



Red Hat Fuse 7.3

Connecting Fuse Online to Applications and Services

Instructions for Using Provided Connectors

Red Hat Fuse 7.3 Connecting Fuse Online to Applications and Services

Instructions for Using Provided Connectors

Legal Notice

Copyright © 2019 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, the Red Hat 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 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

PREFACE	5
CHAPTER 1. CONNECTORS THAT ARE SUPPORTED BY FUSE ONLINE	6
CHAPTER 2. CONNECTING TO AMAZON S3	8
2.1. PREREQUISITES FOR CREATING AN AMAZON S3 CONNECTION	8
2.2. CREATING AMAZON S3 CONNECTIONS	9
2.3. OBTAINING DATA FROM AMAZON S3 TO TRIGGER INTEGRATION EXECUTION	10
2.4. ADDING DATA TO AMAZON S3 TO FINISH AN INTEGRATION	11
2.5. ADDING DATA TO AMAZON S3 IN THE MIDDLE OF AN INTEGRATION	12
CHAPTER 3. CONNECTING TO AMQ	14
3.1. CREATING AN AMQ CONNECTION	14
3.2. ADDING AN AMQ CONNECTION TO TRIGGER INTEGRATION EXECUTION UPON RECEIVING MESSAGES	15
3.3. PUBLISHING AMQ MESSAGES TO FINISH AN INTEGRATION	17
3.4. PUBLISHING AMQ MESSAGES IN THE MIDDLE OF AN INTEGRATION	18
CHAPTER 4. CONNECTING TO AMQP	21
4.1. CREATING AN AMQP CONNECTION	21
4.2. ADDING AN AMQP CONNECTION TO TRIGGER INTEGRATION EXECUTION UPON RECEIVING MESSAGES	22
4.3. PUBLISHING MESSAGES TO AMQP IN THE MIDDLE OF A FLOW OR TO FINISH AN INTEGRATION	24
CHAPTER 5. CONNECTING TO DROPBOX	27
5.1. REGISTERING FUSE ONLINE AS A DROPBOX CLIENT	27
5.2. CREATING A DROPBOX CONNECTION	28
5.3. OBTAINING FILES FROM DROPBOX TO TRIGGER INTEGRATION EXECUTION	29
5.4. ADDING FILES TO DROPBOX TO FINISH AN INTEGRATION	30
5.5. ACCESSING DROPBOX IN THE MIDDLE OF AN INTEGRATION	31
CHAPTER 6. CONNECTING TO FHIR	33
6.1. CREATING A CONNECTION TO A FHIR SERVER	33
6.2. OBTAINING A RESOURCE FROM A FHIR SERVER	34
6.3. QUERYING A FHIR SERVER FOR RESOURCES	35
6.4. CREATING A RESOURCE ON A FHIR SERVER	36
6.5. UPDATING ALL FIELDS IN A RESOURCE ON A FHIR SERVER	37
6.6. UPDATING SPECIFIED FIELDS IN A RESOURCE ON A FHIR SERVER	38
6.7. CREATING RESOURCES OF DIFFERENT TYPES ON A FHIR SERVER	40
6.8. DELETING A RESOURCE FROM A FHIR SERVER	40
CHAPTER 7. CONNECTING TO AN FTP OR SFTP SERVER	42
7.1. CREATING AN FTP OR SFTP CONNECTION	42
7.2. OBTAINING FILES FROM AN FTP OR SFTP SERVER	43
7.3. UPLOADING FILES TO AN FTP OR SFTP SERVER	45
CHAPTER 8. CONNECTING TO GOOGLE APPLICATIONS	47
8.1. REGISTERING FUSE ONLINE AS A GOOGLE CLIENT APPLICATION	47
8.2. CONNECTING TO GMAIL	49
8.2.1. Creating a Gmail connection	50
8.2.2. Alternatives for populating email to send	51
8.2.3. Triggering an integration when polling returns a Gmail message	52
8.2.4. Sending an email from a Gmail account	53
8.3. CONNECTING TO GOOGLE CALENDAR	54

8.3.1. Creating a Google Calendar connection	54
8.3.2. Triggering an integration when polling returns an event from a Google Calendar	55
8.3.3. Obtaining a particular event from a Google Calendar	56
8.3.4. Adding an event to a Google Calendar	57
8.3.5. Updating an event in a Google Calendar	57
8.4. CONNECTING TO GOOGLE SHEETS	58
8.4.1. Creating a Google Sheets connection	59
8.4.2. Triggering an integration when polling returns spreadsheet data	60
8.4.3. Triggering an integration when polling returns spreadsheet properties	62
8.4.4. Creating a spreadsheet	62
8.4.5. Updating data in a sheet	63
8.4.6. Appending data to a sheet	65
8.4.7. Updating spreadsheet properties	66
8.4.8. Adding a chart to a sheet	67
8.4.9. Adding a pivot table to a sheet	70
CHAPTER 9. CONNECTING TO HTTP AND HTTPS ENDPOINTS	74
9.1. CREATING A CONNECTION TO AN HTTP OR HTTPS ENDPOINT	74
9.2. ADDING AN HTTP OR HTTPS CONNECTION TO AN INTEGRATION	74
CHAPTER 10. CONNECTING TO IRC	77
10.1. CREATING AN IRC CONNECTION	77
10.2. ADDING AN IRC CONNECTION TO TRIGGER INTEGRATION EXECUTION UPON RECEIVING MESSAGES	78
10.3. ADDING AN IRC CONNECTION TO SEND A MESSAGE	79
CHAPTER 11. CONNECTING TO KAFKA	81
11.1. CREATING A CONNECTION TO A KAFKA BROKER	81
11.2. OBTAINING DATA FROM A KAFKA BROKER TO TRIGGER INTEGRATION EXECUTION	81
11.3. PUBLISHING DATA TO A KAFKA BROKER	83
CHAPTER 12. CONNECTING TO APACHE KUDU	85
12.1. CREATING A CONNECTION TO AN APACHE KUDU DATA STORE	85
12.2. TRIGGERING AN INTEGRATION WHEN SCANNING RETURNS RECORDS FROM A KUDU TABLE	85
12.3. INSERTING RECORDS INTO A KUDU TABLE	86
CHAPTER 13. CONNECTING TO MQTT	88
13.1. CREATING A CONNECTION TO AN MQTT BROKER	88
13.2. OBTAINING A MESSAGE FROM AN MQTT BROKER TO TRIGGER INTEGRATION EXECUTION	89
13.3. PUBLISHING A MESSAGE TO AN MQTT BROKER	90
CHAPTER 14. CONNECTING TO ODATA	92
14.1. CREATING A CONNECTION TO AN ODATA SERVICE	92
14.2. TRIGGERING AN INTEGRATION WHEN POLLING RETURNS DATA FROM AN ODATA SERVICE	93
14.3. UPDATING, CREATING, AND DELETING DATA THAT IS MANAGED BY AN ODATA SERVICE	95
CHAPTER 15. CONNECTING TO REST APIS	97
15.1. REGISTERING FUSE ONLINE AS A REST API CLIENT	97
15.2. CREATING A REST API CLIENT CONNECTION	98
15.3. ADDING AN API CLIENT CONNECTION TO AN INTEGRATION	98
CHAPTER 16. CONNECTING TO SALESFORCE	100
16.1. REGISTERING FUSE ONLINE AS A SALESFORCE CLIENT APPLICATION	100
16.2. CREATING A SALESFORCE CONNECTION	101
16.3. ADDING A SALESFORCE CONNECTION TO AN INTEGRATION	102

CHAPTER 17. CONNECTING TO SAP CONCUR	104
17.1. USER ROLES FOR CONNECTING TO SAP CONCUR	104
17.2. HOW TO OBTAIN SAP CONCUR IMPLEMENTATION SITE OAUTH CREDENTIALS	104
17.3. HOW TO OBTAIN SAP CONCUR PRODUCTION SITE OAUTH CREDENTIALS	105
17.4. CONFIGURING THE SAP CONCUR CONNECTOR	105
17.5. CREATING A SAP CONCUR CONNECTION	106
17.6. ADDING A SAP CONCUR CONNECTION TO AN INTEGRATION	107
17.7. EXAMPLE OF MAPPING INTEGRATION DATA FIELDS TO SAP CONCUR FIELDS	107
CHAPTER 18. CONNECTING TO SERVICENOW	109
18.1. CREATING A SERVICENOW CONNECTION	109
18.2. OBTAINING RECORDS FROM SERVICENOW TO TRIGGER INTEGRATION EXECUTION	110
18.3. CREATING AN IMPORT SET IN SERVICENOW	111
18.4. COPYING RECORDS TO SERVICENOW DURING OR TO FINISH AN INTEGRATION	112
18.5. EXAMPLE SALESFORCE TO SERVICENOW INTEGRATIONS	113
CHAPTER 19. CONNECTING TO SLACK	115
19.1. CREATING A SLACK CONNECTION	115
19.2. ADDING A SLACK CONNECTION TO TRIGGER INTEGRATION EXECUTION UPON RECEIVING MESSAGES	116
19.3. ADDING A SLACK CONNECTION TO SEND A MESSAGE TO A SLACK CHANNEL OR USER	117
CHAPTER 20. CONNECTING TO SQL DATABASES	118
20.1. CREATING A DATABASE CONNECTION	118
20.2. STARTING AN INTEGRATION BY ACCESSING A DATABASE	119
20.3. ACCESSING A DATABASE IN THE MIDDLE OR TO COMPLETE AN INTEGRATION	120
20.4. CONNECTING TO PROPRIETARY DATABASES	121
CHAPTER 21. CONNECTING TO TELEGRAM	122
21.1. CREATING A TELEGRAM CONNECTION	122
21.2. ADDING A TELEGRAM CONNECTION TO TRIGGER INTEGRATION EXECUTION UPON RECEIVING MESSAGES	123
21.3. ADDING A TELEGRAM CONNECTION TO SEND A MESSAGE TO A TELEGRAM CHAT	123
CHAPTER 22. TRIGGERING INTEGRATION EXECUTION WITH A TIMER	125
22.1. ADDING A TIMER CONNECTION TO TRIGGER INTEGRATION EXECUTION	125
22.2. CREATING A REPLACEMENT TIMER CONNECTION	125
CHAPTER 23. CONNECTING TO TWITTER	127
23.1. REGISTERING FUSE ONLINE AS A TWITTER CLIENT APPLICATION	127
23.2. CREATING A TWITTER CONNECTION	128
23.3. ADDING A TWITTER CONNECTION TO TRIGGER INTEGRATION EXECUTION	129
CHAPTER 24. TRIGGERING INTEGRATION EXECUTION WITH AN HTTP REQUEST (WEBHOOK)	131
24.1. GENERAL PROCEDURE FOR USING THE FUSE ONLINE WEBHOOK	131
24.2. USING THE WEBHOOK CONNECTION	132
24.3. CREATING AN INTEGRATION THAT AN HTTP REQUEST CAN TRIGGER	132
24.4. HOW FUSE ONLINE HANDLES HTTP REQUESTS	133
24.5. GUIDELINES FOR AN HTTP CLIENT THAT INVOKES A FUSE ONLINE WEBHOOK	134
24.6. ABOUT THE JSON SCHEMA FOR SPECIFYING REQUEST PARAMETERS	134
24.7. HOW TO SPECIFY HTTP REQUESTS	135

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, Connecting to Amazon S3](#)
- [Chapter 3, Connecting to AMQ](#)
- [Chapter 4, Connecting to AMQP](#)
- [Chapter 5, Connecting to Dropbox](#)
- [Chapter 6, Connecting to FHIR](#)
- [Chapter 7, Connecting to an FTP or SFTP server](#)
- [Chapter 8, Connecting to Google applications](#)
- [Chapter 9, Connecting to HTTP and HTTPS endpoints](#)
- [Chapter 10, Connecting to IRC](#)
- [Chapter 11, Connecting to Kafka](#)
- [Chapter 12, Connecting to Apache Kudu](#)
- [Chapter 13, Connecting to MQTT](#)
- [Chapter 14, Connecting to OData](#)
- [Chapter 15, Connecting to REST APIs](#)
- [Chapter 16, Connecting to Salesforce](#)
- [Chapter 17, Connecting to SAP Concur](#)
- [Chapter 18, Connecting to ServiceNow](#)
- [Chapter 19, Connecting to Slack](#)
- [Chapter 20, Connecting to SQL databases](#)
- [Chapter 21, Connecting to Telegram](#)
- [Chapter 22, Triggering integration execution with a timer](#)
- [Chapter 23, Connecting to Twitter](#)
- [Chapter 24, Triggering integration execution with an HTTP request \(Webhook\)](#)

CHAPTER 1. CONNECTORS THAT ARE SUPPORTED BY FUSE ONLINE

Fuse Online supports the following connectors.

Name	Description
Amazon S3	Retrieve data from an Amazon S3 bucket or copy data into a bucket.
AMQ	Obtain messages from a Red Hat AMQ (ApacheMQ) broker or publish messages to a Red Hat AMQ (ApacheMQ) broker.
AMQP	Obtain messages from an Advanced Message Queue Protocol broker or publish messages to an AMQP broker.
Dropbox	Download files from Dropbox or upload files to Dropbox.
FHIR	Obtain resources from a FHIR server or update resources on a FHIR server.
FTP/SFTP	Download files from an FTP or SFTP server or upload files to an FTP or SFTP server.
Gmail	Obtain messages sent to a particular Gmail account and send messages from a particular Gmail account.
Google Calendar	Obtain events from a Google Calendar that you specify or add/update events in a Google Calendar that you specify.
Google Sheets	Obtain data from a Google Sheets spreadsheet that you specify, add/update spreadsheet data, create charts, or create pivot tables in a spreadsheet that you specify.
HTTP/HTTPS	Connect to an HTTP or HTTPS endpoint and execute the GET, PUT, POST, DELETE, HEAD, OPTIONS, TRACE, OR PATCH method.
IRC	Receive messages that are sent to an IRC nickname or channel, or send messages to an IRC nickname on a particular channel.
Kafka	Obtain streams of records from or publish streams of records to a Kafka topic that you specify.
Kudu	Obtain records from a table in an Apache Kudu data store or add records to a table in a Kudu data store.
MQTT	Obtain messages from an MQ Telemetry Transport broker or publish messages to an MQTT broker.
OData	Obtain entities from an OData service or update, create, or delete entities that are managed by an OData service.

Name	Description
REST APIs	<p>Create a custom REST API client connector by uploading an OpenAPI document. You can then create a connection to that REST API.</p> <p>Create a REST API provider integration by uploading an OpenAPI document that defines operations that trigger integration execution. See Creating an integration that is triggered by a REST API call in Integrating Applications with Fuse Online.</p>
Salesforce	Create, update, fetch, or delete a Salesforce record.
SAP Concur	Perform any one of a large variety of SAP Concur actions.
ServiceNow	Obtain records from or copy records to your ServiceNow instance.
Slack	Obtain messages from a channel or send a message to a Slack channel or user.
SQL databases	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.
Telegram	Obtain messages from a chat or send messages to a chat by using a Telegram chat bot.
Timer	Set a simple timer or a cron timer to trigger execution of a simple integration.
Twitter	Trigger execution of a simple integration upon tweets that mention you or that contain data that you specify.
Webhook	Trigger execution of simple integrations with HTTP GET or POST 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, how to use Fuse tooling to develop extensions](#)
- [Integrating Applications with Fuse Online, how to code extensions](#)

CHAPTER 2. CONNECTING 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 flow. For details, see:

- [Section 2.1, “Prerequisites for creating an Amazon S3 connection”](#)
- [Section 2.2, “Creating Amazon S3 connections”](#)
- [Section 2.3, “Obtaining data from Amazon S3 to trigger integration execution”](#)
- [Section 2.4, “Adding data to Amazon S3 to finish an integration”](#)
- [Section 2.5, “Adding 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. CREATING 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 an Amazon S3 connection](#) .

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. OBTAINING DATA FROM AMAZON S3 TO TRIGGER INTEGRATION EXECUTION

To start an integration by obtaining data from an Amazon S3 bucket, add an Amazon S3 connection as a simple integration's start connection.

