



## **Red Hat Fuse 7.2**

# **Connecting Fuse Online to Applications and Services**

Instructions for Using Provided Connectors



# Red Hat Fuse 7.2 Connecting Fuse Online to Applications and Services

---

Instructions for Using Provided Connectors

## Legal Notice

Copyright © 2018 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

Follow step-by-step instructions for creating connections and adding them to integrations.

## Table of Contents

<b>PREFACE</b> .....	<b>5</b>
<b>CHAPTER 1. CONNECTORS THAT ARE SUPPORTED BY FUSE ONLINE</b> .....	<b>6</b>
<b>CHAPTER 2. CONNECT TO AMAZON S3</b> .....	<b>8</b>
2.1. PREREQUISITES FOR CREATING AN AMAZON S3 CONNECTION	8
2.2. CREATE AMAZON S3 CONNECTIONS	9
2.3. START AN INTEGRATION BY OBTAINING DATA FROM AMAZON S3	9
2.4. FINISH AN INTEGRATION BY ADDING DATA TO AMAZON S3	11
2.5. ADD DATA TO AMAZON S3 IN THE MIDDLE OF AN INTEGRATION	12
<b>CHAPTER 3. CONNECT TO AMQ</b> .....	<b>14</b>
3.1. CREATE AN AMQ CONNECTION	14
3.2. START AN INTEGRATION BASED ON RECEIVING AMQ MESSAGES	15
3.3. FINISH AN INTEGRATION BY PUBLISHING AMQ MESSAGES	17
3.4. PUBLISH AMQ MESSAGES IN THE MIDDLE OF AN INTEGRATION	18
<b>CHAPTER 4. CONNECT TO AMQP</b> .....	<b>21</b>
4.1. CREATE AN AMQP CONNECTION	21
4.2. START AN INTEGRATION BASED ON RECEIVING AMQP MESSAGES	22
4.3. FINISH AN INTEGRATION BY PUBLISHING AMQP MESSAGES	24
4.4. PUBLISH MESSAGES TO AMQP IN THE MIDDLE OF AN INTEGRATION	25
<b>CHAPTER 5. CONNECT TO DROPBOX</b> .....	<b>28</b>
5.1. REGISTER FUSE ONLINE AS A DROPBOX CLIENT	28
5.2. CREATE A DROPBOX CONNECTION	29
5.3. START AN INTEGRATION BY OBTAINING FILES FROM DROPBOX	30
5.4. FINISH AN INTEGRATION BY ADDING FILES TO DROPBOX	31
5.5. ACCESS DROPBOX IN THE MIDDLE OF AN INTEGRATION	32
<b>CHAPTER 6. CONNECT TO AN FTP OR SFTP SERVER</b> .....	<b>35</b>
6.1. CREATE AN FTP OR SFTP CONNECTION	35
6.2. OBTAIN FILES FROM AN FTP OR SFTP SERVER	36
6.3. UPLOAD FILES TO AN FTP OR SFTP SERVER	38
<b>CHAPTER 7. CONNECT TO GMAIL</b> .....	<b>40</b>
7.1. REGISTER FUSE ONLINE AS A GMAIL CLIENT APPLICATION	40
7.2. ENABLE ACCESS TO THE GMAIL API	42
7.3. CREATE A GMAIL CONNECTION	43
7.4. ALTERNATIVES FOR POPULATING EMAIL TO SEND	44
7.5. TRIGGER AN INTEGRATION WHEN POLLING RETURNS A GMAIL MESSAGE	45
7.6. SEND AN EMAIL FROM A GMAIL ACCOUNT	46
<b>CHAPTER 8. CONNECT TO GOOGLE CALENDAR</b> .....	<b>48</b>
8.1. REGISTER FUSE ONLINE AS A GOOGLE CALENDAR CLIENT APPLICATION	48
8.2. ENABLE ACCESS TO THE GOOGLE CALENDAR API	50
8.3. CREATE A GOOGLE CALENDAR CONNECTION	51
8.4. TRIGGER AN INTEGRATION WHEN POLLING RETURNS AN EVENT FROM A GOOGLE CALENDAR	52
8.5. OBTAIN A PARTICULAR EVENT FROM A GOOGLE CALENDAR	53
8.6. ADD AN EVENT TO A GOOGLE CALENDAR	54
8.7. UPDATE AN EVENT IN A GOOGLE CALENDAR	55
<b>CHAPTER 9. CONNECT TO HTTP AND HTTPS ENDPOINTS</b> .....	<b>57</b>
9.1. CREATE A CONNECTION TO AN HTTP OR HTTPS ENDPOINT	57

---

9.2. ADD AN HTTP OR HTTPS CONNECTION TO AN INTEGRATION	57
<b>CHAPTER 10. CONNECT TO KAFKA</b>	<b>60</b>
10.1. CREATE A CONNECTION TO A KAFKA BROKER	60
10.2. OBTAIN DATA FROM A KAFKA BROKER	60
10.3. PUBLISH DATA TO A KAFKA BROKER	62
<b>CHAPTER 11. CONNECT TO AN INTEGRATION'S LOG</b>	<b>64</b>
11.1. COMPARISON OF LOG STEPS AND LOG CONNECTIONS	64
11.2. ADD A LOG CONNECTION TO AN INTEGRATION	65
11.3. CREATE A REPLACEMENT LOG CONNECTION	66
<b>CHAPTER 12. CONNECT TO MQTT</b>	<b>68</b>
12.1. CREATE A CONNECTION TO AN MQTT BROKER	68
12.2. OBTAIN A MESSAGE FROM AN MQTT BROKER	69
12.3. PUBLISH A MESSAGE TO AN MQTT BROKER	70
<b>CHAPTER 13. CONNECT TO REST APIS</b>	<b>72</b>
13.1. REGISTER FUSE ONLINE AS AN API CLIENT	72
13.2. CREATE A AN API CLIENT CONNECTION	73
13.3. ADD AN API CLIENT CONNECTION TO AN INTEGRATION	73
<b>CHAPTER 14. CONNECT TO SALESFORCE</b>	<b>75</b>
14.1. REGISTER FUSE ONLINE AS A SALESFORCE CLIENT APPLICATION	75
14.2. CREATE A SALESFORCE CONNECTION	76
14.3. ADD A SALESFORCE CONNECTION TO AN INTEGRATION	77
<b>CHAPTER 15. CONNECT TO SAP CONCUR</b>	<b>79</b>
15.1. USER ROLES FOR CONNECTING TO SAP CONCUR	79
15.2. HOW TO OBTAIN SAP CONCUR IMPLEMENTATION SITE OAUTH CREDENTIALS	79
15.3. HOW TO OBTAIN SAP CONCUR PRODUCTION SITE OAUTH CREDENTIALS	80
15.4. CONFIGURE THE SAP CONCUR CONNECTOR	80
15.5. CREATE A SAP CONCUR CONNECTION	81
15.6. ADD A SAP CONCUR CONNECTION TO AN INTEGRATION	82
15.7. EXAMPLE OF MAPPING INTEGRATION DATA FIELDS TO SAP CONCUR FIELDS	82
<b>CHAPTER 16. CONNECT TO SERVICENOW</b>	<b>84</b>
16.1. CREATE A SERVICENOW CONNECTION	84
16.2. START AN INTEGRATION BY OBTAINING RECORDS FROM SERVICENOW	85
16.3. COPY RECORDS TO SERVICENOW DURING OR TO FINISH AN INTEGRATION	86
<b>CHAPTER 17. CONNECT TO SLACK</b>	<b>88</b>
17.1. CREATE A SLACK CONNECTION	88
17.2. ADD A SLACK CONNECTION TO TRIGGER EXECUTION UPON RECEIVING MESSAGES	89
17.3. ADD A SLACK CONNECTION TO SEND A MESSAGE TO A SLACK CHANNEL OR USER	90
<b>CHAPTER 18. CONNECT TO SQL DATABASES</b>	<b>91</b>
18.1. CREATE A DATABASE CONNECTION	91
18.2. START AN INTEGRATION BY ACCESSING A DATABASE	92
18.3. ACCESS A DATABASE IN THE MIDDLE OR TO COMPLETE AN INTEGRATION	93
18.4. CONNECT TO PROPRIETARY DATABASES	94
<b>CHAPTER 19. CONNECT TO TELEGRAM</b>	<b>95</b>
19.1. CREATE A TELEGRAM CONNECTION	95
19.2. ADD A TELEGRAM CONNECTION TO TRIGGER INTEGRATION EXECUTION UPON RECEIVING MESSAGES	96

---

19.3. ADD A TELEGRAM CONNECTION TO SEND A MESSAGE TO A TELEGRAM CHAT	96
<b>CHAPTER 20. TRIGGER INTEGRATION EXECUTION WITH A TIMER</b> .....	<b>98</b>
20.1. ADD A TIMER CONNECTION TO TRIGGER INTEGRATION EXECUTION	98
20.2. CREATE A REPLACEMENT TIMER CONNECTION	98
<b>CHAPTER 21. CONNECT TO TWITTER</b> .....	<b>100</b>
21.1. REGISTER FUSE ONLINE AS A TWITTER CLIENT APPLICATION	100
21.2. CREATE A TWITTER CONNECTION	101
21.3. ADD A TWITTER CONNECTION TO AN INTEGRATION	102
<b>CHAPTER 22. TRIGGER INTEGRATION EXECUTION WITH AN HTTP REQUEST (WEBHOOK)</b> .....	<b>103</b>
22.1. GENERAL PROCEDURE FOR USING THE FUSE ONLINE WEBHOOK	103
22.2. USE THE WEBHOOK CONNECTION	104
22.3. CREATE AN INTEGRATION THAT AN HTTP REQUEST CAN TRIGGER	104
22.4. HOW FUSE ONLINE HANDLES HTTP REQUESTS	105
22.5. GUIDELINES FOR AN HTTP CLIENT THAT INVOKES A FUSE ONLINE WEBHOOK	106
22.6. ABOUT THE JSON SCHEMA FOR SPECIFYING REQUEST PARAMETERS	106
22.7. HOW TO SPECIFY HTTP REQUESTS	107





---

## PREFACE

To integrate applications, you create a connection to each application or service that you want to integrate. You then create an integration and add a connection to it for each integration or service that you want to integrate.

Fuse Online [supports numerous connectors](#) that serve as templates for creating connections. The following topics provide details for creating connections and adding them to integrations:

- [Chapter 2, \*Connect to Amazon S3\*](#)
- [Chapter 3, \*Connect to AMQ\*](#)
- [Chapter 4, \*Connect to AMQP\*](#)
- [Chapter 5, \*Connect to Dropbox\*](#)
- [Chapter 6, \*Connect to an FTP or SFTP server\*](#)
- [Chapter 7, \*Connect to Gmail\*](#)
- [Chapter 8, \*Connect to Google Calendar\*](#)
- [Chapter 9, \*Connect to HTTP and HTTPS endpoints\*](#)
- [Chapter 10, \*Connect to Kafka\*](#)
- [Chapter 11, \*Connect to an integration's log\*](#)
- [Chapter 12, \*Connect to MQTT\*](#)
- [Chapter 13, \*Connect to REST APIs\*](#)
- [Chapter 14, \*Connect to Salesforce\*](#)
- [Chapter 15, \*Connect to SAP Concur\*](#)
- [Chapter 16, \*Connect to ServiceNow\*](#)
- [Chapter 17, \*Connect to Slack\*](#)
- [Chapter 18, \*Connect to SQL databases\*](#)
- [Chapter 19, \*Connect to Telegram\*](#)
- [Chapter 20, \*Trigger integration execution with a timer\*](#)
- [Chapter 21, \*Connect to Twitter\*](#)
- [Chapter 22, \*Trigger integration execution with an HTTP request \(Webhook\)\*](#)

# CHAPTER 1. CONNECTORS THAT ARE SUPPORTED BY FUSE ONLINE

Fuse Online supports the following connectors.

Name	Description
<a href="#">Amazon S3</a>	Retrieve data from an Amazon S3 bucket or copy data into a bucket.
<a href="#">AMQ</a>	Obtain messages from an ApacheMQ broker or publish messages to an ApacheMQ broker.
<a href="#">AMQP</a>	Obtain messages from an Advanced Message Queue Protocol broker or publish messages to an AMQP broker.
<a href="#">Dropbox</a>	Download files from Dropbox or upload files to Dropbox.
<a href="#">FTP/SFTP</a>	Download files from an FTP or SFTP server or upload files to an FTP or SFTP server.
<a href="#">Gmail</a>	Obtain messages sent to a particular Gmail account and send messages from a particular Gmail account.
<a href="#">Google Calendar</a>	Obtain events from a Google Calendar that you specify or add/update events in a Google Calendar that you specify.
<a href="#">HTTP/HTTPS</a>	Connect to an HTTP or HTTPS endpoint and execute the <b>GET</b> , <b>PUT</b> , <b>POST</b> , <b>DELETE</b> , <b>HEAD</b> , <b>OPTIONS</b> , <b>TRACE</b> , OR <b>PATCH</b> method.
<a href="#">Kafka</a>	Obtain streams of records from or publish streams of records to a Kafka topic that you specify.
<a href="#">Log</a>	Send information about messages that an integration processes to the integration's log.
<a href="#">MQTT</a>	Obtain messages from an MQ Telemetry Transport broker or publish messages to an MQTT broker.
<a href="#">REST APIs</a>	<p>Create a custom REST API client connector by uploading an OpenAPI schema. You can then create a connection to that REST API.</p> <p>Create a REST API provider integration by uploading an OpenAPI schema that defines operations that trigger integration execution. See <a href="#">triggering integration execution with API calls</a> in Integrating Applications with Fuse Online.</p>
<a href="#">Salesforce</a>	Create, update, fetch, or delete a Salesforce record.
<a href="#">SAP Concur</a>	Perform any one of a large variety of SAP Concur actions.

Name	Description
<a href="#">ServiceNow</a>	Obtain records from or copy records to your ServiceNow instance.
<a href="#">Slack</a>	Obtain messages from a channel or send a message to a Slack channel or user.
<a href="#">SQL databases</a>	Invoke a SQL statement or a SQL stored procedure on an Apache Derby, MySQL, or PostgreSQL database. To connect to other types of SQL databases, you upload a Fuse Online library extension that contains a JDBC driver for that database.
<a href="#">Telegram</a>	Obtain messages from a chat or send messages to a chat by using a Telegram chat bot.
<a href="#">Timer</a>	Set a simple timer or a <b>cron</b> timer to trigger integration execution.
<a href="#">Twitter</a>	Trigger execution of an integration upon tweets that mention you or that contain data you specify.
<a href="#">Webhook</a>	Trigger integrations with HTTP <b>GET</b> or <b>POST</b> requests.

If Fuse Online does not provide a connector that you need, an experienced developer can create an extension that defines a custom connector. For information about coding the extension and creating its **.jar** file, which you upload to Fuse Online, see [Tooling User Guide, Develop extensions for Fuse Online integrations](#) and [Integrating Applications with Fuse Online, Develop extensions](#).

## CHAPTER 2. CONNECT TO AMAZON S3

An integration can retrieve data from an Amazon S3 bucket or copy data into an Amazon S3 bucket. To do this, you create an Amazon S3 connection and then add that Amazon S3 connection to an integration. For details, see:

- [Section 2.1, “Prerequisites for creating an Amazon S3 connection”](#)
- [Section 2.2, “Create Amazon S3 connections”](#)
- [Section 2.3, “Start an integration by obtaining data from Amazon S3”](#)
- [Section 2.4, “Finish an integration by adding data to Amazon S3”](#)
- [Section 2.5, “Add data to Amazon S3 in the middle of an integration”](#)

### 2.1. PREREQUISITES FOR CREATING AN AMAZON S3 CONNECTION

To create an Amazon S3 connection, you must know the following:

- Amazon S3 access key ID that is associated with the Amazon Web Services (AWS) account that created, or will create, the bucket that you want the connection to access.  
You can create a connection that accesses a bucket that does not yet exist. In this case, when the integration starts running then it use the AWS account associated with this access key ID to try to create the bucket.
- Amazon S3 secret access key that is associated with the AWS account that created or will try to create (when the integration starts running) the bucket that you want the connection to access.
- Name of the bucket that you want to access or its Amazon Resource Name (ARN).  
If the bucket you specify does not yet exist then the connection tries to create a bucket with the name that you specify. Because S3 allows a bucket to be used as a URL that can be accessed publicly, the bucket name that you specify must be globally unique. Also, it must meet [S3 bucket naming requirements](#).  
  
If the bucket you specify does not exist in the AWS account that is associated with the Amazon S3 access key ID, but it does exist in another AWS account, then the connection does not create the bucket and an integration that uses this connection cannot start running.
- Region in which the bucket is located or the region in which you want the connection to create the bucket.

A user with the login credentials for the AWS account that created or will create the bucket obtains the Amazon S3 keys as follows:

1. Go to <https://aws.amazon.com/s3/>.
2. Hover over **My Account**, select **AWS Management Console** and sign in to the console with the AWS account that created the bucket that you want to access or with the account that you want the connection to use to create the bucket.
3. In the console, in the upper right, click the down arrow next to the user name and click **My Security Credentials**.
4. Expand **Access Keys** and click **Create New Access Keys**.

5. Follow the prompts to obtain the keys.

## 2.2. CREATE AMAZON S3 CONNECTIONS

You must create an Amazon S3 connection before you can add an Amazon S3 connection to an integration.

### Prerequisite

See [Prerequisites for creating Amazon S3 connections](#).

### Procedure

1. In Fuse Online, in the left panel, click **Connections** to display any available connections.
2. In the upper right, click **Create Connection** to display Fuse Online connectors.
3. Click the **Amazon S3** connector.
4. In the **Access Key** field, enter the Amazon S3 access key ID, provided by AWS, for the AWS account that created the bucket that you want this connection to access. If the bucket you want the connection to access does not already exist then when Fuse Online tries to start running the integration, it uses the AWS account associated with this access key to create the bucket. However, if the bucket already exists in some other AWS account, then the connection cannot create the bucket and the integration cannot start.
5. In the **Bucket Name or Amazon Resource Name** field, enter the name of the bucket that you want this connection to access or enter the bucket's ARN. If the bucket does not already exist in either the AWS account being used or in any other AWS account, then the connection creates it.
6. In the **Region** field, select the AWS region in which the bucket resides. If the connection creates the bucket, then it creates it in the selected region.
7. In the **Secret Key** field, enter the Amazon S3 secret access key, provided by AWS, for the account that created, or will create, the bucket that you want this connection to access.
8. Click **Validate**. Fuse Online immediately tries to validate the connection and displays a message that indicates whether or not validation is successful. If validation fails, revise the configuration details as needed and try again.
9. When validation is successful, click **Next**.
10. In the **Connection Name** field, enter your choice of a name that helps you distinguish this connection from any other connections. For example, enter **Obtain S3 Data**.
11. In the **Description** field, optionally enter any information that is helpful to know about this connection. For example, enter **Sample S3 connection that obtains data from the northeast bucket**.
12. In the upper right, click **Create** to see that the connection you created is now available. If you entered the example name, you would see that **Obtain S3 Data** appears as a connection that you can choose to add to an integration.

## 2.3. START AN INTEGRATION BY OBTAINING DATA FROM AMAZON S3

To start an integration by obtaining data from an Amazon S3 bucket, add an Amazon S3 connection as the start connection in an integration.

## Prerequisite

You created an Amazon S3 connection.

## Procedure

1. In the Fuse Online panel on the left, click **Integrations**.
2. Click **Create Integration**.
3. On the **Choose a Start Connection** page, click the Amazon S3 connection that you want to use to start the integration.
4. On the **Choose an Action** page, click the action that you want the connection to perform:
  - **Get Object** obtains a file from the bucket that the connection accesses. In the **File Name** field, enter the name of the file that you want to obtain. If the specified file is not in the bucket, it is a runtime error.
  - **Poll an Amazon S3 Bucket** periodically obtains files from the bucket that the connection accesses. To configure this action:
    - a. In the **Delay** field, accept the default of 500 milliseconds as the time that elapses between polls. Or, to specify a different polling interval, enter a number and select its time unit.
    - b. In the **Maximum Objects to Retrieve** field, enter the largest number of files that one poll operation can obtain. The default is 10.  
To have no limit on the number of files that can be obtained, specify **0** or a negative integer. When **Maximum Objects to Retrieve** is unlimited, the poll action obtains all files in the bucket.  
  
If the bucket contains more than the specified maximum number of files then the action obtains the files that were most recently modified or created.
    - c. In the **Prefix** field, optionally specify a regular expression that evaluates to a string. If you specify a prefix then this action retrieves a file only when its name starts with that string.
    - d. Indicate whether you want to **Obtain files and then delete them from the bucket**
5. After you configure the action, click **Done** to specify the action's output type.
6. In the **Select Type** field, if the data type does not need to be known, accept **Type specification not required** and then, at the bottom, click **Done**. You do not need to follow the rest of these instructions.  
Otherwise, select one of the following as the schema type:
  - **JSON schema** is a document that describes the structure of JSON data. The document's media type is **application/schema+json**.
  - **JSON instance** is a document that contains JSON data. The document's media type is **application/json**.

- **XML schema** is a document that describes the structure of XML data. The document's file extension is `.xsd`.
  - **XML instance** is a document that contains XML data. The document's file extension is `.xml`.
7. In the **Definition** input box, paste a definition that conforms to the schema type you selected. For example, if you select **JSON schema** then you would paste the content of a JSON schema file, which has a media type of **application/schema+json**.
  8. In the **Data Type Name** field, enter a name that you choose for the data type. For example, suppose you are specifying a JSON schema for vendors. You can specify **Vendor** as the data type name.  
You will see this data type name when you are creating or editing an integration that uses the connection for which you are specifying this type. Fuse Online displays the type name in the integration visualization panel and in the data mapper.
  9. In the **Data Type Description** field, provide information that helps you distinguish this type. This description appears in the data mapper when you hover over the step that processes this type.
  10. Click **Done**.

The connection appears in the integration flow in the location where you added it.

## 2.4. FINISH AN INTEGRATION BY ADDING DATA TO AMAZON S3

To finish an integration by copying data to Amazon S3, add an Amazon S3 connection as the finish connection in an integration.

### Prerequisite

You created an Amazon S3 connection.

### Procedure

1. Start creating the integration.
2. Add and configure the start connection.
3. On the **Choose a Finish Connection** page, click the Amazon S3 connection that you want to use to finish the integration.
4. Click the action that you want the connection to perform:
  - a. **Copy Object** adds one or more objects to the bucket.  
To add one file to the bucket, you can enter its name in the **File Name** field.

To add multiple files to the bucket, do not specify a file name. In this case, the action adds all objects that it obtains from the previous integration step(s).

If you used the poll action to obtain multiple files and you specify a file name then the **Copy Object** action adds only the last file that was received from the poll action.

- b. **Delete Object** deletes an object from the bucket. In the **File Name** field, specify the name of the object that you want to delete. If the specified file is not in the bucket, the integration continues with no error.

5. After you configure the chosen action, click **Next** to specify the action's input type.
6. In the **Select Type** field, if the data type does not need to be known, accept **Type specification not required** and then, at the bottom, click **Done**. You do not need to follow the rest of these instructions.  
Otherwise, select one of the following as the schema type:
  - **JSON schema** is a document that describes the structure of JSON data. The document's media type is **application/schema+json**.
  - **JSON instance** is a document that contains JSON data. The document's media type is **application/json**.
  - **XML schema** is a document that describes the structure of XML data. The document's file extension is **.xsd**.
  - **XML instance** is a document that contains XML data. The document's file extension is **.xml**.
7. In the **Definition** input box, paste a definition that conforms to the schema type you selected. For example, if you select **JSON schema** then you would paste the content of a JSON schema file, which has a media type of **application/schema+json**.
8. In the **Data Type Name** field, enter a name that you choose for the data type. For example, suppose you are specifying a JSON schema for vendors. You can specify **Vendor** as the data type name.  
You will see this data type name when you are creating or editing an integration that uses the connection for which you are specifying this type. Fuse Online displays the type name in the integration visualization panel and in the data mapper.
9. In the **Data Type Description** field, provide information that helps you distinguish this type. This description appears in the data mapper when you hover over the step that processes this type.
10. Click **Done**.

The connection appears in the integration flow in the location where you added it.

## 2.5. ADD DATA TO AMAZON S3 IN THE MIDDLE OF AN INTEGRATION

In the middle of an integration, to add data to Amazon S3, add an Amazon S3 connection between the start and finish connections of an integration.

