

演習 1 - シンプルなAPIの作成

この演習では、既存のAPIを呼び出すAPIをAPI Connectに登録してテストを行います。

演習 1 - 目的

この演習では、以下の内容を理解できます。

- API Managerへのログイン方法
- 既存APIを呼び出すAPIの定義方法
- API ManagerでのAPIの開発、単体テストの基本的な操作方法

1.1 - 事前作業としてのゲートウェイの構成

1. API Managerにアクセスしてログインします。

ユーザー・レジストリーの選択画面が表示された場合には、API Manager User Registry を選択し、ユーザー名、パスワードを入力して サインイン をクリックしてAPI Managerにログインします。

IBM API Connect

API Manager

IBM Cloud Private user registry ユーザー・レジストリーを使用してサインイン

IBM Cloud Private user registry

OR

API Manager User Registry

Cloud Manager User Registry

IBM API Connect

API Manager

API Manager ユーザー・レジストリーを使用してサインイン

ユーザー名

パスワード

サインイン

パスワードを忘れた場合

OR

IBM Cloud Private user registry

Cloud Manager User Registry

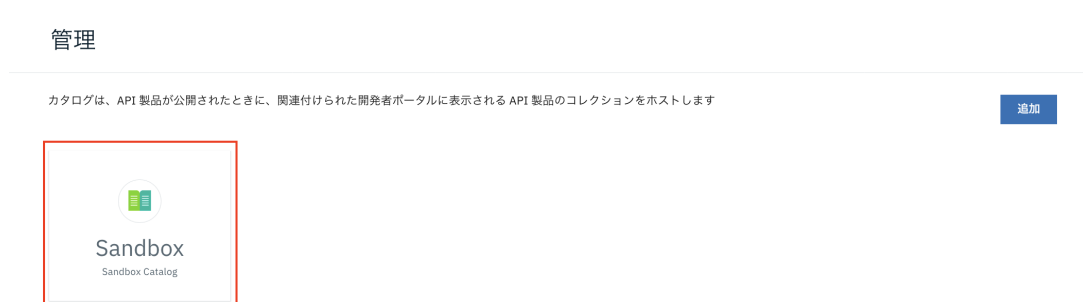


今後、API Managerにログインする場合には、この手順でログインしてください。

1. カタログのゲートウェイ設定を確認して設定します。API Managerのトップページで **カタログの管理** をクリックします。



2. **Sandbox** を選択します。



1. 左のメニューから **設定** をクリックします。



2. ゲートウェイ・サービス メニューを開きます。ゲートウェイ・サービスが登録されていることを確認してください。登録されていない場合は、後続の手順で登録します。登録されている場合は、**1.2 - APIの作成** に進んでください。

編集 をクリックします。

概要

ゲートウェイ・サービス

編集

ゲートウェイ・サービス

ライフサイクル承認

ロール

オンボーディング

API ユーザー・レジストリー

OAuth プロバイダー

API エンドポイント

TLS クライアント・プロファイル

ポータル

プロパティ

タイトル	タイプ	URL
 項目が見つかりません		

3. 利用可能なゲートウェイ・サービスが表示されています。利用するゲートウェイ・サービスにチェックが入っていない場合には、チェックを入れて **保存** をクリックします。

ゲートウェイ・サービスの有効化

<input checked="" type="checkbox"/>	タイトル	タイプ
<input checked="" type="checkbox"/>	gateway1	DataPower API Gateway

キャンセル **保存**



4. 左上の をクリックして戻ります。



API Connect v2018 では2種類のゲートウェイ・タイプが利用できます。API Connect v5ベースのランタイム・アーキテクチャーの *DataPower Gateway (v5 互換)* と、新しい *DataPower API Gateway* です。新しい *DataPower API Gateway* では、API処理に特化しており、ランタイムのパフォーマンスが向上しています。それぞれのゲートウェイ・タイプで一部利用できないポリシーがあるため注意してください。詳細は以下のリンクで確認できます。

- マニュアル: API Connect のゲートウェイ・タイプ: :
https://www.ibm.com/support/knowledgecenter/ja/SSMNED_2018/com.ibm.apic.overview.doc/rpic_gateway_types.html
- APIGW Porting Notes : : <https://github.com/ibm-apiconnect/apigw/wiki/APIGW-Porting-Notes>