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 at the beginning of the integration flow.

2.4. ADDING DATA TO AMAZON S3 TO FINISH AN INTEGRATION

To finish an integration by copying data to Amazon S3, add an Amazon S3 connection as a simple integration's finish connection.

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:

- **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.

- **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 at the end of the integration flow.

2.5. ADDING 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 to the middle of a flow.

Prerequisite

- You created an Amazon S3 connection.
- You are creating or editing a flow and Fuse Online is prompting you to choose a step.

Procedure

1. Click the Amazon S3 connection that you want to use in the middle of a flow.

2. Click the action that you want the connection to perform:
 - **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.
 - **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.
3. After you configure the chosen action, click **Next** to specify the action's input type.
4. 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**.
5. 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**.
6. 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.
7. 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.
8. Click **Done**.

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

CHAPTER 3. CONNECTING TO AMQ

In an integration, you can obtain messages from a Red Hat AMQ broker or publish messages to a Red Hat AMQ broker. Red Hat AMQ uses the OpenWire protocol for communication between clients and message brokers. To communicate with the following broker types, use the Red Hat 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 Red Hat AMQ connector, see:

- [Section 3.1, "Creating an AMQ connection"](#)
- [Section 3.2, "Adding an AMQ connection to trigger integration execution upon receiving messages"](#)
- [Section 3.3, "Publishing AMQ messages to finish an integration"](#)
- [Section 3.4, "Publishing AMQ messages in the middle of an integration"](#)

3.1. CREATING AN AMQ CONNECTION

In an integration, to obtain messages from or to publish messages to:

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

Create a Red Hat AMQ connection, which you can add to an integration.

Prerequisites

For the Red Hat 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 **Red Hat AMQ** 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 Red Hat AMQ broker's PEM certificate text. This is required except when you disable checking the certificates.
 - g. In the **Client Certificate** field, paste the Red Hat 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 **Red Hat AMQ 1**.
8. In the **Description** field, optionally enter any information that is helpful to know about this connection. For example, enter **Sample Red Hat 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 **Red Hat AMQ 1** appears as a connection that you can choose to add to an integration.

3.2. ADDING AN AMQ CONNECTION TO TRIGGER INTEGRATION EXECUTION UPON RECEIVING MESSAGES

To trigger execution of an integration upon receiving a message from a Red Hat AMQ broker, add a Red Hat AMQ connection as the start connection.

Prerequisite

You created a Red Hat AMQ connection to the Red Hat AMQ 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 Red Hat 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 that 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, if you are specifying a JSON schema for vendors then you might 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 at the beginning of the integration flow.

3.3. PUBLISHING AMQ MESSAGES TO FINISH AN INTEGRATION

To finish a simple integration by publishing messages to a Red Hat AMQ broker, add a Red Hat AMQ connection as the simple integration's finish connection.

Prerequisites

- You created a connection to the Red Hat 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 Red Hat 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 that 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, if you are specifying a JSON schema for vendors then you might 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 at the end of the integration flow.

3.4. PUBLISHING AMQ MESSAGES IN THE MIDDLE OF AN INTEGRATION

In the middle of an integration, to publish messages to a Red Hat AMQ broker, add a Red Hat AMQ connection to the middle of a flow.

Prerequisites

- You created a connection to the Red Hat AMQ broker that you want to publish messages to.
- You are creating or editing a flow and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to choose a finish connection.

Procedure

1. Click the Red Hat AMQ connection that you want in the middle of the flow.
2. On the **Choose an Action** page, select one of the following actions:
 - **Publish messages** action to publish messages to the queue or topic that 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 that 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).
3. Click **Next** to specify the action's input type and then the action's output type.
 4. 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**.
 5. 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**.
 6. In the **Data Type Name** field, enter a name that you choose for the data type. For example, if you are specifying a JSON schema for vendors then you might 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.
 7. 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.
 8. Click **Done**.

Result

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

CHAPTER 4. CONNECTING 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 Red Hat 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, "Creating an AMQP connection"](#)
- [Section 4.2, "Adding an AMQP connection to trigger integration execution upon receiving messages"](#)
- [Section 4.3, "Publishing messages to AMQP in the middle of a flow or to finish an integration"](#)

4.1. CREATING 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. ADDING AN AMQP CONNECTION TO TRIGGER INTEGRATION EXECUTION UPON RECEIVING MESSAGES

To trigger execution of an integration upon receiving messages from an AMQP broker, add an AMQP connection as the integration's 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 that 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, if you are specifying a JSON schema for vendors then you might 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 at the beginning of the integration flow.

4.3. PUBLISHING MESSAGES TO AMQP IN THE MIDDLE OF A FLOW OR TO FINISH AN INTEGRATION

You can publish messages to an AMQP broker in the middle of a flow or to finish a simple integration. To do this, add an AMQP connection to the middle of a flow or as a simple integration's finish connection.

Prerequisites

- You created a connection to the AMQP broker that you want to publish messages to.
- You are creating or editing a flow and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to choose a simple integration's finish connection.

Procedure

1. Click the AMQP connection that you want to use to publish messages.
2. On the **Choose an Action** page, select one of the following actions:
 - **Publish messages** to publish messages to the queue or topic that you specify without receiving 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. Select **Persistent** to guarantee message delivery even if a connection fails.
 - **Request response using messages** to publish messages to the queue or topic that you specify and receive a response.



NOTE

When the finish connection in a simple integration is an AMQP connection that performs the **Request response using messages** action, the connection publishes the messages but the response is discarded. To avoid losing the response, add an AMQP connection as a middle connection that performs the **Request response using messages** action and finish the simple integration with a log step.

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 **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 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.

3. Click **Next** to specify the action's input and output type.
4. 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**.
5. 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**.
 6. In the **Data Type Name** field, enter a name that you choose for the data type. For example, if you are specifying a JSON schema for vendors then you might 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.
 7. 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.
 8. Click **Done**.

Result

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

CHAPTER 5. CONNECTING 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, "Registering Fuse Online as a Dropbox client"](#)
- [Section 5.2, "Creating a Dropbox connection"](#)
- [Section 5.3, "Obtaining files from Dropbox to trigger integration execution"](#)
- [Section 5.4, "Adding files to Dropbox to finish an integration"](#)
- [Section 5.5, "Accessing Dropbox in the middle of an integration"](#)

5.1. REGISTERING 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. CREATING 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. OBTAINING FILES FROM DROPBOX TO TRIGGER INTEGRATION EXECUTION

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 at the beginning of the integration flow.

5.4. ADDING FILES TO DROPBOX TO FINISH AN INTEGRATION

To finish an integration by uploading files to Dropbox, add a Dropbox connection as the integration's 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 at the end of the integration flow.

5.5. ACCESSING DROPBOX IN THE MIDDLE OF AN INTEGRATION

To upload a file to Dropbox in the middle of a flow, add a Dropbox connection to the middle of the flow.

Prerequisites

- You created a Dropbox connection.
- You are creating or editing a flow and Fuse Online is prompting you to choose a step.

Procedure

1. Click the Dropbox connection that you want the integration to use.

2. 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 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.
 - 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.
3. Click **Next** to specify the action's input and output type.
4. 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**.
5. 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**.
6. 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.

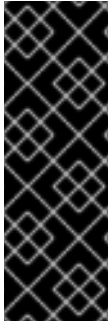
You will see this data type name when you are creating or editing an integration that uses the connection for which you are specifying this type. Fuse Online displays the type name in the integration visualization panel and in the data mapper.
7. In the **Data Type Description** field, provide information that helps you distinguish this type. This description appears in the data mapper when you hover over the step that processes this type.
8. Click **Done**.

Result

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

CHAPTER 6. CONNECTING TO FHIR

Fast Healthcare Interoperability Resources (FHIR) is a standard for exchanging healthcare data. In an integration, you can obtain one or more FHIR resources, create or update a resource, update one or more fields in a resource, or use a transaction to create multiple resources. To do this, add a FHIR connection to the middle of a flow or as a simple integration's finish connection.



IMPORTANT

The FHIR connector is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process. For more information about the support scope of Red Hat Technology Preview features, see <https://access.redhat.com/support/offerings/techpreview/>.

Fuse Online connections to FHIR:

- Can communicate with FHIR servers that use basic or bearer token (OAuth) authentication.
- Can operate on all FHIR resources that comply with FHIR DSTU3.
- Cannot operate on extensions, which allow for storing custom data in a resource. If an extension is in data that is received from a FHIR server, the integration silently ignores it.

In an integration that connects to FHIR, in a data mapper step, you can map only one level of depth for a FHIR list field. In other words, if a FHIR list contains a list then you cannot map the fields in the nested list. To mitigate this limitation, Fuse Online converts some FHIR resource list fields to single-value fields, for example, an address that is in a list field might be displayed in the data mapper as a single field.

Details for connecting to FHIR are in the following topics:

- [Section 6.1, "Creating a connection to a FHIR server"](#)
- [Section 6.2, "Obtaining a resource from a FHIR server"](#)
- [Section 6.3, "Querying a FHIR server for resources"](#)
- [Section 6.4, "Creating a resource on a FHIR server"](#)
- [Section 6.5, "Updating all fields in a resource on a FHIR server"](#)
- [Section 6.6, "Updating specified fields in a resource on a FHIR server"](#)
- [Section 6.7, "Creating resources of different types on a FHIR server"](#)
- [Section 6.8, "Deleting a resource from a FHIR server"](#)

6.1. CREATING A CONNECTION TO A FHIR SERVER

In an integration, to operate on FHIR resources, create a connection to a FHIR server and then add that connection to the middle of a flow or as a simple integration's finish connection.

A FHIR connection cannot be the start connection of a simple integration. However, you can start a simple integration with a timer that periodically triggers a FHIR connection.

Prerequisites

- You must know the URL for the FHIR server that you want to connect to.
- You must have authorization credentials for accessing the FHIR server. You should obtain these from the FHIR server administrator. Rarely, a FHIR server does not require authentication and you can create a connection without specifying credentials. For example, a public FHIR server or a FHIR server in a private network might not require authentication.

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 **FHIR** connector and configure the connection as follows:
 - a. In the **FHIR version** field, accept **DSTU3**, which is required.
 - b. In the **FHIR server URL** field, enter the URL for the FHIR server that you want to access. While creating an integration, you might accept the default test server **<http://fhirtest.uhn.ca/baseDstu3>**.
 - c. In the **FHIR server username** field, enter your username. This is required by FHIR servers that use basic authentication.
 - d. In the **FHIR server password** field, enter your password, which is also required by FHIR servers that use basic authentication.
 - e. In the **FHIR server bearer token** field, enter your token. This is required by FHIR servers that use OAuth authentication.
4. 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.
5. If validation is successful, 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, you might enter **FHIR West**.
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 that the connection you created is now available. If you entered the example name, you would see that **FHIR West** appears as a connection that you can choose to add to an integration.

6.2. OBTAINING A RESOURCE FROM A FHIR SERVER

In an integration, you can obtain one resource that is of the type that you specify. To do this, add a FHIR connection to the middle of a flow.

Prerequisites

- You created a connection to the FHIR server that has the resource that you want to obtain.
- You are creating or editing a flow and Fuse Online is prompting you to choose a step.

Procedure

1. Click the FHIR connection that can access the resource that you want to obtain.
2. On the **Choose an Action** page, click **Read**.
3. Click in the **Resource Type** field to display a list of FHIR resource types. Select or enter the type of resource that you want to obtain.
4. In the **Contained Resource Types** field, if the resource type that you want contains other resource types, select them. Use the **Ctrl** key if you need to select more than one. Identifying contained resource types lets the data mapper display the fields that are in the contained resources. If you do not specify a contained resource type then you cannot map from any fields that are in the contained resource. There is no harm in selecting a resource that is not actually contained.
5. In the **Resource Id** field, enter the ID of the resource that you want. Or, leave this field blank if you plan to map it from an earlier step that is in the flow.
6. In the **Resource version** field, optionally specify a version ID for the resource. Leave this field blank to obtain the most recent version of the resource or to map the version ID of the resource that you want to obtain from a previous step.
7. Click **Next** to add this connection to the flow.

Result

The connection appears in the integration flow in the location where you added it. During execution, this connection returns one resource.

Next steps

To map a value to the **Resource Id** field or the **Resource version** field, add a data mapper step before this connection.

6.3. QUERYING A FHIR SERVER FOR RESOURCES

In an integration, you can query a FHIR server to obtain instances of a particular FHIR resource that satisfy a query that you specify. For example, suppose you want to ensure that all patients who are 65 and older received a pneumonia vaccination. Specify a query that obtains the resources for patients who were born before 1955. To do this, add a FHIR connection to the middle of a flow.

Prerequisites

- You created a connection to the FHIR server that has the resources that you want to obtain.
- You are creating or editing a flow and Fuse Online is prompting you to choose a step.

Procedure

1. Click the FHIR connection that has the resources that you want to obtain.

2. On the **Choose an Action** page, click **Search**.
3. Click in the **Resource Type** field to display a list of FHIR resource types. Select or enter the type of resource that you want to obtain.
4. In the **Query** field, specify a FHIR query or leave this field blank to map the query from a previous step. For details about forming a FHIR query, see [FHIR Release 3 Search](#).
5. Click **Next** to add this connection to the flow.

Result

The connection appears in the integration flow in the location where you added it. During execution, this connection returns a list of resources that satisfy the query that you specified. Fuse Online treats this list as a collection.

Next steps

If you did not specify a query, then add a data mapper step before this connection and map the query to this connection. To operate on individual returned resources, rather than on the returned list collection, after this connection, add a split step to the flow.

6.4. CREATING A RESOURCE ON A FHIR SERVER

In an integration, you can create and add a resource to a FHIR server. When you configure the connection's **Create** action, you specify the type of the resource that you want to create. For example, you might add new patient resource to a FHIR server. To create a resource, add a FHIR connection to the middle of a flow, or as a simple integration's finish connection.

To populate the values in the new resource, add a data mapper step to the flow just before the FHIR connection that you are adding in this procedure. It is expected that previous steps in the flow provide the data that you want the new resource to contain. Map fields in previous steps to fields in the target resource that this FHIR connection is creating. The new resource contains only the fields that you map to.

When a connection performs the **Create** action to create a new resource on a FHIR server, the new resource has an automatically-generated FHIR resource ID. If you want to specify the resource ID for a new resource, instead of selecting the **Create** action choose the **Update action** for the connection to perform.

Prerequisites

- You created a connection to the FHIR server that you want to add a resource to.
- You are creating or editing a flow and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to add a finish connection.

Procedure

1. Click the FHIR connection to the server that you want to add a resource to.
2. On the **Choose an Action** page, click **Create**.
3. Click in the **Resource Type** field to display a list of FHIR resource types. Select or enter the type of resource that you want to create.

4. In the **Contained Resource Types** field, if the resource type that you want to create contains other resource types, select them. Use the **Ctrl** key if you need to select more than one. Identifying contained resource types lets the data mapper display the fields that are in the contained resources. If you do not specify a contained resource type then you cannot map to or from any fields that are in the contained resource. There is no harm in selecting a resource that is not actually contained.
5. Click **Next** to add this connection to the flow.

Result

The connection appears in the integration flow in the location where you added it. During execution, the connection creates a new resource on the FHIR server. The connection returns a **MethodOutcome** resource that includes the new resource's resource ID, which you might want to map to a subsequent step in the flow.

Next steps

If this flow needs additional steps to obtain the data that you want the new resource to contain, add them before this FHIR connection. After those steps are in place, add a data mapper step to the flow just before the FHIR connection that creates a new resource. In the data mapper step, map to the fields in the resource to be created.

CAUTION

Without a data mapper step just before a FHIR connection that creates a new resource, the connection tries to create an empty resource. Depending on the FHIR server configuration, this might or might not work.

6.5. UPDATING ALL FIELDS IN A RESOURCE ON A FHIR SERVER

In an integration, you can update a resource that is on a FHIR server. To provide the updated values, add a data mapper step to the flow just before a FHIR connection that updates a resource. It is expected that previous steps in the flow provide the data that you want the updated resource to contain. Map fields in previous steps to fields in the target resource that this FHIR connection updates.

