Dynamic authentication for connectors in WebSphere Cast Iron, Part 2: Authenticating JDE, Salesforce.com, and Force.com Bulk API connectors dynamically

Madhavi Kota (madhkota@in.ibm.com)
Staff Software Engineer
IBM

Aparna Srinivasan (apasrini@in.ibm.com)
Software Engineer
IBM

22 July 2015

Part 2 of this two-part series describes how connectors such as JD Edwards, Salesforce.com®, and Force.com® Bulk API are authenticated dynamically in IBM® WebSphere® Cast Iron Studio. This tutorial shows how to dynamically pass the connection parameters, connection pool options, and proxy parameters, without the need of stopping the project in the Appliance. This method allows you connect to different endpoints or change endpoint configurations during runtime without stopping the orchestration.

Introduction

WebSphere Cast Iron (hereafter called Cast Iron) is an offering from IBM that provides clients with a platform for integrating cloud-based applications from leading Software as a Service (SaaS) providers with on-premise applications from IBM and other companies. IBM Cast Iron Studio (hereafter called Studio) is a development tool that is used to design, test, and publish integration projects to an IBM Cast Iron Integration Appliance. Studio provides numerous activities that you can drag into a workspace and configure as part of one or more business-process orchestrations that comprise an integration project.

Dynamic endpoint connection option allows you to connect to different endpoints at runtime or appliance. For example, to change the user name and password dynamically without stopping the project or orchestration, you can add optional parameters for the connection properties to enable this feature.
Objectives of this tutorial

Part 1 of this series discussed how to dynamically provide the connection parameters and connection pool options for authentication in Cast Iron for SAP and HTTP connectors. Part 2 explains how to dynamically authenticate various connectors such as JDE, Salesforce.com, and Force.com Bulk API connectors in Studio. It also describes how to dynamically pass the values of the authentication in Cast Iron Appliance and Cast Iron Live.

Prerequisites

You need to be familiar with Cast Iron Studio and Cast Iron Appliance. Basic knowledge about the terms used in Studio and knowledge about the different connectors supported by Studio would be helpful. Also, reading through the following tutorial, Static and dynamic ways of providing input to connector activities using WebSphere Cast Iron Studio, will help you better understand this tutorial.

System requirements

The following products and assets are required to configure and deploy the module:

- IBM WebSphere Cast Iron Studio
- IBM WebSphere Cast Iron Appliance

Dynamic authentication with Cast Iron connectors

Dynamic authentication is a way to connect or change endpoint configurations without stopping the project during runtime. Adding optional parameters of the connection properties allow you to enable this feature. If a value is not provided for an element in the optional parameters of the Connection properties, then no exception is thrown during runtime because it is picked from the endpoint configuration panel, which is mandatory. If the values are provided in the optional parameters of the connection properties, then the precedence is provided to the optional parameters and this will override the endpoint values. Also, this feature enables you to connect to different endpoint servers for different activities in an orchestration.

Dynamic authentication for the Salesforce.com connector

The Salesforce.com connector supports dynamic authentication to connect to different Salesforce.com endpoints. You can enable this feature by adding optional parameters in the "Map Input" section of a Salesforce.com activity. There are many headers, such as MruHeader, Email Header, Package Version Header, and other headers, for which optional values are provided. Along with these, you can provide input to the connection parameter for dynamic authentication. If you want to dynamically connect to a Salesforce.com account other than the one configured in the endpoint panel, dynamic authentication is the right choice as this is done without stopping the project or orchestration.

In Cast Iron, you can connect to Salesforce.com through the Salesforce.com connector through a user name and password with a security token and then a proxy server. You can even provide the session information of a particular session, such as the session URL and session ID from the previous call made to the Salesforce.com session. You can retrieve this value from the map output of the previous activity.
Complete the following steps to create dynamic authentication for the Salesforce.com connector:

1. Create a new project in Cast Iron Studio and create an endpoint for Salesforce.com connector. Provide the value for the user name and password, which is a combination of a password and security token. Also add the details of the proxy server if needed. Create the configuration properties for each if needed. Verify the test connection by clicking on the **Test Connection** button to ensure the connection is made with the server.

2. In the orchestration workspace, create an FTP endpoint where the XML of the input value (user name and password of the Salesforce.com endpoint) is already present.

