

- 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. APIの種類と性質
 - 4.2. プログラム開発における注意点
 - 4.3. 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. Office系→PDF変換
 - 5.2. HTML→PDF変換
- 6. ステータスコード表
 - 6.1. Office系→PDF変換
 - 6.2. HTML→PDF変換
- 7. サポート
- 8. 付録
 - 8.1. 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
 - 8.2. IotheCommonTempFiles

| 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。 <ul style="list-style-type: none">■ 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。 <ul style="list-style-type: none">■ 「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。 <ul style="list-style-type: none">■ 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。 <ul style="list-style-type: none">■ 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。 <ul style="list-style-type: none">■ 「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。 <ul style="list-style-type: none">■ 「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。 <ul style="list-style-type: none">■ 「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。 <ul style="list-style-type: none">■ 「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。 <ul style="list-style-type: none">■ 「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。 <ul style="list-style-type: none">■ 「チュートリアル」の構成、および、記述を変更<ul style="list-style-type: none">■ 「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

2024-04-01

第15版 下記を追加・変更しました。

- 「[APIリストについて](#)」にAPIに関するコラムを追加
- 「[IotheCommonTempFiles](#)」を追加

2024-10-01

第16版 下記を追加・変更しました。

- 「[プログラミング](#)」
 - 「[動作概念](#)」を削除
 - 「[タイムアウトについて](#)」の見出しを「[大量のPDF変換を指示した場合、タイムアウトが発生する可能性があります](#)」に変更し、「[プログラム開発における注意点](#)」に移動、および、記述を変更
 - 「[プログラム開発における注意点](#)」の構成を変更
 - 「[HTML→PDF変換の実行方法には制限があります](#)」を追加
 - 「[PDF変換を行うまでの待機時間が短い場合、正常に変換されない可能性があります](#)」を追加
 - 「[IM-Workflowのコンテンツ画面をPDF変換する場合、遷移可能なURLを指定してください](#)」を追加
- 「[チュートリアル](#)」
 - 「[Office系→PDF変換](#)」を追加
 - 「[HTML→PDF変換](#)」を追加
 - 「[チュートリアル](#)」の記述を、「[Office系→PDF変換](#)」に移動し、構成、および、記述を変更
 - 「[JSPファイルの作成](#)」のプログラムを変更
- 「[ステータスコード表](#)」
 - 「[Office系→PDF変換](#)」を追加
 - 「[HTML→PDF変換](#)」を追加
 - 「[ステータスコード表](#)」の記述を、「[Office系→PDF変換](#)」に移動
- 「[IotheCommonTempFiles](#)」
 - 「[copyFrom\(srcFilePath, ext\)](#)」のExceptionの記述を変更
 - 「[copyTo\(tempFile, destFilePath\)](#)」のExceptionの記述を変更
 - 「[close\(\)](#)」にExceptionの記述を追加

変更年月日

変更内容

2025-04-01

第17版 下記を追加・変更しました。

- 「[プログラミング](#)」
- 「[HTML→PDF変換 の実行方法には制限があります](#)」の制限に関する記述を変更
- 「[チュートリアル](#)」
- 「[HTML→PDF変換](#)」
 - 「[前提条件](#)」に非同期処理に関する記述を追加
 - 「[同期処理](#)」を追加
 - 「HTML→PDF変換（JavaEE開発モデル）」の見出しを「[スクリプト開発モデル](#)」に変更し「[同期処理](#)」に移動、および、「[非同期処理](#)」にあわせて記述を見直し
 - 「HTML→PDF変換（スクリプト開発モデル）」の見出しを「[JavaEE開発モデル](#)」に変更し「[同期処理](#)」に移動、および、「[非同期処理](#)」にあわせて記述を見直し
 - 「[非同期処理](#)」を追加

2025-10-01

第18版 下記を追加・変更しました。

- 「[大量のPDF変換を指示した場合、タイムアウトが発生する可能性があります](#)」のHTML→PDF変換のAPIについて記述を変更、および、注意を追加
 - 「[PDF変換を行うまでの待機時間が短い場合、正常に変換されない可能性があります](#)」のデフォルト値の記述を変更
-

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」



コラム

IM-PDFAutoConverter for Accel Platform は、一時ファイルを操作するスクリプト開発モデル用のAPIも用意しています。

IM-PDFAutoConverter for Accel Platform でPDF変換する際、ファイルの絶対パスが必要となります。IM-LogicDesigner上ではJavaScript定義時の制限により、絶対パスを取得するAPIが一部利用できず、PDF変換できない場合があります。

そのような場合は、一時ファイルを操作するAPIを使用し、対象ファイルを一時ファイルにコピー後、一時ファイルの絶対パスを指定することで処理が可能となります。

上記APIの詳細については、「[IotheCommonTempFiles](#)」を参照してください。

IM-LogicDesigner上でOffice系→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。

JavaScript定義時の制限については、「[IM-LogicDesignerチュートリアルガイド](#)」-「[参考：ユーザ定義（JavaScript）における制限](#)」を参照してください。

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 | <p>パッケージ クラス 使用 階層ツリー 非推奨 API 索引 ヘルプ</p> <p>前のクラス 次のクラス 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</p> <p>yss.autoconverterex.soap クラス AutoExRemote</p> <p>jaya.lang.Object └ yss.autoconverterex.soap.AutoExRemote</p> <p>public final class AutoExRemote extends java.lang.Object</p> <p>PDF変換サーバに変換指示を出すクラス。</p> <p>フィールドの概要</p> <table border="1"><tr><td>static int</td><td>SEC128ACC_DISABLE 128ビットセキュリティ(アクセス):許可しない</td></tr><tr><td>static int</td><td>SEC128ACC_ENABLE 128ビットセキュリティ(アクセス):許可する</td></tr><tr><td>static int</td><td>SEC128COPY_DISABLE 128ビットセキュリティ(コピー):許可しない</td></tr><tr><td>static int</td><td>SEC128COPY_ENABLE 128ビットセキュリティ(コピー):許可する</td></tr><tr><td>static int</td><td>SEC128DOCCCHANGE_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</td></tr><tr><td>static int</td><td>SEC128DOCCCHANGE_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</td></tr><tr><td>static int</td><td>SEC128DOCCCHANGE_DISABLE 128ビットセキュリティ(文書変更):許可しない</td></tr><tr><td>static int</td><td>SEC128DOCCCHANGE_ENABLE 128ビットセキュリティ(文書変更):許可する</td></tr><tr><td>static int</td><td>SEC128DOCCCHANGE_FORMAT1</td></tr></table> | static int | SEC128ACC_DISABLE 128ビットセキュリティ(アクセス):許可しない | static int | SEC128ACC_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 | SEC128DOCCCHANGE_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 | SEC128DOCCCHANGE_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 | SEC128DOCCCHANGE_DISABLE 128ビットセキュリティ(文書変更):許可しない | static int | SEC128DOCCCHANGE_ENABLE 128ビットセキュリティ(文書変更):許可する | static int | SEC128DOCCCHANGE_FORMAT1 |
| static int | SEC128ACC_DISABLE 128ビットセキュリティ(アクセス):許可しない | | | | | | | | | | | | | | | | | | |
| static int | SEC128ACC_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 | SEC128DOCCCHANGE_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 | SEC128DOCCCHANGE_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 | SEC128DOCCCHANGE_DISABLE 128ビットセキュリティ(文書変更):許可しない | | | | | | | | | | | | | | | | | | |
| static int | SEC128DOCCCHANGE_ENABLE 128ビットセキュリティ(文書変更):許可する | | | | | | | | | | | | | | | | | | |
| static int | SEC128DOCCCHANGE_FORMAT1 | | | | | | | | | | | | | | | | | | |

スクリプト開発モデル

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

- APIの種類と性質
- プログラム開発における注意点
 - ファイルパスの指定方法について注意してください
 - ファイルサイズの大きいPDFファイルは、出力処理に時間がかかる場合があります
 - HTML→PDF変換 の実行方法には制限があります
 - 大量のPDF変換を指示した場合、タイムアウトが発生する可能性があります
 - PDF変換を行うまでの待機時間が短い場合、正常に変換されない可能性があります
 - IM-Workflow のコンテンツ画面をPDF変換する場合、遷移可能なURLを指定してください
 - 体験版ライセンスにおける注意点

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

ファイルパスの指定方法について注意してください

変換対象ファイルや、出力先ファイルのパスを指定する際、 intra-mart Accel Platform からアクセス可能な絶対パスを指定してください。

ファイルサイズの大きいPDFファイルは、出力処理に時間がかかる場合があります

ファイルサイズの大きいPDFファイルを作成する際、出力処理に時間がかかり、APIのレスポンスと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。

サイズの大きいPDFファイルを作成する場合は、出力処理の時間を考慮し、十分な時間が経過した後に、出力されたPDFファイルにアクセスするようにしてください。

HTML→PDF変換 の実行方法には制限があります

HTML→PDF変換 の実行方法には、制限があります。

詳細については、「 [IM-PDFAutoConverter for Accel Platform リリースノート](#) 」 - 「 [HTML→PDF変換 の実行方法には制限があります](#) 」を参照してください。

大量のPDF変換を指示した場合、タイムアウトが発生する可能性があります

IM-PDFAutoConverter for Accel Platform 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、次のように処理される仕組みです。

- Office系→PDF変換 : 1件ずつ処理される
- HTML→PDF変換 : 設定ファイルで設定された起動数ずつ処理される

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。

IM-PDFAutoConverter for Accel Platform のAPIは、次のタイムアウトを利用します。

- Office系→PDF変換

| JavaEE開発モデル | スクリプト開発モデル | 説明 |
|-------------------------------------|---|---|
| setBeforeTimeoutSec(int timeoutSec) | setBeforeTimeoutSec(Number 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。タイムアウト時間は過ぎると、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。 |
| setTimeoutSec(int timeoutSec) | setTimeoutSec(Number 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。 |
| setTransTimeoutSec(int timeoutSec) | setTransTimeoutMilliSec(Number timeoutMilliSec) | 【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ないため、設定する際は、setTimeoutSec(timeoutSec)の10倍以上など、できるだけ長く設定することを推奨します。通常は、0（タイムアウトしない）を指定してください。 |

- HTML→PDF変換

| JavaEE開発モデル | スクリプト開発モデル | 説明 |
|-------------------------------|----------------------------------|--|
| setTimeoutSec(int timeoutSec) | setTimeoutSec(Number timeoutSec) | PDFファイルへの変換時、待機する時間を設定します。未設定の場合は設定ファイルから値を取得します。未設定かつ、設定ファイルの値が存在しない場合は3600秒(1時間)待機します。 |



注意

setTimeoutSec(timeoutSec)にて設定した時間を超過した場合、タイムアウトエラーとして例外をスローしますが、既に実行済みの変換処理は継続します。

そのため、上記APIで設定する時間は、API呼び出し側が待機する時間として扱ってください。



コラム

PDF変換処理のタイムアウトを変更した場合は、intra-mart Accel Platform のセッションタイムアウトについても確認してください。

PDF変換処理が完了する前に intra-mart Accel Platform 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 Accel Platform のセッションタイムアウトの設定については、intra-mart Accel Platform のドキュメントを参照してください。



コラム

APIの詳細については、「IM-PDFAutoConverter for Accel Platform API ドキュメント」を参照してください。

PDF変換を行うまでの待機時間が短い場合、正常に変換されない可能性があります

HTML→PDF変換において、画面表示後、PDF変換を行うまでの待機時間が短い場合、正常に変換されない可能性があります。

次のAPIを利用し、適切な待機時間を設定することを推奨します。

| JavaEE開発モデル | スクリプト開発モデル | 説明 |
|---|--|---|
| setWaitMillisecond(int waitMillisecond) | setWaitMillisecond(Number waitMillisecond) | 画面表示後、PDF変換を行うまでの待機時間をミリ秒単位で設定します。未設定の場合は設定ファイルから値を取得します。未設定かつ、設定ファイルの値が存在しない場合は5000ミリ秒(5秒)待機します。 |

IM-Workflow のコンテンツ画面をPDF変換する場合、遷移可能なURLを指定してください

IM-Workflow のコンテンツ画面のURLについては、「[IM-Workflowの色々なコンテンツ画面へ遷移するURL](#)」を参照してください。

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。

Office系→PDF変換

概要

本チュートリアルでは、プログラミング言語を使用した Microsoft Office 系ファイルからPDFファイルへの基本的な変換方法を説明します。

前提条件

本チュートリアルを進めるにあたり、次の事前準備が行われていることが前提となります。

- IM-PDFAutoConverter for Accel Platform (Office系→PDF変換) のセットアップが完了していること。
- intra-mart Accel Platform のテナント環境セットアップが完了していること。