CAUTION

The updated resource contains only the fields that you map to. In other words, in addition to mapping the fields whose values are changing, you must map the fields that you want to be in the resource and whose values are not changing. If you do not map a particular resource field then the connection deletes that field from the updated resource.

Be sure to map the resource ID from a previous step to the resource ID in this connection. This is the only resource field that the connection does not change. If the FHIR server does not already have a resource that has the resource ID for the resource to be updated, then the connection creates a new resource with that resource ID. This is the only way to add a new resource that has a resource ID that you choose.

To update or create a resource, add a FHIR connection to the middle of a flow, or as a simple integration's finish connection.

Prerequisites

- You created a connection to the FHIR server on which you want to update or create a resource.

- You are creating or editing a flow and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to add a finish connection.

Procedure

1. Click the FHIR connection that you want to use to update a resource on a FHIR server.
2. On the **Choose an Action** page, click **Update**.
3. Click in the **Resource Type** field to display a list of FHIR resource types. Select or enter the type of resource that you want to update or create.
4. In the **Contained Resource Types** field, if the resource type that you want to update or create contains other resource types, select them. Use the **Ctrl** key if you need to select more than one.
Identifying contained resource types lets the data mapper display the fields that are in the contained resources. If you do not specify a contained resource type then you cannot map to or from any fields that are in the contained resource. There is no harm in selecting a resource that is not actually contained.
5. Click **Next** to add this connection to the flow.
6. If this flow needs additional steps to obtain the data that you want the updated resource to contain, add them before this FHIR connection.
7. In the integration visualization panel, click the plus sign that is before the FHIR connection that updates a resource.
8. Click **Data Mapper**.
9. In the data mapper:
 - a. Map a resource ID to the target resource ID. This is the only way to specify the resource ID for a new resource.
 - b. Map to each resource field that you want the updated or new resource to contain. Be sure to map fields whose values are not changing, as well as fields whose values need to be updated.
 - c. Click **Done** to add the data mapper step to the flow.

Result

The connection appears in the integration flow in the location where you added it. During execution, the connection updates or creates a resource on the FHIR server and returns a **MethodOutcome** resource that includes the updated/created resource's resource ID, which you might want to map to a subsequent step in the flow.

6.6. UPDATING SPECIFIED FIELDS IN A RESOURCE ON A FHIR SERVER

In an integration, you can update individual fields in a resource that is on a FHIR server. To do this, add a FHIR connection to the middle of a flow or as a simple integration's finish connection.

Prerequisites

- You created a connection to the FHIR server that has the resource that you want to update.
- You are creating or editing a flow and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to choose a finish connection.

Procedure

1. Click the FHIR connection that you want to use to update the resource.
2. On the **Choose an Action** page, click **Patch**.
3. Click in the **Resource Type** field to display a list of FHIR resource types. Select or enter the type of the resource whose fields you want to update.
4. In the **Number of Operations** field, specify the number of fields that you want to update.
5. Leave the **Resource id** field blank if you want to map the value from a previous step in the flow, which is typically what you want to do. Otherwise, specify the resource ID of the resource whose fields you want to update.
6. Specify a JSON patch that describes the updates. The JSON patch must have the format described in [What is a JSON Patch?](#). For some updates, you must specify the JSON patch when you configure the **Patch** action. For example, if you want to copy the value of a resource field to another field. For other updates, you can leave the **JSON Patch** field blank and instead map values that define each update.
7. Click **Done** to add this connection to the flow.
8. If you did not specify a JSON patch and/or a resource ID, then add a data mapper step:
 - a. In the flow visualization panel, click the plus sign that is before the FHIR connection that you just added.
 - b. On the **Choose a Step** page, click **Data Mapper**.
The **Target** panel displays a numbered folder for each field that you want to update. When you configured the **Patch** action, you specified how many fields to update. The data mapper displays this number of folders in the **Target** panel. For example, if you specified 3 as the number of fields to update, you would see three target folders with the labels **1**, **2**, and **3**.
 - c. For each field that you want to update, map to the fields in one target folder:
 - i. In the **Target** panel, expand a folder to display three fields for **op**, **path**, and **value**.
 - ii. Map a source field, constant, or property to the target **path** field. The path identifies the resource field that you want to update. In a path value, a number indicates the index of a list field, and a slash leads to a child field. For example, map this path **/name/1/given/1/value** to update the value of the given name field.
 - iii. Accept the default update operation, which replaces the current value of the field, or map a source field, constant, or property to the **op** field to indicate how you want to update the field. For details about possible operations, see [JSON patch operations](#).
 - iv. If you are updating a field to have a new value, then map a source field, constant, or property to the target **value**. This is the new value that you want the field to contain.
 - d. If you did not specify a resource ID when you configured the **Patch** action, then map the resource ID to the target **id** field.

- e. Click **Done** to add the data mapper step to the flow.

Result

The connection appears in the integration flow in the location where you added it. During execution, the connection updates the specified resource field(s) and returns a **MethodOutcome** resource that contains the ID of the resource that was updated.

6.7. CREATING RESOURCES OF DIFFERENT TYPES ON A FHIR SERVER

In an integration, you can create two or more resources on a FHIR server and each resource can be of a different type. For example, you can add a new patient resource and a new provider resource in one connection to a FHIR server. To do this, add a FHIR connection to the middle of a flow, or as a simple integration's finish connection.

Prerequisites

- You created a connection to the FHIR server that you want to add resources to.
- You are creating or editing a flow and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to choose a finish connection.

Procedure

1. Click the FHIR connection that you want to use to create and add new resources to a FHIR server.
2. On the **Choose an Action** page, click **Transaction**.
3. In the **Included Resource Types** field, select the resource types that this connection creates. Use the **Ctrl** key to select multiple resource types. The connection creates one resource of each type that you select.
4. Click **Next** to add this connection to the flow.
5. To populate fields in the new resources, add a data mapper step before this connection:
 - a. In the flow visualization on the left, click the plus sign that is before the connection that you just added.
 - b. Click **Data Mapper**.
 - c. In the data mapper, map source fields to the target resource fields. The new resources contain only the target fields that you map to.
 - d. Click **Done** to add the data mapper step to the flow.

Result

The connection appears in the integration flow in the location where you added it. During execution, the connection adds all resources to the FHIR server or, if there is an error, the connection does not add any resources to the FHIR server. A successful action returns a transaction resource, which contains the resource ID for each new resource. You can map these resource IDs to subsequent steps in the flow.

6.8. DELETING A RESOURCE FROM A FHIR SERVER

In an integration, you can delete a resource from a FHIR server. To do this, add a FHIR connection to the middle of a flow or as a simple integration's finish connection.

Prerequisites

- You created a connection to the FHIR server that has the resource that you want to delete.
- You are creating or editing a flow and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to select a finish connection.

Procedure

1. Click the FHIR connection that you want to use to delete the resource.
2. On the **Choose an Action** page, click **Delete**.
3. Click in the **Resource Type** field to display a list of FHIR resource types. Select or enter the type of resource that you want to delete.
4. In the **Resource Id** field, specify the resource ID of the resource that you want to delete or leave this field blank if you plan to map the resource ID from an earlier step in the integration.
5. In the **Resource version** field, optionally specify a version ID for the resource. Leave this field blank to delete the most recent version of the resource or to map the version ID of the resource that you want to delete from a previous step.
6. Click **Done** to add this connection to the flow. .

Result

The connection appears in the integration flow in the location where you added it. During execution, the connection deletes the specified resource on the FHIR server and returns a **MethodOutcome** resource, which includes the resource ID for the deleted resource.

Next steps

To map a value to the **Resource Id** field or the **Resource version** field, add a data mapper step before this connection.

CHAPTER 7. CONNECTING 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 and then add it to an integration flow. The following topics provide details:

- [Section 7.1, "Creating an FTP or SFTP connection"](#)
- [Section 7.2, "Obtaining files from an FTP or SFTP server"](#)
- [Section 7.3, "Uploading files to an FTP or SFTP server"](#)

7.1. CREATING 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 used** 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.

7.2. OBTAINING 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 at the beginning of the integration flow.

7.3. UPLOADING FILES TO AN FTP OR SFTP SERVER

In an integration, you can upload files to an FTP or SFTP server in the middle of a flow or to finish a simple integration. To do this, add an FTP or SFTP connection to the middle of a flow or as a simple integration's finish connection.

Prerequisite

- You created an FTP or SFTP connection.
- You are creating or editing a flow and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to choose a finish connection.

Procedure

1. Click the FTP or SFTP connection that you want to use to upload files.
2. On the **Choose an Action** page, click **Upload**.
3. 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.
4. 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 step.
5. 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.
6. 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.

7. 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.
8. Click **Next** to specify the action's input type.
9. 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**.
10. 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**.
11. 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.
12. 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.
13. Click **Done**.

Result

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

CHAPTER 8. CONNECTING TO GOOGLE APPLICATIONS

An integration can connect to these Google applications: Gmail, Calendar, Sheets. See the following topics for details:

- [Section 8.1, “Registering Fuse Online as a Google client application”](#)
- [Section 8.2, “Connecting to Gmail”](#)
- [Section 8.3, “Connecting to Google Calendar”](#)
- [Section 8.4, “Connecting to Google Sheets”](#)

8.1. REGISTERING FUSE ONLINE AS A GOOGLE CLIENT APPLICATION

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

With registration in place, you can create multiple connections to Gmail, Google Calendar, and Google Sheets. You can use each connection in multiple integrations. While each connection to a Google application can use the same Google client ID and Google client secret, which you obtain during registration, each connection can access a different Google account, which you choose.

This procedure instructs you to enable the Gmail API, the Google Calendar API, and the Google Sheets API. However, you can enable only the API(s) that you need rather than all three APIs. At a later time, if you need to create connections to the applications whose APIs you did not enable, you can return here and follow the instructions to enable their APIs.



IMPORTANT


You must create a new Google client application for Fuse Online. The credentials that Google provides for a new client application contain a refresh token that is used for refreshing expired access tokens. This refresh token is available only the first time that the Fuse Online client application uses the credentials. In Fuse Online, connections to Gmail, Google Calendar, and Google Sheets can all use the same Google client ID and Google client secret. If they do, the refresh token is available to all connections to Google applications. When you view connection details in the Fuse Online user interface, do not click the **Validate** button. Validation is a second use of the credentials and the refresh token is no longer part of the client credentials. You can, however, re-connect to Google applications.



In development environments, be careful if you choose to use the Google client ID and Google client secret that you are using for some other, non-Fuse Online, OAuth client. Fuse Online requires offline access that is requested on the first OAuth exchange. If another OAuth client already entered the OAuth exchange and did not request offline access, then Fuse Online cannot obtain offline access on subsequent OAuth exchanges. If you are unsure whether offline access was requested on the first exchange, create a new Google client application for Fuse Online.



Prerequisites

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

Procedure

1. In Fuse Online:
 - 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. Check that you are signed into 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.

The name of the current Google project appears at the top of the Google APIs page.
 - b. To use the current project to grant authorization to Fuse Online, continue to the next step. To grant authorization to Fuse Online in another Google project, choose or create that project. If this Google account does not already have a project, you must create one.
 - c. Enable Google APIs:
 - i. In the upper left corner, click  and select **APIs and Services > Library**.
 - ii. If necessary, scroll down to see the **G Suite** row of cards.
 - iii. Click the **Google Calendar API** card, which displays a page that indicates that the Google Calendar API is enabled.
 - iv. Scroll down and click the **Gmail API** card, which displays a page that indicates that the Gmail API is enabled.
 - v. In the upper left corner, click  and select **APIs and Services > Library**.
 - vi. If necessary, scroll down to see the **G Suite** row of cards.
 - vii. Click the **Google Sheets API** card, which displays a page that indicates that the Google Sheets API is enabled.
 - d. Give your client application a name:
 - i. Click **MANAGE**.
 - ii. In the page that appears, in the left navigation panel, click **Credentials** and then on the right, click **Credentials in APIs & Services**
 - iii. Click the **OAuth consent screen** tab.
 - iv. 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**.
 - v. Skip the other fields.
 - vi. 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 and do the following:
 - a. Expand the entries for Gmail, Google Calendar, and Google Sheets.
 - b. In the **Client ID** field for each Google application, paste the Google 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 in the entry for each Google application:
 - a. In the **Client Secret** field, paste the Google client secret that you just copied.
 - b. Click **Save**. You should get a **Registration Successful!** notification.
 - c. Click **Ok** to collapse the entry.

Results

For each Google application API that you enabled, you can create a connection to that application.

All connections from Fuse Online to Google applications use the same Google client ID and Google client secret.



IMPORTANT

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

8.2. CONNECTING TO GMAIL

To trigger execution of an integration when a particular Gmail account receives an email, add a Gmail connection to a simple integration as its start connection. In an integration, to send an email from a particular Gmail account, do either of the following:

- Add a Gmail connection to the middle of a flow.
- Add a Gmail connection to finish a simple integration.

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

1. [Registering Fuse Online as a Google client application](#) .
2. Creating 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. Adding a Gmail connection to an integration flow.
5. For a Gmail connection that sends an email, optionally mapping integration data to the email fields.

Information and instructions are in the following topics:

- [Section 8.2.1, "Creating a Gmail connection"](#)
- [Section 8.2.2, "Alternatives for populating email to send"](#)
- [Section 8.2.3, "Triggering an integration when polling returns a Gmail message"](#)
- [Section 8.2.4, "Sending an email from a Gmail account"](#)

8.2.1. Creating 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.

Prerequisites

- You [registered Fuse Online as a Google client application](#) and enabled the Gmail API.
- The Fuse Online **Settings** page entry for Gmail has values for the client ID and client secret, which you obtained by registering Fuse Online as a Google 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 display, your Fuse Online environment is not registered as a Google

client application. See [Registering Fuse Online as a Google client application](#) . When you try to create a Gmail connection and your Fuse Online environment is not registered as a Google 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, it is not recommended.

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.

8.2.2. Alternatives for populating email to send

A Gmail connection that finishes a simple integration or that is in the middle of a flow 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 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 a flow, 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** as the email address.

When you add a Gmail connection that sends an email, all action configuration parameters are optional. 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 action 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.

8.2.3. Triggering 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 a simple 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**.

Prerequisites

- 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. Leave the **Labels** field 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.

8.2.4. Sending an email from a Gmail account

In an integration, you can send an email from a Gmail account either in the middle of a flow or to finish a simple integration. To do this, add a Gmail connection to the middle of a flow or as a simple 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.
- Fuse Online is prompting you to choose a step or to choose the finish connection for a simple integration.

Procedure

1. Click the Gmail connection that you want to use to send an email.
2. On the **Choose an Action** page, click **Send Email**.
3. 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 of the following **Send Email** fields. Separate multiple email addresses with a comma.
 - a. **Email To**
 - b. **Email Subject**
 - c. **Email Text**
 - d. **Email cc**
 - e. **Email bcc**
 - 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.

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

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.

8.3. CONNECTING TO GOOGLE CALENDAR

To trigger integration execution when a poll returns an update to a Google calendar, add a Google Calendar connection to a simple integration as its start connection. To add an event to a calendar or update an event in a calendar, you can do either of the following:

- Add a Google Calendar connection to the middle of a flow.
- Add a Google Calendar connection to finish a simple integration.

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

- [Section 8.3.1, "Creating a Google Calendar connection"](#)
- [Section 8.3.2, "Triggering an integration when polling returns an event from a Google Calendar"](#)
- [Section 8.3.3, "Obtaining a particular event from a Google Calendar"](#)
- [Section 8.3.5, "Updating an event in a Google Calendar"](#)
- [Section 8.3.4, "Adding an event to a Google Calendar"](#)

8.3.1. Creating 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.

Prerequisites

- You [registered Fuse Online as a Google client application](#) and enabled the Google Calendar API.
- The Fuse Online **Settings** page entry for Google Calendar has values for the client ID and client secret, which you obtained by registering Fuse Online as a Google 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.

If **Connect Google Calendar** does not display, then your Fuse Online environment is not registered as a Google client application with the Google Calendar API enabled. See [Registering Fuse Online as a Google 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, it is not recommended.

