Red Hat Integration 2020-Q4

API プロバイダーインテグレーションの開発およびデプロイ

API プロバイダーインテグレーションの開発およびデプロイ

Last Updated: 2022-07-30

Red Hat Integration 2020-Q4 API プロバイダーインテグレーションの開発およびデプロイ

API プロバイダーインテグレーションの開発およびデプロイ

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律上の通知

概要

Red Hat Fuse Online、API Designer、および 3scale API Management での API プロバイダーインテグレーションの定義およびデプロイ

目次

前書き

Red Hat Fuse Online、API Designer、および 3scale API Management を使用して REST API を定義およびデプロイします。

第1章 API プロバイダーインテグレーションの開発およびデプロイの概要

アプリケーションプログラミングインターフェース(API)は、アプリケーションソフトウェアを構築するためのツール、定義、およびプロトコルのセットです。API により、製品またはサービスは、実装方法を把握せずに他の製品およびサービスと通信できます。

異なるアプリケーションとサービス間でデータを共有するビジネスユーザーは、Red Hat Integration 分散統合プラットフォームを使用して以下を行うことができます。

- API プロバイダーインテグレーションを開発し、REST API に接続します。
- REST API によって公開されるデータまたは機能を使用するアプリケーションを作成します。
- レガシーシステムと Things(ipmitool)のインターネットなど、製品およびサービスに接続します。
- セキュリティーと制御を維持しつつ、リソースへのアクセスを開きます。アクセスを開き、誰が自分のものを作る方法。

Red Hat Integration は、API 接続、データ変換、サービス構成、オーケストレーション、リアルタイムのメッセージング、クロスデータセンターメッセージストリーミング、および API 管理を提供し、ハイブリッドクラウドアーキテクチャー全体でアプリケーションを接続し、API 中心のビジネスサービスを可能にするために設計された、高速で柔軟な統合およびメッセージング技術のセットです。

Red Hat Integration には、以下の製品が含まれます。

Fuse Online

Red Hat Fuse は、分散型のクラウドネイティブインテグレーションプラットフォームです。Fuse Online は Red Hat の web ベースの Fuse ディストリビューションです。これは OpenShift Online Professional 層に事前インストールされています（また、オンプレミス OpenShift クラスターへのインストールにも使用できます）。Fuse Online は、コード開発を好むビジネスユーザー向けです。

API Designer

Red Hat は、この軽量バージョンの Apicurio(https://www.apicur.io/)を提供し、OpenAPI 形式で API 定義を作成できます。API プロバイダーインテグレーションの編集時に、API Designer は Fuse Online 内でアクセスできます。

3scale API Management

Red Hat 3scale API Management では、アクセスポリシーを設定したり、制御を一元化して、API に高可用性を提供することができます。

前提条件

以下の概念に関する作業知識が必要です。

- Fuse Online の概念
- REST API の概念
- 3scale API Management の概念

ステップの概要

1. Fuse Online で、API 呼び出しによってトリガーされるインテグレーションを作成します。
    a. REST API サービスとのインテグレーションを開始します。既存の REST API 定義を指定す

るか、または API デザイナーで新しい REST API 定義を作成します。
Fuse Online は、各 REST API オペレーションに対してインテグレーションフローと呼ばれる実行パスを作成します。

b. 各オペレーションのフローで、そのオペレーションを実行するコネクションおよびその他のステップを追加します。
各 REST API クライアント呼び出しはオペレーションを呼び出し、そのオペレーションを実行するインテグレーションフローのみの実行をトリガーします。

c. Fuse Online インテグレーションをパブリッシュします。これにより、OpenShift で API サービスとして利用できるようになります。

2. 3scale API Management で、公開された API サービスを検出します。その後、API のセキュリティーを確保し、アクセスポリシーを設定して起動することができます。

1.1. API プロバイダーインテグレーションを作成する利点、概要、およびワークフロー

API プロバイダーインテグレーションは、REST API サービスから開始します。この REST API サービスは、API プロバイダーインテグレーションの作成時に提供する OpenAPI 3 (または 2) ドキュメントによって定義されます。API プロバイダーインテグレーションをパブリッシュした後、Fuse Online は REST API サービスを OpenShift にデプロイします。API プロバイダーインテグレーションの利点は、REST API クライアントがインテグレーションの実行をトリガーする呼び出しを実行できることです。

複数の実行フロー

API プロバイダーインテグレーションには、フローと呼ばれる複数の実行パスがあります。OpenAPI ドキュメントが定義する各オペレーションには独自のフローがあります。Fuse Online では、OpenAPI ドキュメントが定義する各オペレーションに対して、コネクションおよびその他のステップをそのオペレーションの実行フローに追加します。これらのステップは、特定のオペレーションに必要なデータを処理します。

実行フローの例

たとえば、Fuse Online によって利用可能になった REST API サービスを呼び出す人事アプリケーションがあるとします。新しい従業員を追加する操作が呼び出されたとします。この呼び出しを処理する操作のフローは、以下のとおりです。

- 新入社員のハードウェアに関する経費報告書を作成するアプリケーションに接続します。
- 新しいハードウェアを設定するための社内チケットを追加する SQL データベースに接続します。
- 新入社員にオリエンテーションの情報を提供するメッセージを送信する Google メールに接続します。

実行をトリガーする方法

インテグレーションの実行をトリガーする REST API を呼び出す方法は複数あります。これには以下が含まれます。

- データ入力を取得し、呼び出しを生成する Web ブラウザーページ。
- **curl** ユーティリティーなどの REST API を明示的に呼び出すアプリケーション。
- REST API を呼び出す他の API (Webhook など)。

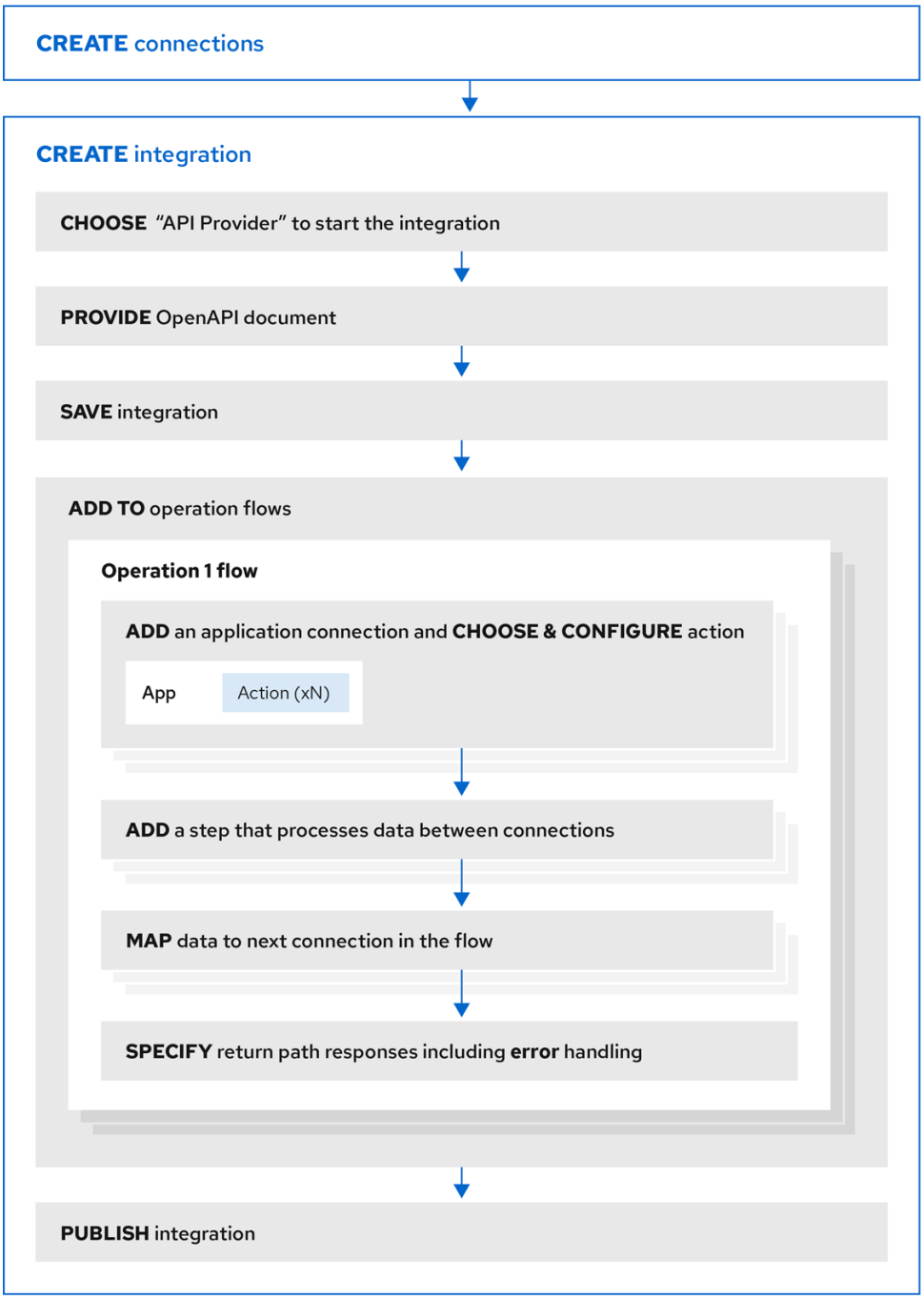
フローを編集する方法

以下を行って、各オペレーションのフローを編集できます。

- データを処理する必要があるアプリケーションにコネクションを追加します。
- 分割、集計、およびデータマッピングステップを含む、コネクション間のステップを追加します。
- コネクションエラーメッセージを、フローを終了する HTTP 応答の戻りコードにマッピングします。この応答は、インテグレーションの実行をトリガーした呼び出しを実行したアプリケーションに送信されます。