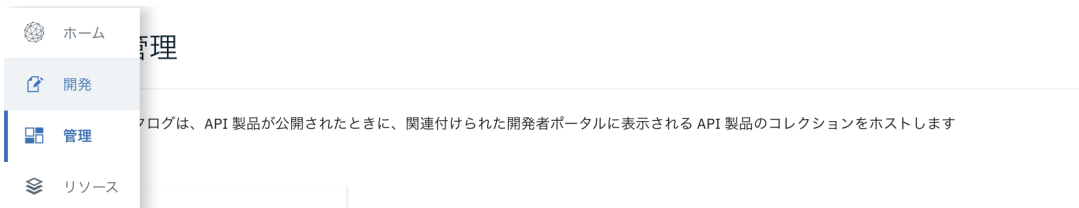
以上でAPIを作成する準備作業が終了しました。



この作業は、新しいカタログを利用する場合に、最初に必要ななります。**Sandbox**カタログはデフォルトのカタログです。カタログとは、APIのデプロイ先となるゲートウェイ上の環境の単位です。APIを公開する際に、公開先のカタログを指定します。APIを公開するとAPIのエンドポイントにカタログ名が含まれます。

1.2 - APIの作成

1. API Managerの左のメニューから **開発** に移動します。



2. 開発 画面で、追加 メニューから API を選択します。



3. ターゲット・サービスから を選択し、次へ をクリックします。



4. 作成するAPIの情報を以下のように入力し、次へ をクリックします。

項目	入力値	備考

タイトル	branch	APIの名前
バージョン	1.0.0	デフォルトで 1.0.0 が入力されます
基本パス	/branch	デフォルトでAPIのタイトルと同じ名前が入力されます
ターゲット・サービス URL	https://apictutorials.mybluemix.net/branches	このAPIが呼び出すターゲットサービスを入力します

情報

この API の詳細を入力します

タイトル

branch

名前

branch

バージョン

1.0.0

基本パス (オプション)

/branch

説明 (オプション)

ターゲット・サービス URL

プロキシ処理するターゲット・サービスの URL を入力します

ターゲット・サービス URL

<https://apictutorials.mybluemix.net/branches>

キャンセル

次へ



ターゲット・サービスURL に入力するURLは、店舗情報をGET要求で取得するAPIです。ブラウザーの別のタブを開き、**ターゲット・サービスURL** に入力したURLをコピー&ペーストして、APIのレスポンスが返ることを確認してください。

- まずは、単純にプロキシするAPIを作成するため、**クライアント ID**を使用した**保護** のチェックをはずし、「次へ」をクリックします。

セキュア

この API のセキュリティを構成します

☐ クライアント ID を使用した保護

☒ CORS

戻る

キャンセル

次へ

6. APIの雛形が完成します。**APIの編集** をクリックして、APIの設定の確認・編集を行います。

要約

✓

OpenAPI 2.0 定義が生成されました

API の編集

7. API設計のメニューから **プロパティ** を開くと、`target-url` というプロパティが自動定義されています。`target-url` をクリックすると、デフォルト値に、API作成時のウィザードで入力したターゲット・サービスURLがセットされています。

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

カテゴリ

アクティビティ・ログ

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

プロパティの編集

名前

target-url

デフォルト値 (オプション)

https://apictutorials.mybluemix.net/branches

説明 (オプション)

URL of the proxy policy

☐ エンコード済み

このプロパティのカテゴリ固有の値を定義します

追加

カタログ

値

削除



プロパティは、デプロイしたカタログに応じてAPIのパラメーターを変えたい場合に便利です。プロパティを利用すれば、デプロイするカタログを変える際に、API定義を修正する必要がありません。ここでは、API Gatewayから呼び出すURLをカタログごとに変えることができるように、プロパティが設定されています。

