The IBM® Cognos® Mashup Service (CMS) allows you to easily extract Cognos Business Intelligence report content and integrate it into applications such as Google Maps, Google Earth, Yahoo desktop widgets, Adobe Flex, IBM Lotus Notes, and other third party Flash or charting engines. This article introduces CMS and includes several examples showing what you can accomplish using this new Web service.

Introduction

Cognos Mashup Service (CMS) is a new Web service that is included as part of the IBM Cognos Software Development Kit. This article introduces CMS and explains its two different interfaces: REST and SOAP. The article shows you some of what you can accomplish with CMS by providing sample code and highlighting some of the samples that are installed with the Cognos Mashup Service. To get the most out of this article, you should have basic knowledge of IBM Cognos 8 Business Intelligence (BI) and a basic understanding of programming languages and Web technologies.

System prerequisites

In order to execute the examples in this article, you need the following:

- IBM Cognos 8.4.1 BI
- IBM Cognos 8.4.1 Software Development Kit

What can you accomplish with CMS?

The IBM Cognos Mashup Service provides a way for you to integrate Cognos reports with other applications. You can use either of two interfaces to accomplish this integration:

- Representational State Transfer (REST). This interface uses basic HTTP requests.
Simple Object Access Protocol (SOAP). This interface can be used to programmatically access the CMS API.

The example below demonstrates a complete CMS integration. Figure 1 shows a portion of a sample IBM Cognos 8.4.1 Report Studio report. The report contains sales revenue data for a fictitious company. The data is broken down according to the location of each of the company's offices. Figure 2 shows the result when the report is integrated with a Google Earth map. The sales figures for each office are overlaid on the map according to the office's location. This is accomplished using the Google Maps API along with the CMS REST interface.

Note: This article is geared towards an introductory walk-through of CMS and does not cover the specifics of how to use the Google Maps API.

Figure 1. Sample IBM Cognos report results in HTML format

<table>
<thead>
<tr>
<th>Sales territory</th>
<th>Country</th>
<th>Region</th>
<th>City</th>
<th>Address line 1</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Central Europe</td>
<td>Germany</td>
<td>Berlin</td>
<td>Berlin</td>
<td>Nestorstraße 30, Berlin, Germany</td>
<td>$422,001.86</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Berlin, Germany</td>
<td>Nestorstraße 30, Berlin, Germany</td>
<td>$1,206,306.02</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Berlin, Germany</td>
<td>Nestorstraße 30, Berlin, Germany</td>
<td>$1,206,306.02</td>
<td></td>
</tr>
<tr>
<td>Brandenburg, Germany</td>
<td>Potsdam, Germany</td>
<td>Alhambrastraße 82, Potsdam, Germany</td>
<td>$602,139.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Potsdam, Germany</td>
<td></td>
<td></td>
<td></td>
<td>$602,139.00</td>
</tr>
</tbody>
</table>

Figure 2. IBM Cognos report results overlaid on Google Earth using IBM Cognos Mashup Service

How is CMS different from the Cognos Software Development Kit (SDK)?

Table 1 compares and contrasts the Cognos SDK with CMS.

Table 1. Cognos SDK compared with CMS

<table>
<thead>
<tr>
<th></th>
<th>SDK</th>
<th>CMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report authoring</td>
<td>Supports authoring</td>
<td>Does not support authoring</td>
</tr>
<tr>
<td>Programming languages</td>
<td>Java®, .NET Languages, VB</td>
<td>Java, .Net Languages, REST interface</td>
</tr>
</tbody>
</table>
Output formats | CSV, HTML, HTML Fragment, MHT, PDF, single XLS, spreadsheet ML, XHT, XLS, XLWA, XML | LDX, XML, HTML, HTML Fragment, JSON Note: LDX stands for Layout Data XML. It is a new format.
---|---
Major functionality | Almost everything that can be done with Cognos Connection, in addition to automating operations | Run and get the report outputs and enable access to report output elements to the smallest granular level possible
Authentication | Yes | Yes - via SDK or own authentication service, for REST you can also use Cognos 8 logon.
API | Java: Jar files available in sdk folder, Microsoft®.Net: .dll files available in sdk folder | Use of WSDL files to consume available functionality through Web services

**Using the CMS**

The examples in this section illustrate how to use the REST and SOAP interfaces to extract different components of a Cognos report.

