



# Virtual API Jam - API 管理

[Labs on Github](#)

Updated: June, 2020

---

# ラボ 1 - OpenAPI 仕様による API プロキシの設計と作成

所要時間: 15 分

登場人物: API チーム

## ユースケース

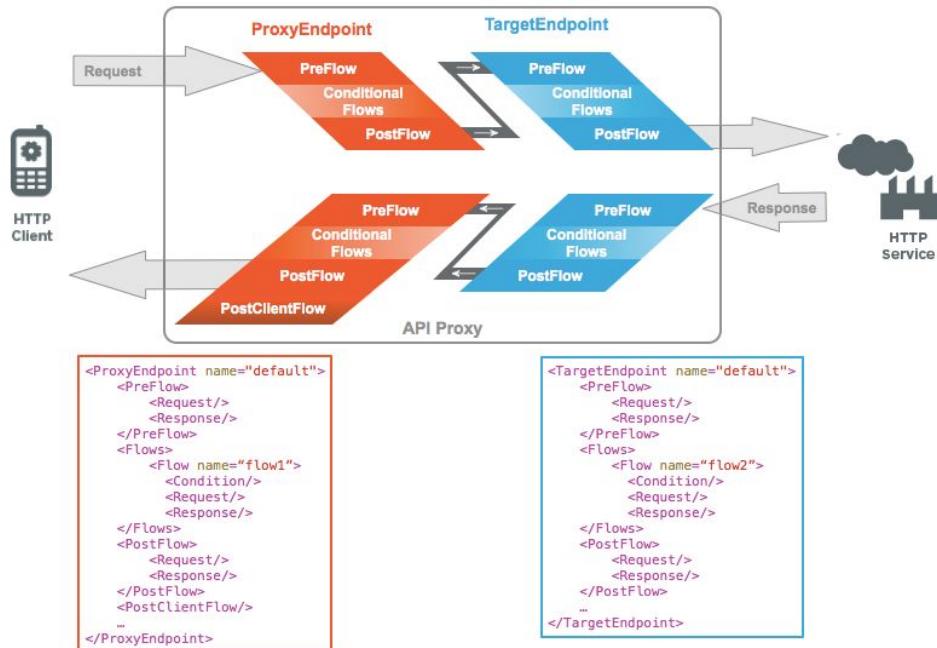
インターネットからのリクエストを受け取るためのリバース プロキシを作成し、そのリクエストを既存のサービスに転送する要件があります。デザインファーストのアプローチに従い、再利用可能なコンポーネントと、API プロキシの構築、API ドキュメントと API テストケースの生成などに使用できる仕様を、OpenAPI 仕様形式で構築することにしました。API プロキシをゼロから構築する代わりに、OpenAPI 仕様 (Swagger) を使用して、Apigee API プロキシを生成する予定です。

## Apigee Edge の利点

Apigee Edge を使用すると、サービスを API として迅速に公開できます。既存の API エンドポイント、汎用 HTTP サービス、アプリケーション (Node.js など) といった、公開したいサービスのファサードを提供する [API プロキシ](#)を作成することで、サービスの迅速な公開が実現します。API プロキシを使用すると、デベロッパーが利用する API エンドポイントからサービス実装が切り離されます。これにより、デベロッパーは将来的なサービスへの変更から守られます。サービスに変更があってもデベロッパーには隠されているため、API の呼び出しが中断されることはありません。Apigee Edge では、API プロキシは API 管理機能を適用するためのランタイム ポリシー構成を行う場所でもあります。。詳しくは、[API と API プロキシに関する詳細情報](#)をご覧ください。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)



Apigee Edge では [OpenAPI 仕様](#)をそのまま使用でき、API プロキシを自動的に生成できます。Apigee Edge には、OpenAPI 仕様の設計や保守に使用できる、OpenAPI 仕様の編集と保管機能が組み込まれています。

```

1  swagger: '2.0'
2  info:
3    title: Hipster Products API
4    version: "1"
5    description: Products API for Hipster Application
6    host: cloud.hipster.s.apigee.com
7    basePath:
8      /
9    schemes:
10      - http
11    consumes:
12      - application/json
13    produces:
14      - application/json
15    paths:
16      /products:
17        get:
18          operationId: GetProducts
19          responses:
20            '200':
21              description: A successful response.
22              schema:
23                $ref: '#/definitions/hipstershopListProductsResponse'
24          tags:
25            - ProductCatalogService
26        /products/{productId}:
27          get:
28            operationId: GetProductDetails
29            parameters:
30              - in: path
31                name: productId
32                type: string
33                required: true
34            responses:
35              '200':
36                description: A successful response.
37                schema:

```

このラボでは、以下の方法を学習します。

- 既存の HTTP サービスの OpenAPI 仕様を設計し、この仕様を Apigee Edge プラットフォーム内に格納する
- 既存の HTTP サービスに受信リクエストを転送する API プロキシを作成する

## Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

## 前提条件

---

- OpenAPI 仕様 (Swagger) に関する基本的な知識があること
- API をテストするための HTTP クライアント (cURL、Postman など) にアクセスできること。これらが利用できない場合は、ブラウザ ウィンドウから [こちらの REST クライアント](#)をご利用ください。

## 手順

---

注: このワークショップでは、複数のユーザーが共有する Apigee 組織内で作業する場合があります。その場合、組織内のすべてのアセット名の前に自分のイニシャルを付けてください。たとえば、仕様名は {your-initials}\_{spec name}、API プロキシ名は {your-initials}\_{proxy name} のようにします。

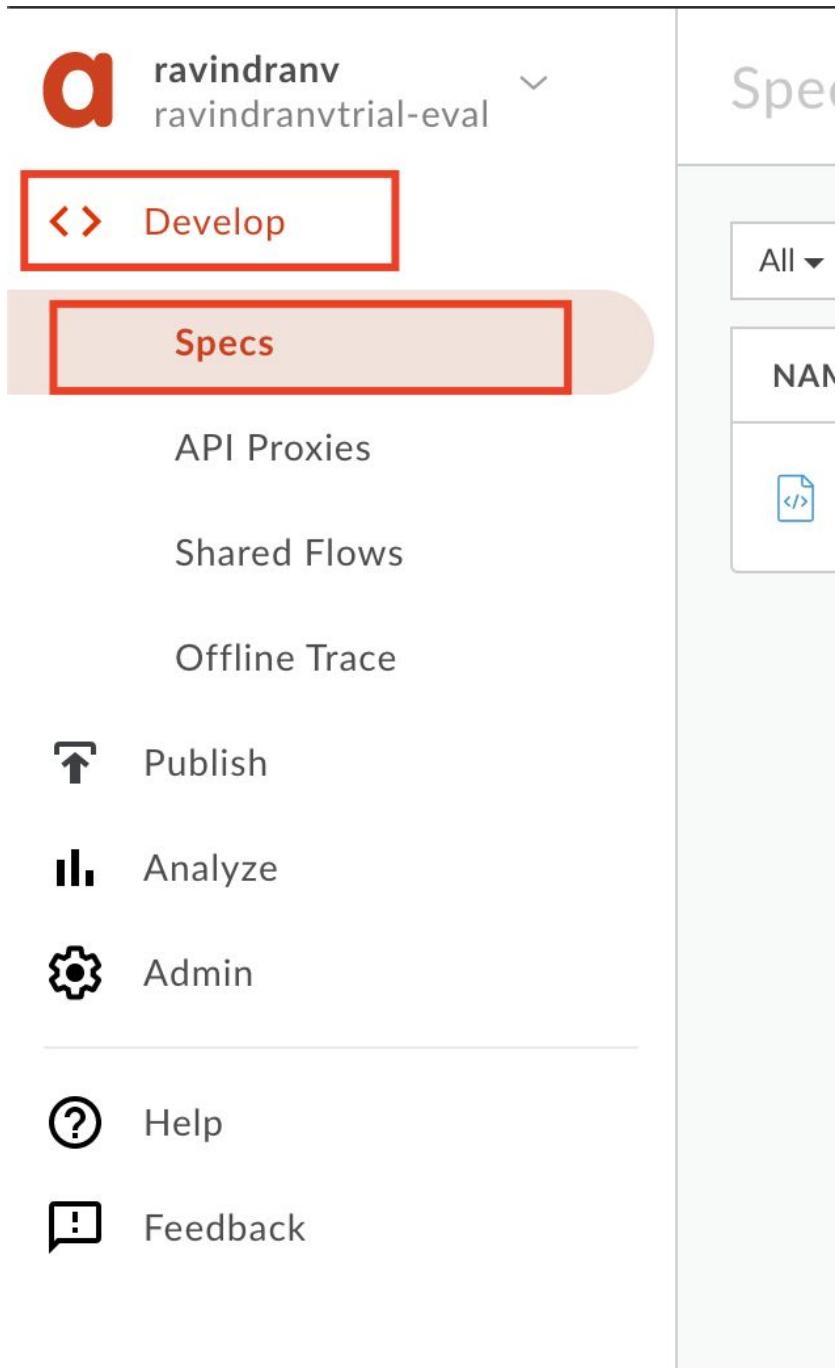
## Open API 仕様を作成する

---

このラボで API エンドポイントとして公開するサンプル HTTP サービスは、<http://cloud.hipster.s.apigee.com/products> にある Hipster Products サービスです。最初に、異なる複数のリソースのエンドポイント (/products と /products/{productId}) 向けに OpenAPI 仕様を設計、作成します。

1. <https://apigee.com/edge> にアクセスしてログインします。これは Edge 管理 UI です。
2. 横のナビゲーションメニューから [Develop] → [Specs] を選択します。

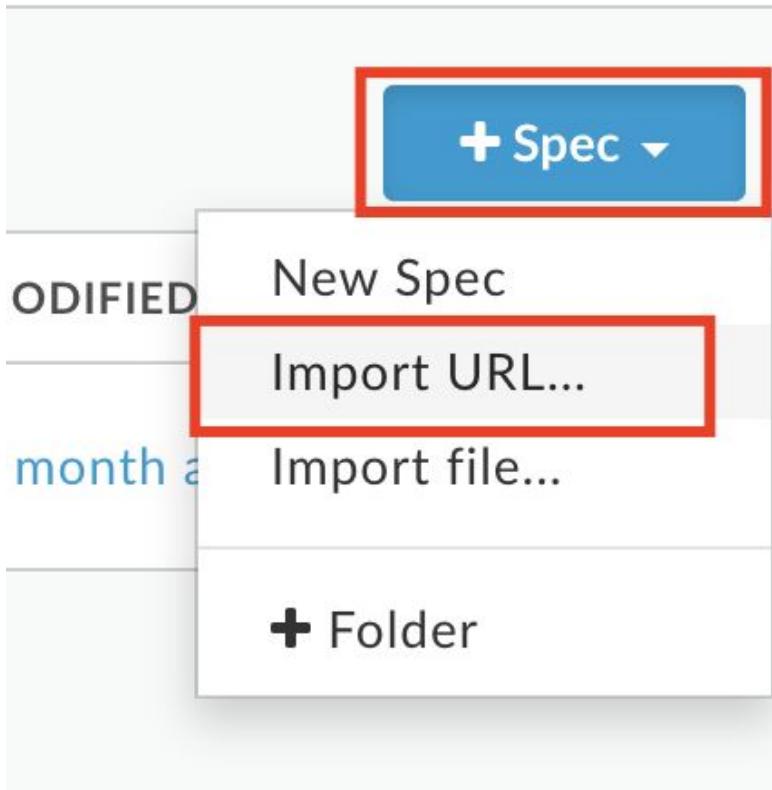




3. このラボでは設計済みの仕様のサンプルが利用できますので、これをApigee 組織の仕様ストアにインポートします。[+Spec] をクリックします。[Import URL] をクリックして、新しい仕様を既存のソースから追加します。

Google Cloud

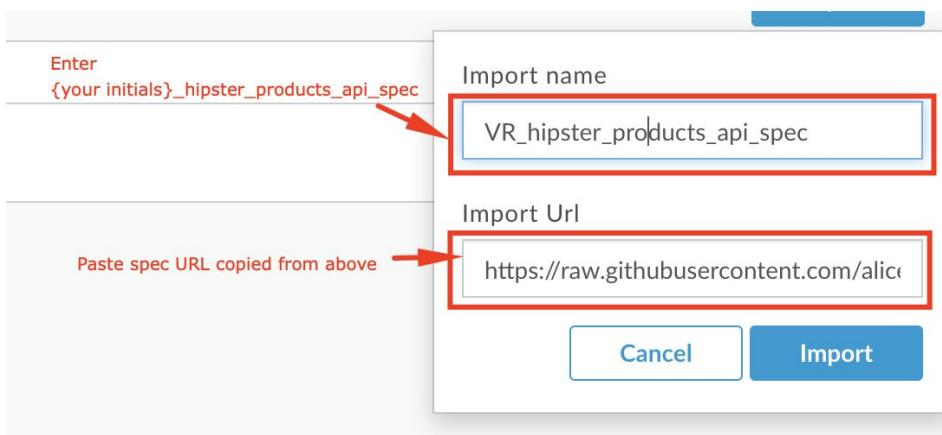
For more information visit [google.com/cloud](http://google.com/cloud)



4. 仕様の詳細を入力します。{your-initials} は自分のイニシャルに置き換えます。

- ファイル名: {your-initials}\_hipster\_products\_api\_spec
- URL:

<https://raw.githubusercontent.com/aliceinapiland/apijam/master/Module-1/Resources/products-catalog-spec.yaml>



## Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

5. 値を確認し、[Import] をクリックします。仕様が Apigee Edge にインポートされ、利用できるようになります。インポートした仕様がリストに表示されます。次に例を示します。

The screenshot shows the Apigee Edge dashboard under the 'Develop' tab. On the left, there's a sidebar with options like 'API Proxies', 'Shared Flows', 'Offline Trace', 'Publish', 'Analyze', and 'Admin'. The main area is titled 'Specs' and shows a table with two rows. The first row has a red box around it and a callout text 'Spec that you just created!'. The second row is for 'Products Backend' and was modified '1 month ago'. There are icons for edit and delete next to each row.

6. リストから {your-initials}\_hipster\_products\_api\_spec をクリックすると、API の詳細と API リソースが記載されている OpenAPI 仕様エディタとインタラクティブなドキュメントにアクセスできます。

The screenshot shows the 'OpenAPI Spec Editor' for the 'VR\_hipster\_products\_api\_spec'. On the left, the API definition is shown as a code editor with lines 1 through 38. Lines 1-10 show basic metadata like swagger version, info, title, version, and host. Lines 11-38 define the '/products' endpoint with a 'GET' operation for listing products and a 'GET' operation for getting details by product ID. On the right, there's a preview window titled 'Hipster Products API' showing the 'Preview of Interactive API Docs' for 'ProductCatalogService'. It lists two endpoints: '/products' (GET) and '/products/{productId}' (GET). Below the endpoints, there's a 'Models' section with 'hipstershopListProductsResponse' and 'hipstershopMoney'.

## API プロキシを作成する

1. OpenAPI 仕様から API プロキシを作成しましょう。横のナビゲーションメニューから [Develop] → [API Proxies] をクリックします。

The screenshot shows the API Platform interface. On the left, there's a sidebar with various options: Develop (highlighted with a red border), Specs, API Proxies (highlighted with a red border), Shared Flows, Offline Trace, Publish, Analyze, Admin, Help, and Feedback. To the right, there's a main panel titled 'Proxi' which lists environment names: Enviro, NAME, Emplc, oauth, Mock, hellov, and oauth.

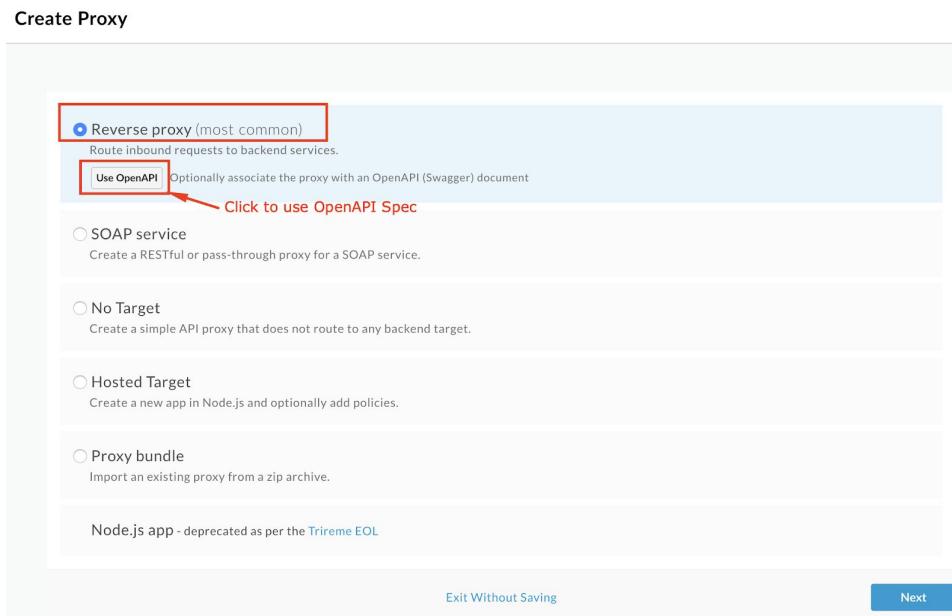
2. [+Proxy] をクリックします。プロキシの構築ウィザードが表示されます。



Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

3. [Reverse proxy] を選択し、[Reverse proxy] オプションの下にある [Use OpenAPI] をクリックします。



4. 仕様一覧のポップアップダイアログが表示されます。  
{your-initials}\_hipster\_products\_api\_spec を選択して [Select] をクリックします。

My Specs

Upload file Import from a URL

NAME LAST ACCESSED ↓

VR_hipster_products_api_spec	12 Sep 2019, 06:34
Products Backend	31 Jul 2019, 07:49

Cancel Select

5. [Reverse Proxy] オプションの下に、選択した OpenAPI 仕様の URL が表示されます。[Next] をクリックして続行します。

Create Proxy

Reverse proxy (most common)  
Route inbound requests to backend services.

SOAP service  
Create a RESTful or pass-through proxy for a SOAP service.

No Target  
Create a simple API proxy that does not route to any backend target.

Hosted Target  
Create a new app in Node.js and optionally add policies.

Proxy bundle  
Import an existing proxy from a zip archive.

Node.js app - deprecated as per the [Trireme EOL](#)

Spec updated to selected

Next

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

6. プロキシ ウィザードに詳細を入力します。{your-initials} は自分のイニシャルに置き換えます。

- Proxy Name: {your\_initials}\_Hipster-Products-API
- Proxy Base Path: /v1/{your\_initials}\_hipster-products-api
- Existing API: OpenAPI 仕様から自動入力されたフィールドの値を確認します。

#### Create Proxy

The screenshot shows the 'Create Proxy' wizard in progress, specifically the 'Details' step. The navigation bar at the top includes tabs for Type, Details, Flows, Security, Virtual Hosts, Build, and Summary. The 'Details' tab is currently selected.

Below the tabs, a section titled 'Specify the proxy details.' contains the following fields:

- Proxy Name \***: A text input field containing 'VR\_Hipster-Products-API'. This field is highlighted with a red box.
- Proxy Base Path \***: A text input field containing '/v1/{your initials}\_hipster-products-api'. This field is also highlighted with a red box.
- Existing API \***: A text input field containing 'http://cloud.hipster.sapigee.com/'. A small note below it says: 'Defines the target URL invoked on behalf of this API proxy. Any URL that is accessible over the open Internet can be used. Example: https://mocktarget.apigee.net'
- Description**: A text input field containing 'Products API for Hipster Application'.

At the bottom of the screen, there are three buttons: 'Previous', 'Exit Without Saving', and 'Next' (which is highlighted in blue).

7. 値を確認し、[Next] をクリックします。

8. OpenAPI 仕様から入力済みの API プロキシリソース一覧の選択や、選択解除ができます。すべてを選択して [Next] をクリックします。

## Create Proxy

Select the operations to manage with policies. You can add more later.  
2 of 2 operations selected

PATH	VERB	OPERATION ID	SUMMARY
/products	GET	GetProducts	
/products/{productId}	GET	GetProductDetails	

Previous      Exit Without Saving      Next

9. このプロキシではセキュリティ ポリシーを適用しないので、[Authorization] では [Pass through (none)] を選択します。[Next] をクリックします。

## Create Proxy

Secure access for users and clients.

Authorization  Pass through (none)  API Key  OAuth 2.0

Browser  Add CORS headers

Previous      Exit Without Saving      Next

10. デフォルトの仮想ホストの構成に進みます。



For more information visit [google.com/cloud](http://google.com/cloud)

## Create Proxy

NAME	ENVIRONMENT	HOST ALIASES
default	prod	http://ravindrantrial- eval-prod.apigee.net http://ravindrantrial- eval-test.apigee.net
secure	prod	https://ravindrantrial- eval-prod.apigee.net https://ravindrantrial- eval-test.apigee.net

11. デプロイ対象として test 環境だけが選択されていることを確認し、[Build and Deploy] をクリックします。

## Create Proxy

You are ready to build and deploy your API proxy.

Deploy Environments  prod  test

Proxy Name VR\_Hipster-Products-API

Proxy Type Reverse proxy

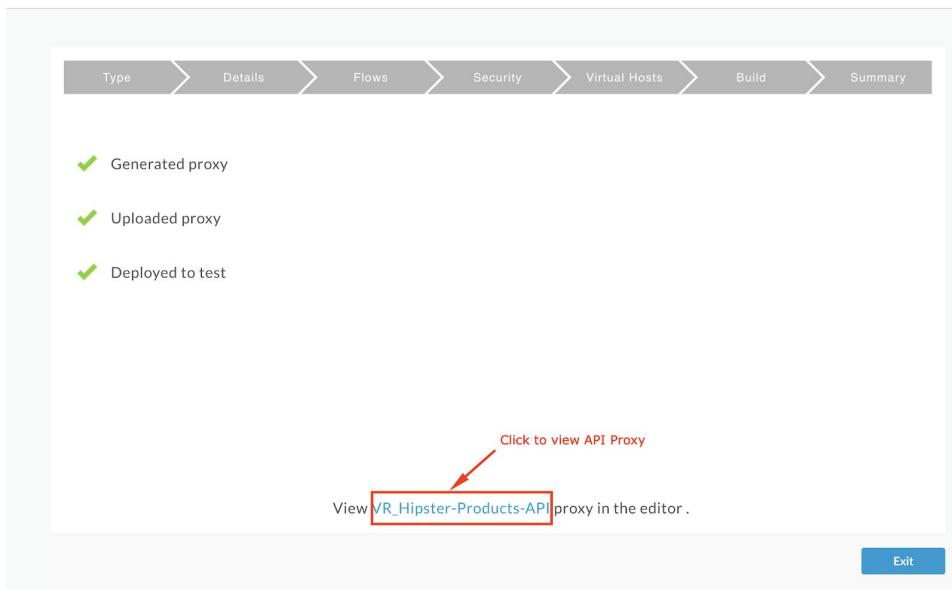
Virtual Hosts secure

12. API プロキシが構築、デプロイされたら、リンクをクリックしてください。プロキシ エディタにプロキシが表示されます。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

## Create Proxy



13. これで完了です。既存のバックエンド サービス向けにリバース プロキシが構築されました。プロキシの [Overview] 画面が表示されます。

API Proxies > VR\_Hipster-Products-API > Overview > 1

API Proxy Operations

Proxy revision management

Deployment to environments

Proxy overview

Proxy configurations

Runtime

Revision 1 Summary

Created: a few seconds ago , Updated: a few seconds ago .

Description: Products API for Hipster Application

Deployments

Environment: test      Deployed environment: 1      Status: ●      URL: https://ravindrantrial-eval-test.apigee.net/v1/vr\_hipster-products-api

API endpoint URL

Click to view API resource paths

Expand All   Collapse All

Proxy Endpoints

Name: default      Base Path: /v1/vr\_hipster-products-api      Target Endpoints: default

Endpoint Flow Name	Method	Path / Condition
PreFlow	ALL	n/a
GetProducts	GET	/products
GetProductDetails	GET	/products/*
PostFlow	ALL	n/a

Target Endpoints

Name: default      Target: http://cloud.hipster.s.apigee.com/      Used by Proxy Endpoints: default

Expand All   Collapse All

**Google Cloud**

For more information visit [google.com/cloud](http://google.com/cloud)

# API プロキシをテストする

新たに構築された API プロキシをテストしてみましょう。curl などのターミナル HTTP クライアントか、[こちらのサンプル REST クライアント](#)などのブラウザベース クライアントを使用できます。

## cURL の使用

org には組織名を、env には API がデプロイされる環境を指定します。

```
curl -X GET "https://{{org}}-{{env}}.apigee.net/{{your initials}}_hipster-products-api/products"
```

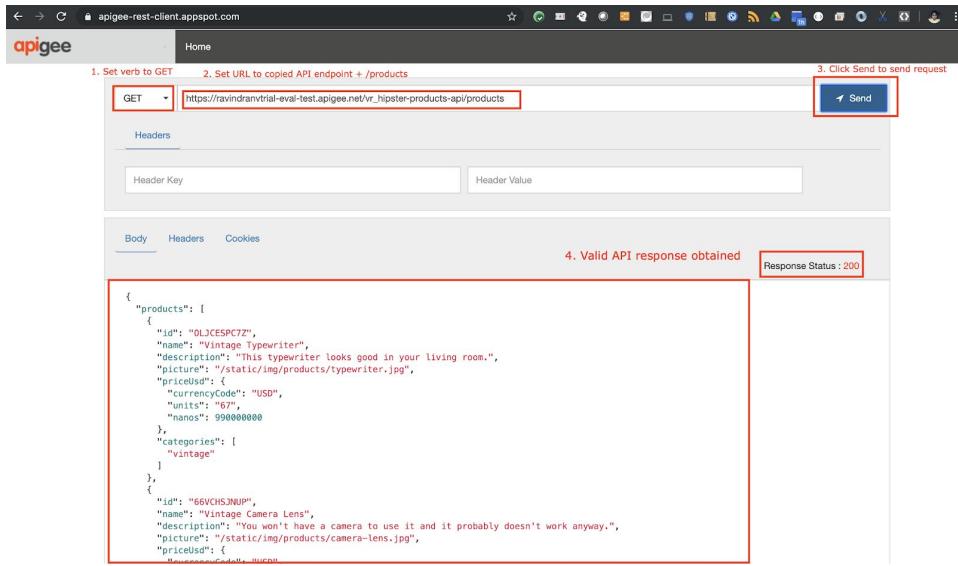
## サンプル REST クライアントの使用:

- 新しいブラウザ ウィンドウで REST クライアントを開きます。
- API プロキシの URL をコピーします。

The screenshot shows the Apigee API Proxies Overview page for the VR\_Hipster-Products-API. It displays the following details:

- Revision 1 Summary:** Created: 2 minutes ago, Updated: 2 minutes ago. Description: Products API for Hipster Application.
- Deployments:** A table showing one deployment entry for environment 'test' with revision 1. The URL column contains two entries:
  - [http://ravindrantrial-eval-test.apigee.net/vr\\_hipster-products-api](http://ravindrantrial-eval-test.apigee.net/vr_hipster-products-api) [ - ]
  - [https://ravindrantrial-eval-test.apigee.net/vr\\_hipster-products-api](https://ravindrantrial-eval-test.apigee.net/vr_hipster-products-api) API endpoint - copy
- Proxy Endpoints:** A table showing one endpoint named 'default' with base path '/vr\_hipster-products-api'.
- Target Endpoints:** A table showing one target endpoint named 'default' with target URL 'http://cloud.hipster.s.apigee.com'.

- それを REST クライアントに貼り付けます。リソースパスの /products を付けて GET 呼び出しを行います。
- 成功した場合のレスポンスは次のようにになります。



## API プロキシを保存する

1. API プロキシを API バンドルとしてローカルに保存し、他のラボで再利用できるようにします。
2. プロキシバンドルをダウンロードして API プロキシを保存します。手順についてはスクリーンショットをご覧ください。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

The screenshot shows the 'Overview' page for the 'VR\_Hipster-Products-API'. At the top, there are buttons for 'Project ▾', 'Save', 'Revision 1 ▾', and 'Deployment ▾'. A context menu is open over the 'Revision 1' button, with 'Download Revision' highlighted by a red box and arrow. The menu also includes options like 'Save as New Revision', 'Save as New API Proxy...', 'Undo All', 'Delete API Proxy...', and 'Delete Current Revision...'. Below the menu, a table lists the API Proxy History with one entry: Revision 1, Status (green dot), URL (http://ravindra and https://ravindr). The 'Proxy Endpoints' and 'Target Endpoints' sections are also visible.