3. Drag and drop the **Poll Directory** activity of the FTP connector and then add **Read XML** activity to read the XML input, which is sent from the FTP server.

4. Drag and drop one activity of Salesforce.com, for example, **Create object activity**. For more information about how to provide input to connector activities dynamically using Cast Iron Studio, refer to [Static and dynamic ways of providing input to connector activities using WebSphere Cast Iron Studio](https://www.ibm.com/support/knowledgecenter/SSSHXK_9.1.0/com.ibm.wci.doc/using/using_cis_dev_dynamic_input.html).

5. In the Configure panel of the activity, select **Account object** as an example. In the Map Input section, provide the value of name, address, and so on.

6. Right-click on **Object** in the **To Activity** of the Map Input section and select **Show Optional Parameters**.

7. Once selected, the optional parameters are displayed. The connection parameter (optional) lists the parameters for dynamic authentication as shown in Figure 1.

8. Dynamically pass the values of the user name and password as obtained from the XML file from the FTP location as shown in Figure 1. You can change the values of the user name and password dynamically when the project is in a running state.
9. You can also provide the session information of a particular session such as a session URL and session ID from the previous call made to the Salesforce.com session. You can retrieve this value from the map output of the previous activity. Map the output of the create object activity to the orchestration and add one more activity to the orchestration, for example, the Update object activity.

10. In the Update object activity located in the map input section, right-click on the object in the To Activity section and add the optional parameters. From the XML file, pass the value of session URL and session ID if needed and change the values of the session ID and URL dynamically as shown in Figure 2.
Dynamic authentication for the Force.com Bulk API connector

The Bulk API is based on REST principles and is optimized to load or delete large sets of data. You can use it to query, insert, update, upsert, or delete a large number of records asynchronously by submitting batches, which are processed in the background by Salesforce.com. The Bulk API connector is dependent on the Bulk API of Salesforce.com and allows you to do a Bulk Insert, Bulk Upsert, Bulk Update, Bulk Delete, Get Batch Status, and Get Batch Results Activities/Operations. By dynamically authenticating the Force.com Bulk API connector, you can dynamically authenticate the connector to change or overwrite the values provided in the endpoint panel.

In Cast Iron, you can connect to the Force Bulk API operations through the Force Bulk API connector and through a user name and password or proxy connection, which is the same as the Salesforce.com endpoint details.

Complete the following steps to create a dynamic authentication for the Force.com Bulk API connector:

1. Create a new project in Cast Iron Studio and create an endpoint for the Force.com Bulk API connector. Provide the value for the user name and password, which is a combination of a password and security token. Also add the details of the proxy server if needed. Create the configuration properties if needed. Verify the test connection.
2. In the orchestration workspace, create an FTP endpoint where the XML of the input value (user name and password of the Force.com Bulk API endpoint) is already present.
3. Drag and drop the **Poll Directory** activity of the FTP connector, and then add **Read XML** activity to read the XML input.

4. Drag and drop one activity of Force.com Bulk API connector, for example, **Create object activity**. For more information about how to provide input to the connector activities dynamically using Cast Iron Studio, refer to [Static and dynamic ways of providing input to connector activities using WebSphere Cast Iron Studio](https://www.ibm.com/developerworks/).

5. In the Configure panel of the activity, select **Account object** as an example. In the Map Input section, provide the name, address, and so on.

6. Right-click on **Object** in the **To Activity** of the Map Input section and select **Show Optional Parameters**.

7. Once selected, the optional parameters are displayed. The proxy connection parameter (optional) lists the parameters for dynamic authentication as shown in Figure 4.

8. Dynamically pass the values of the user name and password as obtained from the XML file from the FTP location, as shown in Figure 3. You can also change the values of the proxy details dynamically under the "(optional) header" input fields.

**Figure 3. Dynamically passing connection details to the Force.com Bulk API connector**

9. As shown in Figure 4, you can also add the proxy connection details separately from the **Proxy connection panel**.

Dynamic authentication for connectors in WebSphere Cast Iron, Part 2: Authenticating JDE, Salesforce.com, and Force.com Bulk API connectors dynamically
Dynamic authentication for the JDE Connector

The JD Edwards (JDE) Connector provides the ability for the Cast Iron Appliance or Cast Iron Live to interact with the JD Edwards EnterpriseOne (E1) server and to execute JDE E1 business functions and receive real-time events, all without the need for coding. The connector works with the JDE E1 server by using the Java™ Dynamic Connector provided in the JDE tools API. This connector has two activities – Poll Real Time Events and Invoke Business Function.

