
変更年月日	変更内容
-------	------

2013-02-01	初版
------------	----

2014-012-01	第2版 下記を修正しました。
-------------	----------------

- 目次項目の見直しを行いました。
-

概要

サンプルプログラムを通して、SAP リアルタイム連携 API による開発の方法を読者に伝えることを目的とします。本書を通して、SAP リアルタイム連携 API の利用方法を理解することができ、intra-mart 上のWeb システムから SAP 上のデータをリアルタイムに更新・取得するアプリケーションを作成する手順を行います。

対象読者

- 本書は SAP リアルタイム連携 API による開発を行おうとする開発者の方を対象にしています。
- 本書の対象読者として、以下の知識は本書を読み進める際に必要な前提知識とします。
 - Java 言語の説明
 - JavaScript 言語の説明
- JSP および Servlet などの ServerSideJava プログラミングに関する説明
- モデリング言語(主に UML)に関する説明
- JavaEE 技術に関する説明
- SAP の基本操作に関する説明
- SAP の業務モジュールに関する説明
- SAP の BAPI (RFC 汎用モジュール) の定義・構造等に関する説明
- 本書ではスクリプト開発モデルのサンプルプログラミングを中心に使用方法を説明しています。
- SAP の基本的な操作や汎用モジュールビルダ (t-cd : SE37) による BAPI (RFC 汎用モジュール) の動作確認等を理解している読者を対象としています(また、それを強くお勧めします)。
- SAP 財務会計・管理会計・経費購買・販売管理モジュール (FI・CO・MM) に関連する用語・モジュール知識等を事前に理解している読者を対象としています。

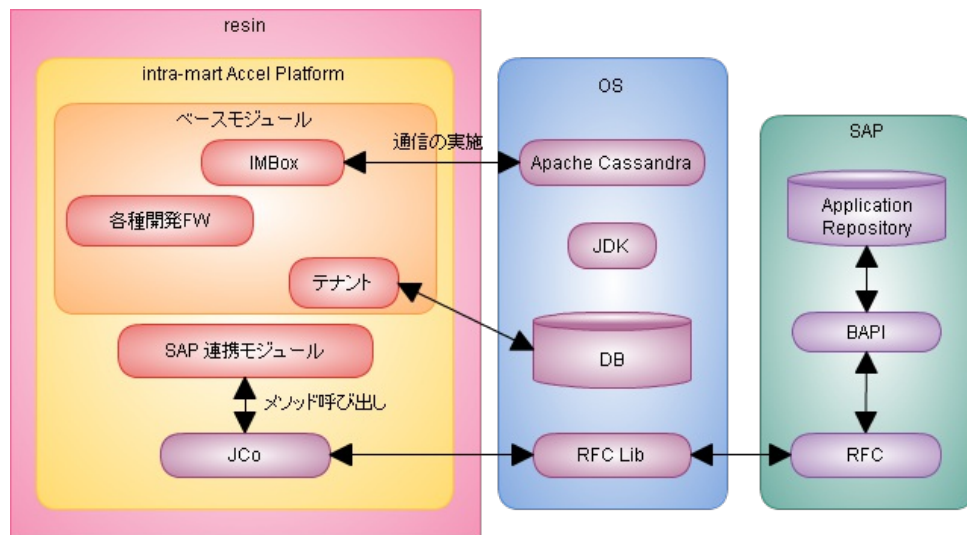
本書の構成

本書では以下の4つのカテゴリで構成されています。

- [SAPリアルタイム連携APIの構造](#)
IM-ERP Real Connect for Accel Platformで提供されているSAP連携APIの動作の原理について説明します。
- [汎用向けAPI](#)
IM-ERP Real Connect for Accel Platformで提供されている汎用向けSAP連携APIを利用した開発方法について説明します。
- [業務向けAPI](#)
IM-ERP Real Connect for Accel Platformで提供されている業務用向けSAP連携APIを利用した開発方法について説明します。
- [Tips](#)
エラー処理、トランザクション、コネクションなどSAP連携時に必要な制御方法について説明します。

SAP リアルタイム連携 API とは、intra-mart 上の Web システムから SAP 内のデータを取得・更新するためのJavaAPI 群の総称です。SAP には外部システムから SAP 内のデータにアクセスするためのインタフェースであるBAPI (RFC 汎用モジュール) が実装されています。このインタフェースは、ビジネスプロセスをカプセル化したコンポーネントのメソッドで構成されます。また、SAP 社は BAPI を操作するための JavaAPI 群である JCo (JavaConnector) を提供しています。SAP リアルタイム連携 API は、JCo をラップした JavaAPI 群であり、JCo を意識することなく、容易に SAP 内のデータの取得・更新を実現する機能を提供します。提供されている各APIの仕様に関しては、以下を参照してください。

- 汎用向けAPI
- 業務向けAPI



注意

intra-mart の提供範囲は、上記図の点線で囲まれた SAP 連携モジュール (SAP リアルタイム連携 API) のみです。SAP 社の提供する SAP 環境及び、「JCo」、「RFC Lib」は intra-mart には含まれません。

本項では汎用APIの概要と利用方法について説明を行います。

汎用向けAPIの役割

汎用向けAPIには以下の役割が存在します。

1. 各業務向けAPIのコアAPIとしての機能

各業務にて、SAP 上のデータを取得/更新する際、SAP との通信開始→BAPIデータ取得・更新→通信終了の処理は、中核に位置する汎用向け API が実行しています。

業務向けAPIは、汎用向けAPIに引数として値を設定して、汎用向けAPIが処理した結果を戻り値として受け取る処理のみを担っています。

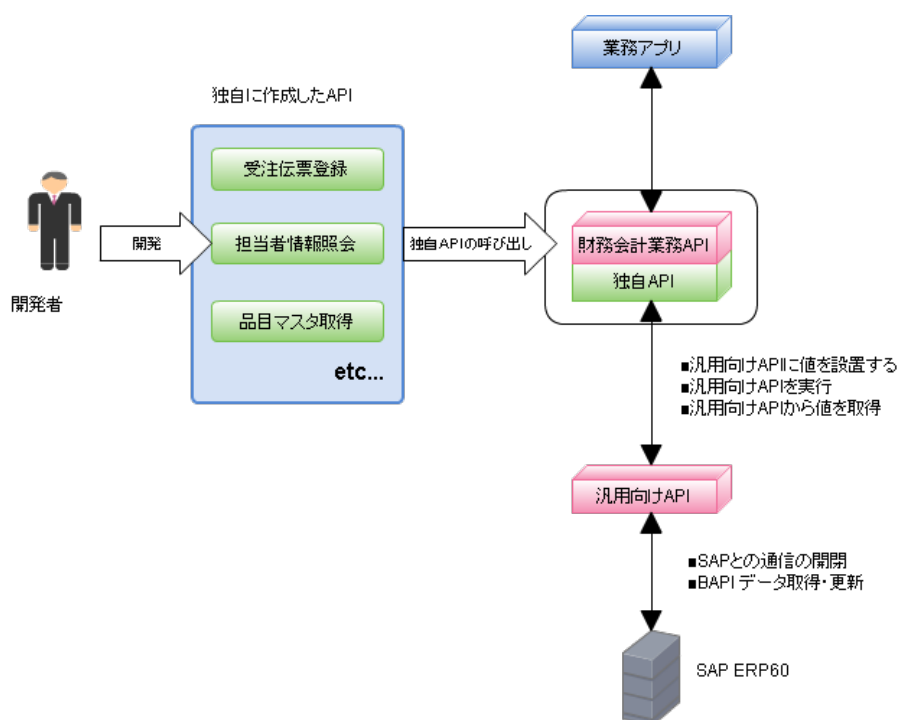
そのため開発者は、汎用向け API を意識することなく業務向けAPIを利用することになります。

2. 独自APIの作成サポート

現在、intra-mart では SAP の FI・CO・MM モジュールに関する一部のデータをターゲットとした、財務・管理会計業務、経費購買向け API が提供されています。

ただし、案件によっては、HR、SD 等のモジュールに関するデータや、intra-mart の会計業務向け API ではサポート外のデータを取得・更新するニーズが発生するケースも存在すると思われます。

汎用向け API では、SAP との通信開始→データ取得・更新→通信終了といった一連のプロセスを直接制御するための Java クラス群を提供しています。



汎用向けプロパティファイルの編集

本項では、汎用向けのプロパティファイルの構成と記述方法について説明します。

汎用向けプロパティファイルについて

開発者が汎用向け API を使用して、SAP 上から任意のデータを取得・更新する際には、使用する BAPI の定義情報をファイルに設定する必要があります。

汎用プロパティファイルは、使用する BAPI の定義情報を記述することで、記述した定義に従ったデータの取得・更新を可能にするためのファイルです。

汎用プロパティファイルは、一般的にJavaアプリケーションで利用する「.propertiesファイル」になっています。

使用可能な KEY 値は、以下のようになっています。

KEY 値	説明
servertype	SAP の接続先を識別する為の servertype の値。

KEY 値	説明
commit	true : トランザクション処理を実施 (更新系) false : トランザクション処理を実施しない (照会系)
bapiname	SAP 上のデータ取得・更新のために使用する BAPI の名称。
import.import.[parameter]	名称が[parameter]である import 型のパラメータ。value値は、対応する Java の HashMap の Key 値を指定する。
import.structure.[parameter].[component]	名称が[parameter]である import 型の sturcture の内、名称が [component]のコンポーネント。value値は、対応する Java の HashMap の Key 値を指定する。
import.table.[parameter].[component]	名称が[parameter]である import 型の table の内、名称 が[component]のコンポーネント。value 値は、対応するJava の HashMap の Key 値を指定する。
export.export.[parameter]	名称が[parameter]である export 型のパラメータ。value値は、対応する Java の HashMap の Key 値を指定する。
export.structure.[parameter].[component]	名称が[parameter]である export 型の sturcture の内、名称が [component]のコンポーネント。value値は、対応する Java の HashMap の Key 値を指定する。
export.table.[parameter].[component]	名称が[parameter]である export 型の table の内、名称が [component]のコンポーネント。value 値は、対応するJava の HashMap の Key 値を指定する。

プロパティファイルの編集

以下では、会計業務向け API で使用しているプロパティファイルを例にして、記述方法を説明します。

```

servertype= . . . . . 説明 1
commit=false . . . . . 説明 2
bapiname=BAPI_PROFITCENTER_GETLIST
#import 定義 . . . . . 説明 3
import.import.CONTROLLINGAREA=controllingarea
import.import.PERSONINCHARGE=personincharge
import.import.DATE=date
#table 定義 . . . . . 説明 4
export.table.PROFITCENTER_LIST.PROFIT_CTR=profit_ctr
export.table.PROFITCENTER_LIST.VALID_TO=valid_to
export.table.PROFITCENTER_LIST.PCTR_NAME=pctr_name
export.table.PROFITCENTER_LIST.IN_CHARGE=in_charge
    
```

- 説明1

sap_auth_info.properties の認証情報で servertypeを使用していない場合は、value 値を記述する必要はありません。
- 説明2

利益センタの一覧を取得する照会系の BAPI を使用するので、value 値は「false」となります。
- 説明3

BAPI_PROFITCENTER_GETLIST の import 型のパラメータを記述します。
 この項目は BAPI に対しての引数として処理されます。
 value 値は、引数をセットする際に使用する Java の HashMap の key 値になります。
 value 値には、任意の半角文字を指定することが可能です。
- 説明4

BAPI_PROFITCENTER_GETLIST の export 型の table パラメータを記述します。
 パラメータ名「PROFITCENTER_LIST」のコンポーネント「PROFIT_CTR」「VALID_TO」「PCTR_NAME」「IN_CHARGE」の情報を table 型の戻り値として複数レコード取得することが可能です。
 value 値は、戻り値を取得する際に使用する Java の HashMap の key 値になります。
 value 値には、任意の半角文字を指定することが可能です。

SAP リアルタイム連携 API では、汎用向け API を利用するためのクラスとして、BapiController クラスが提供されています。
 汎用向け API を利用して、独自のデータ取得・更新 API を作成するためには、BapiController クラスの持つ各メソッドを利用する必要があります。
 BapiController クラスの持つ各メソッドは、以下の通りです。
 引数、戻り値についての詳細な情報は、API リストを参照してください。

メソッド一覧

コンストラクタ / メソッド名	説明
BapiController	第1引数あり。 コンストラクタ。 引数で指定した汎用プロパティファイルの情報を読み込む。
putImportParam	第1、第2引数あり。 import 型のパラメータの値を設定する。 第1引数は、汎用プロパティファイル内の該当するパラメータの value 値を指定する。 第2引数は、BAPI に引渡す値を指定する。
putStructParam	第1～第3引数あり。 import 型の structure パラメータの値を設定する。 第1引数は、汎用プロパティファイル内の該当するパラメータ、コンポーネントの value 値を指定する。 第2引数は、BAPI に引渡す値を指定する。 第3引数は、structure のパラメータ名を指定する。
putTableParam	第1～第4引数あり。 import 型の table パラメータの値を設定する。 第1引数は、汎用プロパティファイル内の該当するパラメータ、コンポーネントの value 値を指定する。 第2引数は、BAPI に引渡す値を指定する。 第3引数は、table のパラメータ名を指定する。 第4引数は、レコード番号を指定する (table は複数件のデータが存在する可能性がある)。
executeBapi	BAPI を実行する。
isSuccess	BAPI 実行が正常に行われたか判定する。 標準の設定では、BAPI 実行結果である「RETURN」パラメータの Type 情報が、全て正常「S」、または空文字であった場合、true を返す。 それ以外は false を返す。
getErrorMessage	BAPI 実行時に発生したエラーメッセージを取得する。 標準の設定では、Type 情報が正常「S」または空文字のいずれでもない、全ての「RETURN」パラメータのメッセージを連結して返す。
getMessage	BAPI 実行時に発生したメッセージを取得する。 標準の設定では、Type 情報が正常「S」または空文字の全ての「RETURN」パラメータのメッセージを連結して返す。
getExportParam	第1引数あり。 export 型のパラメータの値を取得する。 第1引数は、汎用プロパティファイル内の該当するパラメータの value 値を指定する。
getStructureParam	第1引数あり。 export 型の structure パラメータの各コンポーネント (※) の値を配列で取得する。 第1引数は、該当する structure の名称を指定する。 ※プロパティファイル内で export 型にて定義したコンポーネントの値のみ取得可能。

コンストラクタ / メソッド名	説明
getTableParam	<p>第1引数あり。</p> <p>export 型の table パラメータの各コンポーネント（※1）の値を2次元配列（※2）で取得する。</p> <p>第1引数は、該当する table の名称を指定する。</p> <p>※1 プロパティファイル内で export 型にて定義したコンポーネントの値のみ取得可能。</p> <p>※2 複数データ存在する場合があるので、2次元配列になる。</p>
getReturnList	<p>BAPI 内で table パラメータとして定義されている「RETURN」パラメータの各種情報を取得する。</p>
getReturn	<p>BAPI 内で structure パラメータとして定義されている「RETURN」パラメータの各種情報を取得する。</p>

サンプルプログラムによる作成例

サンプルプログラムの概要

サンプルとして実装する機能についての仕様は以下の通りです。

- 機能概要
 - 会社一覧画面で任意の会社を選択すると、選択した会社の詳細情報が次画面にて表示される。詳細画面にて表示される項目は以下の通りとする。
 - 会社コード
 - 会社名
 - 市区町村
 - 国コード
 - 通貨コード
 - 勘定コード表
 - 会計年度バリエーション
- 使用する BAPI
 - BAPI_COMPANYCODE_GETDETAIL 標準 BAPI
- 使用する汎用プロパティファイル名
 - sapapi_tutorial.properties
- 備考
 - 業務用APIの会社取得APIのサンプルプログラムを元に作成する。

汎用プロパティファイルの作成

汎用モジュールビルダ (t-cd : SE37) で、「BAPI_COMPANYCODE_GETDETAIL」のパラメータ名、コンポーネント名等を確認して、汎用プロパティファイル「sapapi_tutorial.properties」を作成します。ファイルを保存する場所は、intra-mart のアプリケーションディレクトリ (例 : C:/imart/doc/imart) 直下の/WEB-INF/classes フォルダです。

```

servertype=
commit=false

bapiname=BAPI_COMPANYCODE_GETDETAIL

import.import.COMPANYCODEID=imp_key1

export.structure.COMPANYCODE_DETAIL.COMP_CODE=imp_str_key1
export.structure.COMPANYCODE_DETAIL.COMP_NAME=imp_str_key2
export.structure.COMPANYCODE_DETAIL.CITY=imp_str_key3
export.structure.COMPANYCODE_DETAIL.COUNTRY=imp_str_key4
export.structure.COMPANYCODE_DETAIL.CURRENCY=imp_str_key5
export.structure.COMPANYCODE_DETAIL.CHRT_ACCTS=imp_str_key6
export.structure.COMPANYCODE_DETAIL.FY_VARIANT=imp_str_key7

```

サンプルのコーディング

以下のファイルをサンプル用の画面、及び処理として実装します。

- 入力画面(im_sapapi.html)

```

<imart type="head">
  <title>チュートリアル</title>
  <script type="text/javascript">
    // 前の画面へと移動.
    function refresh() {
      $('#refresh-form').submit();
    };
    function back() {
      $('#back-form').submit();
    };
  </script>
</imart>

<header class="imui-title">
  <h1>SAP リアルタイム連携API チュートリアル（会社取得 API）</h1>
</header>
<div class="imui-toolbar-wrap">
  <div class="imui-toolbar-inner">
    <!-- ツールバー左側 -->
    <ul class="imui-list-toolbar">
      <!-- 戻る -->
      <li><a href="javascript:back();" class="imui-toolbar-icon"
        title="戻る"><span class="im-ui-icon-common-16-back"></span></a></li>
    </ul>
    <!-- ツールバー右側 -->
    <ul class="imui-list-toolbar-utility">
      <li><a href="javascript:refresh();" class="imui-toolbar-icon" title="最新情報"><span
        class="im-ui-icon-common-16-refresh"></span></a></li>
    </ul>
  </div>
</div>

<div class="imui-form-container-wide">
  <div class="imui-chapter-title">
    <h2>会社取得(BAPI_COMPANYCODE_GETLIST)</h2>
  </div>
  <label>※ 実行時の注意点</label><br />
  <label>1 件以上の会社が SAP 上に存在すること。</label><br />
  <form method="POST" action="sapapi_tutorial/tutorial3-4-3/result">
    <div class="imui-operation-parts">
      <input type="submit" value="チュートリアル実行" class="imui-large-button" />
    </div>
  </form>
</div>

<!-- 画面更新用フォーム -->
<form id="back-form" name="backForm" action="sapapi_tutorial/tutorial_menu" method="POST"></form>
<form id="refresh-form" name="refreshForm" action="sapapi_tutorial/tutorial3-4-3/im_sapapi" method="POST"></form>

```

- 入力画面処理(im_sapapi.js)

```
function init(){
}
```

- 一覧画面(result.html)

```
<imart type="head">
<title>チュートリアル</title>
<script type="text/javascript">
// 前の画面へと移動.
function back() {
  $('#back-form').submit();
};
</script> </imart>
<header class="imui-title">
  <h1>SAP リアルタイム連携API チュートリアル実行結果（会社取得 API）</h1>
</header>
<div class="imui-toolbar-wrap">
  <div class="imui-toolbar-inner">
    <!-- ツールバー左側 -->
    <ul class="imui-list-toolbar">
      <!-- 戻る -->
      <li><a href="javascript:back();" class="imui-toolbar-icon"
        title="戻る"><span class="im-ui-icon-common-16-back"></span></a></li>
    </ul>
  </div>
</div>
<div class="imui-form-container-wide">
  <div class="imui-chapter-title">
    <h2>
      会社一覧（
      <imart type="string" value=listCount />
      件）
    </h2>
  </div>
  <br />
  <table class="imui-table mt-20">
    <tbody>
      <tr>
        <th><label>No</label></th>
        <th><label>会社コード</label></th>
        <th><label>会社名</label></th>
      </tr>
      <imart type="repeat" list=listResult item="record" index="idx">
      <tr>
        <td><a rel="alternate"
          href="sapapi_tutorial/tutorial3-4-3/detail"
          name=record.LINK_KEY><imart type="string" value=idx></imart>
        </a>
        </td>
        <td><imart type="string" value=record.COMP_CODE></imart></td>
        <td><imart type="string" value=record.COMP_NAME></imart></td>
      </tr>
      </imart type="repeat">
    </tbody>
  </table>
</div>

<!-- 実行画面遷移用フォーム -->
<form id="back-form" name="backForm"
  action="sapapi_tutorial/tutorial3-4-3/im_sapapi" method="POST"></form>
```

- 一覧画面処理(result.js)

```
var listResult;
var listCount;

function init(request) {
  var objCon;
  var idx;

  //インスタンス生成
  objCon = new CompanyController();

  // コンストラクタでのエラーチェック
  if(objCon.getErrCode() != "0"){
    //エラーメッセージ出力
    Debug.browse(objCon.getErrMsg());
  }

  //会社の一覧を取得
  listResult = objCon.getCompanyList();

  //会社の一覧にページ間リンク用のKEYを追加
  for (idx=0; idx < listResult.length; idx++){
    listResult[idx]["LINK_KEY"] = listResult[idx]["COMP_CODE"]
  }

  //一覧の件数を取得
  listCount = listResult.length;
}
}
```

- 詳細画面(detail.html)

```

<imart type="head">
<title>チュートリアル</title>
<script type="text/javascript">
// 前の画面へと移動.
function back() {
    $('#back-form').submit();
};
</script> </imart>
<header class="imui-title">
<h1>SAP リアルタイム連携API チュートリアル実行結果（会社取得 API）</h1>
</header>
<div class="imui-toolbar-wrap">
<div class="imui-toolbar-inner">
<!-- ツールバー左側 -->
<ul class="imui-list-toolbar">
<!-- 戻る -->
<li><a href="javascript:back();" class="imui-toolbar-icon"
    title="戻る"><span class="im-ui-icon-common-16-back"></span></a></li>
</ul>
</div>
</div>
<div class="imui-form-container-wide">
<div class="imui-chapter-title">
<h2>会社情報 詳細</h2>
</div>
<table class="imui-table mt-20">
<tbody>
<tr>
<th class="wd-20"><label>会社コード</label></th>
<td><imart type="string" value=compCode> </imart></td>
</tr>
<tr>
<th class="wd-20"><label>会社名</label></th>
<td><imart type="string" value=compName> </imart></td>
</tr>
<tr>
<th class="wd-20"><label>市区町村名</label></th>
<td><imart type="string" value=city> </imart></td>
</tr>
<tr>
<th class="wd-20"><label>国コード</label></th>
<td><imart type="string" value=country> </imart></td>
</tr>
<tr>
<th class="wd-20"><label>通貨コード</label></th>
<td><imart type="string" value=currency> </imart></td>
</tr>
<tr>
<th class="wd-20"><label>勘定コード表</label></th>
<td><imart type="string" value=chrtAccts> </imart></td>
</tr>
<tr>
<th class="wd-20"><label>会計年度バリエーション</label></th>
<td><imart type="string" value=fyVariant></imart></td>
</tr>
</tbody>
</table>
</div>

<!-- 実行画面遷移用フォーム -->
<form id="back-form" name="backForm"
    action="sapapi_tutorial/tutorial3-4-3/result" method="POST"></form>

```

- 詳細画面(detail.js)

```

var compCode; //会社コード
var compName; //会社名
var city; //市区町村
var country; //国コード
var currency; //通貨コード
var chrtAccts; //勘定コード表
var fyVariant; //会計年度バリエーション

function init(request) {

    var objCon;
    var objResult;
    var strResult;

    //インスタンス生成
    objCon = new BapiController("sapapi_tutorial");

    //コンストラクタでのエラーチェック
    if(objCon.getErrCode() != "0"){
        //エラーメッセージ出力
        Debug.browse(objCon.getErrMsg());
    }

    //importの設定 (会社コード)
    objCon.putImportParam("imp_key1", request.link_key);

    //BAPI実行
    objResult = objCon.executeBapi();

    //要素[ERR_CODE]に何らかの値がセットされている場合
    if (objResult["ERR_CODE"] != null) {
        //エラー情報を出力する。
        Debug.browse(objResult);
    } else if (!objCon.isSuccess()) {
        //BAPIエラーメッセージを出力する。
        Debug.browse(objCon.getErrorMessage());
    }

    //structureの取得
    strResult = objCon.getStructureParam("COMPANYCODE_DETAIL");

    compCode = strResult["imp_str_key1"]; //会社コード
    compName = strResult["imp_str_key2"]; //会社名
    city = strResult["imp_str_key3"]; //市区町村
    country = strResult["imp_str_key4"]; //国コード
    currency = strResult["imp_str_key5"]; //通貨コード
    chrtAccts = strResult["imp_str_key6"]; //勘定コード表
    fyVariant = strResult["imp_str_key7"]; //会計年度バリエーション
}

```



コラム

スクリプト開発で作成されたシステムのメニューの登録方法、実行方法は「スクリプト開発プログラミングガイド」の内容を参照してください。

本項では業務向けAPIの概要と利用方法について説明を行います。

IM-ERP Real Connectにて、利用できる業務向けAPIのモジュールは以下の4つになります。

財務会計業務向け API

財務会計業務向けAPIの各APIで使用するBAPI情報を定義したプロパティファイルが以下のフォルダに配備されています。

“WEB-INF/classes/jp/co/intra_mart/foundation/sap/module”

ファイル名を変更すると、APIを利用することができませんので、注意してください。

また、認証情報ファイル「sap_auth_info.properties」の接続先情報でservertypeを使用している場合は、各APIのプロパティファイルにservertypeを指定する必要がありますので、servertypeの設定を行ってください。

財務会計業務向け APIの機能は以下となります。

事業領域 取得 API(BusinessAreaController)

事業領域の情報を取得するためには、BusinessAreaController クラスの持つ各メソッドを利用する必要があります。

引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
BusinessAreaController	コンストラクタ（引数無し）。 言語が「JA」、かつ ISO 639 準拠の言語が「JA」である事業領域情報のみ、No3～4 のメソッドにて取得することができるようになる。
BusinessAreaController	コンストラクタ（第1、第2引数あり）。 第1引数で指定した言語、第2引数で指定した ISO 639 準拠の言語で有効な事業領域の情報を、No3～4のメソッドにて取得することができるようになる。
getBusinessAreaList	引数なしコンストラクタの場合、言語が「JA」、かつ ISO 639 準拠の言語が「JA」である事業領域情報（事業領域コード、事業領域名）が格納された配列を取得する。 引数ありコンストラクタの場合、引数で指定した言語、ISO 639 準拠の言語に紐づく事業領域情報（事業領域コード、事業領域名）が格納された配列を取得する。
getBusArDes	引数で指定した「事業領域コード」に対応する事業領域名を取得する。

サンプルプログラム

```

var busArDes;
var listResult;
var listCount;

function init(request) {

    var list;
    var objCon;
    var code;

    //インスタンス生成
    objCon = new BusinessAreaController(); // 説明1

    // コンストラクタでのエラーチェック
    if (objCon.getErrCode() != "0"){

        //エラーメッセージ出力
        Debug.browse(objCon.getErrMsg());
    }

    //事業領域の一覧を取得
    list = objCon.getBusinessAreaList(); // 説明2

    //事業領域の一覧から1件目の「事業領域コード」を取得
    code = list[0]["BUS_AREA"];

    //1件目のデータの事業領域名を取得
    busArDes = objCon.getBusArDes(code); // 説明3

    //一覧の件数を取得
    listCount = list.length; // 説明4
}

```

- 事業領域取得 API のインスタンス生成(説明1)

BusinessAreaController クラスのインスタンスを生成します。
 インスタンスは、以下の2つのいずれかの方法で生成することができます。
 インスタンスを生成すると、BAPI を呼出して事業領域を取得する処理が実行されます。

1. 言語、ISO 639 準拠の言語を指定しないで事業領域を取得する場合

引数なしのインスタンスを生成します。
 言語に「JA」、ISO 639 準拠の言語に「JA」を内部で設定しBAPI を呼び出します。
 objCon = new BusinessAreaController(); は、言語「JA」、ISO 639 準拠の言語「JA」に紐付けられている事業領域情報を取得、となります。

2. 言語、ISO 639 準拠の言語を指定して事業領域を取得する場合

インスタンス（第1、第2引数あり）を生成します。
 第1引数で「言語」、第2引数で「ISO 639 準拠の言語」を指定します。
 もし、objCon = new BusinessAreaController("EN","EN");とした場合は、言語コード「EN」、ISO 639 準拠の言語「EN」に紐付けられている事業領域の情報を取得、となります。

- 事業領域の一覧を取得(説明2)

BusinessAreaController クラスの getBusinessAreaList メソッドを使用すると、事業領域の一覧を取得することが可能です。

- 1件目のデータの事業領域名を取得(説明3)

BusinessAreaController クラスの getBusArDes メソッドを使用すると、引数に指定した事業領域コードに対応する事業領域名を取得することが可能です。
 該当するデータが存在しない場合は、何も返却されません。

- 一覧の件数を取得(説明4)

BusinessAreaController クラスのgetBusinessAreaListメソッドで取得した事業領域の一覧は、lengthで件数を取得することが可能です

管理領域の情報を取得するためには、ControllingAreaController クラスの持つ各メソッドを利用する必要があります。
引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
ControllingAreaController	コンストラクタ (引数無し)。
getControllingAreaList	全ての管理領域情報(管理領域コード、管理領域名)が格納された配列を取得する。
getName	引数で指定した「管理領域コード」に対応する管理領域名を取得する。

サンプルプログラム

```
var name;
var listResult;
var listCount;

function init(request) {
    var objCon;
    var code;

    //インスタンス生成
    objCon = new ControllingAreaController(); // 説明1

    // コンストラクタでのエラーチェック
    if(objCon.getErrCode() != "0"){
        //エラーメッセージ出力
        Debug.browse(objCon.getErrMsg());
    }

    //管理領域の一覧を取得
    listResult = objCon.getControllingAreaList(); // 説明2

    //管理領域の一覧から1件目の「管理領域コード」を取得
    code = listResult[0]["CO_AREA"];

    // 1件目のデータの管理領域名を取得
    name = objCon.getName(code); // 説明3

    //一覧の件数を取得
    listCount = listResult.length; // 説明4
}
```

- 管理領域取得 API のインスタンス生成(説明1)
ControllingAreaController クラスのインスタンスを生成します。引数は不要です。
インスタンスを生成すると、BAPI を呼出して管理領域を取得する処理が実行されます。
- 管理領域の一覧を取得(説明2)
ControllingAreaController クラスの getControllingAreaList メソッドを使用すると、管理領域の一覧を取得することが可能です。
- 1件目のデータの管理領域名を取得(説明3)
ControllingAreaController クラスの getName メソッドを使用すると、引数に指定した管理領域コードに対応する管理領域名を取得することが可能です。
該当するデータが存在しない場合は、何も返却されません。
- 一覧の件数を取得(説明4)
ControllingAreaController クラスの getControllingAreaList メソッドで取得した管理領域の一覧は、lengthで件数を取得することが可能です。

原価センタの情報を取得するためには、CostCenterController クラスの持つ各メソッドを利用する必要があります。
 引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
CostCenterController	コンストラクタ (引数無し)。 No4~18 のメソッドで値を設定し、No19 executeBapi メソッドで BAPI を呼び出し原価センタを取得する場合に使用します。
CostCenterController	コンストラクタ (第1引数あり)。 引数で指定した「管理領域コード」配下の原価センタの内、システム日付時点で有効な原価センタの情報のみ、No20~22 のメソッドにて取得することができるようになる。
CostCenterController	コンストラクタ (第1、第2引数あり)。 第1引数で指定した「管理領域コード」配下の原価センタの内、第2引数で指定した日付時点で有効な原価センタの情報のみ、No20~22 のメソッドにて取得することができるようになる。
setControllingarea	第1引数で指定した管理領域を設定する。
setCostcenterFrom	第1引数で指定した原価センタ (開始値) を設定する。
setCostcenterTo	第1引数で指定した原価センタ (終了値) を設定する。
setPersonInChargeFrom	第1引数で指定した責任者 (開始値) を設定する。
setPersonInChargeTo	第1引数で指定した責任者 (終了値) を設定する。
setDateFrom	第1引数で指定した有効期間 (開始値) を設定する。
setDateTo	第1引数で指定した有効期間 (終了値) を設定する。
setCostcentergroup	第1引数で指定した原価センタグループを設定する。
setControllingarea	第1引数で指定した事業領域を設定する。
setMasterDataInactive	第1引数で指定したマスタレコードの無効化を設定する。
setPersonInChargeUserFrom	第1引数で指定した担当ユーザ (開始値) を設定する。
setPersonInChargeUserTo	第1引数で指定した担当ユーザ (終了値) を設定する。
setTblParam	テーブル型の各項目に値を設定する。 第一引数にテーブル名、第二引数に項目名、第三引数に値、第四引数にレコード番号を設定する。
executeBapi	上記のsetメソッドで設定した値に紐づく原価センタを取得する為に BAPI を呼び出します。
getCostCenterList	コンストラクタまたは、上記のsetメソッドで設定した値に紐づく原価センタ情報(管理領域コード、原価センタコード、一般名称、テキスト)が格納された配列を取得する。
getName	引数で指定した「原価センタコード」に対応する原価センタの一般名称を取得する。
getDescript	引数で指定した「原価センタコード」に対応する原価センタのテキストを取得する。
getExtensionInList	BAPI パラメータ EXTENSIONIN の参照構造配列を取得する。
getExtensionOutList	BAPI パラメータ EXTENSIONOUT の参照構造を取得する。

サンプルプログラム

```

var name;
var descript;
var listResult;
var listCount;

function init(request) {
    var objCon;
    var code;

    //インスタンス生成
    objCon = new CostCenterController("1000"); // 説明1

    // コンストラクタでのエラーチェック
    if(objCon.getErrCode() != "0"){
        //エラーメッセージ出力
        Debug.browse(objCon.getErrMsg());
    }

    //原価センタの一覧を取得
    listResult = objCon.getCostCenterList(); // 説明3

    //原価センタの一覧から1件目の「原価センタコード」を取得
    code = listResult[0]["COST_CENTER"];

    // 1件目のデータの一般名称を取得
    name = objCon.getName(code); // 説明4

    // 1件目のデータのテキストを取得
    descript = objCon.getDescript(code); // 説明5

    //一覧の件数を取得
    listCount = listResult.length; // 説明6
}

```

- 原価センタ取得 API のインスタンス生成(説明1)

CostCenterController クラスのインスタンスを生成します。

インスタンスは、以下の3つのいずれかの方法で生成することができます。

インスタンスを生成すると、BAPI を呼出して原価センタを取得する処理が実行されます。

サンプルプログラム内では、管理領域コード「JP10」を直接記述していますが、管理領域取得APIにて取得した管理領域コードがセットされた変数を引数に用いれば、より効率的なプログラムを作成することが可能です。

1. システム日付時点で有効な原価センタを取得する場合

インスタンス（第1引数あり）を生成します。

第1引数で「管理領域コード」を指定します。

objCon = new CostCenterController ("JP10")は、管理領域コード「JP10」に紐付けられている原価センタの情報の内、システム日付時点で有効な情報のみを取得する、となります。

2. 引数で指定した日付時点で有効な原価センタを取得する場合

インスタンス（第1、第2引数あり）を生成します。

第1引数で「管理領域コード」、第2引数で「日付（YYYYMMDD）」を指定します。

もし、objCon = new CostCenterController ("JP10","20040930")とした場合は、管理領域コード「JP10」に紐付けられている原価センタの情報の内、2004年9月30日時点で有効な情報のみを取得する、となります。

3. 引数を指定しないで原価センタを取得する場合

引数なしインスタンスを生成します。

コンストラクタ生成後、管理領域や会社コード、原価センタなどをsetter メソッドにより設定します。

そして executeBapi メソッドにより BAPI を呼び出し、setter メソッドで設定した値に紐付けられている原価センタの情報を取得します。

- table の各項目に値を設定(説明2)

引数なしコンストラクタを生成し、各 setter メソッドで値を設定し executeBapi メソッドで BAPI を呼び出すことができます。

table パラメータの項目に値を設定する場合は、setTblParam メソッドを使用します。

第一引数のtable名称は固定値になっています。

後述のとおり名称を設定して下さい。

COSTCENTERLIST(原価センタ一覧) : CostCenter

EXTENSIONIN(BAPIパラメータ EXTENSIONIN) : ExtensionIn

第二引数の項目名称はSAPの各属性の項目名称を設定してください。

table 名称、項目名称は英字の大文字、小文字は厳密に区別されます。

第三引数に値、第四引数にレコード番号を設定します。

- 原価センタの一覧を取得(説明3)
CostCenterController クラスの getCostCenterList メソッドを使用すると、原価センタの一覧を取得することが可能です。ただし、CostCenterController クラスのインスタンスを生成する際に、引数で指定した管理領域コードに紐付けられている原価センタのみ一覧に含まれます（引数で指定したシステム日付時点で有効でない原価センタは対象外）。
- 1件目のデータの一般名称を取得(説明4)
CostCenterController クラスの getName メソッドを使用すると、引数に指定した原価センタコードに対応する一般名称を取得することが可能です。
該当するデータが存在しない場合は、何も返却されません。
- 1件目のデータのテキストを取得(説明5)
CostCenterController クラスの getDescript メソッドを使用すると、引数に指定した原価センタコードに対応するテキストを取得することが可能です。
該当するデータが存在しない場合は、何も返却されません。
- 一覧の件数を取得(説明6)
CostCenterController クラスの getCostCenterList メソッドで取得した原価センタの一覧は、length で件数を取得することが可能です。

利益センタ 取得 API (ProfitCenterController)

利益センタの情報を取得するためには、ProfitCenterController クラスの持つ各メソッドを利用する必要があります。

引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
ProfitCenterController	コンストラクタ（第1引数あり）。 引数で指定した「管理領域コード」配下の利益センタの内、システム日付時点で有効な利益センタの情報のみ、取得することができるようになる。
ProfitCenterController	コンストラクタ（第1、第2引数あり）。 第1引数で指定した「管理領域コード」配下の利益センタの内、第2引数で指定した日付時点で有効な利益センタの情報を取得することができるようになる。
ProfitCenterController	コンストラクタ（第1～4引数あり）。 第1引数で指定した「管理領域コード」の利益センタの内、第2引数で指定した「利益センタ責任者」、第3引数で指定した「利益センタに責任のあるユーザ」、第4引数で指定した「日付」に紐づく利益センタの情報を取得することができるようになる。
getProfitCenterList	コンストラクタにて指定した値に有効な、利益センタ情報(管理領域コード、利益センタコード、有効終了日、一般名称、責任者、利益センタの責任ユーザ)が格納された配列を取得する。
getValidTo	引数で指定した「利益センタコード」に対応する利益センタの有効終了日を取得する。
getPctrName	引数で指定した「利益センタコード」に対応する利益センタの一般名称を取得する。
getInCharge	引数で指定した「利益センタコード」に対応する利益センタの責任者を取得する。
getInChargeUser	引数で指定した「利益センタコード」に対応する利益センタに責任のあるユーザを取得する。