実践

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/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 timeoutsec ;
13
14 /* インスタンスを作成*/
15 ex = new AutoExRemote();
16
17 ****文書情報、セキュリティ、スタンプの設定
18 ****
19 /* 文書情報を設定*/
20 ex.setDocInf("タイトル", "サブタイトル", "作成者", "アプリケーション", "キーワード");
21
22 /* 文書情報を名前で指定して設定*/
23 /* ex.setDocInfByName("docinf-name"); */
24
25 /* セキュリティ設定を名前で指定して設定*/
26 /* ex.setSecurityByName("security-name"); */
27
28 /* 文書情報及びセキュリティを名前で指定して設定*/
29 /* ex.setDocInfAndSecurity("docinf_security-name"); */
30
31 /* 40ビットセキュリティの指定*/
32 /* ex.setSecurity40("open", "security", true, true, true, true); */
33
34 /* 128ビットセキュリティの指定*/
35 /* ex.setSecurity128("open", "security", AutoExRemote.SEC128PRINT_DISABLE,
36 AutoExRemote.SEC128COPY_DISABLE, AutoExRemote.SEC128DOCCHANGE_DISABLE); */
37
38 /* フォルダ名及びフォルダ別設定の情報を設定*/
39 /* ex.setFolderName("folder1", false); */
40
41 /* スタンプ(名前)の指定*/
42 /* ex.addStamp("stamp1"); */
43
44 /* Web用に最適化の有無*/
45 ex.setFastWebView(true);
46
47 /* プリンタ名の指定*/
48 ex.setPrinter("YSS PDF Converter XP");
49
50 ****
51 /* 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
52 ****/
53 timeoutsec = 60 * 60;
54 timeoutsec = AutoExRemote.TIMEOUT_INFINITE;
55 ex.setBeforeTimeoutSec(timeoutsec);
56
57 ****
58 /* 変換時間に対するタイムアウト(秒)の設定
59 ****/
60 timeoutsec = 60 * 60;
61 timeoutsec = AutoExRemote.TIMEOUT_INFINITE;
62 ex.setTimeoutSec(timeoutsec);
63
64 ****
65 /* 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
66 ****/
67 timeoutsec = 60 * 60 * 1000;
68 timeoutsec = AutoExRemote.TIMEOUT_INFINITE;
69 ex.setTransTimeoutSec(timeoutsec);
70
71 try {
72     ...
73 }
74
75 
```

```

73  /* PDF変換 */
74  outpdf = ex.convert(src, pdf);
75 }
76 catch(Exception e) {
77     e.printStackTrace();
78     message = e.getMessage();
79 }
80
81 %>
82 <imui:head>
83 <title>IM-PDFAutoConverter-チュートリアル-JavaEE開発モデル-convert</title>
84 </imui:head>
85
86 <div class="imui-title">
87     <h1>IM-PDFAutoConverter チュートリアル JavaEE開発モデル convert</h1>
88 </div>
89
90 <div class="imui-form-container">
91     <div class="imui-chapter-title"><h2>実行結果</h2></div>
92     <table class="imui-table">
93         <tbody>
94             <tr>
95                 <th class="wd-20">出力PDFファイル</th>
96                 <td><%= outpdf %></td>
97             </tr>
98             <tr>
99                 <th class="wd-225px">メッセージ</th>
100                <td><%= message %></td>
101            </tr>
102        </tbody>
103    </table>
104 </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。

プログラムの登録

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 * | <table border="1"> <tr><td>日本語 *</td><td>IM-PDFAutoConverter</td></tr> <tr><td>英語</td><td>IM-PDFAutoConverter</td></tr> <tr><td>中国語</td><td>IM-PDFAutoConverter</td></tr> </table> | 日本語 * | IM-PDFAutoConverter | 英語 | IM-PDFAutoConverter | 中国語 | IM-PDFAutoConverter |
| 日本語 * | IM-PDFAutoConverter | | | | | | |
| 英語 | IM-PDFAutoConverter | | | | | | |
| 中国語 | IM-PDFAutoConverter | | | | | | |
| アイコン・画像 | <input checked="" type="radio"/> 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。 <input type="radio"/> CSS Sprites imui://csssprites/ クラス名を入力してください。 | | | | | | |

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 * | <table border="1"> <tr><td>日本語 *</td><td>convert</td></tr> <tr><td>英語</td><td>convert</td></tr> <tr><td>中国語</td><td>convert</td></tr> </table> | 日本語 * | convert | 英語 | convert | 中国語 | convert |
| 日本語 * | convert | | | | | | |
| 英語 | convert | | | | | | |
| 中国語 | convert | | | | | | |
| URL * | convert.jsp <input type="button" value="権限設定"/> | | | | | | |
| 呼び出し方法 | GET <input type="button" value=""/> | | | | | | |
| 引数 | <input type="button" value="+ 行追加"/> <input type="button" value="— 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> </tbody> </table> | キー | 値 | | | | |
| キー | 値 | | | | | | |
| | | | | | | | |
| アイコン・画像 | <input checked="" type="radio"/> 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。 <input type="radio"/> CSS Sprites imui://csssprites/ クラス名を入力してください。 | | | | | | |
| IFRAME表示 | <input type="checkbox"/> | | | | | | |
| ポップアップ表示 | <input type="checkbox"/> | | | | | | |
| 説明 | <input type="text"/> | | | | | | |

5. メニュー設定は完了です。

メニュー設定

エクスポート

グローバルナビ(PC用)

検索キーワードを入力してください。 検索 クリア

- + Top
 - IMBox
 - Portal
 - IM-PDFAutoConverter
 - convert
- Workflow
- テナント管理
- サンプル

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。

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 inFile = 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 = inFile.substring(inFile.lastIndexOf(".") - inFile.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 = execPdftautoconverter(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.exist()) {
30      pdfFileStream = pdfFile.load();
31    }
32  }
33  catch(e) {
34    errorMessage = e.message + (isUndefined(e.stack) ? "" : '\r\n' + e.stack);
35  }
36
37  // 保存したファイルを削除します。
38  inFile.remove();
39  outFile.remove();
40
41  // 生成したPDFファイルをダウンロードします。
42  if(pdfFileStream != null) {
43    let pdfFileName = inFile.substring(0, inFile.lastIndexOf(".")) + ".pdf";
44    Module.download.send(pdfFileStream, pdfFileName);
45  }
46  else {
47    let logger = Logger.getLogger();
48    logger.error(errorMessage);
49    let response = Web.getHTTPResponse();
50    response.sendError(500, "Failed to convert PDF file.");
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 execPdftautoconverter(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 pdftautoconverter = new IMPDFAutoConverter();
65
66  // PDF出力時に設定する文書情報を指定します。
67  // pdftautoconverter.setDocInfo(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出力時に設定する文書情報を名前で指定します。
76 // 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。
77 // 本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 // 本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 // 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。
85 // 本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 // 本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 // 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。
93 // 本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 // 本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 // 本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 // 本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 // 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。
103 // 本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 // 本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 // 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。
116 // 本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 // 本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 // 本メソッドを指定した場合、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 // 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。
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 // 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。
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 // 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。
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 // 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。
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 // 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。
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-pdfa.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。

The screenshot shows the 'Permission Setting' screen for the '画面・処理' (Screens and Processes) category. The top navigation bar includes '画面・処理' (Screens and Processes), 'アクションの種類' (Action Types), and '全てのアクション' (All Actions). A red box highlights the '権限設定を開始する' (Start Permission Setting) button in the top right corner of the header. The main table lists resources and their actions, with checkboxes indicating permission levels for various roles.

| 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 | メニュー運用管理者 | アカウント管理者 | ロール管理者 | カレンダーマネージャー | システム管理者 | |
| 画面・処理 | 実行 > | <input checked="" type="checkbox"/> | | |
| intra-mart Accel Platform | 実行 > | <input checked="" type="checkbox"/> | | |
| welcome-all マッパー | 実行 > | <input checked="" type="checkbox"/> | | |
| PDF変換サンプル | 実行 > | <input checked="" type="checkbox"/> | | |
| IM-BIS | 実行 > | <input checked="" type="checkbox"/> | | |
| BIS監査 | 実行 > | <input checked="" type="checkbox"/> | | |

4. [リソース]を選択し、[リソースの詳細を開く]押下します。

| 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 | | |
|---------------------------|-------|--------|----------|--------|----------|---------|-------|
| | | 全て許可 | 全て禁止 | 全て未設定 | 条件の新規作成 | | 条件の編集 |
| 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 |
| 画面・処理 | | | | | | | |
| 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 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| BIS業務管理 | 実行 | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ |
| BIS共通 | 実行 | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ |
| BIS管理共通 | 実行 | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ |

5. [配下にリソースを新規作成]を押下します。

リソースの詳細

| リソースグループID | http-services | リソースの編集 | リソースの削除 | リソースの閉塞 |
|------------|--|---------|---------|---------|
| リソースグループ名 | 画面・処理 | | | |
| リソースURI | | | | |
| 説明 | 各種業務画面へのアクセスや、画面上で行う各種処理の権限に関するリソースグループです。 | | | |
| リソースの階層 | リソースグループ名 画面・処理 | 開く | | |

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 | | |
| <small>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。</small> | | |
| 説明 | 日本語 | |
| | 英語 | |
| | 中国語 (中国) | |

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 | | |
| <small>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。</small> | | |

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 | |
| <small>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。</small> | | |

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 マッパー | 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 * | <table border="1"> <tr> <td>日本語 *</td> <td>IM-PDFAutoConverter</td> </tr> <tr> <td>英語</td> <td>IM-PDFAutoConverter</td> </tr> <tr> <td>中国語</td> <td>IM-PDFAutoConverter</td> </tr> </table> | 日本語 * | IM-PDFAutoConverter | 英語 | IM-PDFAutoConverter | 中国語 | IM-PDFAutoConverter |
| 日本語 * | IM-PDFAutoConverter | | | | | | |
| 英語 | IM-PDFAutoConverter | | | | | | |
| 中国語 | IM-PDFAutoConverter | | | | | | |
| アイコン 画像 | <input checked="" type="radio"/> ファイルパス <input type="text" value="コンテキストパス配下のURLを入力してください。"/> <input type="radio"/> CSS Sprites <input type="text" value="imui://csssprites/ クラス名を入力してください。"/> | | | | | | |
| <input type="button" value="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 `pd़fa/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 * | 5i7j7cuoodion1 | | | | | |
|--------------|--|---------|----|---|--|--|
| メニューアイテム名 * | 日本語 * | convert | | | | |
| | 英語 | convert | | | | |
| | 中国語 | convert | | | | |
| URL * | pdaf/test | | | | | |
| 呼び出し方法 | GET | | | | | |
| 引数 | <input type="button" value="行追加"/> <input type="button" value="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> </tbody> </table> | | キー | 値 | | |
| キー | 値 | | | | | |
| | | | | | | |
| アイコン画像 | <input checked="" type="radio"/> ファイルパス <input type="text" value="コンテキストパス配下のURLを入力してください。"/> <input type="radio"/> CSS Sprites <input type="text" value="imui://csssprites/"/> クラス名を入力してください。 | | | | | |
| IFRAME表示 | <input type="checkbox"/> | | | | | |
| ポップアップ表示 | <input type="checkbox"/> | | | | | |
| 説明 | <input type="text"/> | | | | | |

5. メニュー設定は完了です。

Intra-mart®

Top Workflow テナント管理 サンプル サイトマップ

メニュー設定

エクスポート

グローバルナビ(PC用)

検索キーワードを入力してください。 検索 クリア

+ Top

- IMBox
- Portal
- IM-PDFAutoConverter
 - convert

Workflow

テナント管理

サンプル

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。



コラム

IM-LogicDesigner上で、IM-PDFAutoConverter for Accel PlatformのAPIを使用し、Office系→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。

HTML→PDF変換

本チュートリアルでは、プログラミング言語を使用した IM-Workflow の画面（HTMLファイル）からPDFファイルへの基本的な変換方法を説明します。

前提条件

本チュートリアルを進めるにあたり、次の事前準備が行われていることが前提となります。

- 次のバージョンの IM-PDFAutoConverter for Accel Platform （HTML→PDF変換）のセットアップが完了していること。
 - 同期処理の場合：2024 Autumn 以降
 - 非同期処理の場合：2025 Spring 以降
- IM-FormaDesigner for Accel Platform のセットアップが完了していること。
- IM-Workflow のセットアップが完了していること。
- intra-mart Accel Platform のテナント環境セットアップが完了していること。
- intra-mart Accel Platform のサンプルデータの投入が完了していること。

実践

同期処理

JavaEE開発モデル

本チュートリアルでは、JavaEE開発モデル用APIを使用したIM-Workflowのユーザプログラムを作成し、そのプログラムを実行することで、IM-Workflowの画面（HTMLファイル）をPDFファイルに変換します。