API プロバイダーインテグレーションを作成するためのワークフロー

API プロバイダーインテグレーションを作成するための **一般的な** ワークフローを以下の図に示します。



API プロバイダーインテグレーションのパブリッシュ

API プロバイダーインテグレーションをパブリッシュした後、Fuse Online はインテグレーションの summary ページに REST API サービスの外部 URL を表示します。この外部 URL は、クライアントが REST API サービスを呼び出すために使用するベース URL です。

OCP 上の Fuse Online 環境では、Red Hat 3scale の API プロバイダーインテグレーションの検出が有効になっている可能性があります。この場合、3scale ではサービスを呼び出すための URL が公開されます。

API プロバイダーインテグレーションのテスト

API プロバイダーインテグレーションのフローをテストするには、**curl** ユーティリティーを使用できます。たとえば、以下の **curl** コマンドは、REST API サービス URL https://i-task-api-proj319352.6a63.fuse-ignite.openshiftapps.com/api/ の **Get Task by ID** オペレーションに対し、フローの実行をトリガーします。

HTTP **GET** コマンドはデフォルトのリクエストであるため、**GET** を指定する必要はありません。URL の最後の部分は、取得するタスクの ID を指定します。

```
curl -k https://i-task-api-proj319352.6a63.fuse-ignite.openshiftapps.com/api/todo/1
```

1.2. OPENAPI オペレーションを API プロバイダーインテグレーションフローと関連させる方法

API プロバイダーインテグレーションの OpenAPI ドキュメントは、REST API クライアントが呼び出しできるオペレーションを定義します。各 OpenAPI オペレーションには、独自の API プロバイダーインテグレーションフローがあります。そのため、各オペレーションは独自の REST API サービス URL を持つこともできます。各 URL は API サービスのベース URL で定義され、任意でサブパスによって定義されます。REST API 呼び出しは、オペレーションの URL を指定し、そのオペレーションのフローの実行をトリガーします。

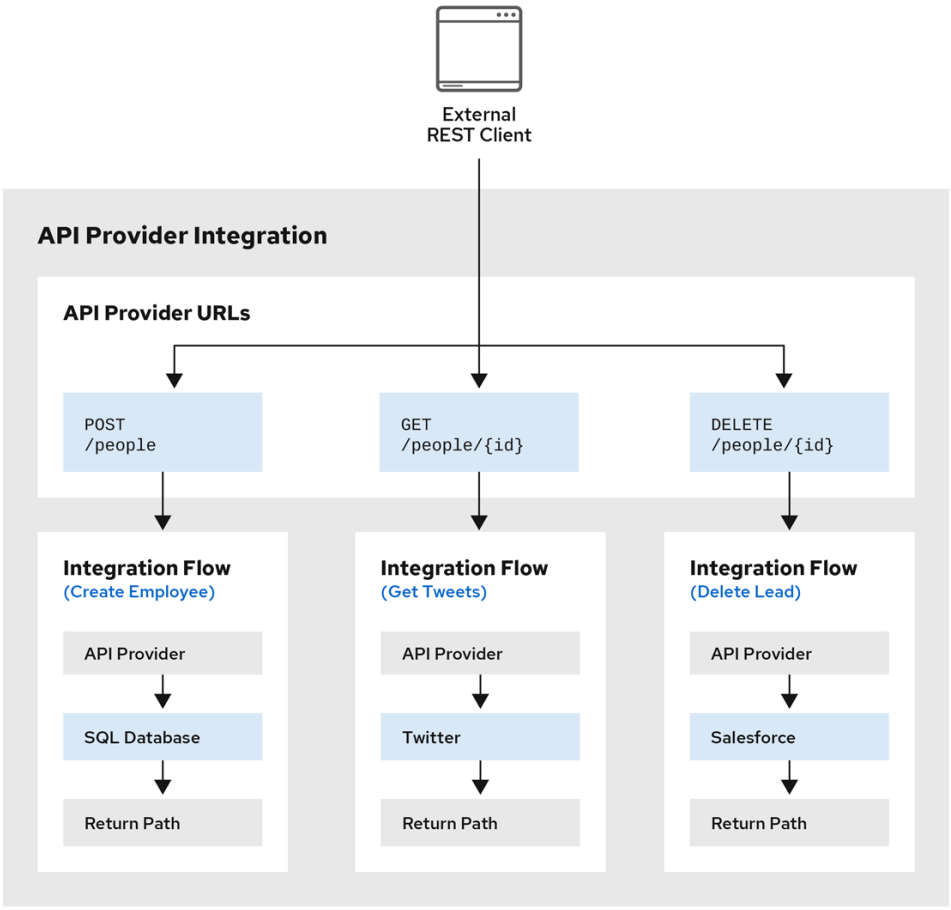
OpenAPI ドキュメントは、REST API サービス URL への呼び出しに指定できる HTTP 動詞 (**GET**、**POST**、**DELETE** など) を決定します。API プロバイダー URL への呼び出しの例は、「 API プロバイダークイックスタートの例を試す手順」を 参照してください。

OpenAPI ドキュメントで、オペレーションによって返信可能な HTTP ステータスコードも判断されます。オペレーションのリターンパスでは、OpenAPI ドキュメントで定義される応答のみが処理されます。たとえば、ID を基にしてオブジェクトが削除されるオペレーションでは、以下のような応答を定義できます。

```
"responses": {
  "204": {
          "description": "Task deleted"
        },
  "404": {
          "description": "No Record found with this ID"
        },
  "500": {
           "description": "Server Error"
       }
}
```

API プロバイダーインテグレーションの例

以下の図は、人に関するデータを処理する API プロバイダーインテグレーションを示しています。外部の REST API クライアントは、API プロバイダーインテグレーションによってデプロイされた REST API URL を呼び出します。URL が呼び出されると、1 つの REST オペレーションに対するフローの実行がトリガーされます。この API プロバイダーインテグレーションには 3 つのフローがあります。各フローは、Fuse Online で利用可能なすべてのコネクションまたはステップを使用できます。REST API とそのフローは、1 つの OpenShift Pod にデプロイされる Fuse Online API プロバイダーインテグレーションの 1 つです。



API プロバイダーインテグレーションの作成時における OpenAPI ドキュメントの編集

API プロバイダーインテグレーションの OpenAPI ドキュメントを指定した後、API オペレーションの実行フローを定義するときに必要に応じてドキュメントを更新できます。これには、API プロバイダーインテグレーションを編集するページの右上にある **View/Edit API Definition** をクリックします。これにより、API Designer エディターに OpenAPI ドキュメントが表示されます。ドキュメントを編集および保存し、Fuse Online で変更が反映されるようにします。

OpenAPI ドキュメントの編集時に以下を考慮します。

- **同期化の operationId プロパティー。**
  API Designer エディターの OpenAPI ドキュメントのバージョンと、Fuse Online インテグレーションエディターの OpenAPI ドキュメントのバージョンの同期は、ドキュメントに定義される各オペレーションに割り当てられた一意な **operationId** プロパティーに応じて行われます。特定の **operationId** プロパティー値を各オペレーションに割り当てるか、Fuse Online が自動生成する値を使用します。

- **リクエストと応答の定義**
  オペレーションのリクエストおよび応答を定義する JSON スキーマを各オペレーションの定義に提供できます。Fuse Online は以下のように JSON スキーマを使用します。

  - オペレーションの入力および出力データシェイプのベースとして使用します。

  - データマッパーでオペレーションフィールドを表示するために使用します。

- **循環スキーマ参照がない**
  API プロバイダーインテグレーションオペレーションの JSON スキーマにはスキーマの循環参照を含めることはできません。たとえば、リクエストまたは応答ボディーを指定する JSON スキーマは、そのスキーマ自体を全体的に参照することはできず、中間スキーマを介してそのス

キーマ自体を部分的に参照することもできません。

第2章 API の開発およびデプロイメント用の RED HAT INTEGRATION プロダクトの設定

オンプレミスの OpenShift Container Platform(OCP)の管理者は、API 開発およびデプロイメント用に Red Hat Integration 製品を設定する必要があります。

