



Red Hat Fuse 7.3

Fuse Online のアプリケーションおよびサービス への接続

提供されるコネクタの使用手順

Red Hat Fuse 7.3 Fuse Online のアプリケーションおよびサービスへの接続

提供されるコネクタの使用手順

法律上の通知

Copyright © 2021 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.

概要

ステップごとの手順にしたがって、コネクションを作成し、インテグレーションに追加します。

目次

| | |
|---|----|
| 前書き | 5 |
| 第1章 FUSE ONLINE によってサポートされるコネクタ | 6 |
| 第2章 AMAZON S3 への接続 | 8 |
| 2.1. AMAZON S3 コネクションの作成における前提条件 | 8 |
| 2.2. AMAZON S3 コネクションの作成 | 9 |
| 2.3. AMAZON S3 からデータを取得してインテグレーションの実行をトリガー | 9 |
| 2.4. AMAZON S3 にデータを追加してインテグレーションを終了 | 11 |
| 2.5. インテグレーションの途中で AMAZON S3 にデータを追加 | 12 |
| 第3章 AMQ への接続 | 14 |
| 3.1. AMQ コネクションの作成 | 14 |
| 3.2. AMQ コネクションを追加してメッセージ受信時にインテグレーションの実行をトリガー | 15 |
| 3.3. AMQ メッセージをパブリッシュしてインテグレーションを終了 | 17 |
| 3.4. インテグレーションの途中で AMQ メッセージをパブリッシュ | 18 |
| 第4章 AMQP への接続 | 21 |
| 4.1. AMQP コネクションの作成 | 21 |
| 4.2. AMQP コネクションを追加してメッセージ受信時にインテグレーションの実行をトリガー | 22 |
| 4.3. フローの途中で AMQP へメッセージをパブリッシュまたは AMQP へメッセージをパブリッシュしてインテグレーションを終了 | 24 |
| 第5章 DROPBOX への接続 | 27 |
| 5.1. FUSE ONLINE を DROPBOX クライアントとして登録 | 27 |
| 5.2. DROPBOX コネクションの作成 | 28 |
| 5.3. DROPBOX からファイルを取得してインテグレーションの実行をトリガー | 29 |
| 5.4. DROPBOX にファイルを追加してインテグレーションを終了 | 30 |
| 5.5. インテグレーションの途中で DROPBOX にアクセス | 31 |
| 第6章 FHIR への接続 | 34 |
| 6.1. FHIR サーバーへのコネクションの作成 | 34 |
| 6.2. FHIR サーバーからのリソースの取得 | 36 |
| 6.3. リソースの FHIR サーバーのクエリー | 36 |
| 6.4. FHIR サーバーでのリソースの作成 | 37 |
| 6.5. FHIR サーバー上のリソースにあるすべてのフィールドを更新 | 38 |
| 6.6. FHIR サーバー上のリソースにある指定のフィールドを更新 | 40 |
| 6.7. FHIR サーバーで異なるタイプのリソースの作成 | 41 |
| 6.8. FHIR サーバーからのリソースの削除 | 42 |
| 第7章 FTP または SFTP サーバーへの接続 | 44 |
| 7.1. FTP または SFTP コネクションの作成 | 44 |
| 7.2. FTP または SFTP サーバーからのファイルの取得 | 45 |
| 7.3. FTP または SFTP サーバーへのファイルのアップロード | 47 |
| 第8章 GOOGLE アプリケーションへの接続 | 49 |
| 8.1. FUSE ONLINE を GOOGLE クライアントアプリケーションとして登録 | 49 |
| 8.2. GMAIL への接続 | 52 |
| 8.2.1. Gmail コネクションの作成 | 52 |
| 8.2.2. 送信メールの代替の作成方法 | 53 |
| 8.2.3. ポーリングによる Gmail メッセージの返信時にインテグレーションをトリガー | 54 |
| 8.2.4. Gmail アカウントからのメールの送信 | 55 |
| 8.3. GOOGLE カレンダーへの接続 | 56 |
| 8.3.1. Google カレンダーコネクションの作成 | 57 |

| | |
|--|------------|
| 8.3.2. ポーリングが Google カレンダーからイベントを返すときにインテグレーションをトリガー | 57 |
| 8.3.3. Google カレンダーからの特定イベントの取得 | 59 |
| 8.3.4. イベントの Google カレンダーへの追加 | 60 |
| 8.3.5. Google カレンダーでのイベントの更新 | 60 |
| 8.4. GOOGLE スプレッドシートへの接続 | 61 |
| 8.4.1. Google スプレッドシートコネクションの作成 | 62 |
| 8.4.2. ポーリングがスプレッドシートデータを返すときにインテグレーションをトリガー | 63 |
| 8.4.3. ポーリングがスプレッドシートプロパティを返すときにインテグレーションをトリガー | 65 |
| 8.4.4. スプレッドシートの作成 | 65 |
| 8.4.5. シートのデータの更新 | 66 |
| 8.4.6. データをシートに追加 | 68 |
| 8.4.7. スプレッドシートプロパティの更新 | 69 |
| 8.4.8. チャートをシートに追加 | 71 |
| 8.4.9. ピボットテーブルをシートに追加 | 74 |
| 第9章 HTTP および HTTPS エンドポイントへの接続 | 77 |
| 9.1. HTTP または HTTPS エンドポイントへのコネクションの作成 | 77 |
| 9.2. HTTP または HTTPS コネクションのインテグレーションへの追加 | 77 |
| 第10章 IRC への接続 | 80 |
| 10.1. IRC コネクションの作成 | 80 |
| 10.2. メッセージの受信時にインテグレーションの実行をトリガーするために IRC コネクションを追加 | 81 |
| 10.3. IRC コネクションを追加してメッセージを送信 | 82 |
| 第11章 KAFKA への接続 | 84 |
| 11.1. KAFKA ブローカーへのコネクションの作成 | 84 |
| 11.2. KAFKA ブローカーからデータを取得してインテグレーションの実行をトリガー | 84 |
| 11.3. KAFKA ブローカーへのデータのパブリッシュ | 86 |
| 第12章 APACHE KUDU への接続 | 88 |
| 12.1. APACHE KUDU データストアへのコネクションの作成 | 88 |
| 12.2. スキャンが KUDU テーブルからレコードを返すときにインテグレーションをトリガー | 88 |
| 12.3. KUDU テーブルへのレコードの挿入 | 89 |
| 第13章 MQTT への接続 | 91 |
| 13.1. MQTT ブローカーへのコネクションの作成 | 91 |
| 13.2. MQTT ブローカーからメッセージを取得してインテグレーションの実行をトリガー | 92 |
| 13.3. MQTT ブローカーへのメッセージのパブリッシュ | 93 |
| 第14章 ODATA への接続 | 95 |
| 14.1. ODATA サービスへのコネクションの作成 | 95 |
| 14.2. ポーリングが ODATA からデータを返すときにインテグレーションをトリガー | 96 |
| 14.3. ODATA サービスによって管理されるデータの更新、作成、および削除 | 99 |
| 第15章 REST API への接続 | 100 |
| 15.1. FUSE ONLINE を REST API クライアントとして登録 | 100 |
| 15.2. REST API クライアントコネクションの作成 | 101 |
| 15.3. API クライアントのインテグレーションへの追加 | 101 |
| 第16章 SALESFORCE への接続 | 103 |
| 16.1. FUSE ONLINE を SALESFORCE クライアントアプリケーションとして登録 | 103 |
| 16.2. SALESFORCE コネクションの作成 | 104 |
| 16.3. SALESFORCE コネクションのインテグレーションへの追加 | 105 |
| 第17章 SAP CONCUR への接続 | 107 |
| 17.1. SAP CONCUR へ接続するためのユーザーロール | 107 |

| | |
|--|------------|
| 17.2. SAP CONCUR 実装サイト用の OAUTH クレデンシャルの取得方法 | 107 |
| 17.3. SAP CONCUR 本番サイト用の OAUTH クレデンシャルの取得方法 | 108 |
| 17.4. SAP CONCUR コネクターの設定 | 108 |
| 17.5. SAP CONCUR コネクションの作成 | 109 |
| 17.6. SAP CONCUR コネクションのインテグレーションへの追加 | 110 |
| 17.7. インテグレーションデータフィールドを SAP CONCUR フィールドにマップする例 | 110 |
| 第18章 SERVICENOW への接続 | 112 |
| 18.1. SERVICENOW コネクションの作成 | 112 |
| 18.2. SERVICENOW からレコードを取得してインテグレーションの実行をトリガー | 113 |
| 18.3. SERVICENOW でのインポートセットの作成 | 114 |
| 18.4. インテグレーションの間またはインテグレーションを終了するためにレコードを SERVICENOW にコピー | 115 |
| 18.5. SALESFORCE から SERVICENOW へのインテグレーションの例 | 116 |
| 第19章 SLACK への接続 | 118 |
| 19.1. SLACK コネクションの作成 | 118 |
| 19.2. メッセージの受信時に SLACK コネクションを追加してインテグレーションの実行をトリガー | 119 |
| 19.3. SLACK コネクションを追加してメッセージを SLACK チャンネルまたはユーザーに送信 | 120 |
| 第20章 SQL データベースへの接続 | 122 |
| 20.1. データベースコネクションの作成 | 122 |
| 20.2. データベースにアクセスしてインテグレーションを開始 | 123 |
| 20.3. インテグレーションの途中またはインテグレーションを完了するためにデータベースにアクセス | 124 |
| 20.4. プロプライエタリーデータベースへの接続 | 125 |
| 第21章 TELEGRAM への接続 | 126 |
| 21.1. TELEGRAM コネクションの作成 | 126 |
| 21.2. メッセージの受信時に TELEGRAM コネクションを追加してインテグレーションの実行をトリガー | 127 |
| 21.3. メッセージを TELEGRAM チャットに送信するため TELEGRAM コネクションを追加 | 127 |
| 第22章 タイマーによるインテグレーション実行のトリガー | 129 |
| 22.1. インテグレーションの実行をトリガーするためタイマーコネクションを追加 | 129 |
| 22.2. 代替タイマーコネクションの作成 | 129 |
| 第23章 TWITTER への接続 | 131 |
| 23.1. FUSE ONLINE を TWITTER クライアントアプリケーションとして登録 | 131 |
| 23.2. TWITTER コネクションの作成 | 132 |
| 23.3. TWITTER コネクションを追加してインテグレーションの実行をトリガー | 133 |
| 第24章 HTTP リクエストによるインテグレーション実行のトリガー (WEBHOOK) | 135 |
| 24.1. FUSE ONLINE WEBHOOK を使用するための一般的な手順 | 135 |
| 24.2. WEBHOOK コネクションの使用 | 136 |
| 24.3. HTTP リクエストがトリガー可能なインテグレーションの作成 | 136 |
| 24.4. FUSE ONLINE による HTTP リクエストの処理方法 | 137 |
| 24.5. FUSE ONLINE WEBHOOK を呼び出す HTTP クライアントのガイドライン | 138 |
| 24.6. リクエストパラメーターを指定するための JSON スキーマ | 138 |
| 24.7. HTTP リクエストの指定方法 | 139 |

前書き

アプリケーションを統合するには、統合する各アプリケーションまたはサービスへのコネクションを作成します。コネクションの作成後にインテグレーションを作成し、統合するインテグレーションまたはサービスごとにコネクションを追加します。

Fuse Online は、コネクションを作成するためのテンプレートとなる [さまざまなコネクタをサポートします](#)。以下のトピックでは、コネクションを作成し、インテグレーションに追加するための詳細を説明しています。

- [2章Amazon S3 への接続](#)
- [3章AMQ への接続](#)
- [4章AMQP への接続](#)
- [5章Dropbox への接続](#)
- [6章FHIR への接続](#)
- [7章FTP またはSFTP サーバーへの接続](#)
- [8章Google アプリケーションへの接続](#)
- [9章HTTP およびHTTPS エンドポイントへの接続](#)
- [10章IRC への接続](#)
- [11章Kafka への接続](#)
- [12章Apache Kudu への接続](#)
- [13章MQTT への接続](#)
- [14章OData への接続](#)
- [15章REST API への接続](#)
- [16章Salesforce への接続](#)
- [17章SAP Concur への接続](#)
- [18章ServiceNow への接続](#)
- [19章Slack への接続](#)
- [20章SQL データベースへの接続](#)
- [21章Telegram への接続](#)
- [22章タイマーによるインテグレーション実行のトリガー](#)
- [23章Twitter への接続](#)
- [24章HTTP リクエストによるインテグレーション実行のトリガー \(Webhook\)](#)

第1章 FUSE ONLINE によってサポートされるコネクタ

Fuse Online では、以下のコネクタがサポートされます。

| 名前 | 説明 |
|-----------------|---|
| Amazon S3 | Amazon S3 バケットからデータを取得したり、データを Amazon S3 バケットにコピーしたりします。 |
| AMQ | Red Hat AMQ (ApacheMQ) ブローカーからメッセージを取得したり Red Hat AMQ (ApacheMQ) ブローカーにメッセージをパブリッシュします。 |
| AMQP | AMQP (Advanced Message Queue Protocol) ブローカーからメッセージを取得するか、AMQP ブローカーにメッセージをパブリッシュします。 |
| Dropbox | Dropbox からファイルをダウンロードしたり、Dropbox にファイルをアップロードしたりします。 |
| FHIR | FHIR サーバーからリソースを取得したり、FHIR サーバーのリソースを更新したりします。 |
| FTP/SFTP | FTP または SFTP サーバーからファイルをダウンロードしたり、FTP または SFTP サーバーへファイルをアップロードしたりします。 |
| Gmail | 特定の Gmail アカウントへ送信されたメッセージを取得し、特定の Gmail アカウントからメッセージを送信します。 |
| Google カレンダー | イベントを指定する Google カレンダーから取得したり、指定する Google カレンダーのイベントを追加または更新したりします。 |
| Google スプレッドシート | 指定する Google スプレッドシートからデータを取得、スプレッドシートのデータを追加/更新、チャートを作成、または指定するスプレッドシートでピボットテーブルを作成します。 |
| HTTP/HTTPS | HTTP または HTTPS エンドポイントに接続し、 GET 、 PUT 、 POST 、 DELETE 、 HEAD 、 OPTIONS 、 TRACE 、または PATCH メソッドを実行します。 |
| IRC | IRC ニックネームまたはチャンネルに送信されたメッセージを受信したり、特定のチャンネルの IRC ニックネームにメッセージを送信したりします。 |
| Kafka | 指定する Kafka トピックからレコードのストリームを取得したり、指定する Kafka トピックにレコードのストリームをパブリッシュします。 |
| Kudu | Apache Kudu データストアのテーブルからレコードを取得したり、Kudu データストアのテーブルにレコードを追加したりします。 |
| MQTT | MQTT (MQ Telemetry Transport) ブローカーからメッセージを取得したり、MQTT ブローカーへメッセージをパブリッシュしたりします。 |

| 名前 | 説明 |
|----------------------------|---|
| OData | OData サービスからエンティティを取得したり、OData サービスによって管理されるエンティティを更新、作成、または削除します。 |
| REST API | <p>OpenAPI ドキュメントをアップロードして、カスタム REST API クライアントコネクタを作成します。その後、その REST API へのコネクションを作成します。</p> <p>インテグレーションの実行をトリガーする操作を定義する OpenAPI ドキュメントをアップロードして、REST API プロバイダーインテグレーションを作成します。『Integrating Applications with Fuse Online』の「Creating an integration that is triggered by a REST API call」を参照してください。</p> |
| Salesforce | Salesforce レコードを作成、更新、取得、または削除します。 |
| SAP Concur | さまざまな SAP Concur アクションの1つを実行します。 |
| ServiceNow | ServiceNow インスタンスからレコードを取得したり、ServiceNow インスタンスへレコードをコピーしたりします。 |
| Slack | チャンネルからメッセージを取得したり、Slack チャンネルまたはユーザーにメッセージを送信したりします。 |
| SQL データベース | Apache Derby、MySQL、または PostgreSQL データベースで SQL ステートメントまたは SQL ストアドプロシージャを呼び出します。他のタイプの SQL データベースに接続するには、そのデータベースの JDBC ドライバーが含まれる Fuse Online ライブラリーエクステンションをアップロードします。 |
| Telegram | Telegram チャットボットを使用して、チャットからメッセージを取得したり、チャットにメッセージを送信したりします。 |
| Timer | シンプルなインテグレーションの実行をトリガーするため、簡単なタイマーまたは cron タイマーを設定します。 |
| Twitter | ユーザーをメンションするツイートまたは指定のデータが含まれるツイートによって、シンプルなインテグレーションの実行をトリガーします。 |
| Webhook | HTTP の GET または POST リクエストによってシンプルなインテグレーションの実行をトリガーします。 |

Fuse Online が必要なコネクタを提供しない場合、経験のある開発者であれば、カスタムコネクタを定義するエクステンションを作成して対応することができます。Fuse Online にアップロードする、エクステンションのコーディングやその **.jar** ファイルの作成に関する詳細は、以下を参照してください。

- [『Tooling User Guide』の Fuse ツーリングを使用したエクステンションの開発](#)
- [『Integrating Applications with Fuse Online』のエクステンションのコーディング方法](#)

第2章 AMAZON S3 への接続

インテグレーションは Amazon S3 バケットからデータを取得したり、Amazon S3 バケットへデータをコピーしたりすることができます。これには、Amazon S3 コネクションを作成し、そのコネクションをインテグレーションフローに追加します。詳細は以下を参照してください。

- 「Amazon S3 コネクションの作成における前提条件」
- 「Amazon S3 コネクションの作成」
- 「Amazon S3 からデータを取得してインテグレーションの実行をトリガー」
- 「Amazon S3 にデータを追加してインテグレーションを終了」
- 「インテグレーションの途中で Amazon S3 にデータを追加」

2.1. AMAZON S3 コネクションの作成における前提条件

Amazon S3 コネクションを作成するには、以下を知っておく必要があります。

- コネクションがアクセスするバケットを作成した (またはこれから作成する) Amazon Web Services (AWS) アカウントに関連する Amazon S3 アクセスキー ID。
現時点では存在しないバケットにアクセスするコネクションを作成できます。この場合、インテグレーションの実行が開始されると、このアクセスキー ID に関連する AWS アカウントを使用してバケットの作成を試みます。
- コネクションがアクセスするバケットを作成した (またはインテグレーションの実行が開始されたときに作成を試みる) AWS アカウントに関連する Amazon S3 シークレットアクセスキー。
- アクセスするバケットの名前、または Amazon リソースネーム (ARN)。
指定したバケットが存在しない場合、コネクションは指定した名前でバケットを作成しようとします。S3 では、バケットをパブリックにアクセスできる URL として使用することが可能であるため、指定するバケット名はグローバルで一意である必要があります。さらに、[S3 バケットの命名要件](#) に準拠する必要があります。

指定するバケットが Amazon S3 アクセスキー ID に関連する AWS アカウントに存在せず、別の AWS アカウントに存在する場合、コネクションはバケットを作成せず、このコネクションを使用するインテグレーションは実行を開始することができません。

- バケットのあるリージョン、またはコネクションがバケットを作成するリージョン。

バケットを作成した AWS アカウントまたはこれからバケットを作成する AWS アカウントのログインクレデンシャルを持つユーザーは、以下のように Amazon S3 キーを取得します。

1. <https://aws.amazon.com/s3/> にアクセスします。
2. **アカウント** で **AWS マネジメントコンソール** を選択します。アクセスするバケットを作成した AWS アカウントまたはバケットを作成するためにコネクションが使用するアカウントでコンソールにサインインします。
3. コンソール右上のユーザー名の横にある下矢印をクリックし、**セキュリティ認証情報** をクリックします。
4. **アクセスキー** を展開し、**新しいアクセスキーの作成スキーの作成** をクリックします。
5. プロンプトにしたがってキーを取得します。

2.2. AMAZON S3 コネクションの作成

Amazon S3 コネクションをインテグレーションに追加する前に Amazon S3 コネクションを作成する必要があります。

前提条件

「[Amazon S3 コネクション作成の前提条件](#)」を参照してください。

手順

1. Fuse Online の左パネルで **Connections** をクリックし、利用可能なコネクションを表示します。
2. 右上にある **Create Connection** をクリックし、Fuse Online コネクタを表示します。
3. **Amazon S3** コネクタをクリックします。
4. **Access Key** フィールドに、このコネクションがアクセスするバケットを作成した AWS アカウントの Amazon S3 アクセスキー ID を入力します。コネクションがアクセスするバケットが存在しない場合は、Fuse Online がインテグレーションの実行を開始しようとするときに、そのアクセスキーに関連する AWS アカウントを使用してバケットを作成します。しかし、他の AWS アカウントにバケットが存在する場合は、コネクションはバケットを作成できず、インテグレーションを開始できません。
5. **Bucket Name or Amazon Resource Name** フィールドに、このコネクションがアクセスするバケットの名前またはバケットの ARN を入力します。使用している AWS のアカウントまたは他の AWS アカウントにバケットが存在しない場合は、コネクションがこのバケットを作成します。
6. **Region** フィールドではバケットが存在する AWS リージョンを選択します。コネクションがバケットを作成する場合にここで選択されたリージョンに作成されます。
7. **Secret Key** フィールドに、このコネクションがアクセスするバケットを作成した (またはこれから作成する) アカウントの Amazon S3 シークレットアクセスキーを入力します。このキーは AWS によって提供されます。
8. **Validate** をクリックします。Fuse Online は即座にコネクションの検証を試み、検証の成功または失敗を示すメッセージを表示します。検証に失敗した場合は、必要に応じて設定の詳細を修正し、再度検証を行います。
9. 検証に成功したら **Next** をクリックします。
10. **Connection Name** フィールドに、このコネクションを別のコネクションと区別するために使用する名前を入力します。たとえば、**Obtain S3 Data** を入力します。
11. **Description** は任意のフィールドで、このコネクションに関する便利な情報を入力します。たとえば、**Sample S3 connection that obtains data from the northeast bucket.** を入力します。
12. 右上にある **Create** をクリックし、作成したコネクションが利用可能になったことを確認します。上記の例の名前を入力した場合は、インテグレーションに追加できるコネクションとして **Obtain S3 Data** が表示されます。

2.3. AMAZON S3 からデータを取得してインテグレーションの実行をトリガー

Amazon S3 バケットからデータを取得してインテグレーションを開始するには、Amazon S3 コネクションをシンプルなインテグレーションの最初のコネクションとして追加します。

前提条件

Amazon S3 コネクションが作成済みである必要があります。

手順

1. Fuse Online パネルの左側にある **Integrations** をクリックします。
2. **Create Integration** をクリックします。
3. **Choose a Start Connection** ページで、インテグレーションを開始するために使用する Amazon S3 コネクションをクリックします。
4. **Choose an Action** ページで、コネクションが実行するアクションをクリックします。
 - **Get Object** は、コネクションがアクセスするバケットからファイルを取得します。**File Name** フィールドでは、取得するファイルの名前を入力します。指定したファイルがバケットにない場合は、ランタイムエラーが発生します。
 - **Poll an Amazon S3 Bucket** は、コネクションがアクセスするバケットからファイルを定期的に取得します。このアクションを設定するには、以下を行います。
 - a. **Delay** フィールドで、デフォルトの 500 ミリ秒をポーリング間の経過時間として受け入れます。別のポーリング間隔を指定する場合は、数値を入力し、時間の単位を選択します。
 - b. **Maximum Objects to Retrieve** フィールドに、1 回のポーリング操作で取得可能なファイルの最大数を入力します。デフォルトは 10 です。
取得するファイルの数を無制限にするには、**0** または負の整数を指定します。**Maximum Objects to Retrieve** が無制限の場合、ポーリングはバケットのすべてのファイルを取得します。

バケットに指定された最大数を超える数のファイルが含まれている場合、編集または作成日時が最も新しいファイルから取得します。
 - c. **Prefix** フィールドには、文字列を評価する正規表現を任意で指定します。接頭辞を指定した場合、名前がこの文字列で始まるファイルのみを取得します。
 - d. **Obtain files and then delete them from the bucket**(ファイルを取得し、取得済みのファイルをバケットから削除) を行うかどうかを示します。
5. アクションの設定後、**Done** をクリックしてアクションの出力タイプを指定します。
6. データタイプの認識が必要がない場合は、**Select Type** フィールドで **Type specification not required** を受け入れ、下部にある **Done** をクリックします。残りの手順を行う必要はありません。
その他の場合は、以下の 1 つをスキーマタイプとして選択します。
 - **JSON schema** は JSON データの構造を記述するドキュメントです。ドキュメントのメディアタイプは **application/schema+json** です。
 - **JSON instance** は JSON データが含まれるドキュメントです。ドキュメントのメディアタイプは **application/json** です。

- **XML schema** は XML データの構造を記述するドキュメントです。このドキュメントのファイル拡張子は **.xsd** です。
 - **XML instance** は XML データが含まれるドキュメントです。このドキュメントのファイル拡張子は **.xml** です。
7. **Definition** 入力ボックスに、選択したスキーマタイプに準拠する定義を貼り付けます。たとえば、**JSON schema** を選択した場合は、メディアタイプが **application/schema+json** の JSON スキーマファイルの内容を貼り付けます。
 8. **Data Type Name** フィールドにはデータタイプに選択した名前を入力します。たとえば、ベンダーの JSON スキーマを指定する場合は、データ型名として **Vendor** を指定できます。このタイプを指定するコネクションを使用するインテグレーションを作成または編集するときにこのデータタイプ名が表示されます。Fuse Online では、インテグレーション仮想化パネルとデータマッパーにタイプ名が表示されます。
 9. **Data Type Description** フィールドには、このタイプを区別するのに便利な情報を提供します。データマッパーでこのタイプを処理するステップにカーソルを合わせると、この説明が表示されます。
 10. **Done** をクリックします。

インテグレーションフローの最初にコネクションが表示されます。

2.4. AMAZON S3 にデータを追加してインテグレーションを終了

データを Amazon S3 にコピーしてインテグレーションを終了するには、Amazon S3 コネクションをシンプルなインテグレーションの最後のコネクションとして追加します。

前提条件

Amazon S3 コネクションが作成済みである必要があります。

手順

1. インテグレーションの作成を開始します。
2. 最初のコネクションを追加および設定します。
3. **Choose a Finish Connection** ページで、インテグレーションの終了に使用する Amazon S3 コネクションをクリックします。
4. コネクションが実行するアクションをクリックします。
 - **Copy Object** は1つ以上のオブジェクトをバケットに追加します。1つのファイルをバケットに追加するには、その名前を **File Name** フィールドに入力します。

複数のファイルをバケットに追加する場合はファイル名を指定しません。この場合、アクションによって以前のインテグレーションステップから取得されるすべてのオブジェクトが追加されます。

ポーリングを使用して複数のファイルを取得し、ファイル名を指定する場合、**Copy Object** アクションはポーリングで受け取った最後のファイルのみを追加します。

- **Delete Object** はバケットからオブジェクトを削除します。**File Name** フィールドに削除するオブジェクトの名前を指定します。指定したファイルがバケットにない場合、エラーは発生せず、インテグレーションが継続されます。
5. 指定したアクションの設定後、**Next** をクリックしてアクションの入力タイプを指定します。
 6. データタイプの認識が必要がない場合は、**Select Type** フィールドで **Type specification not required** を受け入れ、下部にある **Done** をクリックします。残りの手順を行う必要はありません。
その他の場合は、以下の1つをスキーマタイプとして選択します。
 - **JSON schema** は JSON データの構造を記述するドキュメントです。ドキュメントのメディアタイプは **application/schema+json** です。
 - **JSON instance** は JSON データが含まれるドキュメントです。ドキュメントのメディアタイプは **application/json** です。
 - **XML schema** は XML データの構造を記述するドキュメントです。このドキュメントのファイル拡張子は **.xsd** です。
 - **XML instance** は XML データが含まれるドキュメントです。このドキュメントのファイル拡張子は **.xml** です。
 7. **Definition** 入力ボックスに、選択したスキーマタイプに準拠する定義を貼り付けます。たとえば、**JSON schema** を選択した場合は、メディアタイプが **application/schema+json** の JSON スキーマファイルの内容を貼り付けます。
 8. **Data Type Name** フィールドにはデータタイプに選択した名前を入力します。たとえば、ベンダーの JSON スキーマを指定する場合は、データ型名として **Vendor** を指定できます。
このタイプを指定するコネクションを使用するインテグレーションを作成または編集するときこのデータタイプ名が表示されます。Fuse Online では、インテグレーション仮想化パネルとデータマッパーにタイプ名が表示されます。
 9. **Data Type Description** フィールドには、このタイプを区別するのに便利な情報を提供します。データマッパーでこのタイプを処理するステップにカーソルを合わせると、この説明が表示されます。
 10. **Done** をクリックします。

インテグレーションフローの最後にコネクションが表示されます。

2.5. インテグレーションの途中で AMAZON S3 にデータを追加

インテグレーションの途中で Amazon S3 にデータを追加するには、Amazon S3 コネクションをフローの途中に追加します。

前提条件

- Amazon S3 コネクションが作成済みである必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。

手順

1. フローの途中に使用する Amazon S3 コネクションをクリックします。

2. コネクションが実行するアクションをクリックします。

- **Copy Object** は1つ以上のオブジェクトをバケットに追加します。
1つのファイルをバケットに追加するには、その名前を **File Name** フィールドに入力します。

複数のファイルをバケットに追加する場合はファイル名を指定しません。この場合、アクションによって以前のインテグレーションステップから取得されるすべてのオブジェクトが追加されます。

ポーリングを使用して複数のファイルを取得し、ファイル名を指定する場合、**Copy Object** アクションはポーリングで受け取った最後のファイルのみを追加します。

- **Delete Object** はバケットからオブジェクトを削除します。**File Name** フィールドに削除するオブジェクトの名前を指定します。指定したファイルがバケットにない場合、エラーは発生せず、インテグレーションが継続されます。

3. 指定したアクションの設定後、**Next** をクリックしてアクションの入力タイプを指定します。

4. データタイプの認識が必要がない場合は、**Select Type** フィールドで **Type specification not required** を受け入れ、下部にある **Done** をクリックします。残りの手順を行う必要はありません。

その他の場合は、以下の1つをスキーマタイプとして選択します。

- **JSON schema** は JSON データの構造を記述するドキュメントです。ドキュメントのメディアタイプは **application/schema+json** です。
- **JSON instance** は JSON データが含まれるドキュメントです。ドキュメントのメディアタイプは **application/json** です。
- **XML schema** は XML データの構造を記述するドキュメントです。このドキュメントのファイル拡張子は **.xsd** です。
- **XML instance** は XML データが含まれるドキュメントです。このドキュメントのファイル拡張子は **.xml** です。

5. **Definition** 入力ボックスに、選択したスキーマタイプに準拠する定義を貼り付けます。たとえば、**JSON schema** を選択した場合は、メディアタイプが **application/schema+json** の JSON スキーマファイルの内容を貼り付けます。

6. **Data Type Name** フィールドにはデータタイプに選択した名前を入力します。たとえば、ベンダーの JSON スキーマを指定する場合は、データ型名として **Vendor** を指定できます。
このタイプを指定するコネクションを使用するインテグレーションを作成または編集するとき、このデータタイプ名が表示されます。Fuse Online では、インテグレーション仮想化パネルとデータマッパーにタイプ名が表示されます。

7. **Data Type Description** フィールドには、このタイプを区別するのに便利な情報を提供します。データマッパーでこのタイプを処理するステップにカーソルを合わせると、この説明が表示されます。

8. **Done** をクリックします。

インテグレーションフローで追加した場所にコネクションが表示されます。

第3章 AMQ への接続

インテグレーションでは、Red Hat AMQ ブローカーからメッセージを取得したり、Red Hat AMQ ブローカーへメッセージをパブリッシュすることができます。Red Hat AMQ は、クライアントとメッセージブローカーとの間の通信に OpenWire プロトコルを使用します。以下のブローカータイプと通信するには、Red Hat AMQ コネクタを使用して、対象のブローカーへのコネクションを作成します。

- AMQP をサポートしない Apache ActiveMQ ブローカー
- AMQ 6 ブローカー

以下のブローカータイプの1つと通信するには、[AMQP コネクタを使用](#) して、対象のブローカーへのコネクションを作成します。

- AMQP をサポートする Apache ActiveMQ ブローカー
- Apache ActiveMQ Artemis
- AMQ 7 ブローカー
- EnMasse (オープンソースのメッセージングプラットフォーム)

Red Hat AMQ コネクタを使用するには、以下を参照してください。

- [「AMQ コネクションの作成」](#)
- [「AMQ コネクションを追加してメッセージ受信時にインテグレーションの実行をトリガー」](#)
- [「AMQ メッセージをパブリッシュしてインテグレーションを終了」](#)
- [「インテグレーションの途中で AMQ メッセージをパブリッシュ」](#)

3.1. AMQ コネクションの作成

インテグレーションでは、以下からメッセージを取得したり、以下にメッセージをパブリッシュすることがあります。

- AMQP をサポートしない Apache ActiveMQ ブローカー
- AMQ 6 ブローカー

このような場合、インテグレーションに追加できる Red Hat AMQ コネクションを作成します。

前提条件

接続する Red Hat AMQ ブローカーには以下が必要となります。

- ブローカーの URL
- ユーザーアカウントのクレデンシャル
- ブローカーの PEM 形式の証明書

手順

1. Fuse Online の左パネルで **Connections** をクリックし、利用可能なコネクションを表示します。

2. 右上の **Create Connection** をクリックしてコネクタを表示します。
3. **Red Hat AMQ** コネクタを表示します。
4. 以下を入力してコネクションを設定します。
 - a. **Broker URL** フィールドにデータの送信先またはデータの取得元の場所を入力します (例: **tcp://localhost:61616**)。
 - b. **User Name** フィールドに、このブローカーへアクセスするために使用するアカウントのユーザー名を入力します。
 - c. **Password** フィールドに、このブローカーへアクセスするために使用するアカウントのパスワードを入力します。
 - d. **Client ID** フィールドに、メッセージを見逃さずにコネクションを開閉できるようにする ID を入力します。宛先のタイプはトピックにする必要があります。
 - e. このコネクションが開発環境で使用される場合は、**Check Certificates** を無効にすると時間を一部節約することができます。証明書確認の無効化は開発環境で便利です。セキュアな本番環境では、常に **Check Certificates** を有効にするようにしてください。
 - f. **Broker Certificate** フィールドに Red Hat AMQ ブローカーの PEM 形式の証明書テキストを貼り付けます。これは、証明書確認を無効にした場合以外は必須です。
 - g. **Client Certificate** フィールドに Red Hat クライアントの PEM 形式の証明書テキストを貼り付けます。このフィールドは常に任意です。
5. **Validate** をクリックします。Fuse Online は即座にコネクションを検証しようとし、検証の成功または失敗を示すメッセージを表示します。検証に失敗した場合は、必要に応じて設定の詳細を修正し、再度検証を行います。
6. 検証に成功した場合は **Next** をクリックします。
7. **Connection Name** フィールドに、このコネクションを別のコネクションと区別するために使用する名前を入力します。たとえば、**Red Hat AMQ 1** を入力します。
8. **Description** は任意のフィールドで、このコネクションに関する便利な情報を入力します。たとえば、**Sample Red Hat AMQ connection that uses a provided broker.** を入力します。
9. 右上にある **Create** をクリックし、作成したコネクションが利用可能になったことを確認します。上記の例の名前を入力した場合は、インテグレーションに追加できるコネクションとして **Red Hat AMQ 1** が表示されます。

3.2. AMQ コネクションを追加してメッセージ受信時にインテグレーションの実行をトリガー

Red Hat AMQ ブローカーからメッセージを受信したときにインテグレーションの実行をトリガーするには、Red Hat AMQ コネクションを最初のコネクションとして追加します。

前提条件

メッセージの取得元となる Red Hat AMQ ブローカーへの Red Hat AMQ コネクションが作成されている必要があります。

手順

1. Fuse Online パネルの左側にある **Integrations** をクリックします。
2. **Create Integration** をクリックします。
3. **Choose a Start Connection** ページで、インテグレーションを開始するために使用する Red Hat AMQ コネクションをクリックします。
4. **Choose an Action** ページで、**Subscribe for messages** アクションをクリックし、指定したキューまたはトピックからメッセージを受信します。
5. アクションを設定するには以下を行います。
 - a. **Destination Name** フィールドに、データの受信元となるキューまたはトピックの名前を入力します。
 - b. **Destination Type** では **Queue** を許可するか、**Topic** を選択します。
 - c. メッセージを見逃さずにコネクションを開閉できるようにするため、**Durable Subscription ID** フィールドに永続サブスクリプション ID を入力します。宛先のタイプはトピックにする必要があります。
 - d. 特定の条件を満たすデータのみを受信する場合は、**Message Selector** フィールドにフィルター式を入力します。
メッセージセクターは式が含まれる文字列です。式の構文は、SQL92 条件式構文のサブセットを基にしています。以下の例のメッセージセクターは、値が **Sports** または **Opinion** に設定された **NewsType** プロパティがあるメッセージをすべて選択します。

NewsType = 'Sports' OR NewsType = 'Opinion'

メッセージコンシューマーは、ヘッダーとプロパティがメッセージセクターの式と一致するメッセージのみを受信します。メッセージセクターは、メッセージのボディー部の内容を基にしてメッセージを選択することはできません。

6. **Next** をクリックして、アクションの出力タイプを指定します。
7. データタイプの認識が必要がない場合は、**Select Type** フィールドで **Type specification not required** を受け入れ、下部にある **Done** をクリックします。残りの手順を行う必要はありません。
その他の場合は、以下の1つをスキーマタイプとして選択します。
 - **JSON schema** は JSON データの構造を記述するドキュメントです。ドキュメントのメディアタイプは **application/schema+json** です。
 - **JSON instance** は JSON データが含まれるドキュメントです。ドキュメントのメディアタイプは **application/json** です。
 - **XML schema** は XML データの構造を記述するドキュメントです。このドキュメントのファイル拡張子は **.xsd** です。
 - **XML instance** は XML データが含まれるドキュメントです。このドキュメントのファイル拡張子は **.xml** です。
8. **Definition** 入力ボックスに、選択したスキーマタイプに準拠する定義を貼り付けます。たとえば、**JSON schema** を選択した場合は、メディアタイプが **application/schema+json** の JSON スキーマファイルの内容を貼り付けます。

○ **Destination Name** フィールドには、データのタイプに選択した名前を入力します。たとえば、**news**

9. **Data Type Name** フィールドにはデータタイプに選択した名前を入力します。たとえば、ベンダーの JSON スキーマを指定する場合は、**Vendor** をデータタイプ名として指定することができます。
このタイプを指定するコネクションを使用するインテグレーションを作成または編集するとき、このデータタイプ名が表示されます。Fuse Online では、インテグレーション仮想化パネルとデータマッパーにタイプ名が表示されます。
10. **Data Type Description** フィールドには、このタイプを区別するのに便利な情報を提供します。データマッパーでこのタイプを処理するステップにカーソルを合わせると、この説明が表示されます。
11. **Done** をクリックします。

結果

インテグレーションフローの最初にコネクションが表示されます。

3.3. AMQ メッセージをパブリッシュしてインテグレーションを終了

メッセージを Red Hat AMQ ブローカーにパブリッシュしてシンプルなインテグレーションを終了するには、Red Hat AMQ コネクションをシンプルなインテグレーションの最後のコネクションとして追加します。

