



- 1. 改訂情報
- 2. はじめに
 - 2.1. 本書の目的
 - 2.2. 対象読者
 - 2.3. 本書の構成
- 3. APIリスト
 - 3.1. APIリストについて
 - 3.2. JavaEE開発モデル
 - 3.3. スクリプト開発モデル
- 4. プログラミング
 - 4.1. 動作概念
 - 4.2. タイムアウトについて
 - 4.3. APIの種類と性質
 - 4.4. プログラム開発における注意点
 - 4.5. 体験版ライセンスにおける注意点
- 5. チュートリアル
 - 5.1. JSPプログラムの作成 (JavaEE開発モデル)
 - 5.2. jsプログラムの作成 (スクリプト開発モデル)
- 6. ステータスコード表
 - 6.1. PDFオートコンバータEX のステータスコード一覧
 - 6.2. 通信関連のステータスコード一覧
- 7. サポート
- 8. 付録
 - 1. IM-PDFAutoConverter for Accel Platform を使って IM-LogicDesigner でファイルをPDFに変換する方法

変更年月日	変更内容
2013-10-11	初版
2014-04-01	第2版
2016-08-01	第3版 下記を追加・変更しました。 ■ エラーコードの記載を追加
2017-12-01	第4版 下記を追加・変更しました。 ■ 「プログラミング」を変更
2018-12-01	第5版 下記を追加・変更しました。 ■ 表記のゆれを訂正
2020-04-01	第6版 下記を追加・変更しました。 ■ Windows 7 / Windows Server 2008 の記述を削除
2020-08-01	第7版 下記を追加・変更しました。 ■ 「はじめに」のトラブルシューティングに関する記載を削除 ■ 「サポート」の内容を変更
2020-12-01	第8版 下記を追加・変更しました。 ■ 「動作概念」の記述を変更 ■ 「PDF変換サーバ環境」を「PDF変換サーバ（Windows）環境」から変更
2021-08-01	第9版 下記を追加・変更しました。 ■ 「PDFオートコンバータEX インストール・ガイド」を更新
2021-12-01	第10版 下記を追加・変更しました。 ■ 「PDFオートコンバータEX インストール・ガイド」を更新
2022-06-01	第11版 下記を追加・変更しました。 ■ 「エラーコード」の見出しを「ステータスコード表」へ変更 ■ 「PDFオートコンバータEX のエラーコード一覧」の見出しを「PDFオートコンバータEX のステータスコード一覧」へ変更 ■ 「PDFオートコンバータEX のステータスコード一覧」の「PDFオートコンバータEX インストール・ガイド」のエラーコード一覧を参照するよう促す文言を削除し、ステータスコードの一覧表、および、注意を追加 ■ 「通信関連のエラーコード一覧」の見出しを「通信関連のステータスコード一覧」へ変更
2022-12-01	第12版 下記を追加・変更しました。 ■ 「チュートリアル」の構成、および、記述を変更 ■ 「JSPプログラムの作成 (JavaEE開発モデル)」を追加 ■ 「jsプログラムの作成 (スクリプト開発モデル)」を追加 ■ 「前提条件」を削除し、記述を「JSPプログラムの作成 (JavaEE開発モデル)」へ移動 ■ 「環境」を削除 ■ 「プログラムの作成」を削除 ■ 「JSPプログラムの作成」を削除し、記述を「JSPプログラムの作成 (JavaEE開発モデル)」へ移動 ■ 「プログラム実行」を削除し、記述を「JSPプログラムの作成 (JavaEE開発モデル)」へ移動

変更年月日	変更内容
2023-04-01	第13版 下記を追加・変更しました。 ■ 「APIリストについて」の スクリプト開発モデル の記述を変更 ■ 「スクリプト開発モデル」の記述を変更 ■ 「APIの種類と性質」の スクリプト開発モデル の記述を変更 ■ 「チュートリアル」に CookBook のコラムを追加
2023-10-01	第14版 下記を追加・変更しました。 ■ 「APIリストについて」のAPIリストの所在を変更 ■ 「本書の目的」の主旨がプログラム開発になるよう記述を変更 ■ 「タイムアウトについて」の変換処理やタイムアウトの仕組みについての記述を変更 ■ 「チュートリアル」のコラムのリンク先を変更 ■ 「JSPファイルの作成」のサンプルプログラムを変更、および、try-catch文について注意を追加 ■ 「jsファイルの作成」のサンプルプログラムを変更、および、try-catch文について注意を追加 ■ 「サポート」のサポート窓口先の記述を変更 ■ 「付録」を追加

目次

- [本書の目的](#)
- [対象読者](#)
- [本書の構成](#)

本書の目的

本書では、IM-PDFAutoConverter for Accel Platform を利用した基本的なプログラム開発や注意点等について説明します。

対象読者

本書は、開発をスムーズに開始するための手引書となっています。

したがって、実際に IM-PDFAutoConverter for Accel Platform を利用したアプリケーションを開発するプログラマの方が対象です。

- 以下のいずれかを理解していることが必須です。
 - JavaEE開発モデル（Java）
 - スクリプト開発モデル（サーバサイドJavaScript）

また、本書は、以下に列挙する技術に関する知識を有することを前提として構成されています。

これらの技術に関して不明な点がある場合、本ドキュメントの内容を正しく理解することが困難になることがありますので、予めご了承ください。

なお、前提知識となる技術に関しては、一般の専門書籍等を参照してください。

- Javaプログラミング言語
- Java Servlet および JSP
- オペレーティングシステム
- ネットワーク

本書の構成

- [API リスト](#)

利用できるAPIについて説明します。

- [プログラミング](#)

プログラム開発の際の注意点や、プログラムの方法などを説明します。

- [チュートリアル](#)

本製品のAPI を利用して実際にプログラムを作成する過程を学びます。

- [ステータスコード表](#)

ステータスコードについて説明します。

- [サポート](#)

製品サポートおよび技術情報の公開について説明します。

目次

■

APIリストについて

■

JavaEE開発モデル

■

スクリプト開発モデル

APIリストについて

IM-PDFAutoConverter for Accel Platform には、JavaEE開発モデル 用のAPI、および、 スクリプト開発モデル 用のAPIが用意されています。

IM-PDFAutoConverter for Accel Platform のAPIリストは、次の通りです。

- [IM-PDFAutoConverter for Accel Platform API ドキュメント](#)

JavaEE開発モデル

IM-PDFAutoConverter for Accel Platform は、JavaEE開発モデル で利用可能なJava-API（クラス）を用意しています。

すべてのクラス
[AutoExRemote](#)

パッケージ **クラス** 使用 階層ツリー 非推奨 API 索引 ヘルプ

前のクラス 次のクラス [フレームあり](#) [フレームなし](#)
概要: [入れ子](#) | [フィールド](#) | [コンストラクタ](#) | [メソッド](#) 詳細: [フィールド](#) | [コンストラクタ](#) | [メソッド](#)

yss.autoconverterex.soap

クラス AutoExRemote

java.lang.Object
↳ yss.autoconverterex.soap.AutoExRemote

public final class **AutoExRemote**
extends java.lang.Object

PDF変換サーバに変換指示を出すクラス。

フィールドの概要

static int	SEC128ACQ_DISABLE	128ビットセキュリティ(アクセス):許可しない
static int	SEC128ACQ_ENABLE	128ビットセキュリティ(アクセス):許可する
static int	SEC128COPY_DISABLE	128ビットセキュリティ(コピー):許可しない
static int	SEC128COPY_ENABLE	128ビットセキュリティ(コピー):許可する
static int	SEC128DOCCHANGE_ADDNOTE	128ビットセキュリティ(文書変更):フォーム入力と注釈追加を許可する
static int	SEC128DOCCHANGE_ASSEMBLE	128ビットセキュリティ(文書変更):アセンブリを許可する
static int	SEC128DOCCHANGE_DISABLE	128ビットセキュリティ(文書変更):許可しない
static int	SEC128DOCCHANGE_ENABLE	128ビットセキュリティ(文書変更):許可する
static int	SEC128DOCCHANGE_FORMATTING	128ビットセキュリティ(文書変更):フォーマットを許可する

スクリプト開発モデル

IM-PDFAutoConverter for Accel Platform は、スクリプト開発モデル で利用可能なスクリプトAPI（クラス）を用意しています。

Class: IMPDFAutoConverter

[Home](#)

Classes

IMPDFAutoConverter

IMPDFAutoConverter()

PDF変換処理のAPIです。
PDFファイルへ変換するための指示を、PDF変換サーバに出します。

Constructor

`new IMPDFAutoConverter()`

インスタンスオブジェクトの作成。

Author: 株式会社ワイ・エス・エス

Returns:

生成されたインスタンスオブジェクト

Example

```
// PDF変換処理のインスタンスを生成します。  
var pdfautoconverter = new IMPDFAutoConverter();
```