- 同じ OpenShift クラスターに Fuse Online と 3scale API Management をデプロイします（OpenShift クラスターの管理者権限が必要です）。
- 3scale で API の検出を有効にするように Fuse Online を設定します（Fuse Online を OpenShift プロジェクトにインストールする権限）。
- 3scale でサービス検出を設定します（3scale の管理者権限が必要です）。

さらに、以下の設定が必要です。

- Fuse Online のデフォルト動作では、API は 3scale の自動検出の対象になりません。Fuse Online をインストールする OpenShift プロジェクトの検出を有効にする必要があります。
- 3scale API Management の管理者は、サービス検出用に 3scale を設定する必要があります。たとえば、管理者はユーザーが検出可能なサービスを含むクラスタープロジェクトを表示するための適切なパーミッションを持っていることを確認する必要があります。

2.1. 3SCALE で API の検出を有効化する FUSE ONLINE の設定

API プロバイダーインテグレーションを作成する場合、Red Hat 3scale でそのインテグレーションの API の検出を有効することがあります。デフォルトの動作では、API は 3scale の自動検出の対象になりません。検出を有効にするには、3scale ユーザーインターフェースの URL を指定する必要があります。

Fuse Online をインストールする前に 3scale で API の検出を有効にするよう Fuse Online を設定するには、「Fuse Online の設定に使用するカスタムリソース属性の説明」を参照してください。

インストール後に **syndesis** カスタムリソースを更新すると、検出を有効にすることができます。本トピックでは、この手順を説明します。検出を有効にすると、リソースの更新時に接続されている OpenShift プロジェクトのみに適用されます。

3scale サービス検出を有効にすると、以下のようになります。

- デフォルトの動作として、3scale は API プロバイダーインテグレーションをパブリッシュします。3scale が API プロバイダーインテグレーションをパブリッシュする場合は、以下のとおりです。
  - Fuse Online は、稼働している API プロバイダーインテグレーションの外部 URL を提供しません。
  - API には 3scale からのみアクセスできます。エンドポイントを公開するために 3scale の設定が必要です。詳細は、Red Hat 3scale API Management『Admin Portal Guide』の「Service Discovery」を参照してください。
- API プロバイダーインテグレーションの作成者は、そのインテグレーションの 3scale 検出を無効にすることができます。つまり、各 API プロバイダーインテグレーション作成者は、そのインテグレーションの API を検出可能にするかどうかを選択できます。

前提条件

- オンサイトで OCP に Fuse Online がインストールされている必要があります。
- **oc** クライアントツールがインストール済みであり、Fuse Online がインストールされている OCP クラスターに接続されている必要があります。
- API の検出を有効にするプロジェクトで Fuse Online をインストールするパーミッションが必要です。
- OpenShift クラスター上の 3scale プロジェクトの名前を把握している必要があります。

手順

1. Fuse Online をインストールする権限を持つアカウントで OpenShift にログインします。以下に例を示します。
   oc login -u developer -p developer
2. Fuse Online が稼働している OpenShift プロジェクトに切り替えます。このプロジェクトのみで検出を有効にします。以下に例を示します。
   oc project my-fuse-online-project
3. 3scale プロジェクトによる Fuse Online インテグレーションの表示を可能にするパーミッションを追加します。
   oc adm policy add-cluster-role-to-user view system:serviceaccount:<3scale-project>:amp

   たとえば、OpenShift の 3scale プロジェクトの名前が **my3scale** が場合は、次のようになります。

   oc adm policy add-cluster-role-to-user view system:serviceaccount:my3scale:amp
4. **syndesis** カスタムリソースを編集します。
   a. 以下のコマンドを実行します。通常、エディターでリソースが開かれます。
      oc edit syndesis
   b. 3scale ユーザーインターフェースの URL に **managementUrlFor3scale** を設定し、リソースを編集します。結果は以下のようになります。

```
spec:
  components:
    server:
      features:
        managementUrlFor3scale: https://3scale-admin.apps.mycluster.com
```

   c. リソースを保存します。
5. 任意手順:切り替え先のプロジェクトで検出が有効になっていることを確認するには、以下のコマンドを実行します。
   oc describe dc/syndesis-server

   検出が有効な場合、**OPENSHIFT_MANAGEMENT_URL_FOR3SCALE** 環境変数がカスタムリソースで指定した URL に設定されていることがこのコマンドで出力されます。

結果

syndesis カスタムリソースに対するこの変更により、Fuse Online のインストールを担う **syndesis-operator** が **syndesis-server** を再デプロイするようトリガーされます。切り替え先の OpenShift プロジェクトでは、新しいデフォルト動作として API が 3scale で検出の対象となります。

syndesis-server DeploymentConfig オブジェクトを編集して **OPENSHIFT_MANAGEMENT_URL_FOR3SCALE** 環境変数を設定しないでください。**syndesis-operator** により変更が元に戻るため、これは動作しません。**syndesis-operator** は、常に **syndesis** カスタムリソースのみに従って Fuse Online がデプロイされるようにします。

2.2. サービスディスカバリーを有効にするための OPENSHIFT 設定

3scale の管理者は、Open Authorization (OAuth) サーバーの使用/不使用とは独立に、サービスディスカバリーを設定することができます。

OAuth サーバーと共に 3scale サービスディスカバリーを設定する場合、ユーザーが 3scale にサインインした際のフローは以下のとおりです。

- ユーザーが OAuth サーバーにリダイレクトされる。
- ユーザーがまだ OAuth サーバーにログインしていなければ、ログインを求められる。
- ユーザーが 3scale のサービスディスカバリーと SSO の組み合わせを初めて実装する場合には、OAuth サーバーは必要なアクションを実行するために承認を要求する。
- ユーザーが再度 3scale にリダイレクトされる。

OAuth サーバー と組み合わせてサービスディスカバリーを設定する場合には、以下のオプションがあります。

- OpenShift OAuth サーバーと組み合わせてサービスディスカバリーを設定する
- RH-SSO サーバー (Keycloak) と組み合わせてサービスディスカバリーの設定する

OAuth サーバーと組み合わせずにサービスディスカバリーを設定する 場合、ユーザーが 3scale にサインインしてもリダイレクトされません。その代わりに、サービスディスカバリーのために、3scale Single Service Account がクラスターに対してシームレスに認証を行います。3scale を通じて API サービスを検出する間、すべての 3scale テナント管理ユーザーはクラスターに対して同じアクセスレベルを持ちます。

2.2.1. OpenShift OAuth サーバーと組み合わせたサービスディスカバリーの設定

3scale のシステム管理者は、ユーザーが OpenShift の組み込み OAuth サーバーを使用して、個別に認証、3scale を承認して API を検出するのを許可することができます。

前提条件

- OpenShift クラスター（バージョン 3.11 以降）に 3scale 1.2 をデプロイする必要があります。
- 3scale を OpenShift にデプロイするには、3scale-amp-openshift-templates を使用する必要があります。
- 3scale でサービスディスカバリーを使用する 3scale ユーザーは、OpenShift クラスターにアクセスできる必要があります。

手順

1. 3scale 用の OpenShift OAuth クライアントを作成します。詳細は、OpenShift の認証に関するドキュメント を参照してください。以下の例で、**<provide-a-client-secret>** を生成するシークレットに、**<3scale-master-domain-route>** を 3scale マスター管理ポータルにアクセスするための URL に、それぞれ置き換えます。

```
$ oc project default
$ cat <<-EOF | oc create -f -
kind: OAuthClient
apiVersion: v1
metadata:
 name: 3scale
secret: "<provide-a-client-secret>"
redirectURIs:
 - "<3scale-master-domain-route>"
grantMethod: prompt
EOF
```

2. 3scale サービスディスカバリーの設定ファイルを開きます。

```
$ oc project <3scale-project>
$ oc edit configmap system
```

3. 以下のように設定します。

```
service_discovery.yml:
  production:
    enabled: true
    authentication_method: oauth
    oauth_server_type: builtin
    client_id: '3scale'
    client_secret: '<choose-a-client-secret>'
```

4. ユーザーが適切なアクセス権限を持ち、検出可能なサービスが含まれるクラスタープロジェクトを表示できるようにしてください。
<user> で表される管理ユーザーに、検出されるサービスが含まれる **<namespace>** プロジェクトを表示する権限を付与するには、以下のコマンドを使用します。

```
oc adm policy add-role-to-user view <user> -n <namespace>
```

5. **configmap** を変更したら、**system-app** および **system-sidekiq** Pod を再デプロイし、変更を適用する必要があります。

```
oc rollout latest dc/system-app
oc rollout latest dc/system-sidekiq
```

6. ロールアウトのステータスを表示し、読み込みが完了したことを確認します。

```
oc rollout status dc/system-app
oc rollout status dc/system-sidekiq
```

補足説明

OpenShift OAuth トークンについての詳細は、『認証』の「内部 OAuth サーバーの設定」を参照してください。

2.2.2. RH-SSO サーバー (Keycloak) と組み合わせたサービスディスカバリーの設定

システム管理者は、ユーザーが OpenShift 向け Red Hat Single Sign-On を使用して、個別に認証、3scale を承認してサービスを検出するのを許可することができます。

