



OPROARTS Designer ユーザーガイド (PDF/ヘッダー明細/一覧)

Ver.1.1

改訂履歴

Ver.	改訂日	改訂内容
1.0	2019/07/18	新規作成
1.1	2021/12/02	3.4 その他のコンポーネント を追加 ラベルの「動的」にチェックを入れる操作について強調するよう編集

本書に記載されている会社名、製品名、サービス名などは、提供各社の商標、登録商標、商品名です。
なお、本文中に TM マーク、©マークは明記しておりません。

本書の使い方

本資料では、簡単な帳票見本を作成する中で、OPROARTS Designer をご利用するにあたって最低限必要な基本操作手順を理解することを目的としています。

各画面のボタンやコンポーネントの詳細などについては製品ヘルプをご参照ください。

本書の表記

本書では、以下の表記で記載しています。

表記方法	内容
注意	操作上の注意事項について記載しています。
Point	操作上で知っている便利なポイントについて記載しています。
参照	本書における参照先を記載しています。
[]	ボタン名やタブ名、キーボードのキーなどの表記で使用します。
「 」	システム名、メニュー名、画面名、項目名、参照先などの表記で使用します。

目次

1. はじめに	5
1.1 ヘッダー明細型	5
1.2 一覧型.....	7
2. 全体の流れ	8
3. テンプレートのデザイン	9
3.1 テンプレートの新規作成	10
3.2 基本操作の紹介.....	11
3.3 実際に作る	13
3.4 その他のコンポーネント	16
4. 項目のマッピング	18
5. 出力方法	24
5.1 HTML ページからの出力	24
5.2 API からの出力	26

1. はじめに

ここでは OPROARTS Designer で出力する帳票を作成するにあたって重要となる CSV データと帳票の対応、また「ヘッダー明細型」「一覧型」の概念について説明します。

1.1 ヘッダー明細型

ヘッダー(またはフッター)と明細のある帳票を 2 つのグループに分けて考えます。ヘッダーやフッター部分を「非明細部」と呼びます。それ以外の部分を「明細部」と呼びます。この非明細部と明細部に分けることのできる帳票を、「ヘッダー明細型」と定義しています。

御請求書

発行日: 2017/09/01
請求番号: 0001

日本オプロ株式会社 御中
東京都港区芝3-43-15
芝信三田ビル

デモ商事株式会社
〇〇県〇〇市
〇〇1-1111 〇〇ビルズ
11-1111-1111 / 11-1111-1112

下記の通りご請求申し上げます。

ご請求額 ¥756,000

請求締日 20日
支払期限 2017/9/20

振込先 〇〇銀行 三田支店 123456

製品名	単価	数量	金額
商品1	100,000	1	100,000
商品2	100,000	1	100,000
商品3	100,000	1	100,000
商品4	100,000	1	100,000
商品5	100,000	1	100,000
商品6	100,000	1	100,000
商品7	100,000	1	100,000

