ファイルリファレンス」-「最大コネクション数設定」を参照してください。

- Resin のスレッド数
  Resin のスレッド数を増やす場合は、以下のファイルの該当箇所を変更してください。
  <%RESIN_HOME%>/conf/resin.properties

```
# Throttle the number of active threads for a port
port_thread_max : 256
accept_thread_max : 32
accept_thread_min : 4
    # keepalive_max : 512
```

クラスタリング環境を想定した負荷試験を行うための設定

クラスタリング環境を想定した負荷試験を行うために、以下の値を設定します。

- 最大通信スレッド数
  サーバ間通信に利用するスレッドの最大数を設定する必要があります。クラスタグループを構成するサーバ台数の 2 倍を目安に設定してください。
  サーバ間通信はバックエンドでスレッドを生成して行われます。クラスタグループを構成するサーバ台数が多くなるとスレッド数が不足しサーバ間の通信が行えない場合があります。
  負荷試験実施時にその様な事象が確認できた場合は、本設定を調整してください。
  省略時のデフォルト値は 10 です。マルチキャスト設定を行っている場合のデフォルト値は 8 です。
  最大通信スレッド数についての詳細は「設定ファイルリファレンス」-「最大通信スレッド数」を参照してください。
  WEB-INF/conf/network-agent-config.xml

```
<network-agent-config>
 <max-threads>10</max-threads>
</network-agent-config>
```

- 到達確認／再送用最大通信スレッド数
  到達確認や再送などの通信に利用するスレッド数を設定する必要があります。最大通信スレッド数と同様に、クラスタグループを構成するサーバ台数の 2 倍を目安に設定してください。
  クラスタグループを構成するサーバ台数が多くなるとスレッド数が不足しサーバ間の通信が行えない場合があります。
  負荷試験実施時にその様な事象が確認できた場合は、本設定を調整してください。
  省略時のデフォルト値は 8 です。
  到達確認／再送用最大通信スレッド数についての詳細は「設定ファイルリファレンス」-「到達確認／再送用最大通信スレッド数」を参照してください。
  WEB-INF/conf/network-agent-config.xml

```
<network-agent-config>
 <max-oob-threads>8</max-oob-threads>
</network-agent-config>
```

- セッション管理 組み込みHazelcast連携 - 通信スレッド数設定
  セッション管理 組み込みHazelcast連携を利用する場合は、クラスタグループ内の通信に利用するスレッド数を別途設定する必要があります。ここで設定した数のスレッドが、セッションの読み込みや書き込み時に利用されます。弊社では、クラスタグループを構成するサーバ台数を設定して負荷試験を行っております。よって、クラスタグループを構成するサーバ台数を目安に設定を行ってください。
  負荷試験実施時にセッションの読み込みや書き込みが遅延する等の事象が確認できた場合は、本設定を調整してください。
  省略時のデフォルト値は 3 です。
  通信スレッド数設定についての詳細は「設定ファイルリファレンス」-「通信スレッド数設定」を参照してください。
  WEB-INF/conf/hazelcast-config.xml

```
<hazelcast-config>
 <base>
  <io-thread-count>3</io-thread-count>
 </base>
</hazelcast-config>
```

> **注意**
> 以上の設定は、負荷試験を行う際に必要となる設定項目です。本番の運用環境では、必ず環境に応じた適切な設定を行うようにします。

バックアップ・リストア（復元）

> **注意**
>
> お客様の運用環境によって、バックアップ・リストア（復元）の対象、および、作業方法が異なります。
> ここで説明する内容は、基本的な内容とその注意点です。実際の作業について、事前に作業の洗い出しと作業手順について十分にご検討頂き、お客様の責任において作業を実施して頂くようお願いします。
> 作業中に発生したデータ破損および、データ不整合による運用障害については、弊社では保証致しかねます。予めご了承ください。

- 弊社製品のアップデートやパッチの適用のためにバックアップを取得する場合、各設定ファイルに変更が加わるため、バックアップ対象データのフルバックアップを行ってください。
- 正常なバックアップ・リストア（復元）を行うためには、関連するシステムの静止点が一致している必要があります。そのため、コールドバックアップ（関連するシステムの停止）を推奨します。

> **コラム**
>
> intra-mart Accel PlatformのStorage領域と、データベースの静止点が異なる場合、整合性が失われ、申請したワークフローが消失するなどの恐れがあります。

項目

- バックアップ対象
  - IM-Juggling関連データ
  - データベース
  - ストレージ
  - Apache Cassandra （IMBoxをご利用の場合）
  - Apache Solr （IM-ContentsSearch for Accel Platformをご利用の場合）
  - ミドルウェア製品
- リストア（復元）手順

バックアップ対象

> **注意**
>
> バーチャルテナントによる各テナントごとのバックアップを取得する場合には、各テナントで使用しているデータベーススキーマ、パブリックストレージ、Apache Cassandra、Apache Solrのデータを取得する必要があります。

IM-Juggling関連データ

必要に応じて、下記のデータをバックアップしてください。

- WARファイル
  - IM-Jugglingからエクスポート（出力）したWARファイル
- 静的コンテンツ
  - IM-Jugglingからエクスポート（出力）、またはWeb Serverの %WEB_PATH% ディレクトリ配下に配置した静的コンテンツ
- IM-Juggling上のプロジェクトファイルや設定ファイル
- リポジトリデータ（製品版メディアイメージ、または、ダウンロードしたリポジトリデータをご利用の場合）
- 独自に作成したアプリケーションのファイル（プログラムソース、設定ファイル、および、ユーザモジュールのファイル）

データベース

データベースデータ（intra-mart Accel Platformが接続している表領域）をエクスポートします。
詳細については、データベースのマニュアルを参照してください。

対象となるデータベースは以下の通りです。

- **システムデータベース** （システムのデータを保存するデータベース）
- **テナントデータベース** （テナント内で利用するデータを保存する、テナントごとのデータベース）
- **シェアードデータベース** （intra-mart Accel Platform外のデータを保存するデータベース）（利用している場合のみ）

ストレージ

- **パブリックストレージ** （%PUBLIC_STORAGE_PATH% ディレクトリ 配下のファイル・ディレクトリ）

  テナント単位で利用されるストレージです。アップロードされたファイルや利用者間で共有したいファイルを保存する領域です。

- **システムストレージ** （%STORAGE_PATH%/system ディレクトリ配下のファイル・ディレクトリ）

  intra-mart AccelPlatform のシステム内部で利用しているシステム領域です。
  システム環境のデータ、各テナントの設定情報等が格納されています。

Apache Cassandra （IMBoxをご利用の場合）

IMBoxの投稿内容の情報です。
詳細は 「Cassandra管理者ガイド」-「スナップショット」を参照してください。

Apache Solr （IM-ContentsSearch for Accel Platformをご利用の場合）

IM-ContentsSearch for Accel Platformの全文検索用インデックス情報です。
詳細は 「Solr管理者ガイド」-「Solrのバックアップ」を参照してください。

ミドルウェア製品

上記のファイルやデータを除き、必要に応じてミドルウェア製品をインストールしたサーバのイメージバックアップやインストールディレクトリのバックアップを行ってください。

- データベース

  詳細はご利用のデータベースのマニュアルを参照してください。

- Resin

  %RESIN_HOME% ディレクトリ配下をバックアップしてください。

- Web Server

  詳細はご利用のWeb Serverのマニュアルを参照してください。

- Apache Cassandra

  Apache Cassandraをセットアップしたディレクトリ配下をバックアップしてください。

- Apache Solr

  Apache Solrを動作しているWeb Application Server（Jetty、Resin、Tomcat）をセットアップしたディレクトリ配下をバックアップしてください。

リストア（復元）手順

> **注意**
> リストア（復元）先は、新規のディレクトリ先を対象としてください。
> 障害発生後の環境（ディレクトリ）へのリストア（復元）は、データ不整合等の事象が発生する可能性があります。

1. どの範囲でリストア（復元）を行うのかを確認します。
   - ミドルウェア製品からリストア（復元）を行う場合

     新しいディレクトリにミドルウェア製品を新規に構築します。
     ミドルウェア製品によっては以前の環境を完全に削除する必要があります。詳細は各製品のマニュアルを参照してください。

   - intra-mart（プログラム資材、運用データ）のみリストア（復元）を行う場合

     バックアップ前と同じミドルウェア環境を利用される場合、既存データが残っているため、各ミドルウェア製品の管理者とリストア先の検討を実施してください（新規スキーマにリストア（復元）する等）。
     バックアップ前と同じディレクトリ先へリストア（復元）する場合は、既存データが削除されている必要があります。

     - アンデプロイの手順は「*WAR ファイルのアンデプロイ*」を参照してください。
     - 運用データについては「*アンインストール*」を参照してください。

