

IBM Cloud Integration for Bluemix - User Guide

Contents

- What is Cloud Integration for Bluemix? 5
 - What is in it for me? 6
- Connecting to an Add-On 7
- Managing Secure Connections 9
- Creating a Cloud Integration API - overview 9
- How To Docs..... 11
 - How to add and install a Standard Secure Connector 11
 - To install the secure connector: 12
 - How to start and stop connectors on Windows (Service) 14
 - How to start and stop connectors on Windows (application)..... 14
 - How to start and stop secure connectors on Linux 14
 - How to upgrade Secure Connectors 15
 - How to re-configure the Secure Connector 15
 - Creating a basic connection 16
- How to manage DataPower Secure Connections..... 19
 - How to create the Enterprise Secure Connector keys and certificates..... 21
 - How to export self-signed certificates 23
 - How to create a cloud gateway service 23
 - How to add a cloud connector certificate..... 31
 - How to add the cloud connector certificate to the validation credentials certificate list..... 32

How to create an API from a database endpoint (DB2 or Oracle)	34
How to create an Enterprise API from an SAP endpoint	40
How to create an API from Cast Iron Live Orchestrations	45
How to create an API from Bluemix applications	47
How to sign up for Cast Iron Live Evaluation version	48
Creating an on-premises API endpoint	50
Configuring your application for TLS (public) or mutual TLS (private) access	52
Creating a new Cloud Integration REST API that links to an existing on-premises API	53
Publishing an API as a private service	57
Connecting to a user-defined endpoint	58
How to create an integration (click-through)	59
How to view an API	61
How to call an API	61
Downloading an SDK	62
Creating a Data Sync	64
Request payload generation rules	69
Troubleshooting	70
AGENT_LOCKED error when starting the standard Secure Connector	71
Common errors thrown when you invoke the API	71
Getting started with Cloud Integration service	72
Connecting an Add-On	72
Before you begin	72
Managing the Enterprise Secure Connector	73
Creating the Enterprise Secure Connector keys and certificates	73
Exporting self-signed certificates	74

Creating a Cloud Gateway Service	74
Adding a cloud connector certificate	75
Adding the cloud connector certificate to the validation credentials certificate list	75
Managing secure connections - Standard Secure Connector	75
Starting and stopping Secure Connectors on Windows (Installed as a Windows Service).....	77
Starting and stopping Secure Connectors on Windows (Installed as a Windows Application).....	77
Starting and stopping Secure Connectors on Linux	78
Upgrading Secure Connectors	78
Updating (re-configuring) the Secure Connector	79
Creating a basic connection	80
Creating an Integration	81
Signing up for Cast Iron Evaluation Version.....	81
Connecting to a user-defined endpoint.....	82
Creating Cloud Integration APIs.....	82
Creating an API from a database endpoint (DB2 or Oracle).....	83
Creating an Enterprise API - SAP endpoint	86
Creating an API from Cast Iron Live Orchestrations	87
Creating an API from Bluemix applications.....	87
Creating an on-premises API endpoint.....	87
Configuring your application for TLS (public) or mutual TLS (private) access	88
Creating a new Cloud Integration REST API that links to an existing on-premises API	89
Publishing an API as a private service	90
Viewing APIs.....	90
Using the API URL.....	90
Downloading an SDK.....	91

Creating a Data Sync	93
Request payload generation rules	96
Troubleshooting.....	96
How can I use multiple Secure Connectors in a single machine?	97
AGENT_LOCKED error when starting the standard Secure Connector	97
Common errors thrown when you invoke the API	98
Secure Connector window hangs and no information is displayed.....	98
The Secure Connector installer on a 64-bit Linux machine throws error	98
How to set the debug value of the Secure Connector logs to True.....	99
The application that was used to create the data sync is removed	99
The on-premises database password has changed	99
You accidentally click Migrate twice during a data sync.....	99
Tips and tricks.....	99

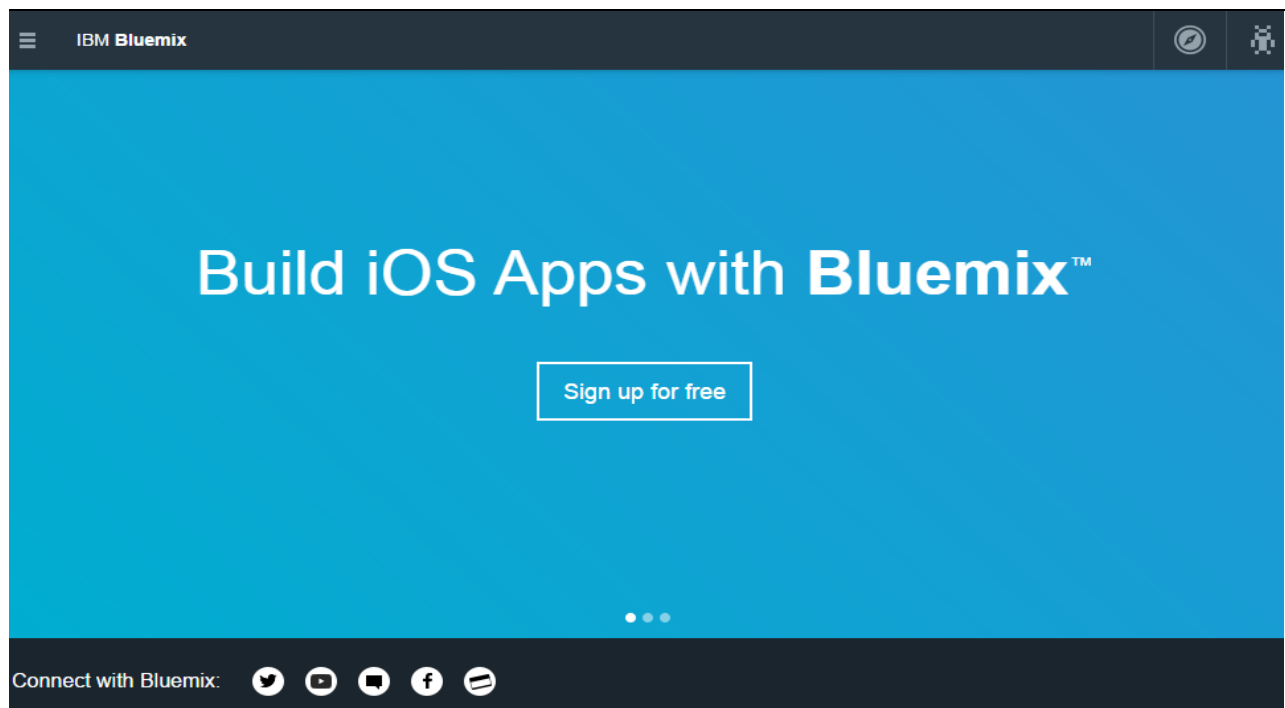
What is Cloud Integration for Bluemix?

IBM® Bluemix is the new Platform-as-a-Service (PaaS) offering, built on Cloud Foundry open source technology. Bluemix is an open platform for developing and deploying mobile and Web applications. Bluemix provides a broad set of services and runtimes that gives the integration developer control and flexibility to build an application.

IBM® Cloud Integration for Bluemix service enables you to integrate cloud services with enterprise systems of record. This service exposes the backend systems of record as ReST APIs to be used by applications.

Cloud Integration for Bluemix enables you to integrate cloud and on-premise applications. The cloud code leverages the Cloud Integration service to interact with the backend databases such as DB2, Oracle, and SAP to create database APIs.

Figure 1: Bluemix home page

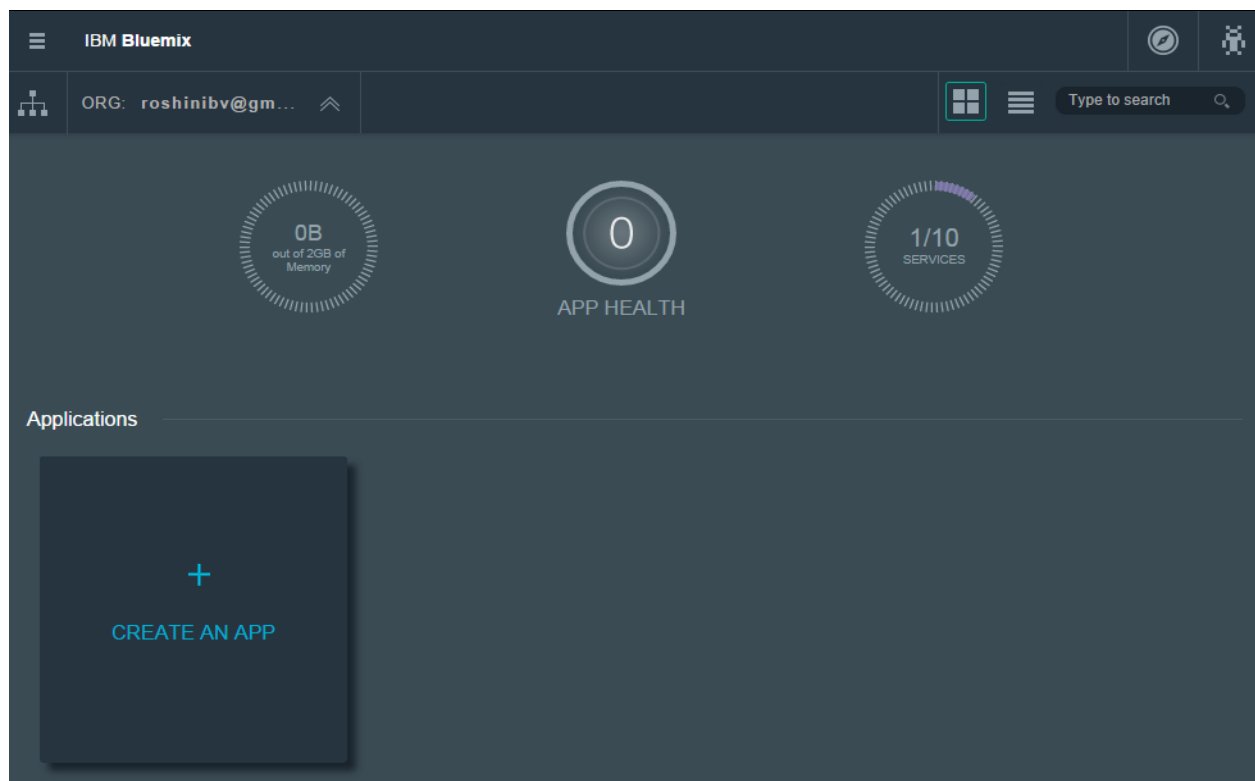


What is in it for me?

IBM® Cloud Integration for Bluemix service offers you:

- Fast and simple service: Cloud Integration for Bluemix enables you to quickly integrate cloud services with enterprise systems of record.
- Connectivity: Cloud Integration for Bluemix exposes backend systems of record as ReST APIs to be used by applications.
- Secure communication: Cloud Integration for Bluemix enables secure communication with on-premise Secure Connectors.
- Expose only those schemas and tables that you want to expose to the application.

Figure 2: Bluemix Dashboard

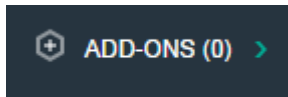


Connecting to an Add-On

Cloud Integration is an add-on service offering in Bluemix. To start working on Cloud Integration, you must first connect to an add-on service.

To connect to an add-on:

1. Click the **ADD-ONS** option in the Dev pane.



The number next to the option name indicates the number of add-ons that have been created.

2. Click the **Connect an Add-On** option. The Add-Ons page is displayed.

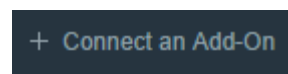
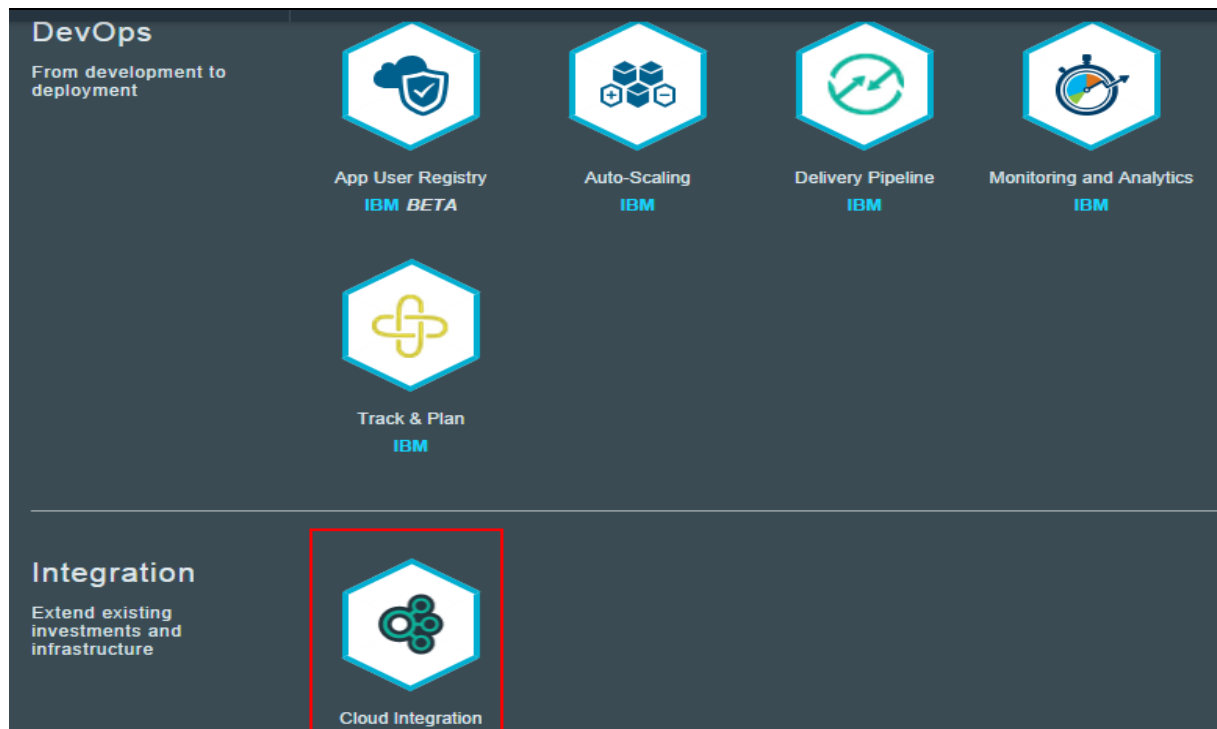


Figure 3: Add-Ons page



3. Click the Cloud Integration tile. The Cloud Integration catalog page is displayed.

Figure 4: Cloud Integration Catalog page

The screenshot displays the Cloud Integration service page. On the left, there is a logo for Cloud Integration by IBM, along with the publish date (6/27/2014) and type (Add-On). A 'VIEW DOCS' button is also present. The main content area describes the service as enabling users to integrate cloud services with enterprise systems of record as ReST APIs. It lists two key features: 'Fast and Simple' and 'Connectivity'. Below this is a 'Pick a plan' section with a dropdown for 'Monthly prices shown are for country or region: United States'. A table lists the 'Cloud Integration Plan' with its features and pricing. The 'Add Service' pane on the right shows the 'App' set to 'Leave unbound' and the 'Selected Plan' as 'Cloud Integration Plan', with a prominent 'CREATE' button.

Cloud Integration
IBM

PUBLISH DATE
6/27/2014

TYPE
Add-On

VIEW DOCS

Bluemix Cloud Integration enables users to integrate cloud services with enterprise systems of record. Bluemix Cloud Integration exposes the backend systems of record as ReST APIs to be used by applications

- **Fast and Simple**
Bluemix Cloud Integration enables users to quickly integrate cloud services with enterprise systems of record
- **Connectivity**
Bluemix Cloud Integration exposes the backend systems of record as ReST APIs to be used by applications

Pick a plan Monthly prices shown are for country or region: United States

Plan	Features	Price
✓ Cloud Integration Plan	Free - 5 relational database end-points are free. Any additional endpoint and/or any Application Integration endpoint will be charged. Standard - \$150 per Month/Integration Endpoint beyond the free end points and/or for any Application Integration Endpoint. Unlimited - \$750 per Month for usage of unlimited number of end points. To be converted automatically after 5 charged Integration Endpoints.	\$150.00 USD/ENDPOINT

i Integrate with 5 enterprise database endpoints for free and there after at \$150 per endpoint per month

TERMS

Connect an Add-On:

App:
Leave unbound

Selected Plan:
Cloud Integration Plan

CREATE

Currently, only the Standard plan is offered for Cloud Integration service. The plan, features, and price are listed in the **Pick a plan** section. You can also choose the plan pricing country-wise.

4. Select an application, and the plan in the **Add Service** pane.
5. Click **Create**.

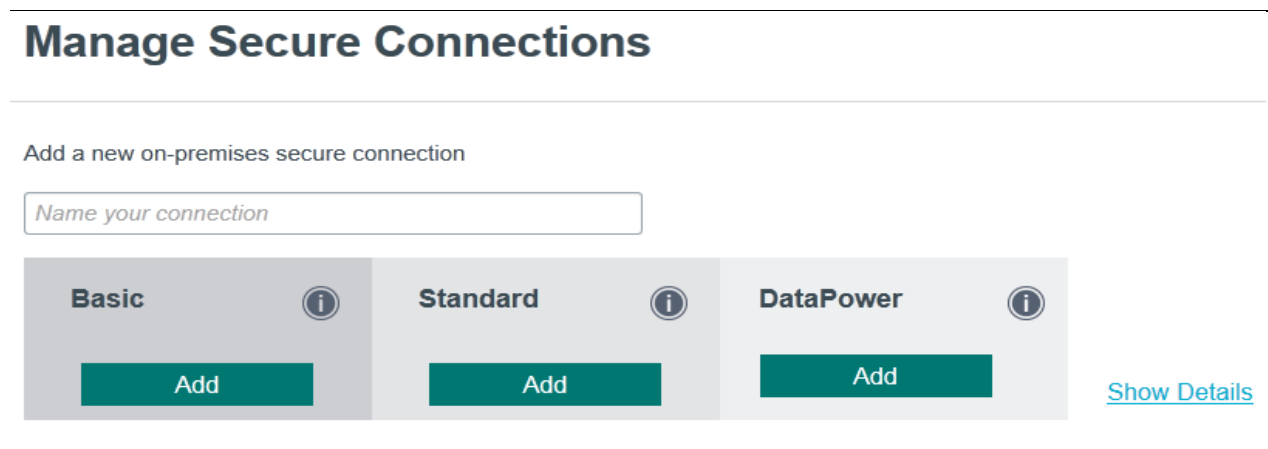
You have now successfully connected to the Cloud Integration add-on service.

Managing Secure Connections

Since the enterprise server is behind a firewall, you will be able to access the server only through a secure connection. You can add an on-premise secure connector by adding and configuring the secure connector:

- **Basic Secure Connector** - The Basic (Software) Secure Connection is a simple software based connection that establishes a tunnel between Bluemix applications and the network on which it is installed, leveraging a secure (SSH) access that eliminates the need for a firewall port.
- **Standard Secure Connector** - The Standard (Cast Iron) Secure Connector is a simple software based connector that establishes a tunnel between Bluemix applications and the network on which it is installed, leveraging a secure (HTTPS) access that eliminates the need for a firewall port.
- **DataPower Secure Connector** - The Enterprise (DataPower) Secure Connector leverages an on-premise DataPower deployment as a secure gateway connection between backend resources (behind the enterprise firewall) and Bluemix applications, ensuring high availability/fail-over and load balancing requirements.

Figure 5: Managing Secure Connections



Creating a Cloud Integration API - overview

This section describes the various steps involved in creating Cloud Integration APIs. You can create Cloud Integration APIs in the following ways:

- **Enterprise Endpoint** - You can effectively use Cloud Integration for Bluemix to quickly connect, browse, and access your database residing on your premise. By creating enterprise endpoint APIs, you can easily access your database tables in a ReST-based

fashion. These APIs can then be used to integrate or access your database from other applications.

- **Cast Iron Live Orchestrations** - You can use Cloud Integration for Bluemix to maintain your Cast Iron Live Orchestration projects that have HTTP Receive activity. You can maintain a catalog of these projects, that can be used by anyone.
- **Bluemix applications** - When you implement an application on Bluemix, you generally access that service through a URL. For effective maintenance of Bluemix applications, you can maintain a catalog of Bluemix apps by creating the APIs for these Bluemix applications, that can be used by anyone.