**REST interface**

The generic URL for a REST request is:

```
http://webservername:portnumber/cognos8/cgi-bin/cognos.cgi/rds/resource_type/source_type/source_id?option1=val1&option2=val2...
```

Table 2 shows the main building components of the REST API.

<table>
<thead>
<tr>
<th>Resource types</th>
<th>Source types</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>auth/logon</td>
<td>conversationID</td>
<td>async</td>
</tr>
<tr>
<td>auth/logoff</td>
<td>path</td>
<td>burstID</td>
</tr>
<tr>
<td>atom</td>
<td>report</td>
<td>burstKey</td>
</tr>
<tr>
<td>promptPage</td>
<td>searchPath</td>
<td>contextId</td>
</tr>
<tr>
<td>reportData</td>
<td></td>
<td>direction</td>
</tr>
</tbody>
</table>

For more information, including additional options, refer to the *Mashup Services Developer Guide* (see the Related topics section for details on where to find the guide).

**REST example 1:**

This first REST API example demonstrates how to get all the charts available in a report that contains a combination of a list and charts. The example returns all the charts from the report in Layout Data XML (LDX) format.

The following parameters are used to build the URL for the example:

- Resource type: `reportData`
- Source type: `report`
• Resource_id: report store id; for example, iE0CDFEF149F8408F96F1C27DCFFA6BCB
• Options: fmt=layoutDataXML, xpath=//chart

The URL itself looks like this:

http://localhost/c841/cgi-bin/cognos.cgi/rds/reportData/report/iE0CDFEF149F8408F96F1C27DCFFA6BCB?fmt=layoutDataXML&xpath=//chart

Listing 1 shows the resulting LDX.

Listing 1. LDX resulting from the sample REST call

```xml
<filterResultSet xmlns="http://www.ibm.com/xmlns/prod/cognos/layoutData/200904">
  <filterResult>
    <filterType>XPATH</filterType>
    <filterValue>//chart</filterValue>
    <reportElement>
      <chart>
        <palette>
          <paletteItem>
            <pattern>
              <type>ePatternSolid</type>
              <fgColor>
                <red>51</red>
                <green>51</green>
                ...
              </fgColor>
            </pattern>
          </paletteItem>
        </palette>
      </chart>
    </reportElement>
  </filterResult>
</filterResultSet>
```

REST example 2:

This example accesses data from the same report as the first example. It returns a single chart from the report as an HTML Fragment.

The following parameters are used to build the URL for the example:

• Resource type: reportData
• Source type: report
• Resource_id: report store id; for example, iE0CDFEF149F8408F96F1C27DCFFA6BCB
• Options: fmt=HTMLFragment, selection=Chart1

The URL itself looks like this:

http://localhost/c841/cgi-bin/cognos.cgi/rds/reportData/report/iE0CDFEF149F8408F96F1C27DCFFA6BCB?fmt=HTMLFragment&selection=Chart1

Figure 3 shows the resulting chart generated from the returned HTML Fragment.
Before this operation is feasible, the user must be authenticated with IBM Cognos 8. One method to authenticate is to use the CMS authentication service logon method described in the Mashup Services Developer Guide (see the Related topics section for details on where to find the guide).

REST example 3:

This example shows how to run a report in HTML format with a maximum of three rows of data returned on each page.

The following parameters are used to build the URL for the example:

- Resource type: reportData
- Source type: report
- Resource_id: report store id; for example, i6243D3E1A36A4E0FAF9D90652C171FC2
- Options: fmt=HTML, selection=List1, rowLimit=3

The URL itself looks like this:

http://localhost/c841/cgi-bin/cognos.cgi/rds/reportData/report/i6243D3E1A36A4E0FAF9D90652C171FC2?fmt=HTML&selection=List1&rowLimit=3

Figure 4 shows the resulting table generated from the returned HTML.