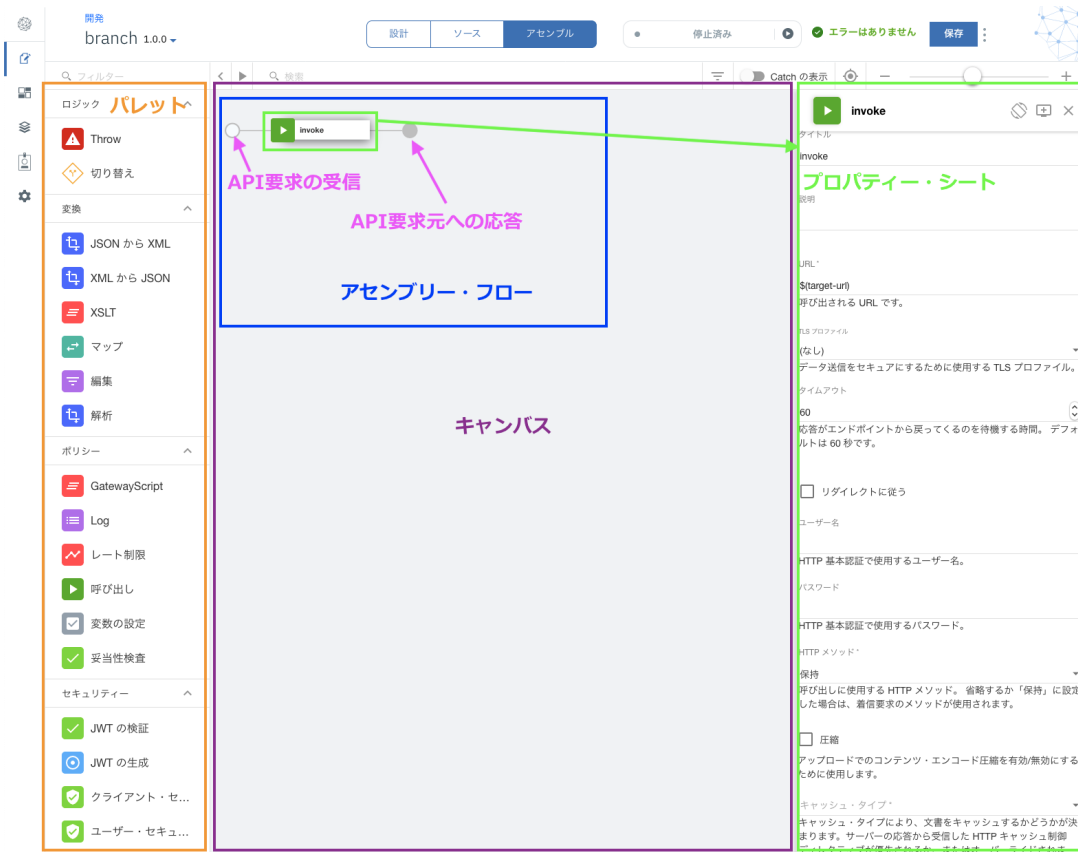
ここまででシンプルなAPIのインターフェース定義が完了しました。

1.3 - アセンブルの定義

1. アセンブルでは、受け付けたAPI要求に対する処理内容を定義します。API開発画面上部から **アセンブル** をクリックして、アセンブル画面に移動します。



2. アセンブル画面が表示されます。アセンブルでは、受け付けたAPI要求に対する処理内容を定義します。

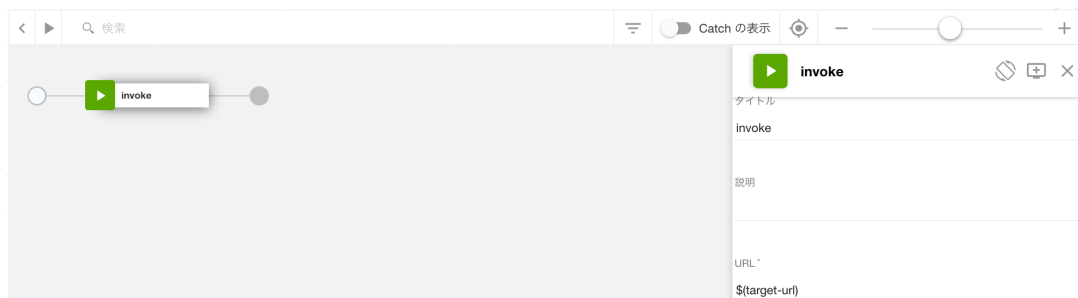


左側の **パレット** には、アセンブルで利用することができるコンポーネントが表示されています。コンポーネントを左側のパレットから真ん中の **キャンバス** 上の **アセンブリ・フロー** 上にドラッグ&ドロップして、処理を組み立てることができます。

アセンブリ・フロー では、受け付けたAPI要求に対する処理内容を定義します。API Connectは、API呼び出しが行われると、まずセキュリティが適用され、次にレート制限が適用された上で、**アセンブリ・フロー** に定義された内容に基づいて処理を行い、API要求元に対して応答を行います。


アセンブリ・フロー 上に配置されている **ポリシー** をクリックすると、右側にその **ポリシー** の **プロパティ・シート** が表示されます。 **プロパティ・シート** にポリシー内での処理を設定します。 **アセンブリ・フロー** には、 **invoke** ポリシーがすでに配置されています。 **invoke** ポリシーは外部のサービスを呼び出すポリシーです。 **invoke** ポリシーをクリックすると、右側に **プロパティ・シート**

開きます。`invoke` ポリシーの URL には、プロパティで定義している `${target-url}` が入力されています。プロパティで定義した値は、`${}` の形式で指定します。