2. 取得していた各バックアップファイルを再配置します。
   - データベース

     バックアップしたエクスポートファイルを、新規で作成したスキーマにインポートします。詳細については、データベースのマニュアルを参照してください。

   - Web Server
     仮想ディレクトリ（エイリアス）を設定したディレクトリに、バックアップした静的コンテンツを配置します。

- Storageデータ
  バックアップしたStorageデータを配置します。
- Apache Cassandra
  詳細は 「Cassandra管理者ガイド」-「スナップショットデータによる復旧（リストア）」を参照してください。
- Apache Solr
  「Solr管理者ガイド」-「Solrのバックアップ」にて、バックアップしたデータをSolrの環境に合わせて配置します。

3. Web Application Serverを起動して、WARファイルのデプロイを行ってください。

アンインストール

- この章では intra-mart Accel Platform に関わる全てのファイル、データのアンインストールを行います。
  ミドルウェア製品等のアンインストールについては、同製品のマニュアルをご確認ください。

> **注意**
> バーチャルテナントによる複数テナント で作成した任意のテナントを削除する場合は、「システム管理者ガイド 」-「テナント情報を更新する 」を参照してください。

項目

- WAR ファイルのアンデプロイ
  - webapps ディレクトリに WAR ファイルを直接配置した場合のアンデプロイ
- アプリケーションの削除
- Storage 領域の削除
- ミドルウェア製品の削除

WAR ファイルのアンデプロイ

以下の方法でアンデプロイを行います。

- *webapps ディレクトリに WAR ファイルを直接配置した場合のアンデプロイ*

> **コラム**
> WARファイルによる複数テナントをご利用の場合は、各テナントにアンデプロイを実施してください。

webapps ディレクトリに WAR ファイルを直接配置した場合のアンデプロイ

> **注意**
> Resin が停止している必要があります。

1. 以下の WAR ファイルを手動で削除してください。

   <%RESIN_HOME%/webapps> 配下に配置したWARファイル

2. 以下のファイルを手動で削除してください。

   <%RESIN_HOME%/webapps> 配下に配置した WAR ファイルと同名のディレクトリ

> **注意**
> 分散環境の場合、各サーバ上で実施する必要があります。

アプリケーションの削除

> **注意**
> Resin が停止している必要があります。
> 停止方法については *Web Application Server の起動・停止*を参照してください。

1. Resin 上に展開されている以下のファイル・ディレクトリを削除します。

<%RESIN_HOME%/resin-data> 配下のファイル・ディレクトリ

<%RESIN_HOME%/webapps> 配下のディレクトリ

Storage 領域の削除

1. <%STORAGE_PATH%> 配下のファイル・ディレクトリを削除します。
2. <%PUBLIC_STORAGE_PATH%> 配下のファイル・ディレクトリを削除します。

ミドルウェア製品の削除

- Resin

  <%RESIN_HOME%> ディレクトリを削除します。

- データベース

  ご利用の データベース のマニュアルを参照してください。

- Web Server

  ご利用の Web Server のマニュアルを参照してください。

  > **注意**
  > 静的ファイルを配置している <%WEB_PATH%> ディレクトリ配下のファイル・ディレクトリを削除します。

- Apache Cassandra

  Apache Cassandra をセットアップしたディレクトリを削除します。

- Apache Solr

  Apache Solr を動作している Web Application Server (Jetty 、Resin 、Tomcat) をセットアップしたディレクトリを削除します。

サンプルデータの投入

サンプルデータをご利用される場合は、以下の手順を実行してください。

> **注意**
> テナント環境セットアップの直後に、サンプル・セットアップを実行してください。
> サンプル・セットアップ前にユーザやロールなどのマスタデータが変更されていると、サンプル・セットアップの実行に失敗する可能性があります。

1. システム管理者のメニュー画面を表示します。
   メニューから「テナント環境セットアップ」をクリックします。

2. 「サンプルデータセットアップ」をクリックします。

コラム

テナント環境・サンプルデータセットアップの途中で失敗した場合

接続先のデータベースのデータを削除し、再度テナント環境セットアップを行ってください。
データの削除については *アンインストール* を参照してください。

データの削除後、 Resin を起動し、再度テナント環境セットアップを行ってください。

Apache Cassandra をご利用されている場合は、合わせてデータの削除および起動を行ってください。

セットアップのログについては *テナント環境セットアップ・サンプルデータセットアップに失敗した場合* を参照してください。

セットアップで困ったら・・・

- セットアップ中のトラブルシューティングについて次を参考にしてください。

Web Server

.NET Framework のセットアップ

- intra-mart Accel Platform を稼働させる Windows Server では、必要に応じたバージョンの「.NET Framework」をセットアップを行う必要があります。
  ・ Resin ：「.NET Framework 3.5（.NET 2.0 および3.0を含む）」が必要
  ・ Internet Information Services（IIS） ：「.NET Framework 4（4.5を含む）」が必要

コラム

各OSに.NET Frameworkをインストールする方法の詳細は以下を参照してください。

https://docs.microsoft.com/ja-jp/dotnet/framework/install/ （日本語）
https://docs.microsoft.com/en-us/dotnet/framework/install/ （英語）
https://docs.microsoft.com/zh-cn/dotnet/framework/install/ （中国語）

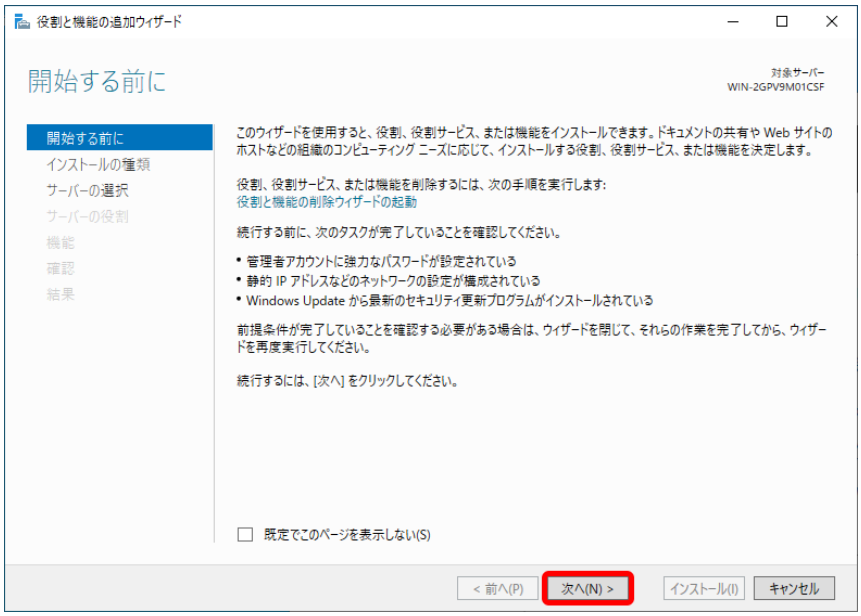
セットアップ手順(Windows Server 2022)

下記のセットアップ手順は、 Windows Server 2022 に「.NET Framework 3.5（.NET 2.0 および3.0を含む）」をセットアップする例です。

