

演習 3 - インポートによるAPIの作成

この演習では、yamlファイルのインポートによりAPIを作成します。

演習 3 - 目的

この演習では、以下の内容を理解できます。

- インポートによるAPI作成の方法
- API設計における `パス`、および、`定義` の設定方法

3.1 - yamlファイルのダウンロード

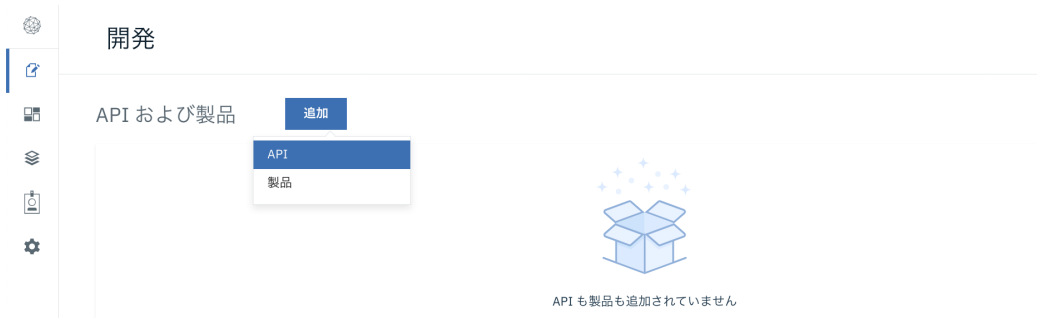
1. インポートするyamlファイルをダウンロードします。以下のURLのテキストをローカルのファイルにコピーして、`findbranch.yaml` という名前で保存します。
 - https://www.ibm.com/support/knowledgecenter/ja/SSMNED_2018/com.ibm.apic.apionprem.doc/findbranch_v6.txt

3.2 - APIの作成

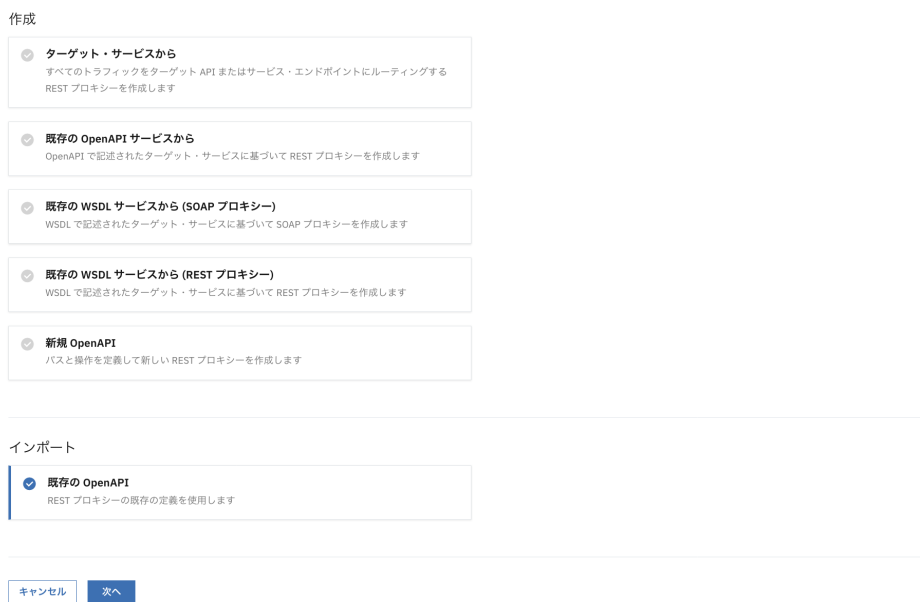
1. API Managerにログインしていない場合には、ログインします。
2. 左のメニューから `開発` を選択し、開発メニューに進みます。



3. 開発 画面で、追加 メニューから API を選択します。



4. 既存のOpenAPI にチェックを入れて、次へ を選択します。



5. 参照 ボタンをクリックして、ローカルに保存した findbranch.yaml ファイルを指定し、次へ をクリックします。

[詳細情報](#)

[詳細情報](#)[詳細情報](#)[詳細情報](#)API の編集

8. 定義されているAPIの情報を確認してみましょう。タイトル は FindBranch です。

API のセットアップ

セキュリティ定義

セキュリティ

パス

定義

プロパティ

ターゲット・サービス

カテゴリ

アクティビティ・ログ

情報

API 要約の詳細を入力してください

タイトル

FindBranch

名前

findbranch

9. 下にスクロールして、基本パスを確認します。/findbranch となっています。

API のセットアップ

セキュリティ定義

セキュリティ

パス

定義

プロパティ

ターゲット・サービス

カテゴリ

アクティビティ・ログ

基本パス

基本パスは API の最初の URL セグメントであり、ホスト名や、パスおよび操作の追加セグメントは含まれません。

基本パス (オプション)

