

**Edition Flex** ▲ **CMS** ▲

ECサイトアプリケーション連携の標準的実装方法

## 1.はじめに

本資料は、一般的なECサイトアプリケーションにおいて、商品注文時にEdition Flexの編集機能と連携し、編集結果データを注文データと紐付けして管理するための標準的な実装方法について解説しております。

\* 特定のECサイトアプリケーションとの連携方法を記載したものではありません。

\* Edition Flex APIの詳細については、ホームページに掲載の「[Flex API仕様書](#)」を参照下さい。

## 2. サイトイメージ1 - エンドユーザ画面

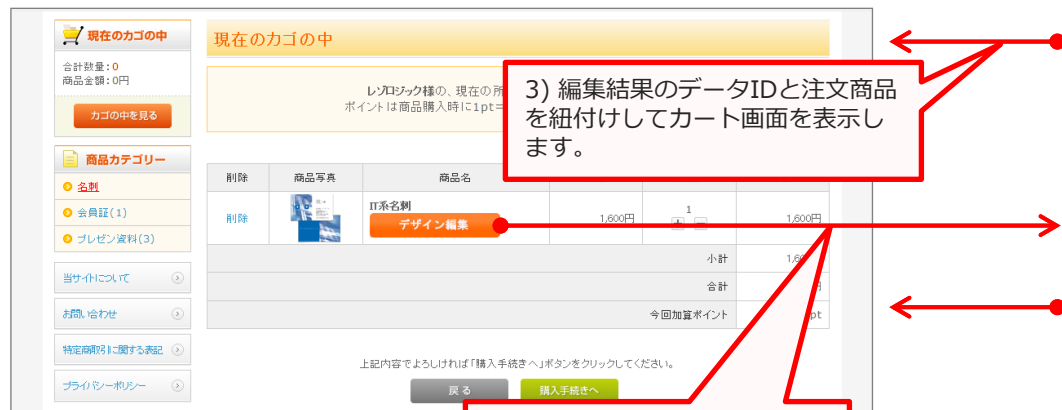
下記のようなECサイトとEdition Flexの連携を実装する際の方法について説明します。

エンドユーザ画面では、商品を選択すると、Edition Flexでデザインを編集して注文できるようにします。

ECサイトの画面はサンプルです。  
実際に稼働しているものではありません。



2) Edition Flex編集画面またはウィザード編集画面で編集し、保存します。



カートの[デザイン編集]ボタンをクリックすると再編集します。

### 3. サイトイメージ2 - マスタメンテナンス画面

マスタメンテナンス画面では、商品マスタで登録される商品情報の項目に、Edition Flexとの連携に必要な情報を追加します。

基本情報管理	<b>商品管理</b>	会員管理	受注管理	売上集計	メルマガ管理	コンテンツ管理	デザイン管理	システム設定	オーナーズストア
商品管理 > 商品登録									
基本情報									
商品ID									
商品名 *	<input type="text"/> (上限50文字)								
商品カテゴリ *	<input type="button" value="← 登録"/> <input type="button" value="削除 →"/> <ul style="list-style-type: none"> <li>&gt;雑貨</li> <li>&gt;シール</li> <li>&gt;シール&gt;びん用シール</li> <li>&gt;シール&gt;お菓子</li> <li>&gt;シール&gt;お菓子&gt;アイス</li> <li>&gt;レシピ</li> </ul>								
公開・非公開 *	<input type="radio"/> 公開 <input checked="" type="radio"/> 非公開								
商品ステータス	<input type="checkbox"/> NEW <input type="checkbox"/> 残りわずか <input type="checkbox"/> ポイント2倍 <input type="checkbox"/> オススメ <input type="checkbox"/> 限定品								
商品種別 *	<input type="radio"/> 通常商品 <input type="radio"/> ダウンロード商品 <input checked="" type="radio"/> <b>デザイン編集商品</b>								
ダウンロード商品ファイル名 *	<input type="text"/> (上限50文字)								
ダウンロード商品用ファイルアップロード *	<input type="button" value="参照"/> ファイルが選択されていません。 <input type="button" value="アップロード"/> 登録可能拡張子: zip,lzh,jpe,jpeg,gif,png,mp3,pdf,csv (パラメーター DOWNLOAD_EXTENSION)								
テンプレートID *	<input type="text"/> (上限50文字)								
商品コード *	<input type="text"/> (上限50文字)								
通常価格	<input type="text"/> 円 (半角数字で入力)								
販売価格 *	<input type="text"/> 円 (半角数字で入力)								
在庫数 *	<input type="text"/> <input type="checkbox"/> 無制限								
商品送料	<input type="text"/> 円 (半角数字で入力) ※現在無効です (パラメーター OPTION_PRODUCT_DELIV_FEE)								
ポイント付与率 *	<input type="text"/> 0 % (半角数字で入力)								
発送日目安	<input type="text"/> 選択済								

