



Embed BA into Web Applications

Copyright Page

This document supports Pentaho Business Analytics Suite 5.3 GA and Pentaho Data Integration 5.3 GA, documentation revision January 15th, 2015, copyright © 2015 Pentaho Corporation. No part may be reprinted without written permission from Pentaho Corporation. All trademarks are the property of their respective owners.

Help and Support Resources

To view the most up-to-date help content, visit <https://help.pentaho.com>.

If you do not find answers to your questions here, please contact your Pentaho technical support representative.

Support-related questions should be submitted through the Pentaho Customer Support Portal at <http://support.pentaho.com>.

For information about how to purchase support or enable an additional named support contact, please contact your sales representative, or send an email to sales@pentaho.com.

For information about instructor-led training, visit <http://www.pentaho.com/training>.

Liability Limits and Warranty Disclaimer

The author(s) of this document have used their best efforts in preparing the content and the programs contained in it. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, express or implied, with regard to these programs or the documentation contained in this book.

The author(s) and Pentaho shall not be liable in the event of incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of the programs, associated instructions, and/or claims.

Trademarks

The trademarks, logos, and service marks ("Marks") displayed on this website are the property of Pentaho Corporation or third party owners of such Marks. You are not permitted to use, copy, or imitate the Mark, in whole or in part, without the prior written consent of Pentaho Corporation or such third party. Trademarks of Pentaho Corporation include, but are not limited, to "Pentaho", its products, services and the Pentaho logo.

Trademarked names may appear throughout this website. Rather than list the names and entities that own the trademarks or inserting a trademark symbol with each mention of the trademarked name, Pentaho Corporation states that it is using the names for editorial purposes only and to the benefit of the trademark owner, with no intention of infringing upon that trademark.

Third-Party Open Source Software

For a listing of open source software used by each Pentaho component, navigate to the folder that contains the Pentaho component. Within that folder, locate a folder named licenses. The licenses folder contains HTML files that list the names of open source software, their licenses, and required attributions.

Contact Us

Global Headquarters Pentaho Corporation Citadel International, Suite 460

5950 Hazeltine National Drive Orlando, FL 32822

Phone: +1 407 812-OPEN (6736)

Fax: +1 407 517-4575

<http://www.pentaho.com>

Sales Inquiries: sales@pentaho.com

Introduction

This document is intended for **developers** with HTML and JavaScript experience. It explains ways to integrate BA Server functionality and content into your own web application.

Prerequisites

This document assumes you have a correctly installed and configured Pentaho BA Server EE running locally on your machine. It also assumes you know how to start and stop the server if necessary. Please refer to the following sections for help with installing, configuring and administering a Pentaho BA server:

- [Install Pentaho Software](#)
- [Configure Pentaho Systems](#)
- [Fine-Tune Pentaho Systems](#)

Integration Samples

This document comes with a set of sample web pages that show ways to integrate BA Server functionality and content into your web application. This document describes the techniques used by the various samples, so we recommend you deploy the samples on your BA Server when following this document.

Obtaining the Samples

The samples are packaged into a file named **pentaho-sdk<version>.zip** accessible for download in your [Customer Support Portal](#). The **readme** file contains the information for deployment.

Deploying the Samples

To deploy the samples to your Pentaho server, follow these steps:

- Stop the BA Server.
- Run SDK > Installer.bat (or the corresponding script for your OS) to uncompress these two folders to your selected location: the **integration-examples** folder and **platform-plugins**.
- Copy the **integration-examples** directory to your `biserver-ee/tomcat/webapps` directory.
- Open the **platform-plugins** directory and copy these two folders: **example-visualization** and **oem-tools**.
- Paste them into the `.../biserver-ee/pentaho-solutions/system` directory.
- Restart the BA Server.

The samples are deployed and can be accessed by opening a web browser and navigating to <http://localhost:8080/integration-examples>.

If you want to access the OEM tools, browse to <http://localhost:8080/pentaho/content...web/index.html>.

If you want to access the **example-visualization**, look under the Charts menu in Analyzer. It will be listed as **Example KPI**.

Integrating BA Server Functionality

To deliver content as well as expose various services, we use a REST web interface. These sections comment on the various samples and how they use the REST URLs.

For the full reference of the Pentaho REST API, please refer to <http://javadoc.pentaho.com>.

General Notes

The REST APIs often allow interaction with repository files. The REST API expects references to repository files to be specified using the `:` character as file separator.

- [Reporting Samples](#)
- [Publish a File to the BA Server](#)
- [Analyzer Samples](#)
- [Interactive Reporting Samples](#)
- [Integrating Pentaho Mobile for OEM Software Vendors](#)
- [Configuring the Proxy Trusting Filter](#)
- [Security Integration](#)

Reporting Samples

Executing a report from a HTML form (POSTing parameters)

This sample renders a prpt report by posting the report parameters to the report REST URL:

<http://localhost:8080/pentaho/api/repos/<path>/generatedContent>

The POST request contains all parameters the report expects, plus the additional rendering parameter `output-type`, which controls the rendering format (HTML, PDF, XLS, etc.)