/findbranch

10. 左のメニューから セキュリティ定義 と セキュリティ を確認します。セキュリティ定義 に clientID が設定されており、セキュリティ メニューで、clientID にチェックが入っています。

API のセットアップ

セキュリティ定義

セキュリティ

パス

定義

プロパティ

セキュリティ定義

セキュリティ定義により、API キー検証、アプリケーション・ユーザー認証、OAuth を含む、API エンドポイントへのクライアント・アクセスが制御されます。詳細情報

追加

名前	タイプ	場所
clientID	apiKey	ヘッダー

API のセットアップ

セキュリティ定義

セキュリティ

パス

定義

プロパティ

セキュリティ

ここで選択されたセキュリティ定義は API 全体に適用されますが、個別の操作についてオーバーライドできます。詳細情報

追加

セキュリティ定義

☒ clientID

apiKey

11. 左のメニューから パス を確認します。パスは /details のみが設定されています。/details をクリックしてパスの詳細を確認します。

API のセットアップ

セキュリティ定義

セキュリティ

パス

定義

プロパティ

パス

追加

名前

/details



APIのURLは、<カタログごとのAPIエンドポイント><基本パス><パス> となります。

12. /details パスでは、操作として、GET 操作のみが設定されています。このパスではGET要求のみ受け付けられます。操作 GET をクリックして詳細を表示します。

パス

パスにより、API で公開されている REST リソースが特定されます。操作では、パスを要求および応答の HTTP 動詞、パラメーター、および定義と組み合わせます。 [詳細情報](#)

パス名

/details

パス・パラメーター

追加

必須	名前	場所	タイプ	説明	削除
操作					
追加					
名前					
GET					

13. この GET 操作に必要なパラメーターなど、追加のオプションを設定できます。ここではパラメーターは設定されていません。応答のスキーマに branch と設定されています。branch のデータ構造は、定義で設定されているので後続の手順で確認します。ここでは、特に変更を加えずにこのまま利用します。

操作

操作 ID (オプション)

要約 (オプション)

タグ (オプション)

説明 (オプション)

☐ API セキュリティー定義のオーバーライド
☐ API 生成タイプのオーバーライド
☐ API コンシューム・タイプのオーバーライド

パラメーター

追加

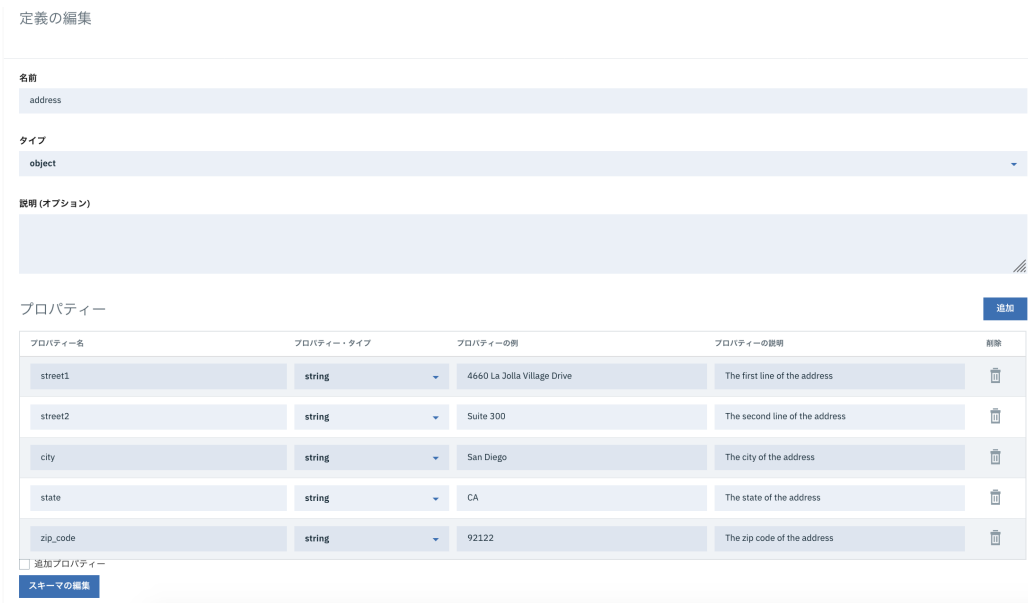
必須	名前	場所	タイプ	説明	削除
Response					
追加					
状況コード	スキーマ	説明	削除		
200	branch	200 OK			


14. 左上の  を2回クリックして、API設計画面に戻ります。

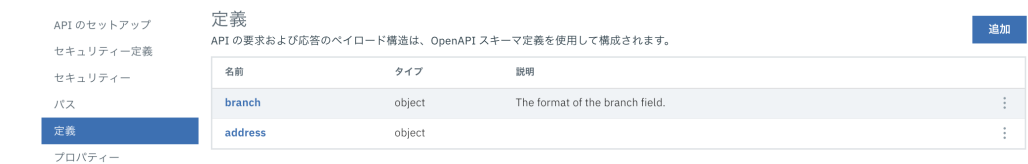
15. API設計画面の左のメニューから 定義 を選択します。 定義 では、APIの要求や応答のペイロードのデータ構造を定義できます。 address をクリックして詳細表示します。