OpenShift の承認ゲートウェイとして RH-SSO デプロイメントを使用する OpenShift の設定例については、この ワークフロー を参照してください。

前提条件

- OpenShift クラスター（バージョン 3.11 以降）に 3scale 1.2 をデプロイする必要があります。
- 3scale を OpenShift にデプロイするには、3scale-amp-openshift-templates を使用する必要があります。
- 3scale でサービスディスカバリーを使用する 3scale ユーザーは、OpenShift クラスターにアクセスできる必要があります。

手順

1. Red Hat OAuth サーバー (Keycloak) に、3scale 用の OAuth クライアントを作成します。

注記

クライアント設定で、OpenShift がアカウントをリンクすることができるように、**username** が **preferred_username** にマッピングされていることを確認します。

2. 3scale サービスディスカバリーの設定を編集します。

```
$ oc project <3scale-project>
$ oc edit configmap system
```

3. 以下のように設定されていることを確認します。ここで、**<the-client-secret-from-Keycloak>** は、OAuth クライアントの作成時に Keycloak が自動生成した値です。

```
service_discovery.yml:
  production:
    enabled: true
    authentication_method: oauth
    oauth_server_type: rh_sso
    client_id: '3scale'
    client_secret: '<the-client-secret-from-Keycloak>'
```

4. ユーザーが適切なアクセス権限を持ち、検出可能なサービスが含まれるクラスタープロジェクトを表示できるようにしてください。
たとえば、**<user>** に **<namespace>** プロジェクトを表示する権限を付与するには、以下のコマンドを使用します。

```
oc adm policy add-role-to-user view <user> -n <namespace>
```

5. **configmap** を変更したら、**system-app** および **system-sidekiq** Pod を再デプロイし、変更を適用する必要があります。

補足説明:

- トークンの有効期限:『Keycloak Server Administration Guide』の「Session and Token Timeouts」に記載されるように、デフォルトではセッショントークンの有効期限は 1 分です。ただし、タイムアウトを妥当な値 (たとえば 1 日) に設定することを推奨します。

2.2.3. OAuth サーバーと組み合わせないサービスディスカバリーの設定

OAuth サーバーを使用しない状況で 3scale のサービスディスカバリーを設定する場合には、3scale Single Service Account を使用して OpenShift API サービスに対する認証を行うことができます。

前提条件

- OpenShift クラスター（バージョン 3.11 以降）に 3scale 1.2 をデプロイする必要があります。
- 3scale を OpenShift にデプロイするには、3scale-amp-openshift-templates を使用する必要があります。
- 3scale でサービスディスカバリーを使用する 3scale ユーザーは、OpenShift クラスターにアクセスできる必要があります。

手順

1. 3scale プロジェクトをアクティブなプロジェクトにします。

```
$ oc project <3scale-project>
```

2. エディターで 3scale サービスディスカバリーの設定を開きます。

```
$ oc edit configmap system
```

3. 以下のように設定されていることを確認します。

```
service_discovery.yml:
  production:
    enabled: <%= cluster_token_file_exists = File.exists?(cluster_token_file_path =
'/var/run/secrets/kubernetes.io/serviceaccount/token') %>
    bearer_token: "<%= File.read(cluster_token_file_path) if cluster_token_file_exists %>"
    authentication_method: service_account
```

4. 以下のいずれかのオプションに従って、3scale デプロイメントの **amp** サービスアカウントに、検出可能なサービスが含まれるプロジェクトを表示するための適切な権限を付与します。
   - Fuse Online プロジェクトに切り替え、**view** cluster level パーミッションを持つ 3scale デプロイメントの **amp** サービスアカウントを付与します。

```
oc project <fuseonline project>

oc adm policy add-cluster-role-to-user view system:serviceaccount:<3scale-project>:amp
```

   - 『Developer Guide』の「Service Accounts」で説明するように、より厳しいポリシーを適用する。

2.3. OPENSHIFT プロジェクトへの 3SCALE アクセスの承認

OAuth トークンが有効ではない場合、OpenShift プロジェクトの管理者は 3scale ユーザーが namespace にアクセスするのを承認することができます。

前提条件

- OpenShift プロジェクトの管理者としてのクレデンシャルが設定されている。
- OpenShift の管理者が、OpenShift クラスターにサービスディスカバリーを設定している。たとえば、Fuse Online API の場合、OpenShift 管理者は Fuse Online **Syndesis CRD** ファイルの **managementUrlFor3scale** 設定を 3scale 管理ポータルの URL に設定し、3scale プロジェクトによる Fuse Online インテグレーションの表示を可能にするパーミッションを追加する必要があります。
- 「サービスディスカバリーについて」 で説明するように、3scale の管理者が 3scale デプロイメントでサービスディスカバリーを設定している。
- API サービスの名前およびその OpenShift プロジェクトの namespace を把握している。
- API サービスが、3scale をインストールしたクラスターと同じ OpenShift クラスターにデプロイされている。
- 「サービスディスカバリーについて」 で説明するように、サービスディスカバリーを有効にする正しいアノテーションが API に付けられている。

手順

1. **Authenticate to enable this option** のリンクをクリックします。
2. namespace 管理者のクレデンシャルを使用して OpenShift にログインします。
3. **Allow selected permissions** をクリックし、3scale ユーザーのアクセスを承認します。

次のステップ

API 管理の詳細については、Red Hat 3scale API Management のドキュメント を参照してください。

第3章 API 呼び出しによってトリガーされる FUSE ONLINE インテグレーションの作成

必要時にインテグレーションの実行をトリガーするには、指定した REST API サービスでインテグレーションを開始します。この方法で開始するインテグレーションは、**API プロバイダーインテグレーション** と呼ばれます。API プロバイダーインテグレーションでは、REST API クライアントはインテグレーションの実行をトリガーするコマンドを呼び出しできます。

Fuse Online が API プロバイダーインテグレーションをパブリッシュすると、インテグレーションエンドポイントにネットワークアクセスできるクライアントはすべてインテグレーションの実行をトリガーできます。

デフォルトでは、Fuse Online は 3scale で使用する API プロバイダーインテグレーションの API サービス定義にアノテーションを付けます。

API プロバイダーインテグレーションを作成するための情報および手順は以下を参照してください。

- 「API プロバイダーインテグレーションの作成」
- 「API プロバイダーインテグレーションのオペレーションフローの定義」

3.1. API プロバイダーインテグレーションの作成

API プロバイダーインテグレーションを作成するには、インテグレーションが実行できるオペレーションを定義する OpenAPI ドキュメント (**.json**、**9.yaml**、または **.yml** ファイル) を提供します。Fuse Online は各オペレーションの実行フローを作成します。各オペレーションのフローを編集し、そのオペレーションの要件に応じてインテグレーションデータを処理するコネクションおよびステップを追加します。

前提条件

- インテグレーションが実行する REST API オペレーションの OpenAPI ドキュメントを提供または定義できる必要があります。
  検証するには、API プロバイダークイックスタート の OpenAPI ドキュメントである raw の **task-api.json** ファイルをダウンロード します。Fuse Online が OpenAPI ドキュメントの提供を要求したときに、このファイルをアップロードできます。この代わりに、raw の **task-api.json** ファイルである https://raw.githubusercontent.com/syndesisio/syndesis-quickstarts/1.11/api-provider/task-api.json を指定できます。
- 各 OpenAPI オペレーションのフローが計画されている必要があります。
- オペレーションのフローを追加する各アプリケーションまたはサービスの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 の左ナビゲーションパネルで **Integrations** をクリックします。
2. **Create Integration** をクリックします。
3. **Choose a connection** ページで **API Provider** をクリックします。
4. **Start integration with an API call** ページで以下を行います。

- REST API オペレーションを定義する OpenAPI ドキュメントがある場合は、OpenAPI ドキュメントをアップロードします。
- OpenAPI ドキュメントを定義する必要がある場合は、**Create a new OpenAPI 3.x document** または **Create a new OpenAPI 2.x document**を選択します。

5. **Next** をクリックします。
   - ドキュメントをアップロードした場合は、これを確認または編集します。
     a. **Review/Edit** をクリックして API Designer エディターを開きます。
     b. 必要に応じて確認や編集を行います。
     任意の手順: ドキュメントによって OpenAPI 2 仕様が使用される場合に、API Designer でドキュメントを変換して OpenAPI 3 仕様に準拠するようにするには、**Convert to OpenAPI 3** をクリックします。
     c. 右上の **Save** または **Cancel** をクリックし、エディターを閉じます。
     d. **Next** をクリックします。
   - ドキュメントを作成する場合は、Fuse Online で起動される API Designer エディターで以下を行います。
     a. 「API Designer での REST API 定義の作成」の説明どおりに OpenAPI ドキュメントを定義 します。
     b. 右上の **Save** をクリックし、エディターを閉じます。
     c. **Next** をクリックします。