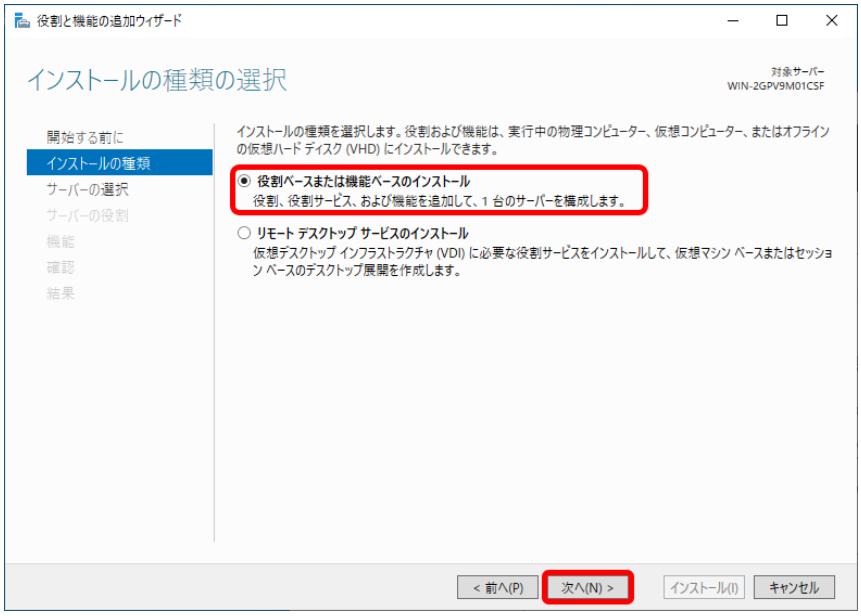
1. Windows Server に付属する「サーバマネージャ」を開きます。

2. 右ペインより「役割と機能の追加」をクリックします。

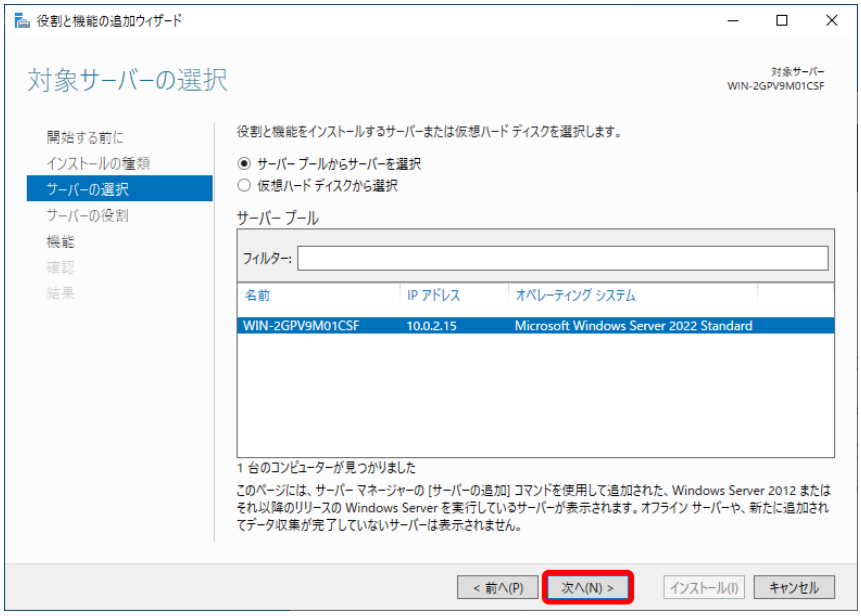
3. 別ウィンドウで表示された「役割と機能の追加ウィザード」に従い、
.NET Framework のセットアップを進めます。
→「次へ（N）＞」をクリックします。



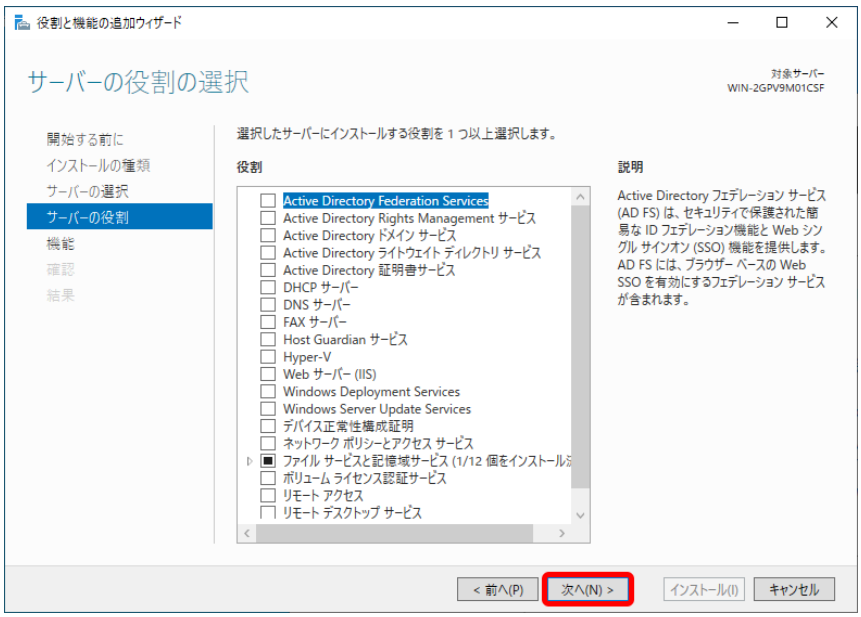
4. 「インストールの種類の選択」より、「役割ベースまたは機能ベースのインストール」を選択します。
   →「次へ（N）＞」をクリックします。



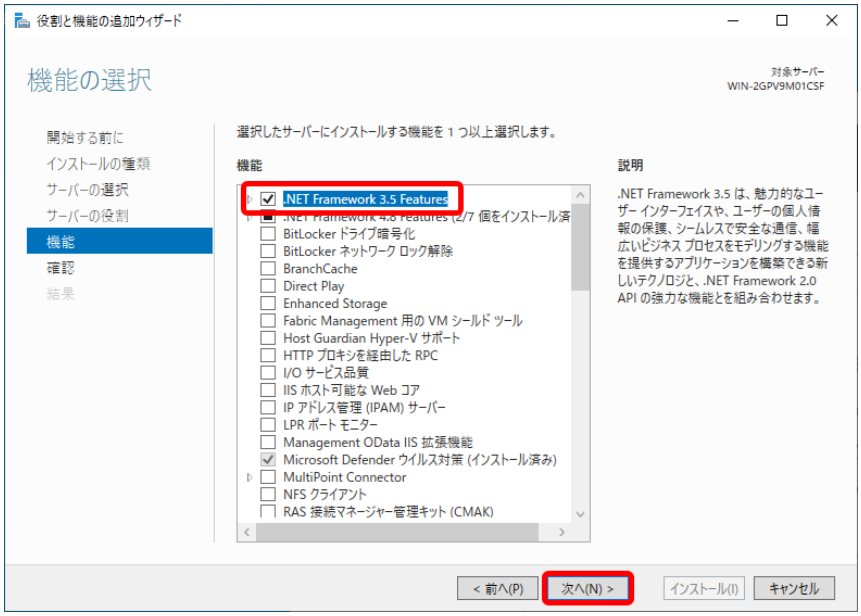
5. 対象サーバを選択します。
   →「次へ（N）＞」をクリックします。



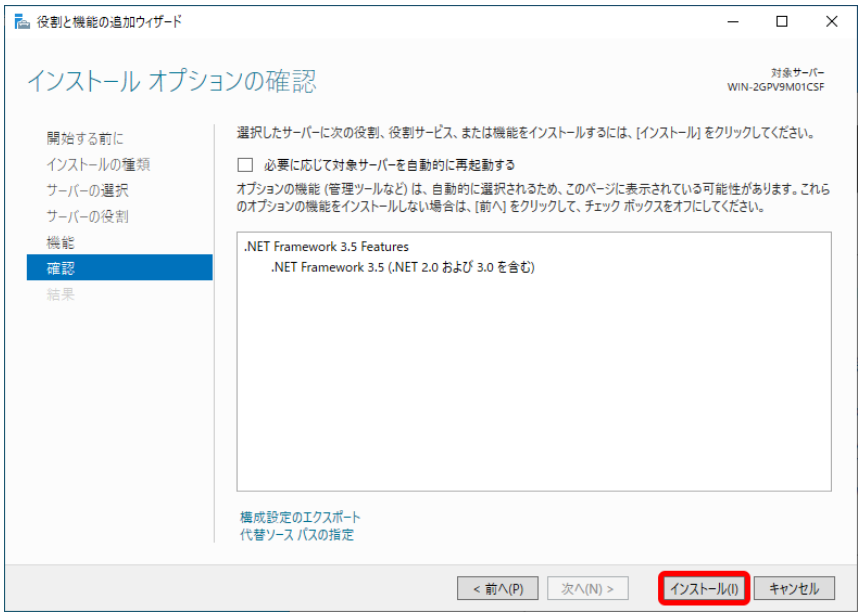
6. 「サーバの役割」の選択は必要ありません。
   →「次へ（N）＞」をクリックします。



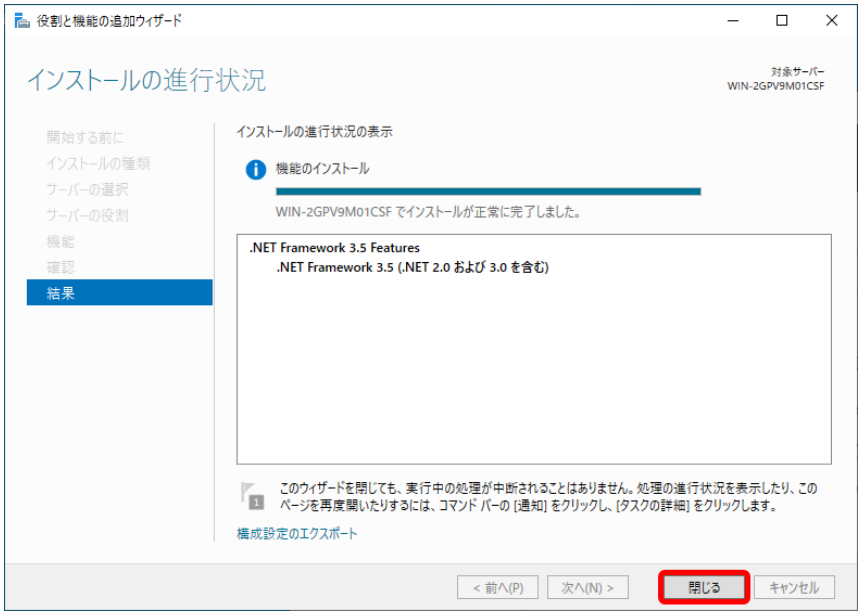
7. 「機能の選択」より、「.NET Framework 3.5 Features」を選択します。
   →「次へ（N）＞」をクリックします。