処理のタイミングは、IM-Workflowの承認ノードのアクション処理となります。

チュートリアルを実施するにあたり、次のzipファイルをダウンロードし、解凍してください。

<[htmltopdf_sync_javaTutorial.zip](#)>

解凍したファイルの構成は、次の通りです。

| フォルダ名／ファイル名 | 説明 |
|----------------------------------|---|
| import/ | HTML→PDF変換インポート関連フォルダ |
| htmltopdf_sync_java_forma.zip | IM-FormaDesigner for Accel Platformで作成したアプリケーションのアプリケーション情報ファイル |
| htmltopdf_sync_java_workflow.xml | IM-Workflowのコンテンツ定義、ルート定義、および、フロー定義の定義情報ファイル |
| javaee/ | JavaEE開発モデル用フォルダ |
| HmlToPdfProcess.java | JavaEE開発モデル用プログラム |

<import/htmltopdf_sync_java_forma.zip>を、IM-FormaDesigner for Accel Platformのアプリケーション情報インポート画面からインポートしてください。

<import/htmltopdf_sync_java_workflow.xml>を、IM-Workflowのインポート画面からインポートしてください。

次の手順に沿って、チュートリアルを進めます。

PDFファイルの変換

本項目では、JavaEE開発モデル用APIを使用したユーザプログラムを作成後、IM-Workflowの承認アクション処理としてユーザプログラムを実行し、承認時の処理詳細画面（HTMLファイル）をPDFファイルに変換します。

手順

- プログラムを作成する
 - PDFファイル変換処理用の Java ファイルを作成する
- プログラムを登録する
- プログラムを実行・確認する
 - 申請する
 - 承認する

プログラムを作成する

PDFファイル変換処理用の Java ファイルを作成する

1. <javaee/HtmlToPdfProcess.java>をテキストエディタで開きます。

2. 9行目を次のように修正し、クラスを指定します。

- スタンドアローン構成 の場合

```
import jp.co.iothe.pdfa_htmltopdf.HtmlToPdf;
```

- 分散構成 の場合

```
import jp.co.iothe.pdfa_htmltopdf.HtmlToPdfRemote;
```

3. 31行目を次のように修正し、PDFファイル名の接頭文字を指定します。

```
String prefix = "htmltopdf_";
```

4. 39行目を次のように修正し、PDFファイルの出力先フォルダを指定します。

```
String dirPath = "pdfa/tutorial/htmltopdf/sync";
```

5. 44行目を次のように修正し、クラスを指定します。

- スタンドアローン構成 の場合

```
HtmlToPdf htmlToPdf = new HtmlToPdf();
```

- 分散構成 の場合

```
HtmlToPdfRemote htmlToPdf = new HtmlToPdfRemote();
```

6. 上書き保存します。



注意

文字コードを UTF-8 にして保存してください。

7. <javaee/HtmlToPdfProcess.java>をコンパイルします。

8. クラスファイル、または、JARファイルを<%RESIN_HOME%/webapps/{warファイルと同名のディレクトリ}/WEB-INF>配下の<classes>、または、<lib>に設置します。



注意

クラスパスが<jp.co.pdfa_htmltopdf.tutorial.sync.HtmlToPdfProcess>となるように設置してください。

9. <%PUBLIC_STORAGE_PATH% /pdfa/tutorial/htmltopdf/sync>ディレクトリを作成します。

プログラムを登録する

設定したプログラムを環境に適用するため、Web Application Server を再起動します。

プログラムを実行・確認する

申請する

1. サンプルユーザの「上田辰男」（ユーザコード：ueda パスワード：ueda）で、一般ユーザ画面<http://<HOST>:

2. 「サイトマップ」 - 「ワークフロー」 - 「一覧」 - 「申請一覧」をクリックします。



3. 「フロー」タブ - 「HTML→PDF変換 同期 JavaEE開発モデル」をクリックします。



4. 申請画面が表示されるため、適切な値を入力し、「申請」をクリックします。

立替経費申請書

記入日 : 2025/01/09 所属 : 開発部

ユーザーID : dev12345 氏名 : 上田辰男

申請情報

支払金額 :

支払先 : XXXX文具店

支払目的 : XXXXXXXXのため

支払内容 :
XXXXXX (2個) : ¥5,000
XXXXXX (1個) : ¥2,345

申請 **一時保存**

**コラム**

エラーが発生した場合は、エラーメッセージの内容に従い入力値の修正等を行ってください。

5. 「申請」をクリックします。

申請 [申請]

フロー

| | |
|----------|---------------------------|
| 案件名 * | HTML→PDF変換 同期 JavaEE開発モデル |
| 申請者 | 上田辰男 |
| 申請基準日 | 2025/01/09 |
| 担当組織 * | サンプル課 2 2 |
| 優先度 | 通常 |
| + コメント | |
| + 添付ファイル | |

申請

6. 「決定」をクリックします。

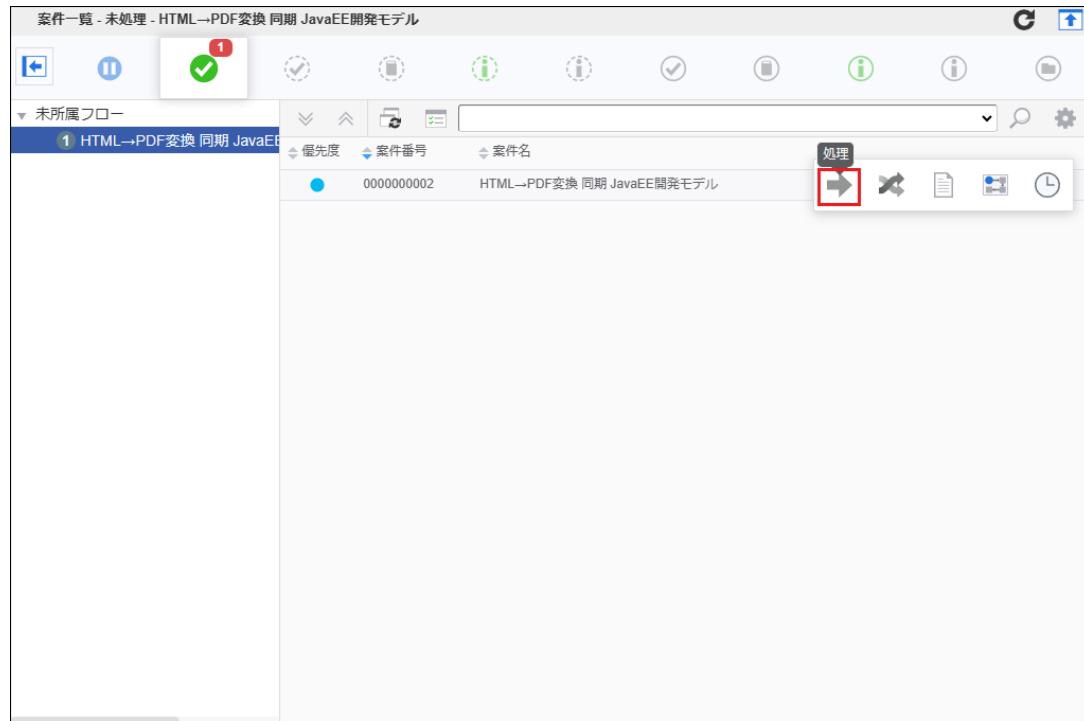


[承認する](#)

- サンプルユーザの「青柳辰巳」（ユーザコード : aoyagi パスワード:aoyagi）で、一般ユーザ画面<http://<HOST>:<PORT>/<CONTEXT_PATH>/login>にログインします。
- 「サイトマップ」 - 「ワークフロー」 - 「一覧」 - 「案件一覧」をクリックします。



3. 「未処理」タブの一覧から、「[申請する](#)」で申請した案件を選択し、「処理」アイコンをクリックします。



4. 承認画面が表示されるため、「承認」をクリックします。

立替経費申請書

| | | | |
|-------------|--|------|------|
| 記入日 : | 2025/01/09 | 所属 : | 開発部 |
| ユーザーID : | dev12345 | 氏名 : | 上田辰男 |
| 申請情報 | | | |
| 支払金額 : | 12,345 | | |
| 支払先 : | XXXX文房具店 | | |
| 支払目的 : | XXXXXXXXXのため | | |
| 支払内容 : | XXXXXX (2個) : ¥5,000 XXXXXX (1個) : ¥2,345 | | |
| 承認 | | | |

5. 「承認」をクリックします。

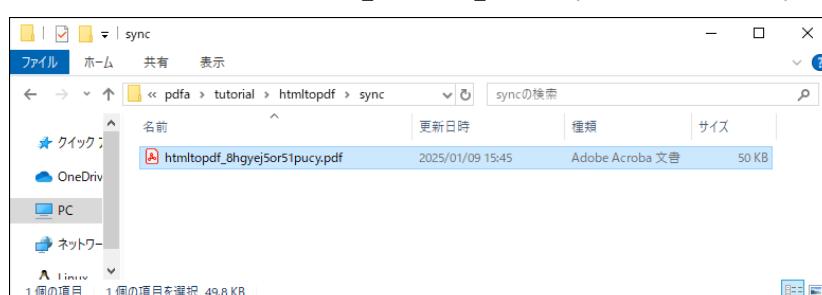
処理 [承認]

| | | | | | | | |
|-----------|---|-----|------|-------|------------|-----|------------|
| 処理種別 * | 承認 | | | | | | |
| 案件番号 | 000000002 | | | | | | |
| 案件名 | HTML→PDF変換 同期 JavaEE開発モデル | | | | | | |
| 申請情報 | <table border="1"> <tr> <td>申請者</td> <td>上田辰男</td> </tr> <tr> <td>申請基準日</td> <td>2025/01/09</td> </tr> <tr> <td>申請日</td> <td>2025/01/09</td> </tr> </table> | 申請者 | 上田辰男 | 申請基準日 | 2025/01/09 | 申請日 | 2025/01/09 |
| 申請者 | 上田辰男 | | | | | | |
| 申請基準日 | 2025/01/09 | | | | | | |
| 申請日 | 2025/01/09 | | | | | | |
| 処理者 * | 青柳辰巳 | | | | | | |
| 担当組織 * | サンプル課 1 1 | | | | | | |
| 承認 | | | | | | | |

6. 「決定」をクリックします。



プログラムが実行され、<%PUBLIC_STORAGE_PATH% /pdfa/tutorial/htmltopdf/sync>にPDFファイルが出力されます。

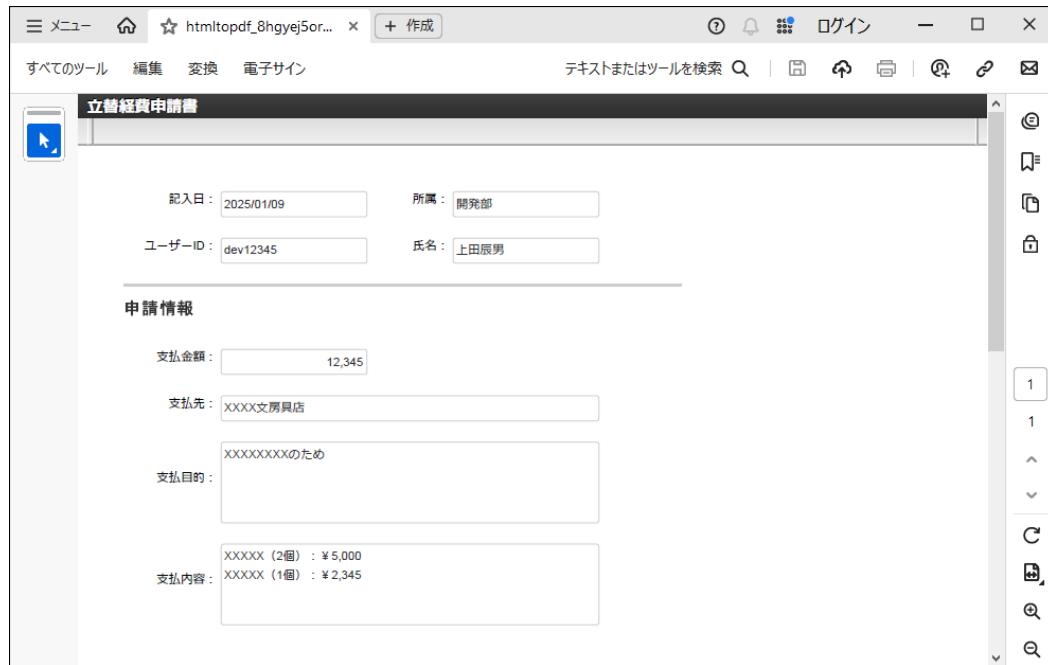




コラム

エラーが発生した場合は、例外ログの内容に従いファイルの修正等を行ってください。

7. 出力されたPDFファイルをPDFビューア（Adobe Acrobat Readerなど）で開き、正しく表示されることを確認します。



スクリプト開発モデル

本チュートリアルでは、スクリプト開発モデル用APIを使用したIM-Workflowのユーザプログラムを作成し、そのプログラムを実行することで、IM-Workflowの画面（HTMLファイル）をPDFファイルに変換します。