## ラボの動画

動画を視聴して学習を進めるには、OpenAPI 仕様でリバース プロキシを作成する動画（<https://www.youtube.com/watch?v=3XBG9QOUPzg>）をご覧ください。

## さらに理解を深める

OpenAPI 仕様を使用してリバース プロキシを作成しました。[Develop] タブをクリックすると、OpenAPI 仕様から取得されるフロー条件について詳しく学ぶことができます。また、OpenAPI 仕様エディタの上部にあるリンクを使用して、OpenAPI 仕様の編集と API プロキシの生成が可能な OpenAPI 仕様エディタについて詳しく学ぶこともできます。プロキシの Overview ページにある [Trace] タブを試してみましょう。

## 理解度チェック

## Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

1. ダウンロードしたプロキシバンドルはどのようにしてインポートしますか？
2. Apigee Edge は API のバージョニングをどのように処理しますか？
3. API プロキシを作成、更新、削除する管理 API は Apigee にありますか？

## まとめ

---

これでハンズオン レッスンは終了です。このラボでは、OpenAPI 仕様と Apigee Edge プロキシ ウィザードを使用して、既存のバックエンド向けにプロキシを作成する方法を学びました。

## 参考資料

---

- API プロキシに関する Apigee ドキュメントのリンクをご紹介します。
  - シンプルな API プロキシの構築 -  
<http://docs.apigee.com/api-services/content/build-simple-api-proxy>
  - API プロキシの設計と開発に関するベスト プラクティス -  
<http://docs.apigee.com/api-services/content/best-practices-api-proxy-design-and-development>
- 「Anatomy of an API proxy」の 4 分間の動画 - <https://youtu.be/O5DJuCXXIRg>

## ラボ 2 - Apigee Edge での API セキュリティと API プロデューサー / コンシューマの関係

所要時間: 20 分

登場人物: API チーム

### ユースケース

さまざまなアプリ（API クライアント）での利用を目指し、保護しながら公開したい API があります。API に承認済みアクセス権を設定するだけでなく、API を呼び出しているアプリを特定し、管理できる必要があります。これで、呼び出し元を基準とする API 動作のカスタマイズ、さまざまなアプリ別の利用パターンに基づくデータ収集、分析ダッシュボードへのデータ表示が可能になります。

### Apigee Edge の利点

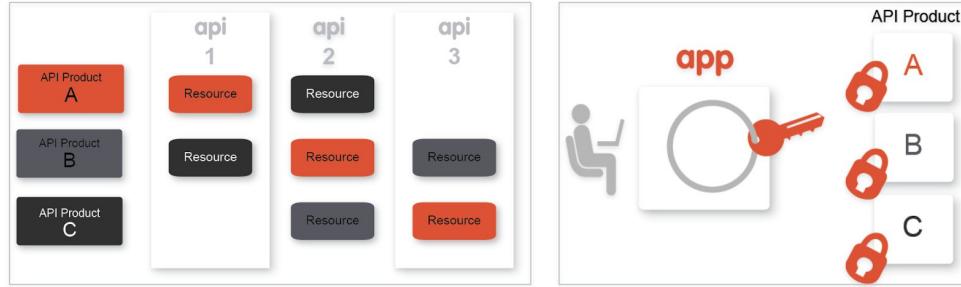
### API プロキシ - API プロダクト - アプリの関係

Apigee Edge では、API を保護し、コンシューマによるアクセスを管理するため、まず、API プロキシ、API プロダクト、アプリの関係を知る必要があります。

API プロキシを使用すると、API 設計仕様に対応した API エンドポイントを公開できるだけでなく、フロントエンド（クライアントアプリ）から API バックエンド（ターゲットサービス）が分離され、API の生産と利用が分離されます。このためには API を利用する方法を定めた構成である「API プロダクト」を作成する必要があります。API プロダクト構成には、特定の API プロダクトを通じた API の利用ルールを定めたメタデータが含まれことがあります。このルールには利用が認められた割り当て量（例: 1分あたり 100 回の API 呼び出し）、公開設定（一般、限定、内部）、API リソース制限（例: /products/{product ID} ではなく、/products リソース URL のみ呼び出し可能）、呼び出し元がアクセスできる API デプロイ環境（例: test 環境や prod 環境）などが含まれています。作成された API プロダクトはクライアントアプリにサブスクライブされます。サブスクライブ時に、Apigee はアプリ用の API Key-Secret ペアを自動的に生成、プロビジョニングします。生成された認証情報を使用して、認証・認可を伴う API エンドポイントをアプリコード内から呼び出すことができます。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)



## API プロキシ構成

Apigee では、さまざまな方法 ([API キー](#)、[OAuth](#)、[JWT トークン](#)、[SAML](#) など) で API を保護し、API 呼び出しを認可します。このラボではシンプルな API キー検証を使用して API を保護する方法を取り上げます。

API プロキシ内で [Verify API Key ポリシー](#) を使用し、API キー検証に基づいた受信 API 呼び出しを認証、認可します。API キー検証が完了すると、[Verify API Key ポリシー](#) によって、呼び出しを行うアプリに関する詳細、アプリのデベロッパー、呼び出しに関連する API プロダクトなどが API プロキシランタイム コンテキストに追加されます。このコンテキストは、クライアントアプリに基づいた割り当て量の適用やルーティングなどの API の動作に反映するため、他のポリシーのパラメータ化に使用できます。また、このデータは Apigee Analytics を通じてビジネス分析情報を取得するために抽出、使用できます。

### このラボの内容

- 既存の API プロキシに [Verify API Key ポリシー](#) を設定し、Apigee Trace ツールを使用して動作中のポリシーを確認する
- API プロキシを API プロダクトにバンドルする
- この API プロダクトをサブスクリーブするアプリを組織内に登録し、認可された API 利用をテストする

## 前提条件

このラボでは、現時点で保護されていない API プロキシが必要になります。利用できる API プロキシがない場合は、[API 設計: OpenAPI 仕様からリバース プロキシを作成する](#) ラボに戻って準備してから以降の手順に沿って操作してください。

## 手順

### 保護対象の API プロキシを選択してテストする

- Apigee Edge 管理 UI (<https://login.apigee.com>) にログインします。  
[Develop] → [API Proxies] に移動し、[ラボ 1](#)で作成した API プロキシ（{your\_initials}\_Hipster-Products-API）を選択します。

The screenshot shows the Apigee Edge management interface. On the left, a sidebar has tabs for 'Develop', 'Specs', 'API Proxies' (which is highlighted with a red box), 'Shared Flows', 'Offline Trace', 'Publish', 'Analyze', 'Admin', 'Help', and 'Feedback'. The main area is titled 'Proxies' with a sub-header 'Environment prod All'. It lists three proxies: 'VR\_Hipster-Products-API' (status: 0, last modified: 3 months ago), 'helloworld' (status: 0, last modified: 3 months ago), and 'oauth' (status: 0, last modified: 3 months ago). The 'prod' environment is indicated by a green dot next to the proxy names.

- [Overview] ページで、API プロキシが環境にデプロイされていることを確認します。選択されたリビジョンの API プロキシがデプロイされる環境は、緑色の丸で示されます。デプロイされていない場合、[Deployment] プルダウンから環境をクリックし、その環境に API プロキシをデプロイします。

The screenshot shows the 'Overview' page for the 'VR\_Hipster-Products-API' proxy. At the top, it says 'API Proxies > VR\_Hipster-Products-API > Overview > 1 Active revision'. Below this, there's a 'Revision 1' dropdown set to 'prod'. To its right is a 'Deployment' dropdown with 'prod' selected. Further right are buttons for 'OVERVIEW', 'DEVELOP' (highlighted with a red box), 'TRACE' (also highlighted with a red box), and 'PERFOR'. Below these are sections for 'Revision 1 Summary' (Created: Sep 12th, 2019, Updated: Sep 12th, 2019) and 'Deployments'. The 'Deployments' table shows one entry: Environment 'test', Revision '1', Status '●' (green), and URL 'http://ravindrantrial-eval-test.apigee.net/vr\_hipster-products-api [+]'.

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

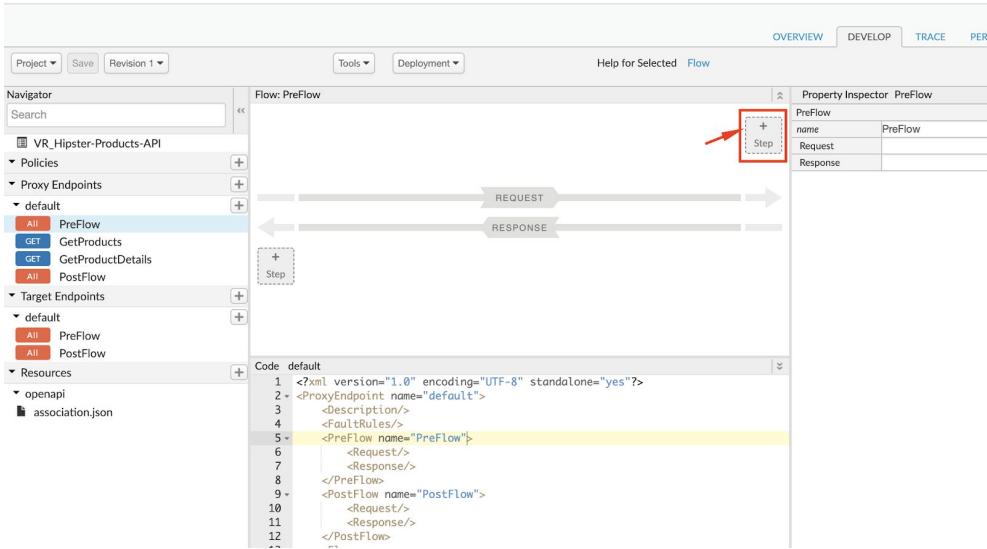
3. 組み込みの Trace ツールを使用して、呼び出しが正常に行われていることを確認します。

- ウィンドウの上部近くにある [Trace] タブをクリックします。
- [Trace] ビューではトレースを開始できます。最大 10 分間のトレースが可能で、その間すべてのリクエストが取り込まれ（リクエストの送信元が Trace ツールか他のクライアントかは関係ありません）、リクエストのトレースが表示されます。
- [Start Trace Session] をクリックしてトレース セッションを開始します。
- 前のラボで構築した API プロキシを使用している場合、同じ API 呼び出しリクエストを送信する前に、/products リソースパスを URL に追加します。[Send] をクリックしてリクエストを送信します。
- API 呼び出しが成功すると 2xx というレスポンスが表示されます（トレース結果が表示されるまで数秒かかる場合があります）。

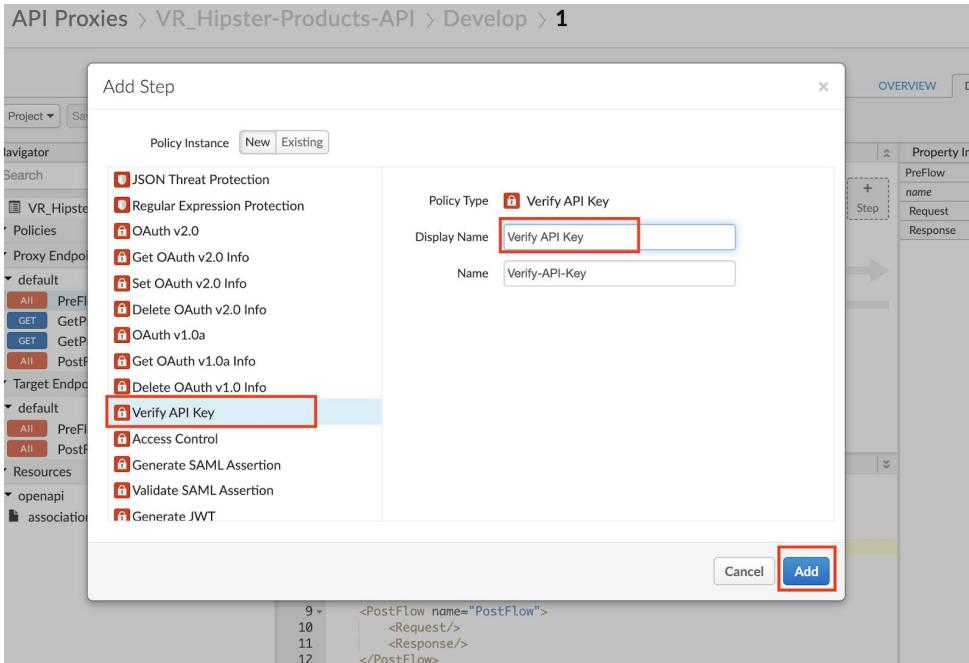
The screenshot shows the Apigee Trace interface for the VR\_Hipster-Products-API. The top navigation bar includes 'Click to Start/Stop Trace', 'OVERVIEW', 'DEVELOP', 'TRACE' (which is selected), and 'PERFORMANCE'. Below this, a 'Transactions' table lists a single entry: 1 200 GET /vr\_hipster-products-api/products, with a duration of 16 ms. A red box highlights this row. To the right, there's a 'Send Requests' section with a 'Method' dropdown set to 'GET', a 'URL' input field containing 'http://ravindrantrial-eval-test.apigee.net/vr\_hipster-products-api/products', and a 'Send' button. A status indicator shows '200'. Below these are sections for 'Transaction Map' (showing a flowchart of the request path) and 'Phase Details' (Request Received from Client: GET /vr\_hipster-products-api/products, Response Sent to Client: 200 OK). The bottom left contains 'View Options' with checkboxes for 'Show Disabled Policies (none)', 'Show Skipped Phases (1)', 'Show All Flowinfos (13)', and 'Automatically Compare Selected Phase'. The bottom right shows detailed 'Request Headers' and 'Response Headers' tables.

## Verify API Key ポリシーを追加する

1. 横のメニューから [Develop] > [API Proxies] に移動し、API プロキシを開きます。[Develop] タブ（ページの右上）をクリックし、フローエディタを表示します（ペインを動かさないと、リクエストとレスポンスのフローラインが完全に表示されないことがあります）。
2. リクエストフローで [+Step] をクリックします。



3. リストの [Security] セクションから [Verify API Key] ポリシーを選択します。ポリシーの名前は変更できますが、デフォルトのままでもかまいません。[Add] をクリックします。



- ポリシーは、[Request] フローに以前から含まれているポリシーの後に追加されます。新たに追加されたポリシーを最初に起動したいので、このポリシーをフロー内のポリシーの左端にドロップします。

## Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

- [Verify API Key] ポリシーを選択すると、このポリシーの構成が表示されます（デフォルトのポリシー構成は以下のとおりです）。API キーは request.queryparam.apikey 変数としてコンテキストから取得されていることに注意してください。これがデフォルトですが、希望するパラメータからキーを取得するようにポリシーを構成できます。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<VerifyAPIKey async="false" continueOnError="false" enabled="true"
name="Verify-API-Key">
  <DisplayName>Verify API Key</DisplayName>
  <Properties/>
  <APIKey ref="request.queryparam.apikey"/>
</VerifyAPIKey>
```



- API プロキシを保存します。
- ウィンドウの上部近くにある [Trace] タブをクリックし、[Start Trace Session] をクリックしてトレース セッションを開始します。
    - 前のラボで構築した API プロキシを使用している場合、同じ API 呼び出しリクエストを送信する前に、/products リソースパスを URL に追加します。[Send] をクリックしてリクエストを送信します。
    - API プロキシはクエリ パラメータとして API キーを期待していたので、この API 呼び出しに対しては 401 (認可失敗) レスポンスが表示されます。以下のトレース セッションをご覧ください。

注: この後の手順で、この API呼び出しを正常に完了できるようにする API キーを取得します。

The screenshot shows the API Proxies interface for the VR\_Hipster-Products-API. A trace session is active, showing a single transaction. The transaction details are as follows:

- Status: 401
- Method: GET
- URI: /vr\_hipster-products-api/products
- Elapsed: 18 ms

The URL for the request is http://ravindrantrial-eval.apigee.net/vr\_hipster-products-api/products. The response status is also 401.

The Transaction Map shows the flow of the transaction, with a red box highlighting the final step where the response is sent back to the client.

Phase Details indicate the error was sent to the client with the status code 401 Unauthorized.

Error Headers show Content-Type: application/json.

Error Content displays the JSON response body: {"fault": {"faultstring": "Failed to resolve API Key variable request.queryparam.apikey", "detail": {"errorcode": "steps.oauth.v2.FailedToResolveAPIKey"}}}

View Options and Phase Details sections are also visible on the left.

## API プロダクトを作成する

1. 横のナビゲーションメニューから [Publish] → [API Products] を選択し、 [+API Product] ボタンをクリックします。

The screenshot shows the API Products page. On the left, there is a navigation menu with several items: Develop, Publish (highlighted), API Products (highlighted), Portals, Developer Programs, Developers, Apps, Analyze, Admin, and Help. The main area is titled "API Products" and contains a table with one row of data. The table columns are DISPLAY NAME, ENVIRONMENTS, DESCRIPTION, and MODIFIED. The data row is: helloworld, prod. test, helloworld product for Apigee, 3 months ago. To the right of the table is a red arrow pointing to a blue "+ API Product" button.

2. 以下のフィールドに入力します。

注: 他のデベロッパーが誤って変更することのないように、{{your initials}} を自分のイニシャルに置き換えてください。たとえば、API プロダクトの Name を「xx\_Hipster-Products-API-Product」などにします。

- セクション: Product details

## Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

- Name: {{your initials}}\_Hipster-Products-API-Product
- Display name: {{your initials}}\_Hipster Products API Product
- Description: Product that provides access to the Hipster Products API.
- Environment: test
- Access: Public
- セクション: API resources
  - セクション: API Proxies
    - [Add a proxy] リンクをクリックします。

API Products > New Product

**Product details**

Name: VR\_Hipster-Products-API-Product

Access: Public

Description: Product that provides access to the Hipster Products /

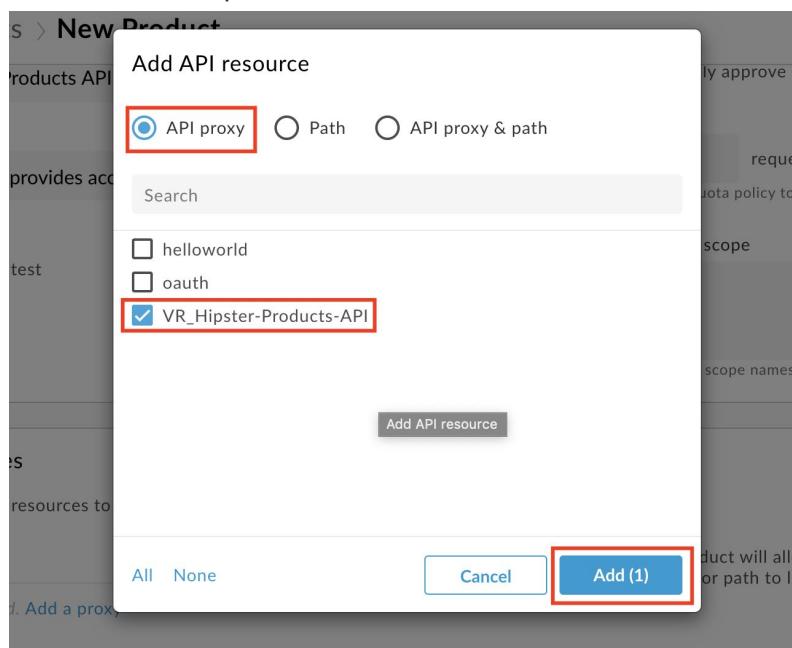
Environment: prod (unchecked), test (checked)

**API resources**

Specify the API resources to allow access to by adding the API proxies and paths to the API Product.

No proxies added. [Add a proxy](#)

- Hipster Products API プロキシを選択し、[Add] をクリックします。



# Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

- [Add a Path] リンクをクリックし、Hipster Products API プロキシの / オプションを選択して [Add] をクリックします。

API Products > New Product

Cancel Save

VR\_Hipster Products API Product

Description  
Product that provides access to the Hipster Products /

Environment  
 prod  test

Quota  
(Optional) requests every

Allowed OAuth scope  
(Optional)

Comma separated scope names

API resources

Specify the API resources to allow access to by adding the API proxies and paths to the API Product.

API proxies

This API Product will allow access to the following paths

/vr\_hipster-products-api/

No paths added. [Add a path](#)

Add API resource

API proxy  Path  API proxy & path

API Proxy: VR\_Hipster-Products-API

Revision: 1

Path: / (Supports the base path and all sub-URIs)

Cancel Add Add & Start another

# Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

### 3. API プロダクトを保存します。

API Products > New Product

The screenshot shows the 'New Product' configuration page. It includes fields for Product details (Name: VR\_Hipster-Products-API-Product, Display name: VR\_Hipster Products API Product, Description: Product that provides access to the Hipster Products /), Access (Public), and Quota (Optional). The 'Save' button at the top right is highlighted with a red arrow.

注: ここでは1つのAPIプロキシ全体をAPIプロダクトに追加しています。同じように簡単に1つ以上のAPIプロキシから1つ以上のオペレーションを選択し、APIプロダクトにまとめてバンドルできます。

## 管理UIからアプリを手動で登録し、APIキーを生成する

1. 横のナビゲーションメニューから [Publish] → [Developers] を選択し、[+Developer] ボタンをクリックします。

The screenshot shows the 'Developers' list page. It includes a search bar, a table with columns for NAME, EMAIL, USERNAME, APPS, and MEMBER SINCE, and a pagination indicator showing 1 - 3 of 3. The '+ Developer' button at the top right is highlighted with a red arrow.

2. 以下の詳細を入力します。

- First Name: {{your initials}}\_Test
- Last Name: Developer
- Username: {{your initials}}\_testdev
- Email: {{your initials}}\_testdev@test.com

3. [Create] をクリックします。

The screenshot shows the 'Create a Developer' form. It has fields for First Name (VR\_Test), Last Name (Developer), Username (vr\_testdev), and Email (vr\_testdev@test.com). The 'Create' button at the bottom right is highlighted with a red arrow.



4. 横のナビゲーションメニューから [Publish] → [Apps] を選択し、[+App] ボタンをクリックします。

The screenshot shows the Google Cloud Platform navigation bar. The 'Apps' menu item is highlighted with a red box and a red arrow pointing to it from the left. The 'Publish' menu item is also highlighted with a red box and a red arrow pointing to it from the top-left. The 'Apps' section is open, displaying a table with one row: 'helloApp' by 'helloworld dev', status 'approved', registered '3 months ago'. A red box highlights the '+ App' button in the top right corner of the table header.

5. 以下の詳細を入力します。

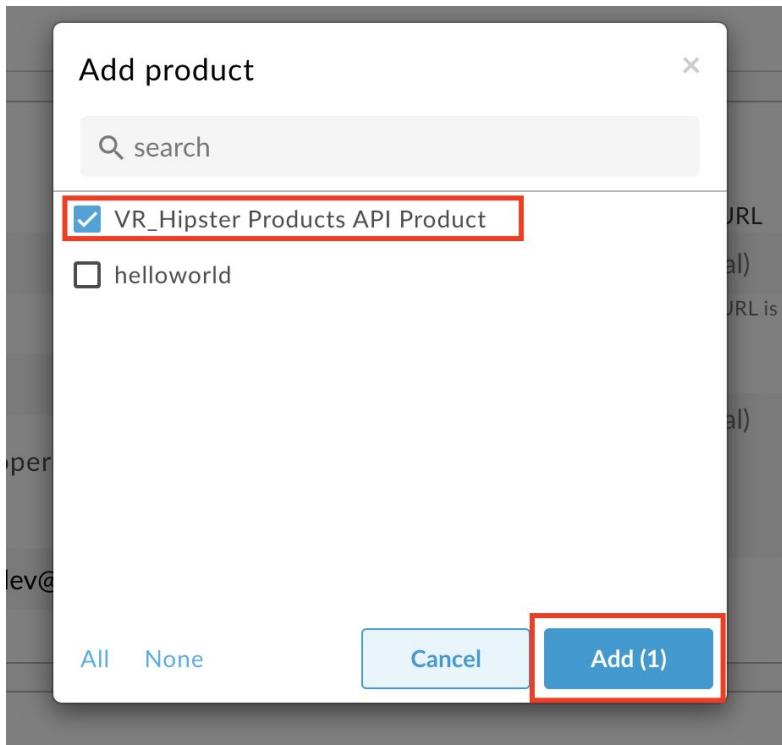
- Name: {{your initials}}\_Hipster-Products-App
- Display Name: {{your initials}}\_Hipster Products App
- [Developer] ラジオボタンを選択します。
- 先ほど作成したデベロッパーを選択します。
- [Add Product] ボタンをクリックします。

The screenshot shows the 'New App' creation form. The 'App Details' section is filled out with:

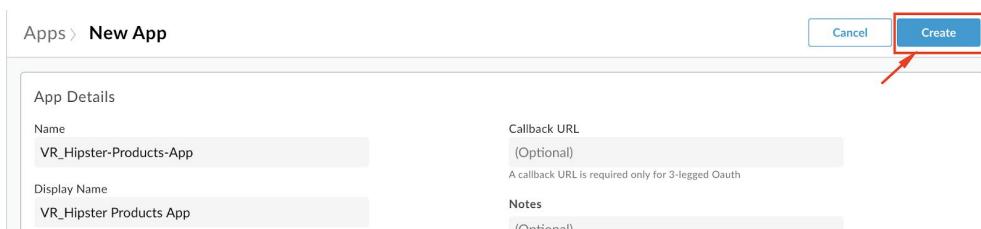
- Name: VR\_Hipster-Products-App
- Display Name: VR\_Hipster Products App
- Developer: VR\_Test Developer (vr\_testdev@test.com) (radio button selected)

The 'Credentials' section has an 'Expiry' dropdown set to 'Never'. Below the table, a note says 'At least one product is required.' and a red box highlights the 'Add product' button.

- 先ほど作成した API プロダクトを選択し、[Add] をクリックします。



6. [Create] ボタンをクリックして保存します。



7. アプリが作成されると、Apigee がこのアプリ用の API キーとシークレットのペアを生成します。キーの隣にある [Show] リンクをクリックし、この API キーをコピーしておきます。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

Apps > VR\_Hipster-Products-App

Edit Delete

#### App Details

Name	VR_Hipster-Products-App	App status	<span>Approved</span>
Display Name	VR_Hipster Products App	Callback URL	
Registered	Sep 16 2019 12:33:55 PM (UTC)	Notes	
Developer	VR_Test Developer (vr_testdev@test.com)		

#### Credentials

Status	<span>Approved</span>	<span>Copy this API Key to test your API</span>
Key	<span>Hide</span> <span>GnorPbx4l3jGqeNrHKTGRwEbbuRsGoIC</span>	
Secret	<span>Show</span> <span>*****</span>	
Issued	Sep 16 2019 12:33:55 PM (UTC)	
Expiry	Never	

これで、この API キーを使用して、API プロキシに有効な API リクエストを送れるようになりました。

## 有効な API キーで API をテストする

1. 横のメニューから [Develop] > [API Proxies] に移動し、API プロキシを開きます。ウィンドウの上部近くにある [Trace] タブをクリックし、[Start Trace Session] をクリックしてトレースセッションを開始します。
2. サンプル API 呼び出しリクエストを送信する前に、/products リソースパスを URL に追加します。さらに、コピーした API キーの値を指定した `apikey` というクエリ パラメータを追加します。[Send] をクリックしてリクエストを送信します。
3. Verify API Key ポリシーが API キーを有効であると確認したので、API リクエストから 200 OK レスポンスが戻ります。

The screenshot shows the Google Cloud API Proxies Trace interface for the VR\_Hipster-Products-App. The main pane displays a trace session with one transaction listed:

- Transactions:** 1 200 (Status: 200, Method: GET, URL: http://ravindrantrial-eval-test.apigee.net/vr\_hipster-products-api/products?apikey=GnorPbx4l3jGqeNrHKTGRwEbbuRsGoIC, Elapsed: 1470 ms)
- Send Requests:** A button labeled "Send" is shown next to the transaction details.
- Transaction Map:** A visual representation of the API flow, showing a sequence of nodes connected by arrows. One node is highlighted with a red box and a lock icon, indicating a security check like the Verify API Key policy.
- Phase Details:** A table showing the execution phases of the proxy request. It includes rows for "Proxy Request Flow Started" and "Verify API Key".
- Proxy Endpoint:** Details for the proxy endpoint, including name (default) and variables (apigee.api.product.name, unknown\_client\_id).

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

## ラボの動画

---

このトピックを取り上げている動画を視聴するには、ブラウザで[こちらを開いてください。](#)

## 理解度チェック

---

- Verify API Key ポリシーが、通常、リクエスト PreFlow で最初のポリシーになっているのはなぜですか？このポリシーが「すべての」 PreFlow ではなく、条件付きの PreFlow に含まれることがあるのはどのような場合ですか？
- デフォルトのクエリ パラメータの場所ではなく「Api-Key」というヘッダーから API キーを取得するには、どのようにポリシーを構成しますか？

## まとめ

---

このラボでは、API の提供側の詳細を API の利用側から隠蔽してくれる、API プロキシ、API プロダクト、アプリの関係と、Verify API Key ポリシーを使用して API プロキシを保護する方法について学びました。そしてポリシーを実装し、組み込みの Trace ツールを使用してテストを実施しました。

## 参考資料

---

- Apigee ドキュメントページへのリンク
  - Verify Api Key ポリシー  
<https://docs.apigee.com/api-platform/reference/policies/verify-api-key-policy>

## ラボ 3 - API 割り当て (Quota) を使って階層型 API プロダクトのサブスクリプションを管理する

所要時間: 15 分

登場人物: API プロダクトチームと API 開発チーム

## ユースケース

---

API 階層化はプロダクトとしての API の新しいトレンドです。階層化によってベースレベル（ブロンズなど）を無料オプションで提供します。このオファーはアップセルで提供するデー

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

タを活用するためのエントリ ポイントです。目標は追加の機能レベルにアップセルすることです。これは「フリーミアム」としても知られています。フリーミアムは非常に基本的なアプローチを採用しています。つまり、基本的な機能か、呼び出しの割り当てを入門レベルとして提供し、さらなるデータへのアクセスまたは機能を必要とする場合は有料で提供します。これにより、デベロッパーは本格的な購入を決定する前に動作するプロトタイプを作り、実際的なシナリオに基づいて API を検討できます。

## Apigee の利点

---

Apigee は API プロキシの機能的なロジックから抽象化された API プロダクトの概念を提供します。プロキシは API プロダクトで定義される制限値にアクセスできます。これにより、API 開発チームがこれらの制限値を使ってパラメータ化された動作を実装する一方で、API プロダクトチームは（プロダクトごとの割り当ての設定などの）ビジネス ロジックに注力できます。

## 前提条件

---

[ラボ 1](#) と [ラボ 2](#) を完了している場合、受講の前提条件を満たしています。

このラボを受講するためには、少なくとも「Verify API Key」ポリシーを持つデプロイ済の API プロキシと、API プロダクトにアプリを登録できるデベロッパーが必要です。

## 手順

---

このラボでは、付与される割り当ては異なるが同一の API プロキシをバンドルする、さまざまな API プロダクトを作成します。

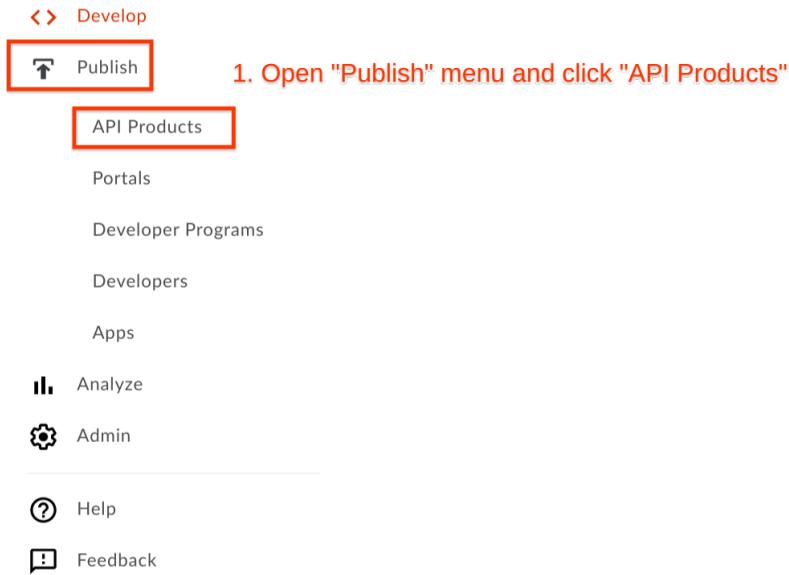
## API プロダクト（ブロンズ、シルバー、ゴールド）を作成する

---

- <https://apigee.com/edge> にアクセスしてログインします。これは Edge 管理 UI です。
- [Publish] → [API Products] を選択します。
- [+API Product] をクリックします。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)



The screenshot shows a list titled 'API Products'. At the top right is a blue button labeled '+ API Product' (highlighted with a red box). Below it, a message says '2. Click "+ API Product" 1 - 1 of 1 < >'.

- 以下のフィールドに入力します。
  - セクション: Product details
    - Name: {yourInitials}\_Hipster-Products-API-Product-Bronze
    - Display name: {yourInitials}\_Hipster Products API Product Bronze
    - Description: Free version of the Hipster Product API
    - Environment: test
    - Access: Public
    - Quota: 5 requests every 1 Minute
  - セクション: API resources
    - [Add a proxy] リンクをクリックします。

[Cancel](#) [Save](#)

**Product details**

Name: VR\_Hipster-Products-API-Product-Bronze

Display name: VR\_Hipster Products API Product Bronze

Description: Free version of the Hipster Product API

Environment:  prod  test

**1. Populate product attributes**

Access: Public

Visible to any registered developer during app registration

Automatically approve access requests

Quota: 5 requests every 1 Minute

You must add a Quota policy to the API proxy to enforce a quota based on these values. [Learn more](#)

Allowed OAuth scope: (Optional)

---

**API resources**

Specify the API resources to allow access to by adding the API proxies and paths to the API Product.

**API proxies**

No proxies added [Add a proxy](#)

**Paths**

- {yourInitials}\_Hipster-Products-API Proxy を選択して [Add] をクリックします。

### Add API resource

API proxy    Path    API proxy & path

Search

fg\_Hipster-Products-API

helloworld

oauth

**1. Select your proxy**

**2. Click "Add"**

All None [Cancel](#) [Add \(1\)](#)

API プロダクトには、許可する一定期間あたりのリクエスト数（1秒あたり 5 件のリクエストなど）を構成できる「Quota」という一連のフィールドがあります。これを構成するだけで割り

## Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

当てが適用されるわけではありません。これは、次の手順で定義する Quota ポリシーが動的に読み込む定義にすぎません。

- [Save] をクリックして API プロダクトを作成します。

続いて、異なる Quota 設定を持つ、シルバーとゴールド プロダクトを表す 2 つのよく似たプロダクトを作成します。以下の手順を実行すると、別の API プロダクトが作成されます。

- [Publish] → [API Products] をクリックします。
- [+API Product] をクリックします。
- シルバー プロダクトの場合は、以下のフィールドに入力します。
  - セクション: Product details
    - Name: {yourInitials}\_Hipster-Products-API-Product-Silver
    - Display name: {yourInitials}\_Hipster Products API Product Silver
    - Description: Basic version of the Hipster Product API
    - Environment: test
    - Access: Public
    - Quota: 20 requests every 1 Minute
  - セクション: API resources
    - [Add a proxy] リンクをクリックします。
    - {yourInitials}\_Hipster-Products-API Proxy を選択して [Add] をクリックします。
- ゴールド プロダクトの場合は、以下のフィールドに入力します。
  - セクション: Product details
    - Name: {yourInitials}\_Hipster-Products-API-Product-Gold
    - Display name: {yourInitials}\_Hipster Products API Product Gold
    - Description: Deluxe version of the Hipster Product API
    - Environment: test
    - Access: Public
    - Quota: 1000 requests every 1 Minute
  - セクション: API resources
    - [Add a proxy] リンクをクリックします。
    - {yourInitials}\_Hipster-Products-API Proxy を選択して [Add] をクリックします。

これで、プロダクト階層戦略にマッチした、3 つの API プロダクトが作成されました。

API Products

+ API Product

All ▾				1 - 4 of 4 < >
DISPLAY NAME ▾	ENVIRONMENTS	DESCRIPTION	MODIFIED	
helloworld	prod, test	helloworld product for Apigee	3 months ago	
VR_Hipster Products API Pr...	test	Free version of the Hipster Product API	9 minutes ago	
VR_Hipster Products API Pr...	test	Deluxe version of the Hipster Product API	6 minutes ago	
VR_Hipster Products API Pr...	test	Basic version of the Hipster Product API	a few seconds ago	

## 各プロダクト用のアプリを作成する

- [Publish] → [Apps] を選択します。
- [+API Product] をクリックします。

< > Develop

**Publish**

API Products

Portals

Developer Programs

Developers

**Apps**

1. Open "Publish" menu and click "Apps"

Analyze

Admin

Help

Feedback

**+ App**

2. Click "+ App"

All ▾	NAME ▾	DEVELOPER	APP STATUS	REGISTERED SINCE
Pending product approval only				

- 以下のフィールドに入力します。
  - セクション: App Details

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

- Name: {yourInitials}\_Hipster Android App Free
- Display name: {yourInitials}\_Hipster Android App Free
- Developer: 既存のデベロッパーを選択します。
- セクション: Credentials
  - [Add product] をクリックします。

Apps > New App

Cancel
Create

**App Details**

Name fg_Hipster Android App Free	Callback URL (Optional) A callback URL is required only for 3-legged OAuth
Display Name fg_Hipster Android App Free	1. Fill in the values Notes (Optional)
<input type="radio"/> Company <input checked="" type="radio"/> Developer	
Developer helloworld dev (helloworld@apigee.com)	

**Credentials**

Expiry	Product	Status
Never		

At least one product is required.

Add product 2. Click "Add product"

- Hipster Product API Product Bronze を選択して [Add] をクリックします。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

Add product X

search

fg\_Hipster Products API Product Gold

fg\_Hipster Products API Product Silver

fg\_Hipster Products API Product Bronze

**1. Select API product**

**2. Click "Add"**

All None Cancel Add (1)

- [Create] をクリックしてアプリを作成します。
- アプリのプロパティで [Show] をクリックし、後から使用する場合に備えてキーを記録しておきます。

Apps > fg\_Hipster Android App Free Edit Delete

App Details	
Name	fg_Hipster Android App Free
Display Name	fg_Hipster Android App Free
Registered	Aug 12 2019 1:59:48 PM (UTC)
Developer	helloworld dev (helloworld@apigee.com)
App status <span style="color: green;">Approved</span>	
Callback URL	
Notes	

Credentials	
Status	<span style="color: green;">Approved</span>
Key	<span style="border: 2px solid red; padding: 2px;">Show</span> *****
Secret	<span style="border: 2px solid red; padding: 2px;">Show</span> *****
Issued	Aug 12 2019 1:59:48 PM (UTC)
Expiry	Never
Product Status	
fg_Hipster Products API Product Bronze <span style="color: green;">Approved</span>	

以下の値を使用して、シルバー枠とゴールド枠を使用するアプリでプロセスを繰り返します。

- シルバー API プロダクトを使用するアプリ
  - セクション: App Details

## Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

- Name: {yourInitials}\_Hipster Android App Basic
  - Display name: {yourInitials}\_Hipster Android App Basic
  - Developer: 既存のデベロッパーを選択します。
- セクション: Credentials
  - [Add product] をクリックします。
  - Hipster Product API Product Silver を選択して [Add] をクリックします。
- ゴールド API プロダクトを使用するアプリ
  - セクション: App Details
    - Name: {yourInitials}\_Hipster Android App Deluxe
    - Display name: {yourInitials}\_Hipster Android App Deluxe
    - Developer: 既存のデベロッパーを選択します。
  - セクション: Credentials
    - [Add product] をクリックします。
    - Hipster Product API Product Gold を選択して [Add] をクリックします。

これで異なる3つのAPIキーを持つアプリが3つ作成されました。キーは記録しておいてください。

Apps			
All	Pending product approval only	1 - 3 of 3	< >
NAME	DEVELOPER	APP STATUS	REGISTERED SINCE
fg_Hipster Android App Basic	helloworld dev	approved	a minute ago
fg_Hipster Android App Deluxe	helloworld dev	approved	a few seconds ago
fg_Hipster Android App Free	helloworld dev	approved	14 minutes ago

## Quota ポリシーを作成して構成する

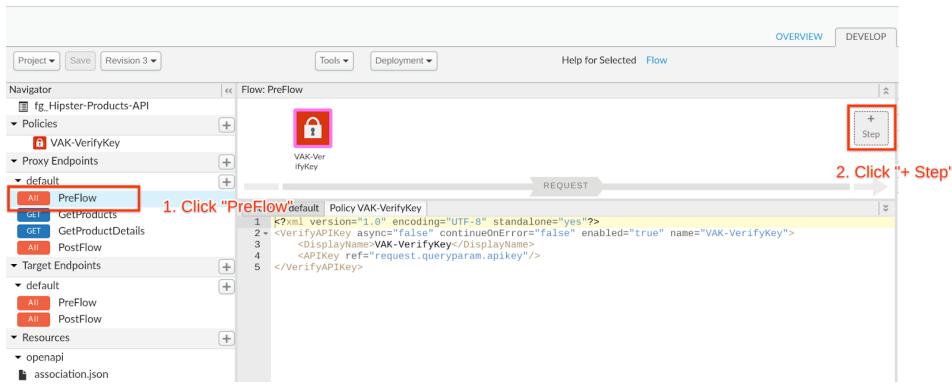
上記で説明したように、Quota ポリシーをプロキシに追加しないと割り当てが適用されません。API プロダクトの Quota フィールドの構成により、プロキシ内で使える必要な変数が設定され、Quota ポリシーで読み込めるようになります。

1. 横のナビゲーションメニューから [Develop] → [API Proxies] をクリックします。前提条件にあった、既存の API プロキシを開きます。
2. Verify API Key のポリシーが正しい名前で存在していることを確認します。ポリシーネームをクリックして下記の XML 構成を確認します。



3. [PreFlow] と [+ Step] をクリックし、新しいポリシーを追加します。

API Proxies > fg\_Hipster-Products-API > Develop > 3



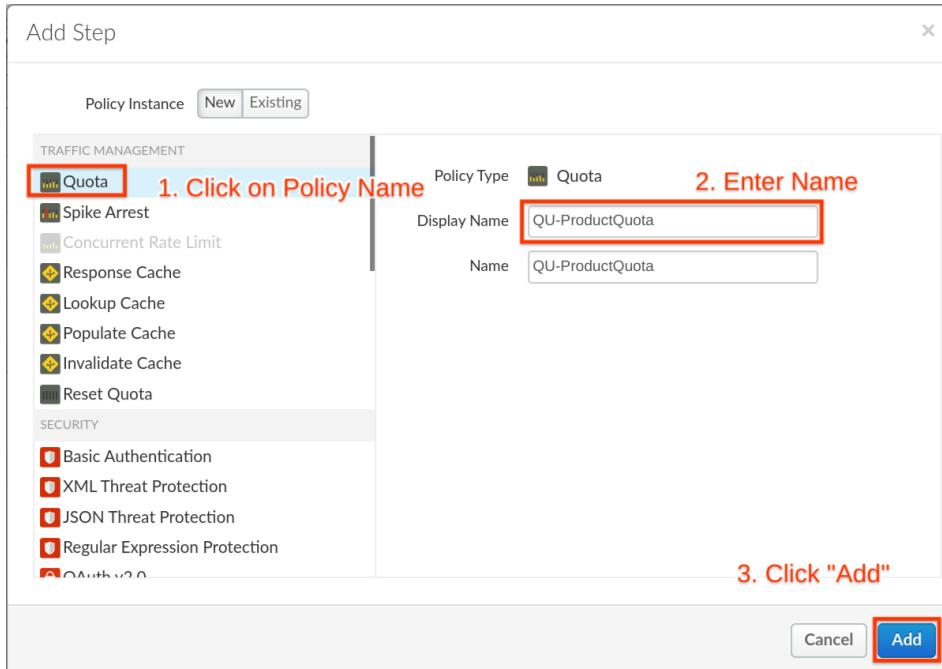
4. [Quota] ポリシーをクリックし、以下のフィールドに入力します。

- i. Display Name: QU-ProductQuota

[Add] をクリックしてフローにポリシーを追加します。

# Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)



5. 前提条件の VAK-VerifyKey で構成した VerifyAPIKey ポリシーにより、API キーが検証されます。API キーが、[Quota] フィールドに「1分あたり 3 件のリクエスト」と設定された API プロダクトを含むため、検証後に以下の変数が設定されます。

```
verifyapikey.VAK-VerifyKey.apiproduct.developer.quota.limit = 3
verifyapikey.VAK-VerifyKey.apiproduct.developer.quota.interval = 1
verifyapikey.VAK-VerifyKey.apiproduct.developer.quota.timeunit = minute
```

次に、上記の変数を参照し、ポリシー構成でこのコードを使用する QU-ProductQuota という Quota ポリシーを設定します。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Quota async="false" continueOnError="false" enabled="true"
name="QU-ProductQuota" type="calendar">
  <DisplayName>QU-ProductQuota</DisplayName>
  <Allow count="3" />
  <countRef="verifyapikey.VAK-VerifyKey.apiproduct.developer.quota.limit"/>
  <Interval
    ref="verifyapikey.VAK-VerifyKey.apiproduct.developer.quota.interval">1</Interval>
  <TimeUnit
    ref="verifyapikey.VAK-VerifyKey.apiproduct.developer.quota.timeunit">minute</TimeUnit>
  <Identifier ref='verifyapikey.VAK-VerifyKey.client_id'/>
  <Distributed>true</Distributed>
```

## Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

```

<Synchronous>true</Synchronous>
<StartTime>2019-01-01 12:00:00</StartTime>
</Quota>

```

The screenshot shows the Apigee Studio interface with the following details:

- Policies:** A policy named "QU-ProductQuota" is selected.
- Type:** Quota
- Display Name:** QU-ProductQuota
- Code:**

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Quota async="false" continueOnError="false" enabled="true" name="QU-ProductQuota" type="calendar">
  <DisplayName>QU-ProductQuota</DisplayName>
  <Allow count="3" countRef="verifyapikey.VAK-VerifyKey.apiproduct.developer.quota.limit"/>
  <Interval ref="verifyapikey.VAK-VerifyKey.apiproduct.developer.quota.interval">1</Interval>
  <TimeUnit ref="verifyapikey.VAK-VerifyKey.apiproduct.developer.quota.timeunit">minute</TimeUnit>
  <Identifier ref="verifyapikey.VAK-VerifyKey.client_id"/>
  <Distributed>true</Distributed>
  <Synchronous>true</Synchronous>
  <StartTime>2019-01-01 12:00:00</StartTime>
</Quota>

```

注: API プロダクトで [Quota] フィールドが 設定されていない場合、ここで設定された値である「1分あたり 3 件の呼び出し」が使用されます。

6. 前の手順でポリシーを変更したら、[Save] をクリックします。

API Proxies > fg\_Hipster-Products-API > Develop > 3

The screenshot shows the Apigee Studio interface with the following details:

- Project:** fg\_Hipster-Products-API
- Revision:** Revision 3
- Tools:** Tools
- Deployment:** Deployment
- Policies:** A policy named "QU-ProductQuota" is selected.
- Type:** Quota
- Display Name:** QU-ProductQuota
- Name:** QU-ProductQuota

## 結果をテストする

API プロキシに移動し、[Trace] タブでいくつかのテストを実行できます。

- [Trace] をクリックします。
- [Start Trace Session] をクリックします。
- ブロンズ apikey 値を URL にクエリ パラメータとして追加します（例: [http://{{yourapigeeorg}}-test.apigee.net/v1/{{yourInitials}}\\_hipster-products-api/products?apikey=GYuZekimsQ2TLdWWMHkqB1poAquHaGsv](http://{{yourapigeeorg}}-test.apigee.net/v1/{{yourInitials}}_hipster-products-api/products?apikey=GYuZekimsQ2TLdWWMHkqB1poAquHaGsv)）。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

- [Send] ボタンを複数回クリックしてテストを行います。

API Proxies > fg\_Hipster-Products-API > Trace > 4

1. Go to "Trace" tab  
2. Start new trace session  
3. Put in url with apikey  
4. Click "Send"

- 呼び出しが 6 回に達し、Quota の 5 回の無料呼び出しを超過すると、Quota ポリシーに赤の感嘆符記号が表示されます。

API Proxies > fg\_Hipster-Products-API > Trace > 4

Request ID	Status	Method	URI	Elapsed
6	429	GET	/v1/fg_hipster-products/api/products?apikey=K0mARBeVqamXadWSx4P2V30ZtbM	33 ms
5	200	GET	/v1/fg_hipster-products/api/products?apikey=K0mARBeV...	15 ms
4	200	GET	/v1/fg_hipster-products/api/products?apikey=K0mARBeV...	219 ms
3	200	GET	/v1/fg_hipster-products/api/products?apikey=K0mARBeV...	41 ms

次に API プロダクトを切り替え、アプリのシルバー apikey 値を URL にクエリ パラメータとして追加します（例：

[http://yourapigeeorg-test.apigee.net/v1/\[yourInitials\]\\_hipster-products-api/products?apikey=GYuZekimsQ2TLdWWMHkqb1poAquHaGsv](http://yourapigeeorg-test.apigee.net/v1/[yourInitials]_hipster-products-api/products?apikey=GYuZekimsQ2TLdWWMHkqb1poAquHaGsv)）。

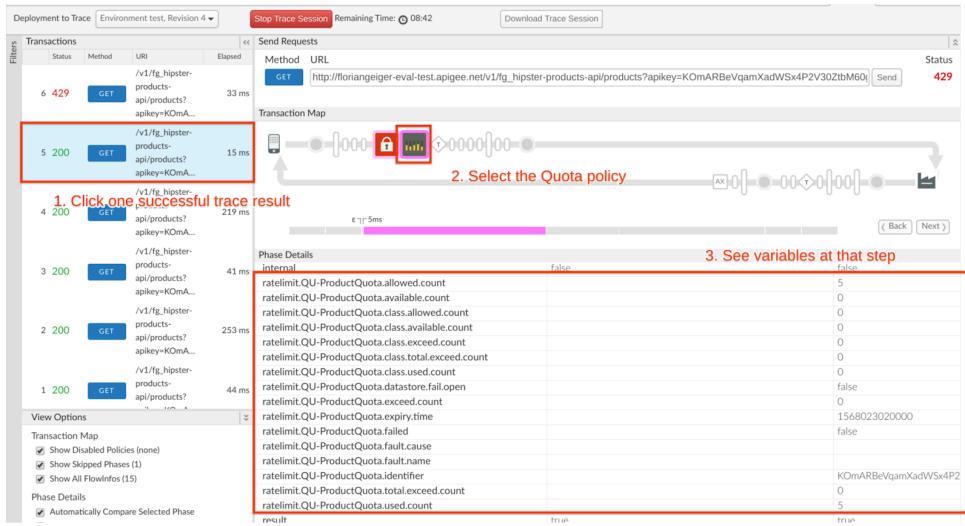
- apikey パラメータをシルバーアプリの認証情報に一致するように変更します。
- [Start Trace Session] をクリックする前に、[Send] ボタンを 15 回前後クリックしてテストを行います。
- トレース セッションを開始し、上限に到達する前に [Send] ボタンを再び複数回クリックします。

トレース結果を確認してみましょう。

- 左側に表示されている成功したトレース結果（ステータスが緑色の 200 のもの）の 1 つをクリックします。
- [Transaction Map] の Quota ポリシー アイコンをクリックします。

# Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)



- 最後の呼び出しでは、元の 20 回 (ratelimit.QU-ProductQuota.allowed.count) のうち、1回だけが使用可能 (ratelimit.QU-ProductQuota.available.count) であることがわかります。
- プロキシフローに含まれる使用可能な他の変数についても見てみましょう。

ここでは、デラックス または ゴールド バージョンのプロダクトについては省略しますが、デベロッパーがこのバージョンのプロダクトの上限には簡単に到達しないことがわかるでしょう。

## 理解度チェック

- Quota ポリシーで [Identifier] タグを省略すると、どうなるでしょうか？

## まとめ

このラボでは、階層化モデルを提供し、プロダクトごとに異なる割り当てを付与するという、API プロダクト戦略に沿った 3 つのプロダクトを作成しました。API プロキシ内では制限を定義せず、API プロダクトで割り当て量を定めることで、同じプロキシを利用できるようにしました。

## 参考資料

### Apigee ドキュメントへのリンク

**Google Cloud**

For more information visit [google.com/cloud](https://google.com/cloud)

<https://docs.apigee.com/api-platform/reference/policies/quota-policy>

## 動画（4mv4d）

---

[https://www.youtube.com/watch?v=z8Rj\\_VzSbh4](https://www.youtube.com/watch?v=z8Rj_VzSbh4)

<https://www.youtube.com/watch?v=1RDDpH0wOdc>

---



For more information visit [google.com/cloud](http://google.com/cloud)

# ラボ 4 - Apigee 統合デベロッパー ポータルを使用したア プリ デベロッパー エクスペリエンスの構築

所要時間: 30 分

登場人物: API チーム

## ユースケース

デベロッパー ポータルから API プロダクトを利用するアプリ デベロッパー向けに、簡単なセルフサービス オンボーディング エクスペリエンスを提供し、管理することを目指します。アプリ デベロッパーが API について学び、利用登録し、API を使い始められるようにすると同時に、さまざまな API プロダクトの可視性とアクセスを管理します。

## Apigee Edge の利点

Apigee Edge にはデベロッパー ポータル向けに複数のオプションが用意されています。Apigee は、シンプルなターンキー ソリューションから、完全にカスタマイズ可能で拡張可能なソリューションまで、多様なデベロッパー ポータル ソリューションをサポートしています。ターンキー統合デベロッパー ポータル オプションは、テーマ、ロゴ、ページコンテンツなど、サイトの多くのブランディングとカスタマイズをサポートします。このオプションは管理 UI から直接、数秒で公開できます。Drupal ベースのポータルも提供しており、こちらではより完全な管理と、Drupal マーケットの何百もの Drupal モジュールを活用が可能です。このラボでは統合デベロッパー ポータルを取り上げます。

## デベロッパー プログラム、チーム、オーディエンスの管理

Apigee Edge では、[デベロッパー プログラム](#)はデベロッパー ポータルごとに関連付けられる構成セットです。具体的には、アプリのデベロッパー アカウント、アプリのデベロッパー ID 管理構成、アプリのデベロッパー チーム、ポータルに公開されるコンテンツにアクセスするためのオーディエンス構成が含まれます。

アプリのデベロッパーは、他のデベロッパーとアプリの責任を共有する[チーム](#)を作成することもできます。チーム内の各デベロッパーには、共有するアプリへのアクセスレベルを定義するロール（オーナー、アプリの管理者、閲覧者）が割り当てられます。

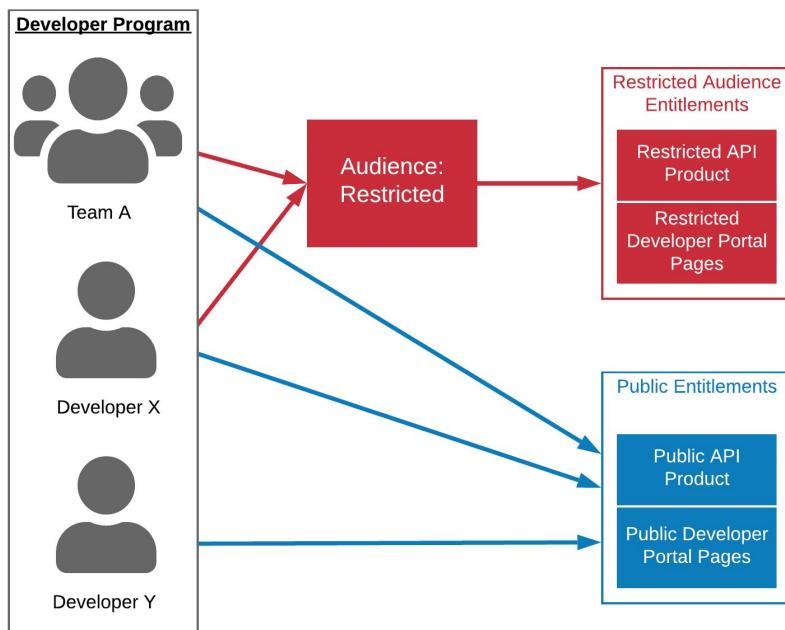
Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

オーディエンス構成は、次のリソースへのアクセス権を管理するため、ユーザーまたはデベロッパー チームを分割するために使用されます。

- ポータル内のページ
- 公開済みの API プロダクト

次の図は、一連のリソースへのアクセス権を管理するためにオーディエンスがどのように使用されているかを示しています。



このラボでは、API プロダクトの公開先である統合デベロッパー ポータルを作成します。このポータルで、アプリのデベロッパーは以下のことが可能です。

- OpenAPI 仕様ベースのインターフェイクティブなドキュメントで API の利用方法を学習
- API プロダクトを利用するアプリの登録
- API 利用のためのアプリのクライアント認証情報（API キーとシークレット）を生成

## 前提条件

このモジュールの [ラボ 1](#)、[ラボ 2](#)、[ラボ 3](#) の受講を完了している必要があります。

以下の準備が必要です。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

- 組織内の仕様ストアにOpenAPI仕様をアップロードしておきます。この仕様からAPIのドキュメントを生成します。このラボで使用できるOpenAPI仕様がない場合は、[ラボ1 - OpenAPI仕様によるAPIプロキシの設計と作成](#)をもう一度受講してください。
- APIプロキシをバンドルしたAPIプロダクト。APIプロダクトが構成されていない場合は、[ラボ2 - Apigee EdgeでのAPIセキュリティとAPIプロデューサー/コンシューマの関係](#)と[ラボ3 - API呼び出し割り当てから階層型APIプロダクトのサブスクリプションを管理する](#)をもう一度受講してください。

## 手順

---

### CORSサポートのためにAPIプロキシを更新する

CORS（クロスオリジンリソースシェアリング）は、非オリジンドメインのリソースとやり取りするウェブページで実行される、JavaScript XMLHttpRequest（XHR）呼び出しを許可するための標準のメカニズムです。CORSは、あらゆるブラウザで実施されている同一オリジンポリシーに対して一般的に実装されているソリューションです。たとえば、ブラウザで実行されるJavaScriptコードからAPIプロキシにXHR呼び出しを行うと、この呼び出しが失敗します。これは、ブラウザにそのページを提供しているドメインが、APIを提供しているドメイン（{your org name}-{environment name}.apigee.netなど）と同じでないためです。CORSは、クロスオリジンリソースシェアリングの提供を希望しているサーバーに「オプトイン」することを許可することで、この問題に対するソリューションを提供します。

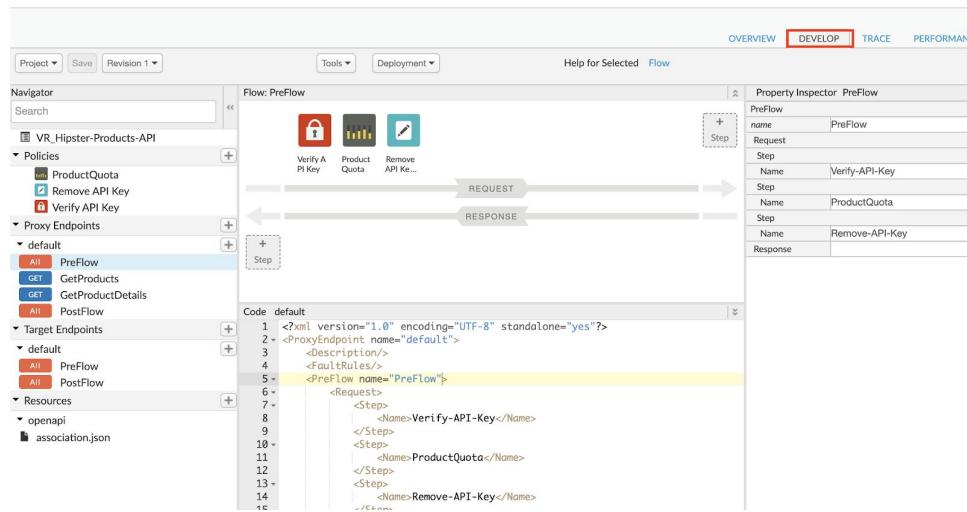
この手順では、APIプロキシがインタラクティブなドキュメントページ内で呼び出されるデベロッパー ポータルに公開される前に、APIプロキシに対してCORSサポートを実装している必要があります。

詳細については、[APIプロキシにCORSサポートを追加する](#)をご覧ください。

- [Develop] → [API Proxies] に移動し、APIプロキシ "{your initials}\_Hipster-Products-API" を選択します。このプロキシの[Develop]タブを開き、

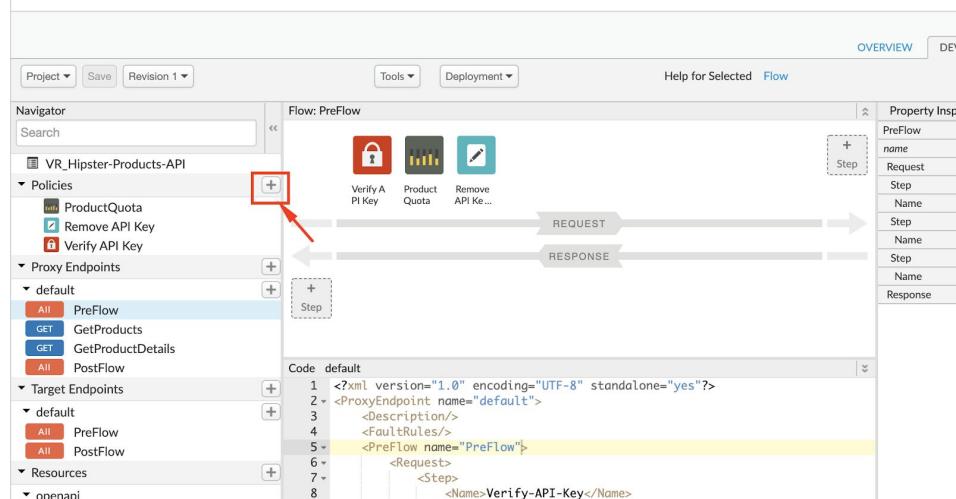
## プロキシ構成を編集します。

API Proxies > VR\_Hipster-Products-API > Develop > 1



2. ポリシーの左側のパネルで [+] オプションを選択し、新しいポリシーを追加します。

API Proxies > VR\_Hipster-Products-API > Develop > 1

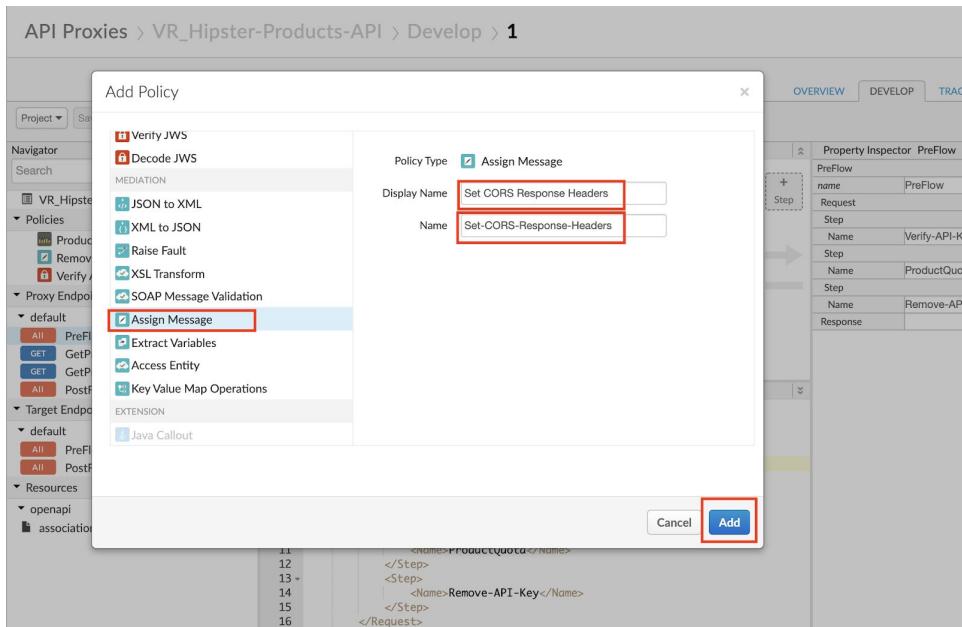


[Assign Message] オプションを選択し、次の詳細のポリシーを追加します。

Display Name: Set CORS Response Headers Name: Set-CORS-Response-Headers

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)



次のようにポリシー構成を編集します。

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<AssignMessage async="false" continueOnError="false" enabled="true"
name="Set-CORS-Response-Headers">
<DisplayName>Set CORS Response Headers</DisplayName>
<FaultRules/>
<Properties/>
<Set>
<Headers>
<Header name="Access-Control-Allow-Origin">{request.header.origin}</Header>
<Header name="Access-Control-Allow-Headers">origin, x-requested-with, accept,
content-type</Header>
<Header name="Access-Control-Max-Age">3628800</Header>
<Header name="Access-Control-Allow-Methods">GET, PUT, POST, DELETE</Header>
</Headers>
</Set>
<IgnoreUnresolvedVariables>true</IgnoreUnresolvedVariables>
<AssignTo createNew="false" transport="http" type="response"/>
</AssignMessage>
...