### Prerequisite

You created an Amazon S3 connection.

### Procedure

1. Add the start and finish connections.
2. In the left panel, hover over the plus sign that is in the location where you want to add the Amazon S3 connection.
3. In the pop-up, click **Add a Connection**.
4. Click the Amazon S3 connection that you want to use as a middle connection in the integration.



5. Click the action that you want the connection to perform:
  - a. **Copy Object** adds one or more objects to the bucket.  
To add one file to the bucket, you can enter its name in the **File Name** field.  
  
To add multiple files to the bucket, do not specify a file name. In this case, the action adds all objects that it obtains from the previous integration step(s).  
  
If you used the poll action to obtain multiple files and you specify a file name then the **Copy Object** action adds only the last file that was received from the poll action.
  - b. **Delete Object** deletes an object from the bucket. In the **File Name** field, specify the name of the object that you want to delete. If the specified file is not in the bucket, the integration continues with no error.
6. After you configure the chosen action, click **Next** to specify the action's input type.
7. In the **Select Type** field, if the data type does not need to be known, accept **Type specification not required** and then, at the bottom, click **Done**. You do not need to follow the rest of these instructions.  
Otherwise, select one of the following as the schema type:
  - **JSON schema** is a document that describes the structure of JSON data. The document's media type is **application/schema+json**.
  - **JSON instance** is a document that contains JSON data. The document's media type is **application/json**.
  - **XML schema** is a document that describes the structure of XML data. The document's file extension is **.xsd**.
  - **XML instance** is a document that contains XML data. The document's file extension is **.xml**.
8. In the **Definition** input box, paste a definition that conforms to the schema type you selected. For example, if you select **JSON schema** then you would paste the content of a JSON schema file, which has a media type of **application/schema+json**.
9. In the **Data Type Name** field, enter a name that you choose for the data type. For example, suppose you are specifying a JSON schema for vendors. You can specify **Vendor** as the data type name.  
You will see this data type name when you are creating or editing an integration that uses the connection for which you are specifying this type. Fuse Online displays the type name in the integration visualization panel and in the data mapper.
10. In the **Data Type Description** field, provide information that helps you distinguish this type. This description appears in the data mapper when you hover over the step that processes this type.
11. Click **Done**.

The connection appears in the integration flow in the location where you added it.

## CHAPTER 3. CONNECT TO AMQ

In an integration, you can obtain messages from an ApacheMQ (AMQ) broker or publish messages to an AMQ broker. AMQ uses the OpenWire protocol for communication between clients and message brokers. To communicate with the following broker types, use the AMQ connector to create a connection to the broker of interest:

- Apache ActiveMQ broker that does not support AMQP
- AMQ 6 broker

To communicate with one of the following broker types, [use the AMQP connector](#) to create a connection to the broker of interest:

- Apache ActiveMQ broker that supports AMQP
- Apache ActiveMQ Artemis
- AMQ 7 broker
- EnMasse, which is an open source messaging platform

To use the AMQ connector, see:

- [Section 3.1, “Create an AMQ connection”](#)
- [Section 3.2, “Start an integration based on receiving AMQ messages”](#)
- [Section 3.3, “Finish an integration by publishing AMQ messages”](#)
- [Section 3.4, “Publish AMQ messages in the middle of an integration”](#)

### 3.1. CREATE AN AMQ CONNECTION

In an integration, to obtain messages from or to publish messages to an Apache ActiveMQ broker that does not support AMQP or an AMQ 6 broker, create an AMQ connection, which you can add to an integration.

#### Prerequisites

For the AMQ broker that you want to connect to, you have the following: \* Broker URL \* User account credentials \* Broker’s PEM certificate text

#### Procedure

1. In Fuse Online, in the left panel, click **Connections** to display any available connections.
2. In the upper right, click **Create Connection** to display connectors.
3. Click the **AMQ Message Broker** connector.
4. Configure the connection by entering:
  - a. In the **Broker URL** field, enter the location that you want to send data to or obtain data from, for example, **tcp://localhost:61616**.

- b. In the **User Name** field, enter the user name for the account that you want to use to access this broker.
  - c. In the **Password** field, enter the password for the account that you want to use to access this broker.
  - d. In the **Client ID** field, enter the ID that allows connections to close and reopen without missing messages. The destination type must be a topic.
  - e. If this connection will be used in a development environment, you can save some time by disabling **Check Certificates**. Disabling the checking of certificates is a convenience for development environments. For secure production environments, always enable **Check Certificates**.
  - f. In the **Broker Certificate** field, paste the broker's PEM certificate text. This is required except when you disable checking the certificates.
  - g. In the **Client Certificate** field, paste the client's PEM certificate text. Content in this field is always optional.
5. Click **Validate**. Fuse Online immediately tries to validate the connection and displays a message that indicates whether validation is successful. If validation fails, revise the configuration details as needed and try again.
  6. If validation is successful, click **Next**.
  7. In the **Connection Name** field, enter your choice of a name that helps you distinguish this connection from any other connections. For example, you might enter **AMQ 1**.
  8. In the **Description** field, optionally enter any information that is helpful to know about this connection. For example, enter **Sample AMQ connection that uses a provided broker**.
  9. In the upper right, click **Create** to see that the connection you created is now available. If you entered the example name, you would see that **AMQ 1** appears as a connection that you can choose to add to an integration.

## 3.2. START AN INTEGRATION BASED ON RECEIVING AMQ MESSAGES

To trigger execution of an integration based on receiving a message from an AMQ broker, add an AMQ connection as the start connection.

### Prerequisite

You created an AMQ connection to the broker that you want to obtain messages from.

### Procedure

1. In the Fuse Online panel on the left, click **Integrations**.
2. Click **Create Integration**.
3. On the **Choose a Start Connection** page, click the AMQ connection that you want to use to start the integration.

4. On the **Choose an Action** page, click the **Subscribe for messages** action to receive messages from the queue or topic you specify.
5. To configure the action:
  - a. In the **Destination Name** field, enter the name of the queue or topic to receive data from.
  - b. For the **Destination Type**, accept **Queue** or select **Topic**.
  - c. In the **Durable Subscription ID** field, to allow connections to close and reopen without missing messages, enter the durable subscription ID. The destination type must be a topic.
  - d. In the **Message Selector** field, if you want to receive only data that satisfies a particular condition, enter a filter expression.  
A message selector is a string that contains an expression. The syntax of the expression is based on a subset of the SQL92 conditional expression syntax. The message selector in the following example selects any message that has a **NewsType** property whose value is set to **Sports** or **Opinion**:

```
NewsType = 'Sports' OR NewsType = 'Opinion'
```

The message consumer receives only those messages whose headers and properties match the message selector expression. A message selector cannot select messages on the basis of the content of the message body.

6. Click **Next** to specify the action's output type.
7. In the **Select Type** field, if the data type does not need to be known, accept **Type specification not required** and then, at the bottom, click **Done**. You do not need to follow the rest of these instructions.  
Otherwise, select one of the following as the schema type:
  - **JSON schema** is a document that describes the structure of JSON data. The document's media type is **application/schema+json**.
  - **JSON instance** is a document that contains JSON data. The document's media type is **application/json**.
  - **XML schema** is a document that describes the structure of XML data. The document's file extension is **.xsd**.
  - **XML instance** is a document that contains XML data. The document's file extension is **.xml**.
8. In the **Definition** input box, paste a definition that conforms to the schema type you selected. For example, if you select **JSON schema** then you would paste the content of a JSON schema file, which has a media type of **application/schema+json**.
9. In the **Data Type Name** field, enter a name that you choose for the data type. For example, suppose you are specifying a JSON schema for vendors. You can specify **Vendor** as the data type name.  
You will see this data type name when you are creating or editing an integration that uses the connection for which you are specifying this type. Fuse Online displays the type name in the integration visualization panel and in the data mapper.
10. In the **Data Type Description** field, provide information that helps you distinguish this type. This description appears in the data mapper when you hover over the step that processes this type.

11. Click **Done**.

## Result

The connection appears in the integration flow in the location where you added it.

## 3.3. FINISH AN INTEGRATION BY PUBLISHING AMQ MESSAGES

To finish an integration by publishing messages to an AMQ broker, add an AMQ connection as the finish connection.

### Prerequisites

- You created a connection to the AMQ broker that you want to publish messages to.
- You are creating an integration and Fuse Online is prompting you to choose the finish connection, or you are editing an integration to change the finish connection.

### Procedure

1. On the **Choose a Finish Connection** page, click the AMQ connection that you want to use to finish the integration.
2. On the **Choose an Action** page, click the **Publish messages** action to publish messages to the queue or topic you specify.
3. In the **Destination Name** field, enter the name of the queue or topic to send messages to.
4. For the **Destination Type**, accept **Queue** or select **Topic**.
5. Select **Persistent** to guarantee message delivery even if a connection fails.
6. Click **Next** to specify the action's input/output type.
7. In the **Select Type** field, if the data type does not need to be known, accept **Type specification not required** and then, at the bottom, click **Done**. You do not need to follow the rest of these instructions.

Otherwise, select one of the following as the schema type:

- **JSON schema** is a document that describes the structure of JSON data. The document's media type is **application/schema+json**.
  - **JSON instance** is a document that contains JSON data. The document's media type is **application/json**.
  - **XML schema** is a document that describes the structure of XML data. The document's file extension is **.xsd**.
  - **XML instance** is a document that contains XML data. The document's file extension is **.xml**.
8. In the **Definition** input box, paste a definition that conforms to the schema type you selected. For example, if you select **JSON schema** then you would paste the content of a JSON schema file, which has a media type of **application/schema+json**.

9. In the **Data Type Name** field, enter a name that you choose for the data type. For example, suppose you are specifying a JSON schema for vendors. You can specify **Vendor** as the data type name.  
You will see this data type name when you are creating or editing an integration that uses the connection for which you are specifying this type. Fuse Online displays the type name in the integration visualization panel and in the data mapper.
10. In the **Data Type Description** field, provide information that helps you distinguish this type. This description appears in the data mapper when you hover over the step that processes this type.
11. Click **Done**.

## Result

The connection appears in the integration flow in the location where you added it.

## 3.4. PUBLISH AMQ MESSAGES IN THE MIDDLE OF AN INTEGRATION

In the middle of an integration, to publish messages to an AMQ broker, add an AMQ connection between the start and finish connections.

### Prerequisites

- You created a connection to the AMQ broker that you want to publish messages to.
- You are creating or editing an integration.
- The integration's start and finish connections have already been added.

### Procedure

1. In the integration visualization panel on the left, click the plus sign that is in the location where you want to add the connection.
2. Click **Add a connection**.
3. On the **Choose a Connection** page, click the AMQ connection that you want the integration to use after the start connection and before the finish connection.
4. On the **Choose an Action** page, select one of the following actions:
  - **Publish messages** action to publish messages to the queue or topic you specify. To configure this action:
    - a. In the **Destination Name** field, enter the name of the queue or topic to send messages to.
    - b. For the **Destination Type**, accept **Queue** or select **Topic**.
    - c. Select **Persistent** to guarantee message delivery even if a connection fails.
  - **Request response using messages** to send messages to the JMS destination you specify and receive a response. To configure this action:
    - a. In the **Destination Name** field, enter the name of the queue or topic to send messages to.

- b. For the **Destination Type**, accept **Queue** or select **Topic**.
  - c. In the **Message Selector** field, if you want to receive only responses that satisfy a particular condition, enter a filter expression.  
A message selector is a string that contains an expression. The syntax of the expression is based on a subset of the SQL92 conditional expression syntax. The message selector in the following example selects any message that has a **NewsType** property whose value is set to **Sports** or **Opinion**:  
  
**NewsType = 'Sports' OR NewsType = 'Opinion'**  
  
The message consumer receives only those messages whose headers and properties match the message selector expression. A message selector cannot select messages on the basis of the content of the message body.
  - d. In the **Named Reply To** field, enter the name of a queue or topic. The destination sends its response to this queue or topic.
  - e. Select **Persistent** to guarantee message delivery even if a connection fails.
  - f. In the **Response Time Out** field, specify the number of milliseconds that this connection waits for a response message before throwing a runtime exception. The default is 5000 milliseconds (5 seconds).
5. Click **Next** to specify the action's input type and then the action's output type.
  6. In the **Select Type** field, if the data type does not need to be known, accept **Type specification not required** and then, at the bottom, click **Done**. You do not need to follow the rest of these instructions.  
Otherwise, select one of the following as the schema type:
    - **JSON schema** is a document that describes the structure of JSON data. The document's media type is **application/schema+json**.
    - **JSON instance** is a document that contains JSON data. The document's media type is **application/json**.
    - **XML schema** is a document that describes the structure of XML data. The document's file extension is **.xsd**.
    - **XML instance** is a document that contains XML data. The document's file extension is **.xml**.
  7. In the **Definition** input box, paste a definition that conforms to the schema type you selected. For example, if you select **JSON schema** then you would paste the content of a JSON schema file, which has a media type of **application/schema+json**.
  8. In the **Data Type Name** field, enter a name that you choose for the data type. For example, suppose you are specifying a JSON schema for vendors. You can specify **Vendor** as the data type name.  
You will see this data type name when you are creating or editing an integration that uses the connection for which you are specifying this type. Fuse Online displays the type name in the integration visualization panel and in the data mapper.
  9. In the **Data Type Description** field, provide information that helps you distinguish this type. This description appears in the data mapper when you hover over the step that processes this type.

10. Click **Done**.

### **Result**

The connection appears in the integration flow in the location where you added it.



## CHAPTER 4. CONNECT TO AMQP

In an integration, you can obtain messages from or publish messages to an Advanced Message Queue Protocol (AMQP) broker. AMQP defines communication between clients and message brokers. To communicate with the following broker types, use the AMQP connector to create a connection to the broker of interest:

- Apache ActiveMQ broker that supports AMQP
- Apache ActiveMQ Artemis
- AMQ 7 broker
- EnMasse, which is an open source messaging platform

To communicate with one of the following broker types, [use the AMQ connector](#) to create a connection to the broker of interest:

- Apache ActiveMQ broker that does not support AMQP
- AMQ 6 broker



### NOTE

It is possible to use the AMQP connector to create a connection to an Apache ActiveMQ broker that does not support AMQP or to an AMQ 6 broker. Doing this requires transport configuration in the broker. For information about configuring the broker, see [Red Hat JBoss A-MQ Managing and Monitoring Brokers, Adding Client Connection Points](#). For information about the configuration values to specify, see [Red Hat JBoss A-MQ Connection Reference, Advanced Message Queuing Protocol \(AMQP\)](#).

To use the AMQP connector, see:

- [Section 4.1, “Create an AMQP connection”](#)
- [Section 4.2, “Start an integration based on receiving AMQP messages”](#)
- [Section 4.3, “Finish an integration by publishing AMQP messages”](#)
- [Section 4.4, “Publish messages to AMQP in the middle of an integration”](#)

### 4.1. CREATE AN AMQP CONNECTION

In an integration, to obtain messages from or publish messages to an AMQP broker, create an AMQP connection, which you can add to an integration.

#### Prerequisites

For the AMQP broker that you want to connect to, you have the following:

- Its URI
- User account credentials
- Its PEM certificate text

## Procedure

1. In Fuse Online, in the left panel, click **Connections** to display any available connections.
2. In the upper right, click **Create Connection** to display connectors.
3. Click the **AMQP Message Broker** connector.
4. Configure the connection by entering:
  - a. In the **Connection URI** field, enter the location you want to send data to or obtain data from.
  - b. In the **User Name** field, enter the user name for the account that you want to use to access this broker.
  - c. In the **Password** field, enter the password for the account that you want to use to access this broker.
  - d. In the **Client ID** field, enter the ID that allows connections to close and reopen without missing messages. The destination type must be a topic.
  - e. If this connection will be used in a development environment, you can save some time by disabling **Check Certificates**. Disabling the checking of certificates is a convenience for development environments. For secure production environments, always enable **Check Certificates**.
  - f. In the **Broker Certificate** field, paste the broker's PEM certificate text. This is required except when disable checking the certificates.
  - g. In the **Client Certificate** field, paste the client's PEM certificate text. Content in this field is always optional.
5. Click **Validate**. Fuse Online immediately tries to validate the connection and displays a message that indicates whether validation is successful. If validation fails, revise the configuration details as needed and try again.
6. If validation is successful, click **Next**.
7. In the **Connection Name** field, enter your choice of a name that helps you distinguish this connection from any other connections. For example, you might enter **AMQP 1**.
8. In the **Description** field, optionally enter any information that is helpful to know about this connection. For example, enter **Sample AMQP connection**.
9. In the upper right, click **Create** to see that the connection you created is now available. If you entered the example name, you would see that **AMQP 1** appears as a connection that you can choose to add to an integration.

## 4.2. START AN INTEGRATION BASED ON RECEIVING AMQP MESSAGES

To trigger execution of an integration based on receiving messages from an AMQP broker, add an AMQP connection as the start connection.

### Prerequisites

- You created a connection to the AMQP broker that you want to receive messages from.
- You are creating an integration and Fuse Online is prompting you to choose the start connection.

## Procedure

1. On the **Choose a Start Connection** page, click the AMQP connection that you want to use to start the integration.
2. On the **Choose an Action** page, click the **Subscribe for messages** action to receive messages from the queue or topic you specify.
3. To configure the action:
  - a. In the **Destination Name** field, enter the name of the queue or topic to receive data from.
  - b. For the **Destination Type**, accept **Queue** or select **Topic**.
  - c. In the **Durable Subscription ID** field, to allow connections to close and reopen without missing messages, enter the durable subscription ID. The destination type must be a topic.
  - d. In the **Message Selector** field, if you want to receive only data that satisfies a particular condition, enter a filter expression.

A message selector is a string that contains an expression. The syntax of the expression is based on a subset of the SQL92 conditional expression syntax. The message selector in the following example selects any message that has a **NewsType** property whose value is set to **Sports** or **Opinion**:

```
NewsType = 'Sports' OR NewsType = 'Opinion'
```

The message consumer receives only those messages whose headers and properties match the message selector expression. A message selector cannot select messages on the basis of the content of the message body.

4. Click **Next** to specify the action's output type.
5. In the **Select Type** field, if the data type does not need to be known, accept **Type specification not required** and then, at the bottom, click **Done**. You do not need to follow the rest of these instructions.

Otherwise, select one of the following as the schema type:

- **JSON schema** is a document that describes the structure of JSON data. The document's media type is **application/schema+json**.
  - **JSON instance** is a document that contains JSON data. The document's media type is **application/json**.
  - **XML schema** is a document that describes the structure of XML data. The document's file extension is **.xsd**.
  - **XML instance** is a document that contains XML data. The document's file extension is **.xml**.
6. In the **Definition** input box, paste a definition that conforms to the schema type you selected. For example, if you select **JSON schema** then you would paste the content of a JSON schema file, which has a media type of **application/schema+json**.

7. In the **Data Type Name** field, enter a name that you choose for the data type. For example, suppose you are specifying a JSON schema for vendors. You can specify **Vendor** as the data type name.  
You will see this data type name when you are creating or editing an integration that uses the connection for which you are specifying this type. Fuse Online displays the type name in the integration visualization panel and in the data mapper.
8. In the **Data Type Description** field, provide information that helps you distinguish this type. This description appears in the data mapper when you hover over the step that processes this type.
9. Click **Done**.

## Result

The connection appears in the integration flow in the location where you added it.

## 4.3. FINISH AN INTEGRATION BY PUBLISHING AMQP MESSAGES

To finish an integration by publishing messages to an AMQP broker, add an AMQP connection as the finish connection.

### Prerequisites

- You created an AMQP connection to the broker that you want to publish messages to.
- You are creating or editing an integration.
- The integration has its the start connection.

### Procedure

1. On the **Choose a Finish Connection** page, click the AMQP connection that you want to use to finish the integration.
2. On the **Choose an Action** page, click **Publish messages** to publish messages to the queue or topic you specify.
3. In the **Destination Name** field, enter the name of the queue or topic to send messages to.
4. For the **Destination Type**, accept **Queue** or select **Topic**.
5. Select **Persistent** to guarantee message delivery even if a connection fails.
6. Click **Next** to specify the action's input and output type.
7. In the **Select Type** field, if the data type does not need to be known, accept **Type specification not required** and then, at the bottom, click **Done**. You do not need to follow the rest of these instructions.  
Otherwise, select one of the following as the schema type:
  - **JSON schema** is a document that describes the structure of JSON data. The document's media type is **application/schema+json**.
  - **JSON instance** is a document that contains JSON data. The document's media type is **application/json**.

- **XML schema** is a document that describes the structure of XML data. The document's file extension is `.xsd`.
  - **XML instance** is a document that contains XML data. The document's file extension is `.xml`.
8. In the **Definition** input box, paste a definition that conforms to the schema type you selected. For example, if you select **JSON schema** then you would paste the content of a JSON schema file, which has a media type of **application/schema+json**.
  9. In the **Data Type Name** field, enter a name that you choose for the data type. For example, suppose you are specifying a JSON schema for vendors. You can specify **Vendor** as the data type name.  
You will see this data type name when you are creating or editing an integration that uses the connection for which you are specifying this type. Fuse Online displays the type name in the integration visualization panel and in the data mapper.
  10. In the **Data Type Description** field, provide information that helps you distinguish this type. This description appears in the data mapper when you hover over the step that processes this type.
  11. Click **Done**.

## Result

The connection appears in the integration flow in the location where you added it.

## 4.4. PUBLISH MESSAGES TO AMQP IN THE MIDDLE OF AN INTEGRATION

In the middle of an integration, to publish messages to an AMQP broker, add an AMQP connection between the start and finish connections.

### Prerequisites

- You created a connection to the AMQP broker that you want to publish messages to.
- You are creating or editing an integration.
- You already added the start and finish connections to the integration.

### Procedure

1. In the integration visualization panel on the left, click the plus sign in the location where you want to add the connection.
2. Click **Add a connection**.
3. On the **Choose a Connection** page, click the AMQP connection that you want the integration to use after the start connection and before the finish connection.
4. On the **Choose an Action** page, select one of the following actions:
  - **Publish messages** to publish messages to the queue or topic you specify. To configure this action:

- a. In the **Destination Name** field, enter the name of the queue or topic to send messages to.
- b. For the **Destination Type**, accept **Queue** or select **Topic**.
- c. Select **Persistent** to guarantee message delivery even if a connection fails.
- **Request response using messages** to send messages to the JMS destination you specify and receive a response. To configure this action:
  - a. In the **Destination Name** field, enter the name of the queue or topic to send messages to.
  - b. For the **Destination Type**, accept **Queue** or select **Topic**.
  - c. In the **Message Selector** field, if you want to receive only responses that satisfy a particular condition, enter a filter expression.

A message selector is a string that contains an expression. The syntax of the expression is based on a subset of the SQL92 conditional expression syntax. The message selector in the following example selects any message that has a **NewsType** property whose value is set to **Sports** or **Opinion**:

```
NewsType = 'Sports' OR NewsType = 'Opinion'
```

The message consumer receives only those messages whose headers and properties match the message selector expression. A message selector cannot select messages on the basis of the content of the message body.
  - d. In the **Named Reply To** field, enter the name of a queue or topic. The destination sends its response to this queue or topic.
  - e. Select **Persistent** to guarantee message delivery even if a connection fails.
  - f. In the **Response Time Out** field, specify the number of milliseconds that this connection waits for a response message before throwing a runtime exception. The default is 5000 milliseconds (5 seconds).
5. Click **Next** to specify the action's input and output type.
6. In the **Select Type** field, if the data type does not need to be known, accept **Type specification not required** and then, at the bottom, click **Done**. You do not need to follow the rest of these instructions.

Otherwise, select one of the following as the schema type:

  - **JSON schema** is a document that describes the structure of JSON data. The document's media type is **application/schema+json**.
  - **JSON instance** is a document that contains JSON data. The document's media type is **application/json**.
  - **XML schema** is a document that describes the structure of XML data. The document's file extension is **.xsd**.
  - **XML instance** is a document that contains XML data. The document's file extension is **.xml**.

7. In the **Definition** input box, paste a definition that conforms to the schema type you selected. For example, if you select **JSON schema** then you would paste the content of a JSON schema file, which has a media type of **application/schema+json**.
8. In the **Data Type Name** field, enter a name that you choose for the data type. For example, suppose you are specifying a JSON schema for vendors. You can specify **Vendor** as the data type name.  
You will see this data type name when you are creating or editing an integration that uses the connection for which you are specifying this type. Fuse Online displays the type name in the integration visualization panel and in the data mapper.
9. In the **Data Type Description** field, provide information that helps you distinguish this type. This description appears in the data mapper when you hover over the step that processes this type.
10. Click **Done**.