ECサイトの画面はサンプルです。実際に稼働しているものではありません。

通常の商品とデザイン編集可能な商品のカート投入時の処理の区別ができるよう、フラグを設置します。デザイン編集フラグがOnの商品のみ、注文時にEdition Flexを呼び出すようにします（フラグOffの商品は従来通りの動作）。

**デザイン編集** フラグOn時の商品選択ボタン

**カゴに入れる** フラグOff時の商品選択ボタン

Edition FlexのテンプレートID(trackingId)を登録できるように項目を追加します。Edition Flexは、選択された商品に登録されているテンプレートIDを使用して編集画面を開きます。



テンプレートID(trackingId)は、Edition CMSのテンプレート一覧画面のテンプレート名の下に表示されています。

商品マスタテーブル

商品情報

+

デザイン編集フラグ

trackingId

## 4. サイトイメージ3 - 受注管理画面

受注管理画面では、受注した商品の原稿を取得するための機能を設置します。

検索結果一覧

7件 が該当しました。 [CSVダウンロード](#) [CSV出力項目設定](#) [PDF一括出力](#) [メール一括通知](#)

1

受注日	注文番号	お名前	支払方法	購入金額(円)	全商品発送日	対応状況	帳票 <input type="checkbox"/>	編集	メール <input type="checkbox"/>	削除
2013/11/11 12:23	7	レゾロジック	銀行振込	205	未発送	新規受付	<input type="checkbox"/> 一括出力 個別出力	編集 原稿あり	<input type="checkbox"/> 一括通知 個別通知	
2013/11/09 12:21	6	レゾロジック	銀行振込	205	未発送	新規受付	<input type="checkbox"/> 一括出力 個別出力	編集 原稿あり	<input type="checkbox"/> 一括通知 個別通知	
2013/11/09 11:00	5	レゾロジック	銀行振込	205	未発送	新規受付	<input type="checkbox"/> 一括出力 個別出力	編集 原稿あり	<input type="checkbox"/> 一括通知 個別通知	削除
2013/11/08 11:17	4	レゾロジック	銀行振込	205	未発送	新規受付	<input type="checkbox"/> 一括出力 個別出力	編集 原稿あり	<input type="checkbox"/> 一括通知 個別通知	削除
2013/11/08 11:14	3	レゾロジック	銀行振込	205	未発送	新規受付	<input type="checkbox"/> 一括出力 個別出力	編集	<input type="checkbox"/> 一括通知 個別通知	削除

ECサイトの画面はサンプルです。  
実際に稼働しているものではありません。

受注一覧画面で、デザインデータの trackingId を持つ注文には「原稿あり」と表示します。

受注商品情報 [計算結果の確認](#) [商品の追加](#)

商品コード	商品名/規格1/規格2	原稿データ	単価	数量	税込み価格	小計
NC-0001	IT系名刺 <a href="#">変更</a>	ダウンロード	1,600 円	1	1,600 円	1,600 円
小計						1,600 円
値引き						0 円
送料						1600 円
手数料						0 円
合計						1,600 円
お支払い合計						1,600 円
使用ポイント						0 pt
加算ポイント						0 pt

受注商品情報画面の「ダウンロード」リンクをクリックすると、原稿PDFを出力します。

[Flex API](#)  
14ページ

お届け先情報 [お客様情報へお届けする](#) [お届け先を新規追加](#) [複数のお届け先を指定する](#)