処理のタイミングは、IM-Workflowの承認ノードのアクション処理となります。

チュートリアルを実施するにあたり、次のzipファイルをダウンロードし、解凍してください。

< [htmltopdf_sync_js_tutorial.zip](#) >

解凍したファイルの構成は、次の通りです。

| フォルダ名／ファイル名 | 説明 |
|--------------------------------|--|
| import/ | HTML→PDF変換インポート関連フォルダ |
| htmltopdf_sync_js_forma.zip | IM-FormaDesigner for Accel Platformで作成したアプリケーション情報ファイル |
| htmltopdf_sync_js_workflow.xml | IM-Workflowのコンテンツ定義、ルート定義、および、フロー定義の定義情報ファイル |
| jssp/ | スクリプト開発モデル用フォルダ |
| HtmlToPdfProcess.js | スクリプト開発モデル用プログラム |

< import/htmltopdf_sync_js_forma.zip >を、IM-FormaDesigner for Accel Platformのアプリケーション情報インポート画面からインポートしてください。

< import/htmltopdf_sync_js_workflow.xml >を、IM-Workflowのインポート画面からインポートしてください。

次の手順に沿って、チュートリアルを進めます。

PDFファイルの変換

本項目では、スクリプト開発モデル用APIを使用したユーザプログラムを作成後、IM-Workflowの承認アクション処理としてユーザプログラムを実行し、承認時の処理詳細画面（HTMLファイル）をPDFファイルに変換します。

手順

- プログラムを作成する
 - PDFファイル変換処理用のJSファイルを作成する
- プログラムを登録する
- プログラムを実行・確認する
 - 申請する
 - 承認する

プログラムを作成する

PDFファイル変換処理用のJSファイルを作成する

1. <jssp/HtmlToPdfProcess.js>をテキストエディタで開きます。
2. 9行目を次のように修正し、PDFファイル名の接頭文字を指定します。

```
const prefix = "htmltopdf_";
```

3. 15行目を次のように修正し、PDFファイルの出力先フォルダを指定します。

```
const dirPath = "pdfa/tutorial/htmltopdf/sync";
```

4. 20行目を次のように修正し、クラスを指定します。

- スタンドアローン構成 の場合

```
const htmlToPdf = new HtmlToPdf();
```

- 分散構成 の場合

```
const htmlToPdf = new HtmlToPdfRemote();
```

5. 上書き保存します。



注意

文字コードを UTF-8 にして保存してください。

6. <%RESIN_HOME%/webapps/{warファイルと同名のディレクトリ}/WEB-INF/jssp/src/pdfa/tutorial/htmltopdf/sync>ディレクトリを作成します。
7. <jssp/HtmlToPdfProcess.js>を<%RESIN_HOME%/webapps/{warファイルと同名のディレクトリ}/WEB-INF/jssp/src/pdfa/tutorial/htmltopdf/sync>配下に設置します。
8. <%PUBLIC_STORAGE_PATH% /pdfa/tutorial/htmltopdf/sync>ディレクトリを作成します。

プログラムを登録する

設置したプログラムを環境に適用するため、Web Application Server を再起動します。

プログラムを実行・確認する

申請する

1. サンプルユーザの「上田辰男」（ユーザコード：ueda パスワード：ueda）で、一般ユーザ画面<http://<HOST>:<PORT>/<CONTEXT_PATH>/login>にログインします。
2. 「サイトマップ」 - 「ワークフロー」 - 「一覧」 - 「申請一覧」をクリックします。



3. 「フロー」タブ-「HTML→PDF変換 同期 スクリプト開発モデル」をクリックします。

| フロー名 | 備考 |
|--|----|
| HTML→PDF変換 同期 スクリプト開発モデル | |
| 分岐ルート[JavaEE開発モデル] | |
| 分岐ルート[スクリプト開発モデル] | |
| 分岐ルート[処理モーダル] | |
| 横配置ルート[JavaEE開発モデル] | |
| 横配置ルート[スクリプト開発モデル] | |
| 横配置ルート[処理モーダル] | |

4. 申請画面が表示されるため、適切な値を入力し、「申請」をクリックします。

立替経費申請書

記入日 : 2025/01/10 31 所属 : 開発部

ユーザーID : dev12345 氏名 : 上田辰男

申請情報

支払金額 : 12,345

支払先 : XXXX文房具店

支払目的 : XXXXXXXXのため

支払内容 : XXXXX (2個) : ¥5,000
XXXXX (1個) : ¥2,345

操作

申請

一時保存

**コラム**

エラーが発生した場合は、エラーメッセージの内容に従い入力値の修正等を行ってください。

5. 「申請」をクリックします。

申請 [申請]

フロー

| | |
|----------|--------------------------|
| 案件名 * | HTML→PDF変換 同期 スクリプト開発モデル |
| 申請者 | 上田辰男 |
| 申請基準日 | 2025/01/10 |
| 担当組織 * | サンプル課 2 2 |
| 優先度 | 通常 |
| + コメント | |
| + 添付ファイル | |

申請

6. 「決定」をクリックします。

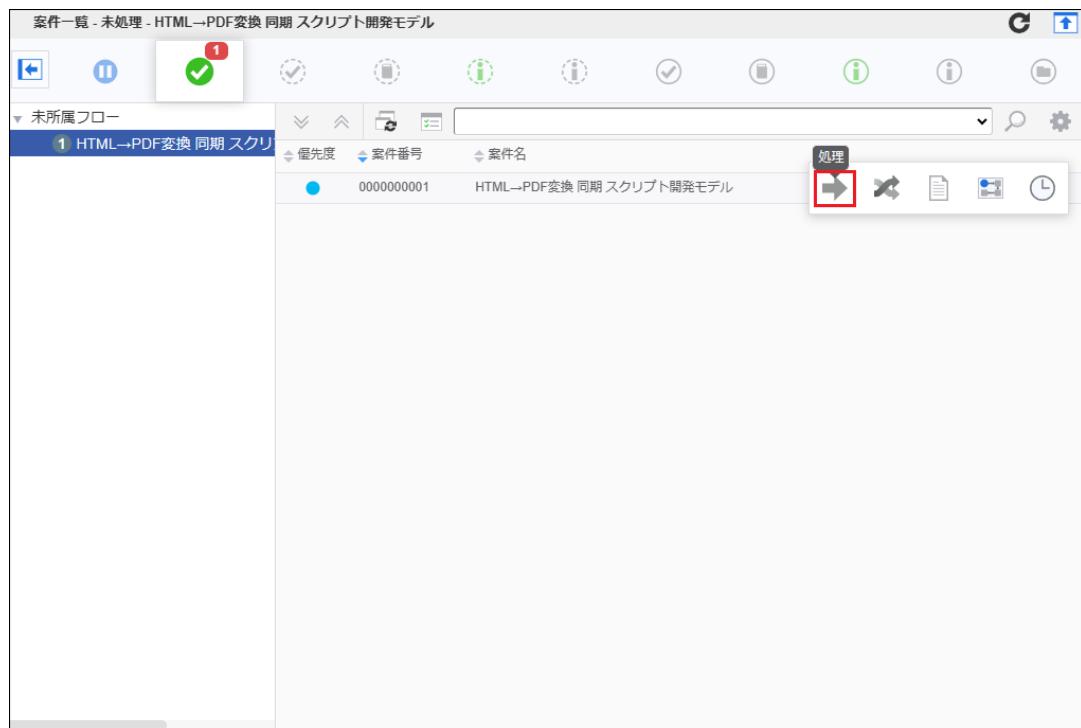


[承認する](#)

- サンプルユーザの「青柳辰巳」（ユーザコード : aoyagi パスワード:aoyagi）で、一般ユーザ画面< http://<HOST>:<PORT>/<CONTEXT_PATH>/login >にログインします。
- 「サイトマップ」 - 「ワークフロー」 - 「一覧」 - 「案件一覧」をクリックします。



3. 「未処理」タブの一覧から、「[申請する](#)」で申請した案件を選択し、「処理」アイコンをクリックします。



4. 承認画面が表示されるため、「承認」をクリックします。

立替経費申請書

| | | | |
|-------------|--|------|------|
| 記入日 : | 2025/01/10 | 所属 : | 開発部 |
| ユーザーID : | dev12345 | 氏名 : | 上田辰男 |
| 申請情報 | | | |
| 支払金額 : | 12,345 | | |
| 支払先 : | XXXX文房具店 | | |
| 支払目的 : | XXXXXXXXXのため | | |
| 支払内容 : | XXXXXX (2個) : ¥5,000 XXXXXX (1個) : ¥2,345 | | |
| 承認 | | | |

5. 「承認」をクリックします。

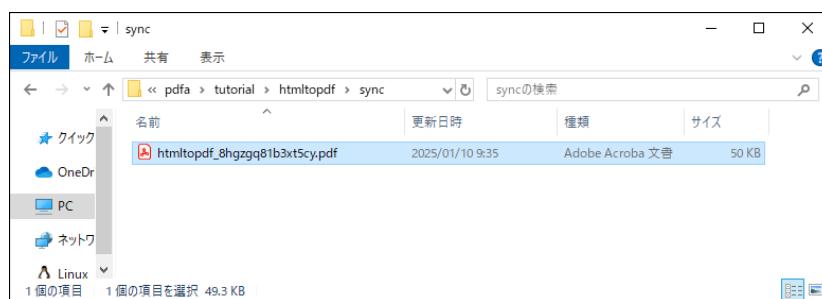
処理 [承認]

| | | | | | | | |
|-----------|---|-----|------|-------|------------|-----|------------|
| 処理種別 * | 承認 | | | | | | |
| 案件番号 | 0000000001 | | | | | | |
| 案件名 | HTML→PDF変換 同期 スクリプト開発モデル | | | | | | |
| 申請情報 | <table border="1"> <tr> <td>申請者</td> <td>上田辰男</td> </tr> <tr> <td>申請基準日</td> <td>2025/01/10</td> </tr> <tr> <td>申請日</td> <td>2025/01/10</td> </tr> </table> | 申請者 | 上田辰男 | 申請基準日 | 2025/01/10 | 申請日 | 2025/01/10 |
| 申請者 | 上田辰男 | | | | | | |
| 申請基準日 | 2025/01/10 | | | | | | |
| 申請日 | 2025/01/10 | | | | | | |
| 処理者 * | 青柳辰巳 | | | | | | |
| 担当組織 * | サンプル課 1 1 | | | | | | |
| 承認 | | | | | | | |

6. 「決定」をクリックします。



プログラムが実行され、<%PUBLIC_STORAGE_PATH% /pdfa/tutorial/htmltopdf/sync>にPDFファイルが出力されます。

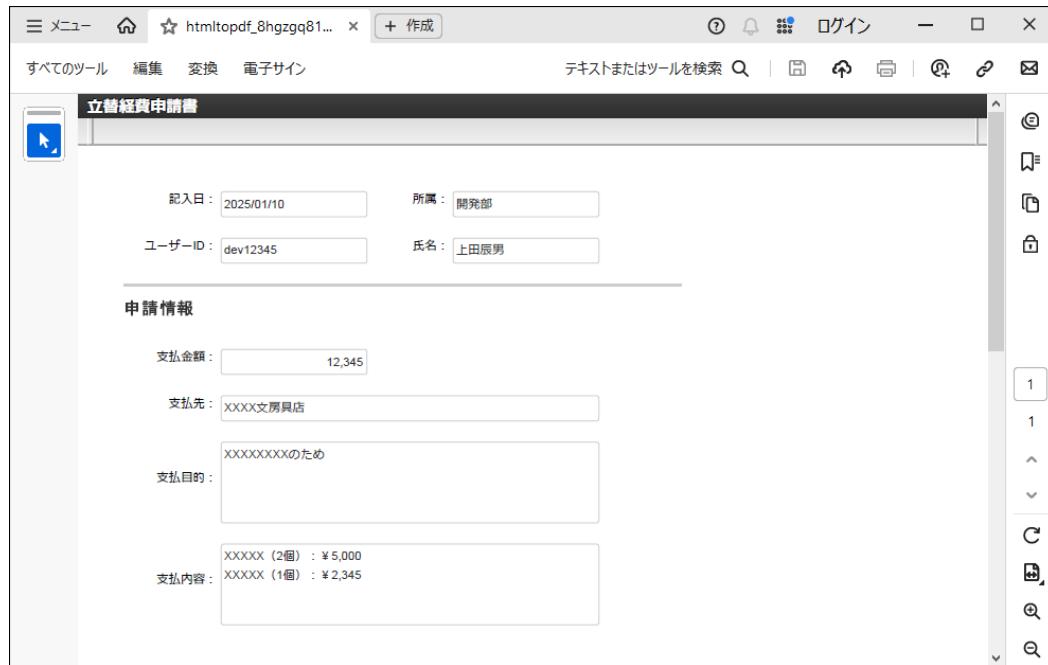




コラム

エラーが発生した場合は、例外ログの内容に従いファイルの修正等を行ってください。

7. 出力されたPDFファイルをPDFビューア（Adobe Acrobat Readerなど）で開き、正しく表示されることを確認します。



非同期処理

JavaEE開発モデル

本チュートリアルでは、JavaEE開発モデル用APIを使用したジョブ、および、IM-Workflowのユーザプログラムを作成し、そのプログラムを非同期で実行することで、IM-Workflowの画面（HTMLファイル）をPDFファイルに変換します。