サンプルプログラム

```

var validTo;
var pctrName;
var inCharge;
var listResult;
var listCount;

function init(request) {
    var objCon;
    var code;

    //インスタンス生成
    objCon = new ProfitCenterController("JP10"); // 説明1

    // コンストラクタでのエラーチェック
    if (objCon.getErrCode() != "0"){
        //エラーメッセージ出力
        Debug.browse(objCon.getErrMsg());
    }

    //利益センタの一覧を取得
    list = objCon.getProfitCenterList(); // 説明2

    //利益センタの一覧から1件目の「利益センタコード」を取得
    code = list[0]["PROFIT_CTR"];

    // 1件目のデータの有効終了日を取得
    validTo = objCon.getValidTo(code); // 説明3

    // 1件目のデータの一般名称を取得
    pctrName = objCon.getPctrName(code); // 説明4

    // 1件目のデータの責任者を取得
    inCharge = objCon.getInCharge(code); // 説明5

    //一覧の件数を取得
    listCount = list.length; // 説明6
}

```

- 利益センタ取得 API のインスタンス生成(説明1)

ProfitCenterController クラスのインスタンスを生成します。

インスタンスは、以下の3つのいずれかの方法で生成することができます。

インスタンスを生成すると、BAPI を呼出して利益センタを取得する処理が実行されます。

サンプルプログラム内では、管理領域コード「JP10」を直接記述していますが、管理領域取得APIにて取得した管理領域コードがセットされた変数を引数に用いれば、より効率的なプログラムを作成することが可能です。

1. システム日付時点で有効な利益センタを取得する場合

インスタンス（第1引数あり）を生成します。

第1引数で「管理領域コード」を指定します。

objCon = new ProfitCenterController ("JP10")は、管理領域コード「JP10」に紐付けられている利益センタの情報の内、システム日付時点で有効な情報のみを取得する、となります。

2. 引数で指定した日付時点で有効な利益センタを取得する場合

インスタンス（第1、2引数あり）を生成します。

第1引数で「管理領域コード」、第2引数で「日付 (YYYYMMDD)」を指定します。

もし、objCon = new ProfitCenterController ("JP10","20040930")とした場合は、管理領域コード「JP10」に紐付けられている利益センタの情報の内、2004年9月30日時点で有効な情報のみを取得する、となります。

3. 引数で指定した管理領域、利益センタ責任者、利益センタに責任のあるユーザ、日付時点で、有効な利益センタを取得する場合

インスタンス（第1～4引数あり）を生成します。

第1引数で「管理領域コード」、第2引数で「利益センタ責任者」、第3引数で「利益センタに責任のあるユーザ」、第4引数で「日付 (YYYYMMDD)」を指定します。

もし、objCon = new ProfitCenterController ("JP10","ERP002","ERP003","20070930")とした場合は、管理領域コード「JP10」に紐付けられている利益センタの情報の内、2007年9月30日時点で有効な情報かつ利益センタ責任者「ERP002」、利益センタに責任のあるユーザ「ERP003」に紐づく情報を取得する、となります。

- 利益センタの一覧を取得

ProfitCenterController クラスの getProfitCenterList メソッドを使用すると、利益センタの一覧を取得することが可能です。ただし、ProfitCenterController クラスのインスタンスを生成する際に、引数で指定した管理領域コードに紐付けられている利益センタのみ一覧に含まれます（引数で指定した日付（またはシステム日付）時点で有効でない利益センタは対象外）。
- 1件目のデータの一般名称を取得

ProfitCenterController クラスの getPctrName メソッドを使用すると、引数に指定した利益センタコードに対応する一般名称を取得することが可能です。

該当するデータが存在しない場合は、何も返却されません。
- 1件目のデータの責任者を取得

ProfitCenterController クラスの getInCharge メソッドを使用すると、引数に指定した利益センタコードに対応する責任者を取得することが可能です。

該当するデータが存在しない場合は、何も返却されません。
- 一覧の件数を取得

ProfitCenterController クラスの getProfitCenterList メソッドで取得した利益センタの一覧は、length で件数を取得することが可能です。

会社 取得 API (CompanyController)

会社の情報を取得するためには、CompanyController クラスの持つ各メソッドを利用する必要があります。引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
CompanyController	コンストラクタ（引数無し）
getCompanyList	全ての会社情報(会社コード、会社名)が格納された配列を取得する。
getCompName	引数で指定した「会社コード」に対応する会社名を取得する。

サンプルプログラム

```

var name;
var listResult;
var listCount;

function init(request) {
    var objCon;
    var code;

    //インスタンス生成
    objCon = new CompanyController(); // 説明1

    // コンストラクタでのエラーチェック
    if(objCon.getErrCode() != "0"){
        //エラーメッセージ出力
        Debug.browse(objCon.getErrMsg());
    }

    //会社の一覧を取得
    listResult = objCon.getCompanyList(); // 説明2

    //会社の一覧から 1 件目の「会社コード」を取得
    code = listResult[0]["COMP_CODE"];

    // 1 件目のデータの会社名を取得
    name = objCon.getCompName(code); // 説明3

    //一覧の件数を取得
    listCount = listResult.length; // 説明4
}

```

- 会社取得 API のインスタンス生成(説明1)

CompanyController クラスのインスタンスを生成します。

引数は不要です。

インスタンスを生成すると、BAPIを呼出して会社を取得する処理が実行されます。

- 会社の一覧を取得(説明2)

CompanyController クラスの getCompanyList メソッドを使用すると、会社の一覧を取得することが可能です。

- 1 件目のデータの会社名を取得(説明3)

CompanyController クラスの getCompName メソッドを使用すると、引数に指定した会社コードに対応する会社名を取得することが可能です。

該当するデータが存在しない場合は、何も返却されません。

- 一覧の件数を取得(説明4)

CompanyController クラスの getCompanyList メソッドで取得した会社の一覧は、length で件数を取得することが可能です。

G/L 勘定 取得 API (GIAccController)

勘定科目の情報を取得するためには、GIAccController クラスの持つ各メソッドを利用する必要があります。

引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
GIAccController	コンストラクタ（第1引数あり）。 引数で指定した「会社コード」配下の勘定科目の内、言語が「JA」、かつ ISO 639 準拠の言語が「JA」である勘定科目の情報取得することができるようになる。
GIAccController	コンストラクタ（第1～3引数あり）。 第1引数で指定した会社コード、第2引数で指定した言語、第3引数で指定した ISO 639 準拠の言語で有効な勘定科目の情報を、No3～5 のメソッドにて取得することができるようになる。
getGIAccList	コンストラクタ（第1引数あり）の場合、指定した会社コード配下の勘定科目の内、言語が「JA」、かつ ISO 639 準拠の言語が「JA」である勘定科目情報(会社コード、G/L 勘定コード、テキスト(短)、テキスト(長))が格納された配列を取得する。 コンストラクタ（第1～3引数あり）にの場合、引数で指定した会社コード、言語、ISO 639 準拠の言語に紐づく勘定科目情報(会社コード、G/L 勘定コード、テキスト(短)、テキスト(長))が格納された配列を取得する。
getShortText	引数で指定した「G/L 勘定コード」に対応する勘定科目の G/L 勘定コードテキスト(短)を取得する。
getLongText	引数で指定した「G/L 勘定コード」に対応する勘定科目の総勘定元帳勘定テキスト(長)を取得する。

サンプルプログラム

```

var shortText;
var longText;
var list;
var listCount;

function init(request) {
  var objCon;
  var code;

  //インスタンス生成
  objCon = new GIAccController("1000"); // 説明1

  // コンストラクタでのエラーチェック
  if (objCon.getErrCode() != "0"){
    //エラーメッセージ出力
    Debug.browse(objCon.getErrMsg());
  }

  //G/L 勘定の一覧を取得
  list = objCon.getGIAccList(); // 説明2

  //G/L 勘定の一覧から1件目の「G/L 勘定コード」を取得
  code = list[0]["GL_ACCOUNT"];

  //1件目のデータのG/L 勘定コードテキスト(短)を取得
  shortText = objCon.getShortText(code); // 説明3

  //1件目のデータの総勘定元帳勘定テキスト(長)を取得
  longText = objCon.getLongText(code); // 説明4

  //一覧の件数を取得
  listCount = list.length; // 説明5
}

```

- G/L 勘定取得 API のインスタンス生成(説明1)

GIAccController クラスのインスタンスを生成します。

インスタンスは、以下の2つのいずれかの方法で生成することができます。

インスタンスを生成すると、BAPI を呼出して勘定科目を取得する処理が実行されます。

サンプルプログラム内では、会社コード「JP10」を直接記述していますが、会社取得 API にて取得した会社コードがセットされた変数を引数に用いれば、より効率的なプログラムを作成することが可能です。

1. 言語、ISO 639 準拠の言語を指定しないで勘定科目を取得する場合

インスタンス（第1引数あり）を生成します。

第1引数で「会社コード」を指定します。言語「JA」、ISO 639 準拠の言語「JA」を内部で設定し BAPI を呼び出します。

objCon = new GIAccController("JP10")は、会社コード「JP10」、言語コード「JA」、ISO 639 準拠の言語「JA」に紐付けられている G/L 勘定の情報を取得する、となります。

インスタンスを生成すると、BAPI を呼出して G/L 勘定を取得する処理が実行されます。

2. 言語、ISO 639 準拠の言語を指定して勘定科目を取得する場合

インスタンス（第1～3引数あり）を生成します。

第1引数で「会社コード」、第2引数で「言語」、第3引数で「ISO 639 準拠の言語」を指定します。

objCon = new GIAccController("JP10","EN","EN")は、会社コード「JP10」、言語コード「EN」、ISO 639 準拠の言語「EN」に紐付けられている G/L 勘定の情報を取得する、となります。インスタンスを生成すると、BAPI を呼出して G/L 勘定を取得する処理が実行されます。

- G/L 勘定の一覧を取得(説明2)

GIAccController クラスの getGIAccList メソッドを使用すると、G/L 勘定の一覧を取得することが可能です。

ただし、GIAccController クラスのインスタンスを生成する際に、引数で指定した会社コードに紐付けられている G/L 勘定のみ一覧に含まれます。

- 1件目のデータの G/L 勘定コードテキスト(短)を取得(説明3)

GIAccController クラスの getShortText メソッドを使用すると、引数に指定した G/L 勘定コードに対応する G/L 勘定コードテキスト(短)を取得することが可能です。

該当するデータが存在しない場合は、何も返却されません。

- 1件目のデータの総勘定元帳勘定テキスト(長)を取得(説明4)
GIAccController クラスの getLongText メソッドを使用すると、引数に指定した G/L 勘定コードに対応する総勘定元帳勘定テキスト(長)を取得することが可能です。
該当するデータが存在しない場合は、何も返却されません。
- 一覧の件数を取得(説明5)
GIAccController クラスの getGIAccList メソッドで取得した G/L 勘定の一覧は、length で件数を取得することが可能です。

銀行 取得 API (BankController)

銀行の情報を取得するためには、BankController クラスの持つ各メソッドを利用する必要があります。
引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
BankController	コンストラクタ (引数無し)。 銀行国コードが「JP」である銀行情報のみ取得することができるようになる。
BankController	コンストラクタ (第1、第2引数あり)。 第1引数で指定した銀行国コード、第2引数で指定した最大該当行数で有効な銀行情報を取得することができるようになる。
getBankList	銀行国コードが「JP」である銀行情報(銀行コード、銀行名、市区町村名)が格納された配列を取得する。
getBankName	引数で指定した「銀行コード」に対応する銀行名を取得する。
getCity	引数で指定した「銀行コード」に対応する市区町村名を取得する。

サンプルプログラム

```

var name;
var city;
var listResult;
var listCount;

function init(request) {
  var objCon;
  var code;

  //インスタンス生成
  objCon = new BankController(); // 説明1

  // コンストラクタでのエラーチェック
  if (objCon.getErrCode() != "0"){
    //エラーメッセージ出力
    Debug.browse(objCon.getErrMsg());
  }

  //銀行の一覧を取得
  list = objCon.getBankList(); // 説明2

  //銀行の一覧から1件目の「銀行コード」を取得
  code = list[0]["BANK_KEY"];

  //1件目のデータの銀行名を取得
  name = objCon.getBankName(code); // 説明3

  //1件目のデータの市区町村名を取得
  city = objCon.getCity(code); // 説明4

  //一覧の件数を取得
  listCount = list.length; // 説明5
}

```

- 銀行取得 API のインスタンス生成(説明1)

BankController クラスのインスタンスを生成します。

インスタンスは、以下の2つのいずれかの方法で生成することができます。

インスタンスを生成すると、BAPI を呼出して銀行情報を取得する処理が実行されます。

 1. 銀行国コード、最大該当行数を指定しないで銀行情報を取得する場合

インスタンス（引数なし）を生成します。

銀行国コード「JP」、最大該当行数ブランクを内部で設定しBAPI を呼び出します。

objCon = new BankController;は、銀行国コード「JP」に紐付けられている銀行情報を取得する、となります。

インスタンスを生成すると、BAPI を呼出して銀行情報を取得する処理が実行されます。

 2. 銀行国コード、最大該当行数を指定して銀行情報を取得する場合

インスタンス（第1、第2引数あり）を生成します。

第1引数で「銀行国コード」、第2引数で「最大該当行数」を指定します。

objCon = new BankController ("JP","100")は、銀行国コード「JP」、最大該当行数「100」に紐付けられている G/L 勘定の情報を取得する、となります。

インスタンスを生成すると、BAPIを呼出して銀行情報を取得する処理が実行されます。

最大該当行数は0以上の値を設定してください。

- 銀行の一覧を取得(説明2)

BankController クラスの getBankList メソッドを使用すると、銀行の一覧を取得することが可能です。

- 1 件目のデータの銀行名を取得(説明3)

BankController クラスの getBankName メソッドを使用すると、引数に指定した銀行コードに対応する銀行名を取得することが可能です。

該当するデータが存在しない場合は、何も返却されません。

- 1 件目のデータの市区町村名を取得(説明4)

BankController クラスの getCity メソッドを使用すると、引数に指定した銀行コードに対応する市区町村名を取得することが可能です。

該当するデータが存在しない場合は、何も返却されません。

- 一覧の件数を取得(説明5)

BankController クラスの getBankList メソッドで取得した銀行の一覧は、length で件数を取得することが可能です。

会計伝票 登録 API (AccEmployeePayPostController)

会計伝票情報を登録するためには、AccEmployeePayPostController クラスの持つ各メソッドを利用する必要があります。

引数、戻り値についての詳細な情報は、API リストを参照してください。

また、登録可能な項目は、会計伝票として必要最低限必要と判断される項目のみとなっています。

各メソッドの説明

コンストラクタ / メソッド名	説明
AccEmployeePayPostController	コンストラクタ（引数無し）
setDocObjType	伝票ヘッダの参照処理に引数の値を設定する。
setDocObjKey	伝票ヘッダのオブジェクトキーに引数の値を設定する。
setDocUsername	伝票ヘッダのユーザ名に引数の値を設定する。
setDocHeaderTxt	伝票ヘッダの伝票ヘッダテキストに引数の値を設定する。
setDocCompCode	伝票ヘッダの会社コードに引数の値を設定する。
setDocAcDocNo	伝票ヘッダの会計伝票番号に引数の値を設定する。
setDocFiscYear	伝票ヘッダの会計年度に引数の値を設定する。
setDocDocDate	伝票ヘッダの伝票の伝票日付に引数の値を設定する。

コンストラクタ / メソッド名	説明
setDocPstngDate	伝票ヘッダの伝票の転記日付に引数の値を設定する。
setDocFisPeriod	伝票ヘッダの会計期間に引数の値を設定する。
setDocDocType	伝票ヘッダの伝票タイプに引数の値を設定する。
setDocRefDocNo	伝票ヘッダの参照伝票番号に引数の値を設定する。
setPaltemnoAcc	第2引数で指定したレコード番号で、仕入先明細の会計伝票明細番号に第1引数の値を設定する。
setPaVendorNo	第2引数で指定したレコード番号で、仕入先明細の仕入先または債権者の勘定コードに第1引数の値を設定する。
setPaGIAccount	第2引数で指定したレコード番号で、仕入先明細の総勘定元帳勘定に第1引数の値を設定する。
setPaBlineDate	第2引数で指定したレコード番号で、仕入先明細の期日計算の支払基準日に第1引数の値を設定する。
setPaPymtMeth	第2引数で指定したレコード番号で、仕入先明細の支払方法に第1引数の値を設定する。
setPaAllocNمبر	第2引数で指定したレコード番号で、仕入先明細のソートキーに第1引数の値を設定する。
setPaltemText	第2引数で指定したレコード番号で、仕入先明細の明細テキストに第1引数の値を設定する。
setGIItemnoAcc	第2引数で指定したレコード番号で、G/L 勘定明細の会計伝票明細番号に第1引数の値を設定する。
setGIGIAccount	第2引数で指定したレコード番号で、G/L 勘定明細の総勘定元帳勘定に第1引数の値を設定する。
setGICompCode	第2引数で指定したレコード番号で、G/L 勘定明細の会社コードに第1引数の値を設定する。
setGIPstngDate	第2引数で指定したレコード番号で、G/L 勘定明細の伝票の転記日付に第1引数の値を設定する。
setGIDocType	第2引数で指定したレコード番号で、G/L 勘定明細の伝票タイプに第1引数の値を設定する。
setGIFiscYear	第2引数で指定したレコード番号で、G/L勘定明細の会計年度に第1引数の値を設定する。
setGIRefKey1	第2引数で指定したレコード番号で、G/L 勘定明細の取引先参照キーに第1引数の値を設定する。
setGIItemText	第2引数で指定したレコード番号で、G/L 勘定明細の明細テキストに第1引数の値を設定する。
setGITaxCode	第2引数で指定したレコード番号で、G/L 勘定明細の消費税コードに第1引数の値を設定する。
setGIAllocNمبر	第2引数で指定したレコード番号で、G/L 勘定明細のソートキーに第1引数の値を設定する。
setGIBusArea	第2引数で指定したレコード番号で、G/L勘定明細の事業領域に第1引数の値を設定する。
setGICostcenter	第2引数で指定したレコード番号で、G/L勘定明細の原価センタに第1引数の値を設定する。
setGIOrderid	第2引数で指定したレコード番号で、G/L勘定明細の指図番号に第1引数の値を設定する。
setGIWbsElement	第2引数で指定したレコード番号で、G/L勘定明細の WBS要素に第1引数の値を設定する。
setTItemnoAcc	第2引数で指定したレコード番号で、税明細の会計伝票明細番号に第1引数の値を設定する。

コンストラクタ / メソッド名	説明
setTaGlAccount	第2引数で指定したレコード番号で、税明細の総勘定元帳勘定に第1引数の値を設定する。
setTaTaxCode	第2引数で指定したレコード番号で、税明細の消費税コードに第1引数の値を設定する。
setCulItemnoAcc	第2引数で指定したレコード番号で、明細通貨項目の会計伝票明細番号に第1引数の値を設定する。
setCuCurrency	第2引数で指定したレコード番号で、明細通貨項目の通貨コードに第1引数の値を設定する。
setCuAmtDoccur	第2引数で指定したレコード番号で、明細通貨項目の伝票通貨額に第1引数の値を設定する。
setCuAmtBase	第2引数で指定したレコード番号で、明細通貨項目の課税基準額(伝票通貨建て)に第1引数の値を設定する。
setStrParam	構造型の各項目に値を設定する。 第1引数に structure 名、第2引数に項目名、第3引数に値を設定する。
setTblParam	テーブル型の各項目に値を設定する。 第1引数にテーブル名、第2引数に項目名、第3引数に値、第4引数にレコード番号を設定する。
executeBapi	setメソッドで設定した値を、接続先の SAP に登録する。登録に失敗した場合、登録処理はロールバックされる。
getObjType	参照処理を取得する。
getObjSys	参照システムを取得する。
getObjKey	参照キーを取得する。
getDocumentHeader	伝票ヘッダ情報を取得する。
getAccountPayableList	仕入先明細一覧を取得する。
getAccountGIList	G/L 勘定明細一覧を取得する。
getCurrencyAmountList	明細通貨項目一覧を取得する。
getTravelAmountList	旅費資金計画通貨項目一覧を取得する。
getTravelList	旅費資金予算明細一覧を取得する。
getExtension1List	カスタマ拡張テーブル一覧を取得する。

サンプルプログラム

```
function init(request) {
  var errList = [];
  var objCon;
  var objResult;

  //インスタンス生成
  objCon = new AccEmployeePayPostController(); // 説明1

  //伝票ヘッダ structure の各項目に値を設定 // 説明2
  objCon.setDocObjType("");
  objCon.setDocObjKey("");
  objCon.setDocUsername("ERP001");
  objCon.setDocHeaderTxt("イントラの伝票番号");
  objCon.setDocCompCode("1000");
  objCon.setDocAcDocNo("");
  objCon.setDocFiscYear("2006");
  objCon.setDocDocDate("20061205");
  objCon.setDocPstngDate("20061210");
  objCon.setDocFisPeriod("09");
  objCon.setDocDocType("AB");
  objCon.setDocRefDocNo("参照伝票番号");
}
```

```

//仕入先明細 table の各項目に値を設定
objCon.setPaltemnoAcc("0000000001", 1);
objCon.setPaVendorNo("0000100011", 1);
objCon.setPaGIAccount("0000412100", 1);
objCon.setPaBlineDate("20061210", 1);
objCon.setPaPymtMeth("T", 1);
objCon.setPaAllocNmbr("10001 ソートキー", 1);
objCon.setPaltemText("高橋啓介", 1);

//G/L 勘定明細 table の各項目に値を設定
objCon.setGlltemnoAcc("0000000002", 1);
objCon.setGIGIAccount("0000823100", 1);
objCon.setGICompCode("1000", 1);
objCon.setGIPstngDate("20061210", 1);
objCon.setGIDocType("AB", 1);
objCon.setGIFiscYear("2006", 1);
objCon.setGIRefKey1("取引参照キー", 1);
objCon.setGlltemText("費用勘定", 1);
objCon.setGITaxCode("V1", 1);
objCon.setGIAllocNmbr("ソートキー", 1);
objCon.setGIBusArea("1200", 1);
objCon.setGICostcenter("1220000000", 1);
objCon.setGIOrderid("000001000000", 1);
objCon.setGIWbsElement("100002", 1);

//税明細 table の各項目に値を設定
objCon.setTaltemnoAcc("0000000003", 1);
objCon.setTaGIAccount("0000148200", 1);
objCon.setTaTaxCode("V1", 1);

//明細通貨項目 table の各項目に値を設定 // 説明3
objCon.setCulitemnoAcc("0000000001", 1);
objCon.setCuCurrency("JPY", 1);
objCon.setCuAmtDoccur("-105000", 1);
objCon.setCuAmtBase("0", 1);
objCon.setCulitemnoAcc("0000000002", 2);
objCon.setCuCurrency("JPY", 2);
objCon.setCuAmtDoccur("100000", 2);
objCon.setCuAmtBase("0", 2);
objCon.setCulitemnoAcc("0000000003", 3);
objCon.setCuCurrency("JPY", 3);
objCon.setCuAmtDoccur("5000", 3);
objCon.setCuAmtBase("100000", 3);

//会計伝票の登録
objResult = objCon.executeBapi(); // 説明 4

//要素[ERR_CODE]に何らかの値がセットされている場合
if (objResult["ERR_CODE"] != null) {
    //エラー情報を出力する。
    Debug.browse(objResult);
}
}
}

```

- 会計伝票登録 API のインスタンス生成(説明1)
AccEmployeePayPostController クラスのインスタンスを生成します。
引数は不要です。
インスタンスを生成すると、会計伝票登録用 BAPI (BAPI_ACC_EMPLOYEE_PAY_POST) の import 型の structure、table パラメータに値を設定することができます。

- 伝票ヘッダ structure の各項目に値を設定
伝票ヘッダは structure パラメータになっています。
以下の2つのいずれかの方法で値を設定することができます。

1. 各項目の setter メソッドを使用する場合

setDoc・・・のメソッドを利用して、伝票ヘッダの各コンポーネントに値を設定します。

2. setStrParam メソッドする場合

第1引数の structure 名称は固定値になっています。

後述のとおり名称を設定して下さい。

第2引数の項目名称は SAP の各属性の項目名称を設定してください。

structure 名称、項目名称は英字の大文字、小文字は厳密に区別されます。第3引数に値を設定します。

SAP structure	セットする固有名称
DOCUMENTHEADER(伝票ヘッダ)	DocumentHeader

- 明細通貨項目 table の各項目に値を設定(説明3)
伝票ヘッダ以外の明細データは、複数件数を登録する場合もあるので、table パラメータになっています。
以下の2つのいずれかの方法で値を設定することができます。

1. 各テーブルの setter メソッドを使用する場合

tableパラメータに値を設定する場合は、objCon.setCulItemnoAcc("0000000001", 1)のように、第1引数に登録する値、第2引数にレコード番号を設定してください。

実際のシステム開発時には、objCon.setCulItemnoAcc(値を設定した属性, idx)のようにして For 文等のループ処理によってレコード番号をインクリメントして、第1引数で使用する属性の値も動的に設定する方法が望ましいでしょう。

2. setTblParam メソッドを使用する場合

第1引数の table 名称は固定値になっています。

後述のとおり名称を設定して下さい。

第2引数の項目名称は SAP の各属性の項目名称を設定してください。table 名称、項目名称は英字の大文字、小文字は厳密に区別されます。

第3引数に値、第4引数にレコード番号を設定します。

SAP table	セットする固有名称
ACCOUNTPAYABLE(仕入先明細)	AccountPayable
ACCOUNTGL(G/L 勘定明細)	AccountGl
ACCOUNTTAX(税明細)	AccountTax
CURRENCYAMOUNT(明細通貨項目)	CurrencyAmount
TRAVEL(旅費資金予算明細)	Travel
TRAVELAMOUNT(旅費資金予算通貨項目)	TravelAmount
EXTENSION1(カスタマ拡張テーブル)	Extension1

- 会計伝票の登録(説明4)
AccEmployeePayPostController クラスの executeBapi メソッドを実行すると、伝票ヘッダ structure、各明細 table にセットされた値が SAP 上に会計伝票データとして登録されます。
登録が成功した場合は、会計伝票登録 API の内部で自動的に commit 処理が実行されます。
登録に失敗した場合は、自動的に rollback処理が実行されます。
成功・失敗の基準は、会計伝票登録用BAPI (BAPI_ACC_EMPLOYEE_PAY_POST) の実行後に RETURN パラメータの TYPE 値を取得して、S (正常終了)、空文字以外の値が1つでも存在するかどうかです。
1つでも存在する場合は、rollback が自動的に実行されます。



注意

会計伝票登録 API に関して、structure、table パラメータを持つ各コンポーネントの必須項目や値の関連チェック等を確認する必要がある場合は、SAP上の汎用モジュールビルダ (t-cd : SE37) 等を使用して、「BAPI_ACC_EMPLOYEE_PAY_POST」の BAPI 仕様を確認してください。

管理会計業務向け API

管理会計業務向けAPIの各APIで使用するBAPI情報を定義したプロパティファイルが以下のフォルダに配備されています。

“WEB-INF/classes/jp/co/intra_mart/foundation/sap/module”

ファイル名を変更すると、APIを利用することができませんので、注意してください。

また、認証情報ファイル「sap_auth_info.properties」の接続先情報でservertypeを使用している場合は、各APIのプロパティファイルにservertypeを指定する必要がありますので、servertypeの設定を行ってください。
 管理会計業務向け APIの機能は以下となります。

原価センタグループ 取得 API(CostCenterGroupGetDetailController)

原価センタの情報を取得するためには、CostCenterGroupGetDetailController クラスの持つ各メソッドを利用する必要があります。
 引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
CostCenterGroupGetDetailController	コンストラクタ（第1、第2引数あり）。 第1引数で指定した「管理領域」配下の原価センタの内、第2引数で指定したグループ名で有効な原価センタの情報のみ取得することができるようになる。
CostCenterGroupGetDetailController	コンストラクタ（第1～4引数あり）。 第1引数で指定した「管理領域」配下の原価センタの内、第2引数で指定したグループ名、第3引数で指定した言語、第4引数で指定したISO 639 準拠の言語で有効な原価センタの情報のみ取得することができるようになる。
getHierarchyNodesList	階層ノード一覧を取得する。
getHierarchyValuesList	階層値一覧を取得する。
getLanguage	言語情報を取得する。

サンプルプログラム

```

var nodeList;
var nodeslistCount;
var valuesList;
var valueslistCount;

function init(request) {
var objCon;

//インスタンス生成
objCon = new CostCenterGroupGetDetailController("1000", "1000"); // 説明1

// コンストラクタでのエラーチェック
if(objCon.getErrCode() != "0"){
//エラーメッセージ出力
Debug.browse(objCon.getErrMsg());
}

//階層ノード一覧を取得
nodesList = objCon.getHierarchyNodesList(); // 説明2
nodeslistCount = nodeList.length; // 説明4

//階層値一覧を取得
valuesList = objCon.getHierarchyValuesList(); // 説明3
valueslistCount = valuesList.length; // 説明5
}
    
```

- 原価センタグループ API のインスタンス生成(説明1)
 CostCenterGroupGetDetailController クラスのインスタンスを生成します。
 インスタンスは、以下の2つのいずれかの方法で制することができます。
 インスタンスを生成すると、BAPI を呼出して原価センタグループを取得する処理が実行されます。
 1. 言語、ISO 639 準拠の言語を指定しないで原価センタグループを取得する場合
 インスタンス（第1、第2引数あり）を生成します。
 第1引数で「管理領域」、第2引数で「グループ名」を指定します。

objCon = new CostCenterGroupGetDetailController("1000", "2000");は、管理領域「1000」、グループ名「2000」に紐付けられている原価センタグループの情報を取得する、となります。

インスタンスを生成すると、BAPI を呼出して原価センタグループを取得する処理が実行されます。

2. 言語、ISO 639 準拠の言語を指定して原価センタグループを取得する場合

インスタンス（第1～4引数あり）を生成します。

第1引数で「管理領域」、第2引数で「グループ名」、第3引数で「言語」、第4引数で「ISO 639準拠の言語」を指定します。

objCon = new CostCenterGroupGetDetailController("1000", "2000", "JA", "JA");は、管理領域「1000」、グループ名「2000」、言語「JA」、ISO 639 準拠の言語「JA」に紐付けられている原価センタグループの情報を取得する、となります。

インスタンスを生成すると、BAPI を呼出して原価センタグループを取得する処理が実行されます。

- 階層ノード一覧を取得(説明2)

CostCenterGroupGetDetailController クラスの getHierarchyNodesList メソッドを使用すると、取得した原価センタの階層ノード一覧を取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

- 階層値一覧を取得(説明3)

CostCenterGroupGetDetailController クラスの getHierarchyValuesList メソッドを使用すると、取得した原価センタの階層値一覧を取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

- 階層ノード一覧の件数を取得(説明4)

CostCenterGroupGetDetailController クラスの getHierarchyNodesList メソッドで取得した原価センタの一覧は、length で件数を取得することが可能です。

- 階層値一覧の件数を取得(説明5)

CostCenterGroupGetDetailController クラスの getHierarchyValuesList メソッドで取得した原価センタの一覧は、length で件数を取得することが可能です。

仕入先詳細 取得 API(CreditorGetDetailController)

仕入先詳細の情報を取得するためには、CreditorGetDetailController クラスの持つ各メソッドを利用する必要があります。

引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
CreditorGetDetailController	コンストラクタ（第1、第2引数あり）。 第1引数で「仕入先」、第2引数で「会社コード」を設定した場合、引数に紐づく情報のみ取得することができるようになる。
getGeneralDataInfo	一般データを取得する。
getCompanyDataInfo	会社データを取得する。
getBankDataList	銀行データを取得する。

サンプルプログラム


```

var list;
var listCount;
// 中略

function init(request) {
    var objCon;
    var gene;
    var comp;
    // インスタンス生成
    objCon = new CreditorGetDetailController("0010000027", "1000"); // 説明1

    // コンストラクタでのエラーチェック
    if (objCon.getErrCode() != "0"){
        // エラーメッセージ出力
        Debug.browse(objCon.getErrMsg());
    }

    // 各データを取得
    gene = objCon.getGeneralDataInfo(); // 説明2
    comp = objCon.getCompanyDataInfo(); // 説明3

    //一般データを取得
    strVendor = gene.Vendor;
    strName = gene.NAME;
    strName_2 = gene.NAME_2;
    strName_3 = gene.NAME_3;
    strName_4 = gene.NAME_4;
    strCity = gene.CITY;
    strDistrict = gene.DISTRICT;
    strPo_box = gene.PO_BOX;
    strPobx_pcd = gene.POBX_PCD;
    strPostl_code = gene.POSTL_CODE;
    strRegion = gene.REGION;
    strStreet = gene.STREET;
    strCountry = gene.COUNTRY;
    strPobx_cty = gene.POBX_CTY;
    strLangu = gene.LANGU;

    //会社データを取得
    strComp_code = comp.COMP_CODE;
    strClerk = comp.CLERK;
    strHd_office = comp.HD_OFFICE;
    strAlt_payee = comp.ALT_PAYEE;
    strCuvd_clear = comp.CUVD_CLEAR;
    strPmnttrms = comp.PMNTTRMS;
    strAct_at_ven = comp.ACT_AT_VEN;
    strVend_user = comp.VEND_USER;
    strInternet = comp.INTERNET;
    strFax = comp.FAX;

    // 銀行データを取得
    list = objCon.getBankDataList(); // 説明4
    listCount = list.length; // 説明5
}

```

- 仕入先詳細 取得 API のインスタンス生成(説明1)

CreditorGetDetailController クラスのインスタンスを生成します。

第1引数に「仕入先」、第2引数に「会社コード」を設定します。

インスタンスを生成すると、BAPI を呼び出して仕入先詳細を取得する処理が実行されます。

- 一般データを取得(説明2)

CreditorGetDetailController クラスの getGeneralDataInfo メソッドを使用すると、取得した一般データを取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

- 会社データを取得(説明3)

CreditorGetDetailController クラスの getCompanyDataInfo メソッドを使用すると、取得した会社データを取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

- 会社データを取得(説明4)
CreditorGetDetailController クラスの getBankDataList メソッドを使用すると、取得した銀行データを取得することが可能です。
- 銀行データの件数を取得(説明5)
CreditorGetDetailController クラスの getBankDataList メソッドで取得した銀行データは、length で件数を取得することが可能です。

得意先詳細 取得 API(DebtorGetDetailController)

得意先詳細の情報を取得するためには、DebtorGetDetailController クラスの持つ各メソッドを利用する必要があります。
引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
DebtorGetDetailController	コンストラクタ（第1、第2引数あり）。 第1引数で「得意先コード」、第2引数で「会社コード」を設定した場合、引数に紐づく情報のみ取得することができるようになる。
getGeneralDataInfo	一般データを取得する。
getCompanyDataInfo	会社データを取得する。
getBankDataList	銀行データを取得する。

サンプルプログラム

```

var list;
var listCount;
// グローバル編集の中略

function init(request) {
    var objCon;
    var gene;
    var comp;

    //インスタンス生成
    objCon = new DebtorGetDetailController("T1000002", "2000"); // 説明 1

    // コンストラクタでのエラーチェック
    if (objCon.getErrCode() != "0"){
        // エラーメッセージ出力
        Debug.browse(objCon.getErrMsg());
    }

    // 各データを取得
    gene = objCon.getGeneralDataInfo(); //説明 2
    comp = objCon.getCompanyDataInfo(); // 説明 3

    //一般データ
    strCustomer = gene.CUSTOMER;
    strName = gene.NAME;
    strName_2 = gene.NAME_2;
    strName_3 = gene.NAME_3;
    strName_4 = gene.NAME_4;
    strCity = gene.CITY;
    strDistrict = gene.DISTRICT;
    strPo_box = gene.PO_BOX;
    strPobx_pcd = gene.POBX_PCD;
    strPostl_code = gene.POSTL_CODE;
    strRegion = gene.REGION;
    strStreet = gene.STREET;
    strCountry = gene.COUNTRY;
    strPobx_cty = gene.POBX_CTY;
    strLangu = gene.LANGU;

    //会社データ
    strComp_code = comp.COMP_CODE;
    strClerk = comp.CLERK;
    strHd_office = comp.HD_OFFICE;
    strAlt_payee = comp.ALT_PAYEE;
    strCuvd_clear = comp.CUVD_CLEAR;
    strPmnttrms = comp.PMNTTRMS;
    strAct_at_cus = comp.ACT_AT_CUS;
    strCust_user = comp.CUST_USER;
    strNext_payee = comp.NEXT_PAYEE;
    strBuying_grp = comp.BUYING_GRP;
    strInternet = comp.INTERNET;
    strFax = comp.FAX;

    // 銀行データをセット
    list = objCon.getBankDataList(); // 説明 4

    // 銀行データ
    listCount = list.length; //説明 5
}

```

- 得意先詳細 取得 API のインスタンス生成(説明1)
DebtorGetDetailControllerクラスのインスタンスを生成します。
第1引数に「得意先コード」、第2引数に「会社コード」を設定します。
インスタンスを生成すると、BAPI を呼び出して得意先詳細を取得する処理が実行されます。
- 一般データを取得(説明2)
DebtorGetDetailController クラスの getGeneralDataInfo メソッドを使用すると、取得した一般データを取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 会社データを取得(説明3)
DebtorGetDetailControllerクラスのgetCompanyDataInfoメソッドを使用すると、取得した会社データを取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

- 会社データを取得(説明4)
DebtorGetDetailController クラスのgetBankDataListメソッドを使用すると、取得した銀行データを取得することが可能です。
- 銀行データの件数を取得(説明5) | DebtorGetDetailController クラスの getBankDataList メソッドで取得した銀行データは、length で件数を取得することが可能です。

為替換算レート 取得 API(ExchangeRateGetDetailController)