前提条件

- メッセージのパブリッシュ先となる Red Hat AMQ ブローカーへのコネクションが作成されている必要があります。
- インテグレーションを作成することになり、Fuse Online は最後のコネクションの選択を要求します。または、インテグレーションを変更して最後のコネクションを変更することになります。

手順

1. **Choose a Finish Connection** ページで、インテグレーションの終了に使用する Red Hat AMQ コネクションをクリックします。
2. **Choose an Action** ページで、**Publish messages** アクションをクリックし、指定したキューまたはトピックにメッセージをパブリッシュします。
3. **Destination Name** フィールドに、メッセージの送信先となるキューまたはトピックの名前を入力します。
4. **Destination Type** では **Queue** を許可するか、**Topic** を選択します。
5. **Persistent** を選択し、コネクションが失敗した場合でもメッセージが確実に送信されるようにします。
6. **Next** をクリックし、アクションの入力および出力タイプを指定します。
7. データタイプの認識が必要がない場合は、**Select Type** フィールドで **Type specification not required** を受け入れ、下部にある **Done** をクリックします。残りの手順を行う必要はありません。
その他の場合は、以下の1つをスキーマタイプとして選択します。

- **JSON schema** は JSON データの構造を記述するドキュメントです。ドキュメントのメディアタイプは **application/schema+json** です。
 - **JSON instance** は JSON データが含まれるドキュメントです。ドキュメントのメディアタイプは **application/json** です。
 - **XML schema** は XML データの構造を記述するドキュメントです。このドキュメントのファイル拡張子は **.xsd** です。
 - **XML instance** は XML データが含まれるドキュメントです。このドキュメントのファイル拡張子は **.xml** です。
8. **Definition** 入力ボックスに、選択したスキーマタイプに準拠する定義を貼り付けます。たとえば、**JSON schema** を選択した場合は、メディアタイプが **application/schema+json** の JSON スキーマファイルの内容を貼り付けます。
 9. **Data Type Name** フィールドにはデータタイプに選択した名前を入力します。たとえば、ベンダーの JSON スキーマを指定する場合は、**Vendor** をデータタイプ名として指定することができます。
このタイプを指定するコネクションを使用するインテグレーションを作成または編集するときこのデータタイプ名が表示されます。Fuse Online では、インテグレーション仮想化パネルとデータマッパーにタイプ名が表示されます。
 10. **Data Type Description** フィールドには、このタイプを区別するのに便利な情報を提供します。データマッパーでこのタイプを処理するステップにカーソルを合わせると、この説明が表示されます。
 11. **Done** をクリックします。

結果

インテグレーションフローの最後にコネクションが表示されます。

3.4. インテグレーションの途中で AMQ メッセージをパブリッシュ

インテグレーションの途中で Red Hat AMQ ブローカーにメッセージをパブリッシュするには、Red Hat AMQ コネクションをフローの途中で追加します。

前提条件

- メッセージのパブリッシュ先となる Red Hat AMQ ブローカーへのコネクションが作成されている必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。または、Fuse Online で最後のコネクションを選択するよう要求されます。

手順

1. フローの途中で追加する Red Hat AMQ コネクションをクリックします。
2. **Choose an Action** ページで、以下のアクションの1つを選択します。
 - **Publish messages** アクションは指定するキューまたはトピックにメッセージをパブリッシュします。このアクションを設定するには、以下を行います。
 - a. **Destination Name** フィールドに、メッセージの送信先となるキューまたはトピックの名前を入力します。

- b. **Destination Type** では **Queue** を許可するか、**Topic** を選択します。
- c. **Persistent** を選択し、コネクションが失敗した場合でもメッセージが確実に送信されるようにします。
- **Request response using messages** で、応答の受信に指定した JMS 宛先にメッセージを送ります。このアクションを設定するには、以下を行います。
 - a. **Destination Name** フィールドに、メッセージの送信先となるキューまたはトピックの名前を入力します。
 - b. **Destination Type** では **Queue** を許可するか、**Topic** を選択します。
 - c. 特定の条件を満たす応答のみを受信する場合は、**Message Selector** フィールドにフィルター式を入力します。
 メッセージセレクトは式が含まれる文字列です。式の構文は、SQL92 条件式構文のサブセットを基にしています。以下の例のメッセージセレクトは、値が **Sports** または **Opinion** に設定された **NewsType** プロパティがあるメッセージをすべて選択します。
NewsType = 'Sports' OR NewsType = 'Opinion'
 メッセージコンシューマーは、ヘッダーとプロパティがメッセージセレクトの式と一致するメッセージのみを受信します。メッセージセレクトは、メッセージのボディ部の内容を基にしてメッセージを選択することはできません。
 - d. **Named Reply To** フィールドには、キューまたはトピックの名前を入力します。宛先によって、応答がこのキューまたはトピックに送信されます。
 - e. **Persistent** を選択し、コネクションが失敗した場合でもメッセージが確実に送信されるようにします。
 - f. **Response Time Out** フィールドには、ランタイム例外をスローする前にこのコネクションが応答メッセージを待つ期間 (ミリ秒単位) を指定します。デフォルトは 5000 ミリ秒 (5 秒) です。
- 3. **Next** をクリックし、アクションの入力タイプを指定してから、アクションの出力タイプを指定します。
- 4. データタイプの認識が必要がない場合は、**Select Type** フィールドで **Type specification not required** を受け入れ、下部にある **Done** をクリックします。残りの手順を行う必要はありません。
 その他の場合は、以下の1つをスキーマタイプとして選択します。
 - **JSON schema** は JSON データの構造を記述するドキュメントです。ドキュメントのメディアタイプは **application/schema+json** です。
 - **JSON instance** は JSON データが含まれるドキュメントです。ドキュメントのメディアタイプは **application/json** です。
 - **XML schema** は XML データの構造を記述するドキュメントです。このドキュメントのファイル拡張子は **.xsd** です。
 - **XML instance** は XML データが含まれるドキュメントです。このドキュメントのファイル拡張子は **.xml** です。

5. **Definition** 入力ボックスに、選択したスキーマタイプに準拠する定義を貼り付けます。たとえば、**JSON schema** を選択した場合は、メディアタイプが **application/schema+json** の JSON スキーマファイルの内容を貼り付けます。
6. **Data Type Name** フィールドにはデータタイプに選択した名前を入力します。たとえば、ベンダーの JSON スキーマを指定する場合は、**Vendor** をデータタイプ名として指定することができます。
このタイプを指定するコネクションを使用するインテグレーションを作成または編集するときにこのデータタイプ名が表示されます。Fuse Online では、インテグレーション仮想化パネルとデータマッパーにタイプ名が表示されます。
7. **Data Type Description** フィールドには、このタイプを区別するのに便利な情報を提供します。データマッパーでこのタイプを処理するステップにカーソルを合わせると、この説明が表示されます。
8. **Done** をクリックします。

結果

インテグレーションフローで追加した場所にコネクションが表示されます。

第4章 AMQP への接続

インテグレーションでは、AMQP (Advanced Message Queue Protocol) ブローカーからメッセージを取得したり、AMQP ブローカーへメッセージをパブリッシュすることができます。AMQP は、クライアントとメッセージブローカーとの間の通信を定義します。以下のブローカータイプで通信するには、AMQP コネクタを使用して対象のブローカーへのコネクションを作成します。

- AMQP をサポートする Apache ActiveMQ ブローカー
- Apache ActiveMQ Artemis
- AMQ 7 ブローカー
- EnMasse (オープンソースのメッセージングプラットフォーム)

以下のブローカータイプの1つと通信するには、[Red Hat AMQ コネクタを使用](#) して、対象のブローカーへのコネクションを作成します。

- AMQP をサポートしない Apache ActiveMQ ブローカー
- AMQ 6 ブローカー



注記

AMQP コネクタを使用して AMQP をサポートしない Apache ActiveMQ ブローカーへのコネクションや AMQ 6 ブローカーへのコネクションを作成できます。これを行うには、ブローカーのトランスポートを設定する必要があります。ブローカーの設定に関する情報は、[『Red Hat JBoss A-MQ Managing and Monitoring Brokers』の「Adding Client Connection Points」](#) を参照してください。指定する設定値に関する情報は、[『Red Hat JBoss A-MQ Connection Reference』の「Advanced Message Queuing Protocol \(AMQP\)」](#) を参照してください。

AMQP コネクタを使用するには、以下を参照してください。

- [「AMQP コネクションの作成」](#)
- [「AMQP コネクションを追加してメッセージ受信時にインテグレーションの実行をトリガー」](#)
- [「フローの途中で AMQP へメッセージをパブリッシュまたは AMQP へメッセージをパブリッシュしてインテグレーションを終了」](#)

4.1. AMQP コネクションの作成

インテグレーションで AMQP ブローカーからメッセージを取得したり AMQP ブローカーへメッセージをパブリッシュする場合は、インテグレーションに追加できる AMQP コネクションを作成します。

前提条件

接続する AMQP ブローカーに以下が必要となります。

- URI
- ユーザーアカウントのクレデンシャル
- PEM 形式の証明書テキスト

手順

1. Fuse Online の左パネルで **Connections** をクリックし、利用可能なコネクションを表示します。
2. 右上の **Create Connection** をクリックしてコネクタを表示します。
3. **AMQP Message Broker** コネクタをクリックします。
4. 以下を入力してコネクションを設定します。
 - a. **Connection URI** フィールドにデータの送信先または取得元となる場所を入力します。
 - b. **User Name** フィールドに、このブローカーへアクセスするために使用するアカウントのユーザー名を入力します。
 - c. **Password** フィールドに、このブローカーへアクセスするために使用するアカウントのパスワードを入力します。
 - d. **Client ID** フィールドに、メッセージを見逃さずにコネクションを開閉できるようにする ID を入力します。宛先のタイプはトピックにする必要があります。
 - e. このコネクションが開発環境で使用される場合は、**Check Certificates** を無効にすると時間を一部節約することができます。証明書確認の無効化は開発環境で便利です。セキュアな本番環境では、常に **Check Certificates** を有効にするようにしてください。
 - f. **Broker Certificate** フィールドにブローカーの PEM 形式の証明書テキストを貼り付けます。これは、証明書確認を無効にした場合以外は必須です。
 - g. **Client Certificate** フィールドにクライアントの PEM 形式の証明書テキストを貼り付けます。このフィールドは常に任意です。
5. **Validate** をクリックします。Fuse Online は即座にコネクションを検証しようとし、検証の成功または失敗を示すメッセージを表示します。検証に失敗した場合は、必要に応じて設定の詳細を修正し、再度検証を行います。
6. 検証に成功した場合は **Next** をクリックします。
7. **Connection Name** フィールドに、このコネクションを別のコネクションと区別するために使用する名前を入力します。たとえば、**AMQP 1** を入力します。
8. **Description** は任意のフィールドで、このコネクションに関する便利な情報を入力します。たとえば、**Sample AMQP connection** を入力します。
9. 右上にある **Create** をクリックし、作成したコネクションが利用可能になったことを確認します。上記の例の名前を入力した場合は、インテグレーションに追加できるコネクションとして **AMQP1** が表示されます。

4.2. AMQP コネクションを追加してメッセージ受信時にインテグレーションの実行をトリガー

AMQP ブローカーからメッセージを受信したときにインテグレーションの実行をトリガーするには、AMQP コネクションをインテグレーションの最初のコネクションとして追加します。

前提条件

- メッセージの受信元となる AMQP ブローカーへの接続が作成されている必要があります。
- インテグレーションを作成することになります。Fuse Online は最初の接続の選択を要求します。

手順

1. **Choose a Start Connection** ページで、インテグレーションを開始するために使用する AMQP コネクションをクリックします。
2. **Choose an Action** ページで、**Subscribe for messages** アクションをクリックし、指定したキューまたはトピックからメッセージを受信します。
3. アクションを設定するには以下を行います。

- a. **Destination Name** フィールドに、データの受信元となるキューまたはトピックの名前を入力します。
- b. **Destination Type** では **Queue** を許可するか、**Topic** を選択します。
- c. メッセージを見逃さずに接続を開閉できるようにするため、**Durable Subscription ID** フィールドに永続サブスクリプション ID を入力します。宛先のタイプはトピックにする必要があります。
- d. 特定の条件を満たすデータのみを受信する場合は、**Message Selector** フィールドにフィルター式を入力します。
メッセージセレクトーは式が含まれる文字列です。式の構文は、SQL92 条件式構文のサブセットを基にしています。以下の例のメッセージセレクトーは、値が **Sports** または **Opinion** に設定された **NewsType** プロパティがあるメッセージをすべて選択します。

NewsType = 'Sports' OR NewsType = 'Opinion'

メッセージコンシューマーは、ヘッダーとプロパティがメッセージセレクトーの式と一致するメッセージのみを受信します。メッセージセレクトーは、メッセージのボディ部の内容を基にしてメッセージを選択することはできません。

4. **Next** をクリックして、アクションの出力タイプを指定します。
5. データタイプの認識が必要がない場合は、**Select Type** フィールドで **Type specification not required** を受け入れ、下部にある **Done** をクリックします。残りの手順を行う必要はありません。
その他の場合は、以下の1つをスキーマタイプとして選択します。

- **JSON schema** は JSON データの構造を記述するドキュメントです。ドキュメントのメディアタイプは **application/schema+json** です。
- **JSON instance** は JSON データが含まれるドキュメントです。ドキュメントのメディアタイプは **application/json** です。
- **XML schema** は XML データの構造を記述するドキュメントです。このドキュメントのファイル拡張子は **.xsd** です。
- **XML instance** は XML データが含まれるドキュメントです。このドキュメントのファイル拡張子は **.xml** です。

6. **Definition** 入力ボックスに、選択したスキーマタイプに準拠する定義を貼り付けます。たとえば、**JSON schema** を選択した場合は、メディアタイプが **application/schema+json** の JSON スキーマファイルの内容を貼り付けます。
7. **Data Type Name** フィールドにはデータタイプに選択した名前を入力します。たとえば、ベンダーの JSON スキーマを指定する場合は、**Vendor** をデータタイプ名として指定することができます。
このタイプを指定するコネクションを使用するインテグレーションを作成または編集するときにこのデータタイプ名が表示されます。Fuse Online では、インテグレーション仮想化パネルとデータマッパーにタイプ名が表示されます。
8. **Data Type Description** フィールドには、このタイプを区別するのに便利な情報を提供します。データマッパーでこのタイプを処理するステップにカーソルを合わせると、この説明が表示されます。
9. **Done** をクリックします。

結果

インテグレーションフローの最初にコネクションが表示されます。

4.3. フローの途中で AMQP ヘッセージをパブリッシュまたは AMQP ヘッセージをパブリッシュしてインテグレーションを終了

フローの途中で AMQP ブローカーヘメッセージをパブリッシュしたり、AMQP ブローカーヘメッセージをパブリッシュしてシンプルなインテグレーションを終了することができます。これには、AMQP コネクションをフローの途中で追加するか、AMQP コネクションをシンプルなインテグレーションの最後のコネクションとして追加します。

前提条件

- メッセージのパブリッシュ先となる AMQP ブローカーへのコネクションが作成されている必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。または、Fuse Online でシンプルなインテグレーションの最後のコネクションを選択するよう要求されます。

手順

1. メッセージのパブリッシュに使用する AMQP コネクションをクリックします。
2. **Choose an Action** ページで、以下のアクションの1つを選択します。
 - **Publish messages** で応答を受信せずに、指定するキューまたはトピックヘメッセージをパブリッシュします。このアクションを設定するには、以下を行います。
 - a. **Destination Name** フィールドに、メッセージの送信先となるキューまたはトピックの名前を入力します。
 - b. **Destination Type** では **Queue** を許可するか、**Topic** を選択します。
 - c. **Persistent** を選択し、コネクションが失敗した場合でもメッセージが確実に送信されるようにします。

- **Request response using messages** で、指定するキューまたはトピックへメッセージをパブリッシュし、応答を受信します。



注記

シンプルなインテグレーションの最後の接続が **Request response using messages** アクションを実行する AMQP 接続である場合、接続はメッセージをパブリッシュしますが、応答は破棄されます。応答を保持するには、**Request response using messages** アクションを実行する途中の接続として AMQP 接続を追加し、ログステップでシンプルなインテグレーションを終了します。

このアクションを設定するには、以下を行います。

- Destination Name** フィールドに、メッセージの送信先となるキューまたはトピックの名前を入力します。
- Destination Type** では **Queue** を許可するか、**Topic** を選択します。
- メッセージを見逃さずに接続を開閉できるようにするため、**Durable Subscription ID** フィールドに永続サブスクリプション ID を入力します。宛先のタイプはトピックにする必要があります。
- 特定の条件を満たす応答のみを受信する場合は、**Message Selector** フィールドにフィルター式を入力します。
メッセージセレクトは式が含まれる文字列です。式の構文は、SQL92 条件式構文のサブセットを基にしています。以下の例のメッセージセレクトは、値が **Sports** または **Opinion** に設定された **NewsType** プロパティがあるメッセージをすべて選択します。

NewsType = 'Sports' OR NewsType = 'Opinion'

メッセージコンシューマーは、ヘッダーとプロパティがメッセージセレクトの式と一致するメッセージのみを受信します。メッセージセレクトは、メッセージのボディ部の内容を基にしてメッセージを選択することはできません。

3. **Next** をクリックし、アクションの入力および出力タイプを指定します。
4. データタイプの認識が必要がない場合は、**Select Type** フィールドで **Type specification not required** を受け入れ、下部にある **Done** をクリックします。残りの手順を行う必要はありません。
その他の場合は、以下の1つをスキーマタイプとして選択します。

- **JSON schema** は JSON データの構造を記述するドキュメントです。ドキュメントのメディアタイプは **application/schema+json** です。
- **JSON instance** は JSON データが含まれるドキュメントです。ドキュメントのメディアタイプは **application/json** です。
- **XML schema** は XML データの構造を記述するドキュメントです。このドキュメントのファイル拡張子は **.xsd** です。
- **XML instance** は XML データが含まれるドキュメントです。このドキュメントのファイル拡張子は **.xml** です。

5. **Definition** 入力ボックスに、選択したスキーマタイプに準拠する定義を貼り付けます。たとえば、JSON schema を選択した場合は、メディアタイプが **application/schema+json** の JSON スキーマファイルの内容を貼り付けます。
6. **Data Type Name** フィールドにはデータタイプに選択した名前を入力します。たとえば、ベンダーの JSON スキーマを指定する場合は、**Vendor** をデータタイプ名として指定することができます。
このタイプを指定するコネクションを使用するインテグレーションを作成または編集するとき、このデータタイプ名が表示されます。Fuse Online では、インテグレーション仮想化パネルとデータマッパーにタイプ名が表示されます。
7. **Data Type Description** フィールドには、このタイプを区別するのに便利な情報を提供します。データマッパーでこのタイプを処理するステップにカーソルを合わせると、この説明が表示されます。
8. **Done** をクリックします。

結果

インテグレーションフローで追加した場所にコネクションが表示されます。

第5章 DROPBOX への接続

インテグレーションでは、Dropbox からファイルをダウンロードしたり、Dropbox にファイルをアップロードすることができます。詳細は以下を参照してください。

- 「Fuse Online を Dropbox クライアントとして登録」
- 「Dropbox コネクションの作成」
- 「Dropbox からファイルを取得してインテグレーションの実行をトリガー」
- 「Dropbox にファイルを追加してインテグレーションを終了」
- 「インテグレーションの途中で Dropbox にアクセス」

5.1. FUSE ONLINE を DROPBOX クライアントとして登録

Fuse Online 環境を Dropbox にアクセスできるクライアントアプリケーションとして登録する必要があります。これにより、Dropbox に接続するインテグレーションをいくつでも作成することができます。そのため、特定の Fuse Online 環境を1度だけ Dropbox に登録する必要があります。

Fuse Online を Dropbox クライアントアプリケーションとして登録できるのは、Fuse Online 環境ごとに1回のみです。しかし、各 Dropbox コネクションは同じ登録を使用しますが、異なるユーザークレデンシャルを使用することができます。

前提条件

ファイルのダウンロードまたはアップロードでインテグレーションが使用する Dropbox アカウントにログインする必要があります。

手順

1. Fuse Online で以下を行います。
 - a. 左側のパネルで **Settings** をクリックします。
 - b. **Settings** ページで、上部付近にあるコールバック URL の右側で  をクリックし、Fuse Online 環境のコールバック URL をクリップボードにコピーします。この手順の最後の方でこの URL が必要になります。
2. 別のブラウザタブで <https://www.dropbox.com> にアクセスし、以下を行います。
 - a. インテグレーションでアクセスするデータがある Dropbox アカウントにログインします。
 - b. ログインした後に <https://www.dropbox.com/developers/apps> にアクセスします。
 - c. **Create App** をクリックします。
 - d. **Dropbox API** を選択します。
 - e. この手順の最初にクリップボードにコピーした URL を、ページの上部近くにある **During registration, enter this callback URL:** で始まる文に貼り付けます。たとえば、貼り付ける URL は `https://app-proj9128.7b63.fuse-ignite.openshiftapps.com/api/v1/credentials/callback` のようになります。

- f. Fuse Online が単一のフォルダーにアクセスするか、またはすべてのフォルダーやファイルにアクセスするかを選択します。
- g. Dropbox アプリケーションの名前を指定します。たとえば、**Fuse Online Access From Aslan LLC** を指定します。Dropbox アプリケーションで一意的な名前を指定する必要があります。
- h. ボックスにチェックマークを入れ、Dropbox API の利用規約に同意します。
- i. **Create App** をクリックします。
- j. 新しいアプリケーションの Dropbox **Settings** ページにある **OAuth2 Redirect URLs** の入力フィールドに、この手順の始めにクリップボードにコピーした Fuse Online URL を貼り付けます。
- k. **Add** をクリックします。

結果

これで Fuse Online 環境が Dropbox クライアントとして登録され、Fuse Online はログインした Dropbox アカウントのコンテンツにアクセスできるようになります。

5.2. DROPBOX コネクションの作成

インテグレーションで Dropbox ファイルをダウンロードまたはアップロードするには、インテグレーションに追加できる Dropbox コネクションを作成します。同じコネクションを任意の数のインテグレーションに追加できます。

前提条件

Fuse Online 環境が Dropbox にアクセスできるアプリケーションとして登録されている必要があります。

手順

1. 新しいブラウザタブで <https://www.dropbox.com> にアクセスし、以下を行います。
 - a. Fuse Online 環境からのアクセスを登録するアプリケーションを作成した Dropbox アカウントにログインします。
 - b. <https://www.dropbox.com/developers/apps> にアクセスします。
 - c. Fuse Online アプリケーションをクリックし、その設定を表示します。
2. Fuse Online の別のブラウザタブで以下を行います。
 - a. 左側のパネルで **Connections** をクリックし、利用できるコネクションを表示します。
 - b. 右上の **Create Connection** をクリックし、利用可能なコネクターを表示します。
 - c. **Dropbox** コネクターをクリックします。
3. アプリケーションの Dropbox 設定表示に戻り、以下を行います。
 - a. 下にスクロールして **Generated Access Token** を確認します。
 - b. **Generate** をクリックします。

- c. 生成されたアクセストークンをクリップボードにコピーします。
4. Fuse Online に戻り、**Configure Connection** ページの **Access Token** フィールドに生成されたアクセストークンを貼り付けます。
5. **Client Identifier** フィールドに Dropbox アプリケーションを作成したときに指定した名前を入力します。
6. **Validate** をクリックします。Fuse Online はこのコネクションを検証できるかを示すメッセージを表示します。検証に失敗した場合は、入力した値が正しいことを確認し、再度検証を行ってください。
7. 検証に成功したら、右上の **Next** をクリックします。
8. **Connection Name** フィールドに、このコネクションを別のコネクションと区別するために使用する名前を入力します。たとえば、**Dropbox Connect 1** を入力します。
9. **Description** は任意のフィールドで、このコネクションに関する便利な情報を入力します。たとえば、**Sample Dropbox connection that can access all content in our company Dropbox account.** を入力します。
10. 右上にある **Create** をクリックし、作成したコネクションが利用可能になったことを確認します。上記の例の名前を入力した場合は、インテグレーションに追加できるコネクションとして **Dropbox Connect 1** が表示されます。

5.3. DROPBOX からファイルを取得してインテグレーションの実行をトリガー

Dropbox からファイルをダウンロードしてインテグレーションを開始するには、Dropbox コネクションを最初のコネクションとして追加します。

前提条件

Dropbox コネクションが作成されている必要があります。

手順

1. Fuse Online パネルの左側にある **Integrations** をクリックします。
2. **Create Integration** をクリックします。
3. **Choose a Start Connection** ページで、インテグレーションを開始するために使用する Dropbox コネクションをクリックします。
4. **Choose an Action** ページで **Download** アクションをクリックし、このコネクションがアクセスする Dropbox アカウントから1つまたは複数のファイルを取得します。
5. アクションを設定するには、**Folder or file name path to download** フィールドにインテグレーションが取得するコンテンツのファイル名のパスを指定します。本リリースでは、単一のファイルのみをダウンロードできます。
6. **Next** をクリックして、アクションの出力タイプを指定します。
7. データタイプの認識が必要がない場合は、**Select Type** フィールドで **Type specification not required** を受け入れ、下部にある **Done** をクリックします。残りの手順を行う必要はありません。

その他の場合は、以下の1つをスキーマタイプとして選択します。

- **JSON schema** は JSON データの構造を記述するドキュメントです。ドキュメントのメディアタイプは **application/schema+json** です。
 - **JSON instance** は JSON データが含まれるドキュメントです。ドキュメントのメディアタイプは **application/json** です。
 - **XML schema** は XML データの構造を記述するドキュメントです。このドキュメントのファイル拡張子は **.xsd** です。
 - **XML instance** は XML データが含まれるドキュメントです。このドキュメントのファイル拡張子は **.xml** です。
8. **Definition** 入力ボックスに、選択したスキーマタイプに準拠する定義を貼り付けます。たとえば、**JSON schema** を選択した場合は、メディアタイプが **application/schema+json** の JSON スキーマファイルの内容を貼り付けます。
 9. **Data Type Name** フィールドにはデータタイプに選択した名前を入力します。たとえば、ベンダーの JSON スキーマを指定する場合は、データ型名として **Vendor** を指定できます。このタイプを指定するコネクションを使用するインテグレーションを作成または編集するときにこのデータタイプ名が表示されます。Fuse Online では、インテグレーション仮想化パネルとデータマッパーにタイプ名が表示されます。
 10. **Data Type Description** フィールドには、このタイプを区別するのに便利な情報を提供します。データマッパーでこのタイプを処理するステップにカーソルを合わせると、この説明が表示されます。
 11. **Done** をクリックします。

結果

インテグレーションフローの最初にコネクションが表示されます。

5.4. DROPBOX にファイルを追加してインテグレーションを終了

ファイルを Dropbox にアップロードしてインテグレーションを終了するには、Dropbox コネクションをインテグレーションの最後のコネクションとして追加します。

前提条件

- Dropbox コネクションが作成されている必要があります。
- インテグレーションを作成または編集することになります。Fuse Online で最後のコネクションを選択するよう要求されます。

手順

1. **Choose a Finish Connection** ページで、インテグレーションの終了に使用する Dropbox コネクションをクリックします。
2. **Choose an Action** ページで **Upload** アクションをクリックし、このコネクションがアクセスする Dropbox アカウントに現在のインテグレーションデータを追加します。

3. **Remote Path** フィールドにアップロードするファイルのローカルファイル名パスを入力します。Dropbox は同じパスと名前ファイルで格納します。本リリースでは、単一のファイルのみをアップロードできます。
4. **Upload mode** では以下を行います。
 - 同じ Dropbox フォルダーに同じ名前のファイルがない場合のみ **Add** を選択してファイルをアップロードします。同じ名前のファイルが同じ Dropbox フォルダーに存在する場合は、ファイルはアップロードされず、インテグレーションが続行されます。これは、アップロードするファイルのコンテンツが更新されたかどうかに関わらず、同じ動作になります。
 - 同じ名前のファイルが同じ Dropbox フォルダーにある場合でも確実にファイルをアップロードするには **Force** を選択します。既存のファイルはアップロードするファイルによって上書きされます。
5. **Next** をクリックしてアクションの入力タイプを指定します。
6. データタイプの認識が必要がない場合は、**Select Type** フィールドで **Type specification not required** を受け入れ、下部にある **Done** をクリックします。残りの手順を行う必要はありません。
その他の場合は、以下の1つをスキーマタイプとして選択します。
 - **JSON schema** は JSON データの構造を記述するドキュメントです。ドキュメントのメディアタイプは **application/schema+json** です。
 - **JSON instance** は JSON データが含まれるドキュメントです。ドキュメントのメディアタイプは **application/json** です。
 - **XML schema** は XML データの構造を記述するドキュメントです。このドキュメントのファイル拡張子は **.xsd** です。
 - **XML instance** は XML データが含まれるドキュメントです。このドキュメントのファイル拡張子は **.xml** です。
7. **Definition** 入力ボックスに、選択したスキーマタイプに準拠する定義を貼り付けます。たとえば、**JSON schema** を選択した場合は、メディアタイプが **application/schema+json** の JSON スキーマファイルの内容を貼り付けます。
8. **Data Type Name** フィールドにはデータタイプに選択した名前を入力します。たとえば、ベンダーの JSON スキーマを指定する場合は、データ型名として **Vendor** を指定できます。
このタイプを指定するコネクションを使用するインテグレーションを作成または編集するときこのデータタイプ名が表示されます。Fuse Online では、インテグレーション仮想化パネルとデータマッパーにタイプ名が表示されます。
9. **Data Type Description** フィールドには、このタイプを区別するのに便利な情報を提供します。データマッパーでこのタイプを処理するステップにカーソルを合わせると、この説明が表示されます。
10. **Done** をクリックします。

結果

インテグレーションフローの最後にコネクションが表示されます。

5.5. インテグレーションの途中で DROPBOX にアクセス

フローの途中で Dropbox にファイルをアップロードするには、Dropbox コネクションをフローの途中に追加します。

前提条件

- Dropbox コネクションが作成されている必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。

手順

1. インテグレーションが使用する Dropbox コネクションをクリックします。
2. **Choose an Action** ページで **Upload** をクリックし、このコネクションがアクセスする Dropbox アカウントに現在のインテグレーションデータを追加します。このアクションを設定するには、以下を行います。
 - a. **Remote Path** フィールドにアップロードするファイルのローカルパスとファイル名を指定します。Dropbox は同じパスと名前で作成されたファイルに格納します。本リリースでは、単一のファイルのみをアップロードできます。
 - b. **Upload mode** では以下を行います。
 - 同じ Dropbox フォルダーに同じ名前のファイルがない場合のみ **Add** を選択してファイルをアップロードします。同じ名前のファイルが同じ Dropbox フォルダーに存在する場合は、ファイルはアップロードされず、インテグレーションが実行されます。これは、アップロードするファイルのコンテンツが更新されたかどうかに関わらず、同じ動作になります。
 - 同じ名前のファイルが同じ Dropbox フォルダーにある場合でも確実にファイルをアップロードするには **Force** を選択します。既存のファイルはアップロードするファイルによって上書きされます。
3. **Next** をクリックし、アクションの入力および出力タイプを指定します。
4. データタイプの認識が必要がない場合は、**Select Type** フィールドで **Type specification not required** を受け入れ、下部にある **Done** をクリックします。残りの手順を行う必要はありません。
その他の場合は、以下の1つをスキーマタイプとして選択します。
 - **JSON schema** は JSON データの構造を記述するドキュメントです。ドキュメントのメディアタイプは **application/schema+json** です。
 - **JSON instance** は JSON データが含まれるドキュメントです。ドキュメントのメディアタイプは **application/json** です。
 - **XML schema** は XML データの構造を記述するドキュメントです。このドキュメントのファイル拡張子は **.xsd** です。
 - **XML instance** は XML データが含まれるドキュメントです。このドキュメントのファイル拡張子は **.xml** です。
5. **Definition** 入力ボックスに、選択したスキーマタイプに準拠する定義を貼り付けます。たとえば、**JSON schema** を選択した場合は、メディアタイプが **application/schema+json** の JSON スキーマファイルの内容を貼り付けます。

6. **Data Type Name** フィールドにはデータタイプに選択した名前を入力します。たとえば、ベンダーの JSON スキーマを指定する場合は、データ型名として **Vendor** を指定できます。
このタイプを指定するコネクションを使用するインテグレーションを作成または編集するときにこのデータタイプ名が表示されます。Fuse Online では、インテグレーション仮想化パネルとデータマッパーにタイプ名が表示されます。
7. **Data Type Description** フィールドには、このタイプを区別するのに便利な情報を提供します。データマッパーでこのタイプを処理するステップにカーソルを合わせると、この説明が表示されます。
8. **Done** をクリックします。

結果

インテグレーションフローで追加した場所にコネクションが表示されます。

第6章 FHIR への接続

FHIR (Fast Healthcare Interoperability Resources) は医療データを交換するための標準です。インテグレーションでは、1つ以上の FHIR リソースの取得、リソースの作成または更新、リソースの1つ以上のフィールドの更新、またはトランザクションを使用した複数リソースの作成を行うことができます。これには、フローの途中に FHIR コネクションを追加するか、FHIR コネクションをシンプルなインテグレーションの最後のコネクションとして追加します。



重要

FHIR コネクターはテクノロジープレビューの機能です。テクノロジープレビューの機能は、Red Hat の本番環境のサービスレベルアグリーメント (SLA) ではサポートされず、機能的に完全ではないことがあるため、Red Hat は本番環境での使用は推奨しません。Red Hat では、これらについて実稼働環境での使用を推奨していません。テクノロジープレビューの機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストやフィードバックの提供を可能にするために提供されます。Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、<https://access.redhat.com/ja/support/offerings/techpreview> を参照してください。

FHIR への Fuse Online コネクションは以下を行うことができます。

- Basic または Bearer トークン (OAuth) 認証を使用する FHIR サーバーと通信できます。
- FHIR DSTU3 に準拠するすべての FHIR リソースで操作することができます。
- リソースにカスタムデータを格納できるようにするエクステンションでは操作できません。FHIR サーバーから受信したデータにエクステンションがある場合、インテグレーションは何もせずに無視します。

FHIR に接続するインテグレーションでは、データマッパーステップで FHIR リストフィールドに対して1レベルの深さのみをマッピングできます。そのため、FHIR リストにリストが含まれる場合は、入れ子のリストでフィールドをマッピングできません。この制限を軽減するため、Fuse Online は一部の FHIR リソースリストフィールドを単一値のフィールドに変換します。たとえば、リストフィールドのアドレスはデータマッパーで単一のフィールドとして表示される可能性があります。

FHIR への接続に関する詳細は、以下を参照してください。

- [「FHIR サーバーへのコネクションの作成」](#)
- [「FHIR サーバーからのリソースの取得」](#)
- [「リソースの FHIR サーバーのクエリー」](#)
- [「FHIR サーバーでのリソースの作成」](#)
- [「FHIR サーバー上のリソースにあるすべてのフィールドを更新」](#)
- [「FHIR サーバー上のリソースにある指定のフィールドを更新」](#)
- [「FHIR サーバーで異なるタイプのリソースの作成」](#)
- [「FHIR サーバーからのリソースの削除」](#)

6.1. FHIR サーバーへのコネクションの作成

インテグレーションで FHIR リソースを操作するには、FHIR サーバーへのコネクションを作成し、そのコネクションをフローの途中に追加するか、シンプルなインテグレーションの最後のコネクションとして追加します。

FHIR コネクションはシンプルなインテグレーションの最初のコネクションとすることはできません。しかし、FHIR コネクションを定期的にトリガーするタイマーを使ってシンプルなインテグレーションを開始することができます。

前提条件

- 接続する FHIR サーバーの URL を知っている必要があります。
- FHIR サーバーへアクセスするための承認クレデンシャルを持っている必要があります。これらのクレデンシャルは FHIR サーバー管理者から取得する必要があります。まれに、FHIR サーバーが認証を必要とせず、クレデンシャルを指定せずにコネクションを作成できることがあります。たとえば、パブリック FHIR サーバーやプライベートネットワークの FHIR サーバーが認証を必要としないことがあります。

手順

1. Fuse Online の左パネルで **Connections** をクリックし、利用可能なコネクションを表示します。
2. 右上の **Create Connection** をクリックしてコネクタを表示します。
3. **FHIR** コネクタをクリックし、以下のようにコネクションを設定します。
 - a. **FHIR version** フィールドで必要となる **DSTU3** を許可します。
 - b. アクセスする FHIR サーバーの URL を **FHIR server URL** フィールドに入力します。インテグレーションの作成中、デフォルトテストサーバー <http://fhirtest.uhn.ca/baseDstu3> を許可することができます。
 - c. **FHIR server username** フィールドにユーザー名を入力します。これは、Basic 認証を使用する FHIR サーバーで必要になります。
 - d. **FHIR server password** フィールドにパスワードを入力します。これも Basic 認証を使用する FHIR サーバーで必要になります。
 - e. **FHIR server bearer token** フィールドにトークンを入力します。これは、OAuth 認証を使用する FHIR サーバーで必要になります。
4. **Validate** をクリックします。Fuse Online は即座にコネクションを検証しようとし、検証の成功または失敗を示すメッセージを表示します。検証に失敗した場合は、入力パラメーターを訂正し、再度検証を行います。
5. 検証に成功した場合は **Next** をクリックします。
6. **Connection Name** フィールドに、このコネクションを別のコネクションと区別するために使用する名前を入力します。たとえば、**FHIR West** を入力します。
7. **Description** は任意のフィールドで、このコネクションに関する便利な情報を入力します。
8. 右上にある **Create** をクリックし、作成したコネクションが利用可能になったことを確認します。上記の例の名前を入力した場合は、インテグレーションに追加できるコネクションとして **FHIR West** が表示されます。

6.2. FHIR サーバーからのリソースの取得

インテグレーションでは、指定タイプのリソースを1つ取得できます。これには、フローの途中で FHIR コネクションを追加します。

前提条件

- 取得するリソースを持つ FHIR サーバーへのコネクションが作成されている必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。

手順

1. 取得するリソースにアクセスできる FHIR コネクションをクリックします。
2. **Choose an Action** ページで **Read** をクリックします。
3. **Resource Type** フィールドをクリックし、FHIR リソースタイプのリストを表示します。取得するリソースのタイプを選択または入力します。
4. 取得するリソースタイプに他のリソースタイプが含まれている場合、**Contained Resource Types** フィールドで含まれているタイプを選択します。複数のタイプを選択する必要がある場合は **Ctrl** キーを使用します。
含まれるリソースタイプを特定すると、データマッパーは含まれるリソースにあるフィールドを表示できるようになります。含まれるリソースタイプを指定しないと、含まれるリソースにあるフィールドからマッピングを行うことができません。実際に含まれないリソースを選択しても問題はありません。
5. **Resource Id** フィールドにリソースの ID を入力します。ただし、フローの初期のステップからマップする予定である場合は、このフィールドを空白のままにしておきます。
6. **Resource version** フィールドには、リソースのバージョン ID を任意で指定します。最新バージョンのリソースを取得する場合や、以前のステップからリソースのバージョン ID をマップする場合は、このフィールドを空白のままにしておきます。
7. **Next** をクリックして、このコネクションをフローに追加します。

結果

インテグレーションフローで追加した場所にコネクションが表示されます。実行中、このコネクションは1つのリソースを返します。

次のステップ

Resource Id フィールドまたは **Resource version** フィールドに値をマップするには、このコネクションの前にデータマッパーステップを追加します。