16. 定義 address には、5つの項目がstring形式で定義されていることが分かります。



17. 左上の  をクリックして、 定義 の画面に戻ります。次に branch をクリックして詳細表示します。



18. ソース・ビューにナビゲート をクリックして、ソースを表示し、definitions に定義されているbranchオブジェクトの構造を確認します。branch オブジェクトには、address 、 type 、 id という3つのプロパティ(項目)があります。 address プロパティは、先ほど確認した address 定義を参照しています。

Edit Definitions

名前

branch

複雑な定義はエディターで編集する必要があります

スキーマの編集

ソース・ビューにナビゲート

```

38 definitions:
39   branch:
40     type: object
41     description: The format of the branch field.
42     properties:
43       address:
44         $ref: '#/definitions/address'
45         properties: {}
46         description: The address of the branch
47       type:
48         type: string
49         description: The type of branch
50         example: atm
51       id:
52         type: string
53         description: The ID of the branch
54         example: "\t9d72ece0-7e7b-11e5-9038-55f9f9c08c06"
55     address:
56       type: object
57       properties:
58         street1:

```

19. 上部の **設計** をクリックして、設計画面に戻ります。

開発

findb... 2.0.0

設計 ソース アセンブル

● 実行中

20. 左のメニューから **プロパティ** 選択して確認します。プロパティには、演習1と同様に **target-url** が設定されています。

API のセットアップ

セキュリティ定義

セキュリティ

パス

定義

プロパティ

ターゲット・サービス

プロパティ

追加

プロパティ名	エンコード済み	説明
target-url	false	The URL of the target service

21. ここまでで、APIのインターフェースの設計メニューの確認が終わりました。次に、上部の **アセンブル** をクリックして **アセンブル** 画面に移動し、APIの処理内容を確認します。

開発

findb... 2.0.0

設計 ソース アセンブル


● 実行中

22. **アセンブル** 画面では、**invoke** ポリシーが配置されており、**invoke** ポリシーのプロパティには、URLに **\$(target-url)** が設定されています。呼び出

すURLとしては、APIの 設計 画面で確認したプロパティ `target-url` に設定されている値が利用されます。

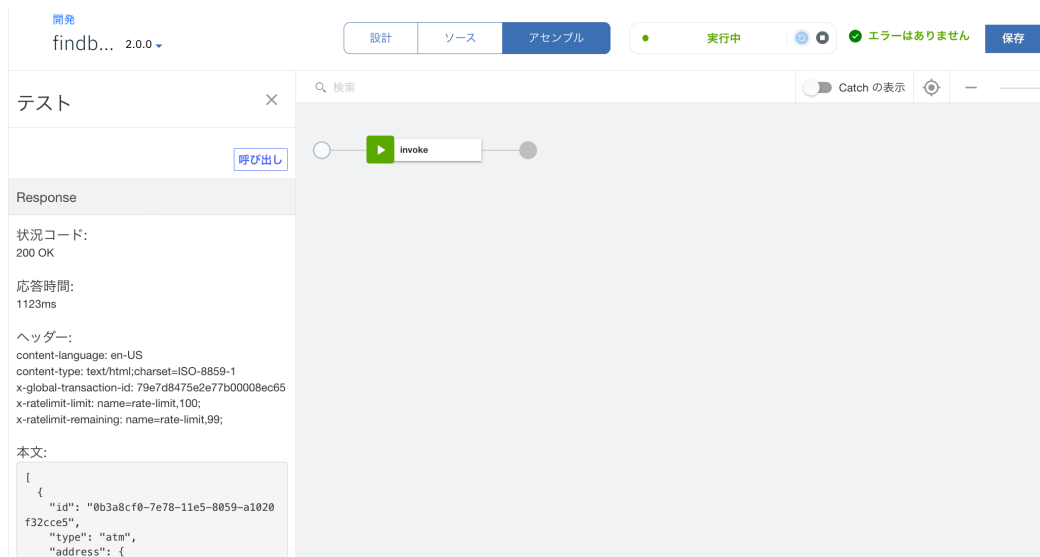



3.3 - APIのテスト

1. 演習1、演習2と同様の手順でAPIのテストを行ってみましょう。画面上のボタン  をクリックしてテストツールを表示します。



2. 製品を再公開 し、呼び出す操作を選択して、 `呼び出し` をクリックしてAPIをテストしてみましょう。応答が正しく返ることを確認してください。



3. 右上の 保存 ボタン  をクリックして、API定義を保存し、ホームメニューをクリックして、ホームに戻ります。

以上で、演習3は終了です。

続いて、 [演習 4 - 製品化と公開](#)に進んでください。