合計 ¥700,000
消費税 ¥56,000
総合計(税込) ¥756,000

備考

1ページ

・非明細部

取引先や住所などのように、帳票上に一度しか表示されない項目が集まる領域。ヘッダーやフッター

・明細部

商品名などといった、帳票によって行数が変わる領域。「繰り返し領域」ともいう。

OPROARTS Designer からこのヘッダー明細型帳票を作る場合

非明細部: 1 行の CSV データ

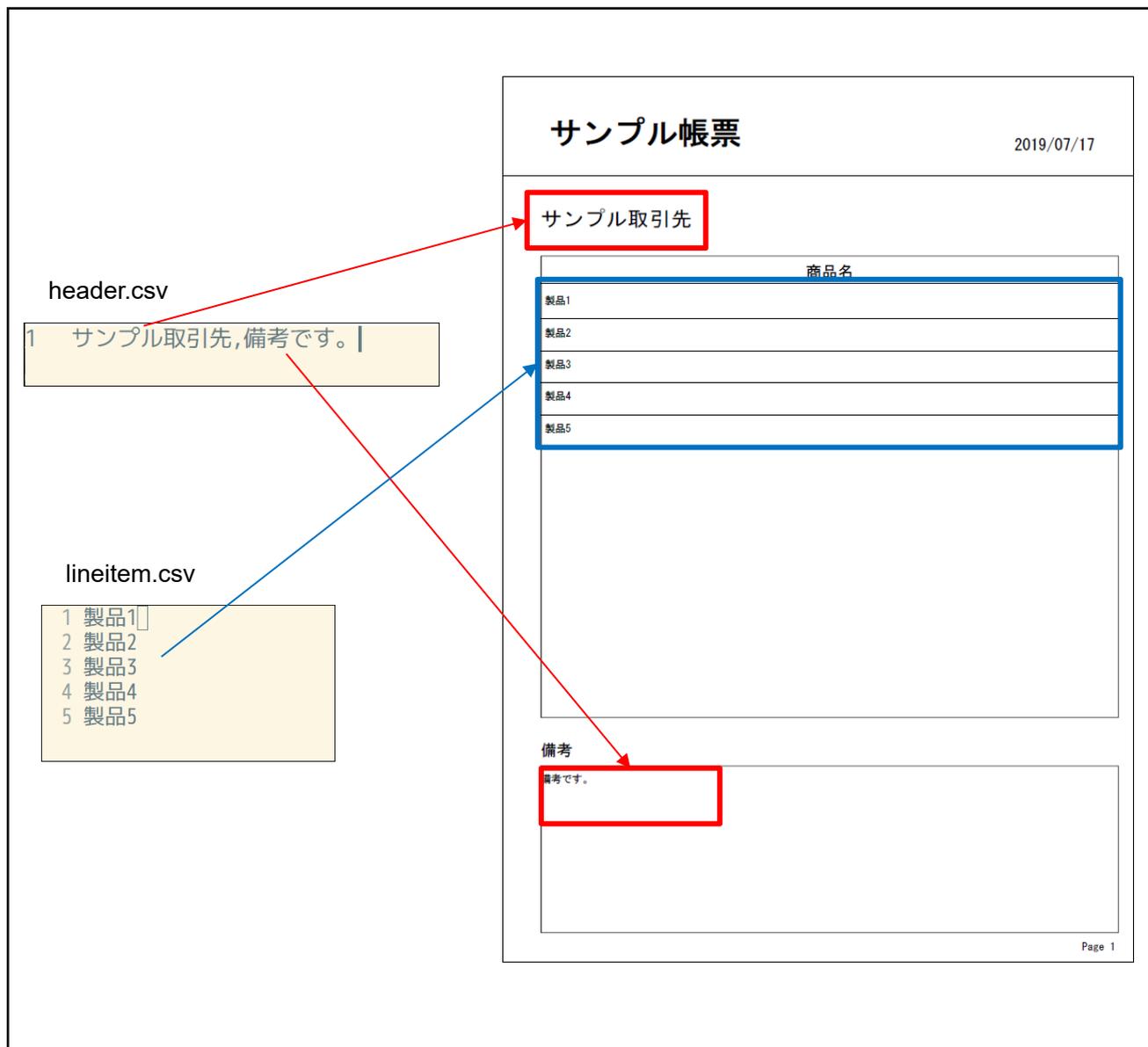
明細部: 複数行の CSV データ

を用いてそれぞれの部分を埋めることになります。

(例)

次の CSV データからは、以下のような帳票が出力されます。

(赤枠で囲まれた部分が header.csv の情報、青枠で囲まれた部分が lineitem.csv の情報です)



1.2 一覧型

ヘッダー明細型とは異なり、一覧型は帳票のヘッダー/フッター部分に動的な値を表示しません。

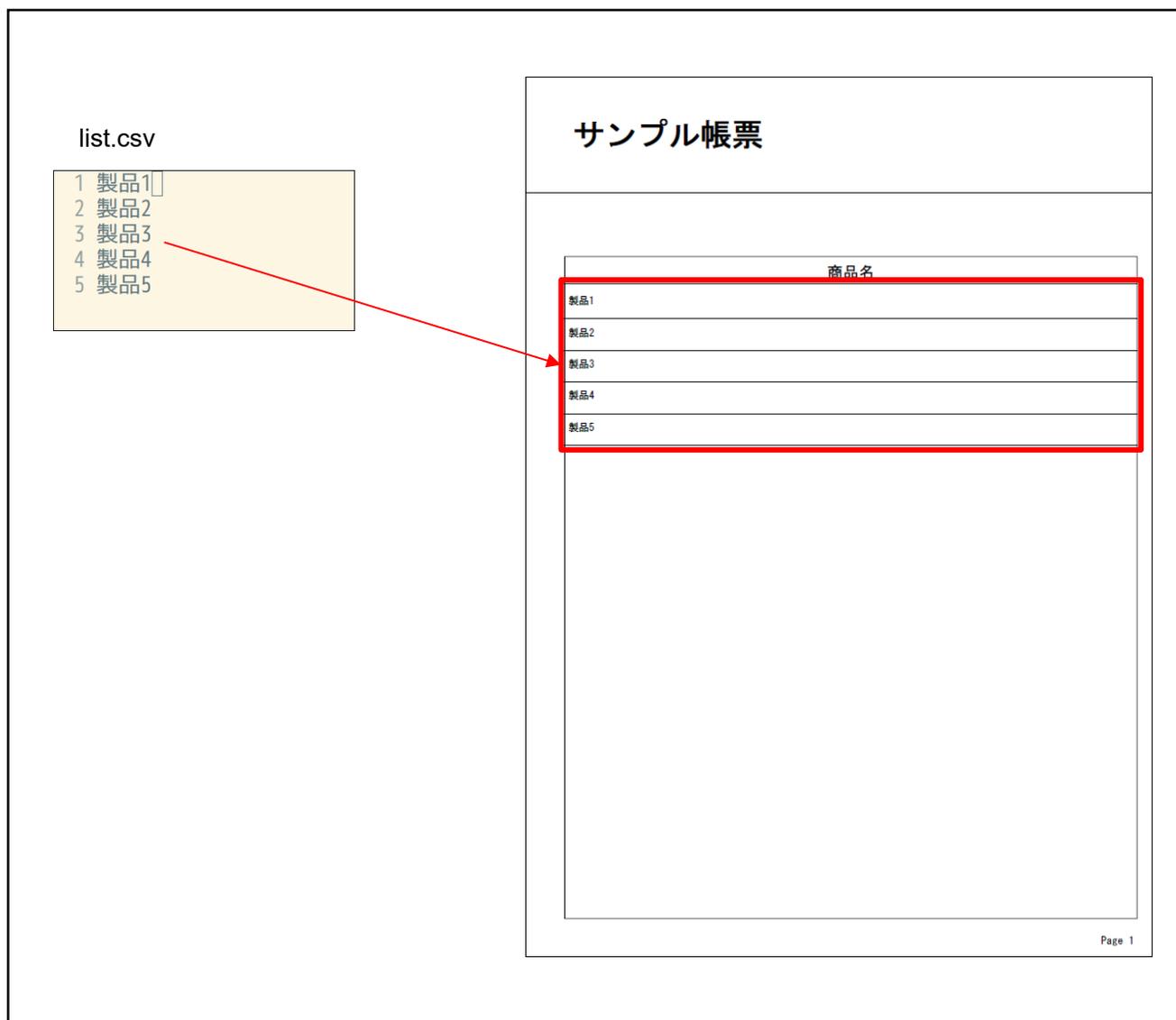
明細部分のみのヘッダー明細型帳票とも言えます。

OPROARTS Designer から一覧型帳票を作る場合、複数行の CSV データを用いて帳票が出力されます。

(例)

次の CSV データからは、以下のような帳票が出力されます。

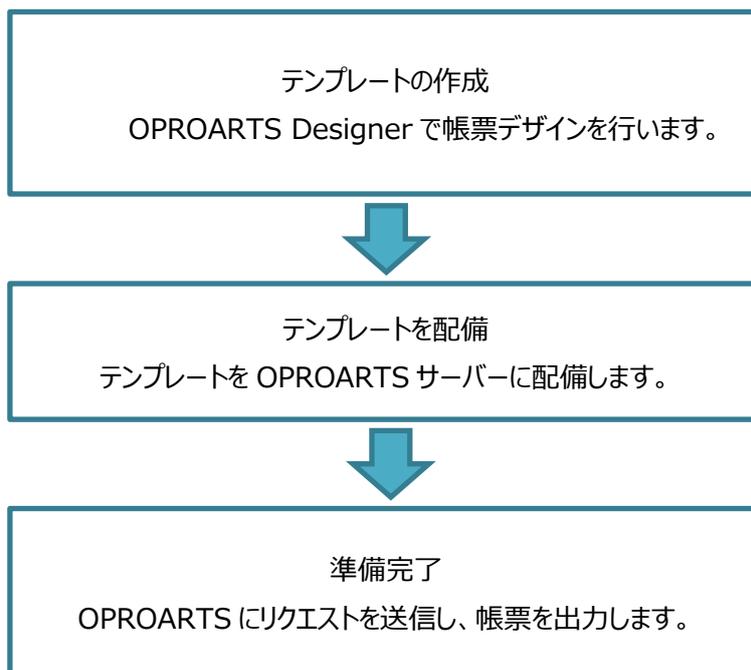
(赤枠で囲まれた部分が list.csv の情報です)