```

# Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

The screenshot shows the API Platform interface with the following details:

- Project:** API Proxies > VR\_Hipster-Products-API > Develop > 1
- Toolbar:** OVERVIEW, DEVELOP, TRACE, PERF
- Navigator:** Policies > Set CORS Response Headers (highlighted with a red box)
- Policy Configuration:**
  - Type: AssignMessage
  - Display Name: Set CORS Response Headers
  - Name: Set-CORS-Response-Headers
  - Attached to: Not used in any flows
  - Code (highlighted with a red box):

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<AssignMessage async="false" continueOnError="false" enabled="true" name="Set-CORS-Response-Headers">
<DisplayName>Set CORS Response Headers</DisplayName>
<Properties/>
<Set>
<Headers>
<Header name="Access-Control-Allow-Origin">{request.header.origin}</Header>
<Header name="Access-Control-Allow-Headers">origin, x-requested-with, accept, content-type</Header>
<Header name="Access-Control-Max-Age">3628800</Header>
<Header name="Access-Control-Allow-Methods">GET, PUT, POST, DELETE</Header>
</Headers>
</Set>
<IgnoreUnresolvedVariables>true</IgnoreUnresolvedVariables>
<AssignTo createNew="false" transport="http" type="response"/>
</AssignMessage>

```
  - Property Inspector (right side):
    - AssignMessage
    - async: false
    - continueOnError: false
    - enabled: true
    - name: Set-COF
    - displayName: Set COF
    - FaultRules
    - Properties
    - Set
    - Headers
    - Header [request]
    - name: Access-\*
    - Header origin, x-
    - name: Access-\*
    - Header 3628800
    - name: Access-\*
    - Header GET, PU
    - name: Access-\*
    - ignoreUnresolvedVariables: true
    - assignTo
    - createNew: false
    - transport: http
    - type: response

3. 上記の手順 2 と 3 を繰り返し、次の詳細の Assign Message ポリシーを追加します。

Display Name: Set OPTIONS Status Code Name: Set-OPTIONS-Status-Code

Policy configuration:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<AssignMessage async="false" continueOnError="false" enabled="true" name="Set-OPTIONS-Status-Code">
<DisplayName>Set OPTIONS Status Code</DisplayName>
<Properties/>
<AssignVariable>
<Name>error.status.code</Name> <Value>200</Value>
</AssignVariable>
<IgnoreUnresolvedVariables>true</IgnoreUnresolvedVariables>
<AssignTo createNew="false" transport="http" type="request"/>
</AssignMessage>

```

4. 左側のパネルの [Target Endpoint] -> [default] の下にある [PreFlow] フローを選択します。次の図のように、[Set CORS Response Headers] ポリシーをフローのレスポンスパイプラインにドラッグ & ドロップします。

The screenshot shows the 'Develop' tab of the API Proxy configuration. On the left, the 'Navigator' pane lists 'VR\_Hipster-Products-API' with its 'Policies' (ProductQuota, Remove API Key, Set CORS Response Headers, Set OPTIONS Status Code, Verify API Key), 'Proxy Endpoints' (default, PreFlow, GET GetProducts, GET GetProductDetails, POST PostFlow), and 'Target Endpoints' (default, PreFlow, PostFlow). The 'Set CORS Response Headers' policy is selected and highlighted with a red box. In the center, the 'Flow: PreFlow' editor shows a sequence of REQUEST and RESPONSE steps. A red arrow points from the selected policy to the REQUEST step, with the text '2. Drag and drop the policy here.' To the right, the 'Property Inspector' pane shows the 'PreFlow' settings: name = PreFlow, Request, Response, Step, Name = Set-CORS-Response-Headers.

これであらゆる有効な API 呼び出しで CORS ヘッダーが返されるようになります。

- [+] ボタンを選択して左側のパネルの [Proxy Endpoint] -> [default] にフローを追加します。

The screenshot shows the 'Develop' tab of the API Proxy configuration. The 'Proxy Endpoint: default' settings are displayed: Name = default, Description = , Base Path = /v1/vr\_hipster-products-api. The 'Proxy Endpoints' section shows 'default' expanded, with 'PreFlow' selected and highlighted with a red box. A red box highlights the '[+]' button next to 'PreFlow'. The 'Property Inspector' pane shows the 'default' settings: ProxyEndpoint, name = default, Description, FaultRules, PreFlow, name = PreFlow, Request, Step, Name = Verify-API-Key, Step, Name = ProductQuota, Step, Name = Remove-API-Key, Response, PostFlow, name = PostFlow, Request, Response, Flows, Flow, name = GetProducts, Description, Request, Response, Condition = (proxy.path == '/v1/vr\_hipster-products-api'), Flow, name = GetProductDetails, Description, Request, Response, Condition = (proxy.path == '/v1/vr\_hipster-products-api'), Flow, name = GetProductDetail, Description, Request, Response, Condition = (proxy.path == '/v1/vr\_hipster-products-api'), Flow, name = GetProductDetail, Description, Request, Response, Condition = (proxy.path == '/v1/vr\_hipster-products-api').

- [Manual] エントリタブを選択し、以下のフロー詳細を入力します。

Flow Name: OptionsPreFlight  
 Description: For CORS preflight requests  
 Condition Type: Custom Condition: request.verb == "OPTIONS"

API Proxies > VR\_Hipster-Products-API > Develop > 1

New Conditional Flow

Manual  From OpenAPI

Flow Name: OptionsPreFlight

Description: For CORS preflight requests

Condition Type:  Custom  Path  Path and Verb

Condition: request.verb == "OPTIONS"

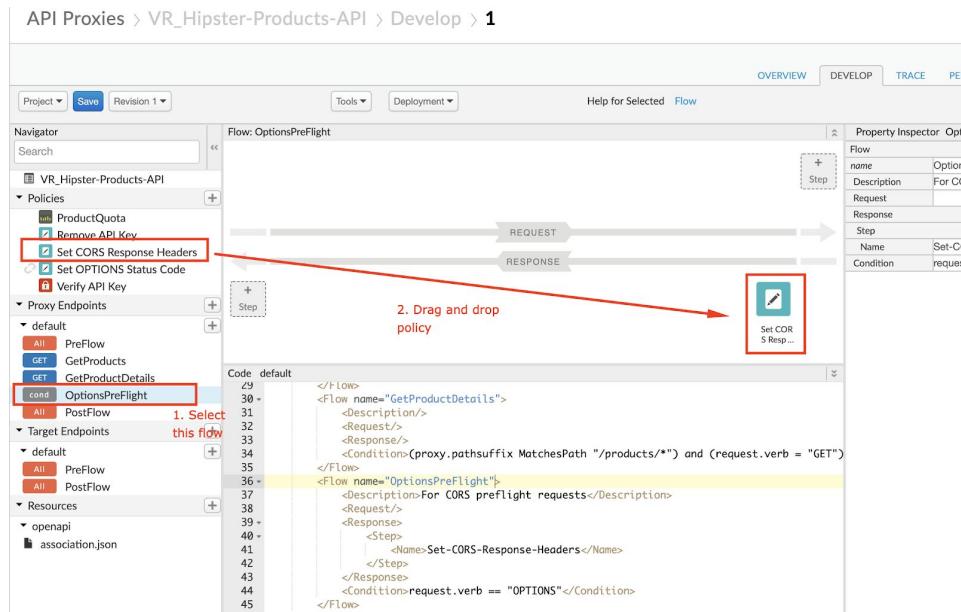
Add

Code default

[Add] をクリックします。

このフローは CORS Preflight OPTIONS リクエストを処理します。

7. 左側のパネルの [Proxy Endpoint] -> [default] の下にある [OptionsPreFlight] フローを選択します。次の図のように、[Set CORS Response Headers] ポリシーをフローのレスポンス パイプラインにドラッグ & ドロップします。



これで CORS Preflight OPTIONS リクエストに対して CORS ヘッダーが返されます。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

左側のパネルの [Proxy Endpoint] -> [default] を選択します。構成の下部までスクロールダウンし、'default' ルールの前に、以下のような RouteRule 設定を追加します。

```

<RouteRule name="NoRoute">
    <Condition>request.verb == "OPTIONS"</Condition>
</RouteRule>

```

```

<RouteRule name="NoRoute">
    <Condition>request.verb == "OPTIONS"</Condition>
</RouteRule>

```

この RouteRule により、CORS Preflight OPTIONS リクエストは API ターゲットサービスに転送されなくなります。

左側のパネルの [Proxy Endpoint] -> [default] を選択します。構成の最上部にスクロールし、次の図のように、DefaultFaultRule 設定を追加します。

```

<DefaultFaultRule name="DefaultFaultRule">
    <Step>
        <Name>Set-CORS-Response-Headers</Name>
    </Step>
    <Step>
        <Name>Set-OPTIONS-Status-Code</Name>
        <Condition>request.verb = "OPTIONS"</Condition>
    </Step>
    <AlwaysEnforce>true</AlwaysEnforce>
</DefaultFaultRule>

```

これで、API プロキシエラーが起きた場合でも、CORS ヘッダーが正しく返され、CORS Preflight OPTIONS リクエストは常に処理されます。たとえば、API キーの検証が失敗した場合などでもです。

## OpenAPI 仕様を更新する

---

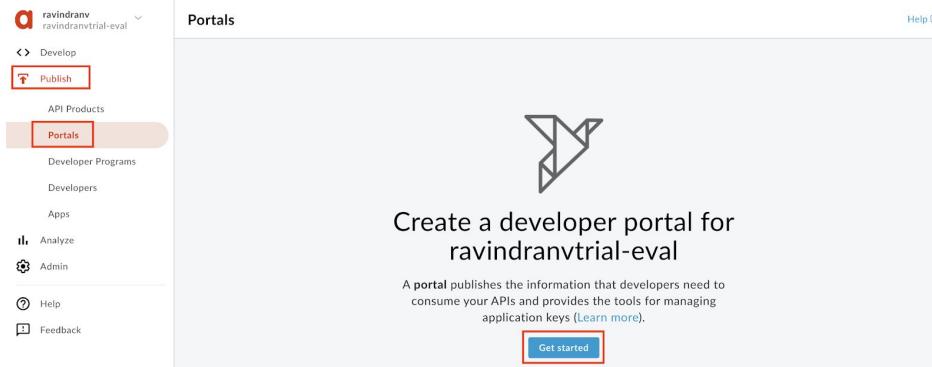
API プロキシ経由で公開される API エンドポイントを正確に記述した、最新の OpenAPI 仕様を用意するため、最初に仕様を修正する必要があります。具体的には、host, basepath, securityDefinitions, security, schemes の各プロパティを変更します。これを行うには、メインメニューで [Develop] → [Specs] に移動し、ラボ 1 で以前インポートした仕様を選択して、次の図のように、host, basepath, securityDefinitions, security, schemes の各プロパティを変更します。

```
host: {{your API proxy host}}
basePath: /{{your initials}}_hipster-products-api
securityDefinitions:
  APIKeyQuery:
    type: "apiKey"
    in: "query"
    name: "apikey"
  security:
    - APIKeyQuery: []
schemes:
  - https
```

## デベロッパー ポータルを作成する

---

1. [Publish] → [Portals] に移動し、[+Portal] または [Get started]（組織内にまだポータルを作成していない場合）をクリックします。



2. ポータル作成ウィザードに詳細を入力します。{{your initials}} は自分のイニシャルに置き換えます。
  - Name: {{your initials}}\_Hipster API Portal
  - Description: Developer portal for consumption of Hipster APIs.
3. [Create] をクリックします。

Create a Portal

Name  
VR\_Hipster API Portal

Description  
Developer portal for consumption of Hipster APIs

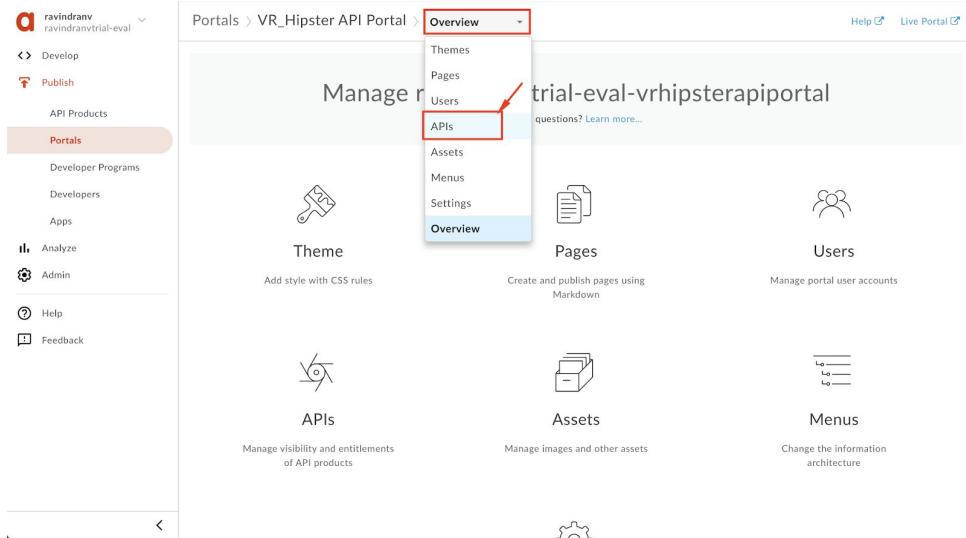
Cancel Create

## ブロンズ（無料）API プロダクトをポータルに公開する

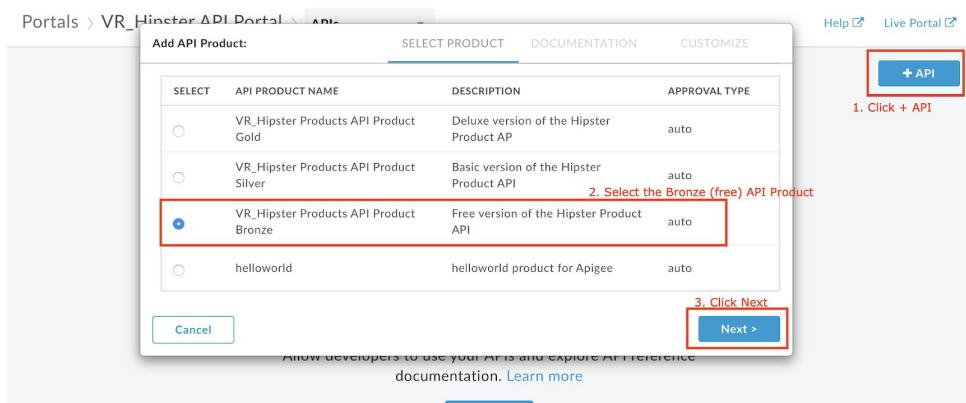
1. ポータルエディタのプルダウンをクリックして [APIs] を選択します。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)



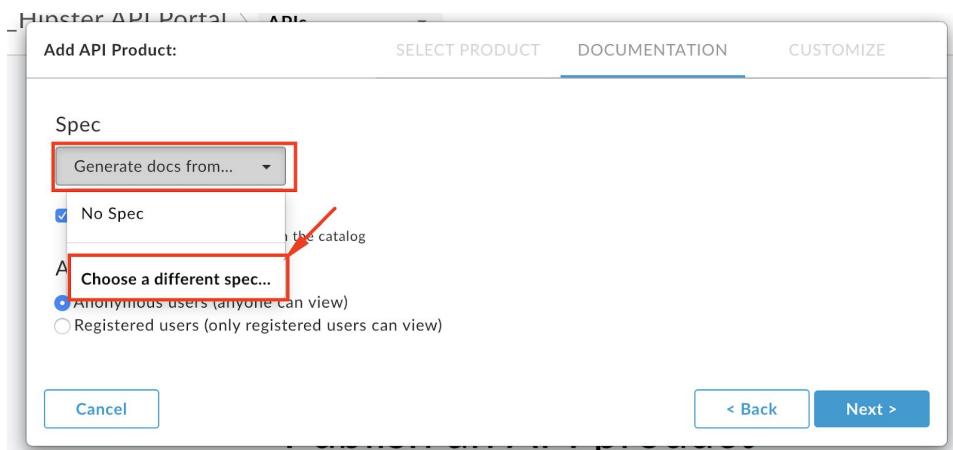
- [+API] をクリックして、ポータルに公開するプロダクツAPIプロダクトを選択します。公開するAPIプロダクトを選択し、[Next]をクリックします。



- [Change Spec Source] プルダウンをクリックして [Choose a different spec...] を選択します。

Google Cloud

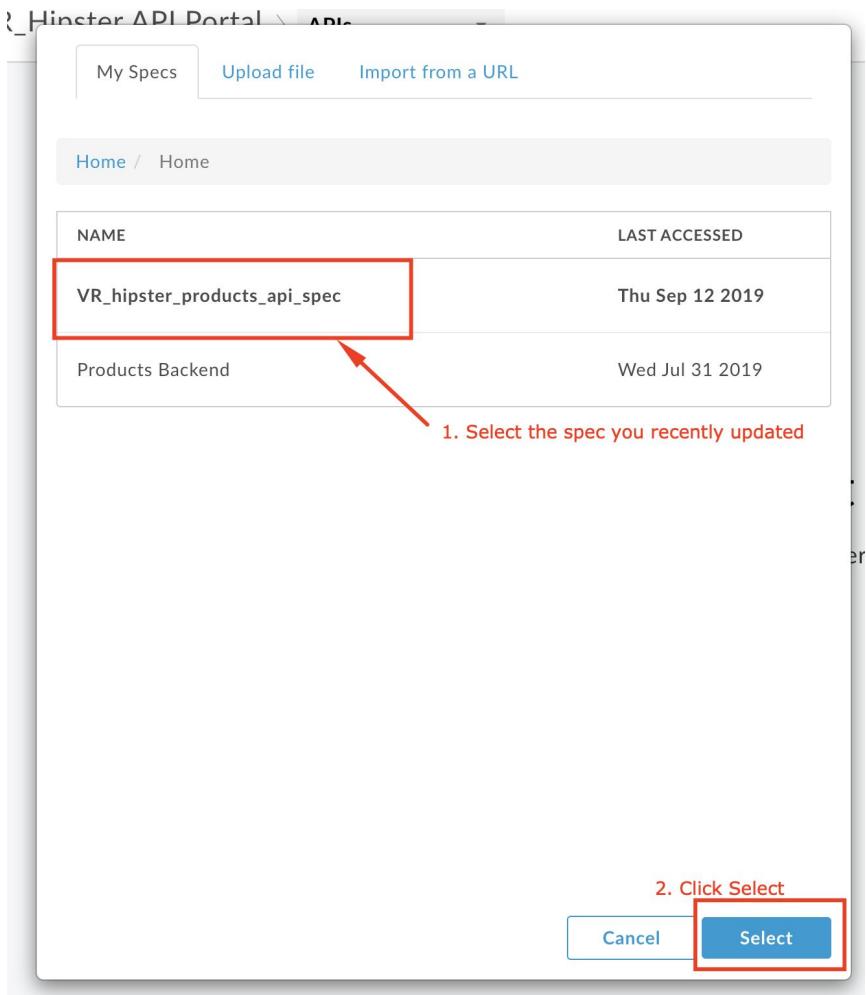
For more information visit [google.com/cloud](http://google.com/cloud)