結果

Fuse Online は OpenAPI ドキュメントが定義するオペレーションの一覧を表示します。

次のステップ

各オペレーションに対して、その オペレーションを実行するフローを定義 します。

3.1.1. API Designer での REST API 定義の作成

以下の手順では、REST API 定義を作成する方法を説明します。

サンプルについて

Task Management API のサンプルは、営業コンサルタントがお客様側の担当者と対話する際に必要なタスクを追跡するために使用する可能性のある単純な API をシミュレートします。「to-do」タスクの例には「create an account for a new contact (新規の担当者のアカウントの作成)」や「place an order for an existing contact (既存の担当者の注文処理)」が含まれる可能性があります。Task Management API のサンプルを実装するには、複数のタスク用のパスと、特定のタスク用のパスの 2 つのパスを作成します。その後、タスクを作成し、すべてのタスクまたは特定のタスクを取得し、タスクを更新してタスクを削除する操作を定義します。

前提条件

- 作成する必要のある API のエンドポイントを把握している必要があります。Task Management API のサンプルの場合、**/todo** および **/todo/{id}** の 2 つのエンドポイントがあります。

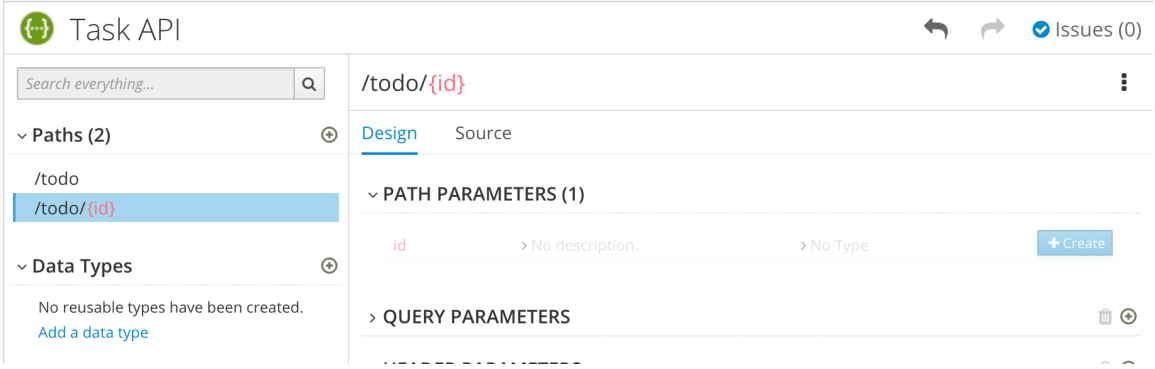
手順

1. **New API** をクリックします。新規の API ページが開きます。
デフォルトでは、API Designer は OpenAPI 3 仕様を使用します。OpenAPI 2 仕様を使用する場合は、**New API** ボタンの横にある矢印をクリックし、**OpenAPI 2** を選択します。

注記

OpenAPI 2 仕様に基づいて API を開く場合、API Designer の **Convert to OpenAPI 3** オプションを使用して、API が OpenAPI 3 仕様に準拠するよう変換することができます。

2. API 名を変更するには、以下を実行します。
   a. 名前にカーソルを合わせ、表示される編集アイコン () をクリックします。
   b. 名前を編集します。たとえば、**Task API** を入力します。
   c. チェックマークアイコンをクリックして、名前変更を確認します。
3. 任意で以下を行います。
   - バージョン番号と説明を提供します。
   - 連絡先情報 (名前、メールアドレス、および URL) を追加します。
   - ライセンスを選択します。
   - タグを定義します。
   - サーバーを 1 つ以上定義します。
   - セキュリティースキームを設定します。
   - セキュリティー要件を指定します。
4. API のそれぞれのエンドポイントへの相対パスを定義します。フィールド名はスラッシュ (/) で開始する必要があります。
Task Management API のサンプルでは、2 つのパスを作成します。
   - 複数のタスクのパス: **/todo**
   - ID 別の特定タスクのパス: **/todo/{id}**



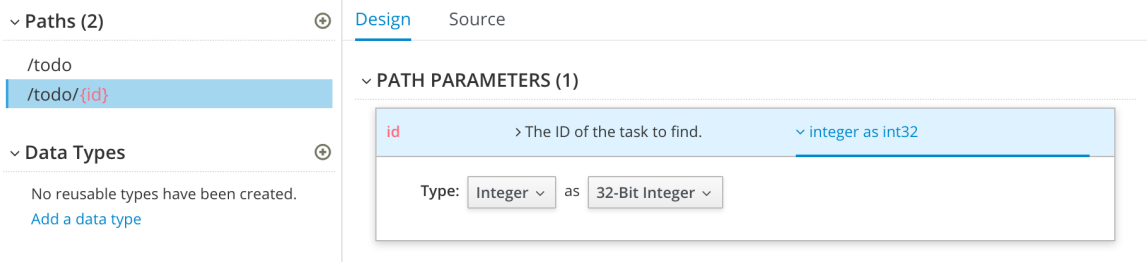
5. 任意のパスパラメーターのタイプを指定します。
id パラメーターのサンプルは、以下のようになります。

a. **Paths** リストで、**/todo/{id}** をクリックします。
id パラメーターが **PATH PARAMETERS** セクションに表示されます。

b. **Create** をクリックします。

c. description (説明) には、**The ID of the task to find.**と入力します。

d. 「type (タイプ)」については、**integer** を **32-Bit integer** として選択します。



6. **Data Types** セクションで、API の再利用可能なタイプを定義します。

a. **Add a data type**をクリックします。

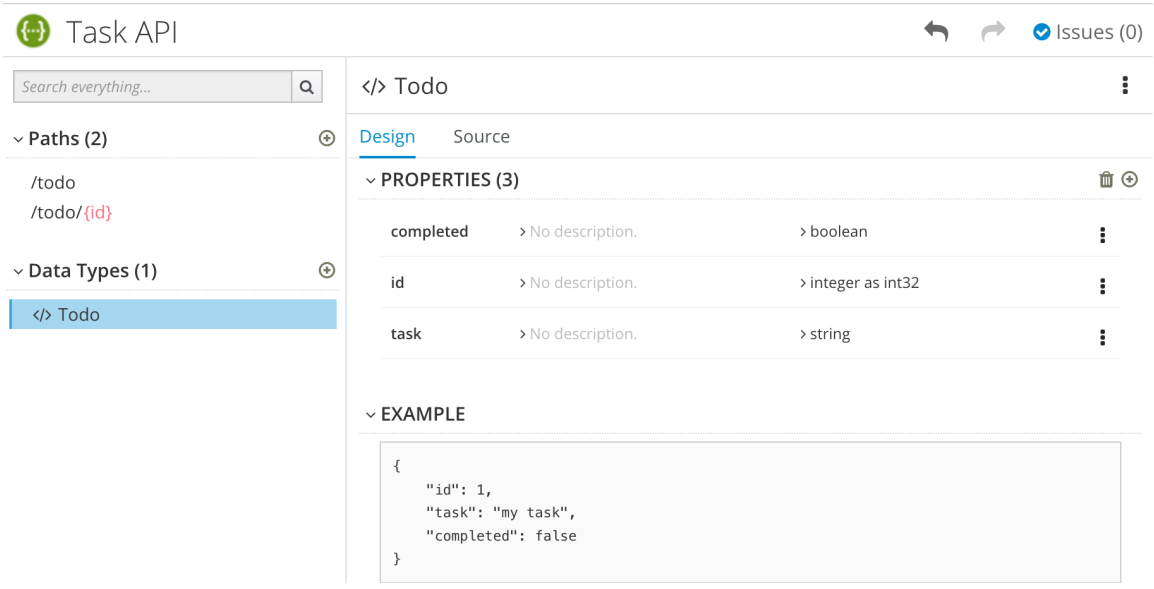
b. **Add Data Type** ダイアログで名前を入力します。Task Management API のサンプルでは、**Todo** と入力します。

c. 任意で、API Designer がデータタイプのスキーマの作成に使用するサンプルを指定できます。生成されるスキーマを編集できます。
Task Management API のサンプルでは、以下の JSON 例で開始します。

```
{
    "id": 1,
    "task": "my task",
    "completed": false
}
```