※一覧型帳票はヘッダー明細型帳票からヘッダー/フッター部分の動的項目を除いたものなので、本ユーザーガイドではヘッダー明細型の説明のみとします。一覧型帳票を作成する場合は、ヘッダー明細型の明細部分の説明をご参照ください。

2. 全体の流れ

OPROARTS Designer では、以下のような流れで帳票出力が可能になります。



3. テンプレートのデザイン

出力したい帳票の見た目を OPROARTS Designer でデザインします。

本ユーザーガイドでは、以下のような帳票を出力するためのテンプレートを作成します。

サンプル帳票	① 2019/07/17
② サンプル取引先	
商品名	
製品1	
製品2	
製品3	
製品4	
製品5	
	③
備考	
備考です。	④
	⑤
	Page 1

番号	内容
①	帳票出力した日付
②	取引先名
③	商品名
④	備考
⑤	ページの番号

番号を振っていない部分の文言は固定文言です。

任意の文言を配置、もしくは何も配置しなくても問題ありません。

※※本テンプレートは帳票テンプレートの構造を理解するため、敢えて非常にシンプルな構成にしています。
このマニュアルで使用しない各帳票コンポーネントは以下の章で紹介していますので、必要に応じてご参照ください。
[その他のコンポーネント](#)

3.1 テンプレートの新規作成

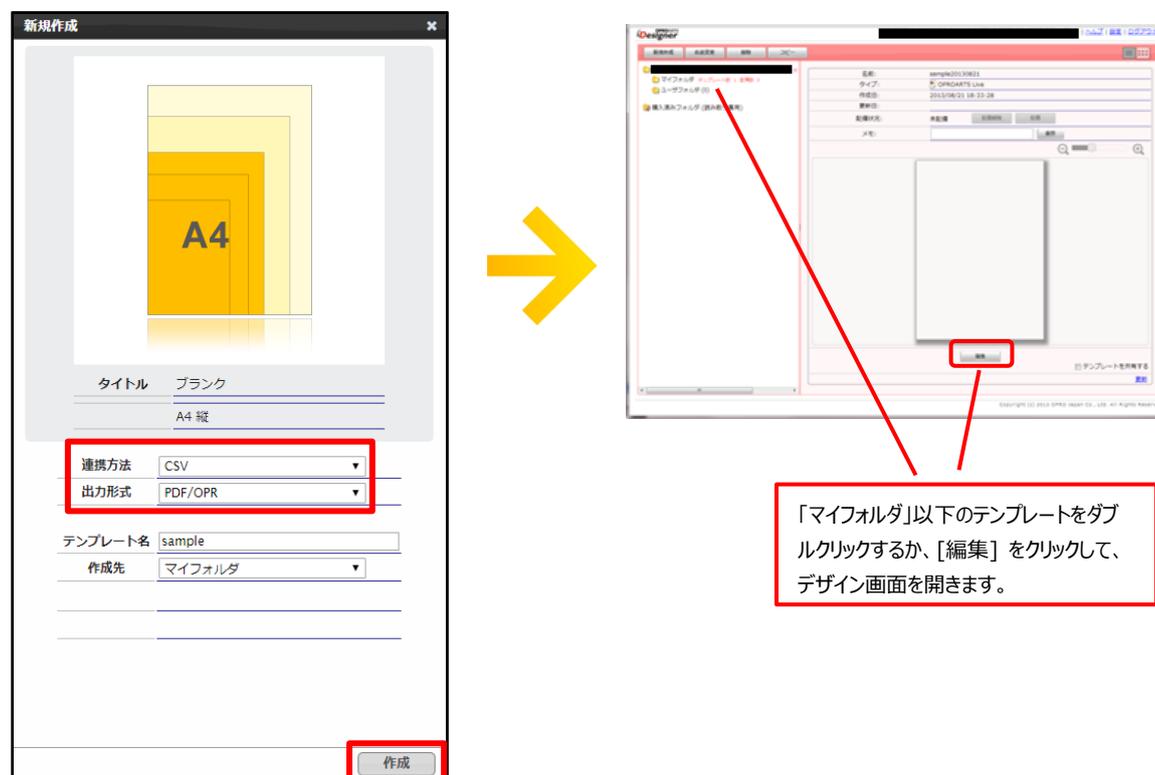
OPROARTS Designer にログインし、左上の[新規作成]をクリックします。



ここでは、[ブランク] タブで A4 縦を選択します。
連携方法「CSV」、出力形式「PDF/OPR」を選択してください。

以下のルールに従って任意のテンプレート名を入力し、[作成] をクリックしてください。

- ・使用できる文字は、英数字とアンダーバー
- ・先頭の文字はアルファベットである
- ・最後の文字がアンダースコアでない
- ・アンダーバーが 2 個以上連続していない