。為替換算レートの情報を取得するためには、ExchangeRateGetDetailController クラスの持つ各メソッドを利用する必要があります。引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
ExchangeRateGetDetailController	コンストラクタ (第1、第2引数あり)。 第1引数で「換算レートタイプ」、第2引数で「換算前通貨」、第3引数で「換算後通貨」、第4引数で「起算日」の設定を行った場合、引数に紐づく情報のみ取得することができるようになる。
getExchRate	換算レート、係数 (通貨ペア)を取得します。

サンプルプログラム

```

var nodeList;
var nodeslistCount;
var valuesList;
var valueslistCount;

function init(request) {
var objCon;

//インスタンス生成
objCon = new CostCenterGroupGetDetailController("1000", "1000"); // 説明 1

// コンストラクタでのエラーチェック
if(objCon.getErrCode() != "0"){
//エラーメッセージ出力
Debug.browse(objCon.getErrMsg());
}

//階層ノード一覧を取得
nodeList = objCon.getHierarchyNodesList(); // 説明 2
nodeslistCount = nodeList.length; // 説明 4

//階層値一覧を取得
valuesList = objCon.getHierarchyValuesList(); // 説明 3
valueslistCount = valuesList.length; // 説明 5
}

```

- 為替換算レート取得 API のインスタンス生成
ExchangeRateGetDetailController クラスのインスタンスを生成します。
第1引数に照会する換算レートタイプ、第2引数に換算前通貨、第3引数に換算後通貨、第4引数に起算日を設定します。
インスタンスを生成すると、BAPI を呼び出して為替換算レートを取得する処理が実行されます。
- 為替換算レート一覧を取得
ExchangeRateGetDetailController クラスの getExchRate メソッドを使用すると、取得した為替換算レート一覧を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

内部指図書詳細 取得 API(InternalOrderGetDetailController)

内部指図書詳細の情報を取得するためには、InternalOrderGetDetailController クラスの持つ各メソッドを利用する必要があります。

引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
InternalOrderGetDetailController	コンストラクタ（第1、第2引数あり）。 第1引数で「指図番号」、第2引数で「ステータスおよび業務トランザクションテキストの言語」（任意入力）を設定した場合、引数に紐付く情報のみ取得することができるようになる。
getMasterData	指図のマスタデータ構造を取得する。
getSystemStatusList	システムステータステーブルを取得する。
getUserStatusList	ユーザステータス/指図ステータステーブルを取得する。
getAllowedBusTractList	使用可能業務トランザクションのテーブルを取得する。

サンプルプログラム

```

var nodeList;
var nodeslistCount;
var valuesList;
var valueslistCount;

function init(request) {
var objCon;

//インスタンス生成
objCon = new CostCenterGroupGetDetailController("1000", "1000"); // 説明 1

// コンストラクタでのエラーチェック
if(objCon.getErrCode() != "0"){
//エラーメッセージ出力
Debug.browse(objCon.getErrMsg());
}

//階層ノード一覧を取得
nodeList = objCon.getHierarchyNodesList(); // 説明 2
nodeslistCount = nodeList.length; // 説明 4

//階層値一覧を取得
valuesList = objCon.getHierarchyValuesList(); // 説明 3
valueslistCount = valuesList.length; // 説明 5
}

```

- 内部指図書詳細 取得 API のインスタンス生成(説明1)
 InternalOrderGetDetailController クラスのインスタンスを生成します。
 第1引数に「指図番号」、第2引数に「ステータスおよび業務トランザクションテキストの言語」を設定します。インスタンスを生成すると、BAPI を呼び出して内部指図書詳細を取得する処理が実行されます。
- マスタデータ構造を取得(説明2)
 InternalOrderGetDetailController クラスの getMasterData メソッドを使用すると、取得した指図のマスタデータ構造を取得することが可能です。
 項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- システムステータステーブルを取得(説明3)
 InternalOrderGetDetailController クラスの getSystemStatusList メソッドを使用すると、取得したシステムステータステーブルを取得することが可能です。
 項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- ユーザステータス/指図ステータステーブルを取得(説明4)
 InternalOrderGetDetailController クラスのgetUserStatusList メソッドを使用すると、取得したユーザステータス/指図ステータステーブルを取得することが可能です。
- 使用可能業務トランザクションのテーブルを取得(説明5)
 InternalOrderGetDetailController クラスの getAllowedBusTractList メソッドを使用すると、取得した使用可能業務トランザク

ションのテーブルを取得することが可能です。

- システムステータステーブルの件数を取得(説明6)
InternalOrderGetDetailController クラスの getSystemStatusList メソッドで取得したシステムステータステーブルは、length で件数を取得することが可能です。
- ユーザステータス/指図ステータステーブルの件数を取得(説明7)
InternalOrderGetDetailController クラスの getUserStatusList メソッドで取得したユーザステータス/指図ステータステーブルは、length で件数を取得することが可能です。
- 使用可能業務トランザクションのテーブルの件数を取得(説明8)
InternalOrderGetDetailController クラスの getAllowedBusTractList メソッドで取得した使用可能業務トランザクションのテーブルは、length で件数を取得することが可能です。intra-mart SAP ERP6.0 連携

内部指図書（一覧）取得 API(InternalOrderGetListController)

内部指図書（一覧）の情報を取得するためには、InternalOrderGetListController クラスの持つ各メソッドを利用する必要があります。引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
InternalOrderGetListController	コンストラクタ（第1、第2引数あり）。 第1引数に「管理領域」、第2引数に「責任原価センタ」、第3引数に「指図タイプ」、第4引数に「指図番号（個別値、または下限値）」、第5引数に「指図番号（上限値）」第6引数に「外部指図番号（個別値、または下限値）」第7引数に「外部指図番号（上限値）」を設定した場合、引数に紐づく情報のみ取得することができるようになる。
getOrderListList	選択指図一覧を取得する。

サンプルプログラム

```

var list;
var count;

function init(request) {

    var objCon;

    //インスタンス生成
    objCon = new InternalOrderGetListController("1000", "", "0100", "", "", "", ""); //説明1

    //コンストラクタでのエラーチェック
    if (objCon.getErrCode() != "0"){
        //エラーメッセージ出力
        Debug.browse(objCon.getErrMsg());
    }

    //選択指図一覧をセット
    list = objCon.getOrderListList(); //説明2
    count = list.length; //説明3
}

```

- 内部指図書（一覧）取得 API のインスタンス生成(説明1)
InternalOrderGetListControllerクラスのインスタンスを生成します。
第1引数に「管理領域」、第2引数に「責任原価センタ」、第3引数に「指図タイプ」、第4引数に「指図番号（個別値、または下限値）」、第5引数に「指図番号（上限値）」第6引数に「外部指図番号（個別値、または下限値）」第7引数に「外部指図番号（上限値）」を設定します。
インスタンスを生成すると、BAPI を呼び出して内部指図書（一覧）を取得する処理が実行されます。
- 内部指図書（一覧）を取得(説明2)
InternalOrderGetListControllerクラスのgetOrderListListメソッドを使用すると、取得した内部指図書一覧を取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

- 内部指図書（一覧）の件数を取得 | InternalOrderGetListController クラスの getOrderListList メソッドで取得した内部指図書（一覧）は、lengthで件数を取得することが可能です。

利益センタグループ 取得 API(ProfitCenterGrpGetDetailController)

利益センタの情報を取得するためには、ProfitCenterGrpGetDetailController クラスの持つ各メソッドを利用する必要があります。引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
ProfitCenterGrpGetDetailController	コンストラクタ（第1、第2引数あり）。 第1引数で指定した「管理領域」配下の原価センタの内、第2引数で指定したグループ名で有効な利益センタの情報のみ取得することができるようになる。
getHierarchyNodesList	階層ノード一覧を取得する。
getHierarchyValuesList	階層値一覧を取得する。

サンプルプログラム

```

var nodeList;
var nodeslistCount;
var valuesList;
var valueslistCount;

function init(request) {

    var objCon;

    //インスタンス生成
    objCon = new ProfitCenterGrpGetDetailController("1000", "1000"); //説明 1

    // コンストラクタでのエラーチェック
    if (objCon.getErrCode() != "0"){
        //エラーメッセージ出力
        Debug.browse(objCon.getErrMsg());
    }

    //階層ノード一覧
    nodeList = objCon.getHierarchyNodesList(); //説明 2

    //階層値一覧
    valueslist = objCon.getHierarchyValuesList(); //説明 3

    //階層ノード一覧を取得
    nodeslistCount = nodeList.length; //説明 4

    //階層値一覧を取得
    valueslistCount = valuesList.length; //説明 5
}

```

- 利益センタグループ API のインスタンス生成(説明1)
ProfitCenterGrpGetDetailController クラスのインスタンスを生成します。
第1引数に照会する管理領域、第2引数にグループ名を設定します。インスタンスを生成すると、BAPI を呼び出して利益センタグループを取得する処理が実行されます。
- 階層ノード一覧を取得(説明1)
ProfitCenterGrpGetDetailController クラスの getHierarchyNodesList メソッドを使用すると、取得した利益センタの階層ノード一覧を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 階層値一覧を取得(説明1) | ProfitCenterGrpGetDetailController クラスの getHierarchyValuesList メソッドを使用すると、取得し

た利益センタの階層値一覧を取得することが可能です。 | 項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

- 階層ノード一覧の件数を取得(説明1) | ProfitCenterGrpGetDetailController クラスの getHierarchyNodesList メソッドで取得した利益センタの一覧は、length で件数を取得することが可能です。
- 階層値一覧の件数を取得(説明1) | ProfitCenterGrpGetDetailController クラスの getHierarchyValuesList メソッドで取得した利益センタの一覧は、length で件数を取得することが可能です。

WBS 情報 取得 API(ProjectGetInfoController)

WBS情報を取得するためには、ProjectGetInfoControllerクラスの持つ各メソッドを利用する必要があります。
引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
ProjectGetInfoController	コンストラクタ (引数無し)
setProjectDefinition	プロジェクト定義識別キーに値を設定する。第1引数にプロジェクト定義を識別するキーを設定します。
setTblParam	テーブル型の各項目に値を設定する。第1引数にテーブル名、第2引数に項目名、第3引数に値、第4引数にレコード番号を設定する。
executeBapi	setProjectDefinition、及びのsetTblParamメソッドで設定した値を、接続先の SAP に登録する。 登録に失敗した場合、登録処理はロールバックされる。
getProjectDefinitionInfo	プロジェクトについての詳細情報を取得する。
getExpWbsElementList	WBS 要素についての詳細情報を取得する。
getExpWbsMilestoneList	WBS マイルストーンに関する詳細情報を取得する。
getExpWbsHierarchieList	WBS 階層についての情報を取得する。
getExpActivityList	活動に関する詳細情報を取得する。

サンプルプログラム


```
// グローバル変数 省略
```

```
function init(request) {

    var objCon;
    var objResult;
    var project;

    //インスタンス生成
    //伝票ヘッダ structure の各項目に値を設定
    objCon = new ProjectGetInfoController(); //説明1

    //プロジェクトの定義
    objCon.setProjectDefinition("2"); //説明2
    //従属活動を設定します。
    objCon.setWithActivities("X");
    //従属 WBS マイルストーンを設定します。
    objCon.setWithMilestones("X");
    //サブプロジェクトを設定します。
    //objCon.setWithSubtree("X");
    //テーブル型の項目に値をセット
    //objCon.setTblParam("ImpWbsElement","wbs_element","21",1); //説明3
    //検索実行
    objResult = objCon.executeBapi(); //説明4

    if(objResult["ERR_CODE"] != null){
        //エラーメッセージ出力
        Debug.browse(objResult);
    }

    //オブジェクトをグローバル領域に保存
    project = objCon.getProjectDefinitionInfo(); //説明5

    //WBS 要素についての詳細情報をセットする
    elementlist = objCon.getExpWbsElementList(); //説明6
    elementlistCount = elementlist.length; //説明10

    //WBS マイルストーンに関する詳細情報をセットする
    milelist = objCon.getExpWbsMilestoneList(); //説明7
    milelistCount = milelist.length; //説明11

    //WBS 階層についての情報をセットする
    hialist = objCon.getExpWbsHierarchieList(); //説明8
    hialistCount = hialist.length; //説明12

    //活動に関する詳細情報をセットする
    activelist = objCon.getExpActivityList(); //説明9
    activelistCount = activelist.length; //説明13

}
```

- WBS 情報取得 API のインスタンス生成(説明1)
ProjectGetInfoController クラスのインスタンスを生成します。
引数は不要です。
インスタンスを生成すると、BAPI を呼び出して WBS 情報を取得する処理が実行されます。
- プロジェクトの定義を設定(説明2)
プロジェクト定義に識別キーを設定する場合は、setProjectDefinition メソッドを使用します。
引数に識別キーを設定します。
- table に値を設定(説明3)
tableパラメータの項目に値を設定する場合は、setTblParam メソッドを使用します。
第1引数にテーブル名、第2引数に項目名、第3引数にセットする値、第4引数にレコード番号を設定します。
- 検索の実行(説明4)
ProjectGetInfoController クラスの execute メソッドを使用すると、各項目に設定した条件に紐づくデータのみを取得します。
- プロジェクトについての詳細情報を取得(説明5)
ProjectGetInfoController クラスの getProjectDefinitionInfo メソッドを使用すると、プロジェクトについての詳細情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

- WBS 要素についての詳細情報を取得(説明6)
 ProjectGetInfoController クラスの getExpWbsElementList メソッドを使用すると、WBS 要素についての詳細情報を取得することが可能です。
 項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- WBS マイルストーンに関する詳細情報を取得(説明7)
 ProjectGetInfoController クラスの getExpWbsMilestoneList メソッドを使用すると、WBS マイルストーンに関する詳細情報を取得することが可能です。
 項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- WBS 階層についての情報を取得(説明8)
 ProjectGetInfoController クラスの getExpWbsHierarchieList メソッドを使用すると、WBS 階層についての情報を取得することが可能です。
 項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 活動に関する詳細情報を取得(説明9)
 ProjectGetInfoController クラスの getExpActivityList メソッドを使用すると、活動に関する詳細情報を取得することが可能です。
 項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- WBS 要素についての詳細情報の件数を取得(説明10)
 ProjectGetInfoController クラスの getExpWbsElementList メソッドで取得した WBS 要素についての詳細情報は、length で件数を取得することが可能です。
- WBS マイルストーンに関する詳細情報の件数を取得(説明11)
 ProjectGetInfoController クラスの getExpWbsMilestoneList メソッドで取得した WBS マイルストーンに関する詳細情報は、length で件数を取得することが可能です。
- WBS 階層についての情報の件数を取得(説明12)
 ProjectGetInfoController クラスの getExpWbsHierarchieList メソッドで取得した WBS 階層についての情報は、length で件数を取得することが可能です。
- 活動に関する詳細情報の件数を取得(説明13)
 ProjectGetInfoController クラスの getExpActivityList メソッドで取得した活動に関する詳細情報は、length で件数を取得することが可能です。

在庫/購買管理業務向け API

在庫/購買管理業務向けAPIの各APIで使用するBAPI情報を定義したプロパティファイルが以下のフォルダに配備されています。
 “WEB-INF/classes/jp/co/intra_mart/foundation/sap/module”

ファイル名を変更すると、APIを利用することができませんので、注意してください。
 また、認証情報ファイル「sap_auth_info.properties」の接続先情報でservertimeを使用している場合は、各APIのプロパティファイルにservertimeを指定する必要がありますので、servertimeの設定を行ってください。
 財務会計業務向け APIの機能は以下となります。

入出庫伝票 登録 API (GoodsMvtCreateController)

入出庫伝票を登録するためには、GoodsMvtCreateController クラスの持つ各メソッドを利用する必要があります。
 引数、戻り値についての詳細な情報は、API リストを参照してください。
 また、登録可能な項目は、入出庫伝票として必要最低限必要と判断される項目のみとなっています。

各メソッドの説明

コンストラクタ / メソッド名	説明
GoodsMvtCreateController	コンストラクタ (引数無し)
GoodsMvtCreateController	コンストラクタ (第1引数あり)。ConnectionManager を使う場合に使用する。第1引数で「ConnectionManager」を設定する。
GoodsMvtCreateController	コンストラクタ (第1引数あり)。SAPClient の接続情報を使う場合に使用する。第1引数で「SAPClient」 (オブジェクト) を設定する。
setTestrun	import 項目 : 在庫移動シミュレーションを設定します。
setStrParam	構造型の各項目に値を設定する。第1引数に structure名、第2引数に項目名、第3引数に値を設定する。

コンストラクタ / メソッド名	説明
setTblParam	テーブル型の各項目に値を設定する。第1引数にテーブル名、第2引数に項目名、第3引数に値、第4引数にレコード番号を設定する。
executeBapi	BapiController により入庫伝票を登録します。
getBapiReturnErrMsg	BAPI実行結果であるBapiReturnのType情報が正常(デフォルト値:"S"/"")以外の BapiReturn オブジェクトのメッセージ情報を全て返します。 全て正常の場合は null が返却されます。
getBapiReturnMessage	BAPI実行結果である BapiReturn の Type 情報が正常(デフォルト値: "S" / "")と判断された Type 情報のメッセージ情報を全て返します。 メッセージが存在しない場合は null が返却されます。
getTestrun	import 項目 : 在庫移動シミュレーションを取得します。
getMatdocumentyear	export 項目 : 入庫伝票会計年度を取得します。
getMaterialdocument	export 項目 : 入庫伝票番号を取得します。
getGoodsMvtCode	import 項目 : 設定した在庫移動に対するコードのトランザクションへの割当情報を取得します。
getGoodsMvtHeader	import 項目 : 設定した入庫伝票ヘッダデータ情報を取得します。
getGoodsMvtRefEwm	import 項目 : BAdl 通信構造: 入庫伝票 EWM 照会情報を取得します。
getGoodsMvtHeadret	export 項目 : 設定した入庫伝票番号/入庫伝票会計年度情報を取得します。
getGoodsMvtItemList	import 項目 : 設定した入庫伝票明細情報を取得します。
getGoodsMvtSerialNumberList	import 項目 : 設定したシリアル番号情報を取得します。
getGoodsMvtServPartDataList	import 項目 : BAPI 通信構造: サービスパーツデータ情報を取得します。
getExtensioninList	Import 項目 : BAPI パラメータ ExtensionIn の参照構造情報を取得します。

サンプルプログラム

```

//■入庫伝票登録
var materialDocument;
var matDocumentYear;
var goodsMvtHeadret;
var ret;

function init(request) {

//-----
//変数宣言
//-----
var objCon;
var objResult;
var objMsg; // executeBapi実行結果メッセージ

//-----
//インスタンス生成
//-----
objCon = new GoodsMvtCreateController(); // 説明 1

//-----
// テスト実行
//-----
objCon.setTestrun(""); //在庫移動シミュレーション // 説明 2

//-----
// 構造体項目セット // 説明 3
//-----
objCon.setStrParam("GoodsMvtHeader","PSTNG_DATE","20061201",1); //伝票の転記日付
objCon.setStrParam("GoodsMvtCode","GM_CODE","01",1); //BAPI 在庫移動トランザクションのコード割当

//-----
// テーブル項目セット // 説明 4
//-----
objCon.setTblParam("GoodsMvtItem","MOVE_TYPE","101",1); //移動タイプ (在庫管理)
objCon.setTblParam("GoodsMvtItem","ENTRY_QNT","1",1); //入力単位での数量
objCon.setTblParam("GoodsMvtItem","PO_NUMBER","4500000015",1); //購買発注番号
objCon.setTblParam("GoodsMvtItem","PO_ITEM","00010",1); //購買伝票の明細番号
objCon.setTblParam("GoodsMvtItem","MVT_IND","B",1); //移動区分

//-----
// executeBapi実行
//-----
objResult = objCon.executeBapi(); // 説明 5

//-----
// エラー判定
//-----
if (objResult["ERR_CODE"] != null){
    Debug.browse(objResult);
}

//入出庫伝票 // 説明 6
materialDocument = objCon.getMaterialdocument();
matDocumentYear = objCon.getMatdocumentyear();
goodsMvtHeadret = objCon.getGoodsMvtHeadret();

//API実行結果を取得します。
objMsg = objCon.getBapiReturnMessage();
Debug.print("objMsg=[" + objMsg + "]);
ret = objMsg;
}

```

- 入出庫伝票登録 API のインスタンス生成(説明1)
GoodsMvtCreateControllerクラスのインスタンスを生成します。
引数は不要です。
インスタンスを生成すると、入出庫伝票登録BAPI (BAPI_GOODSMVT_CREATE) のimport型のimport、structure、table パラメータに値を設定することができます。

- import の各項目に値を設定(説明2)
import パラメータの項目に値を設定する場合は、set メソッドを使用します。
メソッドについては各 API のAPI リストを参照ください。



コラム

サンプルプログラムでは import セットは行っていません。

- structure の各項目に値を設定(説明3)
structure パラメータの項目に値を設定する場合は、setStrParam メソッドを使用します。
第1引数のstructure 名称は固定値になっています。
後述のとおり名称を設定して下さい。
第2引数の項目名称はSAPの各属性の項目名称を設定してください。
structure 名称、項目名称は英字の大文字、小文字は厳密に区別されます。
第3引数に値を設定します。

SAP structure	セットする固有名称
GOODSMVT_CODE (在庫移動に対するコードのトランザクションへの割当)	GoodsMvtCode
GOODSMVT_HEADER 入出庫伝票ヘッダデータ	GoodsMvtHeader
GOODSMVT_REF_EWM (BAdI 通信構造: 入出庫伝票 EWM 照会)	GoodsMvtRefEwm

- table の各項目に値を設定(説明4)
table パラメータの項目に値を設定する場合は、setTblParam メソッドを使用します。
第1引数のtable名称は固定値になっています。
後述のとおり名称を設定して下さい。
第2引数の項目名称は SAP の各属性の項目名称を設定してください。
table 名称、項目名称は英字の大文字、小文字は厳密に区別されます。
第3引数に値、第四引数にレコード番号を設定します。

SAP structure	セットする固有名称
GOODSMVT_ITEM (入出庫伝票明細)	GoodsMvtItem
GOODSMVT_SERIALNUMBER (シリアル番号)	GoodsMvtSerialNumber
GOODSMVT_SERV_PART_DATA (BAPI通信構造:サービスパーツデータ情報)	GoodsMvtServPartData
EXTENSIONIN (BAPI パラメータ ExtensionInの参照構造)	Extensionin

- 入出庫伝票の登録(説明5)
GoodsMvtCreateController クラスの executeBapi メソッドを実行すると、各項目にセットされた値が SAP上に入出庫伝票データとして登録されます。
登録が成功した場合には、入出庫伝票登録APIの内部で自動的に commit 処理が実行されます。
登録に失敗した場合には、自動的に rollback 処理が実行されます。
成功・失敗の基準は、入出庫伝票登録用 BAPI (BAPI_GOODSMVT_CREATE) の実行後に RETURNパラメータの TYPE 値を取得して、S (正常終了)、空文字以外が1つでも存在するかどうかです。
1つでも存在する場合は、rollback が自動的に実行されます。

- サンプルプログラムの中で記載されたデータ取得メソッドの詳細説明については、以下を参照してください。(説明6)
 - Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn を取得
GoodsMvtCreateController クラスの getBapiReturnErrMessage メソッドを使用すると、取得したBapiReturn の Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn オブジェクトのメッセージ情報を取得することが可能です。全て正常の場合は null を取得します。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - Type 情報が正常(デフォルト値: "S" / "")の BapiReturn を取得
GoodsMvtCreateController クラスの getBapiReturnMessage メソッドを使用すると、取得した BapiReturnのType情報が正常(デフォルト値: "S" / "")と判断された Type情報のメッセージ情報を取得することが可能です。
メッセージが存在しない場合は null を取得します。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 在庫移動シミュレーション情報を取得

GoodsMvtCreateController クラスの getTestrund メソッドを使用すると、設定した在庫移動シミュレーション情報を取得することが可能です。

- 入出庫伝票会計年度情報を取得
GoodsMvtCreateController クラスの getMatdocumentyear メソッドを使用すると、取得した入出庫伝票会計年度情報を取得することが可能です。
- 入出庫伝票番号情報を取得
GoodsMvtCreateController クラスの getMaterialdocument メソッドを使用すると、取得した入出庫伝票番号情報を取得することが可能です。
- 在庫移動に対するコードのトランザクションへの割当情報を取得
GoodsMvtCreateController クラスの getGoodsMvtCode メソッドを使用すると、設定した在庫移動に対するコードのトランザクションへの割当情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 入出庫伝票ヘッダデータ情報を取得
GoodsMvtCreateController クラスの getGoodsMvtHeader メソッドを使用すると、設定した入出庫伝票ヘッダデータ情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 入出庫伝票番号/入出庫伝票会計年度情報を取得
GoodsMvtCreateController クラスの getGoodsMvtHeadret メソッドを使用すると、取得した入出庫伝票番号/入出庫伝票会計年度情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- BAdI 通信構造: 入出庫伝票 EWM 照会情報を取得
GoodsMvtCreateController クラスの getGoodsMvtRefEwm メソッドを使用すると、取得した BAdI 通信構造: 入出庫伝票 EWM 照会情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 入出庫伝票明細情報を取得
GoodsMvtCreateController クラスの getGoodsMvtItemList メソッドを使用すると、設定した入出庫伝票明細情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- シリアル番号情報を取得
GoodsMvtCreateController クラスの getGoodsMvtSerialNumberList メソッドを使用すると、設定したシリアル番号情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- BAPI 通信構造: サービスパーツデータ情報を取得
GoodsMvtCreateController クラスの getGoodsMvtServPartDataList メソッドを使用すると、設定した BAPI 通信構造: サービスパーツデータ情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- BAPI パラメータ ExtensionIn の参照構造情報を取得
GoodsMvtCreateController クラスの getExtensioninList メソッドを使用すると、設定した BAPI パラメータ ExtensionIn の参照構造情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

コラム

サンプルプログラム実行結果では、SAP テスト環境のデータが取得されています。SAP 標準では該当データが存在しません。

注意

入出庫伝票登録 API に関して、import、structure、table パラメータを持つ各コンポーネントの必須項目や値の相関チェック等を確認する必要がある場合は、SAP 上の汎用モジュールビルダ (t-cd : SE37) 等を使用して、「BAPI_GOODSMVT_CREATE」の BAPI 仕様を確認してください。

入出庫伝票 一覧 API (GoodsMvtGetItemsController)

入出庫伝票を検索するためには、GoodsMvtGetItemsController クラスの持つ各メソッドを利用する必要があります。引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
GoodsMvtGetItemsController	コンストラクタ (引数無し)
GoodsMvtGetItemsController	コンストラクタ (第 1 引数あり)。 ConnectionManager を使う場合に使用する。第 1 引数で「ConnectionManager」を設定する。
GoodsMvtGetItemsController	コンストラクタ (第 1 引数あり)。 SAPClient の接続情報を使う場合に使用する。第 1 引数で「SAPClient」(オブジェクト)を設定する。
setTblParam	テーブル型の各項目に値を設定する。 第1引数にテーブル名、第2引数に項目名第3引数に値、第4引数にレコード番号を設定する。
executeBapi	BapiController により入出庫伝票一覧を取得します。
getBapiReturnErrMsg	BAPI実行結果である BapiReturn の Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn オブジェクトのメッセージ情報を全て返します。 全て正常の場合は null が返却されます。
getBapiReturnMessage	BAPI実行結果である BapiReturn の Type 情報が正常(デフォルト値: "S" / "")と判断された Type 情報のメッセージ情報を全て返します。 メッセージが存在しない場合は null が返却されます。
getBatchRaList	ロットの範囲情報を取得します。
getGoodsMvtHeaderList	リターン構造、入出庫伝票ヘッダ情報を取得します。
getGoodsMvtItemsList	リターン構造、入出庫伝票明細情報を取得します。
getMaterialRaList	品目コードの範囲情報を取得します。
getMoveTypeRaList	移動タイプの範囲情報を取得します。
getPlantRaList	プラントの範囲情報を取得します。
getPstngDateRaList	登録日付の範囲情報を取得します。
getSpecStockRaList	特殊在庫区分の範囲情報を取得します。
getStgeLocRaList	保管場所の範囲情報を取得します。
getTrEvTypeRaList	トランザクション/イベントタイプの範囲情報を取得します。
getUserNameRaList	ユーザ名の範囲情報を取得します。
getVendorRaList	仕入先の範囲情報を取得します。

サンプルプログラム

```
//■入出庫伝票一覧
var goodsMvtItemsList;
var goodsMvtItemsCount;
var goodsMvtHeaderList;
var goodsMvtHeaderCount;

var vendorRaList;
var vendorRaCount;
var userNameRaList;
var userNameRaCount;
var trEvTypeRaList;
var trEvTypeRaCount;
var stgeLocRaList;
var stgeLocRaCount;
var specStockRaList;
var specStockRaCount;
var pstngDateRaList;
var pstngDateRaCount;
var plantRaList;
var plantRaCount;
```



```

var moveTypeRaList;
var moveTypeRaCount;
var materialRaList;
var materialRaCount;

function init(request) {
    //-----
    //変数宣言
    //-----
    var objCon;
    var objResult;

    //-----
    //インスタンス生成
    //-----
    objCon = new GoodsMvtGetItemsController(); // 説明 1

    //-----
    // テーブル項目セット; // 説明 2
    //-----
    objCon.setTblParam("PstngDateRa", "SIGN", "I", 1); //選択演算子 SIGN (テーブル範囲)
    objCon.setTblParam("PstngDateRa", "OPTION", "BT", 1); //選択演算子 OPTION (範囲テーブル)
    objCon.setTblParam("PstngDateRa", "LOW", "20070120", 1); //伝票の転記日付
    objCon.setTblParam("PstngDateRa", "HIGH", "20070123", 1); //伝票の転記日付

    //-----
    // executeBapi実行; // 説明 3
    //-----
    objResult = objCon.executeBapi();

    //-----
    // エラー判定
    //-----
    if (objResult["ERR_CODE"] != null){
        Debug.browse(objResult);
    }

    //入出庫伝票; // 説明 4
    goodsMvtItemsList = objCon.getGoodsMvtItemsList();
    goodsMvtHeaderList = objCon.getGoodsMvtHeaderList();
    vendorRaList = objCon.getVendorRaList();
    userNameRaList = objCon.getUserNameRaList();
    trEvTypeRaList = objCon.getTrEvTypeRaList();
    stgeLocRaList = objCon.getStgeLocRaList();
    specStockRaList = objCon.getSpecStockRaList();
    pstngDateRaList = objCon.getPstngDateRaList();
    plantRaList = objCon.getPlantRaList();
    moveTypeRaList = objCon.getMoveTypeRaList();
    materialRaList = objCon.getMaterialRaList();

    goodsMvtItemsCount = goodsMvtItemsList.length;
    goodsMvtHeaderCount = goodsMvtHeaderList.length;
    vendorRaCount = vendorRaList.length;
    userNameRaCount = userNameRaList.length;
    trEvTypeRaCount = trEvTypeRaList.length;
    stgeLocRaCount = stgeLocRaList.length;
    specStockRaCount = specStockRaList.length;
    pstngDateRaCount = pstngDateRaList.length;
    plantRaCount = plantRaList.length;
    moveTypeRaCount = moveTypeRaList.length;
    materialRaCount = materialRaList.length;
}

```

- 入出庫伝票一覧照会 API のインスタンス生成(説明1)
GoodsMvtGetItemsController クラスのインスタンスを生成します。
引数は不要です。
インスタンスを生成すると、入出庫伝票一覧照会 BAPI (BAPI_GOODSMVT_GETITEMS) の import 型の table パラメータに値を設定することができます。
- table の各項目に値を設定(説明2)

table パラメータの項目に値を設定する場合は、setTblParam メソッドを使用します。

第1引数のtable名称は固定値になっています。

後述のとおりに名称を設定して下さい。

第2引数の項目名称は SAP の各属性の項目名称を設定してください。

table 名称、項目名称は英字の大文字、小文字は厳密に区別されます。

第3引数に値、第4引数にレコード番号を設定します。

SAP structure	セットする固有名称
BATCH_RA (ロットの範囲)	BatchRa
GOODSMVT_HEADER (リターン構造、入出庫伝票ヘッダ)	GoodsMvtHeader
GOODSMVT_ITEMS (リターン構造、入出庫伝票明細)	GoodsMvtItems
MATERIAL_RA (品目コードの範囲)	MaterialRa
MOVE_TYPE_RA (移動タイプの範囲)	MoveTypeRa
PLANT_RA (プラントの範囲)	PlantRa
PSTNG_DATE_RA(登録日付の範囲)	PstngDateRa
SPEC_STOCK_RA (特殊在庫区分の範囲)	SpecStockRa
STGE_LOC_RA (保管場所の範囲)	StgeLocRa
TR_EV_TYPE_RA (トランザクション/ イベントタイプの範囲)	TrEvTypeRa
USERNAME_RA (ユーザ名の範囲)	UserNameRa
VENDOR_RA (仕入先の範囲)	VendorRa

- 入出庫伝票一覧照会(説明3)
GoodsMvtGetItemsController クラスの executeBapi メソッドを実行すると、各項目にセットされた値で SAPより入出庫伝票一覧データとして照会されます。
入出庫伝票一覧照会用 BAPI (BAPI_GOODSMVT_GETITEMS) の実行後に RETURN パラメータの内容も取得します。
- サンプルプログラムの中で記載されたデータ取得メソッドの詳細説明については、以下を参照してください。(説明4)
 - Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn を取得
GoodsMvtGetItemsController クラスの getBapiReturnErrMessage メソッドを使用すると、取得したBapiReturn の Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn オブジェクトのメッセージ情報を取得することが可能です。
全て正常の場合は null を取得します。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - Type 情報が正常(デフォルト値: "S" / "")の BapiReturn を取得
GoodsMvtGetItemsController クラスの getBapiReturnMessage メソッドを使用すると、取得したBapiReturn の Type 情報が正常(デフォルト値: "S" / "")と判断された Type 情報のメッセージ情報を取得することが可能です。
メッセージが存在しない場合は null を取得します。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - ロットの範囲情報を取得
GoodsMvtGetItemsController クラスのgetBatchRaList メソッドを使用すると、設定したロットの範囲情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - リターン構造、入出庫伝票ヘッダ情報を取得
GoodsMvtGetItemsController クラスの getGoodsMvtHeaderList メソッドを使用すると、取得したリターン構造、入出庫伝票ヘッダ情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - リターン構造、入出庫伝票明細情報を取得
GoodsMvtGetItemsController クラスの getGoodsMvtItemsList メソッドを使用すると、取得したリターン構造、入出庫伝票明細情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 品目コードの範囲情報を取得
GoodsMvtGetItemsController クラスの getMaterialRaList メソッドを使用すると、設定した品目コードの範囲情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 移動タイプの範囲情報を取得
GoodsMvtGetItemsController クラスの getMoveTypeRaList メソッドを使用すると、設定した移動タイプの範囲情報を取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

- プラントの範囲情報を取得
GoodsMvtGetItemsController クラスの getPlantRaList メソッドを使用すると、設定したプラントの範囲情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 登録日付の範囲情報を取得
GoodsMvtGetItemsController クラスの getPstngDateRaList メソッドを使用すると、設定した登録日付の範囲情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 特殊在庫区分の範囲情報を取得
GoodsMvtGetItemsController クラスの getSpecStockRaList メソッドを使用すると、設定した特殊在庫区分の範囲情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 保管場所の範囲情報を取得
GoodsMvtGetItemsController クラスの getStgeLocRaList メソッドを使用すると、設定した保管場所の範囲情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- トランザクション/イベントタイプの範囲情報を取得
GoodsMvtGetItemsController クラスの getTrEvTypeRaList メソッドを使用すると、設定したトランザクション/イベントタイプの範囲情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- ユーザ名の範囲情報を取得
GoodsMvtGetItemsController クラスの getUsernameRaList メソッドを使用すると、設定したユーザ名の範囲情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 仕入先の範囲情報を取得
GoodsMvtGetItemsController クラスの getVendorRaList メソッドを使用すると、設定した仕入先の範囲情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。



コラム

サンプルプログラム実行結果では、SAP テスト環境のデータが取得されています。SAP 標準では該当データが存在しません。

プラントの品目在庫 確認 API (MaterialAvailabilityController)

プラントの品目在庫を確認するためには、MaterialAvailabilityController クラスの持つ各メソッドを利用する必要があります。引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
MaterialAvailabilityController	コンストラクタ (引数無し)
MaterialAvailabilityController	コンストラクタ (第 1 引数あり)。 ConnectionManager を使う場合に使用する。第 1 引数で「ConnectionManager」を設定する。
MaterialAvailabilityController	コンストラクタ (第 1 引数あり)。 SAPClient の接続情報を使う場合に使用する。第 1 引数で「SAPClient」(オブジェクト)を設定する。
setAvQtyPlt	import 項目: プラントレベル利用可能数量を設定します。
setBatch	import 項目: ロットを設定します。
setCheckRule	import 項目: 確認規則を設定します。
setCustomer	import 項目: 得意先コードを設定します。
setDecForRounding	import 項目: 小数点位置 (その位置まで丸める)を設定します。
setDecForRoundingX	import 項目: 関連ユーザデータ項目の更新情報を設定します。

コンストラクタ / メソッド名	説明
setDialogflag	import 項目：区分 (X = 利用不可 / N = 確認なし)を設定します。
setDocNumber	import 項目：伝票番号を設定します。
setEndleadtime	import 項目：補充リードタイムの終了日付を設定します。
setItmNumber	import 項目：明細番号を設定します。
setMaterial	import 項目：品目コードを設定します。
setPlant	import 項目：プラントを設定します。
setStgeLoc	import 項目：保管場所を設定します。
setStockInd	import 項目：特殊在庫区分を設定します。
setUnit	import 項目：表示数量単位を設定します。
setWbsElem	import 項目：WBS 要素を設定します。
setReadAtpLock	import 項目：管理区分:利用可能在庫確認を設定します。
setReadAtpLockX	import 項目：関連ユーザデータ項目の更新情報を設定します。
setStrParam	構造型の各項目に値を設定する。 第一引数に structure 名、第二引数に項目名、第三引数に値を設定する。
setTblParam	テーブル型の各項目に値を設定する。 第1引数にテーブル名、第2引数に項目名第3引数に値、第4引数にレコード番号を設定する。
executeBapi	BapiController により品目在庫を検索します。
getBapiReturnErrorMessage	BAPI実行結果である BapiReturn の Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn オブジェクトのメッセージ情報を全て返します。 全て正常の場合は null が返却されます。
getBapiReturnMessage	BAPI実行結果である BapiReturn の Type 情報が正常(デフォルト値: "S" / "")と判断された Type 情報のメッセージ情報を全て返します。 メッセージが存在しない場合は null が返却されます。
getBatch	import 項目：ロットを取得します。
getCheckRule	import 項目：確認規則を取得します。
getCustomer	import 項目：得意先コードを取得します。
getDecForRounding	import 項目：小数点位置 (その位置まで丸める)を取得します。
getDecForRoundingX	import 項目：関連ユーザデータ項目の更新情報を取得します。
getDialogflag	export 項目：区分 (X = 利用不可 / N = 確認なし)を取得します。
getDocNumber	import 項目：伝票番号を取得します。
getEndleadtime	export 項目：補充リードタイムの終了日付を取得します。
getItmNumber	import 項目：明細番号を取得します。
getMaterial	import 項目：品目コードを取得します。
getAvQtyPlt	export 項目：プラントレベル利用可能数量を取得します。
getPlant	import 項目：プラントを取得します。
getStgeLoc	import 項目：保管場所を取得します。
getUnit	import 項目：表示数量単位を取得します。
getWbsElem	import 項目：WBS 要素を取得します。
getReadAtpLock	import 項目：管理区分:利用可能在庫確認を取得します。
getReadAtpLockX	import 項目:関連ユーザデータ項目の更新情報を取得します。