Methods

`addStamp(stampName)`

PDF出力時に追加するスタンプ設定名を指定します。
本メソッドを指定した場合、setFolderNameで指定したフォルダ別設定は破棄されます。

Parameters:

目次

- 動作概念
- タイムアウトについて
- APIの種類と性質
- プログラム開発における注意点
- 体験版ライセンスにおける注意点

動作概念

通常の JavaEE 開発モデル スクリプト開発モデル プログラムは、ApplicationRuntime で実行されます。

IM-PDFAutoConverter for Accel Platform で提供されるAPI も、そのほとんどはApplicationRuntime で動作しますが、実際にPDFファイルを生成するのは PDF変換サーバ です。

以下の方法でPDF変換を行うことができます。詳しくは、APIリストを参照してください。

No.	メソッド	説明
1	String convert (String inFilePaht, String outFilePaht)	変換対象ファイルのパスと変換後ファイルのパスを渡して、PDF 変換を行います。いずれのパスもApplicationRuntime側のパスを指定してください。



コラム

PDF ファイル作成の際にサーバ間でネットワーク通信が発生します。したがって、PDF生成処理のレスポンス時間はネットワーク環境に影響を受けます。

タイムアウトについて

PDF変換サーバ に依頼された変換処理は、順番待ち（キュー）の状態となり、一度に大量のPDF変換を指示しても1件ずつ処理される仕組みとなっています。

そのため、大量のPDF変換を指示すると、順番待ちとなったファイルは指示開始から変換処理までに数分以上掛かることもあり、タイムアウトが発生する可能性があります。

大量のPDF変換を指示する場合は、上位アプリケーション側でタイムアウトまでの時間延長設定、または、タイムアウト時にリトライする仕組みなどの対応を検討してください。

AutoExRemote クラスは、以下3種類のタイムアウトを利用します。

詳しくは、APIリストを参照してください。

3種類のタイムアウト設定

- 一定時間経過してもPDF変換処理がはじまらない場合
- 一定時間経過してもPDF変換処理が終わらない場合
- 一定時間経過してもネットワークが返ってこない場合

No.	メソッド	説明
1	setBeforeTimeoutSec (int timeoutSec)	変換前のタイムアウト秒数を設定します。PDF 変換依頼を投げてから、PDF 変換処理を開始するまでのタイムアウト時間を設定します。タイムアウト時間を過ぎると、PDF 変換依頼は削除され次の変換処理に移ります。大量のPDF変換処理をおこなうシステムでは処理のキューが溜まりますので、このタイムアウト値は非常に重要です。通常は、0（タイムアウトしない）を指定してください。
2	setTimeoutSec (int timeoutSec)	変換後のタイムアウト秒数を設定します。PDF 変換処理が開始してからのタイムアウト時間を設定します。タイムアウト時間を過ぎると、変換処理は削除され次の変換処理に移ります。このタイムアウトは、必ず設定してください。
3	setTransTimeoutSec (int timeoutSec)	【必須】 SOAPの接続タイムアウト“ミリ”秒数を設定します。短すぎると結果が受け取れません。



コラム

PDF 変換処理のタイムアウトを変更した場合は、intara-mart のセッションタイムアウトについてもご確認ください。PDF 変換処理が完了する前にintra-mart のセッションタイムアウトが発生した場合、変換結果を受け取ることができません。 intra-mart のセッションタイムアウトの設定については、intra-mart のドキュメントをご確認ください。

APIの種類と性質

IM-PDFAutoConverter for Accel Platform は、次のAPIを用意しています。

- JavaEE開発モデル で利用可能なJava-API（クラス）
- スクリプト開発モデル で利用可能なスクリプトAPI（クラス）

プログラム開発における注意点

IM-PDFAutoConverter for Accel Platform が提供するAPIで変換元ファイルのパスを指定する際には、AppRuntimeからアクセス可能なパスを指定してください。

作成するPDFファイルのサイズによっては、ネットワーク、APIのレスポンス、PDFファイルがディスク上に完全に書き出されるタイミングが大きく異なる場合があります。

特にサイズの大きいPDFファイルを作成する場合は、十分な時間が経過した後に作成したPDF ファイルにアクセスするようにして下さい。

体験版ライセンスにおける注意点

試用版ライセンスでご利用のお客様は、30～60 日間の試用期間が終了するとPDF作成APIが自動的に利用できない状態となります。

この状態でPDF作成APIを利用したプログラムを実行した場合に、実行時エラーとなります。

その場合は、正規の製品ライセンスを購入いただき、アンインストール後に再インストールしてください。

アンインストール・再インストールの方法は、インストールマニュアルをご確認ください。

本項では、IM-PDFAutoConverter for Accel Platform での開発の導入として、APIを利用したPDFへのファイル変換処理を作成することによって、IM-PDFAutoConverter for Accel Platform での開発の流れを体験します。

本項のチュートリアルを開始するにあたっての前提条件は次の通りです。

- intra-mart Accel Platform 、および、IM-PDFAutoConverter for Accel Platform が正しくセットアップされていること。

ここでは、JavaEE開発モデル、スクリプト開発モデルそれぞれについて開発の流れを説明します。

- JavaEE開発モデル

JSPプログラムの作成（JavaEE開発モデル）

JavaEE開発モデルとして、JSPのプログラムを作成します。

準備

本チュートリアルでは、Microsoft Word で作成したファイルを変換対象とします。

「sample.docx」のファイル名で作成し、intra-mart Accel Platform サーバの< C:/temp >ディレクトリに配置してください。

JSPファイルの作成

テキストエディタを使用してJSPファイルを作成します。

Resin の場合、< %RESIN_HOME%/webapps/warファイルと同名のディレクトリ/>の配下に「convert.jsp」の名前でファイルを作成し、次のソースを実装します。


```

1      <%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
2      <%@ page import="yss.autoconverterex.soap.*" %>
3      <%@ page import="yss.autoconverterex.soap.com.exception.AutoExException" %>
4      <%@ taglib prefix="imui" uri="http://www.intra-mart.co.jp/taglib/imui" %>
5      <%
6      String src = "C:/temp/sample.docx";
7      String pdf = "C:/temp/out.pdf";
8      String outpdf = "";
9      String message = "Success !!";
10
11     AutoExRemote ex ;
12     int sts ;
13     String docinfname ;
14     int timeoutsec ;
15
16     /* インスタンスを作成 */
17     ex = new AutoExRemote();
18
19     /******
20     文書情報、セキュリティ、スタンプの設定
21     *****/
22     if( false ) {
23
24         if( false ) {
25             /* 設定済みの文書情報及びセキュリティ設定の名前 */
26             docinfname = "docinf-name";
27         }
28
29         if( docinfname == null ) {
30             /* 文書情報を設定 */
31             ex.setDocInf(
32                 "タイトル",
33                 "サブタイトル",
34                 "作成者",
35                 "アプリケーション",
36                 "キーワード");
37         }
38         else {
39             /* 文書情報を名前で指定して設定 */
40             ex.setDocInfByName(docinfname);
41         }
42
43         if( docinfname == null ) {
44             if( false ) {
45                 /* 40ビットセキュリティの指定 */
46                 ex.setSecurity40("open", "security",
47                     true, true, true, true);
48             }
49             else {
50                 /* 128ビットセキュリティの指定 */
51                 ex.setSecurity128("open", "security",
52                     AutoExRemote.SEC128PRINT_DISABLE,
53                     AutoExRemote.SEC128ACC_DISABLE,
54                     AutoExRemote.SEC128COPY_DISABLE,
55                     AutoExRemote.SEC128DOCCHANGE_DISABLE);
56             }
57         }
58         else {
59             /* 文書情報を名前で指定して設定 */
60             ex.setDocInfByName(docinfname);
61         }
62
63         if( docinfname != null ) {
64             /* 文書情報及びセキュリティを名前で指定して設定 */
65             ex.setDocInfAndSecurity(docinfname);
66         }
67
68         if( false ) {
69             /* スタンプ(名前)の指定 */
70             ex.addStamp("stamp1");
71         }
72
73         /* Web用に最適化の有無 */
74         ex.setFastWebView(true);
75     }