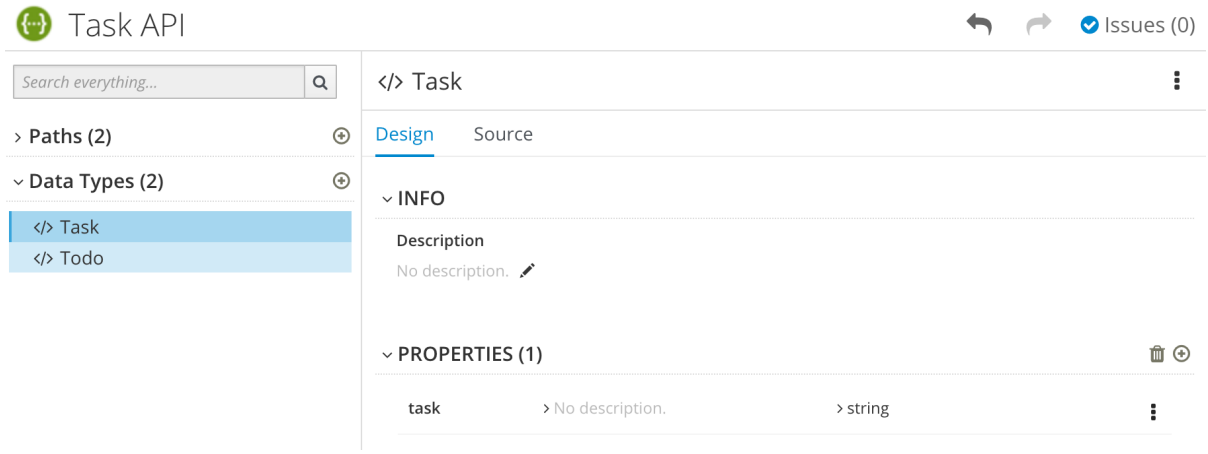
d. オプションで、該当するデータタイプを指定して REST リソースの作成を選択できます。

e. **Save** をクリックします。サンプルを指定している場合、API Designer はそのサンプルからスキーマを生成します。



7. オプションで、スキーマのプロパティーを編集し、新しいプロパティーを追加できます。

8. Task Management API のサンプルの場合、タイプ **string** の **task** という名前の 1 つのプロパティーを使用して、 **Task** という名前の別のデータタイプを作成します。



9. それぞれのパスについて、操作 (GET、PUT、POST、DELETE、 OPTIONS、HEAD、または PATCH) を定義します。
Task Management API のサンプルの場合、以下の表で説明されているように操作を定義します。

表3.1 Task Management API 操作

パス	操作	説明	リクエストボディー	応答
/todo	**POST**	新しいタスクを作成します。	メディアタイプ: **application/json** データタイプ: Task	• ステータスコード: 201 説明: **Task created** 応答ボディー: メディアタイプ: **application/json** データタイプ: **Todo**
/todo	**GET**	すべてのタスクを取得します。	**該当なし**	• ステータスコード: 200 説明: **Task deleted** • ステータスコード: 400 説明: **Task not deleted**

パス	操作	説明	リクエストボディー	応答
/todo/{id}	**GET**	ID 別にタスクを取得します。	**該当なし**	• ステータスコード: 200 説明: **Task found for ID** 応答ボディー: メディアタイプ: **application/json** データタイプ: **Task** • ステータスコード: 404 説明: **No task with provided identifier found.**
/todo/{id}	**UPDATE**	ID 別にタスクを更新します。	リクエストボディーのタイプ: **Task**	• ステータスコード: 200 説明: **Completed** • ステータスコード: 400 説明: **Task not updated**
/todo/{id}	**DELETE**	ID 別にタスクを削除します。	**該当なし**	• ステータスコード: 200 説明: **Task deleted** • ステータスコード: 400 説明: **Task not deleted**

10. 「API Designer での検証問題の解決」の説明どおりに問題 を解決します。

関連情報

- OpenAPI 仕様の詳細は、「https://github.com/OAI/OpenAPI-Specification/blob/master/versions/2.0.md」を参照してください。

3.1.2. API Designer での検証問題の解決

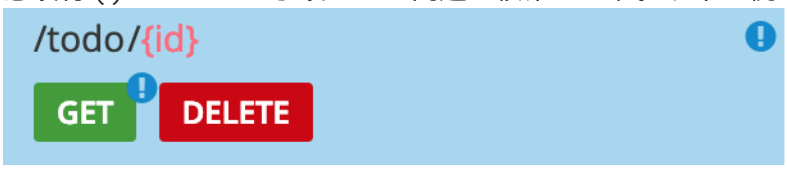
API の作成および編集時に、API Designer は感嘆符 (!) アイコンと、API Designer タイトルバーの問題のリストを使って解決する必要のある問題を特定します。

前提条件

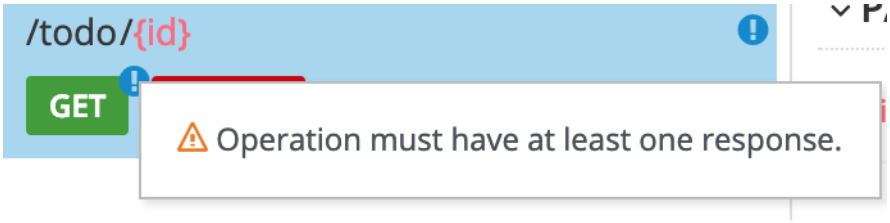
- API Designer で API を開いている必要があります。

手順

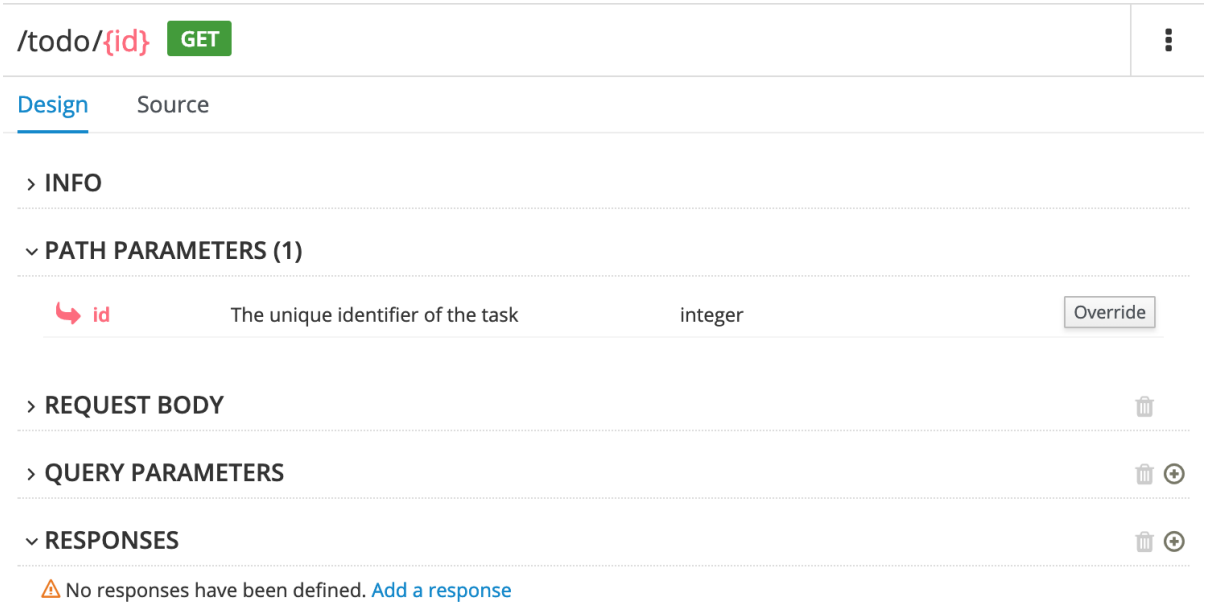
1. 感嘆符 (!) アイコンで示唆される問題を検索します。以下に例を示します。



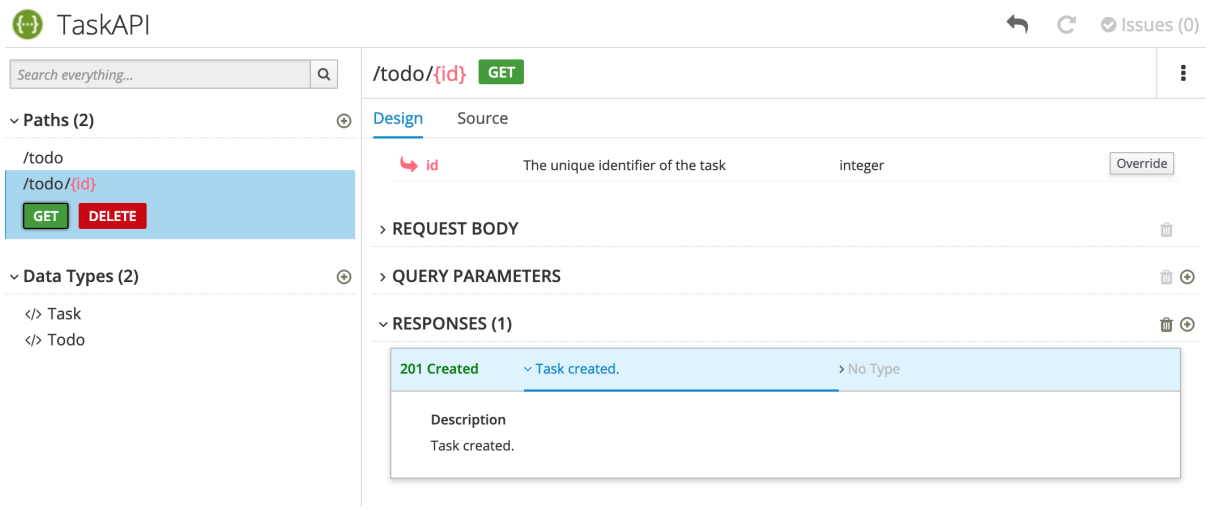
2. 感嘆符アイコンをクリックして、問題の説明を表示します。以下に例を示します。



