

SAP BusinessObjects Business Intelligence Suite
Document Version: 4.0 Support Package 6 - 2013-09-02

SAP Crystal Reports JavaScript API

目次

1	SAP Crystal Reports JavaScript API について	4
1.1	文書履歴	4
2	SAP Crystal Reports JavaScript API の使用	5
2.1	レポートビューアを作成し、レポートを開く	5
2.2	ビューアコントロールの表示/非表示を切り替える	7
2.3	JavaScript ビューアのルックアンドフィールの変更	8
2.4	イベントリスナ	8
3	API リファレンス[api リファレンス]	9
3.1	SAP.CR.Viewer.ActionListener	9
3.1.1	onEvent	9
3.1.2	removeEvent	10
3.2	SAP.CR.Viewer.CanvasListener	11
3.2.1	onEvent	11
3.2.2	removeEvent	12
3.3	SAP.CR.Parameter	14
3.3.1	addValue	15
3.3.2	setReportName	15
3.4	SAP.CR.Parameter.RangeValue	16
3.4.1	setBeginValue	16
3.4.2	setEndValue	17
3.4.3	setLowerBound	18
3.4.4	setUpperBound	19
3.5	SAP.CR.Viewer	19
3.5.1	addActionListener	20
3.5.2	addCanvasListener	20
3.5.3	batchExecute	20
3.5.4	作成	21
3.5.5	ドリルダウン	22
3.5.6	getInstance	22
3.5.7	refresh	22
3.5.8	removeActionListener	23
3.5.9	removeCanvasListener	23
3.5.10	setDisplayBreadcrumb	24
3.5.11	setDisplayLeftPanel	24
3.5.12	setDisplayStatusBar	24
3.5.13	setDisplayToolbar	25
3.5.14	setHasLogo	25

3.5.15	setHasRefreshButton.	26
3.5.16	setLogo.	26
3.5.17	setPageNumber.	27
3.5.18	setParameters.	27
3.5.19	setPrintMode.	28
3.5.20	setPromptOnRefresh.	29
3.5.21	setReportMode.	29
3.5.22	setReportSource.	30
3.6	SAP.CR.Viewer.ThemeManager.	30
3.6.1	setThemeColor.	30
3.6.2	setThemeFont.	31

1 SAP Crystal Reports JavaScript API について

SAP Crystal Reports JavaScript API を使用すると、クライアント側にコンポーネントをインストールしなくても、組み込み Web アプリケーションで Crystal レポートのコンテンツを表示できます。JavaScript API を使用して、レポートビューアをカスタマイズしたり、Crystal レポートのコンテンツに双方向性を追加することができます。レポートは DHTML で生成されるため、JavaScript を使用する任意の言語で Web アプリケーションを開発できます。

API 機能は次のとおりです。

- マウスクリックイベントとマウスオーバーイベントのイベントリスナ。
- 印刷イベントとエクスポートイベントのアクションリスナ。
- 色とフォントをカスタマイズするためのメソッド。
- ビューアコンポーネントを表示/非表示にするためのメソッド。

JavaScript API は、ほとんどの OpenDocument パラメータをサポートします。これまでに BI platform リポジトリに保存されたレポートにリンクするために OpenDocument URL を使用していた場合は、その機能を失うことなく JavaScript API に移行できます。

1.1 文書履歴

次の表に、文書の重要な変更点をまとめます。

バージョン	日付	説明
SAP BusinessObjects Business Intelligence platform 4.0 Feature Pack 3	2012 年 2 月 20 日	この文書の最初のリリース。

2 SAP Crystal Reports JavaScript API の使用

SAP Crystal Reports JavaScript API は、SAP BusinessObjects Business Intelligence Platform インストールに付属します。BIP サーバ内の `ViewerSeed.js` ファイルを参照することで、この API にアクセスできます。

このガイドで詳細に説明する一般的なワークフロー:

1. `ViewerSeed.js` ファイルの場所を見つけます。通常の BIP インストールの場合、このファイルは `http://localhost:8080/clientapi/CR/ViewerSeed.js` にあります
2. BI platform 認証に使用するログオントークンを取得します。
3. `ViewerSeed.js` ファイルへの参照を Web アプリケーションに追加します。
4. CR Embedded ビューアを Web アプリケーションに追加します。
5. レポートソースを設定します。

i 注記

Web サーバにデプロイするのではなく、ローカルコンピュータでアプリケーションを開発している場合は、Adobe Flash Player が Flex のプロンプトダイアログを表示しないことがあります。この問題を解決するには、アプリケーションを Web サーバにデプロイするか、Flash Player のグローバルセキュリティ設定で、アプリケーションを信頼できるコンテンツであると指定してください。