8. インストール オプションの確認を行います。
   →「インストール（I）」をクリックします。



9. インストールが完了したら「閉じる」をクリックします。
   必要に応じてOSの再起動を行ってください。



IM-Juggling （モジュールの取得、WARの作成）

IM-Juggling を利用中にエラーが発生してしまう場合

- IM-Juggling を利用中に発生するエラーの原因と対処方法を説明します。

IM-Juggling が最新版になっている必要があります

- 最新のモジュールを取得する時に、エラーが発生してしまう場合があります。
  - オンライン状態の場合

    IM-Juggling の起動時に最新状態への自動更新を行うメッセージが表示されます。
    このメッセージにしたがい、 IM-Juggling を最新版に更新してください。

  - オフライン状態の場合

    IM-Juggling の自動更新は、インターネットに接続可能な状態でのみ実行できます。
    インターネットに接続できない環境の場合は、インターネットが接続可能な環境より プロダクトファイルダウンロード より最新版をダウンロードしてください。

- Java 11 以降がインストールされた環境では、起動できない場合があります。
  - オンライン状態の場合

以下の手順で IM-Juggling の自動更新を行ってください。

1. 環境変数に Java 8 以前のパスを設定し、IM-Juggling を起動します。
2. 起動後、最新状態への自動更新を行うメッセージが表示されます。
3. 表示されたメッセージにしたがい、IM-Juggling を最新版に更新します。

- オフライン状態の場合

  IM-Juggling の自動更新は、インターネットに接続可能な状態でのみ実行できます。
  インターネットに接続できない環境の場合は、インターネットが接続可能な環境より プロダクトファイルダウンロード より最新版をダウンロードしてください。

ローカル上のファイルが古い可能性があります。

IM-Juggling を利用してプロジェクトの作成や、WARファイルの作成を行っている際にリポジトリ情報の取得等の
エラーが発生した場合、古い情報が残っているためにエラーとなる場合があります。
下記のディレクトリにあるデータを削除して再度、 IM-Juggling を起動して試行してください。

%OSユーザディレクトリ%/juggling/workspace/.repository ディレクトリ

この古いファイルを削除する事で、最新のデータが再取得され問題を回避します。

WARファイルのデプロイ

Linux環境でWARファイルのデプロイ中にファイル入出力エラーが発生する場合

- Linux環境でwarファイルのデプロイ中にIOException, FileNotFoundException 等が発生した場合の原因と対処方法を説明します。

Web Application Server のエラーログを確認します

- Web Application Server のプロセスが利用できるファイル数の上限がOSにより制限されている場合、エラーログにIOException, FileNotFoundException 等が出力されデプロイに失敗する場合があります。
- Web Application Server 別のエラー内容参照元

Web Application Server	エラー内容の参照元
Resin	起動時のコンソール情報
WebSphere Application Server 9.0.5	%WEBSPHERE_ROOT%/profiles/AppSrv01/logs/server1/SystemErr.log
Oracle WebLogic Server 12c R2(12.2.1)	%WEBLOGIC_ROOT%/%USER_PROJECT%/domains/base_domain/servers/AdminServer/logs/AdminServer.log

ファイルディスクリプタの上限を変更します

/etc/system/limits.conf または/etc/security/limits.confで設定されるOSのファイルディスクリプタ数を環境に合わせた
以下の数を追加することにより、回避できますので、適切な値に変更してください。

```
* soft nofile 32768
* hard nofile 32768
root soft nofile 32768
root hard nofile 32768
```

※ ユーザおよび値はサンプルです、環境に合わせて適切な値を設定してください。

コラム

ファイルディスクリプタの現在の設定値は、

```
ulimit -n
```

で確認できます。

Resin でWARファイルのデプロイ中にエラーが発生する場合

- warファイルのデプロイ時に「タイムアウト」のメッセージが表示された場合の原因と対処方法を説明します。

Resin 起動時のコンソール情報の確認と対処方法

Resin のデプロイ時に次のメッセージが出力されているかを確認します。

```
java.lang.IllegalStateException: future timeout
```

原因と対応方法

このメッセージが確認できた場合、デプロイするwarファイルのサイズが大きく「%RESIN_HOME%/conf/resin.xml」に
指定されている「web-app-deploy dependency-check-interval」より時間がかかる場合（デフォルト2秒）に発生します。

「%RESIN_HOME%/conf/resin.xml」の「dependency-check-interval」の値を大きくします。
値の単位にはs（秒）のほかに Y（年）/M（月）/W（週）/D（日）/h（時）/m（分）/ms（ミリ秒）を設定できます。

```
<dependency-check-interval>300s</dependency-check-interval>
```

※設定値は環境などによって異なる場合があります。

または、「%RESIN_HOME%/conf/resin.xml」の「web-app-deploy」に「redeploy-mode=”manual”」を設定します。

```
<host-default>
 <!-- creates the webapps directory for .war expansion -->
 <web-app-deploy path="webapps" redeploy-mode="manual"
                 expand-preserve-fileset="WEB-INF/work/**"
                 multiversion-routing="${webapp_multiversion_routing}"
                 path-suffix="${elastic_webapp?resin.id:''}"/>
</host-default>
```

コラム

「dependency-check-interval」の値に単位を設定しない場合、s（秒）が適用されます。

コラム

「redeploy-mode=”manual”」を設定する事でwarファイルに対する更新チェックが行われないため、上記メッセージが表示されなくなります。

Resin でWARファイルの再デプロイが正常にできない場合

- *WAR ファイルの再デプロイ*（アップデート適用）、*WAR ファイルの再デプロイ*（パッチ適用）において、
  再デプロイを行ったが、追加したモジュール（資材）がデプロイ先に反映されないといった事象が発生した場合、次の手順を行ってください。

原因

- 稼働するサーバOS上でセキュリティソフトが常駐している場合などに、
  %RESIN_HOME%/resin-data ディレクトリ内のファイルが整合性が取れない状態となる可能性があります。

対応方法

- 「resin-data」ディレクトリ配下のファイルを削除します。

  アンデプロイ（アンデプロイコマンドおよび、<%RESIN_HOME%/webapps/再デプロイ対象のWARファイルのディレクトリ配下>の削除後、
  次のディレクトリ配下のファイル群を削除します。

```
<%RESIN_HOME%/resin-data/*>
```

> **注意**
>
> Resin は停止した状態で行ってください。

削除後、再デプロイを行ってください。

> **注意**
>
> WARファイルによる複数テナント の場合、再デプロイ対象外のテナントも全てアンデプロイ、ファイルの削除を行ってください。

> **注意**
>
> 分散構成として複数の Resin 環境を構築している場合、全ての Resin 環境において、同様の対応を行ってください。

Web Application Server 起動時

Web Application Server 起動後にログインができない（データベースに接続できない）場合

- 接続するデータベース設定に誤りがある可能性があります。この場合の原因の確認方法を説明します。

Web Application Server 起動時のコンソール情報の確認

2019 Summer(Waltz) 以前のバージョンの場合

Web Application Server 起動時に次のメッセージが出力されているかを確認します。

```
[WARN] c.c.s.w.WebApp - [] java.lang.RuntimeException: resource: iwp/platform/schema/exists_table.sql is not found.
java.lang.RuntimeException: resource: iwp/platform/schema/exists_table.sql is not found.
```

このメッセージが確認できた場合、データベース設定に誤りがあります。
より詳しい原因を確認するために下記に説明する「データベースログの設定」を変更します。

2019 Winter(Xanadu) 以降のバージョンの場合

Web Application Server 起動時に次のメッセージが出力されているかを確認します。

```
[ERROR] j.c.i.s.s.PlatformServletContextListener - [] [E.IWP.DATABASE.00024] Failed to connect system database.
jp.co.intra_mart.system.servlet.listener.PlatformLifecycleException: [E.IWP.DATABASE.00024] Failed to connect system database.
```

このメッセージが確認できた場合、データベース設定に誤りがあります。
より詳しい原因を確認するためにコンソールに出力されているスタックトレースを確認してください。

データベースログの設定

Resin の場合、<%RESIN_HOME%/webapps> 配下のwarファイルと同名のディレクトリ /WEB-INF/conf/log/im_logger_database.xml ファイルをエディタで開き、L.43行目を次のように変更します。

変更前 <level value=”off” />
変更後 <level value=”trace” />

編集後、 Web Application Server を再起動します。
この設定変更によりデータベース関連のトレース情報が有効となり、JDBCドライバからのログ情報がコンソール上で確認できます。

> **コラム**
>
> Resin 以外の Web Application Server の場合、各 Web Application Server の管理コーンソールにある、データソース設定画面より、接続確認を行います。

> **コラム**
>
> **この場合における主な原因としては次が考えられます。**
>
> - 接続先のデータベースのIPアドレス、ポート番号、データベースユーザ、パスワードの間違い
> - データベースユーザの権限不足（権限不足により接続ができない。テーブル作成ができない。）

