目次

改訂情報

変更年月日	変更内容
2016-04-01	初版
2016-08-01	第2版 下記の誤字を訂正しました。 ・「 *SAML認証のボタン表示方法を制御する* 」の誤字を修正

はじめに

項目

本書の目的

このガイドでは、SAML認証機能の拡張方法および注意点について解説します。

対象読者

- intra-mart Accel Platform を理解している開発者
- Javaを理解している開発者
- SAML 認証機能を利用したい開発者

本書の構成

本書は、以下のような内容で構成されています。

- *ログインフォームの表示方法を制御する*
- *SAML認証のボタン表示方法を制御する*
- *SAML認証時のテナントIDの解決方法をカスタマイズする*

ログインフォームの表示方法を制御する

項目

この機能について

SAML認証のログインボタンが表示されている状態で通常のログインフォームを表示するかどうかを制御します。

実装済みの機能

標準で定義されている機能は以下の通りです。

クラス名	jp.co.intra_mart.foundation.saml.login_form.viewer.SAMLInputFormResolvedTenantViewer
説明	ログイン画面が表示されている時に解決されているテナントIDがパラメータに存在する場合は通常のログインフォームを表示します。 パラメータに値がない場合は常に表示します。
パラメータ	テナントIDをカンマ区切りで設定します。

クラス名	jp.co.intra_mart.foundation.saml.login_form.viewer.SAMLInputFormRegexViewer
説明	ログイン画面のURL(コンテキストパスまで)がパラメータの正規表現に一致した場合は通常のログインフォームを表示します。

パラメータ	ログイン画面のURL(コンテキストパスまで)に対する正規表現を設定します。 【例1】 コンテキストパスまでのURLが **https://tenant1.sample.com/imart** 正規表現が **^http.?://(?!localhost)** 上記の場合は **https://** に一致するためログインフォームが表示されます。 【例2】 コンテキストパスまでのURLが **https://localhost/imart** 正規表現が **^http.?://(?!localhost)** 上記の場合は一致しないためログインフォームが表示されません。

実装方法

ログインフォームの表示方法を制御するクラスの実装は、以下のインタフェースを実装して作成します。

jp.co.intra_mart.foundation.saml.login_form.viewer.SAMLInputFormViewer

以下の関数に処理を実装します。

view(final HttpServletRequest request, final SAMLLoginFormConfiguration loginFormConfig, final String formViewerParam)

ログインフォームを表示する場合はこの関数の戻り値をtrueにします。

【関数のパラメータ】

request	HTTPリクエストです。
loginFormConfig	ログインフォーム表示方法の設定内容です。
formViewerParam	画面で設定したパラメータの内容です。

作成したプログラムはコンパイル後にWEB-INF/classes配下に配置します。

設定方法

1. システム管理者でログインします。
2. ツールバーの「システム管理」-「SAML認証設定」-「SAML認証環境設定」をクリックします。
3. 「表示タイプ」で「動的に判定する」を選択します。

4. 利用したいクラス名とパラメータを設定します。

5. 更新ボタンをクリックして、更新します。

SAML認証のボタン表示方法を制御する

項目

この機能について

対象のIdPに対するSAML認証のログインボタンを表示するかどうかを制御します。

実装済みの機能

標準で定義されている機能は以下の通りです。

クラス名	jp.co.intra_mart.foundation.saml.login_form.viewer.SAMLIdpResolvedTenantViewer
説明	ログイン画面が表示されている時に解決されているテナントIDがパラメータに存在する場合は対象のIdPに対するSAML認証のログインボタンを表示します。 パラメータに値がない場合は常に表示します。
パラメータ	テナントIDをカンマ区切りで設定します。

クラス名	jp.co.intra_mart.foundation.saml.login_form.viewer.SAMLIdpRegexViewer
説明	ログイン画面のURL(コンテキストパスまで)がパラメータの正規表現に一致した場合は対象のIdPに対するSAML認証のログインボタンを表示します。

パラメータ	ログイン画面のURL(コンテキストパスまで)に対する正規表現を設定します。 【例1】 コンテキストパスまでのURLが **https://tenant1.sample.com/imart** 正規表現が **^http.?://(?!localhost)** 上記の場合は **https://** に一致するためSAML認証のログインボタンが表示されます。 【例2】 コンテキストパスまでのURLが **https://localhost/imart** 正規表現が **^http.?://(?!localhost)** 上記の場合は一致しないためSAML認証のログインボタンが表示されません。