```

```

75     }
76
77     /* プリンタ名の指定 */
78     ex.setPrinter("YSS PDF Converter XP");
79
80     /*
81     変換前の別のタスクの処理に対するタイムアウト(秒)の設定
82     *****/
83     timeoutsec = 60 * 60;
84     timeoutsec = AutoExRemote.TIMEOUT_INFINITE;
85     ex.setBeforeTimeoutSec(timeoutsec);
86
87     /*
88     変換時間に対するタイムアウト(秒)の設定
89     *****/
90     timeoutsec = 60 * 60;
91     timeoutsec = AutoExRemote.TIMEOUT_INFINITE;
92     ex.setTimeoutSec(timeoutsec);
93
94     /*
95     ファイル送信から受信までのタイムアウト (ミリ秒) の設定
96     *****/
97     timeoutsec = 60 * 60 * 1000;
98     timeoutsec = AutoExRemote.TIMEOUT_INFINITE;
99     ex.setTransTimeoutSec(timeoutsec);
100
101     try {
102         /* PDF変換 */
103         outpdf = ex.convert(src, pdf);
104     }
105     catch (Exception e) {
106         e.printStackTrace();
107         message = e.getMessage();
108     }
109
110     %>
111     <imui:head>
112     <title>IM-PDFAutoConverter-チュートリアル-JavaEE開発モデル-convert</title>
113     </imui:head>
114
115     <div class="imui-title">
116     <h1>IM-PDFAutoConverter チュートリアル JavaEE開発モデル convert</h1>
117     </div>
118
119     <div class="imui-form-container">
120     <div class="imui-chapter-title"><h2>実行結果</h2></div>
121     <table class="imui-table">
122     <tbody>
123     <tr>
124     <th class="wd-20">出力PDFファイル</th>
125     <td><%= outpdf %></td>
126     </tr>
127     <tr>
128     <th class="wd-225px">メッセージ</th>
129     <td><%= message %></td>
130     </tr>
131     </tbody>
132     </table>
133     </div>

```

**注意**

文字コードを UTF-8 にして保存してください。

**注意**

PDF変換処理であるAutoExRemote.convert(inFilePath, outFilePath)は、エラー発生時にExceptionをスローします。

エラー情報を取得するために、上記メソッドをtry-catch文で括ってください。

作成したJSPファイルを環境に適用するため、Web Application Server を再起動してください。

再起動後、プログラムをメニューに設定します。

メニュー設定

1. テナント管理者でログインし、次のメニューを設定します。
2. [テナント管理]-[メニュー]画面を開きます。
3. フォルダを作成します。

メニューフォルダの編集

メニューフォルダID *	5i7j7cume8ghkn1	
メニューフォルダ名 *	日本語 *	IM-PDFAutoConverter
	英語	IM-PDFAutoConverter
	中国語	IM-PDFAutoConverter
アイコン画像	☒ ファイルパス	コンテキストパス配下のURLを入力してください。
	☐ CSS Sprites	imui://csssprites/ クラス名を入力してください。

更新

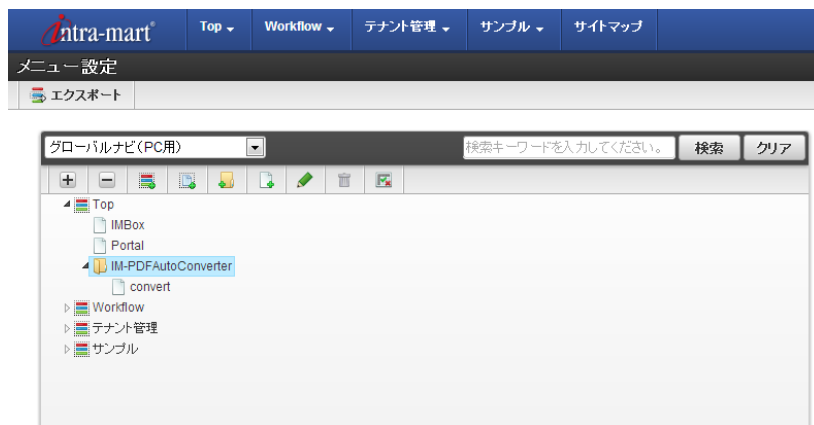
4. URLに、convert.jsp を設定し、メニューを追加します。

メニューアイテムの編集

メニューアイテムID *	5i7j7cuoooodion1													
メニューアイテム名 *	日本語 *	convert												
	英語	convert												
	中国語	convert												
URL *	convert.jsp													
呼び出し方法	GET													
引数	+ 行追加 - 選択行削除 <table border="1"> <thead> <tr> <th>キー</th> <th>値</th> </tr> </thead> <tbody> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> </tbody> </table>		キー	値										
	キー	値												
アイコン画像	☒ ファイルパス ☐ CSS Sprites													
IFRAME表示	☐													
ポップアップ表示	☐													
説明														

更新

5. メニュー設定は完了です。



プログラムの実行と確認

メニューで「convert」を選択することにより、作成したプログラムが実行されます。

実行後は intra-mart Accel Platform サーバの< C:/temp >ディレクトリに、変換されたPDFファイル「out.pdf」が出力されます。

PDFビューア（Adobe Acrobat Reader など）でファイルが正しく表示されることを確認し、このチュートリアルは完了です。

- スクリプト開発モデル

jsプログラムの作成（スクリプト開発モデル）

スクリプト開発モデルとして、HTML/JavaScriptのプログラムを作成します。

準備

本チュートリアルでは、後述で作成する画面から変換対象ファイルをアップロードすることで変換処理を実行します。

変換対象のファイルをご用意ください。

jsファイルの作成

テキストエディタを使用してhtmlファイルとjsファイルを作成します。

Resin の場合、< %RESIN_HOME%/webapps/warファイルと同名のディレクトリ/WEB-INF/jssp/src/pdfa >の配下にそれぞれ「convert.html」「convert.js」の名前でファイルを作成し、次のソースを実装します。

convert.html

```

1  <imart type="head">
2    <title>IM-PDFAutoConverter-チュートリアル-スクリプト開発モデル-convert</title>
3    <script type="text/javascript">
4      $(function(){
5        $("#convert_submit").click(function(){
6          if($("#in_file_path").val().length == 0){
7            imuiAlert("ファイルを選択してください。", "警告");
8            return;
9          }
10         $("#convert_form").submit();
11       });
12     };
13   </script>
14 </imart>
15
16 <div class="imui-title">
17   <h1>IM-PDFAutoConverter チュートリアル スクリプト開発モデル convert</h1>
18 </div>
19
20 <div class="imui-form-container">
21   <div class="imui-chapter-title"><h2>convert プログラム実行</h2></div>
22   <div class="imui-box-supplementation">
23     <div class="supplementation-left-m">
24       <span class="im-ui-icon-common-24-information"></span>
25     </div>
26     <p class="imui-pgh-section supplementation-left-m">
27       アップロードしたファイルをPDFに変換し、変換後PDFをダウンロードします。<br>
28       PDFに変換するファイルを指定し、「PDF変換」ボタンを押下してください。
29     </p>
30   </div>
31   <imart type="form" action="convertPDF" method="POST" id="convert_form" enctype="multipart/form-data">
32     <table class="imui-table">
33       <tbody>
34         <tr>
35           <th class="wd-225px">変換対象ファイル</th>
36           <td><input type="file" id="in_file_path" name="in_file_path"></td>
37         </tr>
38       </tbody>
39     </table>
40     <div class="imui-operation-parts">
41       <imart type="imuiButton" value="PDF変換" class="imui-medium-button" id="convert_submit"></imart>
42     </div>
43   </imart>
44 </div>
45 </div>

```



注意

文字コードを UTF-8 にして保存してください。

convert.js


```

1  /**
2   * アップロードしたファイルをPDFファイルに変換します。
3   * @param {Object} request リクエスト
4   */
5   function convertPDF(request) {
6
7       // リクエストからアップロードしたファイル情報を取得します。
8       let uploadFile = request.getParameter("in_file_path");
9       let inFileStream = uploadFile.getValueAsStream();
10      let inFileName = uploadFile.getFileName();
11
12      // アップロードしたファイルを一時ファイルに保管します。
13      let sessionId = Client.identifier();
14      let inFileExt = inFileName.substr(inFileName.lastIndexOf(".") - inFileName.length);
15      let inFile = File.createTempFile(sessionId, inFileExt, "", false);
16      inFile.save(inFileStream);
17
18      // 出力するPDF ファイルパスを作成します。
19      let outFile = File.createTempFile(sessionId, ".pdf", "", false);
20      let pdfFileStream = null;
21      let errorMessage;
22
23      try {
24          // IM-PDFAutoConverterを実行し、PDFファイルに変換します。
25          let outPdfPath = execPdfautoconverter(inFile.path(), outFile.path());
26
27          // PDFファイルを取得します。
28          let pdfFile = new File(outPdfPath);
29          if(pdfFile.exists()) {
30              pdfFileStream = pdfFile.load();
31          }
32      } catch(e) {
33          errorMessage = e.message + (isUndefined(e.stack) ? '' : '\r\n' + e.stack);
34      }
35
36      // 保存したファイルを削除します。
37      inFile.remove();
38      outFile.remove();
39
40      // 生成したPDF ファイルをダウンロードします。
41      if(pdfFileStream != null) {
42          let pdfFileName = inFileName.substr(0, inFileName.lastIndexOf(".")) + ".pdf";
43          Module.download.send(pdfFileStream, pdfFileName);
44      }
45      else {
46          let logger = Logger.getLogger();
47          logger.error(errorMessage);
48          let response = Web.getHTTPResponse();
49          response.sendError(500, "Failed to convert PDF file.");
50      }
51  }
52  }
53
54  /**
55   * IM-PDFAutoConverterを実行し、PDFファイルに変換します。
56   * @param {String} inFilePath 変換対象のファイルパス
57   * @param {String} outFilePath 変換後の出力先PDFファイルパス
58   * @return {String} 出力したPDFファイルパス
59   */
60  function execPdfautoconverter(inFilePath, outFilePath)
61  {
62      // PDF変換処理のインスタンスを生成します。
63      // @return {Object} PDF変換処理のインスタンス
64      let pdfautoconverter = new IMPDFAutoConverter();
65
66      // PDF出力時に設定する文書情報を指定します。
67      // pdfautoconverter.setDocInf(title, subTitle, creator, app, keyword);
68      // @param {String} title 文書タイトル
69      // @param {String} subTitle 文書サブタイトル
70      // @param {String} creator 作成者
71      // @param {String} app 作成アプリケーション名
72      // @param {String} keyword キーワード
73      pdfautoconverter.setDocInf("文書タイトル", "文書サブタイトル", "作成者", "作成アプリケーション名", "キーワード");
74
75      // PDFファイルの一時保管する文書情報を生成して変換します。