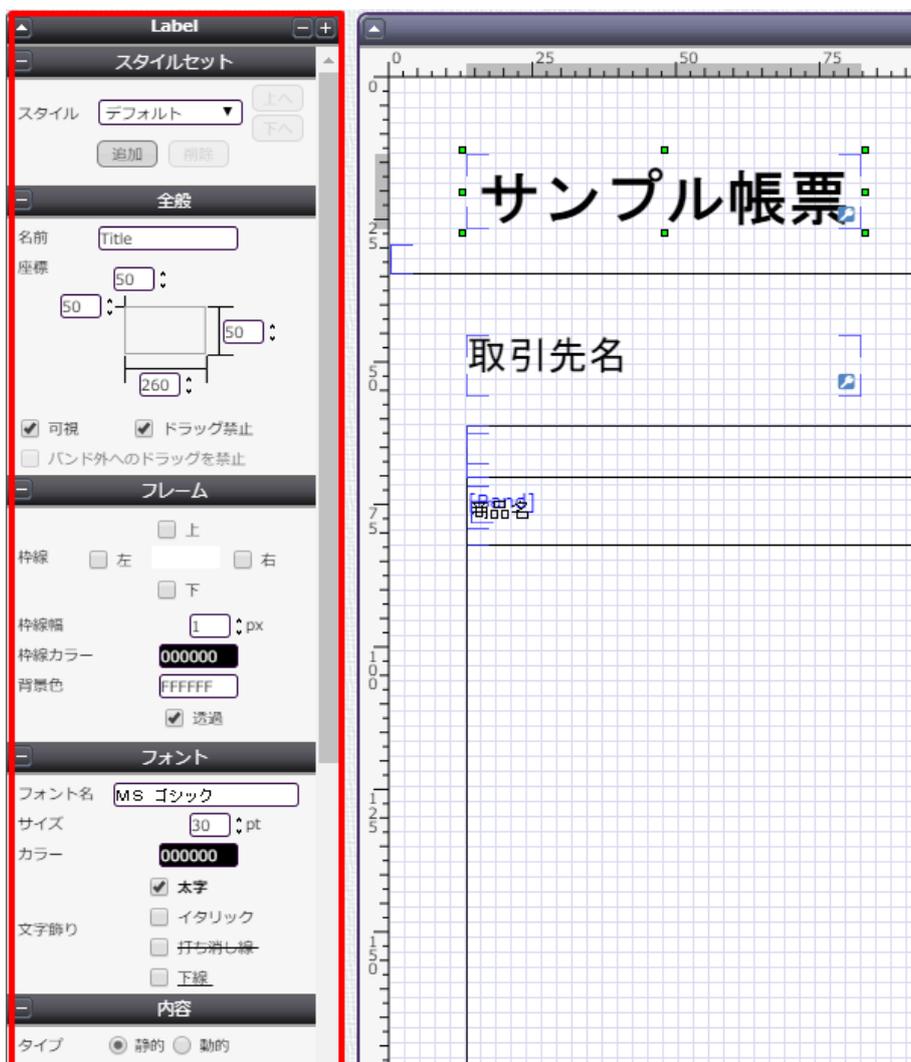
※ViewFramer、D3Worker と連携する場合は、連携方法で「ViewFramer/D3Worker」を選択してください。その後の設定方法は、「CSV」を選択した場合と変わりません。

3.2 基本操作の紹介

左上にあるコンポーネントのリストから配置したいコンポーネントを選択してからレイアウト上をクリックすることで選択したコンポーネントを配置できます。



また、画面左側に現れるプロパティを変更することで現在選択されているコンポーネントの詳細を設定できます。



その他に、知っておくと便利な基本のショートカットキーをご紹介します。

ショートカットキー	動作
[Ctrl] + [矢印 (↑↓←→)]	コンポーネントを 1px ずつ移動
[Ctrl] + [Shift] + [矢 印 (↑↓←→)]	コンポーネントをグリッドの間隔単位で移動
[Shift] + [矢印 (↑↓←→)]	コンポーネントのサイズを変更
[Ctrl] + 選択	コンポーネントを複数選択する(※)
Ctrl+Z	取消
Ctrl+Y	やり直し
Ctrl+X	切り取り
Ctrl+C	コピー
Ctrl+V	貼り付け
Ctrl+A	全て選択(※)
Esc	選択解除
Delete	削除

(※)以下のコンポーネントを組み合わせ、同時に選択することはできません。

- ・他のコンポーネントを内包するコンポーネント
- ・他のコンポーネントに内包されたコンポーネント

3.3 実際に作る

ここでは、実際にテンプレートをデザインしながら Designer の使い方を学んでいきます。

1. 罫線や枠線を引く

Line コンポーネントや Rectangle コンポーネントを用いて仕切り線を引きます。

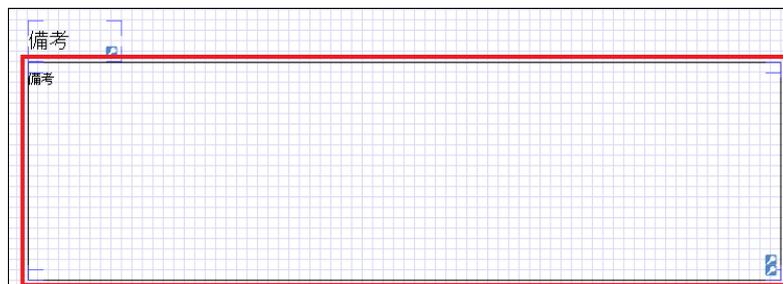


実際にレイアウトに配置するとこのようになります。

罫線



枠線

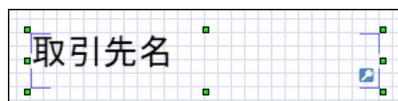


2. ラベルを配置、動的項目にチェック

文字列を表示したい部分には、Label コンポーネントを配置します。



実際にレイアウトに配置するとこのようになります。



ラベルの詳細内容は以下のように設定します。

※ここでは取引先名を表示するラベルを一例として取り上げています。配置したそれぞれのラベルコンポーネントに合わせて設定してください。



- **タイプ**
このラベルは取引先名の値を表示するので、「動的」に設定します。
固定文言の場合、「静的」を指定します。
- **アライン**
水平位置、垂直位置ともに、ラベルのどの位置に文字が表示されるのか指定します。
- **スタイル**
文字数が多くても全体が表示されるよう、「縮小して全体を表示」にチェックを入れます。
- **行間隔**
行の折り返しはしないので、0のままにしておきます。