変換処理のタイミングは、IM-Workflowの案件終了処理となります。

処理の全体の流れは、次の通りです。

1. 承認ノードのアクション処理

セッション情報を取得し、案件プロパティにて保持します。

セッション情報は同期処理でのみ取得可能です。

対象のプログラムファイルは<SessionSaveProcess.java>です。

2. 案件終了処理

セッション情報と案件情報を指定し、ジョブを実行します。

対象のプログラムファイルは<JobExecuteProcess.java>です。

3. ジョブ

セッション情報と案件情報をもとに、HTML→PDF変換処理を実行します。

対象のプログラムファイルは<HtmlToPdfJob.java>です。

チュートリアルを実施するにあたり、次のzipファイルをダウンロードし、解凍してください。

<[htmltopdf_async_java_tutorial.zip](#)>

解凍したファイルの構成は、次の通りです。

| フォルダ名／ファイル名 | 説明 |
|--|---|
| import/ | HTML→PDF変換インポート関連フォルダ |
| htmltopdf_async_java_forma.zip | IM-FormaDesigner for Accel Platformで作成したアプリケーションのアプリケーション情報ファイル |
| htmltopdf_async_java_workflow.xml | IM-Workflowのコンテンツ定義、ルート定義、フロー定義、および、案件プロパティ定義の定義情報ファイル |
| htmltopdf_async_java_job-scheduler.xml | HTML→PDF変換用のジョブ情報ファイル |

| フォルダ名／ファイル名 | 説明 |
|-------------------------|--------------------|
| javaee/ | JavaEE開発モデル 用フォルダ |
| HtmlToPdfJob.java | JavaEE開発モデル 用プログラム |
| JobExecuteProcess.java | |
| SessionSaveProcess.java | |

< import/htmltopdf_async_java_forma.zip >を、 IM-FormaDesigner for Accel Platform のアプリケーション情報インポート画面からインポートしてください。

< import/htmltopdf_async_java_workflow.xml >を、 IM-Workflow のインポート画面からインポートしてください。

次の手順に沿って、チュートリアルを進めます。

PDFファイルの変換

本項目では、 JavaEE開発モデル 用APIを使用したジョブ、および、ユーザプログラムを作成後、 IM-Workflow の案件終了処理としてユーザプログラムを実行し、承認時の処理詳細画面（HTMLファイル）をPDFファイルに変換します。

手順

- プログラムを作成する
 - PDFファイル変換処理のジョブ用の Java ファイルを作成する
 - PDFファイル変換処理のジョブ実行用の Java ファイルを作成する
 - セッション情報取得用の Java ファイルを設置する
- プログラムを登録する
- プログラムを実行・確認する
 - 申請する
 - 承認する

プログラムを作成する

PDFファイル変換処理のジョブ用の Java ファイルを作成する

1. < javaee/HtmlToPdfJob.java >をテキストエディタで開きます。
2. 8行目を次のように修正し、クラスを指定します。
 - スタンドアローン構成 の場合

```
import jp.co.iothe.pdfa_htmltopdf.HtmlToPdf;
```

- 分散構成 の場合

```
import jp.co.iothe.pdfa_htmltopdf.HtmlToPdfRemote;
```

3. 41行目を次のように修正し、PDFファイル名の接頭文字を指定します。

```
String prefix = "htmltopdf_";
```

4. 49行目を次のように修正し、PDFファイルの出力先フォルダを指定します。

```
String dirPath = "pdfa/tutorial/htmltopdf/async";
```

5. 54行目を次のように修正し、クラスを指定します。

- スタンドアローン構成 の場合

```
HtmlToPdf htmlToPdf = new HtmlToPdf();
```

- 分散構成 の場合

```
HtmlToPdfRemote htmlToPdf = new HtmlToPdfRemote();
```

6. 56行目の引数< sessionKey >を次のように修正し、セッションIDのCookie名を指定します。

```
htmlToPdf.convertWorkflowScreenWithSessionInfo(systemMatterId, userDataId, contextRoot, "JSESSIONID", sessionId,  
tempFile.getPath());
```



注意

Cookie名のデフォルト値は< JSESSIONID >です。

Cookie名は、設定ファイルにて任意の値を設定することができます。

デフォルト値から変更している場合は、引数< sessionKey >の値を適宜変更してください。

設定ファイルの詳細については、「[intra-mart Accel Platform 設定ファイルリファレンス](#)」 - 「Cookie名」を参照してください。

7. 上書き保存します。



注意

文字コードを UTF-8 にして保存してください。

8. < javaee/HtmlToPdfJob.java >をコンパイルします。

9. クラスファイル、または、JARファイルを< %RESIN_HOME%/webapps/{warファイルと同名のディレクトリ}/WEB-INF >配下の< classes >、または、< lib >に設置します。



注意

クラスパスが< jp.co.pdfa_htmltopdf.tutorial.async.HtmlToPdfJob >となるように設置してください。

10. < %PUBLIC_STORAGE_PATH% /pdfa/tutorial/htmltopdf/async >ディレクトリを作成します。

PDFファイル変換処理のジョブ実行用の Java ファイルを作成する

1. < javaee/JobExecuteProcess.java >をテキストエディタで開きます。

2. 31行目を次のように修正し、ジョブIDを指定します。

```
String jobId = "htmltopdf-async-java-job";
```

3. 33行目を次のように修正し、ジョブネットIDを指定します。

```
String jobnetId = "htmltopdf-async-java-jobnet";
```

4. 上書き保存します。



注意

文字コードを UTF-8 にして保存してください。

5. < javaee/JobExecuteProcess.java >をコンパイルします。

6. クラスファイル、または、JARファイルを< %RESIN_HOME%/webapps/{warファイルと同名のディレクトリ}/WEB-INF >配下の< classes >、または、< lib >に設置します。



注意

クラスパスが< jp.co.pdfa_htmltopdf.tutorial.async.JobExecuteProcess >となるように設置してください。

セッション情報取得用の Java ファイルを設置する

1. < javaee/SessionSaveProcess.java >をコンパイルします。

2. クラスファイル、または、JARファイルを< %RESIN_HOME%/webapps/{warファイルと同名のディレクトリ}/WEB-INF >配下の< classes >、または、< lib >に設置します。



注意

クラスパスが< jp.co.pdfa_htmltopdf.tutorial.async.SessionSaveProcess >となるように設置してください。

1. 設置したプログラムを環境に適用するため、 Web Application Server を再起動します。
2. < import/htmltopdf_async_java_job-scheduler.xml >をテキストエディタで開きます。
3. 25行目の実行パラメータ< contextRoot >に、コンテキストルート< http://<HOST>:<PORT>/<CONTEXT_PATH> >を指定し、上書き保存します。

```
<parameter key="contextRoot">%コンテキストルート%</parameter>
```



注意

文字コードを UTF-8 にして保存してください。

4. < import/htmltopdf_async_java_job-scheduler.xml >を、 intra-mart Accel Platform のジョブインポート用のジョブからインポートします。

プログラムを実行・確認する

申請する

1. サンプルユーザの「上田辰男」（ユーザコード : ueda パスワード:ueda）で、一般ユーザ画面< http://<HOST>:<PORT>/<CONTEXT_PATH>/login >にログインします。
2. 「サイトマップ」 - 「ワークフロー」 - 「一覧」 - 「申請一覧」をクリックします。



3. 「フロー」タブ-「HTML→PDF変換 非同期 JavaEE開発モデル」をクリックします。

申請一覧

申請基準日: 2025/01/09 [31]

Search: Search

フロー名 備考

- HTML→PDF変換 非同期 JavaEE開発モデル**
- 分歧ルート[JavaEE開発モデル]
- 分歧ルート[スクリプト開発モデル]
- 分歧ルート[処理モーダル]
- 横配置ルート[JavaEE開発モデル]
- 横配置ルート[スクリプト開発モデル]
- 横配置ルート[処理モーダル]

4. 申請画面が表示されるため、適切な値を入力し、「申請」をクリックします。

立替経費申請書

記入日: 2025/01/09 [31] 所属: 開発部

ユーザーID: dev12345 氏名: 上田辰男

申請情報

支払金額: 12,345

支払先: XXXX文房具店

支払目的: XXXXXXXXのため

支払内容: XXXXX (2個) : ¥5,000
XXXXX (1個) : ¥2,345

**コラム**

エラーが発生した場合は、エラーメッセージの内容に従い入力値の修正等を行ってください。

5. 「申請」をクリックします。

案件名 * HTML→PDF変換 非同期 JavaEE開発モデル
申請者 上田辰男
申請基準日 2025/01/09
担当組織 * サンプル課 2 2
優先度 通常
+ コメント
+ 添付ファイル

申請

6. 「決定」をクリックします。



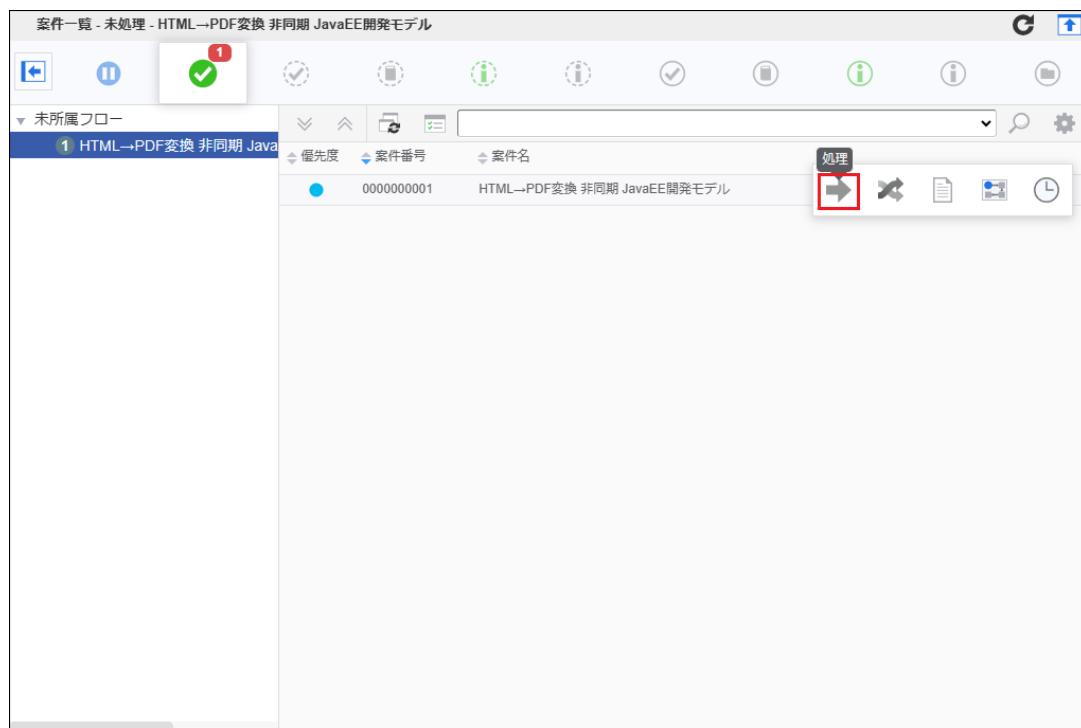
[承認する](#)

- サンプルユーザの「青柳辰巳」（ユーザコード : aoyagi パスワード:aoyagi）で、一般ユーザ画面< http://<HOST>:<PORT>/<CONTEXT_PATH>/login >にログインします。
- 「サイトマップ」 - 「ワークフロー」 - 「一覧」 - 「案件一覧」をクリックします。

サイトマップ

- FileExchange
- ポータル
- ワークフロー
 - 印影設定
 - 代理設定
 - 代理先設定
 - 代理元確認
 - 検索
 - 一覧
 - 申請一覧
 - 案件一覧**
- ContentsSearch
- 共通マスター
- 個人設定
- サンプル

- 「未処理」タブの一覧から、「[申請する](#)」で申請した案件を選択し、「処理」アイコンをクリックします。



4. 承認画面が表示されるため、「承認」をクリックします。

A screenshot of the '立替経費申請書' (Reimbursement Application Form) dialog box. It contains fields for '記入日' (Input Date), '所属' (Department), 'ユーザーID' (User ID), and '氏名' (Name). Below these are sections for '申請情報' (Application Information) including '支払額' (Amount), '支払先' (Payee), '支払目的' (Purpose), and '支払内容' (Content). At the bottom is a large red-bordered '承認' (Approve) button.