注文詳細テーブル

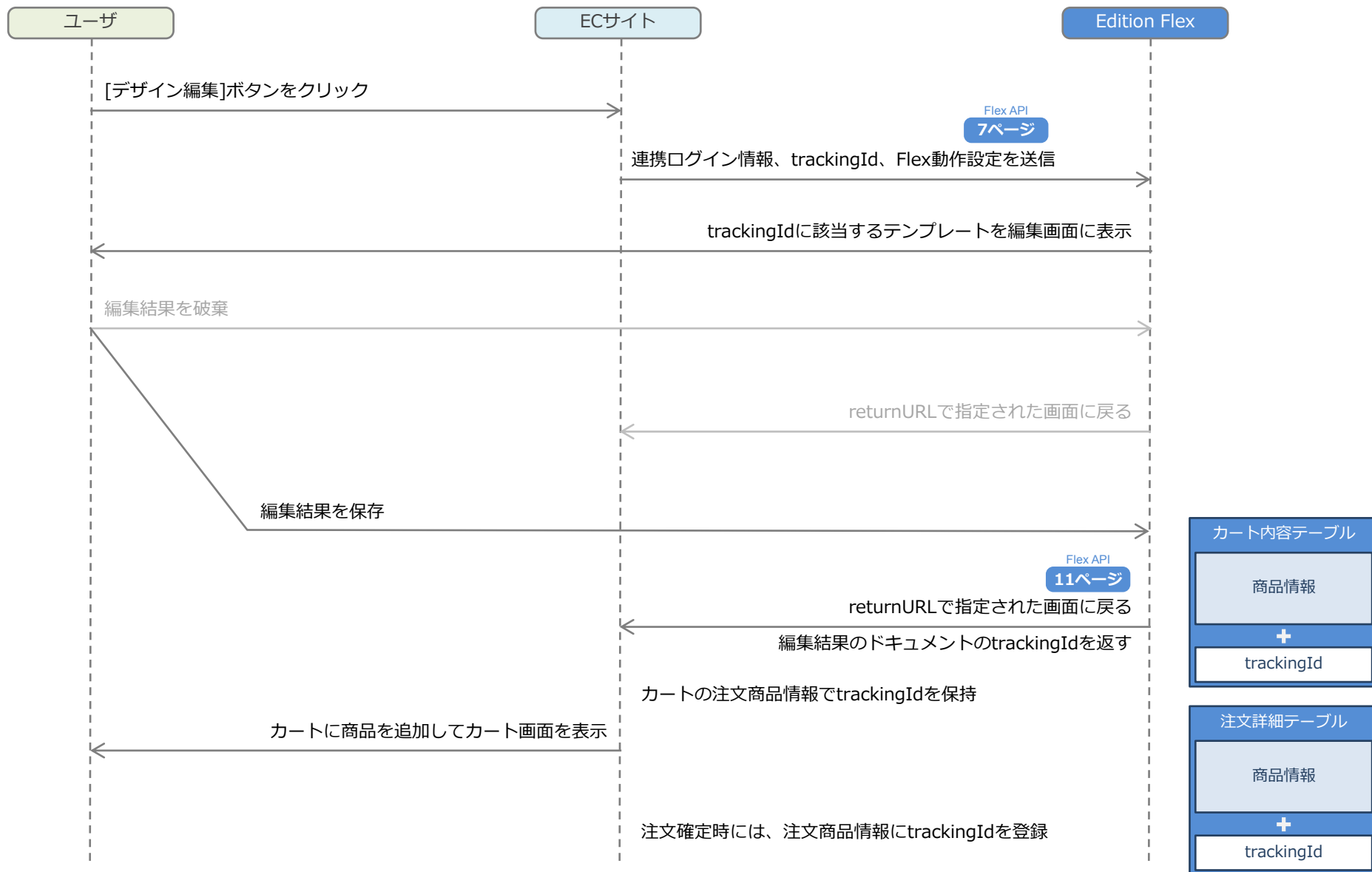
商品情報

+

trackingId

## 5. 注文時の流れ

選択した商品をEdition Flex編集画面で編集後、カートに追加するまでの処理の流れです。



## 6.Flex APIによるEdition Flexの呼び出し1 – デザイン編集 (1/3)

Edition Flex編集画面を開く場合は、[デザイン編集]ボタンをクリックした際に、次のパラメータをEdition FlexにPOSTします。

```
<form method="post" name="editForm" action="http://<Edition FlexのURL>/document/layout" > (1)
  <input type="hidden" name="relayUserId" value="demo" /> (2)
  <input type="hidden" name="dtime" value="Tue, 05 Aug 2014 16:29:23 +0900" /> (3)
  <input type="hidden" name="clientId" value="a73ae5e277d0ea11412ace5ac88c5fa3" /> (4)

  <input type="hidden" name="resultType" value="html" /> (5)
  <input type="hidden" name="returnURL" value="http://<ECサイトのエラー表示ページ>" /> (6)
  <input type="hidden" name="errorURL" value="http://<ECサイト戻り先ページ>" /> (7)

  <input type="hidden" name="editorType" value="general" /> (8)
  <input type="hidden" name="purposeName" value="user" /> (9)
  <input type="hidden" name="resourceType" value="template" /> (10)
  <input type="hidden" name="pageSaveMode" value="document"/> (11)
  <input type="hidden" name="pageSavePath"
value="/documents/ + <ドキュメントの保存先パス>" /> (12)
  <input type="hidden" name="savedAtEnd" value="true"/> (13)

  <input type="hidden" name="srcType" value="trackingId" /> (14)
  <input type="hidden" name="trackingId" value="<テンプレートのtrackingId>" /> (15)

  <input type="hidden" name="forceLayout" value="true" /> (16)
  <input type="hidden" name="multiMode" value="false"/> (17)
</form>
```