The steps and procedures are explained in detail in the How To Docs sections.

Figure 6: Creating an Enterprise API

Create an API

How can others in this organization find your API?

Generate from an Enterprise Endpoint Create from Cast Iron Live Orchestrations Create from a Bluemix App Create from an On-premises API

Cancel

How To Docs

This section describes the various steps followed to create an API, view and API, register and API and so on.

How to add and install a Standard Secure Connector

Since the enterprise server is behind a firewall, you will be able to access this server only through a Secure Connector. Follow the procedure given below to install and run a basic Secure Connector.

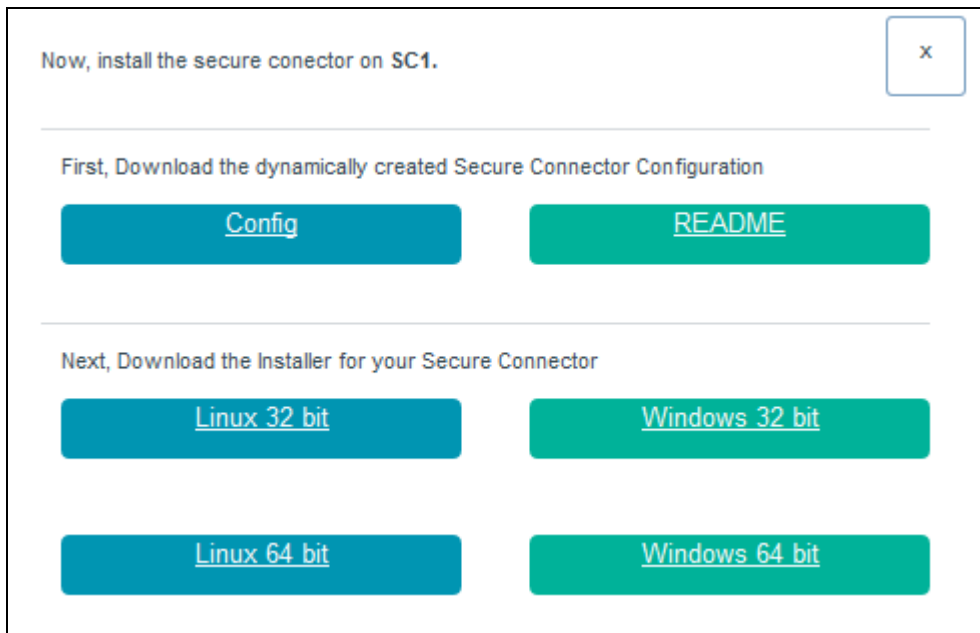
After you create a Secure Connector in the cloud, you must configure a machine behind the firewall to facilitate communication between the Secure Connector and a specific endpoint behind the firewall. Use the Secure Connector installer to configure the machine behind the firewall.

Note: The machine on which you choose to run the installer must have access to the endpoint. You do not have to run the installer on the same machine as the endpoint.

To install the Standard Secure Connector:

1. On the Manage Connections page, specify the name of your connection, in the field present below **Want to install your own basic connector?**.
2. Click the **Add and Install** button. The name of your connection is added to the list. At first the status of your connection is **Not Connected**.
3. Click the **Install** button. The following page is displayed:

Figure 10: Installing a Secure Connector



4. Download the configuration file and the readme file.
5. Download the appropriate installer for your Secure Connector.
6. On the Manage Secure Connections page, click Done.

Now, start creating your APIs by connecting through this newly created secure channel.

To install the secure connector:

1. Launch the Secure Connector installer you downloaded.
 - windows-secure-connector-installer-32bit.exe
 - windows-secure-connector-installer-64bit.exe
 - linux-secure-connector-installer-32bit.sh
 - linux-secure-connector-installer-64bit.sh

The Secure Connector Installer Wizard is displayed.

2. Click **Next** then read and accept the licensing agreement.

Note: On a 32-bit or 64-bit Linux machine, you must download a 32-bit or 64-bit Secure Connector, respectively, and complete the upgrade process. You cannot upgrade a 32-bit Secure Connector on a 64-bit Linux machine.

Note: If you already have a Secure Connector installation that is higher than or same as the latest version, a warning message states that you have an existing installation and alternatively you can upgrade the existing installation.

Note: You must stop the Secure Connector (if already started) before upgrading.

Note: Before you proceed with the Secure Connector upgrade process, ensure that you have:

- Stopped the Secure Connector and corresponding projects.
- Taken a manual backup of the certificates (if any) located at **<secure_connector_install_path>/etc/security** or **jre/lib**. You may want to replace/add your certificates after upgrade.

3. Click **Next** and choose an installation directory.
4. Click **Next**. A message window states the location where the target directory will be created.

Note: If an install directory exists, a warning message displays and you must confirm that you want to install and overwrite existing files.

5. Click OK.
6. Set up shortcut options to start, stop, and edit a Secure Connector.
 - a. Select one or both of the following options:
 - i. Create shortcuts in the Start menu.
 - ii. Create additional shortcuts on the desktop.
 - b. Select a program group from which you will access the shortcuts.
 - c. Choose to create shortcuts for the current user or all users.
7. Click Next. The installation progress displays.
8. Select a Secure Connector configuration file. If you have not already downloaded a Secure Connector configuration file, download one now.
9. Click Next.
10. For Windows installation, choose to install and run the Secure Connector as a Windows Service. If you choose install the Secure Connector as a Windows Service, you can control the Secure Connector using the Windows Services control panel (recommended). If you choose not to install and run the Secure Connector as a Windows Service, then the Secure Connector is installed as a Windows application. To run the Secure Connector as a Windows Service, you must specify the following service account information:
 - a. Service Start Mode
 - b. Service Account Domain
 - c. Service Account User
 - d. Service Account Password
11. Click Next. The installation is complete.
12. Click Done.

How to start and stop connectors on Windows (Service)

1. Open the Windows Services window: **Start > Control Panel > Administrative Tools > Services**.
2. Scroll down the list of services to locate the IBM Secure Connector service.
3. Right-click on the IBM Secure Connector service and select the appropriate command: Start, Stop, Pause, Resume, or Restart.

How to start and stop connectors on Windows (application)

1. Start the Secure Connector from either the Windows Start menu shortcut or desktop shortcut.
 - From the Windows Start button, select All Programs > IBM > Cast Iron Secure Connector <connector_name> > Start Secure Connector
 - From the Windows desktop, click the Start Secure Connector shortcut to start the Secure Connector.
2. Stop the Secure Connector from either the Windows Start menu shortcut or desktop shortcut.
 - From the Windows Start button, select All Programs > IBM > Cast Iron Secure Connector <connector_name> > Stop Secure Connector.
 - From the Windows desktop, click the Stop Secure Connector shortcut to stop the Secure Connector.

How to start and stop secure connectors on Linux

1. Start the Secure Connector from either the menu shortcut , desktop shortcut or command line. Choose one of the following options:
 1. Select <application> > IBM > Cast Iron® Secure Connector <connector_name> > Start Secure Connector.
 2. From the desktop, click the Start Secure Connector shortcut to start the Secure Connector.
 3. From the command prompt, enter `runclient_osgi.sh start`.
2. Stop the Secure Connector from either the menu shortcut, desktop shortcut, or command line. Choose one of the following options:
 - Select <application> > IBM > Cast Iron Secure Connector <connector_name> > Stop Secure Connector.
 - From the desktop, click the Stop Secure Connector shortcut to stop the Secure Connector.
 - From the command prompt, enter `runclient_osgi.sh stop`.

How to upgrade Secure Connectors

This topic provides information about upgrading Secure Connectors.

2. Create a new Secure Connector.
3. Download the latest version of the Secure Connector installer, based on your operating system. For example, Windows or Linux.
4. On a Windows or Linux machine, launch the Secure Connector installer. The Cast Iron Secure Connector wizard guides you through the upgrade process.

Note: On a 32-bit or 64-bit Linux machine, you must download a 32-bit or 64-bit Secure Connector, respectively, and complete the upgrade process. You cannot upgrade a 32-bit Secure Connector on a 64-bit Linux machine.

Note: If you already have a Secure Connector installation that is higher than or same as the latest version, a warning message states that you have an existing installation and alternatively you can upgrade the existing installation.

Note: You must stop the Secure Connector (if already started) before upgrading.

Note: Before you proceed with the Secure Connector upgrade process, ensure that you have:

- Stopped the Secure Connector and corresponding projects.
 - Taken a manual backup of the certificates (if any) located at **<secure_connector_install_path>/etc/security** or **jre/lib**. You may want to replace/add your certificates after upgrade.
5. Click the Upgrade option. The Select the installed path list box is displayed.
 6. Select the Secure Connector installed path, if it is displayed in the list box. Else, click Browse button to select the installed path.
 7. Click Next, then read and accept the licensing agreement.
 8. Click Next. The installation progress is displayed. A message is displayed stating that the installation has been completed successfully. The path to the installer program is also displayed.
 9. Click Done.
 10. Start the Secure Connector.

How to re-configure the Secure Connector

If you have a new Secure Connector configuration file and you want to re-configure your secure connector, complete the steps described below:

1. Launch the Secure Connector Configuration wizard. To launch the wizard:
 - a. Windows machine: Go to **Start > All Programs > b > Cast Iron Secure Connector <connector_name> > Secure Connector Configuration**.
 - b. Linux machine: Select **<application> > IBM > Cast Iron Secure Connector <connector_name> > Secure Connector Configuration**.
2. The Secure Connector configuration wizard guides you through the upgrade process.

3. Click **Next**. The current Secure Connector Configurations are displayed if the Secure Connector is already configured.
4. Update the Secure Connector configuration by clicking the **Previous** button and then browse and select the new **Secure Connector** file.
5. Click **Next** and verify the configuration settings.
6. Click **Next**.
7. Specify settings for a proxy server: Proxy Server, Proxy Port, Login ID, Login Password, and Retype Password. These parameters are only required if your network requires that the Secure Connector uses a proxy to connect to the Cast Iron® Cloud Gateway.
8. The **Create Vendor JAR** window is displayed. In the **Connector** column, select the connector for which you want to install additional files. Files that have already been installed are displayed in the **Installed Files** column.
9. Click **Add** and select the library files to upload. In the appliance, the valid files are .jar and .dll are the valid library file types. The files that you select are displayed in the **Files to Add** column.
10. Click Update.

The files that display in the **Files to Add** column are not committed until you click Update.

11. A confirmation dialog box is displayed. Follow the instructions in the dialog box.
12. Click **Next**. The upgrade is complete.
13. Restart the Secure Connector.

Creating a basic connection

By using the Cloud Integration basic connection feature you can connect to any on-premises system from the Bluemix cloud. Setting up a connection is the first step to establishing one or more tunnels to your on-premises endpoints.

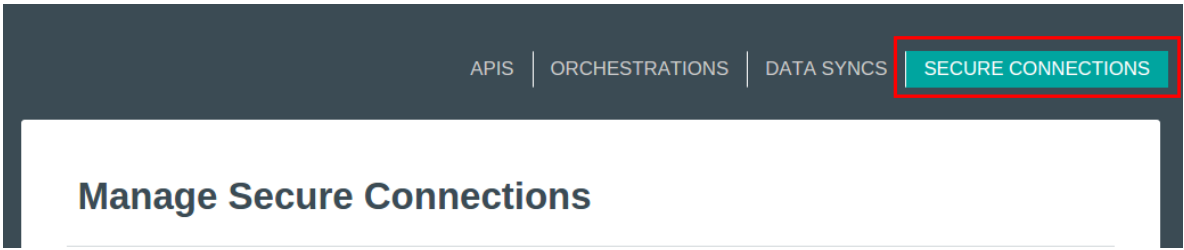
Because the enterprise server is behind a firewall, you can only access this server through a Secure Connector. Follow the procedure that is given below to install and run a basic Secure Connector.

After you create a Secure Connector in the cloud, you must configure a machine behind the firewall to facilitate communication between the Secure Connector and a specific endpoint behind the firewall. Use the Secure Connector installer to configure the machine behind the firewall.

Note: Only one basic connection can be created per Bluemix organization. Through this single connection, multiple endpoints can be attached, thus enabling access to more than one enterprise system.

To create a basic connection, complete the following steps.

1. Go to the Cloud Integration home page and click **SECURE CONNECTIONS**.



2. Enter a name for the new connection, then in the Basic column, click Add.

Add a new on-premises secure connection

Basic ⓘ Add	Standard ⓘ Add	DataPower ⓘ Add
------------------------------	---------------------------------	----------------------------------

3. A row is added listing your new basic connection. Click Install.

Type	Status	Driver	Action	Detail
Basic	Not Connected	INSTALL	⌂	➔

4. From the dialog that appears, select the installer for your system.

Note: Currently, only Ubuntu 64-bit v12.04 or higher is supported

Now, install the secure connection on **Basic_connector_on_phoenix1**.



Download the Installer and follow the instructions in the README to configure your connection

Linux 64-bit Installer

README

Now, upload the public key that you generated using your local machine

No file selected

Browse

5. Copy the downloaded file to the target system, then extract it and the archived file that it contains. You end up with a directory structure that looks like the following example:

```
drwxr-xr-x 2 vagrant vagrant 4096 Oct 2 01:43 ./
drwxr-xr-x 3 vagrant vagrant 4096 Oct 9 20:15 ../
-rw-r--r-- 1 vagrant vagrant 96 Oct 2 01:43 installfiles.md5sum
-rw-r--r-- 1 vagrant vagrant 20480 Oct 2 01:43 installfiles.tar
-rwxr-xr-x 1 vagrant vagrant 4767 Oct 2 01:43 install.sh*
```

6. Run `install.sh`.

Note: You are asked to provide the path to Java if it is not on your system path. The output looks similar to the following example:

```
vagrant@vagrant:/vagrant/connector/napinstall$ sudo ./install.sh
Checking MD5 sum...
Installing Native API Environment
useradd: user 'nativeapiadmin' already exists
Generating public/private rsa key pair.
Created directory '/home/nativeapiadmin/.ssh'.
Your identification has been saved in /home/nativeapiadmin/.ssh/id_rsa.
Your public key has been saved in /home/nativeapiadmin/.ssh/id_rsa.pub.
The key fingerprint is:
63:ed:69:b9:a8:ef:1c:ab:e1:01:0d:39:6d:5b:4a:6e nativeapiad-min@vagrant
id_rsa.pub for nativeapiadmin copied to current directory
stop nativeapimgmt if running
stop: Unknown instance:
nativeapimgmt start/running, process 2608
nativeapimgmt installed and started successfully
```

7. In the list of connectors, click the Refresh icon. The connection now appears as **Connected**.

Name	Type	Status	Driver	Action	Detail
Basic_connector_on_phoenix1 Modified -0 January 2014	Basic	Connected	DOWNLOAD		

- Optional: If you experience problems with getting the basic connection to show as Connected, try restarting the connection service on your remote machine. To do this, from the same directory from which you ran install.sh, run the following command:

```
sudo service nativeapimgmt restart
```

A public key named id_rsa.pub is listed. Upload this public key back to Cloud Integration, then close the dialog.

Now, upload the public key that you generated using your local machine

id_rsa.pub

Browse

The new basic connection is now successfully created and configured. Now, proceed to create a new endpoint. For more information, see *Creating an on-premises API endpoint*

How to manage DataPower Secure Connections

- On the **Manage Secure Connections** page, specify the name of your connection in the **Name your connection** field.

Figure 11: DataPower connection name

Manage Secure Connections

Add a new on-premises secure connection

Name your connection

Basic i

Add

Standard i

Add

DataPower i

Add

[Show Details](#)

- Click the **Add** button. Your enterprise secure connection name is displayed in the list of connectors.

Figure 12: List of connectors

Name	Type	Status	Driver	Action	Detail
BEN_530_729	Enterprise	Configured	CONFIGURE	Ⓢ	→
ben_test_0728	Standard	Not Connected	INSTALL	Ⓢ	→
dev-test	Standard	Not Connected	INSTALL	Ⓢ	→
cody_test	Standard	Not Connected	INSTALL	Ⓢ	→

3. Click **CONFIGURE**. The **Configure a DataPower Secure Connector** window is displayed.

Figure 13: Configure a DataPower Secure Connector window

Configure a DataPower Secure Connection ✕

Configure Mutual Authentication for access to
DP_Sec_Conn

:

Upload the **Certificate Authorities certificate** (*.pem file) that signed your DataPower server certificate

No file selected Browse

Now, install the **Cloud Integration Service certificate** to your DataPower

Download Certificate

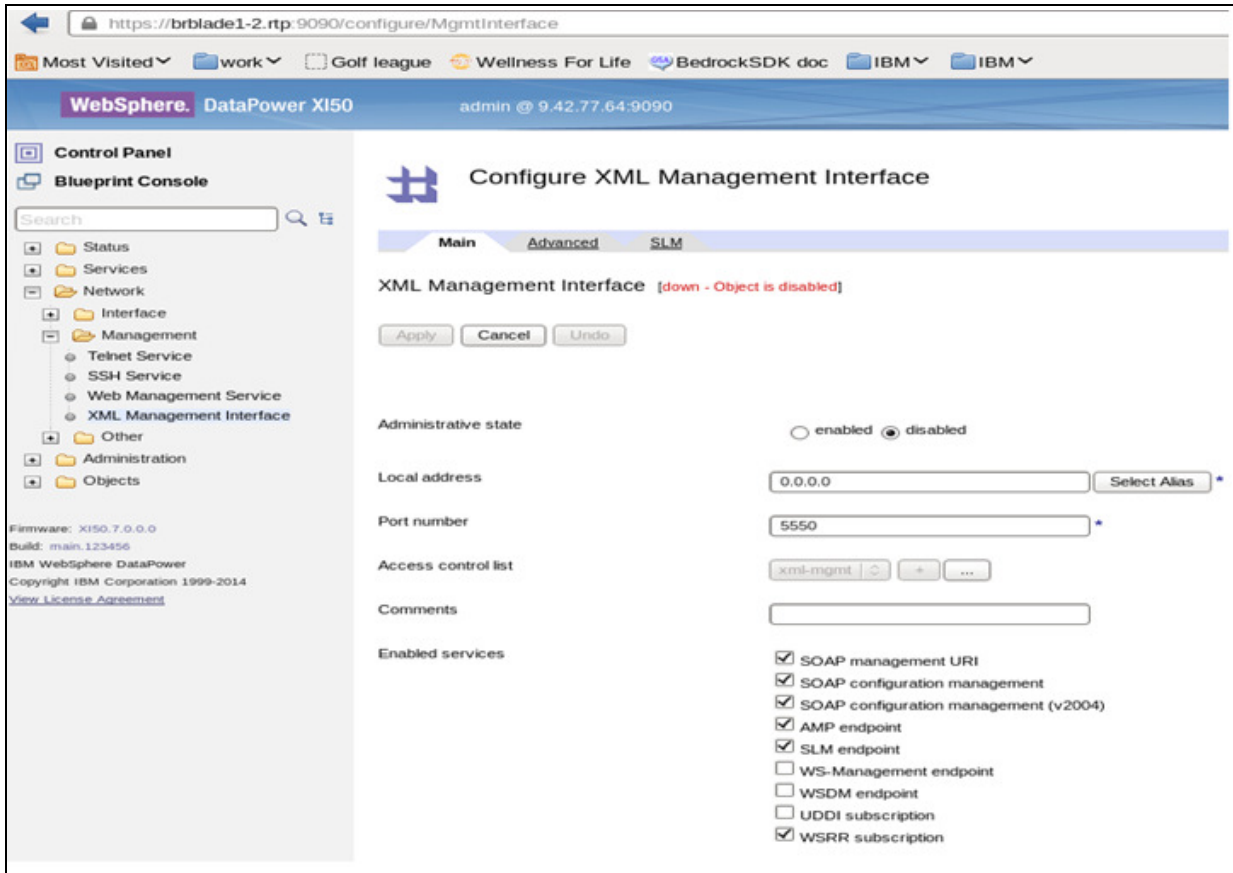
4. Configure the enterprise secure connector by completing the following:
 - a. DataPower hostname or IP address
 - b. Port number
 - c. Browse for the DataPower server certificate
 - d. Download the Cloud Integration service certificate
5. Click **Done** to return to the Cloud Integration home page.

Once the connection is established, you can successfully get started with creating APIs using the newly create DataPower connector.