2.1 レポートビューアを作成し、レポートを開く

1. `head`、`style`、および `body` の開始タグと終了タグを設定します。
2. `head` タグ内にスクリプトタグを追加し、そこで `ViewerSeed.js` を参照します。

```
<script src="http://localhost:8080/clientapi/CR/ViewerSeed.js">
```

3. ログオントークンを保持するトークン変数を作成し、それを `null` に設定します。

```
var token = null;
```

4. `init` という名前の関数を作成し、その中で、`token` の値をログオントークンに設定し、新しい JavaScript ビューアインスタンスを作成します。

ログオントークンの生成については、Business Intelligence Platform RESTful Web Services Developer Guide を参照してください。

```
init(){
    token = "logonToken";
    SAP.CR.Viewer.create("viewerName", "container", onViewerInit);
}
```

5. `onViewerInit` という名前の関数を追加し、その中でレポートソースを設定します。

```
function onViewerInit(viewerInstance){
    viewerInstance.setReportSource('reportID', token);
}
```

reportID は、レポートに割り当てられる数値コードです。

レポート ID は、プログラムまたはセントラル管理コンソール(CMC)で調べることができます。CMC でレポートを右クリックすると、レポート ID がレポートプロパティと共に表示されます。詳細については、SAP Crystal Reports RESTful Web サービス API を参照してください。

6. body タグ内で、init 関数と同等の onload イベントハンドラーを設定します。

```
<body onload="init()">
```

ページが完全にロードされると、このイベントハンドラーが初期化関数を呼び出し、本文内にコード化されたテキストやボタンがレポートビューアの前にロードされます。

7. style タグ内で、JavaScript ビューアの位置とサイズを定義する新しいクラスを作成します。次のコードは、ビューアに絶対位置を指定しています。ビューアの幅と高さは、表示されるページの幅と高さの 75% になります。ページ内のビューアの位置は、left プロパティと top プロパティによって指定されます。

```
.viewerStyle
{
  position : absolute;
  left : 12.5%;
  top : 20.5%;
  width : 75%;
  height : 75%;
}
```

8. body タグ内に、ビューアのスタイルを設定するためのコードを追加します。

```
<div id="container" class="viewerStyle"></div>
```



例

レポートビューアを作成し、レポートを開く

次の例は、crystalViewer という名前のレポートビューアでレポート ID 1234 のレポートを開きます。init 関数の token 変数に設定されている値は例です。実際に使用する値は、ユーザが生成する必要があります。ビューアには、絶対位置が指定されています。ビューアの幅と高さは、表示されるページの幅と高さの 75% になります。ページ内のビューアの位置は、left プロパティと top プロパティによって指定されます。onViewerFailure 関数はオプションです。

```
<head>

<script src="http://computername/clientapi/CR/ViewerSeed.js">

<script>
  var token = null;

  function init() {
    token = "COMMANDCOM-LCM:6400@{3&2=5328,U3&p=40676
      .8926203819,Y7&4F=12,U3&63=secEnterprise
      ,0P&66=60,03&68=secEnterprise:Administrator
      ,0P&qe=100,U3&vz=IVD21LbMCB0eRiI4at
      z9sNL18Ux5anRBdYB9fFv5NrY,UP}";
    SAP.CR.Viewer.create("crystalViewer", 'viewerContainer1', onViewerInit,
onViewerFailure);
  }

  function onViewerInit() {
    viewerInstance.setReportSource('1234', token);
  }

  function onViewerFailure(instance, error) {
    alert(error);
  }
}
```



```

    }

    </script>
</head>
<style>
    .viewerStyle
    {
        position : absolute;
        left : 12.5%;
        top : 20.5%;
        width : 75%;
        height : 75%;
    }
</style>
<body>
    <div id="viewerContainer1" class = "viewerStyle"></div>
</body>

```

2.2 ビューアコントロールの表示/非表示を切り替える

JavaScript API を使用して、JavaScript ビューアの以下のブラウザコントロールの表示/非表示を切り替えることができます。

- 左のパネル (検索、パラメータの設定、および[グループツリー](#)の表示に使用)。
- ツールバー。
- ステータスバー。
- ページナビゲーションバー。

表示を切り替えるには、コントロールを `true` または `false` に設定します。デフォルトでは、すべてのブラウザコントロールが `true` に設定されています。

例

次のコードは、ステータスバーを除くすべてのコントロールバーとパネルを表示します。

```

var viewerInstance = SAP.CR.Viewer.getInstance("crystalViewer");

viewerInstance.setDisplayToolbar(true);
viewerInstance.setDisplayLeftPanel(true);
viewerInstance.setDisplayStatusbar(false);
viewerInstance.setDisplayBreadcrumb(true);