- (1) POSTでの呼び出しには、Edition FlexのURL の後に、/document/API を指定します。  
API : layout . . . 編集開始
- (2) relayUserId : 編集時のユーザID  
値 : 携ログイン用ユーザID  
編集時のユーザID を指定します。  
Edition Flex の利用アカウントを指定します。
- (3) dtime : ログイン情報を生成した時刻  
値 : 連携ログイン日時  
ログイン情報を生成した時刻をRFC1123 形式で指定します。  
Edition Flex 呼び出し時に生成してください。  
認証情報有効期間が、0 以外に設定されている場合、  
有効期間の範囲外では呼び出しエラーとなります。
- (4) clientId : 呼び出し元の正当性を示すID  
値 : 連携ログインクライアントID (32Byte の半角英数字)  
dtime と共通鍵を連結した文字列のMD5 の値を指定します。  
共通鍵は、Edition Flex アカウント毎に設定されています。
- (5) resultType : 結果表示方法  
値 : "html" . . . 標準で用意されている画面を表示します。  
編集終了時やエラー時などの結果表示方法を指定します。
- (6) returnUrl : 戻り先URL  
値 : ECサイトの戻り先ページ  
編集終了時に戻るURL を指定します。  
編集結果を取得するためのID を付加して呼び出します。
- (7) errorURL : エラー発生時戻り先URL  
値 : ECサイトのエラー表示ページ  
エラー発生時時に戻るURL を指定します。  
指定しない場合は戻り先URL に戻ります。

Edition Flex編集画面



## 6.Flex APIによるEdition Flexの呼び出し1 - デザイン編集 (2/3)

Edition Flex編集画面を開く場合は、[デザイン編集]ボタンをクリックした際に、次のパラメータをEdition FlexにPOSTします。

```
<form method="post" name="editForm" action="http://<Edition FlexのURL>/document/layout"> (1)
  <input type="hidden" name="relayUserId" value="demo" /> (2)
  <input type="hidden" name="dtime" value="Tue, 05 Aug 2014 16:29:23 +0900" /> (3)
  <input type="hidden" name="clientId" value="a73ae5e277d0ea11412ace5ac88c5fa3" /> (4)

  <input type="hidden" name="resultType" value="html" /> (5)
  <input type="hidden" name="returnURL" value="http://<ECサイトのエラー表示ページ>" /> (6)
  <input type="hidden" name="errorURL" value="http://<ECサイト戻り先ページ>" /> (7)

  <input type="hidden" name="editorType" value="general" /> (8)
  <input type="hidden" name="purposeName" value="user" /> (9)
  <input type="hidden" name="resourceType" value="template" /> (10)
  <input type="hidden" name="pageSaveMode" value="document"/> (11)
  <input type="hidden" name="pageSavePath"
value="/documents/ + <ドキュメントの保存先パス>" /> (12)
  <input type="hidden" name="savedAtEnd" value="true"/> (13)

  <input type="hidden" name="srcType" value="trackingId" /> (14)
  <input type="hidden" name="trackingId" value="<テンプレートのtrackingId>" /> (15)

  <input type="hidden" name="forceLayout" value="true" /> (16)
  <input type="hidden" name="multiMode" value="false"/> (17)
</form>
```