デプロイ直後の Web Application Server 起動時にエラーが発生する場合

Web Application Server の起動ログの一番最初に出力されるエラー内容を確認します。

- デプロイ直後の Web Application Server の起動時にエラーが発生した場合、
  根本となるエラーに起因して別のエラーが発生する事があります。
  このような場合は、 Web Application Server の起動ログの一番最初に出力されるエラー内容を確認してください。
- Web Application Server 別のエラー内容参照元

Web Application Server	エラー内容の参照元
Resin	起動時のコンソール情報
WebSphere Application Server 9.0.5	%WEBSPHERE_ROOT%/profiles/AppSrv01/logs/server1/startServer.log
Oracle WebLogic Server 12c R2(12.2.1)	%WEBLOGIC_ROOT%/%USER_PROJECT%/domains/base_domain/servers/AdminServer/logs/AdminServer.log

初回アクセス時に「[E.IWP.ADMIN.CONTEXT.10004] Tenant ID cannot be resolved.」が発生します。

現象

- 以下のスタックトレースが出力されます。

```
jp.co.intra_mart.foundation.admin.tenant.InvalidTenantIdException: [E.IWP.ADMIN.CONTEXT.10004] Tenant ID cannot be resolved.
  at jp.co.intra_mart.system.admin.context.StandardTenantIdValidator.validate(StandardTenantIdValidator.java:189)
  at jp.co.intra_mart.system.admin.tenant.TenantIdProvider.validate(TenantIdProvider.java:107)
  at jp.co.intra_mart.system.admin.tenant.TenantIdProvider.getTenantId(TenantIdProvider.java:75)
  at
jp.co.intra_mart.system.admin.context.TenantInfoAccountContextDecorator.decorate(TenantInfoAccountContextDecorator.java:23)
  at jp.co.intra_mart.foundation.context.core.ContextBuilderSupport.decorate(ContextBuilderSupport.java:131)
  at jp.co.intra_mart.foundation.context.core.ContextBuilderSupport.build(ContextBuilderSupport.java:54)
  at jp.co.intra_mart.system.context.core.cache.CachingContextBuilderSupport.build(CachingContextBuilderSupport.java:40)
  at jp.co.intra_mart.system.context.impl.command.LifecycleBeginOperation.buildContext(LifecycleBeginOperation.java:96)
  at jp.co.intra_mart.system.context.impl.command.LifecycleBeginOperation.execute(LifecycleBeginOperation.java:64)
  at jp.co.intra_mart.system.context.impl.LifecycleImpl.begin(LifecycleImpl.java:82)
  at jp.co.intra_mart.system.context.web.impl.ContextFilter.doContextFilter(ContextFilter.java:114)
  at jp.co.intra_mart.system.context.web.impl.PreContextFilterChain.doFilter(PreContextFilterChain.java:47)
  at jp.co.intra_mart.system.context.web.impl.ContextFilter.doFilter(ContextFilter.java:78)
  以下省略
```

原因

- IM-SSO または 統合Windows認証モジュールを利用している場合に発生します。
  IM-SSO または 統合Windows認証モジュールのテナントID自動解決機能が、テナントIDを自動解決できなかったためです。

回避方法

- IM-SSO、統合Windows認証をモジュールを利用しない場合
  IM-SSO、統合Windows認証をモジュールを構成から削除し、WARファイルを再作成後、再デプロイします。
  または、以下の URL に従って、機能を無効化します。

IM-SSO	「IM-SecureSignOnを無効化するには」
統合Windows認証	「統合Windows認証機能を無効化するには」

- IM-SSO を利用する場合
  IM-SSO の設定が間違っている可能性があります。以下の URL を参照し、設定を確認してください。

  「認証を行うテナントidの解決方法について」

- 統合Windows認証モジュールを利用する場合

統合Windows認証モジュールの設定が間違っている可能性があります。以下の URL を参照し、設定を確認してください。

「テナント解決プラグインの設定」

テナント環境セットアップ

テナント環境セットアップ・サンプルデータセットアップに失敗した場合

- セットアップに失敗した原因の確認方法を説明します。

セットアップ実行結果ログの確認

テナント環境セットアップ、サンプルデータセットアップの実行結果ログを確認します。

セットアップ実行結果ログの出力先は以下のとおりです。

- intra-mart Accel Platform 2013 Winter 以前
  - テナント環境セットアップの場合

    <STORAGE_PATH>/system/storage/import_result/basic/import-result-detail-data_yyyy-MM-dd_HH-mm-ss.xml

  - サンプルデータセットアップの場合

    <STORAGE_PATH>/system/storage/import_result/sample/import-result-detail-data_yyyy-MM-dd_HH-mm-ss.xml

- intra-mart Accel Platform 2014 Spring 以降
  - テナント環境セットアップの場合

    <STORAGE_PATH>/system/storage/import_result/basic/<テナントID>/import-result-detail-data_yyyy-MM-dd_HH-mm-ss.xml

  - サンプルデータセットアップの場合

    <STORAGE_PATH>/system/storage/import_result/sample/<テナントID>/import-result-detail-data_yyyy-MM-dd_HH-mm-ss.xml

セットアップに失敗した場合、該当の処理では次のようなログが出力されます。（失敗した処理部分のみを抜粋。）
処理結果（import-result-detail-data@success）が false となり、エラーメッセージが出力されます。

```
:
<import-result-detail-data success="false">
  <module-id>im_workflow</module-id>
  <execute-id>36b27536-5a95-46d2-9752-489b261b20a1</execute-id>
  <import-type>DML</import-type>
  <importer-id>jp.co.intra_mart.import.StandardAuthzPolicyXmlImporter</importer-id>
  <target-name>products/import/basic/im_workflow/im_workflow-authz-policy.xml</target-name>
  <message>[E.IWP.IMPORTEXPORT.IMPORTER.10001] データのインポートに失敗しました。 実行クラス =
jp.co.intra_mart.system.authz.services.admin.batch.imex.policy.PolicyXmlImporter, 実行ID = 36b27536-5a95-46d2-9752-
489b261b20a1</message>
</import-result-detail-data>
:
```

より詳しい情報を確認したい場合は、下記の「インポート処理結果ログ」を確認します。

インポート処理結果ログの確認

インポート処理結果ログを確認します。

インポート処理結果ログの出力先は以下の通りです。

<STORAGE_PATH>/system/storage/log/import-export/<%処理実行ID%>.log

<%処理実行ID%> には、セットアップ実行結果ログにおける処理実行ID（execute-id）が適用されます。

セットアップで失敗した場合は、失敗したインポート処理の実行IDによってインポート処理結果ログを特定し、内容を参照することで、例外発生の詳細情報を確認できます。

> **コラム**
> セットアップ実行結果ログの詳細については「テナント環境セットアップ 仕様書」を参照してください。

コラム
例外情報の追跡方法については「ログ 仕様書 - インポートエラーの原因をログから追跡する」もあわせて参照してください。

トランザクション・タイムアウト設定の確認

Webアプリケーションサーバによってはインポート処理中にトランザクション・タイムアウトが発生し、テナント環境セットアップが失敗する場合があります。
トランザクション・タイムアウトが発生した場合タイムアウト時間を長くするよう設定を変更してください。設定方法については下記のガイドを参照してください。

- WebSphereのタイムアウト設定
- WebLogicのタイムアウト設定

テナント環境セットアップ中にタイムアウトが発生した場合

- タイムアウトが発生した場合の原因と対応方法を説明します。

現象・条件

テナント環境セットアップを実施中、Web Application Server 側のログ更新が停止しているのにも関わらず、ブラウザが処理中のステータスのままとなっている（例えば1時間以上経過しているにも関わらず状況が変わらないなど）。

コラム
Web Server 、ブラウザのタイムアウトに限らず、仮にブラウザを閉じてしまっても、テナント環境セットアップ処理はそのまま継続され、Web Application Server 側のログ更新が継続されます。
このような場合は、Web Application Server 側のログ更新が完了するまで待機します。

原因

Web Server 、ブラウザのタイムアウトにより結果画面に遷移していない可能性があります。

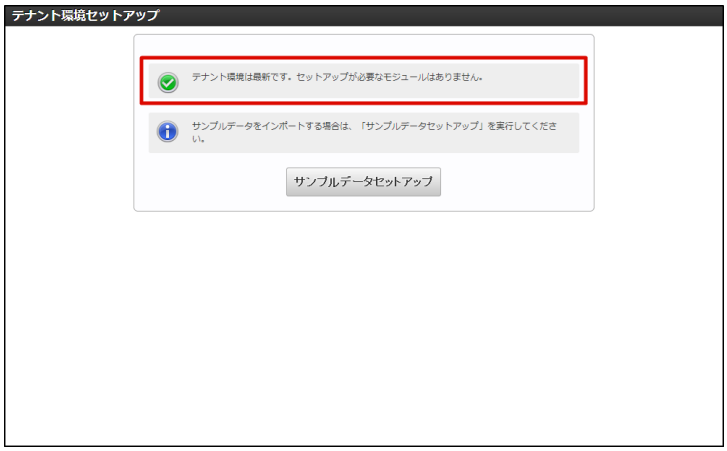
コラム
Web Application Server や データベース 側でエラーが発生する場合、Web Application Server や データベース 側のタイムアウトが原因である可能性があります。