コンストラクタ / メソッド名	説明
getMaterialEvg	import 項目 : 品目コード(長)情報を取得します。
getWmdvexList	出力テーブル (日付および ATP 数量)情報を取得します。
getWmdvsxList	入力テーブル (日付/数量)情報を取得します。

サンプルプログラム

```

var wmdvexList;
var wmdvexCount;
var wmdvsxList;
var wmdvsxCount;
var endleadtme;
var dialogflag;
var avqtypIt;

function init(request) {

    //-----
    //変数宣言
    //-----
    var objCon;
    var objResult;

    //-----
    //インスタンス生成 // 説明 1
    //-----
    objCon = new MaterialAvailabilityController();

    //-----
    // 単一項目セット // 説明 2
    //-----
    objCon.setUnit("ST"); //表示数量単位
    objCon.setPlant("1110"); //プラント
    objCon.setMaterial("00000000000000011"); //品目コード

    //-----
    // executeBapi実行 // 説明 5
    //-----
    objResult = objCon.executeBapi();

    //-----
    // エラー判定
    //-----
    if ( objResult["ERR_CODE"] != null ) {
        Debug.browse(objResult);
    }
    //セッションから在庫情報を取得 // 説明 6
    wmdvexList = objCon.getWmdvexList();
    wmdvexCount = wmdvexList.length;

    wmdvsxList = objCon.getWmdvsxList();
    wmdvsxCount = wmdvsxList.length;

    endleadtme = objCon.getEndleadtme()
    dialogflag = objCon.getDialogflag();
    avqtypIt = objCon.getAvQtyPlt();
}

```

- プラントの品目在庫確認 API のインスタンス生成(説明1)
MaterialAvailabilityController クラスのインスタンスを生成します。
引数は不要です。
インスタンスを生成すると、プラントの品目在庫確認 BAPI (BAPI_MATERIAL_AVAILABILITY) の import 型の import、table パラメータに値を設定することができます。
- import の各項目に値を設定(説明2)

import パラメータの項目に値を設定する場合は、set メソッドを使用します。

メソッドについては各 API のAPI リストを参照ください。

- structure の各項目に値を設定(説明3)

structure パラメータの項目に値を設定する場合は、setStrParam メソッドを使用します。

第1引数のstructure 名称は固定値になっています。

後述のとおりに名称を設定して下さい。

第2引数の項目名称はSAP の各属性の項目名称を設定してください。

structure 名称、項目名称は英字の大文字、小文字は厳密に区別されます。

第3引数に値を設定します。

- table の各項目に値を設定(説明4)

table パラメータの項目に値を設定する場合は、setTblParam メソッドを使用します。第一引数のtable名称は固定値になっています。

後述のとおりに名称を設定して下さい。

第2引数の項目名称は SAP の各属性の項目名称を設定してください。

table 名称、項目名称は英字の大文字、小文字は厳密に区別されます。

第3引数に値、第4引数にレコード番号を設定します。

SAP structure	セットする固有名称
WMDVEX (出カテール(日付および ATP 数量))	Wmdvex
WMDVSX (入カテール (日付/数量))	Wmdvsx

- プラントの品目在庫確認(説明5)

MaterialAvailabilityController クラスの executeBapi メソッドを実行すると、各項目にセットされた値で SAPよりプラントの品目在庫確認データとして照会されます。

プラントの品目在庫確認用 BAPI (BAPI_MATERIAL_AVAILABILITY) の実行後に RETURN パラメータの内容も取得します。

- サンプルプログラムの中で記載されたデータ取得メソッドの詳細説明については、以下を参照してください。(説明6)

- Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn を取得
MaterialAvailabilityController クラスの getBapiReturnErrMessage メソッドを使用すると、取得したBapiReturn の Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn オブジェクトのメッセージ情報を取得することが可能です。
全て正常の場合は null を取得します。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- Type 情報が正常(デフォルト値: "S" / "")の BapiReturn を取得
MaterialAvailabilityController クラスの getBapiReturnMessage メソッドを使用すると、取得したBapiReturn の Type 情報が正常(デフォルト値: "S" / "")と判断された Type 情報のメッセージ情報を取得することが可能です。
メッセージが存在しない場合は null を取得します。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- プラントレベル利用可能数量情報を取得
MaterialAvailabilityController クラスのgetAvQtyPlt メソッドを使用すると、取得したプラントレベル利用可能数量情報を取得することが可能です。
- ロット情報を取得
MaterialAvailabilityController クラスのgetBatch メソッドを使用すると、設定したロット情報を取得することが可能です。
- 確認規則情報を取得
MaterialAvailabilityController クラスの getCheckRule メソッドを使用すると、設定した確認規則情報を取得することが可能です。
- 得意先コード情報を取得
MaterialAvailabilityController クラスの getCustomer メソッドを使用すると、設定した得意先コード情報を取得することが可能です。
- 小数点位置 (その位置まで丸める)情報を取得
MaterialAvailabilityControllerクラスのgetDecForRoundingメソッドを使用すると、設定した小数点位置(その位置まで丸める)情報を取得することが可能です。
- 関連ユーザデータ項目の更新情報情報を取得
MaterialAvailabilityController クラスの getDecForRoundingX メソッドを使用すると、設定した関連ユーザデータ項目の更新情報情報を取得することが可能です。
- 区分 (X = 利用不可 /N = 確認なし)情報を取得
MaterialAvailabilityController クラスの getDialogflag メソッドを使用すると、取得した区分 (X = 利用不可 /N = 確認なし)情報を取得することが可能です。
- 伝票番号情報を取得
MaterialAvailabilityController クラスの getDocNumber メソッドを使用すると、設定した伝票番号情報を取得することが可能です。

- 補充リードタイムの終了日付情報を取得
MaterialAvailabilityController クラスの getEndleadtime メソッドを使用すると、取得した補充リードタイムの終了日付情報を取得することが可能です。
- 明細番号情報を取得
MaterialAvailabilityController クラスの getitmNumber メソッドを使用すると、設定した明細番号情報を取得することが可能です。
- 品目コード情報を取得
MaterialAvailabilityController クラスの getMaterial メソッドを使用すると、設定した品目コード情報を取得することが可能です。
- プラント情報を取得
MaterialAvailabilityController クラスの getPlant メソッドを使用すると、設定したプラント情報を取得することが可能です。
- 保管場所情報を取得
MaterialAvailabilityController クラスの getStgeLoc メソッドを使用すると、設定した保管場所情報を取得することが可能です。
- 表示数量単位情報を取得
MaterialAvailabilityController クラスの getUnit メソッドを使用すると、設定した表示数量単位情報を取得することが可能です。
- WBS 要素情報を取得
MaterialAvailabilityController クラスの getWbsElem メソッドを使用すると、設定した WBS 要素情報を取得することが可能です。
- 管理区分:利用可能在庫確認情報を取得
MaterialAvailabilityController クラスの getReadAtpLock メソッドを使用すると、設定した管理区分:利用可能在庫確認情報を取得することが可能です。
- 関連ユーザデータ項目の更新情報を取得
MaterialAvailabilityController クラスの getReadAtpLockX メソッドを使用すると、設定した関連ユーザデータ項目の更新情報を取得することが可能です。
- 品目コード（長）情報を取得
MaterialAvailabilityController クラスの getMaterialEvg メソッドを使用すると、設定した品目コード（長）情報を取得することが可能です。
- 出力テーブル（日付および ATP 数量）情報を取得
MaterialAvailabilityController クラスの getWmdvexList メソッドを使用すると、取得した出力テーブル（日付および ATP 数量）情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 入力テーブル（日付/数量）情報を取得
MaterialAvailabilityController クラスの getWmdvsxList メソッドを使用すると、設定した入力テーブル（日付/数量）情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。



コラム

サンプルプログラム実行結果では、SAP テスト環境のデータが取得されています。SAP 標準では該当データが存在しません。

品目マスタ 照会 API (MaterialGetAllController)

品目マスタを照会するためには、MaterialGetAllController クラスの持つ各メソッドを利用する必要があります。
引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
MaterialGetAllController	コンストラクタ（引数無し）
MaterialGetAllController	コンストラクタ（第1引数あり）。 ConnectionManager を使う場合に使用する。 第1引数で「ConnectionManager」を設定する。
MaterialGetAllController	コンストラクタ（第1引数あり）。 SAPClient の接続情報を使う場合に使用する。 第1引数で「SAPClient」（オブジェクト）を設定する。
setCompanycode	import 項目：会社コードを設定します。

コンストラクタ / メソッド名	説明
setDistributionchannel	import 項目：流通チャンネルを設定します。
setLifovaluationlevel	import 項目：LIFO 評価レベルを設定します。
setMaterial	import 項目：品目コードを設定します。
setPlant	import 項目：プラントを設定します。
setSalesorganisation	import 項目：販売組織を設定します。
setStorageelocation	import 項目：保管場所を設定します。
setStorageetype	import 項目：保管域タイプを設定します。
setValuationarea	import 項目：評価レベルを設定します。
setValuationtype	import 項目：評価タイプを設定します。
setWarehousenumber	import 項目：倉庫番号/複合倉庫を設定します。
setStrParam	構造型の各項目に値を設定する。 第1引数に structure 名、第2引数に項目名、第3引数に値を設定する。
setTblParam	テーブル型の各項目に値を設定する。 第1引数にテーブル名、第2引数に項目名第3引数に値、第4引数にレコード番号を設定する。
executeBapi	BapiController により品目コードを検索します。
getBapiReturnErrMsg	BAPI実行結果である BapiReturn の Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn オブジェクトのメッセージ情報を全て返します。 全て正常の場合は null が返却されます。
getBapiReturnMessage	BAPI実行結果である BapiReturn の Type 情報が正常(デフォルト値: "S" / "")と判断された Type 情報のメッセージ情報を全て返します。 メッセージが存在しない場合は null が返却されます。
getCompanycode	import 項目：会社コードを取得します。
getDistributionchannel	import 項目：流通チャンネルを取得します。
getLifovaluationlevel	import 項目：LIFO 評価レベルを取得します。
getMaterial	import 項目：品目コードを取得します。
getPlant	import 項目：プラントを取得します。
getSalesorganisation	import 項目：販売組織を取得します。
getStorageelocation	import 項目：保管場所を取得します。
getStorageetype	import 項目：保管域タイプを取得します。
getValuationarea	import 項目：評価レベルを取得します。
getValuationtype	import 項目：評価タイプを取得します。
getWarehousenumber	import 項目：倉庫番号/複合倉庫を取得します。
getMaterialEvg	import 項目：品目コード(長)情報を取得します。
getClientData	品目データ (クライアントレベル)情報を取得します。
getSalesData	販売データ情報を取得します。
getStorageLocationData	品目データ (保管場所レベル)情報を取得します。
getExtensionOutList	BAPI パラメータ EXTENSIONIN/EXTENSIONOUT の参照構造情報を取得します。 取得されるデータについては、カラム順の連続データとなる。
getForecastParameters	予測パラメータ情報を取得します。

コンストラクタ / メソッド名	説明
getInternationArticleNumbersList	EAN 情報を取得します。
getLifoValuationData	LIFO 関連品目データ情報を取得します。
getMaterialDescriptionList	品目テキスト情報を取得します。
getMaterialTextList	テキスト (長)情報を取得します。
getPlanningData	品目マスタ/製品グループの変更伝票構造情報を取得します。
getPlantData	品目データ (プラントレベル)情報を取得します。
getProductionResourceToolData	品目マスタの生産資源/治工具 (PRT) 項目情報を取得します。
getSalesData	販売データ情報を取得します。
getStorageLocationData	品目データ (保管場所レベル)情報を取得します。
getStorageTypeData	保管域タイプデータ情報を取得します。
getTaxClassificationsList	制御データ情報を取得します。
getUnitSofmeasureList	数量単位情報を取得します。
getValuationData	評価データ情報を取得します。
getWarehouseNumberData	倉庫番号データ情報を取得します。

サンプルプログラム

```

var clientData;
var plantData;
var forecastParameters;
var planningData;
var storageLocationData;
var valuationData;
var warehouseNumberData;
var salesData;
var storageTypeData;
var productionResourceToolData;
var lifoValuationData;

var materialDescriptionList;
var materialDescriptionCnt;
var unitSofmeasureList;
var unitSofmeasureCnt;
var internationArticleNumbersList;
var internationArticleNumbersCnt;
var materialTextList;
var materialTextCnt;
var taxClassificationsList;
var taxClassificationsCnt;
var extensionOutList;
var extensionOutCnt;

function init(request) {

    //-----
    //変数宣言
    //-----
    var objCon;
    var objResult;

    //-----
    //インスタンス生成
    //-----
    objCon = new MaterialGetAllController(); //説明 1

    //-----
    //単一項目セット //説明 2
    //-----
    objCon.setMaterial("000000000000000724"); //品目コード
    objCon.setPlant("1110"); //プラント

```



```

objCon.setStorageLocation("1111"); //保管場所
objCon.setSalesorganisation("1100"); //販売組織
objCon.setDistributionchannel("1A"); //流通チャンネル

//-----
// executeBapi実行 //説明 5
//-----
objResult = objCon.executeBapi();

//-----
// エラー判定
//-----
if ( objResult["ERR_CODE"] != null ) {
    Debug.browse(objResult);
}

//セッションから品目マスタ照会情報を取得 //説明 6
clientData = objCon.getClientData()
plantData = objCon.getPlantData()
forecastParameters = objCon.getForecastParameters()
planningData = objCon.getPlanningData()
storageLocationData = objCon.getStorageLocationData()
valuationData = objCon.getValuationData()
warehouseNumberData = objCon.getWarehouseNumberData()
salesData = objCon.getSalesData()
storageTypeData = objCon.getStorageTypeData()
productionResourceToolData = objCon.getProductionResourceToolData()
lifoValuationData = objCon.getLifoValuationData()

//品目テキストの内容と件数を取得
materialDescriptionList = objCon.getMaterialDescriptionList()
materialDescriptionCnt = materialDescriptionList.length;

//数量単位の内容と件数を取得
unitSofmeasureList = objCon.getUnitSofmeasureList()
unitSofmeasureCnt = unitSofmeasureList.length;

//EANの内容と件数を取得
internationArticleNumbersList = objCon.getInternationArticleNumbersList()
internationArticleNumbersCnt = internationArticleNumbersList.length;

//テキスト(長)の内容と件数を取得
materialTextList = objCon.getMaterialTextList()
materialTextCnt = materialTextList.length;

//制御データの内容と件数を取得
taxClassificationsList = objCon.getTaxClassificationsList()
taxClassificationsCnt = taxClassificationsList.length;

//BAPI パラメータ EXTENSIONIN/EXTENSIONOUT の参照構造の内容と件数を取得
extensionOutList = objCon.getExtensionOutList()
extensionOutCnt = extensionOutList.length;
}

```

- 品目マスタ照会 API のインスタンス生成(説明1)
MaterialGetAllController クラスのインスタンスを生成します。
引数は不要です。
インスタンスを生成すると、品目マスタ照会 BAPI (BAPI_MATERIAL_GETALL) の import 型の import、table パラメータに値を設定することができます。
- import の各項目に値を設定(説明2)
import パラメータの項目に値を設定する場合は、set メソッドを使用します。
メソッドについては各 API の API リストを参照ください。
- structure の各項目に値を設定(説明3)
structure パラメータの項目に値を設定する場合は、setStrParam メソッドを使用します。
第1引数のstructure 名称は固定値になっています。
後述のとおり名称を設定して下さい。

第2引数の項目名称はSAPの各属性の項目名称を設定してください。

structure 名称、項目名称は英字の大文字、小文字は厳密に区別されます。

第3引数に値を設定します。

SAP structure	セットする固有名称
MATERIAL_EVG(品目コード(長))	MaterialEvg

..warning::

サンプルプログラムにはstructureのセットは行なっていません。

- table の各項目に値を設定(説明4)

table パラメータの項目に値を設定する場合は、setTblParam メソッドを使用します。

第1引数のtable名称は固定値になっています。

後述のとおり名称を設定して下さい。

第2引数の項目名称はSAPの各属性の項目名称を設定してください。

table 名称、項目名称は英字の大文字、小文字は厳密に区別されます。

第3引数に値、第四引数にレコード番号を設定します。

SAP table	セットする固有名称
EXTENSIONOUT (BAPIパラメータ EXTENSIONIN/EXTENSIONOUTの参照構造)	ExtensionOut
INTERNATIONARTICLENUMBERS(EAN)	InternationArticleNumbers
MATERIALDESCRIPTION(品目テキスト)	MaterialDescription
MATERIALTEXT(テキスト(長))	MaterialText
TAXCLASSIFICATIONS(制御データ)	TaxClassifications
UNITSOFMEASURE(数量単位)	UnitSofmeasure

..warning::

サンプルプログラムにはtableのセットは行なっていません。

- 品目マスタ照会(説明5)

MaterialGetAllController クラスの executeBapi メソッドを実行すると、各項目にセットされた値でSAPより品目マスタデータとして照会されます。

品目マスタ照会用 BAPI (BAPI_MATERIAL_GETALL) の実行後に RETURN パラメータの内容も取得します。

- サンプルプログラムの中で記載されたデータ取得メソッドの詳細説明については、以下を参照してください。(説明6)

- Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn を取得

MaterialGetAllControllerクラスのgetBapiReturnErrMsgメソッドを使用すると、取得したBapiReturnのType情報が正常(デフォルト値:"S"/"")以外のBapiReturnオブジェクトのメッセージ情報を取得することが可能です。

全て正常の場合はnullを取得します。

項目の値を取得するには、KEYにSAPの各属性の項目名称を設定してください。

- Type 情報が正常(デフォルト値: "S" / "")の BapiReturn を取得

MaterialGetAllController クラスの getBapiReturnMessage メソッドを使用すると、取得した BapiReturnのType情報が正常(デフォルト値: "S" / "")と判断された Type情報のメッセージ情報を取得することが可能です。

メッセージが存在しない場合は null を取得します。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

- 会社コード情報を取得

MaterialGetAllController クラスの getCompanycode メソッドを使用すると、設定した会社コード情報を取得することが可能です。

- 流通チャンネル情報を取得

MaterialGetAllController クラスの getDistributionchannel メソッドを使用すると、設定した流通チャンネル情報を取得することが可能です。

- LIFO 評価レベル情報を取得

MaterialGetAllController クラスの getLifovaluationlevel メソッドを使用すると、設定した LIFO 評価レベル情報を取得することが可能です。

- 品目コード情報を取得

MaterialGetAllController クラスの getMaterial メソッドを使用すると、設定した品目コード情報を取得することが可能です。

- プラント情報を取得

MaterialGetAllController クラスの getPlant メソッドを使用すると、設定したプラント情報を取得することが可能です。

- 販売組織情報を取得

- MaterialGetAllController クラスのgetSalesorganisation メソッドを使用すると、設定した販売組織情報を取得することが可能です。
- 保管場所情報を取得
MaterialGetAllController クラスの getStorageelocation メソッドを使用すると、設定した保管場所情報を取得することが可能です。
 - 保管域タイプ情報を取得
MaterialGetAllController クラスの getStoragetype メソッドを使用すると、設定した保管域タイプ情報を取得することが可能です。
 - 評価レベル情報を取得
MaterialGetAllController クラスの getValuationarea メソッドを使用すると、設定した評価レベル情報を取得することが可能です。
 - 評価タイプ情報を取得
MaterialGetAllController クラスの getValuationtype メソッドを使用すると、設定した評価タイプ情報を取得することが可能です。
 - 倉庫番号/複合倉庫情報を取得
MaterialGetAllController クラスの getWarehousenumber メソッドを使用すると、設定した倉庫番号/複合倉庫情報を取得することが可能です。
 - 品目コード (長)情報を取得
MaterialGetAllController クラスの getMaterialEvg メソッドを使用すると、設定した品目コード (長)情報を取得することが可能です。
 - 品目データ (クライアントレベル)情報を取得
MaterialGetAllController クラスの getClientData メソッドを使用すると、取得した品目データ (クライアントレベル)情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - BAPI パラメータ EXTENSIONIN/EXTENSIONOUT の参照構造情報を取得
MaterialGetAllController クラスの getExtensionOutList メソッドを使用すると、取得した BAPI パラメータ EXTENSIONIN/EXTENSIONOUT の参照構造情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 予測パラメータ情報を取得
MaterialGetAllController クラスの getForecastParameters メソッドを使用すると、取得した予測パラメータ情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - EAN 情報を取得
MaterialGetAllController クラスの getInternationArticleNumbersList メソッドを使用すると、取得した EAN 情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - LIFO 関連品目データ情報を取得
MaterialGetAllController クラスの getLifoValuationData メソッドを使用すると、取得した LIFO 関連品目データ情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 品目テキスト情報を取得
MaterialGetAllController クラスの getMaterialDescriptionList メソッドを使用すると、取得した品目テキスト情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - テキスト (長)情報を取得
MaterialGetAllController クラスの getMaterialTextList メソッドを使用すると、取得したテキスト (長)情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 品目マスタ/製品グループの変更伝票構造情報を取得
MaterialGetAllController クラスの getPlanningData メソッドを使用すると、取得した品目マスタ/製品グループの変更伝票構造情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 品目データ (プラントレベル)情報を取得
MaterialGetAllController クラスの getPlantData メソッドを使用すると、取得した品目データ (プラントレベル)情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 品目マスタの生産資源/治工具 (PRT) 項目情報を取得
MaterialGetAllController クラスの getProductionResourceToolData メソッドを使用すると、取得した品目マスタの生産資源/治工具 (PRT) 項目情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

- 販売データ情報を取得

MaterialGetAllController クラスの `getSalesData` メソッドを使用すると、取得した販売データ情報を取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 品目データ (保管場所レベル)情報を取得

MaterialGetAllController クラスの `getStorageLocationData` メソッドを使用すると、取得した品目データ(保管場所レベル)情報を取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 保管域タイプデータ情報を取得

MaterialGetAllController クラスの `getStorageTypeData` メソッドを使用すると、取得した保管域タイプデータ情報を取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 制御データ情報を取得

MaterialGetAllController クラスの `getTaxClassificationsList` メソッドを使用すると、取得した制御データ情報を取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 数量単位情報を取得

MaterialGetAllController クラスの `getUnitSofmeasureList` メソッドを使用すると、取得した数量単位情報を取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 評価データ情報を取得

MaterialGetAllController クラスの `getValuationData` メソッドを使用すると、取得した評価データ情報を取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 倉庫番号データ情報を取得

MaterialGetAllController クラスの `getWarehouseNumberData` メソッドを使用すると、取得した倉庫番号データ情報を取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。



コラム

サンプルプログラム実行結果では、SAP テスト環境のデータが取得されています。
SAP 標準では該当データが存在しません。

品目コード 検索 API (MaterialGetListController)

品目コードを検索するためには、MaterialGetListController クラスの持つ各メソッドを利用する必要があります。
引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
MaterialGetListController	コンストラクタ (引数無し)
setMaxRows	選択予定品目の最大数に値を設定する。
setTblParam	テーブル型の各項目に値を設定する。 第1引数にテーブル名、第2引数に項目名、第3引数に値、第4引数にレコード番号を設定する。
execute	setMaxRows、setTblParamのメソッドで設定した値で、品目コードを検索する。
getMatnrlistlist	検索結果の品目コード、テキスト一覧を取得する。

サンプルプログラム

```
// ■品目コード一覧
var materialList;
var materialCount;

function init(request) {
  var objCon;
  var objResult;

  // インスタンス生成
  objCon = new MaterialGetListController(); // 説明 1

  // 選択予定品目の最大数に値を設定
  objCon.setMaxRows("200"); // 説明 2

  // 品目コードの選択オプションの各項目に値を設定
  // 1 明細
  objCon.setTblParam("MatnrSelection", "SIGN", "I", 1);
  objCon.setTblParam("MatnrSelection", "OPTION", "BT", 1);
  objCon.setTblParam("MatnrSelection", "MATNR_LOW", "000000000000000000", 1);
  objCon.setTblParam("MatnrSelection", "MATNR_HIGH", "999999999999999999", 1);

  // 品目コードの検索
  objResult = objCon.executeBapi(); // 説明 4

  // 要素[ERR_CODE]に何らかの値がセットされている場合
  if (objResult["ERR_CODE"] != null) {
    // エラーメッセージ出力
    Debug.browse(objResult);
  }

  // 伝票明細
  materialList = objCon.getMatnrlistlist(); // 説明 5
  materialCount = materialList.length;
}
}
```

- 品目コード検索 API のインスタンス生成(説明1)
MaterialGetListController クラスのインスタンスを生成します。
引数は不要です。
インスタンスを生成すると、
品目コード検索BAPI (BAPI_MATERIAL_GETLIST) のimport型のimport、table パラメータに値を設定することができます。
- 選択予定品目の最大数に値を設定(説明2) | 選択予定品目の最大数の項目に値を設定する場合は、setMaxRows メソッドを使用します。
- table の各項目に値を設定(説明3)
table パラメータの項目に値を設定する場合は、setTblParam メソッドを使用します。
第一引数のtable名称は固定値になっています。
後述のとおり名称を設定して下さい。
第二引数の項目名称は SAP の各属性の項目名称を設定して下さい。
table 名称、項目名称は英字の大文字、小文字は厳密に区別されます。
第三引数に値、第四引数にレコード番号を設定します。

SAP table	セットする固有名称
MATNRSELECTION (品目コードの選択オプション)	MatnrSelection
MATERIALSHORTDESCSEL (品目テキストの選択オプション)	MaterialShortDescsel
MANUFACTURERPARTNUMB (製造業者と製造者製品コード)	ManufacturerPartNumb
PLANTSELECTION (プラントの選択オプション)	PlantSelection
STORAGELOCATIONSELECT (保管場所の選択オプション)	StorageLocationSelect
SALESORGANISATIONSELECTION (販売組織の選択オプション)	SalesOrganisationSelection
DISTRIBUTIONCHANNELSELECTION (流通チャネルの選択オプション)	DistributionChannelSelection

- 品目コードの検索(説明4)

MaterialGetListControllerクラスのexecuteメソッドを実行すると、各項目にセットされた値を条件にSAPの品目コードを検索します。

- 品目コード、テキスト一覧を取得(説明5)
MaterialGetListController クラスの getMatnrlistlist メソッドを使用すると、検索された品目コード、テキストを取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

品目マスタ 登録／変更／削除 API (MaterialSaveReplicaController)

品目マスタを更新するためには、MaterialSaveReplicaController クラスの持つ各メソッドを利用する必要があります。

引数、戻り値についての詳細な情報は、API リストを参照してください。

また、MaterialSaveReplicaController クラスで使用する品目マスタ登録 BAPI (BAPI_MATERIAL_SAVEREPLICA) は SAP BAPI プログラム内部で commit 処理が行われます。

ConnectionManager によるトランザクション制御が行えませんので、ご注意ください。

各メソッドの説明

コンストラクタ / メソッド名	説明
MaterialSaveReplicaController	コンストラクタ (引数無し)
MaterialSaveReplicaController	コンストラクタ (第 1 引数あり)。 ConnectionManager を使う場合に使用する。 第 1 引数で「ConnectionManager」を設定する。
MaterialSaveReplicaController	コンストラクタ (第 1 引数あり)。 SAPClient の接続情報を使う場合に使用する。 第 1 引数で「SAPClient」 (オブジェクト) を設定する。
setFlagCadCall	import 項目 : CAD システムからコールを設定します。
setInpfldcheck	import 項目 : 項目無効時の応答を設定します。
setNoappllog	import 項目 : アプリケーションログ書込なしを設定します。
setNochangedoc	import 項目 : 変更伝票書込なしを設定します。
setTestrun	import 項目 : 書込 BAPI のシミュレーションセッションへの切替を設定します。
setFlagOnline	import 項目 : No ALE Field Selection を設定します。
setNoRollbackWork	import 項目 : Override Rollback Work を設定します。
setTblParam	テーブル型の各項目に値を設定する。 第1引数にテーブル名、第2引数に項目名第3引数に値、第4引数にレコード番号を設定する。
executeBapi	BapiController により品目マスタを更新します。
getBapiReturnErrMessage	BAPI実行結果である BapiReturn の Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn オブジェクトのメッセージ情報を全て返します。 全て正常の場合は null が返却されます。
getBapiReturnMessage	BAPI実行結果である BapiReturn の Type 情報が正常(デフォルト値: "S" / "")と判断された Type 情報のメッセージ情報を全て返します。 メッセージが存在しない場合は null が返却されます。
getFlagCadCall	import 項目 : CAD システムからコールを取得します。
getInpfldcheck	import 項目 : 項目無効時の応答を取得します。
getNoappllog	import 項目 : アプリケーションログ書込なしを取得します。
getNochangedoc	import 項目 : 変更伝票書込なしを取得します。
getTestrun	import 項目 : 書込 BAPI のシミュレーションセッションへの切替を取得します。

コンストラクタ / メソッド名	説明
getFlagOnline	import 項目 : No ALE Field Selection を取得します。
getNoRollbackWork	import 項目 : Override Rollback Work を取得します。
getClientDataList	import 項目 : 設定した品目データ (クライアントレベル)情報を取得します。
getExtensionInList	BAPI パラメータ EXTENSIONIN/EXTENSIONOUT の参照構造情報を取得します。 取得されるデータについては、カラム順の連続データとなる。
getExtensionInXList	import 項目 : 設定した拡張イン/拡張アウトのチェックボックス構造情報を取得します。
getForecastParametersList	import 項目 : 設定した需要予測パラメータ情報を取得します。
getForecastValuesList	import 項目 : 設定した需要予測値のデータ転送情報を取得します。
getHeadDataList	import 項目 : 設定した管理情報設定ヘッダセグメント情報を取得します。
getInternationalArtnosList	import 項目 : 設定した国際商品コード (EAN)情報を取得します。
getMaterialDescriptionList	import 項目 : 設定した品目テキスト情報を取得します。
getMaterialLongTextList	import 項目 : 設定したテキスト (長)情報を取得します。
getPlanningDataList	import 項目 : 設定した品目マスタ/製品グループの変更伝票構造情報を取得します。
getPlantDataList	import 項目 : 設定した品目データ (プラントレベル)情報を取得します。
getPrtDataList	import 項目 : 設定した品目マスタの生産資源/治工具 (PRT) 項目情報を取得します。
getReturnMessagesList	import 項目 : 設定したリターンパラメータ BAPIRET2 の代替構造情報を取得します。
getSalesDataList	import 項目 : 設定した販売データ情報を取得します。
getStorageLocationDataList	import 項目 : 設定した品目データ (保管場所レベル)情報を取得します。
getStorageTypeDataList	import 項目 : 設定した保管域タイプデータ情報を取得します。
getTaxClassificationsList	import 項目 : 設定した制御データ情報を取得します。
getTotalConsumptionList	import 項目 : 設定した品目総消費量のデータ転送情報を取得します。
getUnitSofmeasureList	import 項目 : 設定した数量単位情報を取得します。
getUnplndConsumptionList	import 項目 : 設定した計画外消費のデータ転送情報を取得します。
getValuationDataList	import 項目 : 設定した評価データ情報を取得します。
getWarehouseNumberDataList	import 項目 : 設定した倉庫番号データ情報を取得します。

サンプルプログラム

```

var ret;
var ret_msg;

function init(request) {

//-----
//変数宣言
//-----
var objCon;

```

```

var objResult;

var IN_MODE = "INS";
var IN_MATERIAL = "000000000000000788";
var IN_TEXT_NAME = "販売管理テスト品目(788-1110-1100-1A)";

//-----
//インスタンス生成 //説明 1
//-----
objCon = new MaterialSaveReplicaController();

//-----
// テスト実行
//-----
objCon.setTestrun(""); //書込 BAPI のシミュレーションセッションへの切替

//-----
// 単一項目セット //説明 2
//-----
objCon.setNoappllog("X"); //アプリケーションログ書込なし
objCon.setNochangedoc("X"); //アプリケーションログ書込なし
objCon.setInpfldcheck("E"); //項目無効時の応答

//-----
// テーブル項目セット //説明 3
//-----
//管理情報設定ヘッダセグメント
objCon.setTblParam("HeadData", "FUNCTION", IN_MODE, 1); //BAPI 機能
objCon.setTblParam("HeadData", "MATERIAL", IN_MATERIAL, 1); //品目コード
objCon.setTblParam("HeadData", "IND_SECTOR", "1", 1); //産業コード
objCon.setTblParam("HeadData", "MATL_TYPE", "HAWA", 1); //品目タイプ
objCon.setTblParam("HeadData", "BASIC_VIEW", "X", 1); //基本データビュー
objCon.setTblParam("HeadData", "SALES_VIEW", "X", 1); //販売ビュー
objCon.setTblParam("HeadData", "PURCHASE_VIEW", "X", 1); //購買管理ビュー
objCon.setTblParam("HeadData", "MRP_VIEW", "X", 1); //資材所要量計画 (MRP) ビュー
objCon.setTblParam("HeadData", "FORECAST_VIEW", "X", 1); //需要予測ビュー
objCon.setTblParam("HeadData", "WORK_SCHED_VIEW", "X", 1); //作業計画ビュー
objCon.setTblParam("HeadData", "PRT_VIEW", "X", 1); //生産資源/治工具 (PRT) ビュー
objCon.setTblParam("HeadData", "STORAGE_VIEW", "X", 1); //保管ビュー
objCon.setTblParam("HeadData", "WAREHOUSE_VIEW", "X", 1); //倉庫管理ビュー
objCon.setTblParam("HeadData", "QUALITY_VIEW", "X", 1); //品質管理ビュー
objCon.setTblParam("HeadData", "ACCOUNT_VIEW", "X", 1); //会計ビュー
objCon.setTblParam("HeadData", "COST_VIEW", "X", 1); //原価計算ビュー

//品目マスタ (クライアントレベル)
objCon.setTblParam("ClientDataList", "FUNCTION", IN_MODE, 1); //BAPI 機能
objCon.setTblParam("ClientDataList", "MATERIAL", IN_MATERIAL, 1); //品目コード
objCon.setTblParam("ClientDataList", "MATL_GROUP", "01", 1); //品目グループ
objCon.setTblParam("ClientDataList", "BASE_UOM", "ST", 1); //基本数量単位
objCon.setTblParam("ClientDataList", "SIZE_DIM", "100", 1); //サイズ/寸法
objCon.setTblParam("ClientDataList", "NET_WEIGHT", "200", 1); //正味重量
objCon.setTblParam("ClientDataList", "UNIT_OF_WT", "KG", 1); //重量単位
objCon.setTblParam("ClientDataList", "TRANS_GRP", "0001", 1); //輸送グループ
objCon.setTblParam("ClientDataList", "DIVISION", "01", 1); //製品部門

//品目データ (プラントレベル)
objCon.setTblParam("PlantDataList", "FUNCTION", IN_MODE, 1); //BAPI 機能
objCon.setTblParam("PlantDataList", "MATERIAL", IN_MATERIAL, 1); //品目コード
objCon.setTblParam("PlantDataList", "PLANT", "1110", 1); //プラント
objCon.setTblParam("PlantDataList", "MRP_TYPE", "ND", 1); //MRP タイプ
objCon.setTblParam("PlantDataList", "MRP_CTRLER", "001", 1); //MRP 管理者
objCon.setTblParam("PlantDataList", "PROC_TYPE", "X", 1); //調達タイプ
objCon.setTblParam("PlantDataList", "LOADINGGRP", "0001", 1); //積載グループ
objCon.setTblParam("PlantDataList", "AVAILCHECK", "CH", 1); //利用可能在庫確認の確認グループ
objCon.setTblParam("PlantDataList", "PROFIT_CTR", "PCA_DUMMY", 1); //利益センタ
objCon.setTblParam("PlantDataList", "MRP_GROUP", "0001", 1); //MRP グループ
objCon.setTblParam("PlantDataList", "SALES_VIEW", "X", 1); //販売ビュー
objCon.setTblParam("PlantDataList", "PURCH_VIEW", "X", 1); //購買管理ビュー
objCon.setTblParam("PlantDataList", "MRP_VIEW", "X", 1); //資材所要量計画 (MRP) ビュー
objCon.setTblParam("PlantDataList", "FORECAST_VIEW", "X", 1); //需要予測ビュー
objCon.setTblParam("PlantDataList", "WORK_SCHED_VIEW", "X", 1); //作業計画ビュー
objCon.setTblParam("PlantDataList", "PRT_VIEW", "X", 1); //生産資源/治工具 (PRT) ビュー
objCon.setTblParam("PlantDataList", "STORAGE_VIEW", "X", 1); //保管ビュー