4. ソースとして使用する、先ほど更新した OpenAPI 仕様を選択します。選択した OpenAPI 仕様の最新バージョン（スナップショット）を使って、ポータル内にこの API プロダクトのドキュメントを生成します。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)



5. [Registered users] オプションを選択すると、デベロッパー ポータルに登録済みのデベロッパーだけが、ポータルを通じてこの API を表示できるようになります。[Next] をクリックします。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

Hipster API Portal > APIs

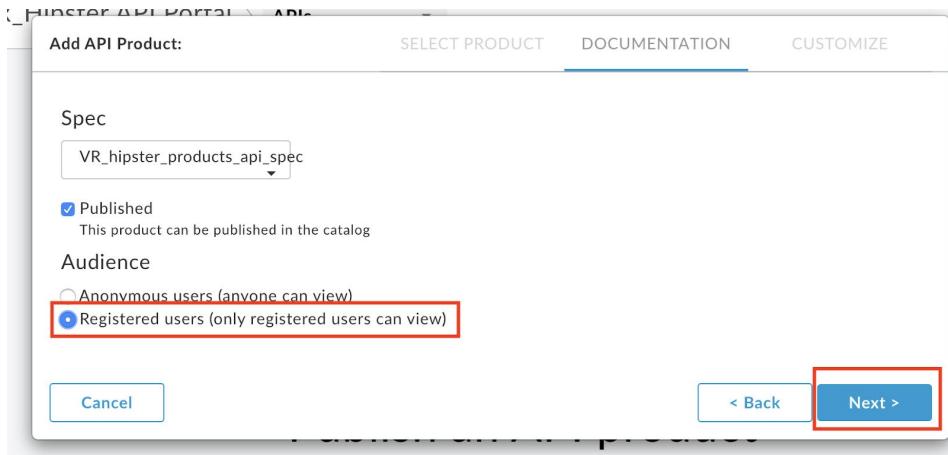
Add API Product: SELECT PRODUCT DOCUMENTATION CUSTOMIZE

Spec  
VR\_hipster\_products\_api\_spec

Published  
This product can be published in the catalog

Audience  
 Anonymous users (anyone can view)  
 Registered users (only registered users can view)

[Cancel](#) [Next >](#)



6. [Image] ボタンを選択して、この API プロダクトに関連付けられているアイコンイメージを更新します。

Hipster API Portal > APIs

Add API Product: SELECT PRODUCT DOCUMENTATION CUSTOMIZE

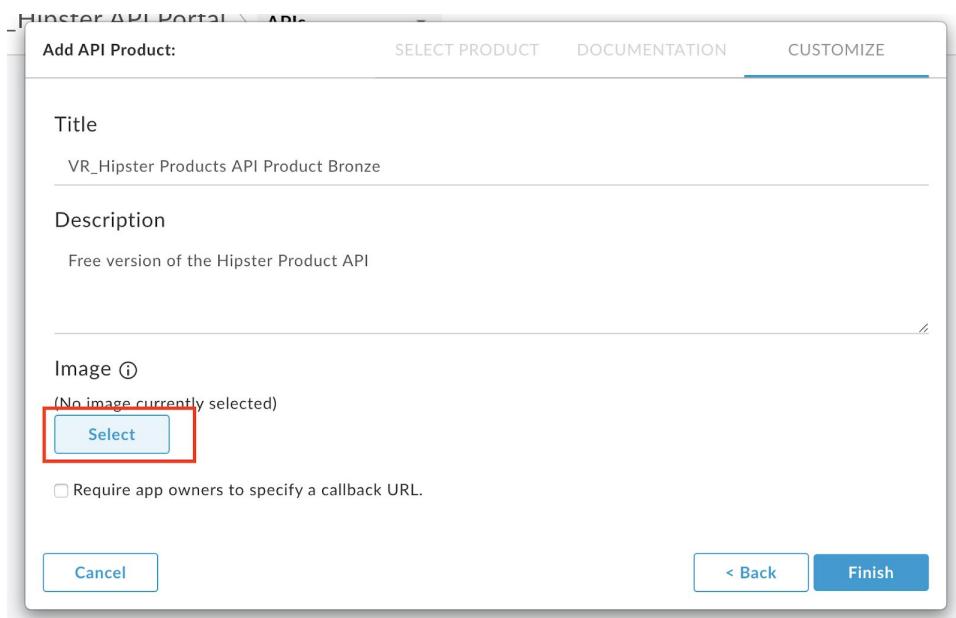
Title  
VR\_Hipster Products API Product Bronze

Description  
Free version of the Hipster Product API

Image ⓘ  
(No image currently selected)  
[Select](#)

Require app owners to specify a callback URL.

[Cancel](#) [Back](#) [Finish](#)

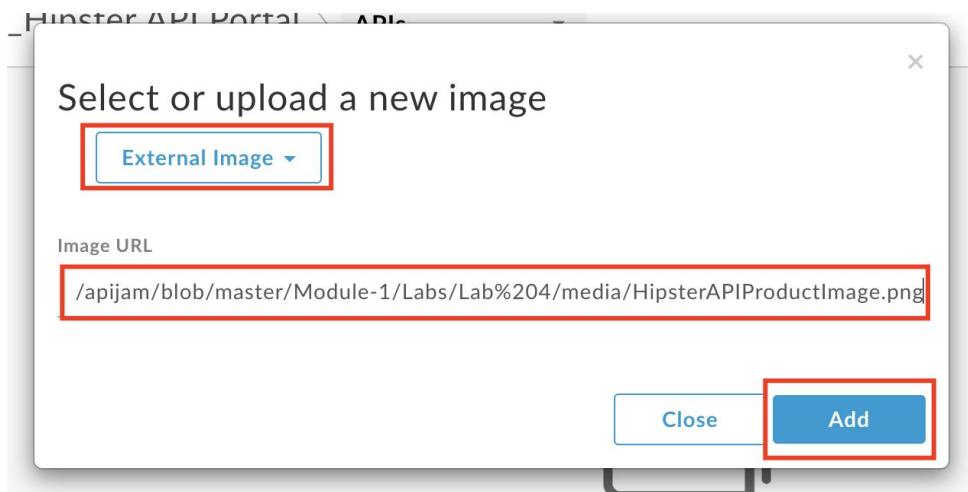


7. [External Image] を選択し、イメージをインポートする以下の URL を指定します。

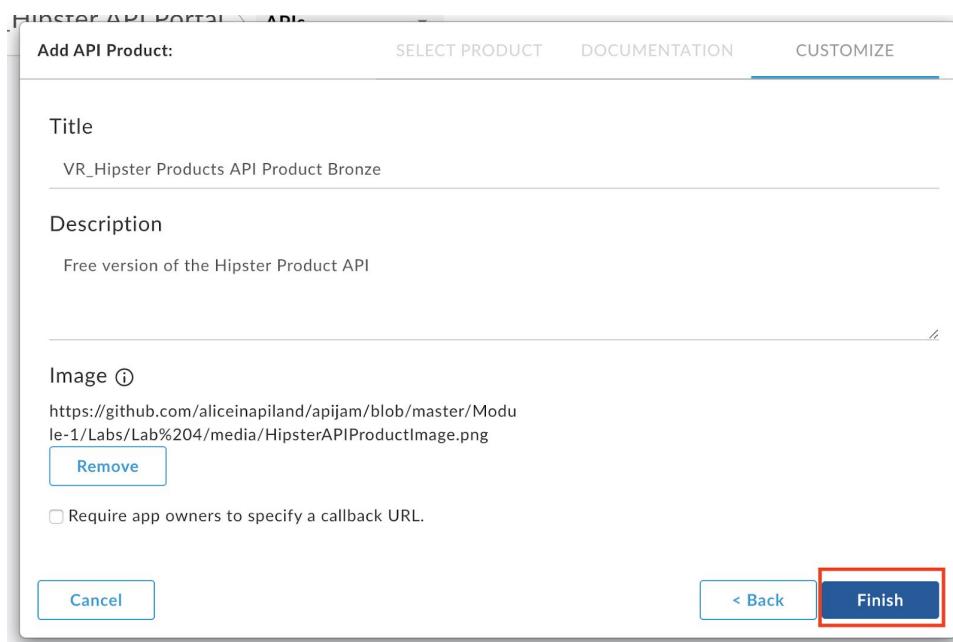
- Image URL:  
<https://raw.githubusercontent.com/aliceinapiland/apijam/master/Module-1/Labs/Lab%204/media/HipsterAPIProductImage.png>

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)



8. [Finish] をクリックします。



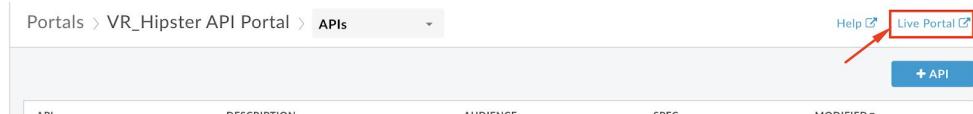
これでデベロッパー ポータルに API プロダクトが公開されました。

## アプリ デベロッパーの登録

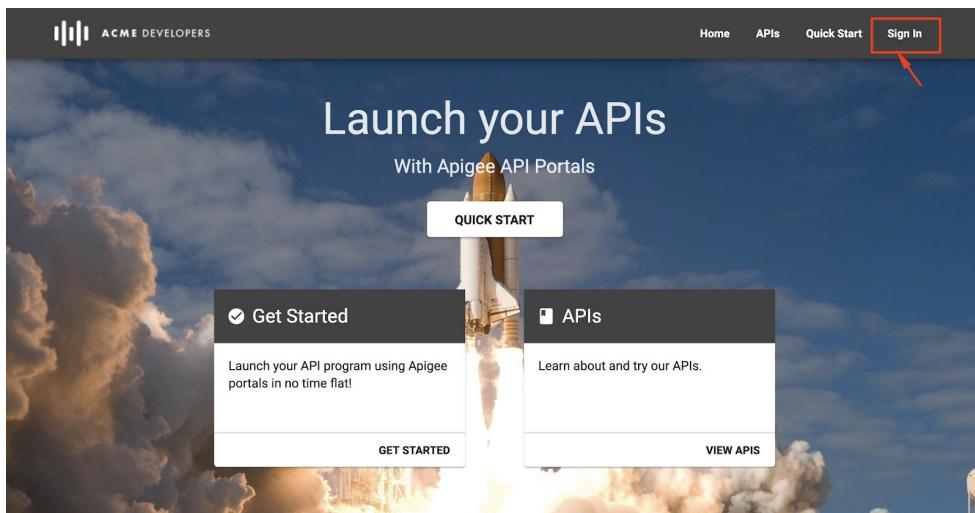
1. [Live Portal] リンクをクリックし、ブラウザタブまたはウィンドウで新たなデベロッパー ポータルを開きます。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

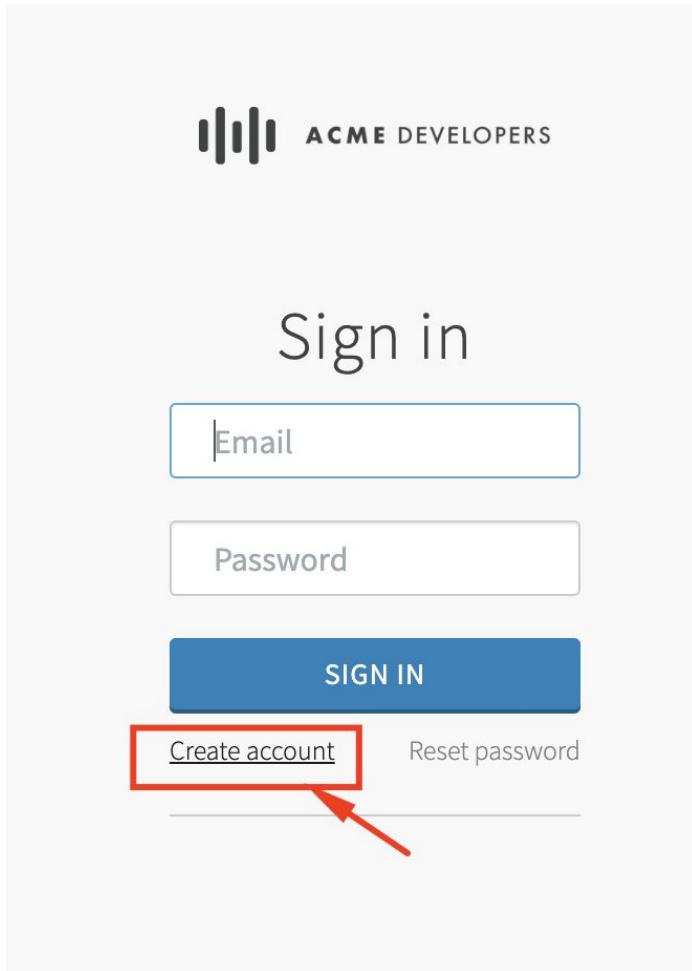


2. デベロッパー ポータルで [Sign In] メインメニュー オプションをクリックします。アブリ デベロッパーのログインページが表示されます。ここで [Create Account] をクリックします。



Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)



次の詳細情報を指定し、[Create Account] をクリックします。

*First Name: {{your first name}}*  
*Last Name: {{your last name}}*  
*Email: {{your email address}}*  
*Password: {{enter a password}}*  
[I agree to terms.] チェックボックスをオンにします。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)



## Create your account

First Name

Last Name

Email

Password  
 ⓘ

I agree to the [terms](#).

Have an account? [Sign in](#).

4. アカウントの作成時に、アプリ デベロッパーはアカウントの検証リンクが含まれたメール通知を受け取ります。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)



# Check your inbox

We sent a confirmation email to you.

Nothing in your inbox yet?  
Don't forget to check your spam folder.

このラボでは、自分のメールアドレスをアプリ デベロッパーとして提供しているので、この通知を受け取るはずです。通知リンクをクリックまたはブラウザに貼り付け、アカウントを検証します。

 **ravindrantrial-eval-vrhipsterapiportal <noreply@apigee.com>** • 7:34 AM (4 minutes ago)   

to me ▾

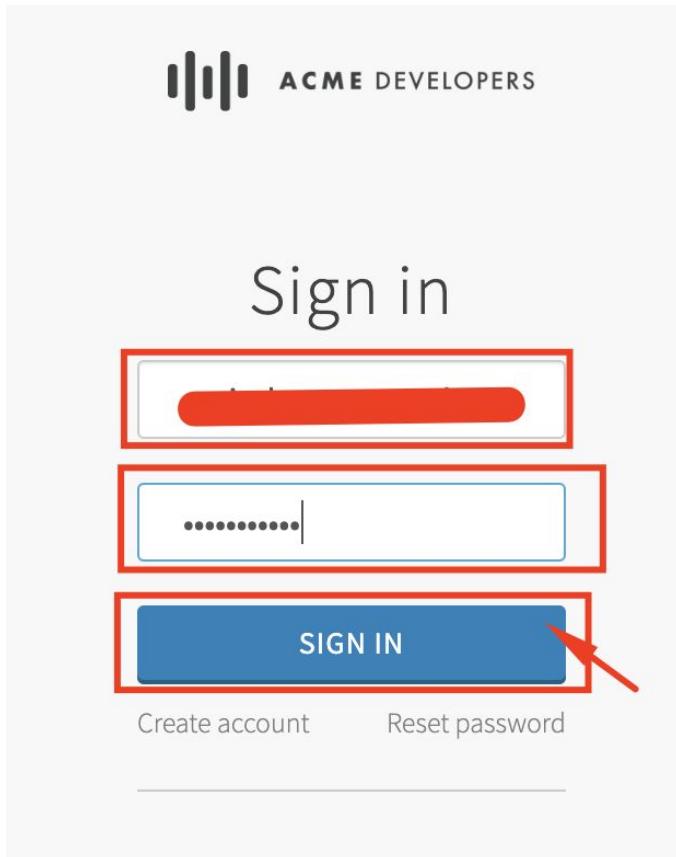
Welcome Vidya, Thank you for signing up for an account. To verify your identity, please visit this link within 10 minutes:  
<https://apigee.com/u/rp1011/https://apigee.com/u/rp1011/identity/verify?code=12345>

5. アカウントが検証されると、アプリ デベロッパーは認証情報を使用してポータルにログインできます。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

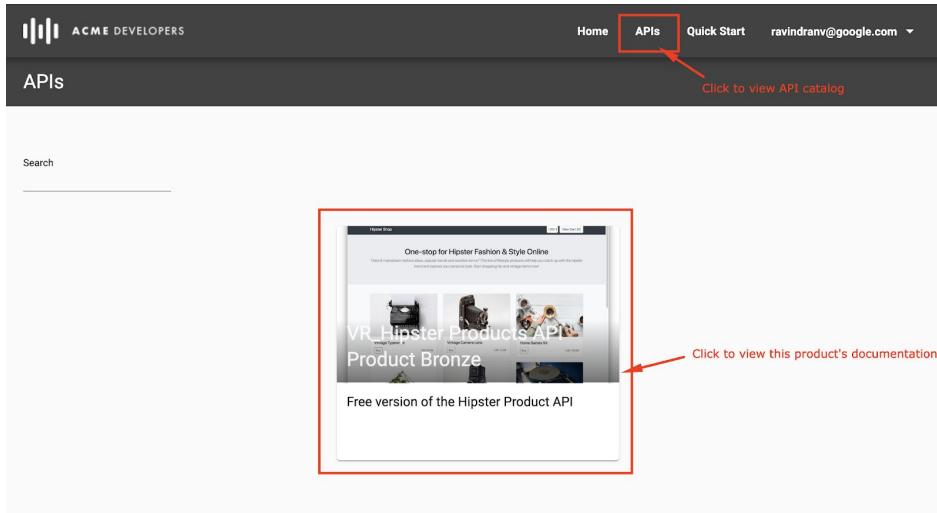


## API ドキュメントを表示する

1. 前の手順で作成したアカウントの認証情報を使用してアプリのデベロッパーとしてログインします。デベロッパー ポータルで [APIs] メニューリンクをクリックします。API カタログのページが表示されます。先ほど公開した API プロダクトが、登録されているすべてのデベロッパーに表示されていることがわかります。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)



2. カタログのブロンズ プロダクトの API プロダクトアイコンをクリックし、ドキュメントを表示します。API プロダクトの公開時に関連付けられた OpenAPI 仕様から生成されるインタラクティブなドキュメントページが表示されます。

Method	Endpoint	Description
GET	/products	

3. ドキュメントの左側のパネルから API リソースパスのいずれか1つを選択し、[Execute] をクリックします。右側のパネルに未承認レスポンス 401 が表示されます。

このレスポンスは、登録されているアプリがなく、認可された API 呼び出しに使用する API キーがないために表示されます。

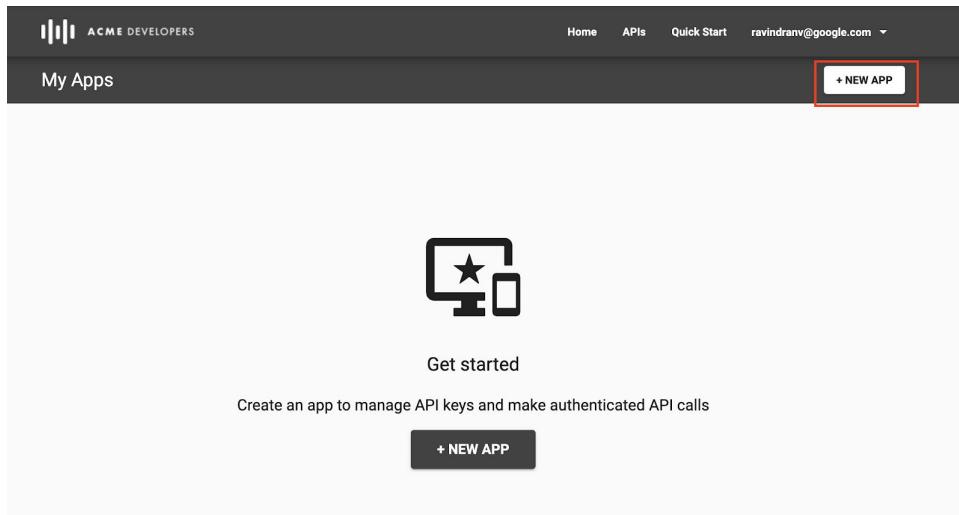
## アプリを登録する

1. 画面右上隅のデベロッパー アカウントのプルダウン メニューに移動し、[Apps] リンクを選択します。

2. [New App] ボタン（このページ上か、以下のように上部パネルにあります）をクリックします。

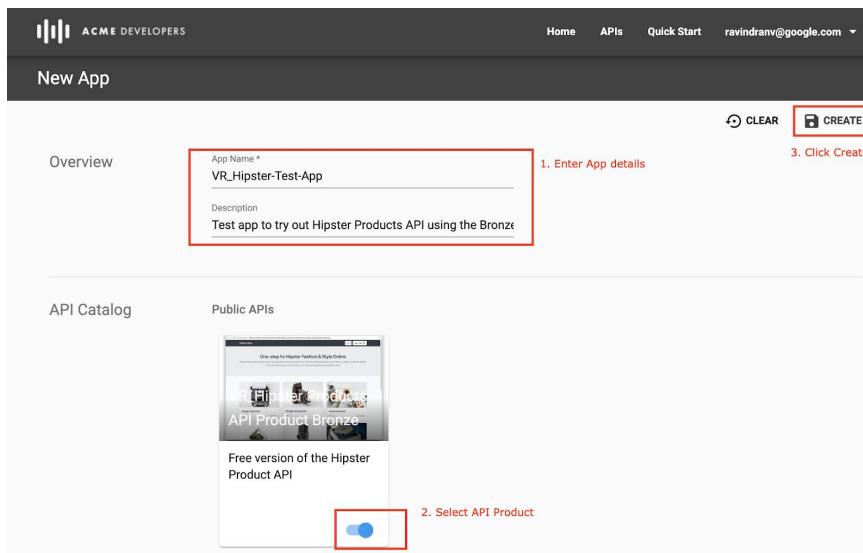
**Google Cloud**

For more information visit [google.com/cloud](http://google.com/cloud)



3. アプリの詳細を次のように入力し、[Create] をクリックします。  
App Name: {{your initials}}\_Hipster-Test-App Description: Test app to try out Hipster Products API using the Bronze (Free) API Product

サブスクリプションで使用できるブロンズ（無料の）API プロダクトを選択します。



4. 新たに作成されたアプリに API Key-Secret ペアが生成されます。これで、この API キーを使用して API をテストできるようになりました。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

The screenshot shows the ACME Developers API management interface. At the top, there's a navigation bar with links for Home, APIs, Quick Start, and a user account. Below the navigation is a dark header bar with the text "VR\_Hipster-Test-App". The main content area has a form for creating an application. The "Overview" tab is selected. The "App Name \*" field contains "VR\_Hipster-Test-App". The "Description" field contains "Test app to try out Hipster Products API using the Bronze (Free) API Product". The "App ID" field shows "1db61041-f970-4aed-a23d-ad728664ce2a". In the "API Keys" section, there's a table with one row. The row contains a key value "nIOGZlr4VZWwNA6cs82vsjj...", a "Copy to clipboard" button, a "Show" button (which is highlighted with a red box), a "Status" column showing "Active", a "Created" column showing "2019-09-20", and an "Expires" column showing "never". There's also a "DELETE" button and a "SAVE" button at the top right of the table. Below the table is a "ADD KEY" button. At the bottom of the page, there are tabs for "API Catalog" and "Public APIs".

5. API カタログに移動し、ブロンズ API プロダクトのドキュメントを選択して GET /products リソースをもう一度テストします。今回のテストでは、[Authorize] ボタンを最初にクリックし、新たに作成されたアプリの認証情報を選択して、API 呼び出しの認可情報（API キー）を設定します。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

1. Click on 'APIs' to get to API catalog  
>> click on API Product icon for docs

Home APIs Quick Start ravindranv@google.com

HIPSTER PRODUCTS API

PATHS

RESOURCES

GET /products

HTTP request

Response Types

Body

Authentication Requirements

Request parameters

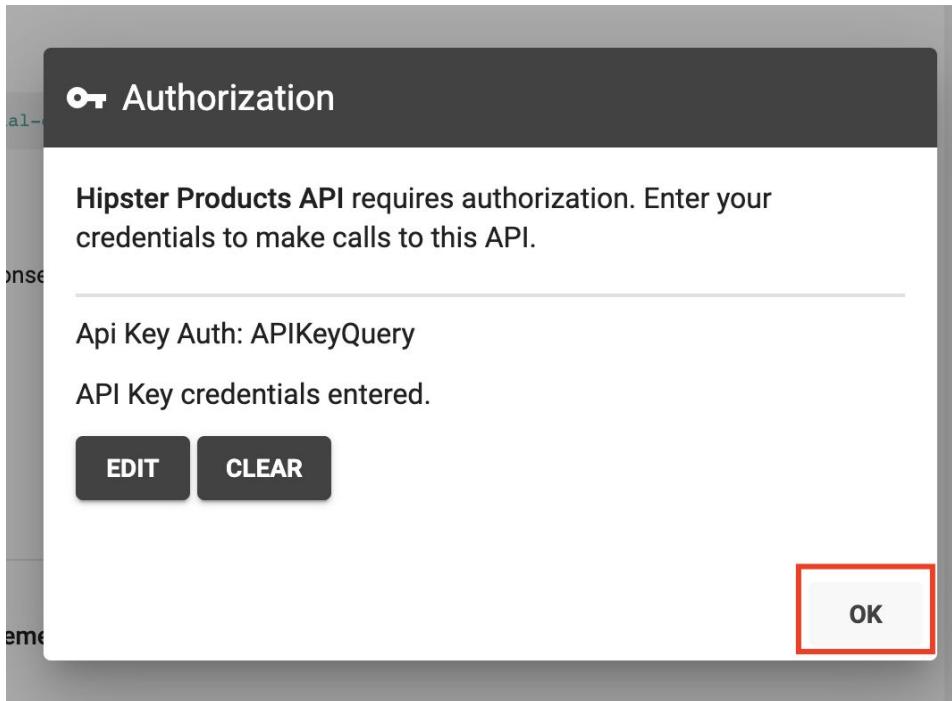
Credentials

EXECUTE

AUTHORIZATION

Try this API

認可情報が設定されたら [OK] をクリックします。



今度は [Execute] をクリックします。有効な 200 OK API レスポンスを受け取ります。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

## チーム機能とオーディエンス管理機能を有効にする

1. チーム機能とオーディエンス管理機能を利用するには、最初に Apigee Edge 組織でベータプログラムに参加する必要があります。それには、管理 UI で [Publish] -> [Developer Programs] に移動し、統合デベロッパー ポータルに関連付けられているデベロッパー プログラムを選択します。

2. [Enroll] ボタンをクリックして、「チームとオーディエンス管理のベータ版に参加」します。

**Google Cloud**

For more information visit [google.com/cloud](http://google.com/cloud)

ravindranv  
ravindrantrial-eval

Develop

Publish

API Products

Portals

**Developer Programs**

Developers

Apps

Analyze

Admin

Developer Programs > ravindrantrial-eval-vrhipsterapiportal

Developer Accounts Configuration

Enroll in beta for team and audience management features: ⓘ  
Teams: Allows responsibility for an app to be shared by multiple portal users. Learn more  
Audiences: Segment portal users to control access to content. Learn more

Enroll

Name	Email*	Status	Teams
Vidya Ravindran	ravindranv@google.com	Active	

参加が完了すると、デベロッパー プログラム内でチーム機能とオーディエンス管理機能が使用できるようになります。

Developer Programs > ravindrantrial-eval-vrhipsterapiportal

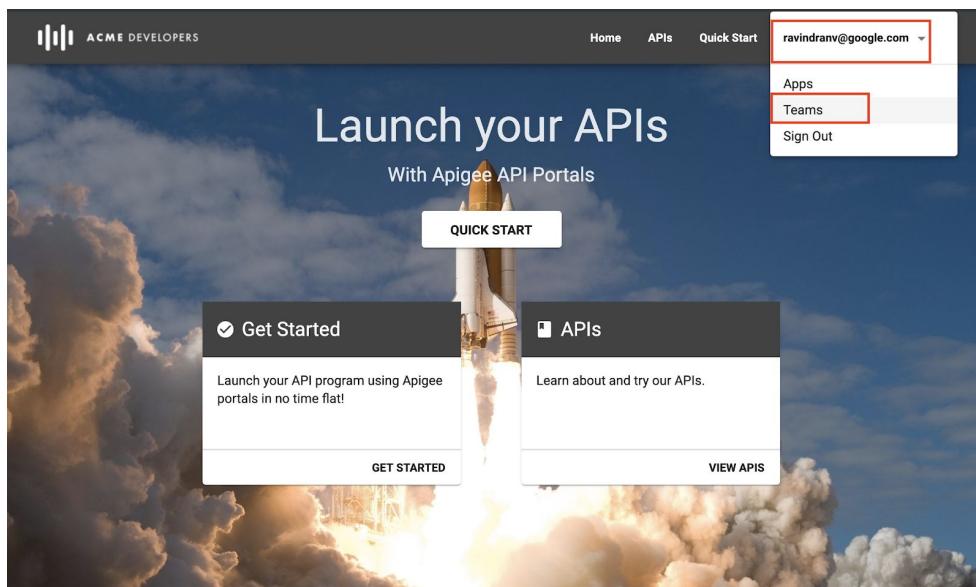
Developer Accounts Teams [beta] Audiences [beta] Configuration

Developer Accounts

Name	Email*	Status	Teams
Vidya Ravindran	ravindranv@google.com	Active	

## デベロッパー チームを作成する

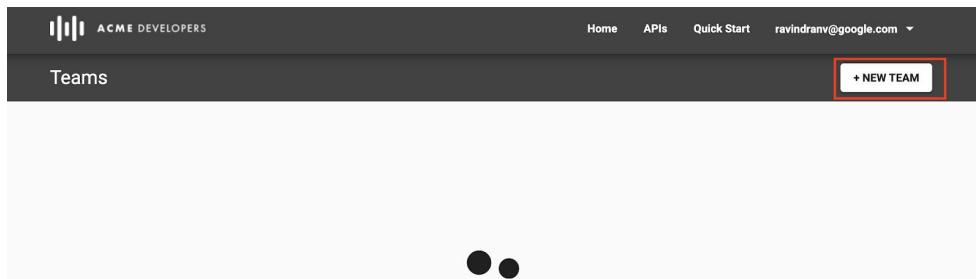
1. 自分のアプリのデベロッパー認証情報でデベロッパー ポータルにログインしていることを確認します。アカウントのプルダウンから [Teams] メニューに移動します。



Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

2. [+ New Team] ボタンをクリックします。



チームの詳細を入力して [Create] をクリックします。

[Overview] セクション: \* Team name: {{your initials}}\_Hipster App Team \* Description: Team that will work together on Hipster apps, and share API Keys. \* Point of contact: {your App Developer email address}

Members:

アプリのデベロッパーのメール ID がすでに「オーナー」ロールでチームに追加されています。

必要に応じて、別のロールのデベロッパーを追加できます。

Email address	Role
ravindranv@google.com	Owner
ravindranv+teammember1@google.com	App Admin
ravindranv+teammember2@google.com	Viewer

3. チームが作成されると、デベロッパー ポータルからチームのメンバーとして、このチームにアクセスできるようになります。API プロデューサーとしては、Apigee 管理 UI 内のデベロッパー プログラムで、作成されたチームを見ることができます。

# Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

The screenshot shows the ACME Developers portal interface. At the top, there's a navigation bar with links for Home, APIs, Quick Start, and a user account (ravindranv@google.com). Below the navigation is a dark header bar with the text "VR\_Hipster App Team".