実装方法

SAML認証のボタン表示を方法を制御するクラスの実装は、以下のインタフェースを実装して作成します。

jp.co.intra_mart.foundation.saml.login_form.viewer.SAMLIdpViewer

以下の関数に処理を実装します。

view(final HttpServletRequest request, final ProviderConfiguration providerConfig, final String idpViewerParam)

対象のIdPに対するSAML認証のログインボタンを表示する場合はこの関数の戻り値をtrueにします。

【関数のパラメータ】

request	HTTPリクエストです。
providerConfig	プロバイダ設定情報の内容です。
idpViewerParam	画面で設定したパラメータの内容です。

作成したプログラムはコンパイル後にWEB-INF/classes配下に配置します。

設定方法

1. システム管理者でログインします。
2. ツールバーの「システム管理」-「SAML認証設定」-「IdP一覧」をクリックします。

3. ツールバーの「新規登録」または一覧から既存のIdP名のリンクをクリックします。

4. 「SP設定」タブをクリックします。

5. 「IdP表示方法」で「動的に判定する」を選択します。

6. 利用したいクラス名とパラメータを設定します。

7. 更新または新規登録ボタンをクリックして、保存します。

SAML認証時のテナントIDの解決方法をカスタマイズする

項目

この機能について

対象のIdPでSAML認証を行った後のテナントIDの解決方法をカスタマイズします。

実装済みの機能

標準で定義されている機能は以下の通りです。

クラス名	jp.co.intra_mart.foundation.saml.tenant.resolver.SAMLDefaultTenantIdResolver
説明	対象のIdPでSAML認証を行った後のテナントIDはデフォルトのテナントIDを利用して解決します。
パラメータ	使用しません。

クラス名	jp.co.intra_mart.foundation.saml.tenant.resolver.SAMLFixTenantIdResolver
説明	対象のIdPでSAML認証を行った後のテナントIDはパラメータに設定したテナントIDを利用して解決します。
パラメータ	テナントIDを設定します。

クラス名	jp.co.intra_mart.foundation.saml.tenant.resolver.SAMLTenantIdRegexResolver
説明	対象のIdPでSAML認証を行った後のテナントIDはログイン画面のURL(コンテキストパスまで)からパラメータの正規表現で抽出した値をテナントIDとして解決します。 パラメータの正規表現で()で囲まれた部分に一致した文字列をテナントIDとします。 マッチしない場合はテナント解決に失敗しますのでパラメータには必ずマッチする正規表現を設定してください。

パラメータ	ログイン画面のURLからテナントIDを抽出する正規表現を設定します。 正規表現の()で囲まれた部分に一致した文字列がテナントIDとなるように設定してください。 【例】 コンテキストパスまでのURLが **https://tenant1.sample.com/imart** 正規表現が **https://(.+)\.sample\.com** 上記の場合は **tenant1** がテナントIDとして解決されます。

実装方法

SAML認証時のテナントIDの解決方法をカスタマイズするクラスの実装は、以下のインタフェースを実装して作成します。

jp.co.intra_mart.foundation.saml.tenant.resolver.SAMLTenantIdResolver

以下の関数に処理を実装します。

getTenantId(final HttpServletRequest request, ProviderConfiguration providerConfig, String tenantResolverParam)

対象のIdPでSAML認証を行った後のテナントIDを戻り値とします。

【関数のパラメータ】

request	HTTPリクエストです。
providerConfig	プロバイダ設定情報の内容です。
tenantResolverParam	画面で設定したパラメータの内容です。

作成したプログラムはコンパイル後にWEB-INF/classes配下に配置します。

設定方法

1. システム管理者でログインします。
2. ツールバーの「システム管理」-「SAML認証設定」-「IdP一覧」をクリックします。
3. ツールバーの「新規登録」または一覧から既存のIdP名のリンクをクリックします。
4. 「SP設定」タブをクリックします。
5. 「テナント解決方法」で利用したいクラス名とパラメータを設定します。

6. 更新または新規登録ボタンをクリックして、保存します。

注意

「テナント解決方法」の項目はテナントが複数ある場合のみ表示されます。