```

```

75 // PDF出力時に設定する文書情報を名前で指定します。
76 // メソッドを指定した場合、setFolderNameで指定したフォルダ別設定は破棄されます。
77 // メソッドを指定した場合、setDocInfで指定した文書情報は破棄されます。
78 // メソッドを指定した場合、setDocInfAndSecurityで指定した文書情報及びセキュリティ設定は破棄されます。
79 // setDocInfByName(name);
80 // @param {String} name 文書情報設定の名称
81 // pdfautoconverter.setDocInfByName("docinf-name");
82
83 // PDF出力時に設定するセキュリティ設定を名前で指定します。
84 // メソッドを指定した場合、setFolderNameで指定したフォルダ別設定は破棄されます。
85 // メソッドを指定した場合、setSecurity40、setSecurity128で指定したセキュリティ設定は破棄されます。
86 // メソッドを指定した場合、setDocInfAndSecurityで指定した文書情報及びセキュリティ設定は破棄されます。
87 // setSecurityByName(name);
88 // @param {String} name セキュリティ設定の名称
89 // pdfautoconverter.setSecurityByName("security-name");
90
91 // PDF出力時に設定する文書情報とセキュリティを名前で指定します。
92 // メソッドを指定した場合、setFolderNameで指定したフォルダ別設定は破棄されます。
93 // メソッドを指定した場合、setDocInfで指定した文書情報は破棄されます。
94 // メソッドを指定した場合、setDocInfByNameで指定した文書情報は破棄されます。
95 // メソッドを指定した場合、setSecurity40、setSecurity128で指定したセキュリティ設定は破棄されます。
96 // メソッドを指定した場合、setSecurityByNameで指定したセキュリティ設定は破棄されます。
97 // setDocInfAndSecurity(name);
98 // @param {String} name 文書情報とセキュリティ設定の名称
99 // pdfautoconverter.setDocInfAndSecurity("docinf_security-name");
100
101 // PDF出力時に設定する40ビットセキュリティ情報を指定します。
102 // メソッドを指定した場合、setFolderNameで指定したフォルダ別設定は破棄されます。
103 // メソッドを指定した場合、setSecurityByNameで指定したセキュリティ設定は破棄されます。
104 // メソッドを指定した場合、setDocInfAndSecurityで指定した文書情報及びセキュリティ設定は破棄されます。
105 // setSecurity40(openPassword, securityPassword, noPrint, noEdit, noCopy, noAddNote);
106 // @param {String} openPassword オープンパスワード
107 // @param {String} securityPassword セキュリティパスワード
108 // @param {boolean} noPrint 印刷を許可しない場合は true、それ以外は false
109 // @param {boolean} noEdit アクセス（編集）を許可しない場合は true、それ以外は false
110 // @param {boolean} noCopy コピーを許可しない場合は true、それ以外は false
111 // @param {boolean} noAddNote 文書変更（注釈追記）を許可しない場合は true、それ以外は false
112 // pdfautoconverter.setSecurity40("open", "security", false, false, false, false);
113
114 // PDF出力時に設定する128ビットセキュリティ情報を指定します。
115 // メソッドを指定した場合、setFolderNameで指定したフォルダ別設定は破棄されます。
116 // メソッドを指定した場合、setSecurityByNameで指定したセキュリティ設定は破棄されます。
117 // メソッドを指定した場合、setDocInfAndSecurityで指定した文書情報及びセキュリティ設定は破棄されます。
118 // setSecurity128(openPassword, securityPassword, print, acc, copy, change);
119 // @param {String} openPassword オープンパスワード
120 // @param {String} securityPassword セキュリティパスワード
121 // @param {String} print 印刷セキュリティを表す文字列
122 // "PRINT_DISABLE" : 許可しない
123 // "PRINT_DEGRADED" : 低解像度で許可する
124 // "PRINT_ENABLE" : 許可する
125 // @param {String} acc アクセス（編集）セキュリティを表す文字列
126 // "ACC_DISABLE" : 許可しない
127 // "ACC_ENABLE" : 許可する
128 // @param {String} copy コピーセキュリティを表す文字列
129 // "COPY_DISABLE" : 許可しない
130 // "COPY_ENABLE" : 許可する
131 // @param {String} change 文書変更（注釈追記）セキュリティを表す文字列
132 // "DOCCHANGE_DISABLE" : 許可しない
133 // "DOCCHANGE_ASSEMBLE" : アセンブリを許可する
134 // "DOCCHANGE_FORMFILL" : フォーム入力を許可する
135 // "DOCCHANGE_ADDNOTE" : フォーム入力と注釈追加を許可する
136 // "DOCCHANGE_ENABLE" : 許可する
137 // pdfautoconverter.setSecurity128("open", "security", "PRINT_DISABLE", "ACC_DISABLE", "COPY_DISABLE",
138 // "DOCCHANGE_DISABLE");
139
140 // フォルダ名及びフォルダ別設定の情報を設定します。
141 // メソッドを指定した場合、その他のメソッドで指定した全ての設定は破棄されます。
142 // setFolderName(name, overwrite);
143 // @param {String} name フォルダ名
144 // @param {boolean} overwrite フォルダ別設定を上書き設定する場合は true、それ以外は false
145 // pdfautoconverter.setFolderName("folder1", false);
146
147 // PDF出力時に追加するスタンプ設定名を指定します。
148 // メソッドを指定した場合、setFolderNameで指定したフォルダ別設定は破棄されます。
149 // addStamp(stampName);

```

```

150 // @param {String} stampName スタンプ設定名
151 // pdfautoconverter.addStamp("stamp1");
152
153 // WEB用に最適化する/しないを指定します。
154 // メソッドを指定した場合、setFolderNameで指定したフォルダ別設定は破棄されます。
155 // setFastWebView(fastWebView);
156 // @param {boolean} fastWebView 最適化する場合は true、それ以外は false
157 pdfautoconverter.setFastWebView(true);
158
159 // PDF出力時に使用するプリンタの名前を指定します。
160 // メソッドを指定した場合、setFolderNameで指定したフォルダ別設定は破棄されます。
161 // setPrinter(name);
162 // @param {String} name プリンタ名
163 pdfautoconverter.setPrinter("YSS PDF Converter XP");
164
165 // 変換前のタイムアウト秒数を設定します。
166 // メソッドを指定した場合、setFolderNameで指定したフォルダ別設定は破棄されます。
167 // setBeforeTimeoutSec(timeoutSec);
168 // @param {Number} timeoutSec タイムアウト時間（秒） 0：タイムアウトしない（待ち続ける）
169 pdfautoconverter.setBeforeTimeoutSec(0);
170
171 // 変換後のタイムアウト秒数を設定します。
172 // メソッドを指定した場合、setFolderNameで指定したフォルダ別設定は破棄されます。
173 // setTimeoutSec(timeoutSec);
174 // @param {Number} timeoutSec タイムアウト時間（秒） 0：タイムアウトしない（待ち続ける）
175 pdfautoconverter.setTimeoutSec(0);
176
177 // PDF変換サーバへのファイル転送のタイムアウトミリ秒を設定します。
178 // setTransTimeoutMilliSec(timeoutMilliSec);
179 // @param {Number} timeoutMilliSec タイムアウト時間（ミリ秒）
180 pdfautoconverter.setTransTimeoutMilliSec(60 * 60 * 1000);
181
182 // 対象ファイルをPDFに変換します。
183 // convert(inFilePath, outFilePath);
184 // @param {String} inFilePath 変換対象のファイルパス
185 // @param {String} outFilePath 変換後の出力先PDFファイルパス
186 // @return {String} 出力したPDFファイルパス
187 return pdfautoconverter.convert(inFilePath, outFilePath);
}