## Result

The connection appears in the integration flow in the location where you added it.

## CHAPTER 5. CONNECT TO DROPBOX

In an integration, you can download files from Dropbox or upload files to Dropbox. The following topics provide the details:

- [Section 5.1, “Register Fuse Online as a Dropbox client”](#)
- [Section 5.2, “Create a Dropbox connection”](#)
- [Section 5.3, “Start an integration by obtaining files from Dropbox”](#)
- [Section 5.4, “Finish an integration by adding files to Dropbox”](#)
- [Section 5.5, “Access Dropbox in the middle of an integration”](#)

### 5.1. REGISTER FUSE ONLINE AS A DROPBOX CLIENT


You must register your Fuse Online environment as a client application that can access Dropbox. This lets you create any number of integrations that connect to Dropbox. In other words, you need to register a particular Fuse Online environment with Dropbox only once.

In each Fuse Online environment, there can be only one registration of Fuse Online as a Dropbox client application. However, while each Dropbox connection uses the same registration, it can use different user credentials.

#### Prerequisite

You can sign in to the Dropbox account that you want an integration to use to download or upload files.

#### Procedure

1. In Fuse Online:
  - a. In the left panel, click **Settings**.
  - b. On the **Settings** page, near the top, to the right of the callback URL, click  to copy the callback URL for your Fuse Online environment to the clipboard. You will need this URL toward the end of this procedure.
2. In another browser tab, go to <https://www.dropbox.com> and do the following:
  - a. Sign in to the Dropbox account that has the data that you want to access in an integration.
  - b. After signing in, go to <https://www.dropbox.com/developers/apps>.
  - c. Click **Create App**.
  - d. Select **Dropbox API**.
  - e. Near the top of the page, in the sentence that starts with **During registration, enter this callback URL:**, paste the URL that you copied to the clipboard at the beginning of this procedure. For example, the URL that you paste is something like this: **`https://app-proj9128.7b63.fuse-ignite.openshiftapps.com/api/v1/credentials/callback`**.
  - f. Choose whether Fuse Online can access a single folder or all of the folders and files.



- g. Specify a name for your Dropbox app. For example, you might specify **Fuse Online Access From Aslan LLC**. The name you specify must be unique in the set of Dropbox app names.
- h. Check the box to indicate that you agree to Dropbox API terms and conditions.
- i. Click **Create App**.
- j. In the Dropbox **Settings** page for your new app, in the input field for **OAuth2 Redirect URIs**, paste your Fuse Online URL, which you copied to the clipboard at the beginning of this procedure.
- k. Click **Add**.

## Result

Your Fuse Online environment is now registered as a Dropbox client, which means that Fuse Online can access content in the Dropbox account that you signed into.

## 5.2. CREATE A DROPBOX CONNECTION

In an integration, to download or upload Dropbox files, create a Dropbox connection, which you can add to an integration. You can add the same connection to any number of integrations.

### Prerequisite

You registered your Fuse Online environment as an application that can access Dropbox.

### Procedure

1. In a new browser tab, go to <https://www.dropbox.com> and do the following:
  - a. Sign in to the Dropbox account in which you created the app that registers access from your Fuse Online environment.
  - b. Go to <https://www.dropbox.com/developers/apps>.
  - c. Click the Fuse Online app to display its settings.
2. In another browser tab, in Fuse Online, do the following:
  - a. In the left panel, click **Connections** to display any available connections.
  - b. In the upper right, click **Create Connection** to display the available connectors.
  - c. Click the **Dropbox** connector.
3. Go back to the Dropbox settings display for your app and do the following:
  - a. Scroll down to see **Generated Access Token**.
  - b. Click **Generate**.
  - c. Copy the generated access token to the clipboard.
4. Back in Fuse Online, in the **Configure Connection** page, in the **Access Token** field, paste the generated access token.

5. In the **Client Identifier** field, enter the name that you specified when you created the Dropbox app.
6. Click **Validate**. Fuse Online displays a message that indicates whether it can validate this connection. If validation fails, try again and be sure to enter the correct values.
7. When validation is successful, in the upper right, click **Next**.
8. In the **Connection Name** field, enter your choice of a name that helps you distinguish this connection from any other connections. For example, enter **Dropbox Connect 1**.
9. In the **Description** field, optionally enter any information that is helpful to know about this connection. For example, enter **Sample Dropbox connection that can access all content in our company Dropbox account**.
10. In the upper right, click **Create** to see that the connection you created is now available. If you entered the example name, you would see that **Dropbox Connect 1** appears as a connection that you can choose to add to an integration.

### 5.3. START AN INTEGRATION BY OBTAINING FILES FROM DROPBOX

To start an integration by downloading files from Dropbox, add a Dropbox connection as the start connection.

#### Prerequisite

You created a Dropbox connection.

#### Procedure

1. In the Fuse Online panel on the left, click **Integrations**.
2. Click **Create Integration**.
3. On the **Choose a Start Connection** page, click the Dropbox connection that you want to use to start the integration.
4. On the **Choose an Action** page, click the **Download** action to obtain one or more files from the Dropbox account that this connection accesses.
5. To configure the action, in the **Folder or file name path to download** field, specify the filename path for the content that you want the integration to obtain. In this release, you can download only a single file.
6. Click **Next** to specify the action's output type.
7. In the **Select Type** field, if the data type does not need to be known, accept **Type specification not required** and then, at the bottom, click **Done**. You do not need to follow the rest of these instructions.  
Otherwise, select one of the following as the schema type:
  - **JSON schema** is a document that describes the structure of JSON data. The document's media type is **application/schema+json**.
  - **JSON instance** is a document that contains JSON data. The document's media type is **application/json**.

- **XML schema** is a document that describes the structure of XML data. The document's file extension is `.xsd`.
  - **XML instance** is a document that contains XML data. The document's file extension is `.xml`.
8. In the **Definition** input box, paste a definition that conforms to the schema type you selected. For example, if you select **JSON schema** then you would paste the content of a JSON schema file, which has a media type of **application/schema+json**.
  9. In the **Data Type Name** field, enter a name that you choose for the data type. For example, suppose you are specifying a JSON schema for vendors. You can specify **Vendor** as the data type name.  
You will see this data type name when you are creating or editing an integration that uses the connection for which you are specifying this type. Fuse Online displays the type name in the integration visualization panel and in the data mapper.
  10. In the **Data Type Description** field, provide information that helps you distinguish this type. This description appears in the data mapper when you hover over the step that processes this type.
  11. Click **Done**.

## Result

The connection appears in the integration flow in the location where you added it.

## 5.4. FINISH AN INTEGRATION BY ADDING FILES TO DROPBOX

To finish an integration by uploading files to Dropbox, add a Dropbox connection as the finish connection.

### Prerequisites

- You created a Dropbox connection.
- You are creating or editing an integration and Fuse Online is prompting you to choose the finish connection.

### Procedure

1. On the **Choose a Finish Connection** page, click the Dropbox connection that you want to use to finish the integration.
2. On the **Choose an Action** page, click the **Upload** action to add the current integration data to the Dropbox account that this connection accesses.
3. In the **Remote Path** field, enter the local filename path for file that you want to upload. Dropbox stores the file with the same path and name. In this release, you can upload only a single file.
4. For the **Upload mode**,
  - Select **Add** to upload a file only when a file with the same name is not already in the same Dropbox folder. If a file with the same name is already in the same Dropbox folder, then the file is not uploaded and the integration continues. This is the behavior regardless of whether the content in the file you are trying to upload has been updated.

- Select **Force** to ensure that the file is uploaded even if a file with the same name is present in the same Dropbox folder. Dropbox overwrites the file that it already has with the file that you are uploading.
5. Click **Next** to specify the action's input type.
  6. In the **Select Type** field, if the data type does not need to be known, accept **Type specification not required** and then, at the bottom, click **Done**. You do not need to follow the rest of these instructions.  
Otherwise, select one of the following as the schema type:
    - **JSON schema** is a document that describes the structure of JSON data. The document's media type is **application/schema+json**.
    - **JSON instance** is a document that contains JSON data. The document's media type is **application/json**.
    - **XML schema** is a document that describes the structure of XML data. The document's file extension is **.xsd**.
    - **XML instance** is a document that contains XML data. The document's file extension is **.xml**.
  7. In the **Definition** input box, paste a definition that conforms to the schema type you selected. For example, if you select **JSON schema** then you would paste the content of a JSON schema file, which has a media type of **application/schema+json**.
  8. In the **Data Type Name** field, enter a name that you choose for the data type. For example, suppose you are specifying a JSON schema for vendors. You can specify **Vendor** as the data type name.  
You will see this data type name when you are creating or editing an integration that uses the connection for which you are specifying this type. Fuse Online displays the type name in the integration visualization panel and in the data mapper.
  9. In the **Data Type Description** field, provide information that helps you distinguish this type. This description appears in the data mapper when you hover over the step that processes this type.
  10. Click **Done**.

## Result

The connection appears in the integration flow in the location where you added it.

## 5.5. ACCESS DROPBOX IN THE MIDDLE OF AN INTEGRATION

To upload a file to Dropbox in the middle of an integration, add a Dropbox connection between the start and finish connections.

### Prerequisites

- You created a Dropbox connection.
- You are creating or editing an integration.
- The integration already has its start and finish connections.

## Procedure

1. In the integration visualization panel on the left, hover over the plus sign that is in the location where you want to add a Dropbox connection.
2. In the popup, click **Add a connection**.
3. On the **Choose a Connection** page, click the Dropbox connection that you want the integration to use.
4. On the **Choose an Action** page, click **Upload** to add the current integration data to the Dropbox account that this connection accesses. To configure this action:
  - a. In the **Remote Path** field, specify the local path and file name of the file you want to upload. Dropbox stores the file with the same path and name. In this release, you can upload only a single file.
  - b. For the **Upload mode**,
    - Select **Add** to upload a file only when a file with the same name is not already in the same Dropbox folder. If a file with the same name is already in the same Dropbox folder, then the file is not uploaded and the integration continues. This is the behavior regardless of whether the content in the file you are trying to upload has been updated.
    - Select **Force** to ensure that the file is uploaded even if a file with the same name is present in the same Dropbox folder. Dropbox overwrites the file that it already has with the file that you are uploading.
5. Click **Next** to specify the action's input and output type.
6. In the **Select Type** field, if the data type does not need to be known, accept **Type specification not required** and then, at the bottom, click **Done**. You do not need to follow the rest of these instructions.  
Otherwise, select one of the following as the schema type:
  - **JSON schema** is a document that describes the structure of JSON data. The document's media type is **application/schema+json**.
  - **JSON instance** is a document that contains JSON data. The document's media type is **application/json**.
  - **XML schema** is a document that describes the structure of XML data. The document's file extension is **.xsd**.
  - **XML instance** is a document that contains XML data. The document's file extension is **.xml**.
7. In the **Definition** input box, paste a definition that conforms to the schema type you selected. For example, if you select **JSON schema** then you would paste the content of a JSON schema file, which has a media type of **application/schema+json**.
8. In the **Data Type Name** field, enter a name that you choose for the data type. For example, suppose you are specifying a JSON schema for vendors. You can specify **Vendor** as the data type name.  
You will see this data type name when you are creating or editing an integration that uses the connection for which you are specifying this type. Fuse Online displays the type name in the integration visualization panel and in the data mapper.

9. In the **Data Type Description** field, provide information that helps you distinguish this type. This description appears in the data mapper when you hover over the step that processes this type.
10. Click **Done**.

## **Result**

The connection appears in the integration flow in the location where you added it.

## CHAPTER 6. CONNECT TO AN FTP OR SFTP SERVER

In an integration, you can connect to an FTP or SFTP server to download or upload files. To do this, create an FTP or SFTP connection. You can then add this connection to any number of integrations. The following topics provide details:

- [Section 6.1, “Create an FTP or SFTP connection”](#)
- [Section 6.2, “Obtain files from an FTP or SFTP server”](#)
- [Section 6.3, “Upload files to an FTP or SFTP server”](#)

### 6.1. CREATE AN FTP OR SFTP CONNECTION

In an integration, to download or upload files from/to an FTP or SFTP server, create a connection to that FTP or SFTP server. You can add the same connection to any number of integrations.

#### Prerequisite

You must know the host name of the server you want to connect to.

#### Procedure

1. In Fuse Online, in the left panel, click **Connections** to display any available connections.
2. In the upper right, click **Create Connection** to display Fuse Online connectors.
3. To create a connection that uses File Transfer Protocol, click the **FTP** connector.  
Or, to create a connection that uses Secure File Transfer Protocol, click the **SFTP** connector.
4. Configure the connection.
  - For an FTP connection:
    - **Host** is the only parameter that you must specify. Enter the host name of the server that you want to connect to. For example, if the name of your FTP host is **FTP.WEST**, then you would enter exactly that, **FTP.WEST**. Do not specify the protocol, for example, you should not specify something like this: **ftp://FTP.WEST**.
    - **Port** is required and has a default value of **21**. This is the port that the FTP server is listening on.
    - All other parameters are either not required or have default values. The defaults are suitable for most integrations. Descriptions of these parameters are after this procedure.
  - For an SFTP connection, there must be values for these parameters:
    - **Host** is the host name of the SFTP server that you want to connect to. For example, if the name of your SFTP host is **SFTP.EAST**, then you would enter exactly that, **SFTP.EAST**. Do not specify the protocol, for example, you should not specify something like this: **sftp://SFTP.EAST**.
    - **Port** has a default of **22**. This is the port that the SFTP server is listening on.
    - **User name** of the account that you want to use to access the SFTP server.

- **Password** that is associated with that user name.
  - All other parameters have default values. The defaults are suitable for most integrations. Descriptions of these parameters are after this procedure.
5. Click **Validate**. Fuse Online immediately tries to validate the connection and displays a message that indicates whether or not validation is successful. If validation fails, revise the configuration details as needed and try again.
  6. When validation is successful, click **Next**.
  7. In the **Connection Name** field, enter your choice of a name that helps you distinguish this connection from any other connections. For example, enter **XLIGHT FTP Server**.
  8. In the **Description** field, optionally enter any information that is helpful to know about this connection.
  9. In the upper right, click **Create** to see that the connection you created is now available. If you entered the example name, you would see that **XLIGHT FTP Server** appears as a connection that you can choose to add to an integration.

### Descriptions of other parameters

- **Connect timeout** defaults to **10000** milliseconds and indicates a maximum wait of 10 seconds to establish the connection. If 10 seconds elapse without a connection then Fuse Online waits for the number of milliseconds defined by **Reconnect delay** and then tries to reconnect.
- **Reconnect delay** defaults to **1000** milliseconds and indicates the wait time before trying to reconnect again.
- **Maximum reconnect attempts** defaults to **3**. Fuse Online tries as many as 3 times to establish a connection.
- **Binary file transfer mode** is used by default. Select **No** for ASCII transfer mode.
- **Passive connection mode** defaults to **Yes**, which is usually the preferred mode. In passive mode, the client opens communication channels with the server as a way to avoid firewall issues. If you select **No** then active mode is used.
- **Disconnect from the server after use** defaults to **No**. The connection remains established after it performs the action. Select **Yes** if you want to disconnect from the server after the connection performs the upload or download.
- **Data timeout** defaults to **30000** milliseconds and indicates the maximum length of time that Fuse Online waits for a reply.

## 6.2. OBTAIN FILES FROM AN FTP OR SFTP SERVER

To trigger integration execution when an FTP or SFTP connection finds the file(s) you are interested in, add an FTP or SFTP connection as an integration's start connection.

### Prerequisite

You created an FTP or SFTP connection.

### Procedure



1. In the Fuse Online panel on the left, click **Integrations**.
2. Click **Create Integration**.
3. On the **Choose a Start Connection** page, click the FTP or SFTP connection that you want to use to poll an FTP or SFTP server.
4. On the **Choose an Action** page, click **Download**.
5. In the **File name expression** field, if you are interested in a single file, then enter an [Apache Camel Simple language](#) expression that resolves to a file name. You cannot specify a regular expression. The connection polls (periodically checks) the server for this file and downloads it when it is found. Leave this field blank if you want to download more than one file.
6. In the **FTP directory** field, enter the absolute or relative path of the server directory to poll. The connection watches this directory for any content and downloads all files when it finds any content.
7. In the **Milliseconds before polling starts** field, accept the default of **1000** milliseconds or change the number of milliseconds.
8. In the **Milliseconds before the next poll** field, accept the default of **500** milliseconds or change the number of milliseconds. This is the interval between polls.
9. In the **Delete after download** field, accept the default of **No** or select **Yes** to download the file(s) and then delete it(them) from the server.
10. Click **Next** to specify the action's output type.
11. In the **Select Type** field, if the data type does not need to be known, accept **Type specification not required** and then, at the bottom, click **Done**. You do not need to follow the rest of these instructions.  
Otherwise, select one of the following as the schema type:
  - **JSON schema** is a document that describes the structure of JSON data. The document's media type is **application/schema+json**.
  - **JSON instance** is a document that contains JSON data. The document's media type is **application/json**.
  - **XML schema** is a document that describes the structure of XML data. The document's file extension is **.xsd**.
  - **XML instance** is a document that contains XML data. The document's file extension is **.xml**.
12. In the **Definition** input box, paste a definition that conforms to the schema type you selected. For example, if you select **JSON schema** then you would paste the content of a JSON schema file, which has a media type of **application/schema+json**.
13. In the **Data Type Name** field, enter a name that you choose for the data type. For example, suppose you are specifying a JSON schema for vendors. You can specify **Vendor** as the data type name.  
You will see this data type name when you are creating or editing an integration that uses the connection for which you are specifying this type. Fuse Online displays the type name in the integration visualization panel and in the data mapper.

14. In the **Data Type Description** field, provide information that helps you distinguish this type. This description appears in the data mapper when you hover over the step that processes this type.
15. Click **Done**.

## Result

The connection appears in the integration flow in the location where you added it.

## 6.3. UPLOAD FILES TO AN FTP OR SFTP SERVER

To finish an integration by uploading files to an FTP or SFTP server, add an FTP or SFTP connection as the integration's finish connection. You can also upload files to an FTP or SFTP server in the middle of an integration. To do this, add an FTP or SFTP connection as a middle connection.

### Prerequisite

You created an FTP or SFTP connection.

### Procedure

1. Start creating the integration.
2. Add and configure the start connection.
3. On the **Choose a Finish Connection** page, do one of the following:
  - To finish an integration by uploading files, click the FTP or SFTP connection that you want to use.
  - To upload files in the middle of an integration, click the connection you want to use to finish the integration. Configure that connection. When the finish connection is part of the integration, in the left panel, hover over the plus sign where you want to add an FTP or SFTP connection and click **Add a connection**. Click the FTP or SFTP connection that you want to use to upload files in the middle of an integration.
4. On the **Choose an Action** page, click **Upload**.
5. In the **File name expression** field, if you want to upload only one particular file, then enter an [Apache Camel Simple language](#) expression that resolves to a file name. This is the name of the file that the action uploads to the server. You cannot specify a regular expression. To upload more than one file, leave this field blank.
6. In the **FTP directory** field, enter the absolute or relative name of a server directory. If the **File name expression** field contains an expression, then the connection stores the specified file in this directory. If the **File name expression** field is blank, then the connection uploads to this directory all files that were received from the previous integration step.
7. In the **If file exists** field, indicate the behavior when you are uploading a file that has the same path and name as a file that is on the server. Accept the default, **Override**, to overwrite the file that is on the server with the file that you are uploading. Or, select one of the following:
  - **Append** adds the content in the file being uploaded to the file that is on the server.
  - **Fail** throws `GenericFileOperationException`. The integration does not enter an error state.

- **Ignore** does not upload the file. The integration continues running under the assumption that everything is okay.
  - **Move** renames one of the files.
  - **TryRename** uploads the file with a temporary name and renames the file to the desired name. This operation does not check for the existence of a file with the desired name, which makes the operation faster on most servers than when existence checks are done.
8. In the **Temporary file prefix while copying** field, specify a string. The connection prepends this string to the name of a file while it is being uploaded. This enables the connection to write to a temporary file on the server and then rename that temporary file to have the correct name. This is useful for reducing locks when uploading very large files.
  9. In the **Temporary file name while copying** field, specify a string. The connection renames a file being uploaded to have this name while it is being uploaded. This enables the connection to write to a temporary file on the server and then rename that temporary file to have the correct name. This is useful for reducing locks when uploading very large files.
  10. Click **Next** to specify the action's input type.
  11. In the **Select Type** field, if the data type does not need to be known, accept **Type specification not required** and then, at the bottom, click **Done**. You do not need to follow the rest of these instructions.  
Otherwise, select one of the following as the schema type:
    - **JSON schema** is a document that describes the structure of JSON data. The document's media type is **application/schema+json**.
    - **JSON instance** is a document that contains JSON data. The document's media type is **application/json**.
    - **XML schema** is a document that describes the structure of XML data. The document's file extension is **.xsd**.
    - **XML instance** is a document that contains XML data. The document's file extension is **.xml**.
  12. In the **Definition** input box, paste a definition that conforms to the schema type you selected. For example, if you select **JSON schema** then you would paste the content of a JSON schema file, which has a media type of **application/schema+json**.
  13. In the **Data Type Name** field, enter a name that you choose for the data type. For example, suppose you are specifying a JSON schema for vendors. You can specify **Vendor** as the data type name.  
You will see this data type name when you are creating or editing an integration that uses the connection for which you are specifying this type. Fuse Online displays the type name in the integration visualization panel and in the data mapper.
  14. In the **Data Type Description** field, provide information that helps you distinguish this type. This description appears in the data mapper when you hover over the step that processes this type.
  15. Click **Done**.

## Result

The connection appears in the integration flow in the location where you added it.

## CHAPTER 7. CONNECT TO GMAIL

To trigger execution of an integration when a particular Gmail account receives an email, add a Gmail connection to an integration as its start connection. In an integration, to send an email from a particular Gmail account, add a Gmail connection as the integration's finish connection, or as a middle connection.

The general steps for connecting to Gmail in an integration are:

1. Register Fuse Online as a Google client application.
2. Create a Gmail connection. When you do this you choose the Gmail account that the connection is authorized to access.
3. If your integration sends an email from a Gmail account, decide how to populate an email to send.
4. Add a Gmail connection to an integration.
5. For a Gmail connection that sends an email, optionally map integration data to the email fields.

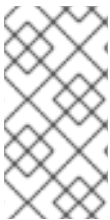
Information and instructions are in the following topics:

- [Section 7.1, “Register Fuse Online as a Gmail client application”](#)
- [Section 7.2, “Enable access to the Gmail API”](#)
- [Section 7.3, “Create a Gmail connection”](#)
- [Section 7.4, “Alternatives for populating email to send”](#)
- [Section 7.5, “Trigger an integration when polling returns a Gmail message”](#)
- [Section 7.6, “Send an email from a Gmail account”](#)

### 7.1. REGISTER FUSE ONLINE AS A GMAIL CLIENT APPLICATION

In an integration, to be able to connect to Gmail, you must register your Fuse Online environment as a Google client application. During registration, you enable the Gmail API and create credentials that Fuse Online uses to access Google APIs that you have enabled.

With registration in place, you can create any number of Gmail connections and any number of integrations that connect to Gmail. While each Gmail connection uses the same registration, each connection can access the mail for a different Google account.





#### NOTE



If you already created a Google Calendar connection, then you already created the credentials that your Fuse Online environment needs to access enabled Google APIs. You do not need to follow the procedure documented here. To be able to create a Gmail connection, you just need to [enable the Gmail API](#).

#### Prerequisite

You must be able to sign in to the Google account that you want to use to register Fuse Online as a Gmail client application.