```



```

objCon.setTblParam("PlantDataList", "WAREHOUSE_VIEW", "X", 1); //倉庫管理ビュー
objCon.setTblParam("PlantDataList", "QUALITY_VIEW", "X", 1); //品質管理ビュー
objCon.setTblParam("PlantDataList", "ACCOUNT_VIEW", "X", 1); //会計ビュー
objCon.setTblParam("PlantDataList", "COST_VIEW", "X", 1); //原価計算ビュー

//品目マスタ/製品グループの変更伝票構造
objCon.setTblParam("PlanningDataList", "FUNCTION", IN_MODE, 1); //BAPI 機能
objCon.setTblParam("PlanningDataList", "MATERIAL", IN_MATERIAL, 1); //品目コード
objCon.setTblParam("PlanningDataList", "PLANT", "1110", 1); //プラント
objCon.setTblParam("PlanningDataList", "PLNG_MATL", "0000000000000000100", 1); //計画品目
objCon.setTblParam("PlanningDataList", "PLNG_PLANT", "1110", 1); //計画プラント
objCon.setTblParam("PlanningDataList", "CONVFACTOR", "1", 1); //計画品目用変換係数

//品目データ (保管場所レベル)
objCon.setTblParam("StorageLocationDataList", "FUNCTION", IN_MODE, 1); //BAPI 機能
objCon.setTblParam("StorageLocationDataList", "MATERIAL", IN_MATERIAL, 1); //品目コード
objCon.setTblParam("StorageLocationDataList", "PLANT", "1110", 1); //プラント
objCon.setTblParam("StorageLocationDataList", "STGE_LOC", "1111", 1); //保管場所

//評価データ
objCon.setTblParam("ValuationDataList", "FUNCTION", IN_MODE, 1); //BAPI 機能
objCon.setTblParam("ValuationDataList", "MATERIAL", IN_MATERIAL, 1); //品目コード
objCon.setTblParam("ValuationDataList", "VAL_AREA", "1110", 1); //評価レベル
objCon.setTblParam("ValuationDataList", "STD_PRICE", "1000", 1); //標準原価
objCon.setTblParam("ValuationDataList", "PRICE_UNIT", "1", 1); //価格単位
objCon.setTblParam("ValuationDataList", "VAL_CLASS", "3100", 1); //評価クラス
objCon.setTblParam("ValuationDataList", "MATL_USAGE", "X", 1); //品目の用途
objCon.setTblParam("ValuationDataList", "ACCOUNT_VIEW", "X", 1); //会計ビュー
objCon.setTblParam("ValuationDataList", "COST_VIEW", "X", 1); //原価計算ビュー

//販売データ
objCon.setTblParam("SalesDataList", "FUNCTION", IN_MODE, 1); //BAPI 機能
objCon.setTblParam("SalesDataList", "MATERIAL", IN_MATERIAL, 1); //品目コード
objCon.setTblParam("SalesDataList", "SALES_ORG", "1100", 1); //販売組織
objCon.setTblParam("SalesDataList", "DISTR_CHAN", "1A", 1); //流通チャンネル
objCon.setTblParam("SalesDataList", "SALES_UNIT", "DZ", 1); //販売単位
objCon.setTblParam("SalesDataList", "DELYG_PLNT", "1110", 1); //出荷プラント

//品目テキストをセット
objCon.setTblParam("MaterialDescription", "FUNCTION", IN_MODE, 1); //BAPI 機能
objCon.setTblParam("MaterialDescription", "MATERIAL", IN_MATERIAL, 1); //品目コード
objCon.setTblParam("MaterialDescription", "LANGU", "J", 1); //言語キー
objCon.setTblParam("MaterialDescription", "MATL_DESC", IN_TEXT_NAME, 1); //品目テキスト

//数量単位
objCon.setTblParam("UnitSofmeasure", "FUNCTION", IN_MODE, 1); //BAPI 機能
objCon.setTblParam("UnitSofmeasure", "MATERIAL", IN_MATERIAL, 1); //品目コード
objCon.setTblParam("UnitSofmeasure", "ALT_UNIT", "DZ", 1); //在庫保管単位の代替数量単位
objCon.setTblParam("UnitSofmeasure", "NUMERATOR", "1", 1); //基本数量単位への換算分子
objCon.setTblParam("UnitSofmeasure", "DENOMINATR", "1", 1); //分母: 基本数量単位へ変換する場合
objCon.setTblParam("UnitSofmeasure", "GROSS_WT", "400", 1); //総重量
objCon.setTblParam("UnitSofmeasure", "UNIT_OF_WT", "KG", 1); //重量単位

//制御データ
objCon.setTblParam("TaxClassifications", "FUNCTION", IN_MODE, 1); //BAPI 機能
objCon.setTblParam("TaxClassifications", "MATERIAL", IN_MATERIAL, 1); //品目コード
objCon.setTblParam("TaxClassifications", "DEPCOUNTRY", "JP", 1); //出荷国 (商品発送国)
objCon.setTblParam("TaxClassifications", "TAX_TYPE_1", "MWST", 1); //税カテゴリ (消費税等)
objCon.setTblParam("TaxClassifications", "TAXCLASS_1", "1", 1); //品目の税分類

//-----
// executeBapi実行 //説明 4
//-----
objResult = objCon.executeBapi();

//-----
// エラー判定
//-----
if(objResult["ERR_CODE"] != null) {
    //インスタンス生成時に例外が発生した場合
    //エラーメッセージを出力
    Debug.browse(objResult);
}

```

```
//API実行結果を取得します。 //説明5
ret = objCon.getBapiReturnMessage();
ret_msg = objCon.getReturnMessagesList();
}
```

- 品目マスタ登録 API のインスタンス生成(説明1)
MaterialSaveReplicaControllerクラスのインスタンスを生成します。
引数は不要です。
インスタンスを生成すると、品目マスタ登録BAPI (BAPI_MATERIAL_SAVEREPLICA) のimport型のimport、table パラメータに値を設定することができます。
- import の各項目に値を設定(説明2)
import パラメータの項目に値を設定する場合は、set メソッドを使用します。
メソッドについては各 API のAPI リストを参照ください。
- table の各項目に値を設定(説明3)
table パラメータの項目に値を設定する場合は、setTblParam メソッドを使用します。
第1引数のtable名称は固定値になっています。
後述のとおり名称を設定して下さい。
第2引数の項目名称は SAP の各属性の項目名称を設定してください。
table 名称、項目名称は英字の大文字、小文字は厳密に区別されます。
第3引数に値、第4引数にレコード番号を設定します。
品目マスタ登録 BAPI (BAPI_MATERIAL_SAVEREPLICA) の import 型の table パラメータには変更ツールバーが存在しますが、各項目に値を設定することにより、品目マスタ登録APIの内部で変更ツールバーの各項目にフラグを設定します。

SAP table	セットする固有名称
CLIENTDATA (品目データ (クライアントレベル))	ClientDataList
EXTENSIONIN (BAPIパラメータ 「EXTENSIONIN/EXTENSIONOUT」 の参照構造)	ExtensionIn
FORECASTPARAMETERS (需要予測パラメータ)	ForecastParametersList
FORECASTVALUES (需要予測値のデータ転送)	ForecastValues
HEADDATA (管理情報設定ヘッダセグメント)	HeadData
INTERNATIONALARTNOS (国際商品コード (EAN))	InternationalArtnos
MATERIALDESCRIPTION (品目テキスト)	MaterialDescription
MATERIALLONGTEXT (テキスト (長))	MaterialLongText
PLANNINGDATA (品目マスタ/製品グループの変更伝票構造)	PlanningDataList
PLANTDATA (品目データ (プラントレベル))	PlantDataList
PRTDATA (品目マスタの生産資源/治工具(PRT) 項目)	PrtData
SALESDATA (販売データ)	SalesDataList
STORAGELOCATIONDATA (品目データ (保管場所レベル))	StorageLocationDataList
STORAGETYPEDATA (保管域タイプデータ)	StorageTypeDataList
TAXCLASSIFICATIONS (制御データ)	TaxClassifications
TOTALCONSUMPTION (品目総消費量のデータ転送)	TotalConsumption
UNITSOFMEASURE (数量単位)	UnitSofmeasure
UNPLNDCONSUMPTION (計画外消費のデータ転送)	UnplndConsumption
VALUATIONDATA (評価データ)	ValuationDataList
WAREHOUSENUMBERDATA (倉庫番号データ)	WarehouseNumberDataList

- 品目マスタの登録(説明4)
MaterialSaveReplicaController クラスの executeBapi メソッドを実行すると、各項目にセットされた値が SAP上に品目マスタデータとして登録されます。
登録が成功した場合には、品目マスタ登録 API の内部で自動的に commit 処理が実行されます。

登録に失敗した場合には、自動的に rollback 処理が実行されます。

成功・失敗の基準は、品目マスタ登録用 BAPI (BAPI_MATERIAL_SAVEREPLICA) の実行後に RETURN パラメータの TYPE 値を取得して、S (正常終了)、空文字以外が 1 つでも存在するかどうかです。

1 つでも存在する場合は、rollback が自動的に実行されます。

- サンプルプログラムの中で記載されたデータ取得メソッドの詳細説明については、以下を参照してください。(説明5)
 - Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn を取得
MaterialSaveReplicaController クラスの getBapiReturnErrMessage メソッドを使用すると、取得したBapiReturn の Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn オブジェクトのメッセージ情報を取得することが可能です。
全て正常の場合は null を取得します。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - Type 情報が正常(デフォルト値: "S" / "")の BapiReturn を取得
MaterialSaveReplicaController クラスの getBapiReturnMessage メソッドを使用すると、取得したBapiReturn の Type 情報が正常(デフォルト値: "S" / "")と判断された Type 情報のメッセージ情報を取得することが可能です。
メッセージが存在しない場合は null を取得します。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - CAD システムからコール情報を取得
MaterialSaveReplicaController クラスの getFlagCadCall メソッドを使用すると、設定した CAD システムからコール情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 項目無効時の応答情報を取得
MaterialSaveReplicaControllerクラスのgetInpFldcheckメソッドを使用すると、設定した項目無効時の応答情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - アプリケーションログ書込なし情報を取得
MaterialSaveReplicaControllerクラスのgetNoappllogメソッドを使用すると、設定したアプリケーションログ書込なし情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 変更伝票書込なし情報を取得
MaterialSaveReplicaController クラスの getNochangedoc メソッドを使用すると、設定した変更伝票書込なし情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 書込 BAPI のシミュレーションセッションへの切替情報を取得
MaterialSaveReplicaController クラスの getTestrun メソッドを使用すると、設定した書込 BAPI のシミュレーションセッションへの切替情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - Override Rollback Work 情報を取得
MaterialSaveReplicaController クラスの getNoRollbackWork メソッドを使用すると、設定した Override Rollback Work 情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - No ALE Field Selection 情報を取得
MaterialSaveReplicaController クラスの getFlagOnline メソッドを使用すると、設定した No ALE Field Selection 情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 品目データ (クライアントレベル)情報を取得
MaterialSaveReplicaController クラスの getClientDataList メソッドを使用すると、設定した品目データ(クライアントレベル)情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - BAPI パラメータ EXTENSIONIN/EXTENSIONOUT の参照構造情報を取得
MaterialSaveReplicaController クラスの getExtensionInList メソッドを使用すると、設定した BAPI パラメータ EXTENSIONIN/EXTENSIONOUT の参照構造情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 拡張イン/拡張アウトのチェックボックス構造情報を取得
MaterialSaveReplicaController クラスの getExtensionInXList メソッドを使用すると、設定した拡張イン/拡張アウトのチェックボックス構造情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 需要予測パラメータ情報を取得
MaterialSaveReplicaController クラスの getForecastParametersList メソッドを使用すると、設定した需要予測パラメータ情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 需要予測値のデータ転送情報を取得
MaterialSaveReplicaControllerクラスのgetForecastValuesListメソッドを使用すると、設定した需要予測値のデータ転送情

報を取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

- 管理情報設定ヘッダセグメント情報を取得
MaterialSaveReplicaController クラスの getHeadDataList メソッドを使用すると、設定した管理情報設定ヘッダセグメント情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 国際商品コード (EAN)情報を取得
MaterialSaveReplicaController クラスの getInternationalArtnosList メソッドを使用すると、設定した国際商品コード (EAN)情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 品目テキスト情報を取得
MaterialSaveReplicaController クラスの getMaterialDescriptionList メソッドを使用すると、設定した品目テキスト情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- テキスト (長)情報を取得
MaterialSaveReplicaController クラスの getMaterialLongTextList メソッドを使用すると、設定したテキスト (長)情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 品目マスタ/製品グループの変更伝票構造情報を取得
MaterialSaveReplicaController クラスの getPlanningDataList メソッドを使用すると、設定した品目マスタ/製品グループの変更伝票構造情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 品目データ (プラントレベル)情報を取得
MaterialSaveReplicaController クラスの getPlantDataList メソッドを使用すると、設定した品目データ(プラントレベル)情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 品目マスタの生産資源/治工具 (PRT) 項目情報を取得
MaterialSaveReplicaController クラスの getPrtDataList メソッドを使用すると、設定した品目マスタの生産資源/治工具 (PRT) 項目情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- リターンパラメータ BAPIRET2 の代替構造情報を取得
MaterialSaveReplicaController クラスの getReturnMessagesList メソッドを使用すると、取得したリターンパラメータ BAPIRET2 の代替構造情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 販売データ情報を取得
MaterialSaveReplicaController クラスの getSalesDataList メソッドを使用すると、設定した販売データ情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 品目データ (保管場所レベル)情報を取得
MaterialSaveReplicaController クラスの getStorageLocationDataList メソッドを使用すると、設定した品目データ (保管場所レベル)情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 保管域タイプデータ情報を取得
MaterialSaveReplicaController クラスの getStorageTypeDataList メソッドを使用すると、設定した保管域タイプデータ情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 制御データ情報を取得
MaterialSaveReplicaController クラスの getTaxClassificationsList メソッドを使用すると、設定した制御データ情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 品目総消費量のデータ転送情報を取得
MaterialSaveReplicaController クラスの getTotalConsumptionList メソッドを使用すると、設定した品目総消費量のデータ転送情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 数量単位情報を取得
MaterialSaveReplicaController クラスの getUnitSofmeasureList メソッドを使用すると、設定した数量単位情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 計画外消費のデータ転送情報を取得
MaterialSaveReplicaController クラスの getUnplndConsumptionList メソッドを使用すると、設定した計画外消費のデータ

転送情報を取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

- 評価データ情報を取得

MaterialSaveReplicaController クラスの getValuationDataList メソッドを使用すると、設定した評価データ情報を取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

- 倉庫番号データ情報を取得

MaterialSaveReplicaController クラスの getWarehouseNumberDataList メソッドを使用すると、設定した倉庫番号データ情報を取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。



注意

品目マスタ登録 API に関して、import、table パラメータを持つ各コンポーネントの必須項目や値の相関チェック等を確認する必要がある場合は、SAP 上の汎用モジュールビルダ (t-cd : SE37) 等を使用して、「BAPI_MATERIAL_SAVEREPLICA」の BAPI 仕様を確認してください。

購買伝票承認対象一覧取得 API (PurchaseGetItemSrelController)

購買伝票承認対象一覧を取得するためには、PurchaseGetItemSrelController クラスの持つ各メソッドを利用する必要があります。

引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
PurchaseGetItemSrelController	コンストラクタ (第 1~3 引数あり)。 第 1 引数に照会する承認グループ、第 2 引数に承認コード、第 3 引数に承認待ち購買発注のみ照会区分を設定する。
getPurchaseHeadersList	伝票ヘッダー一覧を取得する。
getPurchaseItemsList	伝票明細一覧を取得する。

サンプルプログラム

```

//■品目コード一覧
var headerList;
var headerCount;
var itemList;
var itemCount;

function init(request) {
    var objCon;

    //インスタンス生成
    objCon = new PurchaseGetItemSrelController("01", "AA", "X"); // 説明 1

    // コンストラクタでのエラーチェック
    if (objCon.getErrCode() != "0"){
        //エラーメッセージ出力
        Debug.browse(objCon.getErrMsg());
    }

    //オブジェクトをグローバル領域に保存
    //伝票ヘッダー一覧
    headerList = objCon.getPurchaseHeadersList(); // 説明 2

    //伝票明細一覧
    itemList = objCon.getPurchaseItemsList(); // 説明 3

    //オブジェクトを取得
    //伝票ヘッダ
    headerCount = headerList.length;

    //伝票明細
    itemCount = itemList.length;
}

```

- 伝票承認対象一覧取得 API のインスタンス生成(説明1)

PurchaseGetItemSrelController クラスのインスタンスを生成します。

第1引数に照会する承認グループ、第2引数に承認コード、第3引数に承認待ち購買発注のみ照会区分を設定します。

インスタンスを生成すると、BAPIを呼び出して伝票承認対象一覧取得処理が実行されます。

承認待ち購買発注のみ照会区分は、承認待ち購買発注伝票のみ照会する場合には「X」、すでに承認された購買発注伝票のみ照会する場合には、ブランクを設定します。
- 伝票ヘッダー一覧を取得(説明2)

PurchaseGetItemSrelController クラスの getPurchaseHeadersList メソッドを使用すると、取得した承認対象の伝票ヘッダー一覧を取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 伝票ヘッダー一覧を取得(説明3)

PurchaseGetItemSrelController クラスの getPurchaseItemsList メソッドを使用すると、取得した承認対象の伝票明細一覧を取得することが可能です。

項目の値を取得するには、KEYにSAPの各属性の項目名称を設定してください。

購買伝票登録 API (PurchaseOrderPayPostController)

購買伝票を登録するためには、PurchaseOrderPayPostController クラスの持つ各メソッドを利用する必要があります。

引数、戻り値についての詳細な情報は、API リストを参照してください。

また、購買伝票登録用 BAPI (BAPI_PO_CREATE1) の項目すべてにおいて登録が可能です。

各メソッドの説明

コンストラクタ / メソッド名	説明
PurchaseOrderPayPostController	コンストラクタ (引数無し)

コンストラクタ / メソッド名	説明
setStrParam	構造型の各項目に値を設定する。 第一引数に structure 名、第二引数に項目名、第三引数に値を設定する。
setTblParam	テーブル型の各項目に値を設定する。 第1引数にテーブル名、第2引数に項目名、第3引数に値、第4引数にレコード番号を設定する。
executeBapi	上記のsetメソッドで設定した値を、接続先の SAP に登録する。 登録に失敗した場合、登録処理はロールバックされる。
getPoNumber	伝票登録後の購買伝票番号を取得する。
getDocumentHeader	伝票登録後の伝票ヘッダの結果を取得する。
getExportHeader	伝票登録後の貿易管理ヘッダの結果を取得する。
getExtensionList	伝票登録後の得意先拡張項目の結果を取得する。
getAllVersionsList	伝票登録後の全バージョンの結果を取得する。
getPurchaseShippingList	伝票登録後の出荷データの結果を取得する。
getPurchaseShippingExpList	伝票登録後の出荷データエクスポートの結果を取得する。
getPurchaseComponentsList	伝票登録後のコンポーネントの結果を取得する。

サンプルプログラム

```

//■伝票番号
var strPo_number;
//■ヘッダデータ
var strCompCode;
var strDocType;
var strStatus;
var strCreateDate;
var strCreateBy;
var strVendor;
var strPmnttrms;
var strPurchOrg;
var strPurGroup;
var strCurrency;
var strDocDate;
var strPoRelInd;
var strRelStatus;
//■貿易管理ヘッダ
var strTransportMode;
var strTCustoms;
//■伝票明細
var itemList;
var itemCount;
//■拡張得意先項目一覧
var extensionList;
var extensionCount;
//■全バージョン一覧
var allVersionList;
var allVersionCount;

function init(request) {
    var objCon;
    var objResult;
    var header;
    var expoHeader;

    //インスタンス生成
    //伝票ヘッダstructureの各項目に値を設定
    objCon = new PurchaseOrderPayPostController(); // 説明1

    //伝票ヘッダの structure の各項目に値を設定 // 説明2

```

```
objCon.setStrParam("PurchaseHeaderPost","COMP_CODE","1000");
objCon.setStrParam("PurchaseHeaderPost","DOC_TYPE","NB");
objCon.setStrParam("PurchaseHeaderPost","VENDOR","0000100011");
objCon.setStrParam("PurchaseHeaderPost","PMNTRMS","1000");
objCon.setStrParam("PurchaseHeaderPost","PURCH_ORG","1200");
objCon.setStrParam("PurchaseHeaderPost","PUR_GROUP","001");
objCon.setStrParam("PurchaseHeaderPost","CURRENCY","JPY");
```

//仕入先住所structureの各項目に値を設定

```
objCon.setStrParam("VendorAddress","REGION","埼玉県");
objCon.setStrParam("VendorAddress","POSTL_COD1","3650039");
objCon.setStrParam("VendorAddress","CITY","鴻巣市1-5-31");
objCon.setStrParam("VendorAddress","COUNTRY","JP");
objCon.setStrParam("VendorAddress","TEL_NUMBR","048-539-6790");
objCon.setStrParam("VendorAddress","FAX_NUMBER","048-539-6790");
```

//明細tableの各項目に値を設定 // 説明3

// 1 明細

```
objCon.setTblParam("PurchaseItemPost","PO_ITEM","00010",1);
// objCon.setTblParam("PurchaseItemPost","SHORT_TEXT","",1);
objCon.setTblParam("PurchaseItemPost","MATERIAL","0000000000000000057",1);
objCon.setTblParam("PurchaseItemPost","PLANT","1220",1);
objCon.setTblParam("PurchaseItemPost","STGE_LOC","1112",1);
objCon.setTblParam("PurchaseItemPost","TRACKINGNO","追跡番号",1);
objCon.setTblParam("PurchaseItemPost","MATL_GROUP","01",1);
objCon.setTblParam("PurchaseItemPost","QUANTITY","300",1);
objCon.setTblParam("PurchaseItemPost","PO_UNIT","ST",1);
objCon.setTblParam("PurchaseItemPost","NET_PRICE","30000",1);
objCon.setTblParam("PurchaseItemPost","PRICE_UNIT","1",1);
objCon.setTblParam("PurchaseItemPost","TAX_CODE","V1",1);
objCon.setTblParam("PurchaseItemPost","NO_MORE_GR","",1);
objCon.setTblParam("PurchaseItemPost","ACCTASSCAT","K",1);
objCon.setTblParam("PurchaseItemPost","DISTRIB","",1);
objCon.setTblParam("PurchaseItemPost","GR_IND","X",1);
objCon.setTblParam("PurchaseItemPost","IR_IND","",1);
objCon.setTblParam("PurchaseItemPost","GR_BASEDIV","",1);
objCon.setTblParam("PurchaseItemPost","PREQ_NAME","api登録",1);
```

//2 明細

```
objCon.setTblParam("PurchaseItemPost","PO_ITEM","00020",2);
objCon.setTblParam("PurchaseItemPost","SHORT_TEXT","プリンタ1",2);
objCon.setTblParam("PurchaseItemPost","MATERIAL","0000000000000000039",2);
objCon.setTblParam("PurchaseItemPost","PLANT","1220",2);
objCon.setTblParam("PurchaseItemPost","STGE_LOC","1112",2);
objCon.setTblParam("PurchaseItemPost","TRACKINGNO","1Group",2);
objCon.setTblParam("PurchaseItemPost","MATL_GROUP","01",2);
objCon.setTblParam("PurchaseItemPost","QUANTITY","2",2);
objCon.setTblParam("PurchaseItemPost","PO_UNIT","ST",2);
objCon.setTblParam("PurchaseItemPost","NET_PRICE","10000",2);
objCon.setTblParam("PurchaseItemPost","PRICE_UNIT","1",2);
objCon.setTblParam("PurchaseItemPost","TAX_CODE","V1",2);
objCon.setTblParam("PurchaseItemPost","ACCTASSCAT","K",2);
objCon.setTblParam("PurchaseItemPost","GR_IND","X",2);
```

//納入日程tableの各項目に値を設定

```
objCon.setTblParam("PurchaseSchedule","PO_ITEM","00010",1);
objCon.setTblParam("PurchaseSchedule","SCHED_LINE","0001",1);
objCon.setTblParam("PurchaseSchedule","DEL_DATCAT_EXT","D",1);
```

```
objCon.setTblParam("PurchaseSchedule","PO_ITEM","00020",2);
objCon.setTblParam("PurchaseSchedule","SCHED_LINE","0001",2);
objCon.setTblParam("PurchaseSchedule","DEL_DATCAT_EXT","D",2);
```

//勘定設定tableの各項目に値を設定

```
objCon.setTblParam("PurchaseAccount","PO_ITEM","00010",1);
objCon.setTblParam("PurchaseAccount","SERIAL_NO","01",1);
objCon.setTblParam("PurchaseAccount","QUANTITY","100",1);
objCon.setTblParam("PurchaseAccount","GL_ACCOUNT","0000826700",1);
objCon.setTblParam("PurchaseAccount","COSTCENTER","1210000000",1);
```

```
objCon.setTblParam("PurchaseAccount","PO_ITEM","00020",2);
objCon.setTblParam("PurchaseAccount","SERIAL_NO","01",2);
objCon.setTblParam("PurchaseAccount","QUANTITY","2",2);
objCon.setTblParam("PurchaseAccount","GL_ACCOUNT","0000826700",2);
```



```
objCon.setTblParam("PurchaseAccount","COSTCENTER","1210000000",2);
```

```
//明細テキストtableの各項目に値を設定
```

```
objCon.setTblParam("ItemText","PO_ITEM","00010",1);
objCon.setTblParam("ItemText","TEXT_ID","F01",1);
objCon.setTblParam("ItemText","TEXT_LINE","APIテキスト1明細",1);
```

```
objCon.setTblParam("ItemText","PO_ITEM","00020",2);
objCon.setTblParam("ItemText","TEXT_ID","F01",2);
objCon.setTblParam("ItemText","TEXT_LINE","APIテキスト2明細",2);
```

```
//会計伝票の登録
```

```
objResult = objCon.executeBapi(); // 説明4
```

```
//要素[ERR_CODE]に何らかの値がセットされている場合
```

```
if (objResult["ERR_CODE"] != null) {
    //エラーメッセージ出力
    Debug.browse(objResult);
}
```

```
//オブジェクトを取得
```

```
//伝票番号
```

```
strPo_number = objCon.getPoNumber(); // 説明5
```

```
//伝票ヘッダ
```

```
header = objCon.getDocumentHeader(); // 説明6
```

```
strCompCode = header.COMP_CODE;
strDocType = header.DOC_TYPE;
strStatus = header.STATUS;
strCreateDate = header.CREAT_DATE;
strCreateBy = header.CREATED_BY;
strVendor = header.VENDOR;
strPmnttrms = header.PMNTTRMS;
strPurchOrg = header.PURCH_ORG;
strPurGroup = header.PUR_GROUP;
strCurrency = header.CURRENCY;
strDocDate = header.DOC_DATE;
strPoRelInd = header.PO_REL_IND;
strRelStatus = header.REL_STATUS;
```

```
//貿易管理ヘッダ
```

```
expoHeader = objCon.getExportHeader();
strTransportMode = expoHeader.TRANSPORT_MODE;
strTcustoms = expoHeader.CUSTOMS;
```

```
//伝票明細
```

```
itemList = objCon.getPurchaseItemPostList();
itemCount = itemList.length;
```

```
//拡張得意先項目
```

```
extensionList = objCon.getExtensionList(); // 説明7
extensionCount = extensionList.length;
```

```
//全バージョン
```

```
allVersionList = objCon.getAllVersionsList();
allVersionCount = allVersionList.length;
```

```
}
```

- 購買伝票登録 API のインスタンス生成(説明1)
PurchaseOrderPayPostController クラスのインスタンスを生成します。
引数は不要です。
インスタンスを生成すると、購買伝票登録 BAPI (BAPI_PO_CREATE1) のimport型のimport、structure、tableパラメータに値を設定することができます。
- structure の各項目に値を設定(説明2)
structure パラメータの項目に値を設定する場合は、setStrParam メソッドを使用します。
第一引数の structure 名称は固定値になっています。
後述のとおり名称を設定して下さい。
第二引数の項目名称は SAP の各属性の項目名称を設定してください。structure 名称、項目名称は英字の大文字、小文字は厳密に区別されます。
第三引数に値を設定します。

購買伝票登録 BAPI (BAPI_PO_CREATE1) の import 型の structure パラメータには変更ツールバーが存在しますが、各項目に値を設定することにより、購買伝票登録 API の内部で変更ツールバーの各項目にフラグを設定します。

- table の各項目に値を設定(説明3)

table パラメータの項目に値を設定する場合は、setTblParam メソッドを使用します。

第一引数の table 名称は固定値になっています。

後述のとおり名称を設定して下さい。

第二引数の項目名称は SAPの各属性の項目名称を設定してください。

table 名称、項目名称は英字の大文字、小文字は厳密に区別されます。

第三引数に値、第四引数にレコード番号を設定します。

購買伝票登録 BAPI (BAPI_PO_CREATE1) のimport型の table パラメータには変更ツールバーが存在しますが、各項目に値を設定することにより、購買伝票登録 API の内部で変更ツールバーの各項目にフラグを設定します。

SAP structure	セットする固有名称
POHEADER(伝票ヘッダ)	PurchaseOrderPayPost
POADDRVENDOR(仕入先住所)	VendorAddress
POEXPIMPHEADER(貿易管理:ヘッダデータ)	ExportHeader
VERSIONS(バージョン管理)	Versions

- 購買伝票の登録(説明4)

PurchaseOrderPayPostControllerクラスのexecuteBapiメソッドを実行すると、各項目にセットされた値がSAP上に購買伝票データとして登録されます。

登録が成功した場合には、購買伝票登録APIの内部で自動的にcommit処理が実行されます。

登録に失敗した場合には、自動的にrollback処理が実行されます。

成功・失敗の基準は、購買伝票登録用BAPI (BAPI_PO_CREATE1) の実行後にRETURNパラメータのTYPE値を取得して、S (正常終了)、空文字以外が1つでも存在するかどうかです。

1つでも存在する場合は、rollbackが自動的に実行されます。

SAP table	セットする固有名称
POITEM(明細データ)	PurchaseItemPost
POADDRDELIVERY(入荷伝票のアドレス)	DeliveryAddress
POSCHEDULE(納入日程)	PurchaseSchedule
POACCOUNT(勘定設定項目)	PurchaseAccount
POACCOUNTPROFITSEGMENT 入出庫予定イベントオブジェクト	ProfitSegment
POCONDHEADER(条件 (ヘッダ))	CondHeader
POCOND(条件 (明細))	CondItem
POLIMITS(外注サービス:制限)	Limits
POCONTRACTLIMITS(外注サービス:基本契約制限)	ContractLimits
POSERVICES(外注サービス:サービス行)	ContractService
POSRVACCESSVALUES(外注サービス:サービス行の勘定設定)	ContractServiceAccount
POSERVICESTEXT(外注サービス:サービステキスト (長))	ContractServiceText
EXTENSIONIN(得意先独自の項目)	Extension
POEXPIMPITEM(貿易管理:明細データ)	ExportItem
POTEXTHEADER(ヘッダテキスト)	HeaderText
POTEXTITEM(明細テキスト)	ItemText
POPARTNER(取引先)	Partner
POSHIPPING(在庫転送オーダーの 出荷データ)	PurchaseShipping
POSHIPPINGEXP(出荷データのエクスポート)	PurchaseShippingExp
POCOMPONENTS(コンポーネントの BAPI 構造)	PurchaseComponents

- 購買伝票番号を取得(説明5)

PurchaseOrderPayPostControllerクラスのgetPoNumberメソッドを使用すると、登録された購買伝票の伝票番号を取得することが可能です。
- structureデータを取得(説明6)

伝票ヘッダ、貿易管理ヘッダについては、PurchaseOrderPayPostControllerクラスのexecuteメソッドを実行すると購買伝票登録APIの内部で自動的に登録結果を取得します。

伝票ヘッダはgetDocumentHeaderメソッド、貿易管理ヘッダはgetExportHeaderメソッドで取得が可能です。

その他のstructureについては、入力設定した項目について取得が可能です。項目の値を取得するには、KEYにSAPの各属性の項目名称を設定してください。
- tableデータを取得(説明7)

得意先独自の項目、全バージョン、出荷データのエクスポートについては、PurchaseOrderPayPostControllerクラスのexecuteメソッドを実行すると購買伝票登録APIの内部で自動的に登録結果を取得します。

得意先独自の項目はgetExtensionListメソッド、全バージョンはgetAllVersionsListメソッド、出荷データのエクスポートはgetPurchaseShippingExpListメソッドで取得が可能です。その他のtableについては、入力設定した項目について取得が可能です。項目の値を取得するには、KEYにSAPの各属性の項目名称を設定してください。

サンプルプログラム実行結果では、全バージョンの登録結果が取得されていますが、理解しやすいようにSAPのカスタマイズにより、バージョン管理をしている為です。

SAP標準では該当データが存在しません。

その他のstructure、tableデータについても、SAP標準ではデータが取得できない場合があります。

購買伝票 更新 API (PurchaseOrderPostChangeController)

購買伝票を更新するためには、PurchaseOrderPostChangeController クラスの持つ各メソッドを利用する必要があります。引数、戻り値についての詳細な情報は、API リストを参照してください。

また、購買伝票登録用 BAPI (BAPI_PO_CHANGE) の項目すべてにおいて更新が可能です。

各メソッドの説明

コンストラクタ / メソッド名	説明
PurchaseOrderPostChangeController	コンストラクタ (引数無し)
setPoNumber	更新する購買伝票番号を設定する。
setStrParam	構造型の各項目に値を設定する。 第1引数に structure 名、第2引数に項目名、第3引数に値を設定する。
setTblParam	テーブル型の各項目に値を設定する。 第1引数にテーブル名、第2引数に項目名、第3引数に値、第4引数にレコード番号を設定する。
executeBapi	上記のsetメソッドで設定した値を、接続先の SAP に更新する。 更新に失敗した場合、更新処理はロールバックされる。
getDocumentHeader	伝票更新後の伝票ヘッダの結果を取得する。
getExportHeader	伝票更新後の貿易管理ヘッダの結果を取得する。
getExtensionList	伝票更新後の得意先拡張項目の結果を取得する。
getAllVersionsList	伝票更新後の全バージョンの結果を取得する。
getPurchaseShippingList	伝票登録後の出荷データの結果を取得する。
getPurchaseShippingExpList	伝票登録後の出荷データエクスポートの結果を取得する。
getPurchaseComponentsList	伝票登録後のコンポーネントの結果を取得する。
getPurchaseConfirmationList	伝票登録後の転送構造：照会／一覧の結果を取得する。
getPurchaseHistoryList	伝票登録後の購買発注履歴の転送構造の結果を取得する。
getPurchaseHistoryTotalsList	伝票登録後の購買発注履歴の転送構造: 合計の結果を取得する。

```

//■伝票番号
var strPo_number;
//■ヘッダデータ
var strCompCode;
var strDocType;
var strStatus;
var strCreateDate;
var strCreateBy;
var strVendor;
var strPmnttrms;
var strPurchOrg;
var strPurGroup;
var strCurrency;
var strDocDate;
var strPoRelInd;
var strRelStatus;
//■伝票明細
var itemList;
var itemCount;

function init(request) {

    var objCon;
    var objResult;
    var header;

    //インスタンス生成
    objCon = new PurchaseOrderPostChangeController(); // 説明 1

    //伝票番号importの各項目に値を設定
    objCon.setPoNumber("4500000011"); // 説明 2

    //明細tableの各項目に値を設定
    objCon.setTblParam("PurchaseItemPost","PO_ITEM","00010",1);
    objCon.setTblParam("PurchaseItemPost","MATERIAL","0000000000000000057",1);
    objCon.setTblParam("PurchaseItemPost","QUANTITY","320",1);
    objCon.setTblParam("PurchaseItemPost","PO_UNIT","ST",1); // 説明 3

    //会計伝票の登録
    objResult = objCon.executeBapi(); // 説明 4

    //要素[ERR_CODE]に何らかの値がセットされている場合
    if (objResult["ERR_CODE"] != null) {

        //要素[ERR_CODE]に何らかの値がセットされている場合
        if (objResult["ERR_CODE"] != null) {
            //エラーメッセージ出力
            Debug.browse(objResult);
        }

        //オブジェクトを取得
        //伝票番号
        strPo_number = objCon.getPoNumber();

        header = objCon.getDocumentHeader(); // 説明 5

        // 省略
    }
}

```

- 購買伝票更新 API のインスタンス生成(説明1)
PurchaseOrderPostChangeController クラスのインスタンスを生成します。
引数は不要です。
インスタンスを生成すると、購買伝票更新 BAPI (BAPI_PO_CHANGE) の import 型の import、structure、table パラメータに値を設定することができます。
- import の購買伝票番号に値を設定(説明2)
import パラメータの購買伝票番号に更新する伝票番号の値を設定します。
- structure、table の各項目に値を設定(説明3)

structure パラメータ、table パラメータの各項目に値を設定します。

SAP structure	セットする固有名称
POITEM(明細データ)	PurchaseItemPost
POADDRDELIVERY(入荷伝票のアドレス)	DeliveryAddress
POSCHEDULE(納入日程)	PurchaseSchedule
POACCOUNT(勘定設定項目)	PurchaseAccount
POACCOUNTPROFITSEGMENT 入出庫予定イベントオブジェクト	ProfitSegment
POCONDHEADER(条件 (ヘッダ))	CondHeader
POCOND(条件 (明細))	CondItem
POLIMITS(外注サービス:制限)	Limits
POCONTRACTLIMITS(外注サービス:基本契約制限)	ContractLimits
POSERVICES(外注サービス:サービス行)	ContractService
POSRVACCESSVALUES(外注サービス:サービス行の勘定設定)	ContractServiceAccount
POSERVICESTEXT(外注サービス:サービステキスト (長))	ContractServiceText
EXTENSIONIN(得意先独自の項目)	Extension
POEXPIMPITEM(貿易管理:明細データ)	ExportItem
POTEXTHEADER(ヘッダテキスト)	HeaderText
POTEXTITEM(明細テキスト)	ItemText
POPARTNER(取引先)	Partner
POSHIPPING(在庫転送オーダーの 出荷データ)	PurchaseShipping
POSHIPPINGEXP(出荷データのエクスポート)	PurchaseShippingExp
POCOMPONENTS(コンポーネントの BAPI 構造)	PurchaseComponents

SAP table	セットする固有名称
POCONFIRMATION (転送構造: 照会/一覧 - 仕入先確)	PurchaseConfirmation
POHISTORY (購買発注履歴の転送構造)	PurchaseHistory
POHISTORY_TOTALS (購買発注履歴の転送構造: 合計)	PurchaseHistoryTotals

- 購買伝票の更新(説明4)

PurchaseOrderPostChangeController クラスのexecuteメソッドを実行すると、各項目にセットされた値がSAP上に購買伝票データとして更新されます。

更新が成功した場合には、購買伝票更新APIの内部で自動的に commit 処理が実行されます。

登録に失敗した場合には、自動的に rollback 処理が実行されます。

成功・失敗の基準は、購買伝票更新用 BAPI (BAPI_PO_CHANGE) の実行後に RETURN パラメータのTYPE 値を取得して、S (正常終了)、空文字以外が1つでも存在するかどうかです。

1つでも存在する場合は、rollback が自動的に実行されます。
- structure、table データを取得(説明5)

PurchaseOrderPostChangeController クラスの execute メソッドを実行すると、購買伝票更新 API の内部で自動的に登録結果を取得します。