6.3. リソースの FHIR サーバーのクエリー

インテグレーションでは、FHIR サーバーにクエリーを行って、指定のクエリーを満たす特定の FHIR リソースのインスタンスを取得できます。たとえば、65 歳以上の患者全員が肺炎の予防接種を受けるようにするには、1955 年より前に誕生した患者のリソースを取得するクエリーを指定します。1955 年より前に禁止されていた患者のリソースを取得するクエリーを指定します。これには、フローの途中で FHIR コネクションを追加します。

前提条件

- 取得するリソースを持つ FHIR サーバーへの接続が作成されている必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。

手順

1. 取得するリソースを持つ FHIR コネクションをクリックします。
2. **Choose an Action** ページで **Search** をクリックします。
3. **Resource Type** フィールドをクリックし、FHIR リソースタイプのリストを表示します。取得するリソースのタイプを選択または入力します。
4. **Query** フィールドに FHIR クエリーを指定するか、このフィールドを空白のままにして以前のステップからクエリーをマッピングします。FHIR クエリーの作成に関する詳細は、「[FHIR Release 3 Search](#)」を参照してください。
5. **Next** をクリックして、このコネクションをフローに追加します。

結果

インテグレーションフローで追加した場所にコネクションが表示されます。実行中、このコネクションは指定のクエリーを満たすリソースのリストを返します。Fuse Online はこのリストをコレクションとして扱います。

次のステップ

クエリーを指定しなかった場合は、このコネクションの前にデータマッパーステップを追加し、クエリーをこのコネクションにマップします。返されたリストのコレクションではなく、返された個々のリソースで操作を行うには、フローのこのコネクションの後に分割ステップ (split step) を追加します。

6.4. FHIR サーバーでのリソースの作成

インテグレーションではリソースを作成し、FHIR サーバーに追加することができます。コネクションの **Create** アクションを設定するときに、作成するリソースのタイプを指定します。たとえば、新しい患者のリソースを FHIR サーバーに追加する場合があります。リソースを作成するには、フローの途中に FHIR コネクションを追加するか、FHIR コネクションをシンプルなインテグレーションの最後のコネクションとして追加します。

新しいリソースに値を入力するには、このステップを追加する FHIR コネクションのすぐ前にデータマッパーステップを追加します。フローの以前のステップは、新しいリソースに含まれるデータを提供することが想定されます。この FHIR コネクションが作成するターゲットリソースのフィールドに、以前のステップのフィールドをマップします。新しいリソースにはマップしたフィールドのみが含まれます。

コネクションが **Create** アクションを実行して FHIR サーバー上で新しいリソースを作成する場合、新しいリソースには自動生成された FHIR リソース ID が割り当てられます。新しいリソースのリソース ID を指定する場合は、**Create** アクションを選択する代わりに、コネクションが実行する **Update アクション** を選択します。

前提条件

- リソースを追加する FHIR サーバーへの接続が作成されている必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。または、最後のコネクションを追加するよう要求されます。

手順

1. リソースを追加するサーバーへの FHIR コネクションをクリックします。
2. **Choose an Action** ページで **Create** をクリックします。
3. **Resource Type** フィールドをクリックし、FHIR リソースタイプのリストを表示します。作成するリソースのタイプを選択または入力します。
4. 作成するリソースタイプに他のリソースタイプが含まれている場合、**Contained Resource Types** フィールドで含まれているタイプを選択します。複数のタイプを選択する必要がある場合は **Ctrl** キーを使用します。
含まれるリソースタイプを特定すると、データマッパーは含まれるリソースにあるフィールドを表示できるようになります。含まれるリソースタイプを指定しないと、含まれるリソースにあるフィールドをマップ元またはマップ先としてマッピングすることができません。実際に含まれないリソースを選択しても問題はありません。
5. **Next** をクリックして、このコネクションをフローに追加します。

結果

インテグレーションフローで追加した場所にコネクションが表示されます。実行中、コネクションは FHIR サーバー上に新しいリソースを作成します。コネクションは、新しいリソースのリソース ID が含まれる **MethodOutcome** リソースを返します。これをフローの後続のステップでマップすることができます。

次のステップ

このフローに、新しいリソースに含まれるデータを取得する追加のステップが必要な場合、この FHIR コネクションの前に追加します。これらのステップが追加されたら、新しいリソースを作成する FHIR コネクションの直前にデータマッパーステップを追加します。データマッパーステップで、作成されるリソースのフィールドへマップします。

注意

新しいリソースを作成する FHIR コネクションの直前にデータマッパーステップがないと、コネクションは空のリソースの作成を試みます。空のリソースが作成されるかどうかは、FHIR サーバーの設定によります。

6.5. FHIR サーバー上のリソースにあるすべてのフィールドを更新

インテグレーションでは、FHIR サーバー上のリソースを更新できます。更新された値を提供するには、リソースを更新する FHIR コネクションの直前にデータマッパーステップを追加します。フローの以前のステップは、更新したリソースに含まれるデータを提供することが想定されます。この FHIR コネクションが更新するターゲットリソースのフィールドに、以前のステップのフィールドをマップします。

注意

更新されたリソースには、マップ先のフィールドのみが含まれます。そのため、値が変更するフィールドをマップする他に、リソースに含まれるようにする値の変更しないフィールドをマップする必要があります。特定のリソースフィールドをマップしないと、コネクションは更新されたリソースからそのフィールドを削除します。

以前のステップのリソース ID をこのコネクションのリソース ID にマップするようにしてください。これはコネクションが変更しない唯一のリソースフィールドになります。FHIR サーバーに、更新される

リソースのリソース ID を持つリソースがない場合、コネクションはそのリソース ID で新しいリソースを作成します。これは、ユーザーが選択したリソース ID で新しいリソースを追加する唯一の方法になります。

リソースを更新または作成するには、FHIR コネクションをフローの途中に追加するか、シンプルなインテグレーションの最後のコネクションとして追加します。

前提条件

- リソースを更新または作成する FHIR サーバー上へのコネクションが作成されている必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。または、最後のコネクションを追加するよう要求されます。

手順

1. FHIR サーバーでリソースの更新に使用する FHIR コネクションをクリックします。
2. **Choose an Action** ページで **Update** をクリックします。
3. **Resource Type** フィールドをクリックし、FHIR リソースタイプのリストを表示します。更新または作成するリソースのタイプを選択または入力します。
4. 更新または作成するリソースタイプに他のリソースタイプが含まれている場合、**Contained Resource Types** フィールドで含まれているタイプを選択します。複数のタイプを選択する必要がある場合は **Ctrl** キーを使用します。
含まれるリソースタイプを特定すると、データマッパーは含まれるリソースにあるフィールドを表示できるようになります。含まれるリソースタイプを指定しないと、含まれるリソースにあるフィールドをマップ元またはマップ先としてマッピングすることができません。実際に含まれないリソースを選択しても問題はありません。
5. **Next** をクリックして、このコネクションをフローに追加します。
6. 更新されたリソースに含まれるデータを取得する追加のステップがフローに必要な場合は、この FHIR コネクションの前に追加します。
7. インテグレーション可視化パネルで、リソースを更新する FHIR コネクションの前にあるプラス記号をクリックします。
8. **Data Mapper** をクリックします。
9. データマッパーで以下を行います。
 - a. リソース ID をターゲットリソース ID にマップします。これは、新しいリソースのリソース ID を指定する唯一の方法になります。
 - b. 更新されたリソースまたは新しいリソースに含める各リソースフィールドにマップします。値が変更しないフィールドおよび値を更新する必要があるフィールドにマップするようにしてください。
 - c. **Done** をクリックして、データマッパーステップをフローに追加します。

結果

インテグレーションフローで追加した場所にコネクションが表示されます。実行中、コネクションは FHIR サーバー上でリソースを更新または作成し、更新されたリソースまたは作成されたリソースのリ

ソース ID が含まれる **MethodOutcome** リソースを返します。これをフローの後続のステップでマップすることができます。

6.6. FHIR サーバー上のリソースにある指定のフィールドを更新

インテグレーションでは、FHIR サーバー上にあるリソースの個別のフィールドを更新できます。これには、フローの途中に FHIR コネクションを追加するか、FHIR コネクションをシンプルなインテグレーションの最後のコネクションとして追加します。

前提条件

- 更新するリソースを持つ FHIR サーバーへのコネクションが作成されている必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。または、Fuse Online で最後のコネクションを選択するよう要求されます。

手順

1. リソースの更新に使用する FHIR コネクションをクリックします。
2. **Choose an Action** ページで **Patch** をクリックします。
3. **Resource Type** フィールドをクリックし、FHIR リソースタイプのリストを表示します。更新するフィールドを持つリソースのタイプを選択または入力します。
4. **Number of Operations** フィールドに、更新するフィールドの数を指定します。
5. フローの以前のステップから値をマップする場合は **Resource id** フィールドを空白のままにしておきます。これが通常の作業になります。その他の場合は、更新するフィールドが含まれるリソースのリソース ID を指定します。
6. 更新が記述された JSON パッチを指定します。JSON パッチは [What is a JSON Patch?](#) に記述されている形式である必要があります。更新によっては、**Patch** アクションの設定時に JSON パッチを指定する必要があります。たとえば、リソースフィールドの値を別のフィールドにコピーすることがあります。その他の更新では、**JSON Patch** フィールドを空白のままにし、代わりに各更新を定義する値をマップすることができます。
7. **Done** をクリックして、このコネクションをフローに追加します。
8. JSON パッチやリソース ID を指定しなかった場合は、データマッパーステップを追加します。
 - a. フロー可視化パネルで、追加したばかりの FHIR コネクションの前にあるプラス記号をクリックします。
 - b. **Choose a Step** ページで **Data Mapper** をクリックします。

Target パネルには、更新する各フィールド用の番号が付けられたフォルダーが表示されます。**Patch** アクションを設定するとき、更新するフィールドの数を指定します。データマッパーは、この数のフォルダーを **Target** パネルに表示します。たとえば、更新するフィールドの数として 3 を指定すると、**1**、**2**、および **3** のラベルが付いた 3 つのターゲットフォルダーが表示されます。
 - c. 更新する各フィールドをいずれかのターゲットフォルダーのフィールドにマップします。
 - i. **Target** パネルでフォルダーを展開し、**op**、**path**、および **value** の 3 つのフィールドを表示します。

- ii. ソースフィールド、定数、またはプロパティをターゲット **path** フィールドにマップします。パスは更新するリソースフィールドを識別します。パスの値では、数字はリストフィールドのインデックスを示し、スラッシュは子フィールドに通じます。たとえば、**/name/1/given/1/value** パスをマップし、指定の name フィールドの値を更新します。
 - iii. フィールドの現在の値を置き換えるデフォルトの更新操作を許可するか、ソースフィールド、定数、またはプロパティを **op** フィールドにマップし、フィールドの更新方法を示します。可能な操作の詳細は、「[JSON patch operations](#)」を参照してください。
 - iv. フィールドを更新して新しい値を取得する場合は、ソースフィールド、定数、またはプロパティをターゲット **value** にマップします。これは、フィールドに含まれるようにする新しい値になります。
- d. **Patch** アクションの設定時にリソース ID を指定しなかった場合は、リソース ID をターゲット **id** フィールドにマップします。
- e. **Done** をクリックして、データマッピングステップをフローに追加します。

結果

インテグレーションフローで追加した場所にコネクションが表示されます。実行中、コネクションは指定のリソースフィールドを更新し、更新されたリソースの ID が含まれる **MethodOutcome** リソースを返します。

6.7. FHIR サーバーで異なるタイプのリソースの作成

インテグレーションでは、FHIR サーバーで 2 つ以上のリソースを作成でき、各リソースを異なるタイプにすることができます。たとえば、新しい患者のリソースと新しい医療担当者のリソースを 1 つのコネクションで FHIR サーバーに追加できます。これには、FHIR コネクションをフローの途中に追加するか、シンプルなインテグレーションの最後のコネクションとして追加します。

前提条件

- リソースを追加する FHIR サーバーへのコネクションが作成されている必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。または、Fuse Online で最後のコネクションを選択するよう要求されます。

手順

1. 新しいリソースを作成し、FHIR サーバーに追加するために使用する FHIR コネクションをクリックします。
2. **Choose an Action** ページで **Transaction** をクリックします。
3. **Included Resource Types** フィールドに、このコネクションが作成するリソースタイプを選択します。**Ctrl** キーを使用して複数のリソースタイプを選択します。コネクションは、選択した各タイプのリソースを 1 つ作成します。
4. **Next** をクリックして、このコネクションをフローに追加します。
5. 新しいリソースにフィールドを入力するには、このコネクションの前にデータマッピングステップを追加します。

- a. 左側のフロー可視化パネルで、追加したばかりのコネクションの前にあるプラス記号をクリックします。
- b. **Data Mapper** をクリックします。
- c. データマッパーでソースフィールドをターゲットリソースフィールドにマップします。新しいリソースには、マップ先のターゲットフィールドのみが含まれます。
- d. **Done** をクリックして、データマッパーステップをフローに追加します。

結果

インテグレーションフローで追加した場所にコネクションが表示されます。実行中、コネクションはすべてのリソースを FHIR サーバーに追加します。アクションが無事実行されると、新しい個々のリソースのリソース ID が含まれるトランザクションリソースが返されます。これらのリソース ID をフローの後続のステップにマップできます。

6.8. FHIR サーバーからのリソースの削除

インテグレーションでは、FHIR サーバーからリソースを削除できます。これには、フローの途中で FHIR コネクションを追加するか、FHIR コネクションをシンプルなインテグレーションの最後のコネクションとして追加します。

前提条件

- 削除するリソースを持つ FHIR サーバーへのコネクションが作成されている必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。または、Fuse Online で最後のコネクションを選択するよう要求されます。

手順

1. リソースの削除に使用する FHIR コネクションをクリックします。
2. **Choose an Action** ページで **Delete** をクリックします。
3. **Resource Type** フィールドをクリックし、FHIR リソースタイプのリストを表示します。削除するリソースのタイプを選択または入力します。
4. **Resource Id** フィールドに、削除するリソースのリソース ID を指定します。インテグレーションの初期のステップからリソース ID をマップする予定である場合は、このフィールドを空白のままにしておきます。
5. **Resource version** フィールドには、リソースのバージョン ID を任意で指定します。最新バージョンのリソースを削除する場合や、以前のステップから削除するリソースのバージョン ID をマップする場合は、このフィールドを空白のままにしておきます。
6. **Done** をクリックして、このコネクションをフローに追加します。

結果

インテグレーションフローで追加した場所にコネクションが表示されます。実行中、コネクションは FHIR サーバー上の指定のリソースを削除し、削除されたリソースの ID が含まれる **MethodOutcome** リソースを返します。

次のステップ

Resource Id フィールドまたは **Resource version** フィールドに値をマップするには、この接続の前にデータマップステップを追加します。

第7章 FTP または SFTP サーバーへの接続

インテグレーションでは、FTP または SFTP サーバーに接続して、ファイルをダウンロードまたはアップロードすることができます。これには、FTP または SFTP コネクションを作成し、インテグレーションフローに追加します。詳細は以下を参照してください。

- 「FTP または SFTP コネクションの作成」
- 「FTP または SFTP サーバーからのファイルの取得」
- 「FTP または SFTP サーバーへのファイルのアップロード」

7.1. FTP または SFTP コネクションの作成

インテグレーションで FTP または SFTP サーバーからファイルをダウンロードしたり、アップロードするには、その FTP または SFTP サーバーへのコネクションを作成します。同じコネクションを任意の数のインテグレーションに追加できます。

前提条件

接続するサーバーのホスト名を知っている必要があります。

手順

1. Fuse Online の左パネルで **Connections** をクリックし、利用可能なコネクションを表示します。
2. 右上にある **Create Connection** をクリックし、Fuse Online コネクターを表示します。
3. File Transfer Protocol (FTP) を使用するコネクションを作成する場合は、**FTP コネクター**をクリックします。
Secure File Transfer Protocol (SFTP) を使用するコネクションを作成する場合は、**SFTP コネクター**をクリックします。
4. コネクションを設定します。
 - FTP コネクションの場合は以下を行います。
 - 指定するパラメーターは **Host** のみです。接続するサーバーのホスト名を入力します。たとえば、FTP ホストの名前が **FTP.WEST** の場合、**FTP.WEST** をそのまま入力します。**ftp://FTP.WEST** のようにプロトコルは指定しないでください。
 - **Port** は必要なパラメーターで、デフォルト値は **21** になります。これは、FTP サーバーがリッスンするポートになります。
 - その他のパラメーターは、不必要であるかデフォルト値が指定されています。ほとんどのインテグレーションではデフォルトの値が適しています。これらのパラメーターの説明は、この手順の後に記載されています。
 - SFTP コネクションの場合は、以下のパラメーターに値が必要になります。
 - **Host** は、接続する SFTP サーバーのホスト名です。たとえば、SFTP ホストの名前が **SFTP.EAST** の場合、**SFTP.EAST** をそのまま入力します。**sftp://SFTP.EAST** のようにプロトコルは指定しないでください。
 - **Port** のデフォルト値は **22** になります。これは、SFTP サーバーがリッスンするポートになります。

- **User Name** は SFTP サーバーへのアクセスに使用するアカウントのユーザー名です。
 - **Password** はそのユーザーに関連するパスワードです。
 - その他のパラメーターにはデフォルト値が指定されています。ほとんどのインテグレーションではデフォルトの値が適しています。これらのパラメーターの説明は、この手順の後に記載されています。
5. **Validate** をクリックします。Fuse Online は即座に接続の検証を試み、検証の成功または失敗を示すメッセージを表示します。検証に失敗した場合は、必要に応じて設定の詳細を修正し、再度検証を行います。
 6. 検証に成功したら **Next** をクリックします。
 7. **Connection Name** フィールドに、この接続を別の接続と区別するために使用する名前を入力します。たとえば、**XLIGHT FTP Server** を入力します。
 8. **Description** は任意のフィールドで、この接続に関する便利な情報を入力します。
 9. 右上にある **Create** をクリックし、作成した接続が利用可能になったことを確認します。上記の例の名前を入力した場合は、インテグレーションに追加できる接続として **XLIGHT FTP Server** が表示されます。

その他のパラメーターの説明

- **Connect timeout** のデフォルト値は 10000 ミリ秒で、接続の確立まで最大 10 秒待機することを意味します。10 秒が経過しても接続が確立されなかった場合、Fuse Online は **Reconnect delay** で定義されたミリ秒間待機した後に、再接続を試みます。
- **Reconnect delay** のデフォルト値は 1000 ミリ秒で、再接続を試みる前の待機時間を表します。
- **Maximum reconnect attempts** のデフォルト値は 3 です。Fuse Online は接続の確立を最大 3 回試みます。
- **Binary file transfer mode** がデフォルトで使用されます。転送モードを ASCII にする場合は **No** を選択します。
- **Passive connection mode** のデフォルトは **Yes** で、これが通常の推奨モードになります。パッシブモードでは、ファイアウォールの問題を避けるために、クライアントがサーバーで通信チャネルを開きます。**No** を選択するとアクティブモードが使用されます。
- **Disconnect from the server after use** のデフォルト値は **No** です。アクションの実行後、接続は維持されます。接続がアップロードまたはダウンロードを実行した後にサーバーから接続を切断する場合は **Yes** を選択します。
- **Data timeout** のデフォルト値は 30000 ミリ秒で、Fuse Online が応答を待つ最大時間を示します。

7.2. FTP または SFTP サーバーからのファイルの取得

FTP または SFTP コネクションが対象のファイルを見つけたときにインテグレーションを実行するには、FTP または SFTP コネクションをインテグレーションの最初の接続として追加します。

前提条件

FTP または SFTP コネクションが作成済みである必要があります。

手順

1. Fuse Online パネルの左側にある **Integrations** をクリックします。
2. **Create Integration** をクリックします。
3. **Choose a Start Connection** ページで、FTP または SFTP サーバーのポーリングに使用する FTP または SFTP コネクションをクリックします。
4. **Choose an Action** ページで **Download** をクリックします。
5. ファイルが1つの場合は、**File name expression** フィールドにファイル名に解決される [Apache Camel Simple 言語](#) 表現を入力します。正規表現を指定することはできません。コネクションはこのファイルを見つけるためにサーバーをポーリング (定期的を確認) し、見つかったらそのファイルをダウンロードします。複数のファイルをダウンロードする場合はこのフィールドを空白のままにしておきます。
6. **FTP directory** フィールドに、ポーリングする server ディレクトリーの絶対または相対パスを入力します。コネクションは、このディレクトリーのコンテンツを確認し、コンテンツを見つければファイルをすべてダウンロードします。
7. **Milliseconds before polling starts** フィールドでは、デフォルトの 1000 ミリ秒を使用するか、値を変更します。
8. **Milliseconds before the next poll** フィールドでは、デフォルトの 500 ミリ秒を使用するか、値をミリ秒単位で変更します。これは、ポーリングの間隔になります。
9. **Delete after download** フィールドでは、デフォルト値の **No** を使用するか、**Yes** を選択してファイルをダウンロードした後にサーバーから削除します。
10. **Next** をクリックして、アクションの出力タイプを指定します。
11. データタイプの認識が必要がない場合は、**Select Type** フィールドで **Type specification not required** を受け入れ、下部にある **Done** をクリックします。残りの手順を行う必要はありません。
その他の場合は、以下の1つをスキーマタイプとして選択します。
 - **JSON schema** は JSON データの構造を記述するドキュメントです。ドキュメントのメディアタイプは **application/schema+json** です。
 - **JSON instance** は JSON データが含まれるドキュメントです。ドキュメントのメディアタイプは **application/json** です。
 - **XML schema** は XML データの構造を記述するドキュメントです。このドキュメントのファイル拡張子は **.xsd** です。
 - **XML instance** は XML データが含まれるドキュメントです。このドキュメントのファイル拡張子は **.xml** です。
12. **Definition** 入力ボックスに、選択したスキーマタイプに準拠する定義を貼り付けます。たとえば、**JSON schema** を選択した場合は、メディアタイプが **application/schema+json** の JSON スキーマファイルの内容を貼り付けます。
13. **Data Type Name** フィールドにはデータタイプに選択した名前を入力します。たとえば、ベンダーの JSON スキーマを指定する場合は、データ型名として **Vendor** を指定できます。

このタイプを指定するコネクションを使用する場合は、このタイプを作成または編集するときに

このタイプを指定するコネクションを使用するインテグレーションを作成または編集するときにこのデータタイプ名が表示されます。Fuse Online では、インテグレーション仮想化パネルとデータマッパーにタイプ名が表示されます。

14. **Data Type Description** フィールドには、このタイプを区別するのに便利な情報を提供します。データマッパーでこのタイプを処理するステップにカーソルを合わせると、この説明が表示されます。
15. **Done** をクリックします。

結果

インテグレーションフローの最初にコネクションが表示されます。

7.3. FTP または SFTP サーバーへのファイルのアップロード

インテグレーションでは、フローの途中で FTP または SFTP サーバーへファイルをアップロードしたり、シンプルなインテグレーションを終了することができます。これには、FTP または SFTP コネクションをフローの途中に追加するか、シンプルなインテグレーションの最後のコネクションとして追加します。

前提条件

- FTP または SFTP コネクションが作成済みである必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。または、Fuse Online で最後のコネクションを選択するよう要求されます。

手順

1. ファイルのアップロードに使用する FTP または SFTP コネクションをクリックします。
2. **Choose an Action** ページで **Upload** をクリックします。
3. 特定の1つのファイルのみをアップロードする場合は、**File name expression** フィールドにファイル名に解決される [Apache Camel Simple 言語](#) 表現を入力します。これは、アクションによってサーバーにアップロードされるファイルの名前になります。正規表現を指定することはできません。複数のファイルをアップロードする場合はこのフィールドを空白のままにしておきます。
4. **FTP directory** フィールドには server ディレクトリーの絶対または相対名を入力します。**File name expression** フィールドに表現が含まれる場合、コネクションはこのディレクトリーに指定されたファイルを格納します。**File name expression** フィールドが空白である場合、コネクションは前のステップで取得したすべてのファイルをこのディレクトリーにアップロードします。
5. **If file exists** フィールドには、サーバーのファイルと同じパスや名前を持つファイルをアップロードする場合に行う処置を示します。デフォルトの **Override** を使用してサーバー上のファイルをアップロードするファイルでオーバーライドするか、以下の1つを選択します。
 - **Append** は、アップロードするファイルの内容をサーバー上のファイルに追加します。
 - **Fail** は **GenericFileOperationException** をスローします。インテグレーションはエラー状態になりません。
 - **Ignore** はファイルをアップロードしません。インテグレーションは何も問題がないことを前提で実行を継続します。

- **Move** は1つのファイルの名前を変更します。
 - **TryRename** は一時的な名前でファイルをアップロードし、希望のファイル名に変更します。この操作は、希望の名前を持つファイルの存在を確認しないため、既存ファイルのチェックを行う場合よりもほとんどのサーバーで操作が速くなります。
6. **Temporary file prefix while copying** フィールドには、文字列を指定します。ファイルのアップロード中にこの文字列がファイル名の先頭に追加されます。これにより、コネクションはサーバー上で一時ファイルに書き込みを行い、その一時ファイルの名前を適切な名前に変更することが可能になります。これは、非常に大きなファイルをアップロードする場合にロックを減らすのに便利です。
 7. **Temporary file name while copying** フィールドには、文字列を指定します。ファイルのアップロード中にそのファイルの名前が変更されます。これにより、コネクションはサーバー上で一時ファイルに書き込みを行い、その一時ファイルの名前を適切な名前に変更することが可能になります。これは、非常に大きなファイルをアップロードする場合にロックを減らすのに便利です。
 8. **Next** をクリックしてアクションの入力タイプを指定します。
 9. データタイプの認識が必要がない場合は、**Select Type** フィールドで **Type specification not required** を受け入れ、下部にある **Done** をクリックします。残りの手順を行う必要はありません。
その他の場合は、以下の1つをスキーマタイプとして選択します。
 - **JSON schema** は JSON データの構造を記述するドキュメントです。ドキュメントのメディアタイプは **application/schema+json** です。
 - **JSON instance** は JSON データが含まれるドキュメントです。ドキュメントのメディアタイプは **application/json** です。
 - **XML schema** は XML データの構造を記述するドキュメントです。このドキュメントのファイル拡張子は **.xsd** です。
 - **XML instance** は XML データが含まれるドキュメントです。このドキュメントのファイル拡張子は **.xml** です。
 10. **Definition** 入力ボックスに、選択したスキーマタイプに準拠する定義を貼り付けます。たとえば、**JSON schema** を選択した場合は、メディアタイプが **application/schema+json** の JSON スキーマファイルの内容を貼り付けます。
 11. **Data Type Name** フィールドにはデータタイプに選択した名前を入力します。たとえば、ベンダーの JSON スキーマを指定する場合は、データ型名として **Vendor** を指定できます。
このタイプを指定するコネクションを使用するインテグレーションを作成または編集するときにこのデータタイプ名が表示されます。Fuse Online では、インテグレーション仮想化パネルとデータマッパーにタイプ名が表示されます。
 12. **Data Type Description** フィールドには、このタイプを区別するのに便利な情報を提供します。データマッパーでこのタイプを処理するステップにカーソルを合わせると、この説明が表示されます。
 13. **Done** をクリックします。

結果

インテグレーションフローで追加した場所にコネクションが表示されます。

第8章 GOOGLE アプリケーションへの接続

インテグレーションは、Google アプリケーションである Gmail、Google カレンダー、および Google スプレッドシートに接続できます。以下を参照してください。

- [「Fuse Online を Google クライアントアプリケーションとして登録」](#)
- [「Gmail への接続」](#)
- [「Google カレンダーへの接続」](#)
- [「Google スプレッドシートへの接続」](#)

8.1. FUSE ONLINE を GOOGLE クライアントアプリケーションとして登録

インテグレーションで Gmail、Google カレンダー、または Google スプレッドシートへの接続を可能にするには、Fuse Online 環境を Google クライアントアプリケーションとして登録する必要があります。登録中に、Google API を有効にします。さらに、有効にした Google API へアクセスするために Fuse Online が使用するクレデンシャルも作成します。

登録を完了すると、Gmail、Google カレンダー、および Google スプレッドシートへのコネクションを複数作成することができます。各コネクションを複数のインテグレーションで使用することができます。Google アプリケーションへの各コネクションは、登録中に取得する同じ Google クライアント ID と Google クライアントシークレットを使用できますが、各コネクションはユーザーが選択する別の Google アカウントにアクセスすることもできます。

この手順では、Gmail API、Google Calendar API、および Google Sheet API を有効にしますが、3つの API すべてではなく必要な API のみを有効にすることができます。ただし、3つすべての API ではなく、必要な API のみを有効にします。API を有効にしなかったアプリケーションへのコネクションを後で作成する必要がある場合は、この手順にしたがって API を有効にすることができます。

重要

Fuse Online 用の新しい Google クライアントアプリケーションを作成する必要があります。Google が新しいクライアントアプリケーションに提供するクレデンシャルには、期限切れのアクセストークンを更新するために使用される更新トークンが含まれています。この更新トークンは、Fuse Online クライアントアプリケーションがクレデンシャルを初めて使用する場合のみ使用できます。Fuse Online では、Gmail、Google カレンダー、および Google スプレッドシートへのコネクションはすべて同じ Google クライアント ID および Google クライアントシークレットを使用することができます。この場合、更新トークンは Google アプリケーションへのすべてのコネクションに対して使用できます。Fuse Online ユーザーインターフェースでコネクションの詳細を表示するときに、**Validate** ボタンをクリックしないでください。検証はクレデンシャルを2回目に使用することになり、更新トークンはクライアントクレデンシャルの一部ではなくなります。ただし、Google アプリケーションに再接続することはできます。



開発環境で、Fuse Online ではない他の OAuth クライアントで使用している Google クライアント ID および Google クライアントシークレットを使用する場合は注意してください。Fuse Online はオフラインのアクセスを必要とし、そのアクセスは最初の OAuth の交換で要求されます。他の OAuth クライアントがすでに OAuth の交換を行い、オフラインのアクセスを要求しなかった場合、Fuse Online は後続の OAuth の交換でオフラインのアクセスを取得することはできません。最初の交換でオフラインアクセスの要求があったことを確認できない場合は、Fuse Online の新しい Google クライアントアプリケーションを作成してください。

前提条件

- Fuse Online を Google クライアントアプリケーションとして登録するために使用する Google アカウントにログインできる必要があります。

手順

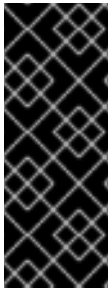
1. Fuse Online で以下を行います。
 - a. 左側のナビゲーションパネルで **Settings** をクリックします。
 - b. **Settings** ページで、上部付近にあるコールバック URL の右側で  をクリックし、Fuse Online 環境のコールバック URL をクリップボードにコピーします。この手順の後半でこの URL が必要になります。
2. 別のブラウザタブで <https://console.developers.google.com> にアクセスし、以下を行います。
 - a. Fuse Online を Google クライアントアプリケーションとして登録するために使用する Google アカウントにログインしていることを確認します。または、別の Google アカウントを選択し、そのアカウントにログインします。
Google API のページの上部に現在の Google プロジェクトの名前が表示されます。
 - b. 現在のプロジェクトを使用して、Fuse Online へ承認を付与するには、次のステップに進みます。別の Google プロジェクトで Fuse Online へ承認を付与するには、そのプロジェクトを選択または作成します。Google アカウントに既存のプロジェクトがない場合は、作成する必要があります。
 - c. Google API を有効にします。
 - i. 左上隅の  をクリックし、**APIs and Services > Library** と選択します。
 - ii. 必要な場合は下方向にスクロールし、**G Suite** 行のカードを確認します。
 - iii. **Google Calendar API** カードをクリックすると、Google Calendar API が有効になっていることを示すページが表示されます。
 - iv. 下方向にスクロールして **Gmail API** カードをクリックすると、Gmail API が有効になっていることを示すページが表示されます。
 - v. 左上隅の  をクリックし、**APIs and Services > Library** と選択します。
 - vi. 必要な場合は下方向にスクロールし、**G Suite** 行のカードを確認します。
 - vii. **Google Sheets API** カードをクリックすると、Google Sheets API が有効になっていることを示すページが表示されます。
 - d. クライアントアプリケーションに名前を付けます。
 - i. **MANAGE** をクリックします。
 - ii. 表示されるページの左側のナビゲーションパネルで **Credentials** をクリックします。右側にある **Credentials in APIs & Services** をクリックします。
 - iii. **OAuth consent screen** タブをクリックします。

- iv. 表示されるページの **Application Name** フィールドに Fuse Online クライアントアプリケーションの名前を入力します。たとえば、**Fuse Online client application** を入力します。
 - v. 他のフィールドは無視します。
 - vi. **Save** をクリックします。
- e. 以下のようにクライアントアプリケーションのクレデンシャルを取得します。
- i. **Create Credentials** の右側の下矢印をクリックしてメニューを表示し、**OAuth client ID** を選択します。
 - ii. 表示されるページで **Web application** を選択し、その他のコンテンツを表示します。
 - iii. **Name** フィールドに、Fuse Online 環境の OAuth クライアント ID の名前を入力します。これは、クライアントアプリケーション自体に入力した名前とは異なります。たとえば、**OAuth client ID for Fuse Online** を入力します。
 - iv. **Authorized JavaScript origins** は無視します。
 - v. この手順の最初で Fuse Online 環境からコピーしたコールバック URL を **Authorized redirect URIs** フィールドに貼り付けます。
 - vi. **Create** をクリックして、Fuse Online 環境のクライアント ID とシークレットを表示します。
- f. クライアント ID フィールドの右側で  をクリックし、クライアント ID をクリップボードにコピーします。
3. Fuse Online **Settings** ページに戻り、以下を行います。
- a. Gmail、Google カレンダー、および Google スプレッドシートのエントリーを展開します。
 - b. 各 Google アプリケーションの **Client ID** フィールドに、コピーした Google クライアント ID を貼り付けます。
4. Google の開発者サイトに戻り、クライアントシークレットフィールドの右側で  をクリックし、クライアントシークレットをクリップボードにコピーします。
5. Fuse Online **Settings** ページに戻り、各 Google アプリケーションのエントリーで以下を行います。
- a. コピーした Google クライアントシークレットを **Client Secret** フィールドに貼り付けます。
 - b. **Save** をクリックします。**Registration Successful!** という通知が表示されるはずです。
 - c. **Ok** をクリックしてエントリーを折りたたみます。

結果

有効にした各 Google アプリケーション API に対して、そのアプリケーションへのコネクションを作成できます。

Fuse Online から Google アプリケーションへのすべてのコネクションは、同じ Google クライアント ID と Google クライアントシークレットを使用します。



重要

Google アプリケーションへのコネクションを持つインテグレーションが継続的に正しく動作するようにするため、Google クライアント ID と Google クライアントシークレットにはトークン更新情報が含まれます。そのため、新しいクレデンシャルを取得 **しない** でください。新しいクレデンシャルを取得すると、各 Google コネクションを再作成し、以前のコネクションを新しいコネクションに置き換え、Google コネクションを使用する各インテグレーションを再パブリッシュする必要があります。

8.2. GMAIL への接続

特定の Gmail アカウントがメールを受け取ったときにインテグレーションの実行をトリガーするには、Gmail コネクションを最初のコネクションとしてシンプルなインテグレーションに追加します。インテグレーションで特定の Gmail アカウントからメールを送信するには、以下のいずれかを行います。

- Gmail コネクションをフローの途中に追加します。
- Gmail コネクションを追加して、シンプルなインテグレーションを完了します。

インテグレーションで Gmail に接続するための一般的なステップは次のとおりです。

1. [Fuse Online を Google クライアントアプリケーションとして登録します](#)。
2. Gmail コネクションを作成します。これには、コネクションがアクセスを許可される Gmail アカウントを選択します。
3. インテグレーションが Gmail アカウントからメールを送信する場合は、送信メールの内容を入力する方法を決定します。
4. インテグレーションフローに Gmail コネクションを追加します。
5. メールを送信する Gmail コネクションでは、任意でインテグレーションデータを電子メールフィールドにマップします。

情報や手順は以下を参照してください。

- [「Gmail コネクションの作成」](#)
- [「送信メールの代替の作成方法」](#)
- [「ポーリングによる Gmail メッセージの返信時にインテグレーションをトリガー」](#)
- [「Gmail アカウントからのメールの送信」](#)

8.2.1. Gmail コネクションの作成

Gmail コネクションを作成する場合、特定の1つの Gmail アカウントにアクセスできるようにコネクションを許可する必要があります。Gmail コネクションの作成後、複数のインテグレーションを追加することができます。

前提条件

- [Fuse Online](#) が [Google クライアントアプリケーションとして登録](#) され、Gmail API が有効になっている必要があります。
- Gmail の Fuse Online **Settings** ページエントリーには、Fuse Online を Google クライアントアプリケーションとして登録して取得したクライアント ID とクライアントシークレットの値があります。

手順

1. Fuse Online の左パネルで **Connections** をクリックし、利用可能なコネクションを表示します。
2. 右上の **Create Connection** をクリックし、利用可能なコネクタを表示します。コネクタは、1つ以上のコネクションを作成するために使用するテンプレートです。
3. **Gmail** コネクタをクリックします。
4. **Configure Connection** ページで **Connect Gmail** をクリックします。Sign in with Gmail ページが表示されます。
Connect Gmail が表示されない場合、Fuse Online 環境は Google クライアントアプリケーションとして登録されていません。「[Fuse Online を Google クライアントアプリケーションとして登録](#)」を参照してください。Fuse Online 環境が Google クライアントアプリケーションとして登録されていない状態で Gmail コネクションを作成しようとすると、Fuse Online に承認情報の入力を要求する複数のフィールドが表示されます。これらのフィールドに値を入力して Gmail コネクションを作成することは可能ですが、推奨されません。
5. Google へのログインを要求するページで、このコネクションが Fuse Online からアクセスする Google アカウントのメールアドレスを入力し、**Next** をクリックします。
6. **openshiftapps.com wants to access your Google Account**に対して **Allow** をクリックし、Fuse Online に戻ります。
7. Fuse Online の **Connection Name** フィールドに、このコネクションを別のコネクションと区別するために使用する名前を入力します。たとえば、**Gmail Connect 1** を入力します。
8. **Description** は任意のフィールドで、このコネクションに関する便利な情報を入力します。たとえば、**Sample Gmail connection that uses jkim Gmail account credentials.** を入力します。
9. 右上にある **Create** をクリックし、作成したコネクションが利用可能になったことを確認します。上記の例の名前を入力した場合は、インテグレーションに追加できるコネクションとして **Gmail Connect 1** が表示されます。

8.2.2. 送信メールの代替の作成方法

シンプルなインテグレーションを完了する Gmail コネクションや、フローの途中にある Gmail コネクションは、アクセスが許可されている Gmail アカウントからメールを送信します。Gmail コネクションが送信するメールの内容を入力する方法は複数あります。メールを送信する Gmail を追加する前に、メールの内容を入力する方法を検討してください。

送信メールの内容を入力する代替方法は次のとおりです。

- メールを送信する Gmail コネクションの直前にデータマッピングステップを追加します。このデータマッピングステップでは、以前のステップから出力されたデータフィールドを Gmail コネクションの **Send Email** アクションフィールドにマップします。**Send Email** アクションフィールドは次のとおりです。