**Overview** section:

- Team name: VR\_Hipster App Team
- Description: Team that will work together on Hipster apps, and share API Keys.
- Point of contact: ravindranv@google.com

**Members** section:

Email address	Role
ravindranv@google.com	Owner
ravindranv+teammember1@google.com	App Admin
ravindranv+teammember2@google.com	Viewer

+ Add another member

**Apps** section:

This team currently has no apps.

**Sidebar Navigation:**

- User profile: ravindranv (ravindrantrial-eval)
- Develop
- Publish
- API Products
- Portals
- Developer Programs** (selected)
- Developers
- Apps
- Analyze

**Developer Programs View:**

- Developer Accounts
- Teams** (selected)
- Audiences
- Configuration

**Teams Table:**

Name	Description	Members	Apps	Audiences	Created
VR_Hipster App Team	Team that will work together on Hipster apps, and share API Keys.	ravindranv@google.com, ... 0			51 seconds ago

## オーディエンスを作成する

特定のオーディエンスだけが表示したりサブスクライブできるような形で、デベロッパー ポータルに API プロダクトを公開する方法について説明します。

1. [Publish] → [Developer Programs] → {デベロッパー ポータルに関連付けられているデベロッパー プログラム} に移動します。[Audiences] タブをクリックし、[+] ボタンをクリックして新しいオーディエンスを追加します。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

Developer Programs > ravindrantrial-eval-vrhipsterapiportal

1. Navigate to Publish -> Developer Programs -> your Developer Portal's Developer Program.

2. Click on the Audiences tab.

3. Click to add new audience.

Name	Description	Assignments	Created
Authenticated	Authenticated Users	..	..

詳細を次のように入力し、[OK] をクリックします。

Name: Hipster-API-Privileged-Audience  
Description: A privileged audience that is allowed access to the Hispter API Silver and Gold products.

Developer Programs > ravindrantrial-eval-vrhipsterapiportal

Audiences

Name	Description	Assignments
Authenticated	Authenticated Users	..

New audience

Name: Hipster-API-Privileged-Audience

Description: A privileged audience that is allowed access to the Hispter API Silver and Gold products.

Cancel OK

- オーディエンスが作成されたら、オーディエンスに割り当てるチームを定義します。これを行うには、オーディエンスの [Assignments] セクションで [+] ボタンをクリックします。

The screenshot shows the 'Audience details' page for 'Hipster-API-Privileged-Audience'. It includes sections for 'Audience details', 'Assignments', and 'Entitlements'. The 'Assignments' section has a red box around the '+ Add' button.

Name	Description	Created
Hipster-API-Privileged-Audience	A privileged audience that is allowed access to the Hispter API Silver and Gold products.	2019-09-20

Email	Type
No assignments have been added yet	

Name	Portal	Type
No audience entitlements have been added yet		

ポップアップに、直前に作成したチームの名前を入力します。チームを選択して [Add(1)] をクリックします。

The screenshot shows the 'Audience details' page with an open 'Add assignments' modal. The modal contains a search bar with 'VR\_Hipster' and a list with one item checked: 'VR\_Hipster App Team'. At the bottom of the modal, the 'Add (1)' button is highlighted with a red box.

Name	Description	Created
Hipster-API-Privileged-Audience	A privileged audience that is allowed access to the Hispter API Silver and Gold products.	2019-09-20

Email
No assignments have been added yet

Name
No audience entitlements have been added yet

[Save] をクリックしてオーディエンスの割り当てを保存します。

Developer Programs > ravindrantrial-eval-vrhipsterapiportal > Audiences > Hipster-API-Privileged-Audience

Audience details

Name: Hipster-API-Privileged-Audience

Description: A privileged audience that is allowed access to the Hispter API Silver and Gold products.

Created: 2019-09-20

1 pending change Show

Cancel Save

Assignments

Email	Type
VR_Hipster App Team	Team

Entitlements

Name	Portal	Type
No audience entitlements have been added yet		

## 新しいオーディエンス資格に対応したシルバー API プロダクトを公開する

- [Publish] → [Portals] → {{your developer portal}} → [APIs] に移動し、[+API] ボタンをクリックします。

Portals > VR\_Hipster API Portal APIs Help Live Portal

1. Navigate to Publish -> Portals -> Your portal.

2. Select the 'APIs' option from this menu.

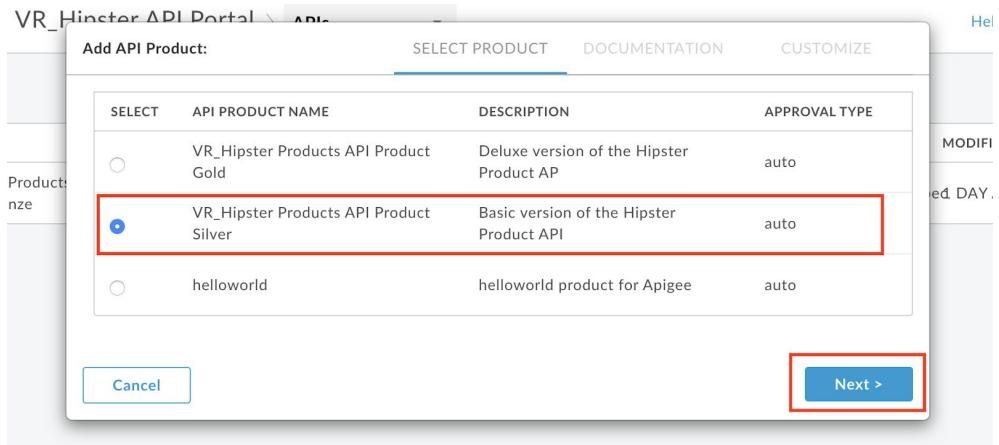
3. Click to publish new API products to the portal.

API	DESCRIPTION	AUDIENCE	SPEC	MODIFIED
VR_Hipster Products API	Free version of the Hipster Product API	Registered users	VR_hipster_products.api.spec	DAY AGO

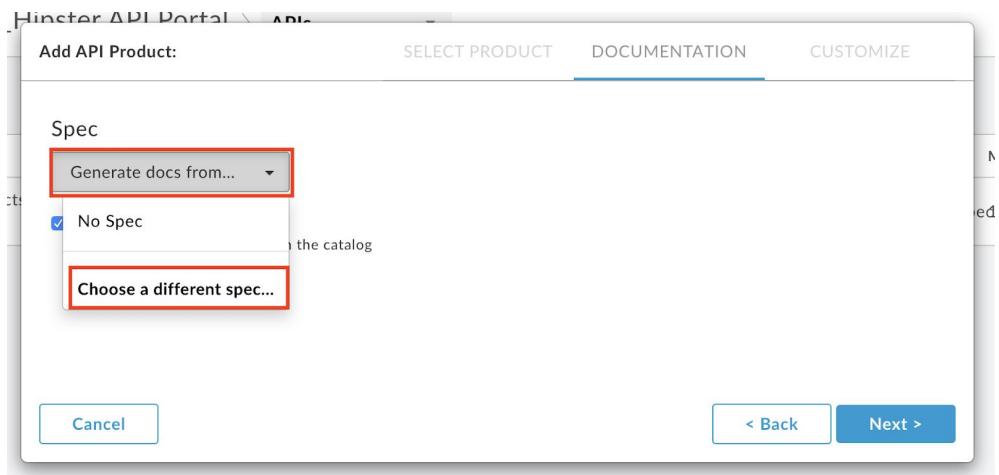
- シルバー API プロダクトを選択して [Next] をクリックします。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)



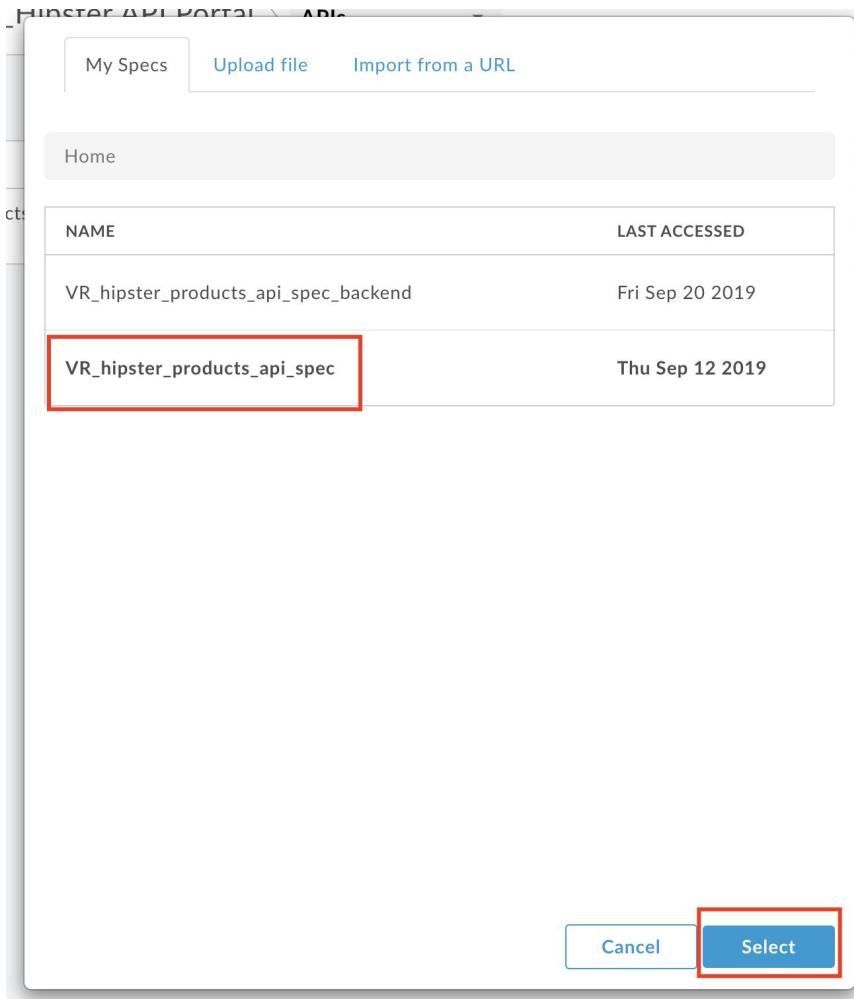
3. [Generate docs from] プルダウンをクリックして [Choose a different spec...] を選択します。



4. ソースとして使用する、最近更新した OpenAPI 仕様を選択します。選択した OpenAPI 仕様の最新バージョン（スナップショット）でポータル内にこの API プロダクトのドキュメントを生成します。

Google Cloud

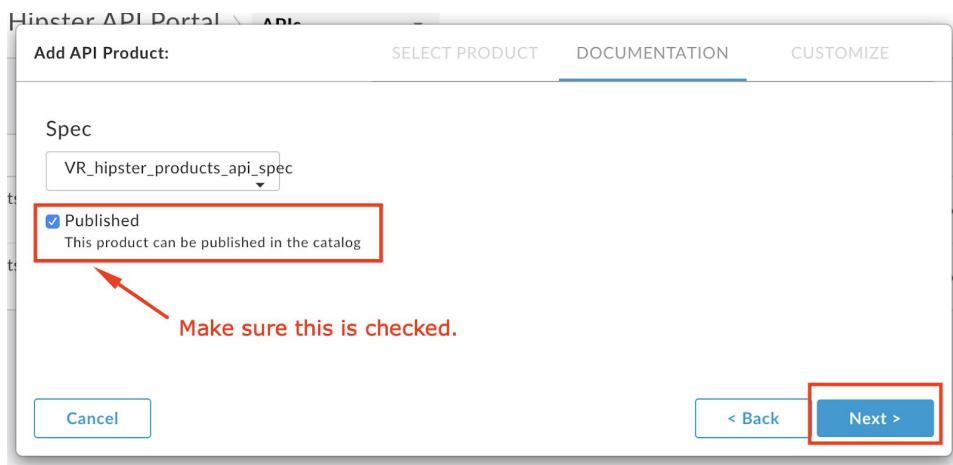
For more information visit [google.com/cloud](http://google.com/cloud)



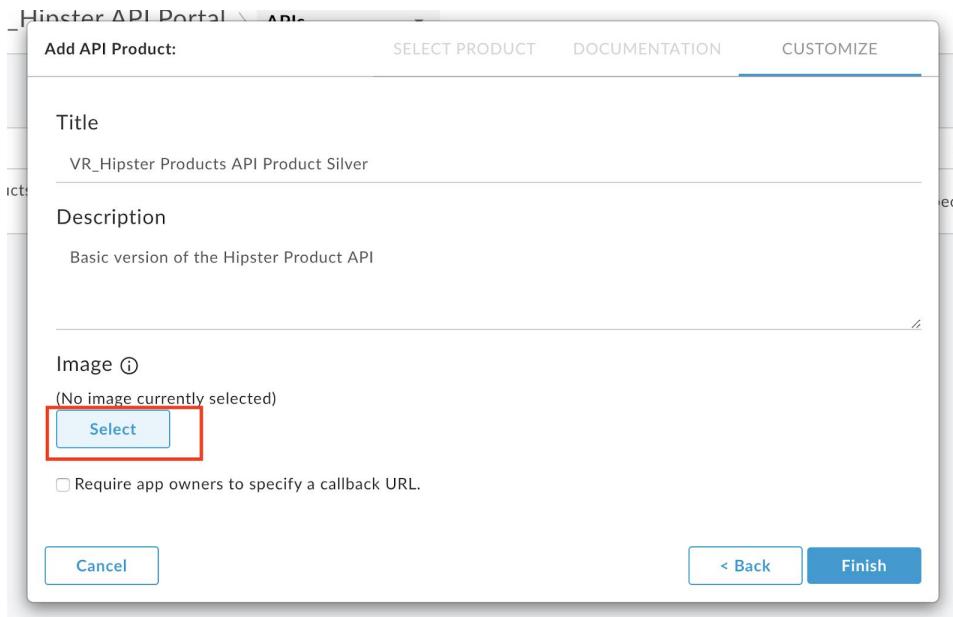
5. [Published] チェックボックスがオンになっていることを確認します。これで、アプリの作成時に API カタログから、シルバー API プロダクトが承認されたアプリのデベロッパーに公開されます。次に [Next] をクリックします。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)



6. [Image] ボタンを選択して、この API プロダクトに関連付けられているアイコンイメージを更新します。



7. [External Image] を選択し、イメージをインポートする以下の URL を指定します。

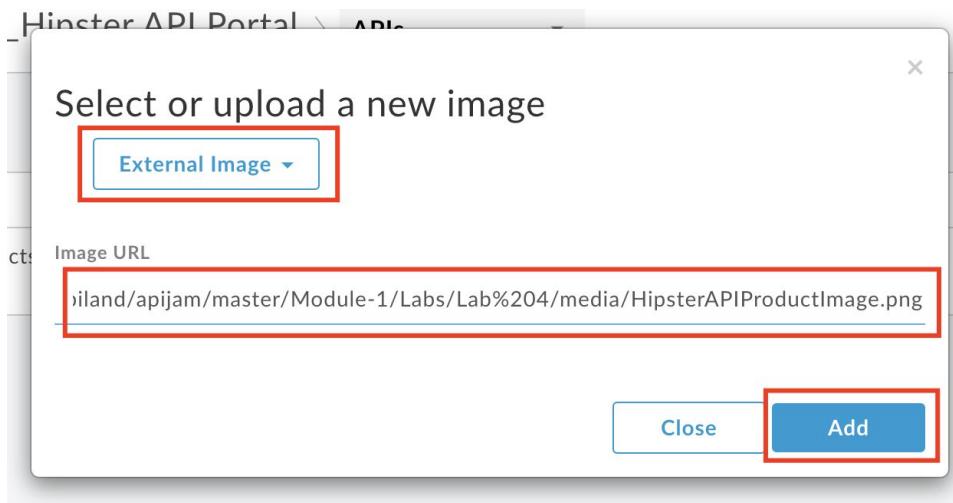
Image URL:

<https://raw.githubusercontent.com/aliceinapiland/apijam/master/Module-1/Labs/Lab%204/media/HipsterAPIProductImage.png>

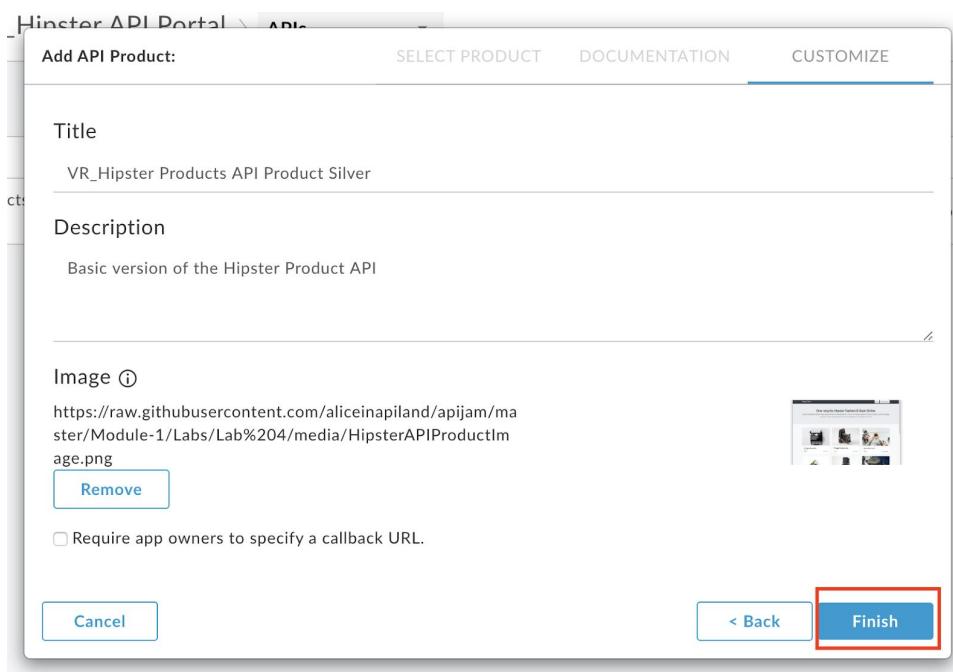
次に [Add] をクリックします。

## Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)



8. [Finish] をクリックします。



9. シルバー API プロダクトの [Audience Visibility] アイコンをクリックします。

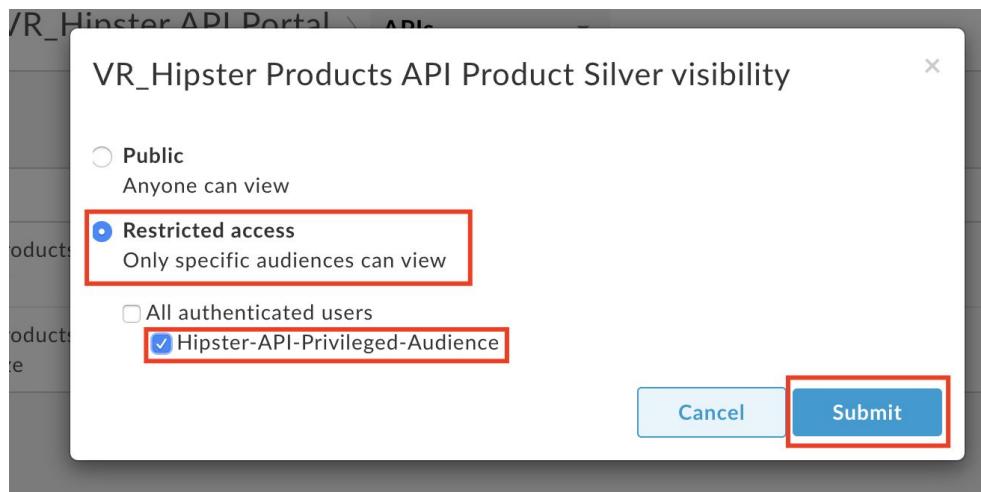
Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

Portals > VR\_Hipster API Portal > APIs

API	DESCRIPTION	AUDIENCE	SPEC	Modify Audience Visibility
VR_Hipster Products API Product Silver	Basic version of the Hipster Product API	Anonymous users	VR_hipster_products_api_spec	
VR_Hipster Products API Product Bronze	Free version of the Hipster Product API	Registered users	VR_hipster_products_api_spec	DAY AGO

- [Restricted access] オプションを選択し、作成したオーディエンスのチェックボックスをオンにします。[Submit] をクリックします。



## チームアプリを作成する

- デベロッパー ポータルで、画面右隅のデベロッパー アカウントのプルダウンメニューに移動し、[Apps] リンクを選択します。[New App] ボタンをクリックします。

ACME DEVELOPERS

My Apps + NEW APP

Name	Description	Owner	Created
VR_Hipster-Test-App	Test app to try out Hipster Products API using the Bronze (Free) API Product	Me	Sep 20, 2019

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

2. アプリの詳細を次のように入力し、[Create] をクリックします。

App Name: {{your initials}}\_Hipster App Description: App that is registered against the team to share App credentials and access Silver tier Hipster API product. Owner: {{your initials}}\_Hipster App Team

サブスクリプションに使用できるシルバー（無料の）API プロダクトを選択します。

New App

Overview

App Name \* VR\_Hipster App

Description App that is registered against the team to share App cre

Owner

App Owner \* VR\_Hipster App Team

API Catalog

Public APIs

API Product Silver

Basic version of the Hipster Product API

API Product Bronze

Free version of the Hipster Product API

3. 新たに作成されたアプリに API Key-Secret ペアが生成されます。これで、この API キーを使用して API をテストできるようになりました。

The screenshot shows the ACME Developers API management interface. At the top, there's a navigation bar with 'ACME DEVELOPERS' logo, 'Home', 'APIs', 'Quick Start', and a user email 'ravindranv@google.com'. Below the navigation is a dark header bar with the text 'VR\_Hipster App'. On the left, there's a sidebar with sections like 'Overview', 'Owner', and 'API Keys'. The main content area shows app details: 'App Name \*' is 'VR\_Hipster App'; 'Description' is 'App that is registered against the team to share App credentials and access Silver tier Hipster API product.'; 'App ID' is '693a1b3e-b906-4bbd-880a-743ee42b8ab6'. Under 'API Keys', there's a table with one row: Key 'czCbmYZpY7UsdtVM8tayRG...', Secret 'Show', Status 'Active', Created '2019-09-20', and Expires 'never'. A red box highlights the key value. At the bottom of this section is a 'ADD KEY' button. Below this is another section titled 'API Catalog' with tabs 'Public APIs' and 'Hipster API'. The 'Hipster API' tab is selected.

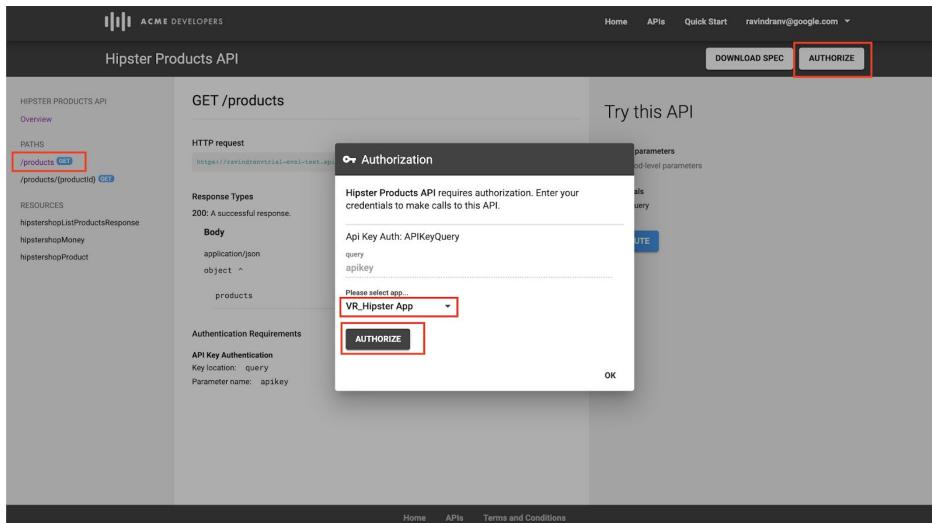
4. [API Catalog] に移動し、シルバー API プロダクトのドキュメントを選択します。

The screenshot shows the ACME Developers API Catalog. At the top, there's a navigation bar with 'ACME DEVELOPERS' logo, 'Home', 'APIs' (which is the active tab, highlighted with a red border), 'Quick Start', and a user email 'ravindranv@google.com'. Below the navigation is a dark header bar with the text 'APIs'. The main content area shows two API products: 'Product Bronze' and 'Product Silver'. The 'Product Silver' card is highlighted with a red box. Both cards have a thumbnail image of a person sitting at a desk with a laptop, the text 'One-stop for Hipster Fashion & Style Online', and the subtext 'Free version of the Hipster Product API' or 'Basic version of the Hipster Product API'.