In Cast Iron, you can work with the JDE server through the JDE Connector by providing the environment, role, username, and password for the JDE server.

1. Create a new project in Cast Iron Studio and add the third party JAR files that were downloaded to the local subdirectory by browsing it. Select **Tools > Install Module Providers** in Cast Iron Studio. You can see the "Add Providers" window in Figure 5. A pop-up appears that asks you for a manual restart of Studio. Restart studio and check if the JARs are added to Add Providers.
2. Once restarted, click on the "new endpoint" icon under the Project name on the right-hand side of the Studio. Click on JDE Endpoint. The Connection Information tab opens. Provide the following values for the connection information:

- **Environment:** This specifies the code representing the JDE environment. It specifies the JDE EnterpriseOne environment name.
- **Role:** This specifies the name of the role that is associated with the user name used to access the JDE EnterpriseOne environment.
- **User name:** This specifies the user name that the integration appliance uses to connect to the JDE application. It specifies the user name to connect to the JDE server.
- **Password:** This specifies the password associated with the application-access user name.

3. Create the configuration properties if needed. Verify the test connection.

4. In the orchestration workspace, create an FTP endpoint where the XML of the input value (environment, role, username, and password of the JDE endpoint) is already present.

5. Drag and drop the **Poll Directory** activity of the FTP connector and then add **Read XML** activity to read the XML input.

6. Drag and drop one activity of the JDE connector, for example, **Invoke Business Function activity**. For more information about how to provide input to the connector activities dynamically using Cast Iron Studio, refer to [Static and dynamic ways of providing input to connector activities using WebSphere Cast Iron Studio](http://ibm.com/developerWorks/).

7. In the Configure panel of the activity, select **CFIN > B0100033 > GetEffectiveAddress object** as an example. In the Map Input section, provide the value for **Mnaddressnumber** such as "1".

8. Right-click on **Object** in the **To Activity** of the Map Input section and select **Show Optional Parameters**.

9. Once selected, the optional parameters are displayed as shown in Figure 6.
10. Dynamically pass the values of the user name, password, and also the Connection Pool properties as obtained from the XML file, which is from the FTP location as shown in Figure 6. You can change the values of the user name and password and the connection pool properties dynamically when the project is in running state.

Conclusion

This tutorial series explained how to dynamically authenticate various connectors in Cast Iron Appliance. Part 2 described how to dynamically authenticate connectors such as Salesforce.com, JD Edwards, and Force.com Bulk API in WebSphere Cast Iron Studio. Part 1 explained how you can dynamically provide the connection parameters and connection pool options for authentication in Cast Iron Appliance for SAP and HTTP connectors.

Acknowledgments

The authors would like to thank C. Vamsi and Hariharan I. Subramanian for their help in reviewing this tutorial and providing valuable input.
Resources

- IBM WebSphere Cast Iron V7 Knowledge Center
- Static and dynamic ways of providing input to connector activities using IBM WebSphere Cast Iron
- Working with JD Edwards Connector or WebSphere Adapter on different runtimes
- Dynamic authentication for connectors in WebSphere Cast Iron, Part 1: Authenticating SAP and HTTP connectors dynamically
- Cast Iron community forums
- developerWorks WebSphere zone
About the authors

Madhavi Kota

Madhavi Kota is a Staff Software Engineer on the WebSphere Cast Iron team. She has worked with the IBM TXSeries for Multiplatforms and IBM WebSphere Adapters teams, and she currently works on the WebSphere Cast Iron team using technologies such as Java and Rational Functional Tester. She has a Bachelor of Technology degree from the Vignan Institute of Technology and Science in Chennai, India.

Aparna Srinivasan

Aparna Srinivasan is a Software Engineer and L3 Support Engineer for the WebSphere Cast Iron and WebSphere Adapters team. She has over 3 years of experience working with technologies such as Java and Rational Functional Tester. She has a Bachelor’s Degree in Engineering from Anna University, Chennai, India.

© Copyright IBM Corporation 2015
Trademark
(www.ibm.com/developerworks/ibm/trademarks/)