3. 問題の説明で提供される情報に基づいて、問題のある場所に移動し、修正します。
たとえば、「Operation must have at least one response」(オペレーションには 1 つ以上の応答が必要) の問題を修正するには、**GET** オペレーションをクリックして開き、**Add a response** をクリックします。



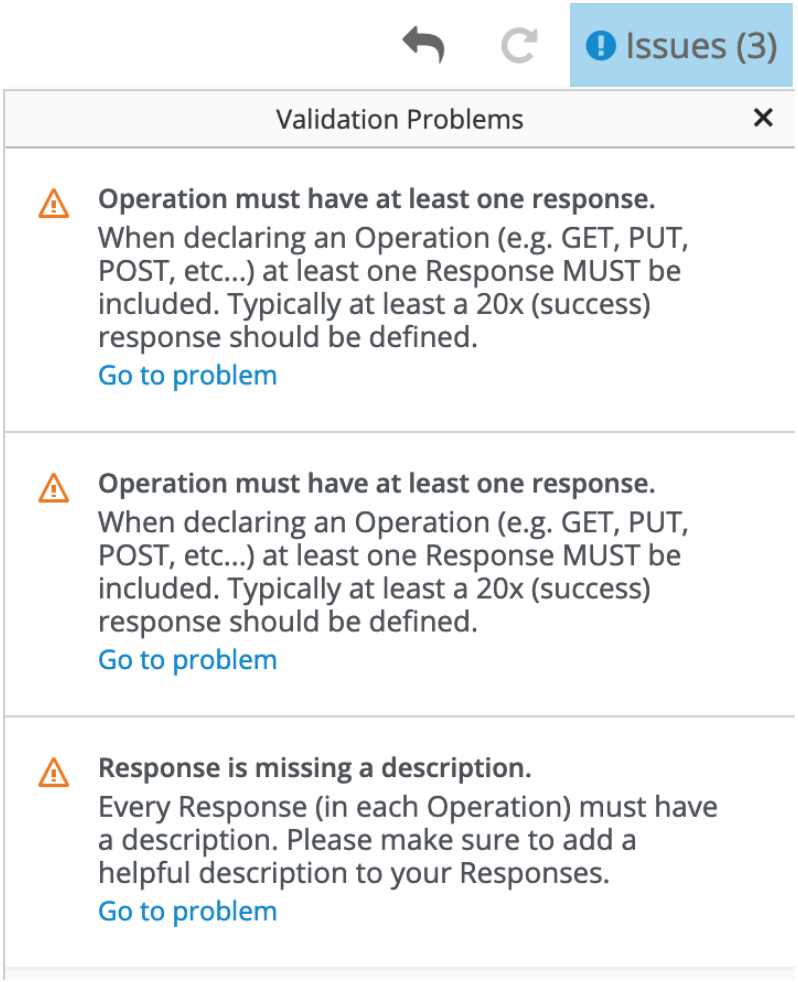
応答の説明の入力後、問題は解決され、感嘆符アイコンは表示されなくなります。



4. すべての問題の概要については、以下のようになります。

   a. 右上にある **Issues** リンクをクリックします。

   b. 特定の問題の **Go to a problem**をクリックし、その問題を解決するために問題のある場所に移動します。



3.2. API プロバイダーインテグレーションのオペレーションフローの定義

REST API サービスを定義する OpenAPI ドキュメントは、サービスが実行できるオペレーションを定義します。API プロバイダーインテグレーションの作成後、各オペレーションのフローを編集できます。

各オペレーションには必ず 1 つのフローがあります。オペレーションフローでは、コネクションを他のアプリケーションやサービスに追加でき、コネクション間のデータで操作するステップも追加できます。

オペレーションフローへ追加すると、API プロバイダーインテグレーションがベースとする OpenAPI ドキュメントの更新が必要であることがあります。これには、API プロバイダーインテグレーションを編集するページの右上にある **View/Edit API Definition** をクリックします。これにより、API Designer エディターにドキュメントが表示されます。OpenAPI 定義では、各オペレーションに固有の **operationId** プロパティーがある限り、API Designer に更新を保存することができ、Fuse Online は API プロバイダーインテグレーションのフロー定義を同期して更新が反映されるようにすることができます。

前提条件

- API プロバイダーインテグレーションを作成し、名前を付け、保存している必要があります。
- オペレーションフローが接続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が作成済みである必要があります。詳細は、コネクションの 作成に関する情報 を参照してください。
- Fuse Online は API が定義するオペレーションのリストを表示します。

手順

1. **Operations** リストページで、定義するフローのオペレーションに対して **Create flow** をクリックします。
2. このフローに追加する各コネクションに対して以下を行います。
   a. フロービジュアライゼーションで、プラス記号をクリックしてその場所にコネクションを追加します。
   b. 使用するコネクションをクリックします。
   c. コネクションが実行するアクションを選択します。
   d. ラベルが付いたフィールドにデータを入力して、アクションを設定します。
   e. **Next** をクリックします。

   フローに必要なコネクションをすべて追加してから、続行します。
3. このオペレーションフローでコネクション間のデータを処理するには、以下を行います。
   a. フロービジュアライゼーションで、ステップを追加する場所にあるプラス記号をクリックします。
   b. 追加するステップをクリックします。
   c. ラベル付が付いたフィールドにデータを入力して、ステップを設定します。
   d. **Next** をクリックします。
   ヘルプは「コネクション 間のステップの追加」を参照して ください。

   コネクションの間のデータを処理する別のステップを追加する場合は、この手順のサブセットを繰り返します。
4. データを次のコネクションのフィールドにマップします。

a. フロービジュアライゼーションで、コネクションが受信データを処理できないことを示す、データタイプ不一致の  アイコンを確認します。ここでは、データマッパーステップを追加する必要があります。

b. フロービジュアライゼーションの各データ不一致アイコンに対して以下を行います。

i. そのステップの直前にあるプラスマークをクリックします。

ii. **Data Mapper** をクリックします。

iii. 必要なマッピングを定義します。ヘルプは「インテグレーションデータを次のコネクションのフィールドにマッピング」を 参照してください。

iv. **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。

5. フロービジュアライゼーションの、**Provided API Return Path**ステップで **Configure** をクリックします。
すべての API プロバイダーインテグレーションは、オペレーションフローの実行をトリガーした REST API の呼び出し元に応答を送信することで、各オペレーションフローを終了します。応答には、オペレーションのフローを終了する **Provided API Return Path**ステップのに設定した戻りコードの 1 つが含まれます。以下のように、Return Path ステップを設定します。

a. **Return Code** フィールドの **Default Response** で、Fuse Online に表示されるデフォルトの応答を指定するか、下向きのキャレットをクリックしてスクロールし、希望のデフォルト応答を選択します。オペレーションフローを実行しても設定されたエラー応答がどれも返されなかった場合、フローはこの応答を送信します。通常、デフォルト応答の戻りコードはオペレーションに成功したことを意味します。

b. 返されたメッセージのボディー部にエラーメッセージが含まれるようにするかどうかを **Error Handling** に指定します。
通常、開発中はエラーメッセージを返すようにします。しかし、実稼働では機密情報が含まれる場合にエラーメッセージを非表示にする場合があります。エラーメッセージは、**responseCode**、**category**、**message**、および **error** 要素が含まれる JSON 形式の文字列です。例を以下に示します。

```
{
  responseCode: 404,
  category:  "ENTITY_NOT_FOUND_ERROR",
  message: "SQL SELECT did not SELECT any records"
  error: SYNDESIS_CONNECTION_ERROR
 }
```

開発中、最も確実にエラーの発生を確認する方法は、呼び出し元への応答の **HTTP_RESPONSE STATUS** ヘッダーをチェックすることです。インテグレーション Pod のログで **INFO** メッセージを確認することもできます。インテグレーションの **Activity** ログには成功した交換が記録され、エラーは **Activity** ログに常に記録されるとは限りません。

c. **Error Response Codes** には、フローのコネクションが返す可能性がある各エラーのエントリーが表示されます。エラーごとに、デフォルトの戻りコードである **200 All is good** を指定するか、クリックして別の HTTP ステータスの戻りコードを選択します。
選択可能な戻りコードは、このフローで実行されるオペレーションのために OpenAPI ドキュメントで定義される戻りコードです。必要な戻りコードが Fuse Online に表示されない場合、OpenAPI ドキュメントを編集して追加できます。