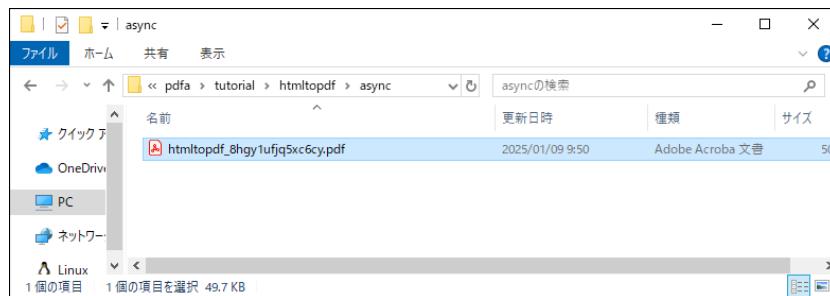
5. 「承認」をクリックします。



6. 「決定」をクリックします。



プログラムが実行され、<%PUBLIC_STORAGE_PATH% /pdfa/tutorial/htmltopdf/async>にPDFファイルが出力されます。



コラム

エラーが発生した場合は、「ジョブネットモニタ詳細」画面のエラーメッセージや、例外ログの内容に従いファイルの修正等を行ってください。

ジョブネットモニタ詳細

| ジョブネット | HTML→PDF変換 非同期 JavaEE開発モデル (htmltopdf-async-java-jobnet) | |
|--------|--|--|
| トリガ | 即時実行 (MT_3mrkcr0nodk21) | |
| ステータス | エラー | |
| 開始日 | 2025/01/09 9:56:55 | |
| 終了日 | 2025/01/09 9:57:04 | |
| メッセージ | | |

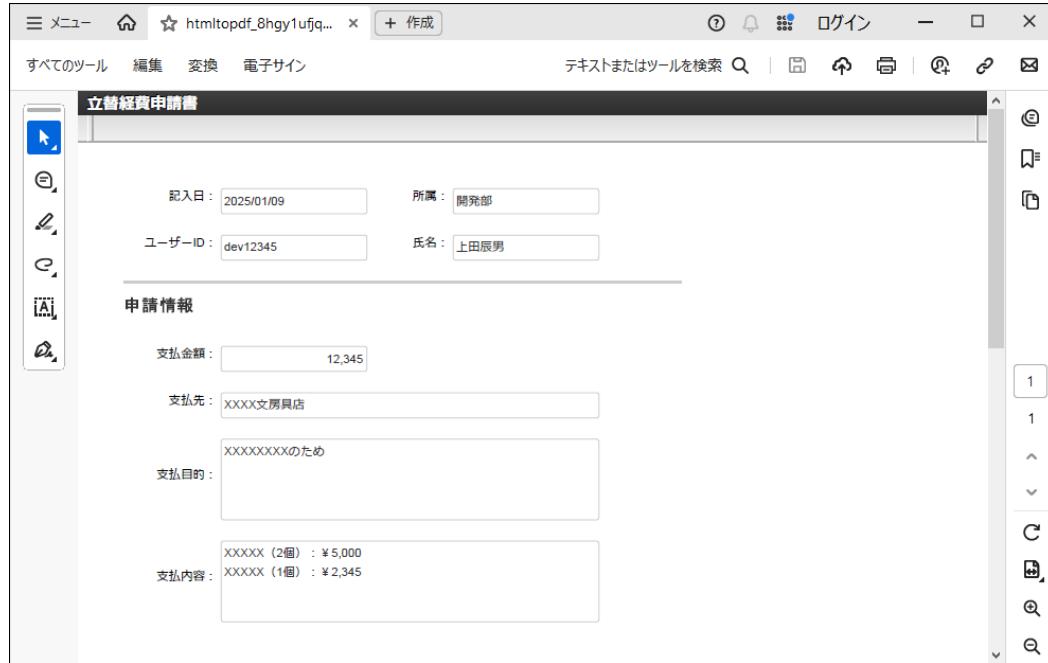
タスク一覧

| タスクID | ジョブ | ステータス | 開始日 | 終了日 | メッセージ |
|-------|---|-------|--------------------|--------------------|------------------|
| 0 | HTML→PDF変換 非同期 JavaEE開発モデル (htmltopdf-a...) | エラー | 2025/01/09 9:56:55 | 2025/01/09 9:57:04 | 予期しないエラーが発生しました。 |

実行中パラメータ

再実行

7. 出力されたPDFファイルをPDFビューア（Adobe Acrobat Readerなど）で開き、正しく表示されることを確認します。



スクリプト開発モデル

本チュートリアルでは、スクリプト開発モデル用APIを使用したジョブ、および、IM-Workflowのユーザプログラムを作成し、そのプログラムを非同期で実行することで、IM-Workflowの画面（HTMLファイル）をPDFファイルに変換します。

変換処理のタイミングは、IM-Workflowの案件終了処理となります。

処理の全体の流れは、次の通りです。

1. 承認ノードのアクション処理

セッション情報を取得し、案件プロパティにて保持します。

セッション情報は同期処理でのみ取得可能です。

対象のプログラムファイルは<SessionSaveProcess.js>です。

2. 案件終了処理

セッション情報と案件情報を指定し、ジョブを実行します。

対象のプログラムファイルは<JobExecuteProcess.js>です。

3. ジョブ

セッション情報と案件情報をもとに、HTML→PDF変換処理を実行します。

対象のプログラムファイルは<HtmlToPdfJob.js>です。

チュートリアルを実施するにあたり、次のzipファイルをダウンロードし、解凍してください。

<[htmltopdf_async_js_tutorial.zip](#)>

解凍したファイルの構成は、次の通りです。

| フォルダ名／ファイル名 | 説明 |
|--------------------------------------|---|
| import/ | HTML→PDF変換インポート関連フォルダ |
| htmltopdf_async_js_forma.zip | IM-FormaDesigner for Accel Platformで作成したアプリケーションのアプリケーション情報ファイル |
| htmltopdf_async_js_workflow.xml | IM-Workflowのコンテンツ定義、ルート定義、フロー定義、および、案件プロパティ定義の定義情報ファイル |
| htmltopdf_async_js_job-scheduler.xml | HTML→PDF変換用のジョブ情報ファイル |
| jssp/ | スクリプト開発モデル用フォルダ |
| HmlToPdfJob.js | スクリプト開発モデル用プログラム |
| JobExecuteProcess.js | |
| SessionSaveProcess.js | |

<import/htmltopdf_async_js_forma.zip>を、IM-FormaDesigner for Accel Platform のアプリケーション情報インポート画面からインポートしてください。

<import/htmltopdf_async_js_workflow.xml>を、IM-Workflow のインポート画面からインポートしてください。

次の手順に沿って、チュートリアルを進めます。

PDFファイルの変換

本項目では、スクリプト開発モデル用APIを使用したジョブ、および、ユーザプログラムを作成後、IM-Workflow の案件終了処理としてユーザプログラムを実行し、承認時の処理詳細画面（HTMLファイル）をPDFファイルに変換します。

手順

- プログラムを作成する
 - PDFファイル変換処理のジョブ用のJSファイルを作成する
 - PDFファイル変換処理のジョブ実行用のJSファイルを作成する
 - セッション情報取得用のJSファイルを設置する
- プログラムを登録する
- プログラムを実行・確認する
 - 申請する
 - 承認する

プログラムを作成する

PDFファイル変換処理のジョブ用のJSファイルを作成する

1. <jssp/HtmIToPdfJob.js>をテキストエディタで開きます。
2. 17行目を次のように修正し、PDFファイル名の接頭文字を指定します。

```
const prefix = "htmltopdf_";
```

3. 23行目を次のように修正し、PDFファイルの出力先フォルダを指定します。

```
const dirPath = "pdfa/tutorial/htmltopdf/async";
```

4. 28行目を次のように修正し、クラスを指定します。

- スタンドアローン構成の場合

```
const htmlToPdf = new HtmIToPdf();
```

- 分散構成の場合

```
const htmlToPdf = new HtmIToPdfRemote();
```

5. 30行目の引数<sessionKey>を次のように修正し、セッションIDのCookie名を指定します。

```
htmlToPdf.convertWorkflowScreenWithSessionInfo(systemMatterId, userDataId, contextRoot, "JSESSIONID", sessionId, tempFile.path());
```



注意

Cookie名のデフォルト値は<JSESSIONID>です。

Cookie名は、設定ファイルにて任意の値を設定することができます。

デフォルト値から変更している場合は、引数<sessionKey>の値を適宜変更してください。

設定ファイルの詳細については、「[intra-mart Accel Platform 設定ファイルリファレンス](#)」 - 「Cookie名」を参照してください。

6. 上書き保存します。



注意

文字コードを UTF-8 にして保存してください。

7. <%RESIN_HOME%/webapps/{warファイルと同名のディレクトリ}/WEB-INF/jssp/src/pdfa/tutorial/htmltopdf/async>ディレクトリを作成します。
8. <jssp/HtmlToPdfJob.js>を<%RESIN_HOME%/webapps/{warファイルと同名のディレクトリ}/WEB-INF/jssp/src/pdfa/tutorial/htmltopdf/async>配下に設置します。
9. <%PUBLIC_STORAGE_PATH% /pdfa/tutorial/htmltopdf/async>ディレクトリを作成します。

PDFファイル変換処理のジョブ実行用のJSファイルを作成する

1. <jssp/JobExecuteProcess.js>をテキストエディタで開きます。
2. 9行目を次のように修正し、ジョブIDを指定します。

```
const jobId = "htmltopdf-async-js-job";
```

3. 11行目を次のように修正し、ジョブネットIDを指定します。

```
const jobnetId = "htmltopdf-async-js-jobnet";
```

4. 上書き保存します。



注意

文字コードを UTF-8 にして保存してください。

5. <jssp/JobExecuteProcess.js>を<%RESIN_HOME%/webapps/{warファイルと同名のディレクトリ}/WEB-INF/jssp/src/pdfa/tutorial/htmltopdf/async>配下に設置します。

セッション情報取得用のJSファイルを設置する

<jssp/SessionSaveProcess.js>を<%RESIN_HOME%/webapps/{warファイルと同名のディレクトリ}/WEB-INF/jssp/src/pdfa/tutorial/htmltopdf/async>配下に設置します。

プログラムを登録する

1. 設置したプログラムを環境に適用するため、Web Application Server を再起動します。
2. <import/htmltopdf_async_js_job-scheduler.xml>をテキストエディタで開きます。
3. 25行目の実行パラメータ<contextRoot>に、コンテキストルート<http://<HOST>:<PORT>/<CONTEXT_PATH>>を指定し、上書き保存します。

```
<parameter key="contextRoot">%コンテキストルート%</parameter>
```



注意

文字コードを UTF-8 にして保存してください。

4. <import/htmltopdf_async_js_job-scheduler.xml>を、intra-mart Accel Platform のジョブインポート用のジョブからインポートします。

プログラムを実行・確認する

申請する

1. サンプルユーザの「上田辰男」（ユーザコード：ueda パスワード：ueda）で、一般ユーザ画面<http://<HOST>:<PORT>/<CONTEXT_PATH>/login>にログインします。
2. 「サイトマップ」 - 「ワークフロー」 - 「一覧」 - 「申請一覧」をクリックします。



3. 「フロー」タブ-「HTML→PDF変換 非同期 スクリプト開発モデル」をクリックします。

The screenshot shows the 'Flow' tab in the application. At the top, there is a search bar and a date selector set to '2025/01/09'. Below the tabs, there is a list of flows:

- HTML→PDF変換 非同期
スクリプト開発モデル** (highlighted with a red box)
- 分歧ルート[JavaEE開発
モデル]
- 分歧ルート[スクリプト開
発モデル]
- 分歧ルート[処理モーダル]
- 横配置ルート[JavaEE開
発モデル]
- 横配置ルート[スクリプト
開発モデル]
- 横配置ルート[処理モーダ
ル]

4. 申請画面が表示されるため、適切な値を入力し、「申請」をクリックします。

立替経費申請書

記入日 : 2025/01/09  所属 : 開発部

ユーザーID : dev12345 氏名 : 上田辰男

申請情報

支払金額 : 12,345

支払先 : XXXX文房具店

支払目的 : XXXXXXXXのため

支払内容 : XXXXX (2個) : ¥5,000
XXXXX (1個) : ¥2,345

操作ボタン

申請 **一時保存**

**コラム**

エラーが発生した場合は、エラーメッセージの内容に従い入力値の修正等を行ってください。

5. 「申請」をクリックします。

申請 [申請]

フロー

| | |
|----------|---------------------------|
| 案件名 * | HTML→PDF変換 非同期 スクリプト開発モデル |
| 申請者 | 上田辰男 |
| 申請基準日 | 2025/01/09 |
| 担当組織 * | サンプル課 2 2 |
| 優先度 | 通常 |
| + コメント | |
| + 添付ファイル | |