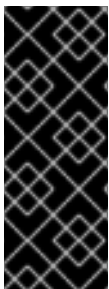
## Procedure

1. In Fuse Online, do the following:
  - a. In the left navigation panel, click **Settings**.
  - b. On the **Settings** page, near the top, to the right of the callback URL, click  to copy the callback URL for your Fuse Online environment to the clipboard. You will need this URL later in this procedure.
2. In another browser tab, go to <https://console.developers.google.com> and do the following:
  - a. Confirm that you are signed in to the Google account that you want to use to register Fuse Online as a Google client application. Or, choose a different Google account and sign in to that account.
  - b. Ensure that you want to use the current Google project to grant authorization to Fuse Online. Or, choose or create another project. If this Google account does not already have a project, you must create one. For information about Google projects and how you might want to organize your use of projects, click the Google help icon in the upper right.
  - c. Enable the Gmail API as follows:
    - i. You should see the **APIs and Services** dashboard. If you do not, then in the upper left corner, click the  and select **APIs and Services > Dashboard**.
    - ii. Click the **Gmail API** card.
    - iii. In the **Gmail API** page, click **ENABLE**.
  - d. Give your client application a name as follows:
    - i. In the page that appears, in the left navigation panel, click **Credentials** and then on the right, click **Credentials in APIs & Services**.
    - ii. Click the **OAuth consent screen** tab.
    - iii. In the page that appears, in the **Application Name** field, enter a name for the Fuse Online client application. For example, enter **Fuse Online client application**.
    - iv. Skip the other fields.
    - v. Click **Save**.
  - e. Obtain client application credentials as follows:
    - i. To the right of **Create Credentials**, click the down arrow to display a menu and select **OAuth client ID**.
    - ii. In the page that appears, select **Web application** to display more content.
    - iii. In the **Name** field, enter a name for the OAuth client ID for your Fuse Online environment. This is different from the name that you entered for the client application itself. For example, enter **OAuth client ID for Fuse Online**.
    - iv. Skip **Authorized JavaScript origins**.

- v. In the **Authorized redirect URIs** field, paste the callback URL that you copied from your Fuse Online environment at the beginning of this procedure.
  - vi. Click **Create** to display the client ID and client secret for your Fuse Online environment.
  - f. To the right of the client ID field, click  to copy the client ID to your clipboard.
3. Return to the Fuse Online **Settings** page, expand the Gmail entry, and in the Gmail **Client ID** field, paste the consumer client ID that you just copied.
  4. Return to the Google developers site and to the right of the client secret field, click  to copy the client secret to your clipboard.
  5. Return to Fuse Online and do the following:
    - a. In the **Settings** page, in the Gmail **Client Secret** field, paste the Gmail client secret that you just copied.
    - b. Click **Save**. You should get a **Registration Successful!** notification.
    - c. Click **Ok** to collapse the fields.

## Results

You can now create a Gmail connection.



### IMPORTANT

The Google client ID and Google client secret contain token refresh information that ensures that your Gmail connections, and integrations that have Gmail connections continuously work correctly. Consequently, you should **not** obtain new credentials. If you do, then you would need to recreate each Gmail connection, replace the old Gmail connections with new Gmail connections, and re-publish each integration that uses a Gmail connection.

## 7.2. ENABLE ACCESS TO THE GMAIL API

To be able to connect to Gmail in an integration, you must register your Fuse Online environment as a Google client application. During registration, you enable the Gmail API and create credentials that Fuse Online uses to access Google APIs that you have enabled.

If you already created a connection to Google Calendar, then Fuse Online has the credentials for access to enabled Google APIs. To be able to create a Gmail connection, you just need to enable the Gmail API and then in the Fuse Online **Settings** page, copy the Google Calendar client ID and client secret to the Gmail settings for client ID and client secret.


If you already registered your Fuse Online environment as a Google client application and during registration you enabled the Gmail API, then you do not need to follow the procedure provided here. You are ready to create a Gmail connection.

### Prerequisite

In the **Settings** page in your Fuse Online environment, the Google Calendar entry displays **Edit** and **Remove**, which indicate that Fuse Online has the credentials to access enabled Google APIs. If **Register** appears in the Google Calendar entry and the Gmail entry, then you must [register your Fuse](#)

Online environment as a Google client application.

### Procedure

1. In a browser, go to <https://console.developers.google.com>.
2. Confirm that you are signed in to the Google account that you used to register your Fuse Online environment as a Google client. If you are not, sign in to that account.
3. Ensure that the current Google project is the one you used for registration. If it is not, change to that project.
4. You should see the **APIs and Services** dashboard. If you do not, then in the upper left, click  and select **APIs and Services > Dashboard**.
5. Click the **Gmail API** card.
6. In the **Gmail API** page, click **ENABLE**.
7. In the Fuse Online **Settings** page:
  - a. Expand the Google Calendar entry
  - b. Expand the Gmail entry.
  - c. Copy the Google Calendar client ID to the input field for the Gmail client ID.
  - d. Copy the Google Calendar client secret to the input field for the Gmail client secret.
  - e. Click **Save**, and then click **Ok**.

### Result

You can now create a Gmail connection.

## 7.3. CREATE A GMAIL CONNECTION

When you create a Gmail connection, you authorize the connection to access one particular Gmail account. After you create a Gmail connection, you can add it to multiple integrations.

### Prerequisite

You registered your Fuse Online environment as a Gmail client application.

### Procedure

1. In Fuse Online, in the left panel, click **Connections** to display any available connections.
2. In the upper right, click **Create Connection** to display the available connectors. A connector is a template that you use to create one or more connections.
3. Click the **Gmail** connector.
4. In the **Configure Connection** page, click **Connect Gmail**, which takes you to a **Sign in with Gmail** page.  
If **Connect Gmail** does not appear, then your Fuse Online environment is not registered as a

Gmail client application. See [Register with Gmail](#). When you try to create a Gmail connection and your Fuse Online environment is not registered as a Gmail client application, then Fuse Online displays multiple fields that prompt for authorization information. While you can create a Gmail connection by entering values in these fields, you should register with Gmail first, and then create a Gmail connection after registration is in place.

5. In the page that is prompting you to sign in to Gmail, enter the email address of the Google account that you want this connection to access from Fuse Online and click **Next**.
6. In response to **openshiftapps.com wants to access your Google Account**, click **Allow** to return to Fuse Online.
7. In the Fuse Online **Connection Name** field, enter your choice of a name that helps you distinguish this connection from any other connections. For example, enter **Gmail Connect 1**.
8. In the **Description** field, optionally enter any information that is helpful to know about this connection. For example, enter **Sample Gmail connection that uses jkim Gmail account credentials**.
9. In the upper right, click **Create** to see that the connection you created is now available. If you entered the example name, you would see that **Gmail Connect 1** appears as a connection that you can choose to add to an integration.

## 7.4. ALTERNATIVES FOR POPULATING EMAIL TO SEND

A Gmail connection that finishes an integration or that is in the middle of an integration sends an email from the Gmail account that the connection is authorized to access. There are several ways to populate the content of the email that the connection sends. Before you add a Gmail connection that sends an email, consider how you want to populate that email.

The alternatives for populating an email to send are as follows:

- Add a data mapper step just before the Gmail connection that sends an email. In this data mapper step, map data fields that are output from previous integration steps to Gmail connection **Send Email** action fields. The **Send Email** action fields are:
  - **Email to**
  - **Email subject**
  - **Email text**
  - **Email cc**
  - **Email bcc**

If you add a data mapper step then you can map one, some, or all **Send Email** action fields.

- When you add a Gmail connection to an integration, configure the action by specifying values in the **Send Email** action fields. You can specify values in one, some, or all fields.
- Use both a data mapper step and **Send Email** action configuration to populate the fields. In other words, you can configure the **Send Email** action by specifying one or more fields and also add a data mapper step that populates other fields.

A value that you specify directly in a **Send Email** action field has precedence over a value that is mapped to the **Send Email** action field. In other words, suppose you populate a **Send Email** field by specifying a value when you add the connection and configure the action and also by



mapping a value to the same field. The value that you specify in the action configuration always overrides the value that was mapped. For example, suppose you specify **people@redhat.com** in the **Email to** action field and you also map an email field from a previous step to the Gmail **Email to** field. The integration always uses **people@redhat.com** and only **people@redhat.com** as the email address.

When you add a Gmail connection that sends an email, no action configuration parameters are required. This is because you might choose to populate an email entirely by mapping integration data to the **Send Email** action fields. However, the presence of an email address in the **Email to** field, either by configuration specification or by mapping, is required. Without an email address to send the message to, Fuse Online generates a runtime error and the integration stops executing.

## 7.5. TRIGGER AN INTEGRATION WHEN POLLING RETURNS A GMAIL MESSAGE

To trigger execution of an integration based on email received by a particular Gmail account, add a Gmail connection as the start connection of an integration. When the integration is running, the Gmail connection checks this account for emails at intervals that you control. When the connection finds an unread email, it passes the email to the next step in the integration and, by default, marks the email as **read**.

### Prerequisite

You created a Gmail connection that is authorized to access the Gmail account that you want to obtain emails from.

### Procedure

1. In the Fuse Online panel on the left, click **Integrations**.
2. Click **Create Integration**.
3. On the **Choose a Start Connection** page, click the Gmail connection that you want to use to start the integration.
4. On the **Choose an Action** page, click the **Receive Email** action.
5. To configure the **Receive Email** action:
  - a. In the **Delay** field, accept the default of 30 seconds or specify how often you want the integration to check for new email.
  - b. In the **Labels** field, leave it blank to obtain any unread email. To obtain only certain emails, specify a comma separated list of labels that are used in the Gmail account that the connection is accessing. By default, the integration obtains the unread emails that have those labels.
  - c. Select **Mark as read** to ensure that the connection does not return the same email twice. If **Mark as read** is not selected, the connection returns emails that are in the account's in box, whether or not they were previously read. Or, if you specify one or more labels, then the integration return emails that have those labels whether or not they were already read.
  - d. In the **Max Results** field, accept the default of **5** or indicate the maximum number of emails that the connection can return for each poll. If the connection finds more than this number of unread emails, then it returns the most recent **Max Results** emails.

When a Gmail connection returns more than one email, the integration processes the emails as a batch. In other words, Fuse Online executes the integration once for the batch.

- e. Click **Done** to add this Gmail connection as the start connection in the integration. The connection appears as the first step in the integration flow.

## 7.6. SEND AN EMAIL FROM A GMAIL ACCOUNT

In an integration, you can send an email from a Gmail account either in the middle of the integration or to finish the integration. To do this, add a Gmail connection to the middle of an integration or as the integration's finish connection.

### Prerequisites

- You created a Gmail connection.
- You are familiar with the [alternatives for populating an email to send](#) and you have a plan for populating such emails.

### Procedure

1. Start creating the integration.
2. Add and configure the start connection.
3. On the **Choose a Finish Connection** page, do one of the following:
  - To finish an integration by sending an email from a Gmail account, click the Gmail connection that is authorized to access the Gmail account that you want to send the email from.
  - To send an email in the middle of an integration:
    - a. Click the connection that you want to use to finish the integration.
    - b. Configure that connection.
    - c. When the finish connection is part of the integration, in the left panel, hover over the plus sign where you want to add a Gmail connection and click **Add a connection**.
    - d. Click the Gmail connection that you want to use to send an email in the middle of an integration.
4. On the **Choose an Action** page, click **Send Email**.
5. On the **Configure Send Email** page, do one of the following:
  - Leave all fields blank if you plan to add a data mapper step just before this connection and you plan to map integration data to the **Send Email** fields.
  - Enter information in one or more fields:
    - a. In the **Email To** field, enter one email address or a comma separated list of email addresses to send the email to.
    - b. In the **Email Subject** field, enter text that states the subject of the email.

- c. In the **Email Text** field, enter the message that you want to send.
  - d. In the **Email cc** field, enter a comma separated list of email addresses to which you want to send copies of the email.
  - e. In the **Email bcc** field, enter a comma separated list of email addresses to which you want to send blind copies of the email.
- Leave some fields blank and enter information in some fields.  
If you plan to add a data mapper step just before this Gmail connection, and you want to map integration data to some email fields, leave those fields blank in the action configuration. Enter values in other action fields as needed.

A value that you specify in a **Send Email** action configuration field has precedence over a value that is mapped from a previous step.

6. Click **Done** to add the connection to the integration.

### **Result and next steps**

The connection appears in the integration flow in the location where you added it. If you plan to add a data mapper step before this connection, first add any other connections that you plan to add to this integration. Then add the data mapper step.

## CHAPTER 8. CONNECT TO GOOGLE CALENDAR

You can add a Google Calendar connection to an integration as its start connection to trigger execution of an integration.. You can add a Google Calendar connection to an integration as its finish connection when you want to add or update an event. In the middle of an integration, you can add a Google Calendar connection when you want to obtain a particular event, add a new event, or update an event.

Details for connecting to Google Calendar are in the following topics:

- [Section 8.1, “Register Fuse Online as a Google Calendar client application”](#)
- [Section 8.2, “Enable access to the Google Calendar API”](#)
- [Section 8.3, “Create a Google Calendar connection”](#)
- [Section 8.4, “Trigger an integration when polling returns an event from a Google calendar”](#)
- [Section 8.5, “Obtain a particular event from a Google calendar”](#)
- [Section 8.7, “Update an event in a Google calendar”](#)
- [Section 8.6, “Add an event to a Google calendar”](#)

### 8.1. REGISTER FUSE ONLINE AS A GOOGLE CALENDAR CLIENT APPLICATION

In an integration, to be able to connect to Google Calendar, you must register your Fuse Online environment as a Google client application. During registration, you enable the Google Calendar API and create credentials that Fuse Online uses to access Google APIs that you have enabled.


With registration in place, you can create any number of Google Calendar connections and any number of integrations that connect to Google Calendar. While each Google Calendar connection uses the same registration, each connection can access the calendars that belong to a different Google account.






#### NOTE

If you already created a Gmail connection, then you already created the credentials that your Fuse Online environment needs to access enabled Google APIs. You do not need to follow the procedure documented here. To be able to create a Google Calendar connection, you just need to [enable the Google Calendar API](#) and then copy Fuse Online settings from Gmail to Google Calendar.

#### Procedure

1. In Fuse Online, obtain the callback URL as follows:
  - a. In the left navigation panel, click **Settings**.
  - b. On the **Settings** page, near the top, to the right of the callback URL, click  to copy the callback URL for your Fuse Online environment to the clipboard. You will need this URL later in this procedure.
2. In another browser tab, go to <https://console.developers.google.com> and do the following:

- a. Confirm that you are signed in to the Google account that you want to use to register Fuse Online as a Google client application. Or, choose a different Google account and sign in to that account.
- b. Ensure that you want to use the current Google project to grant authorization to Fuse Online. Or, choose or create another project. If this Google account does not already have a project, you must create one. For information about Google projects and how you might want to organize your use of projects, click the Google help icon in the upper right.
- c. You should see the **APIs and Services** dashboard. If you do not, then in the upper left corner, click the  and select **APIs and Services > Dashboard**.
- d. Enable the Google Calendar API as follows:
  - i. At the top of the page, click **ENABLE APIS AND SERVICES**.
  - ii. In the page that appears, in the search field, enter **Google Calendar**.
  - iii. In the search results, click the **Google Calendar** card.
  - iv. In the **Google Calendar API** page, click **ENABLE**.
- e. Give your client application a name as follows:
  - i. In the page that appears, in the left navigation panel, click **Credentials** and then on the right, click **Credentials in APIs & Services**.
  - ii. Click the **OAuth consent screen** tab.
  - iii. In the page that appears, in the **Application Name** field, enter a name for the Fuse Online client application. For example, enter **Fuse Online client application**.
  - iv. Skip the other fields.
  - v. Click **Save**.
- f. Obtain client application credentials as follows:
  - i. To the right of **Create Credentials**, click the down arrow to display a menu and select **OAuth client ID**.
  - ii. In the page that appears, select **Web application** to display more content.
  - iii. In the **Name** field, enter a name for the OAuth client ID for your Fuse Online environment. This is different from the name that you entered for the client application itself. For example, enter **OAuth client ID for Fuse Online**.
  - iv. Skip **Authorized JavaScript origins**.
  - v. In the **Authorized redirect URIs** field, paste the callback URL that you copied from your Fuse Online environment at the beginning of this procedure.
  - vi. Click **Create** to display the client ID and client secret for your Fuse Online environment.
- g. To the right of the client ID field, click  to copy the client ID to your clipboard.

3. Return to the Fuse Online **Settings** page, expand the Google Calendar entry, and in the Google Calendar **Client ID** field, paste the consumer client ID that you just copied.
4. Return to the Google developers site and to the right of the client secret field, click  to copy the client secret to your clipboard.
5. Return to the Fuse Online **Settings** page and do the following:
  - a. In the Google Calendar **Client Secret** field, paste the Google Calendar client secret that you just copied.
  - b. Click **Save**. You should get a **Registration Successful!** notification.
  - c. Click **Ok** to collapse the fields.

## Results

You can now create a Google Calendar connection.



### IMPORTANT

The Google client ID and Google client secret contain token refresh information that ensures that your Google Calendar connections and integrations that have Google Calendar connections continuously work correctly. Consequently, you should **not** obtain new credentials. If you do, then you would need to recreate each Google Calendar connection and re-publish each integration that uses a Google Calendar connection.

## 8.2. ENABLE ACCESS TO THE GOOGLE CALENDAR API

To be able to connect to Google Calendar in an integration, you must register your Fuse Online environment as a Google client application. During registration, you enable the Google Calendar API and create credentials that Fuse Online uses to access Google APIs that you have enabled.

If you already created a connection to Gmail, then Fuse Online has the credentials for access to enabled Google APIs. To be able to create a Google Calendar connection, you just need to enable the Google Calendar API and then copy Fuse Online settings from Gmail to Google Calendar.


If you already registered your Fuse Online environment as a Google client application and during registration you enabled the Google Calendar API, then you do not need to follow the procedure provided here. You are ready to create a Google Calendar connection.

### Prerequisite

In the **Settings** page in your Fuse Online environment, the Gmail entry displays **Edit** and **Remove**, which indicate that Fuse Online has the credentials to access enabled Google APIs. If **Register** appears in the Gmail entry, then you must [register your Fuse Online environment as a Google client application](#).

### Procedure

1. In a browser, go to <https://console.developers.google.com> and do the following:
  - a. Confirm that you are signed in to the Google account that you used to register your Fuse Online environment as a Google client. If you are not, sign in to that account.
  - b. Ensure that the current Google project is the one you used for registration. If it is not, change to that project.

- c. You should see the **APIs and Services** dashboard. If you do not, then in the upper left, click  and select **APIs and Services > Dashboard**.
  - d. At the top of the page, click **ENABLE APIS AND SERVICES**.
  - e. In the page that appears, in the search field, enter **Google Calendar**.
  - f. In the search results, click the **Google Calendar** card.
  - g. In the **Google Calendar API** page, click **ENABLE**.
2. In the Fuse Online **Settings** page, do the following:
    - a. Expand the Gmail entry.
    - b. Expand the Google Calendar entry.
    - c. Copy the Gmail client ID to the input field for the Google Calendar client ID.
    - d. Copy the Gmail client secret to the input field for the Google Calendar client secret.
    - e. Click **Save**, and then click **Ok**.

## Result

You can now create a Google Calendar connection.

## 8.3. CREATE A GOOGLE CALENDAR CONNECTION

When you create a Google Calendar connection, you authorize the connection to access the Google Calendars that are associated with one particular Google account. After you create a Google Calendar connection, you can add it to multiple integrations.

### Prerequisite

You registered your Fuse Online environment as a Google client application as described in [Register as a Google Calendar client application](#).

### Procedure

1. In Fuse Online, in the left panel, click **Connections** to display any available connections.
2. In the upper right, click **Create Connection** to display the available connectors. A connector is a template that you use to create one or more connections.
3. Click the **Google Calendar** connector.
4. In the **Configure Connection** page, click **Connect Google Calendar**, which takes you to a Google sign-in page.



## NOTE

If **Connect Google Calendar** does not appear, then your Fuse Online environment is not registered as a Google client application with the Google Calendar API enabled. See [Register as a Google Calendar client application](#). When your environment is not registered with Google, then when you try to create a Google Calendar connection, Fuse Online displays multiple fields that prompt for authorization information. While you can create a Google Calendar connection by entering values in these fields, you should register with Google first, enable the Google Calendar API, and then create a Google Calendar connection.

5. In the Google sign-in page, select the Google account that you want this connection to access from Fuse Online and click **Next**.
6. In response to the **openshiftapps.com wants to access your Google Account** prompt, click **Allow** to return to Fuse Online.
7. In the Fuse Online **Connection Name** field, enter your choice of a name that helps you distinguish this connection from other connections. For example, enter **Google Calendar Work Connection**.
8. In the **Description** field, optionally enter any information that is helpful to know about this connection. For example, enter **Sample Google Calendar connection that uses my Google work account**.
9. In the upper right, click **Create** to see that the connection you created is now available. If you entered the example name, you would see that **Google Calendar Work Connection** appears as a connection that you can choose to add to an integration.

## 8.4. TRIGGER AN INTEGRATION WHEN POLLING RETURNS AN EVENT FROM A GOOGLE CALENDAR

To trigger execution of an integration upon obtaining events from a Google Calendar that you specify, add a Google Calendar connection to an integration as its start connection. When the integration is running, the Google Calendar connection checks the Google calendar for events at intervals that you control. When the connection finds events that comply with the way that you configured the Google Calendar **Get Events** action, the connection passes the events to the next step in the integration.

When a Google Calendar connection returns more than one event, Fuse Online executes the integration for each returned event. For example, if the poll returns 5 events then Fuse Online executes the integration five times.

### Prerequisite

You created a Google Calendar connection.

### Procedure

1. In the Fuse Online panel on the left, click **Integrations**.
2. Click **Create Integration**.
3. On the **Choose a Start Connection** page, click the Google Calendar connection that you want to use to start the integration.



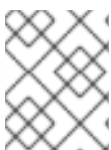
4. On the **Choose an Action** page, click the **Get Events** action.
5. To configure the **Get Events** action:
  - a. In the **Delay** field, accept the default of 30 seconds or specify how often you want the integration to check the calendar.
  - b. In the **Calendar name** field, enter the name of a calendar that is accessible from the Google account that this Google Calendar connection is authorized to access.
  - c. In the **Max Results** field, accept the default of **5** or indicate the maximum number of events that the connection can return for each poll.
  - d. Select one of the following to indicate the date that you want the connection to use to start checking for events. The connection checks that date and subsequent dates.
    - **Consume from the current date ahead** starts on the current date.  
For example, if you select this option, and if **Max Results** is 5, then the Google Calendar connection starts checking for events on the current date and returns no more than the first five events that it finds. It does not matter whether or not these events have been updated or added since the last poll.
    - **Consume from the last event update date on the next poll** starts on the date of the most recently updated event that was returned in the previous poll.  
For example, if you select this option, and if **Max Results** is 5, then the Google Calendar connection returns no more than 5 updated or new events. The connection starts checking for new or updated events on the date of the most recently updated event that was returned in the previous poll.
  - e. Optionally, in the **Query for events** field, specify text to filter the events that the poll can return. The connection returns only events that contain the specified text in at least one event field.  
For example, suppose that you specify **Standup meeting** in the query field. A poll would return only those events that have **Standup meeting** in an event field.
6. Click **Done** to add this Google Calendar connection as the integration's start connection. The connection appears as the first step in the integration.

## 8.5. OBTAIN A PARTICULAR EVENT FROM A GOOGLE CALENDAR

In an integration, you can obtain a particular Google Calendar event in the middle of an integration. Obtaining a particular event is useful, for example, when you want to:

- Update the event in a subsequent Google Calendar connection.
- Announce the event by using a subsequent Twitter connection.

To obtain one event, add a Google Calendar connection in the middle of an integration.




### NOTE

In this release, while obtaining a specific event in an integration's finish connection is supported, it is not particularly useful. This is expected to change in a future release.

### Prerequisites

- A Google Calendar connection is available and this connection is authorized to access the Google calendar that has the event that you want to get.
- You are creating or editing an integration and Fuse Online is prompting you to select the connection that you want to add in the middle of an integration.

### Procedure

1. Click a Google Calendar connection that is authorized to access the calendar that you want to connect to.
2. On the **Choose an Action** page, click **Get a Specific Event**.
3. Specify the name of the calendar or add a finish connection or ender that has the event that you want.
4. Specify the ID of the event to get. To obtain the event ID, do one of the following:
  - Map it from a previous Google Calendar connection.
  - Manually obtain it from the calendar as follows:
    - a. In a browser, display the calendar that contains the event you want to get.
    - b. Append `?gsessionid=OK&eventdeb=1`` to the URL and redisplay the calendar.
    - c. In the calendar, click the event that you want to get.
    - d. In the event popup, click  and select **Troubleshooting info**.
    - e. In the popup that Google Calendar displays, copy the string that follows `eid=`. For example, an event ID looks something like this:  
`p1pva2a4t504gbsha12di9ch6k_20181107T150000Z*`.
5. Click **Done** to add the connection to the integration. The connection appears in the integration flow in the location where you added it.