- (8) editorType : エディタの形式  
値：“general”・・・編集画面あり  
エディタの形式を選択します。
- (9) purposeName : 用途  
値：“user”  
編集時の設定を指定します。  
Edition Flexの編集設定ファイル editor.laml のpurpose と合わせてください。
- (10) resourceType : リソースの種類  
値：“template”・・・テンプレート  
読み込むリソースの種類を指定します。
- (11) pageSaveMode : 保存先プロバイダ  
値：“document”・・・デフォルトドキュメントプロバイダ  
保存先プロバイダを指定します。
- (12) pageSavePath : 保存先パス  
値：“/documents/ + <ドキュメントの保存先パス>”  
Edition CMS上の保存先パスを指定します。
- (13) saveAtEnd : 編集終了時の保存  
値：“true”・・・編集画面終了時に無条件で保存  
編集終了時の保存処理の指定をします。  
trueでは、編集画面終了時に無条件で保存処理が行われ、  
returnURLにパラメータ tracingId=\${保存されたtrackingId}  
が付加されて、GETで画面遷移します。
- (14) srcType : 種別  
値：“trackingId”・・・識別コード  
編集の元データとするテンプレートの指定方法を指定します。
- (15) trackingId : 識別コード  
値：テンプレートのtrackingId  
編集の元にするテンプレートを指定します。

Edition Flex編集画面





## 6.Flex APIによるEdition Flexの呼び出し1 - デザイン編集 (3/3)

Edition Flex編集画面を開く場合は、[デザイン編集]ボタンをクリックした際に、次のパラメータをEdition FlexにPOSTします。

```
<form method="post" name="editForm" action="http://<Edition FlexのURL>/document/layout"> (1)
  <input type="hidden" name="relayUserId" value="demo" /> (2)
  <input type="hidden" name="dtime" value="Tue, 05 Aug 2014 16:29:23 +0900" /> (3)
  <input type="hidden" name="clientId" value="a73ae5e277d0ea11412ace5ac88c5fa3" /> (4)

  <input type="hidden" name="resultType" value="html" /> (5)
  <input type="hidden" name="returnURL" value="http://<ECサイトのエラー表示ページ>" /> (6)
  <input type="hidden" name="errorURL" value="http://<ECサイト戻り先ページ>" /> (7)

  <input type="hidden" name="editorType" value="general" /> (8)
  <input type="hidden" name="purposeName" value="user" /> (9)
  <input type="hidden" name="resourceType" value="template" /> (10)
  <input type="hidden" name="pageSaveMode" value="document"/> (11)
  <input type="hidden" name="pageSavePath"
value="/documents/ + <ドキュメントの保存先パス>" /> (12)
  <input type="hidden" name="savedAtEnd" value="true"/> (13)

  <input type="hidden" name="srcType" value="trackingId" /> (14)
  <input type="hidden" name="trackingId" value="<テンプレートのtrackingId>" /> (15)

  <input type="hidden" name="forceLayout" value="true" /> (16)
  <input type="hidden" name="multiMode" value="false"/> (17)
</form>
```

- (16) forceLayout : 編集画面の起動方法  
値: "true" ……強制的に編集開始  
同一セッションで編集画面が実行中の場合に編集画面を開くかどうか指定します。
- (17) multiMode : 同一セッションでの複数編集画面使用  
値: "false" ……複数使用しない  
同一セッションで複数の編集画面の使用可否を指定します。

Edition Flex編集画面



## 7.Flex APIによるEdition Flexの呼び出し2 - ウィザード編集

ウィザード編集画面を使用する場合は、[デザイン編集]ボタンをクリックした際に、次のパラメータをEdition FlexにPOSTします。

```
<form method="post" name="editForm" action="http://<Edition FlexのURL>/document/wizard"> (1)
  <input type="hidden" name="relayUserId" value="demo" />
  <input type="hidden" name="dtime" value="Tue, 05 Aug 2014 16:29:23 +0900" />
  <input type="hidden" name="clientId" value="a73ae5e277d0ea11412ace5ac88c5fa3" />

  <input type="hidden" name="resultType" value="html" />
  <input type="hidden" name="returnURL" value="http://<ECサイトのエラー表示ページ>" />
  <input type="hidden" name="errorURL" value="http://<ECサイト戻り先ページ>" />

  <input type="hidden" name="editorType" value="general" />
  <input type="hidden" name="purposeName" value="user" />
  <input type="hidden" name="resourceType" value="template" />
  <input type="hidden" name="pageSaveMode" value="document"/>
  <input type="hidden" name="pageSavePath"
  value="/documents/ + <ドキュメントの保存先パス>" />
  <input type="hidden" name="savedAtEnd" value="true"/>

  <input type="hidden" name="srcType" value="trackingId" />
  <input type="hidden" name="productTrackingId" value="<テンプレートのtrackingId>" /> (2)

  <input type="hidden" name="forceLayout" value="true" />
  <input type="hidden" name="multiMode" value="false"/>
</form>
```