How to create the Enterprise Secure Connector keys and certificates

1. Log on to IBM WebSphere DataPower.
2. Go to **Network > Management > XML Management Interface**.

Figure 14 : Configure XML Management Interface



3. On the **Configure XML Management Interface** page, specify the following:
 - a. Administrative state - Select the **enabled** option.
 - b. Local address
 - c. Port number
 - d. Access control list
 - e. Comments
 - f. Enabled services
4. Click **Apply**.
5. Go to **Administration > Miscellaneous > Crypto Tools**. The **Generate Key** page is displayed.

Figure 15: Generate Key page

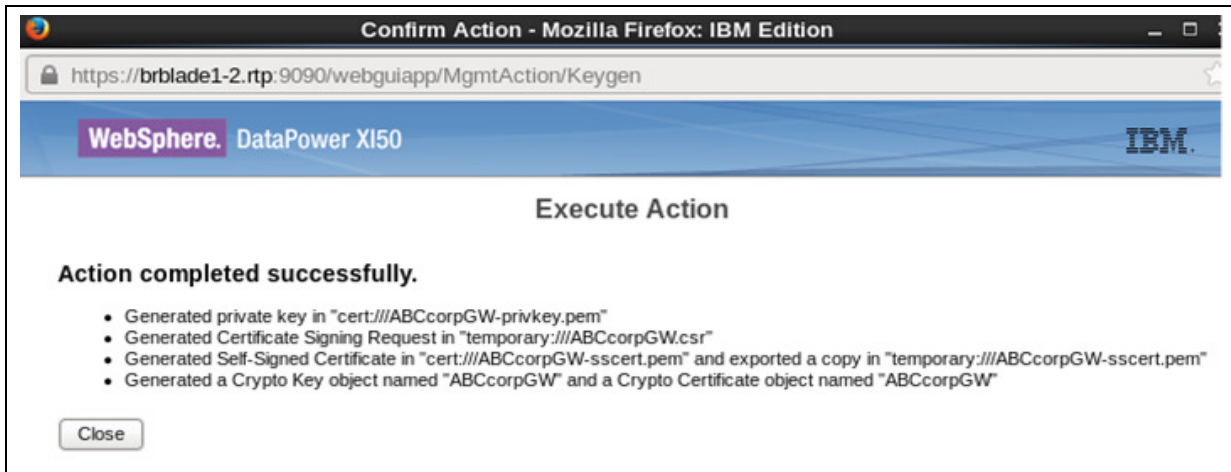
6. On the **Generate Key** page, specify the following:

- a. LDAP (reverse) Order of RDNs
- b. Country Name (C)
- c. State or Province (ST)
- d. Locality (L)
- e. Organization (O)
- f. Organizational Unit (OU)
- g. Organizational Unit 2 (OU)
- h. Organizational Unit 3 (OU)
- i. Organizational Unit 4 (OU)
- j. Common Name (CN)
- k. RSA Key Length
- l. File Name
- m. Validity Period
- n. Password
- o. Password Alias
- p. Export Private Key
- q. Generate Self-Signed Certificate
- r. Generate Key and Certificate Objects
- s. Object Name
- t. Using Existing Key Objects

7. Click **Generate**. The confirmation page is displayed which lists the following keys and certificates that are generated:

- Private Key
- Certificate Signing Request
- Self-Signed Certificate
- Crypto Key object and Crypto Certificate object

Figure 16: Confirmation page



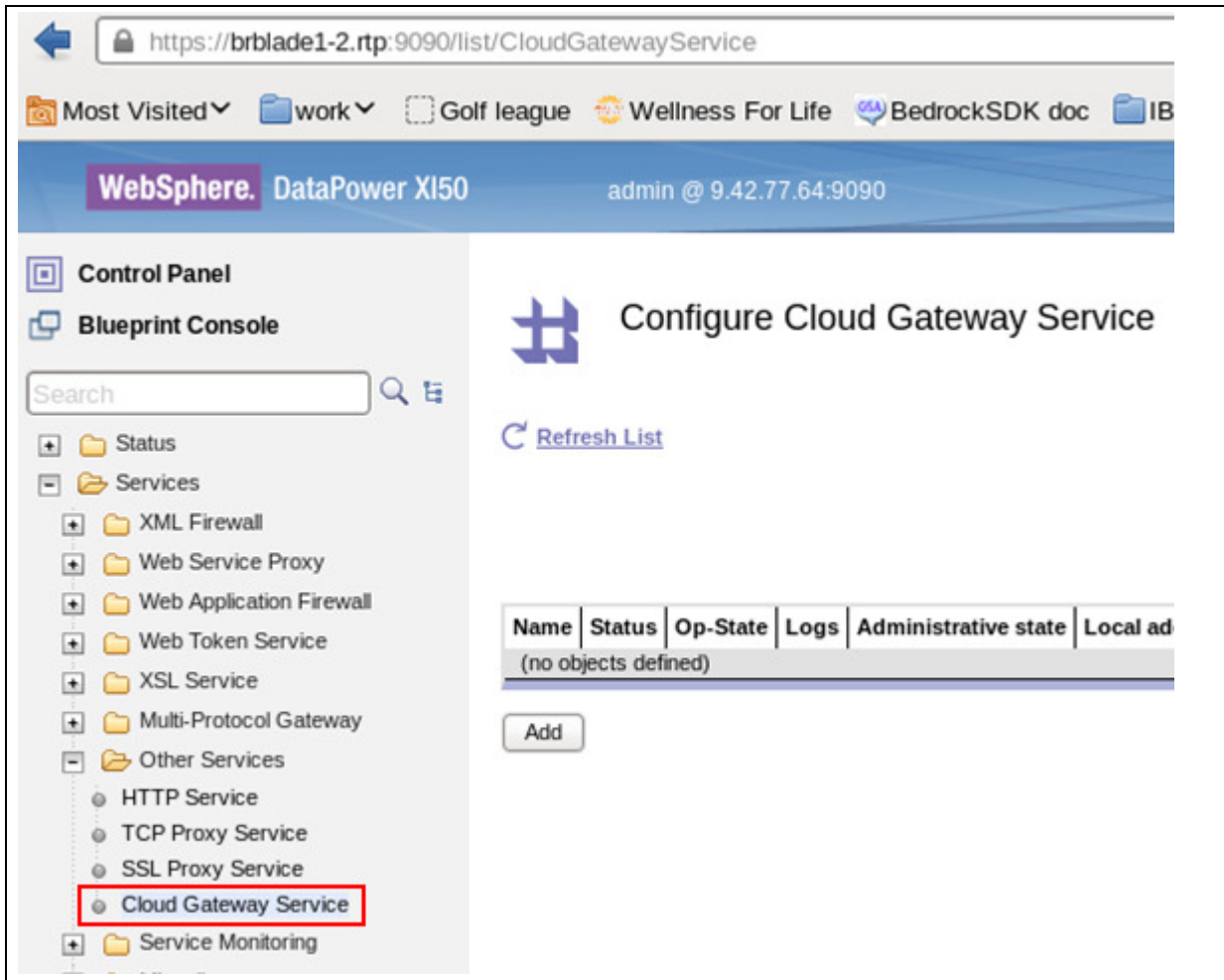
How to export self-signed certificates

1. Log on to IBM WebSphere DataPower.
2. Go to **Administration > Main > File Management**.
3. Browse for the self-signed certificate file.
4. In the **Actions** column, right-click **Edit**.
5. Click **Save Link As...** option. Save the file to your local folder.

How to create a cloud gateway service

1. On the IBM WebSphere Data Power page, go to **Services > Other Services > Cloud Gateway Service**.

Figure 17: Cloud gateway service



2. Click the **Add** button.

Figure 18: Add cloud gateway service

Main

Cloud Gateway Service

Apply Cancel

Name *

General

Administrative state enabled disabled

Comments

Service priority Normal ▾

Transaction timeout Seconds *

Cloud Gateway inbound connections

Local address Select Alias *

Local port *

SSL proxy profile (none) ▾ + ... *

Cloud-based client timeout Seconds *

Enterprise application connections


Allowed enterprise application connections

Service Name	Application host	Application port	Server-side timeout	Maximum connections	SSL proxy profile
(empty)					
					Add

*

3. Specify the required details.
4. In the Cloud Gateway inbound connections section, click the '+' symbol next to the **SSL proxy profile**. The SSL Proxy Profile page is displayed.

Figure 19: Configure SSL Proxy Profile page

 **Configure SSL Proxy Profile**

Main

SSL Proxy Profile

Name *

Administrative state enabled disabled

SSL Direction *

Reverse (Server) Crypto Profile + *

Server-side Session Caching on off

Server-side Session Cache Timeout seconds

Server-side Session Cache Size entries (x 1024)

Client Authentication Is Optional on off

Always Request Client Authentication on off

5. Specify the relevant details to configure the SSL proxy profile. In the **SSL Direction field**, select the **Reverse** option.
6. Click the '+' symbol next to the **Reverse (Server) Crypto Profile**. The **Crypto Profile** page is displayed.

Figure 20: Crypto Profile page

Configure Crypto Profile

Main

Crypto Profile

Apply Cancel

Name ABCcorpServerCryptoProfile *

Administrative state enabled disabled

Identification Credentials ABCcorpGWidcred | ⌵ + ...

Validation Credentials (none) | ⌵ + ...

Ciphers HIGH:MEDIUM:!aNULL:!eNULL:@STRE

Options

- Enable default settings
- Disable SSL version 2
- Disable SSL version 3
- Disable TLS version 1.0
- Permit insecure SSL renegotiation to a legacy SSL client
- Enable compression
- Disable TLS version 1.1
- Disable TLS version 1.2

Send Client CA List on off

7. On the **Crypto Profile** page, specify the relevant details, especially the mandatory fields. Click the '+' symbol next to the **Identification Credentials**. The **Crypto Identification Credentials** page is displayed.

Figure 21: Crypto Identification Credentials page

Configure Crypto Identification Credentials

Main

Crypto Identification Credentials

Apply Cancel

Name

Administrative state enabled disabled

Crypto Key + ...

Certificate + ...

Intermediate CA Certificate
 add + ...

8. On the **Crypto Identification Credentials** page, create the crypto identification credentials for the Cloud Gateway Service.
9. Specify the relevant details.
10. Select the correct **crypto key** and the corresponding **crypto certificate objects**.
11. Click **Apply**. The configuration is now added.
12. Now, validate the credentials. To validate the credentials, go back to the **Crypto Profile** page.
13. On the Crypto Profile page, click the '+' symbol next to the **Validation Credentials**.

Figure 22: Crypto Validation Credentials

Configure Crypto Validation Credentials

This configuration has been added and not yet saved.

Main

Crypto Validation Credentials: ABCcorpGWSValCred [up]

Apply Cancel Undo [Export](#) | [View Log](#)

Administrative state enabled disabled

Certificates (empty) [add] []

Certificate Validation Mode Match exact certificate or immediate issuer []

Use CRL on off

Require CRL on off

CRL Distribution Points Handling Ignore []

Check Dates on off

14. Specify the name of the object and click **Apply**.
15. Apply changes to the Crypto and SSL Profile objects and return to the Cloud Gateway Service object definition window.

Figure 23: Cloud gateway service object definition window

Enterprise application connections

Allowed enterprise application connections

Service Name	Application host	Application port	Server-side timeout	Maximum connections	SSL proxy profile
(empty)					
					<input type="button" value="Add"/>

16. Under Enterprise application connections, click **Add** and specify the required details.

Figure 24: Editing allowed enterprise application connections

WebSphere. DataPower XI50

Edit Allowed enterprise application connections

Service Name *

Application host *

Application port *

Server-side timeout Seconds *

Maximum connections *

SSL proxy profile

Note: Let the value of SSL Proxy Profile remain as none, unless you need to use SSL to the application.

17. Click **Apply** so that the changes to the allowed enterprise application connections are saved.

Figure 25: Applying changes to the gateway service and enterprise application connections

Cloud Gateway Service: ABCcorpGW [up]

Apply Cancel Delete Undo

General

Administrative state enabled disabled

Comments

Service priority Normal

Transaction timeout Seconds *

Cloud Gateway inbound connections

Local address *

Local port *

SSL proxy profile *

Cloud-based client timeout Seconds *

Enterprise application connections

Allowed enterprise application connections

Service Name	Application host	Application port	Server-side timeout	Maximum connector
ABCcorpSampleStaffDB	174.37.244.178	60004	0	100
<input type="text"/>				

*

18. On the object definition page of the gateway service, click **Apply** to apply the changes made to the gateway service and enterprise application connections.

How to add a cloud connector certificate

1. On the IBM WebSphere DataPower page, go to **Objects > Crypto Configuration > Crypto Service** and click **Add**. The Crypto Certificate page is displayed.

Figure 26: Crypto Certificate page

Configure Crypto Certificate

This configuration has been added and not yet saved.

Main

Crypto Certificate

Apply Cancel

Name TomCloud *

Administrative state enabled disabled

File Name cert:///tom-sscert.pem Details... Upload... Fetch... *

Password

Password Alias on off

Ignore Expiration Dates on off

2. On the Crypto Certificate page, specify details in the required fields. Specify the Name of the crypto object.
3. In the **File Name** field, click **Upload** to upload the certificate to the **cert:///** directory.
4. Click **Apply**.

How to add the cloud connector certificate to the validation credentials certificate list

1. On the IBM WebSphere Data Power page, go to **Objects > Crypto Configuration > Crypto Validation Credentials**.

Figure 27: Crypto Validation Credentials page

Control Panel
Blueprint Console

Search

- Status
- Services
- Network
- Administration
- Objects
 - Network Settings
 - Protocol Handlers
 - Service Configuration
 - XML Processing
 - JSON Processing
 - Web Services
 - Policy Configuration
 - Web Applications
 - Monitoring
 - Crypto Configuration
 - Cookie Attribute Policy
 - CRL Retrieval
 - Crypto Certificate
 - Crypto Certificate Monitor
 - Crypto Firewall Credentials
 - Crypto Identification Credentials
 - Crypto Key
 - Crypto Profile
 - Crypto Shared Secret Key
 - Crypto Validation Credentials**

The running configuration of the device contains unsaved changes. [Review](#)

Configure Crypto Validation Credentials

[Refresh List](#)

Name	Status	Op-State	Logs
ABCcorpGWSValCred	new	up	

[Add](#)

2. On the Crypto Validation Credentials page, in the **Certificates** field, search for the cloud crypto certificate by scrolling, and select the certificate.

Figure 28: Select the certificate

Crypto Validation Credentials: ABCcorpGWSValCred [up]

Apply Cancel Delete Undo

Administrative state enabled disabled

Certificates

TomCloud		
TomCloud	add	+

Certificate Validation Mode Match exact certificate or immediate issuer

Use CRL on off

Require CRI

3. Click **Add** to add the selected certificate to the certificate list.
4. Click **Apply**.

How to create an API from a database endpoint (DB2 or Oracle)

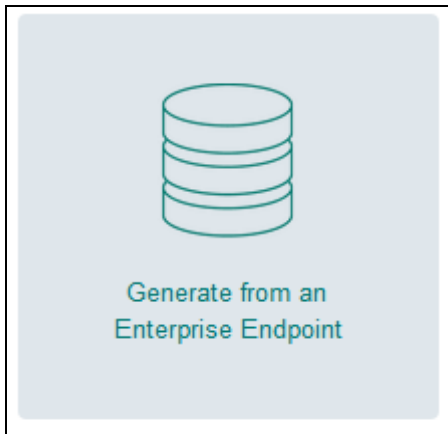
Note: Cloud Integration for Bluemix supports the following versions of Oracle:

- Oracle 11g (R1)
- Oracle 10g R1 and R2
- Oracle 9i R1 and R2
- Oracle 8i R2 (8.1.6) and R3 (8.1.7)

Complete the following steps to create an API from an Enterprise endpoint:

1. Go to the Cloud Integration home page. Click the **Create an API** button. The Create an API page is displayed.
2. Specify the name and description of your API.
3. Click **Generate from an Enterprise Endpoint**. The "Create an Enterprise API" page is displayed.

Figure 29. Generate from an Enterprise Endpoint tile



4. In the "Create an Enterprise API" page, you must begin by connecting to the database you want.
5. Click the **Select your Enterprise Endpoint** arrow and select **Add another Enterprise Endpoint**. The Connect to an Endpoint page is displayed.

Figure 30: Resources page

Create an Enterprise API

MyEntAPI
My enterprise API

Resources

Next, build your API by connecting to the database you want to use.

Database:

Select your Enterprise Endpoint

Cancel

6. Specify where the endpoint is by clicking the Where is the endpoint arrow and selecting one of the choices provided. A new field is displayed.

7. Add the service name.

8. Select either **DB2®** or **Oracle** and specify the following details:

Note: For on-premises applications, someone with a database administrator role on the customer's infrastructure, discusses with the integration developer about the number of users and the access permissions for each of the users. The administrator then grants those users the access required for APIs.

The administrator can also grant the users appropriate access and privileges to database assets. For example, the database administrator can grant access to particular database tables and certain operations the users can perform on the table.

- Database Name
- User name
- Password

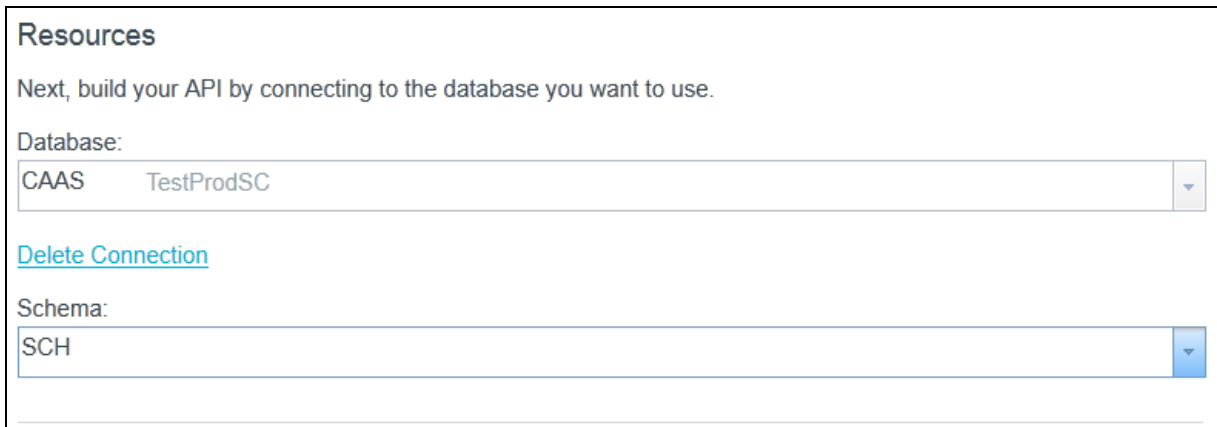
9. Click **Create**.

10. Go to the Cloud Integration dashboard and click **Create API**.

11. Provide a name and description.

12. Select an existing on-premises database and select the required schema.

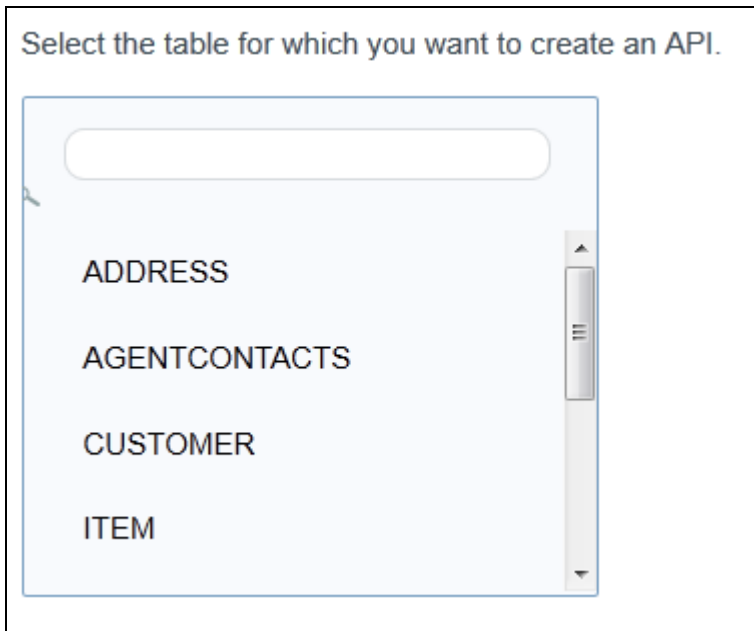
Figure 33: Schema



The screenshot shows a web interface titled "Resources". Below the title is the instruction: "Next, build your API by connecting to the database you want to use." There are two dropdown menus. The first is labeled "Database:" and contains the text "CAAS TestProdSC". Below it is a blue link labeled "Delete Connection". The second dropdown menu is labeled "Schema:" and contains the text "SCH".