3. データセットテーブルを設置

DataSetTable コンポーネントを配置して、明細部分のレイアウトを作成します。



Dataset Table コンポーネントの中に、Band というコンポーネントがあります。



この中に配置されたコンポーネントは、明細オブジェクトレコードの数だけ繰り返し表示されます。

ここでは、商品名の動的項目ラベルと罫線を配置します。

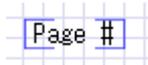


4. ページ番号を挿入

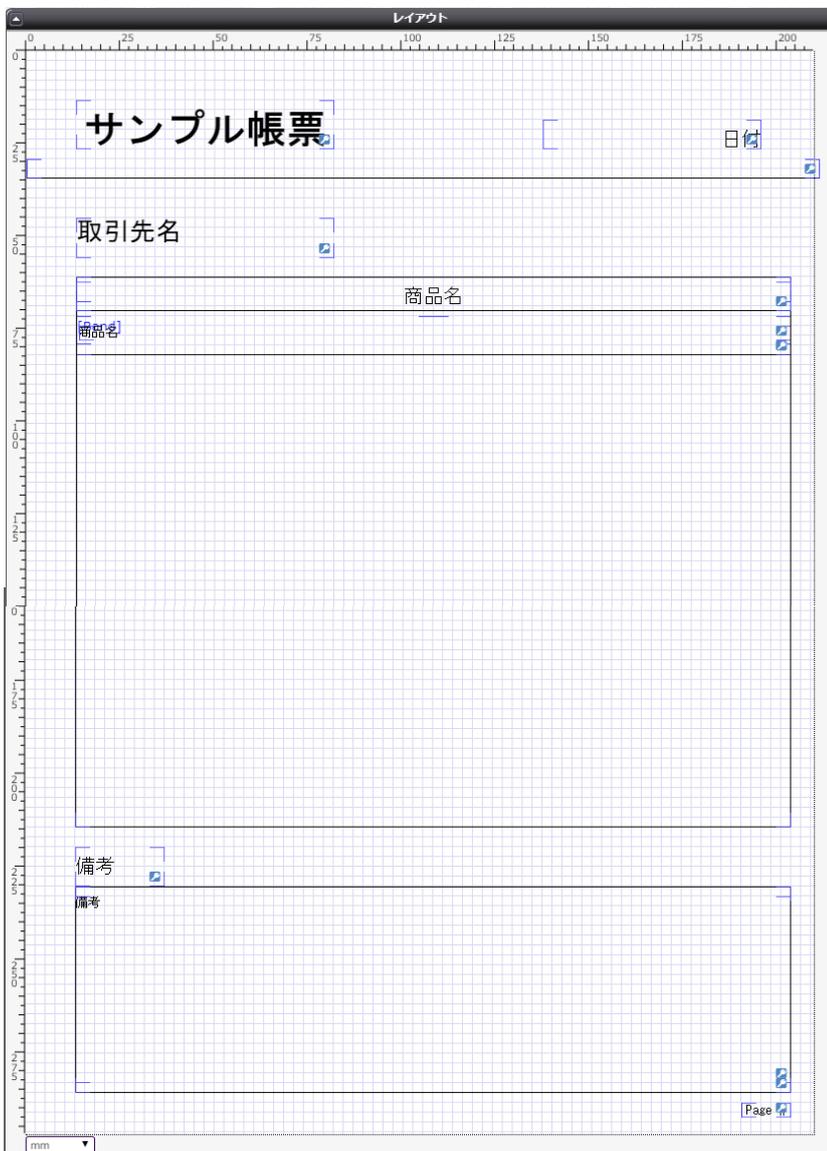
レイアウト右下にページ番号を配置します。



実際にレイアウトに配置するとこのようになります。



完成したレイアウトは以下です。



デザインが出来上がったら、保存ボタン(📁)をクリックして、矢印ボタン(➡)でマッピングへ進みます。

3.4 その他のコンポーネント

ここでは、3.3 までに紹介したコンポーネントを含む全ての使用可能なコンポーネントについてリストアップして説明します。



番号	名称	機能
1	Line	レイアウトにラインを描画します。
2	Rectangle	レイアウトに正方形・長方形を描画します。
3	Circle	レイアウトに円・楕円を描画します。
4	Label	レイアウトにテキストを表示します。テキストは、静的に指定(コンポーネントに直接記述)するだけでなく、ドキュメント生成時に データソースの値を動的に適用することもできます。
5	Image	レイアウトに画像を挿入します。画像は、 ・お使いのブラウザよりアップロードされたもの ・外部サーバから URL で取得できるものを設定できます。
6	Barcode	レイアウトにバーコード画像を挿入します。Barcode の元となるデータは、静的に指定(コンポーネントに直接記述)するだけでなく、ドキュメント生成時にデータソースの値を動的に適用することもできます。
7	Page Number	レイアウトにページ番号を表示します。コンポーネント毎に、ページ番号の採番・表示ルールを設定することができます。
8	Total Page Number	レイアウトに総ページ数を表示します。ただし、OPR 形式での出力のみ有効なコンポーネントです。
9	Dataset Table	データソースのデータとリンクし、バンドを使用してレイアウトにテーブルを作成します。Dataset Table コンポーネントは、内部に Band コンポーネントを持ちます。
1 0	Horizontal Dataset Table	データによって明細行が右方向に伸びる点を除き Dataset Table コンポーネントと同じです。
1 1	Container	内部に複数 DatasetTable を配置可能なコンポーネントです。Container 内に二つの DatasetTable を配置して出力すると、最初の DatasetTable にテーブルの内容すべてを表示した後、二番目の DatasetTable にテーブルの内容を表示します。

Dataset Table 内に設定できるコンポーネント

名称	機能
Band	Dataset Table コンポーネントに繰り返し領域（明細行）を描画します。Dataset Table 配置時に内包されています。
Report Header	Dataset Table コンポーネントにレポートヘッダ領域を描画します。レポートヘッダは、繰り返し領域（明細行）の前に1度だけ描画されるヘッダです。
Report Footer	Report Footer コンポーネントは、Dataset Table コンポーネントにレポートフッタ領域を描画します。レポートフッタは、繰り返し領域（明細行）の後に1度だけ描画されるフッタです。
Column Header	Column Header コンポーネントは、Dataset Table コンポーネントにカラムヘッダを描画します。カラムヘッダは Dataset Table の明細行の直前に描画されるヘッダです。見出しの役割をします。
Group Header	Group Header コンポーネントは、Dataset Table コンポーネントにグループヘッダを描画します。グループヘッダは、Dataset Table のデータを任意項目でグループ化し、その際のヘッダを描画します。
Group Footer	Group Footer コンポーネントは、Dataset Table コンポーネントにグループフッタを描画します。グループフッタは、Dataset Table のデータを任意項目でグループ化し、その際のフッタを描画します。