(1) POSTでの呼び出しには、Edition FlexのURL の後に、/document/API を指定します。

API : wizard . . . ウィザード編集開始

(2) productTrackingId : 識別コード

値 : テンプレートのtrackingId

ウィザード編集定義済みのテンプレートを指定します。

上記以外のパラメータは、7~9ページのデザイン編集と同じです。

## ウィザード編集画面



## 8.Flex APIによるEdition Flexの呼び出し3 — デザイン編集からの戻り

Edition Flex編集画面で [終了]ボタンをクリックすると、7ページのPOSTパラメータ“returnURL”で指定したURLに、編集結果のtrackingIdをパラメータとして付加して遷移します。

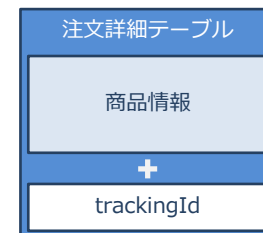
例：http://<ECサイトの戻りURL>?trackingId=a5d1c451c0a80032028d1bc43a36738d&flexId= . . .

ECサイト側のプログラムでは、このtrackingIdの値を取得し、カートに追加した商品と紐付けして保持するようにします。

カート画面から再編集する場合は、Edition Flexへ送信するPOSTパラメータ“trackingId”の値に保持したtrackingIdの値を使用します。

また、注文が完了した際には、注文情報の1項目としてデータベースに登録します。

この情報は、運営側ユーザが原稿データをダウンロードする場合や、エンドユーザが再注文する際に再編集する場合などに使用します。



## 9. Flex APIによるEdition Flexの呼び出し4 — 再編集（Edition Flex編集画面）

カート画面の[デザイン編集]ボタンをクリックした際は、すでに編集したドキュメントを開きます（再編集）。次のパラメータをEdition FlexにPOSTします。

```
<form method="post" name="editForm" action="http://<Edition FlexのURL>/document/layout">
  <input type="hidden" name="relayUserId" value="demo" />
  <input type="hidden" name="mtime" value="Tue, 05 Aug 2014 16:29:23 +0900" />
  <input type="hidden" name="clientId" value="a73ae5e277d0ea11412ace5ac88c5fa3" />

  <input type="hidden" name="resultType" value="html" />
  <input type="hidden" name="returnURL" value="http://<ECサイトのエラー表示ページ>" />
  <input type="hidden" name="errorURL" value="http://<ECサイト戻り先ページ>" />

  <input type="hidden" name="editorType" value="general" />
  <input type="hidden" name="purposeName" value="user" />
  <input type="hidden" name="resourceType" value="document" /> (1)
  <input type="hidden" name="pageSaveMode" value="document"/>
  <input type="hidden" name="pageSaveTrackingId" value="<ドキュメントのtrackingId>" /> (2)
  <input type="hidden" name="savedAtEnd" value="true"/>

  <input type="hidden" name="srcType" value="trackingId" />
  <input type="hidden" name="trackingId" value="<ドキュメントのtrackingId>" /> (3)

  <input type="hidden" name="forceLayout" value="true" />
  <input type="hidden" name="multiMode" value="false"/>
</form>
```

- (1) resourceType : リソースの種類  
値 : "document"・・・ドキュメント  
読み込むリソースの種類を指定します。
  - (2) pageSaveTrackingId : 識別コード  
値 : ドキュメントのtrackingId  
ドキュメントを（上書き）保存するtrackingIdを指定します。
  - (3) trackingId : 識別コード  
値 : ドキュメントのtrackingId  
再編集するドキュメントを指定します。
- \* 再編集ですので、既存のドキュメントを開き、編集結果は上書き保存するようにパラメータを指定します。  
\* 上記以外のパラメータは、7～9ページのデザイン編集と同じです。