## 8.6. ADD AN EVENT TO A GOOGLE CALENDAR

In an integration, you can add an event to a Google calendar in the middle of the integration or to finish the integration. To do this, add a Google Calendar connection to the middle of an integration or as the integration's finish connection.

### Prerequisites

- A Google Calendar connection is available and this connection is authorized to access the Google calendar to which you want to add an event.
- You are creating or editing an integration and Fuse Online is prompting you to add a finish connection or select the connection that you want to add in the middle of an integration.

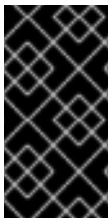
### Procedure

1. Click a Google Calendar connection that is authorized to access the calendar that you want to connect to.

2. On the **Choose an Action** page, click the **Create Event** action.
3. Configure the **Create Event** action:
  - a. Enter a title for the new event.
  - b. Enter the name of the Google Calendar to add the event to.
  - c. For the other input parameters, you can enter data or you might want to map values from an event that you obtained in a previous Google Calendar connection.
4. Click **Done** to add the connection to the integration. The connection appears in the integration flow in the location where you added it.

## 8.7. UPDATE AN EVENT IN A GOOGLE CALENDAR

In an integration, you can update an event in a Google calendar in the middle of the integration or to finish the integration. To do this, add a Google Calendar connection to the middle of an integration or as the integration's finish connection.



### IMPORTANT

In this release, the **Update Event** action requires a value in each event field. In most if not all cases, this means that you must add a Google Calendar connection that [obtains the event that you want to update](#), then add the Google Calendar connection that updates the event, and then insert a data mapper step between the two Google Calendar connections.

### Prerequisites

- A Google Calendar connection is available and this connection is authorized to access the Google calendar that has the event that you want to update.
- In the integration, there is an earlier connection to Google Calendar and that connection obtains the event that you want to update.
- You are creating or editing an integration and Fuse Online is prompting you to add a finish connection or select the connection that you want to add in the middle of an integration.

### Procedure

1. Click a Google Calendar connection that is authorized to access the calendar that you want to connect to.
2. On the **Choose an Action** page, click **Update Event**.
3. To configure the **Update Event** action:
  - a. Enter the title of the event that you want to update.
  - b. Enter content in each event field that you want to update. Do not enter content in an event field when you want the content in that field to remain the same as it was.
4. Click **Done** to add the connection to the integration. The connection appears in the integration visualization flow in the location where you added it.

5. In the integration visualization flow, hover over the plus sign that is just before the connection that you just added and click **Add a step**.
6. Click **Data Mapper**.
7. In the data mapper, for each event field that remains the same in the updated event, map that field from the Google Calendar connection that obtained the event to the corresponding field in the Google Calendar connection that updates the event.  
Do not map fields that you are updating. If you map a field and also specify an update for that field when you configure the **Update Event** action, Fuse Online uses the mapped value.
8. In the upper right, click **Done** to add the data mapper step.

## CHAPTER 9. CONNECT TO HTTP AND HTTPS ENDPOINTS

In an integration, you can connect to HTTP and HTTPS endpoints to execute the **GET**, **PUT**, **POST**, **DELETE**, **HEAD**, **OPTIONS**, **TRACE**, or **PATCH** method. To do this, create an HTTP or HTTPS connection and then add it to an integration. The following topics provide details:

- [Section 9.1, “Create a connection to an HTTP or HTTPS endpoint”](#)
- [Section 9.2, “Add an HTTP or HTTPS connection to an integration”](#)

### 9.1. CREATE A CONNECTION TO AN HTTP OR HTTPS ENDPOINT

In an integration, to execute the HTTP **GET**, **PUT**, **POST**, **DELETE**, **HEAD**, **OPTIONS**, **TRACE**, or **PATCH** method, create a connection to an HTTP or HTTPS endpoint. You can then add the connection to one or more integrations.

#### Prerequisite

You must know the URL for the endpoint that you want to connect to.

#### Procedures

1. In Fuse Online, in the left panel, click **Connections** to display any available connections.
2. In the upper right, click **Create Connection** to display Fuse Online connectors.
3. If you want to use Hyper Text Transfer Protocol to connect to the endpoint, then click the **HTTP** connector. If you want to use Secure Hyper Text Transfer protocol, then click the **HTTPS** connector.
4. In the **Base URL** field, enter the endpoint path. For example, `www.mycompany.com/sales`.
5. Click **Validate**. Fuse Online immediately tries to validate the connection and displays a message that indicates whether validation is successful. If validation fails, revise specification of the base URL and try again.
6. If validation is successful, click **Next**.
7. In the **Connection Name** field, enter your choice of a name that helps you distinguish this connection from any other connections. For example, enter **HTTPS My Company Sales**.
8. In the **Description** field, optionally enter any information that is helpful to know about this connection.
9. In the upper right, click **Create** to see that the connection you created is now available. If you entered the example name, you would see that **HTTPS My Company Sales** appears as a connection that you can choose to add to an integration.

### 9.2. ADD AN HTTP OR HTTPS CONNECTION TO AN INTEGRATION

You can add an HTTP or HTTPS connection to any number of integrations.

#### Prerequisite

You created an HTTP or HTTPS connection.

## Procedure

1. Do one of the following:

- To start an integration by periodically invoking an HTTP or HTTPS endpoint, add an HTTP or HTTPS connection as the integration's start connection.
- To finish an integration by invoking an HTTP or HTTPS endpoint once, add an HTTP or HTTPS connection as the integration's finish connection.
- In the middle of an integration, to invoke an HTTP or HTTPS endpoint once, add an HTTP or HTTPS connection after the start connection and before the finish connection.

If you are creating an integration, Fuse Online prompts you to choose and configure the start connection, and then choose and configure the finish connection. To add a middle connection, hover over the plus sign in the left panel at the location where you want to add the connection, and select **Add a connection**.

2. Click the HTTP or HTTPS connection that you want to add to the integration.

3. Select the action that you want the connection to perform:

- If you are adding a start connection, then **Periodic invoke URL** is the only available action. This action invokes the endpoint at intervals that you specify and triggers the integration if the endpoint returns any data.
- If you are adding a finish or middle connection, then **Invoke URL** is the only available action. This action invokes the endpoint once.

4. In the **URL Path** field, specify the location of the endpoint that you want to invoke.

5. In the **HTTP Method** field, select the method that you want the connection to perform. The default method is **GET**.

- **GET** obtains the content at the URL path.
- **PUT** replaces the content at the URL path with the integration data.
- **POST** stores the integration data at the URL path to create new content.
- **DELETE** removes content at the URL path.
- **HEAD** obtains metadata about the content at the URL path.
- **OPTIONS** obtains communication option settings at the URL path.
- **TRACE** obtains information for testing and diagnostic purposes.
- **PATCH** partially updates the content at the URL path according to the integration data.

6. If you are adding a start connection, which periodically invokes the URL, then in the **Period** field, accept the default interval of **1** second or specify a number and its unit (milliseconds, seconds, minutes, or hours) to indicate how long to wait between invocations.

7. Click **Done** to specify the action's input or output type.

8. In the **Select Type** field, if the data type does not need to be known, accept **Type specification not required** and then, at the bottom, click **Done**. You do not need to follow the rest of these instructions.  
Otherwise, select one of the following as the schema type:
  - **JSON schema** is a document that describes the structure of JSON data. The document's media type is **application/schema+json**.
  - **JSON instance** is a document that contains JSON data. The document's media type is **application/json**.
  - **XML schema** is a document that describes the structure of XML data. The document's file extension is **.xsd**.
  - **XML instance** is a document that contains XML data. The document's file extension is **.xml**.
9. In the **Definition** input box, paste a definition that conforms to the schema type you selected. For example, if you select **JSON schema** then you would paste the content of a JSON schema file, which has a media type of **application/schema+json**.
10. In the **Data Type Name** field, enter a name that you choose for the data type. For example, suppose you are specifying a JSON schema for vendors. You can specify **Vendor** as the data type name.  
You will see this data type name when you are creating or editing an integration that uses the connection for which you are specifying this type. Fuse Online displays the type name in the integration visualization panel and in the data mapper.
11. In the **Data Type Description** field, provide information that helps you distinguish this type. This description appears in the data mapper when you hover over the step that processes this type.
12. Click **Done**.

## Result

The connection appears in the integration flow in the location where you added it.

## CHAPTER 10. CONNECT TO KAFKA

Apache Kafka is a distributed streaming platform that you can use to obtain and publish data. In an integration, you can subscribe for data from a Kafka topic that you specify or publish data to a Kafka topic that you specify. To do this, create a connection to Kafka and then add that connection to an integration. Details are in the following topics:

- [Section 10.1, “Create a connection to a Kafka broker”](#)
- [Section 10.2, “Obtain data from a Kafka broker”](#)
- [Section 10.3, “Publish data to a Kafka broker”](#)

### 10.1. CREATE A CONNECTION TO A KAFKA BROKER

In an integration, to subscribe for data from a Kafka topic or publish data to a Kafka topic, create a connection to Kafka and then add that connection to an integration.

#### Prerequisite

You must know the URI for the Kafka broker that you want to connect to.

#### Procedure

1. In Fuse Online, in the left panel, click **Connections** to display any available connections.
2. In the upper right, click **Create Connection** to display connectors.
3. Click the **Kafka** connector.
4. In the **Kafka bootstraps URI** field, enter a comma separated list of Kafka broker URIs. Each URI should be in the form *host:port*.
5. Click **Validate**. Fuse Online immediately tries to validate the connection and displays a message that indicates whether validation is successful. If validation fails, revise the input parameter and try again.
6. If validation is successful, click **Next**.
7. In the **Connection Name** field, enter your choice of a name that helps you distinguish this connection from any other connections. For example, you might enter **Kafka West**.
8. In the **Description** field, optionally enter any information that is helpful to know about this connection.
9. In the upper right, click **Create** to see that the connection you created is now available. If you entered the example name, you would see that **Kafka West** appears as a connection that you can choose to add to an integration.

### 10.2. OBTAIN DATA FROM A KAFKA BROKER

To trigger execution of an integration based on receiving data from a Kafka broker, add a Kafka connection as the start connection. When the integration is running, the Kafka connection continuously watches for data in the Kafka topic that you specify. When the connection finds new data, it passes that data to the next step in the integration.



## Prerequisite

You created a connection to a Kafka broker.

## Procedure

1. In the Fuse Online panel on the left, click **Integrations**.
2. Click **Create Integration**.
3. On the **Choose a Start Connection** page, click the Kafka connection that you want to use to start the integration.
4. On the **Choose an Action** page, click the **Subscribe** action to receive data from the topic that you specify.
5. In the **Topic Name** field, click the down carat to display a list of topics and click the topic that you want to subscribe to.
6. Click **Next** to specify the action's output type.
7. In the **Select Type** field, if the data type does not need to be known, accept **Type specification not required** and then, at the bottom, click **Done**. You do not need to follow the rest of these instructions.  
Otherwise, select one of the following as the schema type:
  - **JSON schema** is a document that describes the structure of JSON data. The document's media type is **application/schema+json**.
  - **JSON instance** is a document that contains JSON data. The document's media type is **application/json**.
  - **XML schema** is a document that describes the structure of XML data. The document's file extension is **.xsd**.
  - **XML instance** is a document that contains XML data. The document's file extension is **.xml**.
8. In the **Definition** input box, paste a definition that conforms to the schema type you selected. For example, if you select **JSON schema** then you would paste the content of a JSON schema file, which has a media type of **application/schema+json**.
9. In the **Data Type Name** field, enter a name that you choose for the data type. For example, suppose you are specifying a JSON schema for vendors. You can specify **Vendor** as the data type name.  
You will see this data type name when you are creating or editing an integration that uses the connection for which you are specifying this type. Fuse Online displays the type name in the integration visualization panel and in the data mapper.
10. In the **Data Type Description** field, provide information that helps you distinguish this type. This description appears in the data mapper when you hover over the step that processes this type.
11. Click **Done**.

## Result

The connection appears in the integration flow in the location where you added it.

## 10.3. PUBLISH DATA TO A KAFKA BROKER

In an integration, you can publish data to a Kafka broker to finish an integration. To do this, add a Kafka connection as the integration's finish connection. To publish data to a Kafka broker in the middle of integration, add a Kafka connection to an integration after the start connection and before the finish connection.

### Prerequisites

- You created a connection to a Kafka broker.
- You are creating or editing an integration and the integration already has a start connection.

### Procedure

1. On the **Choose a Finish Connection** page, do one of the following:
  - To finish an integration by publishing data, click the Kafka connection that you want to use.
  - To publish data in the middle of an integration, click the connection that you want to use to finish the integration. Configure that connection. When the finish connection is part of the integration, in the left panel, hover over the plus sign where you want to add a Kafka connection and click **Add a connection**. Click the Kafka connection that you want to use to publish a message in the middle of an integration.
2. On the **Choose an Action** page, click **Publish**.
3. In the **Topic Name** field, click the down carat to display a list of topics and click the topic that you want to publish to.
4. Click **Next** to specify the action's input type.
5. In the **Select Type** field, if the data type does not need to be known, accept **Type specification not required** and then, at the bottom, click **Done**. You do not need to follow the rest of these instructions.  
Otherwise, select one of the following as the schema type:
  - **JSON schema** is a document that describes the structure of JSON data. The document's media type is **application/schema+json**.
  - **JSON instance** is a document that contains JSON data. The document's media type is **application/json**.
  - **XML schema** is a document that describes the structure of XML data. The document's file extension is **.xsd**.
  - **XML instance** is a document that contains XML data. The document's file extension is **.xml**.
6. In the **Definition** input box, paste a definition that conforms to the schema type you selected. For example, if you select **JSON schema** then you would paste the content of a JSON schema file, which has a media type of **application/schema+json**.
7. In the **Data Type Name** field, enter a name that you choose for the data type. For example, suppose you are specifying a JSON schema for vendors. You can specify **Vendor** as the data type name.

You will see this data type name when you are creating or editing an integration that uses the connection for which you are specifying this type. Fuse Online displays the type name in the integration visualization panel and in the data mapper.

8. In the **Data Type Description** field, provide information that helps you distinguish this type. This description appears in the data mapper when you hover over the step that processes this type.
9. Click **Done**.

## Result

The connection appears in the integration flow in the location where you added it.

## CHAPTER 11. CONNECT TO AN INTEGRATION'S LOG

For each integration step, Fuse Online provides the following activity information:

- The date and time that the step was executed
- How long it took to execute the step
- Whether execution was successful
- The error message if execution was not successful

To view this information in Fuse Online, display the integration's summary and then click the **Activity** tab. See also: [Viewing integration activity information](#).

To obtain further details about integration execution, you can log information about the messages that an integration processes by adding a log step and/or a log connection to an integration.

- [Section 11.1, "Comparison of log steps and log connections"](#)
- [Section 11.2, "Add a log connection to an integration"](#)
- [Section 11.3, "Create a replacement log connection"](#)



### 11.1. COMPARISON OF LOG STEPS AND LOG CONNECTIONS

To log information about the messages that an integration processes, you can add one or more log steps to an integration and/or you can add one or more log connections to an integration.

A log connection cannot be an integration's start connection. Typically, a log connection is most useful as an integration's finish connection.

The following table indicates whether a log step or a log connection can do what you want.

To do this	Add a log step	Add a log connection
Log the ID of each message. (Internally, this ID is the exchange ID.)		✓
Log the headers in each message.	✓	✓
Log the body of each message.	✓	✓
Log text that you specify for each message.	✓	✓
Specify a Camel Simple expression that Fuse Online resolves to text for each message and then sends to the log.	✓	

To do this	Add a log step	Add a log connection
Finish an integration.		
Log messages at the level you specify ( <b>ERROR</b> , <b>WARN</b> , <b>INFO</b> , <b>DEBUG</b> , <b>TRACE</b> , <b>OFF</b> ).		

### Additional resources

- [Add a log step to an integration](#) in Integrating Applications with Fuse Online.
- [Add a log connection to an integration](#)

## 11.2. ADD A LOG CONNECTION TO AN INTEGRATION

To log information about the messages that an integration processes, add one or more log connections to the integration. A log connection takes each input message from the previous step in the integration, sends the details that you specify to the integration's log, and passes the message to the next step in the integration.

Fuse Online provides a log connection that you can add to any number of integrations, and that you can add any number of times to the same integration. You do not need to create a log connection unless you inadvertently delete the provided log connection.

### Prerequisites

- You have chosen to add a log connection rather than a log step to an integration. For help with this, see [Comparison of log step and log connection](#).
- You must be creating or editing an integration. If you are creating an integration then the integration must already have its start connection.
- Fuse Online is prompting you to choose a new integration's finish connection or is prompting you to choose the connection that you want to add to an integration.

### Procedure

1. Click the **Log** connection.  
Fuse Online comes with the **Log** connection; you do not need to have created a log connection yourself. However, if you inadvertently deleted the provided connection and created a new log connection, then you might have given it another name.
2. On the **Choose an Action** page, click **Simple Logger**.
3. Accept the **INFO** log level, or click in the log level input box to select a different level for sending information to the log.  
Each message that an integration handles has a log level field whose value is **ERROR**, **WARN**, **INFO**, **DEBUG**, or **TRACE**. The table below indicates the behavior for each log level.

Select this level	To log information for messages with this log level
<b>ERROR</b>	<b>ERROR</b>
<b>WARN</b>	<b>ERROR, WARN</b>
<b>INFO</b>	<b>ERROR, WARN, INFO</b>
<b>DEBUG</b>	<b>ERROR, WARN, INFO, DEBUG</b>
<b>TRACE</b>	<b>ERROR, WARN, INFO, DEBUG, TRACE</b>
<b>OFF</b>	No logging. After creating and testing an integration, you might no longer need the logging provided by the log connection. Edit the integration, edit its log connection to select <b>OFF</b> , and re-publish the integration.

4. Select one or more checkboxes to indicate what information to send to the log for each message. Choices include:

- **Message ID**
- **Message headers**
- **Message body**

If you do not select any checkboxes, then for each logged message, Fuse Online logs several internal IDs, how long it took to process the message, whether there was an error, and an error message if there was one. This is comparable to the information that Fuse Online provides for each integration step, which you can view in the **Activity** tab of the integration summary.

5. Click **Done** to add the log connection to the integration.

#### Additional resource

- [View integration activity](#)

## 11.3. CREATE A REPLACEMENT LOG CONNECTION

Fuse Online provides a log connection that you can add to an integration to log information about each message that an integration processes. If you inadvertently delete the provided log connection, then you can create a new log connection from the **Log** connector that Fuse Online provides.

#### Procedure

1. In Fuse Online, in the left navigation panel, click **Connections**.
2. In the upper right, click **Create Connection**.
3. Click the **Log** connector.
4. In the **Connection Name** field, enter a name for the connection, for example, enter **Log**.

5. Optionally, enter a description for this connection if you need to distinguish it.
6. In the upper right, click **Create**. This displays the available connections and you can see your new log connection.

## CHAPTER 12. CONNECT TO MQTT

MQ Telemetry Transport (MQTT) is a lightweight, machine-to-machine, internet of things, connectivity protocol. In an integration, you can obtain messages from or publish messages to an MQTT broker. To do this, create a connection to the MQTT broker of interest and then add that connection to an integration. Details are in the following topics:

- [Section 12.1, “Create a connection to an MQTT broker”](#)
- [Section 12.2, “Obtain a message from an MQTT broker”](#)
- [Section 12.3, “Publish a message to an MQTT broker”](#)

### 12.1. CREATE A CONNECTION TO AN MQTT BROKER

In an integration, to obtain messages from or publish messages to an MQTT broker, create a connection to the MQTT broker of interest and then add that connection to an integration.

#### Prerequisite

You must know the URL for the MQTT broker that you want to connect to.

#### Procedure

1. In Fuse Online, in the left panel, click **Connections** to display any available connections.
2. In the upper right, click **Create Connection** to display connectors.
3. Click the **MQTT Message Broker** connector.
4. To configure the connection:
  - a. In the **MQTT broker URL** field, enter the location of the MQTT broker that you want to send data to or obtain data from. This is the only required field.
  - b. In the **User Name** field, optionally enter the user name for the MQTT account whose credentials you want to use to access the broker.
  - c. In the **Password** field, if you specified a user name, then specify the password associated with that account.
  - d. In the **Client ID** field, optionally enter the ID that allows connections to close and reopen without missing messages. The connection must subscribe to or publish to a topic.
5. Click **Validate**. Fuse Online immediately tries to validate the connection and displays a message that indicates whether validation is successful. If validation fails, revise the input parameters and try again.
6. If validation is successful, click **Next**.
7. In the **Connection Name** field, enter your choice of a name that helps you distinguish this connection from any other connections. For example, you might enter **MQTT West**.
8. In the **Description** field, optionally enter any information that is helpful to know about this connection.



- In the upper right, click **Create** to see that the connection you created is now available. If you entered the example name, you would see that **MQTT West** appears as a connection that you can choose to add to an integration.

## 12.2. OBTAIN A MESSAGE FROM AN MQTT BROKER

To trigger execution of an integration based on receiving a message from an MQTT broker, add an MQTT connection as the start connection. When the integration is running, the MQTT connection continuously watches for messages on the MQTT queue or topic that you specify. When the connection finds a message, it passes it to the next step in the integration. An MQTT connection handles one message at a time.

### Prerequisite

You created an MQTT connection.

### Procedure

- In the Fuse Online panel on the left, click **Integrations**.
- Click **Create Integration**.
- On the **Choose a Start Connection** page, click the MQTT connection that you want to use to start the integration.
- On the **Choose an Action** page, click the **Subscribe** action to receive messages from the queue or topic that you specify.
- In the **MQTT queue/topic name** field, enter the name of the queue or topic to subscribe to in order to receive data.
- Click **Next** to specify the action's output type.
- In the **Select Type** field, if the data type does not need to be known, accept **Type specification not required** and then, at the bottom, click **Done**. You do not need to follow the rest of these instructions.  
Otherwise, select one of the following as the schema type:
  - **JSON schema** is a document that describes the structure of JSON data. The document's media type is **application/schema+json**.
  - **JSON instance** is a document that contains JSON data. The document's media type is **application/json**.
  - **XML schema** is a document that describes the structure of XML data. The document's file extension is **.xsd**.
  - **XML instance** is a document that contains XML data. The document's file extension is **.xml**.
- In the **Definition** input box, paste a definition that conforms to the schema type you selected. For example, if you select **JSON schema** then you would paste the content of a JSON schema file, which has a media type of **application/schema+json**.

9. In the **Data Type Name** field, enter a name that you choose for the data type. For example, suppose you are specifying a JSON schema for vendors. You can specify **Vendor** as the data type name.  
You will see this data type name when you are creating or editing an integration that uses the connection for which you are specifying this type. Fuse Online displays the type name in the integration visualization panel and in the data mapper.
10. In the **Data Type Description** field, provide information that helps you distinguish this type. This description appears in the data mapper when you hover over the step that processes this type.
11. Click **Done**.

## Result

The connection appears in the integration flow in the location where you added it.

## 12.3. PUBLISH A MESSAGE TO AN MQTT BROKER

In an integration, you can publish a message to an MQTT broker to finish an integration. To do this, add an MQTT connection as the integration's finish connection. To publish a message to an MQTT broker in the middle of integration, add an MQTT connection to an integration after the start connection and before the finish connection.

### Prerequisites

- You created an MQTT connection.
- You are creating or editing an integration and the integration already has a start connection.

### Procedure

1. On the **Choose a Finish Connection** page, do one of the following:
  - To finish an integration by publishing a message, click the MQTT connection that you want to use.
  - To publish a message in the middle of an integration, click the connection that you want to use to finish the integration. Configure that connection. When the finish connection is part of the integration, in the left panel, hover over the plus sign where you want to add an MQTT connection and click **Add a connection**. Click the MQTT connection that you want to use to publish a message in the middle of an integration.
2. On the **Choose an Action** page, click **Publish**.
3. In the **MQTT queue/topic name** field, specify the name of the queue or topic to publish the message to.
4. Click **Next** to specify the action's input type.
5. In the **Select Type** field, if the data type does not need to be known, accept **Type specification not required** and then, at the bottom, click **Done**. You do not need to follow the rest of these instructions.  
Otherwise, select one of the following as the schema type:
  - **JSON schema** is a document that describes the structure of JSON data. The document's media type is **application/schema+json**.