The following output formats are supported:

Option	Purpose
table/html;page-mode=stream	HTML as a single page, all report pagebreaks are ignored.
table/html;page-mode=page	HTML as a sequence of physical pages, manual and automatic pagebreaks are active.
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet;page-mode=flow	Excel 2007 XLSX Workbook
table/excel;page-mode=flow	Excel 97 Workbook
table/csv;page-mode=stream	CSV output
table/rtf;page-mode=flow	Rich text format
pageable/pdf	PDF output
pageable/text	Plain text
pageable/xml	Pageable layouted XML
table/xml	Table-XML output
pageable/X-AWT-Graphics;image-type=png	A single report page as PNG.
mime-message/text/html	MIME email with HTML as body text and all style and images as inline attachments.

Rendering a report in an IFRAME

This sample renders a prpt report using a regular GET request, by populating an HTML IFRAME with the report REST URL:

<http://localhost:8080/pentaho/api/repos/<path>/generatedContent?<parameters>>

The request contains all parameters as part of the URL.

Rendering report viewer with available parameters and executed report

The sample renders the **Pentaho Report Viewer IU** by populating an HTML IFRAME with the report REST URL:

<http://localhost:8080/pentaho/api/repos/<path>/viewer?<parameters>>

The request contains the initial report parameters as part of the URL.

Publish a File to the BA Server

Follow this process if you want to publish a file to the BA Server:

Click the link that publishes a file to the BA Server: <http://localhost:8080/api/repo/publish/publishfile>. This publishes the file to the provided path in the repository. Note that the file will be overwritten if the overwrite flag is set to `true`.

Table 1. Parameters

Name	Description	Type	Default
pathId	(colon separated path for the repository file)	formdata	false
fileContents	(input stream containing the data)	formdata	false
overwriteFile	(flag to determine whether to overwrite the existing file in the repository or not)	formdata	false
fileInfo	(information about the file being imported)	formdata	false

Sample code for reference purposes can be found at [Github for Pentaho](#). Look for the file called

PublishRestUtil.java under the **pentaho-reporting** repository.

The file is now published to the BA Server and is available to users.

Analyzer Samples

Displaying a report in viewer mode

This sample renders a xanalyzer report using a regular GET request, by populating an HTML IFRAME with the report REST URL:

<http://localhost:8080/pentaho/api/repos/<path>/viewer>

The viewer has limited possibilities for interaction and does not allow changing the report.

Displaying a report in editor mode

The sample renders the report with full **Analyzer IU** by populating an HTML IFRAME with the report REST URL:

<http://localhost:8080/pentaho/api/repos/<path>/editor>

The IFRAME now renders the full Analyzer user interface, allowing users to interact with the data.

Creating a new report

The sample prepares a new report by populating an HTML IFRAME with the report REST URL:

<http://localhost:8080/pentaho/api/repos/xanalyzer/>

editor?catalog=<Schema>&cube=<CubeName>

The mandatory **catalog** and **cube** parameters specify the analysis schema and cube to use for the new report.

Integrating into a custom web application via an IFRAME

The sample opens a new page that mimics a parent application that integrates Analyzer using an IFrame. The parent application has a set of buttons interacting with the Analyzer UI in the embedded IFRAME. Studying this page is a good starting point, if you are trying to deeply integrate analyzer into your own web application.

Specifying the select schema service

The sample is opened with a specified data service.

<http://localhost:8080/pentaho/api/re...e/selectSchema>

After a data service is selected, the sample opens and shows an unsaved Analyzer report using that data service.

Interactive Reporting Samples

Displaying an Interactive report

This sample renders a `.prpti` report using a regular GET request, by populating an HTML IFRAME with the report REST URL. The viewer has limited possibilities for interaction and can't edit the report. The example below shows how to do this by replacing the path within the `<>` with the URL for the report that you want to view.

```
http://localhost:8080/pentaho/api/repos/<%3Ayour%3Apath%3Agoes%3Ahere.
prpti>/prpti.view
```

You can use the `prpti.edit` end-point instead, to allow more user interaction with the report.

```
http://localhost:8080/pentaho/api/repos/<%3Ayour%3Apath%3Agoes%3Ahere.
prpti>/prpti.edit
```

Creating a new interactive report

The sample prepares a new report by populating an HTML IFRAME with the report REST URL.

```
http://localhost:8080/pentaho/api/repos/pentaho-interactive-reporting/prpti.new
```

In order to show the toolbar buttons so that users can create, open, or save Interactive reports, add the following to the end of the URL.

```
?showRepositoryButtons=true
```

Users will be prompted to select the data source they want to work with.

Integrating into a custom web application via an IFRAME

The sample opens a new page that mimics a parent application that integrates the Interactive Reporting editor using an IFrame. The parent application has a set of buttons interacting with the Interactive Reporting UI in the embedded IFRAME. Studying this page is a good starting point, if you are trying to deeply integrate Interactive Reporting into your own web application.