13. After you select the schema, select the table for which you want to create an API.

Figure 34: Table



The screenshot shows a web interface with the heading "Select the table for which you want to create an API." Below the heading is a light blue rounded rectangle containing a search input field at the top. Below the search field is a scrollable list of table names: "ADDRESS", "AGENTCONTACTS", "CUSTOMER", and "ITEM". A vertical scrollbar is visible on the right side of the list.

14. Select the resources that you want to make available when you created the API. You can include resources such as Get, Post, Put, and Delete.

Note: Click the copy icon available next to the resource names to view and copy the API URL.

Figure 35: Property names and resources

The screenshot shows an API configuration interface. On the left, a sidebar lists property names: ADDRESS (highlighted), AGENTCONTACTS, CUSTOMER, and ITEM. On the right, a table lists resources for the ADDRESS resource:

Method	Property Name	Data Type	Include
GET	UNITNO	xs:integer	Yes
GET	STREET	xs:string	
GET	CITY	xs:string	
GET	COUNTRY	xs:string	Yes
GET	CONTACTNUMBER	xs:integer	Yes
GET	POSTALCODE	xs:integer	Yes
GET	ADDRESSID	xs:integer	Yes

Below the table, the text reads: "Now you can choose which resources you want to make available".

Four resource configurations are shown, each with a method button, a URL, and an "Include" toggle:

- GET** (blue button): `database/c564796e-c40e-4ba6-8ad5-1be376ba8daa/SCH/Tables/ADDRESS?{columnname1=columnvalue1&columnname2=columnvalue2&..}` (Include:)
- GET** (blue button): `database/c564796e-c40e-4ba6-8ad5-1be376ba8daa/SCH/Tables/ADDRESS` (Include:)
- POST** (green button): `database/c564796e-c40e-4ba6-8ad5-1be376ba8daa/SCH/Tables/ADDRESS?{columnname1=columnvalue1&columnname2=columnvalue2&..}` (Include:)
- PUT** (yellow button): `database/c564796e-c40e-4ba6-8ad5-1be376ba8daa/SCH/Tables/ADDRESS?{columnname1=columnvalue1&columnname2=columnvalue2&..}` (Include:)
- DELETE** (red button): `database/c564796e-c40e-4ba6-8ad5-1be376ba8daa/SCH/Tables/ADDRESS?{columnname1=columnvalue1&columnname2=columnvalue2&..}` (Include:)

15. Click **Create API**. The API is successfully created with the required resources.

16. Expand the resources to view the JSON examples. Sample JSON script:

GET:

Request JSON:

Response JSON:

```
{
  "rows": {
```

```

        "row": [
            {
                "CUSTOMERID": "xs:integer",
                "NAME": "xs:string",
                "ADDRESSID": "xs:integer",
                "TITLE": "xs:string",
                "FLAG": "xs:string"
            }
        ]
    }
}

POST:

Request JSON:
{
    "rows": {
        "row": [
            {
                "CUSTOMERID": "xs:integer",
                "NAME": "xs:string",
                "ADDRESSID": "xs:integer",
                "TITLE": "xs:string",
                "FLAG": "xs:string"
            }
        ]
    }
}

Response JSON:
{
    "rows": {
        "rowsModifiedCount" : "xs:int"
    }
}

PUT:

Request JSON:
{
    "rows": {
        "row": [
            {
                "CUSTOMERID": "xs:integer",
                "NAME": "xs:string",
                "ADDRESSID": "xs:integer",
                "TITLE": "xs:string",
                "FLAG": "xs:string"
            }
        ]
    }
}

Response JSON:
{
    "rows": {
        "rowsModifiedCount" : "xs:int"
    }
}

DELETE:

```

```
Request JSON:
Response JSON:
  {
    "rows": {
      "rowsModifiedCount" : "xs:int"
    }
  }
```

17. Go to the service instance home page where you can see the newly created API.
18. Click **Create an API** to create a new API for yet another database. Repeat the steps from Step 11 onwards.

You have successfully created an API from a database.

Note:

- The default response format is JSON. You can specify an alternative format in the Accept header. The following response is supported: format:application/json; application/xml; application/text.
- The default request format is JSON. For PUT and POST operations, you can specify an alternative format in the Content-Type header.
- If you are connecting to DB2 on z/OS, and if the value of the rowsModifiedCount is -2 when you invoke the REST API for PUT, POST, and DELETE operations, this means that the operation was successful but the exact number of rows modified was not returned by the JDBC driver.

Now you can publish your API. For more information, see *Publishing an API as a private service*.

How to create an Enterprise API from an SAP endpoint

To create an API from a database, go to the Create an Enterprise API page and complete the following steps:

1. Click the **Generate from an Enterprise Endpoint** tile. (See Figure 11).
2. The Resources page is displayed. Here, you must connect to the SAP database to create your SAP API. Click the dropdown **Select your Enterprise Endpoint**. (See Figure 12).
3. Select **Add another Enterprise Endpoint**. The Connect to a Database or SAP page is displayed.

Figure 38: Connect to SAP

x

Connect to a Database or SAP

DB2OracleSAP

Where is the database?

SC1 ▼

Hostname or IP Address

System Number

Client

Username

Password

CancelConnect

4. Specify the following details:
 - a. Name of the connector
 - b. Host name or IP Address
 - c. System Number
 - d. Client
 - e. User name
 - f. Password

Once you click connect, the database is added to the list of databases. A new field called Schema is added.

5. Select the required schema from the list.

Figure 39: Schema - 2

Resources

Next, build your API by connecting to the database you want to use.

Database:

SAP TestProdSC

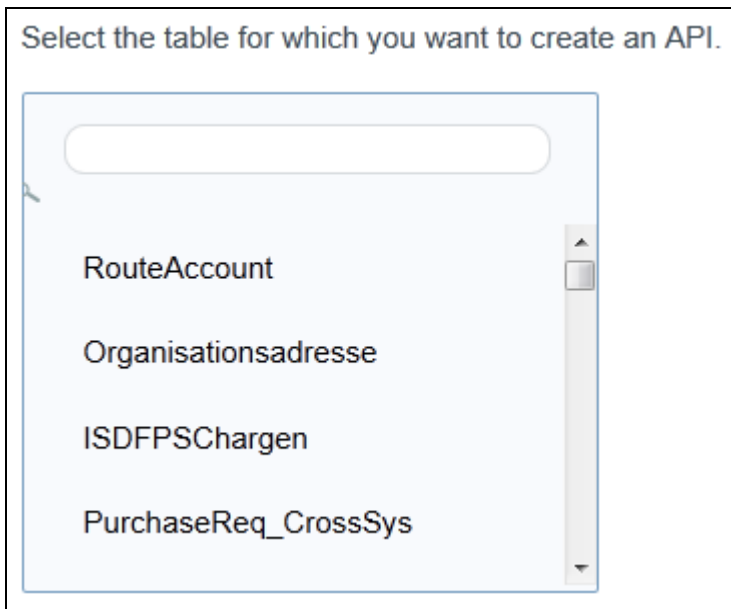
[Delete Connection](#)

Schema:

bapi

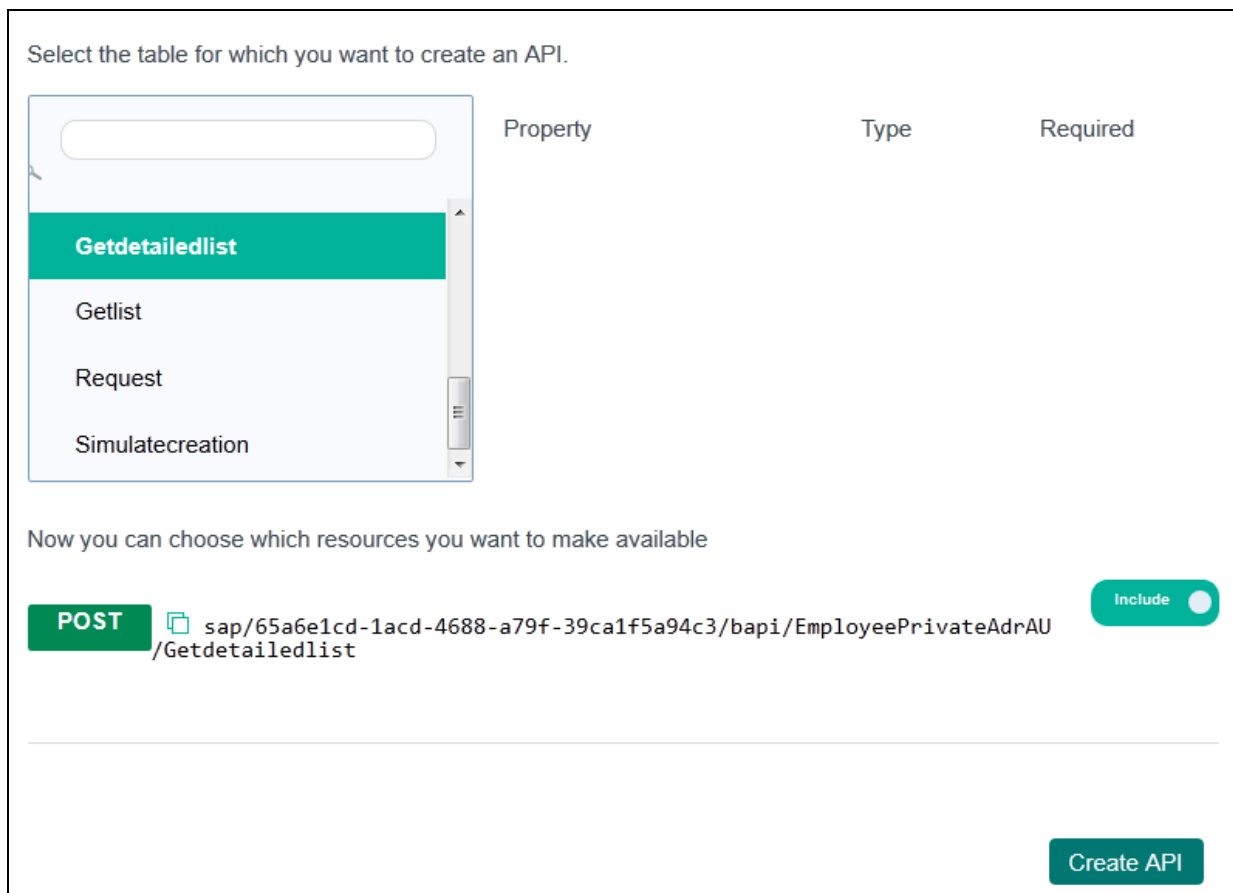
6. Select the table for which you want the API created.

Figure 40: Table




7. Drill-down to another level of selecting tables.

Figure 41: Choosing Resources



8. Choose the resources that you want to make available.

Note: Click the copy icon  available next to the resource names to view and copy the API URL.

9. Click **Create API**.

10. The API is created. The details that are displayed includes the API_SECRET key, the resources and the JSON examples.

The generated API_SECRET key is displayed in the API details.

11. Click the copy icon to view the URL, the API_SECRET key and the user name and password to access the API. (See Figure 18).

Your newly created SAP API tile is now present in the Cloud Integration home page.

How to create an API from Cast Iron Live Orchestrations

Follow the procedure listed below to create an API from Cast Iron Live Orchestrations.

Note: This feature works only with the latest version of Cast Iron Live. Contact IBM Support to get access to the latest version.

1. Go to the Cloud Integration home page. Click the **Create an Enterprise API** button. The Create an Enterprise API page is displayed.
2. Specify the name and description of your API.
3. Click the **Create from Cast Iron Live Orchestrations** button. The Cast Iron Live sign-on page is displayed.

Figure 42: Create from Cast Iron Live Orchestrations button

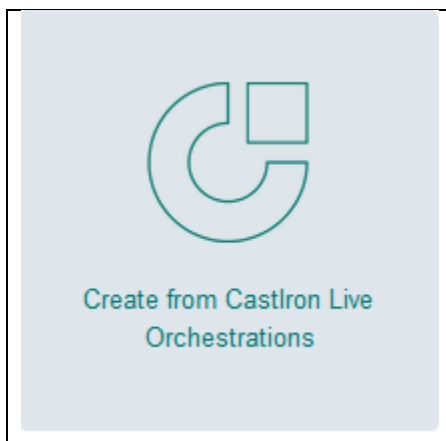


Figure 43: Cast Iron Live sign-on page

live-orchestration

my live orchestration

First, Sign In to Cast Iron Live.
(Available for Cast Iron v7.0 users only)

Forgot your [password?](#)

Evaluation Version

Production Version

Don't have a Cast Iron Live account? [Sign up](#) for one now.

4. Specify your Cast Iron Live logon credentials and click **Sign In**. The Resources page is displayed.

Figure 44: Cast Iron Live Orchestrations resources page

The screenshot shows a web interface titled "Create an Enterprise API". Below the title, there is a section for "live-orchestration" with the text "my live orchestration". A horizontal line separates this from the "Resources" section. Under "Resources", there is a prompt: "Next, build your API by describing the verbs, location, and parameters." Below this, there is a dropdown menu for the resource, currently showing "httpTest1:1.0:Default:Orchestration1 (Development)". To the left of the dropdown is a button labeled "Get" with a small downward arrow. To the right of the dropdown are two circular buttons, one with a plus sign and one with a minus sign. Below the resource dropdown is a text area with the placeholder text "Describe what happens when this is called". Below the text area are two links: "Add parameters" and "Add code sample". At the bottom of the form are two buttons: "Cancel" and "Create API".

5. Select the resources that you want to make available when creating the API. You can include resources such as Get, Post, Put, and Delete.
6. Select the name of your project available in the cloud management console.
7. Add the parameters and code sample, if required.
8. Click Create API. The API is successfully created with the required resources.

You have successfully created an API from Cast Iron Live Orchestrations.

How to create an API from Bluemix applications

Follow the procedure listed below to create an API from Cast Iron Live Orchestrations

1. Go to the Cloud Integration home page. Click the **Create an Enterprise API** button. The Create an Enterprise API page is displayed.
2. Specify the name and description of your API.
3. Click the **Create from Bluemix apps** button. The Resources page is displayed.
4. Select the resources that you want to make available when creating the API. You can include resources such as Get, Post, Put, and Delete.
5. Select the a Bluemix application from the list.
6. Add the parameters and code sample, if required.
7. Click Create API. The API is successfully created with the required resources.

You have successfully created an API from Bluemix applications.

How to sign up for Cast Iron Live Evaluation version

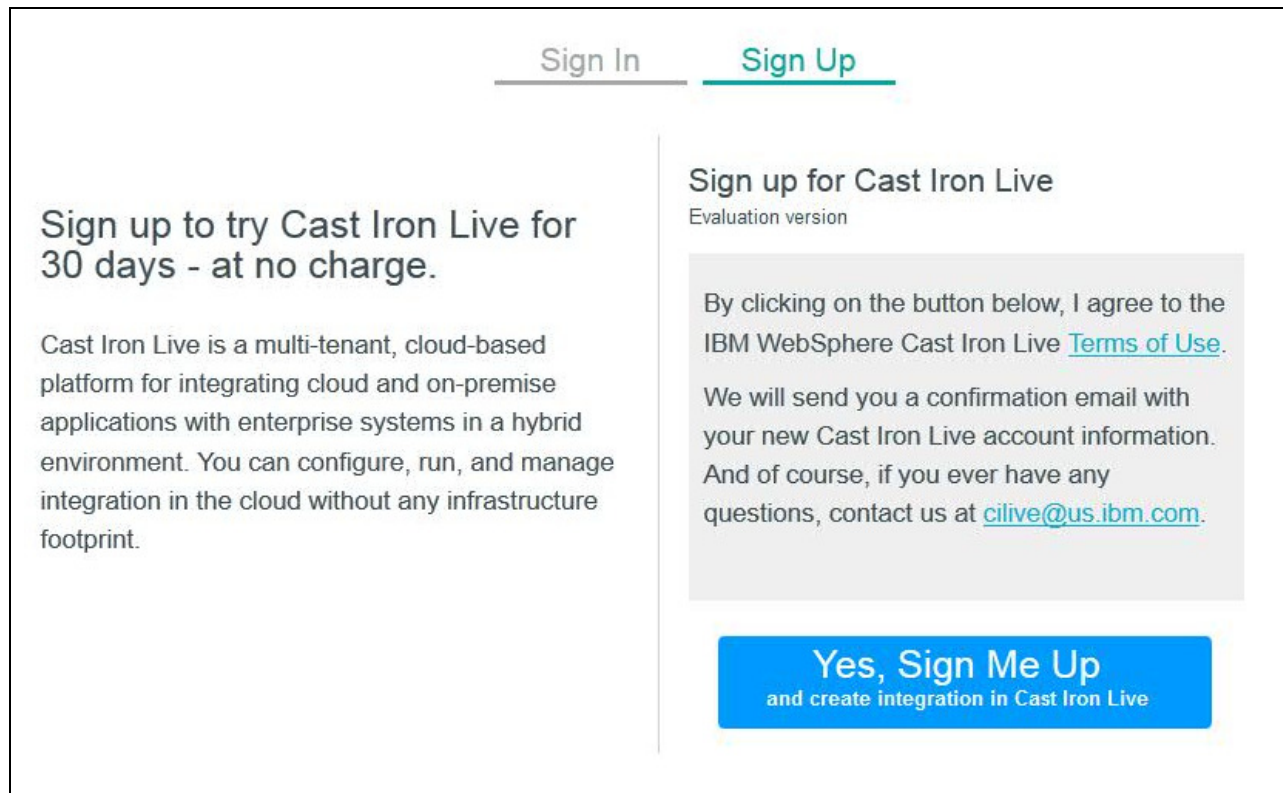
Cast Iron Live is a multi-tenant, cloud-based platform for integrating cloud and on-premise applications with enterprise systems in a hybrid environment. You can configure, run, and manage integration in the cloud without any infrastructure footprint.

Each organization must have an admin who can sign-up and obtain credentials to a Cast Iron Live evaluation cloud. The admin is authorized to create accounts for users within the organization. If the other users within the organization try to sign-up, they will be provided with the contact details of the admin.

Complete the follow steps to sign-up with Cast Iron Live Evaluation version:

1. Go to the Cloud Integration home page. Click the **Create Integration** button. The Cast Iron Live sign on page is displayed.

Figure 45: Cast Iron Evaluation version Sign Up page



2. Click the **Sign Up** link.

The sign-up instructions are displayed, if you are the first to sign-up from your organization (for example the Admin).

3. Click **Yes, Sign Me Up**.

An email is sent to the mail address that your organization is registered with. The email contains the instructions on getting started with Cast Iron Live. This includes the username and password, system requirements, and quick tips.

Figure 46: Sample welcome mail

Welcome to WebSphere Cast Iron Live 30 day trial account! It is our pleasure to welcome you to the IBM WebSphere Cast Iron family. This email contains information you will need to ensure success with your new WebSphere Cast Iron Live session.

Getting Started:
To access your Cast Iron Solution, please go to: **The Cloud Integration service home page from Bluemix. Click the Create Integration button. The Cast Iron Live sign on page is displayed** and use the following credentials to login to Cast Iron Live:

User Name: admin@eval1034879 **Password:** changelt!

System requirements:

OS - Windows 7 or Windows XP
Java - version 6

Quick Tips:

Please refer to following quick tips which will help you to understand Cast Iron Live interface usability and functionality:

From Cast Iron Live Home tab you perform 3 simple steps to get your integration projects up and running quickly!

From the **Create** tab get started with creating a new project by selecting a template from a repository which covers a wide range of applications. You may also choose to create a project from scratch by clicking the option on top left. When you finish creating the project and save it, your project and your view are moved to the **Modify** tab.