- **JSON instance** is a document that contains JSON data. The document's media type is **application/json**.
  - **XML schema** is a document that describes the structure of XML data. The document's file extension is **.xsd**.
  - **XML instance** is a document that contains XML data. The document's file extension is **.xml**.
6. In the **Definition** input box, paste a definition that conforms to the schema type you selected. For example, if you select **JSON schema** then you would paste the content of a JSON schema file, which has a media type of **application/schema+json**.
  7. In the **Data Type Name** field, enter a name that you choose for the data type. For example, suppose you are specifying a JSON schema for vendors. You can specify **Vendor** as the data type name.  
You will see this data type name when you are creating or editing an integration that uses the connection for which you are specifying this type. Fuse Online displays the type name in the integration visualization panel and in the data mapper.
  8. In the **Data Type Description** field, provide information that helps you distinguish this type. This description appears in the data mapper when you hover over the step that processes this type.
  9. Click **Done**.

## Result

The connection appears in the integration flow in the location where you added it.

## CHAPTER 13. CONNECT TO REST APIS

In an integration, to connect to a REST API, you must have created a connector for that API by uploading an OpenAPI schema that describes the API. See [Integrating Applications with Fuse Online, Adding REST API client connectors](#).

When a connector for the REST API you want to connect to is available in Fuse Online, the steps for connecting to that REST API are:

- [Section 13.1, “Register Fuse Online as an API client”](#) if required
- [Section 13.2, “Create a An API client connection”](#)
- [Section 13.3, “Add an API client connection to an integration”](#)

### 13.1. REGISTER FUSE ONLINE AS AN API CLIENT


Before Fuse Online creates an API client connector, it prompts you to indicate the API’s security requirements. For APIs that use the OAuth protocol, when Fuse Online creates the connector it also adds an entry for the API to the Fuse Online **Settings** page. This is where you provide the API client ID and the API client secret that authorize Fuse Online to access the API.

If the API you want to connect to does not use OAuth, skip this section and see [Create a REST API connection](#).

#### Prerequisite

You must know the URL for the OAuth custom application setting page for the REST API that you want to connect to.

#### Procedure

1. In Fuse Online:
  - a. On the **Settings** page, near the top, to the right of the callback URL, click  to copy the callback URL for your Fuse Online environment to the clipboard. You will need this URL later in this procedure.
  - b. Look for the name of the API you want to connect to and click its **Register** button to display its client ID and client secret fields.
2. In another browser window, you must register Fuse Online as an OAuth client of the API you want to connect to. The exact steps for doing this vary for each API service. Typically, the API service provides an OAuth custom application setting page. Go to that page and do the following:
  - a. Provide the Fuse Online callback URL, which you copied at the beginning of this procedure.
  - b. Respond to any other prompts that require your input.
  - c. Obtain the client ID and client secret that the API service assigns to your Fuse Online environment.
3. Return to the Fuse Online **Settings** page entry for the API service you are registering with and do the following:

- a. Paste the assigned client ID.
- b. Paste the assigned client secret.
- c. Click **Save**.

## 13.2. CREATE A AN API CLIENT CONNECTION

In an integration, to connect to a REST API, create a connection to that REST API, which you can then add to any number of integrations.

### Prerequisites

- You created a connector for the REST API that you want to connect to.
- If the REST API uses the OAuth protocol then you registered you Fuse Online environment as a client application that can access that REST API.

### Procedure

1. In Fuse Online, in the left panel, click **Connections** to display any available connections.
2. In the upper right, click **Create Connection** to display available connectors.
3. Click the connector for the API that you want to create a connection for.
4. Respond to prompts for additional information. The definition of the API determines what Fuse Online prompts for. For example, for an API that uses HTTP Basic Authorization, Fuse Online prompts for the user name and password to use to access the API. For an API that uses OAuth, Fuse Online displays a button for you to click so that Fuse Online can verify its registration credentials for connecting to the API.
5. In the **Connection Name** field, enter your choice of a name that helps you distinguish this connection from any other connections.
6. In the **Description** field, optionally enter any information that is helpful to know about this connection.
7. In the upper right, click **Create** to see that you can now choose the connection you created and add to an integration.

## 13.3. ADD AN API CLIENT CONNECTION TO AN INTEGRATION

In an integration, to connect to a REST API, add a connection to that REST API to the integration.

In this release, in an integration, a connection to a REST API can be the finish connection or a middle connection. It cannot be the start connection.

### Prerequisites

- You created a connection to the REST API.
- You are creating or editing an integration.
- The integration already has its start connection.

- Fuse Online is prompting you to select a finish connection or a connection that is not the start connection.

### Procedure

1. On the page that displays available connections, click the REST API connection that you want to add to the integration.
2. Click the action that you want the connection to perform. The actions that are available are based on the resource operations specified in the OpenAPI file that was uploaded to Fuse Online and that describes the API you are connecting to.
3. Depending on the action you select, enter any parameters that Fuse Online prompts for.
4. Click **Done**.

### Result

The connection appears in the integration flow in the location where you added it.

## CHAPTER 14. CONNECT TO SALESFORCE

In an integration, to connect to Salesforce, you must register your Fuse Online environment as a Salesforce client application. This authorizes Fuse Online to access Salesforce. With registration in place, you create a Salesforce connection, which you can then add to any number of integrations. For details, see the following topics:

- [Section 14.1, “Register Fuse Online as a Salesforce client application”](#)
- [Section 14.2, “Create a Salesforce connection”](#)
- [Section 14.3, “Add a Salesforce connection to an integration”](#)

### 14.1. REGISTER FUSE ONLINE AS A SALESFORCE CLIENT APPLICATION


In an integration, to connect to Salesforce, the first thing you must do is register your Fuse Online environment as a client application that can access Salesforce. This lets you create any number of integrations that connect to Salesforce. In other words, you need to register a particular Fuse Online environment with Salesforce only once.

In each Fuse Online environment, there can be only one registration of Fuse Online as a Salesforce client application. However, while each Salesforce connection uses the same registration, it can use different user credentials.


#### Prerequisite

You are logged in to Fuse Online. The URL for accessing Fuse Online is in the email message that welcomed you to the Red Hat Fuse Online Evaluation program.

#### Procedure

1. In Fuse Online:
  - a. In the left panel, click **Settings**.
  - b. On the **Settings** page, near the top, to the right of the callback URL, click  to copy the callback URL for your Fuse Online environment to the clipboard. You will need this URL toward the end of this procedure.
  - c. To the right of the **Salesforce** entry, click **Register** to display the **Client ID** and **Client Secret** fields.
2. In another browser tab, log in to your Salesforce account and follow the steps below to create a connected app. These instructions assume that you are using the Salesforce Classic user interface. To switch from the Salesforce Lightning Experience interface, click your profile icon and select **Switch to Salesforce Classic**. For additional information, see the Salesforce documentation for [Create a Connected App](#).
  - a. In Salesforce, in the upper right, click **Setup**.
  - b. In the left panel, select **Build > Create > Apps**.
  - c. Scroll down to **Connected Apps** and click **New**.
  - d. Enter the required information and then select **Enable OAuth Settings**.

- e. In the **Callback URL** field, paste your Fuse Online URL, which you copied at the beginning of this procedure. For example: **https://app-proj9128.7b63.fuse-ignite.openshiftapps.com/api/v1/credentials/callback**.
  - f. For **OAuth Scopes**, add:
    - **Access and manage your data (api)**
    - **Allow access to your unique identifier (openid)**
    - **Perform requests on your behalf at any time (refresh\_token, offline\_access)**
  - g. Select **Configure ID token** and then **Include Standard Claims**.
  - h. Scroll down and click **Save**.
  - i. Scroll up to see that Salesforce indicates a short wait:
 

**New Connected App** [Help for this Page](#) 

Allow from 2-10 minutes for your changes to take effect on the server before using the connected app.

Continue
Cancel
  - j. Click **Continue**.
  - k. Copy the consumer key that Salesforce provides.
3. Return to your Fuse Online **Settings** page and paste the Salesforce-provided consumer key into the Fuse Online Salesforce **Client ID** field.
  4. Return to Salesforce and copy the consumer secret that Salesforce provides.
  5. Return to your Fuse Online **Settings** page and paste the Salesforce-provided consumer secret into the Fuse Online Salesforce **Client Secret** field.
  6. Click **Save** and then click **Ok**.

## 14.2. CREATE A SALESFORCE CONNECTION

To create an integration that accesses data in Salesforce, you must first create a Salesforce connection. After you create a Salesforce connection, you can use it in multiple integrations.

### Prerequisites

- You are logged in to Fuse Online. The URL for accessing Fuse Online is in the email message that welcomed you to the Red Hat Fuse Online Evaluation program.
- Fuse Online is open in a web browser.
- You must have already registered your Fuse Online environment as an application that can access Salesforce.  
Be sure to wait 2 - 10 minutes after registering your Fuse Online installation as a Salesforce client before you try to create a Salesforce connection.

### Procedure



1. In Fuse Online, in the left panel, click **Connections** to display available connections.
2. In the upper right, click **Create Connection** to display the available connectors. A connector is a template for creating one or more connections.
3. Click the **Salesforce** connector.
4. Click **Connect Salesforce** to display a Salesforce authorization page. You might need to log in to Salesforce before you see the authorization page.



#### NOTE

The following error indicates that Salesforce does not have the correct Fuse Online callback URL:

```
error=redirect_uri_mismatch&error_description=redirect_uri%20must%20match%20configuration
```

If you get this error message, then in Salesforce, ensure that the Fuse Online callback URL is specified according to the instructions in [Register with Salesforce](#).

5. Click **Allow** to return to Fuse Online.
6. In the **Connection Name** field, enter your choice of a name that helps you distinguish this connection from any other connections. For example, enter **SF Connect 1**.
7. In the **Description** field, optionally enter any information that is helpful to know about this connection. For example, enter **Sample Salesforce connection that uses my Salesforce login credentials**.
8. In the upper right, click **Create** to see that the connection you created is now available. If you entered the example name, you would see that **SF Connect 1** appears as a connection that you can choose to add to an integration.

## 14.3. ADD A SALESFORCE CONNECTION TO AN INTEGRATION

In an integration, you can connect to Salesforce to start or finish the integration, or in the middle of an integration. To connect to Salesforce, add a Salesforce connection to the integration.

### Prerequisites

- You created a Salesforce connection.
- You are creating or updating an integration. If you need to, see the [general procedure for creating an integration](#) or see [updating integrations](#).
- Fuse Online is prompting you to select a start connection, a finish connection or a middle connection.

### Procedure

1. On the page that displays available connections, click the Salesforce connection that you want to add to the integration. When the integration uses the connection you select to connect to Salesforce, Fuse Online uses the credentials defined in that connection.

2. Click the action that you want the selected connection to perform. Each Salesforce connection that you add to an integration performs only the action you choose.
3. Specify the Salesforce object that the action operates on, for example, it might be a contact, lead or price book entry. Click in the **Object** field to select from a list of Salesforce objects or enter the name of the object.
4. Click **Done** to add the connection to the integration.

## Result

The connection appears in the integration flow in the location where you added it.

## CHAPTER 15. CONNECT TO SAP CONCUR

SAP Concur manages business expenses, travel, and invoices. In an integration, you can connect to SAP Concur and perform any one of a large variety of SAP Concur actions. For example, suppose that you store project codes in a SQL database and you need the same project codes in SAP Concur so that expenses can be associated with projects. You can create an integration that connects to your SQL database and runs a stored procedure that obtains any new project codes. The integration can then connect to SAP Concur and upload the new project codes to the appropriate list.

See the following topics:

- [Section 15.1, “User roles for connecting to SAP Concur”](#)
- [Section 15.2, “How to Obtain SAP Concur implementation site OAuth credentials”](#)
- [Section 15.3, “How to obtain SAP Concur production site OAuth credentials”](#)
- [Section 15.4, “Configure the SAP Concur connector”](#)
- [Section 15.5, “Create a SAP Concur connection”](#)
- [Section 15.6, “Add a SAP Concur connection to an integration”](#)
- [Section 15.7, “Example of mapping integration data fields to SAP Concur fields”](#)

### 15.1. USER ROLES FOR CONNECTING TO SAP CONCUR

SAP Concur has an implementation site for development work and a production site for live use. The implementation site simulates the production site. This lets you create and refine an integration before you use it in production with the live SAP Concur site.

At any one time, a Fuse Online environment can connect to the SAP Concur implementation site or the SAP Concur production site. In other words, a particular Fuse Online environment does not support connections to both SAP Concur sites at the same time. Each Fuse Online environment requires its own credentials for accessing each SAP Concur site and the credentials for the implementation site are different from the credentials for the production site.

Consequently, the expectation is that someone in a system administrator-like role would do the following:

1. Obtain two sets of OAuth credentials from SAP Concur:
  - One set for access to the SAP Concur implementation site
  - Another set for access to the SAP Concur production site
2. Configure the SAP Connector for access to appropriate SAP Concur site.
3. Create a SAP Concur connection.

A business user can then create an integration that uses the SAP Concur connection.

### 15.2. HOW TO OBTAIN SAP CONCUR IMPLEMENTATION SITE OAUTH CREDENTIALS

To obtain credentials for accessing the SAP Concur implementation site, contact SAP Concur directly and tell them:

- You want to register Fuse Online as a new app that is authorized to access the SAP Concur implementation site.
- You want a new set of OAuth credentials for their implementation site.
- The credentials must include the **LIST** scope.
- Optionally, to obtain permission to perform actions other than the list actions, ask for other [scopes](#).
- The OAuth grant type for these credentials should be **Authorization Grant Flow**. To use a REST client, such as Postman, to send a request to the SAP Concur implementation site, you need credentials with an OAuth grant type of **Password**.
- The Fuse Online callback URL for the environment that that you want credentials for, which is something like this:  
**`https://syndesis.my-minishift.syndesis.io/api/v1/credentials/callback`**

In this URL, **`syndesis.my-minishift.syndesis.io`** matches the OpenShift route to your Fuse Online environment.

The response from SAP Concur should provide:

- A username and password for logging in to the SAP Concur implementation site.
- Client ID and client secret values that you will specify in the Fuse Online development environment **Settings** page to configure the Fuse Online SAP Concur connector.

### 15.3. HOW TO OBTAIN SAP CONCUR PRODUCTION SITE OAUTH CREDENTIALS

To obtain OAuth credentials that authorize Fuse Online to access the SAP Concur production site, go to the [SAP Concur Developer Center Getting Started](#) page and follow the instructions there.

### 15.4. CONFIGURE THE SAP CONCUR CONNECTOR

To connect to SAP Concur in an integration, you must configure the Fuse Online SAP Concur connector. You can then use the connector to create a connection to SAP Concur.

#### Prerequisites

- You know whether you want to configure the SAP Concur connector to access the SAP Concur implementation site or the SAP Concur production site.
- You obtained OAuth credentials that authorize your Fuse Online environment to access the appropriate SAP Concur site.

#### Procedure

1. In Fuse Online, in the left panel, click **Settings**.
2. On the **Settings** page, expand the **SAP Concur** entry.
3. In the **Client ID** field, enter the client ID that you received from SAP Concur.

4. In the **Client Secret** field, enter the client secret that you received from SAP Concur. Fuse Online populates the other fields.
5. Click **Save**.  
Fuse Online immediately tries to validate the configuration. If validation is not successful, correct the input values and try again. If validation is successful, you can create a SAP Concur connection, which you can add to an integration.

### Next steps

To create a SAP Concur connection, click the **create a connection** link in the success message. Or, to create a connection later, click **Ok** to contract the SAP Concur entry in the **Settings** page.

## 15.5. CREATE A SAP CONCUR CONNECTION

To connect to SAP Concur in an integration, you must first create a SAP Concur connection, which you can then add to any number of integrations. When you create a SAP Concur connection, you authorize the connection to access SAP Concur with the login credentials that you enter when you create the connection.

### Prerequisites

- In the Fuse Online environment in which you are creating the connection, you must have configured the SAP Concur connector.
- You should know whether the connector is configured for access to the SAP Concur implementation site or the SAP Concur production site.
- You must have a user name and password for logging in to the SAP Concur site for which the SAP Concur connector is configured.

### Procedure

1. In Fuse Online, in the left panel, click **Connections** to display any available connections.
2. In the upper right, click **Create Connection** to display the available connectors. A connector is a template that you use to create one or more connections.
3. Click the **SAP Concur** connector.
4. In the **Configure Connection** page, click **Connect SAP Concur**, which takes you to the SAP Concur site for which the connector is configured.  
If **Connect SAP Concur** does not appear, then the SAP Concur connector in your Fuse Online environment is not configured. See [Configure SAP Concur connector](#).

If **redirect\_uri is improper or not previously registered** appears, then the SAP Concur connector configuration is incorrect. Return to the Fuse Online **Settings** page to update and validate the connector configuration.

5. On the SAP Concur OAuth site:
  - a. Enter the email address for the SAP Concur account that you want this connection to use to access SAP Concur.
  - b. Ensure that **Username** is selected.

- c. Click **Continue**.
  - d. Enter the SAP Concur password associated with the email address.
  - e. Click **Sign In**, which returns you to Fuse Online.
6. In the Fuse Online **Connection Name** field, enter your choice of a name that helps you distinguish this connection from any other connections. For example, enter **SAP Concur Test1**.
  7. In the **Description** field, optionally enter any information that is helpful to know about this connection. For example, enter **Sample SAP Concur connection to the implementation site**.
  8. In the upper right, click **Create** to see that the connection you created is now available. If you entered the example name, you would see that **SAP Concur Test1** appears as a connection that you can choose to add to an integration.

## 15.6. ADD A SAP CONCUR CONNECTION TO AN INTEGRATION

In an integration, a connection to SAP Concur is a middle or finish connection and not a start connection. A connection to SAP Concur can perform any one of 85 actions.

### Prerequisites

- You created a SAP Concur connection.
- You are creating an integration or updating an integration to add a connection to that integration. If you need to, see the [general procedure for creating an integration](#) or see [updating integrations](#).

### Procedure

The instructions below assume that Fuse Online is prompting you to select a finish connection or a middle connection.

1. On the page that displays available connections, click the SAP Concur connection that you want to add to the integration. When the integration uses the connection that you select to connect to SAP Concur, Fuse Online uses the credentials defined in that connection.
2. Click the action that you want the selected connection to perform. Each SAP Concur connection that you add to an integration performs only the action you choose. For details about SAP Concur actions, visit the [SAP Concur developer center](#) and expand **v3.0**.
3. Click **Done** to add the connection to the integration.

### Result

The connection appears in the integration flow in the location where you added it.

## 15.7. EXAMPLE OF MAPPING INTEGRATION DATA FIELDS TO SAP CONCUR FIELDS

In most integrations, you need to add a data mapping step before a connection to SAP Concur. In the data mapping step, you map data fields that are available from previous integration steps to SAP Concur action parameters. In the data mapper, the **Target** panel displays the SAP Concur parameters for the

action that you selected when you added the SAP Concur connection to the integration.

Suppose that an integration starts by executing a SQL stored procedure that obtains new project codes. The integration finishes by adding new project codes to SAP Concur. The following procedure provides an example of a data mapper step before the SAP Concur connection.

1. In Fuse Online, in the left panel, hover over the plus sign that is between the database connection and the SAP Concur connection to display a pop-up in which you click **Add a Step**.
2. Click **Data Mapper**. When the data fields appear, the **Sources** panel on the left displays the fields that are available from the database connection. In this example, the source fields include **concur-list-id** and **project-code**. The **Target** panel on the right displays the SAP Concur **id** and **content** parameters for the update list action, which the connection performs.
3. Map the **concur-list-id** source field to the **id** target field.
4. Map the **project-code** source field to the **content** target field.
5. In the upper right, click **Done** to add the data mapper step to the integration.

## CHAPTER 16. CONNECT TO SERVICENOW

An integration can retrieve records from a ServiceNow table or add records to a ServiceNow import set. To do this, you create a ServiceNow connection and then add that connection to an integration. For details, see:

- [Section 16.1, “Create a ServiceNow connection”](#)
- [Section 16.2, “Start an integration by obtaining records from ServiceNow”](#)
- [Section 16.3, “Copy records to ServiceNow during or to finish an integration”](#)

### 16.1. CREATE A SERVICENOW CONNECTION

In an integration, to connect to your company’s ServiceNow instance, you must create a ServiceNow connection.

#### Prerequisites

- A ServiceNow administrator at your site must have created a ServiceNow account for you.
- You must know the URL for your ServiceNow instance as well your ServiceNow user name and password.
- If the ServiceNow administrator also created a ServiceNow client ID and client secret for you, then you must know these values.

#### Procedure

1. In Fuse Online, in the left panel, click **Connections** to display any available connections.
2. In the upper right, click **Create Connection** to display Fuse Online connectors.
3. Click the **ServiceNow** connector.
4. Configure this ServiceNow connection:
  - a. In the **Instance Name** field, enter the name of the ServiceNow instance that you want to obtain records from or copy records to. For example, if the URL for your ServiceNow instance is `https://abc12345.service-now.com` then the instance name is **abc12345**.
  - b. In the **User Name** field, enter your ServiceNow user name.
  - c. In the **User Password** field, enter your ServiceNow password.
  - d. Optionally, in the **Client ID** field, enter your client ID if you received one from your ServiceNow administrator.
  - e. Optionally, in the **Client Secret** field, enter your client secret if you received one from your ServiceNow administrator.
  - f. Click **Validate**. Fuse Online immediately tries to validate the connection and displays a message that indicates whether or not validation is successful. If validation fails, revise the configuration details as needed and try again.



- g. When validation is successful, click **Next**.
5. In the **Connection Name** field, enter your choice of a name that helps you distinguish this connection from any other connections. For example, enter **ServiceNow Con 1**.
6. In the **Description** field, optionally enter any information that is helpful to know about this connection. For example, enter **ServiceNow connection that uses administrative credentials**.
7. In the upper right, click **Create** to see that the connection you created is now available. If you entered the example name, you would see that **ServiceNow Con 1** appears as a connection that you can choose to add to an integration.

## 16.2. START AN INTEGRATION BY OBTAINING RECORDS FROM SERVICENOW

To start an integration by obtaining records from ServiceNow, add a ServiceNow connection to an integration as the start connection.

### Prerequisites

- You created a ServiceNow connection.
- You must know the name of the table that you want to retrieve records from.
- You should have already defined a ServiceNow query that obtains only the records that you want. You can find information about ServiceNow queries here: [ServiceNow encoded queries](#).

### Procedure

1. In the Fuse Online panel on the left, click **Integrations**.
2. Click **Create Integration**.
3. On the **Choose a Start Connection** page, click the ServiceNow connection that you want to use to start the integration.
4. On the **Choose an Action** page, click **Retrieve Record** to obtain records from a ServiceNow table that you select.
5. Configure the **Retrieve Record** action:
  - a. Click in **The table name** field and select the table that you want to obtain records from.
  - b. In **The query used to filter the result set** field, enter a ServiceNow query. For example, consider the **state=1^impact=2** query on a table that keeps records about incidents. This query returns records for new incidents (**state=1**) that have a medium impact (**impact=2**). If you do not enter a query and the records in the table do not change, then the connection obtains the same records every time.
  - c. In the **Limit of elements per page** field, enter the maximum number of records that you want the connection to obtain. In this release, you must enter a value, the recommendation is to enter **1000** or less, and pagination is not supported.

- d. In the **Period** field, indicate how often you want to obtain records. The default is every minute. In other words, by default, Fuse Online executes this integration every 60 seconds.
- e. Click **Done**.

## Results

- Fuse Online generates a JSON schema that defines the structure of the obtained records. This enables you to add a data mapping step before the next connection in the integration if data mapping is needed.
- Fuse Online prompts you to choose a finish connection.

## 16.3. COPY RECORDS TO SERVICENOW DURING OR TO FINISH AN INTEGRATION

In the middle of an integration, or to finish an integration, you can copy records to ServiceNow. To do this, add a ServiceNow connection to an integration as the finish connection or as a middle connection.

### Prerequisites


- You are creating or editing an integration.
- The integration has a start connection.
- You know the name of the ServiceNow import set that you want to add records to. Your ServiceNow administrator can help you identify the appropriate import set.
- The ServiceNow import set must have been configured to handle the addition of records.