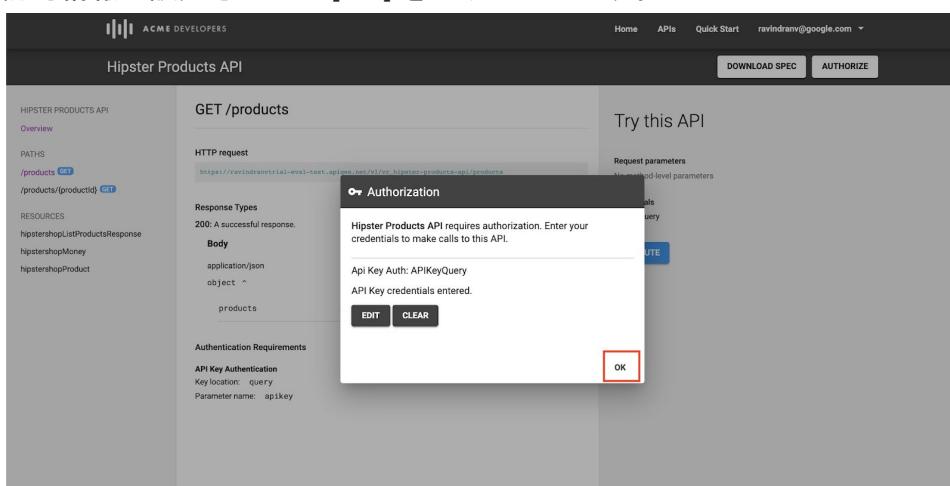
5. GET /products リソースを選択します。[Authorize] ボタンをクリックし、新たに作成したチームアプリの認証情報を選択して、API呼び出しの認可情報（API キー）を設定します。ポップアップから [Authorize] をクリックします。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)



認可情報が設定されたら [OK] をクリックします。



# Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

[Execute] をクリックします。有効な 200 OK API レスポンスを受け取ります。

The screenshot shows the Apigee API portal interface. On the left, there's a sidebar with 'Hipster Products API' and sections for 'PATHS' (including '/products' and '/products/{productId}'), 'RESOURCES' (listing 'hipsterShop.listProductsResponse', 'hipsterShopMoney', and 'hipsterShopProduct'), and 'Authentication Requirements' (API Key Authentication). The main area shows a 'GET /products' endpoint with an 'HTTP request' section containing the URL 'https://revivedtrial-eval-test.apigee.net/v1/\_vt\_hipster-products-api/products'. Below it, the 'Response Types' section indicates a successful 200 response with a JSON object body. A 'Try this API' panel is open, showing the 'EXECUTE' button and a modal window titled '200 OK' displaying a JSON response. The response contains an array of products, with the first item being a typewriter.

```
200 OK
{
  "products": [
    {
      "id": "66VCHSJNQF",
      "name": "Vintage Typewriter",
      "description": "This typewriter looks good in your living room",
      "picture": "/static/img/products/typewriter.jpg",
      "price": {
        "currencyCode": "USD",
        "units": "67",
        "nanos": 99000000
      },
      "categories": [
        "vintage"
      ],
      "id": "66VCHSJNQF"
    }
  ]
}
```

## ラボの動画

このトピックを取り上げている動画を視聴するには、ブラウザで[こちらを開いてください](#)。

## さらに理解を深める

- ポータルに2番目のプロダクトを追加し、ライブポータルを起動してこのプロダクトをテストします。
- 自分のAPI仕様を更新し、その仕様のスナップショットを取得して、ポータルのドキュメントを更新します。

## 理解度チェック

- 複数のAPIプロダクトをデベロッパー ポータルに公開する理由を2つ挙げてください。
- OpenAPI仕様に加えられた変更は、デベロッパー ポータルに自動的に反映されます。正しいか誤りかを選んでください。

## まとめ

以下について学びました。

## Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

- Apigee の軽量デベロッパー ポータルのデプロイ
  - OpenAPI 仕様付きでの API プロダクトの公開
  - デベロッパー ポータル UI を使用して OpenAPI 仕様のスナップショットをデベロッパーとして閲覧
- 

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

# ラボ 5 - Apigee Analytics で API プログラムの成果を評価する

所要時間: 15 分

登場人物: API プロダクトチーム

## ユースケース

---

### 最初の API プロダクト公開後の流れ

いくつかの API プロダクトの稼働開始後は、次の点について、プログラムがどのように実行され、API プロダクトが正常に動作しているかどうか確認する必要があります。

- トラフィックは長期的にどのような傾向があるか。
- トラフィックが最大となっているのはいつか。
- ユーザーが最も多い場所はどこか。
- 最もアクティブなデベロッパーは誰か。

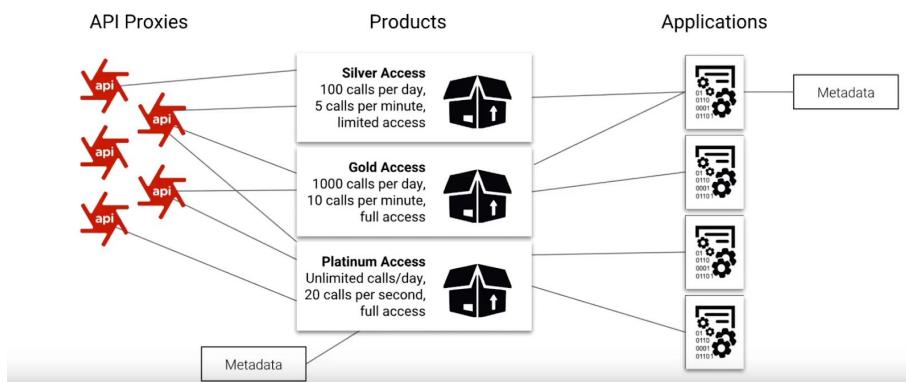
### 分析情報によるキャパシティ プランニングのサポート

キャパシティ プランニングを実行する必要があるとき、利用トレンドデータを使って、API 呼び出しの長期的な傾向を観察し、次の年のトラフィック予測を立てることができます。

Apigee 組み込みの分析機能によってデータを可視化することで、API の傾向とパターンの理解をサポートします。

# トラフィックと API プロダクトの連携

## Products and Applications



アプリケーションを API プロダクトに登録すると、API キーまたは OAuth トークンの検証時にメタデータが読み込まれ、デベロッパー、アプリ、プロダクトが識別できるようになります。このデータはプログラムに関する分析情報にとって重要です。

## 前提条件

- 前のモジュールを受講していれば、そのモジュールで作成したプロキシに対するトラフィックがある程度あり、前の API プロキシをカスタム統計に使用できます。
- 受講していなくてもこのラボを受講できますが、ダッシュボードに表示されるデータは少なくなります。
- (省略可) フル エクスペリエンスを試したい場合、ダッシュボードのデータを閲覧する前に、負荷生成ツールを設定します。

## (省略可) API プロキシに負荷を加える

前のモジュールを受講していない場合、負荷生成ツールを使用してレポートにデータを加えることができます。

デフォルトの API キーポリシーと 3 つの API キーを持つ API プロキシが用意されていると、さまざまなアプリと API プロダクトを容易に作成できます。

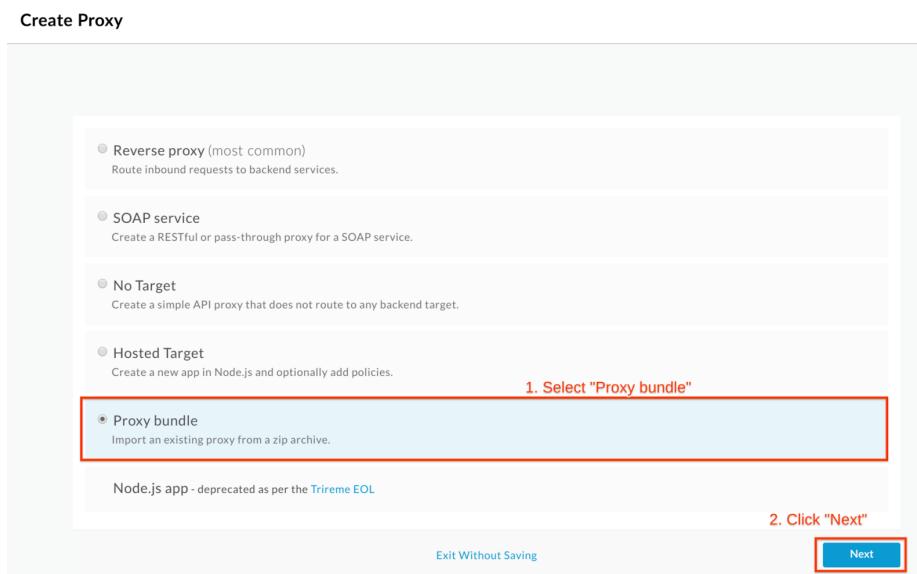
Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

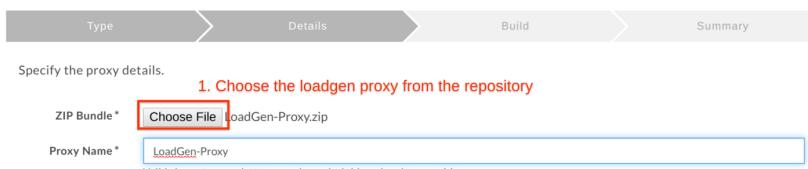
## 負荷生成ツールを構成して実行する

このリポジトリ内の loadgen フォルダにあるプロキシをアップロードすることで、新しいプロキシを作成できます。

1. [Develop] > [API Proxies] > [+ Proxy] の順にクリックして新しいプロキシを作成します。
2. 新しいプロキシ ウィザードでプロキシバンドルのアップロードを選択し、リポジトリから .zip ファイルをアップロードします。この .zip ファイルは[こちら](#)からダウンロードします。



## Create Proxy



Specify the proxy details.

**1. Choose the loadgen proxy from the repository**

ZIP Bundle \* **Choose File** LoadGen-Proxy.zip

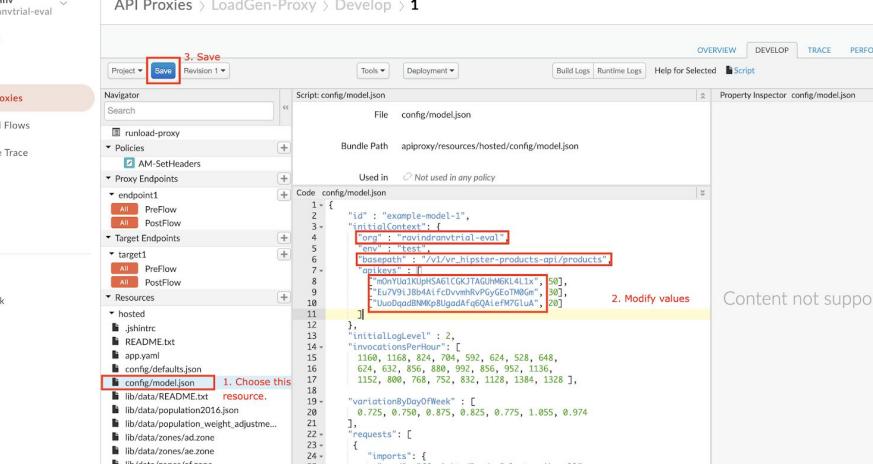
Proxy Name \* **LoadGen-Proxy**

Valid characters are letters, numbers, dash (-), and underscore (\_).

**2. Click "Next"**

Previous Exit Without Saving **Next**

3. [Build] をクリックします。
4. プロキシが作成されたら、それをエディタで直接開き、[Develop] タブに移動してプロキシと API キー関連の値を変更します。左側のパネルの [Resources] セクションで config/model.json リソースに移動し、以下のように値を変更します。



**3. Save**

Project: **LoadGen-Proxy** Revision 1

Navigator: Search: runload-proxy

File: config/model.json

Bundle Path: aliproxy/resources/hosted/config/model.json

Used in: Not used in any policy

```

1+ {
  2+   "id": "example-model-1",
  3+   "env": "context",
  4+   "env": "test",
  5+   "proxiesPath": "/v1/vr_hipster-products-api/products",
  6+   "proxiesPath": "/v1/vr_hipster-products-api/products",
  7+   "proxiesPath": "/v1/vr_hipster-products-api/products",
  8+   "proxiesPath": "/v1/vr_hipster-products-api/products",
  9+   "proxiesPath": "/v1/vr_hipster-products-api/products",
  10+  "proxiesPath": "/v1/vr_hipster-products-api/products"
}
  
```

**2. Modify values**

Content not supported

5. 値を変更したら、プロキシを保存してデプロイし、[Trace] タブからプロキシを実行します。
6. しばらく負荷生成ツールを実行しておき、プロキシによるトラフィックの生成を停止するときはプロキシをデプロイ解除します。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

## ビルトインの Apigee Analytics ダッシュボードを使用して API の使用状況を把握する

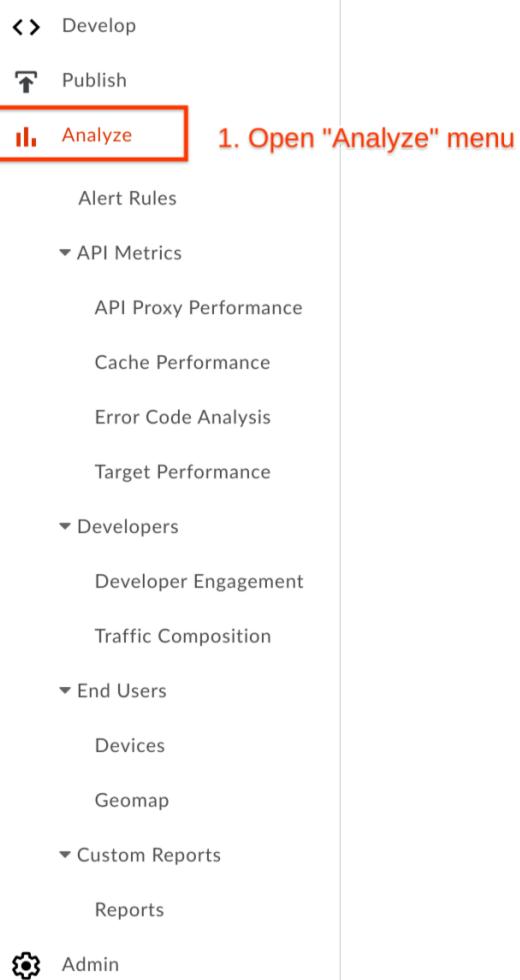
---

Apigee には数多くの組み込みレポートが用意されています。このラボの目的のため、また時間を節約するため、稼働中のデモ環境から取得したスクリーンショットで、さまざまなダッシュボードをご紹介します。各自の環境で手順を実行する場合は、オプションの負荷生成ツールを使用してください。

動画で API ダッシュボードの確認を希望する場合は、こちらをご覧ください。数多くの値が入力済みの、さまざまな指標のダッシュボードの概要が紹介されています。

<https://youtu.be/mEINO44QQFQ?t=126>

レポートに移動するにはサイドバーの [Analyze] メニューを開きます。

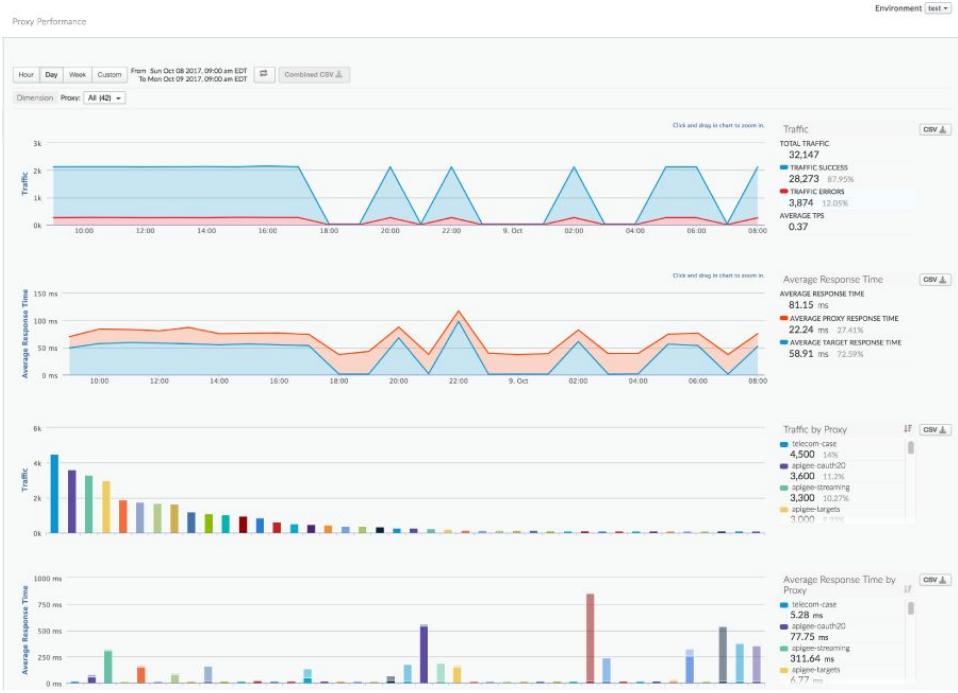


## API プロキシのパフォーマンス

[Proxy Performance] ダッシュボードは、API プロキシのトラフィック パターンと処理時間を確認するために役立ちます。API によって生成されるトラフィックの量と API 呼び出しの処理にかかる時間（Apigee Edge で受信されてからクライアントアプリに返されるまで）を簡単に可視化できます。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)



## キャッシュのパフォーマンス

[Cache Performance] ダッシュボードでは、Apigee Edge キャッシュの値を一目で確認できます。このダッシュボードは、レイテンシの短縮とバックエンドサーバーの負荷の軽減の観点から、キャッシュの効果を可視化するために役立ちます。



Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

## エラーコード分析

[Error Code Analysis] ダッシュボードには、API プロキシおよびターゲットのエラー率に関する情報が表示されます。[Error Code Analysis] ダッシュボードの表示は次のとおりです。

- プロキシエラーはレスポンスコードを使用して計算
- ターゲットエラーはターゲットからのレスポンスコードを使用して計算



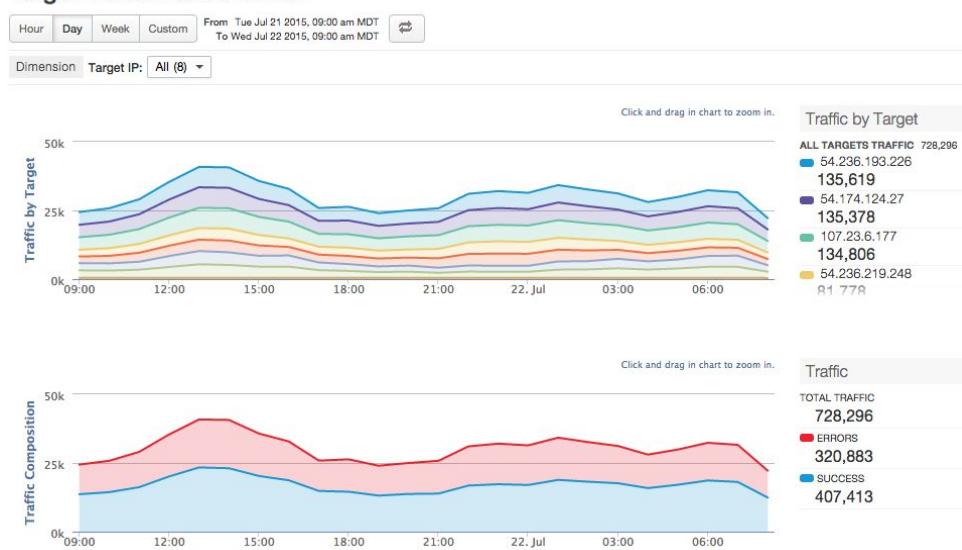
## ターゲットのパフォーマンス

[Target Performance] ダッシュボードは、API プロキシのバックエンドターゲットのトラフィックパターンとパフォーマンス指標を可視化します。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

## Target Performance [Learn more](#)

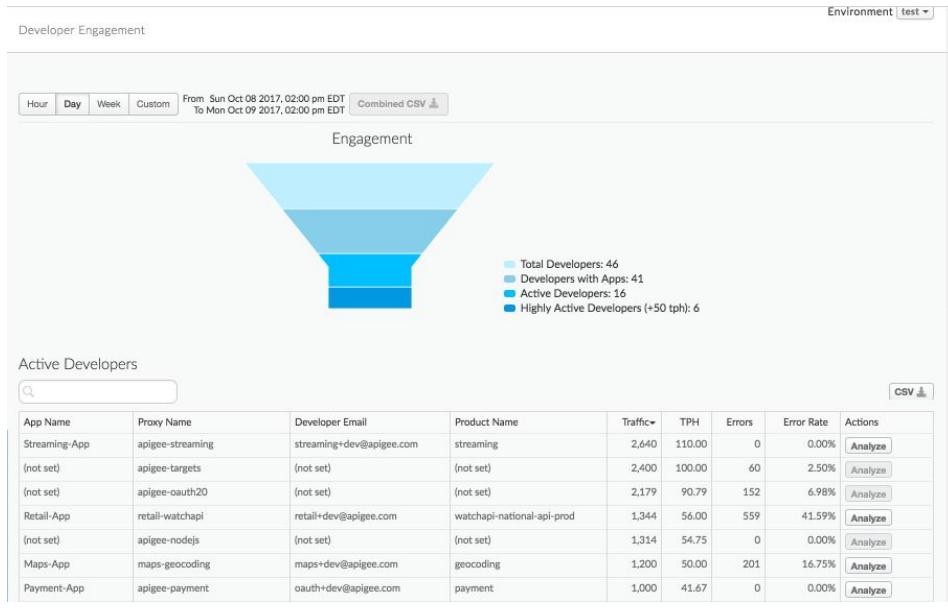


## デベロッパーのエンゲージメント

[Developer Engagement] ダッシュボードでは、最も多くの API トランザクションを生成している登録済みのアプリケーションがわかります。デベロッパーごとに、誰が最も多くの API トランザクションと最も多くのエラーを生成しているかを調べることができます。たとえば、特定のデベロッパーのアプリが他のデベロッパーに比べて多くのエラーを発生している場合は、そのデベロッパーの問題にプロアクティブに対処できます。



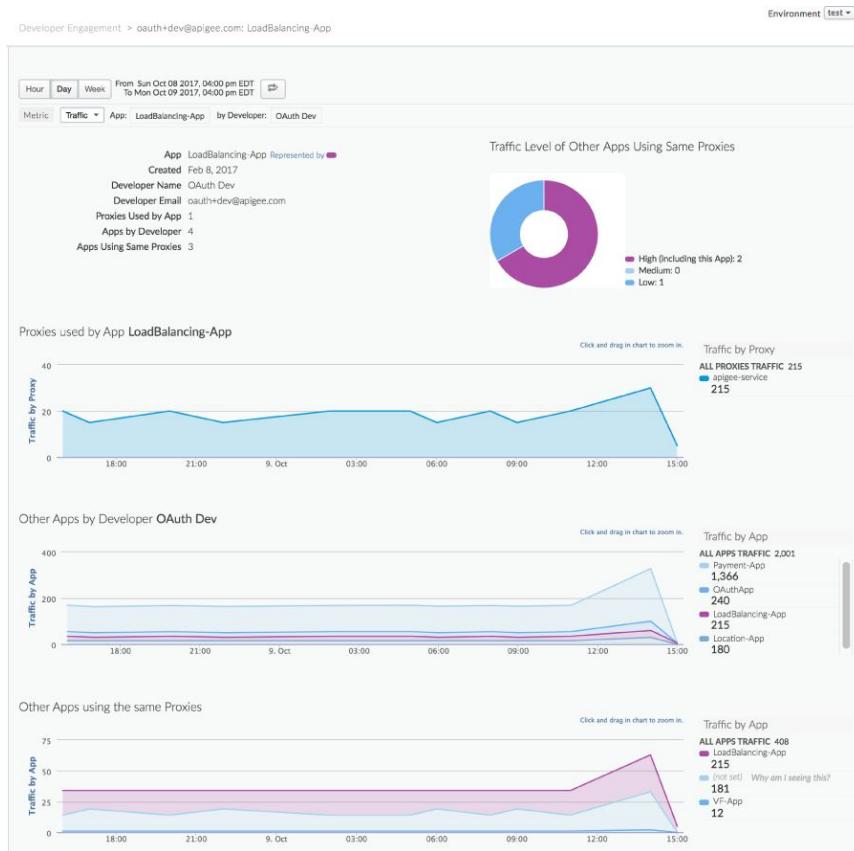
For more information visit [google.com/cloud](http://google.com/cloud)



メインビューが使用できる場合、アプリの [Actions] 列の下にある [Analyze] ボタンを選択し、アプリとアプリのデベロッパーに関する詳細を表示できます。以下のグラフが表示されます。

# Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

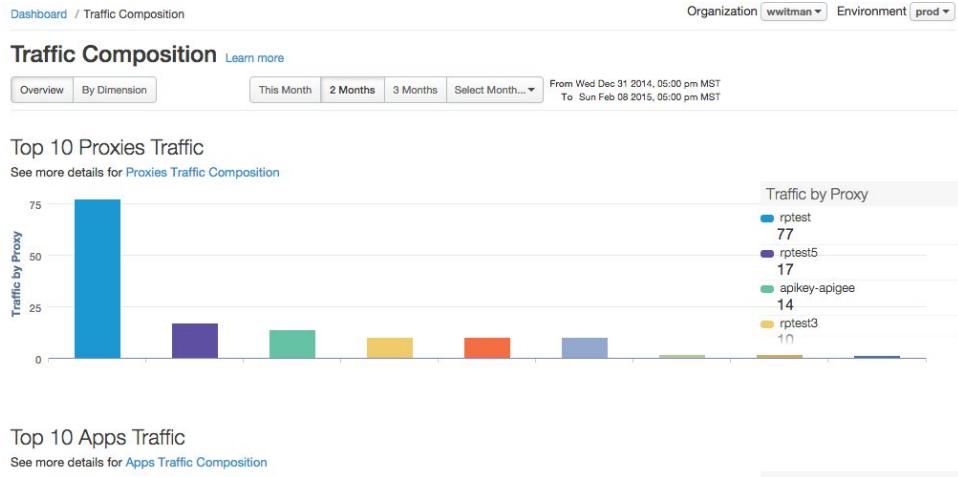


## トラフィック構成

[Traffic Composition] ダッシュボードでは、API プログラム全体に対する上位の API、アプリ、デベロッパー、プロダクトの影響を相対的に測定します。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