```



注意

文字コードを UTF-8 にして保存してください。



注意

PDF変換処理であるIMPdfAutoConverter.convert(inFilePath, outFilePath)は、エラー発生時にExceptionをスローします。

エラー情報を取得するために、上記メソッドをtry-catch文で括ってください。

ルーティング設定ファイルの作成

ルーティング用の xml（ sample-pdf.xml ）を作成します。次のようにファイルのマッピング情報を記述します。

- 認可の設定に当たる authz-default の mapper 属性には “welcome-all” を設定します。
- URLの設定に当たる file-mapping には、 path 属性に画面のURLを、 page 属性に画面のファイルパスをそれぞれ設定します。

Resin の場合、作成した設定ファイルは < %RESIN_HOME%/webapps/{アプリケーション名}/WEB-INF/conf/routing-jssp-config > の配下に設置してください。

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <routing-jssp-config
3 xmlns="http://www.intra-mart.jp/router/routing-jssp-config" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4 xsi:schemaLocation="http://www.intra-mart.jp/router/routing-jssp-config ../schema/routing-jssp-config.xsd ">
5
6     <authz-default mapper="welcome-all" />
7     <file-mapping path="pdfa/convert" page="pdfa/convert">
8     <authz uri="service://pdfa/convert" action="execute" />
9     </file-mapping>
10
11 </routing-jssp-config>

```



注意

文字コードを UTF-8 にして保存してください。

プログラムの登録

作成したhtmlファイルとjsファイルを環境に適用するため、Web Application Server を再起動してください。

再起動後、プログラムを認可とメニューに設定します。

認可設定

1. テナント管理者でログインし、次のメニューを設定します。
2. [テナント管理]-[認可]画面を開きます。
3. [権限設定を開始する]ボタンを押下します。

認可設定 (画面・処理) - part-id: im_authz_impl_router

🔍 エクスポート ⚠️ キャッシュクリア

リソースの種類: 画面・処理 アクションの種類: 全てのアクション 権限設定を開始する

リソース	アクション	認証		組織		ロール								
		ゲストユーザ	認証済みユーザ	サンプル会社	その他会社	テナント管理者	認可管理者	メニュー管理者	メニュー適用管理者	アカウント管理者	ロール管理者	カレンダー管理者	システム管理者	
画面・処理	実行	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖
intra-mart Accel Platform	実行	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖
welcome-all マッパー	実行	✔	✔	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖
PDF変換サンプル	実行	✔	✔	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖
IM-BIS	実行	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖
BIS監査	実行	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖

4. [リソース]を選択し、[リソースの詳細を開く]押下します。



5. [配下にリソースを新規作成]を押下します。



6. リソースグループを作成します。

リソースの設定

リソースグループID *

8gj3gdrpf1996gs

リソースグループ名 *

日本語 *

英語

中国語 (中国)

リソースURI

?

リソースURIを省略するとリソースグループとして、入力するとリソースとして登録します。

説明

日本語

英語

中国語 (中国)

OK

7. リソースグループ名に PDF変換サンプル を設定します。

リソースの設定

リソースグループID *

8gj3gdrpf1996gs

リソースグループ名 *

日本語 *

PDF変換サンプル

英語

中国語 (中国)

リソースURI

?

リソースURIを省略するとリソースグループとして、入力するとリソースとして登録します。

8. リソースURIに `service://pdfa/convert` を設定します。

リソースの設定

リソースグループID *

8gj3gdrpf1996gs

リソースグループ名 *

日本語 *

PDF変換サンプル

英語

中国語 (中国)

リソースURI

service://pdfa/convert

?

リソースURIを省略するとリソースグループとして、入力するとリソースとして登録します。

9. 作成したリソースグループで「認証済みユーザ」に「全て許可」を付与します。

リソース	アクション	認証		組織	
		ゲスト ユーザ	認証済 みユーザ	サンプ ル会社	その他 会社
		▼	▼	▼	▼
画面・処理	実行 >				
intra-mart Accel Platform	実行 >				
welcome-all マッパ	実行 >	✓	✓		
PDF変換サンプル	実行 >	✓	✓		
IM-BIS	実行 >				
BIS監査	実行 >				

メニュー設定

1. テナント管理者でログインし、次のメニューを設定します。
2. [テナント管理]-[メニュー]画面を開きます。
3. フォルダを作成します。

メニューフォルダの編集

メニューフォルダID *

5i7j7cume8ghkn1

メニューフォルダ名 *

日本語 *

IM-PDFAutoConverter

英語

IM-PDFAutoConverter

中国語

IM-PDFAutoConverter

アイコン画像

☒ ファイルパス

コンテキストパス配下のURLを入力してください。

☐ CSS Sprites

imui://csssprites/ クラス名を入力してください。

更新

4. 作成したフォルダの下にメニューアイテムを新規作成し、URLに `pdfa/convert` を設定します。

メニューアイテムの編集

メニューアイテムID * 517j7cuooodion1

メニューアイテム名 *

日本語 *	convert
英語	convert
中国語	convert

URL * pdfa/test 権限設定

呼び出し方法 GET

引数

+ 行追加 - 選択行削除

キー	値
----	---

アイコン画像

☒ ファイルパス コンテキストパス配下のURLを入力してください。

☐ CSS Sprites imui://csssprites/ クラス名を入力してください。

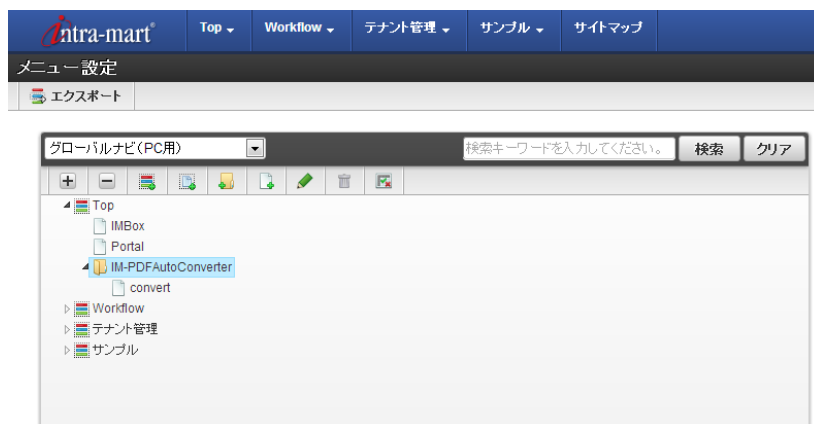
IFRAME表示 ☐

ポップアップ表示 ☐

説明

更新

5. メニュー設定は完了です。



プログラムの実行と確認

メニューで「convert」を選択することにより、作成した画面が表示されます。

画面上で変換対象ファイルをアップロードすることで、変換処理のプログラムが実行され、PDFに変換されたファイルがダウンロードされます。