```

2.3 JavaScript ビューアのルックアンドフィールの変更

JavaScript API を使用すると、JavaScript ビューアで使用される色とフォントをカスタマイズできます。

例

次の例は、ビューアの色を淡い紫色のグラデーションに変更し、フォントを Times に変更します。setThemeColor メソッドの最初のパラメータは、色の 16 進表記です。2 番目のパラメータの boolean 値として false ではなく true を使用すると、ビューアがグラデーションではなく単色になります。

```
SAP.CR.Viewer.ThemeManager.setThemeColor("#CDB7F9", false);
SAP.CR.Viewer.ThemeManager.setThemeFont("times");
```

2.4 イベントリスナ

JavaScript API は、キャンバスイベントリスナとアクションイベントリスナをサポートします。イベントリスナは、特定のイベントが発生したことを検出し、応答をトリガーします。キャンバスイベントは、レポートキャンバス上で発生するマウスイベント（クリックとマウスオーバー）です。アクションリスナは、ボタンのクリックや一覧の項目の選択といったアクションイベントを処理します。

例

次のコードは、マウスオーバーイベントに反応してレポートオブジェクトを緑色で強調表示します。カーソルがレポートオブジェクトから離れると、背景色プロパティがクリアされます。つまり、オブジェクトのデフォルトの背景色は復元されません。

```
var canvasListener = SAP.CR.Viewer.CanvasListener();
var viewerInstance = SAP.CR.Viewer.getInstance("crystalViewer");

canvasListener.onEvent(SAP.CR.Viewer.CanvasEvents.REPORT_ELEMENT_MOUSE_ENTER, function(args) {
    args.target.style.backgroundColor="green";
});

canvasListener.onEvent(SAP.CR.Viewer.CanvasEvents.REPORT_ELEMENT_MOUSE_LEAVE, function(args) {
    args.target.style.backgroundColor=" ";
});

viewerInstance.addCanvasListener(canvasListener);
```

関連リンク

[SAP.CR.Viewer.CanvasListener](#) [ページ 11]

[SAP.CR.Viewer.ActionListener](#) [ページ 9]

3 API リファレンス[api リファレンス]

このセクションでは、JavaScript API に含まれるクラスとメソッドのリファレンスを提供します。

3.1 SAP.CR.Viewer.ActionListener

`ActionListener` クラスを使用すると、ツールバーのほとんどの項目に対して、ビューアアクションイベントのカスタム機能を追加できます。

構文

```
var actionListener = new SAP.CR.Viewer.ActionListener();
```

3.1.1 onEvent

`onEvent` メソッドは、イベントのアクションリスナを登録します。同じイベントに複数のリスナを登録できます。

構文

```
var actionListener = new SAP.CR.Viewer.ActionListener();  
actionListener.onEvent(eventName, listener);
```

パラメータ

- `eventName` - イベントの名前。
指定できる値:

値	説明
<code>SAP.CR.Viewer.ActionEvents .EXPORT</code>	ユーザが [エクスポート] ボタンをクリックした。

値	説明
SAP.CR.Viewer.ActionEvents.PRINT	ユーザが [印刷] ボタンをクリックした。
SAP.CR.Viewer.ActionEvents.DRILL	ユーザがグループツリー、ブレードクラム、コンテンツ、またはチャートからレポートにドリルダウンした。
SAP.CR.Viewer.ActionEvents.GROUP_TREE_NAVIGATE	表示中のページを変更するグループツリーノードをユーザがクリックした。
SAP.CR.Viewer.ActionEvents.PROMPT	レポートがデータベースまたはパラメータの入力を要求した。
SAP.CR.Viewer.ActionEvents.ERROR	ビューアでエラーが発生した。

- listener - [function(arg)] イベントが発生したときの動作を決定するユーザ定義のカスタム関数。パラメータ arg には、JavaScript キーワード名以外の任意の名前を指定できます。これは、レポートビューアのハンドラーによって作成され、関数に渡されます。

3.1.2 removeEvent

removeEvent メソッドは、ActionListener オブジェクトからリスナを削除します。このメソッドを使用して、関連するイベント名を持つ特定のリスナを削除したり、関連するイベント名を持つすべてのイベントを削除することができます。

構文

```
var actionListener = new SAP.CR.Viewer.ActionListener();
actionListener.removeEvent(eventName, listener);
```

パラメータ

- eventName - 削除するイベントの名前。
指定できる値:

値	説明
SAP.CR.Viewer.ActionEvents.EXPORT	ユーザが [エクスポート] ボタンをクリックした。

値	説明
SAP.CR.Viewer.ActionEvents.PRINT	ユーザが [印刷] ボタンをクリックした。
SAP.CR.Viewer.ActionEvents.DRILL	ユーザがグループツリー、ブレードクラム、コンテンツ、またはチャートからレポートにドリルダウンした。
SAP.CR.Viewer.ActionEvents.GROUP_TREE_NAVIGATE	表示中のページを変更するグループツリーノードをユーザがクリックした。
SAP.CR.Viewer.ActionEvents.PROMPT	レポートがデータベースまたはパラメータの入力を要求した。
SAP.CR.Viewer.ActionEvents.ERROR	ビューアでエラーが発生した。