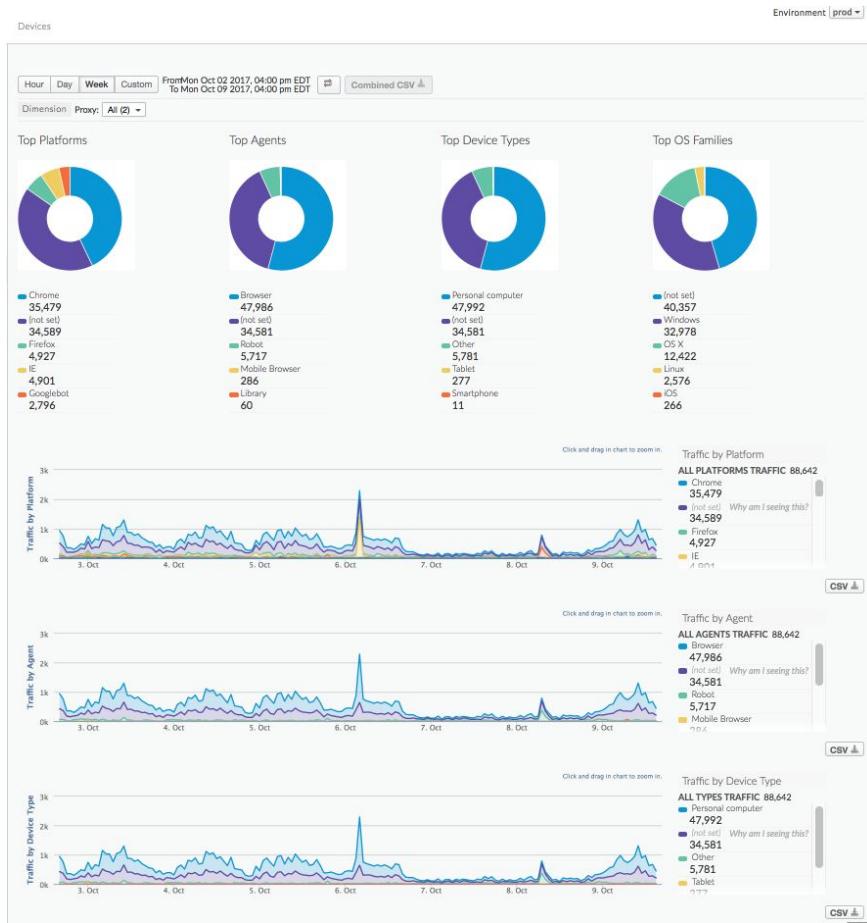


## デバイス

[Device] ダッシュボードには、APIへのアクセスに使用されているデバイスやサーバーに関する情報が表示されます。ユーザーが API にどのようにアクセスしているか、傾向をつかむことができます。たとえば、特定のタイプのデバイスからのトラフィックの増加や減少を把握して、変化に伴いアクションが必要となるかどうかを判断できます。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

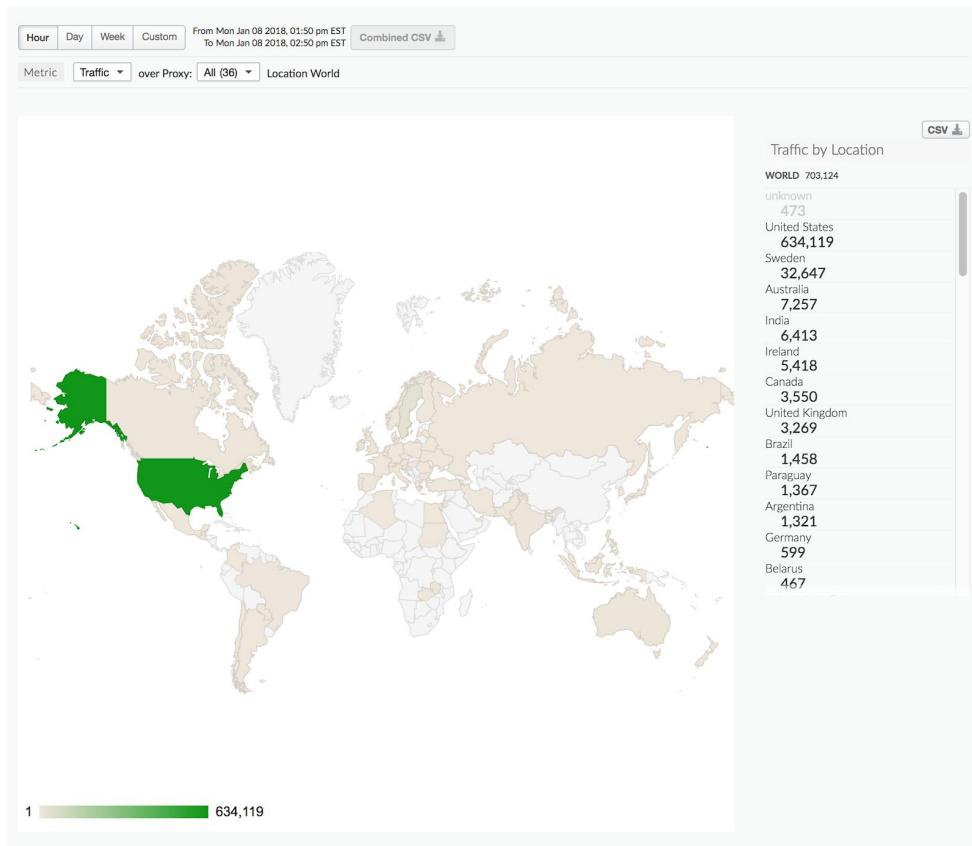


## Geomap

[Geo Map] ダッシュボードでは、さまざまな地理的場所のトラフィックパターン、エラーパターン、サービス品質を追跡できます。すべての API に関する情報を表示することも、特定の API の情報に絞り込むこともできます。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)



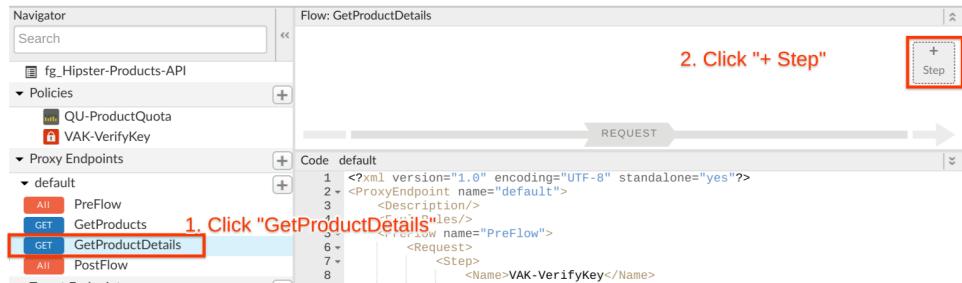
## カスタム分析レポートを作成する

### Statistics Collector を追加する

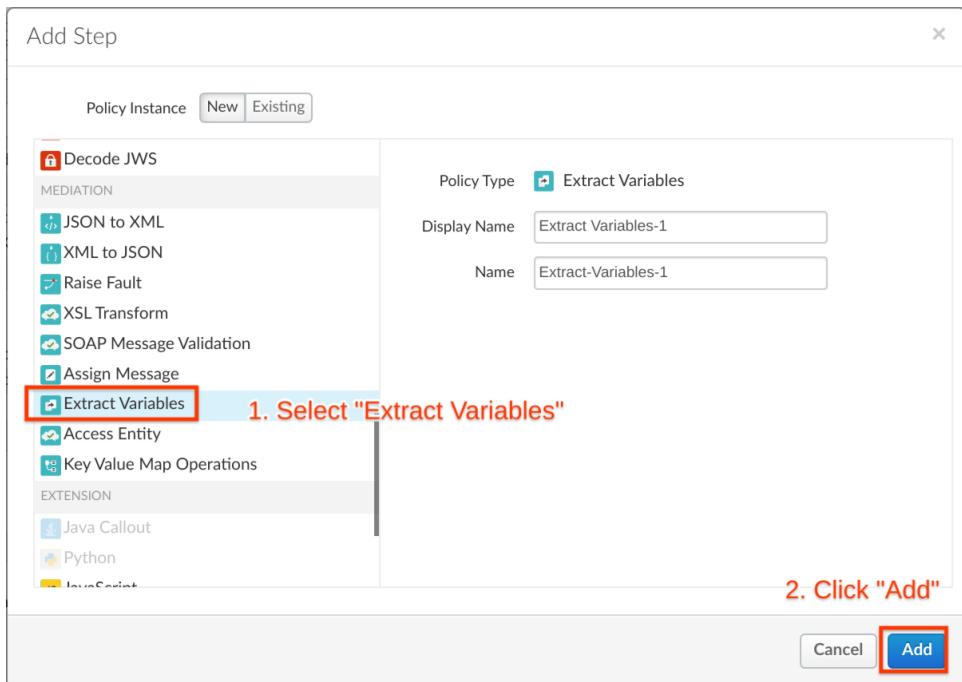
1. 横のナビゲーションメニューから [Develop] → [API Proxies] をクリックします。前のラボで作成した Hipster プロダクトの API プロキシを開き、[Develop] タブに移動します。
2. パスからデータを取得するには、最初にこれを Apigee Flow 変数に抽出する必要があります。それには、Extract Variables ポリシーを追加します。
3. [GetProductDetails] フローで [+ Step] ボタンをクリックして、リクエストに新しいステップを追加します。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)



4. [Mediation] カテゴリの下にある [Extract Variables] ポリシーを追加します。



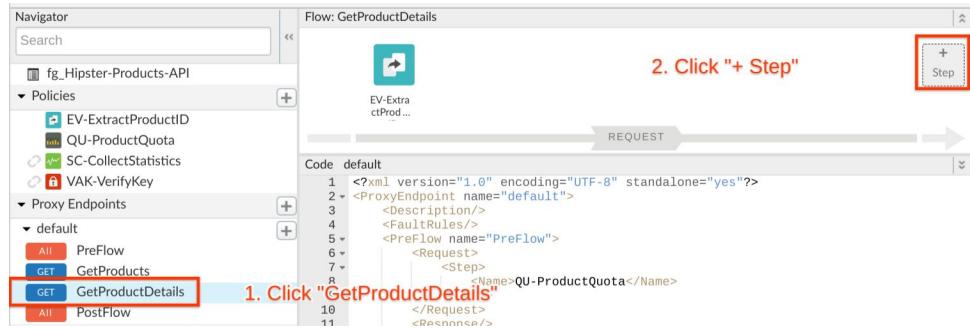
5. [Mediation] カテゴリの下にある [Extract Variables] ポリシーを追加します。

```
<ExtractVariables name="ExtractVariables-1">
  <DisplayName>EV-ExtractProductID </DisplayName>
  <Source>request</Source>
  <URIPath>
    <Pattern ignoreCase="true">/products/{productId}</Pattern>
  </URIPath>
  <VariablePrefix>uriRequest</VariablePrefix>
  <IgnoreUnresolvedVariables>true</IgnoreUnresolvedVariables>
</ExtractVariables>
```

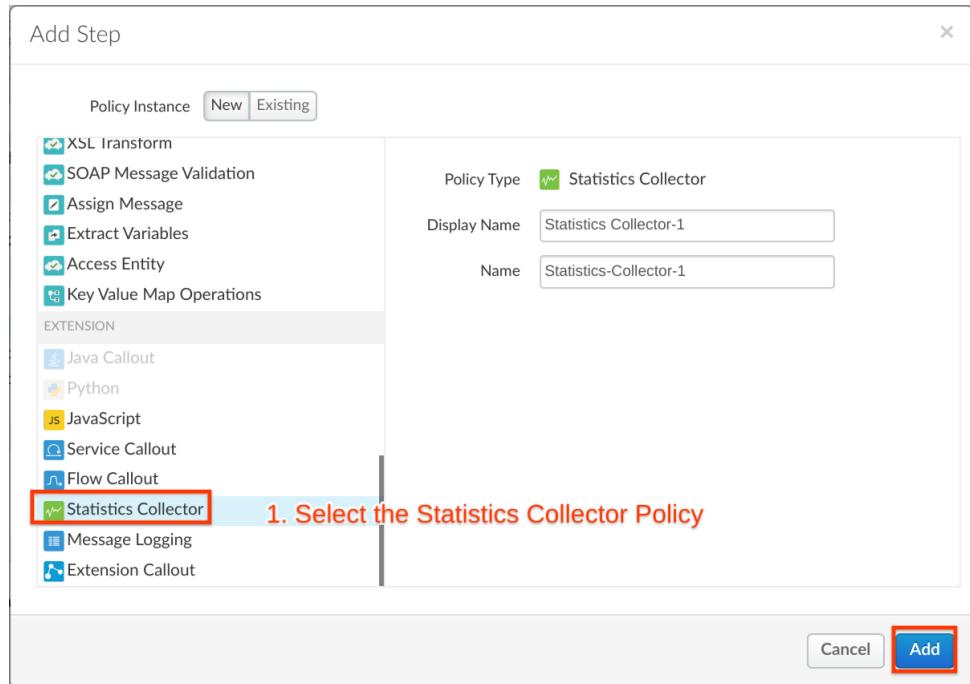
# Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

6. [GetProductDetails] で [+ Step] ボタンをクリックして、リクエストに新しいステップを追加します。



7. [Extensions] カテゴリの下にある [Statistics Collector] ポリシーを追加します。



8. 次の値を指定して新しい Statistics Collector ポリシーを編集します。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<StatisticsCollector async="false" continueOnError="false" enabled="true"
name="Statistics-Collector-1">
  <DisplayName>SC-CollectStatistics</DisplayName>
  <Properties/>
  <Statistics>
```

Google Cloud

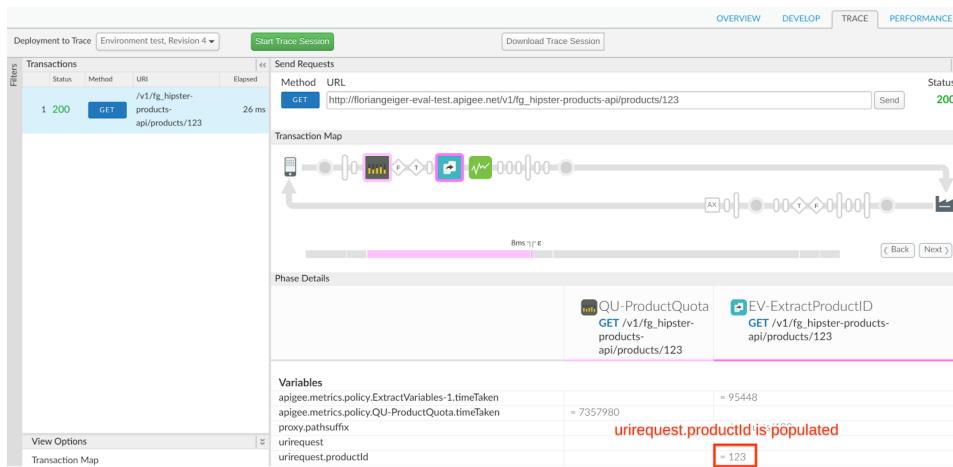
For more information visit [google.com/cloud](http://google.com/cloud)

```

<Statistic name="productId" ref="urirequest.productId" type="STRING"/>
</Statistics>
</StatisticsCollector>

```

9. トレース セッションを実行します。[Trace] タブをクリックし、[Start Trace Session] をクリックしてトレース セッションを開始します。クエリ パラメータとしていずれかのアプリから API キーを入力してから、[Send] をクリックします。[Extract Variables Policy] アイコンをクリックすると、変数内の値が表示されます。この値は Statistics Collector ポリシーによって記録されます。



さまざまな API からいろいろな API キーを試してみます。

注: それぞれの環境で初めて Statistics Collector ポリシーが呼び出された後、新しいカスタム ディメンションがデータベースに追加されるには 30 分ほどかかります。

たとえば、test 環境で Statistics Collector プロキシが呼び出されると、そのカスタム ディメンションのデータが test 環境で利用できるのは 30 分後になります。そして、カスタム ディメンションは、カスタム レポートの [Dimensions] プルダウン リストに表示されます。

## カスタム レポートを作成する

次に API プロキシで収集したデータを確認しましょう。直前に Statistics Collector ポリシーで定義した値を検索するカスタム レポートを作成する必要があります。

- [Analyze] > [Custom Reports] > [Reports] で [Custom Reports] メニューを開きます。

**Google Cloud**

For more information visit [google.com/cloud](http://google.com/cloud)

◀ ▶ Develop

↑ Publish

II Analyze **1. Open the "Analyze" Menu**

- Alert Rules
- ▶ API Metrics
- ▶ Developers
- ▶ End Users
- ▼ Custom Reports **2. Under "Custom Reports" click "Reports"**
- Reports

⚙ Admin

2. [+ Custom Report] をクリックして新しいカスタム レポートを作成します。

#### Custom Reports

1. Click "Custom Report"

REPORT NAME	ENVIRONMENT	METRIC AND DIMENSION	DESCRIPTION	LAST MODIFIED	ACTIONS
				0-0 of 0	<b>+ Custom Report</b>

3. わかりやすいレポート名とレポートの説明を入力します。また、Y 軸に表示する指標を定義します。カスタム統計変数を文字列ではなく整数で定義している場合、その変数もここに表示されます。

たとえば、プロキシのトラフィックを確認してみましょう。

さらに、トラフィックをグループ分けする際のディメンションも定義できます。ディメンションを複数選択すると、レポートをドリルダウンして、トラフィックの指標に関して詳しい情報を得ることができます。

Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

Custom Reports > New Custom Report

**Basics**

Report Name: Custom Products Report **1. Give the report a name**

Report Description:

Chart Type:  Column  Line  
 For Column charts, the x-axis represents groups designated by dimensions. For Line charts, the x-axis represents time.

**Metrics**

The y-axis represents metric values.

Metric	Aggregate Function	Actions
1 Traffic	<input checked="" type="radio"/> Sum <input type="radio"/> Average <input type="radio"/> Min <input type="radio"/> Max	+ Metric

**2. Select the "Traffic" metric**

**Dimensions**

Dimensions have two purposes:

- Initially, the dimension is used to group data, similar to the GROUP BY clause in SQL.
- Once a dimension is selected, it becomes a filter, similar to the WHERE clause in SQL, working as a drill down, and the subsequent dimensions becomes the grouping mechanism.

Dimension	Actions
1 Select...	+ Dimension

**3. Select our dimension**

**Filter**

Filter Conditions	Connector	Name	Operator	Value	Actions
					+ Filter Condition

4. このラボでは、この複数のディメンションを使用してドリルダウン機能を紹介します。  
 ここでは [Country]、[Developer App]、[productId] をディメンションに選択します。

Dimension	Actions
1 Country	<b>x Delete</b>
2 Developer App	<b>x Delete</b>
3 productId	<b>x Delete</b>

+ Dimension

5. 選択できるその他のディメンションをご覧ください。Statistics Collector ポリシーで定義した productId だけでなく、Apigee で定義済みの共通ディメンションがあることがわかります。

# Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

Select...

**Common Dimensions**

- Access Token
- Developer App
- Developer ID
- Flow Resource
- Proxy
- Proxy Client IP
- Proxy Path Suffix
- Request Verb
- Response Status Code
- Target Response Code
- Target URL
- User Agent
- X Forwarded For

**Custom Dimensions**

- productid
- searchterm

**Advanced Dimensions**

- API Product
- Cache Key

6. 選択が終わったら右下の [Save] をクリックします。
7. レポートのフィルタリングとして適当な範囲を選択します。すぐに開始するのか、時間がたってから確認するかを決めてください。デフォルトの1時間では短すぎるかもしれません。

Custom Products Report [Learn more](#)

**1. Click the date range**

Tue 3 Sep 2019 13:03 - Tue 3 Sep 2019 14:03 CEST

Dimension Country

Range Last Hour  
From An hour ago  
Duration 1 hour

To 14:03  
Unit minute

2. Select proper Range values

S	M	T	W	T	F	S	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15	16	17	18	19	20	21
22	23	24	25	26	27	28	29	30	31	22	23	24	25
26	27	28	29	30	31	29	30	31	1	2	3	4	5

Selected Tue 3 Sep 2019 13:03 - Tue 3 Sep 2019 14:03 CEST

Sum of Traffic IE 21

CSV

Summary

Country IE

21 Analyze

- 最初に選択したディメンションで分類されたトラフィックを確認してみましょう。こちらの例ではトラフィックのすべてがIEからのものです。実際のレポートはこの例とは違ったものになり、本番環境ではもっと多様な結果になるでしょう。

Custom Products Report [Learn more](#)

Chart Table Mon 2 Sep 2019 14:00 - Tue 3 Sep 2019 14:00 CEST Combined CSV

Dimension Country All

Proportion

Sum of Traffic IE 24

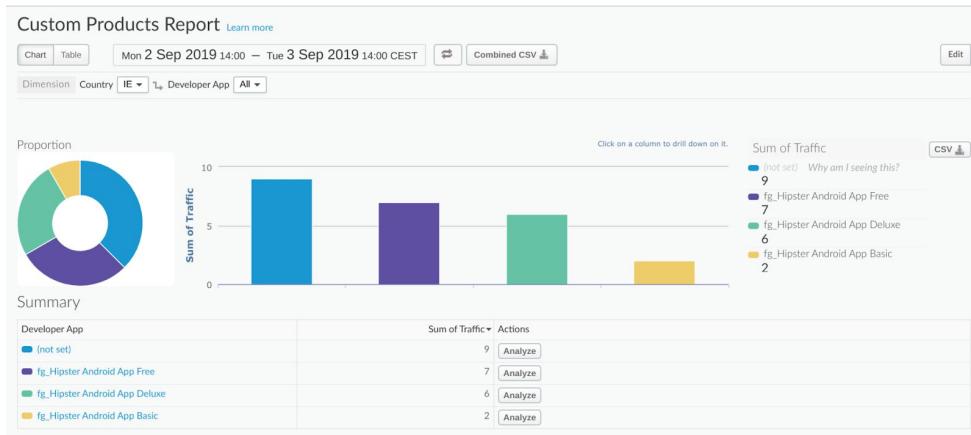
Sum of Traffic Actions 24

Summary

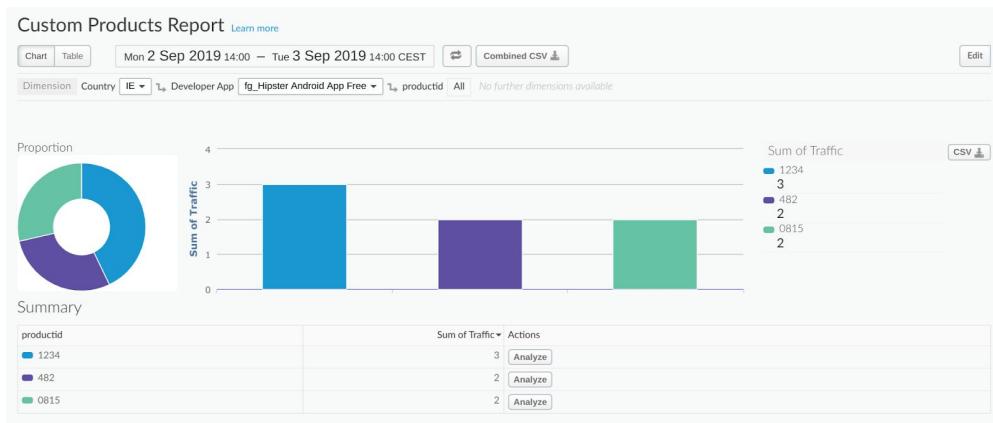
Country IE

- 青の列をクリックすると、レポートの詳細が表示されます。

次にアプリ別のトラフィックを確認しましょう。注目すべき、トラフィックが多いアプリを簡単に見つけることができます。



10. たとえば、[fg\_Hipster Android App Free] を見てみましょう。



11. 収集されたプロダクト ID 情報を見ることができました。

右側のダウンロードボタンをクリックすると、データを CSV 形式でダウンロードできます。

## 管理 API で統計情報を入手する

API呼び出しを簡単に試せる、管理 API ドキュメントを見てみましょう。

[https://apidocs.apigee.com/management/apis/get/organizations/%7Borg\\_name%7D/environments/%7Benv\\_name%7D/stats/%7Bdimension\\_name%7D-0](https://apidocs.apigee.com/management/apis/get/organizations/%7Borg_name%7D/environments/%7Benv_name%7D/stats/%7Bdimension_name%7D-0)



For more information visit [google.com/cloud](http://google.com/cloud)

## 値を入力して API呼び出しを実行する

このドキュメントでは自分の組織に対してインタラクティブに呼び出しを実行し、結果がどうなるか確認できます。

- API呼び出しデータを [Resource URL] フィールドの青い箇所に次のように入力します。
  - {org\_name}: 自分の組織
  - {env\_name}: 自分の環境
  - {dimension\_name}: apiproxy
- 表のデータは次のとおりです。
  - Select: sum(message\_count)
  - timeRange: 08/01/2019 00:00~08/30/2019 23:59
  - Filter: (apiproxy eq 'weather-analytics')

注意: 時間の範囲は、現在の終了範囲の日時に合わせて更新してください。

- 管理 UIへのログインに使用する Apigee ユーザー名とパスワードで認証します。
- [Send this request] ボタンをクリックしてリクエストを送信します。

## Resource URL

`https://api.enterprise.apigee.com/v1/organizations/  
florianeiger-eval/environments/test/stats/apiproxy`

## Query Parameters

Name	Values	Description
<b>select</b> <i>(required)</i>	<code>sum(message_count)</code>	<p>Designates one or more metrics to be aggregated for the report. For descriptions, see the <a href="#">Metrics</a>. The documentation on <a href="#">metrics</a> contains a table that specifies the metric name to use with the <b>select</b> query parameter and the function (sum, avg, min, max) supported by the metric.</p> <p>Following are examples of supported metrics and aggregate functions:</p> <pre>sum(ax_cache_executed) avg min max(ax_cache_l1_count) sum(cache_hit) sum(is_error) sum(message_count) sum(policy_error) avg min max(request_processing_latency) sum avg min max(request_size) avg min max(response_processing_latency) sum avg min max(response_size) sum(target_error) sum avg min max(target_response_time) sum avg min max(total_response_time) tps (Currently must use with another metric)</pre>
<b>timeRange</b> <i>(required)</i>	<code>08/01/2019 00:00~08/30</code>	The start and end time for the desired interval. The date format is MM/DD/YYYY HH:MM. For example, 04/15/2017 00:00~05/15/2017 23:59



For more information visit [google.com/cloud](http://google.com/cloud)

**1. Authenticate**

**2. Send a request**

HTTP Basic  
florianeige...

Send this request  
using the values above

Request Response cURL Reset

**HTTP/1.1 200 OK**

Access-Control-Allow-Headers: origin  
Access-Control-Allow-Methods: GET  
Access-Control-Allow-Origin: \*  
Access-Control-Max-Age: 3628800  
Connection: keep-alive  
Content-Length: 444  
Content-Type: application/json  
Date: Tue, 13 Aug 2019 15:58:51 GMT  
Server: Apigee LB

```
{
  "environments": [
    {
      "dimensions": [
        {
          "metrics": [
            {
              "name": "sum(message_count)",
              "values": [
                "402.0"
              ]
            }
          ],
          "name": "weather-analytics"
        }
      ],
      "name": "test"
    ],
    "metaData": {
      "errors": [],
      "notices": [
        "Source:Postgres",
        "Table used: edge.api.faxgroupusenondn014.agg_api",
        "PG Host:ruappg0ro.apigeeeks.net",
        "query served by:c9a876f8-8f0f-42d3-abe9-d0ce643ca330"
      ]
    }
  ]
}
```

**3. Check the result**

# Google Cloud

For more information visit [google.com/cloud](http://google.com/cloud)

## 理解度チェック

---

- API 仕様では productId のデータ型に数値が使用されているにもかかわらず、このラボでは文字列を使用したのはなぜでしょうか？
- プロキシフローに複数の Statistics Collector を配置するとどうなるでしょうか？

## まとめ

---

このラボでは API プログラムの成果とパフォーマンスを示し、API を最適化するための価値ある分析情報を提供するさまざまなレポートについて説明しました。

## 学習した内容

---

- ビルトイン Apigee ダッシュボードの概要
- カスタム統計データを収集するためのプロキシの変更
- API 呼び出しから収集したデータを使用してカスタム レポートを作成し、ディメンションを定義してレポートをドリルダウンする

## 参考資料

---

Apigee ドキュメントへのリンク

<https://docs.apigee.com/api-platform/analytics/use-analytics-api-measure-api-program-performance>

Analytics ダッシュボードの概要 -

<https://docs.apigee.com/api-platform/analytics/using-analytics-dashboards>

Extract Variables ポリシー -

<https://docs.apigee.com/api-platform/reference/policies/extract-variables-policy>

Statistics Collector ポリシー -

<https://docs.apigee.com/api-platform/reference/policies/statistics-collector-policy>

カスタム レポートの作成と管理 -

<https://docs.apigee.com/api-platform/analytics/create-custom-reports>



Statistics Collector を使用したカスタム レポートの例 -

<https://docs.apigee.com/api-platform/analytics/analyze-api-message-content-using-custom-analytics>

## 動画 (4mv4d)

---

Extract Variables ポリシー - [https://www.youtube.com/watch?v=KqFpnNt\\_8yo](https://www.youtube.com/watch?v=KqFpnNt_8yo)

API の測定と KPI - <https://www.youtube.com/watch?v=x6r7xHKNINU>

API プログラムの成果の測定とレポート - <https://www.youtube.com/watch?v=VcNKpoakVqQ>



For more information visit [google.com/cloud](http://google.com/cloud)