### Procedure

1. Do one of the following:
  - To finish the integration by copying records to ServiceNow, on the **Choose a Finish Connection** page, click the ServiceNow connection that you want to use to finish the integration.
  - To copy records to ServiceNow in the middle of an integration, after the integration has a start connection and a finish connection, in the integration visualization panel on the left, hover over the plus sign that is in the location where you want to copy records to ServiceNow. Click **Add a connection**, and click the ServiceNow connection that you want to add to the integration.
2. Click the **Create Record** action.
3. Click in **The Import Set name** field and select the import set that you want to add records to.
4. Click **Next**.

### Results

- Fuse Online adds the connection to the integration.

- If a data mapper step is needed before this connection, then in the integration visualization panel, Fuse Online displays  on the connection.

## CHAPTER 17. CONNECT TO SLACK

As a business user, you can create an integration that connects to Slack. A connection to Slack can do either one of the following:

- Trigger execution of the integration when a Slack channel that you specify receives a message. The integration passes the message to the next step in the integration. For example, you can monitor a Slack channel for keyword instances such as product names. Upon finding messages that contain those product names, the integration can notify the appropriate contact in a Gmail connection.
- Deliver a message to a particular user or to a channel. For example, this behavior is useful when an integration downloads a file from an FTP server and processes it in some way. You can finish an integration by notifying a Slack channel or user that the process was successful.

To connect to Slack in an integration, create a Slack connection and then add the connection to an integration. Details are in the following topics:

- [Section 17.1, “Create a Slack connection”](#)
- [Section 17.2, “Add a Slack connection to trigger execution upon receiving messages”](#)
- [Section 17.3, “Add a Slack connection to send a message to a Slack channel or user”](#)

### 17.1. CREATE A SLACK CONNECTION

In an integration, a Slack connection can retrieve messages from a channel that you specify or send a message to a channel or user. You can use the same Slack connection in any number of integrations.

#### Prerequisites

- You created a Slack app for connecting Fuse Online to Slack.
- You can obtain the Slack webhook URL for your Slack app.
- You can obtain a legacy token that authorizes access to your Slack app.

#### Procedure

1. In Fuse Online, in the left panel, click **Connections** to display any available connections.
2. In the upper right, click **Create Connection** to display Fuse Online connectors.
3. Click the **Slack** connector.
4. In the **Slack webhook URL** field, enter the webhook URL for your Slack app.
5. In the **Token for accessing Slack API** field, enter the Slack-provided token, which you can [obtain from Slack](#).
6. Optionally, enter values for additional parameters:
  - In the **Sending username for messages** field, enter the user name that the bot has when it sends messages to Slack.

- In the **Message avatar emoji** field, specify one emoji, which the bot uses as the message avatar when it sends a message. Examples: `:smile:`, `:wave:`, `:question:`.

### TIP

Search the web for **emoji cheat sheet** to see some emoji codes.

- In the **Message avatar icon URL** field, specify the URL of the avatar that the bot uses when it sends messages to Slack.

If you specify an emoji and an icon URL, then the integration uses the icon URL. If you specify neither an emoji nor an icon URL, then the message is sent without an avatar.

7. Click **Validate**. Fuse Online immediately tries to validate the connection and displays a message that indicates whether validation is successful. If validation fails, revise the connection configuration values and try again.
8. If validation is successful, click **Next**.
9. In the **Connection Name** field, enter your choice of a name that helps you distinguish this connection from any other connections. For example, enter **Slack for Tracking Company Sales**.
10. In the **Description** field, optionally enter any information that is helpful to know about this connection.
11. In the upper right, click **Create** to see the updated list of available connections, including the connection that you just created. If you entered the example name, you would see that **Slack for Tracking Company Sales** appears as a connection that you can choose to add to an integration. You can add this connection to any number of integrations.

## 17.2. ADD A SLACK CONNECTION TO TRIGGER EXECUTION UPON RECEIVING MESSAGES

A Slack connection that starts an integration triggers execution of the integration when the connection finds messages on a Slack channel that you specify.

### Prerequisites

- You created a Slack connection.
- You are creating an integration and Fuse Online is prompting you to choose how you want to start the integration.

### Procedure

1. Click the Slack connection that you want to use to start the integration.
2. Click the **Read Messages** action.
3. In the **Channel** field, specify the Slack channel from which you want to obtain messages.
4. In the **Delay** field, accept the default of 500 milliseconds as the time that elapses between polls for messages. Or, to specify a different interval for checking for messages, enter a number and select its time unit.

5. In the **Maximum Messages to Retrieve** field, accept the default of 10 or indicate the maximum number of messages that the connection can return for each poll. If the connection finds more than this number of unread messages, then it returns the most recent messages. The connection returns a particular message only once.

For example, if you accept the default of 10 and there are 15 messages at the time of a poll then the connection returns the 10 most recent messages. At the next poll, the connection checks for messages that have a time stamp that is after the most recent message that the connection previously returned. This means that the 5 messages that were not returned in the first example poll are never returned.

When a Slack connection returns more than one message, the integration processes the messages as a batch. In other words, Fuse Online executes the integration once for the batch.

6. Click **Done** to add the connection to the integration.

## 17.3. ADD A SLACK CONNECTION TO SEND A MESSAGE TO A SLACK CHANNEL OR USER

In an integration, you can send a message to a Slack channel or Slack user to finish an integration or in the middle of an integration.

### Prerequisites

- You created a Slack connection.
- You are creating or editing an integration. If you are creating an integration, then Fuse Online might be prompting you to choose a finish connection. To add a middle connection, hover over the plus sign in the left panel in the location where you want to add the connection and select **Add a connection**.

### Procedure

1. Click the Slack connection that you want to add to the integration.
2. Select the action that you want the connection to perform.
  - Click **Username** to send a message to one user. To configure this action, in the **User name** field, specify the Slack name of the user to send the message to.
  - Click **Channel** to publish a message on a channel. To configure this action, in the **Channel** field, specify the channel to publish the message to.
3. Click **Done** to add the connection to the integration.
4. Optionally, add additional connections to the integration. Whether additional connections are needed depends on what you want the integration to do. The important point is to add all connections before you continue.
5. Add a data mapping step just before the Slack connection that you added in this procedure. In the mapping step, map a string from a data mapping source to the Slack **message** field. This string should contain the message that you want to send to the Slack user or channel. See [Add data mapping step](#).

## CHAPTER 18. CONNECT TO SQL DATABASES

In an integration, you can connect to any of the following types of SQL databases:

- Apache Derby
- MySQL
- PostgreSQL

You create a connection to the database that you want to access in an integration and then you add that connection to an integration.

To connect to other types of databases, you must upload a JDBC driver for that database.

See the following topics for details:

- [Section 18.1, “Create a database connection”](#)
- [Section 18.2, “Start an integration by accessing a database”](#)
- [Section 18.3, “Access a database in the middle or to complete an integration”](#)
- [Section 18.4, “Connect to proprietary databases”](#)

### 18.1. CREATE A DATABASE CONNECTION

You create a separate connection for each database that you want to connect to in an integration. You can use the same connection in multiple integrations.

A database connection operates on a database table that you specify or invokes a stored procedure that you specify.

#### Prerequisite

The database table or the stored procedure must exist when an integration connects to the database.

#### Procedure

1. Ensure that the JDBC driver for the database that you want to connect to is on your classpath. If you uploaded a JDBC driver library extension to connect to a proprietary database, then the upload process puts the driver on your classpath. See [Creating JDBC driver library extensions](#).
2. In Fuse Online, in the left panel, click **Connections** to display any available connections.
3. In the upper right, click **Create Connection** to display Fuse Online connectors.
4. Click the **Database** connector.
5. Configure the connection by entering:
  - a. In the **Connection URL** field, enter the JDBC URL for the database that you want to connect to. For example, enter `jdbc:postgresql:ignite-db1234/sampledb`.
  - b. In the **Username** field, enter the name of the account that you want to use to access the database. Ensure that the specified password and user name are for the same account.

- c. In the **Password** field, enter the password associated with the user account you want to use to access the database.
  - d. In the **Schema** field, enter the name of the schema for the database. If the connection URL specifies the schema, ensure that this field indicates the same schema as the connection URL. For example, enter **sampledb**.
6. Click **Validate**. Fuse Online tries to validate the connection and displays a message that indicates whether validation is successful. If validation fails, revise the configuration details as needed and try again.
  7. If validation is successful, click **Next**.
  8. In the **Connection Name** field, enter your choice of a name that helps you distinguish this connection from any other connections. For example, enter **PostgreSQL DB 1**.
  9. In the **Description** field, optionally enter any information that is helpful to know about this connection. For example, enter **Sample PostgreSQL connection that uses my login credentials**.
  10. In the upper right, click **Create** to see that the connection you created is now available. If you entered the example name, you would see that **PostgreSQL DB 1** appears as a connection that you can choose to add to an integration.

## 18.2. START AN INTEGRATION BY ACCESSING A DATABASE

To trigger execution of an integration based on the result of invoking a SQL query or a SQL stored procedures, choose a database connection as the integration's start connection.

### Prerequisite

You created a database connection.

### Procedure

1. In the Fuse Online panel on the left, click **Integrations**.
2. Click **Create Integration**.
3. On the **Choose a Start Connection** page, click the database connection that you want to use to start an integration.
4. On the **Choose an Action** page, click one of the following:
  - **Periodic SQL invocation** obtains data by periodically invoking the SQL query you specify.
  - **Periodic stored procedure invocation** obtains data by periodically invoking the stored procedure you specify or select.
5. If you selected **Periodic SQL invocation**, in the **Query** field, enter a SQL **SELECT** statement to obtain one or more records. For example: **SELECT \* from my\_db\_table**. The database table that contains the data you want must already exist.  
If you selected **Periodic stored procedure invocation**, in the **Procedure name** field, select or enter the stored procedure to invoke to obtain the data of interest. The stored procedure you specify must already exist. The database administrator should have created any stored procedures you need to use in an integration.



6. In the **Period** field, enter an integer and indicate whether the unit is milliseconds, seconds, minutes, hours, or days. For example, if you specify **5 minutes** then the connection invokes the specified query or stored procedure every five minutes.
7. Click **Done**.

## Result

Fuse Online tries to validate the connection, which includes checking that a specified SQL query is syntactically correct and confirming that the query or stored procedure target data exists. If verification is successful then Fuse Online adds the start connection to the integration. If verification fails then Fuse Online displays a message about the problem. Update your input as needed and try again.

## 18.3. ACCESS A DATABASE IN THE MIDDLE OR TO COMPLETE AN INTEGRATION

To finish an integration by accessing a database, add a database connection as the finish connection. To access a database in the middle of an integration, add a database connection between the start and finish connections.

### Prerequisites

- You created a database connection.
- You are creating or editing an integration.
- You are on the Fuse Online **Choose a Finish Connection** page or the **Choose a Connection** page.

### Procedure

1. Click the database connection for the database you want to access.
2. On the **Choose an Action** page, click one of the following:
  - **Invoke SQL** operates on data by executing the SQL statement you specify.
  - **Invoke stored procedure** operates on data by invoking the stored procedure you specify or select.
3. If you selected **Invoke SQL**, in the **SQL Statement** field:
  - For a middle connection, enter a SQL **SELECT** statement that obtains one or more records or enter a SQL **INSERT**, **UPDATE**, or **DELETE** statement that operates on one or more records. The database table that contains the data must already exist.
  - For a finish connection, enter a SQL **INSERT**, **UPDATE** or **DELETE** statement to operate on one or more records.

If you selected **Invoke stored procedure**, in the **Procedure name** field, select or enter the name of the stored procedure to invoke to operate on the data of interest. The stored procedure you specify must already exist. The database administrator should have created any stored procedures you need to use in an integration.

- See the information below about specifying placeholder parameters in queries.

4. Click **Done**.

## Result

Fuse Online tries to validate the connection, which includes checking that a specified SQL query is syntactically correct and confirming that the query or stored procedure target data exists. If verification is successful then Fuse Online adds the connection to the integration. If verification fails then Fuse Online displays a message about the problem. Update your input as needed and try again.

## Specifying parameters in queries

When you access a database in the middle of an integration or to complete an integration, you can specify placeholder parameters in the SQL query or there can be placeholders in the stored procedure. For example:

```
INSERT INTO TODO(task, completed) VALUES(:#param_1, :#param_2)
DELETE FROM TODO WHERE task LIKE :#param_3
```

To specify the values of these placeholders, add a data mapping step to your integration before the database connection. In the data mapping step, map the appropriate source data fields to the target data fields, for example, map source data to the `:#param_1`, `:#param_2`, and `:#param_3` target fields. See [Add a data mapping step](#).

## 18.4. CONNECT TO PROPRIETARY DATABASES

To connect to a proprietary SQL database, the main tasks that must be accomplished are as follows:

1. A developer creates a library extension that contains the JDBC driver for the database that you want to access in an integration. See [Creating JDBC driver library extensions](#).
2. The developer provides a `.jar` file that contains the library extension.
3. You upload that `.jar` file to Fuse Online. See [Making extensions available](#).
4. You create a connection to your database by selecting the Fuse Online **Database** connector and specifying the connection URL for your database. See [Creating database connections](#).
5. In an integration, you add the connection to your database. See [Start an integration by accessing a database](#) or [Access a database in the middle of an integration or to finish an integration](#).

## CHAPTER 19. CONNECT TO TELEGRAM

As a business user, you can create an integration that connects to Telegram. A connection to Telegram can do either one of the following:

- Trigger execution of the integration when a Telegram bot that you specify receives a message. The integration passes the message to the next step in the integration. For example, after an integration receives a message there might be a filter step that watches for keyword instances such as product names. Upon finding messages that contain those product names, the integration can notify the appropriate contact in a Gmail connection or in a different Telegram connection.
- Deliver a message to a particular Telegram chat. For example, this behavior is useful when an integration downloads a file from an FTP server and processes it in some way. You can finish an integration by notifying a Telegram chat that the process was successful.

To connect to Telegram in an integration, create a Telegram connection and then add the connection to an integration. Details are in the following topics:

- [Section 19.1, “Create a Telegram connection”](#)
- [Section 19.2, “Add a Telegram connection to trigger integration execution upon receiving messages”](#)
- [Section 19.3, “Add a Telegram connection to send a message to a Telegram chat”](#)

### 19.1. CREATE A TELEGRAM CONNECTION

In an integration, a Telegram connection can receive messages that have been sent to a particular chat. A chat bot that you create forwards the messages to Fuse Online. A Telegram connection can also send messages to chats that the connection has access to. You can use the same Telegram connection:

- In any number of integrations that use the connection to send messages to a chat.
- In addition, you can use it in one integration that uses the connection to receive messages from a Telegram chat.

In other words, you cannot use the same Telegram connection in more than one integration if you want that connection to receive Telegram chat messages in both integrations. This is a Telegram limitation.

#### Prerequisites

- In Telegram, you created a bot for the chat that you want to obtain messages from and send messages to. For help with this, see [Bots, an introduction for developers](#).
- You obtained a bot authorization token from the Telegram BotFather.

#### Procedure

1. In Fuse Online, in the left panel, click **Connections** to display any available connections.
2. In the upper right, click **Create Connection** to display Fuse Online connectors.
3. Click the **Telegram** connector.

4. In the **Authorization Token** field, enter the Telegram authorization token for the bot that you want to get messages from and send messages to.
5. Click **Next**.
6. In the **Connection Name** field, enter your choice of a name that helps you distinguish this connection from any other connections. For example, enter **Telegram for Product Mentions**.
7. In the **Description** field, optionally enter any information that is helpful to know about this connection.
8. In the upper right, click **Create** to see the updated list of available connections, including the connection that you just created. If you entered the example name, you would see that **Telegram for Product Mentions** appears as a connection that you can choose to add to an integration. You can add this connection to any number of integrations.

## 19.2. ADD A TELEGRAM CONNECTION TO TRIGGER INTEGRATION EXECUTION UPON RECEIVING MESSAGES

A Telegram connection that starts an integration triggers execution of the integration each time that the connection receives a message from a Telegram chat bot that you specify.

### Prerequisites

- You created a [Telegram connection](#).
- You are creating an integration and Fuse Online is prompting you to choose how you want to start the integration.

### Procedure

1. Click the Telegram connection that you want to use to start the integration.
2. Click the **Read Messages** action to receive all messages that are sent to the chat bot. The connection identifies the chat bot that you want to receive messages from. No other configuration is required.

## 19.3. ADD A TELEGRAM CONNECTION TO SEND A MESSAGE TO A TELEGRAM CHAT

In an integration, you can send a message to a Telegram chat to finish an integration or in the middle of an integration.

### Prerequisites

- You created a [Telegram connection](#).
- You are creating or editing an integration. If you are creating an integration, then Fuse Online might be prompting you to choose a finish connection. To add a middle connection, hover over the plus sign in the left panel in the location where you want to add the connection and select **Add a connection**.

### Procedure

1. Click the Telegram connection that you want to add to the integration.
2. Click the **Send a Text Message** action.
3. In the **Chat Id** field, enter the ID for the Telegram chat that you are sending a message to. If you leave this field blank, and if there is an earlier Telegram connection that receives messages, then Fuse Online uses the chat ID that is in an incoming message.

To obtain the ID for a chat:

- a. In Telegram, add a bot to the chat that you want to send a message to. The BotFather response includes a bot access token.
- b. Copy the access token to the clipboard.
- c. In Telegram, send a message directly to that chat. Do not use the bot to send the message.
- d. Read messages that are sent to that chat by invoking a **curl** command in the following format. Replace **bot-access-token** with the access token in the clipboard.  
**curl https://api.telegram.org/bot-access-token/getUpdates\?offset\=0**

The response is in JSON format. The JSON **chat** object contains the chat ID.

4. Click **Done** to add the connection to the integration.

## CHAPTER 20. TRIGGER INTEGRATION EXECUTION WITH A TIMER

To trigger execution of an integration according to a schedule that you specify, add a timer connection as an integration's start connection. Fuse Online provides a timer connection, which you can use to start as many integrations as you like. You do not need to create a timer connection unless you inadvertently delete the provided timer connection. Details are in the following topics:

- [Section 20.1, "Add a timer connection to trigger integration execution"](#)
- [Section 20.2, "Create a replacement timer connection"](#)

### 20.1. ADD A TIMER CONNECTION TO TRIGGER INTEGRATION EXECUTION

To trigger execution of an integration according to a schedule that you specify, add a timer connection as an integration's start connection. A timer connection cannot be in the middle of an integration or at the end of an integration.

#### Procedure

1. In Fuse Online, on the left, click **Integrations**.
2. In the upper right, click **Create Integration**.
3. On the **Choose a Start Connection** page, click the **Timer** connection.  
Fuse Online provides a **Timer** connection; you do not need to create a timer connection. However, if you inadvertently deleted the provided **Timer** connection and created a new timer connection then you might have given it a different name.
4. On the **Choose an Action** page, click **Cron Timer** or **Simple Timer**.
  - A **cron** timer requires a **cron** expression that specifies the schedule for when to trigger integration execution.
  - A simple timer prompts you to specify a period and its time unit, for example, **5 seconds, 1 hour**. Available units are milliseconds, seconds, minutes, hours, days.
5. According the type of timer you are adding, enter a **cron** expression or a period with a selected time unit.
6. Click **Done** to add the **Timer** connection as the integration's start connection.

### 20.2. CREATE A REPLACEMENT TIMER CONNECTION

Fuse Online provides a timer connection; you do not need to create one. However, if you inadvertently delete the provided timer connection then you can use the **Timer** connector to create a timer connection.

#### Procedure

1. In Fuse Online, in the left panel, click **Connections**.
2. In the upper right, click **Create Connection**.

3. Click the **Timer** connector.
4. In the **Add Connection Details** page, enter a name for the connection, for example, **Timer**.
5. Optionally, enter a description that distinguishes this connection.
6. In the upper right, click **Create**. Fuse Online displays a list of available connections and you can see the timer connection that you just created.

## CHAPTER 21. CONNECT TO TWITTER

To connect to Twitter in an integration, the main steps are:

- [Section 21.1, “Register Fuse Online as a Twitter client application”](#)
- [Section 21.2, “Create a Twitter connection”](#)
- [Section 21.3, “Add a Twitter connection to an integration”](#)

You need a Twitter developer account to authorize access from your Fuse Online environment to Twitter. If you do not already have a Twitter developer account, go to <https://apps.twitter.com>, sign in to Twitter, and click **Apply for a developer account**. Obtaining the developer account is quick when you provide a professional email address, such as `me@redhat.com` rather than a personal address such as `you@gmail.com`.

### 21.1. REGISTER FUSE ONLINE AS A TWITTER CLIENT APPLICATION


In an integration, to connect to Twitter, the first thing you must do is register your Fuse Online environment as a client application that can access Twitter. This lets you create any number of integrations that connect to Twitter. In other words, you need to register a particular Fuse Online environment with Twitter only once.

In each Fuse Online environment, there can be only one registration of Fuse Online as a Twitter client application. However, while each Twitter connection uses the same registration, it can use different user credentials.

#### Prerequisite

You are logged in to Fuse Online. The URL for accessing Fuse Online is in the email message that welcomed you to the Red Hat Fuse Online Evaluation program.

#### Procedure

1. In Fuse Online:
  - a. In the left panel, click **Settings**.
  - b. On the **Settings** page, near the top, to the right of the callback URL, click  to copy the callback URL for your installation of Fuse Online to the clipboard. You will need this URL later in this procedure.
  - c. To the right of the **Twitter** entry, click **Register** to display the **Consumer Key** and **Consumer Secret** fields.
2. In another browser tab, go to your [Twitter developer account](#), sign in if you need to, and do the following:
  - a. In the upper right, next to the name of your developer project, click the down caret and click **Apps**.
  - b. In the upper right, click **Create an app**.
  - c. In the **App name** field, enter a name such as **My Fuse Online Client**.
  - d. In the **Application description** field, enter a tip such as **For leveraging tweets**.



- e. In the **Website URL** field, paste the URL that you copied at the beginning of this procedure and remove **api/v1/credentials/callback** from the end of the URL.
  - f. Select **Enable Sign in with Twitter**.
  - g. In the **Callback URLs** field, paste the URL again. It should be something like this:  
**https://app-proj9128.7b63.fuse-ignite.openshiftapps.com/api/v1/credentials/callback.**
  - h. A 100-character description of how you plan to use this app is required. For convenience, you can copy and paste: **This is a demo application so that I can evaluate how to integrate Twitter with my enterprise applications.**
  - i. Click **Create**.
  - j. Click **Create** again to confirm that you reviewed the developer terms.
  - k. At the top of the page for the app you just created, click the **Keys and tokens** tab.
    - l. Copy the API key.
3. Return to your Fuse Online **Settings** page and paste the Twitter consumer API key into the Fuse Online Twitter **Consumer Key** field.
  4. Return to the Twitter **Keys and tokens** tab and copy the consumer API secret key.
  5. Return to your Fuse Online **Settings** page and paste the Twitter consumer API secret key into the Fuse Online Twitter **Consumer Secret** field.
  6. Click **Save** and then click **Ok**.

## 21.2. CREATE A TWITTER CONNECTION

To create an integration that obtains data from Twitter, you must first create a Twitter connection. After you create a Twitter connection, you can use it in any number of integrations.

### Prerequisites

- You are logged in to Fuse Online. The URL for accessing Fuse Online is in the email message that welcomed you to the Red Hat Fuse Online Evaluation program.
- Fuse Online is open in a web browser.
- You registered your Fuse Online environment as an application that can access Twitter.

### Procedure

1. In Fuse Online, in the left panel, click **Connections** to display any available connections.
2. In the upper right, click **Create Connection** to display the available connectors. A connector is a template that you use to create one or more connections.
3. Click the **Twitter** connector.
4. Click **Connect Twitter** to display a Twitter authorization page. You might need to log in to Twitter before you see the authorization page.

5. Click **Authorize app** to return to Fuse Online.
6. In the **Connection Name** field, enter your choice of a name that helps you distinguish this connection from any other connections. For example, enter **Twitter Connect 1**.
7. In the **Description** field, optionally enter any information that is helpful to know about this connection. For example, enter **Sample Twitter connection that uses my Twitter login credentials**.
8. In the upper right, click **Create** to see that the connection you created is now available. If you entered the example name, you would see that **Twitter Connect 1** appears as a connection that you can choose to add to an integration.