From the **Modify** tab you can perform following actions for each project:

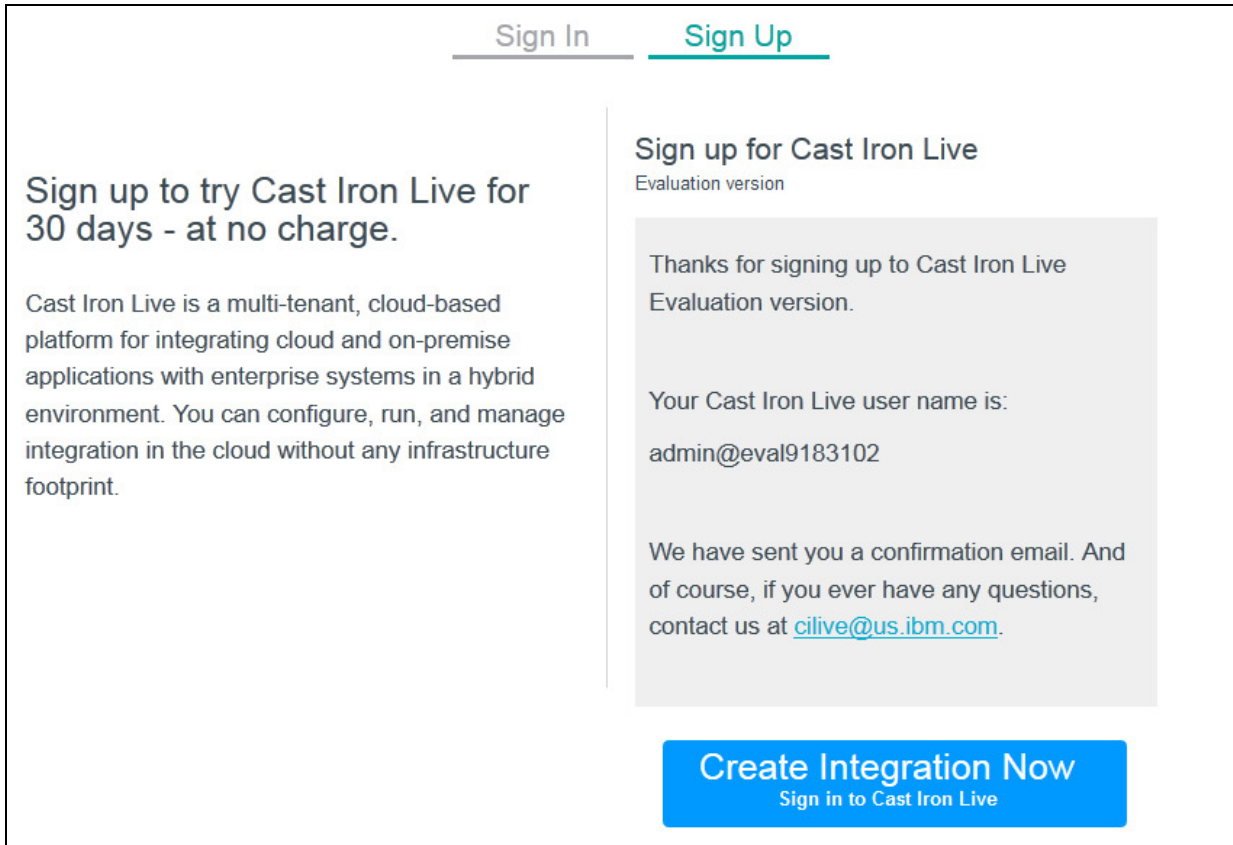
- 1 Edit Project
- 2 Publish
- 3 Delete
- 4 Create New Version

To edit a project, select **Edit Project**, and select Edit Project in Designer

. It launches a web start Cast Iron Studio. The Studio development environment has the same functionality as the on-premise installed version of the WebSphere cast Iron Studio. You can build/modify integration flows, create/modify orchestration built around several activities that define the flow of data.

The Sign up for Cast Iron Live page displays the following:

Figure 47: Sign up page



4. Click **Create Integration Now** to sign in to Cast Iron Live with your newly received user credentials.

Creating an on-premises API endpoint



Before you begin, create a basic connection. For more information, see *Creating a basic connection*

To create a new on-premises API endpoint, complete the following steps.

1. Go to the Cloud Integration home page and click **SECURE CONNECTIONS**.

Manage Secure Connections

- Find the basic connection that you created previously, and click the **Arrow** icon in the "Detail" column.

Name	Type	Status	Driver	Action	Detail
Basic_connector_on_phoenix1 Modified -0 January 2014	Basic	Connected	DOWNLOAD		

- Click **Add an Endpoint** to launch the wizard.

 [Add an Endpoint](#)

- In the "Connect to an Endpoint" section, click **API**.

Connect to an Endpoint

API

User defined

- Complete the remaining fields, including the host name and port of your on-premises system, the relative URI to the root of your API, and any credentials required to access the API.

Where is the endpoint?

MTH_Prod_Basic_Test (CONNECTED) ▾

What is the API type?

REST SOAP

server3.internal.mycompany.com : 9080

/api/v2

(optional) Secret API key

apiuser@mycompany.com

How do you want to secure access to this Endpoint?

Privately Available only to authorized applications
Publicly Available to all applications

Requires mutual TLS/SSL authentication

Use TLS/SSL to encrypt incoming connections

Cancel

Create

- Specify how you want to secure access to the endpoint. For more information about the different security options, see *Configuring your application for TLS (public) or mutual TLS (private) access*
- Click Create.

Your endpoint is created and is displayed in the list of endpoints.

Configuring your application for TLS (public) or mutual TLS (private) access

Securing the communication channel and access control to your application is an important step to providing security for your overall API.

Warning: When you created the endpoint, if you selected the Publicly security option and cleared the Use TLS/SSL to encrypt incoming connections check box, this section does not apply. There will be no security enforced by Cloud Integration between you and your on-

premises endpoint. Data (including authentication details) flowing between IBM® Bluemix and any application using this endpoint will be sent unencrypted.

To use an endpoint that is created publicly with TLS, complete the following steps.

Note: Endpoints that are created with this access type encrypt data between the application client and the on-premises connection, but does not authenticate clients.

1. After you create the endpoint, click the **i** icon to the right of the endpoint name. A window opens.
2. Click the Download Certificate button. A compressed file begins to download.
3. Extract the contents of the compressed file. It contains a certificate file.
4. Add the certificate file to your applications trust store. This process varies depending upon which technology the application is built. Consult the documentation for your application platform about how to add a certificate to the applications trusted store.
5. Continue developing and deploying the application. Communication between the application and the endpoint is secured with TLS.

To use an endpoint that is created Privately (mutual TLS is required), complete the following steps.

Note: Endpoints created with this access type encrypt data between the application client and the on-premises connection, and also require a certificate to authenticate access to the endpoint.

6. After you create the endpoint, click the **i** icon to the right of the endpoint name. A window opens.
7. Click the Download Certificate button. A compressed file begins to download.
8. Extract the contents of the compressed file. It contains a certificate file.
9. Add the server-cert file to your application's trust store.
10. Add the client-cert and private-key files to your application's key store. The application will need to use the key store to authenticate against the endpoint, and the trust store to trust the endpoint's certificate. This process varies depending upon which technology the application is built. Consult the documentation for your application platform for details on how to manage key and trust stores.
11. Continue developing and deploying the application. Communication between the application and the on-premises endpoint is secured with mutual TLS authentication.

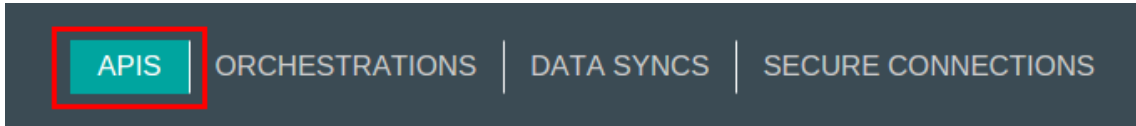
Creating a new Cloud Integration REST API that links to an existing on-premises API

Before you begin, ensure you complete the following tasks.

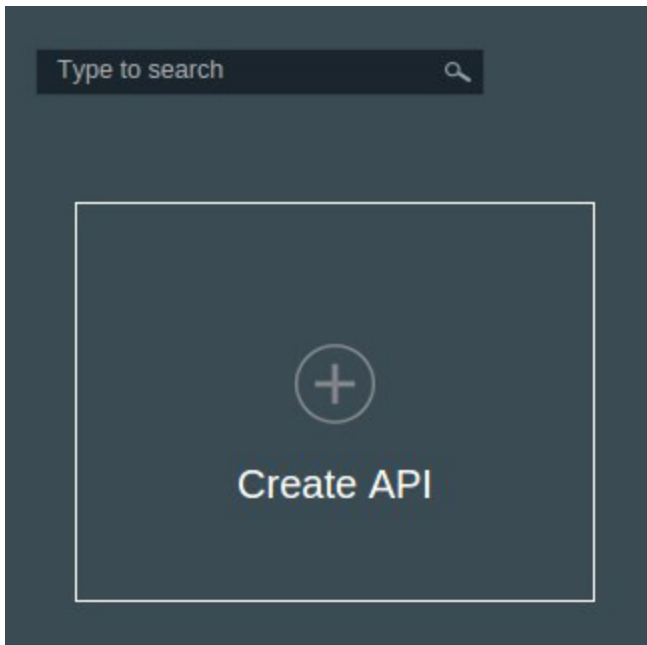
- Creating a basic connection
- Creating an on-premises API endpoint

To create a new API, complete the following steps.

1. Go to the Cloud Integration home page and click **APIS**.



2. Click **Create API**.



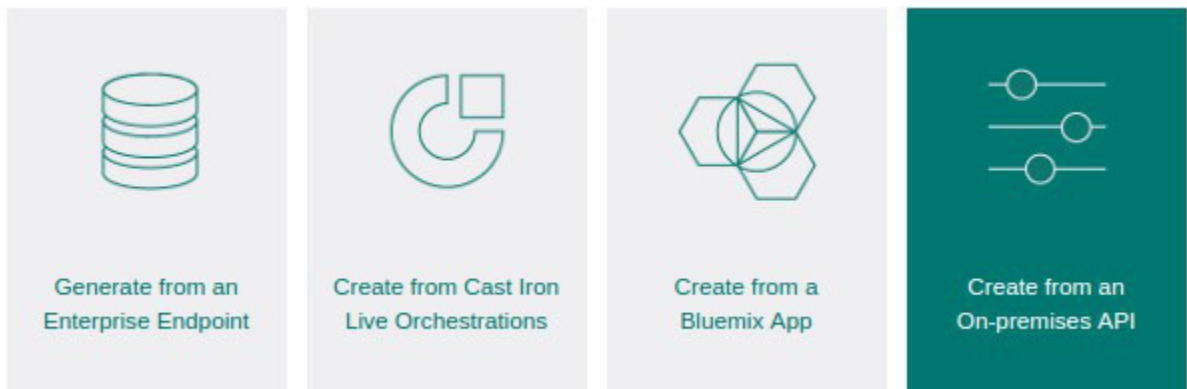
3. Enter a short title for the new API and a longer more detailed description.

How can others in this organization find your API?

Vacation planner

Provides APIs for tracking employee vacation. Allows applications full access to and management of vacation data.

4. Select **Create from an On-premises API**.



5. Click the arrow and select the endpoint that has access to your on-premises API.

First, select the endpoint that accesses your on-premises API

6. The system prompts you to upload a file defining the API. For SOAP APIs, upload a WSDL file; for REST APIs, upload a Swagger file. Select **Load**.

Load a File

Note: Currently, Cloud Integration cannot retrieve a WSDL or Swagger definition from your on-premises API.

7. Cloud Integration parses the resource file and then prompt you for any additional required resource files. Upload any requested resources until all entries show as "Complete". Click **Next**.

Load supporting sources

Filename	Load
/person	Complete

Cancel

Back


Next

8. Review the APIs loaded from the resource files for accuracy.

Next, review the operations that will be made available to your new API


List of Operations

GET

 /person

Parameter	Type	Required
-----------	------	----------

POST

 /person

Parameter	Type	Required
-----------	------	----------

body	Person	Yes
------	--------	-----

- Optional: Upload any supporting API documentation in one of the accepted formats listed.

Upload **additional documents** for your API.

Drop files here, or [browse](#)

Supported formats: doc, wpd, pdf, txt, jpeg, gif, tiff, png
(Max upload size: 16MB)

- Click **Create API** to complete the process.

Your API is created and is available for use, SDK generation, access credentials and so on.

Publishing an API as a private service

Publishing an API as a Private Service exposes your new API as a service in the Bluemix catalog under the Private Services category.

Note: Only organization managers can publish an API as a private service.

To publish your API as a Private Service, complete the following steps.

- Go to the Cloud Integration home page and select the API you want to work with. The "View API" window opens.
- Click **Publish API**.

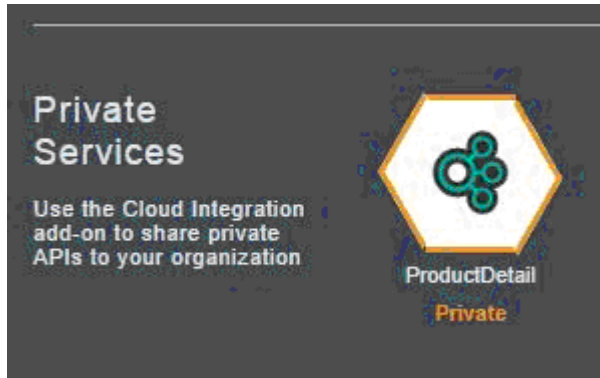
ProductDetail
Oct 8, 2014

Product Summary from on-premises Product database

Access SDKs

Download an SDK that will allow you to easily work with this API in your chosen programming language

Private Services are only visible to members of your organization. Your Private Service shows as a new tile in the Bluemix catalog and are easily distinguishable by the orange border and the orange "Private" tag at the bottom of the tile.

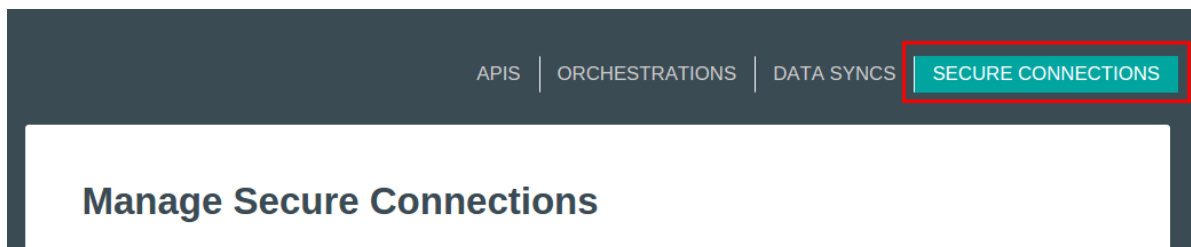


Connecting to a user-defined endpoint

Cloud Integration supports connections to any type of on-premises system. When you create a user-defined endpoint, a host name and port are defined and these can be used to access on-premises systems. This type of endpoint can be secured privately with mutual TLS authentication.

Before you begin, you must create a basic connection. For more information, see *Creating a basic connection*.

1. Go to the Cloud Integration home page and click **SECURE CONNECTIONS**.



2. Find the basic connection that you created and click the **Arrow** icon in the "Detail" column.

Name	Type	Status	Driver	Action	Detail
Basic_connector_on_phoenix1 Modified -0 January 2014	Basic	Connected	DOWNLOAD	🔄	➔

3. Click **Add an Endpoint** to start the wizard.

⊕ Add an Endpoint

4. Select the **User defined** endpoint type.

Connect to an Endpoint ⊗

API **User defined**

5. Enter the host name and port for the on-premises system.

Where is the endpoint?

MTH_Prod_Basic_Test (CONNECTED) ▾

achilleus.internal.mycompany.com : 636

6. Specify how you want to secure access to the endpoint. For more information about the different security options, see *Configuring your application for TLS (public) or mutual TLS (private) access*.

How do you want to secure access to this Endpoint?

Privately
Available only to authorized applications

Publicly
Available to all applications

Requires mutual TLS/SSL authentication

Use TLS/SSL to encrypt incoming connections

7. Click **Create**.

Your endpoint is created successfully and appears in the list of endpoints.

How to create an integration (click-through)

Follow the procedure listed below to create an integration:

Note: This feature works only with the latest version of Cast Iron Live. Contact IBM Support to get access to the latest version.

1. Go to the Cloud Integration home page. Click the **Create Integration** button. The Cast Iron Live sign on page is displayed.

Figure 48: Cast Iron Live Sign In page

←

Sign In Sign Up

Sign In to Cast Iron Live
(Available for Cast Iron v7.0 users only)

Username

Password

Forgot your [password?](#)

Evaluation Version

Production Version

Sign In Cancel

Don't have a Cast Iron Live account? [Sign up](#) for one now.

Sign in with your Cast Iron Live account to link your account to Bluemix and orchestrate integration for your API.

2. Specify your Cast Iron Live logon credentials and click Sign In. The Cloud Management Console is displayed.

Note: On the sign-on page, you can either choose the **Evaluation Version** or the **Production Version**. If you do not have a Cast Iron Live account, you can even sign-up for an account.

3. Create your required Cast Iron orchestrations. The orchestrations must be wrapped in HTTP Receive-Response request.

You have successfully created a Cast Iron Live integration.

How to view an API

The APIs that you have created are displayed as API tiles in the service console.

1. Click the newly created API tile to view details about the API. The description and API resources are displayed.
2. Click the arrow beside the tile to view the parameters and examples related to an API resource.

How to call an API

The APIs created are saved in the VCAP_SERVICES as well. Complete the following steps to use the API URL

1. If you are using Node.js as the runtime for your Bluemix application, use the following sample code for the Cloud Integration API:

Figure 49: Sample Node.JS code

```
if (process.env.VCAP_SERVICES) { //USE FOR CLOUDFOUNDRY DEPLOYMENT
  var env = JSON.parse(process.env.VCAP_SERVICES);
  env_cloudint = env['CloudIntegration-0.1'][0].credentials;
}

var user = env_cloudint .userid;
var password = env_cloudint .password;
for (i=0; i<env_cloudint .apis.length; i++) {
  if (env_cloudint .apis[i].name == api_name)
    myurl = url.parse(env_cloudint .apis[i].url);
}

if (myurl == null)
  console.log ('API Url is not defined!');
options = {
  host: myurl.hostname,
  port: 443,
  path: myurl.path,
  secureProtocol: 'SSLv3_method',
  headers: {'Authorization': 'Basic ' + new Buffer (user+'!'+password).toString('base64')}}
};
//Set the API method to be invoked
options.method = 'GET';

//Make a http request

var request = https.request(options, function(response) {
  //Response handling here
});
```

2. Make an HTTPS call using the API URL.

Note: You can now provide a different username and password to access the on-premise database. This is an optional feature. These credentials will override the credentials provided during the API creation. You will need to pass the Username and Password as HTTP headers. If there is no USERNAME and PASSWORD passed in the headers while invoking the API, then the credentials specified during API creation will be used.

Downloading an SDK

The option to download an SDK is shown when you view a Cloud Integration API. SDKs are dynamically generated from the API definition and enable you to quickly make use of the API in your chosen language. When you download an SDK, it's easier for you to use Cloud Integration APIs from applications.

Before you begin, ensure that you have completed the following tasks.

- Created an API. For more information, see *Creating Cloud Integration APIs*.

SDKs can be downloaded from APIs generated from an endpoint (DB2®, Oracle, or SAP), Cast Iron® Live orchestration, or a Bluemix application. The option to download an SDK is made available when you view the API either through the Cloud Integration Add-on, or as an API in the Private API catalog.

SDKs are available in three languages: JavaScript (running node.js), Java, and Ruby (for example, running on Sinatra or Rails). The SDK is dynamically generated at the time of download and based on the definition of the API. The download is in the form of a compressed file.

The SDK has functions that represent each REST verb available in the API and depending on the language, contain models to represent request and response JSON. Other helper functions are included which allow for authentication and headers to be set in the request. Each SDK contains an example file to demonstrate how these functions and models should be used.

To download an SDK, complete the following steps.

1. Go to the Cloud Integration home page and select the API you want to work with. The "View API" window opens.
2. Select one of the following options to download a compressed file.
 - **Download JavaScript SDK**

The downloaded compressed file includes a number of directories and files.

- Prerequisites for using the client code are defined in the package.json file and can be installed by using the node package manager (npm).
- The client code is contained in the lib directory and is exported through the main.js file in the top-level directory.

The downloaded SDK is used as a node package and therefore more SDKs can be downloaded and combined into one package. This is done by exporting the client code in a single main.js file, updating the package.json file accordingly and uploading the package to your npm repository.

An example.js file is included in the top-level directory that shows you how to use the methods that are defined in the client code.

- **Download Java SDK**

The downloaded compressed file includes a number of JAR files. These are prerequisite JAR files and the precompiled SDK client code, which is contained in -client.jar and -client-sources.jar. The former file can be imported into your Java client code as usual, and is precompiled with debug information, whereas the latter file is provided to help with debugging applications that are using this SDK.