返却されるデータは、伝票ヘッダ、貿易管理ヘッダ、得意先独自の項目、全バージョン、出荷データのエクスポート、購買発注履歴の転送構造、購買発注履歴の転送構造：合計、転送構造：照会/一覧-仕入先確認です。

その他の項目については、入力設定した項目について取得が可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

サンプルプログラム実行結果では、全バージョンの登録結果が取得されていますが、理解しやすいようにSAPのカスタマイズにより、バージョン管理をしている為です。

SAP 標準では該当データが存在しません。

伝票 承認 API (PurchaseReleaseController)

伝票を承認するためには、PurchaseReleaseController クラスの持つ各メソッドを利用する必要があります。
引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
PurchaseReleaseController	コンストラクタ（第 1～3 引数あり）。 第1引数に承認する伝票番号、第2引数に承認コード、第3引数に例外メッセージ出力区分を設定する。
PurchaseReleaseController	コンストラクタ（第 1～4 引数あり）。 第1引数に承認する伝票番号、第2引数に承認コード、第3引数に例外メッセージ出力区分、第4引数にコミットワーク（true/false）を設定する。
executeBapi	コンストラクタで設定した値を、接続先の SAP に更新する。 更新に失敗した場合、更新処理はロールバックされる。
getRelStatusNew	承認後の新承認状況を取得する。
getRelIndicatorNew	承認後の新承認区分を取得する。
getRetCode	承認後のリターン値を取得する。

サンプルプログラム

```
//承認状況
var strStatus;
//承認区分
var strRelInd;
//リターン値
var strRetCode;

function init(request) {

    var objCon;
    var objResult;

    //インスタンス生成
    objCon = new PurchaseReleaseController("4500000011", "AA", "X");// 説明 1

    //購買伝票の承認処理の実行
    objResult = objCon.executeBapi(); // 説明 2

    //要素[ERR_CODE]に何らかの値がセットされている場合
    if (objResult["ERR_CODE"] != null) {
        //エラー情報を出力する。
        Debug.browse(objResult);
    }

    //オブジェクトをグローバル領域に保存
    strStatus = objCon.getRelStatusNew(); // 説明 3
    strRelInd = objCon.getRelIndicatorNew(); // 説明 4
    strRetCode = objCon.getRetCode(); // 説明 5
}
```

- 伝票承認 API のインスタンス生成(説明1)
PurchaseReleaseController クラスのインスタンスを生成します。
インスタンスは、以下の2つのいずれかの方法で生成することができます。

1. コミットワークを指定しない場合

インスタンス（第1～3引数あり）を生成します。

第1引数に承認する伝票番号、第2引数に承認コード、第3引数に例外メッセージ区分を設定します。

コミットワーク（はいいいえ）はブランクが内部で設定されます。

2. コミットワークを指定する場合

インスタンス（第1～4引数あり）を生成します。

第1引数に承認する伝票番号、第2引数に承認コード、第3引数に例外メッセージ区分、第4引数にコミットワークを設定します。

コミットワークに「X」を指定するとコミット処理が実行されません。

ブランクを指定するとコミット処理が実行されます。

- 購買伝票の承認処理の実行(説明2)
PurchaseReleaseController クラスの executeBapi メソッドを実行すると、コンストラクタでセットされた購買伝票の承認処理が実行されます。
承認処理が成功した場合には、購買伝票承認 API の内部で自動的に commit 処理が実行されます。
承認処理に失敗した場合には、自動的にrollback 処理が実行されます。
コンストラクタ生成時にコミットワークに「X」を設定すると、承認処理が成功しても BAPI内部で commit 処理が実行されません。
コミットワークに「X」を設定し commit 処理を実行しない場合には、bapi_po_release.properties の commit プロパティを false に設定する必要があります。
- 新承認状況を取得(説明3)
承認実行後の承認状況を取得します。
- 新承認コードを取得(説明4)
承認実行後の承認状況を取得します。
- リターン値を取得(説明5)
承認実行処理のリターン値を取得します。

伝票 承認取消 API (PurchaseResetReleaseController)

伝票を承認取消するためには、PurchaseResetReleaseController クラスの持つ各メソッドを利用する必要があります。

引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
PurchaseResetReleaseController	コンストラクタ（第 1～3 引数あり）。 第1引数に承認取消する伝票番号、第2引数に承認コード、第 3 引数に例外メッセージ出力区分を設定する。
executeBapi	コンストラクタで設定した値を、接続先のSAPに更新する。 更新に失敗した場合、更新処理はロールバックされる。
getRelStatusNew	承認後の新承認状況を取得する。
getRelIndicatorNew	承認後の新承認区分を取得する。

サンプルプログラム


```

//承認状況
var strStatus;
//承認区分
var strRelInd;
//リターン値
var strRetCode;

function init(request) {
    var objCon;
    var objResult;

    //インスタンス生成
    objCon = new PurchaseResetReleaseController ("4500000011", "AA", "X"); // 説明 1

    //購買伝票の承認取消処理の実行
    objResult = objCon.executeBapi(); // 説明 2

    //要素[ERR_CODE]に何らかの値がセットされている場合
    if (objResult["ERR_CODE"] != null) {
        //エラー情報を出力する。
        Debug.browse(objResult);
    }
    //オブジェクトをグローバル領域に保存
    strStatus = objCon.getRelStatusNew(); // 説明 3
    strRelInd = objCon.getRelIndicatorNew(); // 説明 4
}

```

- 伝票承認 API のインスタンス生成
PurchaseReleaseController クラスのインスタンスを生成します。
第1引数に承認取消する伝票番号、第2引数に承認コード、第3引数に例外メッセージ区分を設定します。
- 購買伝票の承認取消処理の実行
PurchaseReleaseController クラスの executeBapi メソッドを実行すると、コンストラクタでセットされた購買伝票の承認取消処理が実行されます。
承認取消処理が成功した場合には、購買伝票承認取消 API の内部で自動的に commit 処理が実行されます。
承認取消処理に失敗した場合には、自動的に rollback 処理が実行されます。
ただし、標準の購買伝票承認取消用BAPI(BAPI_PO_RESET_RELEASE)では、BAPI内部で commit/rollback処理が実行されます。
ConnectionManagerを使用し commit/rollback処理を制御されたい場合は、BAPI 自体をカスタマイズする必要があります。
- 新承認状況を取得
承認取消実行後の承認状況を取得します。
- 新承認コードを取得
承認取消実行後の承認状況を取得します。

品目コード割当済 照会 API (StdMaterialGetIntNumberController)

割り当てられた品目コードを取得するためには、StdMaterialGetIntNumberController クラスの持つ各メソッドを利用する必要があります。

引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
StdMaterialGetIntNumberController	コンストラクタ（第 1～3 引数あり）。 第 1 引数に品目タイプ、第 2 引数に産業コード、第 3 引数に割当数を設定する。
StdMaterialGetIntNumberController	コンストラクタ（第 1～4 引数あり）。 SAPClient の接続情報を使う場合に使用する。 第 4 引数で「SAPClient」（オブジェクト）を設定する。
getStdMaterialGetIntNumberList	品目コード割当済一覧（結果）を取得します。
getErrCode	コンストラクタでエラー時のエラーコードを取得します。
getErrDesc	コンストラクタでエラー時の実行時エラー内容を取得します。

コンストラクタ / メソッド名	説明
getErrMsg	コンストラクタでエラー時のエラーメッセージを取得します。

サンプルプログラム

```

var listResult;
var listCount;

function init(request) {
    var objCon;

    // 品目タイプ
    var materialType = "FERT";
    // 産業コード
    var industrySector = "1";
    // 必須品目コードの設定 * 割当数
    var requiredNumbers = "1";

    // インスタンス生成 // 説明 1
    objCon = new StdMaterialGetIntNumberController(materialType, industrySector, requiredNumbers);

    // コンストラクタでのエラーチェック
    if(objCon.getErrCode() != "0"){
        // エラーメッセージ出力
        Debug.browse(objCon.getErrMsg());
    }
    // コード一覧をセッションから取得
    listResult = objCon.getStdMaterialGetIntNumberList(); // 説明 2

    // 一覧の件数を取得
    listCount = listResult.length; // 説明 3
}

```

- 品目コード割当済照会 API のインスタンス生成(説明1)

StdMaterialGetIntNumberController クラスのインスタンスを生成します。

第 1 引数に「品目タイプ」、第 2 引数に「産業コード」、第 3 引数に「必須品目コード」を設定します。

インスタンスを生成すると、BAPI を呼び出して割当済の品目コードを取得する処理が実行されます。
- 品目コード割当済一覧を取得(説明2)

StdMaterialGetIntNumberController クラスのgetStdMaterialGetIntNumberList メソッドを使用すると、割り当てられた品目コード一覧を取得することが可能です。
- 割当済品目コードの件数を取得(説明3)

StdMaterialGetIntNumberController クラスのgetStdMaterialGetIntNumberList メソッドで取得した割当済品目コードデータは、length で件数を取得することが可能です。

仕入先コード存在チェック取得 API (SupplierExistenceCheckController)

仕入先コード存在チェックを取得するためには、SupplierExistenceCheckController クラスの持つ各メソッドを利用する必要があります。引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
SupplierExistenceCheckController	コンストラクタ（第1、第2引数あり）。 第1引数で「仕入先」、第2引数で「会社コード」を設定した場合、引数に紐付く情報のみ、下記のメソッドにて取得することができるようになる。
getCheckReturn	チェックリターンコードを取得する。 仕入先コードが存在する時は"1"を、存在しない時は"0"を取得します。

サンプルプログラム

```

var check_code;

function init(request) {
  var objCon;

  //インスタンス生成
  objCon = new SupplierExistenceCheckController("0000100011", ""); // 説明 1

  // コンストラクタでのエラーチェック
  if(objCon.getErrCode() != "0"){
    //エラーメッセージ出力
    Debug.browse(objCon.getErrMsg());
  }
  //オブジェクトを取得
  check_code = objCon.getCheckReturn(); // 説明 2
}

```

- 仕入先コード存在チェック 取得 API のインスタンス生成(説明1)
SupplierExistenceCheckController クラスのインスタンスを生成します。
第1引数に「仕入先」、第2引数に「会社コード」を設定します。
インスタンスを生成すると、BAPI を呼び出して仕入先コード存在チェックを取得する処理が実行されます。
- チェックリターンコードを取得(説明2)
SupplierExistenceCheckController クラスの getCheckReturn メソッドを使用すると、取得したチェックリターンコードを取得することが可能です。

仕入先マスタ詳細 取得 API (SupplierGetDetailController)

仕入先マスタ詳細を取得するためには、SupplierGetDetailController クラスの持つ各メソッドを利用する必要があります。
引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
SupplierGetDetailController	コンストラクタ（第1、第2引数あり）。 第1引数で「仕入先」、第2引数で「会社コード」を設定した場合、 引数に紐付く情報のみ、下記の3つのメソッドにて取得することができるようになる。
getGeneralDataInfo	一般データを取得する。
getCompanyDataInfo	会社データを取得する。
getBankDataList	銀行データを取得する。

サンプルプログラム

```
// グローバル変数 省略
```

```
function init(request) {
  var objCon;
  var list;
  var gene;
  var comp;
  //インスタンス生成
  objCon = new SupplierGetDetailController("0000100011", "1000"); // 説明 1

  // コンストラクタでのエラーチェック
  if(objCon.getErrCode() != "0"){
    //エラーメッセージ出力
    Debug.browse(objCon.getErrMsg());
  }

  //オブジェクトをグローバル領域に保存
  gene = objCon.getGeneralDataInfo(); // 説明 2
  comp = objCon.getCompanyDataInfo(); // 説明 3

  //オブジェクトを取得
  strVendor = gene.VENDOR;
  strName = gene.NAME;
  strName_2 = gene.NAME_2;
  strName_3 = gene.NAME_3;
  strName_4 = gene.NAME_4;
  strCity = gene.CITY;
  strDistrict = gene.DISTRICT;
  strPo_box = gene.PO_BOX;
  strPobx_pcd = gene.POBX_PCD;
  strPostl_code = gene.POSTL_CODE;
  strRegion = gene.REGION;
  strStreet = gene.STREET;
  strCountry = gene.COUNTRY;
  strCountryiso = gene.COUNTRYISO;
  strPobx_cty = gene.POBX_CTY;
  strLangu = gene.LANGU;
  strLangu_iso = gene.LANGU_ISO;
  strTelephone = gene.TELEPHONE;
  strFormofaddr = gene.FORMOFADDR;
  strTelephone2 = gene.TELEPHONE2;
  strComp_code = comp.COMP_CODE;
  strClerk = comp.CLERK;
  strHd_office = comp.HD_OFFICE;
  strAlt_payee = comp.ALT_PAYEE;
  strCuvd_clear = comp.CUVD_CLEAR;
  strPmnttrms = comp.PMNTTRMS;
  strAct_at_ven = comp.ACT_AT_VEN;
  strVend_user = comp.VEND_USER;
  strInternet = comp.INTERNET;
  strFax = comp.FAX;
  strPayment_methods = comp.PAYMENT_METHODS;
  strTel = comp.TEL;

  listResult = objCon.getBankDataList(); // 説明 4
  listCount = listResult.length; // 説明 5
}
```

- 仕入先マスタ詳細照会 取得 API のインスタンス生成(説明1)
SupplierGetDetailController クラスのインスタンスを生成します。
第1引数に「仕入先」、第2引数に「会社コード」を設定します。
インスタンスを生成すると、BAPI を呼び出して得意先詳細を取得する処理が実行されます。
- 一般データを取得(説明2)
SupplierGetDetailController クラスのgetGeneralDataInfoメソッドを使用すると、取得した一般データを取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

- 会社データを取得(説明3)
SupplierGetDetailController クラスの getCompanyDataInfo メソッドを使用すると、取得した会社データを取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 会社データを取得(説明4)
SupplierGetDetailController クラスの getBankDataList メソッドを使用すると、取得した銀行データを取得することが可能です。
- 銀行データの件数を取得(説明5)
SupplierGetDetailController クラスの getBankDataList メソッドで取得した銀行データは、length で件数を取得することが可能です。

販売管理業務向け API

販売管理業務向けAPIの各APIで使用するBAPI情報を定義したプロパティファイルが以下のフォルダに配備されています。
“WEB-INF/classes/jp/co/intra_mart/foundation/sap/module”

ファイル名を変更すると、APIを利用することができませんので、注意してください。
また、認証情報ファイル「sap_auth_info.properties」の接続先情報でservertypeを使用している場合は、各APIのプロパティファイルにservertypeを指定する必要がありますので、servertypeの設定を行ってください。
財務会計業務向け APIの機能は以下となります。

請求伝票登録 API (BillingDocCreateMultipleController)

請求伝票を登録するためには、BillingDocCreateMultipleController クラスの持つ各メソッドを利用する必要があります。
引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
BillingDocCreateMultipleController	コンストラクタ (引数無し)
BillingDocCreateMultipleController	コンストラクタ (第1引数あり)。 ConnectionManager を使う場合に使用する。第1引数で「ConnectionManager」を設定する。
BillingDocCreateMultipleController	コンストラクタ (第1引数あり)。 SAPClient の接続情報を使う場合に使用する。第1引数で「SAPClient」(オブジェクト)を設定する。
setTestrun	import 項目 : シミュレーション実行 ('X')、更新実行 ('')を設定します。
setPosting	import 項目 : 転記カテゴリ: 直接転記を設定します。
setStrParam	構造型の各項目に値を設定する。 第1引数に structure 名、第2引数に項目名、第3引数に値を設定する。
setTblParam	テーブル型の各項目に値を設定する。 第1引数にテーブル名、第2引数に項目名第3引数に値、第4引数にレコード番号を設定する。
executeBapi	BapiController により請求伝票を登録します。
getBapiReturnErrMessage	BAPI実行結果である BapiReturn の Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn オブジェクトのメッセージ情報を全て返します。 全て正常の場合は null が返却されます。
getBapiReturnMessage	BAPI実行結果である BapiReturn の Type 情報が正常(デフォルト値: "S" / "")と判断された Type 情報のメッセージ情報を全て返します。 メッセージが存在しない場合は null が返却されます。

コンストラクタ / メソッド名	説明
getTestrun	import 項目：シミュレーション実行 ('X')、更新実行 ('')を取得します。
getPosting	import 項目：転記カテゴリ: 直接転記を取得します。
getBillingDataInList	import 項目：設定した処理予定明細データのテーブル情報を取得します。
getCCardDataInList	import 項目：設定した処理予定支払方法のテーブル情報を取得します。
getConditionDataInList	import項目：設定した処理予定条件のテーブル情報を取得します。
getCreatorDataIn	import 項目：設定した取り込まれる日付の順序情報を取得します。
getErrorsList	import 項目：設定した前明細の不正処理に関する情報情報を取得します。
getSuccessList	export 項目：設定した正常に処理された明細のテーブル情報を取得します。
getTextDataInList	import項目：設定した請求インタフェースの通信構造テキスト情報を取得します。

サンプルプログラム

```

//■正常に処理された明細のテーブル
var successList;
var successCnt;
//■前明細の不正処理に関する情報
var errorsList;
var errorsCnt;
var ret;

function init(request) {

    //-----
    //変数宣言
    //-----
    var objCon;
    var objResult;

    //-----
    //インスタンス生成 // 説明 1
    //-----
    objCon = new BillingDocCreateMultipleController();

    //-----
    // テスト実行 // 説明 2
    //-----
    objCon.setTestrun("");

    //-----
    // テーブル項目セット // 説明 4
    //-----
    //処理予定明細データのテーブル
    objCon.setTblParam("BillingDataIn", "BILL_DATE", "20070115", 1); //請求索引および印刷に対する請求日
    objCon.setTblParam("BillingDataIn", "REF_DOC", "0080000033", 1); //参照伝票番号
    objCon.setTblParam("BillingDataIn", "REF_DOC_CA", "J", 1); //先行販売管理伝票の伝票カテゴリ

    //-----
    // executeBapi実行 // 説明 5
    //-----
    objResult = objCon.executeBapi();

    //-----
    // エラー判定
    //-----
    if(objResult["ERR_CODE"] != null) {
        //インスタンス生成時に例外が発生した場合
        //エラーメッセージを出力
        Debug.browse(objResult);
    }

    //API実行結果を取得します。 // 説明 6
    //正常に処理された明細のテーブルの内容と件数を取得
    successList = objCon.getSuccessList();
    successCnt = successList.length;

    //前明細の不正処理に関する情報
    errorsList = objCon.getErrorsList();
    errorsCnt = errorsList.length;

    //API実行結果を取得します。
    ret = objCon.getBapiReturnMessage();
}

```

- 請求伝票登録 API のインスタンス生成(説明1)
BillingDocCreateMultipleController クラスのインスタンスを生成します。
引数は不要です。
インスタンスを生成すると、請求伝票登録 BAPI (BAPI_BILLINGDOC_CREATEMULTIPLE) の import 型の import、structure、table パラメータに値を設定することができます。
- import の各項目に値を設定(説明2)

import パラメータの項目に値を設定する場合は、set メソッドを使用します。
メソッドについては各 API のAPI リストを参照ください。

- structure の各項目に値を設定(説明3)
structure パラメータの項目に値を設定する場合は、setStrParam メソッドを使用します。
第1引数の structure 名称は固定値になっています。
後述のとおり名称を設定して下さい。
第2引数の項目名称はSAP の各属性の項目名称を設定してください。
structure 名称、項目名称は英字の大文字、小文字は厳密に区別されます。
第3引数に値を設定します。

SAP structure	セットする固有名称
CREATORDATAIN (取り込まれる日付の順序)	CreatorDataIn

- table の各項目に値を設定(説明4)
table パラメータの項目に値を設定する場合は、setTblParam メソッドを使用します。
第1引数のtable名称は固定値になっています。
後述のとおり名称を設定して下さい。
第2引数の項目名称は SAP の各属性の項目名称を設定してください。
table 名称、項目名称は英字の大文字、小文字は厳密に区別されます。
第3引数に値、第4引数にレコード番号を設定します。

SAP table	セットする固有名称
BILLINGDATAIN (処理予定明細データのテーブル)	BillingDataIn
CCARDDATAIN (処理予定支払方法のテーブル)	CCardDataIn
CONDITIONDATAIN (処理予定条件のテーブル)	ConditionDataIn
ERRORS (前明細の不正処理に関する情報)	Errors
SUCCESS (正常に処理された明細のテーブル)	Success
TEXTDATAIN (請求インタフェースの通信構造テキスト)	TextDataIn

- 請求伝票の登録(説明5)
BillingDocCreateMultipleController クラスの executeBapi メソッドを実行すると、各項目にセットされた値がSAP 上に請求伝票データとして登録されます。
登録が成功した場合には、請求伝票登録 API の内部で自動的に commit 処理が実行されます。
登録に失敗した場合には、自動的に rollback 処理が実行されます。
成功・失敗の基準は、請求伝票登録用 BAPI (BAPI_BILLINGDOC_CREATEMULTIPLE) の実行後にRETURN パラメータの TYPE 値を取得して、S (正常終了)、空文字以外が1つでも存在するかどうかです。
1つでも存在する場合は、rollback が自動的に実行されます。
- サンプルプログラムの中で記載されたデータ取得メソッドの詳細説明については、以下を参照してください。(説明6)

 - Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn を取得
BillingDocCreateMultipleController クラスの getBapiReturnErrorMessage メソッドを使用すると、取得したBapiReturn の Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn オブジェクトのメッセージ情報を取得することが可能です。
全て正常の場合は null を取得します。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - Type 情報が正常(デフォルト値: "S" / "")の BapiReturn を取得
BillingDocCreateMultipleController クラスの getBapiReturnMessage メソッドを使用すると、取得したBapiReturn の Type 情報が正常(デフォルト値: "S" / "")と判断された Type 情報のメッセージ情報を取得することが可能です。
メッセージが存在しない場合は null を取得します。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - シミュレーション実行 ('X')、更新実行 ('')情報を取得
BillingDocCreateMultipleController クラスのgetTestrunメソッドを使用すると、設定したシミュレーション実行 ('X')、更新実行 ('')情報を取得することが可能です。
 - 転記カテゴリ: 直接転記情報を取得
BillingDocCreateMultipleController クラスの getPosting メソッドを使用すると、設定した転記カテゴリ: 直接転記情報を取

得することが可能です。

- 処理予定明細データのテーブル情報を取得
BillingDocCreateMultipleController クラスの getBillingDataInList メソッドを使用すると、設定した処理予定明細データのテーブル情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 処理予定支払方法のテーブル情報を取得
BillingDocCreateMultipleController クラスの getCCardDataInList メソッドを使用すると、設定した処理予定支払方法のテーブル情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 処理予定条件のテーブル情報を取得
BillingDocCreateMultipleController クラスの getConditionDataInList メソッドを使用すると、設定した処理予定条件のテーブル情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 取り込まれる日付の順序情報を取得
BillingDocCreateMultipleController クラスの getCreatorDataInList メソッドを使用すると、設定した取り込まれる日付の順序情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 前明細の不正処理に関する情報情報を取得
BillingDocCreateMultipleController クラスの getErrorsList メソッドを使用すると、取得した前明細の不正処理に関する情報情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 正常に処理された明細のテーブル情報を取得
BillingDocCreateMultipleController クラスの getSuccessList メソッドを使用すると、取得した正常に処理された明細のテーブル情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 請求インタフェースの通信構造テキスト情報を取得
BillingDocCreateMultipleController クラスの getTextDataInList メソッドを使用すると、設定した請求インタフェースの通信構造テキスト情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。



コラム

サンプルプログラム実行結果では、SAP テスト環境のデータが取得されています。SAP 標準では該当データが存在しません。



注意

請求伝票登録 API に関して、import、structure、table パラメータを持つ各コンポーネントの必須項目や値の相関チェック等を確認する必要がある場合は、SAP 上の汎用モジュールビルダ (t-cd : SE37) 等を使用して、「BAPI_BILLINGDOC_CREATEMULTIPLE」の BAPI 仕様を確認してください。

請求伝票詳細 照会 API (BillingDocGetDetailController)

請求伝票詳細を照会するためには、BillingDocGetDetailController クラスの持つ各メソッドを利用する必要があります。
引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
BillingDocGetDetailController	コンストラクタ (引数無し)
BillingDocGetDetailController	コンストラクタ (第 1 引数あり)。 ConnectionManager を使う場合に使用する。第 1 引数で「ConnectionManager」を設定する。
BillingDocGetDetailController	コンストラクタ (第 1 引数あり)。 SAPClient の接続情報を使う場合に使用する。第 1 引数で「SAPClient」(オブジェクト)を設定する。
getBapiReturnErrorMessage	BAPI実行結果である BapiReturn の Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn オブジェクトのメッセージ情報を全て返します。 全て正常の場合は null が返却されます。

コンストラクタ / メソッド名	説明
getBapiReturnMessage	BAPI実行結果である BapiReturn の Type 情報が正常(デフォルト値: "S" / "")と判断された Type 情報のメッセージ情報を全て返します。 メッセージが存在しない場合は null が返却されます
getBillingDocumentDetail	請求伝票に関する詳細情報情報を取得します。
getErrCode	コンストラクタでエラー時のエラーコードを取得します。
getErrDesc	コンストラクタでエラー時の実行時エラー内容を取得します。
getErrMsg	コンストラクタでエラー時のエラーメッセージを取得します。

サンプルプログラム

```

var strBillingdoc;
var strBill_type;
var strBillcateg;
var strSd_doc_cat;
var strCurrency;
var strCurrency_iso;
var strAcctstatus;
var strNet_value;
var strTax_value;
var strCancelled;

function init(request) {
    //-----
    //変数宣言
    //-----
    var billingdocument = "0090000000"; // 請求伝票数

    //-----
    //インスタンス生成 // 説明 1
    //-----
    var objCon = new BillingDocGetDetailController(billingdocument);

    //-----
    // エラー判定
    //-----
    if (objCon.getErrCode() != "0"){
        // エラーが発生した場合

        var msg = objCon.getErrMsg();
        var desc = objCon.getErrDesc();

        writeError(msg,desc);
    }
    else {
        // エラーが発生していない場合

        //オブジェクトを取得 // 説明 2
        var bill = objCon.getBillingDocumentDetail();

        //オブジェクトを取得
        //請求伝票
        strBillingdoc = bill.BILLINGDOC;
        strBill_type = bill.BILL_TYPE;
        strBillcateg = bill.BILLCATEG;
        strSd_doc_cat = bill.SD_DOC_CAT;
        strCurrency = bill.CURRENCY;
        strCurrency_iso = bill.CURRENCY_ISO;
        strAcctstatus = bill.ACCTSTATUS;
        strNet_value = bill.NET_VALUE;
        strTax_value = bill.TAX_VALUE;
        strCancelled = bill.CANCELLED;
    }
}

```

- 請求伝票詳細照会 API のインスタンス生成(説明1)
BillingDocGetDetailController クラスのインスタンスを生成します。
第1引数に照会する請求伝票番号を設定します。
インスタンスを生成すると、請求伝票詳細 BAPI (BAPI_BILLINGDOC_GETDETAIL) を呼び出して請求伝票詳細情報を取得する処理が実行されます。*import 型の import、structure、table パラメータが存在しないため個別に設定することはできません。
API 内部で BAPI が実行されます。

- サンプルプログラムの中で記載されたデータ取得メソッドの詳細説明については、以下を参照してください。(説明2)
 - Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn を取得
BillingDocGetDetailController クラスの getBapiReturnErrMessage メソッドを使用すると、取得したBapiReturn の Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn オブジェクトのメッセージ情報を取得することが可能です。
全て正常の場合は null を取得します。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - Type 情報が正常(デフォルト値: "S" / "")の BapiReturn を取得
BillingDocGetDetailController クラスの getBapiReturnMessage メソッドを使用すると、取得したBapiReturn の Type 情報が正常(デフォルト値: "S" / "")と判断された Type 情報のメッセージ情報を取得することが可能です。
メッセージが存在しない場合は null を取得します。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 請求伝票に関する詳細情報情報を取得
BillingDocGetDetailController クラスの getBillingDocumentDetail メソッドを使用すると、取得した請求伝票に関する詳細情報情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 請求伝票に関する詳細情報情報を取得
BillingDocGetDetailController クラスの getBillingDocumentDetail メソッドを使用すると、取得した請求伝票に関する詳細情報情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - エラー時のエラーコード情報を取得
BillingDocGetDetailController クラスの getErrCode メソッドを使用すると、取得したエラー時のエラーコード情報を取得することが可能です。
 - エラー時の実行時エラー内容情報を取得
BillingDocGetDetailController クラスの getErrDesc メソッドを使用すると、取得したエラー時の実行時エラー内容情報を取得することが可能です。
 - エラー時のエラーメッセージ情報を取得
BillingDocGetDetailController クラスの getErrMsg メソッドを使用すると、取得したエラー時のエラーメッセージ情報を取得することが可能です。



コラム

サンプルプログラム実行結果では、SAP テスト環境のデータが取得されています。SAP 標準では該当データが存在しません。

請求伝票一覧 照会 API (BillingDocGetListController)

請求伝票一覧を照会するためには、BillingDocGetListController クラスの持つ各メソッドを利用する必要があります。
引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
BillingDocGetListController	コンストラクタ (引数無し)
BillingDocGetListController	コンストラクタ (第 1 引数あり)。 ConnectionManager を使う場合に使用する。第 1 引数で「ConnectionManager」を設定する。
BillingDocGetListController	コンストラクタ (第 1 引数あり)。 SAPClient の接続情報を使う場合に使用する。第 1 引数で「SAPClient」(オブジェクト)を設定する。
setStrParam	構造型の各項目に値を設定する。 第 1 引数に structure 名、第 2 引数に項目名、第 3 引数に値を設定する。

コンストラクタ / メソッド名	説明
setTblParam	テーブル型の各項目に値を設定する。 第1引数にテーブル名、第2引数に項目名第3引数に値、第4引数にレコード番号を設定する。
executeBapi	BapiController により後続請求伝票一覧を取得します。
getBapiReturnErrMessage	BAPI実行結果である BapiReturn の Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn オブジェクトのメッセージ情報を全て返します。 全て正常の場合は null が返却されます。
getBapiReturnMessage	BAPI実行結果である BapiReturn の Type 情報が正常(デフォルト値: "S" / "")と判断された Type 情報のメッセージ情報を全て返します。 メッセージが存在しない場合は null が返却されます。
getBillingDocumentDetailListList	後続請求伝票のテーブル情報を取得します。
getSuccessList	後続請求伝票の明細テーブル情報を取得します。

サンプルプログラム

```

var detailList;
var detaillistCount;
var successList;
var successlistCount;

function init(request) {
    //-----
    //変数宣言
    //-----
    var sign = "I";           // 範囲テーブル
    var option = "GT";       // テーブル範囲
    var refdoclow = "0900000005"; // 前伝票の開始伝票番号
    var refdochigh = "";     // 前伝票の終了伝票番号

    var objCon;
    var objResult;

    //-----
    //インスタンス生成 // 説明 1
    //-----
    objCon = new BillingDocGetListController(sign, option, refdoclow, refdochigh);

    //-----
    // executeBapi実行 // 説明 4
    //-----
    objResult = objCon.executeBapi();

    //-----
    // エラー判定
    //-----
    if(objResult["ERR_CODE"] != null) {
        //インスタンス生成時に例外が発生した場合
        //エラーメッセージを出力
        Debug.browse(objResult);
    }

    //請求伝票一覧を取得 // 説明 5
    detailList = objCon.getBillingDocumentDetailListList();
    detaillistCount = detailList.length;

    //請求伝票一覧を取得
    successList = objCon.getSuccessList();
    successlistCount = successList.length;
}

```

- 請求伝票一覧照会 API のインスタンス生成(説明1)

BillingDocGetListController クラスのインスタンスを生成します。

引数は不要です。

インスタンスを生成すると、請求伝票一覧 BAPI (BAPI_BILLINGDOC_GETLIST) の import 型の structure、table パラメータに値を設定することができます。

- structure の各項目に値を設定(説明2)

structure パラメータの項目に値を設定する場合は、setStrParam メソッドを使用します。

第1引数のstructure 名称は固定値になっています。

後述のとおり名称を設定して下さい。

第2引数の項目名称はSAPの各属性の項目名称を設定してください。

structure 名称、項目名称は英字の大文字、小文字は厳密に区別されます。

第3引数に値を設定します。

SAP structure	セットする固有名称
REFDOCRANGE (前伝票の範囲)	RefDocRange

- table の各項目に値を設定(説明3)

table パラメータの項目に値を設定する場合は、setTblParam メソッドを使用します。

第1引数のtable名称は固定値になっています。

後述のとおり名称を設定して下さい。

第2引数の項目名称は SAP の各属性の項目名称を設定してください。

table 名称、項目名称は英字の大文字、小文字は厳密に区別されます。

第3引数に値、第4引数にレコード番号を設定します。

SAP table	セットする固有名称
BILLINGDOCUMENTDETAIL (後続請求伝票のテーブル)	BillingDocumentDetail
SUCCESS (後続請求伝票の明細テーブル)	Success

- 請求伝票一覧照会(説明4)

BillingDocGetListControllerクラスのexecuteBapiメソッドを実行すると、各項目にセットされた値でSAPより請求伝票一覧データとして照会されます。

請求伝票一覧 BAPI (BAPI_BILLINGDOC_GETLIST) の実行後に RETURN パラメータの内容も取得します。
- サンプルプログラムの中で記載されたデータ取得メソッドの詳細説明については、以下を参照してください。(説明5)

 - Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn を取得

BillingDocGetListController クラスの getBapiReturnErrorMessage メソッドを使用すると、取得したBapiReturn の Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn オブジェクトのメッセージ情報を取得することが可能です。

全て正常の場合は null を取得します。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - Type 情報が正常(デフォルト値: "S" / "")の BapiReturn を取得

BillingDocGetListControllerクラスのgetBapiReturnMessageメソッドを使用すると、取得したBapiReturn の Type 情報が正常(デフォルト値: "S" / "")と判断された Type 情報のメッセージ情報を取得することが可能です。

メッセージが存在しない場合は null を取得します。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 後続請求伝票のテーブル情報を取得

BillingDocGetListController クラスの getBillingDocumentDetailListList メソッドを使用すると、取得した後続請求伝票のテーブル情報を取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 後続請求伝票の明細テーブル情報を取得

BillingDocGetListControllerクラスのgetSuccessListメソッドを使用すると、取得した後続請求伝票の明細テーブル情報を取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

引合伝票を登録するためには、InquiryCreateFromData2Controller クラスの持つ各メソッドを利用する必要があります。

引数、戻り値についての詳細な情報は、API リストを参照してください。

また、登録可能な項目は、引合伝票として必要最低限必要と判断される項目のみとなっています。

各メソッドの説明

コンストラクタ / メソッド名	説明
InquiryCreateFromData2Controller	コンストラクタ (引数無し)
InquiryCreateFromData2Controller	コンストラクタ (第 1 引数あり)。 ConnectionManager を使う場合に使用する。 第 1 引数で「ConnectionManager」を設定する。
InquiryCreateFromData2Controller	コンストラクタ (第 1 引数あり)。 SAPClient の接続情報を使う場合に使用する。 第 1 引数で「SAPClient」 (オブジェクト) を設定する。
setBehaveWhenError	import 項目 : エラー処理を設定します。
setBinaryRelationshiptype	import 項目 : バイナリ関連タイプ (専用)を設定します。
setConvert	import 項目 : 変換: 取引先機能 + 受注伝票タイプを設定します。
setIntNumberAssignment	import 項目 : 明細番号内部採番を設定します。
setSalesdocumentin	import 項目 : 販売管理伝票番号を設定します。
setTestrun	import 項目 : テスト実行を設定します。
setStrParam	構造型の各項目に値を設定する。 第 1 引数に structure 名、第 2 引数に項目名、第 3 引数に値を設定する。
setTblParam	テーブル型の各項目に値を設定する。 第1引数にテーブル名、第2引数に項目名第3引数に値、第4引数にレコード番号を設定する。
executeBapi	BapiController により引合伝票を登録します。
getBapiReturnErrMsg	BAPI実行結果である BapiReturn の Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn オブジェクトのメッセージ情報を全て返します。 全て正常の場合は null が返却されます。
getBapiReturnMessage	BAPI実行結果である BapiReturn の Type 情報が正常(デフォルト値: "S" / "")と判断された Type 情報のメッセージ情報を全て返します。 メッセージが存在しない場合は null が返却されます。
getBehaveWhenError	import 項目 : エラー処理を取得します。
getBinaryRelationshiptype	import 項目 : バイナリ関連タイプ (専用)を取得します。
getConvert	import 項目 : 変換: 取引先機能 + 受注伝票タイプを取得します。
getIntNumberAssignment	import 項目 : 明細番号内部採番を取得します。
getSalesdocumentin	import 項目 : 販売管理伝票番号を取得します。
getTestrun	import 項目 : テスト実行を取得します。
getSalesdocument	export 項目 : 販売伝票番号を取得します。
getExtensionInList	import 項目 : 設定した VBAK、VBAP、VBEP のカスタム拡張情報を取得します。
getInquiryCfgsBlobList	import 項目 : 設定した選定: BLOB 内部データ (SCE)情報を取得します。
getInquiryCfgsInstList	import 項目 : 設定した選定: インスタンス情報を取得します。
getInquiryCfgsPartOfList	import 項目 : 設定した選定: 部分設定情報を取得します。
getInquiryCfgsRefList	import 項目 : 設定した選定: 参照データ情報を取得します。

コンストラクタ / メソッド名	説明
getInquiryCfgsRefInstList	import 項目：設定した選定: 参照明細/インスタンス情報を取得します。
getInquiryCfgsValueList	import 項目：設定した選定: 特性値情報を取得します。
getInquiryCfgsVkList	import 項目：設定した選定: バリエーション条件キー情報を取得します。
getInquiryConditionsInList	import 項目：設定した条件情報を取得します。
getInquiryConditionsInXList	import 項目：設定した受注における条件更新の通信項目情報を取得します。
getInquiryHeaderIn	import 項目：設定した引合伝票ヘッダデータ情報を取得します。
getInquiryHeaderInX	import 項目：設定したヘッダデータチェックボックス情報を取得します。
getInquiryItemsInList	import 項目：設定した明細データ情報を取得します。
getInquiryItemsInXList	import 項目：設定した明細データチェックボックス情報を取得します。
getInquiryKeysList	import 項目：設定した参照キーの出力テーブル情報を取得します。
getInquiryPartnersList	import 項目：設定した伝票取引先情報を取得します。
getInquirySchedulesInList	import 項目：設定した納入日程行情報を取得します。
getInquirySchedulesInXList	import 項目：設定した納入日程行チェックボックス情報を取得します。
getInquiryTextList	import 項目：設定したテキスト情報を取得します。
getLogicSwitch	import 項目：設定した内部制御パラメータ情報を取得します。
getPartnerAddressesList	import 項目：設定したアドレス (組織/会社) 用 BAPI 参照構造情報を取得します。
getSender	import 項目：設定した論理システム - 送信者情報を取得します。