## 10. Flex APIによるEdition Flexの呼び出し5 - 再編集（ウィザード編集画面）

ウィザード編集画面で再編集をする場合は、カート画面の[デザイン編集]ボタンをクリックした際に、次のパラメータをEdition FlexにPOSTします。

```
<form method="post" name="editForm" action="http://<Edition FlexのURL>/document/wizard">
  <input type="hidden" name="relayUserId" value="demo" />
  <input type="hidden" name="dtime" value="Tue, 05 Aug 2014 16:29:23 +0900" />
  <input type="hidden" name="clientId" value="a73ae5e277d0ea11412ace5ac88c5fa3" />

  <input type="hidden" name="resultType" value="html" />
  <input type="hidden" name="returnURL" value="http://<ECサイトのエラー表示ページ>" />
  <input type="hidden" name="errorURL" value="http://<ECサイト戻り先ページ>" />

  <input type="hidden" name="editorType" value="general" />
  <input type="hidden" name="purposeName" value="user" />
  <input type="hidden" name="resourceType" value="document" /> (1)
  <input type="hidden" name="pageSaveMode" value="document"/>
  <input type="hidden" name="pageSaveTrackingId" value="<ドキュメントのtrackingId>" /> (2)
  <input type="hidden" name="savedAtEnd" value="true"/>

  <input type="hidden" name="srcType" value="trackingId" />
  <input type="hidden" name="productTrackingId" value="<ドキュメントのtrackingId>" /> (3)

  <input type="hidden" name="forceLayout" value="true" />
  <input type="hidden" name="multiMode" value="false"/>
</form>
```

- (1) resourceType : リソースの種類  
値 : "document"・・・ドキュメント  
読み込むリソースの種類を指定します。
  - (2) pageSaveTrackingId : 識別コード  
値 : ドキュメントのtrackingId  
ドキュメントを（上書き）保存するtrackingIdを指定します。
  - (3) productTrackingId : 識別コード  
値 : ドキュメントのtrackingId  
再編集するドキュメントを指定します。
- \* 再編集ですので、既存のドキュメントを開き、編集結果は上書き保存するようにパラメータを指定します。  
\* 上記以外のパラメータは、10ページのウィザード編集と同じです。

## 11. Flex APIによるEdition Flexの呼び出し6 – PDF出力

受注管理画面で原稿データの[ダウンロード]リンクをクリックした際に、PDF原稿をEdition Flexに生成させる場合は、次のパラメータをEdition FlexにPOSTします。

```
<form method="post" name="editForm" action="http://<Edition FlexのURL>/document/layout" >
  <input type="hidden" name="relayUserId" value="demo" />
  <input type="hidden" name="dtime" value="Tue, 05 Aug 2014 16:29:23 +0900" />
  <input type="hidden" name="clientId" value="a73ae5e277d0ea11412ace5ac88c5fa3" />

  <input type="hidden" name="resultType" value="html" />
  <input type="hidden" name="resultName" value="pdfAuto" /> (1)
  <input type="hidden" name="returnURL" value="http://<ECサイトのエラー表示ページ>" />
  <input type="hidden" name="errorURL" value="http://<ECサイト戻り先ページ>" />

  <input type="hidden" name="editorType" value="none" /> (2)
  <input type="hidden" name="purposeName" value="user" />
  <input type="hidden" name="savedAtEnd" value="false"/> (3)

  <input type="hidden" name="srcType" value="trackingId" />
  <input type="hidden" name="trackingId" value="<ドキュメントのtrackingId>" />

  <input type="hidden" name="forceLayout" value="true" />
  <input type="hidden" name="multiMode" value="false"/>
</form>
```

- (1) resultName : 戻り画面ページ名  
値：“pdfAuto”・・・Edition Flexに登録した戻り画面の名称  
編集終了時に表示する戻り画面を指定します。  
連携ユーザでEdition Flexにログインし、個人設定から戻り画面  
を作成します。  
無指定または、戻り画面が存在しない場合は、デフォルトの編集  
終了画面を表示します
- (2) editorType : エディタの形式  
値：“none”・・・編集画面なし  
エディタの形式を選択します。
- (3) savedAtEnd : 編集終了時の保存  
値：“false”・・・編集画面終了時に戻り画面を表示  
編集終了時の保存処理の指定をします。  
falseでは、編集画面終了時にresultNameで指定した戻り画面  
が表示されます。

\* PDF原稿生成では編集画面を表示しないで、戻り画面を表示するよう  
に指定します。

\* 上記以外のパラメータは、7～9ページのデザイン編集と同じです。

## 12. 戻り画面サンプル

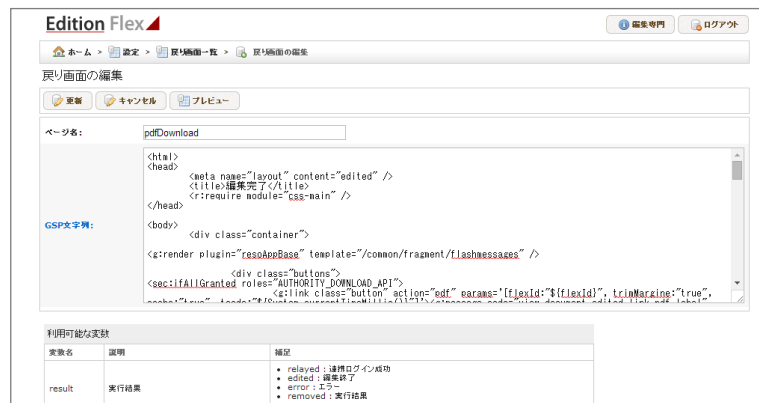
Edition Flexの戻り画面とは、Edition Flex編集画面終了時に表示される画面で、任意の画面を登録できます。