An Example.java file is included in the compressed file and demonstrates the expected use of the functions that are provided in the client code. The client code is made up of a number of packages, all beginning with com.ibm.cloudintegration.

com.ibm.cloudintegration.api

This package contains the class `api_name`, which is where methods that drive the REST API are defined. This is the main class that any code developed with the SDKs will use, and is the only class that is directly used in the `Example.java` file.

com.ibm.cloudintegration.api_name

This package contains classes that represent the structure of request and response objects and are named according to the REST operation they correspond to. Where models are a complex type, submodels are also generated and named accordingly.

com.ibm.cloudintegration.common

This package contains utility classes that are used by the other classes that are listed above. The `ApiInvoker` class includes methods for more advanced configuration of sending REST calls.

- **Download Ruby SDK**

The downloaded compressed file contains a set of Ruby files that can additionally be used as a RubyGem.

The main API class is defined in a Ruby file under the `lib` directory, and that file is named after the API that the SDK was generated from. Classes that represent the structure of request and response objects are stored in the `models` directory, and are named according to the REST operation they correspond to.

Where models are a complex type, submodels are also generated and named accordingly.

An example Ruby script, `example.rb`, is provided in the top-level directory that demonstrates how to use the downloaded Ruby SDK to interact with the API.

3. Depending on your browser settings, a file downloads to your download directory. The file is named using the convention: `api_name-client-language-date-timestamp.zip`
4. Navigate to the downloaded file and import it into your development environment.

Next, you can publish your API to the private API catalog. For more information, see *Publishing an API as a Private service*.

Integrate the SDK into your Bluemix application.

Creating a Data Sync

Create a data sync to replicate data from an on-premises database to a cloud database so that you can use the data in a cloud environment. After you create a data sync, you have faster access to data from cloud applications and you do not have to rely on a live connection to an on-premises database. After you create the data sync and migrate data, the data sync keeps

your cloud table in sync with your on-premises table. Further updates to the on-premises table are automatically propagated to the cloud table.

Before you begin, ensure that you complete the following tasks.

- Create a Bluemix application and bind SQL database service to it.
- Create a Cloud Integration service. For more information, see *Creating an Integration*
- Install a standard secure connector. For more information, see *Managing secure connections – Standard Secure Connector*
- Add an endpoint to the on-premises database. For more information, see *Creating an API from a database endpoint (DB2 or Oracle)*
- Connect a Cloud Integration Add-on. For more information, see *Connecting an Add-On*.

To create a data sync, complete the following steps.

1. Go to the Cloud Integration home page and click **DATA SYNC**.
2. Click **Add a new Data Sync**. The "Create a Data Sync" window is displayed.
3. Provide a name and description for your data sync.
4. Click **Next**.

Create a Data Sync



Provide a name and description for this data sync

NEWDATASYNC

new data sync

Cancel

Next

5. Select an option from each of the following lists.
 - Select your database
 - Select schema
 - Select a table

The table that you choose here, is the one that is going to be replicated in the cloud.

Note: You might be prompted for an on-premises database password.

Restriction: The table that you select must have a primary key and the primary key can only contain one column. The following data types are not supported.

- DBCLOB
- GRAPHIC
- LONG VARGRAPHIC
- VARGRAPHIC
- DB2SQLSTATE
- BLOB
- CLOB
- XML

NEWDATASYNC

On-premises Data

Select on-premises database, schema and table

USERDB	DB2	▼
KMCMUL		▼
CLOUDSTOCK30		▼

Cancel

Back

Next

6. Click **Next**. The "Cloud Data" window opens.
7. Select an option from each of the following lists.
 - Select a Bluemix application. Only Bluemix applications that have SQL database bindings are displayed here.
 - Select your database. Select the database that your application is bound to.
 - Select schema
 - Select table

Only valid tables that match the source table structure are listed here.

Cloud Data

Next, determine which cloud database you want to use.

KATAPP

SQL Database-9c

DB2INST1

Select a table with matching structure

Select a table with matching structure

Add a table with matching structure

CLOUD30

Type: Synchronise from on-premises table to cloud table

Frequency: The source table will be monitored every 5 minutes for changes that need to be migrated to the target table.

Previous

Create Data Sync

Cancel

8. If your table is not listed, you can create one by clicking **Add a table with matching structure** and entering a name for it.

Cloud Data

Next, determine which cloud database you want to use.

KATAPP

SQL Database-9c

DB2INST1

Add a table with matching structure

Settings

Type: Synchronise from on-premises table to cloud table

Frequency: The source table will be monitored every 5 minutes for changes that need to be migrated to the target table.

Create a table

Enter a table name

Create

Cancel

Previous

Create Data Sync

Cancel

A table is automatically created for you.

9. Click **Create Data Sync**. A new window opens that displays the database assets.
10. Required: Before a data sync can work, extra assets (control tables and/or table triggers) must be created on the source and target databases. Click **Download** to download the SQL script that you need to run on the source database. To run the SQL script, complete the following steps:
 - a. To open a DB2® command line, click **Start > All Programs > IBM DB2 > DatabaseInstance > Command Line Tools**, and select **Command Window**. (Where *DatabaseInstance* is your DB2 instance name, which by default is, DB2COPY1 (Default)). A DB2 - CLP window opens.
 - b. To connect to the required database, enter the following command:

```
db2 CONNECT TO DatabaseName
```

where:

- *DatabaseName* is the name of your on-premises database that you selected in [Step 5](#).
- c. Enter the following command to run the script.

```
db2 -f "C:\Users\admin\Downloads\batch.sql" -t -v -s
```

where:

- *batch.sql* is the name of your SQL batch script.
- -f tells the command line processor to read command input from a file instead of from standard input.
- -t tells the command line processor to use a semicolon (;) as the statement termination character.
- -v tells the command line processor to echo command text to standard output.
- -s tells the command line processor to stop execution if errors occur while it is running commands in a batch file or in interactive mode.

The tables are created automatically.

11. When you create all the assets that are required for a data sync to work, click **Migrate now**. Clicking **Migrate now** copies the existing data in the source table to the target table. For large data tables, the migration might take a while.

Warning: The data is migrated via an unsecured channel.

12. You can view the status of the migration, on the "My Data Sync" page.

When you are finished, the "Status" panel displays that the "Overall data sync state" is connected and the "Last sync" field displays a time and date.

Status

Overall data sync state: Connected

Last sync: 07/10/2014 16:40 (Migration completed)

[Refresh status](#)

Settings

Type: Synchronise from on-premises table to cloud table

Frequency: The source table will be monitored every 5 minutes for changes that need to be migrated to the target table.

If the "Status" panel displays that the on-premises database is unreachable, check that the secure connector is connected and then check that your on-premises database is running.

The data that is synchronized can be accessed by any application and not just the application whose binding credential is being used to synchronize data.

Updates that you make to the on-premises database are reflected in the cloud table automatically. Click **Refresh status** to get information about when this update took place and whether it was successful or not.

Attention: There is a 5-minute time delay from the moment that any updates made to the on-premises table are reflected in the cloud table.

For multiple tables, you need to create a data sync for each table. To do this, go to Step 2.

Request payload generation rules

When you provide a JSON input to the created API, conform to the following rules:

1. Case 1: Passing a string value:
Accepted format: "name" : "Varun" (double quotes only).
2. Case 2: Passing inexplicit quotes in a string:
Accepted format: "name" : "'Varun'" (The single quotes will be replaced with explicit double quotes).
3. Case 3: Passing a number starting with a leading 0:
Accepted format: "id" : "_\$_0001\$_" (Any string prefixed with `_s_` and post-fixed with `$_` will be treated as a number).
4. Case 4: Passing a number:
Accepted format: "age" : 30

Troubleshooting

Note: *If you face any issues while creating or accessing an API, check the Cast Iron Secure Connector logs or check the DataPower logs for the DataPower secure connector. For Cast Iron Integration, check the system logs of Cast Iron in your account.*

You may encounter the exception listed below, with Secure Connector, while connecting to the database:

```
com.approuter.maestro.sdk.mpi.ActivityFailedException(Connection error while
getting connection out of the pool.  SQLSTATE: HY000 ERRORCODE: 0 Error
Message: [CastIron Systems][DB2 JDBC Driver]End of stream was detected on a
read.
[CastIron Systems][DB2 JDBC Driver]End of stream was detected on a read.)
FaultTime:2014-05-29T10:48:40.661Z JobID:BB85F0FE3B3EBD8FDEBEDAC80AEF4956
ActivityID:4
.....
com.approuter.module.database.protocol.DBConnectionException: Connection
error while getting connection out of the pool.  SQLSTATE: 08001 ERRORCODE: 0
Error Message: [CastIron Systems][DB2 JDBC Driver]Error establishing socket
to host and port: 9.1.2.1.:50000. Reason: Connection timed out: connect

    at
com.approuter.module.database.protocol.ConnectionManager.getConnection(Connec
tionManager.java:178)
    at
com.approuter.module.database.activity.GenericQueryActivity.process(GenericQu
eryActivity.java:140)
    at
com.approuter.module.database.activity.GenericQueryActivity.execute(GenericQu
eryActivity.java:107)
    at
com.approuter.agent.container.sdk.AgentServiceManager.execute(AgentServiceMan
ager.java:132)
    at
com.approuter.agent.container.protocol.AgentActivityHandler.execute(AgentActi
vityHandler.java:346)
    at
com.approuter.agent.container.serviceimpl.AgentServiceImpl.execute(AgentServi
ceImpl.java:105)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:88)
        at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.
java:55)
            at java.lang.reflect.Method.invoke(Method.java:618)
            at
com.sun.xml.ws.api.server.InstanceResolver$1.invoke(InstanceResolver.
```

This exception occurs if the database is not reachable from the Secure Connector. Verify that the database is accessible from the server machine where the Secure Connector is running.

AGENT_LOCKED error when starting the standard Secure Connector

Symptoms

If the standard secure connector is deleted from the Cloud Management Console, and a new one with the same name is created, the AGENT_LOCKED error will be thrown when starting the Secure Connector on the local machine.

If a secure connector is deleted and a new one with the same name is then created from Cloud Management Console, the secure connector configuration for the new instance is different from the original one. At this time, if the secure connector isn't updated in user's environment, the connector will still use the previous authentication key to connect to Cast Iron Live. Thus, the agent is locked by Cast Iron Live.

Resolving the problem

To resolve the problem, complete the following steps:

1. Contact IBM Cloud Integration Support to unlock the secure connector from the backend.
2. Re-configure the Secure Connector.
 - a. Download a new Secure Connector configuration from Cloud Management Console.
 - b. On Windows, run Secure Connector Configuration from the start menu to import the downloaded configuration.
 - c. On Linux, run agent_configurator.jar from *sc_installation_path*/utils to import the downloaded configuration.
3. Restart the Secure Connector.

Common errors thrown when you invoke the API

1. You do not have permission to access this API.

Solution: Ensure that you pass in the header "API_SECRET" the value for which will be specified in the API details when you created them.

2. (The content of elements must consist of well-formed character data or markup.)

FaultTime:2014-07-22T14:51:45.524+05:30

JobID:8BDB16C024B8B4EFCF04FC6E024C4F72 ActivityID:6796

Solution: Verify the input HTTP header value for Content-Type. If it is 'application/xml', then the input XML input is not well formed. If it is 'application/json', then the input JSON input is not well formed.

3. (An invalid XML character (Unicode: 0x7b) was found in the prolog of the document.)

FaultTime:2014-07-22T14:52:26.972+05:30 JobID:AAB2AC0C96CF7BB297579046BB5D4DD3 ActivityID:6796

Solution: Verify the input HTTP header value for Content-Type. If it is 'application/json', the JSON is not well formed or could not be converted into a valid XML. If it is 'application/xml', then the input XML input is not well formed.

- [Troubleshooting](#)

[Services](#)>[Add-Ons](#)>[Cloud Integration](#)

Getting started with Cloud Integration service

Last Updated: 10/16/2014

TUTORIALS AND SAMPLES

- [IBM Cloud Integration for Bluemix User Guide](#)

RELATED LINKS

- [IBM Bluemix Pricing Sheet](#)
- [IBM Bluemix Prerequisites](#)

IBM® Cloud Integration for Bluemix™ service enables you to integrate cloud services with enterprise systems of record. This service exposes the backend systems of record as ReST APIs to be used by applications.

You can create an instance of the Cloud Integration service through Bluemix. This service can be used to interact with the backend system of record through APIs.

Connecting an Add-On

If you are logging on to Bluemix for the first time, before you proceed with creating a Cloud Integration service, you must connect to an add-on service.

1. From the IBM Bluemix dashboard, click the Connect an Add-On link. The Add-Ons tiles are displayed on the Add-Ons page.
2. Click the Cloud Integration add-on tile. The Cloud Integration Pick a Plan page displays the Plan, Features, and the Price for the add-on service.
3. Select the App and Selected Plan and click Create.

You have successfully connected to the Cloud Integration add-on.

Before you begin

Before you begin exploring the Cloud Integration add-on service, you must ensure that you have the required secure connections and that you have created integrations. Once these two are in

place, you can start creating Cloud Integration Enterprise APIs, Cast Iron Live Orchestration APIs, or Bluemix Apps APIs.

Managing the Enterprise Secure Connector

The Enterprise (DataPower) Secure Connector leverages and on-premise DataPower deployment as a secure gateway connection between backend resources (behind the enterprise firewall) and Bluemix applications ensuring high-availability/fail over and load balancing requirements. Follow the procedure given below to add and configure an enterprise Secure Connector.

1. On the Manage Secure Connections page, specify the name of your connection in the Name your connection field.
2. Click the Add and Configure button. Your enterprise secure connection name is displayed in the list of connectors.
3. Configure the enterprise secure connector by completing the following:
 - a. DataPower hostname or IP address
 - b. Port number
 - c. Browse for the DataPower server certificate
 - d. Download the Cloud Integration service certificate
4. Click **Done** to return to the Cloud Integration home page.
- 5.

Once the connection is established, you can successfully get started with creating APIs using the newly create DataPower connector.

Creating the Enterprise Secure Connector keys and certificates

1. Log on to IBM WebSphere DataPower.
2. Go to Network > Management > XML Management Interface.
3. On the Configure XML Management Interface page, specify the following:
 - a. Administrative state - Select the enabled option.
 - b. Local address
 - c. Port number
 - d. Access control list
 - e. Comments
 - f. Enabled services
4. Click Apply.
5. Go to Administration > Miscellaneous > Crypto Tools. The Generate Key page is displayed.
6. On the Generate Key page, specify the following:
 - a. LDAP (reverse) Order of RDNs
 - b. Country Name (C)
 - c. State or Province (ST)
 - d. Locality (L)
 - e. Organization (O)
 - f. Organizational Unit (OU)
 - g. Organizational Unit 2 (OU)
 - h. Organizational Unit 3 (OU)

- i. Organizational Unit 4 (OU)
 - j. Common Name (CN)
 - k. RSA Key Length
 - l. File Name
 - m. Validity Period
 - n. Password
 - o. Password Alias
 - p. Export Private Key
 - q. Generate Self-Signed Certificate
 - r. Generate Key and Certificate Objects
 - s. Object Name
 - t. Using Existing Key Objects
7. Click Generate. The confirmation page is displayed which lists the following keys and certificates that are generated:
- o Private Key
 - o Certificate Signing Request
 - o Self-Signed Certificate
 - o Crypto Key object and Crypto Certificate object

Exporting self-signed certificates

1. Log on to IBM WebSphere DataPower.
2. Go to Administration > Main > File Management.
3. Browse for the self-signed certificate file.
4. In the Actions column, right-click Edit.
5. Click Save Link As... option. Save the file to your local folder.

Creating a Cloud Gateway Service

1. On the IBM WebSphere Data Power page, go to Services > Other Services > Cloud Gateway Service
2. Click the Add button.
3. Specify the required field details.
4. In the Cloud Gateway inbound connections section, click the '+' symbol next to the SSL proxy profile. The SSL Proxy Profile page is displayed.
5. Specify the relevant details to configure the SSL proxy profile. In the SSL Direction field, select the Reverse option.
6. Click the '+' symbol next to the Reverse (Server) Crypto Profile. The Crypto Profile page is displayed.
7. On the Crypto Profile page, specify the relevant details, especially the mandatory fields. Click the '+' symbol next to the Identification Credentials. The Crypto Identification Credentials page is displayed.
8. On the Crypto Identification Credentials page, create the crypto identification credentials for the Cloud Gateway Service.
9. Specify the relevant details.
10. Select the correct crypto key and the corresponding crypto certificate objects.
11. Click Apply. The configuration is now added.

12. Now, validate the credentials. To validate the credentials, go back to the Crypto Profile page.
13. On the Crypto Profile page, click the '+' symbol next to the Validation Credentials.
14. Specify the name of the object and click Apply.
15. Apply changes to the Crypto and SSL Profile objects and return to the Cloud Gateway Service object definition window.
16. Under Enterprise application connections, click Add and specify all the required details.

Note: Let the value of SSL Proxy Profile remain as `none`, unless you need to use SSL to the application.

17. Click Apply so that the changes to the allowed enterprise application connections are saved.
18. On the main page of the gateway service, click Apply to apply the changes made to the gateway service and enterprise application connections.

Adding a cloud connector certificate

1. On the IBM WebSphere Data Power page, go to Objects > Crypto Configuration > Crypto Service and click Add.
2. On the Crypto Certificate page, specify details in the required fields. Specify the Name of the crypto object.
3. In the File Name field, click Upload to upload the certificate to the `cert:///` directory.
4. Click Apply.

Adding the cloud connector certificate to the validation credentials certificate list

1. On the IBM WebSphere Data Power page, go to Objects > Crypto Configuration > Crypto Validation Credentials.
2. On the Crypto Validation Credentials page, in the Certificates field, search for the cloud crypto certificate by scrolling, and select the certificate.
3. Click Add to add the selected certificate to the certificate list.
4. Click Apply.

Managing secure connections - Standard Secure Connector

Since the enterprise server is behind a firewall, you will be able to access this server only through a Secure Connector. Follow the procedure given below to install and run a basic Secure Connector.

- Ensure that you have the root level permissions to install the Secure Connector on a Linux machine.
- Ensure that you have the administrator permissions to install the Secure Connector on a Windows machine.

After you create a Secure Connector in the cloud, you must configure a machine behind the firewall to facilitate communication between the Secure Connector and a specific endpoint behind the firewall. Use the Secure Connector installer to configure the machine behind the firewall. Note: The machine on which you choose to run the installer must have access to the endpoint. You do not have to run the installer on the same machine as the endpoint.

To download the Secure Connector installer:

1. On the Cloud Integration home page, create a new secure connection by clicking the Manage secure connections button. If you already have a secure connection, it will be listed with the existing Secure Connectors.
2. If you do not have a secure connection, specify the name and description of your secure connection.
3. Click the Install button of the Secure Connector that you would like to install. The links to the latest configuration and readme files are displayed. The download links for Windows (32-bit and 64-bit architecture) and Linux (32-bit and 64-bit architecture) are also displayed.

Note: The configuration file is used to configure the Secure Connector. The readme file contains the links to the Secure Connector downloads and MD5SUM to verify the installers.

4. Click the appropriate link based on your machine configuration.

To install the Secure Connector:

5. Launch the Secure Connector installer you downloaded.
 - o `windows-secure-connector-installer-32bit.exe`
 - o `windows-secure-connector-installer-64bit.exe`
 - o `linux-secure-connector-installer-32bit.sh`
 - o `linux-secure-connector-installer-64bit.sh`

The Secure Connector Installer Wizard is displayed.

6. Click Next then read and accept the licensing agreement.
7. Click Next and choose an installation directory.
8. Click Next. A message window states the location where the target directory will be created.

Note: If an install directory exists, a warning message displays and you must confirm that you want to install and overwrite existing files.

9. Click OK.
10. Set up shortcut options to start, stop, and edit a Secure Connector.
 - a. Select one or both of the following options:
 - Create shortcuts in the Start menu.
 - Create additional shortcuts on the desktop.
 - b. Select a program group from which you will access the shortcuts.
 - c. Choose to create shortcuts for the current user or all users.