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.

8.3.2. Triggering 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 a simple 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.

Prerequisites

- 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 a 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 a 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.

8.3.3. Obtaining a particular event from a Google Calendar

In an integration, you can obtain a particular Google Calendar event in the middle of a flow. 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 to the middle of a flow.



NOTE


In this release, while obtaining a specific event in a simple 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 that you want to get.
- You are creating or editing a flow and Fuse Online is prompting you to choose a step.

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

8.3.4. Adding an event to a Google Calendar

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

Prerequisites

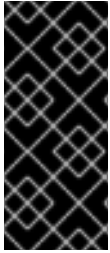
- A Google Calendar connection is available and this connection is authorized to access the Google Calendar to which you want to add an event.
- You are creating or editing a flow and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to choose a finish connection.

Procedure

1. Click a Google Calendar connection that is authorized to access the calendar that you want to add an event 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 flow. The connection appears in the integration flow in the location where you added it.

8.3.5. Updating an event in a Google Calendar

In an integration, you can update an event in a Google Calendar in the middle of a flow or to finish a simple integration. To do this, add a Google Calendar connection to the middle of a flow or as a simple 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 flow, there is an earlier connection to Google Calendar and that connection obtains the event that you want to update.
- You are creating or editing a flow and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to choose a finish connection.

Procedure

1. Click a Google Calendar connection that is authorized to access the calendar that has the event that you want to update.
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 unchanged.
4. Click **Done** to add the connection to the flow. The connection appears in the integration visualization flow in the location where you added it.
5. In the integration visualization flow, click the plus sign that is just before the connection that you just added.
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.

8.4. CONNECTING TO GOOGLE SHEETS

To trigger execution of an integration when a Google Sheets connection returns spreadsheet data or

spreadsheet properties, add a Google Sheets connection to a simple integration as its start connection. To finish a simple integration by updating spreadsheet values or properties, or by inserting a chart or pivot table into a spreadsheet, add a Google Sheets connection to a simple integration as its finish connection.

In the middle of a flow, you can add a Google Sheets connection that obtains, creates or updates values in a spreadsheet, or that inserts a chart or pivot table into a spreadsheet.

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

- [Section 8.4.1, "Creating a Google Sheets connection"](#)
- [Section 8.4.2, "Triggering an integration when polling returns spreadsheet data"](#)
- [Section 8.4.3, "Triggering an integration when polling returns spreadsheet properties"](#)
- [Section 8.4.4, "Creating a spreadsheet"](#)
- [Section 8.4.5, "Updating data in a sheet"](#)
- [Section 8.4.6, "Appending data to a sheet"](#)
- [Section 8.4.7, "Updating spreadsheet properties"](#)
- [Section 8.4.8, "Adding a chart to a sheet"](#)
- [Section 8.4.9, "Adding a pivot table to a sheet"](#)

8.4.1. Creating a Google Sheets connection

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

Prerequisites

- You [registered Fuse Online as a Google client application](#) and enabled the Google Sheets API.
- The Fuse Online **Settings** page entry for Google Sheets has values for the client ID and client secret, which you obtained by registering Fuse Online as a Google 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 Sheets** connector.
4. In the **Configure Connection** page, click **Connect Google Sheets**, which takes you to a Google sign-in page.

If **Connect Google Sheets** does not display, then your Fuse Online environment is not registered as a Google client application with the Google Sheets API enabled. See [Registering Fuse Online as a Google client application](#). When your environment is not registered with

Google, then when you try to create a Google Sheets connection, Fuse Online displays multiple fields that prompt for authorization information. While you can create a Google Sheets connection by entering values in these fields, it is not recommended.

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 Sheets Work Connection**.
8. In the **Description** field, optionally enter any information that is helpful to know about this connection. For example, enter **Sample Google Sheets connection that uses my Google work account**.
9. In the upper right, click **Create** to see that the connection that you created is now available. If you entered the example name, you would see that **Google Sheets Work Connection** appears as a connection that you can choose to add to an integration.

8.4.2. Triggering an integration when polling returns spreadsheet data

To trigger execution of an integration upon obtaining data from a Google Sheets spreadsheet, add a Google Sheets connection to a simple integration as its start connection. When the integration is running, the Google Sheets connection polls the spreadsheet at the interval that you specified, obtains the data that you identified, and passes the data to the next step in the integration.

To obtain data from a particular sheet in a spreadsheet, you specify the sheet name when you configure the **Get sheet values** action for the Google Sheets connection. A particular connection can obtain data from only one sheet.

Between polls, if there are no changes to the sheet values that the connection is configured to return, then the next poll returns the same values as the previous poll.

Prerequisites

- You created a Google Sheets connection that is authorized to access the spreadsheet that you want to obtain data 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 Google Sheets connection that you want to use to start the integration.
4. On the **Choose an Action** page, click the **Get sheet values** action.
5. To configure the **Get sheet values** action:
 - a. In the **SpreadsheetId** field, enter the ID of a Google spreadsheet that is accessible from the Google account that this Google Sheets connection is authorized to access.

To obtain a spreadsheet ID, display the spreadsheet in a browser. The ID is the part of the URL between **d/** and **/edit**.

- b. In the **Range** field, enter Google A1 notation that identifies the data that you want to obtain. The default is **A:A**.

For example, suppose that you want all data from a spreadsheet that has one sheet with 4 columns. Specify **A:D**. Or, to obtain data from rows 5 through 15, specify **A5:D15**.

If the spreadsheet from which you are obtaining data has more than one sheet, specify the sheet name as well as the start cell coordinate and the end cell coordinate. If there is more than one sheet, and you do not specify a sheet name, then the connection obtains data from the first sheet in the spreadsheet. For example, specification of **2019!A1:D5** specifies that you want to obtain data from the sheet whose name is **2019**. In that sheet, you want the data that is in columns A through D for rows 1 through 5.

- c. In the **Major Dimension** field, accept **Rows**, which is the default, or select **Columns**. **Rows** configures the action to return a collection of row objects where each row object contains a value for each desired column. **Columns** configures the action to return a collection of column objects where each column object contains a value for each desired row.
- d. In the **Split Results** field, accept **No**, which is the default, or select **Yes**. A setting of **No** configures the action to return data as a collection of values. That is, the connection passes a collection of row objects or a collection of column objects to the next step in the flow. Select **Yes** to enable the connection to split the returned data according to the setting of **Major Dimension**. For example, if **Major Dimension** is set to **Rows** then the connection returns row objects. Each row object triggers a separate execution of the flow. That is, Fuse Online executes the flow once for each returned row object. For example, if the poll returns 5 rows then Fuse Online executes the flow 5 times.
- Fuse Online also provides discrete split and aggregate steps, which you can add to a flow. If you want to process individual objects in one or more steps and then aggregate the row or column objects, do not split the Google Sheets connection result. Instead, accept the default, **No**, and then add a split step to the flow after this connection. A split step is required if you want an aggregate step in the flow.
- e. In the **Delay** field, accept the default of 30 seconds or specify how often you want the connection to obtain spreadsheet data.
- f. In the **Max Results** field, accept the default of **25** or specify another limit for the minor dimension in the result matrix.
- For example, suppose that the **Major Dimension** is columns and that **Max Results** is 25. The poll returns a column object for each column that the range setting includes. Each column object contains no more than 25 row values.

In this release, it is a known problem that the setting for **Max Results** is observed only when **Split Results** is set to **Yes**. When **Split Results** is set to **No**, the setting of **Max Results** is ignored. In the next release, observance of the **Max Results** setting will not depend on the **Split Results** setting.

6. Click **Done** to add this Google Sheets connection as the integration's start connection. In the integration visualization panel, the connection appears as the first step in the integration.

Next steps

After you add a start connection, Fuse Online prompts you to add the integration's finish connection. With the start and finish connections in the integration, add any other connections that you want in the integration. Then, after the connection that obtains sheet values, add a data mapper step. In the data mapper, Fuse Online displays source fields according to how you configure the **Get sheet values** action.

That is, if the major dimension is **Rows**, then the data mapper lists the column names as fields that you can map to the target. If the major dimension is **COLUMNS**, then the data mapper lists row indexes as fields that you can map to the target.

Additional resource

[Google A1 notation](#) for specifying groups of cells in a spreadsheet

8.4.3. Triggering an integration when polling returns spreadsheet properties

To trigger execution of an integration upon obtaining properties from a Google Sheets spreadsheet, add a Google Sheets connection to a simple integration as its start connection. When the integration is running, the Google Sheets connection polls the spreadsheet at the interval that you specified, obtains the spreadsheet properties, and passes the result to the next step in the integration.

Properties include the spreadsheet's title, locale, and time zone. Between polls, if there are no changes to the sheet properties, then the next poll returns the same values as the previous poll.

Prerequisites

- You created a Google Sheets connection that is authorized to access the spreadsheet that you want to obtain properties 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 Google Sheets connection that you want to use to start the integration.
4. On the **Choose an Action** page, click the **Get spreadsheet properties** action.
5. To configure the **Get spreadsheet properties** action:
 - a. In the **SpreadsheetId** field, enter the ID of a Google spreadsheet that is accessible from the Google account that this Google Sheets connection is authorized to access.
To obtain a spreadsheet ID, display the spreadsheet in a browser. The ID is the part of the URL between **d/** and **/edit**.
 - b. In the **Delay** field, accept the default of 30 seconds or specify how often you want the connection to obtain properties.
6. Click **Done** to add this Google Sheets connection as the integration's start connection. In the integration visualization panel, the connection appears as the first step in the integration's flow.

8.4.4. Creating a spreadsheet

To create a new spreadsheet in the middle of a flow, add a Google Sheets connection between the start and finish connections. While you can also finish a simple integration with a Google Sheets connection that creates a spreadsheet, you cannot add data to a new spreadsheet in the same connection. Therefore, when you want to create a spreadsheet and add data to the spreadsheet in the same flow, the flow requires two Google Sheets connections. One connection creates the spreadsheet and then a subsequent connection adds data to the spreadsheet.

Prerequisites

- You created a Google Sheets connection that is authorized to access the Google account in which you want to create the spreadsheet.
- You are creating or editing a flow and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to choose a finish connection.

Procedure

1. Click a Google Sheets connection that is authorized to access the Google account that you want to use to create the spreadsheet.
2. On the **Choose an Action** page, click **Create spreadsheet**
3. To configure the **Create spreadsheet** action:
 - a. In the **Title** field, enter the title of the new spreadsheet.
 - b. In the **Time Zone** field, enter the time zone that the new spreadsheet should use, for example, **India Standard Time**, or **Tokyo**.
 - c. In the **Locale** field, enter the locale of the new spreadsheet, for example, **Canada**, or **Hong Kong**.
4. Click **Done** to add the connection to the flow. The connection appears in the integration visualization flow in the location where you added it.

Result

A connection that creates a spreadsheet returns the spreadsheet ID for the new spreadsheet. In subsequent steps, you can map this spreadsheet ID to the spreadsheet ID in a connection that accesses the new spreadsheet, for example, to update it.

8.4.5. Updating data in a sheet

In an integration, you can update data in a spreadsheet in the middle of a flow or to finish a simple integration. To do this, add a Google Sheets connection to the middle of a flow or as a simple integration's finish connection.

Prerequisites

- You created a Google Sheets connection that is authorized to access the Google spreadsheet that you want to update.
- You have access to the spreadsheet ID for the spreadsheet that you want to update.
- You are creating or editing a flow and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to choose a finish connection.

Procedure

1. Click a Google Sheets connection that is authorized to access the spreadsheet that you want to update.
2. On the **Choose an Action** page, click **Update sheet values**.

3. To configure the **Update sheet values** action:
 - a. In the **SpreadsheetId** field, do one of the following:
 - Enter the ID of the spreadsheet that you want to update.
 - Leave the field blank. In this case, after you add this connection to the flow, you must add a data mapper step before this connection. In the data mapper step, map the spreadsheet ID from a previous connection to this connection. The previous connection must access the spreadsheet that you want to update in this connection.
 - b. In the **Range** field, use Google A1 notation to specify the spreadsheet cells that you want to update. Specify the start and end coordinates, for example, **A1:D4** updates the first 4 columns in the first 4 rows in the first sheet in the spreadsheet.
The default is **A:A**, which updates the first column in the first sheet in the spreadsheet.
 - c. In the **Major Dimension** field, accept **Rows**, which is the default, or select **Columns**. **Rows** configures the action to use row objects to update the sheet. Each row object contains a value for each column that you want to update. **Columns** configures the action to use column objects to update the sheet. Each column object contains a value for each row that you want to update.
 - d. In the **Value Input Option** field, indicate how you want Google sheets to interpret the data that it receives for updating the sheet. **Unspecified**, which is the default, enables Google Sheets to automatically convert data that it recognizes. For example, if the input data is a date, then Google Sheets formats it as a date. If the input data is a decimal, then Google Sheets formats it as a decimal.
 - **Unspecified** defaults to the Google Sheets API setting, which defaults to **User entered**.
 - **Raw** causes Google Sheets to insert the input data as is.
 - **User entered** enables automatic conversion of recognizable data.
 - e. Click **Done** to add the connection to the flow. The connection appears in the integration visualization flow in the location where you added it.
4. If you want to add any other connections to the flow, add them now and then return to these instructions.
5. After the flow has all desired connections, in the integration visualization flow, click the plus sign that is just before the Google Sheets connection that updates sheet values.
6. Click the **Data Mapper** card to add a data mapping step to the flow.
7. In the data mapper:
 - a. If you did not specify the spreadsheet ID when you configured the **Update sheet values** action, then map a source **spreadsheetId** to the target **spreadsheetId**.
 - b. Specify the data that you want to use to update the sheet by mapping fields from a source step to the target spreadsheet. For example, you might map fields from another spreadsheet or from a database.
If you need to, you can edit the Google Sheets connection that updates sheet values and change the settings for **Range** or **Major Dimension**. Changing these settings causes the data mapper to display different target fields according to your changes.
 - c. In the upper right, click **Done** to add the data mapper step.

8.4.6. Appending data to a sheet

You can append data to a sheet in the middle of a flow or to finish a simple integration. To do this, add a Google Sheets connection to the middle of a flow or as a simple integration's finish connection.

Prerequisites

- You created a Google Sheets connection that is authorized to access the Google spreadsheet in which you want to append data.
- You have access to the ID for the spreadsheet in which you want to append data.
- You are creating or editing a flow and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to choose a finish connection.

Procedure

1. Click a Google Sheets connection that is authorized to access the spreadsheet in which you want to append data.
2. On the **Choose an Action** page, click **Append values to a sheet**
3. To configure the **Append values to a sheet** action:
 - a. In the **SpreadsheetId** field, do one of the following:
 - Enter the ID of the spreadsheet that you want to append values to.
 - Leave the field blank. In this case, after you add this connection to the flow, you must add a data mapper step before this connection. In the data mapper step, map the spreadsheet ID from a previous connection to this connection. The previous connection must access the spreadsheet that you want to append data to in this connection.
 - b. In the **Range** field, use Google A1 notation to specify the spreadsheet range that you want to append data to. Specify the start and end coordinates, for example, **A3:D3** appends data to the first sheet, after the third row for 4 columns. The default is **A:A**, which appends data to the first column in the first sheet in the spreadsheet.
The connection never overwrites data. The connection starts appending data after the range you specify, and then always appends data to the content that is in place.
 - c. In the **Major Dimension** field, accept **Rows**, which is the default, or select **Columns**. **Rows** configures the action to use row objects to append data. Each row object contains a value for each column that you want to append data to. **Columns** configures the action to use column objects to append data. Each column object contains a value for each row that you want to append.
 - d. In the **Value Input Option** field, indicate how you want Google sheets to interpret the data that it receives for appending to a sheet. **Unspecified**, which is the default, enables Google Sheets to automatically convert data that it recognizes. For example, if the input data is a date, then Google Sheets formats it as a date. If the input data is a decimal, then Google Sheets formats it as a decimal.
 - **Unspecified** defaults to the Google Sheets API setting, which defaults to **User entered**.
 - **Raw** does nothing. Google Sheets inserts the input data as is.
 - **User entered** enables automatic conversion of recognizable data.