Figure 4. Limited output in HTML format using IBM Cognos Mashup Service

<table>
<thead>
<tr>
<th>City</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amsterdam</td>
<td></td>
</tr>
<tr>
<td>Bilbao</td>
<td></td>
</tr>
<tr>
<td>Birmingham</td>
<td></td>
</tr>
</tbody>
</table>

SOAP interface

Table 3 shows a few of the generic methods of the SOAP API interface.
Table 3. SOAP methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>drill</td>
<td>Use this secondary method to drill up or down in an existing report session.</td>
</tr>
<tr>
<td>getOutput</td>
<td>Use this method to retrieve the output of a report.</td>
</tr>
<tr>
<td>getPromptAnswers</td>
<td>Use this method after a getPromptPage request to retrieve the answers associated with a prompt page.</td>
</tr>
<tr>
<td>getPromptContent</td>
<td>Use this method to retrieve the content of a prompt page.</td>
</tr>
<tr>
<td>getReportData</td>
<td>Use this method to retrieve the content of a report.</td>
</tr>
</tbody>
</table>

For more information, including additional options, refer to the Mashup Services Developer Guide (see the Related topics section for details on where to find the guide).

SOAP example using Java API:

This example demonstrates how to extract a list that accesses all the report elements from a report that includes a list and a graph.

Sample Java code

```java
ReportDataServicePortProxy proxy = new ReportDataServicePortProxy();
//The endpoint will be the gateway URL for your server
proxy.setEndpoint(url);

ReportDataServicePort mashupService = proxy.getReportDataServicePort();

GetReportDataRequest request = new GetReportDataRequest();
//This sets the cm searchPath as the source of the report, you
//could alternatively use the storeID
request.setSourceID("/content/folder[@name='Samples']/folder[@name='Models']/package[@name='GO Data Warehouse (analysis)']/folder[@name='Query Studio Report Samples']/query[@name='Returns by Product Type']");

//This identifies that the sourceID refers to a CM search path
request.setSourceType(SourceTypeEnum.searchPath);

//This creates an object id filter to only return the List1 element,
//OBJECT_ID is the type of filter that
//filters based on element id.
Filter[] filters = {new Filter("List1", FilterTypeEnum.OBJECT_ID, null)};
request.setFilters(filters);

GetReportDataResponse response = mashupService.getReportData(request);
GetReportDataRequest outputRequest = new GetReportDataRequest();

/*
* This loop is necessary when running asynchronously
*/
while(response.getSession().getStatus() == SessionTypeStatus.working)
{
    outputRequest.setSession(response.getSession());
    response = mashupService.getOutput(outputRequest);
}

/*
* This block of code goes through the list contents and outputs them to the console
*/
ListFrame list1 = response.getOutput().getLDXOutput().getFilterResultSet().getFilterResult()[0];
```
```java
getRowElement()[0].getLst();
Row[] rows = list1.getGroup().getRow();

System.out.println();
System.out.print("Product type");
int spaceOfCell=25-"Product type".length();
for (int n=0;n<spaceOfCell;n++)
{ System.out.print(" ");}

System.out.println("Base product");
spaceOfCell=35-"Base product".length();
for (int n=0;n<spaceOfCell;n++)
{ System.out.print(" ");}

System.out.println("Lost revenue");
spaceOfCell=20-"Lost revenue".length();
for (int n=0;n<spaceOfCell;n++)
{ System.out.print(" ");}

System.out.println();
System.out.println();
for(int i = 0; i < rows.length; i++)
{
Cell[] cells = rows[i].getCell();
System.out.print(cells[0].getItem(0).getTxt().getFmtVal());
spaceOfCell=25-(cells[0].getItem(0).getTxt().getFmtVal().length());
for(int j=0;j<spaceOfCell;j++)
{   System.out.print(" ");}
System.out.print(cells[1].getItem(0).getTxt().getFmtVal());
spaceOfCell=35-cells[1].getItem(0).getTxt().getFmtVal().length();
for(int j=0;j<spaceOfCell;j++)
{   System.out.print(" ");}
System.out.print(cells[5].getItem(0).getTxt().getFmtVal());
spaceOfCell=20-cells[5].getItem(0).getTxt().getFmtVal().length();
for(int j=0;j<spaceOfCell;j++)
{   System.out.print(" ");}
System.out.println();
}
```