- Email to
- Email subject
- Email text
- Email cc
- Email bcc

データマッピングステップを追加した場合、**Send Email** アクションフィールドを1つ、一部、またはすべてマップできます。

- Gmail コネクションをフローに追加するとき、**Send Email** アクションフィールドに値を指定してアクションを設定します。値は1つ、一部、またはすべてのフィールドに指定できます。
- データマッピングステップと **Send Email** アクション設定を両方使用して、フィールドに値を入力します。つまり、1つ以上のフィールドを指定して **Send Email** アクションを設定でき、その他のフィールドに値を入力するデータマッピングステップを追加することもできます。
Send Email アクションフィールドに直接指定する値は、**Send Email** アクションフィールドにマップされる値よりも優先されます。つまり、コネクションの追加時およびアクションの設定時に値を指定して **Send Email** フィールドに値を入力し、さらに同じフィールドに値をマッピングした場合に、アクション設定で指定した値は常にマップされた値よりも優先されます。たとえば、**Email to** アクションフィールドに **people@redhat.com** を指定し、前のステップのメールフィールドも Gmail の **Email to** フィールドにマップする場合、インテグレーションは常に **people@redhat.com** をメールアドレスとして使用します。

メールを送信する Gmail コネクションを追加するとき、すべてのアクション設定パラメーターは任意となります。これは、インテグレーションデータを **Send Email** アクションフィールドにマップして、メールの内容を完全に入力することを選択する場合があるからです。しかし、アクション設定での指定またはマッピングのいずれかを行って、**Email to** フィールドにメールアドレスが存在する必要があります。メッセージ送信先のメールアドレスがないと、Fuse Online はランタイムエラーを生成し、インテグレーションの実行が停止します。

8.2.3. ポーリングによる Gmail メッセージの返信時にインテグレーションをトリガー

特定の Gmail アカウントが受信したメールを基にインテグレーションを実行するには、Gmail コネクションをシンプルなインテグレーションの最初のコネクションとして追加します。インテグレーションの実行時、Gmail コネクションはユーザーが制御する間隔毎に、このアカウントでメールをチェックします。コネクションが未読のメールを見つけると、そのメールをインテグレーションの次のステップに渡し、デフォルトではメールに **read** (既読) マークを付けます。

前提条件

- メールを取得する Gmail アカウントへのアクセスが許可される Gmail コネクションが作成されている必要があります。

手順

1. Fuse Online パネルの左側にある **Integrations** をクリックします。
2. **Create Integration** をクリックします。
3. **Choose a Start Connection** ページで、インテグレーションを開始するために使用する Gmail コネクションをクリックします。

4. **Choose an Action** ページで **Receive Email** アクションをクリックします。

5. **Receive Email** アクションを設定します。

- a. **Delay** フィールドには、デフォルトの 30 秒を使用するか、インテグレーションが新規メールをチェックする頻度を指定します。
- b. 未読のメールをすべて取得する場合は **Labels** フィールドを空白のままにしておきます。特定のメールのみを取得する場合は、コネクションがアクセスしている Gmail アカウントで使用されるラベルのコンマ区切りリストを指定します。デフォルトでは、インテグレーションはこれらのラベルが付けられた未読のメールを取得します。
- c. コネクションが同じメールを 2 度返さないようにするため、**Mark as read** を選択します。**Mark as read** を選択しないと、メールが未読または既読であるかに関わらず、コネクションはアカウントの受信トレイにあるメールを返します。1 つ以上のラベルを指定した場合は、メールが未読または既読であるかに関わらず、指定したラベルが付けられたメールを返します。
- d. **Max Results** フィールドには、デフォルトの 5 を使用するか、各ポーリングに対してコネクションが返すことができるメールの最大数を指定します。コネクションがこの値を超える数の未読メールを見つけた場合、最も新しい **Max Results** メールを返します。
Gmail コネクションが 1 つ以上のメールを返すとき、インテグレーションはメールをバッチとして処理します。つまり、Fuse Online はバッチに対してインテグレーションを 1 度実行します。
- e. **Done** をクリックして、この Gmail コネクションを最初のコネクションとしてインテグレーションに追加します。コネクションはインテグレーションフローの最初のステップとして表示されます。

8.2.4. Gmail アカウントからのメールの送信

インテグレーションでは、フローの途中で Gmail アカウントからメールを送信したり、シンプルなインテグレーションを終了するために Gmail アカウントからメールを送信することができます。これには、Gmail コネクションをフローの途中に追加するか、シンプルなインテグレーションの最後のコネクションとして追加します。

前提条件

- Gmail コネクションが作成されている必要があります。
- [送信するメールの内容を入力する代替方法](#) について熟知し、このようなメールの内容を入力する計画がある必要があります。
- Fuse Online は、ステップの選択またはシンプルなインテグレーションの最後のコネクションの選択を要求します。

手順

1. メール送信に使用する Gmail コネクションをクリックします。
2. **Choose an Action** ページで **Send Email** をクリックします。
3. **Configure Send Email** ページで以下の 1 つを行います。

- この接続の直前にデータマッピングステップを追加する計画があり、インテグレーションデータを **Send Email** フィールドにマップする計画がある場合は、すべてのフィールドを空白にしておきます。
- 次の **Send Email** フィールドの1つ以上に情報を入力します。複数のメールアドレスはコマで区切ります。
 - a. **Email To**
 - b. **Email Subject**
 - c. **Email Text**
 - d. **Email cc**
 - e. **Email bcc**
- 一部のフィールドを空白のままにし、他の一部のフィールドに情報を入力します。この Gmail 接続の直前にデータマッピングステップを追加する計画があり、インテグレーションデータを一部のメールにマップする場合、アクション設定でこれらのフィールドを空白にします。必要に応じて他のアクションフィールドに値を入力します。

Send Email アクション設定フィールドに指定した値は、以前のステップからマップした値よりも優先されます。

4. **Done** をクリックして、接続をフローに追加します。

結果および次のステップ

インテグレーションフローで追加した場所に接続が表示されます。この接続の前にデータマッピングステップを追加する予定がある場合は最初に、追加する予定のある他の接続をこのインテグレーションに追加します。その後、データマッピングステップを追加します。

8.3. GOOGLE カレンダーへの接続

ポーリングが更新を Google カレンダーに返すときにインテグレーションの実行をトリガーするには、Google カレンダー接続を最初のコネクションとしてシンプルなインテグレーションに追加します。イベントをカレンダーに追加するか、カレンダーのイベントを更新するには、以下のいずれかを行います。

- Google カレンダーをフローの途中に追加します。
- Google カレンダーを追加して、シンプルなインテグレーションを完了します。

Google カレンダーへの接続に関する詳細は、以下を参照してください。

- [「Google カレンダー接続の作成」](#)
- [「ポーリングが Google カレンダーからイベントを返すときにインテグレーションをトリガー」](#)
- [「Google カレンダーからの特定イベントの取得」](#)
- [「Google カレンダーでのイベントの更新」](#)
- [「イベントの Google カレンダーへの追加」](#)

8.3.1. Google カレンダーコネクションの作成

Google カレンダーのコネクションを作成するとき、1つの特定の Google アカウントに関連する Google カレンダーへアクセスするためコネクションを承認します。Google カレンダーコネクションの作成後、複数のインテグレーションに追加できます。

前提条件

- [Fuse Online](#) が [Google クライアントアプリケーションとして登録](#) され、Google Calendar API が有効になっている必要があります。
- Google カレンダーの Fuse Online **Settings** ページエントリーには、Fuse Online を Google クライアントアプリケーションとして登録して取得したクライアント ID とクライアントシークレットの値があります。

手順

1. Fuse Online の左パネルで **Connections** をクリックし、利用可能なコネクションを表示します。
2. 右上の **Create Connection** をクリックし、利用可能なコネクタを表示します。コネクタは、1つ以上のコネクションを作成するために使用するテンプレートです。
3. **Google Calendar** コネクタをクリックします。
4. **Configure Connection** ページで **Connect Google Calendar** をクリックします。Google のログインページが表示されます。
Connect Google Calendar が表示されない場合、Fuse Online 環境は Google Calendar API が有効な状態で Google クライアントアプリケーションとして登録されていません。「[Fuse Online を Google クライアントアプリケーションとして登録](#)」を参照してください。Fuse Online 環境が Google クライアントアプリケーションとして登録されていない状態で Google カレンダーコネクションを作成しようとすると、Fuse Online に承認情報の入力を要求する複数のフィールドが表示されます。これらのフィールドに値を入力して Google カレンダーコネクションを作成することは可能ですが、推奨されません。
5. Google のログインページで、このコネクションが Fuse Online からアクセスする Google アカウントを選択し、**Next** をクリックします。
6. **openshiftapps.com wants to access your Google Account** プロンプトに対して **Allow** をクリックし、Fuse Online に戻ります。
7. Fuse Online の **Connection Name** フィールドに、この接続を別の接続と区別するために使用する名前を入力します。たとえば、**Google Calendar Work Connection** を入力します。
8. **Description** は任意のフィールドで、このコネクションに関する便利な情報を入力します。たとえば、**Sample Google Calendar connection that uses my Google work account.** を入力します。
9. 右上にある **Create** をクリックし、作成したコネクションが利用可能になったことを確認します。上記の例の名前を入力した場合は、インテグレーションに追加できるコネクションとして **Google Calendar Work Connection** が表示されます。

8.3.2. ポーリングが Google カレンダーからイベントを返すときにインテグレーションをトリガー

指定した Google カレンダーからイベントを取得したときにインテグレーションの実行をトリガーする

には、Google カレンダーコネクションを最初のコネクションとしてシンプルなインテグレーションに追加します。インテグレーションの実行時、Google カレンダーコネクションはユーザーが制御できる間隔で Google カレンダーのイベントをチェックします。コネクションが、Google カレンダーの **Get Events** アクションの設定方法に準拠するイベントを見つけると、コネクションはイベントをインテグレーションの次のステップに渡します。

Google カレンダーコネクションが 2 つ以上のイベントを返すとき、Fuse Online は返された各イベントに対してインテグレーションを実行します。たとえば、ポーリングによって 5 つのイベントが返されると、Fuse Online はインテグレーションを 5 回実行します。

前提条件

- Google カレンダーコネクションが作成済みである必要があります。

手順

1. Fuse Online パネルの左側にある **Integrations** をクリックします。
2. **Create Integration** をクリックします。
3. **Choose a Start Connection** ページで、インテグレーションを開始するために使用する Google カレンダーコネクションをクリックします。
4. **Choose an Action** ページで **Get Events** アクションをクリックします。
5. **Get Events** アクションを設定します。
 - a. **Delay** フィールドには、デフォルトの 30 秒を使用するか、インテグレーションがカレンダーをチェックする頻度を指定します。
 - b. **Calendar name** フィールドには、この Google カレンダーコネクションによるアクセスが承認されている Google アカウントからアクセス可能なカレンダーの名前を入力します。
 - c. **Max Results** フィールドでは、デフォルトの 5 を使用するか、各ポーリングに対して返すことができるコネクションの最大イベント数を指定します。
 - d. 以下の 1 つを選択し、コネクションが使用するイベントのチェックの開始日を示します。コネクションはこの日付と後続の日付をチェックします。
 - **Consume from the current date ahead** を選択すると、現在の日付が開始日になります。たとえば、このオプションを選択し、**Max Results** が 5 の場合、Google カレンダーコネクションはイベントのチェックは現在の日付から開始され、見つかったイベントを最初から最大 5 つ返します。最後のポーリング後にイベントが更新または追加されたかは関係ありません。
 - **Consume from the last event update date on the next poll** を選択すると、前回のポーリングで返された最新の更新イベントの日付が開始日になります。たとえば、このオプションを選択し、**Max Results** が 5 の場合、Google カレンダーコネクションは最大 5 つの更新または新規イベントを返します。コネクションが新規または送信イベントをチェックする開始日は、前回のポーリングで返された最新の更新イベントの日付になります。
 - e. 任意で、**Query for events** フィールドにテキストを指定し、ポーリングが返すことができるイベントをフィルターすることができます。コネクションは、最低でも 1 つのイベントフィールドに指定されたテキストが含まれるイベントのみを返します。

たとえば、クエリーフィールドに **Standup meeting** を指定するとします。この場合、ポーリングはイベントフィールドに **Standup meeting** が含まれるイベントのみを返します。

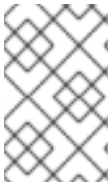
6. **Done** をクリックして、この Google カレンダーコネクションをインテグレーションの最初のコネクションとして追加します。コネクションはインテグレーションの最初のステップとして表示されます。

8.3.3. Google カレンダーからの特定イベントの取得

インテグレーションでは、フローの途中で特定の Google カレンダーイベントを取得できます。特定イベントの取得は、以下のような場合に便利です。

- 後続の Google カレンダーコネクションでイベントを更新する場合。
- 後続の Twitter コネクションを使用してイベントを発表する場合。

1つのイベントを取得するには、Google カレンダーをフローの途中に追加します。




注記

本リリースでは、シンプルなインテグレーションの最後のコネクションで特定のイベントを取得することはサポートされますが、これは特に便利なことではありません。これは今後のリリースで変更される予定です。

前提条件

- Google カレンダーコネクションが利用でき、取得するイベントがある Google カレンダーへのアクセスが承認されている必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。

手順

1. 接続するカレンダーへのアクセスが承認されている Google カレンダーコネクションをクリックします。
2. **Choose an Action** ページで **Get a Specific Event** をクリックします。
3. 取得するイベントがあるカレンダーの名前を指定します。
4. 取得するイベントの ID を指定します。イベント ID を取得するには、以下の1つを行います。
 - 以前の Google カレンダーコネクションからマップします。
 - 以下のようにカレンダーから手動で取得します。
 - a. ブラウザーで、取得するイベントが含まれるカレンダーを表示します。
 - b. URL の末尾に **?gsessionid=OK&eventdeb=1** を追加し、カレンダーを再表示します。
 - c. カレンダーで、取得するイベントをクリックします。
 - d. イベントのポップアップで、 をクリックし、**Troubleshooting info** を選択します。

- e. Google カレンダーが表示するポップアップで、**eid=** に続く文字列をコピーします。たとえば、イベント ID は **p1pva2a4t504gbsha12di9ch6k_20181107T150000Z*** のようになります。
5. **Done** をクリックして、コネクションをフローに追加します。フローで追加した場所にコネクションが表示されます。

8.3.4. イベントの Google カレンダーへの追加

インテグレーションでは、フローの途中で Google カレンダーにイベントを追加したり、シンプルなインテグレーションを終了するために Google カレンダーにイベントを追加することができます。これには、Google カレンダーコネクションをフローの途中に追加するか、シンプルなインテグレーションの最後のコネクションとして追加します。

前提条件

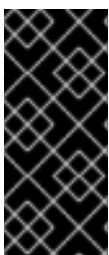
- Google カレンダーコネクションが利用でき、イベントを追加する Google カレンダーへのアクセスが承認されている必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。または、Fuse Online で最後のコネクションを選択するよう要求されます。

手順

1. イベントを追加するカレンダーへのアクセスが承認されている Google カレンダーコネクションをクリックします。
2. **Choose an Action** ページで **Create Event** アクションをクリックします。
3. **Create Event** アクションを設定します。
 - a. 新しいイベントのタイトルを入力します。
 - b. イベントの追加先となる Google カレンダーの名前を入力します。
 - c. その他の入力パラメーターにはデータを入力できますが、以前の Google カレンダーコネクションで取得したイベントから値をマップすることも可能です。
4. **Done** をクリックして、コネクションをフローに追加します。インテグレーションフローで追加した場所にコネクションが表示されます。

8.3.5. Google カレンダーでのイベントの更新

インテグレーションでは、フローの途中で Google カレンダーのイベントを更新したり、シンプルなインテグレーションを終了するために Google カレンダーのイベントを更新することができます。これには、Google カレンダーコネクションをフローの途中に追加するか、シンプルなインテグレーションの最後のコネクションとして追加します。



重要

本リリースでは、**Update Event** アクションには各イベントフィールドに値が必要です。よって、ほぼすべての場合で、[更新するイベントを取得する](#) Google カレンダーコネクションを追加し、次にイベントを更新する Google カレンダーコネクションを追加してから、この 2 つの Google カレンダーコネクションの間にデータマッピングステップを挿入する必要があります。

前提条件

- Google カレンダーコネクションが利用でき、更新するイベントがある Google カレンダーへのアクセスが承認されている必要があります。
- フローには、Google カレンダーへの以前のコネクションがあり、そのコネクションが更新するイベントを取得します。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。または、Fuse Online で最後のコネクションを選択するように要求されます。

手順

1. 更新するイベントがあるカレンダーへのアクセスが承認されている Google カレンダーコネクションをクリックします。
2. **Choose an Action** ページで **Update Event** をクリックします。
3. **Update Event** アクションを設定します。
 - a. 更新するイベントのタイトルを入力します。
 - b. 更新する各イベントフィールドの内容を入力します。フィールドの内容を変更したくない場合は、イベントフィールドに内容を入力しないでください。
4. **Done** をクリックして、コネクションをフローに追加します。インテグレーション可視化フローで追加した場所にコネクションが表示されます。
5. インテグレーション可視化フローで、追加したばかりのコネクションの前にあるプラス記号をクリックします。
6. **Data Mapper** をクリックします。
7. データマッパーで、更新されたイベントの変更のなかった各イベントフィールドを、イベントを取得した Google カレンダーコネクションから、イベントを更新した Google カレンダーコネクションの対応フィールドにマップします。
更新しているフィールドをマップしないでください。**Update Event** アクションの設定時にフィールドをマップし、そのフィールドの更新も指定すると、Fuse Online はマップされた値を使用します。
8. 右上の **Done** をクリックし、データマッパーステップを追加します。

8.4. GOOGLE スプレッドシートへの接続

Google スプレッドシートコネクションがスプレッドシートデータまたはスプレッドシートプロパティを返すときにインテグレーションの実行をトリガーするには、Google スプレッドシートコネクションを最初のコネクションとしてシンプルなインテグレーションに追加します。スプレッドシートの値またはプロパティを更新するか、チャートまたはピボットテーブルをスプレッドシートに挿入して、シンプルなインテグレーションを終了するには、Google スプレッドシートコネクションを最後のコネクションとしてシンプルなインテグレーションに追加します。

フローの途中で、スプレッドシートで値を取得、作成、または更新する Google スプレッドシートコネクションを追加でき、スプレッドシートにチャートやピボットテーブルを挿入する Google スプレッドシートコネクションを追加することもできます。

Google スプレッドシートへの接続に関する詳細は、以下を参照してください。

- 「Google スプレッドシートコネクションの作成」
- 「ポーリングがスプレッドシートデータを返すときにインテグレーションをトリガー」
- 「ポーリングがスプレッドシートプロパティを返すときにインテグレーションをトリガー」
- 「スプレッドシートの作成」
- 「シートのデータの更新」
- 「データをシートに追加」
- 「スプレッドシートプロパティの更新」
- 「チャートをシートに追加」
- 「ピボットテーブルをシートに追加」

8.4.1. Google スプレッドシートコネクションの作成

Google スプレッドシートのコネクションを作成するとき、1つの特定の Google アカウント (ユーザーが選択する) に関連する Google スプレッドシートへアクセスするためにコネクションを承認します。Google スプレッドシートコネクションの作成後、複数のインテグレーションに追加できます。

前提条件

- [Fuse Online](#) が [Google クライアントアプリケーションとして登録](#) され、Google Sheets API が有効になっている必要があります。
- Google スプレッドシートの Fuse Online **Settings** ページエントリーには、Fuse Online を Google クライアントアプリケーションとして登録して取得したクライアント ID とクライアントシークレットの値があります。

手順

1. Fuse Online の左パネルで **Connections** をクリックし、利用可能なコネクションを表示します。
2. 右上の **Create Connection** をクリックし、利用可能なコネクタを表示します。コネクタは、1つ以上のコネクションを作成するために使用するテンプレートです。
3. **Google Sheets** コネクタをクリックします。
4. **Configure Connection** ページで **Connect Google Sheets** をクリックします。Google のログインページが表示されます。
Connect Google Sheets が表示されない場合、Fuse Online 環境は Google Sheets API が有効な状態で Google クライアントアプリケーションとして登録されていません。「[Fuse Online を Google クライアントアプリケーションとして登録](#)」を参照してください。Fuse Online 環境が Google クライアントアプリケーションとして登録されていない状態で Google スプレッドシートコネクションを作成しようとすると、Fuse Online に承認情報の入力を要求する複数のフィールドが表示されます。これらのフィールドに値を入力して Google スプレッドシートコネクションを作成することは可能ですが、推奨されません。
5. Google のログインページで、このコネクションが Fuse Online からアクセスする Google アカウントを選択し、**Next** をクリックします。

6. **openshiftapps.com wants to access your Google Account**プロンプトに対して **Allow** をクリックし、Fuse Online に戻ります。
7. Fuse Online の **Connection Name** フィールドに、この接続を別の接続と区別するために使用する名前を入力します。たとえば、**Google Sheets Work Connection** を入力します。
8. **Description** は任意のフィールドで、このコネクションに関する便利な情報を入力します。たとえば、**Sample Google Sheets connection that uses my Google work account.** を入力します。
9. 右上にある **Create** をクリックし、作成したコネクションが利用可能になったことを確認します。上記の例の名前を入力した場合は、インテグレーションに追加できるコネクションとして **Google Sheets Work Connection** が表示されます。

8.4.2. ポーリングがスプレッドシートデータを返すときにインテグレーションをトリガー

Google スプレッドシートからデータを取得したときにインテグレーションの実行をトリガーするには、Google スプレッドシートコネクションを最初のコネクションとしてシンプルなインテグレーションに追加します。インテグレーションの実行時、Google スプレッドシートは指定された間隔でスプレッドシートをポーリングして特定したデータを取得し、データをインテグレーションの次のステップに渡します。

スプレッドシートの特定のシートからデータを取得するには、Google スプレッドシートコネクションの **Get sheet values** アクションを設定するときにシート名を指定します。1つの特定のコネクションは1つのシートからのみデータを取得できます。

ポーリングの合間に、コネクションが返すように設定されているシート値に変更がない場合は、次のポーリングは前回のポーリングと同じ値を返します。

前提条件

- データの取得元となるスプレッドシートへのアクセスが承認されている Google スプレッドシートコネクションが作成されている必要があります。

手順

1. Fuse Online パネルの左側にある **Integrations** をクリックします。
2. **Create Integration** をクリックします。
3. **Choose a Start Connection** ページで、インテグレーションの開始に使用する Google スプレッドシートコネクションをクリックします。
4. **Choose an Action** ページで **Get sheet values** アクションをクリックします。
5. **Get sheet values** アクションを設定します。
 - a. **SpreadsheetId** フィールドには、この Google スプレッドシートコネクションによるアクセスが承認されている Google アカウントからアクセス可能な Google のスプレッドシートの ID を入力します。
スプレッドシート ID を取得するには、ブラウザでスプレッドシートを表示します。ID は **d/** と **/edit** の間にある URL の一部です。
 - b. **Range** フィールドに、取得するデータを識別する Google A1 表記を入力します。デフォルトは **A:A** です。

たとえば、4つの列が含まれる1つのシートがあるスプレッドシートからすべてのデータを取得する場合、**A:D** を指定します。行 5 から 15 からデータを取得する場合は **A5:D15** を指定します。

データの取得元となるスプレッドシートに複数のシートがある場合は、スプレッドシート名とともに、開始セルの座標と終了セルの座標を指定します。複数のシートがある場合にシート名を指定しないと、コネクションはスプレッドシートの最初のシートからデータを取得します。たとえば、**2019!A1:D5** を指定すると、**2019** という名前のシートからデータを取得します。そのシートで列 A から列 D の行 1 から 5 のデータを取得します。

- c. **Major Dimension** フィールドでは、デフォルトである **Rows** を使用するか、**Columns** を選択します。**Rows** は、行オブジェクトのコレクションを返すようアクションを設定します。**Columns** は、列オブジェクトのコレクションを返すようアクションを設定します。各列オブジェクトには該当する各行の値が含まれます。
- d. **Split Results** フィールドでデフォルトの **No** を許可するか、**Yes** を選択します。**No** を設定すると、データを値のコレクションとして返すアクションを設定します。これにより、コネクションは行オブジェクトのコレクションまたは列オブジェクトのコレクションをフローの次のステップに渡します。**Yes** を選択すると、**Major Dimension** の設定にしたがってコネクションは返されたデータを分割できます。たとえば、**Major Dimension** が **Rows** に設定されている場合、コネクションは行オブジェクトを返します。各行オブジェクトはフローの実行を個別にトリガーします。Fuse Online は返された各行オブジェクトに対してフローを 1 度実行します。たとえば、ポーリングによって 5 つの行が返されると、Fuse Online はフローを 5 回実行します。
Fuse Online は、フローに追加できる個別の分割ステップ (split step) および集約ステップ (aggregate step) も提供します。1 つ以上のステップで個別のオブジェクトを処理し、行または列オブジェクトを集約する場合、Google スプレッドシートコネクションの結果を分割しないでください。代わりに、デフォルトの **No** を使用し、このコネクションの後にフローに分割ステップ (split step) を追加します。フローに集約ステップ (aggregate step) を使用する場合は分割ステップ (split step) が必要になります。
- e. **Delay** フィールドには、デフォルトの 30 秒を使用するか、コネクションがスプレッドシートデータを取得する頻度を指定します。
- f. **Max Results** フィールドにはデフォルトの 25 を使用するか、結果メトリクスのマイナーディメンションに対して他の制限を指定します。
たとえば、**Major Dimension** が列でその **Max Results** が 25 の場合、ポーリングは範囲の設定に含まれる各列の列オブジェクトを返します。各列オブジェクトには 25 以下の行の値が含まれます。

本リリースでは、**Max Results** の設定は **Split Results** が **Yes** に設定されている場合のみ有効である既知の問題が存在します。**Split Results** が **No** に設定された場合、**Max Results** の設定は無視されます。次のリリースでは、**Max Results** の設定は **Split Results** の設定に依存しなくなる予定です。

- 6. **Done** をクリックして、この Google スプレッドシートコネクションをインテグレーションの最初のコネクションとして追加します。インテグレーション可視化パネルで、コネクションはインテグレーションの最初のステップとして表示されます。

次のステップ

最初のコネクションの追加後、インテグレーションの最後のコネクションを追加するよう求められます。インテグレーションの最初および最後のコネクションを追加した後、インテグレーションに必要な他のコネクションを追加します。シートの値を取得するコネクションの後に、データマッパーステップを追加します。Fuse Online は、**Get sheet values** アクションの設定方法に応じて、データマッパーでソースフィールドを表示します。メジャーディメンションが **Rows** である場合、データマッパーはター

ゲットにマップできるフィールドとして列名を表示します。メジャーディメンションが **COLUMNS** である場合、データマッパーは行インデックスをターゲットにマップできるフィールドとしてリストします。

その他のリソース

スプレッドシートでセルのグループを指定する [Google A1 表記](#)

8.4.3. ポーリングがスプレッドシートプロパティを返すときにインテグレーションをトリガー

Google スプレッドシートからプロパティを取得したときにインテグレーションの実行をトリガーするには、Google スプレッドシートコネクションを最初のコネクションとしてシンプルなインテグレーションに追加します。インテグレーションの実行時、Google スプレッドシートは指定された間隔でスプレッドシートをポーリングしてスプレッドシートプロパティを取得し、結果をインテグレーションの次のステップに渡します。

プロパティには、スプレッドシートのタイトル、ロケール、およびタイムゾーンが含まれます。ポーリングの合間にシートプロパティに変更がない場合は、次のポーリングは前回のポーリングと同じ値を返します。

前提条件

- プロパティの取得元となるスプレッドシートへのアクセスが承認されている Google スプレッドシートコネクションが作成されている必要があります。

手順

1. Fuse Online パネルの左側にある **Integrations** をクリックします。
2. **Create Integration** をクリックします。
3. **Choose a Start Connection** ページで、インテグレーションの開始に使用する Google スプレッドシートコネクションをクリックします。
4. **Choose an Action** ページで **Get spreadsheet properties** アクションをクリックします。
5. **Get spreadsheet properties** アクションを設定します。
 - a. **SpreadsheetId** フィールドには、この Google スプレッドシートコネクションによるアクセスが承認されている Google アカウントからアクセス可能な Google のスプレッドシートの ID を入力します。
スプレッドシート ID を取得するには、ブラウザでスプレッドシートを表示します。ID は **d/** と **/edit** の間にある URL の一部です。
 - b. **Delay** フィールドには、デフォルトの 30 秒を使用するか、コネクションがプロパティを取得する頻度を指定します。
6. **Done** をクリックして、この Google スプレッドシートコネクションをインテグレーションの最初のコネクションとして追加します。インテグレーション可視化パネルで、コネクションはインテグレーションのフローの最初のステップとして表示されます。

8.4.4. スプレッドシートの作成

フローの途中で新しいスプレッドシートを作成するには、Google スプレッドシートコネクションを最初のコネクションと最後のコネクションの間に追加します。スプレッドシートを作成する Google スプ

レッドシートコネクションでシンプルなインテグレーションを終了することはできますが、同じコネクションの新しいスプレッドシートにデータを追加することはできません。そのため、スプレッドシートを作成し、同じフローのスプレッドシートにデータを追加する場合、フローには2つの Google スプレッドシートコネクションが必要になります。1つのコネクションはスプレッドシートを作成し、後続のコネクションはデータをスプレッドシートに追加します。

前提条件

- スプレッドシートを作成する Google アカウントへのアクセスが承認される Google スプレッドシートコネクションが作成されている必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。または、Fuse Online で最後のコネクションを選択するよう要求されます。

手順

1. スプレッドシートの作成に使用する Google アカウントへのアクセスが承認される Google スプレッドシートコネクションをクリックします。
2. **Choose an Action** ページで **Create spreadsheet** をクリックします。
3. **Create spreadsheet** アクションを設定します。
 - a. **Title** フィールドに新しいスプレッドシートのタイトルを入力します。
 - b. **Time Zone** フィールドに新しいスプレッドシートが使用するタイムゾーンを入力します (例: **India Standard Time**、**Tokyo**)。
 - c. **Locale** フィールドに新しいスプレッドシートのロケールを入力します (例: **Canada**、**Hong Kong**)。
4. **Done** をクリックして、コネクションをフローに追加します。インテグレーション可視化フローで追加した場所にコネクションが表示されます。

結果

スプレッドシートを作成するコネクションによって新しいスプレッドシートのスプレッドシート ID が返されます。後続のステップで、このスプレッドシート ID を更新などのために新しいスプレッドシートにアクセスするコネクションのスプレッドシート ID にマップできます。

8.4.5. シートのデータの更新

インテグレーションでは、フローの途中でスプレッドシートのデータを更新したり、シンプルなインテグレーションを終了するためにスプレッドシートでデータを更新することができます。これには、Google スプレッドシートコネクションをフローの途中に追加するか、シンプルなインテグレーションの最後のコネクションとして追加します。

前提条件

- 更新する Google スプレッドシートへのアクセスが承認されている Google スプレッドシートコネクションが作成されている必要があります。
- 更新するスプレッドシートのスプレッドシート ID にアクセスできる必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。または、Fuse Online で最後のコネクションを選択するよう要求されます。

手順

1. 更新するスプレッドシートへのアクセスが承認されている Google スプレッドシートコネクションをクリックします。
2. **Choose an Action** ページで **Update sheet values** をクリックします。
3. **Update sheet values** を設定します。
 - a. **SpreadsheetId** フィールドで以下の1つを行います。
 - 更新するスプレッドシートの ID を入力します。
 - フィールドは空白のままにします。この場合、このコネクションをフローに追加した後に、このコネクションの前にデータマッピングステップを追加する必要があります。データマッピングステップでは、スプレッドシート ID を以前のコネクションからこのコネクションにマップします。以前のコネクションはこのコネクションで更新するスプレッドシートにアクセスする必要があります。
 - b. **Range** フィールドでは、Google A1 表記を使用して更新するスプレッドシートのセルを指定します。開始および終了座標を指定します。たとえば、**A1:D4** はスプレッドシートの最初のシートで最初の 4 行の最初の 4 列を更新します。
デフォルトは **A:A** で、スプレッドシートの最初のシートで最初の列を更新します。
 - c. **Major Dimension** フィールドでは、デフォルトである **Rows** を使用するか、**Columns** を選択します。**Rows** は、行オブジェクトを使用してシートを更新するアクションを設定します。各行オブジェクトには、更新する各列の値が含まれます。**Columns** は、列オブジェクトを使用してシートを更新するアクションを設定します。各列オブジェクトには、更新する各行の値が含まれます。
 - d. **Value Input Option** フィールドには、シートを更新するために受信するデータを Google スプレッドシートが解釈する方法を示します。デフォルトの **Unspecified** を使用すると、Google スプレッドシートは認識するデータを自動的に変換します。たとえば、入力データが日付である場合、Google スプレッドシートは日付としてフォーマットします。入力データが 10 進数の場合は、Google スプレッドシートは 10 進数としてフォーマットします。
 - **Unspecified** のデフォルトは Google Sheets API 設定で、このデフォルトは **User entered** になります。
 - **Raw** を選択すると、Google スプレッドシートは入力データをそのまま挿入します。
 - **User entered** は、認識可能なデータの自動変換を有効にします。
 - e. **Done** をクリックして、コネクションをフローに追加します。インテグレーション可視化フローで追加した場所にコネクションが表示されます。
4. フローに他のコネクションを追加する場合はここで追加し、これらの手順に戻ります。
5. フローにすべてのコネクションを追加したら、インテグレーション可視化フローで、シート値を更新する Google スプレッドシートコネクションの直前にあるプラスマークをクリックします。
6. **Data Mapper** カードをクリックして、データマッピングステップをフローに追加します。
7. データマッパーで以下を行います。
 - a. **Update sheet values** アクションを設定したときにスプレッドシート ID を指定しなかった場合、ソース **spreadsheetId** をターゲット **spreadsheetId** にマップします。

- b. フィールドをソースステップからターゲットスプレッドシートにマッピングして、シートの更新に使用するデータを指定します。たとえば、フィールドを別のスプレッドシートやデータベースからマッピングする場合があります。
必要な場合、シートの値を更新する Google スプレッドシートコネクションを編集し、**Range** または **Major Dimension** の設定を変更することができます。これらの設定を変更すると、データマッパーは変更に応じて異なるターゲットフィールドを表示します。
- c. 右上の **Done** をクリックし、データマッパーステップを追加します。

8.4.6. データをシートに追加

フローの途中でシートにデータを追加したり、シンプルなインテグレーションを終了するためにシートにデータを追加することができます。これには、Google スプレッドシートコネクションをフローの途中に追加するか、シンプルなインテグレーションの最後のコネクションとして追加します。

前提条件

- データを追加する Google スプレッドシートへのアクセスが承認される Google スプレッドシートコネクションが作成されている必要があります。
- データを追加するスプレッドシートの ID にアクセスできる必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。または、Fuse Online で最後のコネクションを選択するよう要求されます。

手順

1. データを追加するスプレッドシートへのアクセスが承認される Google スプレッドシートコネクションをクリックします。
2. **Choose an Action** ページで **Append values to a sheet** をクリックします。
3. **Append values to a sheet** アクションを設定します。
 - a. **SpreadsheetId** フィールドで以下の1つを行います。
 - 値を追加するスプレッドシートの ID を入力します。
 - フィールドは空白のままにします。この場合、このコネクションをフローに追加した後に、このコネクションの前にデータマッパーステップを追加する必要があります。データマッパーステップでは、スプレッドシート ID を以前のコネクションからこのコネクションにマップします。以前のコネクションはこのコネクションでデータを追加するスプレッドシートにアクセスする必要があります。
 - b. **Range** フィールドに Google A1 表記を使用し、データを追加するスプレッドシートの範囲を指定します。開始および終了座標を指定します。たとえば、**A3:D3** の場合は最初のシートの 4 列の 3 行目の後にデータを追加します。デフォルトは **A:A** で、スプレッドシートの最初のシートで最初の列にデータを追加します。
コネクションはデータを上書きすることはありません。コネクションは指定した範囲の後でデータの追加を開始し、常にデータを適切な場所にある内容に追加します。
 - c. **Major Dimension** フィールドでは、デフォルトである **Rows** を使用するか、**Columns** を選択します。**Rows** は、行オブジェクトを使用してデータを追加するアクションを設定します。各行オブジェクトには、データを追加する各列の値が含まれます。**Columns** は、列オブジェクトを使用してデータを追加するアクションを設定します。各列オブジェクトには、追加する各行の値が含まれます。

- d. **Value Input Option** フィールドには、シートに追加するために受信するデータを Google スプレッドシートが解釈する方法を示します。デフォルトの **Unspecified** を使用すると、Google スプレッドシートは認識するデータを自動的に変換します。たとえば、入力データが日付である場合、Google スプレッドシートは日付としてフォーマットします。入力データが10進数の場合は、Google スプレッドシートは10進数としてフォーマットします。
 - **Unspecified** のデフォルトは Google Sheets API 設定で、このデフォルトは **User entered** になります。
 - **Raw** を選択すると何もしません。Google スプレッドシートは入力データをそのまま挿入します。
 - **User entered** は、認識可能なデータの自動変換を有効にします。
 - e. **Done** をクリックして、コネクションをフローに追加します。インテグレーション可視化フローで追加した場所にコネクションが表示されます。
Append values to a sheet アクションを設定するために指定した値は、このコネクションをマッピング先またはマッピング元とするためにデータマッパーが表示するフィールドを判断します。
4. フローに他のコネクションを追加する場合はここで追加し、これらの手順に戻ります。
 5. フローにすべてのコネクションを追加したら、インテグレーション可視化フローで、データをシートに追加する Google スプレッドシートコネクションの直前にあるプラスマークをクリックします。
 6. **Data Mapper** カードをクリックして、データマッピングステップをフローに追加します。
 7. データマッパーで以下を行います。
 - a. **Append values to a spreadsheet** アクションを設定したときにスプレッドシート ID を指定しなかった場合、ソース **spreadsheetId** をターゲット **spreadsheetId** にマップします。
 - b. フィールドをソースステップからターゲットスプレッドシートにマッピングして、シートに追加するデータを指定します。たとえば、フィールドを別のスプレッドシートやデータベースからマッピングする場合があります。
 必要な場合、シートの値を追加する Google スプレッドシートコネクションを編集し、**Range** または **Major Dimension** の設定を変更することができます。これらの設定を変更すると、データマッパーは変更に応じて異なるターゲットフィールドを表示します。
 - c. 右上の **Done** をクリックし、データマッパーステップを追加します。

8.4.7. スプレッドシートプロパティの更新

インテグレーションでは、フローの途中でスプレッドシートのプロパティを更新したり、シンプルなインテグレーションを終了するためにスプレッドシートでデータを更新することができます。これには、Google スプレッドシートコネクションをフローの途中に追加するか、シンプルなインテグレーションの最後のコネクションとして追加します。

プロパティには、スプレッドシートのタイトル、ロケール、およびタイムゾーンが含まれます。

Fuse Online が **Update spreadsheet properties** アクションの設定を要求したら、一部またはすべてのフィールドを空白のままにしておきます。フィールドを空白のままにした場合、このコネクションの前のフローのデータマッパーステップ (後で追加) で、フィールドを以前のステップから空白の **Update spreadsheet properties** アクション設定フィールドにマップします。

前提条件

- プロパティを更新する Google スプレッドシートへのアクセスが承認されている Google スプレッドシートコネクションが作成されている必要があります。
- プロパティを更新するスプレッドシートのスプレッドシート ID にアクセスできる必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。または、Fuse Online で最後のコネクションを選択するよう要求されます。