Action Sequence Samples

Running an action sequence to generate a report

This sample renders an xaction report using a regular GET request, by populating an HTML IFRAME with the report REST URL:

[http://localhost:8080/pentaho/api/
repos/<path>/generatedContent?<parameters>](http://localhost:8080/pentaho/api/repos/<path>/generatedContent?<parameters>)

The parameters for the xaction are passed as simple GET parameters on the URL.

Running an action sequence to generate a report with prompts

The sample prepares a form to collect parameters and POSTs them to the xaction REST URL:

<http://localhost:8080/pentaho/api/repos/<path>/generatedContent>

The xaction now retrieves its parameters from the POST request body.

Integrating Pentaho Mobile for OEM Software Vendors

Pentaho uses Sencha Touch as a framework for building navigation features for our mobile product. If you are an independent software vendor who wants to create your own mobile application that integrates with Pentaho Mobile, you can use Sencha Touch or any other mobile SDK that meets your needs. These instructions use Sencha Touch. If you have chosen another mobile SDK, replace all references to Sencha Touch with the appropriate SDK.

These instructions specify Sencha Touch. If you have chosen to use a different Mobile SDK, replace all references to Sencha Touch with the alternate Mobile SDK.

1. Extract and install the `pentaho-mobile-plugin` to `pentaho-solutions/system`.
2. Download the `sencha-touch-2.0.1.1-commercial.zip` file and extract it to a temporary folder. See <http://www.sencha.com/products/touch/download/> for more information.
3. From the extracted files in the temporary folder, copy these files.

```
/sencha-touch-2.0.1.1/resources/css/apple.css
/sencha-touch-2.0.1.1/sencha-touch.js
/sencha-touch-2.0.1.1/sencha-touch-all.js
/sencha-touch-2.0.1.1/sencha-touch-all-debug.js
/sencha-touch-2.0.1.1/sencha-touch-debug.js
```

4. Paste the copied files into the mobile plugin touch folder: `/pentaho-solutions/system/pentaho-mobile-plugin/resources/touch`.
5. Ensure that the BA Server is running. It is not necessary to restart the BA server.

Access your content on an iPad by opening a web browser and entering

`http://<HostName:PortNumber>/pentaho/`.

Configuring the Proxy Trusting Filter

If you have set the BA Server to run on a specific IP address or hostname other than the default 127.0.0.1, you must modify the trusted proxy to match that address or hostname for Pentaho plugins to run as expected. Additionally, if you have applications that access BA Server resources, such as REST APIs, you must add that application's IP address so that the BA Server will accept those requests.

1. Stop the BA Server.
2. Open `biserver-ee/tomcat/webapps/pentaho/WEB-INF/web.xml` and search for `TrustedIpAddr`.

NOTE:

The `param-value` immediately below `TrustedIpAddr` is a comma-separated list of IP addresses that should be trusted.

3. Add the IP address of the host machine.

```
<filter>
  <filter-name>Proxy Trusting Filter</filter-name>
  <filter-class>org.pentaho.platform.web.http.filters.
ProxyTrustingFilter</filter-class>
  <init-param>
    <param-name>TrustedIpAddr</param-name>
    <param-value>127.0.0.1</param-value>
    <description>Comma separated list of IP addresses of a trusted
hosts.</description>
  </init-param>
</filter>
```

4. Start the BA Server.

Security Integration

Pentaho relies on the Spring Security pluggable authentication framework. By default, the BA Server uses a JDBC-based data access object that is tied to a Jackrabbit database. Users and roles are configured through the [Administration perspective](#) in PUC, and content authorization is controlled by the BA Server administrator. However, you can easily configure the server to use existing security tables in a different database, or to authenticate through your existing LDAP (including Active Directory) server or Central Authentication Service. Pentaho's security is also extensible to the point that you can create your own custom data access object, or completely remove all authentication functionality.

Developer Support

The examples in this section are simple and easy to follow, but with more complex requirements come more advanced programs. While reading the source code comments can help quite a bit, you may still need help to develop an application within a reasonable time-frame. Should you need personal assistance, you can have direct access to the most knowledgeable support resources through a Pentaho Enterprise Edition software vendor annual subscription:

[ISV/OEM support options](#)

If phone and email support are not enough, Pentaho can also arrange for an on-site consulting engagement:

[Consultative support options](#)

License Information

Most of the software comprising Pentaho Business Analytics is open source, licensed under the GNU General Public License version 2. Business Analytics also contains a large volume of third-party open source libraries that are licensed under a number of different licenses. Most of this software is freely redistributable, with the notable exceptions of the following Pentaho-authored programs:

- Dashboard Designer
- Analyzer
- Interactive Reporting
- Various individual BI Platform JARs

If you already have regular Pentaho licenses for the BI Platform, Dashboard Designer, Interactive Reporting, and Analyzer, then no further licenses are required to integrate BA Server functionality into a third-party application. If you wish to embed pieces of the BA Server into an application that you intend to sell or distribute, you must familiarize yourself with the licenses of all of the pieces you are including in order to make sure you are complying properly. Proprietary Pentaho software may not be redistributed under any circumstances.

This section is not intended for redistribution. However, the example code and example application that accompany this document may be freely modified or reused.