サンプルプログラム

```

var salesDocument;
var partneraddressesList;
var partneraddresseslistCount;
var extensioninList;
var extensioninlistCount;
var cfgsblobList;
var cfgsbloblistCount;
var textList;
var textlistCount;
var schedulesinxList;
var schedulesinxlistCount;
var schedulesinList;
var schedulesinlistCount;
var partnersList;
var partnerslistCount;
var keysList;
var keyslistCount;
var itemsinxList;
var itemsinxlistCount;
var itemsinList;
var itemsinlistCount;
var cfgsinstList;
var cfgsinstlistCount;
var cfgspartofList;
var cfgspartoflistCount;
var conditionsinxList;
var conditionsinxlistCount;
var conditionsinList;

```

```

var conditionsinlistCount;
var cfigsvkList;
var cfigsvklistCount;
var cfigsvalueList;
var cfigsvaluelistCount;
var cfigsrefinstList;
var cfigsrefinstlistCount;
var cfigsrefList;
var cfigsreflistCount;
var ret;

function init(request) {
//-----
//変数宣言
//-----
var objResult; // executeBapi実行結果

//-----
//インスタンス生成 // 説明1
//-----
var objCon = new InquiryCreateFromData2Controller();

//-----
// テスト実行 // 説明2
//-----
objCon.setTestrun("");

//-----
// 構造体項目セット // 説明3
//-----
// 通信項目: 販売管理伝票ヘッダ
objCon.setStrParam("HeaderIn", "DOC_TYPE", "AF"); // 販売伝票タイプ
objCon.setStrParam("HeaderIn", "QT_VALID_F", "20070301"); // 見積/引合有効開始日
objCon.setStrParam("HeaderIn", "QT_VALID_T", "20070331"); // 見積の有効終了日
objCon.setStrParam("HeaderIn", "PURCH_NO_C", "API_引合伝票登録1"); // 出荷先発注番号

//-----
// テーブル項目セット // 説明4
//-----
// 伝票取引先
objCon.setTblParam("Partners", "PARTN_ROLE", "AG", 1); // 取引先機能
objCon.setTblParam("Partners", "PARTN_NUMB", "0000000012", 1); // 得意先コード1
objCon.setTblParam("Partners", "PARTN_ROLE", "WE", 2); // 取引先機能
objCon.setTblParam("Partners", "PARTN_NUMB", "0000000012", 2); // 得意先コード1
objCon.setTblParam("Partners", "PARTN_ROLE", "RG", 3); // 取引先機能
objCon.setTblParam("Partners", "PARTN_NUMB", "0000000012", 3); // 得意先コード1

//ORDER_ITEMS_IN[明細データ]
objCon.setTblParam("ItemsIn", "MATERIAL", "000000000000000011", 1); // 品目コード

//ORDER_SCHEDULES_IN[納入日程データ]
objCon.setTblParam("SchedulesIn", "REQ_QTY", "3", 1); //販売単位による受注数量

//-----
// executeBapi実行 // 説明5
//-----
objResult = objCon.executeBapi();

//-----
// エラー判定
//-----
//要素[ERR_CODE]に何らかの値がセットされている場合
if (objResult["ERR_CODE"] != null) {
//エラーメッセージ出力
Debug.browse(objResult);
}

// 各データをセット // 説明6
// 販売伝票番号をセット
salesDocument = objCon.getSalesdocument();

// アドレス (組織/会社) 用 BAPI 参照構造

```

```
partneraddressesList = objCon.getPartnerAddressesList();
partneraddresseslistCount = partneraddressesList.length;
```

```
// VBAK、VBAP、VBEP のカスタマ拡張
extensioninList = objCon.getExtensionInList();
extensioninlistCount = extensioninList.length;
```

```
// 選定: BLOB 内部データ (SCE)
cfgsblobList = objCon.getInquiryCfgsBlobList();
cfgsbloblistCount = cfgsblobList.length;
```

```
// テキスト
textList = objCon.getInquiryTextList();
textlistCount = textList.length;
```

```
// 納入日程行チェックボックス
schedulesinxList = objCon.getInquirySchedulesInXList();
schedulesinxlistCount = schedulesinxList.length;
```

```
// 納入日程行
schedulesinList = objCon.getInquirySchedulesInList();
schedulesinlistCount = schedulesinList.length;
```

```
// 伝票取引先
partnersList = objCon.getInquiryPartnersList();
partnerslistCount = partnersList.length;
```

```
// 参照キーの出カテーブル
keysList = objCon.getInquiryKeysList();
keyslistCount = keysList.length;
```

```
// 明細データチェックボックス
itemsinxList = objCon.getInquiryItemsInXList();
itemsinxlistCount = itemsinxList.length;
```

```
// 明細データ
itemsinList = objCon.getInquiryItemsInList();
itemsinlistCount = itemsinList.length;
```

```
// 選定: インスタンス
cfgsinstList = objCon.getInquiryCfgsInstList();
cfgsinstlistCount = cfgsinstList.length;
```

```
// 選定: 部分設定
cfgspartofList = objCon.getInquiryCfgsPartOfList();
cfgspartoflistCount = cfgspartofList.length;
```

```
// 受注における条件更新の通信項目
conditionsinxList = objCon.getInquiryConditionsInXList();
conditionsinxlistCount = conditionsinxList.length;
```

```
// 条件
conditionsinList = objCon.getInquiryConditionsInList();
conditionsinlistCount = conditionsinList.length;
```

```
// 選定: バリエーション条件キー
cfgsvkList = objCon.getInquiryCfgsVkList();
cfgsvklistCount = cfgsvkList.length;
```

```
// 選定: 特性値
cfgsvalueList = objCon.getInquiryCfgsValueList();
cfgsvaluelistCount = cfgsvalueList.length;
```

```
// 選定: 参照明細/インスタンス
cfsrefinstList = objCon.getInquiryCfgrInstList();
cfsrefinstlistCount = cfsrefinstList.length;
```

```
// 選定: 参照データ
cfsrefList = objCon.getInquiryCfgrRefList();
cfsreflistCount = cfsrefList.length;
```

```
//API実行結果を取得します。
ret = objCon.getBapiReturnMessage();
```


}

- 引合伝票登録 API のインスタンス生成(説明1)
 InquiryCreateFromData2Controller クラスのインスタンスを生成します。
 引数は不要です。
 インスタンスを生成すると、引合伝票登録 BAPI (BAPI_INQUIRY_CREATEFROMDATA2) の import 型の import、structure、table パラメータに値を設定することができます。
- import の各項目に値を設定(説明2)
 import パラメータの項目に値を設定する場合は、set メソッドを使用します。
 メソッドについては各API の API リストを参照ください。
- structure の各項目に値を設定(説明3)
 structure パラメータの項目に値を設定する場合は、setStrParam メソッドを使用します。
 第1引数のstructure 名称は固定値になっています。
 後述のとおり名称を設定して下さい。
 第2引数の項目名称はSAP の各属性の項目名称を設定してください。
 structure 名称、項目名称は英字の大文字、小文字は厳密に区別されます。
 第3引数に値を設定します。

SAP structure	セットする固有名称
INQUIRY_HEADER_IN 引合伝票ヘッダデータ	HeaderIn
INQUIRY_HEADER_INX ヘッダデータチェックボックス	HeaderInX
LOGIC_SWITCH 内部制御パラメータ	LogicSwitch
SENDER 論理システム - 送信者	Sender

- table の各項目に値を設定(説明4)
 table パラメータの項目に値を設定する場合は、setTblParam メソッドを使用します。
 第1引数のtable名称は固定値になっています。
 後述のとおり名称を設定して下さい。
 第2引数の項目名称は SAP の各属性の項目名称を設定してください。
 table 名称、項目名称は英字の大文字、小文字は厳密に区別されます。
 第3引数に値、第4引数にレコード番号を設定します。

SAP table	セットする固有名称
EXTENSIONIN VBAK、VBAP、VBEP のカスタマ拡張	ExtensionIn
INQUIRY_CFGS_BLOB 選定: BLOB 内部データ (SCE)	CfgsBlob
INQUIRY_CFGS_INST 選定: インスタンス	CfgsInst
INQUIRY_CFGS_PART_OF 選定: 部分設定	CfgsPartOf
INQUIRY_CFGS_REF 選定: 参照データ	CfgsRef
INQUIRY_CFGS_REFINST 選定: 参照明細/インスタンス	CfgsRefInst
INQUIRY_CFGS_VALUE 選定: 特性値	CfgsValue
INQUIRY_CFGS_VK 選定: バリエーション条件キー	CfgsVk
INQUIRY_CONDITIONS_IN 条件	ConditionsIn
INQUIRY_CONDITIONS_INX 受注における条件更新の通信項目	ConditionsInX
INQUIRY_ITEMS_IN 明細データ	ItemsIn
INQUIRY_ITEMS_INX 明細データチェックボックス	ItemsInX
INQUIRY_KEYS 参照キーの出力テーブル	Keys
INQUIRY_PARTNERS 伝票取引先	Partners

SAP table	セットする固有名称
INQUIRY_SCHEDULES_IN 納入日程行	SchedulesIn
INQUIRY_SCHEDULES_INX 納入日程行チェックボックス	SchedulesInX
INQUIRY_TEXT テキスト	Text
PARTNERADDRESSES アドレス (組織/会社) 用 BAPI 参照構造	PartnerAddresses

- 引合伝票の登録(説明5)

InquiryCreateFromData2Controller クラスの executeBapi メソッドを実行すると、各項目にセットされた値がSAP上に、引合伝票データとして登録されます。

登録が成功した場合には、引合伝票登録 API の内部で自動的に commit 処理が実行されます。

登録に失敗した場合には、自動的に rollback 処理が実行されます。

成功・失敗の基準は、引合伝票登録用 BAPI (BAPI_INQUIRY_CREATEFROMDATA2) の実行後にRETURNパラメータのTYPE値を取得して、S (正常終了)、空文字以外が1つでも存在するかどうかです。

1つでも存在する場合は、rollback が自動的に実行されます。
- サンプルプログラムの中で記載されたデータ取得メソッドの詳細説明については、以下を参照してください。(説明6)
 - Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn を取得

InquiryCreateFromData2Controller クラスの getBapiReturnErrorMessage メソッドを使用すると、取得したBapiReturnのType情報が正常(デフォルト値: "S" / "")以外の BapiReturn オブジェクトのメッセージ情報を取得することが可能です。全て正常の場合は null を取得します。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - Type 情報が正常(デフォルト値: "S" / "")の BapiReturn を取得

InquiryCreateFromData2Controller クラスの getBapiReturnMessage メソッドを使用すると、取得したBapiReturnのType情報が正常(デフォルト値: "S" / "")と判断された Type 情報のメッセージ情報を取得することが可能です。メッセージが存在しない場合は null を取得します。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - エラー処理情報を取得

InquiryCreateFromData2Controller クラスの getBehaveWhenError メソッドを使用すると、設定したエラー処理情報を取得することが可能です。
 - バイナリ関連タイプ (専用)情報を取得

InquiryCreateFromData2Controller クラスの getBinaryRelationshiptype メソッドを使用すると、設定したバイナリ関連タイプ (専用)情報を取得することが可能です。
 - 変換: 取引先機能 + 受注伝票タイプ情報を取得

InquiryCreateFromData2ControllerクラスのgetConvertメソッドを使用すると、設定した変換(取引先機能 + 受注伝票タイプ情報)を取得することが可能です。
 - 明細番号内部採番情報を取得

InquiryCreateFromData2Controller クラスの getIntNumberAssignment メソッドを使用すると、設定した明細番号内部採番情報を取得することが可能です。
 - 販売管理伝票番号情報を取得

InquiryCreateFromData2Controller クラスの getSalesdocumentin メソッドを使用すると、設定した販売管理伝票番号情報を取得することが可能です。
 - テスト実行情報を取得

InquiryCreateFromData2Controller クラスの getTestrun メソッドを使用すると、設定したテスト実行情報を取得することが可能です。
 - 販売伝票番号情報を取得

InquiryCreateFromData2ControllerクラスのgetSalesdocumentメソッドを使用すると、取得した販売伝票番号情報を取得することが可能です。
 - VBAK、VBAP、VBEP のカスタマ拡張情報を取得

InquiryCreateFromData2ControllerクラスのgetExtensionInListメソッドを使用すると、設定したVBAK、VBAP、VBEP のカスタマ拡張情報を取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 選定: BLOB 内部データ (SCE)情報を取得

InquiryCreateFromData2Controller クラスの getInquiryCfgsBlobList メソッドを使用すると、設定した選定: BLOB 内部データ (SCE)情報を取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 選定: インスタンス情報を取得

- InquiryCreateFromData2Controller クラスの getInquiryCfgsInstList メソッドを使用すると、設定した選定: インスタンス情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 選定: 部分設定情報を取得
InquiryCreateFromData2Controller クラスの getInquiryCfgsPartOfList メソッドを使用すると、設定した選定: 部分設定情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 選定: 参照データ情報を取得
InquiryCreateFromData2Controller クラスの getInquiryCfgsRefList メソッドを使用すると、設定した選定: 参照データ情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 選定: 参照明細/インスタンス情報を取得
InquiryCreateFromData2Controller クラスの getInquiryCfgsRefInstList メソッドを使用すると、設定した選定: 参照明細/インスタンス情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 選定: 特性値情報を取得
InquiryCreateFromData2Controller クラスの getInquiryCfgsValueList メソッドを使用すると、設定した選定: 特性値情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 選定: バリエーション条件キー情報を取得
InquiryCreateFromData2Controller クラスの getInquiryCfgsVkList メソッドを使用すると、設定した選定: バリエーション条件キー情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 条件情報を取得
InquiryCreateFromData2Controller クラスの getInquiryConditionsInList メソッドを使用すると、設定した条件情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 受注における条件更新の通信項目情報を取得
InquiryCreateFromData2Controller クラスの getInquiryConditionsInXList メソッドを使用すると、設定した受注における条件更新の通信項目情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 引合伝票ヘッダデータ情報を取得
InquiryCreateFromData2Controller クラスの getInquiryHeaderIn メソッドを使用すると、設定した引合伝票ヘッダデータ情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - ヘッダデータチェックボックス情報を取得
InquiryCreateFromData2Controller クラスの getInquiryHeaderInX メソッドを使用すると、設定したヘッダデータチェックボックス情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 明細データ情報を取得
InquiryCreateFromData2Controller クラスの getInquiryItemsInList メソッドを使用すると、設定した明細データ情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 明細データチェックボックス情報を取得
InquiryCreateFromData2Controller クラスの getInquiryItemsInXList メソッドを使用すると、設定した明細データチェックボックス情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 参照キーの出力テーブル情報を取得
InquiryCreateFromData2Controller クラスの getInquiryKeysList メソッドを使用すると、設定した参照キーの出力テーブル情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 伝票取引先情報を取得
InquiryCreateFromData2Controller クラスの getInquiryPartnersList メソッドを使用すると、設定した伝票取引先情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 納入日程情報を取得
InquiryCreateFromData2Controller クラスの getInquirySchedulesInList メソッドを使用すると、設定した納入日程情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 納入日程チェックボックス情報を取得
InquiryCreateFromData2Controller クラスの getInquirySchedulesInXList メソッドを使用すると、設定した納入日程

チェックボックス情報を取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

- テキスト情報を取得
 InquiryCreateFromData2Controller クラスの getInquiryTextList メソッドを使用すると、設定したテキスト情報を取得することが可能です。
 項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 内部制御パラメータ情報を取得
 InquiryCreateFromData2Controller クラスの getLogicSwitch メソッドを使用すると、設定した内部制御パラメータ情報を取得することが可能です。
 項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- アドレス (組織/会社) 用 BAPI 参照構造情報を取得
 InquiryCreateFromData2Controller クラスの getPartnerAddressesList メソッドを使用すると、設定したアドレス (組織/会社) 用 BAPI 参照構造情報を取得することが可能です。
 項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 論理システム - 送信者情報を取得
 InquiryCreateFromData2Controller クラスの getSender メソッドを使用すると、設定した論理システム - 送信者情報を取得することが可能です。
 項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。



コラム

サンプルプログラム実行結果では、SAP テスト環境のデータが取得されています。SAP 標準では該当データが存在しません。



注意

引合伝票登録 API に関して、import、structure、table パラメータを持つ各コンポーネントの必須項目や値の相関チェック等を確認する必要がある場合は、SAP 上の汎用モジュールビルダ (t-cd : SE37) 等を使用して、「BAPI_INQUIRY_CREATEFROMDATA2」の BAPI 仕様を確認してください。

見積伝票 登録 API (QuotationCreateFromData2Controller)

見積伝票を登録するためには、QuotationCreateFromData2Controller クラスの持つ各メソッドを利用する必要があります。引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
QuotationCreateFromData2Controller	コンストラクタ (引数無し)
QuotationCreateFromData2Controller	コンストラクタ (第 1 引数あり)。 ConnectionManager を使う場合に使用する。 第 1 引数で「ConnectionManager」を設定する。
QuotationCreateFromData2Controller	コンストラクタ (第 1 引数あり)。 SAPClient の接続情報を使う場合に使用する。 第 1 引数で「SAPClient」 (オブジェクト) を設定する。
setBehaveWhenError	import 項目 : エラー処理を設定します。
setBinaryRelationshiptype	import 項目 : バイナリ関連タイプ (専用)を設定します。
setConvert	import 項目 : 取引先機能/受注伝票タイプの変換を設定します。
setIntNumberAssignment	import 項目 : 明細番号内部採番を設定します。
setSalesdocumentin	import 項目 : 見積伝票番号を設定します。
setTestrun	import 項目 : テスト実行を設定します。
setStrParam	構造型の各項目に値を設定する。 第1引数に structure 名、第2引数に項目名、第3引数に値を設定する。

コンストラクタ / メソッド名	説明
setTblParam	テーブル型の各項目に値を設定する。 第1引数にテーブル名、第2引数に項目名第3引数に値、第4引数にレコード番号を設定する。
executeBapi	BapiController により見積伝票を登録します。
getBapiReturnErrMsg	BAPI実行結果である BapiReturn の Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn オブジェクトのメッセージ情報を全て返します。 全て正常の場合は null が返却されます。
getBapiReturnMessage	BAPI実行結果である BapiReturn の Type 情報が正常(デフォルト値: "S" / "")と判断された Type 情報のメッセージ情報を全て返します。 メッセージが存在しない場合は null が返却されます。
getBehaveWhenError	import 項目: エラー処理を取得します。
getBinaryRelationshiptype	import 項目: バイナリ関連タイプ (専用)を取得します。
getConvert	import 項目: 取引先機能/受注伝票タイプの変換を取得します。
getIntNumberAssignment	import 項目: 明細番号内部採番を取得します。
getSalesdocumentin	import 項目: 見積伝票番号を取得します。
getTestrun	import 項目: テスト実行を取得します。
getSalesdocument	export 項目: 販売管理伝票番号を取得します。
getExtensionInList	import項目: 設定したVBAK、VBAP、VBEP のカスタマ拡張情報を取得します。
getLogicSwitch	import 項目: 設定した内部制御パラメータ情報を取得します。
getPartnerAddressesList	import 項目: 設定したアドレス (組織/会社) 用 BAPI 参照構造情報を取得します。
getQuotationCfgsBlobList	import項目: 設定した選定: BLOB 内部データ (SCE)情報を取得します。
getQuotationCfgsInstList	import 項目: 設定した選定: インスタンス情報を取得します。
getQuotationCfgsPartOfList	import 項目: 設定した選定: 部分設定情報を取得します。
getQuotationCfgsRefList	import 項目: 設定した選定: 参照データ情報を取得します。
getQuotationCfgsRefInstList	import 項目: 設定した選定: 参照明細/インスタンス情報を取得します。
getQuotationCfgsValueList	import 項目: 設定した選定: 特性値情報を取得します。
getQuotationCfgsVkList	import 項目: 設定した選定: バリエーション条件キー情報を取得します。
getQuotationConditionsInList	import 項目: 設定した条件情報を取得します。
getQuotationConditionsInXList	import 項目: 設定した受注における条件更新の通信項目情報を取得します。
getQuotationHeaderIn	import 項目: 設定した伝票ヘッダデータ情報を取得します。
getQuotationHeaderInX	import 項目: 設定した伝票ヘッダチェックボックス情報を取得します。
getQuotationItemsInList	import 項目: 設定した明細データ情報を取得します。
getQuotationItemsInXList	import 項目: 設定した明細データチェックボックス情報を取得します。
getQuotationKeysList	import 項目: 設定した参照キーの出力テーブル情報を取得します。
getQuotationPartnersList	import 項目: 設定した伝票取引先情報を取得します。

コンストラクタ / メソッド名	説明
getQuotationSchedulesInList	import 項目：設定した納入日程行データ情報を取得します。
getQuotationSchedulesInXList	import 項目：設定した納入日程行データチェックボックス情報を取得します。
getQuotationTextList	import 項目：設定したテキスト情報を取得します。
getSender	import 項目：設定した論理システム - 送信者情報を取得します。

サンプルプログラム

```

var salesDocument;
var partneraddressesList;

// グローバル変数 省略

var cfgsreflistCount;
var ret;

function init(request) {

//-----
//変数宣言
//-----
var objRet; // executeBapi実行結果
var objMsg; // executeBapi実行結果メッセージ

//-----
//インスタンス生成 // 説明 1
//-----
var objCon = new QuotationCreateFromData2Controller();

//-----
// テスト実行 // 説明 2
//-----
objCon.setTestrun("");

//-----
// 構造体項目セット // 説明 3
//-----
// 通信項目: 販売管理伝票ヘッダ
objCon.setStrParam("HeaderIn", "DOC_TYPE", "AG"); // 販売伝票タイプ
objCon.setStrParam("HeaderIn", "QT_VALID_F", "20070301"); // 見積/引合有効開始日
objCon.setStrParam("HeaderIn", "QT_VALID_T", "20070331"); // 見積の有効終了日
objCon.setStrParam("HeaderIn", "PURCH_NO_C", "API_見積伝票登録1"); // 出荷先発注番号

//-----
// テーブル項目セット // 説明 4
//-----
// 伝票取引先
objCon.setTblParam("Partners", "PARTN_ROLE", "AG", 1); // 取引先機能
objCon.setTblParam("Partners", "PARTN_NUMB", "0000000012", 1); // 得意先コード1
objCon.setTblParam("Partners", "PARTN_ROLE", "WE", 2); // 取引先機能
objCon.setTblParam("Partners", "PARTN_NUMB", "0000000012", 2); // 得意先コード1
objCon.setTblParam("Partners", "PARTN_ROLE", "RG", 3); // 取引先機能
objCon.setTblParam("Partners", "PARTN_NUMB", "0000000012", 3); // 得意先コード1

//ORDER_ITEMS_IN[明細データ]
objCon.setTblParam("ItemsIn", "MATERIAL", "000000000000000011", 1); // 品目コード

//ORDER_SCHEDULES_IN[納入日程データ]
objCon.setTblParam("SchedulesIn", "REQ_QTY", "3", 1); // 販売単位による受注数量

//-----
// executeBapi実行 // 説明 5
//-----
objRet = objCon.executeBapi();

//-----
// エラー判定

```

```

//-----
//要素[ERR_CODE]に何らかの値がセットされている場合
if (objRet["ERR_CODE"] != null) {
    //エラーメッセージ出力
    Debug.browse(objRet);
}

// 各種データを取得 // 説明 6
// 販売伝票番号をセット
salesDocument = objCon.getSalesdocument();

// アドレス (組織/会社) 用 BAPI 参照構造
partneraddressesList = objCon.getPartnerAddressesList();
partneraddresseslistCount = partneraddressesList.length;

// VBAK、VBAP、VBEP のカスタマ拡張
extensioninList = objCon.getExtensionInList();
extensioninlistCount = extensioninList.length;

// 選定: BLOB 内部データ (SCE)
cfsblobList = objCon.getQuotationCfgsBlobList();
cfsbloblistCount = cfsblobList.length;

// 省略

// 選定: 参照データ
cfsrefList = objCon.getQuotationCfgsRefList();
cfsreflistCount = cfsrefList.length;

//API実行結果を取得します。
ret = objCon.getBapiReturnMessage();
}

```

- 見積伝票登録 API のインスタンス生成(説明1)
QuotationCreateFromData2Controller クラスのインスタンスを生成します。
引数は不要です。
インスタンスを生成すると、見積伝票登録 BAPI (BAPI_QUOTATION_CREATEFROMDATA2) の import 型の import、structure、table パラメータに値を設定することができます。
 - import の各項目に値を設定(説明2)
import パラメータの項目に値を設定する場合は、set メソッドを使用します。
メソッドについては各 API のAPI リストを参照ください。
 - structure の各項目に値を設定(説明3)
structure パラメータの項目に値を設定する場合は、setStrParam メソッドを使用します。
第1引数のstructure 名称は固定値になっています。
後述のとおり名称を設定して下さい。
第2引数の項目名称はSAPの各属性の項目名称を設定してください。
structure 名称、項目名称は英字の大文字、小文字は厳密に区別されます。
第3引数に値を設定します。
- | SAP structure | セットする固有名称 |
|------------------------------------|-------------|
| LOGIC_SWITCH 内部制御パラメータ | LogicSwitch |
| QUOTATION_HEADER_IN 伝票ヘッダデータ | HeaderIn |
| QUOTATION_HEADER_INX 伝票ヘッダチェックボックス | HeaderInX |
| SENDER 論理システム - 送信者 | Sender |
- table の各項目に値を設定(説明4)
table パラメータの項目に値を設定する場合は、setTblParam メソッドを使用します。
第1引数のtable名称は固定値になっています。
後述のとおり名称を設定して下さい。
第2引数の項目名称は SAP の各属性の項目名称を設定してください。

table 名称、項目名称は英字の大文字、小文字は厳密に区別されます。

第3引数に値、第4引数にレコード番号を設定します。

SAP table	セットする固有名称
EXTENSIONIN VBAK、VBAP、VBEP のカスタマ拡張	ExtensionIn
PARTNERADDRESSES アドレス (組織/会社) 用 BAPI 参照構造	PartnerAddresses
QUOTATION_CFGS_BLOB 選定: BLOB 内部データ (SCE)	CfgsBlob
QUOTATION_CFGS_INST 選定: インスタンス	CfgsInst
QUOTATION_CFGS_PART_OF 選定: 部分設定	CfgsPartOf
QUOTATION_CFGS_REF 選定: 参照データ	CfgsRef
QUOTATION_CFGS_REFINST 選定: 参照明細/インスタンス	CfgsRefInst
QUOTATION_CFGS_VALUE 選定: 特性値	CfgsValue
QUOTATION_CFGS_VK 選定: バリエーション条件キー	CfgsVk
QUOTATION_CONDITIONS_IN 条件	ConditionsIn
QUOTATION_CONDITIONS_INX 受注における条件更新の通信項目	ConditionsInX
QUOTATION_ITEMS_IN 明細データ	ItemsIn
QUOTATION_ITEMS_INX 明細データチェックボックス	ItemsInX
QUOTATION_KEYS 参照キーの出力テーブル	Keys
QUOTATION_PARTNERS 伝票取引先	Partners
QUOTATION_SCHEDULES_IN 納入日程行	SchedulesIn
QUOTATION_SCHEDULES_INX 納入日程行チェックボックス	SchedulesInX
QUOTATION_TEXT テキスト	Text

- 見積伝票の登録(説明5)

QuotationCreateFromData2Controller クラスの executeBapi メソッドを実行すると、各項目にセットされた値が SAP 上に見積伝票データとして登録されます。

登録が成功した場合には、見積伝票登録 API の内部で自動的に commit 処理が実行されます。

登録に失敗した場合には、自動的に rollback 処理が実行されます。

成功・失敗の基準は、見積伝票登録用 BAPI (BAPI_QUOTATION_CREATEFROMDATA2) の実行後に RETURN パラメータの TYPE 値を取得して、S (正常終了)、空文字以外が1つでも存在するかどうかです。

1つでも存在する場合は、rollback が自動的に実行されます。
- サンプルプログラムの中で記載されたデータ取得メソッドの詳細説明については、以下を参照してください。(説明6)
 - Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn を取得

QuotationCreateFromData2Controller クラスの getBapiReturnErrMsgMessage メソッドを使用すると、取得した BapiReturn の Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn オブジェクトのメッセージ情報を取得することが可能です。

全て正常の場合は null を取得します。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - Type 情報が正常(デフォルト値: "S" / "")の BapiReturn を取得

QuotationCreateFromData2Controller クラスの getBapiReturnMessage メソッドを使用すると、取得した BapiReturn の Type 情報が正常(デフォルト値: "S" / "")と判断された Type 情報のメッセージ情報を取得することが可能です。

メッセージが存在しない場合は null を取得します。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - エラー処理情報を取得

QuotationCreateFromData2Controller クラスの getBehaveWhenError メソッドを使用すると、設定したエラー処理情報を取得することが可能です。
 - バイナリ関連タイプ(専用)情報を取得

QuotationCreateFromData2Controller クラスの getBinaryRelationshiptype メソッドを使用すると、設定したバイナリ関連タイプ(専用)情報を取得することが可能です。
 - 取引先機能/受注伝票タイプの変換情報を取得

QuotationCreateFromData2Controller クラスの getConvert メソッドを使用すると、設定した取引先機能/受注伝票タイプの変換情報を取得することが可能です。

- 明細番号内部採番情報を取得
QuotationCreateFromData2Controller クラスの getIntNumberAssignment メソッドを使用すると、設定した明細番号内部採番情報を取得することが可能です。
- 見積伝票番号情報を取得
QuotationCreateFromData2Controller クラスの getSalesdocumentin メソッドを使用すると、設定した見積伝票番号情報を取得することが可能です。
- テスト実行情報を取得
QuotationCreateFromData2Controller クラスの getTestrun メソッドを使用すると、設定したテスト実行情報を取得することが可能です。
- 販売管理伝票番号情報を取得
QuotationCreateFromData2Controller クラスの getSalesdocument メソッドを使用すると、取得した販売管理伝票番号情報を取得することが可能です。
- VBAK、VBAP、VBEP のカスタマ拡張情報を取得
QuotationCreateFromData2Controller クラスの getExtensionInList メソッドを使用すると、設定したVBAK、VBAP、VBEP のカスタマ拡張情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 内部制御パラメータ情報を取得
QuotationCreateFromData2Controller クラスの getLogicSwitch メソッドを使用すると、設定した内部制御パラメータ情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- アドレス (組織/会社) 用 BAPI 参照構造情報を取得
QuotationCreateFromData2Controller クラスの getPartnerAddressesList メソッドを使用すると、設定したアドレス (組織/会社) 用 BAPI 参照構造情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 選定: BLOB 内部データ (SCE)情報を取得
QuotationCreateFromData2Controller クラスの getQuotationCfgsBlobList メソッドを使用すると、設定した選定: BLOB 内部データ (SCE)情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 選定: インスタンス情報を取得
QuotationCreateFromData2Controller クラスの getQuotationCfgsInstList メソッドを使用すると、設定した選定: インスタンス情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 選定: 部分設定情報を取得
QuotationCreateFromData2Controller クラスの getQuotationCfgsPartOfList メソッドを使用すると、設定した選定: 部分設定情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 選定: 参照データ情報を取得
QuotationCreateFromData2Controller クラスの getQuotationCfgsRefList メソッドを使用すると、設定した選定: 参照データ情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 選定: 参照明細/インスタンス情報を取得
QuotationCreateFromData2Controller クラスの getQuotationCfgsRefInstList メソッドを使用すると、設定した選定: 参照明細/インスタンス情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 選定: 特性値情報を取得
QuotationCreateFromData2Controller クラスの getQuotationCfgsValueList メソッドを使用すると、設定した選定: 特性値情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 選定: バリエーション条件キー情報を取得
QuotationCreateFromData2Controller クラスの getQuotationCfgsVkList メソッドを使用すると、設定した 選定: バリエーション条件キー情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 条件情報を取得
QuotationCreateFromData2Controller クラスの getQuotationConditionsInList メソッドを使用すると、設定した条件情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 受注における条件更新の通信項目情報を取得
QuotationCreateFromData2Controller クラスの getQuotationConditionsInXList メソッドを使用すると、設定した受注における条件更新の通信項目情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

- 伝票ヘッダデータ情報を取得
 QuotationCreateFromData2Controller クラスの getQuotationHeaderIn メソッドを使用すると、設定した伝票ヘッダデータ情報を取得することが可能です。
 項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 伝票ヘッダチェックボックス情報を取得
 QuotationCreateFromData2Controller クラスの getQuotationHeaderInX メソッドを使用すると、設定した伝票ヘッダチェックボックス情報を取得することが可能です。
 項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 明細データ情報を取得
 QuotationCreateFromData2Controller クラスの getQuotationItemsInList メソッドを使用すると、設定した明細データ情報を取得することが可能です。
 項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 明細データチェックボックス情報を取得
 QuotationCreateFromData2Controller クラスの getQuotationItemsInXList メソッドを使用すると、設定した明細データチェックボックス情報を取得することが可能です。
 項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 参照キーの出力テーブル情報を取得
 QuotationCreateFromData2Controller クラスの getQuotationKeysList メソッドを使用すると、設定した参照キーの出力テーブル情報を取得することが可能です。
 項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 伝票取引先情報を取得
 QuotationCreateFromData2Controller クラスの getQuotationPartnersList メソッドを使用すると、設定した伝票取引先情報を取得することが可能です。
 項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 納入日程行データ情報を取得
 QuotationCreateFromData2Controller クラスの getQuotationSchedulesInList メソッドを使用すると、設定した納入日程行データ情報を取得することが可能です。
 項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 納入日程行データチェックボックス情報を取得
 QuotationCreateFromData2Controller クラスの getQuotationSchedulesInXList メソッドを使用すると、設定した納入日程行データチェックボックス情報を取得することが可能です。
 項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- テキスト情報を取得
 QuotationCreateFromData2Controller クラスの getQuotationTextList メソッドを使用すると、設定したテキスト情報を取得することが可能です。
 項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 論理システム - 送信者情報を取得
 QuotationCreateFromData2Controller クラスの getSender メソッドを使用すると、設定した論理システム - 送信者情報を取得することが可能です。
 項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。



コラム

サンプルプログラム実行結果では、SAP テスト環境のデータが取得されています。SAP標準では該当データが存在しません。



注意

見積伝票登録 API に関して、import、structure、table パラメータを持つ各コンポーネントの必須項目や値の関連チェック等を確認する必要がある場合は、SAP 上の汎用モジュールビルダ (t-cd : SE37) 等を使用して、「BAPI_QUOTATION_CREATEFROMDATA2」の BAPI 仕様を確認してください。

受注伝票 登録 API (SalesOrderCreateFromDat2Controller)

受注伝票を登録するためには、SalesOrderCreateFromDat2Controller クラスの持つ各メソッドを利用する必要があります。引数、戻り値についての詳細な情報は、API リストを参照してください。
 また、登録可能な項目は、見積伝票として必要最低限必要と判断される項目のみとなっています。

各メソッドの説明

コンストラクタ / メソッド名	説明
SalesOrderCreateFromDat2Controller	コンストラクタ (引数無し)

コンストラクタ / メソッド名	説明
SalesOrderCreateFromDat2Controller	コンストラクタ (第 1 引数あり)。 ConnectionManager を使う場合に使用する。 第 1 引数で「ConnectionManager」を設定する。
SalesOrderCreateFromDat2Controller	コンストラクタ (第 1 引数あり)。 SAPClient の接続情報を使う場合に使用する。 第 1 引数で「SAPClient」 (オブジェクト) を設定する。
setBehaveWhenError	import 項目: エラー処理を設定します。
setBinaryRelationshiptype	import 項目: バイナリ関連タイプ (専用)を設定します。
setConvert	import 項目: 変換: 取引先機能 + 受注伝票タイプを設定します。
setIntNumberAssignment	import 項目: 明細番号内部採番を設定します。
setSalesdocumentin	import 項目: 販売管理伝票番号を設定します。
setTestrun	import 項目: テスト実行を設定します。
setStrParam	構造型の各項目に値を設定する。 第1引数に structure 名、第2引数に項目名、第3引数に値を設定する。
setTblParam	テーブル型の各項目に値を設定する。 第1引数にテーブル名、第2引数に項目名、第3引数に値、第4引数にレコード番号を設定する。
executeBapi	BapiController により販売伝票を登録します。
getBapiReturnErrMsgage	BAPI実行結果である BapiReturn の Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn オブジェクトのメッセージ情報を全て返します。 全て正常の場合は null が返却されます。
getBapiReturnMessage	BAPI実行結果である BapiReturn の Type 情報が正常(デフォルト値: "S" / "")と判断された Type 情報のメッセージ情報を全て返します。 メッセージが存在しない場合は null が返却されます。
getBehaveWhenError	import 項目: エラー処理を取得します。
getBinaryRelationshiptype	import 項目: バイナリ関連タイプ (専用)を取得します。
getConvert	import 項目: 変換: 取引先機能 + 受注伝票タイプを取得します。
getIntNumberAssignment	import 項目: 明細番号内部採番を取得します。
getSalesdocument	import 項目: 生成された伝票の番号を取得します。
getTestrun	import 項目: テスト実行を取得します。
getSalesdocumentin	export 項目: 販売管理伝票番号を取得します。
getExtensionInList	import項目: 設定したVBAK、VBAP、VBEP のカスタマ拡張情報を取得します。
getLogicSwitch	import 項目: 設定した内部制御パラメータ情報を取得します。
getOrderCcardList	import 項目: 設定したクレジットカードデータ情報を取得します。
getOrderCfgsBlobList	import項目: 設定した選定: BLOB 内部データ (SCE)情報を取得します。
getOrderCfgsInstList	import 項目: 設定した選定: インスタンス情報を取得します。
getOrderCfgsPartOfList	import 項目: 設定した選定: 部分設定情報を取得します。
getOrderCfgsRefList	import 項目: 設定した選定: 参照データ情報を取得します。
getOrderCfgsRefInstList	import 項目: 設定した選定: 参照明細/インスタンス情報を取得します。