PDFビューア（Adobe Acrobat Reader など）で変換後のファイルが正しく表示されることを確認し、このチュートリアルは完了です。

i コラム

IM-LogicDesigner 上で、IM-PDFAutoConverter for Accel Platform のAPIを使用し、PDF変換する方法については、「[IM-PDFAutoConverter for Accel Platform を使って IM-LogicDesigner でファイルをPDFに変換する方法](#)」を参照してください。

目次

- [PDFオートコンバータEX のステータスコード一覧](#)
- [通信関連のステータスコード一覧](#)

PDFオートコンバータEX のステータスコード一覧

ステータスコード	内容
11	関数パラメータ不正です。
12	コマンド引数が不正です。
13	必須パラメータが指定されていません。
20	ファイルが開けません。
21	ファイル読み込みエラーです。
22	ファイル書き込みエラーです。
23	データが不正です。
24	プロセス起動エラーです。
25	ディレクトリアクセスエラーです。
26	印刷処理エラーです。
27	パスワードエラーです。
28	ライブラリがロードできないか、関数が見つかりません。
29	起動したプロセスがエラー終了しました。
30	正しくセットアップされていません。
31	システムコールエラーです。
32	ライセンスエラーです。
33	画像処理エラーです。
34	0割が発生しました。
35	計算エラーが発生しました。
200	内部エラーが発生しました。
201	メモリエラーです。
601	既に起動されています。
602	ターゲットフォルダが存在しません。
603	プリンタが見つかりません。
604	変換処理がタイムアウトしました。
605	変換モジュールが正常を返しましたが、出力がありませんでした。
606	変換モジュールが正常を返しましたが、出力サイズが0でした。
607	拡張子がない為、処理しませんでした。
608	デフォルトの除外拡張子の為、処理しませんでした。
609	変換対象拡張子でない為、処理しませんでした。
610	後処理DLLがエラーを返しました。
611	変換モジュールがエラーを返しました。
612	この拡張子は、シェル(print/printto)印刷をサポートしていません。
613	多重起動監視でタイムアウトが発生しました。

ステータスコード	内容
614	変換対象のファイルサイズが0です。
615	全てのページがコンテンツ無しです。
616	コンテンツ無しページを検出しました。
617	PDF コンバータ(プリンタ)でエラーが発生しました。[Status:コンバータエラー番号]
701	変換対象のページがありませんでした。
751	ワード文書変換エラー
752	ワード環境設定エラー
753	ワード初期化エラー
754	ワード文書オープンエラー(パスワード付きか、存在しない可能性があります)
755	ワードプリンタ設定エラー
756	ワード印刷時エラー
757	ワードリンク情報解析エラー
758	ワード変更履歴非表示エラー
801	エクセル文書変換エラー
802	エクセル環境設定エラー
803	エクセル初期化エラー
804	エクセル文書オープンエラー(パスワード付きか、存在しない可能性があります)
805	エクセルプリンタ設定エラー
806	エクセル印刷時エラー
807	エクセルヘッダフッタ操作時エラー
808	エクセル一括変換エラー
851	パワーポイント文書変換エラー
852	パワーポイント環境設定エラー
853	パワーポイント初期化エラー
854	パワーポイント文書オープンエラー(パスワード付きか、存在しない可能性があります)
855	パワーポイントプリンタ設定エラー
856	パワーポイント印刷時エラー
861	イメージ変換エラー
871	一太郎文書変換エラー
9998	EOFエラーです。
9999	予期しないエラーが発生しました。



注意

「611」のエラーが発生した場合は、「[PDFオートコンバータEX インストール・ガイド](#)」の「6.1 [611] エラー抑止（必須設定）」を参照してください。



注意

「617」のエラーが発生した場合の、コンバータエラー番号については次の通りです。

コンバータエラー番号	内容
-300	PDFコンバータ印刷時の一時ファイルサイズが制限サイズを超えたため、処理を中止しました。

ステータスコード 内容

-1	SOAPでの通信処理でエラーが発生した場合
----	-----------------------

Webにて当製品に対するサポート、および、技術情報を公開しています。

当製品に関して不明な点などがある場合、情報検索、または、「[intra-mart サポートサイト](#)」に問い合わせしてください。

IM-PDFAutoConverter for Accel Platform を使って IM-LogicDesigner でファイルをPDFに変換する方法

本項では、IM-LogicDesigner の JavaScript定義 で、IM-PDFAutoConverter for Accel Platform のAPIを使用したPDF変換について紹介しています。



注意

本サンプルは、IM-PDFAutoConverter for Accel Platform 2022 Winter 以降のバージョンが必要です。

次の完成サンプルをダウンロードし活用してください。

■ [cookbook_im_cookbook_8004-1.0.0.imm](#)

immファイルを適用すると、パブリックストレージ配下に、サンプル実行に必要な次のファイルが設置されます。

フォルダ名／ファイル名	説明
%PUBLIC_STORAGE_PATH%	パブリックストレージパス
im_cookbook_8004 / pdfa_logic	IM-LogicDesigner PDF変換サンプル用フォルダ
import / im_cookbook_8004.zip	PDF変換用ユーザ定義、フロー定義をまとめたzipファイル
data / sample.docx	PDF変換サンプル用のdocxファイル

zipファイル< im_cookbook_8004.zip >を、IM-LogicDesigner のインポート画面からインポートしてください。

本サンプルのユーザ定義（JavaScript定義）では、IM-PDFAutoConverter for Accel Platform の スクリプト開発モデル 用APIを使用しています。

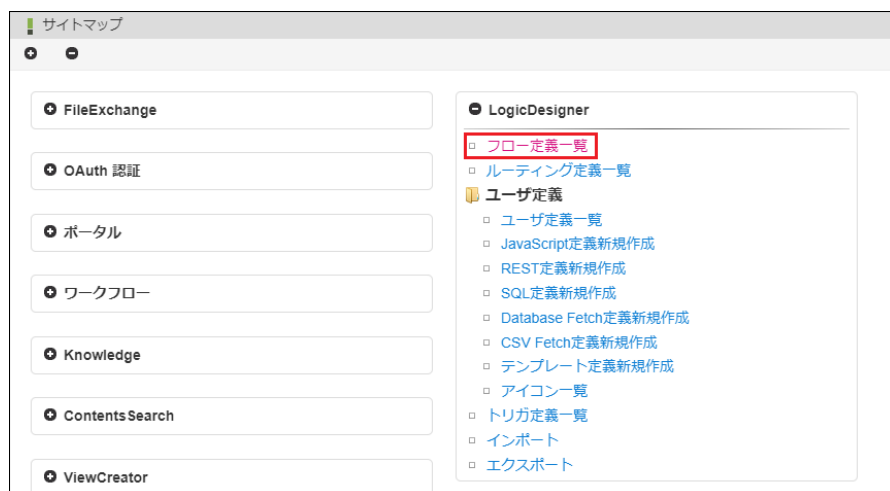
サンプルの実行手順、ユーザ定義の詳細、および、サンプル内で使用しているAPIについては、次を参照してください。

サンプル実行手順

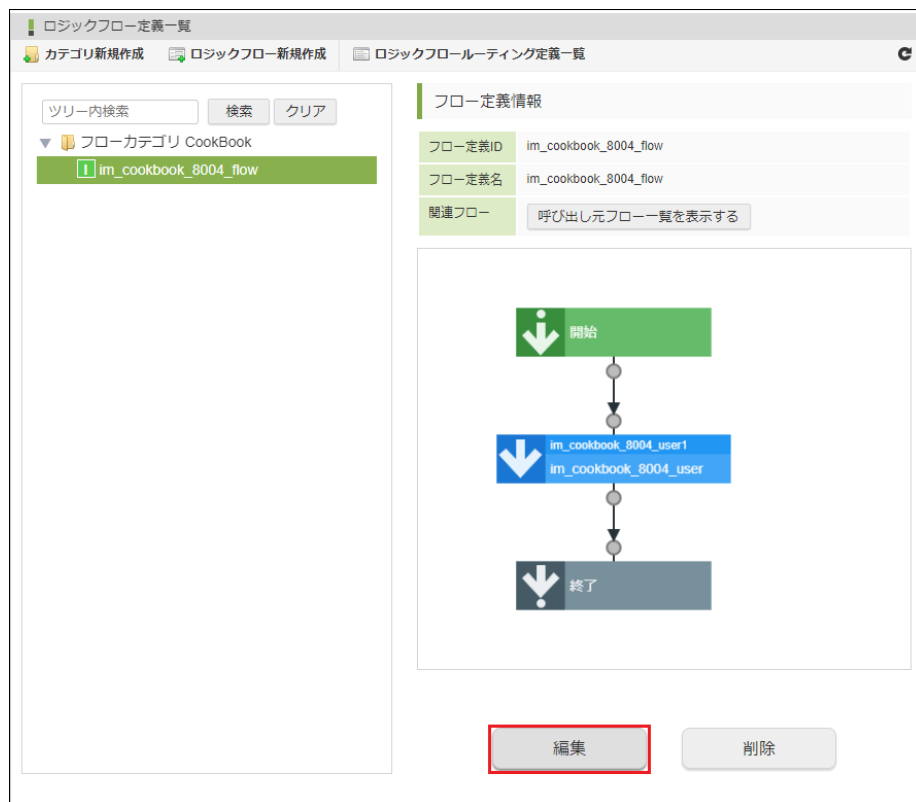
- サンプルのフロー定義のデバッグ画面を開く
- 入力値を設定し、デバッグを実行する
- 実行結果を確認する

サンプルのフロー定義のデバッグ画面を開く

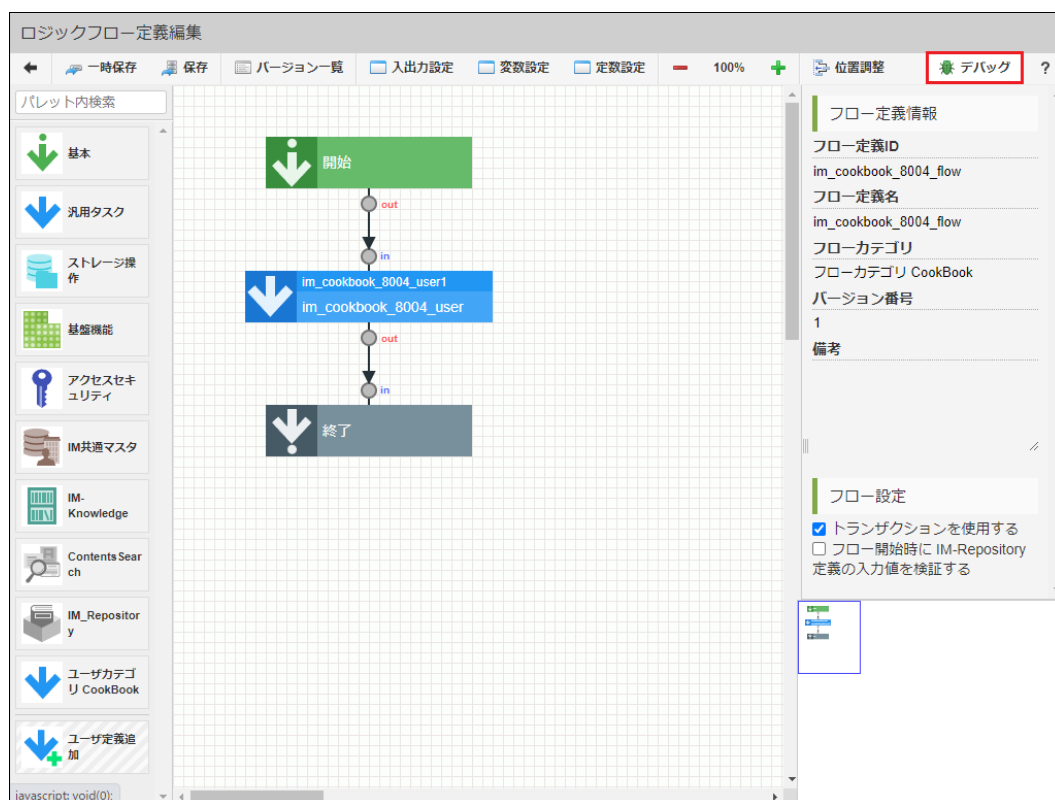
1. 「サイトマップ」 - 「LogicDesigner」 - 「フロー定義一覧」をクリックします。



2. 「フローカテゴリ CookBook」 - 「im_cookbook_8004_flow」を選択し、「編集」ボタンをクリックします。

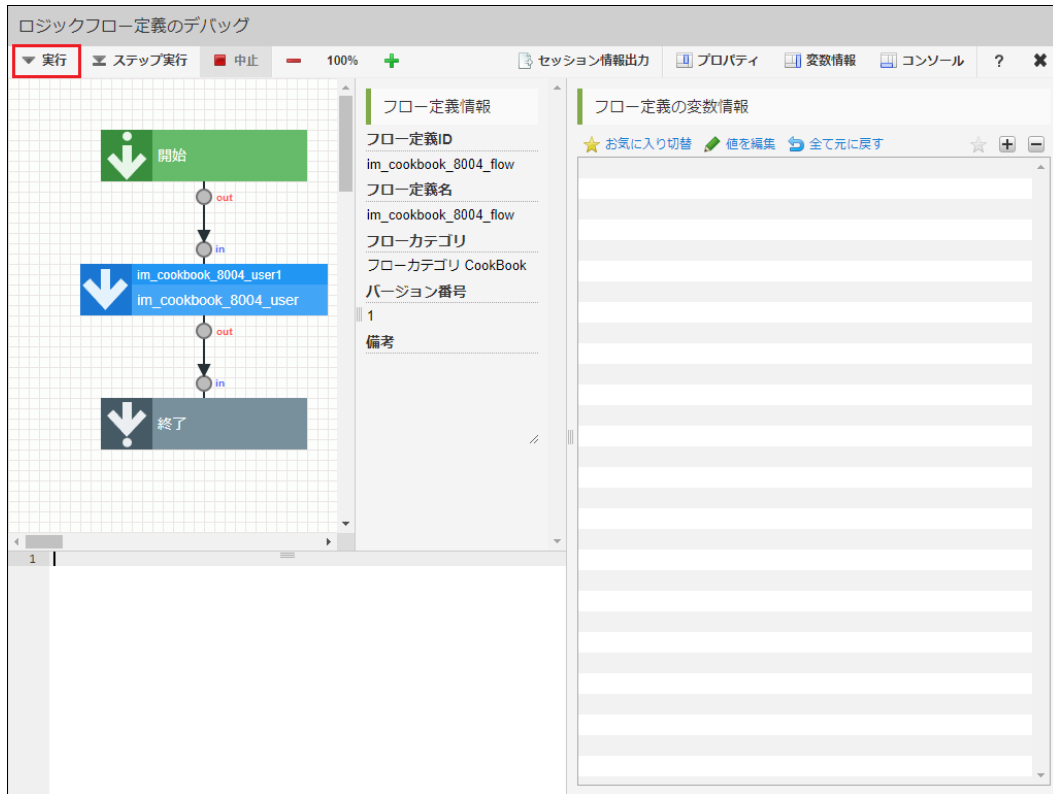


3. 「デバッグ」ボタンをクリックします。



入力値を設定し、デバッグを実行する

1. 「実行」ボタンをクリックします。



2. 「inputFilePath」に変換対象ファイル、「outputFilePath」に出力ファイルのパブリックストレージパスを設定し、「実行」ボタンをクリックします。

設定例は次の通りです。

< 値を編集 >

変数	値
inputFilePath	im_cookbook_8004/pdfa_logic/data/sample.docx
outputFilePath	im_cookbook_8004/pdfa_logic/data/out.pdf

< JSON入力 >