手順

1. プロパティを更新するスプレッドシートへのアクセスが承認されている Google スプレッドシートコネクションをクリックします。
2. **Choose an Action** ページで、**Update spreadsheet properties** をクリックします。
3. **Update spreadsheet properties** アクションを設定します。
 - a. **SpreadsheetId** フィールドで以下の1つを行います。
 - プロパティを更新するスプレッドシートの ID を入力します。
 - フィールドは空白のままにします。この場合、このコネクションをフローに追加した後、このコネクションの前にデータマッピングステップを追加する必要があります。データマッピングステップでは、スプレッドシート ID を以前のコネクションからこのコネクションにマップします。以前のコネクションはこのコネクションで更新するスプレッドシートにアクセスする必要があります。
 - b. 別のフィールドには、プロパティを変更する場合のみ値を入力します。また、フィールドを空白のままにすることもできます。空白にする場合は、このコネクションの直前に後で追加するデータマッピングステップで、変更するフィールドをマップできます。
 - c. **Done** をクリックして、コネクションをフローに追加します。インテグレーション可視化フローで追加した場所にコネクションが表示されます。
4. フローに他のコネクションを追加する場合はここで追加し、これらの手順に戻ります。
5. フローにコネクションがすべて追加されたら、データマッピングステップを使用してスプレッドシートプロパティを更新することができます。
 - a. インテグレーション可視化フローで、プロパティを更新する Google スプレッドシートコネクションの直前にあるプラス記号をクリックします。
 - b. **Data Mapper** をクリックして、データマッピングステップをフローに追加します。
 - c. データマッパーで、**Update spreadsheet properties** アクションを設定したときにスプレッドシートを指定しなかった場合は、ソース **spreadsheetId** をターゲット **spreadsheetId** にマップします。
 - d. **Update spreadsheet properties** アクションを設定したときに更新された値を指定しなかった、変更する各スプレッドシートプロパティに対し、フィールドをソースステップから変更するターゲットスプレッドシートプロパティにマップします。
 - e. 右上の **Done** をクリックし、データマッピングステップを追加します。

8.4.8. チャートをシートに追加

フローの途中で基本的なチャートや円グラフを Google のスプレッドシートに追加できます。これには、Google スプレッドシートコネクションをフローの途中に追加するか、シンプルなインテグレーションの最後のコネクションとして追加します。これには、Google スプレッドシートコネクションをフローの途中に追加するか、シンプルなインテグレーションの最後のコネクションとして追加します。データマッピングステップでは、チャートの場所、プロパティ、および内容を判断するオプションを設定します。

Fuse Online が **Add charts** アクションの設定を要求したら、一部またはすべてのフィールドを空白のままにすることができます。フィールドを空白のままにする場合、このコネクションの直前に追加するデータマッピングステップで値をそのフィールドにマップできます。

前提条件

- チャートを追加する Google スプレッドシートへのアクセスが承認されている Google スプレッドシートコネクションが作成されている必要があります。
- チャートを追加するスプレッドシートの ID にアクセスできる必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。Fuse Online で、インテグレーションを追加するよう要求されるか、シンプルなインテグレーションの最後のコネクションを選択するよう要求されます。

手順

1. チャートを追加するスプレッドシートへのアクセスが承認されている Google スプレッドシートコネクションをクリックします。
2. **Choose an Action** ページで **Add charts** をクリックします。
3. **Add charts** アクションを設定します。
 - a. **SpreadsheetId** フィールドで以下の1つを行います。
 - チャートを追加するスプレッドシートの ID を入力します。
 - フィールドは空白のままにします。このコネクションの直前のデータマッピングステップで、スプレッドシート ID を以前のコネクションからこのコネクションにマップします。以前のコネクションは、このコネクションでチャートを追加するスプレッドシートにアクセスする必要があります。
 - b. **Chart Title** フィールドにタイトルを入力します。
このフィールドを空白のままにすることもできます。このコネクションの直前のデータマッピングステップで、チャートのタイトルをソースフィールドまたはプロパティから **title** ターゲットフィールドにマップします。
 - c. **Subtitle** フィールドにサブタイトルを入力します。
このフィールドを空白のままにすることもできます。このコネクションの直前のデータマッピングステップで、チャートのサブタイトルをソースフィールドまたはプロパティから **subtitle** ターゲットフィールドにマップします。
4. **Done** をクリックして、コネクションをフローに追加します。インテグレーション可視化フローで追加した場所にコネクションが表示されます。
5. フローに他のコネクションを追加する場合はここで追加します。その後、これらの手順に戻ります。

6. インテグレーション可視化フローで、チャートを追加するコネクションの直前にあるプラス記号をクリックします。

7. **Data Mapper** をクリックします。

データマッパーで、チャートを追加するターゲット Google スプレッドシートコネクションにソースプロパティ、ソースフィールド、またはソース定数をマップします。以下の手順は、各ターゲットフィールドにマップする値が含まれるソースフィールド、ソースプロパティ、またはソース定数があることを仮定します。

必要な値のソースがない場合、**Properties** フォルダーの右にあるプラス記号をクリックして、データマッパーでソースプロパティを作成します。**Create Property** ダイアログで、正しいターゲットフィールドへのマッピングを容易にする名前をプロパティに付けます。たとえば、**overlayPosition** をプロパティ名とすることができます。プロパティの値には、コネクションがチャートの作成に使用する値を指定します。チャートのオーバーレイポジションフィールドにはセルを特定する A1 表記が必要です。これは、開始ポジションを指定しますが、終了ポジションを指定する必要はなく、終了ポジションがある場合は無視されます。たとえば、**overlayPosition** ソースプロパティの値として **D4** を指定します。新しいプロパティを保存します。

Add charts アクションを設定したときに、スプレッドシート ID、チャートのタイトル、およびチャートのサブタイトルの値を指定した場合は、このフィールドに値をマップしないでください。

8. チャートをシートに追加するには、以下のようにソースプロパティ、ソースフィールド、またはソース定数をターゲットフィールドにマップします。
 - a. セルを識別する A1 表記を **overlayPosition** ターゲットフィールドにマップして、チャートの場所を設定します。終了ポジションは必要なく、指定があっても無視されます。マップされた値は、Google スプレッドシートがチャートを配置するセルを特定します。このセルはチャート表示の左上隅になります。
overlayPosition フィールドに値をマップせず、チャートを追加するシートを特定する **sheetId** フィールドにも値をマップしないと、Google スプレッドシートが新しいシートを作成し、チャートを A1 に配置するのがデフォルトの動作になります。
sheetId フィールドに値をマップする場合は、**overlayPosition** フィールドに値をマップする必要があります。
 - b. ソース **spreadsheetId** フィールドをターゲット **spreadsheetId** フィールドにマップします。コネクションはこのスプレッドシートのソースデータを使用して、チャートを同じスプレッドシートのシートに追加します。ソースデータとチャートは同じスプレッドシートの異なるシートに配置することができますが、異なるスプレッドシートに配置することはできません。
 - c. ターゲット **title** フィールドにマッピングして、チャートタイトルを設定します。コネクションはこの文字列をチャートのタイトルとして挿入します。
 - d. ターゲット **subtitle** フィールドにマッピングして、チャートのサブタイトルを設定します。コネクションはこの文字列をチャートのサブタイトルとして挿入します。
 - e. 整数を **sheetId** フィールドにマップして、チャートを追加するシートを特定します。
sheetId フィールドに値をマップする場合、**overlayPosition** フィールドに値をマップする必要もあります。

sheetId フィールドに値をマップせず、**overlayPosition** フィールドにも値をマップしない場合、コネクションは新しいシートを作成し、チャートをその新しいシートに追加します。

`sheetId` フィールドに値をマップせずに、`overlayPosition` フィールドに値をマップした場合、コネクションは `sourceSheetId` フィールドによって特定されたシートにチャートを配置します。このシートはチャートのデータを提供します。

0 はスプレッドシートの最初のシートを特定します。後続のシートは、シートをブラウザーで見るときに URL の最後の部分がシート ID となります。たとえば、以下の URL の最後には **gid=206589330** があります。

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

これは、シート ID が **206589330** であることを示しています。これが `sheetId` にマップする値になります。

- f. 整数のシート ID をターゲット `sourceSheetId` フィールドにマップして、チャートのデータを提供するシートを特定します。デフォルトは **0** で、スプレッドシートの最初のシートを特定します。
基本のチャートまたは円グラフを追加できます。追加するチャートの種類によって、以下の手順の1つにしたがいます。
- g. 基本のチャートを追加するには、ターゲットパネルで `basicChart` フォルダーを展開し、ソースフィールドを `basicChart` のターゲットフィールドにマップします。
 - i. ターゲット `type` フィールドにマップして、チャートの種類 (**BAR**、**LINE**、**AREA**、または **COLUMN**) を設定します。デフォルトは **COLUMN** です。
 - ii. ターゲット `axisTitleBottom` フィールドにマップして、チャートの横軸のタイトルを設定します。
 - iii. ターゲット `axisTitleLeft` フィールドにマップして、チャートの縦軸のタイトルを設定します。
 - iv. ターゲット `domainRange` フィールドにマップして、チャートが表す情報の上位カテゴリを設定します。このフィールドは Google スプレッドシートの A1 表記を使用します。たとえば、ソースシートの最初の列が **A2** から **A6** までの5つの製品のリストを提供し、チャートが各製品の一部のデータを示すこととします。この場合、この列の最初のセル (**A1**) には **Products** のようなラベルが含まれる必要があります。`domainRange` の設定は **A1:A6** になります。チャートには **A2** から **A6** までのデータが含まれるようになります。チャートにはラベルは表示されません。
 - v. ターゲット `dataRange` フィールドをマップして、チャートが示すソースデータの場所を特定します。このフィールドは Google スプレッドシートの A1 表記を使用します。チャートは1系列のデータのみを提供します。`domainRange` の例を引き続き使用し、最初の列にリストされた各5製品の販売数をソースシートの2番目の列に表し、このデータをチャートに示すこととします。この場合、この列の最初のセル (**B1**) には **Number Sold** のようなラベルが含まれる必要があります。`dataRange` の設定は **B1:B6** になります。チャートには **B2** から **B6** までのデータが含まれるようになります。チャートにはラベルは表示されません。
- h. 円グラフを追加するには、ターゲットパネルで `pieChart` フォルダーを展開し、ソースフィールドを `pieChart` フォルダーのターゲットフィールドにマップします。
 - i. ターゲット `legendPosition` フィールドにマップして、円グラフの凡例の場所を設定します。デフォルトは **LEFT_LEGEND** です。この値は、**BOTTOM_LEGEND**、**LEFT_LEGEND**、**RIGHT_LEGEND**、**TOP_LEGEND**、または **NO_LEGEND** である必要があります。
 - ii. ターゲット `domainRange` フィールドにマップして、円グラフが表す情報の上位カテゴリ

リーを設定します。このフィールドは Google スプレッドシートの A1 表記を使用します。たとえば、ソースシートの最初の列が **A2** から **A6** までの 5 つの製品のリストを提供し、チャートが各製品の一部のデータを示すこととします。この場合、この列の最初のセル (**A1**) には **Products** のようなラベルが含まれる必要があります。**domainRange** の設定は **A1:A6** になります。チャートには **A2** から **A6** までのデータが含まれるようになります。チャートにはラベルは表示されません。

- iii. ターゲット **dataRange** フィールドをマップして、チャートが示すソースデータの場所を特定します。このフィールドは Google スプレッドシートの A1 表記を使用します。チャートは 1 系列のデータのみを提供します。**domainRange** の例を引き続き使用し、最初の列にリストされた各 5 製品の販売数をソースシートの 2 番目の列に表し、このデータをチャートに示すこととします。この場合、この列の最初のセル (**B1**) には **Number Sold** のようなラベルが含まれる必要があります。**dataRange** の設定は **B1:B6** になります。チャートには **B2** から **B6** までのデータが含まれるようになります。チャートにはラベルは表示されません。

- i. 右上の **Done** をクリックし、データマッパーステップを追加します。

8.4.9. ピボットテーブルをシートに追加

フローの途中でピボットテーブルを Google のスプレッドシートに追加できます。また、シンプルなインテグレーションを完了するために追加することもできます。ピボットテーブルではスプレッドシートデータを集計または並べ替えでき、スプレッドシートデータに関数を適用することもできます。ピボットテーブルを追加するには、Google スプレッドシートコネクションをフローの途中に追加するか、シンプルなインテグレーションの最後のコネクションとして追加します。その後、データマッピングステップをコネクションの前に追加します。データマッピングステップでは、ピボットテーブルの場所、プロパティ、および内容を判断するオプションを設定します。

本リリースでは、**Add pivot table** アクションは 1 つの値グループ、1 つの行ピボットグループ、および 1 つの列ピボットグループの定義に限定されます。単一のアクションでの複数グループのサポートは、今後のリリースで追加される予定です。回避策として、同じソーススプレッドシートを基にしてピボットテーブルを追加する複数の Google スプレッドシートコネクションを追加することができます。

前提条件

- ピボットテーブルを追加する Google スプレッドシートへのアクセスが承認されている Google スプレッドシートコネクションが作成されている必要があります。
- ピボットテーブルのソースデータが含まれるスプレッドシートの ID にアクセスできる必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。Fuse Online で、インテグレーションを追加するよう要求されるか、シンプルなインテグレーションの最後のコネクションを選択するよう要求されます。

手順

1. ピボットテーブルを追加するスプレッドシートへのアクセスが承認されている Google スプレッドシートコネクションをクリックします。
2. **Choose an Action** ページで **Add pivot tables** をクリックします。
3. **Add pivot tables** アクションを設定するには、**SpreadsheetId** フィールドで以下の 1 つを行います。
 - ピボットテーブルを追加するスプレッドシートの ID を入力します。

- フィールドは空白のままにします。この接続の直前に追加するデータマップス
テップで、スプレッドシート ID を以前の接続からこの接続にマップしま
す。以前の接続は、この接続でピボットテーブルを追加するスプレッド
シートにアクセスする必要があります。
4. **Done** をクリックして、接続をフローに追加します。インテグレーション可視化フ
ローで追加した場所に接続が表示されます。
 5. このフローに他の接続を追加する場合はここで追加します。その後、これらの手順に
戻ります。
 6. インテグレーション可視化フローで、ピボットテーブルをシートに追加する接続の直
前にあるプラス記号をクリックします。
 7. **Data Mapper** をクリックします。

データマッパーで、ピボットテーブルを追加するをターゲット Google スプレッドシートコネ
クションにソースプロパティ、ソースフィールド、またはソース定数をマップします。以下
の手順は、各ターゲットフィールドにマップする値が含まれるソースフィールド、ソースプロ
パティ、またはソース定数があることを仮定します。

必要な値のソースがない場合、**Properties** フォルダーの右にあるプラス記号をクリックして、
データマッパーでソースプロパティを作成します。**Create Property** ダイアログで、正しい
ターゲットフィールドへのマッピングを容易にする名前をプロパティに付けます。たとえ
ば、**valueLayout** をプロパティ名とすることができます。ピボットテーブルを作成するた
めに接続が使用する値を指定します。ピボットテーブルの値レイアウトフィールドに
は、**HORIZONTAL** または **VERTICAL** の値が必要であるため、**VERTICAL** を **valueLayout** プ
ロパティの値として指定することがあります。新しいプロパティを保存します。

8. データマッパーで、以下のように新しいピボットテーブルを設定します。

- a. ソース **spreadsheetId** フィールドをターゲット **spreadsheetId** フィールドにマップしま
す。接続はこのスプレッドシートのソースデータを使用して、ピボットテーブル
を同じスプレッドシートのシートに追加します。ソースデータとピボットテーブルは同じ
スプレッドシートの異なるシートに配置することができますが、異なるスプレッドシート
に配置することはできません。
- b. 整数を **sheetId** フィールドにマップして、ピボットテーブルを追加するシートを特定しま
す。デフォルトは **0** で、スプレッドシートの最初のシートを特定します。後続のシート
は、シートをブラウザーで見るときに URL の最後の部分がシート ID となります。たとえ
ば、以下の URL の最後には **gid=206589330** があります。

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

これは、シート ID が **206589330** であることを示しています。これが **sheetId** にマップす
る値になります。

- c. 整数のシート ID をターゲット **sourceSheetId** フィールドにマップして、ピボットテーブル
を生成するためのデータを提供するシートを特定します。**sourceSheetId** に値をマップし
ないと、接続は **sheetId** の値を使用します。**sheetId** にマップされた値がない場
合は、**0** を使用します。**0** はスプレッドシートの最初のシートを特定します。
- d. A1 表記の値をターゲット **sourceRange** フィールドにマップして、ピボットテーブルの生
成にソースデータの範囲を設定します。たとえば、**A2:D5** を設定します。
- e. ターゲット **valueLayout** フィールドにマップして、ピボットテーブルで使用する値のレ
イアウトを設定します。値は、**VERTICAL** または デフォルトの **HORIZONTAL** である必要
があります。**HORIZONTAL** を指定すると、値は列としてレイアウトされます。

- f. A1 表記のセル座標をターゲット **start** フィールドにマップして、ピボットテーブルの左上のセルを設定します。コネクションがピボットテーブルをソースデータが含まれるシートに追加するとき、値を **start** フィールドにマップしないと、デフォルトではピボットテーブルの左上のセルがソースデータ範囲の最初の行にあるセルとなり、ソースデータ範囲の後の最初の空白列にあるセルとなります。ソースデータがあるシートとピボットテーブルがあるシートが異なる場合、デフォルトは **A1** です。
- g. 値グループを追加するには、ターゲットパネルで **valueGroups** 展開し、**valueGroups** 以下で値をフィールドにマップします。
 - i. 値グループ名をターゲット **name** フィールドにマップします。
 - ii. 値グループに適用する関数をターゲット **function** フィールドにマップします。この値は、**SUM**、**COUNT**、**AVERAGE**、**MAX**、**MIN**、または **CUSTOM** である必要があります。デフォルトは **SUM** です。式を定義するときに **CUSTOM** を **function** にマップします。
 - iii. 値 **CUSTOM** を **function** フィールドにマップした場合、値グループに適用するカスタムの式をターゲット **formula** フィールドにマップします。
 - iv. 値グループを構築する座標の列名をターゲット **sourceColumn** フィールドにマップします。
- h. 行ピボットグループを追加するには、ターゲットパネルで **columnGroups** 展開し、**columnGroups** 以下で値をフィールドにマップします。
 - i. 行ピボットグループ名をターゲット **label** フィールドにマップします。
 - ii. 行ピボットグループに適用する並べ替え順序をターゲット **sortOrder** フィールドにマップします。この値は、**ASCENDING** または **DESCENDING** である必要があります。デフォルトは **ASCENDING** です。
 - iii. **true** または **false** をターゲット **showTotals** フィールドにマップします。デフォルトは **true** で、行ピボットグループの合計を表示できます。
 - iv. 行ピボットグループを構築する座標の列名をターゲット **sourceColumn** フィールドにマップします。
- i. 列ピボットグループを追加するには、ターゲットパネルで **rowGroups** 展開し、**rowGroups** 以下で値をフィールドにマップします。
 - i. 列ピボットグループ名をターゲット **label** フィールドにマップします。
 - ii. 列ピボットグループに適用する並べ替え順序をターゲット **sortOrder** フィールドにマップします。この値は、**ASCENDING** または **DESCENDING** である必要があります。デフォルトは **ASCENDING** です。
 - iii. **true** または **false** をターゲット **showTotals** フィールドにマップします。デフォルトは **true** で、列ピボットグループの合計を表示できます。
 - iv. 列ピボットグループを構築する座標の列名をターゲット **sourceColumn** フィールドにマップします。
- j. 右上の **Done** をクリックし、データマッパーステップを追加します。

その他のリソース

[ピボットテーブルに関する開発者向け Google ドキュメント](#)

第9章 HTTP および HTTPS エンドポイントへの接続

インテグレーションでは、HTTP または HTTPS エンドポイントへ接続

し、**GET**、**PUT**、**POST**、**DELETE**、**HEAD**、**OPTIONS**、**TRACE**、**PATCH** メソッドを実行できます。これを行うには、HTTP または HTTPS コネクションを作成し、インテグレーションフローに追加します。詳細は以下を参照してください。

- [「HTTP または HTTPS エンドポイントへのコネクションの作成」](#)
- [「HTTP または HTTPS コネクションのインテグレーションへの追加」](#)

9.1. HTTP または HTTPS エンドポイントへのコネクションの作成

インテグレーションで HTTP の **GET**、**PUT**、**POST**、**DELETE**、**HEAD**、**OPTIONS**、**TRACE**、または **PATCH** メソッドを実行するには、HTTP または HTTPS エンドポイントへのコネクションを作成します。作成後、コネクションを1つ以上のインテグレーションに追加できます。

前提条件

接続するエンドポイントの URL を知っている必要があります。

手順

1. Fuse Online の左パネルで **Connections** をクリックし、利用可能なコネクションを表示します。
2. 右上にある **Create Connection** をクリックし、Fuse Online コネクタを表示します。
3. HTTP (Hyper Text Transfer Protocol) を使用してエンドポイントに接続する場合は、**HTTP** コネクタを使用します。HTTPS (Hyper Text Transfer Protocol Secure) を使用する場合は、**HTTPS** コネクタをクリックします。
4. **Base URL** フィールドにエンドポイントパスを入力します。例: **www.mycompany.com/sales**
5. **Validate** をクリックします。Fuse Online は即座にコネクションを検証しようとし、検証の成功または失敗を示すメッセージを表示します。検証に失敗した場合は、指定したベース URL を訂正し、再度検証を行います。
6. 検証に成功した場合は **Next** をクリックします。
7. **Connection Name** フィールドに、このコネクションを別のコネクションと区別するために使用する名前を入力します。たとえば、**HTTPS My Company Sales** を入力します。
8. **Description** は任意のフィールドで、このコネクションに関する便利な情報を入力します。
9. 右上にある **Create** をクリックし、作成したコネクションが利用可能になったことを確認します。上記の例の名前を入力した場合は、インテグレーションに追加できるコネクションとして **HTTPS My Company Sales** が表示されます。

9.2. HTTP または HTTPS コネクションのインテグレーションへの追加

HTTP または HTTPS コネクションを任意の数のインテグレーションに追加できます。

前提条件

- HTTP または HTTPS コネクションが作成されている必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。Fuse Online はインテグレーションの追加を要求します。または、Fuse Online は最初のコネクションまたは最後のコネクションの選択を要求します。

手順

1. フローに追加する HTTP または HTTPS コネクションをクリックします。
2. コネクションが実行するアクションを選択します。
 - 最初のコネクションを追加する場合、使用できるアクションは **Periodic invoke URL** のみになります。このアクションは、指定する間隔でエンドポイントを呼び出し、エンドポイントがデータを返すとインテグレーションをトリガーします。
 - 途中または最後のコネクションを追加する場合、使用できるアクションは **Invoke URL** のみです。このアクションはエンドポイントを一度呼び出します。
3. **URL Path** フィールドに、呼び出すエンドポイントの場所を指定します。
4. **HTTP Method** フィールドでコネクションが実行するメソッドを選択します。デフォルトのメソッドは **GET** です。
 - **GET** は、URL パスでコンテンツを取得します。
 - **PUT** は、URL パスでコンテンツをインテグレーションデータに置き換えます。
 - **POST** は、URL パスのインテグレーションデータを格納し、新しいコンテンツを作成します。
 - **DELETE** は URL パスのコンテンツを削除します。
 - **HEAD** は、URL パスでコンテンツに関するメタデータを取得します。
 - **OPTIONS** は、URL パスで通信オプション設定を取得します。
 - **TRACE** は、テストおよび診断の目的で情報を取得します。
 - **PATCH** は、インテグレーションデータに応じて URL パスでコンテンツを部分的に更新します。
5. URL を定期的に呼び出す最初のコネクションを追加する場合、**Period** フィールドにデフォルトの間隔である 1 秒を使用するか、数値と単位 (ミリ秒、秒、分、または時間) を指定して呼び出しの間の待ち時間を示します。
6. **Done** をクリックして、アクションの入力または出力タイプを指定します。
7. データタイプの認識が必要がない場合は、**Select Type** フィールドで **Type specification not required** を受け入れ、下部にある **Done** をクリックします。残りの手順を行う必要はありません。
その他の場合は、以下の 1 つをスキーマタイプとして選択します。
 - **JSON schema** は JSON データの構造を記述するドキュメントです。ドキュメントのメディアタイプは **application/schema+json** です。

- **JSON instance** は JSON データが含まれるドキュメントです。ドキュメントのメディアタイプは **application/json** です。
 - **XML schema** は XML データの構造を記述するドキュメントです。このドキュメントのファイル拡張子は **.xsd** です。
 - **XML instance** は XML データが含まれるドキュメントです。このドキュメントのファイル拡張子は **.xml** です。
8. **Definition** 入力ボックスに、選択したスキーマタイプに準拠する定義を貼り付けます。たとえば、**JSON schema** を選択した場合は、メディアタイプが **application/schema+json** の JSON スキーマファイルの内容を貼り付けます。
 9. **Data Type Name** フィールドにはデータタイプに選択した名前を入力します。たとえば、ベンダーの JSON スキーマを指定する場合は、データ型名として **Vendor** を指定できます。このタイプを指定するコネクションを使用するインテグレーションを作成または編集するときにこのデータタイプ名が表示されます。Fuse Online では、インテグレーション仮想化パネルとデータマッパーにタイプ名が表示されます。
 10. **Data Type Description** フィールドには、このタイプを区別するのに便利な情報を提供します。データマッパーでこのタイプを処理するステップにカーソルを合わせると、この説明が表示されます。
 11. **Done** をクリックします。

結果

インテグレーションフローで追加した場所にコネクションが表示されます。

第10章 IRC への接続

IRC (Internet Relay Chat) は、IRC クライアントを使って誰でもリアルタイムでテキストメッセージを送受信できるプロトコルです。インテグレーションでは、IRC へのコネクションは以下のいずれかを行うことができます。

- 指定した IRC ニックネームがプライベートメッセージを受信したときにシンプルなインテグレーションの実行をトリガーします。インテグレーションはメッセージをフローの次のステップに渡します。たとえば、見込み顧客の名前を受信する IRC ニックネームを指定することができます。インテグレーションはメッセージを受信すると Salesforce に接続してリードを作成することができます。
- 特定のチャンネルへメッセージを配信します。メッセージは指定するニックネームから配信されます。たとえば、インテグレーションが FTP サーバーからファイルをダウンロードし、何らかの方法で処理するときに便利です。インテグレーションフローはメッセージを特定のチャンネルに送信し、プロセスの成功を示すことができます。

IRC をインテグレーションに接続するには、IRC コネクションを作成し、コネクションをインテグレーションフローに追加します。詳細は以下を参照してください。

- [「IRC コネクションの作成」](#)
- [「メッセージの受信時にインテグレーションの実行をトリガーするために IRC コネクションを追加」](#)
- [「IRC コネクションを追加してメッセージを送信」](#)

10.1. IRC コネクションの作成

インテグレーションでは、IRC コネクションは指定した IRC ニックネームに送信されたメッセージを取得したり、チャンネルでニックネームにメッセージを送信することができます。任意の数のインテグレーションで同じ IRC コネクションを使用できます。

前提条件

- 接続先の IRC サーバーのホスト名およびポートを知っている必要があります。
- IRC サーバーにパスワードが必要な場合は、そのパスワードを知っている必要があります。

手順

1. Fuse Online の左パネルで **Connections** をクリックし、利用可能なコネクションを表示します。
2. 右上にある **Create Connection** をクリックし、Fuse Online コネクタを表示します。
3. IRC コネクタをクリックします。
4. **Hostname** フィールドに IRC ホスト名を入力する必要があります。IP アドレスまたは名前 (例: **chat.freenode.net**) を指定することができます。
5. **Port** フィールドに IRC サーバーがリッスンしているポート (例: **6665**) を入力します。
6. IRC サーバーにパスワードが必要な場合は **Password** フィールドにそのパスワードを入力します。

7. **Validate** をクリックします。Fuse Online は即座にコネクションを検証しようとし、検証の成功または失敗を示すメッセージを表示します。検証に失敗した場合は、コネクション設定値を修正し、再度検証を行います。
8. 検証に成功した場合は **Next** をクリックします。
9. **Connection Name** フィールドに、このコネクションを別のコネクションと区別するために使用する名前を入力します。たとえば、**IRC Freenode** を入力します。
10. **Description** は任意のフィールドで、このコネクションに関する便利な情報を入力します。
11. 右上にある **Create** をクリックし、作成したばかりのコネクションを含む、利用可能なコネクションの更新済みリストを表示します。上記の例の名前を入力した場合は、インテグレーションに追加できるコネクションとして **IRC Freenode** が表示されます。このコネクションを任意の数のコネクションに追加できます。

10.2. メッセージの受信時にインテグレーションの実行をトリガーするために IRC コネクションを追加

指定した IRC ニックネームに送信されたプライベートメッセージをコネクションが発見すると、インテグレーションを開始する IRC コネクションはインテグレーションの実行をトリガーします。

前提条件

- IRC コネクションが作成されている必要があります。
- メッセージの送受信するための IRC ニックネームが必要です。
- シンプルなインテグレーションを作成することになります。Fuse Online はインテグレーションの開始方法の選択を要求します。

手順

1. インテグレーションの開始に使用する IRC コネクションをクリックします。
2. **IRC Receive Private Message** アクションをクリックします。
3. **Channels** フィールドに任意で1つ以上の IRC チャンネルをコンマで区切って指定します。
4. **Nickname** フィールドには、インテグレーションが使用する IRC ニックネームを指定します。このニックネームは指定されたすべてのチャンネルに参加します。実行中、インテグレーションはこのニックネームに送信されたメッセージを取得し、インテグレーションの次のステップに渡します。
5. 指定のニックネームにパスワードがある場合は **NickServ Password** フィールドにパスワードを入力します。
6. **Next** をクリックして、アクションの出力タイプを指定します。
7. データタイプの認識が必要がない場合は、**Select Type** フィールドで **Type specification not required** を受け入れ、下部にある **Done** をクリックします。残りの手順を行う必要はありません。
その他の場合は、以下の1つをスキーマタイプとして選択します。
 - **JSON schema** は JSON データの構造を記述するドキュメントです。ドキュメントのメディアタイプは **application/schema+json** です。

- **JSON instance** は JSON データが含まれるドキュメントです。ドキュメントのメディアタイプは **application/json** です。
 - **XML schema** は XML データの構造を記述するドキュメントです。このドキュメントのファイル拡張子は **.xsd** です。
 - **XML instance** は XML データが含まれるドキュメントです。このドキュメントのファイル拡張子は **.xml** です。
8. **Definition** 入力ボックスに、選択したスキーマタイプに準拠する定義を貼り付けます。たとえば、**JSON schema** を選択した場合は、メディアタイプが **application/schema+json** の JSON スキーマファイルの内容を貼り付けます。
 9. **Data Type Name** フィールドにはデータタイプに選択した名前を入力します。たとえば、ベンダーの JSON スキーマを指定する場合は、**Vendor** をデータタイプ名として指定することができます。
このタイプを指定するコネクションを使用するインテグレーションを作成または編集するときこのデータタイプ名が表示されます。Fuse Online では、インテグレーション仮想化パネルとデータマッパーにタイプ名が表示されます。
 10. **Data Type Description** フィールドには、このタイプを区別するのに便利な情報を提供します。データマッパーでこのタイプを処理するステップにカーソルを合わせると、この説明が表示されます。
 11. **Done** をクリックして、コネクションをフローに追加します。

結果

インテグレーションフローの最初にコネクションが表示されます。

10.3. IRC コネクションを追加してメッセージを送信

インテグレーションでは、フローの途中でメッセージを1つ以上の IRC チャンネルに送信したり、シンプルなインテグレーションを終了するためにメッセージを1つ以上の IRC チャンネルに送信することができます。これには、IRC コネクションをフローの途中に追加するか、シンプルなインテグレーションの最後のコネクションとして追加します。

前提条件

- IRC コネクションが作成されている必要があります。
- メッセージの送受信するための IRC ニックネームが必要です。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。または、Fuse Online で最後のコネクションを選択するよう要求されます。

手順

1. フローに追加する IRC コネクションをクリックします。
2. **IRC Send Message to Channel** をクリックして、メッセージを1つ以上のチャンネルに送信します。
 - **Channels** フィールドに、コンマで区切られた1つ以上の IRC チャンネルを入力します。実行中、このコネクションは指定した IRC ニックネームからこれらのチャンネルにメッセージを送信します。

3. **Nickname** フィールドに、指定されたチャンネルにメッセージを送信する IRC ニックネームを指定します。
4. 指定したニックネームにパスワードがある場合は、**NickServ password** フィールドにそのパスワードを指定します。
5. **Next** をクリックしてアクションの入力タイプを指定します。
6. データタイプの認識が必要がない場合は、**Select Type** フィールドで **Type specification not required** を受け入れ、下部にある **Done** をクリックします。残りの手順を行う必要はありません。
その他の場合は、以下の1つをスキーマタイプとして選択します。
 - **JSON schema** は JSON データの構造を記述するドキュメントです。ドキュメントのメディアタイプは **application/schema+json** です。
 - **JSON instance** は JSON データが含まれるドキュメントです。ドキュメントのメディアタイプは **application/json** です。
 - **XML schema** は XML データの構造を記述するドキュメントです。このドキュメントのファイル拡張子は **.xsd** です。
 - **XML instance** は XML データが含まれるドキュメントです。このドキュメントのファイル拡張子は **.xml** です。
7. **Definition** 入力ボックスに、選択したスキーマタイプに準拠する定義を貼り付けます。たとえば、**JSON schema** を選択した場合は、メディアタイプが **application/schema+json** の JSON スキーマファイルの内容を貼り付けます。
8. **Data Type Name** フィールドにはデータタイプに選択した名前を入力します。たとえば、ベンダーの JSON スキーマを指定する場合は、**Vendor** をデータタイプ名として指定することができます。
このタイプを指定するコネクションを使用するインテグレーションを作成または編集するときにこのデータタイプ名が表示されます。Fuse Online では、インテグレーション仮想化パネルとデータマッパーにタイプ名が表示されます。
9. **Data Type Description** フィールドには、このタイプを区別するのに便利な情報を提供します。データマッパーでこのタイプを処理するステップにカーソルを合わせると、この説明が表示されます。
10. **Done** をクリックして、コネクションをフローに追加します。

結果

フローで追加した場所にコネクションが表示されます。

第11章 KAFKA への接続

Apache Kafka はデータの取得およびパブリッシュに使用できる分散ストリーミングプラットフォームです。インテグレーションでは、指定した Kafka トピックからデータをサブスクライブでき、または指定した Kafka トピックにデータをパブリッシュすることもできます。これには、Kafka へのコネクションを作成し、そのコネクションをインテグレーションフローに追加します。詳細は以下を参照してください。

- [「Kafka ブローカーへのコネクションの作成」](#)
- [「Kafka ブローカーからデータを取得してインテグレーションの実行をトリガー」](#)
- [「Kafka ブローカーへのデータのパブリッシュ」](#)

11.1. KAFKA ブローカーへのコネクションの作成

インテグレーションで、Kafka トピックからデータをサブスクライブしたり、Kafka トピックにデータをパブリッシュするには、Kafka へのコネクションを作成し、そのコネクションをインテグレーションに追加します。

前提条件

接続する Kafka ブローカーの URI を知っている必要があります。

手順

1. Fuse Online の左パネルで **Connections** をクリックし、利用可能なコネクションを表示します。
2. 右上の **Create Connection** をクリックしてコネクタを表示します。
3. **Kafka** コネクタをクリックします。
4. **Kafka bootstraps URI** フィールドに、Kafka ブローカー URI のコンマ区切りリストを入力します。各 URI は **host:port** 形式で指定する必要があります。
5. **Validate** をクリックします。Fuse Online は即座にコネクションを検証しようとし、検証の成功または失敗を示すメッセージを表示します。検証に失敗した場合は、入力パラメーターを訂正し、再度検証を行います。
6. 検証に成功した場合は **Next** をクリックします。
7. **Connection Name** フィールドに、このコネクションを別のコネクションと区別するために使用する名前を入力します。たとえば、**Kafka West** を入力します。
8. **Description** は任意のフィールドで、このコネクションに関する便利な情報を入力します。
9. 右上にある **Create** をクリックし、作成したコネクションが利用可能になったことを確認します。上記の例の名前を入力した場合は、インテグレーションに追加できるコネクションとして **Kafka West** が表示されます。

11.2. KAFKA ブローカーからデータを取得してインテグレーションの実行をトリガー

Kafka ブローカーからデータを受信したときにインテグレーションの実行をトリガーするには、Kafka

コネクションを最初のコネクションとして追加します。インテグレーションの実行中、Kafka コネクションは指定した Kafka トピックでデータを継続的に監視します。コネクションが新しいデータを見つけると、そのデータをインテグレーションの次のステップに渡します。

前提条件

Kafka ブローカーへのコネクションが作成されている必要があります。

手順

1. Fuse Online パネルの左側にある **Integrations** をクリックします。
2. **Create Integration** をクリックします。
3. **Choose a Start Connection** ページで、インテグレーションを開始するために使用する Kafka コネクションをクリックします。
4. **Choose an Action** ページで、**Subscribe for messages** アクションをクリックし、指定したトピックからデータを受信します。
5. **Topic Name** フィールドで、下向きの矢印をクリックしてトピックのリストを表示し、サブスクライブするトピックをクリックします。
6. **Next** をクリックして、アクションの出力タイプを指定します。
7. データタイプの認識が必要がない場合は、**Select Type** フィールドで **Type specification not required** を受け入れ、下部にある **Done** をクリックします。残りの手順を行う必要はありません。
その他の場合は、以下の1つをスキーマタイプとして選択します。
 - **JSON schema** は JSON データの構造を記述するドキュメントです。ドキュメントのメディアタイプは **application/schema+json** です。
 - **JSON instance** は JSON データが含まれるドキュメントです。ドキュメントのメディアタイプは **application/json** です。
 - **XML schema** は XML データの構造を記述するドキュメントです。このドキュメントのファイル拡張子は **.xsd** です。
 - **XML instance** は XML データが含まれるドキュメントです。このドキュメントのファイル拡張子は **.xml** です。
8. **Definition** 入力ボックスに、選択したスキーマタイプに準拠する定義を貼り付けます。たとえば、**JSON schema** を選択した場合は、メディアタイプが **application/schema+json** の JSON スキーマファイルの内容を貼り付けます。
9. **Data Type Name** フィールドにはデータタイプに選択した名前を入力します。たとえば、ベンダーの JSON スキーマを指定する場合は、データ型名として **Vendor** を指定できます。
このタイプを指定するコネクションを使用するインテグレーションを作成または編集するときこのデータタイプ名が表示されます。Fuse Online では、インテグレーション仮想化パネルとデータマッパーにタイプ名が表示されます。
10. **Data Type Description** フィールドには、このタイプを区別するのに便利な情報を提供します。データマッパーでこのタイプを処理するステップにカーソルを合わせると、この説明が表示されます。
11. **Done** をクリックします。

結果

インテグレーションフローの最初に接続が表示されます。

11.3. KAFKA ブローカーへのデータのパブリッシュ

インテグレーションでは、フローの途中で Kafka ブローカーにデータをパブリッシュしたり、シンプルなインテグレーションを終了するためにデータをパブリッシュしたりすることができます。これには、Kafka 接続をフローの途中に追加するか、シンプルなインテグレーションの最後の接続として追加します。