戻り画面の編集は、連携ユーザアカウント（または管理者アカウント）でEdition Flexにログインし、戻り画面一覧画面から追加・編集ができます。

3秒後に自動的にダウンロードが開始されます。  
開始されない場合は、以下のボタンを押して下さい。

PDFのダウンロード

戻り画面 pdfAuto



Edition Flex 戻り画面一覧

pdfAuto サンプル

```
<html>
<head>
  <meta name="layout" content="edited" />
  <title>PDFダウンロード</title>
  <r:require module="css-main" />
  <meta http-equiv="refresh" content="3;URL=pdf?flexId=${flexId}&trimMargine=true&tcode=${System.currentTimeMillis()}&pdfOptionSetName=print&purge=true">
</head>
<body>
  3 秒後に自動的にダウンロードが開始されます。 <br />
  開始されない場合は、以下のボタンを押して下さい。 <br /> <br />

  <g:link class="button" action="pdf" params="[flexId: '${flexId}', trimMargine: 'true', cache: 'true', tcode: '${System.currentTimeMillis()}', pdfOptionSetName: 'print',
  purge: 'true']">
  <g:message code="view.document.edited.link.pdf.label" default="PDFのダウンロード" /></g:link>
</body>
</html>
```

PDFの出力設定は、Edition Flexに登録されている「PdfOptions」というファイルで設定されています。  
このサンプルでは、PdfOptionsの設定名「print」という設定を参照するようになっています。（次ページ）

## 13. PDF出力設定 (PdfOptions)

PDFの出力設定は、Edition Flexに登録されている「PdfOptions」というファイルで設定されています。

戻り画面の編集は、連携ユーザアカウント（または管理者アカウント）でEdition Flexにログインし、PdfOptionsの更新画面から登録できます。



### Edition Flex PdfOptionsの更新

#### PDFOption.dsl.groovy

```

/*
 * 印刷オプション
 */
PDFOptions {
  // 用途 { // 用途名には英数字のみを利用して下さい。
  //   cmykJpegMode      = "JPEG" | "RAW" | "Pass Through" // CMYK-JPEG-MODE
  //   cmyk              = true | false // カラープロファイル
  //   cmykColorProfileName = "CMYKカラープロファイル名"
  //   rasterFormat      = 無指定 | "" | "JPEG" | "PNG" | "GIF" | "BMP" : 共通: ラスター出力フォーマット(無指定または"" の場合は、PDF)
  //   highResolution    = true | false : 共通: 高解像度
  //   sampleMark        = true | false : 共通: 見本マーク有無
  //   sampleMarkStr     = true | false : 共通: 見本マーク文字列
  //   trimMargine       = true | false : 共通: 余白トリミング
  //   checkFont         = true | false : 共通: フォントチェック
  //   pdfVersion        = "1.3" | "1.4" | "1.5" : PDF: PDFバージョン
  //   cmykJpegMode      = "JPEG" | "RAW" | "Pass Through" : PDF: CMYK-JPEG-MODE
  //   cmyk              = true | PDF: false : PDF: カラープロファイル
  //   pdfMasterPassword = "" : PDF: 権限パスワード
  //   pdfUserPassword  = "" : PDF: ユーザパスワード
  //   pdfPermissions   = "noprint nomodify nocopy noannots" : PDF: 制限(複数ある時は、スペースで区切る)
  //   rasterDpi         = 数値 : JPEG: ラスター出力時のDPI
  //   rasterSmoothing   = true | false : JPEG: ラスター出力時のスムージング処理の有無
  // }

  // PDF/CMYK/高解像度/余白あり/見本マークなし
  print {
    cmyk          = true
    highResolution = true
    sampleMark    = false
    trimMargine   = false
    linearize     = true
  }
  // JPEG/高解像度/余白あり/見本マークなし
  jpeg {
    rasterFormat      = "JPEG"
    highResolution    = true
    sampleMark        = false
    trimMargine       = false
    rasterSmoothing   = true
  }
  rasterPxSize = 400
}

```

設定名「print」