- e. Click **Done** to add the connection to the flow. The connection appears in the integration visualization flow in the location where you added it.
The values that you specify to configure the **Append values to a sheet** action determine the fields that the data mapper displays for mapping to or from this connection.
4. If you want to add any other connections to the flow, add them now and then return to these instructions.
5. After the flow has all desired connections, in the integration visualization flow, click the plus sign that is just before the Google Sheets connection that appends data to a sheet.
6. Click the **Data Mapper** card to add a data mapping step to the flow.
7. In the data mapper:
 - a. If you did not specify the spreadsheet ID when you configured the **Append values to a spreadsheet** action, then map a source **spreadsheetId** to the target **spreadsheetId**.
 - b. Specify the data to append to the sheet by mapping fields from a source step to the target spreadsheet. For example, you might map fields from another spreadsheet or from a database.
If you need to, you can edit the Google Sheets connection that appends sheet values and change the settings for **Range** or **Major Dimension**. Changing these settings causes the data mapper to display different target fields according to your changes.
 - c. In the upper right, click **Done** to add the data mapper step.

8.4.7. Updating spreadsheet properties

In an integration, you can update the properties of a spreadsheet in the middle of a flow or to finish a simple integration. To do this, add a Google Sheets connection to the middle of a flow or as a simple integration's finish connection.

Properties include the spreadsheet's title, locale, and time zone.

When Fuse Online prompts you to configure the **Update spreadsheet properties** action, you can leave some or all fields blank. If you leave a field blank, then in a data mapper step that is in the flow (you add it later) before this connection, you map fields from previous steps to blank **Update spreadsheet properties** action configuration fields.

Prerequisites

- You created a Google Sheets connection that is authorized to access the Google spreadsheet whose properties you want to update.
- You have access to the spreadsheet ID for the spreadsheet whose properties you want to update.
- You are creating or editing a flow and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to choose a finish connection.

Procedure

1. Click a Google Sheets connection that is authorized to access the spreadsheet whose properties you want to update.

2. On the **Choose an Action** page, click **Update spreadsheet properties**.
3. To configure the **Update spreadsheet properties** action:
 - a. In the **SpreadsheetId** field, do one of the following:
 - Enter the ID of the spreadsheet whose properties you want to update.
 - Leave the field blank. In this case, after you add this connection to the flow, you must add a data mapper step that is before this connection. In the data mapper step, map the spreadsheet ID from a previous connection to this connection. The previous connection must access the spreadsheet that you want to update in this connection.
 - b. In the other fields, enter a value only if you want to change the property. Alternatively, you can leave the fields blank. If you do, then in a data mapper step that you add later, just before this connection, you can map the fields that you want to change.
 - c. Click **Done** to add the connection to the flow. The connection appears in the integration visualization flow in the location where you added it.
4. If you want to add any other connections to the flow, add them now and then return to these instructions.
5. After the flow has all desired connections, you might want to use a data mapper step to update spreadsheet properties:
 - a. In the integration visualization flow, click the plus sign that is just before the Google Sheets connection that updates properties.
 - b. Click **Data Mapper** to add a data mapping step to the flow.
 - c. In the data mapper, if you did not specify a spreadsheet ID when you configured the **Update spreadsheet properties** action, map a source **spreadsheetId** to the target **spreadsheetId**.
 - d. For each spreadsheet property that you want to change and for which you did not specify an updated value when you configured the **Update spreadsheet properties** action, map a field from a source step to the target spreadsheet property that you want to change.
 - e. In the upper right, click **Done** to add the data mapper step.

8.4.8. Adding a chart to a sheet

In the middle of a flow, or to finish a simple integration, you can add a basic chart or a pie chart to a Google Sheets spreadsheet. To do this, add a Google Sheets connection to the middle of a flow or as a simple integration's finish connection. Then add a data mapping step just before the connection. In the data mapping step, set options that determine the location, properties, and content of the chart.

When Fuse Online prompts you to configure the **Add charts** action, you can leave some or all fields blank. If you leave a field blank, you can map a value to that field in the data mapper step that you will add just before this connection.

Prerequisites

- You created a Google Sheets connection that is authorized to access the Google spreadsheet that you want to add a chart to.
- You have access to the ID for the spreadsheet that you want to add a chart to.

- You are creating or editing a flow and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to choose the finish connection for a simple integration.

Procedure

1. Click a Google Sheets connection that is authorized to access the spreadsheet that you want to add a chart to.
2. On the **Choose an Action** page, click **Add charts**.
3. To configure the **Add charts** action:
 - a. In the **SpreadsheetId** field, do one of the following:
 - Enter the ID of the spreadsheet that you want to add a chart to.
 - Leave the field blank. In the data mapper step that will be just before this connection, map the spreadsheet ID from a previous connection to this connection. The previous connection must access the spreadsheet that you want to add a chart to in this connection.
 - b. In the **Chart Title** field, enter a title.
Alternatively, leave the field blank. In the data mapper step that is just before this connection, map the chart title from a source field or property to the **title** target field.
 - c. In the **Subtitle** field, enter a subtitle.
Alternatively, leave the field blank. In the data mapper step that is just before this connection, map the chart subtitle from a source field or property to the **subtitle** target field.
4. Click **Done** to add the connection to the flow. The connection appears in the integration visualization flow in the location where you added it.
5. If you want to add any other connections to the flow, add them now. Then return to these instructions.
6. In the integration visualization flow, click the plus sign that is just before the connection that adds a chart.
7. Click **Data Mapper**.
In the data mapper, map source properties, source fields, or source constants to the target Google Sheets connection, which is adding a chart. The following instructions assume that there is a source field, a source property, or a source constant that contains the value that you want to map to each target field.

If there is no source for the required value, then create a source property in the data mapper by clicking the + sign that is to the right of the **Properties** folder. In the **Create Property** dialog, give the property a name that makes it easy to map to the correct target field, for example, specify **overlayPosition** as the property name. For the property value, specify the value that you want the connection to use to create the chart. The chart's overlay position field requires A1 notation that identifies a cell. That is, it specifies a start position but an end position specification is not required, and it is ignored if it is present. For example, specify something like **D4** as the value of the **overlayPosition** source property. Save the new property.

When you configured the **Add charts** action, you might have specified a value for the spreadsheet ID, the chart title, and/or the chart subtitle. If you did, then do not map a value to that field.

8. To add a chart to a sheet, map source properties, source fields or source constants to target fields as follows:

- a. Set the location of the chart by mapping an A1 notation value, which identifies a cell, to the **overlayPosition** target field. An end position is not required and it is ignored if present. The mapped value identifies the cell where Google Sheets places the chart. This cell is the upper left corner of the chart display.

If you do not map a value to the **overlayPosition** field, and you also do not map a value to the **sheetId** field, which identifies the sheet that you want to add the chart to, then the default behavior is that Google Sheets creates a new sheet and places the chart in A1.

You must map a value to the **overlayPosition** field if you plan to map a value to the **sheetId** field.

- b. Map a source **spreadsheetId** field to the target **spreadsheetId** field. The connection uses source data in this spreadsheet to add the chart to a sheet in this same spreadsheet. The source data and the chart can be in two different sheets in the same spreadsheet but they cannot be in two different spreadsheets.
- c. Set a chart title by mapping to the target **title** field. The connection inserts this string as the title of the chart.
- d. Set a chart subtitle by mapping to the target **subtitle** field. The connection inserts this string as a chart subtitle.
- e. Identify the sheet to add the chart to by mapping an integer to the **sheetId** field. If you map a value to the **sheetId** field then you must also map a value to the **overlayPosition** field.

If you do not map a value to the **sheetId** field and you also do not map a value to the **overlayPosition** field, then the connection creates a new sheet and adds the chart to the new sheet.

If you do not map a value to the **sheetId** field but you do map a value to the **overlayPosition** field, then the connection places the chart on the sheet identified by the **sourceSheetId** field. This is the sheet that provides the data for the chart.

0 identifies the first sheet in the spreadsheet. For subsequent sheets, the sheet ID is at the end of the URL when the sheet is visible in a browser. For example, at the end of the following URL, you can see **gid=206589330**:

<https://docs.google.com/spreadsheets/d/1pa...ngQbKkM/edit#gid=206589330>

This indicates that the sheet ID is **206589330** and that is the value that you would map to **sheetId**.

- f. Identify the sheet that provides the data for the chart by mapping its integer sheet ID to the target **sourceSheetId** field. The default is **0**, which identifies the first sheet in the spreadsheet. You can add a basic chart or a pie chart. Follow one of the following sets of instructions according to the kind of chart that you want to add.
- g. To add basic chart, in the target panel, expand the **basicChart** folder and map source fields to target fields that are in the **basicChart** folder:
 - i. Set the chart type (**BAR**, **LINE**, **AREA**, or **COLUMN**) by mapping to the target **type** field. The default is **COLUMN**.

- ii. Set the title for the bottom axis of the chart by mapping to the target **axisTitleBottom** field.
 - iii. Set the title for the left axis of the chart by mapping to the target **axisTitleLeft** field.
 - iv. Set the high level category of information that the chart shows by mapping to the target **domainRange** field. This field uses the Google Sheets A1 notation. For example, suppose that the first column in a source sheet provides a list of 5 products in **A2** through **A6**, and you want the chart to provide some data for each product. In this column, the first cell (**A1**) must contain a label, such as **Products**. The setting of **domainRange** would be **A1:A6**. The chart will contain the data that is in **A2** through **A6**. The chart does not display the label.
 - v. Identify the location of the source data that the chart shows by mapping to the target **dataRange** field. This field uses the Google Sheets A1 notation. A chart can provide only one series of data. Continuing with the **domainRange** example, suppose that the second column in a source sheet shows the number sold for each of the 5 products that are listed in the first column and this is the data that you want the chart to show. In this column, the first cell (**B1**) must contain a label, such as **Number Sold**. The setting of **dataRange** would be **B1:B6**. The chart will contain the data that is in **B2** through **B6**. The chart does not display the label.
- h. To add a pie chart, in the target panel, expand the **pieChart** folder and map source fields to target fields that are in the **pieChart** folder:
- i. Set the location of the pie chart legend by mapping to the target **legendPosition** field. The default is **LEFT_LEGEND**. The value must be **BOTTOM_LEGEND**, **LEFT_LEGEND**, **RIGHT_LEGEND**, **TOP_LEGEND**, or **NO_LEGEND**.
 - ii. Set the high level category of information that the pie chart shows by mapping to the target **domainRange** field. This field uses the Google Sheets A1 notation. For example, suppose that the first column in a source sheet provides a list of 5 products in **A2** through **A6**, and you want the chart to provide some data for each product. In this column, the first cell (**A1**) must contain a label, such as **Products**. The setting of **domainRange** would be **A1:A6**. The chart will contain the data that is in **A2** through **A6**. The chart does not display the label.
 - iii. Identify the location of the source data that the chart shows by mapping to the target **dataRange** field. This field uses the Google Sheets A1 notation. A chart can provide only one series of data. Continuing with the **domainRange** example, suppose that the second column in a source sheet shows the number sold for each of the 5 products that are listed in the first column and this is the data that you want the chart to show. In this column, the first cell (**B1**) must contain a label, such as **Number Sold**. The setting of **dataRange** would be **B1:B6**. The chart will contain the data that is in **B2** through **B6**. The chart does not display the label.
- i. In the upper right, click **Done** to add the data mapper step.

8.4.9. Adding a pivot table to a sheet

In the middle of a flow, or to finish a simple integration, you can add a pivot table to a Google Sheets spreadsheet. A pivot table lets you aggregate, sort, or apply a function to spreadsheet data and display the results in the same spreadsheet. To add a pivot table, add a Google Sheets connection to the middle of a flow or as a simple integration's finish connection. Then add a data mapping step before the connection. In the data mapping step, you set options that determine the location, properties, and content of the pivot table.

In this release, the **Add pivot table** action is limited to defining one value group, one row pivot group, and one column pivot group. Support for multiple groups in a single action is expected to be added in a future release. As a workaround, you can add multiple Google Sheets connections that add pivot tables based on the same source spreadsheet.

Prerequisites

- You created a Google Sheets connection that is authorized to access the Google spreadsheet that you want to add a pivot table to.
- You have access to the ID for the spreadsheet that contains the source data for the pivot table.
- You are creating or editing a flow and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to choose the finish connection for a simple integration.

Procedure

1. Click a Google Sheets connection that is authorized to access the spreadsheet that you want to add the pivot table to.
2. On the **Choose an Action** page, click **Add pivot tables**.
3. To configure the **Add pivot tables** action, in the **SpreadsheetId** field, do one of the following:
 - Enter the ID of the spreadsheet that you want to add a pivot table to.
 - Leave the field blank. In the data mapper step that you will add just before this connection, you will map the spreadsheet ID from a previous connection to this connection. The previous connection must access the spreadsheet that you want to add a pivot table to in this connection.
4. Click **Done** to add the connection to the flow. The connection appears in the integration visualization flow in the location where you added it.
5. If you want to add any other connections to this flow, add them now. Then return to these instructions.
6. In the integration visualization flow, click the plus sign that is just before the connection that adds a pivot table to a sheet.

7. Click **Data Mapper**.

In the data mapper, map source properties, source fields, or source constants to the target Google Sheets connection, which is adding a pivot table. The following instructions assume that there is a source field, source property, or source constant that contains the value that you want to map to each target field.

If there is no source for the required value, then create a source property in the data mapper by clicking the + sign that is to the right of the **Properties** folder. In the **Create Property** dialog, give the property a name that makes it easy to map to the correct target field, for example, specify **valueLayout** as the property name. Specify the value that you want the connection to use to create the pivot table. The pivot table's value layout field requires a value of **HORIZONTAL** or **VERTICAL** so you might specify **VERTICAL** as the value of the **valueLayout** property. Save the new property.

8. In the data mapper, configure a new pivot table as follows:

- a. Map a source **spreadsheetId** field to the target **spreadsheetId** field. The connection uses

source data in this spreadsheet to add the pivot table to a sheet in this same spreadsheet. The source data and the pivot table can be in two different sheets in the same spreadsheet but they cannot be in two different spreadsheets.

- b. Identify the sheet to add the pivot table to by mapping an integer to the **sheetId** field. The default is **0**, which identifies the first sheet in the spreadsheet. For subsequent sheets, the sheet ID is at the end of the URL when the sheet is visible in a browser. For example, at the end of the following URL, you can **gid=206589330**:

<https://docs.google.com/spreadsheets/d/1pa...ngQbKkM/edit#gid=206589330>.

This indicates that the sheet ID is **206589330** and that is the value that you would map to **sheetId**.

- c. Identify the sheet that provides the data for generating the pivot table by mapping its integer sheet ID to the target **sourceSheetId** field. If you do not map a value to **sourceSheetId**, the connection uses the **sheetId** value, or **0** if no value is mapped to **sheetId**. **0** identifies the first sheet in the spreadsheet.
- d. Set the source data range for generating the pivot table by mapping an A1 notation value to the target **sourceRange** field. For example, **A2:D5**.
- e. Set the value layout to be used on the pivot table by mapping to the target **valueLayout** field. The value must be **VERTICAL** or **HORIZONTAL**, which is the default. **HORIZONTAL** specifies that values are laid out as columns.
- f. Set the top left cell of the pivot table by mapping an A1 notation cell coordinate to the target **start** field. When the connection adds the pivot table to the sheet that contains the source data, if you do not map a value to the **start** field, the default is that the top left cell of the pivot table is the cell that is in the first row of the source data range and in the first empty column after the source data range. When the source data is in one sheet and the pivot table will be in another sheet, the default is **A1**.
- g. To add a value group, expand **valueGroups** in the target panel, and map values to the fields under **valueGroups**:
 - i. Map a value group name to the target **name** field.
 - ii. Map a function, which you want to apply to the value group, to the target **function** field. The value must be **SUM**, **COUNT**, **AVERAGE**, **MAX**, **MIN**, or **CUSTOM**. The default is **SUM**. Map **CUSTOM** to **function** when you are defining a formula.
 - iii. Map a custom formula, which you want to apply to the value group, to the target **formula** field if you mapped the **CUSTOM** value to the **function** field.
 - iv. Map a column name, which is the coordinate that builds the value group, to the target **sourceColumn** field.
- h. To add a row pivot group, expand **columnGroups** in the target panel, and map values to the fields under **columnGroups**:
 - i. Map a row pivot group name to the target **label** field.
 - ii. Map the sort order that you want to apply to the row pivot group to the target **sortOrder** field. The value must be **ASCENDING** or **DESCENDING**. The default is **ASCENDING**.