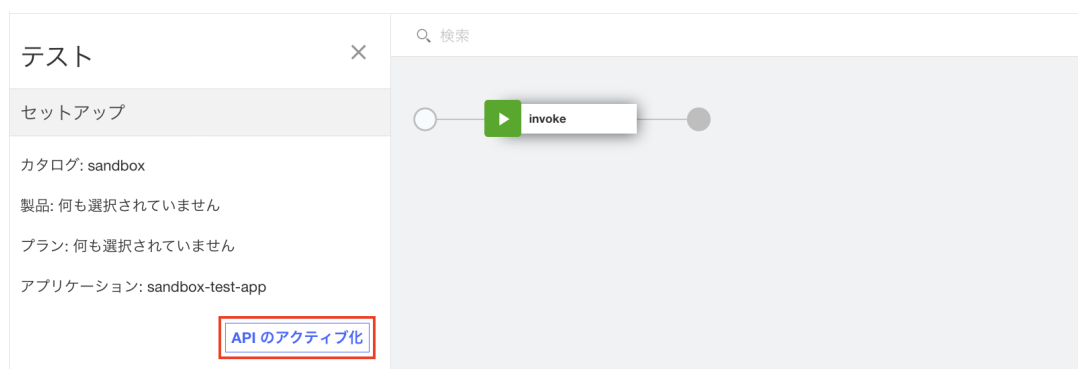
ここまでで、既存のAPIをパススルーして公開するシンプルなAPIの定義が完了しました。

1.4 - APIのテスト

- 作成したAPIのテストを行います。アセンブル画面で利用できるAPIのテストツールを利用します。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。



- APIのアクティブ化 をクリックします。APIのアクティブ化 をクリックすると、テスト用にAPIがカタログ上にデプロイされ、起動されます。デプロイ先のカタログとして `sandbox` が指定されています。APIの公開には、製品やプランも必要ですが、テストツールにより自動的に作成されます。



- しばらくすると、セットアップ・メニュー上に、作成された製品名やプラン名が表示され、右上のAPIのステータスが `実行中` に変更されます。これで準備が整ったのでAPIのテストを実行できます。



1. 操作・メニューで呼び出す操作として `get /` を選択し、呼び出し をクリックしてAPIを呼び出します。



2. API応答が表示されます。状況コードが `200 OK` と表示されており、応答本文にJSON形式の応答が表示されていることを確認します。

Response	
状況コード: 200 OK	
応答時間: 843ms	
ヘッダー: content-language: en-US content-type: text/html;charset=ISO-8859-1 x-global-transaction-id: cdebeb8a5e2684ab006b2ba9 x-ratelimit-limit: name=default,100; x-ratelimit-remaining: name=default,99;	
本文:	
<pre>[{ "id": "0b3a8cf0-7e78-11e5-8059-a1020f32cce5", "type": "atm", "address": { "street1": "600 Anton Blvd.", "street2": "Floor 5", "city": "Costa Mesa", "state": "CA", "zip_code": "92626" } }, { "id": "9d72ece0-7e7b-11e5-9038-55f9f9c08c06",</pre>	



以下のようなエラーが表示された場合には、一度表示されているリンクをクリックして別のタブで開き、証明書の例外を受け入れてください。その後、テストの画面に戻り再度、呼び出し ボタンをクリックしてテストを実行してみてください。

Response	
状況コード: -1	
応答を受信しませんでした。原因としては、ターゲット・サーバーで CORS がサポートされていない、サーバーが使用不可である、信頼できない証明書が輸出された、相互 SSL 認証が必要である、などがあります。	
以下のリンクをクリックすると、新しいタブにサーバーを開きます。ブラウザーに証明書の問題が表示される場合は、それを受け入れることを選択し、ここに戻って再度テストすることができます。 https://gw.10.0.0.1.xip.io/potorg-001/sandbox/branch/	
応答時間: 27ms	

Warning: Potential Security Risk Ahead

Firefox detected a potential security threat and did not continue to gw.10.0.0.1.xip.io. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

[Learn more...](#)

[Go Back \(Recommended\)](#)[Advanced...](#)

Websites prove their identity via certificates. Firefox does not trust this site because it uses a certificate that is not valid for gw.10.0.0.1.xip.io. The certificate is only valid for .

Error code: `MOZILLA_PKIX_ERROR_SELF_SIGNED_CERT`

[View Certificate](#)

[Go Back \(Recommended\)](#)[Accept the Risk and Continue](#)

保存

3. ここまでで、既存のAPIをパススルーで呼び出すシンプルなAPIの定義とテストが完了しました。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1は終了です。

続いて、[演習 2 - セキュリティー定義の付加されたAPIの作成](#)に進んでください。