前提条件

- Kafka ブローカーへの接続が作成されている必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。または、Fuse Online で最後の接続を選択するよう要求されます。

手順

1. メッセージのパブリッシュに使用する Kafka 接続をクリックします。
2. **Choose an Action** ページで **Publish** をクリックします。
3. **Topic Name** フィールドで、下向きの矢印をクリックしてトピックのリストを表示し、パブリッシュするトピックをクリックします。
4. **Next** をクリックしてアクションの入力タイプを指定します。
5. データタイプの認識が必要がない場合は、**Select Type** フィールドで **Type specification not required** を受け入れ、下部にある **Done** をクリックします。残りの手順を行う必要はありません。
その他の場合は、以下の1つをスキーマタイプとして選択します。
 - **JSON schema** は JSON データの構造を記述するドキュメントです。ドキュメントのメディアタイプは **application/schema+json** です。
 - **JSON instance** は JSON データが含まれるドキュメントです。ドキュメントのメディアタイプは **application/json** です。
 - **XML schema** は XML データの構造を記述するドキュメントです。このドキュメントのファイル拡張子は **.xsd** です。
 - **XML instance** は XML データが含まれるドキュメントです。このドキュメントのファイル拡張子は **.xml** です。
6. **Definition** 入力ボックスに、選択したスキーマタイプに準拠する定義を貼り付けます。たとえば、**JSON schema** を選択した場合は、メディアタイプが **application/schema+json** の JSON スキーマファイルの内容を貼り付けます。
7. **Data Type Name** フィールドにはデータタイプに選択した名前を入力します。たとえば、ベンダーの JSON スキーマを指定する場合は、データ型名として **Vendor** を指定できます。
このタイプを指定する接続を使用するインテグレーションを作成または編集するときこのデータタイプ名が表示されます。Fuse Online では、インテグレーション仮想化パネルとデータマッパーにタイプ名が表示されます。

8. **Data Type Description** フィールドには、このタイプを区別するのに便利な情報を提供します。データマッパーでこのタイプを処理するステップにカーソルを合わせると、この説明が表示されます。
9. **Done** をクリックします。

結果

インテグレーションフローで追加した場所にコネクションが表示されます。

第12章 APACHE KUDU への接続

Apache Kudu は Apache Hadoop プラットフォーム向けに開発された列指向ストレージです。Fuse Online のインテグレーションは Kudu データストアに接続してテーブルをスキャンでき、テーブルのすべてのレコードをインテグレーションに返すか、レコードをテーブルに挿入します。詳細は以下を参照してください。

- [「Apache Kudu データストアへの接続の作成」](#)
- [「スキャンが Kudu テーブルからレコードを返すときにインテグレーションをトリガー」](#)
- [「Kudu テーブルへのレコードの挿入」](#)

12.1. APACHE KUDU データストアへの接続の作成

インテグレーションで、Kudu テーブルからレコードを取得したり、レコードを Kudu テーブルに挿入するには、Kudu マスターホストへの接続を作成し、その接続をインテグレーションに追加します。

前提条件

- 接続する Kudu マスターホストの IP アドレスとホスト名を知っている必要があります。
- Kudu がリッスンするポートを知っている必要があります。

手順

1. Fuse Online の左パネルで **Connections** をクリックし、利用可能な接続を表示します。
2. 右上の **Create Connection** をクリックしてコネクターを表示します。
3. **Apache Kudu** コネクターをクリックします。
4. 接続を設定します。
 - a. **Address of Kudu master host** フィールドに、Kudu マスターホストのホスト名または IP アドレスを入力します。
 - b. **Port to establish connection to** フィールドに、Kudu がリッスンするポートを入力します。デフォルトは **7051** です。
5. **Next** をクリックします。
6. **Connection Name** フィールドに、この接続を別の接続と区別するために使用する名前を入力します。たとえば、**Kudu North** を入力します。
7. **Description** は任意のフィールドで、この接続に関する便利な情報を入力します。
8. 右上にある **Create** をクリックし、作成した接続が利用可能になったことを確認します。上記の例の名前を入力した場合は、インテグレーションに追加できる接続として **Kudu North** が表示されます。

12.2. スキャンが KUDU テーブルからレコードを返すときにインテグレーションをトリガー

Kudu テーブルからデータを取得したときにインテグレーションの実行をトリガーするには、Kudu コネクションを最初のコネクションとしてシンプルなインテグレーションに追加します。インテグレーションの実行時、Kudu コネクションは指定の間隔で指定のテーブルをスキャンし、レコードのコレクションをインテグレーションの次のステップに渡します。

Kudu コネクションは1つのテーブルからのみデータを取得できます。スキャンの合間に、コネクションがスキャンするテーブルのデータに変更がない場合は、次のスキャンが同じデータを前回のスキャンとして返します。

前提条件

- Kudu コネクションが作成されている必要があります。
- レコードの取得元となるテーブルが存在する必要があります。

手順

1. Fuse Online パネルの左側にある **Integrations** をクリックします。
2. **Create Integration** をクリックします。
3. **Choose a Start Connection** ページで、インテグレーションを開始するために使用する Kudu コネクションをクリックします。
4. **Choose an Action** ページで **Scan a Kudu table** アクションをクリックします。
5. **Table** フィールドに、レコードを取得するテーブルの名前を入力します。
6. **Period** フィールドで、デフォルトの1分を使用するか、コネクションがテーブルをスキャンし、テーブルのレコードを返す間隔を入力します。
7. **Done** をクリックします。

結果

コネクションは、インテグレーションの最初のコネクションとしてインテグレーションフローに表示されます。

次のステップ

インテグレーションの最後のコネクションと、インテグレーションに含まれるようにする他のコネクションを追加します。インテグレーションに必要なすべてのコネクションが含まれる場合、Kudu コネクションが返すレコードのコレクションを分割する必要があるかどうか検討します。Kudu テーブルから取得した各レコードのインテグレーションステップを実行する場合、Kudu コネクションの後に分割ステップを追加します。また、インテグレーションの後続のコネクションで Kudu から取得したデータをフィールドにマップするデータマッピングステップで、Kudu コネクションにしたがう必要がある可能性が高くなります。

12.3. KUDU テーブルへのレコードの挿入

インテグレーションでは、フローの途中で Kudu テーブルにレコードを追加したり、シンプルなインテグレーションを終了するために Kudu テーブルにレコードを追加することができます。これには、Kudu コネクションをフローの途中に追加するか、シンプルなインテグレーションの最後のコネクションとして追加します。

前提条件

- Kudu コネクションが作成されている必要があります。
- インテグレーションを作成または編集することになります。または、Fuse Online で最後のコネクションを選択するよう要求されます。
- レコードの追加先となるテーブルが存在する必要があります。

手順

1. 使用する Kudu コネクションをクリックします。Kudu コネクションがデータを挿入するとき、コネクションは何も返さないことに注意してください。
2. **Choose an Action** ページで **Insert data into a Kudu table** をクリックします。
3. アクションを設定するには、**Table** フィールドでレコードの追加先となるテーブルの名前を指定します。
レコードの追加先となる Kudu テーブルの設定方法を理解することが重要になります。たとえば、レコードの追加先となる Kudu テーブルには一意のキーがある可能性があります。テーブルにすでに存在するキー値が含まれるレコードを追加しようとしても、Kudu コネクションはそのレコードを追加しません。
4. **Done** をクリックします。

結果

インテグレーションフローで追加した場所にコネクションが表示されます。

次のステップ

レコードのコレクションを Kudu コネクションがテーブルに追加できる個別のレコードに分割する必要があるかどうかを検討します。これには、インテグレーションに分割ステップを追加します。分割ステップは、各レコードに対してインテグレーションの後続のステップを一度実行します。また、レコードとテーブルに追加する Kudu コネクションの前にデータマッピングステップが必要になる可能性が高いでしょう。

第13章 MQTT への接続

MQTT (MQ Telemetry Transport) は、ライトウェイトなマシンツーマシンの IoT (Internet of Things、モノのインターネット) 接続プロトコルです。インテグレーションでは、MQTT ブローカーからメッセージを取得したり、MQTT へメッセージをパブリッシュすることができます。これには、対象の MQTT ブローカーへのコネクションを作成し、そのコネクションをインテグレーションフローに追加します。詳細は以下を参照してください。

- 「MQTT ブローカーへのコネクションの作成」
- 「MQTT ブローカーからメッセージを取得してインテグレーションの実行をトリガー」
- 「MQTT ブローカーへのメッセージのパブリッシュ」

13.1. MQTT ブローカーへのコネクションの作成

インテグレーションで、MQTT ブローカーからメッセージと取得したり、MQTT ブローカーにメッセージをパブリッシュするには、対象の MQTT ブローカーへのコネクションを作成し、そのコネクションをインテグレーションに追加します。

前提条件

接続する MQTT ブローカーの URL を知っている必要があります。

手順

1. Fuse Online の左パネルで **Connections** をクリックし、利用可能なコネクションを表示します。
2. 右上の **Create Connection** をクリックしてコネクタを表示します。
3. **MQTT Message Broker** コネクタをクリックします。
4. コネクションを設定します。
 - a. **MQTT broker URL** フィールドにデータの送信先または取得元となる MQTT ブローカーの場所を入力します。必須のフィールドはこのフィールドのみです。
 - b. 任意で、このブローカーにアクセスするために使用するクレデンシャルを持つ MQTT アカウントのユーザー名を **User Name** フィールドに入力します。
 - c. ユーザー名を指定した場合、**Password** フィールドにそのアカウントに関連するパスワードを指定します。
 - d. 任意で、**Client ID** フィールドに、メッセージを見逃さずにコネクションを開閉できるようにする ID を入力します。コネクションには、トピックのサブスクライブまたはパブリッシュが必要です。
5. **Validate** をクリックします。Fuse Online は即座にコネクションを検証しようとし、検証の成功または失敗を示すメッセージを表示します。検証に失敗した場合は、入力パラメーターを訂正し、再度検証を行います。
6. 検証に成功した場合は **Next** をクリックします。
7. **Connection Name** フィールドに、このコネクションを別のコネクションと区別するために使用する名前を入力します。たとえば、**MQTT West** を入力します。

8. **Description** は任意のフィールドで、このコネクションに関する便利な情報を入力します。
9. 右上にある **Create** をクリックし、作成したコネクションが利用可能になったことを確認します。上記の例の名前を入力した場合は、インテグレーションに追加できるコネクションとして **MQTT West** が表示されます。

13.2. MQTT ブローカーからメッセージを取得してインテグレーションの実行をトリガー

MQTT ブローカーからのメッセージの受信により、インテグレーションの実行をトリガーするには、MQTT コネクションを最初のコネクションとして追加します。インテグレーションの実行中、MQTT コネクションは指定した MQTT キューまたはトピックでメッセージを継続的に監視します。コネクションがメッセージを見つけると、そのメッセージをインテグレーションの次のステップに渡します。MQTT コネクションは一度に1つのメッセージを処理します。

前提条件

MQTT コネクションが作成されている必要があります。

手順

1. Fuse Online パネルの左側にある **Integrations** をクリックします。
2. **Create Integration** をクリックします。
3. **Choose a Start Connection** ページで、インテグレーションを開始するために使用する MQTT コネクションをクリックします。
4. **Choose an Action** ページで、**Subscribe** アクションをクリックし、指定したキューまたはトピックからメッセージを受信します。
5. **MQTT queue/topic name** フィールドに、データを受信するためにサブスクライブするキューまたはトピックの名前を入力します。
6. **Next** をクリックして、アクションの出力タイプを指定します。
7. データタイプの認識が必要がない場合は、**Select Type** フィールドで **Type specification not required** を受け入れ、下部にある **Done** をクリックします。残りの手順を行う必要はありません。
その他の場合は、以下の1つをスキーマタイプとして選択します。
 - **JSON schema** は JSON データの構造を記述するドキュメントです。ドキュメントのメディアタイプは **application/schema+json** です。
 - **JSON instance** は JSON データが含まれるドキュメントです。ドキュメントのメディアタイプは **application/json** です。
 - **XML schema** は XML データの構造を記述するドキュメントです。このドキュメントのファイル拡張子は **.xsd** です。
 - **XML instance** は XML データが含まれるドキュメントです。このドキュメントのファイル拡張子は **.xml** です。
8. **Definition** 入力ボックスに、選択したスキーマタイプに準拠する定義を貼り付けます。たとえば、**JSON schema** を選択した場合は、メディアタイプが **application/schema+json** の JSON スキーマファイルの内容を貼り付けます。

9. **Data Type Name** フィールドにはデータタイプに選択した名前を入力します。たとえば、ベンダーの JSON スキーマを指定する場合は、データ型名として **Vendor** を指定できます。
このタイプを指定するコネクションを使用するインテグレーションを作成または編集するときにこのデータタイプ名が表示されます。Fuse Online では、インテグレーション仮想化パネルとデータマッパーにタイプ名が表示されます。
10. **Data Type Description** フィールドには、このタイプを区別するのに便利な情報を提供します。データマッパーでこのタイプを処理するステップにカーソルを合わせると、この説明が表示されます。
11. **Done** をクリックします。

結果

インテグレーションフローの最初にコネクションが表示されます。

13.3. MQTT ブローカーへのメッセージのパブリッシュ

インテグレーションでは、フローの途中で MQTT ブローカーにメッセージをパブリッシュしたり、シンプルなインテグレーションを終了するために MQTT ブローカーにメッセージをパブリッシュすることができます。これには、MQTT コネクションをフローの途中に追加するか、シンプルなインテグレーションの最後のコネクションとして追加します。

前提条件

- MQTT コネクションが作成されている必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。または、Fuse Online で最後のコネクションを選択するよう要求されます。

手順

1. メッセージのパブリッシュに使用する MQTT コネクションをクリックします。
2. **Choose an Action** ページで **Publish** をクリックします。
3. **MQTT queue/topic name** フィールドに、メッセージをパブリッシュするキューまたはトピックの名前を指定します。
4. **Next** をクリックしてアクションの入力タイプを指定します。
5. データタイプの認識が必要がない場合は、**Select Type** フィールドで **Type specification not required** を受け入れ、下部にある **Done** をクリックします。残りの手順を行う必要はありません。
その他の場合は、以下の1つをスキーマタイプとして選択します。
 - **JSON schema** は JSON データの構造を記述するドキュメントです。ドキュメントのメディアタイプは **application/schema+json** です。
 - **JSON instance** は JSON データが含まれるドキュメントです。ドキュメントのメディアタイプは **application/json** です。
 - **XML schema** は XML データの構造を記述するドキュメントです。このドキュメントのファイル拡張子は **.xsd** です。

- **XML instance** は XML データが含まれるドキュメントです。このドキュメントのファイル拡張子は **.xml** です。
6. **Definition** 入力ボックスに、選択したスキーマタイプに準拠する定義を貼り付けます。たとえば、**JSON schema** を選択した場合は、メディアタイプが **application/schema+json** の JSON スキーマファイルの内容を貼り付けます。
 7. **Data Type Name** フィールドにはデータタイプに選択した名前を入力します。たとえば、ベンダーの JSON スキーマを指定する場合は、データ型名として **Vendor** を指定できます。このタイプを指定するコネクションを使用するインテグレーションを作成または編集するときにこのデータタイプ名が表示されます。Fuse Online では、インテグレーション仮想化パネルとデータマッパーにタイプ名が表示されます。
 8. **Data Type Description** フィールドには、このタイプを区別するのに便利な情報を提供します。データマッパーでこのタイプを処理するステップにカーソルを合わせると、この説明が表示されます。
 9. **Done** をクリックします。

結果

インテグレーションフローで追加した場所にコネクションが表示されます。

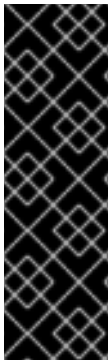
第14章 ODATA への接続

OData (Open Data Protocol) は、REST API を構築および使用するための標準です。OData サービスは、OData 対応の HTTP リクエストを介して Fuse Online などのサービスをクライアントの提供します。Fuse Online インテグレーションは、OData サービスからエンティティを取得でき、OData サービスによって管理されるエンティティを更新、作成、または削除できます。これには、OData コネクションを作成し、そのコネクションをインテグレーションに追加します。

OData の仕様を理解していることを前提とします。

詳細は以下を参照してください。

- 「OData サービスへのコネクションの作成」
- 「ポーリングが OData からデータを返すときにインテグレーションをトリガー」
- 「OData サービスによって管理されるデータの更新、作成、および削除」



重要

OData コネクタはテクノロジープレビューの機能です。テクノロジープレビューの機能は、Red Hat の本番環境のサービスレベルアグリーメント (SLA) ではサポートされず、機能的に完全ではないことがあるため、Red Hat は本番環境での使用は推奨しません。Red Hat では、これらについて実稼働環境での使用を推奨していません。テクノロジープレビューの機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストやフィードバックの提供を可能にするために提供されます。Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、<https://access.redhat.com/ja/support/offerings/techpreview> を参照してください。

14.1. ODATA サービスへのコネクションの作成

インテグレーションで OData サービスからエンティティを取得したり、OData サービスによって管理されるエンティティを作成または削除するには、最初に OData サービスへのコネクションを作成する必要があります。

前提条件

- 接続する OData サービスのベース URL を知っている必要があります。
- サービスが認証を使用する場合はクレデンシャルが必要で、必要に応じて SSL 証明書が必要です。しかし、ほとんどの OData サービスはパブリックサービスであり、認証は必要ありません。

手順

1. Fuse Online の左パネルで **Connections** をクリックし、利用可能なコネクションを表示します。
2. 右上の **Create Connection** をクリックしてコネクタを表示します。
3. **OData** コネクタをクリックします。
4. コネクションを設定します。

- a. **Service Root URL** フィールドに、アクセスする OData サービスのベース URL を入力します。
- b. サービスが Basic 認証を使用する場合は、**User Name** フィールドにユーザー名を入力します。
- c. サービスが Basic 認証を使用する場合は、**Password** フィールドにパスワードを入力します。
- d. サービスが SSL 証明書を要求する場合は、**Server Certificate** フィールドに SSL 証明書の内容を貼り付けます。

通常、パブリックの OData サービスには、ブラウザによって提供される以外の証明書は必要ありません。しかし、内部 OData サービスでは、独自の SSL 署名書を作成し、自己署名した可能性があります。この証明書は認証局によって信用されないため、OData サービスへの接続に失敗します。接続を有効にするには、ここに自己署名証明書を貼り付けます。

ブラウザでサービスをロードするのが証明書を取得する 1 つの方法です。その後のステップは、使用しているブラウザによって異なります。一般的に、アドレスバーの横にある **Not secure** 鍵マークをクリックして、**View Certificate** をクリックし、表示された証明書をファイルにエクスポートしてその証明書をコピーし、このフィールドに貼り付けるステップと似たものになります。

5. **Validate** をクリックします。Fuse Online は即座に接続を検証しようとし、検証の成功または失敗を示すメッセージを表示します。検証に失敗した場合は、入力パラメーターを訂正し、再度検証を行います。
6. 検証に成功した場合は **Next** をクリックします。
7. **Connection Name** フィールドに、この接続を別の接続と区別するために使用する名前を入力します。たとえば、**OData North** を入力します。
8. **Description** は任意のフィールドで、この接続に関する便利な情報を入力します。
9. 右上にある **Create** をクリックし、作成した接続が利用可能になったことを確認します。上記の例の名前を入力した場合は、インテグレーションに追加できる接続として **OData North** が表示されます。

次のステップ

OData 接続をインテグレーションに追加します。

14.2. ポーリングが ODATA からデータを返すときにインテグレーションをトリガー

OData サービスからデータを取得したときにインテグレーションの実行をトリガーするには、OData 接続を最初の接続としてシンプルなインテグレーションに追加します。インテグレーションの実行時、OData 接続は指定された間隔でサービスをポーリングします。接続設定を満たすデータが見つかったら、接続はそのデータをインテグレーションの次のステップに渡します。

前提条件

OData 接続が作成されている必要があります。

手順

1. Fuse Online パネルの左側にある **Integrations** をクリックします。
2. **Create Integration** をクリックします。
3. **Choose a Start Connection** ページで、インテグレーションを開始するために使用する OData コネクションをクリックします。
4. **Choose an Action** ページで **Read** アクションをクリックします。
5. **Read** アクションを設定します。
 - a. **Resource Collection** フィールドで、クエリーを実行する OData リソースの名前を選択します。Fuse Online は OData サービスからデータを取得し、利用可能なリソースコレクションのリストを提供します。
 - b. 特定の1つのエンティティを取得するには、**Entity Key Predicate** フィールドにキー述語を指定してそのエンティティを特定します。たとえば、**UserName='Bob'** や **Categories(1)** を指定します。複数のエンティティを取得する場合はこのフィールドを空白のままにしておきます。
 - c. **Query Options** フィールドに、リソースに適用するクエリーを入力します。OData 構文を使用します。たとえば、**\$filter=startswith(name, 'N')** は、**N** で始まる **name** フィールドを持つリソースで各エンティティのメッセージを返します。
Entity Key Predicate と **Query Options** の両方を指定できます。両方を指定すると、OData サービスは指定されたエンティティを取得し、クエリーをそのエンティティに適用します。たとえば、**Entity Key Predicate** を **UserName='russellwhyte'/Emails** に設定し、**Query Option** を **\$filter=contains(\$it,'example')** に設定した場合、コネクションは以下のようなリクエストを作成します。

[https://services.odata.org/TripPinRESTTierService/People\(UserName='russellwhyte'\)/Emails?\\$filter=contains\(\\$it, 'example'\)](https://services.odata.org/TripPinRESTTierService/People(UserName='russellwhyte')/Emails?$filter=contains($it, 'example'))

 サービスは、ドメイン **example** が含まれる Russell Whyte のメールアドレスをすべて返します。
 - d. **Filter Old Results** チェックボックスを選択し、特定のメッセージを一度のみ取得します。
 - e. メッセージのコレクションでなく、個別のメッセージをコネクションが返すようにするには、**Split Results** チェックボックスを選択します。
 Fuse Online は、フローに追加できる個別の分割ステップ (split step) および集約ステップ (aggregate step) も提供します。1つ以上のステップで個別のメッセージを処理し、メッセージをコレクションに集約する場合は、**Split Results** チェックボックスを選択しないでください。代わりに、このチェックボックスを空のままにし、このコネクションの後にフローに分割ステップ (split step) を追加します。フローに集約ステップ (aggregate step) を使用する場合は分割ステップ (split step) が必要になります。
 - f. **Interval Before Polling Starts** フィールドには、デフォルトの1秒を使用するか、コネクションが OData サービスのポーリングを開始するまでの経過期間を入力します。
 - g. **Delay** フィールドには、デフォルトの30秒を使用するか、コネクションが OData サービスをポーリングする間隔を入力します。
 - h. **Backoff Idle Threshold** フィールドには、デフォルトの1を使用するか、戻りデータがなくても許容されるポーリングの連続数を示します。このポーリング数を超えると、コネクションは後続のポーリングの間隔を長くします。コネクションは、**Delay** の値に **Backoff Multiplier** の値を掛けて、ポーリングの新しい間隔を決定します。

たとえば、ポーリングの間隔 (Delay の値) がデフォルトの 30 秒である場合、**Backoff Idle Threshold** は 5 に設定され、**Backoff Multiplier** は 12 に設定されます。ポーリングが 5 回連続でデータを返さなかった場合、コネクションは 360 秒 (30 x 12) 待機してからポーリングを再開します。コネクションは、ポーリングがデータを返すまで、360 秒間隔でポーリングを続行します。ポーリングがデータを返したら、コネクションはポーリング間隔を 30 秒に戻します。

- i. **Backoff Multiplier** フィールドには、デフォルトの 1 を使用するか、**Backoff Idle Threshold** に設定された値に達したときにポーリングの間隔を長くする乗数を整数で指定します。
Backoff Multiplier にデフォルトの 1 を使用すると、連続して結果を返さなかったポーリングの数に関わらず、コネクションは指定の間隔でポーリングを続行します。

コネクションはデータが返されないと自動的にポーリングの頻度を減らすことができるため、**Backoff Idle Threshold** および **Backoff Multiplier** に指定する値は、CPU オーバーヘッドの削減に便利です。

6. **Done** をクリックします。

結果

コネクションは、インテグレーションの最初のコネクションとしてインテグレーションフローに表示されます。

実行中にコネクションが返すものは **Entity Key Predicate** および **Query Options** フィールドに指定した内容によって異なります。OData コネクションは以下を返すことができます。

- エンティティのコレクションまたはエンティティプロパティのコレクション
たとえば、リソースのすべての **Person** エンティティであったり、すべての **Person** エンティティのすべての **Age** プロパティであったりします。コネクションはコレクションを 1 つのメッセージで返します。Fuse Online はコレクションに対して、フローで各後続ステップを 1 度実行します。しかし、**Read** アクションを設定するときに **Split Results** を選択すると、コネクションは独自のメッセージで各エンティティまたは各プロパティを返します。Fuse Online はフローで各後続ステップを一度各メッセージに対して実行します。
- エンティティまたはエンティティプロパティ
たとえば、**UserName** プロパティが **Bob** である **Person** エンティティであったり、**UserName** が **Bob** である **Person** エンティティの **Age** プロパティであったりします。コネクションは、フローの次のステップに渡すメッセージのエンティティまたはエンティティプロパティを返します。

次のステップ

インテグレーションの最後のコネクションと、インテグレーションに含まれるようにする他のコネクションを追加します。インテグレーションに必要なコネクションがすべて含まれ、OData コネクションがコレクションを返す場合、OData コネクションの後に分割ステップ (split step) を追加する必要があるか検討します。通常、インテグレーションは OData コネクションによって返されるデータをフローの後続のコネクションが使用できるフィールドにマップします。コレクションをマップすることもあります。より多くの場合でターゲットフィールドにマップするためにコレクションを分割する必要があります。

OData コネクションの後、フローにデータマッパーステップを追加します。フローでの正確な場所は、フローが実行することによって異なります。たとえば、OData コネクションの後にベーシックフィルタステップを追加してから、データマッパーステップを追加することがあります。

14.3. ODATA サービスによって管理されるデータの更新、作成、および削除

インテグレーションでは、フローの途中で OData サービスによって管理されるリソースを更新したり、シンプルなインテグレーションを終了するために OData サービスによって管理されるリソースを更新することができます。これには、OData コネクションをフローの途中に追加するか、シンプルなインテグレーションの最後のコネクションとして追加します。

前提条件

- OData コネクションが作成されている必要があります。
- インテグレーションを作成または編集することになります。または、Fuse Online で最後のコネクションを選択するよう要求されます。

手順

1. 使用する OData コネクションをクリックします。OData コネクションがエンティティを更新または削除するときに、コネクションは何も返さないことに注意してください。OData コネクションがエンティティを作成するとき、コネクションは新しいエンティティを返します。
2. **Choose an Action** ページで、コネクションが実行するアクションをクリックします。
 - **Create** はエンティティを OData リソースに追加します。
 - **Delete** はエンティティを OData リソースから削除します。
 - **Update** は OData リソースにあるエンティティを変更します。
3. アクションを設定するには、更新する OData リソースを選択します。
更新する OData リソースの更新方法を理解することが重要になります。OData サービスのバックエンドデータソースは、更新や競合解決のルールを決定します。たとえば、OData コネクションが新規エンティティを作成しようとしたにも拘らず、その Predicate Key が指定されたエンティティがすでに存在した場合に OData サービスは既存のエンティティを上書きするか、既存のエンティティのフィールドの一部を上書きするか、この操作を無視します。この状況で OData サービスがどのように動作するかを知るのはユーザー次第です。
4. **Done** をクリックします。

結果

インテグレーションフローで追加した場所にコネクションが表示されます。

次のステップ

OData コネクションの前にデータマッピングステップを追加します。必要なデータを提供するソースフィールドをマップして新しいエンティティを作成するか、エンティティを削除する必要があります。コレクションをターゲットフィールドにマップする場合、ソースコレクションの最新要素のコンテンツがターゲットフィールドにマップされるのがデフォルトの動作となります。

第15章 REST API への接続

インテグレーションで REST API に接続するには、API を説明する OpenAPI ドキュメントをアップロードして、API のコネクタを作成しておく必要があります。『[Integrating Applications with Fuse Online](#)』の「[Adding and managing REST API client connectors](#)」を参照してください。

接続する REST API のコネクタが Fuse Online で利用可能である場合、REST API に接続するステップは次のとおりです。

- 「[Fuse Online を REST API クライアントとして登録](#)」（必要な場合）
- 「[REST API クライアントコネクションの作成](#)」
- 「[API クライアントのインテグレーションへの追加](#)」

15.1. FUSE ONLINE を REST API クライアントとして登録


Fuse Online が REST API クライアントコネクタを作成する前に、API のセキュリティ要件を示すよう要求されます。OAuth プロトコルを使用する API では、Fuse Online がコネクタを作成すると、API のエントリーも Fuse Online の **Settings** ページに追加します。ここに Fuse Online の API へのアクセスを承認する API クライアント ID と API クライアントシークレットを提供します。

接続する API が OAuth を使用しない場合、このセクションを飛ばして「[REST API クライアントコネクションの作成](#)」を参照してください。

前提条件

接続する REST API の OAuth カスタムアプリケーション設定ページの URL を知っている必要があります。

手順

1. Fuse Online で以下を行います。
 - a. **Settings** ページで、上部付近にあるコールバック URL の右側で  をクリックし、Fuse Online 環境のコールバック URL をクリップボードにコピーします。この手順の後半でこの URL が必要になります。
 - b. 接続する API の名前を見つけ、その **Register** ボタンをクリックしてクライアント ID およびクライアントシークレットフィールドを表示します。
2. 別のブラウザウィンドウで、Fuse Online を接続する API の OAuth クライアントとして登録する必要があります。これを行う正確な手順は、API サービスごとに異なります。通常、API サービスは OAuth カスタムアプリケーション設定ページを提供します。そのページに移動し、以下を行います。
 - a. この手順の最初にコピーした Fuse Online のコールバック URL を提供します。
 - b. 入力が必要なその他のプロンプトに対応します。
 - c. API サービスが Fuse Online 環境に割り当てるクライアント ID およびクライアントシークレットを取得します。
3. 登録している API サービスの Fuse Online **Settings** ページエントリーに戻り、以下を行います。

- a. 割り当てられたクライアント ID を貼り付けます。
- b. 割り当てられたクライアントシークレットを貼り付けます。
- c. **Save** をクリックします。

15.2. REST API クライアントコネクションの作成

インテグレーションで REST API に接続するには、任意の数のインテグレーションを追加できる REST API へのコネクションを作成します。

前提条件

- 接続する REST API のコネクターが作成されている必要があります。
- REST API が OAuth プロトコルを使用する場合、その REST API にアクセスできるクライアントアプリケーションとして Fuse Online 環境が登録されています。

手順

1. Fuse Online の左パネルで **Connections** をクリックし、利用可能なコネクションを表示します。
2. 右上にある **Create Connection** をクリックし、利用できるコネクターを表示します。
3. コネクションを作成する API のコネクターをクリックします。
4. プロンプトに応答し、追加情報を提供します。API の定義は、Fuse Online が要求する内容を決めます。たとえば、HTTP Basic 認証を使用する API の場合、Fuse Online は API にアクセスするために使用するユーザー名およびパスワードを要求します。OAuth を使用する API の場合、Fuse Online が API に接続するための登録クレデンシャルを検証できるようにするため、ユーザーがクリックするボタンを表示します。
5. **Connection Name** フィールドに、このコネクションを別のコネクションと区別するために使用する名前を入力します。
6. **Description** は任意のフィールドで、このコネクションに関する便利な情報を入力します。
7. 右上の **Create** をクリックし、作成したコネクションを選択でき、インテグレーションに追加できることを確認します。

15.3. API クライアントのインテグレーションへの追加

インテグレーションで REST API に接続するには、その REST API へのコネクションをフローに追加します。

本リリースでは、REST API へのコネクションはフローの途中とするか、シンプルなインテグレーションの最後のコネクションとすることができます。よって、REST API へのコネクションを最初のコネクションとすることはできません。

前提条件

- REST API へのコネクションが作成されている必要があります。
- フローを作成または編集することになります。

- フローに最初のコネクションがある必要があります。
- Fuse Online は最後のコネクションの選択またはステップの選択を要求します。

手順

1. 使用できるステップを表示するページで、フローに追加する REST API コネクションをクリックします。
2. コネクションが実行するアクションをクリックします。使用できるアクションは、Fuse Online にアップロードされ、接続先の API を説明する OpenAPI ドキュメントに指定されたリソース操作を基にします。
3. 選択したアクションに応じて、Fuse Online が要求するパラメーターを入力します。
4. **Done** をクリックします。

結果

フローで追加した場所にコネクションが表示されます。

第16章 SALESFORCE への接続

インテグレーションで Salesforce に接続するには、Fuse Online 環境を Salesforce クライアントアプリケーションとして登録する必要があります。これにより、Fuse Online による Salesforce へのアクセスが許可されます。この登録が完了していると、Salesforce コネクションを作成でき、そのコネクションに任意の数のインテグレーションを追加することができます。詳細は以下を参照してください。

- [「Fuse Online を Salesforce クライアントアプリケーションとして登録」](#)
- [「Salesforce コネクションの作成」](#)
- [「Salesforce コネクションのインテグレーションへの追加」](#)

16.1. FUSE ONLINE を SALESFORCE クライアントアプリケーションとして登録

インテグレーションで Salesforce に接続するには、最初に Fuse Online 環境を Salesforce にアクセスできるクライアントアプリケーションとして登録する必要があります。これにより、Salesforce に接続するインテグレーションをいくつでも作成することができます。そのため、特定の Fuse Online 環境を1度だけ Salesforce に登録する必要があります。

Fuse Online を Salesforce クライアントアプリケーションとして登録できるのは、Fuse Online 環境ごとに1回のみです。しかし、各 Salesforce コネクションは同じ登録を使用しますが、異なるユーザークレデンシャルを使用することができます。

前提条件

Fuse Online にログインしている必要があります。Fuse Online にアクセスできる URL は、Red Hat Fuse Online Evaluation プログラムのウェルカムメールメッセージにあります。

手順

1. Fuse Online で以下を行います。
 - a. 左側のパネルで **Settings** をクリックします。
 - b. **Settings** ページで、上部付近にあるコールバック URL の右側で  をクリックし、Fuse Online 環境のコールバック URL をクリップボードにコピーします。この手順の最後の方でこの URL が必要になります。
 - c. **Salesforce** エントリーの右側にある **Register** をクリックし、**Client ID** および **Client Secret** フィールドを表示します。
2. 別のブラウザータブで、Salesforce アカウントにログインし、以下の手順にしたがって接続アプリケーションを作成します。これらの手順では、Salesforce Classic ユーザーインターフェースを使用していることを仮定します。Salesforce Lightning Experience インターフェースから切り替えるには、プロファイルアイコンをクリックして **Switch to Salesforce Classic** を選択します。追加情報は、Salesforce ドキュメントの「[Create a Connected App](#)」を参照してください。
 - a. Salesforce で右上にある **Setup** をクリックします。
 - b. 左側のパネルで **Build > Create > Apps** と選択します。
 - c. **Connected Apps** まで下にスクロールし、**New** をクリックします。

- d. 必要な情報を入力し、**Enable OAuth Settings** を選択します。
- e. **Callback URL** フィールドに、この手順の最初にコピーした Fuse Online URL を貼り付けます。例: **https://app-proj912876.7b63.fuse-ignite.openshiftapps.com/api/v1/credentials/callback**
- f. **OAuth Scopes** では以下を追加します。
 - **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. **Configure ID token** を選択し **Include Standard Claims** を選択します。
- h. 下にスクロールし、**Save** をクリックします。
- i. 上にスクロールし、Salesforce が短い待機時間を示していることを確認します。

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. **Continue** をクリックします。
 - k. Salesforce が提供するコンシューマー鍵をコピーします。
3. Fuse Online の **Settings** ページに戻り、Salesforce が提供したコンシューマー鍵を Fuse Online の Salesforce **Client ID** フィールドに貼り付けます。
 4. Salesforce に戻り、Salesforce が提供するコンシューマーの秘密をコピーします。
 5. Fuse Online の **Settings** ページに戻り、Salesforce が提供したコンシューマーの秘密を Fuse Online の Salesforce **Client Secret** フィールドに貼り付けます。
 6. **Save** をクリックし、**Ok** をクリックします。

16.2. SALESFORCE コネクションの作成

Salesforce でデータにアクセスするインテグレーションを作成するには、最初に Salesforce コネクションを作成する必要があります。Salesforce コネクションの作成後、複数のインテグレーションで使うことができます。

前提条件

- Fuse Online にログインしている必要があります。Fuse Online にアクセスできる URL は、Red Hat Fuse Online Evaluation プログラムのウェルカムメールメッセージにあります。
- Fuse Online がウェブブラウザで開かれている必要があります。
- Fuse Online 環境が Salesforce にアクセスできるアプリケーションとして登録されている必要があります。
- Fuse Online の **Settings** ページで登録した後に受け取った Salesforce クライアント ID およびクライアントシークレットが追加済みである必要があります。

Fuse Online インストールを Salesforce クライアントとして登録した後、必ず 2 - 10 分ほど待ってから Salesforce コネクションを作成してください。

手順

1. Fuse Online で、左側のパネルにある **Connections** をクリックして、使用できるコネクションを表示します。
2. 右上の **Create Connection** をクリックし、利用可能なコネクタを表示します。コネクタは、1つ以上のコネクションを作成するためのテンプレートです。
3. **Salesforce** コネクタをクリックします。
4. **Connect Salesforce** をクリックして、Salesforce の承認ページを表示します。承認ページを表示するには Salesforce にログインする必要があります。
Connect Salesforce が表示されない場合、Fuse Online 環境は Salesforce クライアントアプリケーションとして登録されていません。「[Fuse Online を Salesforce クライアントアプリケーションとして登録](#)」を参照してください。Salesforce コネクションの作成時に Fuse Online 環境が Salesforce クライアントアプリケーションとして登録されていない場合、Fuse Online は承認を要求する複数のフィールドを表示します。これらのフィールドに値を入力して Salesforce コネクションを作成することはできませんが、推奨されません。



注記

以下のエラーは、Salesforce に正しい Fuse Online コールバック URL が指定されていないことを表しています。

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

このエラーメッセージが表示されたら、Salesforce で「[Fuse Online を Salesforce クライアントアプリケーションとして登録](#)」の手順にしたがって Fuse Online コールバック URL が指定されていることを確認してください。

5. **Allow** をクリックして、Fuse Online に戻ります。
6. **Connection Name** フィールドに、このコネクションを別のコネクションと区別するために使用する名前を入力します。たとえば、**SF Connect 1** を入力します。
7. **Description** は任意のフィールドで、このコネクションに関する便利な情報を入力します。たとえば、**Sample Salesforce connection that uses my Salesforce login credentials.** を入力します。
8. 右上にある **Create** をクリックし、作成したコネクションが利用可能になったことを確認します。上記の例の名前を入力した場合は、インテグレーションに追加できるコネクションとして **SF Connect 1** が表示されます。

16.3. SALESFORCE コネクションのインテグレーションへの追加

インテグレーションでは、フローの途中で Salesforce に接続したり、シンプルなインテグレーションを開始または終了できます。これには、Salesforce コネクションをインテグレーションに追加します。