- iii. Map **true** or **false** to the target **showTotals** field. The default is **true**, which enables the display of totals for the row pivot group.
 - iv. Map a column name, which is the coordinate that builds the row pivot group, to the target **sourceColumn** field.
- i. To add a column pivot group, expand **rowGroups** in the target panel, and map values to the fields under **rowGroups**:
- i. Map a column pivot group name to the target **label** field.
 - ii. Map the sort order that you want to apply to the column pivot group to the target **sortOrder** field. The value must be **ASCENDING** or **DESCENDING**. The default is **ASCENDING**.
 - iii. Map **true** or **false** to the target **showTotals** field. The default is **true**, which enables the display of totals for the column pivot group.
 - iv. Map a column name, which is the coordinate that builds the column pivot group, to the target **sourceColumn** field.
- j. In the upper right, click **Done** to add the data mapper step.

Additional resource

[Google developers documentation for pivot tables](#)

CHAPTER 9. CONNECTING 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 flow. The following topics provide details:

- [Section 9.1, "Creating a connection to an HTTP or HTTPS endpoint"](#)
- [Section 9.2, "Adding an HTTP or HTTPS connection to an integration"](#)

9.1. CREATING 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 Hyper Text Transfer Protocol Secure, 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. ADDING 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.

- You are creating or editing a flow and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to choose a start connection, or to choose a finish connection.

Procedure

1. Click the HTTP or HTTPS connection that you want to add to the flow.
2. 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 middle or finish connection, then **Invoke URL** is the only available action. This action invokes the endpoint once.
3. In the **URL Path** field, specify the location of the endpoint that you want to invoke.
4. 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.
5. 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.
6. Click **Done** to specify the action's input or 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.

CHAPTER 10. CONNECTING TO IRC

Internet Relay Chat (IRC) is a protocol that lets anyone with an IRC client send and receive text messages in real time. In an integration, a connection to IRC can do one of the following:

- Trigger execution of a simple integration when an IRC nickname that you specify receives a private message. The integration passes the message to the next step in the flow. For example, you can designate an IRC nickname for receiving names of possible customers. Upon receiving a message, the integration could connect to Salesforce to create a lead.
- Deliver a message to a particular channel. The message is from the nickname that you specify. For example, this is useful when an integration downloads a file from an FTP server and processes it in some way. An integration flow can send a message to a particular channel to indicate that the process was successful.

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

- [Section 10.1, "Creating an IRC connection"](#)
- [Section 10.2, "Adding an IRC connection to trigger integration execution upon receiving messages"](#)
- [Section 10.3, "Adding an IRC connection to send a message"](#)

10.1. CREATING AN IRC CONNECTION

In an integration, an IRC connection can retrieve messages that are sent to an IRC nickname that you specify or send messages to a nickname on a channel. You can use the same IRC connection in any number of integrations.

Prerequisites

- You know the hostname and port of the IRC server that you want to connect to.
- If the IRC server requires a password, you know the password.

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 **IRC** connector.
4. In the **Hostname** field, enter an IRC hostname. You can specify an IP address or a name, for example, **chat.freenode.net**.
5. In the **Port** field, enter the port that the IRC server is listening on, for example, **6665**.
6. In the **Password** field, if the IRC server requires a password, enter it here.
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 **IRC Freenode**.
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 **IRC Freenode** appears as a connection that you can choose to add to an integration. You can add this connection to any number of integrations.

10.2. ADDING AN IRC CONNECTION TO TRIGGER INTEGRATION EXECUTION UPON RECEIVING MESSAGES

An IRC connection that starts an integration triggers execution of the integration when the connection finds private messages that are sent to an IRC nickname that you specify.

Prerequisites

- You created an IRC connection.
- You have an IRC nickname for receiving and sending messages.
- You are creating a simple integration and Fuse Online is prompting you to choose how you want to start the integration.

Procedure

1. Click the IRC connection that you want to use to start the integration.
2. Click the **IRC Receive Private Message** action.
3. In the **Channels** field, optionally specify one or more IRC channels separated with commas.
4. In the **Nickname** field, specify the IRC nickname that the integration uses. For any channels that you specify, this nickname joins the channel. During execution, the integration retrieves messages that are sent to this nickname and passes them to the next step in the integration.
5. In the **NickServ Password** field, if the specified nickname has a password, enter it.
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, if you are specifying a JSON schema for vendors, then you might 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** to add the connection to the flow.

Result

The connection appears at the beginning of the integration flow.

10.3. ADDING AN IRC CONNECTION TO SEND A MESSAGE

In an integration, you can send a message to one or more IRC channels in the middle of a flow or to finish a simple integration. To do this, add an IRC connection to the middle of a flow or as a simple integration's finish connection.

Prerequisites

- You created an IRC connection.
- You have an IRC nickname for receiving and sending messages.
- You are creating or editing a flow and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to choose a finish connection.

Procedure

1. Click the IRC connection that you want to add to the flow.
2. Click **IRC Send Message to Channel** to send a message to one or more channels.
 - In the **Channels** field, enter one or more IRC channels delimited by commas. During execution, this connection sends messages from the IRC nickname that you specify to these channels.
3. In the **Nickname** field, specify the IRC nickname that sends messages to the specified channels.
4. In the **NickServ password** field, if the nickname that you specified has a password, specify it here.
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, if you are specifying a JSON schema for vendors, then you might 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** to add the connection to the flow.

Results

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

CHAPTER 11. CONNECTING 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 flow. Details are in the following topics:

- [Section 11.1, “Creating a connection to a Kafka broker”](#)
- [Section 11.2, “Obtaining data from a Kafka broker to trigger integration execution”](#)
- [Section 11.3, “Publishing data to a Kafka broker”](#)

11.1. CREATING 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.

11.2. OBTAINING DATA FROM A KAFKA BROKER TO TRIGGER INTEGRATION EXECUTION

To trigger execution of an integration upon 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 at the beginning of the integration flow.

11.3. PUBLISHING DATA TO A KAFKA BROKER

In an integration, you can publish data to a Kafka broker in the middle of a flow or to finish a simple integration. To do this, add a Kafka connection to the middle of a flow or as a simple integration's finish connection.

Prerequisites

- You created a connection to a Kafka broker.
- You are creating or editing a flow and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to choose a finish connection.

Procedure

1. Click the Kafka connection that you want to use to publish messages.
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 12. CONNECTING TO APACHE KUDU

Apache Kudu is a columnar storage manager developed for the Apache Hadoop platform. A Fuse Online integration can connect to a Kudu data store to scan a table, which returns all records in the table to the integration, or to insert records into a table. Details are in the following topics:

- [Section 12.1, “Creating a connection to an Apache Kudu data store”](#)
- [Section 12.2, “Triggering an integration when scanning returns records from a Kudu table”](#)
- [Section 12.3, “Inserting records into a Kudu table”](#)

12.1. CREATING A CONNECTION TO AN APACHE KUDU DATA STORE

In an integration, to obtain records from or insert records into a Kudu table, create a connection to a Kudu master host and then add that connection to an integration.

Prerequisite

- You must know the IP address or the hostname for the Kudu master host that you want to connect to.
- You must know the port that Kudu is listening on.

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 **Apache Kudu** connector.
4. To configure the connection:
 - a. In the **Address of Kudu master host** field, enter the hostname or the IP address of the Kudu master host.
 - b. In the **Port to establish connection to** field, enter the port that Kudu is listening on. The default is **7051**.
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, you might enter **Kudu North**.
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 that the connection you created is now available. If you entered the example name, you would see that **Kudu North** appears as a connection that you can choose to add to an integration.

12.2. TRIGGERING AN INTEGRATION WHEN SCANNING RETURNS RECORDS FROM A KUDU TABLE

To trigger execution of an integration upon obtaining data from a Kudu table, add a Kudu connection to a simple integration as its start connection. When the integration is running, the Kudu connection scans the table that you specified at the interval that you specified, obtains all records in the table, and passes a collection of the records to the next step in the integration.

A Kudu connection can obtain data from only one table. Between scans, if there are no changes to the data in the table that the connection scans, then the next scan returns the same data as the previous scan.

Prerequisite

- You created a Kudu connection.
- The table that you want to obtain records from exists.

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 Kudu connection that you want to use to start the integration.
4. On the **Choose an Action** page, click the **Scan a Kudu table** action.
5. In the **Table** field, enter the name of the table that you want to obtain records from.
6. In the **Period** field, accept the default of one minute, or enter the interval at which you want the connection to scan the table and return the records that are in the table.
7. Click **Done**.

Result

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

Next steps

Add the integration's finish connection and any other connections that you want to include in the integration. When the integration contains all the connections that are needed, consider whether you need to split the collection of records that the Kudu connection returns. If you want to execute integration steps for each record that you obtained from the Kudu table, then after the Kudu connection, add a split step. Also, you probably need to follow the Kudu connection with a data mapping step that maps data obtained from Kudu to fields in subsequent connections in the integration.

12.3. INSERTING RECORDS INTO A KUDU TABLE

In an integration, you can add records to a Kudu table in the middle of a flow or to finish a simple integration. To do this, add a Kudu connection to the middle of a flow or as a simple integration's finish connection.

Prerequisites

- You created a Kudu connection.

- You are creating or editing an integration and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to choose a finish connection.
- The table that you want to add records to exists.

Procedure

1. Click the Kudu connection that you want to use. Note that when a Kudu connection inserts data, the connection does not return anything.
2. On the **Choose an Action** page, click **Insert data into a Kudu table**
3. To configure the action, in the **Table** field, specify the name of the table to add records to. It is important for you to have an understanding of how the Kudu table that you are adding records to is set up. For example, a Kudu table that you are adding records to might have a unique key. If you try to add a record that contains a key value that is already in the table, the Kudu connection does not add that record.
4. Click **Done**.

Result

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

Next steps

Consider whether you need to split a collection of records into individual records that a Kudu connection can add to a table. To do this, add a split step to the integration. The split step executes the subsequent steps in the integration once for each record. Also, you probably need a data mapper step before a Kudu connection that adds records to a table.

CHAPTER 13. CONNECTING 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 flow. Details are in the following topics:

- [Section 13.1, “Creating a connection to an MQTT broker”](#)
- [Section 13.2, “Obtaining a message from an MQTT broker to trigger integration execution”](#)
- [Section 13.3, “Publishing a message to an MQTT broker”](#)

13.1. CREATING 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.

13.2. OBTAINING A MESSAGE FROM AN MQTT BROKER TO TRIGGER INTEGRATION EXECUTION

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 at the beginning of the integration flow.

13.3. PUBLISHING A MESSAGE TO AN MQTT BROKER

In an integration, you can publish a message to an MQTT broker in the middle of a flow or to finish a simple integration. To do this, add an MQTT connection to the middle of a flow or as the integration's finish connection.

Prerequisites

- You created an MQTT connection.
- You are creating or editing a flow and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to choose a finish connection.

Procedure

1. Click the MQTT connection that you want to use to publish a message.
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.

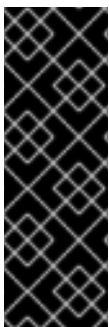
CHAPTER 14. CONNECTING TO ODATA

Open Data Protocol (OData) is a standard for building and consuming REST APIs. An OData service provides services to clients, such as Fuse Online, through OData-compliant HTTP requests. A Fuse Online integration can obtain entities from an OData service, and can update, create, or delete entities managed by an OData service. To do this, create an OData connection, and then add it to an integration.

It is assumed that you are familiar with the [OData specification](#).

Details are in the following topics:

- [Section 14.1, “Creating a connection to an OData service”](#)
- [Section 14.2, “Triggering an integration when polling returns data from an OData service”](#)
- [Section 14.3, “Updating, creating, and deleting data that is managed by an OData service”](#)



IMPORTANT

The OData connector is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process. For more information about the support scope of Red Hat Technology Preview features, see [link:https://access.redhat.com/support/offerings/techpreview/](https://access.redhat.com/support/offerings/techpreview/).

14.1. CREATING A CONNECTION TO AN ODATA SERVICE

In an integration, to obtain entities from an OData service, or to update, create or delete entities that are managed by an OData service, you must first create a connection to that OData service.

Prerequisite

- You must know the base URL for the OData service that you want to connect to.
- If that service uses authentication then you must have the required credentials, and, if required, an SSL certificate. However, most OData services are public and do not require authentication.

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 **OData** connector.
4. To configure the connection:
 - a. In the **Service Root URL** field, enter the base URL for the OData service that you want to access.
 - b. In the **User Name** field, if the service uses basic authentication, then enter your user name.
 - c. In the **Password** field, if the service uses basic authentication, then enter your password.

- d. In the **Server Certificate** field, if the service requires it, paste the content of an SSL certificate.

Typically, a public OData service does not require a certificate in addition to the certificates provided by the browser. However, for an internal OData service, you might have created your own SSL certificate and self-signed it. Since this certificate is not trusted by a certificate authority, connections to your OData service would fail. To enable connections, paste the self-signed certificate here.

One way to get the certificate is to load the service in a browser. The steps after that depend on which browser you are using, but it will be something like the following: click the **Not secure** padlock symbol next to the address bar, then click **View Certificate**, export the displayed certificate to a file, copy the certificate, and paste it into this field.

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(s) 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 **OData North**.
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 **OData North** appears as a connection that you can choose to add to an integration.

Next step

Add your OData connection to an integration.

14.2. TRIGGERING AN INTEGRATION WHEN POLLING RETURNS DATA FROM AN ODATA SERVICE

To trigger execution of an integration upon obtaining data from an OData service, add an OData connection to a simple integration as its start connection. When the integration is running, the OData connection polls the service at intervals that you specify. When the connection finds data that satisfies the connection configuration, the connection passes the data to the next step in the integration.

Prerequisite

You created an OData 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 OData connection that you want to use to start the integration.
4. On the **Choose an Action** page, click the **Read** action.
5. To configure the **Read** action:

- a. In the **Resource Collection** field, select the name of the OData resource that you want to query. Fuse Online fetches data from the OData service to provide a list of available resource collections.
- b. In the **Entity Key Predicate** field, to obtain a particular entity, identify the entity that you want by specifying its key predicate. For example, specify something like **UserName='Bob'** or **Categories(1)**. To obtain multiple entities, leave this field blank.
- c. In the **Query Options** field, enter a query that you want to apply to the resource. Use OData syntax. For example, **\$filter=startswith(name, 'N')** returns a message for each entity in the resource that has a **name** field that starts with **N**.
You can specify both **Entity Key Predicate** and **Query Options**. If you do, the OData service obtains the specified entity and applies the query to that entity. For example, suppose that you set **Entity Key Predicate** to **UserName='russellwhyte'/Emails** and you set **Query Options** to **\$filter=contains(\$it,'example')**. The connection creates a request that looks something like this:

```
https://services.odata.org/TripPinRESTierService/People(UserName='russellwhyte')/Emails
?$filter=contains($it, 'example')
```

The service returns all email addresses for Russell Whyte that contain the domain **example**.