## 21.3. ADD A TWITTER CONNECTION TO AN INTEGRATION

In an integration, a connection to Twitter can trigger execution of an integration when a tweet contains your Twitter handle or when a tweet contains text that you have specified. A connection to Twitter must be the integration's start connection.

### Prerequisites

- You created a Twitter connection.
- You must be creating or editing an integration and Fuse Online is prompting you to choose the start connection.

### Procedure

1. On the page that displays available connections, click the Twitter connection that you want to add to the integration. When the integration uses the selected connection to connect to Twitter, Fuse Online uses the credentials defined in that connection.
2. Click the action that you want the selected connection to perform. A Twitter connection that you add to an integration performs only the action you choose.
  - **Mention** triggers execution of the integration when a tweet contains your handle.
  - **Search** triggers execution of the integration when a tweet contains text that you specify.
3. Optionally, enter the configuration information that Fuse Online prompts for. For example, the **Search** action prompts you to specify how often to search and keywords to search for.
4. Click **Done** to add the connection to the integration.

### Result

The connection appears in the integration flow as the start connection.

## CHAPTER 22. TRIGGER INTEGRATION EXECUTION WITH AN HTTP REQUEST (WEBHOOK)

You can trigger execution of an integration by sending an HTTP **GET** or **POST** request to an HTTP endpoint that Fuse Online exposes. The following topics provide details:

- [Section 22.1, “General procedure for using the Fuse Online webhook”](#)
- [Section 22.2, “Use the Webhook connection”](#)
- [Section 22.3, “Create an integration that an HTTP request can trigger”](#)
- [Section 22.4, “How Fuse Online handles HTTP requests”](#)
- [Section 22.5, “Guidelines for an HTTP client that invokes a Fuse Online Webhook”](#)
- [Section 22.6, “About the JSON schema for specifying request parameters”](#)
- [Section 22.7, “How to specify HTTP requests”](#)

### 22.1. GENERAL PROCEDURE FOR USING THE FUSE ONLINE WEBHOOK

To trigger execution of an integration with an HTTP **GET** or **POST** request, you must do the following:

1. Decide whether you want to send a **GET** or **POST** request to Fuse Online.
2. Plan your integration to handle this request.
3. Create the connection that finishes the integration.  
Fuse Online provides a Webhook connection that you use as the start connection.
4. Create any other connections that you want to add to the integration.
5. Create the integration:
  - a. Add the Webhook connection as the start connection.
  - b. Add the finish connection and then any other connections that are required in the integration. The finish connection and any middle connections handle the HTTP request that triggers execution of the integration. It is up to you to choose and specify the most appropriate HTTP request for accomplishing your goals. Keep the following in mind:
    - Add a connection to the application that contains the data that you want to obtain or that contains the data that you want to update.
    - A **GET** request is limited to specification of key/value parameters.
    - A **POST** request can provide an arbitrary body, such as an XML or JSON instance.
    - Fuse Online returns only an HTTP status header and does not return any data. Consequently, you can define an integration that is triggered by a **GET** request and that updates data rather than obtaining data. Likewise, you can define an integration that is triggered by a **POST** request and that obtains data rather than updating data.

- c. Add a data mapper step after the Webhook connection.  
For a **GET** request, map the parameter fields in the HTTP request to the data fields in the next connection.

For a **POST** request, you might have specified the output data shape in the request by passing a JSON instance, JSON schema, XML instance, or XML schema. If you did not, then when you add a Webhook connection as the start connection of an integration, Fuse Online prompts you to specify the output data type. If you do not, then the default Webhook connection output data type is in JSON format.

- d. Add any other steps that the integration needs.
6. Publish the integration and wait for it to be **Running**.
7. Go to the integration summary page and copy the external URL that Fuse Online provides.
8. Modify this external URL to construct your **GET** or **POST** request.
9. Implement the application that sends the HTTP **GET** or **POST** request to Fuse Online.

## 22.2. USE THE WEBHOOK CONNECTION

Fuse Online provides a Webhook connection that you can use in any number of integrations as the start connection. Each time you add this Webhook connection to an integration, Fuse Online provides a new random token for use with that integration. You paste this token at the end of your **GET** or **POST** request. You do not need to create a Webhook connection unless you inadvertently delete the provided Webhook connection.

### Prerequisite

You inadvertently deleted the Webhook connection that was provided with Fuse Online.

### Procedure:

1. In Fuse Online, in the left panel, click **Connections** to display any available connections.
2. In the upper right, click **Create Connection** to display connectors.
3. Click the **Webhook** connector.
4. In the **Connection Name** field, enter your choice of a name that helps you distinguish this connection from any other connections. For example, you might enter **Webhook 1**.
5. In the **Description** field, optionally enter any information that is helpful to know about this connection. For example, enter **Sample Webhook connection**.
6. In the upper right, click **Create** to see that the connection you created is now available. If you entered the example name, you would see that **Webhook 1** appears as a connection that you can choose to add to an integration.

## 22.3. CREATE AN INTEGRATION THAT AN HTTP REQUEST CAN TRIGGER

To trigger execution of an integration with an HTTP **GET** or **POST** request, add a Webhook connection as the integration's start connection.

## Procedure

1. In the Fuse Online panel on the left, click **Integrations**.
2. Click **Create Integration**.
3. On the **Choose a Start Connection** page, click the Webhook connection.
4. On the **Choose an Action** page, click the **Incoming Webhook** action.  
In the **Webhook Configuration** page, Fuse Online displays the webhook token that Fuse Online generates for this integration.

When you construct the HTTP request, this token is the last part of the URL. After you publish this integration and it is running, Fuse Online displays the Fuse Online external URL, which has this token at the end.

5. Click **Next**.
6. In the **Specify Output Data Type** page:
  - a. In the **Select Type** field, select **JSON schema**.
  - b. In the **Definition** field, paste the JSON schema that defines the data types of the parameters in the HTTP request. See [JSON schema for specifying request parameters](#).
  - c. In the **Data Type Name** field, specify a name for this data type. Although this is optional, if you specify a name, it appears in the data mapper **Sources** list, which can make it easier to correctly map fields.
  - d. Optionally, in the **Data Type Description** field, provide some information that helps you distinguish this data type.
  - e. Click **Done**.
7. Add the finish connection to the integration.
8. Add any other needed connections.
9. Immediately after the start connection, add a data mapper step.
10. Add any other needed steps.
11. Click **Publish**, give the integration a name and optionally, a description, and click **Publish** again.

## 22.4. HOW FUSE ONLINE HANDLES HTTP REQUESTS

You can specify an HTTP **GET** or **POST** request to trigger execution of an integration. Although a **GET** request usually obtains data and a **POST** request usually updates data, you can use either request to trigger an integration that does either operation. Any parameters in the request are available for mapping to data fields in the next connection that is in the integration. For details, see [JSON schema for specifying request parameters](#).

A Webhook connection only passes the data it receives to the next connection in the integration. When Fuse Online receives an HTTP request, it:

- Returns an HTTP status header to the requester. When a request successfully triggers execution of an integration, the Fuse Online return code is **201**. When a request fails to trigger execution of an integration, the Fuse Online return code is **5xx**.
- Does not return any other data to the requester. In other words, there is no data in the HTTP body of the response that contains the status header.
- Passes the data in the request to the next connection in the integration.

This means that you can define an integration that is triggered by a **GET** request and that updates data rather than obtaining data. Likewise, you can define an integration that is triggered by a **POST** request and that obtains data rather than updating data.

## 22.5. GUIDELINES FOR AN HTTP CLIENT THAT INVOKES A FUSE ONLINE WEBHOOK

When you implement a client that sends an HTTP request to Fuse Online, your implementation should:

- Add to the Fuse Online-provided external URL to construct a URL that makes a **GET** or **POST** request.
- In the URL request, specify HTTP header and query parameter values whose data types adhere to the **io:syndesis:webhook** JSON schema. See [JSON schema for specifying request parameters](#). When header and query parameters adhere to this data type specification, then you can map parameter fields to fields that the next connection in the integration can process.
- If the request succeeds, handle a returned success code of **201**.
- If the request fails, handle an HTTP **5xx** error code.
- Not expect any other response from Fuse Online. In other words, sending the request does not directly return data to the requesting client other than the return code.

## 22.6. ABOUT THE JSON SCHEMA FOR SPECIFYING REQUEST PARAMETERS

In an integration, you typically map header and query parameters in the HTTP request to data fields that the next connection in the integration can process. To make this possible, when you add the Webhook connection to the integration, specify the output data type in a JSON schema that has the following structure:

```
{
  "$schema": "http://json-schema.org/schema#",
  "id": "io:syndesis:webhook",
  "type": "object",
  "properties": {
    "parameters": {
      "type": "object",
      "properties": { 1
      }
    },
    "body": {
      "type": "object",
      "properties": { 2
      }
    }
  }
}
```

```

{
  }
  }
  }
}

```

To add the data structures that you need, in the JSON instance for your HTTP request:

- 1 Specify query parameters in the **properties** section under the **parameters** object.
- 2 Specify the HTTP body schema in the **properties** section under the **body** object.

While all data that an HTTP client sends is available in the integration, when a Webhook connection's data shape conforms to this JSON schema, then query parameters and body content are available for mapping.

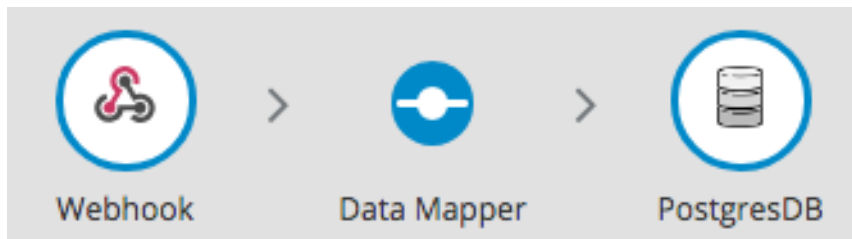
For examples, see [How to specify HTTP requests](#).

## 22.7. HOW TO SPECIFY HTTP REQUESTS

The following examples show how to specify HTTP requests for the Fuse Online Webhook.

### Webhook example of POST request with only HTTP body

Consider an integration that starts with a Webhook connection and then creates a row in the **Todo** table of the Fuse Online-provided database:



During creation of this integration, when you add the Webhook start connection, you specify its output data type with a JSON instance that has this content: `{"todo": "text"}`:

Home > Integrations > New Integration > Configure Incoming Webhook

## Webhook

### Specify Output Data Type

This is a typeless connection. To use type-aware functionality, enter information that defines the data type

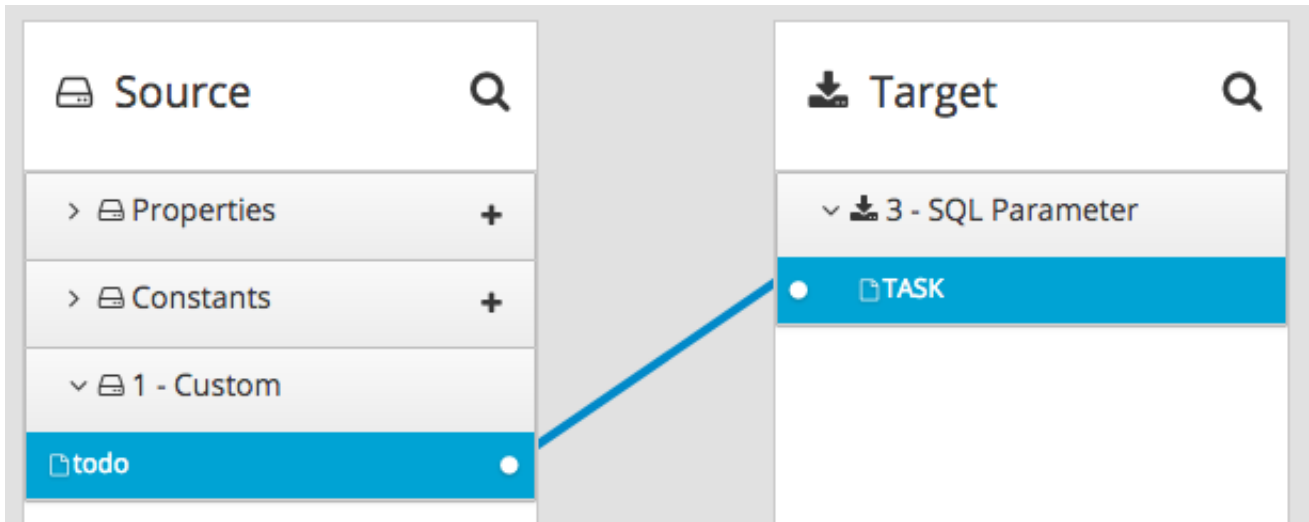
Select Type	<input type="text" value="JSON Instance"/>
Definition ⓘ	<pre>("todo":"text")</pre>
Data Type Name	<input type="text" value="Custom"/>
Data Type Description	<input type="text" value="A user specified data type"/>

When you add the **PostgresDB** connection as the finish connection, you select the **Invoke SQL** action and specify this SQL statement:

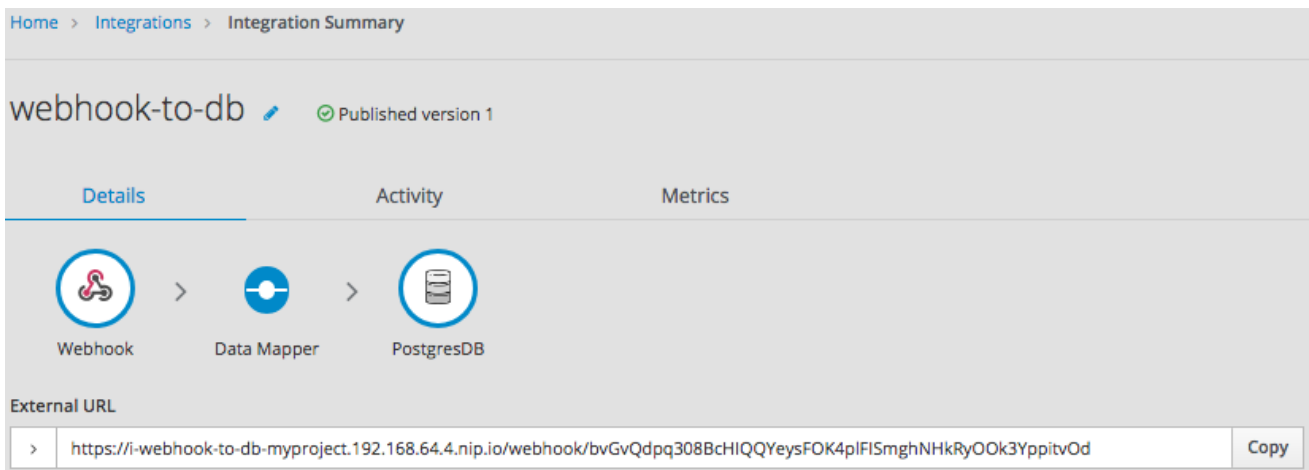
```
INSERT INTO TODO (TASK) VALUES (:#TASK)
```

After you add the database connection, you add a mapping step:





You save and publish the integration. When it is running, you can copy the external URL that Fuse Online provides:



To understand the parts of the external URL, consider this sample URL:

**https://i-webhook-to-db-myproject.192.168.64.4.nip.io/webhook/bvGvQdpq308BcHIQQYeysFOk4p1FISmghNHkRyOOk3Yppitv0d**

Value	Description
<b>i-</b>	Fuse Online always inserts this value at the beginning of the URL.
<b>webhook-to-db</b>	The name of the integration.
<b>myproject</b>	The OpenShift namespace that contains the pod that is running the integration.
<b>192.168.64.4.nip.io</b>	The DNS domain that is configured for OpenShift. This indicates the Fuse Online environment that is providing the webhook.
<b>webhook</b>	Appears in each Webhook connection URL.

Value	Description
<b>bvGvQdpq308BcHIQ QYeysFOk4p1FISmg hNHkRy00k3Yppitv Od</b>	<p>Webhook connection token that Fuse Online provides when you add a Webhook connection to an integration. The token is a random string that provides security in that it makes the URL hard to discern, which prevents anyone else from sending a request.</p> <p>In a request, you can specify the token that Fuse Online provides or you can define your own. If you define your own, ensure that it is hard to guess.</p>

As you can see in the external URL, Fuse Online constructs the host name from the name of the integration, the name of the OpenShift namespace, and the OpenShift DNS domain. Fuse Online removes illegal characters and converts spaces to hyphens. In the sample external URL, this is the host name:

**`https://i-webhook-to-db-myproject.192.168.64.4.nip.io`**

To use `curl` to invoke the webhook, you would specify the command as follows:

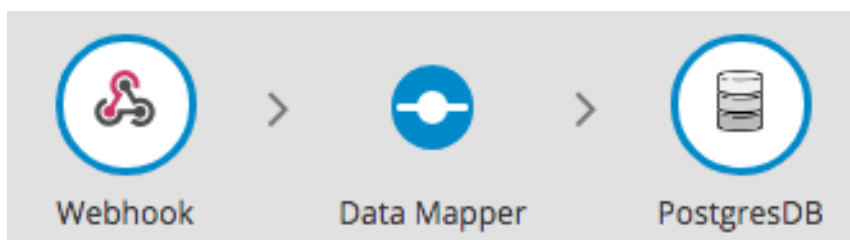
```
curl -H 'Content-Type: application/json' -d '{"todo":"from webhook"}'  
https://i-webhook-to-db-  
myproject.192.168.64.4.nip.io/webhook/bvGvQdpq308BcHIQYeysFOk4p1FISmghNHkRyO  
Ok3YppitvOd
```

- The `-H` option specifies the HTTP **Content-Type** header.
- The `-d` option sets the HTTP method to **POST** by default.

Execution of this command triggers the integration. The database finish connection inserts a new task into the tasks table. To see this, display the **Todo** app at, for example, **`https://todo-myproject.192.168.64.4.nip.io`**, Click **Update** and you should see **from webhook** as a new task.

### Webhook example of POST request with query parameters

For this example, consider the same integration as in the previous example:



But in this example, you define the Webhook connection output data type by specifying a JSON schema with this content:

```
{
  "type": "object",
  "definitions": {},
  "$schema": "http://json-schema.org/draft-07/schema#",
  "id": "io:syndesis:webhook",
```

```

"properties": {
  "parameters": {
    "type": "object",
    "properties": {
      "source": {
        "type": "string"
      },
      "status": {
        "type": "string"
      }
    }
  },
  "body": {
    "type": "object",
    "properties": {
      "company": {
        "type": "string"
      },
      "email": {
        "type": "string"
      },
      "phone": {
        "type": "string"
      }
    }
  }
}
}
}

```

In this JSON schema:

- The **id** must be set to **io.syndesis.webhook**.
- The **parameters** section must specify the HTTP query parameters.
- The **body** section must specify the body content and it can be as complex as you need it to be. For example, it can define nested properties as well as arrays.

This provides the information that the Webhook connector needs to prepare the content for the next step in the integration.

To use **curl** to send an HTTP request, invoke a command such as the following:

```

curl -H 'Content-Type: application/json' -d
'{"company": "Gadgets", "email": "sales@gadgets.com", "phone": "+1-202-555-
0152"}' https://i-webhook-params-to-db-
myproject.192.168.42.235.nip.io/webhook/ZYWrhaW7dVk097vNsLX3YJ1GyxUFMFRteLpw0
z4069MW7d2Kjg?source=web&status=new

```

When the Webhook connection receives this request it creates a JSON instance that looks like this:

```

{
  "parameters": {
    "source": "web",
    "status": "new"
  },

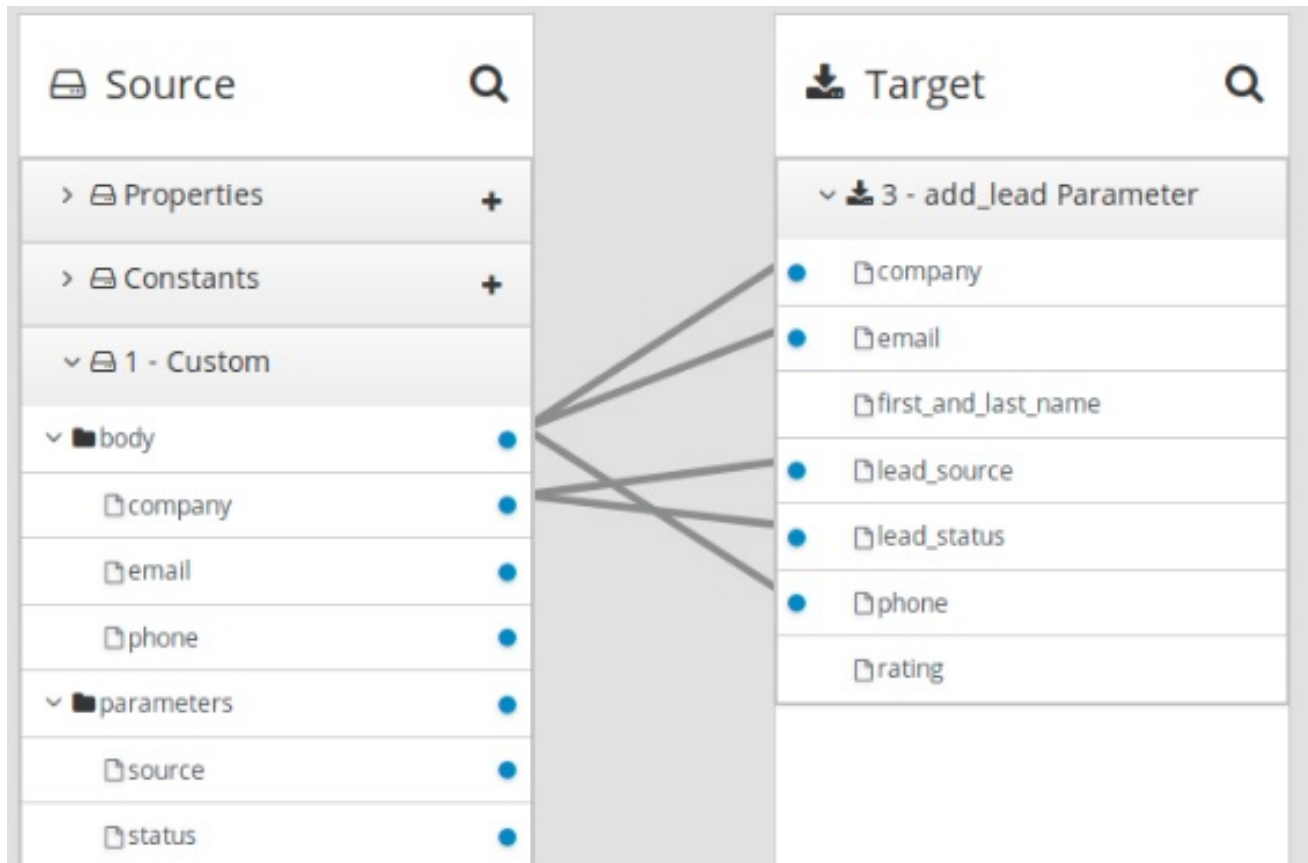
```

```

"body": {
  "company": "Gadgets",
  "email": "sales@gadgets.com",
  "phone": "+1-202-555-0152"
}
}

```

It is this internal JSON instance that enables the following mapping:



## Webhook examples with GET

To trigger an integration with a **GET** request that does not provide input data, specify the Webhook connection output data shape as a JSON instance with the definition `'{}'`. You can then invoke the following **curl** command, which does not specify query parameters:

```

curl 'https://i-webhook-params-to-db-myproject.192.168.42.235.nip.io/webhook/ZYWrhaW7dVk097vNsLX3YJ1GyxUFMRteLpw0z4069MW7d2Kjg'

```

You can change the previous **POST** example to send a **GET** request with query parameters but no body. You would specify the Webhook connection output data shape as a JSON schema with the definition as shown below.

```

{
  "type": "object",
  "definitions": {},
  "$schema": "http://json-schema.org/draft-07/schema#",
  "id": "io:syndesis:webhook",
  "properties": {
    "parameters": {
      "type": "object",

```

```
    "properties": {  
      "source": {  
        "type": "string"  
      },  
      "status": {  
        "type": "string"  
      }  
    }  
  }  
}
```

Here is the **curl** command that sends the **GET** request:

```
curl 'https://i-webhook-params-to-db-  
myproject.192.168.42.235.nip.io/webhook/ZYWrhaW7dVk097vNsLX3YJ1GyxUFMFRteLpw0  
z4069MW7d2Kjg?source=web&status=new'`
```