これを行うには、右上の **View/Edit API Definition**をクリックします。必要に応じて

OpenAPI ドキュメントを編集します。編集し終わったら、OpenAPI ドキュメントを保存します。Fuse Online が **Provided API Return Path**の編集に戻り、保存した変更が反映されます。

d. Next をクリックし、リターンパスの設定を完了します。

6. このフローに、必要なコネクションとステップがすべてあり、データの不一致アイコンがない場合や、現時点でフローを編集しない場合は、以下の 1 つを行います。

- **Publish**: インテグレーションの実行を開始するには、右上の **Publish** をクリックします。これにより、インテグレーションがビルドされ、REST API サービスが OpenShift にデプロイされます。さらにインテグレーションが実行できるようになります。オペレーションフローの作成を完了するときやオペレーションフローを編集するときに、インテグレーションをパブリッシュできます。

- **Save**: オペレーションのリストを表示するには、右上の **Save** をクリックします。

この手順を繰り返して、別のオペレーションフローを編集します。

API プロバイダーインテグレーションのテスト

- 以下のプラットフォームの 1 つで稼働している API プロバイダーインテグレーションのテスト
  - OpenShift Online
  - OpenShift Dedicated
  - **API 検出が無効になっている** 場合の OpenShift Container Platform

  curl ユーティリティーを使用すると、インテグレーションが想定どおりに動作していることを確認できます。**curl** コマンドに、API プロバイダーインテグレーションのパブリッシュ後に Fuse Online に表示される外部 URL を指定します。この例については、「API プロバイダークイックスタートインテグレーションの例のテストを」を 参照してください。

- **API 検出が有効になっている** 場合の OpenShift Container Platform で稼働している API プロバイダーインテグレーションのテスト
  Red Hat 3scale は API プロバイダーインテグレーションをパブリッシュします。インテグレーションをテストするには、3scale ダッシュボードを開き、インテグレーションの URL を取得します。

  たとえば、Red Hat 3scale でインテグレーションの API へのアクセスを制御したくない場合や、Fuse Online で API プロバイダーをテストしたい場合は、API プロバイダーインテグレーションの検出を無効にできます。検出を無効にすると、インテグレーションが Fuse Online によって再パブリッシュされ、インテグレーションの実行を呼び出しおよびテストする外部 URL が提供されます。これには、Fuse Online でインテグレーションの概要ページに移動します。このページの **Disable discovery** をクリックします。Fuse Online によってインテグレーションが再パブリッシュされ、インテグレーションの URL が提供されます。インテグレーションのテスト方法の例については、「API プロバイダークイックスタートインテグレーションの例 のテスト」を参照してください。テスト後に、API プロバイダーインテグレーションの検出を再度有効にすると、3scale でパブリッシュできるようになります。

  各 API プロバイダーインテグレーションの検出を有効または無効にすることができます。

第4章 API の 3SCALE へのインポート

Red Hat 3scale API Management のサービス検出機能を使用して、OpenShift からサービスをインポートすることができます。

4.1. サービスディスカバリーについて

サービスディスカバリーを使用すると、同じ OpenShift クラスター内で実行されている検出可能な API サービスの有無をスキャンし、関連する API 定義を 3scale に自動的にインポートすることができます。

いつでも API インテグレーションおよび OpenAPI Specification を更新して、それらを後からクラスターと再度同期することもできます。

サービスディスカバリーにより、以下の機能を利用することができます。

- クラスター API を使用して、正しく検出のアノテーションが付けられたサービスのクエリーを行う。
- クラスター内の内部エンドポイントを使用してサービスにアクセスするように 3scale を設定する。
- サービスに関連付けられた OpenAPI Specification を 3scale ActiveDocs としてインポートする。
- OpenShift と Red Hat Single Sign-On (RH-SSO) の承認フローをサポートする。
- Fuse バージョン 7.2 以降の Red Hat Fuse と協調する。

検出可能なサービスをインポートする場合、その namespace はサービスが属するプロジェクト内に維持されます。インポートされたサービスは、新しい顧客がアクセスする API (プロダクト) およびそれに対応する内部 API (バックエンド) になります。

- オンプレミス型 3scale では、3scale API プロバイダーは固有の namespace およびサービスを持つ場合があります。検出されたサービスは、3scale の既存ネイティブサービスと共存することができます。
- Fuse の検出可能サービスは、Fuse のプロダクション namespace にデプロイされます。

4.1.1. 検出可能サービスの条件

3scale が OpenShift クラスターの API を検出できるためには、その API は以下に示す各要素の条件を満たしている必要があります。

Content-Type ヘッダー

API 仕様の **Content-Type** ヘッダーの値は、以下のいずれかでなければなりません。

- **application/swagger+json**
- **application/vnd.oai.openapi+json**
- **application/json**

OpenShift Service Object YAML 定義

- OpenShift Service Object YAML 定義には、以下のメタデータが含まれている必要があります。
  - **discovery.3scale.net** ラベル (必須):「true」に設定します。検出が必要なすべてのサービスを探すために 3scale がセレクター定義を実行する際に、このラベルが使用されます。
  - また、以下のアノテーションが含まれている必要があります。
    discovery.3scale.net/discovery-version (オプション): 3scale の検出プロセスのバージョン。

    discovery.3scale.net/scheme (必須): サービスをホストする URL のスキーム部分。設定可能な値は「http」および「https」です。

    discovery.3scale.net/port (必須): クラスター内のサービスのポート番号。

    discovery.3scale.net/path (オプション): サービスをホストする URL の相対ベースパス。パスがルート (「/」) の場合には、このアノテーションを省略することができます。

    discovery.3scale.net/description-path: サービスの OpenAPI サービス記述ドキュメントへのパス。

    以下に例を示します。

```
metadata:
  annotations:
    discovery.3scale.net/scheme: "https"
    discovery.3scale.net/port: '8081'
    discovery.3scale.net/path: "/api"
    discovery.3scale.net/description-path: "/api/openapi/json"
  labels:
    discovery.3scale.net: "true"
  name: i-task-api
  namespace: fuse
```

  - 管理者権限を持つ OpenShift ユーザーであれば、OpenShift のコンソールで API サービスの YAML ファイルを表示することができます。
    1. **Applications > Services** の順に選択します。
    2. サービスを選択し (例: **i-task-api**)、その Details のページを表示します。
    3. **Actions > Edit YAML** の順に選択し、YAML ファイルを開きます。
    4. ファイルの表示を終えたら、**Cancel** を選択します。

ovs-networkpolicy プラグインを持つクラスター

- OpenShift と 3scale プロジェクト間のトラフィックを許可する場合、**ovs-networkpolicy** プラグインを持つクラスターには、アプリケーションプロジェクト内で作成された **NetworkPolicy** オブジェクトが必要です。
- **NetworkPolicy** オブジェクトの設定についての詳細は、『Networking』の「About network policy」を参照してください。

注記

Fuse Online で API プロバイダーインテグレーションを作成すると、API に必要なアノテーションが自動的に含まれます。

4.2. 検出されたサービスのインポート

OpenShift クラスターから、OpenAPI Specification に準拠する新しい API サービスをインポートすることができます。この API を 3scale で管理することができます。

前提条件

- OpenShift の管理者が、OpenShift クラスターにサービスディスカバリーを設定している。たとえば、OpenShift の管理者は、Fuse Online カスタムリソースを編集して 3scale ユーザーインターフェースの URL を指定し、3scale の検出機能を有効にしている必要があります。
- 「サービスディスカバリー について」で説明するように、3scale の管理者が 3scale デプロイメントでサービス ディスカバリーを設定している。
- 3scale の管理者が、3scale ユーザーまたはサービスアカウント (設定されている認証モードによる) に、API サービスおよびその namespace を表示するのに必要な権限を付与している。詳細は、「 OpenShift プロジェクトへの 3scale アクセスの承認 」を参照してください。
- 「検出可能サービス の条件」で説明するように、サービスディスカバリーを有効にする正しいアノテーションが API に 付けられている。
- API サービスが、3scale をインストールしたクラスターと同じ OpenShift クラスターにデプロイされている。
- API のサービス名およびその namespace (OpenShift プロジェクト) を把握している。

手順

1. 3scale 管理ポータルにログインします。
2. 管理ポータルの Dashboard で、**NEW API** をクリックします。
3. **Import from OpenShift** を選択します。
    - OAuth トークンが有効ではない場合、OpenShift プロジェクトの管理者は、「 OpenShift プロジェクトへの 3scale アクセスの承認」で説明するように 3scale ユーザーのアクセスを承認する 必要があります。
4. **Namespace** フィールドで、API が含まれる OpenShift プロジェクトを指定または選択します (例: **fuse**)。
5. **Name** フィールドで、上記 namespace 内の OpenShift サービスの名前を入力または選択します (例: **i-task-api**)。
6. **Create Service** をクリックします。
7. 新しい API サービスが 3scale に非同期的にインポートされるのを待ちます。管理ポータル右上のセクションに、**The service will be imported shortly. You will receive a notification when it is done.** というメッセージが表示されます。

次のステップ

API 管理の詳細については、Red Hat 3scale API Management のドキュメント を参照してください。