コンストラクタ / メソッド名	説明
getOrderCfgsValueList	import 項目：設定した選定: 特性値情報を取得します。
getOrderCfgsVkList	import 項目：設定した選定: バリエーション条件キー情報を取得します。
getOrderConditionsInList	import 項目：設定した条件情報を取得します。
getOrderConditionsInXList	import 項目：設定した条件チェックボックス情報を取得します。
getOrderHeaderIn	import 項目：設定した受注伝票ヘッダ情報を取得します。
getOrderHeaderInX	import 項目：設定した受注伝票チェックリスト情報を取得します。
getOrderItemsInList	import 項目：設定した明細データ情報を取得します。
getOrderItemsInXList	import 項目：設定した明細データチェックボックス情報を取得します。
getOrderKeysList	import 項目：設定した参照キーの出力テーブル情報を取得します。
getOrderPartnersList	import 項目：設定した伝票取引先情報を取得します。
getOrderSchedulesInList	import 項目：設定した納入日程行データ情報を取得します。
getOrderSchedulesInXList	import 項目：設定した納入日程行データチェックボックス情報を取得します。
getOrderTextList	import 項目：設定したテキスト情報を取得します。
getPartnerAddressesList	import 項目：設定したアドレス (組織/会社) 用 BAPI 参照構造情報を取得します。
getSender	import 項目：設定した論理システム - 送信者情報を取得します。

サンプルプログラム

```

var salesDocument;
var partneraddressesList;

// グローバル変数 省略

var cfgsreflistCount;
var ret;

function init(request) {
    //-----
    //変数宣言
    //-----
    var objRet; // executeBapi実行結果
    var objMsg; // executeBapi実行結果メッセージ

    //-----
    //インスタンス生成 // 説明 1
    //-----
    var objCon = new SalesOrderCreateFromDat2Controller();

    //-----
    // テスト実行 // 説明 2
    //-----
    objCon.setTestrun("");

    //-----
    // 構造体項目セット // 説明 3
    //-----
    // 通信項目: 販売管理伝票ヘッダ
    objCon.setStrParam("HeaderIn", "DOC_TYPE", "TA"); // 販売伝票タイプ
    objCon.setStrParam("HeaderIn", "PRICE_DATE", "20070301"); // 見積/引合有効開始日
    objCon.setStrParam("HeaderIn", "PURCH_NO_C", "API_受注伝票登録1"); // 出荷先発注番号

    //-----

```

```

// テーブル項目セット // 説明 4
//-----
// 伝票取引先
objCon.setTblParam("Partners", "PARTN_ROLE", "AG", 1); // 取引先機能
objCon.setTblParam("Partners", "PARTN_NUMB", "0000000012", 1); // 得意先コード1
objCon.setTblParam("Partners", "PARTN_ROLE", "WE", 2); // 取引先機能
objCon.setTblParam("Partners", "PARTN_NUMB", "0000000012", 2); // 得意先コード1
objCon.setTblParam("Partners", "PARTN_ROLE", "RG", 3); // 取引先機能
objCon.setTblParam("Partners", "PARTN_NUMB", "0000000012", 3); // 得意先コード1

//ORDER_ITEMS_IN[明細データ]
objCon.setTblParam("ItemsIn", "ITM_NUMBER", "000010", 1); // 販売伝票明細
objCon.setTblParam("ItemsIn", "MATERIAL", "000000000000000011", 1); // 品目コード

//ORDER_SCHEDULES_IN[納入日程行データ]
objCon.setTblParam("SchedulesIn", "ITM_NUMBER", "000010", 1); // 販売伝票明細
objCon.setTblParam("SchedulesIn", "SCHED_LINE", "0010", 1); // 納入日程行
objCon.setTblParam("SchedulesIn", "REQ_DATE", "20070302", 1); // 納入日程日付
objCon.setTblParam("SchedulesIn", "REQ_QTY", "1", 1); // 販売単位による受注数量

objCon.setTblParam("SchedulesIn", "ITM_NUMBER", "000010", 2); // 販売伝票明細
objCon.setTblParam("SchedulesIn", "SCHED_LINE", "0020", 2); // 納入日程行
objCon.setTblParam("SchedulesIn", "REQ_DATE", "20070303", 2); // 納入日程日付
objCon.setTblParam("SchedulesIn", "REQ_QTY", "2", 2); // 販売単位による受注数量

//-----
// executeBapi実行 // 説明 5
//-----
objRet = objCon.executeBapi();

//-----
// エラー判定
//-----
//要素[ERR_CODE]に何らかの値がセットされている場合
if (objRet["ERR_CODE"] != null) {
    //エラーメッセージ出力
    Debug.browse(objRet);
}

// オブジェクトを取得 // 説明 6
// 販売伝票番号をセット
salesDocument = objCon.getSalesdocument();

// アドレス (組織/会社) 用 BAPI 参照構造
partneraddressesList = objCon.getPartnerAddressesList();
partneraddresseslistCount = partneraddressesList.length;

// VBAK、VBAP、VBEP のカスタマ拡張
extensioninList = objCon.getExtensionInList();
extensioninlistCount = extensioninList.length;

// 省略

// 選定: 参照データ
cfsrefsrefList = objCon.getOrderCfgrsRefList();
cfsrefsreflistCount = cfsrefsrefList.length;

//API実行結果を取得します。
objMsg = objCon.getBapiReturnMessage();
ret = objMsg;
}

```

- 受注伝票登録 API のインスタンス生成(説明1)
SalesOrderCreateFromDat2Controller クラスのインスタンスを生成します。
引数は不要です。
インスタンスを生成すると、受注伝票登録 BAPI (BAPI_SALESORDER_CREATEFROMDAT2) の import 型の import、structure、table パラメータに値を設定することができます。
- import の各項目に値を設定(説明2)

import パラメータの項目に値を設定する場合は、set メソッドを使用します。メソッドについては各 API のAPI リストを参照ください。

- structure の各項目に値を設定(説明3)

structure パラメータの項目に値を設定する場合は、setStrParam メソッドを使用します。

第1引数のstructure 名称は固定値になっています。

後述のとおり名称を設定して下さい。

第2引数の項目名称はSAPの各属性の項目名称を設定してください。

structure 名称、項目名称は英字の大文字、小文字は厳密に区別されます。

第3引数に値を設定します。

SAP structure	セットする固有名称
LOGIC_SWITCH 内部制御パラメータ	LogicSwitch
ORDER_HEADER_IN 受注伝票ヘッダ	HeaderIn
ORDER_HEADER_INX 受注伝票チェックリスト	HeaderInX
SENDER 論理システム - 送信者	Sender

- table の各項目に値を設定(説明4)

table パラメータの項目に値を設定する場合は、setTblParam メソッドを使用します。

第1引数のtable名称は固定値になっています。

後述のとおり名称を設定して下さい。

第2引数の項目名称は SAPの各属性の項目名称を設定してください。

table 名称、項目名称は英字の大文字、小文字は厳密に区別されます。

第3引数に値、第4引数にレコード番号を設定します。

SAP table	セットする固有名称
EXTENSIONIN VBAK、VBAP、VBEP のカスタマ拡張	ExtensionIn
ORDER_CCARD クレジットカードデータ	OrderCcard
ORDER_CFGS_BLOB 選定: BLOB 内部データ (SCE)	CfgsBlob
ORDER_CFGS_INST 選定: インスタンス	CfgsInst
ORDER_CFGS_PART_OF 選定: 部分設定	CfgsPartOf
ORDER_CFGS_REF 選定: 参照データ	CfgsRef
ORDER_CFGS_REFINST 選定: 参照明細/インスタンス	CfgsRefInst
ORDER_CFGS_VALUE 選定: 特性値	CfgsValue
ORDER_CFGS_VK 選定: バリエーション条件キー	CfgsVk
ORDER_CONDITIONS_IN 条件	ConditionsIn
ORDER_CONDITIONS_INX 条件チェックボックス	ConditionsInX
ORDER_ITEMS_IN 明細データ	ItemsIn
ORDER_ITEMS_INX 明細データチェックボックス	ItemsInX
ORDER_KEYS 参照キーの出力テーブル	Keys
ORDER_PARTNERS 伝票取引先	Partners
ORDER_SCHEDULES_IN 納入日程行データ	SchedulesIn
ORDER_SCHEDULES_INX 納入日程行データチェックボックス	SchedulesInX
ORDER_TEXT テキスト	Text
PARTNERADDRESSES アドレス (組織/会社) 用 BAPI 参照構造	PartnerAddresses

- 受注伝票の登録(説明5)

SalesOrderCreateFromDat2Controller クラスの executeBapi メソッドを実行すると、各項目にセットされた値が SAP 上に受注伝票データとして登録されます。

登録が成功した場合には、受注伝票登録 API の内部で自動的に commit 処理が実行されます。

登録に失敗した場合には、自動的に rollback 処理が実行されます。

成功・失敗の基準は、受注伝票登録用 BAPI (BAPI_SALESORDER_CREATEFROMDAT2) の実行後に RETURN パラメータの TYPE 値を取得して、S (正常終了)、空文字以外が1つでも存在するかどうかです。

1つでも存在する場合は、rollback が自動的に実行されます。

- サンプルプログラムの中で記載されたデータ取得メソッドの詳細説明については、以下を参照してください。(説明6)
 - Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn を取得
SalesOrderCreateFromDat2Controller クラスの getBapiReturnErrorMessage メソッドを使用すると、取得した BapiReturn の Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn オブジェクトのメッセージ情報を取得することが可能です。
全て正常の場合は null を取得します。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - Type 情報が正常(デフォルト値: "S" / "")の BapiReturn を取得
SalesOrderCreateFromDat2Controller クラスの getBapiReturnMessage メソッドを使用すると、取得した BapiReturn の Type 情報が正常(デフォルト値: "S" / "")と判断された Type 情報のメッセージ情報を取得することが可能です。
メッセージが存在しない場合は null を取得します。
項目の値には、KEY に SAP の各属性の項目名称を設定してください。
 - エラー処理情報を取得
SalesOrderCreateFromDat2Controller クラスの getBehaveWhenError メソッドを使用すると、設定したエラー処理情報を取得することが可能です。
 - バイナリ関連タイプ (専用)情報を取得
SalesOrderCreateFromDat2Controller クラスの getBinaryRelationshiptype メソッドを使用すると、設定したバイナリ関連タイプ (専用)情報を取得することが可能です。
 - 変換: 取引先機能 + 受注伝票タイプ情報を取得
SalesOrderCreateFromDat2Controller クラスの getConvert メソッドを使用すると、設定した変換: 取引先機能 + 受注伝票タイプ情報を取得することが可能です。
 - 明細番号内部採番情報を取得
SalesOrderCreateFromDat2Controller クラスの getIntNumberAssignment メソッドを使用すると、設定した明細番号内部採番情報を取得することが可能です。
 - 生成された伝票の番号情報を取得
SalesOrderCreateFromDat2Controller クラスの getSalesdocument メソッドを使用すると、設定した生成された伝票の番号情報を取得することが可能です。
 - テスト実行情報を取得
SalesOrderCreateFromDat2Controller クラスの getTestrun メソッドを使用すると、設定したテスト実行情報を取得することが可能です。
 - 販売管理伝票番号情報を取得
SalesOrderCreateFromDat2Controller クラスの getSalesdocumentin メソッドを使用すると、取得した販売管理伝票番号情報を取得することが可能です。
 - VBAK、VBAP、VBEP のカスタマ拡張情報を取得
SalesOrderCreateFromDat2Controller クラスの getExtensionInList メソッドを使用すると、設定した VBAK、VBAP、VBEP のカスタマ拡張情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 内部制御パラメータ情報を取得
SalesOrderCreateFromDat2Controller クラスの getLogicSwitch メソッドを使用すると、設定した内部制御パラメータ情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - クレジットカードデータ情報を取得
SalesOrderCreateFromDat2Controller クラスの getOrderCcardList メソッドを使用すると、設定したクレジットカードデータ情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 選定: BLOB 内部データ (SCE)情報を取得
SalesOrderCreateFromDat2Controller クラスの getOrderCfgsBlobList メソッドを使用すると、設定した選定: BLOB 内部データ (SCE)情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
 - 選定: インスタンス情報を取得
SalesOrderCreateFromDat2Controller クラスの getOrderCfgsInstList メソッドを使用すると、設定した選定: インスタンス情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

- 選定: 部分設定情報を取得
SalesOrderCreateFromDat2Controller クラスの getOrderCfgsPartOfList メソッドを使用すると、設定した選定: 部分設定情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 選定: 参照データ情報を取得
SalesOrderCreateFromDat2Controller クラスの getOrderCfgsRefList メソッドを使用すると、設定した選定: 参照データ情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 選定: 参照明細/インスタンス情報を取得
SalesOrderCreateFromDat2Controller クラスの getOrderCfgsRefInstList メソッドを使用すると、設定した選定: 参照明細/インスタンス情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 選定: 特性値情報を取得
SalesOrderCreateFromDat2Controller クラスの getOrderCfgsValueList メソッドを使用すると、設定した選定: 特性値情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 選定: バリエーション条件キー情報を取得
SalesOrderCreateFromDat2Controller クラスの getOrderCfgsVkList メソッドを使用すると、設定した選定: バリエーション条件キー情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 条件情報を取得
SalesOrderCreateFromDat2Controller クラスの getOrderConditionsInList メソッドを使用すると、設定した条件情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 条件チェックボックス情報を取得
SalesOrderCreateFromDat2Controller クラスの getOrderConditionsInXList メソッドを使用すると、設定した条件チェックボックス情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 受注伝票ヘッダ情報を取得
SalesOrderCreateFromDat2Controller クラスの getOrderHeaderIn メソッドを使用すると、設定した受注伝票ヘッダ情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 受注伝票チェックリスト情報を取得
SalesOrderCreateFromDat2Controller クラスの getOrderHeaderInX メソッドを使用すると、設定した受注伝票チェックリスト情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 明細データ情報を取得
SalesOrderCreateFromDat2Controller クラスの getOrderItemsInList メソッドを使用すると、設定した明細データ情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 明細データチェックボックス情報を取得
SalesOrderCreateFromDat2Controller クラスの getOrderItemsInXList メソッドを使用すると、設定した明細データチェックボックス情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 参照キーの出力テーブル情報を取得
SalesOrderCreateFromDat2Controller クラスの getOrderKeysList メソッドを使用すると、設定した参照キーの出力テーブル情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 伝票取引先情報を取得
SalesOrderCreateFromDat2Controller クラスの getOrderPartnersList メソッドを使用すると、設定した伝票取引先情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 納入日程行データ情報を取得
SalesOrderCreateFromDat2Controller クラスの getOrderSchedulesInList メソッドを使用すると、設定した納入日程行データ情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 納入日程行データチェックボックス情報を取得
SalesOrderCreateFromDat2Controller クラスの getOrderSchedulesInXList メソッドを使用すると、設定した納入日程行データチェックボックス情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- テキスト情報を取得

SalesOrderCreateFromDat2Controller クラスのgetOrderTextListメソッドを使用すると、設定したテキスト情報を取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

- アドレス (組織/会社) 用 BAPI 参照構造情報を取得

SalesOrderCreateFromDat2Controller クラスの getPartnerAddressesList メソッドを使用すると、設定したアドレス (組織/会社) 用 BAPI 参照構造情報を取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

- 論理システム - 送信者情報を取得

SalesOrderCreateFromDat2Controller クラスの getSender メソッドを使用すると、設定した論理システム - 送信者情報を取得することが可能です。

項目の値を取得するには、KEY に SAP R/4 の各属性の項目名称を設定してください。



コラム

サンプルプログラム実行結果では、SAP テスト環境のデータが取得されています。SAP 標準では該当データが存在しません。



注意

受注伝票登録 API に関して、import、structure、table パラメータを持つ各コンポーネントの必須項目や値の関連チェック等を確認する必要がある場合は、SAP 上の汎用モジュールビルダ (t-cd : SE37) 等を使用して、「BAPI_SALESORDER_CREATEFROMDAT2」の BAPI 仕様を確認してください。

販売伝票一覧 照会 API (SalesOrderGetListController)

販売伝票一覧を照会するためには、SalesOrderGetListController クラスの持つ各メソッドを利用する必要があります。引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
SalesOrderGetListController	コンストラクタ (引数無し)
SalesOrderGetListController	コンストラクタ (第 1 引数あり)。 ConnectionManager を使う場合に使用する。 第 1 引数で「ConnectionManager」を設定する。
SalesOrderGetListController	コンストラクタ (第 1 引数あり)。 SAPClient の接続情報を使う場合に使用する。 第 1 引数で「SAPClient」 (オブジェクト) を設定する。
setCustomerNumber	import 項目 : 得意先コードを設定します。
setDocumentDate	import 項目 : 登録日付を設定します。
setDocumentDateTo	import 項目 : 入力日付 (以前/以降)を設定します。
setMaterial	import 項目 : 品目コードを設定します。
setPurchaseOrder	import 項目 : 得意先発注番号を設定します。
setPurchaseOrderNumber	import 項目 : 得意先発注番号を設定します。
setSalesOrganization	import 項目 : 販売組織を設定します。
setTransactionGroup	import 項目 : 一文字フラグを設定します。
setStrParam	構造型の各項目に値を設定する。 第1引数に structure 名、第2引数に項目名、第3引数に値を設定する。
setTblParam	テーブル型の各項目に値を設定する。 第1引数にテーブル名、第2引数に項目名第3引数に値、第4引数にレコード番号を設定する。
executeBapi	BapiController により販売伝票を取得します。

コンストラクタ / メソッド名	説明
getBapiReturnErrMsg	BAPI実行結果である BapiReturn の Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn オブジェクトのメッセージ情報を全て返します。 全て正常の場合は null が返却されます。
getBapiReturnMessage	BAPI実行結果であるBapiReturnのType情報が正常(デフォルト値: "S" / "")と判断された Type 情報のメッセージ情報を全て返します。 メッセージが存在しない場合は null が返却されます。
getCustomerNumber	import 項目：得意先コードを取得します。
getDocumentDate	import 項目：登録日付を取得します。
getDocumentDateTo	import 項目：入力日付 (以前/以降)を取得します。
getMaterial	import 項目：品目コードを取得します。
getPurchaseOrder	import 項目：得意先発注番号を取得します。
getPurchaseOrderNumber	import 項目：得意先発注番号を取得します。
getSalesOrganization	import 項目：販売組織を取得します。
getTransactionGroup	import 項目：一文字フラグを取得します。
getMaterialEvg	import 項目：品目コード(長)情報を取得します。
getSalesOrdersList	得意先受注テーブル情報を取得します。

サンプルプログラム

```
//■販売伝票一覧
var salesOrderList;
var salesOrderCount;

function init(request) {
//-----
//変数宣言
//-----
var objCon;
var objResult;

//-----
//インスタンス生成 // 説明 2
//-----
objCon = new SalesOrderGetListController();

//-----
// 単一項目セット // 説明 2
//-----
objCon.setCustomerNumber("0000000012");//得意先コード
objCon.setSalesOrganization("1100");//販売組織
objCon.setTransactionGroup("0");//一文字フラグ

//-----
// executeBapi実行 // 説明 5
//-----
objResult = objCon.executeBapi();

//-----
// エラー判定 // 説明 6
//-----
if (objResult["ERR_CODE"] != null){
    Debug.browse(objResult);
}

//オブジェクトを取得
//伝票明細
salesOrderList = objCon.getSalesOrdersList();
salesOrderCount = salesOrderList.length;
}
}
```

- 販売伝票一覧照会 API のインスタンス生成(説明1)
SalesOrderGetListController クラスのインスタンスを生成します。
引数は不要です。
インスタンスを生成すると、販売伝票一覧BAPI (BAPI_SALESORDER_GETLIST) のimport型のimport、table パラメータに値を設定することができます。
- import の各項目に値を設定(説明2)
import パラメータの項目に値を設定する場合は、set メソッドを使用します。
メソッドについては各 API のAPI リストを参照ください。
- structure の各項目に値を設定 (説明3)
structure パラメータの項目に値を設定する場合は、setStrParam メソッドを使用します。
第1引数のstructure 名称は固定値になっています。
後述のとおり名称を設定して下さい。
第2引数の項目名称はSAP の各属性の項目名称を設定してください。
structure 名称、項目名称は英字の大文字、小文字は厳密に区別されます。
第三引数に値を設定します。

SAP structure	セットする固有名称
MATERIAL_EVG (品目コード (長)	MaterialEvg
)	

- table の各項目に値を設定 (説明4)
table パラメータの項目に値を設定する場合は、setTblParam メソッドを使用します。

第1引数のtable名称は固定値になっています。

後述のとおり名称を設定して下さい。

第2引数の項目名称は SAP の各属性の項目名称を設定してください。

table 名称、項目名称は英字の大文字、小文字は厳密に区別されます。

第3引数に値、第四引数にレコード番号を設定します。

SAP table	セットする固有名称
SALES_ORDERS (得意先受注テーブル)	SalesOrders

- 販売伝票一覧照会(説明5)

SalesOrderGetListControllerクラスのexecuteBapiメソッドを実行すると、各項目にセットされた値でSAPより販売伝票一覧データとして照会されます。

販売伝票一覧照会用 BAPI (BAPI_GOODSMVT_GETITEMS) の実行後に RETURN パラメータの内容も取得します。

- サンプルプログラムの中で記載されたデータ取得メソッドの詳細説明については、以下を参照してください。(説明6)
 - Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn を取得

SalesOrderGetListController クラスの getBapiReturnErrorMessage メソッドを使用すると、取得したBapiReturn の Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn オブジェクトのメッセージ情報を取得することが可能です。

全て正常の場合は null を取得します。

項目の値を取得するには、KEYにSAPの各属性の項目名称を設定してください。
 - Type 情報が正常(デフォルト値: "S" / "")の BapiReturn を取得

SalesOrderGetListController クラスの getBapiReturnMessageメソッドを使用すると、取得した BapiReturn の Type 情報が正常(デフォルト値: "S" / "")と判断された Type 情報のメッセージ情報を取得することが可能です。

メッセージが存在しない場合は null を取得します。

項目の値を取得するには、KEYにSAPの各属性の項目名称を設定してください。
 - 得意先コード情報を取得

SalesOrderGetListControllerクラスのgetCustomerNumberメソッドを使用すると、設定した得意先コード情報を取得することが可能です。
 - 登録日付情報を取得

SalesOrderGetListController クラスの getDocumentDate メソッドを使用すると、設定した登録日付情報を取得することが可能です。
 - 入力日付 (以前/以降)情報を取得

SalesOrderGetListController クラスの getDocumentDateTo メソッドを使用すると、設定した入力日付(以前/以降)情報を取得することが可能です。
 - 品目コード情報を取得

SalesOrderGetListController クラスの getMaterial メソッドを使用すると、設定した品目コード情報を取得することが可能です。
 - 得意先発注番号情報を取得

SalesOrderGetListController クラスの getPurchaseOrder メソッドを使用すると、設定した得意先発注番号情報を取得することが可能です。
 - 得意先発注番号情報を取得

SalesOrderGetListController クラスの getPurchaseOrderNumber メソッドを使用すると、設定した得意先発注番号情報を取得することが可能です。

- 販売組織情報を取得

SalesOrderGetListController クラスの getSalesOrganization メソッドを使用すると、設定した販売組織情報を取得することが可能です。
- 一文字フラグ情報を取得

SalesOrderGetListController クラスの getTransactionGroup メソッドを使用すると、設定した一文字フラグ情報を取得することが可能です。
- 品目コード (長)情報を取得

SalesOrderGetListController クラスの getMaterialEvg メソッドを使用すると、設定した品目コード (長)情報を取得することが可能です。
- 得意先受注テーブル情報を取得

SalesOrderGetListController クラスの getSalesOrdersList メソッドを使用すると、設定した得意先受注テーブル情報を取得することが可能です。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。



コラム

サンプルプログラム実行結果では、SAP テスト環境のデータが取得されています。SAP 標準では該当データが存在しません。

販売伝票ステータス一覧 照会 API (SalesOrderGetStatusController)

販売伝票ステータス一覧を照会するためには、SalesOrderGetStatusController クラスの持つ各メソッドを利用する必要があります。引数、戻り値についての詳細な情報は、API リストを参照してください。

各メソッドの説明

コンストラクタ / メソッド名	説明
SalesOrderGetStatusController	コンストラクタ (引数無し)
SalesOrderGetStatusController	コンストラクタ (第 1 引数あり)。 ConnectionManager を使う場合に使用する。 第 1 引数で「ConnectionManager」を設定する。
SalesOrderGetStatusController	コンストラクタ (第 1 引数あり)。 SAPClient の接続情報を使う場合に使用する。 第 1 引数で「SAPClient」 (オブジェクト) を設定する。
setSalesdocument	import 項目 : 販売管理伝票番号を設定します。
setTblParam	テーブル型の各項目に値を設定する。 第1引数にテーブル名、第2引数に項目名第3引数に値、第4引数にレコード番号を設定する。
executeBapi	BapiController により販売伝票ステータスを取得します。
getBapiReturnErrMsg	BAPI実行結果である BapiReturn の Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn オブジェクトのメッセージ情報を全て返します。 全て正常の場合は null が返却されます。
getBapiReturnMessage	BAPI実行結果である BapiReturn の Type 情報が正常(デフォルト値: "S" / "")と判断された Type 情報のメッセージ情報を全て返します。 メッセージが存在しない場合は null が返却されます。
getSalesdocument	import 項目 : 販売管理伝票番号を取得します。
getStatusInfoList	ステータス情報情報を取得します。

サンプルプログラム

```
//■販売伝票ステータス一覧
var statusInfoList;
var statusInfoCount;

function init(request) {
//-----
//変数宣言
//-----
var objCon;
var objResult;

//-----
//インスタンス生成 // 説明 1
//-----
objCon = new SalesOrderGetStatusController();

//-----
// 単一項目セット // 説明 2
//-----
objCon.setSalesdocument("0000000023"); //販売管理伝票番号

//-----
// executeBapi実行 // 説明 4
//-----
objResult = objCon.executeBapi();

//-----
// エラー判定
//-----
if(objResult["ERR_CODE"] != null) {
//インスタンス生成時に例外が発生した場合
//エラーメッセージを出力
Debug.browse(objResult);
}

//オブジェクトを取得 // 説明 5
statusInfoList = objCon.getStatusInfoList();
statusInfoCount = statusInfoList.length;
}
}
```

- 販売伝票ステータス一覧照会 API のインスタンス生成(説明1)
SalesOrderGetStatusController クラスのインスタンスを生成します。
引数は不要です。
インスタンスを生成すると、販売伝票ステータス一覧 BAPI (BAPI_SALESORDER_GETSTATUS) の import 型の import、table パラメータに値を設定することができます。
- import の各項目に値を設定(説明2)
import パラメータの項目に値を設定する場合は、set メソッドを使用します。
メソッドについては各 API のAPI リストを参照ください。
- table の各項目に値を設定(説明3)
table パラメータの項目に値を設定する場合は、setTblParam メソッドを使用します。
第1引数のtable名称は固定値になっています。
後述のとおり名称を設定して下さい。
第2引数の項目名称は SAP の各属性の項目名称を設定してください。
table 名称、項目名称は英字の大文字、小文字は厳密に区別されます。
第3引数に値、第四引数にレコード番号を設定します。

SAP table	セットする固有名称
STATUSINFO (ステータス情報)	StatusInfo

- 販売伝票ステータス一覧照会(説明4)
SalesOrderGetStatusController クラスの executeBapi メソッドを実行すると、各項目にセットされた値で SAPより販売伝票ステータス一覧データとして照会されます。
販売伝票ステータス一覧 BAPI (BAPI_SALESORDER_GETSTATUS) の実行後に RETURN パラメータの内容も取得します。
- サンプルプログラムの中で記載されたデータ取得メソッドの詳細説明については、以下を参照してください。(説明5)
 - Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn を取得

SalesOrderGetStatusController クラスの getBapiReturnErrMessage メソッドを使用すると、取得したBapiReturn の Type 情報が正常(デフォルト値: "S" / "")以外の BapiReturn オブジェクトのメッセージ情報を取得することが可能です。

全て正常の場合は null を取得します。

項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。

- Type 情報が正常(デフォルト値: "S" / "")の BapiReturn を取得
SalesOrderGetStatusController クラスの getBapiReturnMessage メソッドを使用すると、取得したBapiReturn の Type 情報が正常(デフォルト値: "S" / "")と判断された Type 情報のメッセージ情報を取得することが可能です。
メッセージが存在しない場合は null を取得します。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。
- 販売管理伝票番号情報を取得
SalesOrderGetStatusController クラスの getSalesdocument メソッドを使用すると、設定した販売管理伝票番号情報を取得することが可能です。
- ステータス情報情報を取得
SalesOrderGetStatusController クラスの getStatusInfoList メソッドを使用すると、取得したステータス情報情報を取得することが可能です。
項目の値を取得するには、KEY に SAP の各属性の項目名称を設定してください。



コラム

サンプルプログラム実行結果では、SAP テスト環境のデータが取得されています。SAP標準では該当データが存在しません。

Tips

本項では、独自でBAPI実行時の制御を行うための実装方法について説明します。
実装方法の内容は以下のメニューから参照してください。

エラー制御

SAP リアルタイム連携 API における BAPI 実行後の成功・失敗の標準の基準は、各 API での BAPI 実行後に RETURN パラメータの TYPE 値を取得して、S（正常終了）または空文字以外が1つでも存在するかどうかがです。
失敗の場合、以下の例外やエラーコードが返却されます。

スクリプト開発モデル エラーコード	JAVAEE 開発モデル 例外	概要
6	SAPAccountingAppException	財務会計業務系アプリケーションエラー
8	SAPPurchaseAppException	経費購買業務系アプリケーションエラー
10	SAPFinancialAppException	管理会計業務系アプリケーションエラー
12	SAPSalesAppException	販売管理業務系アプリケーションエラー

RETURN パラメータの TYPE 値は、一般的に以下になります。

RETURN-TYPE 値	概要
S	正常終了
W	警告
I	情報
E	エラー

たとえば、「警告」の場合も例外やエラーコードが返却されずに登録処理を行いたいという場合もあります。

その場合、各プロパティファイルにてこれを柔軟に設定することができます。

プロパティファイルに、以下のように ignoreErrMsgReturnTypes に W を追加することで、「警告」時に例外やエラーコードが返却されずに処理を終了させることができます。

```
### エラーメッセージを無視する RETURN-TYPE の設定
# カンマ区切りで指定。一般的に、RETURN-TYPE は以下ようになる。
# S : 正常終了 / W : 警告 / I : 情報 / E : エラー
# デフォルト値: S
ignoreErrMsgReturnTypes=S,W
# 空文字を含めるかどうか(true/false)
# デフォルト値: true
ignoreErrMsgReturnTypes.includeBlank=true
```

またこのプロパティの設定は、以下のメソッドに影響します。

- BapiController#getErrorMessage()
ignoreErrMsgReturnTypes の TYPE 値以外のエラーメッセージをすべて取得します。
各業務向けAPI はこのメッセージが存在している場合に、上記の例外やエラーコードを返却します。
- BapiController#getMessage()
ignoreErrMsgReturnTypes の TYPE 値のメッセージをすべて取得します。
- BapiController#isSuccess()
BAPI 実行が正常に行われたか判断します。
正常とは、実行結果に ignoreErrMsgReturnTypes のTYPE 値以外が含まれている場合になります。

トランザクション制御

ConnectionManager を使用することで、プログラム側で SAP への commit/rollback 処理を実行することができます。

これを利用することで、たとえば、複数の伝票をまとめて登録したり、伝票ヘッダと明細が別の BAPI によって登録されるような場合にも同

ートランザクション内で登録処理を行うことができます。

ただし、同一トランザクション内で実行する BAPI 内部に commit/rollback 処理が行われていないことが前提となります。

BAPI 内部の処理をご理解いただいた上で、本機能をご利用いただくことをお勧めいたします。

ConnectionManager についての詳細な情報は、API リストを参照してください。

現在、汎用向け API の BapiController に加え、以下の登録・更新系 API において、ConnectionManager を使用し、プログラム側で SAP への commit/rollback 処理を実行することができます。

使用方法についての詳細な情報は、API リストを参照してください。

BAPI名	概要
AccEmployeePayPostController	会計伝票登録 API
PurchaseOrderPayPostController	購買伝票登録 API
PurchaseOrderPostChangeController	購買伝票更新 API
PurchaseReleaseController	購買伝票承認 API (注)
PurchaseResetReleaseController	購買伝票承認取消 API (注)
MaterialSaveReplicaController	品目マスタ登録/変更/削除 API(注)
GoodsMvtCreateController	入在庫伝票登録 API
InquiryCreateFromData2Controller	引合伝票登録 API
QuotationCreateFromData2Controller	見積伝票登録 API
SalesOrderCreateFromDat2Controller	受注伝票登録 API
BillingDocCreateMultipleController	請求伝票登録 API

注意

(注)マークが付いている API で使用する標準 BAPI では、BAPI 内部でトランザクション制御(commit/rollback 処理)が行われません。

そのため、これらの API に対して ConnectionManager を使用する場合、BAPI 自体をカスタマイズする必要があります。標準 BAPI を使用すると、同一トランザクション内のそこまでの処理がすべて commit または rollback されてしまいます。

以下にConnectionManager を使用した簡単なサンプルを示します。

```

// -----
// 更新系APIをトランザクション制御して実行する手順
// -----

function init() {
  var objCon;
  var errObj;

  // コネクションマネージャーのインスタンス生成
  var connectionManager = new ConnectionManager();

  while(true) {

    // 引数に ConnectionManager を指定し、インスタンス生成
    objCon = new BapiController("xxxxx", connectionManager);

    // BAPI 実行
    errObj = objCon.executeBapi();

    if (checkErr(errObj)) {
      // エラーの場合、ロールバック処理
      connectionManager.rollback();
      break;
    }

    // 引数に ConnectionManager を指定し、インスタンス生成
    objCon = new BapiController("yyyyy", connectionManager);

    // BAPI 実行
    errObj = objCon.executeBapi();

    if (checkErr(errObj)) {
      // エラーの場合、ロールバック処理
      connectionManager.rollback();
      break;
    }

    // すべて成功の場合、コミット処理
    connectionManager.commit();
    break;
  }
}

```

また、更新系APIと参照系APIを同じトランザクションで複数回実行する場合は、コネクションをまず取得し、そのコネクションを使いまわしてください。

下記のサンプルの場合、commit処理が行われるまでに複数回実行されるBAPIの内のいずれかでエラーが発生した場合、これまでの処理がすべてロールバックされます。

```
// -----
// 更新系 -> 参照系 -> 更新系 をトランザクション制御して実行する手順
// -----

var objCon;
var errObj;

// 1. 準備 : SapClientのインスタンスの取得とコネクションマネージャの作成
var sapClient = SapClientManager.getSapClient();
var connectionManager = new ConnectionManager(sapClient);

while(true) {
    // 引数にConnectionManagerを指定し、インスタンス生成
    objCon = new XxxController(connectionManager); // 更新系
    // BAPI 実行
    errObj = objCon.executeBapi();
    if (checkErr(errObj)) {
        // エラーの場合、ロールバック処理
        connectionManager.rollback();
        break;
    }

    // 引数にConnectionManagerを指定し、インスタンス生成
    objCon = new YyyController(sapClient); // 参照系
    // BAPI 実行
    errObj = objCon.executeBapi();
    if (checkErr(errObj)) {
        // エラーの場合、ロールバック処理
        connectionManager.rollback();
        break;
    }

    // 引数にConnectionManagerを指定し、インスタンス生成
    objCon = new ZzzController(connectionManager); // 更新系
    // BAPI 実行
    errObj = objCon.executeBapi();
    if (checkErr(errObj)) {
        // エラーの場合、ロールバック処理
        connectionManager.rollback();
        break;
    }
}

// すべて成功の場合、コミット処理
connectionManager.commit();
break;
}
```

コネクション方式

「IM-ERP Real Connect for Accel Platform セットアップガイド」にも示している通り、SAP との接続には SAP JCo を利用しており、接続方式には以下の 2 つがあります。

- ダイレクトコネクション方式
- プールコネクション方式

パフォーマンスではプールコネクション方式の方が優位となりますが、ダイレクトコネクション方式では SAP への接続時に接続ユーザを変更することができます。

ただし、その変更したユーザは、SAP に存在するユーザでなければなりません。

またそのユーザに適切な権限が付与されている必要があります。

以下に、intra-mart のログインユーザ ID を SAP への接続ユーザ ID として使用する簡単なサンプルを示します。

ただし、認証情報 sap_auth_info.properties に以下のような servertype が sample のダイレクトコネクション用の定義がされているとします。

- sap_auth_info.properties

```
jco.client.client.sample=250
jco.client.user.sample =user001
jco.client.passwd.sample =pass001
jco.client.lang.sample =ja
jco.client.ashost.sample =erpsv01
```

- sample.js

```
function init() {

    // ログインユーザID を取得
    var sessionInfo = AccessSecurityManager.getSessionInfo();
    var userId = sessionInfo.user;

    // そのログインユーザのパスワードを取得
    var accountManager = new AccountManager(sessionInfo.loginGroup);
    var account = accountManager.getAccount(userId);
    var passwd = account.password;

    // SAP クライアントを取得し、ログインユーザID とパスワードを設定
    // ※接続先 IP アドレス等その他の値はプロパティファイルの値が使用されます。
    var sapClient = SapClientManager.getSapClient("sample");
    sapClient.setUser(userId);
    sapClient.setPasswd(passwd);

    // SAP クライアントを任意のコントローラクラスのコンストラクタの引数に渡し、
    // インスタンス生成。これにより、SAP への接続ユーザは上記で設定した
    // ユーザで接続され、そのユーザで BAPI の実行を試みる。
    var objCon = new BankController(sapClient);

}
```

コミット処理の同期化

ConnectionManager#commit() 処理は、デフォルトではSAPへのコミット処理を非同期で実行しています。

SAPへのコミット処理を同期でおこなう場合、ConnectionManager#setCommitAndWait(boolean)を利用してコミット処理の同期、非同期を選択できます。

ConnectionManager#setCommitAndWait(boolean)の仕様に関しては、APIリストを参照してください。

```
function init() {

    //準備 : SapClientのインスタンスの取得とコネクションマネージャの作成
    var sapClient = SapClientManager.getSapClient();
    var connectionManager = new ConnectionManager(sapClient);

    // コミット処理を同期化する
    connectionManager.setCommitAndWait(true);

    // 引数にConnectionManager を指定し、インスタンス生成
    var objCon = new XxxController(connectionManager); // 更新系

    // BAPI 実行
    var errObj = objCon.executeBapi();

    if (checkErr(errObj)) {
        // エラーの場合、ロールバック処理
        connectionManager.rollback();
    } else {
        // コミット
        connectionManager.commit();
    }
}
```