前提条件

- Salesforce コネクションが作成されている必要があります。

- インテグレーションを作成または更新することになります。
- Fuse Online は最初のコネクション、最後のコネクション、またはステップの選択を要求します。

手順

1. フローに追加する Salesforce コネクションをクリックします。インテグレーションが Salesforce に接続するために選択したコネクションを使用する場合、Fuse Online はそのコネクションに定義されたクレデンシャルを使用します。
2. 選択したコネクションが実行するアクションをクリックします。フローに追加した各 Salesforce コネクションは選択したアクションのみを実行します。
3. アクションが操作する Salesforce オブジェクトを指定します。たとえば、コンタクト、リード、または価格表のエントリーなどがこれに該当します。**Object** フィールドをクリックして Salesforce オブジェクトのリストから選択するか、オブジェクトの名前を入力します。
4. **Done** をクリックして、コネクションをフローに追加します。

結果

インテグレーションフローで追加した場所にコネクションが表示されます。

第17章 SAP CONCUR への接続

SAP Concur は、経費、旅費、および請求書を管理します。インテグレーションで SAP Concur に接続し、SAP Concur のさまざまなアクションの1つを実行できます。たとえば、SQL データベースにプロジェクトコードを保存して、経費をプロジェクトに関連付けることができるように、SAP Concur に同じプロジェクトコードが必要な場合に、SQL データベースを接続するインテグレーションを作成して、新規プロジェクトコードを取得するストアードプロシージャを実行できます。その後、インテグレーションは SAP Concur に接続し、新しいプロジェクトコードを該当のリストにアップロードできます。

以下を参照してください。

- [「SAP Concur へ接続するためのユーザーロール」](#)
- [「SAP Concur 実装サイト用の OAuth クレデンシャルの取得方法」](#)
- [「SAP Concur 本番サイト用の OAuth クレデンシャルの取得方法」](#)
- [「SAP Concur コネクターの設定」](#)
- [「SAP Concur コネクションの作成」](#)
- [「SAP Concur コネクションのインテグレーションへの追加」](#)
- [「インテグレーションデータフィールドを SAP Concur フィールドにマップする例」](#)

17.1. SAP CONCUR へ接続するためのユーザーロール

SAP Concur には、開発作業の実装サイトと実稼働で使用するための本番サイトがあります。実装サイトは本番サイトをシミュレートします。これにより、SAP Concur のサイトで実稼働で使用する前にインテグレーションを作成および改良することができます。

Fuse Online はいつでも SAP Concur 実装サイトまたは本番サイトに接続できます。よって、特定の Fuse Online 環境で両方の SAP Concur サイトへのコネクションを同時に行うことはサポートされません。各 Fuse Online 環境には、各 SAP Concur サイトにアクセスするための独自のクレデンシャルが必要で、実装サイトのクレデンシャルと本番サイトのクレデンシャルは異なります。

そのため、システム管理者と同等の権限を持つユーザーが以下を行うことが想定されています。

1. SAP Concur から OAuth クレデンシャルを 2 セット取得します。
 - 1 セットは SAP Concur 実装サイトにアクセスするためのクレデンシャル
 - もう 1 セットは SAP Concur 本番サイトにアクセスするためのクレデンシャル
2. 適切な SAP Concur サイトにアクセスするために、SAP Concur コネクターを設定します。
3. SAP Concur コネクションを作成します。

その後、ビジネスユーザーは SAP Concur コネクションを使用するインテグレーションを作成できます。

17.2. SAP CONCUR 実装サイト用の OAUTH クレデンシャルの取得方法

SAP Concur 実装サイトへアクセスするためのクレデンシャルを取得するには、SAP Concur に直接連絡し、以下をリクエストします。

- Fuse Online を SAP Concur 実装サイトへのアクセスが許可される新しいアプリケーションとして登録します。
- 実装サイトに新しいセットの OAuth クレデンシャルが必要です。
- クレデンシャルには **LIST** スコープが含まれている必要があります。
- 任意で、list アクション以外のアクションを実行する権限を取得するには、他の **スコープ** をリクエストします。
- これらのクレデンシャルの OAuth 付与タイプは **Authorization Grant Flow** である必要があります。
Postman などの REST クライアントを使用してリクエストを SAP Concur 実装サイトに送信するには、OAuth 付与タイプが **Password** のクレデンシャルが必要です。
- クレデンシャルが必要な環境の Fuse Online コールバック URL は次のようになります。
`https://syndesis.my-minishift.syndesis.io/api/v1/credentials/callback`

この URL の **syndesis.my-minishift.syndesis.io** は、Fuse Online 環境への OpenShift ルートと一致します。

SAP Concur からの応答は以下を提供する必要があります。

- SAP Concur 実装サイトへログインするためのユーザー名およびパスワード。
- Fuse Online の SAP Concur コネクターを設定するために Fuse Online 開発環境の **Settings** に指定するクライアント ID およびクライアントシークレットの値。

17.3. SAP CONCUR 本番サイト用の OAUTH クレデンシャルの取得方法

Fuse Online の SAP Concur 本番サイトへのアクセスを許可する OAuth クレデンシャルを取得するには、SAP Concur Developer Center の「[Getting Started](#)」ページにアクセスし、記載されている手順にしたがいます。

17.4. SAP CONCUR コネクターの設定

インテグレーションで SAP Concur に接続するには、Fuse Online の SAP Concur コネクターを設定する必要があります。設定後、コネクターを使用して SAP Concur へのコネクションを作成できます。

前提条件

- SAP Concur コネクターを設定して SAP Concur 実装サイトまたは本番サイトにアクセスするかどうかを認識している必要があります。
- Fuse Online の適切な SAP Concur サイトへのアクセスを許可する OAuth クレデンシャルを取得している必要があります。

手順

1. Fuse Online の左側のパネルで **Settings** をクリックします。
2. **Settings** ページの **SAP Concur** エントリーを展開します。
3. **Client ID** フィールドに、SAP Concur から受け取ったクライアント ID を入力します。

4. **Client Secret** フィールドに、SAP Concur から受け取ったクライアントシークレットを入力します。
他のフィールドは Fuse Online によって入力されます。
5. **Save** をクリックします。
Fuse Online は即座に設定の検証を試みます。検証に成功しなかった場合は、入力値を修正して再度検証を行います。検証に成功した場合は、インテグレーションに追加できる SAP Concur コネクションを作成することができます。

次のステップ

SAP Concur コネクションを作成するには、成功メッセージにある **create a connection** リンクをクリックします。作成を後で作成する場合は **Ok** をクリックし、**Settings** ページの SAP Concur エントリーを縮小します。

17.5. SAP CONCUR コネクションの作成

インテグレーションで SAP Concur に接続するには、任意の数のインテグレーションを追加できる SAP Concur コネクションを最初に作成する必要があります。SAP Concur コネクションを作成するとき、コネクションの作成時に入力するログインクレデンシャルを使用してコネクションが SAP Concur にアクセスすることを許可します。

前提条件

- コネクションを作成する Fuse Online 環境で、SAP Concur コネクターを設定する必要があります。
- SAP Concur 実装サイトまたは SAP Concur 本番サイトへアクセスするためにコネクターが設定されたかどうかを認識する必要があります。
- SAP Concur コネクターが設定された SAP Concur サイトへログインするためのユーザー名およびパスワードが必要です。

手順

1. Fuse Online の左パネルで **Connections** をクリックし、利用可能なコネクションを表示します。
2. 右上の **Create Connection** をクリックし、利用可能なコネクターを表示します。コネクターは、1つ以上のコネクションを作成するために使用するテンプレートです。
3. **SAP Concur** コネクターをクリックします。
4. **Configure Connection** ページで **Connect SAP Concur** をクリックし、コネクターが設定された SAP Concur サイトに移動します。
Connect SAP Concur が表示されない場合は、Fuse Online 環境で SAP Concur コネクターが設定されていません。「[SAP Concur コネクターの設定](#)」を参照してください。

redirect_uri is improper or not previously registered が表示された場合、SAP Concur コネクター設定は正しくありません。Fuse Online の **Settings** ページに戻りコネクター設定を更新および検証します。

5. SAP Concur の OAuth サイトで以下を行います。
 - a. SAP Concur へアクセスするためにこのコネクションが使用する SAP Concur アカウントのメールアドレスを入力します。

- b. **Username** が選択されていることを確認してください。
 - c. **Continue** をクリックします。
 - d. このメールアドレスに関連する SAP Concur パスワードを入力します。
 - e. **Sign In** をクリックし、Fuse Online に戻ります。
6. Fuse Online の **Connection Name** フィールドに、このコネクションを別のコネクションと区別するために使用する名前を入力します。たとえば、**SAP Concur Test1** を入力します。
 7. **Description** は任意のフィールドで、このコネクションに関する便利な情報を入力します。たとえば、**Sample SAP Concur connection to the implementation site.** を入力します。
 8. 右上にある **Create** をクリックし、作成したコネクションが利用可能になったことを確認します。上記の例の名前を入力した場合は、インテグレーションに追加できるコネクションとして **SAP Concur Test1** が表示されます。

17.6. SAP CONCUR コネクションのインテグレーションへの追加

SAP Concur コネクションをフローの途中に追加したり、シンプルなインテグレーションの最後のコネクションとして追加することができます。SAP Concur へのコネクションは 85 個のアクションのいずれかを実行できます。

前提条件

- SAP Concur コネクションが設定されている必要があります。
- フローを作成または編集することになります。
- Fuse Online は、ステップの選択またはシンプルなインテグレーションの最後のコネクションの選択を要求します。

手順

1. フローに追加する SAP Concur コネクションをクリックします。インテグレーションが SAP Concur に接続するために選択したコネクションを使用する場合、Fuse Online はそのコネクションに定義されたクレデンシャルを使用します。
2. 選択したコネクションが実行するアクションをクリックします。フローに追加した各 SAP Concur コネクションは選択したアクションのみを実行します。
SAP Concur アクションに関する詳細は、[SAP Concur developer center](#) にアクセスし、**v3.0** を展開してください。
3. **Done** をクリックして、コネクションをフローに追加します。

結果

インテグレーションフローで追加した場所にコネクションが表示されます。

17.7. インテグレーションデータフィールドを SAP CONCUR フィールドにマップする例

ほとんどのインテグレーションでは、SAP Concur へのコネクションの前にデータマッピングステップを追加する必要があります。データマッピングステップでは、以前のインテグレーションステップから

利用可能なデータフィールドを SAP Concur アクションパラメーターにマップします。データマッパーでは、インテグレーションに SAP Concur コネクションを追加したときに選択したアクションの SAP Concur パラメーターが **Target** パネルに表示されます。

新しいプロジェクトコードを取得する SQL ストアドプロシージャを実行してインテグレーションが起動するとします。インテグレーションは、新しいプロジェクトコードを SAP Concur に追加して終了します。以下の手順は、SAP Concur コネクションの前のデータマッピングステップの例を提供します。インテグレーション可視化フローは左にあり、Fuse Online はステップの選択を要求します。

1. **Data Mapper** をクリックします。データフィールドが表示されると、左側の **Sources** パネルにデータベースコネクションから利用できるフィールドが表示されます。この例では、ソースフィールドに **concur-list-id** および **project-code** が含まれます。右側の **Target** パネルには、コネクションが実行する update list アクションの SAP Concur **id** および **content** パラメーターが表示されます。
2. **concur-list-id** ソースフィールドを **id** ターゲットフィールドにマップします。
3. **project-code** ソースフィールドを **content** ターゲットフィールドにマップします。
4. 右上の **Done** をクリックし、データマッピングステップをインテグレーションに追加します。

第18章 SERVICENOW への接続

インテグレーションは ServiceNow テーブルからレコードを取得したり、ServiceNow がテーブルを更新するために使用する ServiceNow インポートセットにレコードを追加することができます。インテグレーションで ServiceNow に接続するには、ServiceNow コネクションを作成した後、そのコネクションをインテグレーションに追加します。詳細は以下を参照してください。

- 「ServiceNow コネクションの作成」
- 「ServiceNow からレコードを取得してインテグレーションの実行をトリガー」
- 「ServiceNow でのインポートセットの作成」
- 「インテグレーションの間またはインテグレーションを終了するためにレコードを ServiceNow にコピー」
- 「Salesforce から ServiceNow へのインテグレーションの例」

18.1. SERVICENOW コネクションの作成

インテグレーションで、所属企業の ServiceNow インスタンスに接続するには、ServiceNow コネクションを作成する必要があります。

前提条件

- 所属企業の ServiceNow 管理者がユーザーの ServiceNow アカウントを作成済みである必要があります。
- ServiceNow インスタンスの URL と、ServiceNow のユーザー名およびパスワードを知っている必要があります。
- ServiceNow 管理者が ServiceNow クライアント ID およびクライアントシークレットも作成した場合、これらの値を知っている必要があります。

手順

1. Fuse Online の左パネルで **Connections** をクリックし、利用可能なコネクションを表示します。
2. 右上にある **Create Connection** をクリックし、Fuse Online コネクターを表示します。
3. **ServiceNow** コネクターをクリックします。
4. この ServiceNow コネクションを設定します。
 - a. **Instance Name** フィールドに、レコードの取得先または取得元となる ServiceNow インスタンスの名前を入力します。たとえば、ServiceNow インスタンスの URL が **https://abc12345.service-now.com** である場合、インスタンス名は **abc12345** になります。
 - b. **User Name** フィールドに ServiceNow ユーザー名を入力します。
 - c. **User Password** フィールドに ServiceNow パスワードを入力します。
 - d. 任意で、ServiceNow 管理者からクライアント ID を受け取った場合は **Client ID** フィールドにそのクライアント ID を入力します。

- e. 任意で、ServiceNow 管理者からクライアントシークレットを受け取った場合は **Client Secret** フィールドにそのクライアントシークレットを入力します。
 - f. **Validate** をクリックします。Fuse Online は即座に接続を検証しようとし、検証の成功または失敗を示すメッセージを表示します。検証に失敗した場合は、必要に応じて設定の詳細を更新し、再度検証を行います。
 - g. 検証に成功したら **Next** をクリックします。
5. **Connection Name** フィールドに、この接続を別の接続と区別するために使用する名前を入力します。たとえば、**ServiceNow Con 1** を入力します。
 6. **Description** は任意のフィールドで、この接続に関する便利な情報を入力します。たとえば、**ServiceNow connection that uses administrative credentials.** を入力します。
 7. 右上にある **Create** をクリックし、作成した接続が利用可能になったことを確認します。上記の例の名前を入力した場合は、インテグレーションに追加できる接続として **ServiceNow Con 1** が表示されます。

18.2. SERVICENOW からレコードを取得してインテグレーションの実行をトリガー

ServiceNow からレコードを受信したときにインテグレーションの実行をトリガーするには、ServiceNow 接続を最初の接続としてシンプルなインテグレーションに追加します。

前提条件

- ServiceNow 接続が作成されている必要があります。
- レコードを取得するテーブルの名前を知っている必要があります。
- 必要なレコードのみを取得する ServiceNow クエリーがすでに定義されている必要があります。ServiceNow クエリーの詳細は、ServiceNow の「[Encoded query strings](#)」を参照してください。

手順

1. Fuse Online パネルの左側にある **Integrations** をクリックします。
2. **Create Integration** をクリックします。
3. **Choose a Start Connection** ページで、インテグレーションを開始するために使用する ServiceNow 接続をクリックします。
4. **Choose an Action** ページで **Retrieve Record** をクリックし、選択した ServiceNow テーブルからレコードを取得します。
5. **Retrieve Record** アクションを設定します。
 - a. **Table name** をクリックし、レコードを取得するテーブルを選択します。
 - b. **Query to filter the result set** フィールドに ServiceNow クエリーを入力します。たとえば、インシデントに関するレコードを保持するテーブルでの **state=1^impact=2** クエリーについて考えてみましょう。このクエリーは中程度の影響 (**state=1**) を持つ新しいインシデント (**impact=2**) のレコードを返します。クエリーを入力せず、テーブルのレコードが変更しないと、接続は同じレコードを毎回取得します。

- c. **Limit of elements per page** フィールドに、コネクションが取得する最大レコード数を入力します。本リリースでは、値を入力する必要があり、**1000** 以下の値を入力することが推奨されます。ページネーションはサポートされません。
- d. **Period** フィールドにレコードを取得する頻度を指定します。デフォルト毎分です。そのため、デフォルトでは Fuse Online はこのインテグレーションを 60 秒ごとに実行します。
- e. **Done** をクリックします。

結果

- Fuse Online は、取得したレコードの構造を定義する JSON スキーマを生成します。これにより、データマッピングが必要な場合にインテグレーションで次のコネクションの前にデータマッピングステップを追加することができます。
- Fuse Online は最後のコネクションの選択を要求します。

18.3. SERVICENOW でのインポートセットの作成

Fuse Online インテグレーションでは、ServiceNow コネクションは直接 ServiceNow テーブルを更新できません。インテグレーションで ServiceNow データを更新するには、ServiceNow テーブルの更新をステージングするインポートセットが事前に必要になります。

多くの場合で、ServiceNow 管理者が ServiceNow インバウンド Web サービスを作成することが、最も簡単にインポートセットを作成する方法になります。これにより、ServiceNow インポートセットが自動的に作成されます。インポートセットは、Web サービスの作成時に管理者が選択する ServiceNow ターゲットテーブルを基にしています。ServiceNow コネクションはこのインポートセットを更新し、ServiceNow はインポートセットを使用して対応する ServiceNow テーブルを更新します。

前提条件

ServiceNow の管理者権限が必要です。

ServiceNow の手順

1. ServiceNow で **web services** を検索します。
2. 結果の **System Web Services > Inbound** を選択し、**Create New** をクリックします。
3. **Create Web Service** ページで以下を行います。
 - a. **Label** フィールドで Web サービスの表示名を指定します。Web サービスのインポートセットにもこの名前が使用されます。
 - b. **Target table** フィールドに、このサービスが更新する ServiceNow テーブルを選択します。
 - c. **Copy fields from target table** を選択します。これにより、ターゲットテーブルと同じスキーマを持つインポートセットが作成されます。インポートセットはターゲットテーブルを更新するためのステージングエリアです。
 - d. **Create transform map** を選択します。このマップによって、ServiceNow がインポートセットからターゲットテーブルにデータをコピーまたは変換することができます。
4. **Create** をクリックします。

5. 次の表示の **Related Links** で **Auto Map Matching Fields** をクリックし、選択したターゲットテーブルにあるフィールドのリストを表示します。
6. correlation (相関) または coalesce (結合) フィールドとなるターゲットテーブルフィールドを1つ選択します。
インポートセットのレコードにある correlation フィールドの値が、ターゲットテーブルの correlation フィールドの値と一致する場合、ServiceNow は新しいレコードを作成せずに、ターゲットテーブルのレコードを更新します。
7. **Coalesce** を選択し、選択したフィールドを correlation フィールドとして特定します。
correlation フィールドがない場合、ServiceNow はインポートセットの各レコードをターゲットテーブルとして追加します。
8. **Update** をクリックします。
9. 変換マップ (transform map) が作成されたことを確認します。
 - a. ServiceNow で **transform map** を検索します。
 - b. **System Import Sets** で **Transform Maps** をクリックし、変換マップのリストを表示します。
 - c. このリストで、作成した Web サービスのラベルに指定した値が名前になっている変換マップがあることを確認します。

その他のリソース

ServiceNow ドキュメントの「[Import sets](#)」

18.4. インテグレーションの間またはインテグレーションを終了するためにレコードを SERVICENOW にコピー

フローの途中でレコードを ServiceNow にコピーしたり、シンプルなインテグレーションを終了するためにレコードを ServiceNow にコピーすることができます。これには、ServiceNow コネクションをフローの途中に追加するか、シンプルなインテグレーションの最後のコネクションとして追加します。

前提条件


- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。または、Fuse Online で最後のコネクションを選択するよう要求されます。
- レコードの追加先となる ServiceNow インポートセットの名前を知っている必要があります。ServiceNow 管理者は適切なインポートセットを特定することができます。インポートセットが存在しない場合は、「[ServiceNow でのインポートセットの作成](#)」を参照してください。
- 追加のレコードを処理するよう ServiceNow インポートセットが設定されている必要があります。

手順

1. インテグレーションを追加する ServiceNow コネクションをクリックします。
2. **Add Record** アクションをクリックします。
3. **Import Set** フィールドをクリックし、レコードに追加するインポートセットを選択します。

4. **Next** をクリックします。

結果

- Fuse Online はコネクションをフローに追加します。
- 通常、このコネクションの前にデータマッピングステップが必要になります。インテグレーションビジュアライゼーションパネルで、Fuse Online はコネクション上に  を表示します。ほとんどのインテグレーションでは、ソースフィールドを ServiceNow テーブルの correlation フィールドとして使用されるターゲットフィールドにマップする必要があります。

18.5. SALESFORCE から SERVICENOW へのインテグレーションの例

この例では、2つのシンプルなインテグレーションについて説明します。

- 1つのインテグレーションは、Salesforce から新しいケースを取得し、インシデントとして ServiceNow に追加します。
- もう1つのインテグレーションでは、Salesforce から更新されたケースを取得し、ServiceNow の対応するインシデントを更新します。

ServiceNow における前提条件

両方のインテグレーションの前提条件として、ServiceNow の管理者がインバウンド Web サービスを作成する必要があります。これは、インテグレーションがデータを ServiceNow に追加するために必要な ServiceNow インポートセットを暗黙的に作成します。この例では、以下の条件が適用されます。

- Web サービスの名前は **Cases from Salesforce** です。
- サービスのターゲットテーブルは **Incidents** です。このテーブルには更新するデータが含まれます。
- **Copy fields from target table**が選択されているため、インポートセットには **Incidents** テーブルと同じスキーマがあります。インポートセットの名前も **Cases from Salesforce** になります。
- **Create transform map**が選択されているため、ServiceNow は **Cases from Salesforce** インポートセットを **Incidents** ターゲットテーブルにコピーおよび変換します。
- **correlation_Id** フィールドが選択され、その後に **Coalesce** が選択されているため、インポートセットでは correlation フィールドは **correlation_Id** になります。

Fuse Online での手順の概要

Fuse Online で Salesforce および ServiceNow コネクションを作成します。

以下のステップのとおり、2つのインテグレーションは、Salesforce コネクションで実行されるアクション以外は同じになります。インテグレーションを作成するには、これらのステップを2回実行し、1回目は各インテグレーションを作成します。

1. Salesforce コネクションを最初のコネクションとして追加します。
 - 新しいケースを取得するインテグレーションを作成するには、**On create** アクションを選択します。

- 更新されたケースを取得するインテグレーションを作成するには、**On update** アクションを選択します。
2. アクションが操作する **Case** オブジェクトを選択します。この Salesforce コネクションは、選択したアクションに応じて、新しい **Case** オブジェクトまたは更新された **Case** オブジェクトを返します。
 3. ServiceNow コネクションを最後のコネクションとして追加します。
 4. **Create Record** アクションを選択した後、インテグレーションが更新するインポートセットとして **Cases from Salesforce** を選択します。
この ServiceNow コネクションは、新しいまたは更新された Salesforce ケースを **Cases from Salesforce** インポートセットに追加します。
 5. 以下をマップするデータマップステップを追加します。
 - Salesforce ケース **id** から ServiceNow インポートセットの **correlation_Id**
 - Salesforce **subject** から ServiceNow **short_description**
 - Salesforce **description** から ServiceNow **description**
 6. インテグレーションに名前を付け、パブリッシュします。
 - 新しいケースを取得するインテグレーションでは、**On SF Create Case** を使用します。
 - 更新されたケースを取得するインテグレーションでは、**On SF Update Case** を使用します。

インテグレーションの動作確認

両方のインテグレーションが実行しているときに、インテグレーションの動作を確認することができます。

1. Salesforce でケースを作成します。
2. Fuse Online で **On SF Create Case** インテグレーションの概要を表示します。**Activity** タブをクリックし、Fuse Online がインテグレーションを1度実行したことを確認します。
3. ServiceNow で、**Incidents** テーブルを表示します。Salesforce で指定した件名と説明のある新しいインシデントが表示されるはずです。
4. Salesforce に戻り、作成したケースの件名を変更して、そのケースを更新します。
5. Fuse Online で **On SF Update Case** インテグレーションの概要を表示します。**Activity** タブをクリックし、Fuse Online がインテグレーションを1度実行したことを確認します。
6. ServiceNow で **Incidents** テーブルを表示し、これまで新規だったインシデントのエントリを展開します。このインシデントに更新された **short_description** の値があることが確認できるはずです。ServiceNow は、**correlation_Id** の値のインポートセットエントリをチェックします。この値がすでに **Incidents** テーブルに存在する場合、ServiceNow はその値があるインシデントを更新します。

第19章 SLACK への接続

ビジネスユーザーは Slack に接続するインテグレーションを作成できます。Slack へのコネクションは以下の1つを行うことができます。

- 指定の Slack チャンネルがメッセージを受信するときにシンプルなインテグレーションの実行をトリガーします。インテグレーションはメッセージをフローの次のステップに渡します。たとえば、製品名などのキーワードインスタンスを Slack チャンネルで監視することができます。これらの製品名が含まれるメッセージを見つけた後、インテグレーションは Gmail コネクションの適切な連絡先に通知することができます。
- メッセージを特定ユーザーまたはチャンネルに配信します。たとえば、この動作は、インテグレーションが FTP サーバーからファイルをダウンロードし、何らかの方法で処理する場合に便利です。インテグレーションフローは、プロセスに成功したことを Slack チャンネルまたはユーザーに通知できます。

インテグレーションで Slack に接続するには、Slack コネクションを作成した後にコネクションをインテグレーションフローに追加します。詳細は以下を参照してください。

- [「Slack コネクションの作成」](#)
- [「メッセージの受信時に Slack コネクションを追加してインテグレーションの実行をトリガー」](#)
- [「Slack コネクションを追加してメッセージを Slack チャンネルまたはユーザーに送信」](#)

19.1. SLACK コネクションの作成

インテグレーションでは、Slack コネクションは指定したチャンネルからメッセージを取得したり、メッセージをチャンネルまたはユーザーに送信することができます。同じ Slack コネクションを任意の数のインテグレーションで使用することができます。

前提条件

- Fuse Online を Slack に接続する Slack アプリケーションが作成されている必要があります。
- Slack アプリケーションの Slack Webhook URL を取得できる必要があります。
- Slack アプリへのアクセスを許可するレガシートークンを取得できる必要があります。

手順

1. Fuse Online の左パネルで **Connections** をクリックし、利用可能なコネクションを表示します。
2. 右上にある **Create Connection** をクリックし、Fuse Online コネクタを表示します。
3. **Slack** コネクタをクリックします。
4. **Slack webhook URL** フィールドに Slack アプリケーションに Webhook URL を入力します。
5. **Token for accessing Slack API** フィールドに、[Slack から取得](#) できる Slack 提供のトークンを入力します。
6. 任意で、追加パラメーターの値を入力します。

- **Sending username for messages** フィールドに、Slack のメッセージを送信するときに

- **Sending username for messages** フィールドに、Slack ハメッセンを返信するときにボットが持っているユーザー名を入力します。
- **Message avatar emoji** フィールドに、ボットがメッセージの送信時にメッセージアバターとして使用する絵文字を1つ指定します。例: **:smile:**、**:wave:**、**:question:**

ヒント

絵文字コードについては、**emoji cheat sheet** を Web 検索してください。

- **Message avatar icon URL** フィールドに、ボットがメッセージを Slack に送信するときに使用するアバターの URL を指定します。

絵文字とアイコン URL を指定した場合、インテグレーションはアイコン URL を使用します。絵文字とアイコン URL を両方指定しなかった場合、メッセージはアバターなしで送信されます。

7. **Validate** をクリックします。Fuse Online は即座にコネクションを検証しようとし、検証の成功または失敗を示すメッセージを表示します。検証に失敗した場合は、コネクション設定値を修正し、再度検証を行います。
8. 検証に成功した場合は **Next** をクリックします。
9. **Connection Name** フィールドに、このコネクションを別のコネクションと区別するために使用する名前を入力します。たとえば、**Slack for Tracking Company Sales** を入力します。
10. **Description** は任意のフィールドで、このコネクションに関する便利な情報を入力します。
11. 右上にある **Create** をクリックし、作成したばかりのコネクションを含む、利用可能なコネクションの更新済みリストを表示します。上記の例の名前を入力した場合は、インテグレーションに追加できるコネクションとして **Slack for Tracking Company Sales** が表示されます。このコネクションを任意の数のコネクションに追加できます。

19.2. メッセージの受信時に SLACK コネクションを追加してインテグレーションの実行をトリガー

指定した Slack チャンネルでコネクションがメッセージを見つけると、インテグレーションを開始する Slack コネクションはインテグレーションの実行をトリガーします。

前提条件

- Slack コネクションが作成されている必要があります。
- インテグレーションを作成することになります。Fuse Online はインテグレーションの開始方法の選択を要求します。

手順

1. インテグレーションの開始に使用する Slack コネクションをクリックします。
2. **Read Messages** アクションを設定します。
3. **Channel** フィールドに、メッセージを取得する Slack チャンネルを指定します。

4. **Delay** フィールドで、デフォルトの 500 ミリ秒をメッセージのポーリング間の経過時間として受け入れます。別の間隔でメッセージをチェックする場合は、数値を入力し、時間の単位を選択します。
5. **Maximum Messages to Retrieve** フィールドには、デフォルトの 10 を使用するか、各ポーリングでコネクションが返すことができるメッセージの最大数を指定します。コネクションがこの値を超える数の未読メールを見つけた場合、最も新しいメッセージを返します。コネクションは特定のメッセージを 1 度のみ返します。
たとえば、デフォルトの 10 を使用し、ポーリング時に 15 個のメッセージがあった場合、コネクションは新しい順に 10 個のメッセージを返します。次のポーリングでは、コネクションが返した最新のメッセージよりも新しいタイムスタンプを持つメッセージをチェックします。そのため、最初のポーリングで返されなかった 5 個のメッセージは永遠に返されません。

Slack コネクションが 1 つ以上のメッセージを返すとき、インテグレーションはメッセージをバッチとして処理します。つまり、Fuse Online はバッチに対してインテグレーションを 1 度実行します。
6. **Done** をクリックしてコネクションをインテグレーションに追加します。

19.3. SLACK コネクションを追加してメッセージを SLACK チャネルまたはユーザーに送信

インテグレーションでは、フローの途中でメッセージを Slack チャネルに送信したり、シンプルなインテグレーションを終了するためにメッセージを Slack チャネルに送信することができます。これには、Slack コネクションをフローの途中に追加するか、シンプルなインテグレーションの最後のコネクションとして追加します。

前提条件

- Slack コネクションが作成されている必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。または、Fuse Online で最後のコネクションを選択するよう要求されます。

手順

1. フローに追加する Slack コネクションをクリックします。
2. コネクションが実行するアクションを選択します。
 - **Username** をクリックし、メッセージを 1 名のユーザーに送信します。このアクションを設定するには、**User name** フィールドにメッセージの送信先となるユーザーの Slack 名を指定します。
 - **Channel** をクリックして、チャネルにメッセージをパブリッシュします。このアクションを設定するには、**Channel** フィールドにメッセージをパブリッシュするチャネルを指定します。
3. **Done** をクリックして、コネクションをフローに追加します。
4. 任意で、追加のコネクションをフローに追加します。追加のコネクションが必要であるかどうかは、フローが何を行うかによって異なります。すべてのコネクションを追加してから手順を続行することが重要になります。
5. この手順を追加する Slack コネクションの直前にデータマッパーステップを追加します。マッピングステップでは、文字列をデータマッピングソースから Slack **message** フィールドにマッ

ブします。この文字列には、Slack ユーザーまたはチャンネルに送信するメッセージが含まれる必要があります。「[Adding a data mapping step](#)」を参照してください。

第20章 SQL データベースへの接続

インテグレーションでは、以下の SQL データベースに接続することができます。

- Apache Derby
- MySQL
- PostgreSQL

インテグレーションでアクセスするデータベースへのコネクションを作成し、そのコネクションを次のインテグレーションに追加します。

他の種類のデータベースに接続するには、そのデータベースの JDBC ドライバーをアップロードする必要があります。

以下を参照してください。

- [「データベースコネクションの作成」](#)
- [「データベースにアクセスしてインテグレーションを開始」](#)
- [「インテグレーションの途中またはインテグレーションを完了するためにデータベースにアクセス」](#)
- [「プロプライエタリーデータベースへの接続」](#)

20.1. データベースコネクションの作成

インテグレーションで接続する各データベースの個別のコネクションを作成します。複数のインテグレーションで同じコネクションを使用できます。

データベースコネクションは指定のデータベーステーブルで操作するか、指定のストアドプロシージャを呼び出します。

前提条件

インテグレーションがデータベースに接続するときにデータベーステーブルまたはストアドプロシージャが存在する必要があります。

手順

1. 接続するデータベースの JDBC ドライバーがクラスパスにあるようにしてください。プロプライエタリーデータベースに接続するために JDBC ドライバーライブラリーエクステンションをアップロードした場合、ドライバはアップロードプロセスによってクラスパス上に置かれます。「[Creating JDBC driver library extensions](#)」を参照してください。
2. Fuse Online の左パネルで **Connections** をクリックし、利用可能なコネクションを表示します。
3. 右上にある **Create Connection** をクリックし、Fuse Online コネクターを表示します。
4. **Database** コネクターをクリックします。
5. 以下を入力してコネクションを設定します。

- a. **Connection URL** フィールドに、接続するデータベースの JDBC URL を入力します。たとえば、**jdbc:postgresql://ignite-db1234/sampledb** を入力します。
 - b. **Username** フィールドに、データベースのアクセスに使用するアカウントの名前を入力します。同じアカウントのパスワードとユーザー名を指定するようにしてください。
 - c. **Password** フィールドに、データベースのアクセスに使用する、ユーザーアカウントに関連するパスワードを入力します。
 - d. **Schema** フィールドにデータベースのスキーマの名前を入力します。コネクション URL がスキーマを指定する場合、このフィールドがコネクション URL と同じスキーマを示すようにしてください。たとえば、**sampledb** を入力します。
6. **Validate** をクリックします。Fuse Online はコネクションの検証を試み、検証の成功または失敗を示すメッセージを表示します。検証に失敗した場合は、必要に応じて設定の詳細を修正し、再度検証を行います。
 7. 検証に成功した場合は **Next** をクリックします。
 8. **Connection Name** フィールドに、このコネクションを別のコネクションと区別するために使用する名前を入力します。たとえば、**PostgreSQL DB 1** を入力します。
 9. **Description** は任意のフィールドで、このコネクションに関する便利な情報を入力します。たとえば、**Sample PostgreSQL connection that uses my login credentials.** を入力します。
 10. 右上にある **Create** をクリックし、作成したコネクションが利用可能になったことを確認します。上記の例の名前を入力した場合は、インテグレーションに追加できるコネクションとして **PostgreSQL DB 1** が表示されます。

20.2. データベースにアクセスしてインテグレーションを開始

SQL クエリーまたは SQL ストアドプロシージャの呼び出し結果を基にインテグレーションの実行をトリガーするには、データベースコネクションをインテグレーションの最初のコネクションとして選択します。

前提条件

データベースコネクションが作成されている必要があります。

手順

1. Fuse Online パネルの左側にある **Integrations** をクリックします。
2. **Create Integration** をクリックします。
3. **Choose a Start Connection** ページで、インテグレーションを開始するために使用するデータベースコネクションをクリックします。
4. **Choose an Action** ページで、以下の1つをクリックします。
 - **Periodic SQL invocation** は、指定した SQL クエリーを定期的に呼び出して、データを取得します。
 - **Periodic stored procedure invocation** は、選択または指定したストアドプロシージャを定期的に呼び出して、データを取得します。

5. **Periodic SQL invocation** を選択した場合、**Query** フィールドに SQL **SELECT** ステートメントを入力し、1つ以上のレコードを取得します。例: **SELECT * from my_db_table** 必要なデータが含まれるデータベーステーブルが存在する必要があります。
Periodic stored procedure invocation を選択した場合、**Procedure name** フィールドに必要なデータを取得するために呼び出すストアドプロシージャを選択または入力します。指定するストアドプロシージャが存在する必要があります。データベース管理者は、インテグレーションで使用する必要があるストアドプロシージャを作成しておく必要があります。
6. **Period** フィールドに整数を入力し、単位がミリ秒、秒、分、時間、または日であることを示します。たとえば、**5 minutes** を指定した場合、コネクションは指定のクエリーまたはストアドプロシージャを 5 分間隔で呼び出します。
7. **Done** をクリックします。

結果

Fuse Online はコネクションの検証を試みます。これには、指定の SQL クエリーの構文チェックや、クエリーまたはストアドプロシージャターゲットデータの存在確認が含まれます。これには、指定の SQL クエリーの構文チェックや、クエリーまたはストアドプロシージャターゲットデータの存在確認が含まれます。検証に失敗した場合、問題に関するメッセージが Fuse Online に表示されます。必要に応じて入力を更新し、再試行します。

20.3. インテグレーションの途中またはインテグレーションを完了するためにデータベースにアクセス

インテグレーションでは、フローの途中でデータベースにアクセスしたり、シンプルなインテグレーションを終了するためにデータベースにアクセスすることができます。これには、データベースコネクションをフローの途中に追加するか、シンプルなインテグレーションの最後のコネクションとして追加します。

前提条件

- データベースコネクションが作成されている必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。または、Fuse Online で最後のコネクションを選択するよう要求されます。

手順

1. アクセスするデータベースのデータベースコネクションをクリックします。
2. **Choose an Action** ページで、以下の1つをクリックします。
 - **Invoke SQL** は、指定の SQL ステートメントを実行してデータを操作します。
 - **Invoke stored procedure** は指定または選択したストアドプロシージャを呼び出してデータを操作します。
3. **Invoke SQL** を選択した場合、**SQL Statement** フィールドで以下を行います。
 - 途中のコネクションでは、1つ以上のレコードを取得する SQL **SELECT** ステートメントを入力するか、1つ以上のレコードを操作する SQL **INSERT**、**UPDATE**、または **DELETE** ステートメントを入力します。データが含まれるデータベースが存在する必要があります。
 - 最後のコネクションでは、SQL **INSERT**、**UPDATE**、または **DELETE** ステートメントを入力し、1つ以上のレコードを操作します。

Invoke stored procedure を選択した場合、**Procedure name** フィールドに必要なデータを操作するために呼び出すストアドプロシージャの名前を選択または入力します。指定するストアドプロシージャが存在する必要があります。データベース管理者は、インテグレーションで使用する必要があるストアドプロシージャを作成しておく必要があります。

- クエリーでのプレースホルダーパラメーターの指定に関する情報は以下を参照してください。

4. **Done** をクリックします。

結果

Fuse Online はコネクションの検証を試みます。これには、指定の SQL クエリーの構文チェックや、クエリーまたはストアドプロシージャターゲットデータの存在確認が含まれます。検証に成功した場合、Fuse Online でコネクションがフローに追加されます。検証に失敗した場合、問題に関するメッセージが Fuse Online に表示されます。必要に応じて入力を更新し、再試行します。

クエリーでのパラメーターの指定

フローの途中でデータベースにアクセスしたり、インテグレーションを終了するためにデータベースにアクセスする場合、SQL クエリーにプレースホルダーパラメーターを指定でき、ストアドプロシージャにプレースホルダーを追加することができます。以下に例を示します。