- listener - [function(arg)] 削除する特定のリスナ。イベントリスナの作成時にユーザによって定義されるカスタム関数。

3.2 SAP.CR.Viewer.CanvasListener

CanvasListener クラスを使用すると、レポートページで発生するマウスイベントにカスタム機能を追加できます。

構文

```
var canvasListener = new SAP.CR.Viewer.CanvasListener();
```

3.2.1 onEvent

onEvent メソッドは、イベントのキャンバスリスナを登録します。同じイベントに複数のリスナを登録できます。

構文

```
var canvasListener = new SAP.CR.Viewer.CanvasListener();
canvasListener.onEvent(eventName, listener);
```

パラメータ

- eventName - イベントの名前。

指定できる値:

値	説明
SAP.CR.Viewer.CanvasEvents.REPORT_ELEMENT_MOUSE_ENTER	カーソルがレポート要素に入った。
SAP.CR.Viewer.CanvasEvents.REPORT_ELEMENT_MOUSE_LEAVE	カーソルがレポート要素から離れた。
SAP.CR.Viewer.CanvasEvents.REPORT_ELEMENT_CLICK	ユーザがレポート要素をクリックした。
SAP.CR.Viewer.CanvasEvents.REPORT_ELEMENT_RIGHT_CLICK	ユーザがレポート要素を右クリックした。
SAP.CR.Viewer.CanvasEvents.REPORT_CANVAS_MOUSE_ENTER	カーソルがレポートキャンバスに入った。
SAP.CR.Viewer.CanvasEvents.REPORT_CANVAS_MOUSE_LEAVE	カーソルがレポートキャンバスから離れた。
SAP.CR.Viewer.CanvasEvents.REPORT_CANVAS_CLICK	ユーザがレポートキャンバスをクリックした。
SAP.CR.Viewer.CanvasEvents.REPORT_CANVAS_RIGHT_CLICK	ユーザがレポートキャンバスを右クリックした。

- listener - [function(arg)] イベントが発生したときの動作を決定するユーザ定義のカスタム関数。パラメータ arg には、JavaScript キーワード名以外の任意の名前を指定できます。これは、レポートビューアのハンドラーによって作成され、関数に渡されます。

3.2.2 removeEvent

`removeEvent` メソッドは、`CanvasListener` オブジェクトからリスナを削除します。このメソッドを使用して、関連するイベント名を持つ特定のリスナを削除したり、関連するイベント名を持つすべてのイベントを削除することができます。

構文

```
var canvasListener = new SAP.CR.Viewer.CanvasListener();
canvasListener.removeEvent(eventName, listener);
```

パラメータ

- eventName - 削除するイベントの名前。
指定できる値:

値	説明
SAP.CR.Viewer.CanvasEvents .REPORT_ELEMENT_MOUSE_ENTER	カーソルがレポート要素に入った。
SAP.CR.Viewer.CanvasEvents .REPORT_ELEMENT_MOUSE_LEAVE	カーソルがレポート要素から離れた。
SAP.CR.Viewer.CanvasEvents .REPORT_ELEMENT_CLICK	ユーザがレポート要素をクリックした。
SAP.CR.Viewer.CanvasEvents .REPORT_ELEMENT_RIGHT_CLICK	ユーザがレポート要素を右クリックした。
SAP.CR.Viewer.CanvasEvents .REPORT_CANVAS_MOUSE_ENTER	カーソルがレポートキャンバスに入った。
SAP.CR.Viewer.CanvasEvents .REPORT_CANVAS_MOUSE_LEAVE	カーソルがレポートキャンバスから離れた。
SAP.CR.Viewer.CanvasEvents .REPORT_CANVAS_CLICK	ユーザがレポートキャンバスをクリックした。
SAP.CR.Viewer.CanvasEvents .REPORT_CANVAS_RIGHT_CLICK	ユーザがレポートキャンバスを右クリックした。

- listener - [function(arg)] 削除する特定のリスナ。イベントリスナの作成時にユーザによって定義されるカスタム関数。

3.3 SAP.CR.Parameter

SAP.CR.Parameter クラスを使用すると、パラメータを作成してビューインスタンスに追加できます。このクラスを使用して、サブレポートのパラメータ値を設定することもできます。

構文

```
var param = new SAP.CR.Parameter(parameterName,parameterType);
```

パラメータ

- parameterName - [String] レポートのパラメータの名前。多くの場合、この値はレポートであらかじめ定義されています。
- parameterType - パラメータの型。

指定できる値:

値	説明
SAP.CR.Parameter .DataTypes.BOOLEAN	Boolean 値 (true または false)。
SAP.CR.Parameter .DataTypes.CURRENCY	数値。
SAP.CR.Parameter .DataTypes.DATE_TIME	JavaScript Date 関数を使用して指定された日時。
SAP.CR.Parameter .DataTypes.DATE	JavaScript Date 関数を使用して指定された日付。
SAP.CR.Parameter .DataTypes.TIME	JavaScript Date 関数を使用して指定された時刻。
SAP.CR.Parameter .DataTypes.NUMBER	数値。
SAP.CR.Parameter .DataTypes.STRING	文字列値。

3.3.1 addValue

パラメータの値を設定します。

構文

```
var param = new SAP.CR.Viewer.Parameter(parameterName,parameterType);
param.addValue(val);
```

パラメータ

- `val` - パラメータの値。型は、`parameterType` によって決定されます。離散値または範囲値にすることができます。

3.3.2 setReportName

サブレポートのパラメータを設定します。`setReportName` メソッドを使用しない場合、これらのパラメータは自動的にメインレポートに設定されます。

構文

```
var param = new SAP.CR.Viewer.Parameter(parameterName,parameterType);
param.setReportName(subReportName);
```

パラメータ

- `subReportName` - [String] パラメータを設定するサブレポートの名前。

例

次の例は、`SAP.CR.DataTypes.NUMBER` 型の新しいパラメータを作成し、それを `subReport1` という名前のサブレポートに追加します。

```
var instance = SAP.CR.Viewer.getInstance("crystalViewer");

var param = new
SAP.CR.Parameter("subNumberParameter",SAP.CR.Parameter.DataTypes.NUMBER);
param.setReportName("subReport1");
```



```
param.addValue(10);
instance.setParameters([param]);
```

3.4 SAP.CR.Parameter.RangeValue

SAP.CR.Parameter.Range クラスは、パラメータの値の範囲を指定するために使用されます。

SAP.CR.Parameter.Range オブジェクトは、addValue メソッドを使用して SAP.CR.Parameter オブジェクトに追加されます。

構文

```
var range = new SAP.CR.Parameter.RangeValue();
```

3.4.1 setBeginValue

値の範囲の最初を設定します。

構文

```
var range = new SAP.CR.Parameter.RangeValue();
range.setBeginValue(val);
```

パラメータ

- val - 値の範囲の下限值。
val に指定できる型:

値	説明
SAP.CR.Parameter. .DataTypes.BOOLEAN	Boolean 値 (true または false)。

値	説明
SAP.CR.Parameter .DataTypes.CURRENCY	数値。
SAP.CR.Parameter .DataTypes.DATE_TIME	JavaScript Date 関数を使用して指定された日時。
SAP.CR.Parameter .DataTypes.DATE	JavaScript Date 関数を使用して指定された日付。
SAP.CR.Parameter .DataTypes.TIME	JavaScript Date 関数を使用して指定された時刻。
SAP.CR.Parameter .DataTypes.NUMBER	数値。
SAP.CR.Parameter .DataTypes.STRING	文字列値。

この型は、範囲値の追加先のパラメータの型と同じでなければなりません。

3.4.2 setEndValue

値の範囲の最後を設定します。

構文

```
var range = new SAP.CR.Parameter.RangeValue();
range.setEndValue(val);
```

パラメータ

- val - 値の範囲の上限値。
val に指定できる型:

値	説明
SAP.CR.Parameter .DataTypes.BOOLEAN	Boolean 値 (true または false)。

値	説明
<code>SAP.CR.Parameter .DataTypes.CURRENCY</code>	数値。
<code>SAP.CR.Parameter .DataTypes.DATE_TIME</code>	JavaScript Date 関数を使用して指定された日時。
<code>SAP.CR.Parameter .DataTypes.DATE</code>	JavaScript Date 関数を使用して指定された日付。
<code>SAP.CR.Parameter .DataTypes.TIME</code>	JavaScript Date 関数を使用して指定された時刻。
<code>SAP.CR.Parameter .DataTypes.NUMBER</code>	数値。
<code>SAP.CR.Parameter .DataTypes.STRING</code>	文字列値。

この型は、範囲値の追加先のパラメータの型と同じでなければなりません。

3.4.3 setLowerBound

範囲パラメータの下限値のタイプを設定します。

構文

```
var range = new SAP.CR.Parameter.RangeValue();
range.setLowerBound(boundType);
```

パラメータ

- boundType - 下限値のタイプを決定します。
指定できる値:

値	説明
<code>SAP.CR.Parameter.Parameter .RangeBoundTypes.INCLUSIVE</code>	値の範囲に開始値が含まれます。