- d. Select the **Filter Old Results** checkbox to obtain a particular message only once.
- e. Select the **Split Results** checkbox if you want the connection to return individual messages rather than a collection of messages.
Fuse Online also provides discrete split and aggregate steps, which you can add to a flow. If you want to process individual messages in one or more steps and then aggregate the messages into a collection, do not select the **Split Results** checkbox. Instead, leave the checkbox empty and then add a split step to the flow after this connection. A split step is required if you want an aggregate step in the flow.
- f. In the **Interval Before Polling Starts** field, accept the default of 1 second, or enter the length of time that you want to elapse before the connection starts to poll the OData service.
- g. In the **Delay** field, accept the default of 30 seconds, or enter the interval at which you want the connection to poll the OData service.
- h. In the **Backoff Idle Threshold** field, accept the default of 1 or enter an integer that indicates the number of consecutive polls that can return no data. After this number of polls, the connection increases the interval between subsequent polls. The connection determines the new length of the polling interval by multiplying the **Delay** value by the **Backoff Multiplier** value.
For example, suppose that the polling interval (the **Delay** value) is the default of 30 seconds, **Backoff Idle Threshold** is set to 5, and **Backoff Multiplier** is set to 12. After 5 consecutive polls that return no data, the connection waits 360 seconds (30 x 12) before polling again. The connection continues to poll every 360 seconds until a poll returns data. After a poll returns data, the connection resumes polling every 30 seconds.
- i. In the **Backoff Multiplier** field, accept the default of 1 or enter an integer that indicates the multiplier for increasing the polling interval if the value set for the **Backoff Idle Threshold** is reached.
If you accept the default of 1 for the **Backoff Multiplier**, then the connection continues to poll at the specified interval no matter how many consecutive polls return no results.

The values that you specify for **Backoff Idle Threshold** and **Backoff Multiplier** are useful for reducing CPU overhead since the connection can automatically poll less often during quiet periods.

6. Click **Done**.

Result

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

During execution, what the connection returns depends on what you specified in the **Entity Key Predicate** and **Query Options** fields. The OData connection can return:

- A collection of entities or a collection of entity properties
For example, this might be all **Person** entities in the resource, or perhaps all **Age** properties for all **Person** entities. The connection returns the collection in one message. Fuse Online executes each subsequent step in the flow once for the collection. However, when you configure the **Read** action, if you select **Split Results**, then the connection returns each entity or each property in its own message. Fuse Online executes each subsequent step in the flow once for each message.
- An entity or an entity property
For example, this might be the **Person** entity whose **UserName** property is **Bob**, or perhaps the **Age** property for the **Person** entity whose **UserName** is **Bob**. The connection returns the entity or the entity property in a message that it passes to the next step in the flow.

Next steps

Add the integration's finish connection and any other connections that you want to include in the integration. When the integration contains all the connections that are needed, if the OData connection returns a collection, consider whether you need to add a split step after the OData connection. An integration usually needs to map the data returned by the OData connection to fields that a subsequent connection in the flow can use. Sometimes you can map a collection, but more often you need to split the collection in order to map to the target fields.

After the OData connection, add a data mapper step to the flow. Exactly where in the flow depends on what you want the flow to do. For example, after the OData connection, you might add a basic filter step and then add a data mapper step.

14.3. UPDATING, CREATING, AND DELETING DATA THAT IS MANAGED BY AN ODATA SERVICE

In an integration, you can update a resource that is managed by an OData service in the middle of a flow or to finish a simple integration. To do this, add an OData connection to the middle of a flow or as a simple integration's finish connection.

Prerequisites

- You created an OData connection.
- You are creating or editing an integration and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to choose a finish connection.

Procedure

1. Click the OData connection that you want to use. Note that when an OData connection updates or deletes an entity, the connection does not return anything. When the OData connection creates an entity, the connection returns the new entity.
2. On the **Choose an Action** page, click the action that you want the connection to perform:
 - **Create** adds an entity to an OData resource.
 - **Delete** removes an entity from an OData resource.
 - **Update** changes an entity that is in an OData resource.
3. To configure the action, select the OData resource that you want to update.

It is important for you to have an understanding of how the OData resource that you are updating is set up. The backing data source for an OData service determines the rules for updates and resolving conflicts. For example, suppose an OData connection tries to create a new entity but an entity with that predicate key already exists. The OData service might overwrite the existing entity, or it might update some fields in the existing entity, or it might ignore the operation. It is up to you to know how the OData service behaves in this situation.
4. Click **Done**.

Result

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

Next steps

Add a data mapper step before the OData connection. You must map source fields that provide the data that is required to create a new entity, to update an entity, or to delete an entity. If you map a collection to a target field, the default behavior is that the content in the last element that is in the source collection is mapped to the target field.

CHAPTER 15. CONNECTING 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 document that describes the API. See [Integrating Applications with Fuse Online, Adding and managing 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 15.1, "Registering Fuse Online as a REST API client"](#) if required
- [Section 15.2, "Creating a REST API client connection"](#)
- [Section 15.3, "Adding an API client connection to an integration"](#)

15.1. REGISTERING FUSE ONLINE AS A REST API CLIENT


Before Fuse Online creates a REST 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 [Creating a REST API client 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**.

15.2. CREATING A REST 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 your 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 Authentication, 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.

15.3. ADDING AN API CLIENT CONNECTION TO AN INTEGRATION

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

In this release, a connection to a REST API can be in the middle of a flow or it can be the finish connection in a simple integration. In other words, a connection to a REST API cannot be a start connection.

Prerequisites

- You created a connection to the REST API.
- You are creating or editing a flow.

- The flow already has its start connection.
- Fuse Online is prompting you to select a finish connection or to choose a step.

Procedure

1. On the page that displays available steps, click the REST API connection that you want to add to the flow.
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 document that was uploaded to Fuse Online and that describes the API that 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 flow in the location where you added it.

CHAPTER 16. CONNECTING 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 16.1, “Registering Fuse Online as a Salesforce client application”](#)
- [Section 16.2, “Creating a Salesforce connection”](#)
- [Section 16.3, “Adding a Salesforce connection to an integration”](#)

16.1. REGISTERING 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-proj912876.7b63.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.



- 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**.

16.2. CREATING 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.
- You added the Salesforce client ID and client secret that you received after registration to the Fuse Online **Settings** page.
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.

If **Connect Salesforce** does not appear, then your Fuse Online environment is not registered as a Salesforce client application. See [Registering Fuse Online as a Salesforce client application](#) . When you try to create a Salesforce connection and your Fuse Online environment is not registered as a Salesforce client application, then Fuse Online displays multiple fields that prompt for authorization information. While you can create a Salesforce connection by entering values in these fields, it is not recommended.



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 [Registering Fuse Online as a Salesforce client application](#).

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.

16.3. ADDING A SALESFORCE CONNECTION TO AN INTEGRATION

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

Prerequisites

- You created a Salesforce connection.
- You are creating or updating an integration.

- Fuse Online is prompting you to choose a start connection, or to choose a finish connection, or to choose a step.

Procedure

1. Click the Salesforce connection that you want to add to the flow. When the integration uses the connection that 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 a flow 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 flow.

Result

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

CHAPTER 17. CONNECTING 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 17.1, "User roles for connecting to SAP Concur"](#)
- [Section 17.2, "How to Obtain SAP Concur implementation site OAuth credentials"](#)
- [Section 17.3, "How to obtain SAP Concur production site OAuth credentials"](#)
- [Section 17.4, "Configuring the SAP Concur connector"](#)
- [Section 17.5, "Creating a SAP Concur connection"](#)
- [Section 17.6, "Adding a SAP Concur connection to an integration"](#)
- [Section 17.7, "Example of mapping integration data fields to SAP Concur fields"](#)

17.1. 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 Concur connector for access to the appropriate SAP Concur site.
3. Create a SAP Concur connection.

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

17.2. 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.

17.3. 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.

17.4. CONFIGURING 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.

17.5. CREATING 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 [Configuring the 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.

17.6. ADDING A SAP CONCUR CONNECTION TO AN INTEGRATION

You can add a SAP Concur connection to the middle of a flow or as the finish connection in a simple integration. A connection to SAP Concur can perform any one of 85 actions.

Prerequisites

- You created a SAP Concur connection.
- You are creating or editing a flow.
- Fuse Online is prompting you to choose a step, or to choose the finish connection in a simple integration.

Procedure

1. Click the SAP Concur connection that you want to add to the flow. 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 a flow 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 flow.

Result

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

17.7. 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. The integration visualization flow is on the left and Fuse Online is prompting you to choose a step.

1. 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.
2. Map the **concur-list-id** source field to the **id** target field.
3. Map the **project-code** source field to the **content** target field.
4. In the upper right, click **Done** to add the data mapper step to the integration.

CHAPTER 18. CONNECTING TO SERVICENOW

An integration can retrieve records from a ServiceNow table or add records to a ServiceNow import set, which ServiceNow uses to update a table. In an integration, to connect to ServiceNow, create a ServiceNow connection and then add that connection to an integration. For details, see:

- [Section 18.1, "Creating a ServiceNow connection"](#)
- [Section 18.2, "Obtaining records from ServiceNow to trigger integration execution"](#)
- [Section 18.3, "Creating an import set in ServiceNow"](#)
- [Section 18.4, "Copying records to ServiceNow during or to finish an integration"](#)
- [Section 18.5, "Example Salesforce to ServiceNow integrations"](#)

18.1. CREATING 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 and your ServiceNow user name and password.
- If the ServiceNow administrator also created a ServiceNow client ID and client secret for you, 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**, 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 validation is successful. If validation fails, update the configuration details as needed and try again.
 - g. When validation is successful, click **Next**.
5. In the **Connection Name** field, enter a name that helps you distinguish this connection from 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.

18.2. OBTAINING RECORDS FROM SERVICENOW TO TRIGGER INTEGRATION EXECUTION

To trigger execution of an integration upon receiving records from ServiceNow, add a ServiceNow connection to a simple integration as its 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 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, 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.

18.3. CREATING AN IMPORT SET IN SERVICENOW

In a Fuse Online integration, a ServiceNow connection cannot directly update a ServiceNow table. To update ServiceNow data in an integration, a prerequisite is an import set that stages updates for a ServiceNow table.

Often, the easiest way to create an import set is for a ServiceNow administrator to create a ServiceNow inbound web service. This implicitly creates a ServiceNow import set. The import set is based on the ServiceNow target table that the administrator selects when creating the web service. A ServiceNow connection updates this import set, and ServiceNow uses the import set to update the corresponding ServiceNow table.

Prerequisites

You must have ServiceNow administrative privileges.

Procedure in ServiceNow

1. In ServiceNow, search for **web services**.
2. In the results, under **System Web Services > Inbound**, click **Create New**.
3. In the **Create Web Service** page:
 - a. In the **Label** field, specify the display name for the web service. The web service's import set will also have this name.
 - b. In the **Target table** field, select the ServiceNow table that this service updates.
 - c. Select **Copy fields from target table**. This creates an import set that has the same schema as the target table. The import set is a staging area for updating the target table.
 - d. Select **Create transform map**. This map enables ServiceNow to copy and transform data from the import set to the target table.
4. Click **Create**.
5. In the next display, under **Related Links**, click **Auto Map Matching Fields** to display a list of fields in the target table that you selected.

6. Select one target table field to be the correlation (or coalesce) field.
When the value of the correlation field in a record in the import set matches the value of a correlation field in the target table, ServiceNow updates that record in the target table rather than creating a new record.
7. Select **Coalesce** to identify the field that you just selected as the correlation field. Without a correlation field, ServiceNow adds each record in the import set to the target table.
8. Click **Update**.
9. Confirm that the transform map was created:
 - a. In ServiceNow, search for **transform map**.
 - b. Under **System Import Sets**, click **Transform Maps** to display a list of transform maps.
 - c. In this list, confirm that there is a transform map whose name is the value that you specified for the label of the web service that you created.

Additional resource

[ServiceNow documentation for import sets](#)

18.4. COPYING RECORDS TO SERVICENOW DURING OR TO FINISH AN INTEGRATION

You can copy records to ServiceNow in the middle of a flow or to finish a simple integration. To do this, add a ServiceNow connection to the middle of a flow or as a simple integration's finish connection.

Prerequisites


- You are creating or editing a flow and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to choose a finish 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. If the import set does not exist, see [Creating an import set in ServiceNow](#) .
- The ServiceNow import set is configured to handle the addition of records.

Procedure

1. Click the ServiceNow connection that you want to add to the integration.
2. Click the **Add Record** action.
3. Click in the **Import Set** field and select the import set that you want to add records to.
4. Click **Next**.

Results

- Fuse Online adds the connection to the flow.

- Typically, a data mapper step is needed before this connection. In the integration visualization panel, Fuse Online displays  on the connection. In most integrations, you need to map a source field to the target field used as the correlation field in the ServiceNow table.

18.5. EXAMPLE SALESFORCE TO SERVICENOW INTEGRATIONS

This example describes two simple integrations:

- One integration obtains new cases from Salesforce and adds them as incidents to ServiceNow.
- The other integration obtains updated cases from Salesforce and updates a corresponding incident in ServiceNow.

Prerequisite (in ServiceNow)

For both integrations, a prerequisite is for a ServiceNow administrator to create an inbound web service. This implicitly creates a ServiceNow import set, which is required for an integration to add data to ServiceNow. For this example:

- The name of the web service is **Cases from Salesforce**.
- The service's target table is **Incidents**. This table contains the data to be updated.
- The import set has the same schema as the **Incidents** table because of the selection of **Copy fields from target table**. The name of the import set is also **Cases from Salesforce**.
- ServiceNow can copy and transform data from the **Cases from Salesforce** import set to the **Incidents** target table because of the selection of **Create transform map**.
- In the import set, the correlation field is **correlation_Id** because of the selection of the **correlation_Id** field followed by selection of **Coalesce**.

Procedure overview (in Fuse Online)

In Fuse Online, you create the Salesforce and ServiceNow connections.

The two integrations are the same except for the action performed by the Salesforce connection, as noted in the following steps. To create the integrations, you would perform these steps twice, that is, once to create each integration:

1. Add the Salesforce connection as the start connection.
 - To create the integration that obtains new cases, select the **On create** action.
 - To create the integration that obtains updated cases, select the **On update** action.
2. Select the **Case** object for the action to operate on. This Salesforce connection returns new **Case** objects or updated **Case** objects according to the chosen action.
3. Add the ServiceNow connection as the finish connection.
4. Select the **Create Record** action, and then select **Cases from Salesforce** as the import set that the integration updates. This ServiceNow connection adds new or updated Salesforce cases to the **Cases from Salesforce** import set.
5. Add a data mapper step that maps:

- Salesforce case **id** to the ServiceNow import set's **correlation_id**
 - Salesforce **subject** to the ServiceNow **short_description**
 - Salesforce **description** to the ServiceNow **description**
6. Give the integration a name and then publish it.
 - For the integration that obtains new cases, use **On SF Create Case**
 - For the integration that obtains updated cases, **On SF Update Case**

Confirm that the integrations work

When both integrations are running, you can confirm that the integrations work:

1. In Salesforce, create a case.
2. In Fuse Online, view the summary for the **On SF Create Case** integration. Click its **Activity** tab to see that Fuse Online executed the integration once.
3. In ServiceNow, view the **Incidents** table. You should see a new incident that has the subject and description that you specified in Salesforce.
4. Back in Salesforce, update the case that you just created by changing its subject.
5. In Fuse Online, view the summary for the **On SF Update Case** integration. Click its **Activity** tab to see that Fuse Online executed the integration once.
6. In ServiceNow, view the **Incidents** table and expand the entry for the incident that was previously new. You should see that this incident has an updated **short_description** value. ServiceNow checks the import set entry for the value of its **correlation_id**. If this value already exists in the **Incidents** table, ServiceNow updates the incident that has that value.

CHAPTER 19. CONNECTING 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 a simple integration when a Slack channel that you specify receives a message. The integration passes the message to the next step in the flow. 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. An integration flow can notify 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 flow. Details are in the following topics:

- [Section 19.1, “Creating a Slack connection”](#)
- [Section 19.2, “Adding a Slack connection to trigger integration execution upon receiving messages”](#)
- [Section 19.3, “Adding a Slack connection to send a message to a Slack channel or user”](#)

19.1. CREATING 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.

19.2. ADDING A SLACK CONNECTION TO TRIGGER INTEGRATION 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.

19.3. ADDING 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 in the middle of a flow or to finish a simple integration. To do this, add a Slack connection to the middle of a flow or as a simple integration's finish connection.

Prerequisites

- You created a Slack connection.
- You are creating or editing a flow and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to choose a finish connection.