申請

6. 「決定」をクリックします。

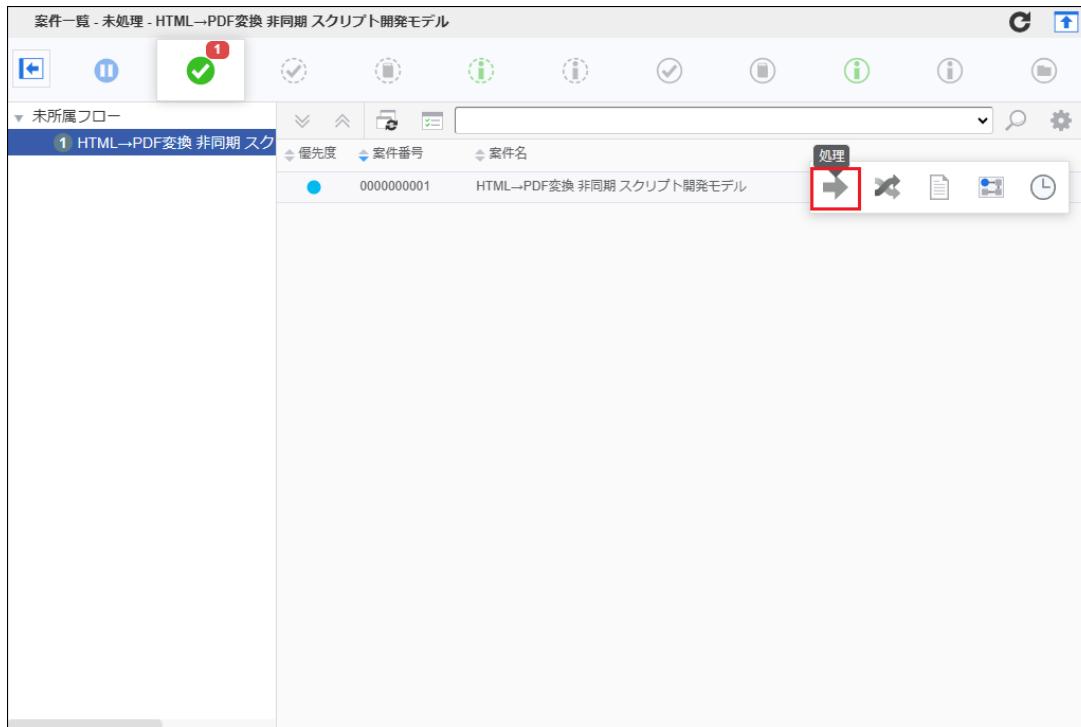


[承認する](#)

- サンプルユーザの「青柳辰巳」（ユーザコード : aoyagi パスワード:aoyagi）で、一般ユーザ画面<http://<HOST>:<PORT>/<CONTEXT_PATH>/login>にログインします。
- 「サイトマップ」 - 「ワークフロー」 - 「一覧」 - 「案件一覧」をクリックします。



3. 「未処理」タブの一覧から、「[申請する](#)」で申請した案件を選択し、「処理」アイコンをクリックします。



4. 承認画面が表示されるため、「承認」をクリックします。

立替経費申請書

| | | | |
|-------------|---|------|------|
| 記入日 : | 2025/01/09 | 所属 : | 開発部 |
| ユーザーID : | dev12345 | 氏名 : | 上田辰男 |
| 申請情報 | | | |
| 支払金額 : | 12,345 | | |
| 支払先 : | XXXX文房具店 | | |
| 支払目的 : | XXXXXXXXXのため | | |
| 支払内容 : | XXXXXX (2個) : ¥5,000 XXXXX (1個) : ¥2,345 | | |
| 承認 | | | |

5. 「承認」をクリックします。

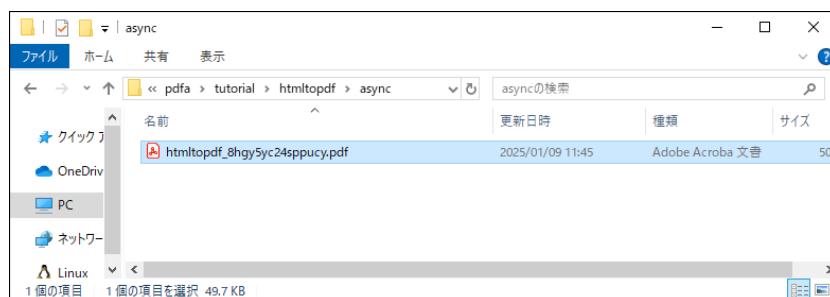
処理 [承認]

| | | | | | | | |
|-----------|---|-----|------|-------|------------|-----|------------|
| 処理種別 * | 承認 | | | | | | |
| 案件番号 | 0000000001 | | | | | | |
| 案件名 | HTML→PDF変換 非同期 スクリプト開発モデル | | | | | | |
| 申請情報 | <table border="1"> <tr> <td>申請者</td> <td>上田辰男</td> </tr> <tr> <td>申請基準日</td> <td>2025/01/09</td> </tr> <tr> <td>申請日</td> <td>2025/01/09</td> </tr> </table> | 申請者 | 上田辰男 | 申請基準日 | 2025/01/09 | 申請日 | 2025/01/09 |
| 申請者 | 上田辰男 | | | | | | |
| 申請基準日 | 2025/01/09 | | | | | | |
| 申請日 | 2025/01/09 | | | | | | |
| 処理者 * | 青柳辰巳 | | | | | | |
| 担当組織 * | サンプル課 1 1 | | | | | | |
| 承認 | | | | | | | |

6. 「決定」をクリックします。



プログラムが実行され、<%PUBLIC_STORAGE_PATH% /pdfa/tutorial/htmltopdf/async>にPDFファイルが出力されます。





コラム

エラーが発生した場合は、「ジョブネットモニタ詳細」画面のエラーメッセージや、例外ログの内容に従いファイルの修正等を行ってください。

ジョブネットモニタ詳細

| モニタ詳細 (intra-martAPP:10.0.75.1:52001736391101414) | |
|---|---|
| ジョブネット | HTML→PDF変換 非同期 スクリプト開発モデル (htmloptopdf-async-js-jobnet) |
| トリガ | 即時実行 (MT_1iew9pr90cmhm) |
| ステータス | エラー |
| 開始日 | 2025/01/09 11:52:45 |
| 終了日 | 2025/01/09 11:52:54 |
| メッセージ | <input type="text"/> |

タスク一覧

| タスクID | ジョブ | ステータス | 開始日 | 終了日 | メッセージ |
|-------|---|-------|---------------------|---------------------|---|
| 0 | HTML→PDF変換 非同期 スクリプト開発モデル (htmloptopdf-async-js-jobnet) | エラー | 2025/01/09 11:52:45 | 2025/01/09 11:52:54 | yss.htmloptopdf.exception.HtmlToPdfException: |

実行中パラメータ

再実行

7. 出力されたPDFファイルをPDFビューア（Adobe Acrobat Readerなど）で開き、正しく表示されることを確認します。

立替経費申請書

記入日： 2025/01/09 所属： 開発部

ユーザーID： dev12345 氏名： 上田辰男

申請情報

支払金額： 12,345

支払先： XXXX文房具店

支払目的： XXXXXXのため

支払内容： XXXXX (2個) : ¥5,000
XXXXX (1個) : ¥2,345

目次

- Office系→PDF変換
 - 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
- HTML→PDF変換

Office系→PDF変換

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。 |

ステータスコード 内容

| | |
|------|--|
| 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。 |
| 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](#)」 - 「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。 |
|------|--|

通信関連のステータスコード一覧

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 |
|----|-----------------------|

HTML→PDF変換

HTML→PDF変換 は、ステータスコードを返しません。

そのため、スローされたエラーを確認してください。

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。

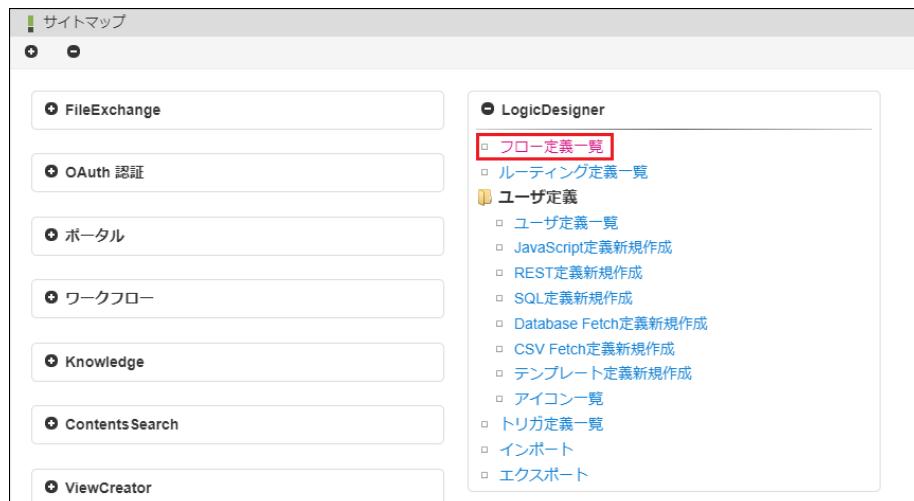
サンプルの実行手順、ユーザ定義の詳細、および、サンプル内で使用しているAPIについては、次を参照してください。

サンプル実行手順

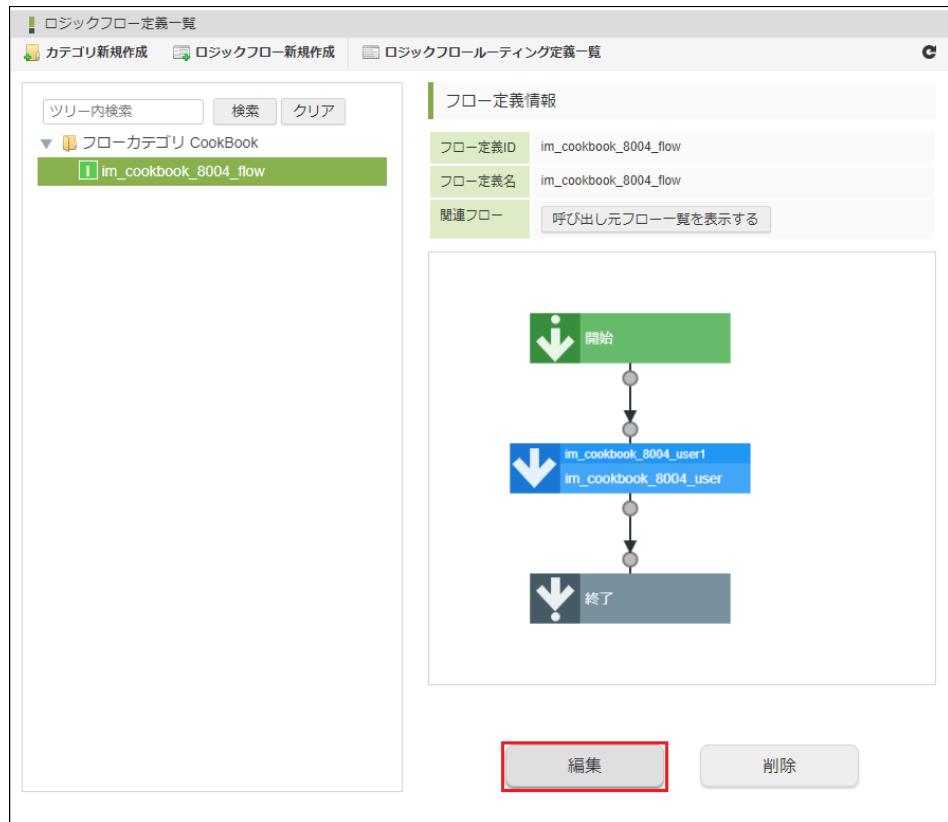
- サンプルのフロー定義のデバッグ画面を開く
- 入力値を設定し、デバッグを実行する
- 実行結果を確認する