値	説明
SAP.CR.Parameter.Parameter .RangeBoundTypes.EXCLUSIVE	値の範囲に開始値が含まれません。
SAP.CR.Parameter.Parameter .RangeBoundTypes.UNBOUNDED	範囲の開始を指定しません。

3.4.4 setUpperBound

範囲パラメータの上限値のタイプを設定します。

構文

```
var range = new SAP.CR.Parameter.RangeValue();
range.setUpperBound(boundType);
```

パラメータ

- boundType - 上限値のタイプを決定します。
指定できる値:

値	説明
SAP.CR.Parameter.Parameter .RangeBoundTypes.INCLUSIVE	値の範囲に終了値が含まれます。
SAP.CR.Parameter.Parameter .RangeBoundTypes.EXCLUSIVE	値の範囲に終了値が含まれません。
SAP.CR.Parameter.Parameter .RangeBoundTypes.UNBOUNDED	範囲の終了を指定しません。

3.5 SAP.CR.Viewer

SAP.CR.Viewer クラスには、レポートビューアのインスタンスを作成し、それをカスタマイズしたり、機能を追加するためのメソッドが含まれます。

3.5.1 addActionListener

ビューアインスタンスにアクションイベントリスナを追加します。

構文

```
var viewerInstance = SAP.CR.Viewer.getInstance(viewerName);
viewerInstance.addActionListener(aListener);
```

パラメータ

- `aListener` - 追加するアクションリスナの名前。SAP.CR.ActionListener のインスタンスでなければなりません。

3.5.2 addCanvasListener

ビューアインスタンスにキャンバスイベントリスナを追加します。

構文

```
var viewerInstance = SAP.CR.Viewer.getInstance(viewerName);
viewerInstance.addCanvasListener(cListener);
```

パラメータ

- `cListener` - 追加するキャンバスリスナの名前。SAP.CR.CanvasListener のインスタンスでなければなりません。

3.5.3 batchExecute

`batchExecute` メソッドは、API 呼び出しが同期的に実行されるようにします。

構文

```
var viewerInstance = SAP.CR.Viewer.getInstance(viewerName);
viewerInstance.batchExecute(function());
```

パラメータ

- `function` - [function()] 実行する API 呼び出しを実行する順序で格納するユーザ定義の関数。

例

次の例は、1つのグループパスをドリルダウンしてから、ページ番号を 2 に設定します。`batchExecute` メソッドを使用しない場合は、2つの非同期呼び出しが同時にサーバに送信されます。このため、この 2つの API 呼び出しの一方だけが行われます。

```
var viewerInstance = SAP.CR.Viewer.getInstance("crystalViewer");

viewerInstance.batchExecute(function() {
    viewerInstance.drillDown(["0"]);
    viewerInstance.setPageNumber(2);
});
```

3.5.4 作成

CR Embedded ビューアインスタンスを作成します。

構文

```
SAP.CR.Viewer.create(viewerName, containerID, initCB, failCB)
```

パラメータ

- `containerID` - [String] ビューアを保持する要素の ID。
- `viewerName` - [String] ビューアの名前。
- `initCB` - [function(instance)] ビューアインスタンスを初期化する関数。この関数は、メソッド呼び出し内で定義されません。パラメータ `instance` はビューアインスタンスです。
- `failCB` - [function(instance,error)] 何かが失敗した場合に実行される関数。この関数は、メソッド呼び出し内で定義されません。これはオプションパラメータです。パラメータ `instance` はビューアインスタンスです。

3.5.5 ドリルダウン

グループにドリルダウンします。

構文

```
var viewerInstance = SAP.CR.Viewer.getInstance(viewerName);
viewerInstance.drilldown([groupPath]);
```

パラメータ

- groupPath - ドリルダウンしたパスを表示する整数の配列。

3.5.6 getInstance

ビューア名に基づいてビューアインスタンスを取得します。

構文

```
var viewerInstance = SAP.CR.Viewer.getInstance(viewerName);
```

パラメータ

- viewerName - [String] 取得するビューアの名前。

3.5.7 refresh

JavaScript ビューアに表示されているレポートを最新表示します。このメソッドは、表示されているレポートのデータを最新表示します。必要に応じて、パラメータやログオン情報の入力を要求します。

構文

```
var viewerInstance = SAP.CR.Viewer.getInstance(viewerName);
viewerInstance.refresh();
```

3.5.8 removeActionListener

ビューアインスタンスからアクションイベントリスナを削除します。

構文

```
var viewerInstance = SAP.CR.Viewer.getInstance(viewerName);
viewerInstance.removeActionListener(aListener);
```

パラメータ

- aListener - 削除するアクションリスナの名前。

3.5.9 removeCanvasListener

ビューアインスタンスからキャンバスイベントリスナを削除します。

構文

```
var instance = SAP.CR.Viewer.getInstance(viewerName);
instance.removeCanvasListener(cListener);
```