Procedure

1. Click the Slack connection that you want to add to the flow.
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 flow.
4. Optionally, add additional connections to the flow. Whether additional connections are needed depends on what you want the flow 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 [Adding a data mapping step](#).

CHAPTER 20. CONNECTING 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 following.

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

See the following topics for details:

- [Section 20.1, "Creating a database connection"](#)
- [Section 20.2, "Starting an integration by accessing a database"](#)
- [Section 20.3, "Accessing a database in the middle or to complete an integration"](#)
- [Section 20.4, "Connecting to proprietary databases"](#)

20.1. CREATING 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.

20.2. STARTING 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 that 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.

20.3. ACCESSING A DATABASE IN THE MIDDLE OR TO COMPLETE AN INTEGRATION

In an integration, you can access a database in the middle of a flow or to finish a simple integration. To do this, add a database connection to the middle of a flow or as a simple integration's finish connection.

Prerequisites

- You created a database connection.
- You are creating or editing a flow and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to choose a finish connection.

Procedure

1. Click the database connection for the database that 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 that 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 flow. 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 a flow or to finish 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 the flow 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 [Adding a data mapping step](#).

20.4. CONNECTING 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 a database connection](#).
5. In an integration, you add the connection to your database. See [Starting an integration by accessing a database](#) or [Accessing a database in the middle or to complete an integration](#).

CHAPTER 21. CONNECTING 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 a simple integration when a Telegram bot that you specify receives a message. The integration passes the message to the next step in the flow. For example, after a simple 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 add a Telegram connection that notifies 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 flow. Details are in the following topics:

- [Section 21.1, "Creating a Telegram connection"](#)
- [Section 21.2, "Adding a Telegram connection to trigger integration execution upon receiving messages"](#)
- [Section 21.3, "Adding a Telegram connection to send a message to a Telegram chat"](#)

21.1. CREATING 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.

21.2. ADDING 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 a simple 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 configuration identifies the chat bot that you want to receive messages from. No other configuration is required.

21.3. ADDING A TELEGRAM CONNECTION TO SEND A MESSAGE TO A TELEGRAM CHAT

In an integration, you can send a message to a Telegram chat in the middle of a flow or to finish a simple integration. To do this, add a Telegram connection to the middle of a flow or as a simple integration's finish connection.

Prerequisites

- You created a Telegram connection.
- You are creating or editing a flow and Fuse Online is prompting you to choose a step. Or, Fuse Online is prompting you to choose a finish connection.

Procedure

1. Click the Telegram connection that you want to add to the flow.

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 flow.

CHAPTER 22. TRIGGERING INTEGRATION EXECUTION WITH A TIMER

To trigger execution of a simple integration according to a schedule that you specify, add a timer connection as a simple integration's start connection. Fuse Online provides a timer connection, which you can use to start as many simple 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 22.1, "Adding a timer connection to trigger integration execution"](#)
- [Section 22.2, "Creating a replacement timer connection"](#)

22.1. ADDING 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 a simple integration's start connection. A timer connection cannot be in the middle of a flow nor at the end of a flow.

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 triggering 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 that 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.

22.2. CREATING 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 23. CONNECTING TO TWITTER

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

- [Section 23.1, “Registering Fuse Online as a Twitter client application”](#)
- [Section 23.2, “Creating a Twitter connection”](#)
- [Section 23.3, “Adding a Twitter connection to trigger integration execution”](#)

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**.

23.1. REGISTERING 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 API Key** and **ConsumerAPI Secret Key** 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-proj912876.7b63.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 consumer API key.
3. Return to your Fuse Online **Settings** page and paste the Twitter consumer API key into the Fuse Online Twitter **Consumer API 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 API Secret Key** field.
 6. Click **Save** and then click **Ok**.

23.2. CREATING 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.
- You added the Twitter consumer API key and consumer API secret key that you received after registration to the Fuse Online **Settings** page.

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.
If **Connect Twitter** does not appear, then your Fuse Online environment is not registered as a Twitter client application. See [Registering Fuse Online as a Twitter client application](#) . When you try to create a Twitter connection and your Fuse Online environment is not registered as a Twitter client application, then Fuse Online displays multiple fields that prompt for authorization information. While you can create a Twitter connection by entering values in these fields, it is not recommended.
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.

23.3. ADDING A TWITTER CONNECTION TO TRIGGER INTEGRATION EXECUTION

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. 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 at the beginning of the integration flow.

CHAPTER 24. TRIGGERING INTEGRATION EXECUTION WITH AN HTTP REQUEST (WEBHOOK)

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

- [Section 24.1, "General procedure for using the Fuse Online webhook"](#)
- [Section 24.2, "Using the Webhook connection"](#)
- [Section 24.3, "Creating an integration that an HTTP request can trigger"](#)
- [Section 24.4, "How Fuse Online handles HTTP requests"](#)
- [Section 24.5, "Guidelines for an HTTP client that invokes a Fuse Online Webhook"](#)
- [Section 24.6, "About the JSON schema for specifying request parameters"](#)
- [Section 24.7, "How to specify HTTP requests"](#)

24.1. 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.

24.2. USING THE WEBHOOK CONNECTION

Fuse Online provides a Webhook connection that you can use in any number of simple 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.

24.3. CREATING 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 [About the 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. Add any other needed steps.
10. Immediately after the start connection, add a data mapper step.
11. Click **Publish**, give the integration a name and optionally, a description, and click **Publish** again.

24.4. HOW FUSE ONLINE HANDLES HTTP REQUESTS

You can specify an HTTP **GET** or **POST** request to trigger execution of a simple 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 a simple integration that is triggered by a **GET** request and that updates data rather than obtaining data. Likewise, you can define a simple integration that is triggered by a **POST** request and that obtains data rather than updating data.

24.5. 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 [About the 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.

24.6. 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.

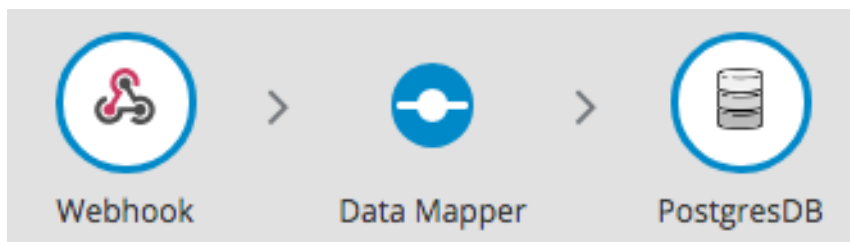
For examples, see [How to specify HTTP requests](#).

24.7. 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

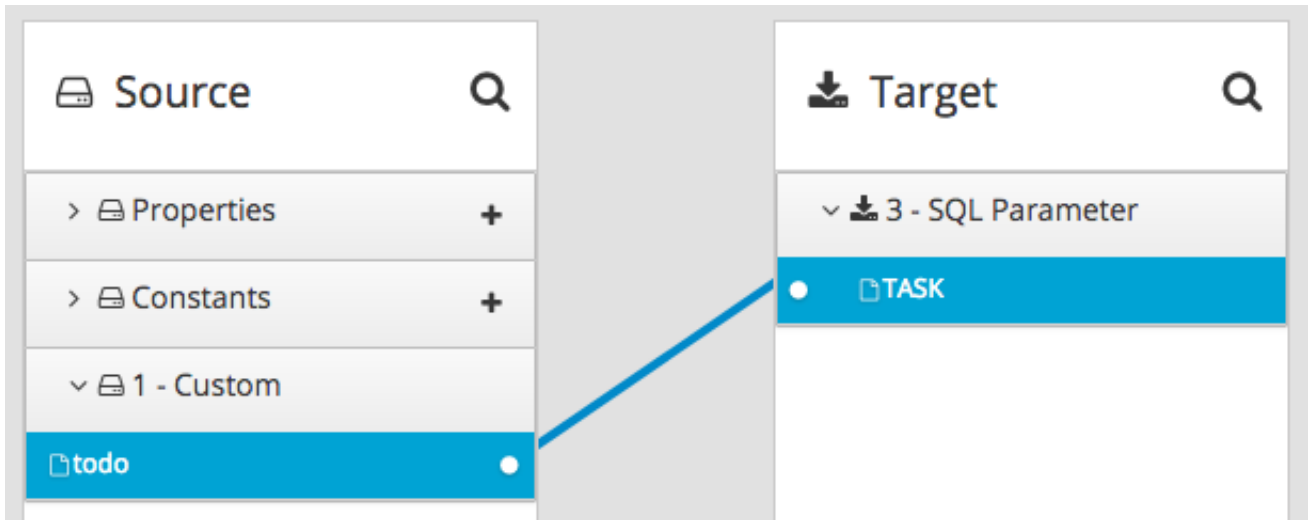
This is a typeless connection. To use type-aware functionality, enter information that defines the data type

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"/>

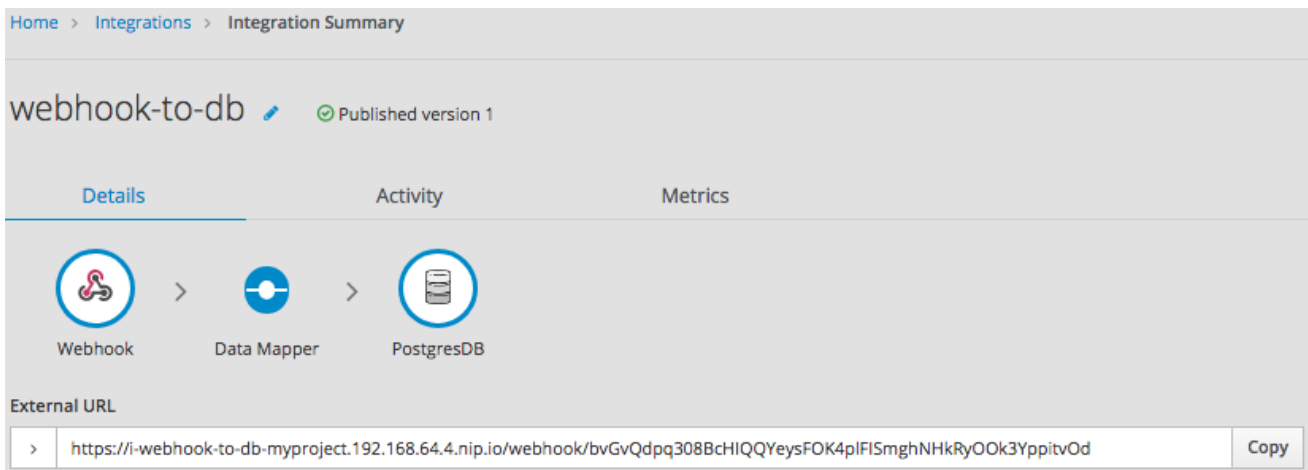
When you add the **PostgresDB** connection as the finish connection, you select the **Invoke SQL** action and specify this SQL statement:

```
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/bvGvQdpq308BcHIQQYeysFOK4pIFISmghNHkRyOOK3YppitvOd

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

Value	Description
bvGvQdpq308BcHIQ QYeysFOK4pIFISmg hNHkRyOOk3Yppitv Od	<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:

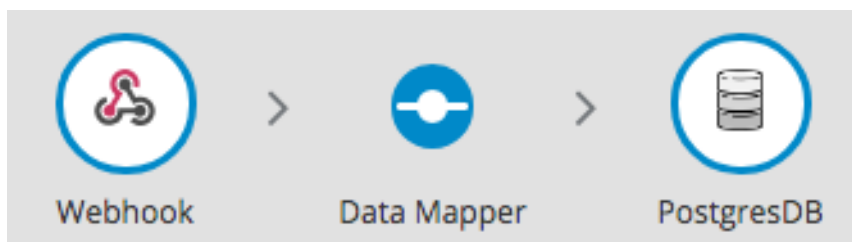
```
curl -H 'Content-Type: application/json' -d '{"todo":"from webhook"}' https://i-webhook-to-db-myproject.192.168.64.4.nip.io/webhook/bvGvQdpq308BcHIQQYeysFOK4pIFISmgghNHkRyOOk3YppitvOd
```

- 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:synthesis: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/ZYWrhaW7dVk097vNsLX3YJ1GyxUFMFRteLpw0z4O69
MW7d2Kjg?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",

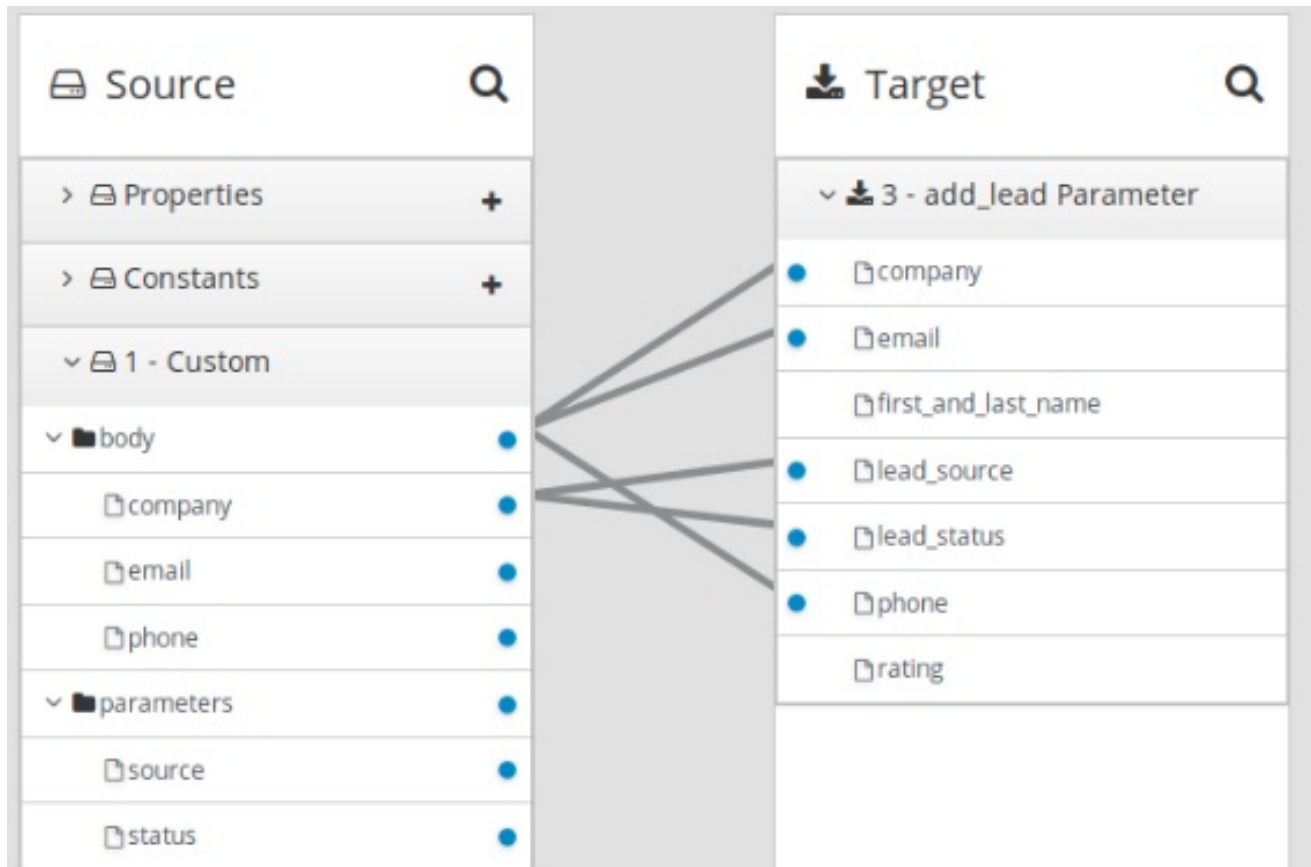
```

```

    "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/ZYW7dVvK097vNsLX3YJ1GyxUFMFRteLpw0z4O69MW7d2Kjg'
```

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:synthesis: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/ZYWrhoW7dVk097vNsLX3YJ1GyxUFMFRteLpw0z4O69  
MW7d2Kjg?source=web&status=new'
```