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

これらのプレースホルダーの値を指定するには、フローのデータベースコネクションの前にデータマッピングステップを追加します。データマッピングステップで、適切なソースデータフィールドをターゲットデータフィールドにマップします。たとえば、ソースデータを **:#param_1**、**:#param_2**、および **:#param_3** ターゲットフィールドにマップします。「[Adding a data mapping step](#)」を参照してください。

20.4. プロプライエタリーデータベースへの接続

プロプライエタリー SQL データベースに接続する場合、以下の主なタスクを行う必要があります。

- 開発者は、インテグレーションで接続するデータベースの JDBC ドライバーが含まれるライブラリーエクステンションを作成します。「[Creating JDBC driver library extensions](#)」を参照してください。
- 開発者はライブラリーエクステンションが含まれる **.jar** ファイルを提供します。
- その **.jar** ファイルを Fuse Online にアップロードします。「[Making extensions available](#)」を参照してください。
- Fuse Online の **Database** コネクタを選択し、データベースのコネクション URL を指定して、データベースへのコネクションを作成します。「[データベースコネクションの作成](#)」を参照してください。
- インテグレーションで、コネクションをデータベースに追加します。「[データベースレコードを取得してインテグレーションの実行をトリガー](#)」または「[インテグレーションの途中またはインテグレーションを完了するためにデータベースにアクセス](#)」を参照してください。

第21章 TELEGRAM への接続

ビジネスユーザーは Telegram に接続するインテグレーションを作成できます。Telegram への接続は以下の1つを行うことができます。

- 指定の Telegram ボットがメッセージを受信するときにシンプルなインテグレーションの実行をトリガーします。インテグレーションはメッセージをフローの次のステップに渡します。たとえば、シンプルなインテグレーションがメッセージを受信した後に、製品名などのキーワードインスタンスを監視するフィルタステップがあることがあります。これらの製品名が含まれるメッセージを見つけた後、インテグレーションは Gmail コネクションまたは異なる Telegram コネクションの適切な連絡先に通知することができます。
- メッセージを特定の Telegram チャットに配信します。たとえば、この動作は、インテグレーションが FTP サーバーからファイルをダウンロードし、何らかの方法で処理する場合に便利です。プロセスに成功したことを Telegram チャットに通知する Telegram コネクションを追加できます。

インテグレーションで Telegram に接続するには、Telegram コネクションを作成した後に接続をインテグレーションフローに追加します。詳細は以下を参照してください。

- [「Telegram コネクションの作成」](#)
- [「メッセージの受信時に Telegram コネクションを追加してインテグレーションの実行をトリガー」](#)
- [「メッセージを Telegram チャットに送信するため Telegram コネクションを追加」](#)

21.1. TELEGRAM コネクションの作成

インテグレーションでは、Telegram コネクションは特定のチャットに送信されたメッセージを受信できます。作成したチャットボットはメッセージを Fuse Online に転送します。Telegram コネクションもメッセージをコネクションがアクセスできるチャットに送信することができます。以下で、同じ Telegram コネクションを使用できます。

- コネクションを使用してメッセージをチャットに送信する任意の数のインテグレーション。
- さらに、Telegram チャットからメッセージを受信するためにコネクションを使用する1つのインテグレーションで 사용할 ことができます。

そのため、コネクションが両方のインテグレーションで Telegram チャットメッセージを受信する場合、同じ Telegram コネクションを複数のインテグレーションで使用できません。これが Telegram の制限になります。

前提条件

- Telegram で、メッセージの取得元となるチャットのボットが作成されている必要があります。ヘルプが必要な場合は [「Bots, an introduction for developers」](#) を参照してください。
- Telegram BotFather からボット承認トークンを取得済みである必要があります。

手順

1. Fuse Online の左パネルで **Connections** をクリックし、利用可能なコネクションを表示します。
2. 右上にある **Create Connection** をクリックし、Fuse Online コネクターを表示します。

3. **Telegram** コネクターをクリックします。
4. **Authorization Token** フィールドに、メッセージの受信元および送信先となるボットの Telegram 承認トークンを入力します。
5. **Next** をクリックします。
6. **Connection Name** フィールドに、このコネクションを別のコネクションと区別するために使用する名前を入力します。たとえば、**Telegram for Product Mentions** を入力します。
7. **Description** は任意のフィールドで、このコネクションに関する便利な情報を入力します。
8. 右上にある **Create** をクリックし、作成したばかりのコネクションを含む、利用可能なコネクションの更新済みリストを表示します。上記の例の名前を入力した場合は、インテグレーションに追加できるコネクションとして **Telegram for Product Mentions** が表示されます。このコネクションを任意の数のコネクションに追加できます。

21.2. メッセージの受信時に TELEGRAM コネクションを追加してインテグレーションの実行をトリガー

インテグレーションを開始する Telegram コネクションは、コネクションが指定の Telegram チャットボットからメッセージを受信するたびにインテグレーションの実行をトリガーします。

前提条件

- Telegram コネクションが作成されている必要があります。
- シンプルなインテグレーションを作成することになります。Fuse Online はインテグレーションの開始方法の選択を要求します。

手順

1. インテグレーションの開始に使用する Telegram コネクションをクリックします。
2. **Read Messages** アクションをクリックし、チャットボットに送信されたすべてのメッセージを受信します。
コネクション設定は、メッセージの受信元となるチャットボットを特定します。他の設定は必要ありません。

21.3. メッセージを TELEGRAM チャットに送信するため TELEGRAM コネクションを追加

インテグレーションでは、フローの途中でメッセージを Telegram チャットに送信したり、シンプルなインテグレーションを終了するために Telegram チャットを送信することができます。これには、Telegram コネクションをフローの途中に追加するか、シンプルなインテグレーションの最後のコネクションとして追加します。

前提条件

- Telegram コネクションが作成されている必要があります。
- フローを作成または編集することになります。Fuse Online はステップの選択を要求します。または、Fuse Online で最後のコネクションを選択するよう要求されます。

手順

1. フローに追加する Telegram コネクションをクリックします。
2. **Send a Text Message** アクションをクリックします。
3. **Chat Id** フィールドに、メッセージの送信先となる Telegram チャットの ID を入力します。
このフィールドを空白のままにし、メッセージを受信する以前の Telegram コネクションがある場合、Fuse Online は受信メッセージにあるチャット ID を使用します。

チャットの ID を取得するには以下を行います。

- a. Telegram で、ボットをメッセージの送信先となるチャットに追加します。BotFather 応答にはボットアクセストークンが含まれます。
- b. アクセストークンをクリップボードにコピーします。
- c. Telegram で、メッセージを直接そのチャットに送信します。メッセージの送信にボットは使用しないでください。
- d. 以下の形式で **curl** コマンドを呼び出し、チャットに送信されたメッセージを読み取ります。**bot-access-token** はクリップボードのアクセストークンに置き換えてください。

curl https://api.telegram.org/bot-access-token/getUpdates\?offset\=0

応答は JSON 形式になります。JSON **chat** オブジェクトにはチャット ID が含まれます。

4. **Done** をクリックして、コネクションをフローに追加します。

第22章 タイマーによるインテグレーション実行のトリガー

指定のスケジュールに応じてシンプルなインテグレーションの実行をトリガーするには、タイマーコネクションをシンプルなインテグレーションの最初のコネクションとして追加します。Fuse Online はタイマーコネクションを提供し、任意の数のシンプルなインテグレーションを開始するために使用できます。提供されたタイマーを誤って削除しない限り、タイマーコネクションを作成する必要はありません。詳細は以下を参照してください。

- [「インテグレーションの実行をトリガーするためタイマーコネクションを追加」](#)
- [「代替タイマーコネクションの作成」](#)

22.1. インテグレーションの実行をトリガーするためタイマーコネクションを追加

指定のスケジュールに応じてシンプルなインテグレーションの実行をトリガーするには、タイマーコネクションをシンプルなインテグレーションの最初のコネクションとして追加します。タイマーコネクションをフローの途中やフローの最後に追加することはできません。

手順

1. Fuse Online で左側にある **Integrations** をクリックします。
2. 右上の **Create Integration** をクリックします。
3. **Choose a Start Connection** ページで **Timer** コネクションをクリックします。
Fuse Online は **Timer** コネクションを提供するため、タイマーコネクションを作成する必要はありません。しかし、提供された **Timer** コネクションを誤って削除し、新しいタイマーコネクションを作成した場合、異なる名前を付けた可能性があります。
4. **Choose an Action** ページで **Cron Timer** または **Simple Timer** をクリックします。
 - **cron** タイマーには、インテグレーションの実行をトリガーするスケジュールを指定する **cron** 式が必要です。
 - 期間とその時間単位の指定を要求されます (例:**5 seconds**、**1 hour**)。使用可能な単位はミリ秒 (milliseconds)、秒 (seconds)、分 (minutes)、時間 (hours)、および日 (days) です。
5. 追加するタイマーのタイプに応じて、**cron** 式または選択した時間単位の期間を入力します。
6. **Done** をクリックして **Timer** コネクションをインテグレーションの最初のコネクションとして追加します。

22.2. 代替タイマーコネクションの作成

Fuse Online はタイマーコネクションを提供するため、タイマーコネクションを作成する必要はありません。しかし、提供されたタイマーコネクションを誤って削除してしまった場合、**Timer** コネクタを使用してタイマーコネクションを作成できます。

手順

1. Fuse Online の左側のパネルで **Connections** をクリックします。
2. 右上の **Create Connection** をクリックします。

3. **Timer** コネクタをクリックします。
4. **Add Connection Details** ページに、コネクションの名前を入力します。たとえば、**Timer** を入力します。
5. 任意で、このコネクションを区別するための説明を入力します。
6. 右上の **Create** をクリックします。Fuse Online は使用できるコネクションのリストを表示し、作成したタイマーコネクションも表示されます。

第23章 TWITTER への接続

インテグレーションで Twitter に接続するための主なステップは次のとおりです。

- 「[Fuse Online を Twitter クライアントアプリケーションとして登録](#)」
- 「[Twitter コネクションの作成](#)」
- 「[Twitter コネクションを追加してインテグレーションの実行をトリガー](#)」

Fuse Online 環境から Twitter へのアクセスを承認するため、Twitter 開発者アカウントが必要です。Twitter 開発者アカウントを持っていない場合は、<https://apps.twitter.com> にアクセスし、Twitter にログインしてから **Apply for a developer account** をクリックします。**you@gmail.com** のような個人アドレスではなく、**me@redhat.com** のような企業のアドレスを提供すると、迅速に開発者アカウントを取得できます。

23.1. FUSE ONLINE を TWITTER クライアントアプリケーションとして登録


インテグレーションで Twitter に接続するには、最初に Fuse Online 環境を Twitter にアクセスできるクライアントアプリケーションとして登録する必要があります。これにより、Twitter に接続するインテグレーションをいくつでも作成することができます。そのため、特定の Fuse Online 環境を1度だけ Twitter に登録する必要があります。

Fuse Online を Twitter クライアントアプリケーションとして登録できるのは、Fuse Online 環境ごとに1回のみです。しかし、各 Twitter コネクションは同じ登録を使用しますが、異なるユーザークレデンシャルを使用することができます。

前提条件

Fuse Online にログインしている必要があります。Fuse Online にアクセスできる URL は、Red Hat Fuse Online Evaluation プログラムのウェルカムメールメッセージにあります。

手順

1. Fuse Online で以下を行います。
 - a. 左側のパネルで **Settings** をクリックします。
 - b. **Settings** ページで、上部付近にあるコールバック URL の右側で  をクリックし、Fuse Online インストールのコールバック URL をクリップボードにコピーします。この手順の後半でこの URL が必要になります。
 - c. **Twitter** エントリーの右側にある **Register** をクリックし、**Consumer API Key** および **ConsumerAPI Secret Key** フィールドを表示します。
2. 別のブラウザータブで、[Twitter 開発者アカウント](#) にアクセスし、必要な場合はログインしてから以下を行います。
 - a. 右上で、開発者プロジェクトの名前の横にある下向きのチャレットをクリックし、**Apps** をクリックします。
 - b. 右上の **Create an app** をクリックします。
 - c. **App name** フィールドに **My Fuse Online Client** などの名前を入力します。

- d. **Application description** フィールドに、**For leveraging tweets** などのヒントを入力します。
 - e. **Website URL** フィールドに、この手順の最初でコピーした URL を貼り付け、URL の最後にある **api/v1/credentials/callback** を削除します。
 - f. **Enable Sign in with Twitter** を選択します。
 - g. **Callback URLs** フィールドに URL を再度貼り付けます。 **https://app-proj912876.7b63.fuse-ignite.openshiftapps.com/api/v1/credentials/callback** のようになるはずです。
 - h. このアプリの使用目的を 100 文字以内で説明する必要があります。 **This is a demo application so that I can evaluate how to integrate Twitter with my enterprise applications.** をコピーし、貼り付けてもいいでしょう。
 - i. **Create** をクリックします。
 - j. 再度 **Create** をクリックし、開発者規約を読んだことを確認します。
 - k. 作成したアプリのページ上部にある **Keys and tokens** タブをクリックします。
 - l. コンシューマー API キーをコピーします。
3. Fuse Online の **Settings** ページに戻り、Twitter のコンシューマー API キーを Fuse Online の Twitter **Consumer API Key** フィールドに貼り付けます。
 4. Twitter の **Keys and tokens** タブに戻り、コンシューマー API シークレットキーをコピーします。
 5. Fuse Online の **Settings** ページに戻り、Twitter のコンシューマー API シークレットキーを Fuse Online の Twitter **Consumer API Secret Key** フィールドに貼り付けます。
 6. **Save** をクリックし、**Ok** をクリックします。

23.2. TWITTER コネクションの作成

Twitter からデータを取得するインテグレーションを作成するには、最初に Twitter コネクションを作成する必要があります。Twitter コネクションを作成した後、任意の数のインテグレーションで使用できます。

前提条件

- Fuse Online にログインしている必要があります。Fuse Online にアクセスできる URL は、Red Hat Fuse Online Evaluation プログラムのウェルカムメールメッセージにあります。
- Fuse Online がウェブブラウザで開かれている必要があります。
- Fuse Online 環境が Twitter にアクセスできるアプリケーションとして登録されている必要があります。
- Fuse Online の **Settings** ページで登録した後に受け取った Twitter コンシューマー API キーおよびコンシューマー API シークレットキーが追加済みである必要があります。

手順

1. Fuse Online の左パネルで **Connections** をクリックし、利用可能なコネクションを表示します。
2. 右上の **Create Connection** をクリックし、利用可能なコネクタを表示します。コネクタは、1つ以上のコネクションを作成するために使用するテンプレートです。
3. **Twitter** コネクタをクリックします。
4. **Connect Twitter** をクリックし、Twitter の承認ページを表示します。承認ページを表示するには Twitter にログインする必要があります。承認ページを表示するには Twitter にログインする必要があります。
Connect Twitter が表示されない場合、Fuse Online 環境は Twitter クライアントアプリケーションとして登録されていません。「[Fuse Online を Twitter クライアントアプリケーションとして登録](#)」を参照してください。Twitter コネクションの作成時に Fuse Online 環境が Twitter クライアントアプリケーションとして登録されていない場合、Fuse Online は承認情報を要求する複数のフィールドを表示します。これらのフィールドに値を入力して Twitter コネクションを作成することはできますが、推奨されません。
5. **Authorize app** をクリックし、Fuse Online に戻ります。
6. **Connection Name** フィールドに、このコネクションを別のコネクションと区別するために使用する名前を入力します。たとえば、**Twitter Connect 1** を入力します。
7. **Description** は任意のフィールドで、このコネクションに関する便利な情報を入力します。たとえば、**Sample Twitter connection that uses my Twitter login credentials.** を入力します。
8. 右上にある **Create** をクリックし、作成したコネクションが利用可能になったことを確認します。上記の例の名前を入力した場合は、インテグレーションに追加できるコネクションとして **Twitter Connect 1** が表示されます。

23.3. TWITTER コネクションを追加してインテグレーションの実行をトリガー

インテグレーションでは、ツイートに Twitter ハンドル名や指定のテキストが含まれている場合に Twitter へのコネクションはインテグレーションの実行をトリガーすることができます。Twitter へのコネクションはインテグレーションの最初のコネクションである必要があります。

前提条件

- Twitter コネクションが作成されている必要があります。
- インテグレーションを作成または編集することになります。Fuse Online は最初のコネクションの選択を要求します。

手順

1. インテグレーションに追加する Twitter コネクションをクリックします。インテグレーションが Twitter にコネクションするために選択したコネクションを使用する場合、Fuse Online はそのコネクションに定義されたクレデンシャルを使用します。
2. 選択したコネクションが実行するアクションをクリックします。インテグレーションに追加した Twitter コネクションは選択したアクションのみを実行します。
 - **Mention** は、ツイートにハンドル名が含まれる場合にインテグレーションの実行をトリガーします。

- **Search** はツイートに指定のテキストが含まれる場合にインテグレーションの実行をトリガーします。
3. 任意で、Fuse Online が要求する設定情報を入力します。たとえば、**Search** アクションは、検索の頻度と検索するキーワードを指定するよう要求します。
 4. **Done** をクリックしてコネクションをインテグレーションに追加します。

結果

インテグレーションフローの最初にコネクションが表示されます。

第24章 HTTP リクエストによるインテグレーション実行のトリガー (WEBHOOK)

HTTP **GET** または **POST** リクエストを Fuse Online が公開する HTTP エンドポイントに送信して、シンプルなインテグレーションの実行をトリガーできます。詳細は以下のトピックを参照してください。

- [「Fuse Online Webhook を使用するための一般的な手順」](#)
- [「Webhook コネクションの使用」](#)
- [「HTTP リクエストがトリガー可能なインテグレーションの作成」](#)
- [「Fuse Online による HTTP リクエストの処理方法」](#)
- [「Fuse Online Webhook を呼び出す HTTP クライアントのガイドライン」](#)
- [「リクエストパラメーターを指定するための JSON スキーマ」](#)
- [「HTTP リクエストの指定方法」](#)

24.1. FUSE ONLINE WEBHOOK を使用するための一般的な手順

HTTP **GET** または **POST** リクエストでインテグレーションの実行をトリガーするには、以下を行う必要があります。

1. **GET** または **POST** リクエストを Fuse Online に送信するかどうかを決定します。
2. このリクエストを処理するようインテグレーションを計画します。
3. インテグレーションを終了するコネクションを作成します。
Fuse Online は、最初のコネクションとして使用する Webhook コネクションを提供します。
4. インテグレーションに追加する他のコネクションを作成します。
5. インテグレーションを作成します。
 - a. Webhook コネクションを最初のコネクションとして追加します。
 - b. 最後のコネクションを追加した後、インテグレーションで必要な他のコネクションを追加します。最後のコネクションと途中のコネクションは、インテグレーションの実行をトリガーする HTTP リクエストを処理します。目的を達成するために最も適切な HTTP リクエストを選択および指定するのはユーザー自身です。これには以下を考慮してください。
 - 取得または更新するデータが含まれるアプリケーションへのコネクションを追加します。
 - **GET** リクエストは、キー/値パラメーターの指定に限定されます。
 - **POST** リクエストは、XML や JSON インスタンスなどの任意のボディを提供します。
 - Fuse Online は HTTP ステータスヘッダーのみを返し、データは返しません。そのため、**GET** リクエストによってトリガーされるインテグレーションや、データを取得せずにデータを更新するインテグレーションを定義できます。同様に、**POST** リクエストによってトリガーされるインテグレーションや、データを更新せずにデータを取得するインテグレーションを定義することもできます。

- c. Webhook コネクションの後にデータマップステップを追加します。
GET リクエストでは、HTTP リクエストのパラメーターフィールドを次のコネクションのデータフィールドにマップします。

POST リクエストの場合、JSON インスタンス、JSON スキーマ、XML インスタンス、または SML スキーマを渡して、リクエストに出力データシェイプを指定した可能性があります。指定しなかった場合は、Webhook コネクションをインテグレーションの最初のコネクションとして追加します。指定しないと、Webhook コネクションの出力データタイプのデフォルトは JSON 形式になります。
 - d. インテグレーションに必要な他のステップを追加します。
6. インテグレーションをパブリッシュし、**Running** 状態になるまで待ちます。
 7. インテグレーション概要ページに移動し、Fuse Online が提供する外部 URL をコピーします。
 8. 外部 URL を編集して、**GET** または **POST** リクエストを作成します。
 9. HTTP **GET** または **POST** リクエストを Fuse Online に送信するアプリケーションを実装します。

24.2. WEBHOOK コネクションの使用

Fuse Online は、任意の数のシンプルなインテグレーションで最初のコネクションとして使用できる Webhook コネクションを提供します。この Webhook コネクションをインテグレーションに追加するたびに、Fuse Online はこのインテグレーションと使用する新しいランダムなトークンを提供します。**GET** または **POST** リクエストの最後にこのトークンを貼り付けます。提供された Webhook コネクションを誤って削除しない限り、Webhook コネクションを作成する必要はありません。

前提条件

Fuse Online によって提供された Webhook コネクションを誤って削除した場合。

手順:

1. Fuse Online の左パネルで **Connections** をクリックし、利用可能なコネクションを表示します。
2. 右上の **Create Connection** をクリックしてコネクタを表示します。
3. **Webhook** コネクタをクリックします。
4. **Connection Name** フィールドに、このコネクションを別のコネクションと区別するために使用する名前を入力します。たとえば、**Webhook 1** を入力します。
5. **Description** は任意のフィールドで、このコネクションに関する便利な情報を入力します。たとえば、**Sample Webhook connection.** を入力します。
6. 右上にある **Create** をクリックし、作成したコネクションが利用可能になったことを確認します。上記の例の名前を入力した場合は、インテグレーションに追加できるコネクションとして **Webhook 1** が表示されます。

24.3. HTTP リクエストがトリガー可能なインテグレーションの作成

HTTP **GET** または **POST** リクエストでインテグレーションの実行をトリガーするには、Webhook コネクションをインテグレーションの最初のコネクションとして追加します。

手順

1. Fuse Online パネルの左側にある **Integrations** をクリックします。
2. **Create Integration** をクリックします。
3. **Choose a Start Connection** ページで Webhook コネクションをクリックします。
4. **Choose an Action** ページで **Incoming Webhook** アクションをクリックします。
Webhook Configuration ページで、Fuse Online はこのインテグレーションのために生成した Webhook トークンを表示します。

HTTP リクエストを作成するとき、このトークンは URL の最後の部分になります。このインテグレーションをパブリッシュし、稼働状態になった後、Fuse Online はこのトークンが末尾にある Fuse Online 外部 URL を表示します。

5. **Next** をクリックします。
6. **Specify Output Data Type** ページで以下を行います。
 - a. **Select Type** フィールドで **JSON schema** を選択します。
 - b. **Definition** フィールドに、HTTP リクエストでパラメーターのデータタイプを定義する JSON スキーマを貼り付けます。「[リクエストパラメーターを指定するための JSON スキーマ](#)」を参照してください。
 - c. **Data Type Name** フィールドに、このデータタイプの名前を指定します。これは任意のフィールドですが、名前を指定すると、データマッパーの **Sources** リストに表示されるため、フィールドを正しくマップしやすくなります。
 - d. 必要に応じて、このデータタイプを区別するための情報を **Data Type Description** フィールドに入力します。
 - e. **Done** をクリックします。
7. 最後のコネクションをインテグレーションに追加します。
8. 必要な他のコネクションを追加します。
9. 必要な他のステップを追加します。
10. 最初のコネクションの直後に、データマッピングステップを追加します。
11. **Publish** をクリックし、インテグレーションの名前を付け、任意で説明を追加した後、再度 **Publish** をクリックします。

24.4. FUSE ONLINE による HTTP リクエストの処理方法

HTTP **GET** または **POST** リクエストを指定して、シンプルなインテグレーションの実行をトリガーできます。通常 **GET** リクエストはデータを取得し、**POST** リクエストはデータを更新しますが、いずれかのリクエストを使用して、いずれかのオペレーションを行うインテグレーションをトリガーできます。リクエストのパラメーターはすべてインテグレーションの次のコネクションにあるデータフィールドへのマッピングに利用できます。詳細は「[リクエストパラメーターを指定するための JSON スキーマ](#)」を参照してください。

Webhook コネクションは受信するデータのみをインテグレーションの次のコネクションに渡します。Fuse Online が HTTP リクエストを受信すると、以下を行います。

- HTTP ステータスヘッダーを要求元に返します。リクエストがインテグレーションの実行を正常にトリガーした場合、Fuse Online の戻りコードは **201** になります。リクエストがインテグレーション実行のトリガーに失敗した場合、戻りコードは **5xx** になります。
- 他のデータを要求元に返しません。ステータスヘッダーが含まれる 応答 の HTTP ボディーにはデータがありません。
- リクエストのデータをインテグレーションの次のコネクションに渡します。

そのため、**GET** リクエストによってトリガーされるシンプルなインテグレーションや、データを取得せずにデータを更新するインテグレーションを定義できます。同様に、**POST** リクエストによってトリガーされるシンプルなインテグレーションや、データを更新せずにデータを取得するインテグレーションを定義することもできます。

24.5. FUSE ONLINE WEBHOOK を呼び出す HTTP クライアントのガイドライン

HTTP リクエストを Fuse Online に送信するクライアントを実装する場合、実装は以下を行う必要があります。

- **GET** または **POST** リクエストを作成する URL を作成するため、Fuse Online が提供する外部 URL に追加します。
- URL リクエストに、**io:syndesis:webhook** JSON スキーマに準拠するデータタイプを持つ、HTTP ヘッダーとクエリーパラメーターの値を指定します。「[リクエストパラメーターを指定するための JSON スキーマ](#)」を参照してください。ヘッダーとクエリーパラメーターがこのデータタイプの指定に準拠する場合、パラメーターフィールドをインテグレーションの次のコネクションが処理できるフィールドにマップできます。
- リクエストに成功した場合、返された成功コード **201** を処理します。
- リクエストに失敗した場合、HTTP の **5xx** エラーコードを処理します。
- Fuse Online からの他の応答を想定しません。そのため、リクエストを送信しても戻りコード以外の戻りデータは直接要求元のクライアントに返されません。

24.6. リクエストパラメーターを指定するための JSON スキーマ

インテグレーションでは、通常 HTTP リクエストのヘッダーおよびクエリーパラメーターをインテグレーションの次のコネクションで処理できるデータフィールドにマップします。これを可能にするには、Webhook コネクションをインテグレーションに追加するときに、以下の構造を持つ JSON スキーマの出力データタイプを指定します。

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

```

    "properties": { ❷
      }
    }
  }
}

```

必要なデータ構造を追加するには、HTTP リクエストの JSON インスタンスで以下を行います。

- ❶ **properties** オブジェクト下の **parameters** セクションにクエリーパラメーターを指定します。
- ❷ **properties** オブジェクト下の **body** セクションに HTTP ボディースキーマを指定します。

HTTP クライアントが送信するデータはすべてインテグレーションで利用できますが、Webhook コネクションのデータシェイプがこの JSON スキーマに準拠する場合はクエリーパラメーターとボディーのコンテンツをマッピングに使用できます。

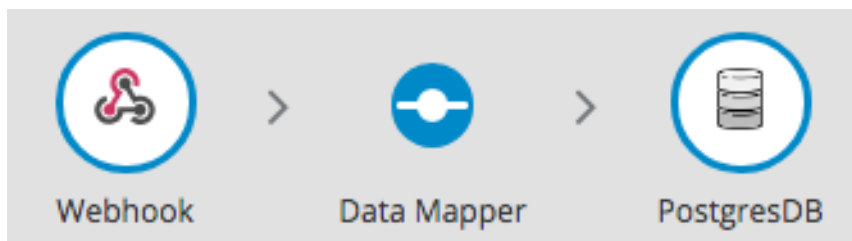
例については「[HTTP リクエストの指定方法](#)」を参照してください。

24.7. HTTP リクエストの指定方法

以下の例は、Fuse Online Webhook に HTTP リクエストを指定する方法を示しています。

Webhook における HTTP ボディーのみの POST リクエストの例

Webhook コネクションで開始し、Fuse Online が提供するデータベースの **Todo** テーブルの行を作成するインテグレーションについて考えます。



このインテグレーションの作成中、Webhook の最初のコネクションを追加するときに **{"todo":"text"}** がコンテンツにある JSON インスタンスで出力データタイプを指定します。

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"/> |
| <input type="button" value="Done"/> | |

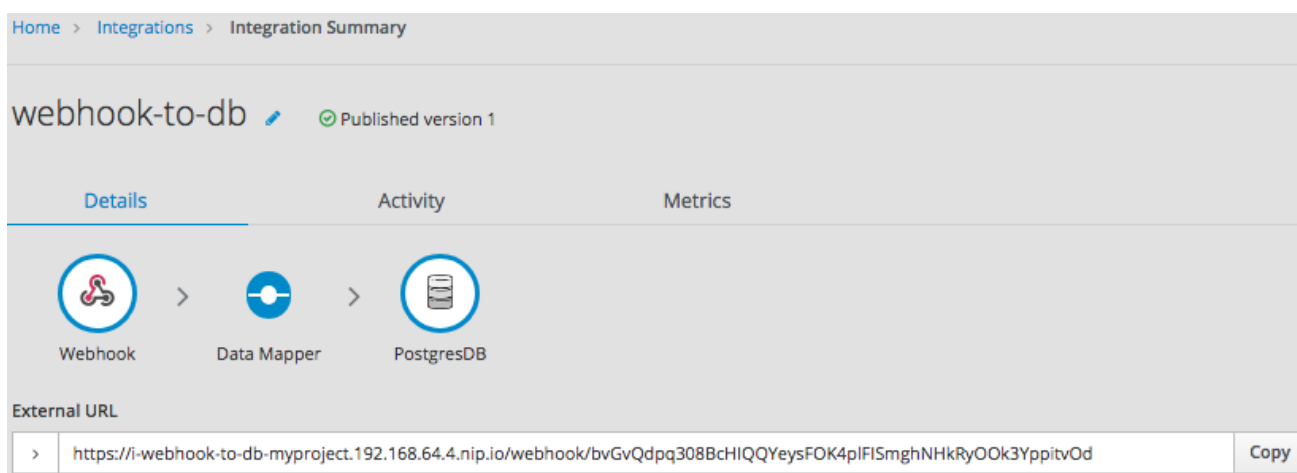
PostgresDB コネクションを最後のコネクションとして追加するとき、**Invoke SQL** アクションを選択し、この SQL ステートメントを指定します。

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

データベースコネクションを追加した後、マッピングステップを追加します。



インテグレーションを保存し、パブリッシュします。実行中に Fuse Online が提供する外部 URL をコピーできます。



外部 URL の一部を理解するため、以下の URL の例を見てください。

https://i-webhook-to-db-myproject.192.168.64.4.nip.io/webhook/bvGvQdpq308BcHIQQYeysFOK4pIFISmghNHkRyOOk3Yp pitvOd

| 値 | 説明 |
|----------------------------|---|
| i- | Fuse Online は常にこの値を URL の最初に挿入します。 |
| webhook-to-db | インテグレーションの名前。 |
| myproject | インテグレーションを実行している Pod が含まれる OpenShift namespace。 |
| 192.168.64.4.nip.io | OpenShift 用に設定された DNS ドメイン。これは、Webhook を提供している Fuse Online 環境を示しています。 |
| webhook | 各 Webhook コネクション URL に表示されます。 |

| 値 | 説明 |
|--|--|
| bvGvQdpq308BcHIQ QYeysFOK4pIFISmg hNHkRyOOk3Yppitv Od | <p>Webhook コネクションをインテグレーションに追加するときに Fuse Online が提供する Webhook コネクショントークン。トークンは、URL を識別しにくくすることでセキュリティを提供する無作為の文字列です。これにより、該当の送信者以外がリクエストを送信できないようにします。</p> <p>リクエストでは、Fuse Online が提供するトークンを指定するか、独自のトークンを定義します。独自に定義する場合は、必ず簡単に推測できないものにしてください。</p> |

外部 URL が確認できたら、Fuse Online はインテグレーションの名前、OpenShift namespace の名前、および OpenShift DNS ドメインからホスト名を作成します。Fuse Online は使用できない文字を削除し、空白文字をハイフンに変換します。上記の外部 URL の例では、ホスト名は次のようになります。

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

curl を使用して Webhook を呼び出すには、コマンドを以下のように指定します。

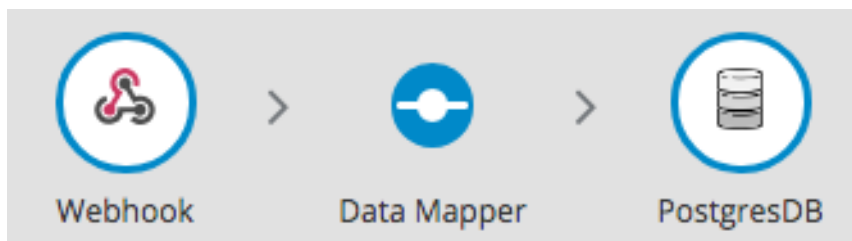
curl -H 'Content-Type: application/json' -d '{"todo": "from webhook"}' https://i-webhook-to-db-myproject.192.168.64.4.nip.io/webhook/bvGvQdpq308BcHIQQYeysFOK4pIFISmg hNHkRyOOk3YppitvOd

- **-H** オプションは HTTP **Content-Type** ヘッダーを指定します。
- **-d** オプションは、デフォルトで HTTP メソッドを **POST** に設定します。

このコマンドの実行により、インテグレーションがトリガーされます。データベースの最後のコネクションは新しいタスクをタスクテーブルに挿入します。これを確認するには、たとえば **https://todo-myproject.192.168.64.4.nip.io** で **https://todo-myproject.192.168.64.4.nip.io** アプリケーションを表示し、Update をクリックします。**from webhook** が新しいタスクとして表示されるはずです。

Webhook におけるクエリーパラメーターでの POST リクエストの例

この例では、前述の例と同じインテグレーションを使用します。



しかし、この例では以下のコンテンツを持つ JSON スキーマを指定して Webhook コネクションの出力データタイプを定義します。

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

```

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

```

この JSON スキーマでは以下を行います。

- **id** が **io.syndesis.webhook** に設定される必要があります。
- **parameters** セクションは HTTP クエリーパラメーターを指定する必要があります。
- **body** セクションはボディーのコンテンツを指定し、必要に応じて複雑に指定することができます。たとえば、入れ子のプロパティやアレイを定義できます。

これは、Webhook コネクターがインテグレーションの次のステップのコンテンツを準備するために必要な情報を提供します。

curl を使用して HTTP リクエストを送信するには、以下のようなコマンドを呼び出します。

```

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/ZYW7dV7k097vNsLX3YJ1GyxUFMFRteLpw0z4O69
MW7d2Kjg?source=web&status=new

```

Webhook コネクションがこのリクエストを受信すると、以下のような JSON インスタンスを作成します。

```

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

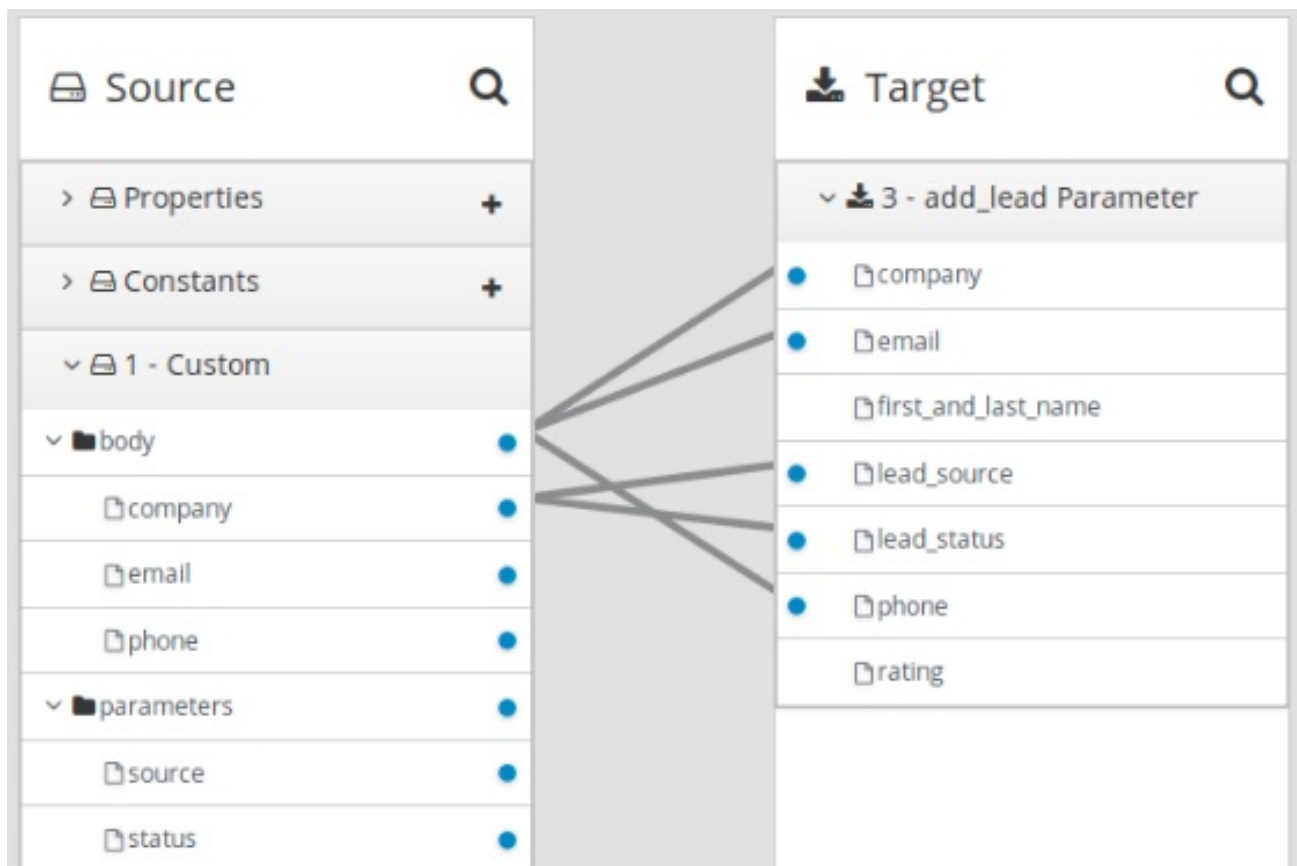
```

```

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

```

この JSON インスタンスが以下のマッピングを有効にします。



Webhook での GET の例

入力データを提供しない **GET** リクエストでインテグレーションをトリガーするには、Webhook コネクションの出力データシェイプを、定義 '{}' を持つ JSON インスタンスとして指定します。その後、クエリーパラメーターを指定しない以下の **curl** コマンドを呼び出すことができます。

```
curl 'https://i-webhook-params-to-db-myproject.192.168.42.235.nip.io/webhook/ZYWrrhaW7dVk097vNsLX3YJ1GyxUFMFRteLpw0z4O69MW7d2Kjg'
```

前述の **POST** の例を変更して、クエリーパラメーターがありボディーはない **GET** リクエストを送信します。Webhook コネクションの出力データシェイプを、以下の定義を持つ JSON スキーマとして指定します。

```

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

```

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

以下の **curl** コマンドは **GET** リクエストを送信します。

```
curl 'https://i-webhook-params-to-db-
myproject.192.168.42.235.nip.io/webhook/ZYWrhaW7dVk097vNsLX3YJ1GyxUFMFRteLpw0z4O69
MW7d2Kjg?source=web&status=new'
```