Web Application Server については下記を参照してください。
「*Oracle WebLogic Server 12c R2(12.2.1) でテナント環境セットアップに失敗した場合*」、「 WebSphere Application Server 9.0.5 - トランザクション・タイムアウトの設定 」
データベース については、製品元のドキュメントを参照してください。

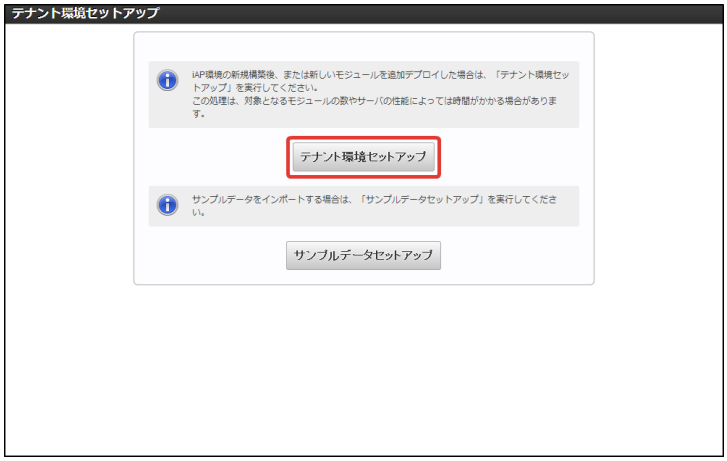
対応方法

- テナント環境セットアップが正常に完了しているかを確認します。
  システム管理者の「メニュー」画面を表示します。
  メニューから「テナント環境セットアップ」をクリックします。

  下図のように、「テナント環境は最新です。セットアップが必要なモジュールはありません。」という旨のメッセージが表示されていれば、テナント環境セットアップは正常に完了しています。



下図のように、「テナント環境セットアップ」ボタンが表示されている場合、テナント環境セットアップは未完了です。「テナント環境セットアップ」ボタンをクリックしセットアップを再度実施します。



> **注意**
>
> 再度セットアップを実施する前に、現象が再発しないよう Web Server 等のタイムアウト設置値を変更します。
> または、Web Application Server 経由でセットアップを実施する事を推奨します。

> **コラム**
>
> サンプルをセットアップするボタンは処理結果に関わらず、常に表示される仕様です。
> このため、サンプルデータ投入中にエラーが発生した場合、再度セットアップを実施すると
> データベース で一意制約違反が発生します。
> この場合は、「*アンインストール*」を行い、改めてセットアップからやり直す事を推奨します。

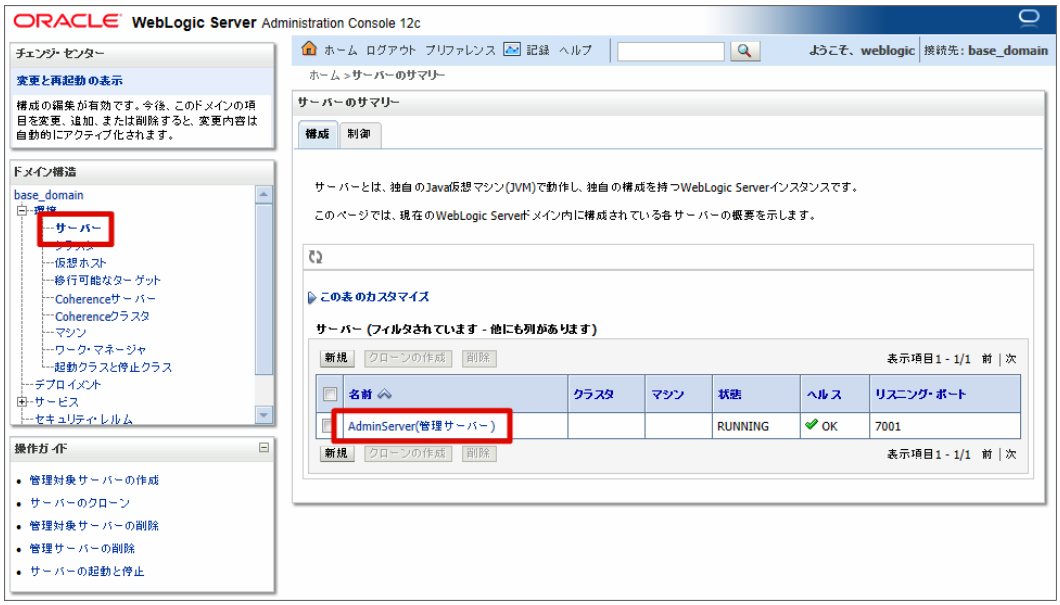
Oracle WebLogic Server 12c R2(12.2.1) でテナント環境セットアップに失敗した場合

- Oracle WebLogic Server 12c R2(12.2.1) 環境でテナント環境セットアップに失敗した場合の原因と対処方法を説明します。

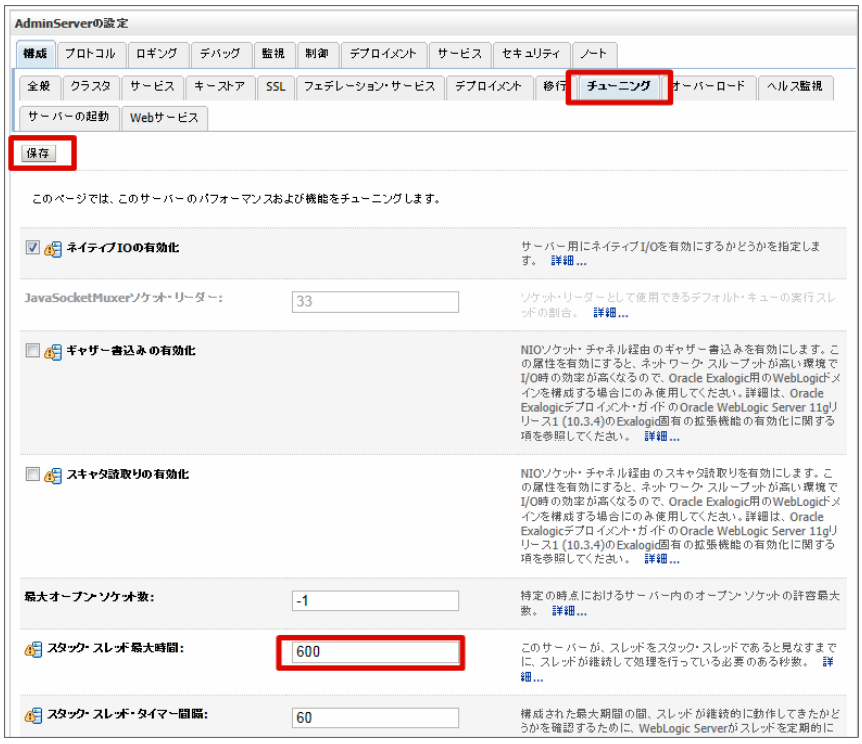
原因と対応方法

データベースの性能等により、テナント環境セットアップに時間がかかりスレッドをスタック状態とみなされセットアップが失敗する場合があります。
下記の手順で設定を変更してください。

1. 左メニューの [環境]-[サーバ] を選択します。
   右画面の使用するサーバを選択します。



2. [チューニング] タブを選択し、[スタック・スレッド最大時間] にデフォルト値より長い時間を入力します。
   入力後、[保存]ボタンをクリックします。



3. Oracle WebLogic Server 12c R2(12.2.1) を再起動します。

テナント環境セットアップ後に、テナント管理者でログインができない場合

- テナント環境セットアップ時に作成したテナント管理者でログインができない、また
  アカウントの設定で「ライセンス」を有効としても更新時にデータベースエラーとなる場合の原因と対処方法を説明します。

発生条件

- この現象は データベース に、 Microsoft SQL Server 2008 R2 、 Microsoft SQL Server 2012 を利用している場合に確認されています。
  これ以外の データベース を利用されている場合においても、下記に挙げる原因によっては該当する事が考えられます。

原因と対応方法

アカウントの設定で「ライセンス」を有効としても、更新時にエラーとなった場合、Resin のコンソール上に次のメッセージが出力されているかを確認します。

```
[ERROR] j.c.i.s.j.i.UserLicenseObject - [] [E.IWP.LICENSE.USER.00003] データベースエラーが発生したため、ライセンスの登録に失敗しました。
jp.co.intra_mart.system.secure.license.exception.LicenseException: [E.IWP.LICENSE.USER.00003] データベースエラーが発生したため、ライセンスの登録に失敗しました。
```

このメッセージが確認できた場合、次の原因が考えられます。

- JDBCドライバのバージョンが古い

  JDBC ドライバ を参照してください。
  この章で記載しているリンク先のJDBCドライバである必要があります。

  > **注意**
  >
  > Microsoft SQL Server 2008 R2 、 Microsoft SQL Server 2012 以外の データベース の場合でも
  > 使用しているJDBCドライバが古いと同様に正常に動作しない可能性が考えられます。