パラメータ

- cListener - 削除するキャンバスリスナの名前。

3.5.10 setDisplayBreadcrumb

ブレッdkラムの表示/非表示を切り替えます。

構文

```
var viewerInstance = SAP.CR.Viewer.getInstance(viewerName);
viewerInstance.setDisplayBreadcrumb(isBreadcrumb);
```

パラメータ

- `isBreadcrumb` - [Boolean] ブレッdkラムを表示する場合は `true`、ブレッdkラム非表示にする場合は `false`。

3.5.11 setDisplayLeftPanel

検索パネルとして使用される左パネルの表示/非表示を切り替えます。これにより、[グループツリー](#)を表示したり、パラメータを表示および編集することができます。

構文

```
var viewerInstance = SAP.CR.Viewer.getInstance(viewerName);
viewerInstance.setDisplayLeftPanel(isLeftPanel);
```

パラメータ

- `isLeftPanel` - [Boolean] パネルを表示する場合は `true`、パネルを非表示にする場合は `false` に設定します。

3.5.12 setDisplayStatusBar

レポートビューアの下端に表示されるステータスバーの表示/非表示を切り替えます。

構文

```
var viewerInstance = SAP.CR.Viewer.getInstance(viewerName);
viewerInstance.setDisplayStatusBar(isStatusBar);
```

パラメータ

- `isStatusBar` - [Boolean] ステータスバーを表示する場合は `true`、ステータスバーを非表示にする場合は `false`。

3.5.13 setDisplayToolbar

ツールバーの表示/非表示を切り替えます。

構文

```
var viewerInstance = SAP.CR.Viewer.getInstance(viewerName);
viewerInstance.setDisplayToolbar(isToolbar);
```

パラメータ

- `isToolbar` - [Boolean] ツールバーを表示する場合は `true`、ツールバー非表示にする場合は `false`。

3.5.14 setHasLogo

レポートビューアの上右隅にあるロゴの表示/非表示を切り替えます。デフォルトでは、ロゴが表示されます。

i 注記

ビューアを初期化する前に呼び出す必要があります。そうしないと、例外がスローされます。

構文

```
var viewerInstance = SAP.CR.Viewer.getInstance(viewerInstance);
viewerInstance.setHasLogo(isLogo);
```

パラメータ

- `isLogo` - [Boolean] ログを表示する場合は `true`、非表示にする場合は `false` に設定します。

3.5.15 setHasRefreshButton

レポートビューアのツールバーの最新表示ボタンを表示または非表示にします。デフォルトでは、最新表示ボタンは非表示に設定されています。

i 注記

ビューアを初期化する前に呼び出す必要があります。そうしないと、例外がスローされます。

構文

```
var viewerInstance = SAP.CR.Viewer.getInstance(viewerName);
viewerInstance.setHasRefreshButton(isRefreshButton);
```

パラメータ

- `isRefreshButton` - [Boolean] 最新表示ボタンを表示する場合は `true`、非表示にする場合は `false` に設定します。

3.5.16 setLogo

画像を右上隅に表示するロゴとして使用するように設定します。ロゴに URL をリンクしたり、ロゴの上にマウスを置いたときに表示されるツールヒントを設定することができます。

i 注記

ビューアを初期化する前に呼び出す必要があります。そうしないと、例外がスローされます。

i 注記

画像のサイズは、100 x 125 ピクセル以下にする必要があります。

構文

```
var viewerInstance = SAP.CR.Viewer.getInstance(viewerName);
viewerInstance.setLogo(url, link, tooltip);
```

パラメータ

- `url` - [String] ログとして使用する画像の相対パスまたは絶対パス。
- `link` - [String] ユーザがロゴをクリックすると開く URL。
- `tooltip` - [String] ログの上にマウスを置くと表示されるツールヒント。ツールヒントに英文字以外の文字が含まれる場合は、それをエンコードする必要があります。

3.5.17 setPageNumber

表示されるレポートページのページ番号を設定します。

構文

```
var viewerInstance = SAP.CR.Viewer.getInstance(viewerName);
viewerInstance.setPageNumber(pageNumber);
```

パラメータ

- `pageNumber` - [number] 正の数値。レポートのページ数より大きい場合、ページ番号には、レポートの最後のページが設定されます。

3.5.18 setParameters

レポートパラメータを設定します。1 回のメソッド呼び出しで複数のパラメータを設定できます。

構文

```
var viewerInstance = SAP.CR.Viewer.getInstance(viewerName);
viewerInstance.setParameters([param]);
```

パラメータ

- param - ビューアに設定するパラメータの配列。

3.5.19 setPrintMode

レポートビューアの印刷モードを ACTIVEX または PDF に設定します。

i 注記

ビューアを初期化する前に呼び出す必要があります。そうしないと、例外がスローされます。

構文

```
var viewerInstance = SAP.CR.Viewer.getInstance(viewerName);
viewerInstance.setReportMode(mode);
```