サンプルのフロー定義のデバッグ画面を開く

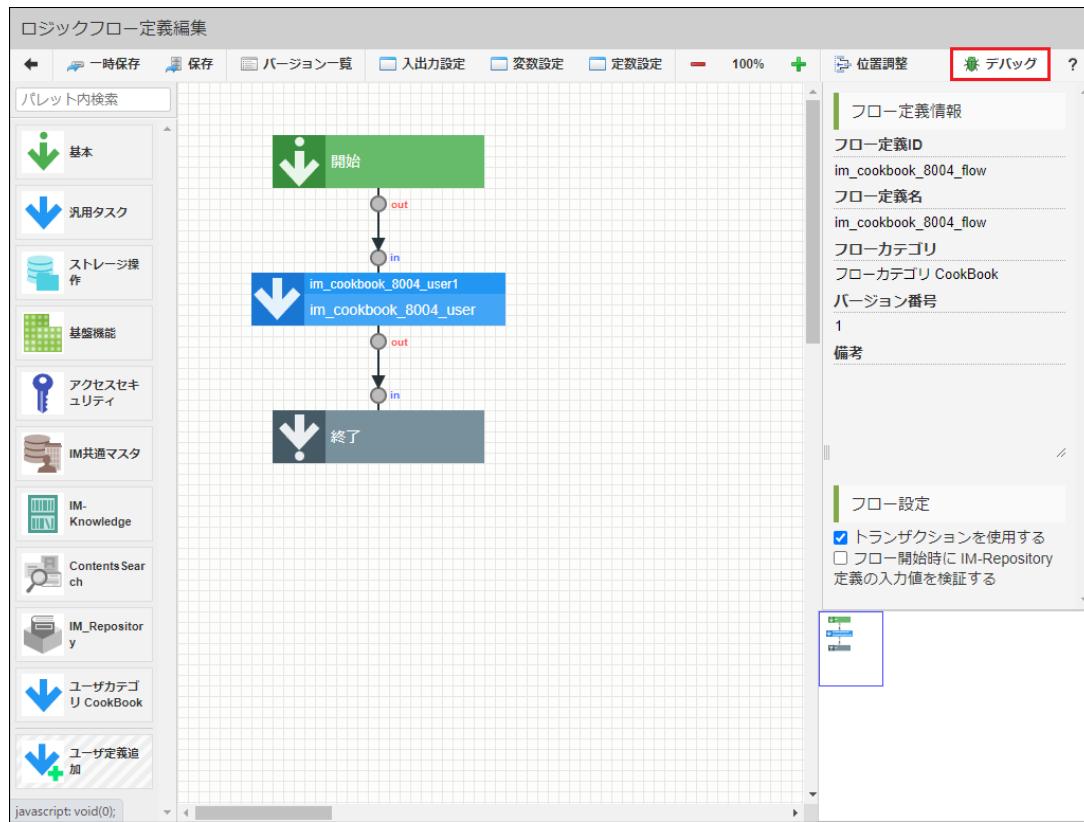
- 「サイトマップ」 - 「LogicDesigner」 - 「フロー定義一覧」をクリックします。



- 「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。

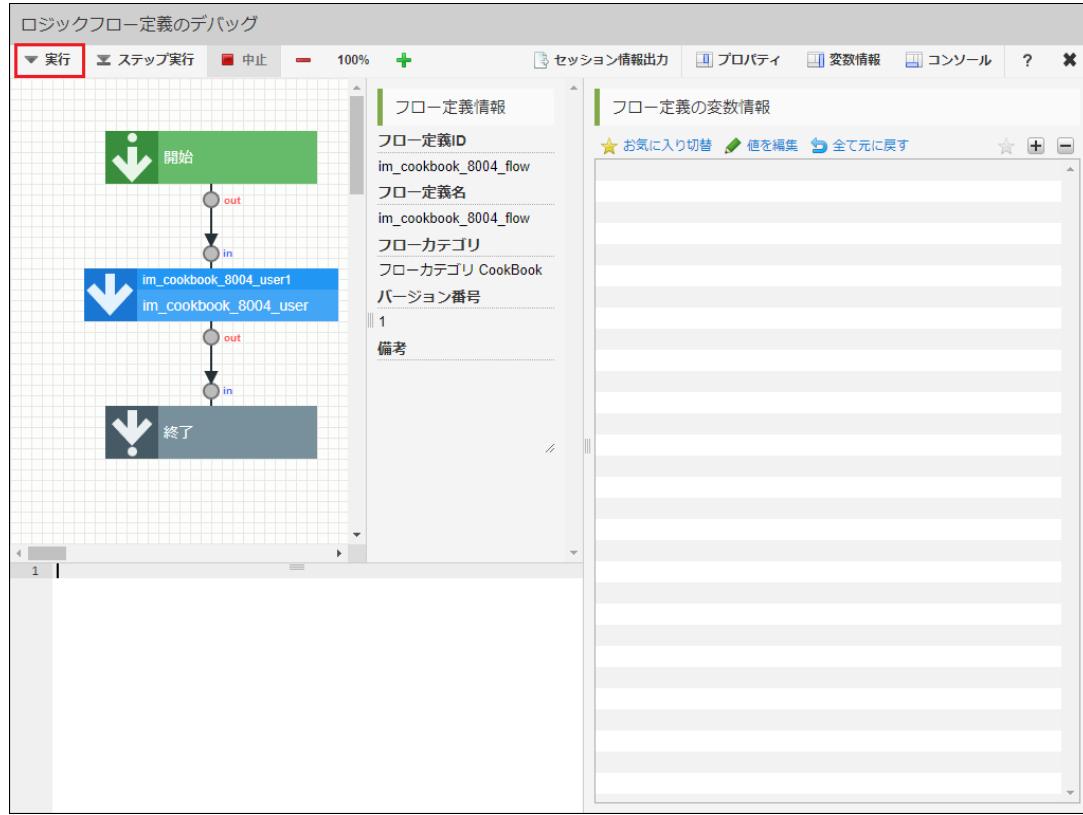


3. 「デバッグ」ボタンをクリックします。



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

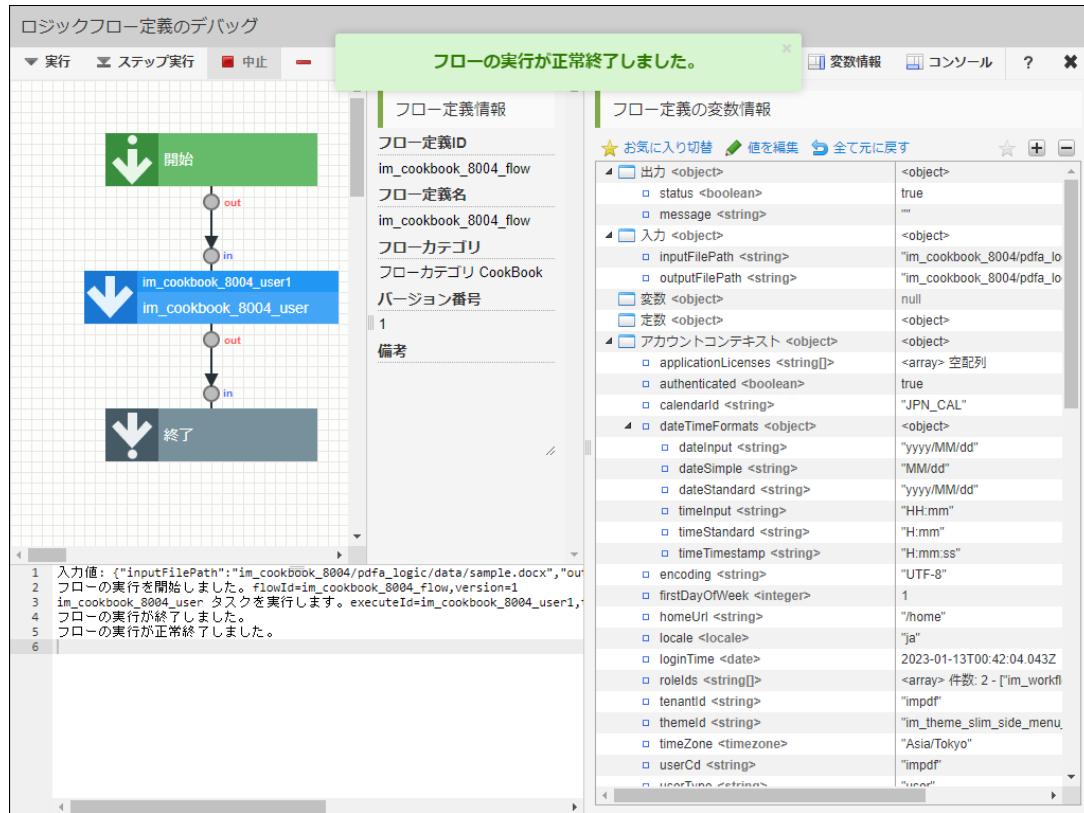
設定例の「inputFilePath」には、パブリックストレージ配下に設置されたPDF変換サンプル用のdocxファイルを指定しています。

3. 「決定」ボタンをクリックします。



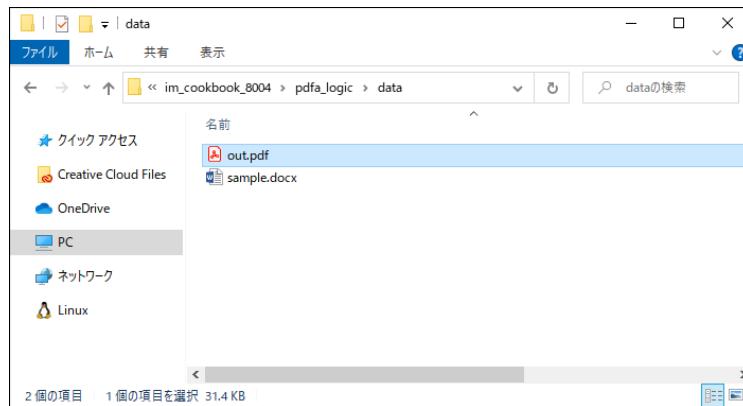
4. デバッグが開始されます。

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。

The screenshot shows the LogicFlow debugger interface with the following details:

- Workflow Diagram:** A simple workflow starting with a green "開始" (Start) node, followed by a blue "im_cookbook_8004_user" node, and ending with a grey "終了" (End) node.
- LogicFlow Information:**
 - フロー定義ID: im_cookbook_8004_flow
 - フロー定義名: im_cookbook_8004_flow
 - フローカテゴリ: CookBook
 - バージョン番号: 1
 - 備考:
- Variable Information:**

| 変数 | 値 |
|------------------------|----------------------|
| out | <object> |
| 入力 | <object> |
| 変数 | null |
| 定数 | <object> |
| アカウントコンテキスト | <object> |
| 外部ユーザコンテキスト | <object> |
| セッション情報 | <object> |
| 処理結果情報 | <object> |
| ユーザコンテキスト | <object> |
| im_cookbook_8004_user1 | <object> |
| status | false |
| message | "コピー元の対象ファイルが開けません。" |
- Log Output:**

```

1 入力値: {"inputFilePath": "im_cookbook_8004/pdfa_logic/data/sample.pdf"} の実行を開始しました。
2 flowId=im_cookbook_8004_flowタスクを実行します。executeId=im_cookbook_8004_user
3 フローの実行が終了しました。
4 フローの実行が正常終了しました。
5 フローの実行が正常終了しました。
6
    
```

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>
└─ outputPath <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。


```

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を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 |
| 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。

lotheCommonTempFiles

一時ファイルを操作するクラスです。

一時ファイルは、「[intra-mart Accel Platform 設定ファイルリファレンス](#)」 - 「[一時ファイルディレクトリ](#)」で設定したディレクトリ配下に作成されます。

コラム

本クラスの使用例については、「[スクリプト](#)」を参照してください。

また、スクリプトの9行目を次のように読み替えてください。

```
const tempFiles = new lotheCommonTempFiles();
```

注意

本クラスは、IM-PDFAutoConverter for Accel Platform 2024 Spring 以降のバージョンで使用可能です。

2023 Autumn 以前のバージョンの場合は、2024 Spring 以降のバージョンにアップデートしてください。

Constructor

`new lotheCommonTempFiles()`

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。

| 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

Exception

以下のエラーメッセージを持つErrorオブジェクト

- 指定された引数extの値が不正な場合：「接尾辞文字列の値が不正です。」

`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 |
| 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

Exception

以下のエラーメッセージを持つErrorオブジェクト

- 指定された引数srcFilePathの値が不正な場合：「コピー元の対象ファイルパスの値が不正です。」
- 指定された引数extの値が不正な場合：「接尾辞文字列の値が不正です。」
- ファイルのコピーに失敗した場合：「ファイルのコピーに失敗しました。」

`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 |

Exception

以下のエラーメッセージを持つErrorオブジェクト

- 指定された引数tempFileの値が不正な場合：「コピー元の対象ファイルの値が不正です。」
- 指定された引数destFilePathの値が不正な場合：「コピー先の対象ファイルパスの値が不正です。」
- ファイルのコピーに失敗した場合：「ファイルのコピーに失敗しました。」

`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。

!

注意

当該メソッドを実行しない場合、一時ファイルディレクトリに一時ファイルが蓄積されていきます。

一時ファイルを残す必要が無い場合は、finallyブロック等を使い、必ず当該メソッドを実行して一時ファイルを削除してください。

Exception

以下のエラーメッセージを持つErrorオブジェクト

- 一時ファイルの削除に失敗した場合：「一時ファイルの削除に失敗しました。」