- Resin の照合順序の設定

  データベースサーバ を参照してください。
  この章で記載している照合順序の設定が必要です。

ベースURLを設定した環境でシステム管理者ログイン画面表示時にエラーが発生してしまう場合

- ベースURLを設定した場合に発生するエラーの原因と対処方法を説明します。

適切でないベースURLが設定されていることが考えられます。

- ベースURLにローカルホストなどの不適なURLをしていることが考えられます。

  適切なURLを設定して再度環境セットアップを行ってください。

> **コラム**
>
> ベースURLとは？
>
> intra-mart Accel Platform のシステムを外部から参照する際に利用される基底のURLです。
> 一般的には、 *http://example.org/imart* 等に設定されています。
>
> このベースURLは、あくまでも外部から参照される際に利用するURLです。
> 従って、localhostや、127.0.0.1（ループバックアドレス）等をURLとして指定した場合、外部からの接続時に適切にサーバに接続されません。

Resin で PreparedStatement の キャッシュサイズに大きな値を指定している場合にテナント環境セットアップが失敗する

- 本事象は利用するデータベースによって事象内容が異なります。

弊社での検証済み環境

データベース	事象内容
PostgreSQL	テナント環境セットアップ中にエラーが発生します。 org.postgresql.util.PSQLException: ERROR: キャッシュした計画は結果型を変更してはなりません org.postgresql.util.PSQLException: ERROR: cached plan must not change result type
Oracle	テナント環境セットアップ中に応答がなくなります。 エラーは発生せず、ブラウザ側で処理中のまま応答がなくなります。
SQL Server	テナント環境セットアップが正常に終了します。 弊社での検証済み環境では、本事象は確認されていません。

原因

- PreparedStatementで使用するクエリが参照するテーブル構成が変更されている場合、PreparedStatementのキャッシュにより、テーブル構成が変更する前の古いクエリが発行されてしまうためです。

回避方法

- テナント環境セットアップ処理中のみ resin-web.xml prepared-statement-cache-size を「0」にします。
  ただし、PostgreSQLはJDBCドライバのバージョンによって設定が異なります。
  Version 9.4-1202以降では<init-param>に **preparedStatementCacheQueries** を設定する必要があります。

詳しくは、「設定ファイルリファレンス」の「プリペアドステートメントキャッシュ設定」を参照してください。
テナント環境セットアップ後に再度値を変更してResinの再起動を行ってください。

その他

SAStruts版ポートレットが404エラーでアクセスできない場合

事象内容

- 作成・登録したSAStruts版ポートレットが404でアクセスできない事象が発生します。
  登録したSAStruts版ポートレットにアクセス可能なURLを直接要求すると表示されますが、Web Application Server を再起動すると事象が再発します。

原因と対応方法

- 「*SAStruts用設定ファイル（SAStruts版ポートレットを利用する場合）*」を参照してください。
  ※上記対応を行った後、再度WARを作成し、再デプロイを実施する必要があります。

Apache Solr で利用するJavaのバージョンによって、起動パラメータを設定する必要があります。

- 詳細と対応内容は「 Solr管理者ガイド 」 - 「Java SE Development Kit 7u40 以上を利用する場合 」を参照してください。

Apache POI 5.2.3 を利用している機能で Microsoft Office ファイル読み込み時にエラーとなる場合

- IM-ContentsSearch のクローリングや、IM-LogicDesigner のユーザ定義タスク等 Apache POI 5.2.3 を利用している機能において、Microsoft Office ファイルの読み込みエラーが発生した場合の原因と対処方法を説明します。

原因と対応方法

Apache POI 5.2.3 では読み込み時の最大テキストサイズ上限等がデフォルトで設定されているため、その値を超えるとエラーが出力されます。
intra-mart Accel Platform 2023 Autumn(Hollyhock)（Apache POI 5.2.3-PATCH_002） 以降のバージョンではシステムプロパティから上限値を設定できるようになりました。
エラー例を参考に下記システムプロパティを設定してください。
※大きなファイルを取り扱うリスクもご確認の上設定してください。

- 設定可能なシステムプロパティ

設定内容	プロパティ値	型
レコード長の最大サイズ	jp.co.intra_mart.system.repackage.poi_5_2_3.org.apache.poi.byteArrayMax	integer
ファイル展開時の最大サイズ	jp.co.intra_mart.system.repackage.poi_5_2_3.org.apache.poi.maxEntrySize	long
最大テキストサイズ	jp.co.intra_mart.system.repackage.poi_5_2_3.org.apache.poi.maxTextSize	long
最小圧縮率	jp.co.intra_mart.system.repackage.poi_5_2_3.org.apache.poi.minInflateRatio	double

レコード長のサイズが上限を超過している場合

以下が対象ファイルのレコード長のサイズが上限を超過していた場合のエラー例です。

- エラー例

```
jp.co.intra_mart.system.repackage.poi_5_2_3.org.apache.poi.util.RecordFormatException: Tried to allocate an array of length
xxx,xxx,xxx, but the maximum length for this record type is xxx,xxx,xxx.
If the file is not corrupt and not large, please open an issue on bugzilla to request
increasing the maximum allowable size for this record type.
You can set a higher override value with IOUtils.setByteArrayMaxOverride()
```

jp.co.intra_mart.system.repackage.poi_5_2_3.org.apache.poi.byteArrayMax がレコード長のサイズ（ ログの *Tried to allocate an array of length xxx,xxx,xxx,* ）より大きくなるように設定します。
デフォルトの設定値は 100000000 です。

- 設定例

```
jvm_args : -Djp.co.intra_mart.system.repackage.poi_5_2_3.org.apache.poi.byteArrayMax=200000000
```

ファイル展開時のサイズが上限を超過している場合

以下が対象ファイルの展開サイズが上限を超過していた場合のエラー例です。

- エラー例

```
java.io.IOException: Zip bomb detected! The file would exceed the max size of the expanded data in the zip-file.
This may indicates that the file is used to inflate memory usage and thus could pose a security risk.
You can adjust this limit via ZipSecureFile.setMaxEntrySize() if you need to work with files which are very large.
Uncompressed size: xxxxxxx, Raw/compressed size: xxxxxxx
Limits: MAX_ENTRY_SIZE: xxxxxxx, Entry:xxxxxxx
```

jp.co.intra_mart.system.repackage.poi_5_2_3.org.apache.poi.maxEntrySize がファイルの展開サイズ（ ログの *size: xxxxxxx* ）より大きくなるように設定します。
デフォルトの設定値は 4294967295 です。

- 設定例

```
jvm_args : -Djp.co.intra_mart.system.repackage.poi_5_2_3.org.apache.poi.maxEntrySize=4500000000
```

抽出テキストサイズが上限を超過している場合

以下が対象ファイルの文字数が上限を超過していた場合のエラー例です。

- エラー例

```
java.lang.IllegalStateException: The text would exceed the max allowed overall size of extracted text. By default this is prevented as some documents may exhaust available memory and it may indicate that the file is used to inflate memory usage and thus could pose a security risk. You can adjust this limit via ZipSecureFile.setMaxTextSize() if you need to work with files which have a lot of text. Size: xxxxxxxxxx, limit: MAX_TEXT_SIZE: xxxxxxxxxx
```

jp.co.intra_mart.system.repackage.poi_5_2_3.org.apache.poi.maxTextSize がファイルのテキストサイズ（ ログの *Size: xxxxxxxxxx* ）より大きくなるように設定します。
ログは *MAX_TEXT_SIZE* を超過したタイミングで出力されるため、実際のテキストサイズは *Size: xxxxxxxxxx* より大きい場合があります。
デフォルトの設定値は 10485760 です。

- 設定例

```
jvm_args : -Djp.co.intra_mart.system.repackage.poi_5_2_3.org.apache.poi.maxTextSize=20000000
```

圧縮率が下限を下回る場合

以下が対象ファイルの圧縮率が最小圧縮率より小さい場合のエラー例です。

- エラー例

```
java.io.IOException: Zip bomb detected! The file would exceed the max. ratio of compressed file size to the size of the expanded data.
This may indicate that the file is used to inflate memory usage and thus could pose a security risk.
You can adjust this limit via ZipSecureFile.setMinInflateRatio() if you need to work with files which exceed this limit.
Uncompressed size: xxxxxx, Raw/compressed size: xxxxx, ratio: x.xxxxxxxx
Limits: MIN_INFLATE_RATIO: x.xxxxxxxx, Entry: xxxxxxx
```

jp.co.intra_mart.system.repackage.poi_5_2_3.org.apache.poi.minInflateRatio がファイルの圧縮率（ ログの *ratio: x.xxxxxxxx* ）より小さくなるように設定します。
デフォルトの設定値は 0.01 です。

- 設定例

```
jvm_args : -Djp.co.intra_mart.system.repackage.poi_5_2_3.org.apache.poi.minInflateRatio=0.001
```