```
{
  "inputFilePath": "im_cookbook_8004/pdfa_logic/data/sample.docx",
  "outputFilePath": "im_cookbook_8004/pdfa_logic/data/out.pdf"
}
```



コラム

ユーザ定義の入出力値については「[ユーザ定義タスク](#)」を参照してください。



コラム

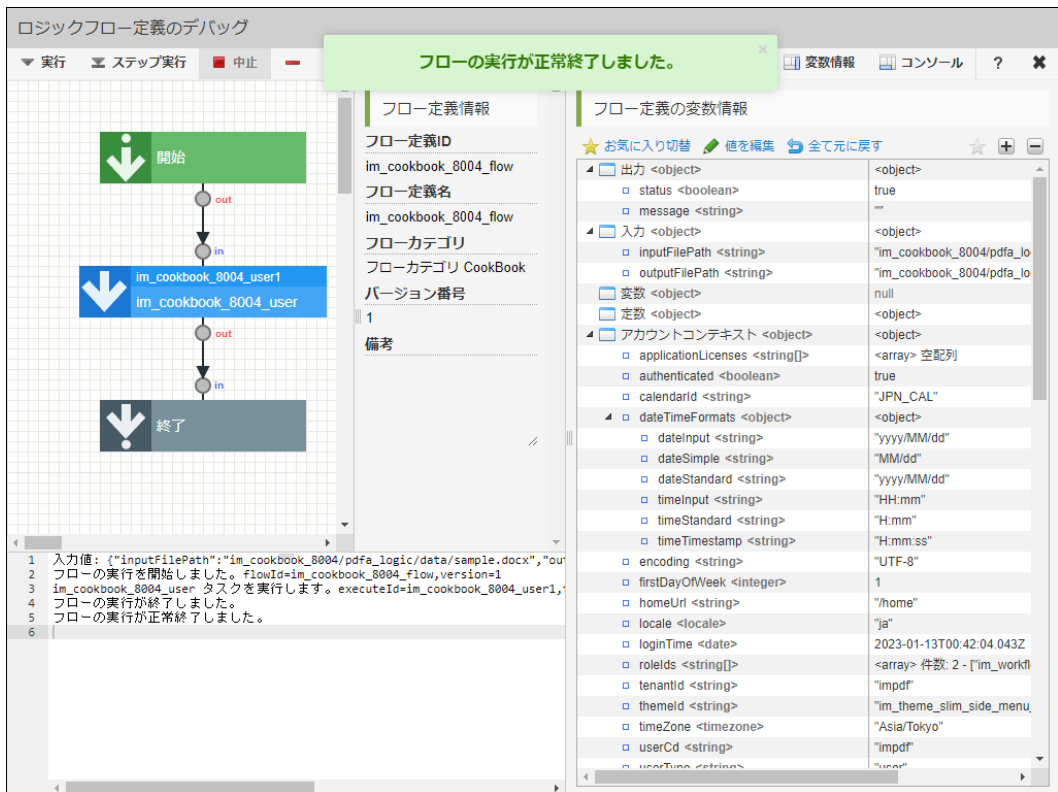
設定例の「inputFilePath」には、パブリックストレージ配下に設置されたPDF変換サンプル用のdocxファイルを指定しています。

3. 「決定」ボタンをクリックします。



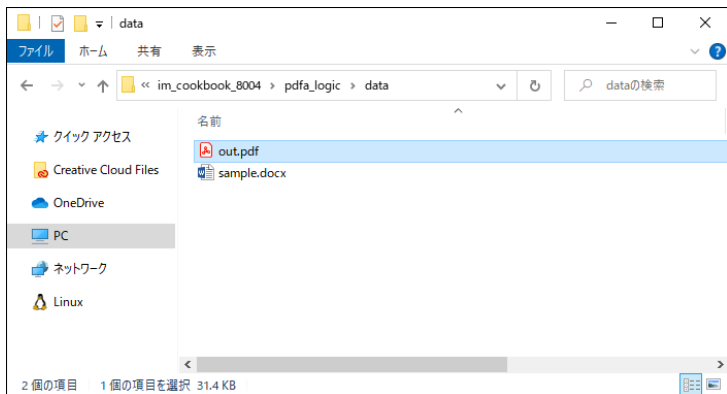
4. デバッグが開始されます。

正常にデバッグが終了した場合、その旨のメッセージが表示され、変数情報ペイン、および、コンソールペインが更新されます



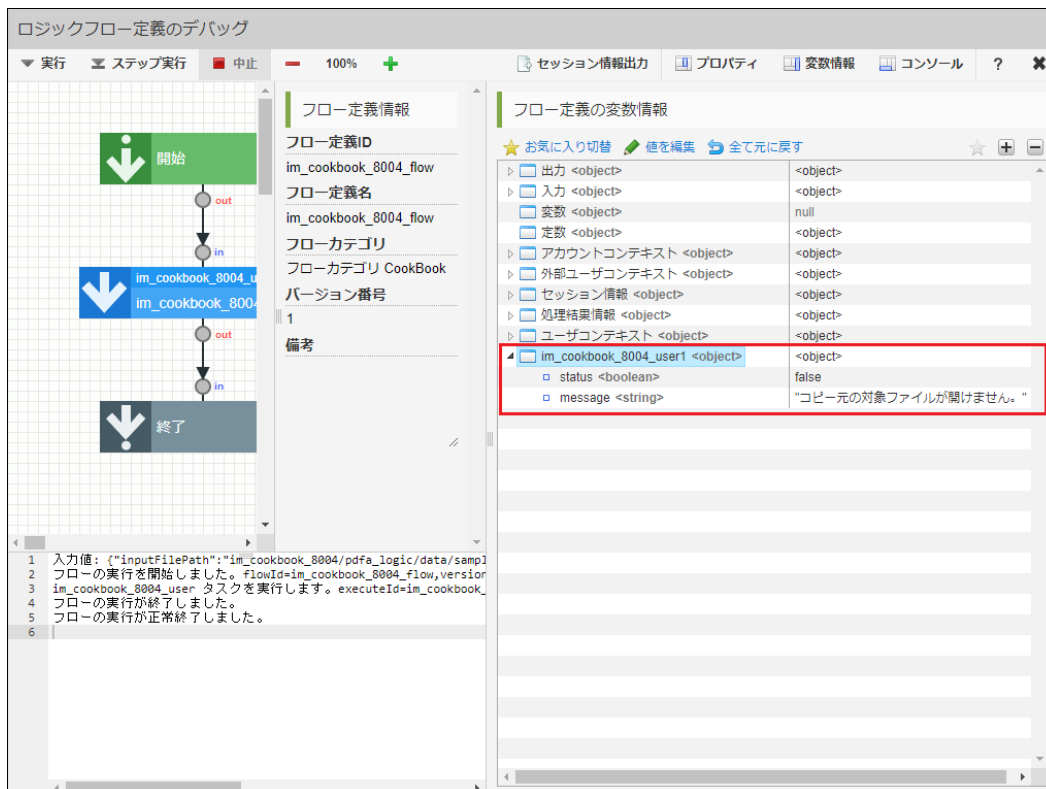
実行結果を確認する

1. 「outputFilePath」に指定した出力先に、PDFファイルが出力されていることを確認します。





PDFの変換処理に失敗した場合は、デバッグ実行時のユーザ定義の返却値「status」、および、「message」を確認してください。



以上で、全ての手順は終了です。

ユーザ定義タスク

PDF変換

編集	ユーザ定義ID	ユーザ定義名	種別	ユーザカテゴリ	呼出元
	im_cookbook_8004_user	im_cookbook_8004_user	javascript	ユーザカテゴリ CookBook	

入力値