11. Click Next. The installation progress displays.
12. Select a Secure Connector configuration file. If you have not already downloaded a Secure Connector configuration file, download one now.
13. Click Next.
14. For Windows installation, choose to install and run the Secure Connector as a Windows Service. If you choose install the Secure Connector as a Windows Service, you can control the Secure Connector using the Windows Services control panel (recommended). If you choose not to install and run the Secure Connector as a Windows Service, then the Secure Connector is installed as a Windows application. To run the Secure Connector as a Windows Service, you must specify the following service account information:
 - Service Start Mode
 - Service Account Domain
 - Service Account User
 - Service Account Password
15. The Create Vendor JAR window is displayed. In the Connector column, select the connector for which you want to install additional files. Files that have already been installed are displayed in the Installed Files column.
16. Click Add and select the library files to upload. In the appliance, the valid files are `.jar` and `.dll` are the valid library file types. The files that you select are displayed in the Files to Add column.
17. Click Update. The files that display in the Files to Add column are not committed until you click Update.
18. A confirmation dialog box is displayed. Follow the instructions in the dialog box.
19. Click Next. The installation is complete.
20. Click Done.

Starting and stopping Secure Connectors on Windows (Installed as a Windows Service)

1. Open the Windows Services window : Start > Control Panel > Administrative Tools > Services.
2. Scroll down the list of services to locate the IBM Secure Connector service.
3. Right-click on the IBM Secure Connector service and select the appropriate command: Start, Stop, Pause, Resume, or Restart.

Starting and stopping Secure Connectors on Windows (Installed as a Windows Application)

1. Start the Secure Connector from either the Windows Start menu shortcut or desktop shortcut.
 - From the Windows Start button, select All Programs > IBM > Cast Iron Secure Connector <connector_name> > Start Secure Connector
 - From the Windows desktop, click the Start Secure Connector shortcut to start the Secure Connector.
2. Stop the Secure Connector from either the Windows Start menu shortcut or desktop shortcut.
 - From the Windows Start button, select All Programs > IBM > Cast Iron Secure Connector <connector_name> > Stop Secure Connector.

- From the Windows desktop, click the Stop Secure Connector shortcut to stop the Secure Connector.

Starting and stopping Secure Connectors on Linux

1. Start the Secure Connector from either the menu shortcut , desktop shortcut or command line. Choose one of the following options:
 - Select *<application>* > IBM > Cast Iron® Secure Connector *<connector_name>* > Start Secure Connector.
 - From the desktop, click the Start Secure Connector shortcut to start the Secure Connector.
 - From the command prompt, enter `runclient_osgi.sh start`.
2. Stop the Secure Connector from either the menu shortcut, desktop shortcut, or command line. Choose one of the following options:
 - Select *<application>* > IBM > Cast Iron Secure Connector *<connector_name>* > Stop Secure Connector.
 - From the desktop, click the Stop Secure Connector shortcut to stop the Secure Connector.
 - From the command prompt, enter `runclient_osgi.sh stop`.
 -

Upgrading Secure Connectors

This topic provides information about upgrading Secure Connectors.

1. Create a new Secure Connector.
2. Download the latest version of the Secure Connector installer, based on your operating system. For example, Windows or Linux.
3. On a Windows or Linux machine, launch the Secure Connector installer. The Cast Iron Secure Connector wizard guides you through the upgrade process.

Note: On a 32-bit or 64-bit Linux machine, you must download a 32-bit or 64-bit Secure Connector, respectively, and complete the upgrade process. You cannot upgrade to a 32-bit Secure Connector on a 64-bit Linux machine.

Note: If you already have a Secure Connector installation that is higher than or same as the latest version, a warning message states that you have an existing installation and alternatively you can upgrade the existing installation.

Note: You must stop the Secure Connector (if already started) before upgrading.

Note: Before you proceed with the Secure Connector upgrade process, ensure that you have:

- Stopped the Secure Connector and corresponding projects.

- Taken a manual backup of the certificates (if any) located at `<secure_connector_install_path>/etc/security` or `jre/lib`. You may want to replace/add your certificates after upgrade.
- 4. Click the Upgrade option. The Select the installed path list box is displayed.
- 5. Select the Secure Connector installed path, if it is displayed in the list box. Else, click Browse button to select the installed path.
- 6. Click Next, then read and accept the licensing agreement.
- 7. Click Next. The installation progress is displayed. A message is displayed stating that the installation has been completed successfully. The path to the installer program is also displayed.
- 8. Click Done.
- 9. Start the Secure Connector.

Updating (re-configuring) the Secure Connector

If you have a new Secure Connector configuration file and you want to re-configure your secure connector, complete the steps described below:

1. Launch the Secure Connector Configuration wizard. To launch the wizard:
 - a. Windows machine: Go to Start > All Programs > IBM > Cast Iron Secure Connector `<connector_name>` > Secure Connector Configuration.
 - b. Linux machine: Select `<application>` > IBM > Cast Iron Secure Connector `<connector_name>` > Secure Connector Configuration.
2. The Secure Connector configuration wizard guides you through the upgrade process.
3. Click Next. The current Secure Connector Configurations are displayed if the Secure Connector is already configured.
4. Update the Secure Connector configuration by clicking the Previous button and then browse and select the new Secure Connector file.
5. Click Next and verify the configuration settings.
6. Click Next.
7. Specify settings for a proxy server: Proxy Server, Proxy Port, Login ID, Login Password, and Retype Password. These parameters are only required if your network requires that the Secure Connector uses a proxy to connect to the Cast Iron® Cloud Gateway.
8. The Create Vendor JAR window is displayed. In the Connector column, select the connector for which you want to install additional files. Files that have already been installed are displayed in the Installed Files column.
9. Click Add and select the library files to upload. In the appliance, the valid files are `.jar` and `.dll` are the valid library file types. The files that you select are displayed in the Files to Add column.
10. Click Update. The files that display in the Files to Add column are not committed until you click Update.
11. A confirmation dialog box is displayed. Follow the instructions in the dialog box.
12. Click Next. The upgrade is complete.
13. Restart the Secure Connector.

Creating a basic connection

By using the Cloud Integration basic connection feature you can connect to any on-premises system from the Bluemix cloud. Setting up a connection is the first step to establishing one or more tunnels to your on-premises endpoints.

Because the enterprise server is behind a firewall, you can only access this server through a Secure Connector. Follow the procedure that is given below to install and run a basic Secure Connector.

After you create a Secure Connector in the cloud, you must configure a machine behind the firewall to facilitate communication between the Secure Connector and a specific endpoint behind the firewall. Use the Secure Connector installer to configure the machine behind the firewall.

Note: Only one basic connection can be created per Bluemix organization. Through this single connection, multiple endpoints can be attached, thus enabling access to more than one enterprise system.

To create a basic connection, complete the following steps.

1. Go to the Cloud Integration home page and click SECURE CONNECTIONS.
2. Enter a name for the new connection, then in the Basic column, click Add.
3. A row is added listing your new basic connection. Click Install.
4. From the dialog that appears, select the installer for your system.

Note: Currently, only Ubuntu 64-bit v12.04 or higher is supported

5. Copy the downloaded file to the target system, then extract it and the archived file that it contains. You end up with a directory structure that looks like the following example:
6.

```
drwxr-xr-x 2 vagrant vagrant 4096 Oct 2 01:43 ./
```
7.

```
drwxr-xr-x 3 vagrant vagrant 4096 Oct 9 20:15 ../
```
8.

```
-rw-r--r-- 1 vagrant vagrant 96 Oct 2 01:43 installfiles.md5sum
```
9.

```
-rw-r--r-- 1 vagrant vagrant 20480 Oct 2 01:43 installfiles.tar
```
10.

```
-rwxr-xr-x 1 vagrant vagrant 4767 Oct 2 01:43 install.sh*
```

10. 6. Run install.sh.

Note: You are asked to provide the path to Java if it is not on your system path. The output looks similar to the following example:

```
vagrant@vagrant:/vagrant/connector/napinstall$ sudo ./install.sh
Checking MD5 sum...
Installing Native API Environment
useradd: user 'nativeapiadmin' already exists
Generating public/private rsa key pair.
Created directory '/home/nativeapiadmin/.ssh'.
Your identification has been saved in /home/nativeapiadmin/.ssh/id_rsa.
Your public key has been saved in /home/nativeapiadmin/.ssh/id_rsa.pub.
The key fingerprint is:
```



```
63:ed:69:b9:a8:ef:1c:ab:e1:01:0d:39:6d:5b:4a:6e nativeapiad-min@vagrant
id_rsa.pub for nativeapiadmin copied to current directory
stop nativeapimgmt if running
stop: Unknown instance:
nativeapimgmt start/running, process 2608
nativeapimgmt installed and started successfully
```

11. In the list of connectors, click the Refresh icon. The connection now appears as `Connected`.
12. Optional: If you experience problems with getting the basic connection to show as `Connected`, try restarting the connection service on your remote machine. To do this, from the same directory from which you ran `install.sh`, run the following command:

```
sudo service nativeapimgmt restart
```

A public key named `id_rsa.pub` is listed. Upload this public key back to Cloud Integration, then close the dialog.

The new basic connection is now successfully created and configured. Now, proceed to create a new endpoint. For more information, see [Creating an on-premises API endpoint](#).

Creating an Integration

Follow the procedure listed below to create an integration:

Note: This feature works only with the latest version of Cast Iron Live. Contact IBM Support to get access to the latest version.

1. Go to the Cloud Integration home page. Click the Create Integration button. The Cast Iron Live sign on page is displayed.
2. Specify your Cast Iron Live logon credentials and click Sign In. The Cloud Management Console is displayed.

Note: On the sign-on page, you can either choose the Evaluation Version or the Production Version. If you do not have a Cast Iron Live account, you can even sign-up for an account.

3. Create your required Cast Iron orchestrations. The orchestrations must be wrapped in HTTP Receive-Response request.

You have successfully created a Cast Iron Live integration.

Signing up for Cast Iron Evaluation Version

Cast Iron Live is a multi-tenant, cloud-based platform for integrating cloud and on-premise applications with enterprise systems in a hybrid environment. You can configure, run, and manage integration in the cloud without any infrastructure footprint. Each organization must have an admin who can sign-up and obtain credentials to a Cast Iron Live evaluation cloud. The

admin is authorized to create accounts for users within the organization. If the other users within the organization try to sign-up, they will be provided with the contact details of the admin.

Complete the follow steps to sign-up with Cast Iron Live Evaluation version:

1. Go to the Cloud Integration home page. Click the Create Integration button. The Cast Iron Live sign on page is displayed.
2. Click the Sign Up link. The sign-up instructions are displayed, if you are the first to sign-up from your organization (for example the Admin).
3. Click Yes, Sign Me Up.

An email is sent to the mail address that your organization is registered with. The email contains the instructions on getting started with Cast Iron Live. This includes the username and password, system requirements, and quick tips.

4. The Sign up for Cast Iron Live page displays the thank you message, system generated username, and email Id for any queries.
5. Click Create Integration Now to sign in to Cast Iron Live with your newly received user credentials.

Connecting to a user-defined endpoint

Cloud Integration supports connections to any type of on-premises system. When you create a user-defined endpoint, a host name and port are defined and these can be used to access on-premises systems. This type of endpoint can be secured privately with mutual TLS authentication.

Before you begin, you must create a basic connection. For more information, see [Creating a basic connection](#).

1. Go to the Cloud Integration home page and click SECURE CONNECTIONS.
2. Find the basic connection that you created and click the Arrow icon in the "Detail" column.
3. Click Add an Endpoint to start the wizard.
4. Select the User defined endpoint type.
5. Enter the host name and port for the on-premises system.
6. Specify how you want to secure access to the endpoint. For more information about the different security options, see [Configuring your application for TLS \(public\) or mutual TLS \(private\) access](#).
7. Click Create.

Your endpoint is created successfully and appears in the list of endpoints.

Creating Cloud Integration APIs

Follow the procedure listed below to create a new Cloud Integration API.

1. On the Cloud Integration home page, you can create APIs by first setting up a secure connection and then creating an integration. For more information about adding and configuring Secure Connectors, see [Managing secure connections - Standard Secure Connector](#) and [Managing the Enterprise Secure Connector](#). For more information about creating an integration, see [Creating an Integration](#).
2. Click the Create an Enterprise API button. The Create an Enterprise API page is displayed.
3. Specify the name and description of your API.

You can create APIs from the following:

- [Enterprise Endpoint - DB2/Oracle](#) and [Enterprise Endpoint - SAP](#)
 - [Cast Iron Live Orchestrations](#)
 - [Bluemix Apps](#)
4. The newly created API tiles are present on the Cloud Integration page.

Creating an API from a database endpoint (DB2 or Oracle)

Note: Cloud Integration for Bluemix supports the following versions of Oracle:

- Oracle 11g (R1)
- Oracle 10g R1 and R2
- Oracle 9i R1 and R2
- Oracle 8i R2 (8.1.6) and R3 (8.1.7)

Complete the following steps to create an API from an Enterprise endpoint:

1. Go to the Cloud Integration home page. Click the Create an API button. The Create an API page is displayed.
2. Specify the name and description of your API.
3. Click Generate from an Enterprise Endpoint. The "Create an Enterprise API" page is displayed.
4. In the "Create an Enterprise API" page, you must begin by connecting to the database you want.
5. Click the Select your Enterprise Endpoint arrow and select Add another Enterprise Endpoint. The Connect to an Endpoint page is displayed.
6. Specify where the endpoint is by clicking the Where is the endpoint arrow and selecting one of the choices provided. A new field is displayed.
7. Add the service name.
8. Select either DB2® or Oracle and specify the following details:

Note: For on-premise applications, someone with a database administrator role on the customer's infrastructure, discusses with the integration developer about the number of users and the access permissions for each of the users. The administrator then grants those users the access required for APIs.

The administrator can also grant the users appropriate access and privileges to database assets. For example, the database administrator can grant access to particular database tables and certain operations the users can perform on the table.

- Database Name
 - User name
 - Password
9. Click Create
 10. Go to the Cloud Integration dashboard and clickCreate API.
 11. Provide and name and description.
 12. Select an existing on-premises database and select the required schema.
 13. After you select the schema, select the table for which you want to create an API.
 14. Select the resources that you want to make available when you creat the API. You can include resources such as Get, Post, Put, and Delete.

Note: Click the copy icon available next to the resource names to view and copy the API URL.

15. Click Create API. The API is successfully created with the required resources.
16. Expand the resources to view the JSON examples. Sample JSON script:

GET:

Request JSON:

Response JSON:

```
{
  "rows": {
    "row": [
      {
        "CUSTOMERID": "xs:integer",
        "NAME": "xs:string",
        "ADDRESSID": "xs:integer",
        "TITLE": "xs:string",
        "FLAG": "xs:string"
      }
    ]
  }
}
```

POST:

Request JSON:

```
{
  "rows": {
    "row": [
      {
        "CUSTOMERID": "xs:integer",
        "NAME": "xs:string",
        "ADDRESSID": "xs:integer",
        "TITLE": "xs:string",
        "FLAG": "xs:string"
      }
    ]
  }
}
```

Response JSON:

```
{
  "rows": {
    "rowsModifiedCount" : "xs:int"
  }
}
```

```

    }
PUT:
Request JSON:
{
  "rows": {
    "row": [
      {
        "CUSTOMERID": "xs:integer",
        "NAME": "xs:string",
        "ADDRESSID": "xs:integer",
        "TITLE": "xs:string",
        "FLAG": "xs:string"
      }
    ]
  }
}
Response JSON:
{
  "rows": {
    "rowsModifiedCount" : "xs:int"
  }
}
DELETE:
Request JSON:
Response JSON:
{
  "rows": {
    "rowsModifiedCount" : "xs:int"
  }
}

```

17. Go to the service instance home page where you can see the newly created API.
18. Click Create an API to create a new API for yet another database. Repeat the steps from [Step 11](#) onwards.

You have successfully created an API from a database.

Note:

- The default response format is JSON. You can specify an alternative format in the Accept header. The following response is supported: `format:application/json;`
`application/xml;` `application/text.`
- The default request format is JSON. For PUT and POST operations, you can specify an alternative format in the Content-Type header.
- If you are connecting to DB2 on z/OS, and if the value of the `rowsModifiedCount` is -2 when you invoke the REST API for PUT, POST, and DELETE operations, this means that the operation was successful but the exact number of rows modified was not returned by the JDBC driver.

Now you can publish your API. For more information, see [Publishing an API as a private service](#).

Creating an Enterprise API - SAP endpoint

Follow the procedure listed below to create an enterprise API from an SAP endpoint:

1. Go to the Cloud Integration home page. Click the Create an Enterprise API button. The Create an Enterprise API page is displayed.
2. Specify the name and description of your API.
3. Click the Generate from an Enterprise Endpoint button. The Resources page is displayed. In the Resources page, you must begin by connecting to the database you want.

Note: If you are creating an enterprise endpoint for the first time, you must click the Connect to your first Enterprise Endpoint button.

4. On the Resources page, select the Add another Enterprise Endpoint option from the Select your Enterprise Endpoint drop-down list. The Connect to a Database or SAP page is displayed.
5. Click the SAP button specify the following details:

Note: For on-premise applications, someone with a database administrator role on the customer's infrastructure, discusses with the integration developer about the number of users and the access permissions for each of the users. The administrator then grants those users the access required for APIs.

The administrator can also grant the users appropriate access and privileges to database assets. For example, the database administrator can grant access to particular database tables and certain operations the users can perform on the table.

- Name of the secure connector
 - Hostname or IP Address
 - Port
 - System Number
 - Client
 - User name
 - Password
6. Click Connect. Once you click connect, the database is added to the list of databases. A new field called Schema is added.
 7. Select an already existing on-premise database and select the required schema.
 8. After selecting the schema, select the table for which you want to create an API.
 9. Select the resources that you want to make available when creating the API. You can include resources such as Get, Post, Put, and Delete.

Note: Click the copy icon available next to the resource names to view and copy the API URL.

10. Click Create API. The API is successfully created. The details that are displayed includes the API_SECRET key, the resources and the JSON examples. The generated API_SECRET key is displayed in the API details.
11. Click the copy icon to view the URL, the API_SECRET key and the user name and password to access the API.

You have successfully created an API from SAP. Your newly created SAP API tile is now present in the Cloud Integration home page.

Creating an API from Cast Iron Live Orchestrations

Ensure that you:

- Enable the pop-ups on your browser.
- Use your v7.x credentials to log on to Cast Iron Live.

Follow the procedure listed below to create an API from Cast Iron Live Orchestrations:

Note: This feature works only with the latest version of Cast Iron Live. Contact IBM Support to get access to the latest version.

1. Go to the Cloud Integration home page. Click the Create an API button. The Create an API page is displayed.
2. Specify the name and description of your API.
3. Click the Create from Cast Iron Live Orchestrations button The Cast Iron Live sign-on page is displayed.
4. Specify your Cast Iron Live logon credentials and click Sign In The Resources page is displayed.
5. Select the resources that you want to make available when creating the API. You can include resources such as Get, Post, Put, and Delete.
6. Select the name of your project available in the cloud management console.
7. Add the parameters and code sample, if required.
8. Click Create API. The API is successfully created with the required resources.

You have successfully created an API from Cast Iron Live Orchestrations.

Creating an API from Bluemix applications

Follow the procedure listed below to create an API from Cast Iron Live Orchestrations

1. Go to the Cloud Integration home page. Click the Create an API button. The Create an API page is displayed.
2. Specify the name and description of your API.
3. Click the Create from Bluemix apps button. The Resources page is displayed.
4. Select the resources that you want to make available when creating the API. You can include resources such as Get, Post, Put, and Delete.
5. Select the a Bluemix application from the list.
6. Add the parameters and code sample, if required.
7. Click Create API. The API is successfully created with the required resources.

You have successfully created an API from Bluemix applications.

Creating an on-premises API endpoint

Before you begin, create a basic connection. For more information, see [Creating a basic connection](#)

To create a new on-premises API endpoint, complete the following steps.

1. Go to the Cloud Integration home page and click **SECURE CONNECTIONS**.
2. Find the basic connection that you created previously, and click the arrow icon in the "Detail" column.
3. Click **Add an Endpoint** to launch the wizard.
4. In the "Connect to an Endpoint" section, click **API**.
5. Complete the remaining fields, including the host name and port of your on-premises system, the relative URI to the root of your API, and any credentials required to access the API.
6. Specify how you want to secure access to the endpoint. For more information about the different security options, see [Configuring your application for TLS \(public\) or mutual TLS \(private\) access](#).
7. Click **Create**.

Your endpoint is created and is displayed in the list of endpoints.

Configuring your application for TLS (public) or mutual TLS (private) access

Securing the communication channel and access control to your application is an important step to providing security for your overall API.