DocuWorks Content Filter のインストール方法

項目

- DocuWorks Content Filter について
  - DocuWorks とは
  - DocuWorks Content Filter とは
  - ダウンロードについて
  - ライセンスについて
- DocuWorks 9.1 （Windows環境の場合）のインストール
  - 検証済みバージョン
  - DocuWorks 9.1 の入手
  - DocuWorks 9.1 のインストール
- DocuWorks Content Filter for Linux 7.0 （Linux環境の場合）のインストール
  - 検証済みバージョン
  - DocuWorks Content Filter for Linux 7.0 の入手
  - DocuWorks Content Filter for Linux 7.0 のインストール

DocuWorks Content Filter について

IM-ContentsSearch for Accel Platform 2016 Spring(Maxima) より、**DocuWorks文書（拡張子 .xdw）** および **DocuWorksバインダー（拡張子 .xbd）** ファイルに含まれるテキスト情報を全文検索対象に含めることが可能になりました。
本項ではテキスト抽出処理を実現するために必要な **DocuWorks Content Filter** のインストール方法について記載します。

> **コラム**
>
> DocuWorks は富士フイルムビジネスイノベーション株式会社の商標です。
> 本項における富士フイルムビジネスイノベーション株式会社が提供するサービスやサイトの内容は、2023年10月時点の情報です。

DocuWorks とは

DocuWorks は富士フイルムビジネスイノベーション株式会社が提供するドキュメントハンドリング・ソフトウェアです。
2023年10月時点の最新バージョンは **DocuWorks 9.1（バージョン: 9.1.6）** です。

詳細については下記「**商品情報ページ - DocuWorks 9.1** 」を参照してください。
https://www.fujifilm.com/fb/product/software/docuworks

DocuWorks Content Filter とは

富士フイルムビジネスイノベーション株式会社が提供する **DocuWorks文書** および **DocuWorksバインダー** のファイル内に含まれるテキスト情報を抽出するためのライブラリです。

- Windows版の DocuWorks Content Filter は DocuWorks 9.1 本体に同梱されています。
  DLL として提供されています。
- Linux版の DocuWorks Content Filter は DocuWorks Content Filter for Linux 7.0 として提供されています。
  Linux形式の実行コマンド として提供されています。

ダウンロードについて

DocuWorks 9.1 をダウンロードするには、富士フイルムビジネスイノベーション株式会社が提供している「**富士フイルムBIダイレクト**」サイトにユーザ登録する必要があります。

詳細については下記「**富士フイルムBIダイレクト**」ページを参照してください。
https://www.fujifilm.com/fb/support/direct

> **注意**
>
> 「**富士フイルムBIダイレクト**」サイトでの DocuWorks 9 以前のバージョンおよび DocuWorks Content Filter for Linux 7.0 の新規インストーラのダウンロード提供は2023年1月15日をもって終了しています。
>
> 詳細については下記「**富士フイルムBIダイレクト**」ページを参照してください。
> https://www.fujifilm.com/fb/support/direct

ライセンスについて

DocuWorks 9.1 および DocuWorks Content Filter for Linux 7.0 をご利用される際は、富士フイルムビジネスイノベーション株式会社の使用許諾

条件の準拠してください。

DocuWorks 9.1 （Windows環境の場合）のインストール

intra-mart Accel Platform をインストールするサーバOSに Windows を利用される場合、DocuWorks Content Filter が同梱された DocuWorks 9.1 をインストールする必要があります。
intra-mart Accel Platform が動作する全てのサーバにインストールする必要があります。

検証済みバージョン

弊社では下記サーバOSにて動作を確認しております。

- Windows Server 2022

> **コラム**
>
> DocuWorks 9.1 のリリースノートでは対応OSに Windows Server 2022 は含まれておりませんが、DocuWorks Content Filter の動作については弊社にて動作検証を実施しております。
> DocuWorks Content Filter 以外の機能に関しては動作検証を実施していないため、リリースノート通りにサポート対象外です。

DocuWorks 9.1 の入手

1. 「**富士フイルムBIダイレクト**」サイトを表示してください。
   https://www.fujifilm.com/fb/support/direct
2. 「**富士フイルムBIダイレクト**」サイトにログインします。
3. 「**お客様のご利用サービス**」->「**ソフトウェア/クラウドサービス**」->「**DocuWorks**」を選択してください。
4. 右ペインの「**ダウンロード**」->「**DocuWorks 9.1**」を選択してください。
5. 「**DocuWorks 9.1 体験版・ライセンス認証版**」を展開し、「**ダウンロードページはこちら**」を選択してください。
6. 「**DocuWorks 9.1 体験版・ライセンス認証版**」を展開し、ご利用になるOSに合わせて「**DocuWorks 9.1 体験版・ライセンス認証版（64ビット）**」または「**DocuWorks 9.1 体験版・ライセンス認証版（32ビット）**」のいずれかを選択してください。
7. 「**注意事項**」タブから注意事項の内容を確認してください。
8. 「**使用許諾条件**」タブから使用許諾条件の内容を確認してください。
9. 「**使用許諾条件に同意しダウンロード**」ボタンを押下してください。

> **注意**
>
> 「**富士フイルムBIダイレクト**」サイトを利用するには、事前にユーザ登録を行う必要があります。

DocuWorks 9.1 のインストール

製品に同梱されているリリースノートに記載された「**インストールについて**」を参考にインストールを実施してください。

> **コラム**
>
> DocuWorks 9.1 のインストールには、Microsoft .NET Framework 3.5 がインストールされている必要があります。
>
> インストール手順の詳細については下記「**Windows 8、Windows 8.1、および Windows 10 への .NET Framework 3.5 のインストール**」ページを参照してください。
> https://msdn.microsoft.com/ja-jp/library/hh506443.aspx
>
> 上記のページには Windows Server 2022 の記載はありませんが、同様の手順でインストール可能です。

DocuWorks Content Filter for Linux 7.0 （Linux環境の場合）のインストール

intra-mart Accel Platform をインストールするサーバOSに Linux（Red Hat Enterprise Linux）を利用される場合、DocuWorks Content Filter for Linux 7.0 をインストールする必要があります。
intra-mart Accel Platform が動作する全てのサーバにインストールする必要があります。

検証済みバージョン

弊社では下記サーバOSにて動作を確認しております。

- Red Hat Enterprise Linux 8.8
- Red Hat Enterprise Linux 9.2

DocuWorks Content Filter for Linux 7.0 の入手

お手持ちの DocuWorks Content Filter for Linux 7.0 の新規インストール用モジュールをご利用ください。

> **注意**
>
> 「**富士フイルムBIダイレクト**」サイトでの DocuWorks Content Filter for Linux 7.0 の新規インストール用モジュールの提供は2023年1月15日をもって終了しています。
>
> 詳細については下記「**富士フイルムBIダイレクト**」ページを参照してください。
> https://www.fujifilm.com/fb/support/direct

DocuWorks Content Filter for Linux 7.0 のインストール

製品に同梱されているリリースノートに記載された「**インストールについて**」を参考にインストールを実施してください。
尚、弊社にて検証する際には *yum install* コマンドにて *libstdc++.so.6* をインストールし、 *linux_libstdc++6_rhel52* ディレクトリ配下の *xdw2text* を利用しております。

> **コラム**
>
> インストール後、intra-mart Accel Platform を実行するユーザから有効な実行パス（/usr/local/bin など）に対して、コピーまたはシンボリックリンクを貼ることを推奨します。
> コピーまたはシンボリックリンクを貼ることにより、テキスト抽出設定（solr-extractor-config.xml）において実行ファイルのパス指定を変更する必要がなくなります。

intra-mart Accel Platform のヘルスチェック

- Resin のヘルス機能等で intra-mart Accel Platform の状態を監視する場合、以下のURL をご利用ください。
  ヘルスチェック画面： *http://<HOST>:<PORT>/<CONTEXT_PATH>/availability_check/index.jsp*

  > **注意**
  >
  > 分散環境を構築している場合、それぞれのAP サーバに対して監視を行ってください。
  > 外部からのアクセスでは起動しているAP サーバのヘルスチェック画面にアクセスするため死活監視にはなりません。

- ヘルスチェック画面では以下のチェックを行っており、全て正常であればレスポンスコード「200」を返却します。
  1件でもエラーが発生するとレスポンスコード「503」を返却します。
  - intra-mart Accel Platform 各種サービスの起動確認
  - テナントデータベースの接続確認
  - システムデータベースの接続確認
  - パブリックストレージの接続確認
  - システムストレージの接続確認

  > **コラム**
  >
  > ヘルスチェックを実施する上で intra-mart Accel Platform 各種サービスの起動確認が不要な場合は、以下のURLをご利用ください。
  >
  > **ヘルスチェック画面：** *http://<HOST>:<PORT>/<CONTEXT_PATH>/availability_check/index.jsp?skipServiceAliveMonitoring=true*

  > **コラム**
  >
  > intra-mart Accel Platform 各種サービスは、「サービス仕様書 - intra-mart 各種サービスの概要」を参照してください。