4. 項目のマッピング

デザインしたテンプレートに CSV フィールドをマッピングします。

1. 画面左上の「コンポーネントグループ」内、「ページ」と「DatasetTable1」の両方について、CSV 定義を追加します。



「ページ」にチェックを入れ、[CSV 定義を追加]をクリックします。



「DatasetTable1」についても同様にします。



2. それぞれのコンポーネントグループについて、CSVデータのフィールドを決定します。
[フィールドを定義]をクリックし、[追加]ボタンでフィールドを追加します。

ページ

1. ページ 向け CSV

フィールドを定義 マッピング
↑ ↓ 削除

CSV 定義

区切り記号付き 固定長
フィールド区切り記号: ,
文字列の引用符: "
 先頭行を無視する
 データ出力条件

フィールド名	データタイプ	形式
AccountName	text	
Remarks	text	

追加 削除 上へ 下へ 形式の読み込み... サンプルCSV

Dataset Table1

2. DatasetTable1 向け CSV

フィールドを定義 マッピング
↑ ↓ 削除

CSV 定義

区切り記号付き 固定長
フィールド区切り記号: ,
文字列の引用符: "
 先頭行を無視する
 データ出力条件

フィールド名	データタイプ	形式
ProductName	text	

追加 削除 上へ 下へ 形式の読み込み... サンプルCSV

※「先頭行を無視する」のチェックを必ず外してください。外さないと、CSVデータが正しく連携されません。

3. フィールドを追加したら[マッピング]をクリックし、それぞれのフィールドについてテンプレートのどの動的項目と対応するかを決定します。

設定したいフィールドの行をクリックして、画面左側の「CSV フィールド」から対応させたい項目を「データ」欄にドラッグ&ドロップします。

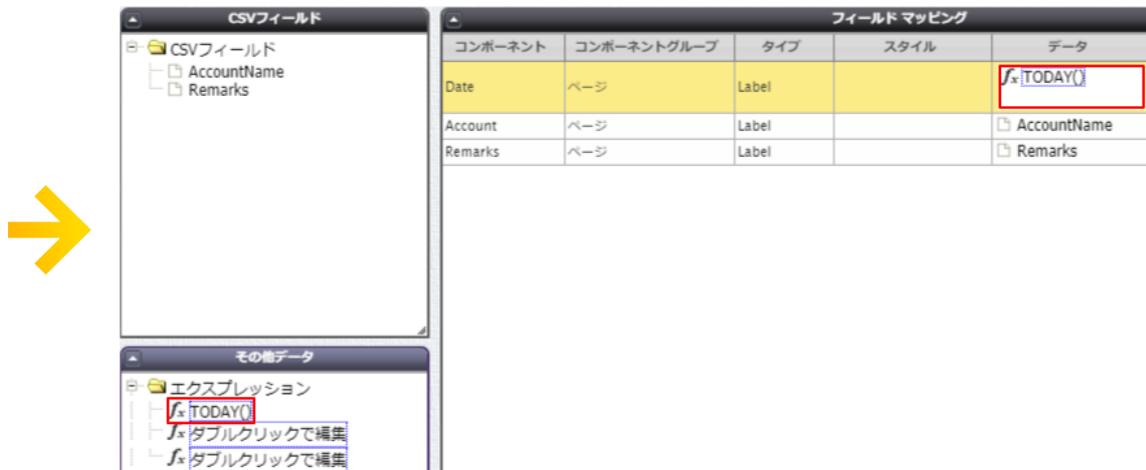
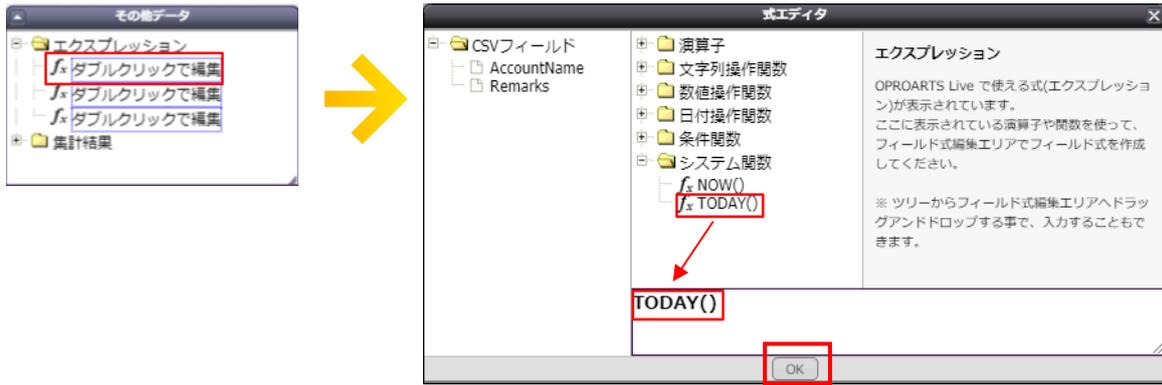
ページ



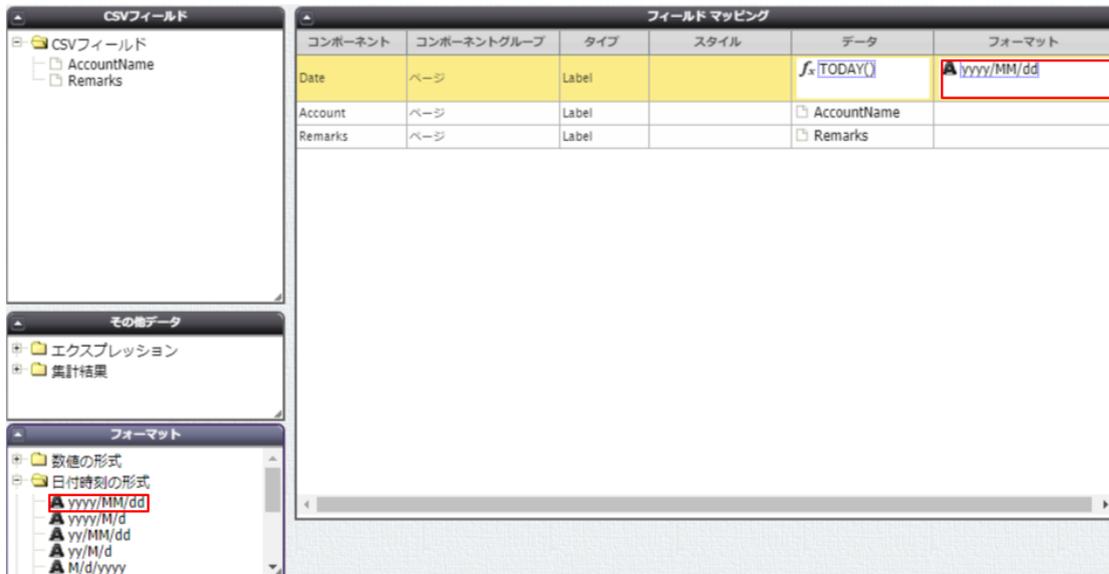
The image shows two side-by-side windows. The left window, titled 'CSVフィールド', shows a tree view with 'AccountName' and 'Remarks' under 'CSVフィールド'. The right window, titled 'フィールド マッピング', shows a table with columns: 'コンポーネント', 'コンポーネントグループ', 'タイプ', 'スタイル', and 'データ'. The 'Date' row has 'Date' in the first column, 'ページ' in the second, 'Label' in the third, and 'f TODAY()' in the fifth. The 'Account' row has 'Account' in the first column, 'ページ' in the second, 'Label' in the third, and 'AccountName' in the fifth. The 'Remarks' row has 'Remarks' in the first column, 'ページ' in the second, 'Label' in the third, and 'Remarks' in the fifth. The 'AccountName' and 'Remarks' cells in the 'データ' column are highlighted with red rectangles.