Warning: When you created the endpoint, if you selected the **Publicly** security option and cleared the **Use TLS/SSL to encrypt incoming connections** check box, this section does not apply. There will be no security enforced by Cloud Integration between you and your on-premises endpoint. Data (including authentication details) flowing between IBM Bluemix and any application using this endpoint will be sent unencrypted.


To use an endpoint that is created publicly with TLS, complete the following steps.

Note: Endpoints that are created with this access type encrypt data between the application client and the on-premises connection, but does not authenticate clients.

1. After you create the endpoint, click the **i** icon to the right of the endpoint name. A window opens.
2. Click the **Download Certificate** button. A compressed file begins to download.
3. Extract the contents of the compressed file. It contains a certificate file.
4. Add the certificate file to your applications trust store. This process varies depending upon which technology the application is built. Consult the documentation for your application platform about how to add a certificate to the applications trusted store.
5. Continue developing and deploying the application. Communication between the application and the endpoint is secured with TLS.

To use an endpoint that is created **Privately** (mutual TLS is required), complete the following steps.

Note: Endpoints created with this access type encrypt data between the application client and the on-premises connection, and also require a certificate to authenticate access to the endpoint.

6. After you create the endpoint, click the  icon to the right of the endpoint name. A window opens.
7. Click the Download Certificate button. A compressed file begins to download.
8. Extract the contents of the compressed file. It contains a certificate file.
9. Add the server-cert file to your application's trust store.
10. Add the client-cert and private-key files to your application's key store. The application will need to use the key store to authenticate against the endpoint, and the trust store to trust the endpoint's certificate. This process varies depending upon which technology the application is built. Consult the documentation for your application platform for details on how to manage key and trust stores.
11. Continue developing and deploying the application. Communication between the application and the on-premises endpoint is secured with mutual TLS authentication.

Creating a new Cloud Integration REST API that links to an existing on-premises API

Before you begin, ensure you complete the following tasks.

- [Creating a basic connection](#)
- [Creating an on-premises API endpoint](#)

To create a new API, complete the following steps.

1. Go to the Cloud Integration home page and click APIs.
2. Click Create API.
3. Enter a short title for the new API and a longer more detailed description.
4. Select Create from an On-premises API.
5. Click the arrow and select the endpoint that has access to your on-premises API.
6. The system prompts you to upload a file defining the API. For SOAP APIs, upload a WSDL file; for REST APIs, upload a Swagger file. Select Load.

Note: Currently, Cloud Integration cannot retrieve a WSDL or Swagger definition from your on-premises API.

7. Cloud Integration parses the resource file and then prompt you for any additional required resource files. Upload any requested resources until all entries show as "Complete". Click Next.
8. Review the APIs loaded from the resource files for accuracy.
9. Optional: Upload any supporting API documentation in one of the accepted formats listed.
10. Click Create API to complete the process.

Your API is created and is available for use, SDK generation, access credentials and so on.

Publishing an API as a private service

Publishing an API as a Private Service exposes your new API as a service in the Bluemix catalog under the Private Services category.

Note: Only organization managers can publish an API as a private service.

To publish your API as a Private Service, complete the following steps.

1. Go to the Cloud Integration home page and select the API you want to work with. The "View API" window opens.
2. Click Publish API.

Private Services are only visible to members of your organization. Your Private Service shows as a new tile in the Bluemix catalog and are easily distinguishable by the orange border and the orange "Private" tag at the bottom of the tile.

Viewing APIs

The APIs that you have created are displayed as API tiles in the service console.

1. Click the newly created API tile to view details about the API. The description and API resources are displayed.
2. Click the arrow beside the tile to view the parameters and examples related to an API resource.

Using the API URL

The APIs created are saved in the VCAP_SERVICES as well. Complete the following steps to use the API URL

1. If you are using Node.js as the runtime for your Bluemix application, use the following sample code for the Cloud Integration API:

```
2.   if (process.env.VCAP_SERVICES) { //USE FOR CLOUDFOUNDRY DEPLOYMENT
3.       var env = JSON.parse(process.env.VCAP_SERVICES);
4.       env_cloudint = env['CloudIntegration-0.1'][0].credentials;
5.   }
6.
7.       var user = env_cloudint .userid;
8.       var password = env_cloudint .password;
9.   for (i=0; i<env_cloudint .apis.length; i++) {
10.        if (env_cloudint .apis[i].name == api_name)
11.            myurl = url.parse(env_cloudint .apis[i].url);
12.   }
13.
14.   if (myurl == null)
15.       console.log ('API Url is not defined!');
16.   options = {
17.       host: myurl.hostname,
18.       port: 443,
19.       path: myurl.path,
```

```

20.         secureProtocol: 'SSLv3_method',
21.         headers: {'Authorization': 'Basic ' + new Buffer
    (user+':'+password).toString('base64')}
22.     };
23.     //Set the API method to be invoked
24.     options.method = 'GET';
25.
26.     //Make a http request
27.
28.     var request = https.request(options, function(response) {
29.         //Response handling here
    });

```

30. Make an HTTPS call using the API URL.

Note: You can now provide a different username and password to access the on-premise database. This is an optional feature. These credentials will override the credentials provided during the API creation. You will need to pass the Username and Password as HTTP headers. If there are no USERNAME and PASSWORD passed in the headers while invoking the API, then the credentials specified during API creation would be used.

Downloading an SDK

The option to download an SDK is shown when you view a Cloud Integration API. SDKs are dynamically generated from the API definition and enable you to quickly make use of the API in your chosen language. When you download an SDK, it's easier for you to use Cloud Integration APIs from applications.

Before you begin, ensure that you have completed the following tasks.

- Created an API. For more information, see [Creating Cloud Integration APIs](#).
- Connected your API to an endpoint. For more information, see [Connecting to a user-defined endpoint](#).

SDKs can be downloaded from APIs generated from an endpoint (DB2, Oracle, or SAP), Cast Iron Live orchestration, or a Bluemix application. The option to download an SDK is made available when you view the API either through the Cloud Integration Add-on, or as an API in the Private API catalog.

SDKs are available in three languages: JavaScript (running node.js), Java™, and Ruby (for example, running on Sinatra or Rails). The SDK is dynamically generated at the time of download and based on the definition of the API. The download is in the form of a compressed file.

The SDK has functions that represent each REST verb available in the API and depending on the language, contain models to represent request and response JSON. Other helper functions are included which allow for authentication and headers to be set in the request. Each SDK contains an example file to demonstrate how these functions and models should be used.

To download an SDK, complete the following steps.

1. Go to the Cloud Integration home page and select the API you want to work with. The "View API" window opens.
2. Select one of the following options to download a compressed file.
 - o Download JavaScript SDK

The downloaded archive file includes a number of directories and files.

- Prerequisites for using the client code are defined in the package.json file and can be installed by using the node package manager (npm).
- The client code is contained in the lib directory and is exported through the main.js file in the top-level directory.

The downloaded SDK is used as a node package and therefore more SDKs can be downloaded and combined into one package. This is done by exporting the client code in a single main.js file, updating the package.json file accordingly and uploading the package to your npm repository.

An example.js file is included in the top-level directory that shows you how to use the methods that are defined in the client code.

- o Download Java SDK

The downloaded archive file includes a number of JAR files. These are prerequisite JAR files and the precompiled SDK client code, which is contained in -client.jar and -client-sources.jar. The former file can be imported into your Java client code as usual, and is precompiled with debug information, whereas the latter file is provided to help with debugging applications that are using this SDK.

An Example.java file is included in the archive file and demonstrates the expected use of the functions that are provided in the client code. The client code is made up of a number of packages, all beginning with `com.ibm.cloudintegration`.

```
com.ibm.cloudintegration.api
```

This package contains the class *api_name*, which is where methods that drive the REST API are defined. This is the main class that any code developed with the SDKs will use, and is the only class that is directly used in the Example.java file.

```
com.ibm.cloudintegration.api_name
```

This package contains classes that represent the structure of request and response objects and are named according to the REST operation they correspond to. Where models are a complex type, submodels are also generated and named accordingly.

```
com.ibm.cloudintegration.common
```

This package contains utility classes that are used by the other classes that are listed above. The `ApiInvoker` class includes methods for more advanced configuration of sending REST calls.

- Download Ruby SDK

The downloaded archive file contains a set of Ruby files that can additionally be used as a RubyGem.

The main API class is defined in a Ruby file under the `lib` directory, and that file is named after the API that the SDK was generated from. Classes that represent the structure of request and response objects are stored in the `models` directory, and are named according to the REST operation they correspond to.

Where models are a complex type, submodels are also generated and named accordingly.

An example Ruby script, `example.rb`, is provided in the top-level directory that demonstrates how to use the downloaded Ruby SDK to interact with the API.

3. Depending on your browser settings, a file downloads to your download directory. The file is named using the convention: `api_name-clientlanguage-date-timestamp.zip`
4. Navigate to the downloaded file and import it into your development environment.

Upload the SDK to a Bluemix application so that application developers can bind your service to their applications and start using the APIs exposed in that service.

Creating a Data Sync

Create a data sync to replicate data from an on-premises database to a cloud database so that you can use the data in a cloud environment. After you create a data sync, you have faster access to data from cloud applications and you do not have to rely on a live connection to an on-premises database. After you create the data sync and migrate data, the data sync keeps your cloud table in sync with your on-premises table. Further updates to the on-premises table are automatically propagated to the cloud table.

Before you begin, ensure that you complete the following tasks.

- Create a Bluemix application and bind SQL database service to it.
- Create a Cloud Integration service. For more information, see [Creating an Integration](#).
- Install a standard secure connector. For more information, see [Managing secure connections - Standard Secure Connector](#).
- Add an endpoint to the on-premises database. For more information, see [Creating an API from a database endpoint \(DB2 or Oracle\)](#).
- Connect a Cloud Integration Add-on. For more information, see [Connecting an Add-On](#).

To create a data sync, complete the following steps.

1. Go to the Cloud Integration home page and click DATA SYNCs.

2. Click Add a new Data Sync. The "Create a Data Sync" window is displayed.
3. Provide a name and description for your data sync.
4. Click Next.
5. Select an option from each of the following lists.
 - Select your database
 - Select schema
 - Select a table

The table that you choose here, is the one that is going to be replicated in the cloud.

Note: You might be prompted for an on-premises database password.

Restriction: The table that you select must have a primary key and the primary key can only contain one column. The following data types are not supported.

- DBCLOB
 - GRAPHIC
 - LONG VARGRAPHIC
 - VARGRAPHIC
 - DB2SQLSTATE
 - BLOB
 - CLOB
 - XML
6. Click Next. The "Cloud Data" window opens.
 7. Select an option from each of the following lists.
 - Select a Bluemix application. Only Bluemix applications that have SQL database bindings are displayed here.
 - Select your database. Select the database that your application is bound to.
 - Select schema
 - Select table

Only valid tables that match the source table structure are listed here.

8. If your table is not listed, you can create one by clicking Add a table with matching structure and entering a name for it.
9. Click Create Data Sync. A new window opens that displays the database assets.
10. Required: Before a data sync can work, extra assets (control tables and/or table triggers) must be created on the source and target databases. Click Download to download the SQL script that you need to run on the source database. To run the SQL script, complete the following steps:
 - a. To open a DB2 command line, click Start > All Programs > IBM DB2 > *DatabaseInstance* > Command Line Tools, and select Command Window. (Where *DatabaseInstance* is your DB2 instance name, which by default is, DB2COPY1 (Default)). A DB2 - CLP window opens.
 - b. To connect to the required database, enter the following command:

```
db2 CONNECT TO DatabaseName
```

where:

- *DatabaseName* is the name of your on-premises database that you selected in [Step 5](#).

c. Enter the following command to run the script.

```
db2 -f "C:\Users\admin\Downloads\batch.sql" -t -v -s
```

where:

- *batch.sql* is the name of your SQL batch script.
- `-f` tells the command line processor to read command input from a file instead of from standard input.
- `-t` tells the command line processor to use a semicolon (;) as the statement termination character.
- `-v` tells the command line processor to echo command text to standard output.
- `-s` tells the command line processor to stop execution if errors occur while it is running commands in a batch file or in interactive mode.

The tables are created automatically.

11. When you create all the assets that are required for a data sync to work, click **Migrate now**. Clicking **Migrate now** copies the existing data in the source table to the target table. For large data tables, the migration might take a while.

Warning: The data is migrated via an unsecured channel.

12. You can view the status of the migration, on the "My Data Sync" page.

When you are finished, the "Status" panel displays that the "Overall data sync state" is connected and the "Last sync" field displays a time and date.

If the "Status" panel displays that the on-premises database is unreachable, check that the secure connector is connected and then check that your on-premises database is running.

The data that is synchronized can be accessed by any application and not just the application whose binding credential is being used to synchronize data.

Updates that you make to the on-premises database are reflected in the cloud table automatically. Click **Refresh status** to get information about when this update took place and whether it was successful or not.

Attention: There is a 5-minute time delay from the moment that any updates made to the on-premises table are reflected in the cloud table.

For multiple tables, you need to create a data sync for each table. To do this, go to [Step 2](#)

Request payload generation rules

When you provide a JSON input to the created API, conform to the following rules:

- **Case 1:** Passing a string value:

Accepted format: "name" : "Varun" (double quotes only).

- **Case 2:** Passing inexplicit quotes in a string:

Accepted format: "name" : "'Varun'" (The single quotes will be replaced with explicit double quotes).

- **Case 3:** Passing a number starting with a leading 0:

Accepted format: "id" : "_\$_0001_\$_" (Any string prefixed with `_s_` and post-fixed with `_$_` will be treated as a number).

- **Case 4:** Passing a number:

Accepted format: "age" : 30

Troubleshooting

Note: If you face any issues while creating or accessing an API, check the Cast Iron Secure Connector logs or check the DataPower® logs for the DataPower secure connector. For Cast Iron Integration, check the system logs of Cast Iron in your account.

You can encounter the exception given below with Secure Connector, while connecting to the database:

```
com.approuter.maestro.sdk.mpi.ActivityFailedException(Connection error
while getting connection out of the pool.  SQLSTATE: HY000 ERRORCODE: 0 Error
Message: [CastIron Systems][DB2 JDBC Driver]End of stream was detected on a
read.
[CastIron Systems][DB2 JDBC Driver]End of stream was detected on a read.)
FaultTime:2014-05-29T10:48:40.661Z JobID:BB85F0FE3B3EBD8FDEBEDAC80AEF4956
ActivityID:4
.....
com.approuter.module.database.protocol.DBConnectionException: Connection
error while getting connection out of the pool.  SQLSTATE: 08001 ERRORCODE: 0
Error Message: [CastIron Systems][DB2 JDBC Driver]Error establishing socket
to host and port: 9.1.2.1.:50000. Reason: Connection timed out: connect

    at
com.approuter.module.database.protocol.ConnectionManager.getConnection(Connec
tionManager.java:178)
    at
com.approuter.module.database.activity.GenericQueryActivity.process(GenericQu
eryActivity.java:140)
```



```
    at
com.approuter.module.database.activity.GenericQueryActivity.execute (GenericQueryActivity.java:107)
    at
com.approuter.agent.container.sdk.AgentServiceManager.execute (AgentServiceManager.java:132)
    at
com.approuter.agent.container.protocol.AgentActivityHandler.execute (AgentActivityHandler.java:346)
    at
com.approuter.agent.container.serviceimpl.AgentServiceImpl.execute (AgentServiceImpl.java:105)
    at sun.reflect.NativeMethodAccessorImpl.invoke0 (Native Method)
    at
sun.reflect.NativeMethodAccessorImpl.invoke (NativeMethodAccessorImpl.java:88)
    at
sun.reflect.DelegatingMethodAccessorImpl.invoke (DelegatingMethodAccessorImpl.java:55)
    at java.lang.reflect.Method.invoke (Method.java:618)
    at
com.sun.xml.ws.api.server.InstanceResolver$1.invoke (InstanceResolver.
```

This exception occurs if the database is not reachable from the Secure Connector. Verify that the database is accessible from the server machine where the Secure Connector is running.

How can I use multiple Secure Connectors in a single machine?

To use multiple Secure Connectors in a single machine, complete the following steps: You must install the new secure connections on different paths (different directories). Configure the newly installed Secure Connectors by changing the port numbers (Listen on Port and Transmit on Port). Contact with your network administrator and get new numbers for the Listen to Port and Transmit on Port fields.

AGENT_LOCKED error when starting the standard Secure Connector

If the standard secure connector is deleted from the Cloud Management Console, and a new one with the same name is created, the `AGENT_LOCKED` error will be thrown when starting the Secure Connector on the local machine.

If a secure connector is deleted and a new one with the same name is then created from Cloud Management Console, the secure connector configuration for the new instance is different from the original one. At this time, if the secure connector isn't updated in user's environment, the connector will still use the previous authentication key to connect to Cast Iron Live. Thus, the agent is locked by Cast Iron Live.

To resolve the problem, complete the following steps:

1. Contact IBM Cloud Integration Support to unlock the secure connector from the backend.
2. Re-configure the Secure Connector.
 - a. Download a new Secure Connector configuration from Cloud Management Console.
 - b. On Windows, run Secure Connector Configuration from the start menu to import the downloaded configuration.

- c. On Linux, run `agent_configurator.jar` from `*sc_installation_path*/utils` to import the downloaded configuration.
3. Restart the Secure Connector.

Common errors thrown when you invoke the API

- You do not have permission to access this API.

Solution: Ensure that you pass in the header "API_SECRET" the value for which will be specified in the API details when you created them.

- (The content of elements must consist of well-formed character data or markup.)
FaultTime:2014-07-22T14:51:45.524+05:30 JobID:8BDB16C024B8B4EFCF04FC6E024C4F72
ActivityID:6796

Solution: Verify the input HTTP header value for Content-Type. If it is 'application/xml', then the input XML input is not well formed. If it is 'application/json', then the input JSON input is not well formed.

- (An invalid XML character (Unicode: 0x7b) was found in the prolog of the document.)
FaultTime:2014-07-22T14:52:26.972+05:30 JobID:AAB2AC0C96CF7BB297579046BB5D4DD3
ActivityID:6796

Solution: Verify the input HTTP header value for Content-Type. If it is 'application/json', the JSON is not well formed or could not be converted into a valid XML. If it is 'application/xml', then the input XML input is not well formed.

Secure Connector window hangs and no information is displayed

Secure Connector window hangs and no information is displayed.

Solution: You must always run the Secure Connector as an Administrative user.

The Secure Connector installer on a 64-bit Linux machine throws error

The following error is thrown when you run the Secure Connector installer on a 64-bit Linux machine:

```
Invalid MIT-MAGIC-COOKIE-1 keyException in thread "main"  
java.lang.NoClassDefFoundError: Could not initialize class  
sun.awt.X11.XToolkit
```

Solution: To install Secure Connector on Linux , you must enable X or install VNC on Linux machines to get the Secure Connector installation wizard. After creating the connection, run the following commands to launch the UI based Secure Connector installation wizard:

```
sudo chmod +x linux-secure-connector-installer_64.sh sudo  
./linux-secure-connector-installer_64.sh
```

How to set the debug value of the Secure Connector logs to True

To set the `debug` value of the Secure Connector logs to `True`, complete the following steps:

1. Go to the root directory where your Secure Connector is installed.
2. Open the `config` folder.
3. Open the `localConfig_ws.xml` file for editing.
4. Change the value of the `<debug>` tag to `True`. By default, the value of the `<debug>` tag is `False`.

The application that was used to create the data sync is removed

If the application whose binding credential was used by data sync gets removed, then the data sync fails to work. The cloud database status displays `unreachable`.

Solution: Remove the data sync and start again.

The on-premises database password has changed

Changing the password for an on-premises database causes the data sync to fail. This happens even when you can change the password for the database endpoint.

Solution: Remove the data sync and start again.

You accidentally click Migrate twice during a data sync

If you accidentally hit Migrate and the target table has data in it, then you get a SQL error.


Solution: Empty the target table.


Tips and tricks

- 1) How can I to use multiple Secure Connectors in a single machine?

To user multiple Secure Connectors in a single machine, complete the following steps:

- a. You must install these secure connections on different paths (different directories).
- b. Configure the newly installed Secure Connectors by changing the port numbers.

User Data
Step 5 of 10 



Secure Connector Configuration

Name:

Tenant Id:

Environment Id:

Gateway:

Listen on Port:

Transmit on Port:

Auth Key:

Click the "Previous" button to load a Secure Connector configuration file.

(Made with IzPack - <http://izpack.org/>)

c. Contact with your network administrator and get new numbers for the **Listen to Port** and **Transmit on Port**.