```
im_cookbook_8004_user <object>
├─ inputFilePath <string>
└─ outputFilePath <string>
```

項目名	必須/任意	型	配列/リスト	説明
inputFilePath	必須	string	なし	PDF変換対象ファイルのパブリックストレージパス
outputFilePath	必須	string	なし	PDF変換出力ファイルのパブリックストレージパス

出力値

```
im_cookbook_8004_user <object>
├─ status <boolean>
└─ message <string>
```

項目名	型	配列/リスト	説明
status	boolean	なし	true : PDF変換成功時 false : PDF変換失敗時
message	string	なし	PDF変換成功時 : 空文字 PDF変換失敗時 : エラーメッセージ

スクリプト

サンプル内で使用しているAPIについては「[API](#)」を参照してください。



コラム

文書情報を設定する場合は、スクリプトの27、32行目のコメントを外してください。



コラム

セキュリティ情報を設定する場合は、スクリプトの35、41行目のコメントを外してください。


```

1  /**
2   * run.
3   *
4   * @param input {Object} - task input data.
5   * @return {Object} task result.
6   */
7  function run(input) {
8
9      const tempFiles = new PdftempFiles();
10
11     try {
12         if (!input.inputFilePath) {
13             throw new Error("PDF変換対象ファイルパスにnull、または、空文字が指定されています。");
14         }
15
16         if (!input.outputFilePath) {
17             throw new Error("PDF変換出力ファイルパスにnull、または、空文字が指定されています。");
18         }
19
20         if (input.inputFilePath.indexOf('.') !== -1) {
21             const ex = new IMPDFAutoConverter();
22             const srcFileExt = "." + input.inputFilePath.split('.').pop();
23             const tempSrcFile = tempFiles.copyFrom(input.inputFilePath, srcFileExt);
24             const tempOutFile = tempFiles.create();
25
26             /* 文書情報を設定 */
27             /* ex.setDocInf(
28              "タイトル",
29              "サブタイトル",
30              "作成者",
31              "アプリケーション",
32              "キーワード"); */
33
34             /* 128ビットセキュリティ情報を設定 */
35             /* ex.setSecurity128(
36              "open",
37              "sec",
38              "PRINT_DISABLE",
39              "ACC_DISABLE",
40              "COPY_DISABLE",
41              "DOCCHANGE_DISABLE"); */
42
43             /* Web用に最適化の有無を設定 */
44             ex.setFastWebView(true);
45
46             /* プリンタ名を設定 */
47             ex.setPrinter("YSS PDF Converter XP");
48
49             /* 変換前のタイムアウト秒数を設定 */
50             ex.setBeforeTimeoutSec(0);
51
52             /* 変換後のタイムアウト秒数を設定 */
53             ex.setTimeoutSec(1500);
54
55             /* PDF変換サーバへのファイル転送のタイムアウトミリ秒を設定 */
56             ex.setTransTimeoutMilliSec(1560000);
57
58             /* PDF変換 */
59             ex.convert(tempSrcFile.path(), tempOutFile.path());
60
61             tempFiles.copyTo(tempOutFile, input.outputFilePath);
62         } else {
63             throw new Error("PDF変換対象ファイルの拡張子がありません。");
64         }
65     } catch (error) {
66         return {
67             status: false,
68             message: error.message
69         };
70     } finally {
71         tempFiles.close();
72     }
73
74     return {

```

```

75     status: true,
76     message: ""
77 };
78 }

```

API

サンプル内で使用しているAPIについて示します。

IMPDFAutoConverter

IMPDFAutoConverterクラスの詳細については、「[IM-PDFAutoConverter for Accel Platform API ドキュメント](#)」- スクリプト開発モデル「[IMPDFAutoConverter](#)」を参照してください。



注意

IMPDFAutoConverter.setPrinter(name)の引数の値は、PDFコンバータのプリンタ名に合わせて変更してください。
デフォルトのプリンタ名は「YSS PDF Converter XP」です。

PdfaTempFiles

サンプル用の一時ファイル进行操作するクラスです。



注意

本クラスは公開されているAPIではなく、サンプル用に用意したロジックになります。
そのため、サンプル用のimmファイルを適用していない場合、本クラスは使用できません。

Constructor

new PdfaTempFiles()

インスタンスオブジェクトを作成します。

Returns

生成されたインスタンスオブジェクト

Methods

create(ext)

空のファイルを作成します。

Parameters

Name	Type	Description
ext	String	ファイル名を生成するために使用される接尾辞文字列 null、または、未指定時は".tmp" が使用されます。

Returns

新規作成された空のファイルを示すFileオブジェクト

copyFrom(srcFilePath, ext)

指定したファイルをコピーし、コピー先のファイルを返します。

Parameters

Name	Type	Description
srcFilePath	String	コピー元の対象ファイルのパブリックストレージパス

Name	Type	Description
ext	String	コピー先のファイル名を生成するために使用される接尾辞文字列 null、または、未指定時は".tmp" が使用されます。

Returns

コピー先ファイルを示すFileオブジェクト

copyTo(tempFile, destFilePath)

指定したファイルを、指定先へコピーします。

Parameters

Name	Type	Description
tempFile	File	コピー元の対象ファイル
destFilePath	String	コピー先の対象ファイルのパブリックストレージパス

close()

一時ファイルを削除し、一時ファイルに関する操作を終了します。