コンポーネント	コンポーネントグループ	タイプ	スタイル	データ
Date	ページ	Label		f TODAY()
Account	ページ	Label		AccountName
Remarks	ページ	Label		Remarks

フィールドには組み込みの関数を使用することもできます。ここでは Date 項目にシステム関数の TODAY()を使用しています。



表示の際のフォーマットを指定したい場合は、画面左側の「フォーマット」からお好みの項目をフィールド行の「フォーマット」欄にドラッグ&ドロップします。ここでは、日付の表示を「yyyy/MM/dd(年四桁、月二桁、日二桁)」に指定しています。

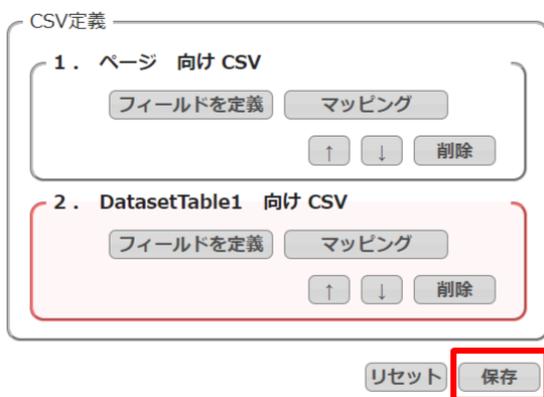


4. 明細部分についても、同様にマッピングします。

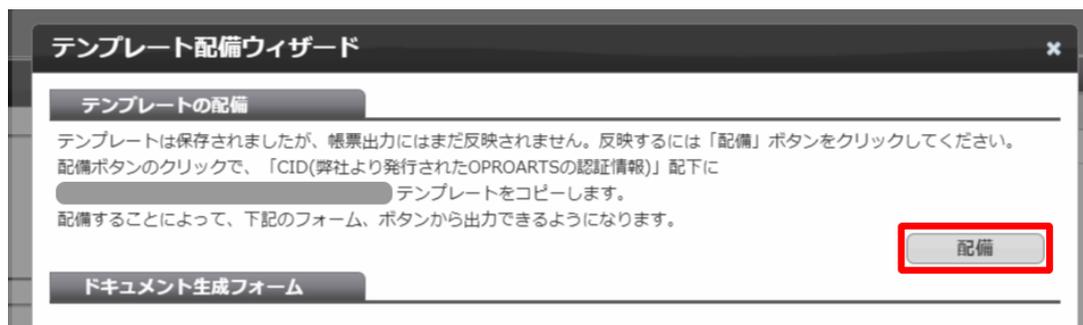
Dataset Table1



5. マッピングが終わったら、[保存]をクリックします。



6. テンプレート配備ウィザードにて[配備]をクリックしてテンプレートを帳票出力に使用できる状態にします。



※テンプレートの編集をした際も、必ず[配備]をクリックしてください。配備をしないと変更点が帳票出力に反映されません。

5. 出力方法

OPROARTS にリクエストを送信して、帳票を出力する方法を説明します。

5.1 HTML ページからの出力

1. テンプレート配備ウィザードの「ドキュメント生成フォーム」の各項目を設定し、[HTML 取得]ボタンをクリックします。

ドキュメント生成フォーム

このフォームを使ってOPROARTSサーバへリクエストを行います。パラメータを編集してリクエストを行い、ドキュメントをプレビューしてください。OPROARTSへのリクエスト形式、OPROARTSからのレスポンスについては、以下を参照ください。
OPROARTSへのリクエスト形式、OPROARTSからのレスポンスについて

OPROARTS認証情報			
パラメータ	意味	値	備考
OID	顧客ID		※ 必須パラメータ
UID	ユーザID		※ 必須パラメータ
LPW	ユーザパスワード	※ 必須パラメータ

基本			
パラメータ	意味	値	備考
OP	操作指示	dip: pdfドキュメントのダウンロード	※ 必須パラメータ
FE	PDFへのフォントの埋め込み	<input checked="" type="checkbox"/>	

テンプレート			
パラメータ	意味	値	備考
TP1	テンプレート名	live_simple_quotation_for_manual_vf	※ 必須パラメータ(テンプレート名に接頭辞live_を付与してください)
TD1_1	テキストデータセット	Choose File header.csv リセット	
TD1_2	テキストデータセット	Choose File lineitem.csv リセット	

完了待ち自動更新			
パラメータ	意味	値	備考
RF	リフレッシュ間隔	:1	

送信 **HTML取得**

OPROARTSでは、Java APIも準備されています。「APIサンプル」ボタンのクリックで、ドキュメント生成フォームのパラメータに基づいたソースコードを生成します。OPROARTS APIの詳細は、以下を参照ください。
OPROARTS API ドキュメント
OPROARTS API本体のダウンロード