パラメータ

- mode - 印刷モード。
指定できる値:

値	説明
SAP.CR.PrintMode.ACTIVEX	ActiveX プリントを使用して出力します。Internet Explorer でのみ動作します。
SAP.CR.PrintMode.PDF	PDF に出力します。

3.5.20 setPromptOnRefresh

ビューアが最新表示後に新しいパラメータ値の入力を要求するかどうかを設定します。デフォルトでは、ビューアは新しいパラメータ値の入力を要求します。

i 注記

ビューアを初期化する前に呼び出す必要があります。そうしないと、例外がスローされます。

構文

```
var viewerInstance = SAP.CR.Viewer.getInstance(viewerName);
viewerInstance.setPromptOnRefresh(isPrompt);
```

パラメータ

- `isPrompt` - [Boolean] 新しいパラメータの入力を要求する場合は `true` に設定し、既存のパラメータ値を使用する場合は `false` に設定します。

3.5.21 setReportMode

レポートビューア内のレポートページの背景色と配置を設定します。

i 注記

ビューアを初期化する前に呼び出す必要があります。そうしないと、例外がスローされます。

構文

```
var viewerInstance = SAP.CR.Viewer.getInstance(viewerName);
viewerInstance.setReportMode(mode);
```

パラメータ

- `mode` - レポートビューア内のレポートページの色と配置を設定します。
指定できる値:

値	説明
SAP.CR.Viewer .ReportMode.PRINT	レポートビューアの背景を灰色に設定します。
SAP.CR.Viewer .ReportMode.WEB	レポートビューアの背景を白色に設定します。レポートページは、ビューアの中央に揃えられません。

3.5.22 setReportSource

数値で示されるオブジェクト ID と Enterprise セッションに基づいて、レポートソースを設定します。必要に応じて、ドキュメントのロケールも設定できます。

構文

```
var viewerInstance = SAP.CR.Viewer.getInstance(viewerName);
viewerInstance.setReportSource(infoObjectID, token, locale);
```

パラメータ

- `infoObjectID` - [String] Info View/CMC 内のレポートの ID。
- `token` - [String] ログオントークン。数分間アイドル状態が続くとトークンが期限切れになり、トークンを再生成する必要があります。ログオントークンの生成方法については、Business Intelligence Platform RESTful Web Services Developer Guide を参照してください。
- `locale` - [String] ドキュメントのロケール。

3.6 SAP.CR.Viewer.ThemeManager

`ThemeManager` クラスを使用すると、ビューアで使用する色とフォントスタイルを設定できます。

3.6.1 setThemeColor

ビューアの色を設定します。

構文

```
SAP.CR.Viewer.ThemeManager.setThemeColor(hexColor, isSolid);
```

パラメータ

- `hexColor` - [String] 使用する色の 16 進表記。値には必ず先頭に # 記号を付けます。
- `isSolid` - [Boolean] 単色にする場合は `true`、グラデーションカラーにする場合は `false` に設定します。

3.6.2 setThemeFont

JavaScript ビューア UI で使用するファミリフォント名を設定します。これは、表示するレポートに使用されるフォントに影響しません。

構文

```
SAP.CR.Viewer.ThemeManager.setThemeFont(font);
```

パラメータ

- `font` - [String] ビューアで使用するフォント。ブラウザでサポートされるすべてのフォントを使用できます。



www.sap.com/contactsap

© 2013 SAP AG or an SAP affiliate company. All rights reserved.

本書のいかなる部分も SAP AG の明示的許可なしに、いかなる形式、目的を問わず、複写、または送信することを禁じます。本書に記載された情報は、予告なしに変更されることがあります。

SAP AG がライセンス、またはその頒布業者が頒布するソフトウェア製品には、他のソフトウェア会社の専有ソフトウェアコンポーネントが含まれています。製品仕様は、国ごとに変わる場合があります。

これらの文書は SAP AG およびその関連会社（「SAP グループ」）が情報提供のためにのみ提供するもので、いかなる種類の表明および保証を伴うものではなく、SAP グループは文書に関する錯誤又は脱漏等に対する責任を負うものではありません。SAP グループの製品およびサービスに対する唯一の保証は、当該製品およびサービスに伴う明示的保証がある場合に、これに規定されたものに限られます。本書のいかなる記述も、追加の保証となるものではありません。

SAP、および本書で言及されるその他 SAP の製品およびサービス、ならびにそれらのロゴは、ドイツおよびその他諸国における SAP AG の商標または登録商標です。

商標に関する情報および表示の詳細については、<http://www.sap.com/corporate-en/legal/copyright/index.epx> をご覧ください。