Figure 5 shows an example of what the output generated by the above sample code would look like:

**Figure 5. Results from Java example**

<table>
<thead>
<tr>
<th>Product type</th>
<th>Base product</th>
<th>Lost revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insect Repellents</td>
<td>BugShield Lotion</td>
<td>$159,170.37</td>
</tr>
<tr>
<td>Navigation</td>
<td>Trail Star</td>
<td>$483,691.20</td>
</tr>
<tr>
<td>Insect Repellents</td>
<td>BugShield Lotion Lite</td>
<td>$26,641.48</td>
</tr>
<tr>
<td>Sunscreen</td>
<td>Sun Shield</td>
<td>$89,374.32</td>
</tr>
<tr>
<td>Lanterns</td>
<td>EverGlow Lamp</td>
<td>$434,454.32</td>
</tr>
<tr>
<td>Insect Repellents</td>
<td>BugShield Extreme</td>
<td>$174,837.10</td>
</tr>
<tr>
<td>Sunscreen</td>
<td>Sun Blocker</td>
<td>$31,759.65</td>
</tr>
<tr>
<td>Sunscreen</td>
<td>Sun Shelter 15</td>
<td>$44,971.96</td>
</tr>
<tr>
<td>Sunscreen</td>
<td>Sun Shelter 30</td>
<td>$80,952.30</td>
</tr>
<tr>
<td>Tents</td>
<td>Star Gazer 6</td>
<td>$398,970.00</td>
</tr>
<tr>
<td>Tents</td>
<td>Star Gazer 2</td>
<td>$1,963,638.14</td>
</tr>
<tr>
<td>Sunscreen</td>
<td>Sun Shelter Stick</td>
<td>$25,666.96</td>
</tr>
<tr>
<td>Putters</td>
<td>Blue Steel Max Putter</td>
<td>$325,192.62</td>
</tr>
</tbody>
</table>
Sample applications

Currently, CMS includes several samples. The samples are installed by default as part of the IBM Cognos Software Development Kit.

The REST samples are written in HTML and JavaScript. These samples can be run from a Web server and are installed in the following directory:

<\c8_install>/webcontent/samples/sdk/cms

The SOAP API samples are written in C# and Java. These samples are installed in the following directory:

<\c8_install>/sdk/cms

For more information on the samples, refer to the Mashup Services Developer Guide (see the Related topics section for details on where to find the guide).

Figure 6 shows a screenshot of the main menu of the JavaScript samples to give you an idea of what they contain. The samples are not meant to be end-user applications; they only demonstrate what you can accomplish with CMS.

Figure 6. Sample programs

Summary

Thank you for your interest in this introductory article to IBM Cognos Mashup Services. The purpose of the article was to give you an idea of what can be accomplished with CMS as well as how to perform some basic tasks using the two different interfaces.
Related topics

- The *Mashup Services Developer Guide* is included with the installation of IBM Cognos software. After installing Cognos, you can find the English version of the guide here: *IBM Cognos installation directory/webcontent/documentation/en/dg_cms.pdf*
- "IBM Cognos Mashup Service Product description and demo"
- "IBM Cognos SDK Validation Tool" (developerWorks, August 2008), provides information about the Cognos SDK validation tool that you can use to run a complete validation check of all the reports in the Content Store.

© Copyright IBM Corporation 2010
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)