APIサンプル

閉じる

2. ダウンロードされた HTML をブラウザで開きます。
CID/UID/UPW を入力し、[送信]をクリックすると帳票出力ができます。

OPROARTS パラメータ送信サンプル		
OPROARTSへのリクエストを実行するフォームのサンプルです。各パラメータの意味については、 OPROARTS readme をご覧ください。		
OPROARTS認証情報		
パラメータ	値	役割
CID	<input type="text"/>	CIDを入力してください。
UID	<input type="text"/>	UIDを入力してください。
UPW	<input type="password"/>	UPWを入力してください。
基本		
パラメータ	値	役割
OP	<input type="text" value="dip"/>	dip:PDFダウンロード, dlo:OPRダウンロード, mlp:PDF添付メール, mlp:OPR添付メール, dp:直接印刷
FE	<input checked="" type="checkbox"/>	PDFへのフォントの埋め込み (フォントを埋め込むには、サーバー側で特別な設定が必要です。通常は「false」を指定してください。)
テンプレート		
パラメータ	値	役割
TP1	<input type="text" value="live_simple_quotation_for_manual_vf"/>	テンプレート名
TD1_1	<input type="text" value="Choose File header.csv"/>	TP1で使用するCSVデータを選択します。
TD1_2	<input type="text" value="Choose File lineitem.csv"/>	TP1で使用するCSVデータを選択します。
完了待ち自動更新		
パラメータ	値	役割
RF	<input type="text" value="-1"/>	更新間隔 (秒)
<input type="button" value="Submit"/>		

5.2 API からの出力

OPROARTS へのリクエストは、API からも可能です。

1. テンプレート配備ウィザードの「ドキュメント生成フォーム」の各項目を設定し、[API サンプル]ボタンをクリックします。

ドキュメント生成フォーム

このフォームを使って OPROARTS サーバへリクエストを行います。パラメータを編集してリクエストを行い、ドキュメントをプレビューしてください。OPROARTS へのリクエスト形式、OPROARTS からのレスポンスについては、以下を参照ください。

OPROARTS へのリクエスト形式、OPROARTS からのレスポンスについて

パラメータ	意味	値	備考
CD	顧客ID	CXA7PJ2YDC48SR	※ 必須パラメータ
UID	ユーザID	tsdemo	※ 必須パラメータ
UPW	ユーザパスワード	※ 必須パラメータ

基本

パラメータ	意味	値	備考
OP	操作指示	dip.pdfドキュメントのダウンロード	※ 必須パラメータ
FE	PDFへのフォントの埋め込み	<input type="checkbox"/>	

テンプレート

パラメータ	意味	値	備考
TP1	テンプレート名	live_simple_quotation_for_manual_vf	※ 必須パラメータ(テンプレート名に接続辞(ue)を付与してください)
TD1.1	テキストデータセット	Choose File header.csv リセット	
TD1.2	テキストデータセット	Choose File lineitem.csv リセット	

完了待ち自動更新

パラメータ	意味	値	備考
BE	リフレッシュ間隔	1	

送信 HTML取得

OPROARTS では、Java API も準備されています。「API サンプル」ボタンのクリックで、ドキュメント生成フォームのパラメータに基づいたソースコードを生成します。OPROARTS API の詳細は、以下を参照ください。

OPROARTS API ドキュメント

OPROARTS API 本体のダウンロード

API サンプル

閉じる

2. API リクエスト用のサンプル Java コードが表示されます。

こちらのソースコードを用いて帳票出力が可能です。

```
// これは、OPROARTSサーバで生成された OPROARTS API の利用サンプルです。
// OPROARTS API の情報は、https://www2.opro.net/jp_arts/support.html からご覧になれます
package oproartsapi.sample;
```

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.net.MalformedURLException;
```

```
// 以下のクラスは、OPROARTSAPI.jar に含まれています。
// OPROARTSAPI.jar は、https://www2.opro.net/jp_arts/support.html より入手できます
import net.opro.product.odcaw.oproarts.api.OPROARTSHandler;
import net.opro.product.odcaw.oproarts.api.OPROARTSParameter;
import net.opro.product.odcaw.oproarts.api.TemplateParameter;
import net.opro.product.odcaw.oproarts.api.OPROARTSAPIConnectTimeoutException;
import net.opro.product.odcaw.oproarts.api.OPROARTSAPIConnectionException;
import net.opro.product.odcaw.oproarts.api.OPROARTSAPIException;
import net.opro.product.odcaw.oproarts.api.OPROARTSAPIReadTimeoutException;
import net.opro.product.odcaw.oproarts.api.OPROARTSAPISavingException;
import net.opro.product.odcaw.oproarts.api.OPROARTSAPIConcurrentRequestException;
import net.opro.product.odcaw.oproarts.api.OPROARTSAPIIllegalRequestException;
import net.opro.product.odcaw.oproarts.api.OPROARTSRequestTooLargeException;
import net.opro.product.odcaw.oproarts.api.OPROARTSRestrictedException;
import net.opro.product.odcaw.oproarts.api.OPROARTSUnauthorizedException;
import net.opro.product.odcaw.oproarts.api.OPROARTSException;
```

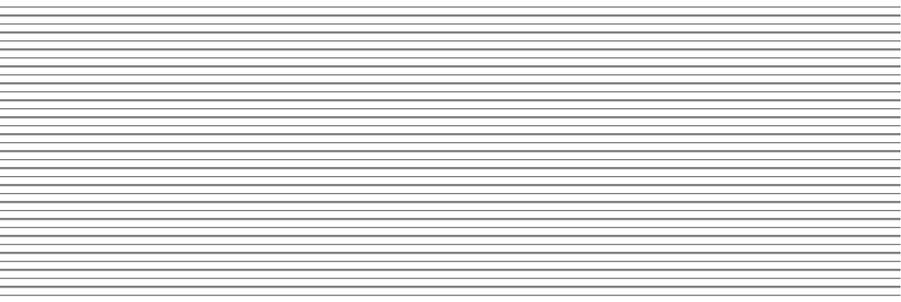
```
public class OPROARTSAPISample {
    public static void main(String[] args) throws FileNotFoundException {
        // OPROARTS 認証情報を指定し、パラメータ定義を作成します
```

詳細は、以下の URL をご参照ください。

[OPROARTS へのリクエスト形式、OPROARTS からのレスポンスについて](#)

[OPROARTS API ドキュメント \(日本語\)](#)

[OPROARTS API 本体のダウンロード](#)



OPROARTS Designer
ユーザーガイド
(PDF/ヘッダー明細/一覧)

