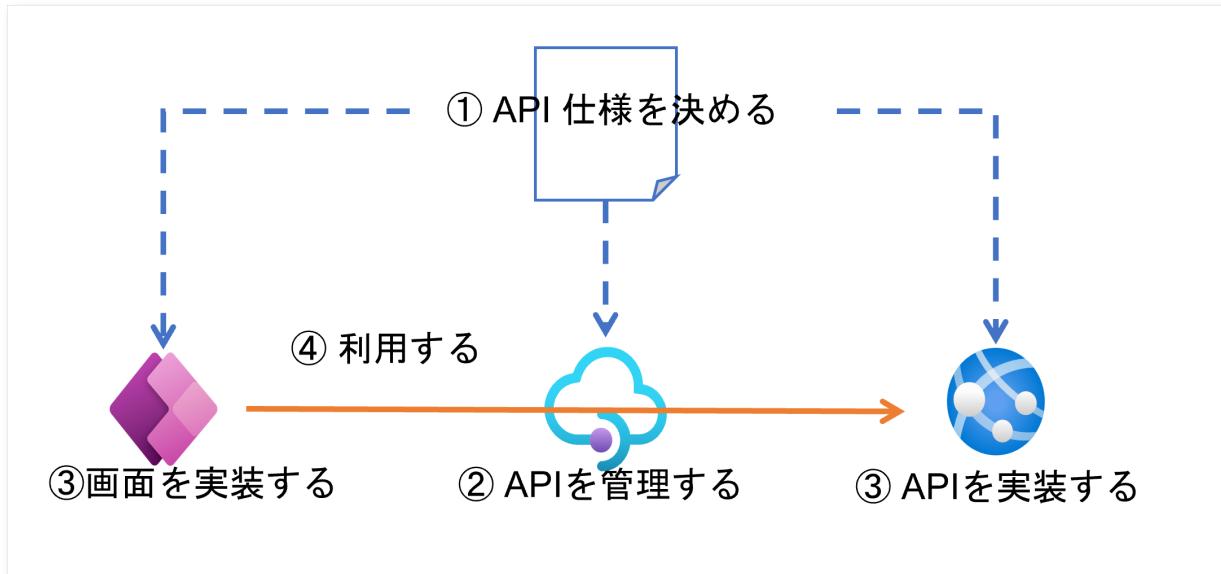


Fusion 開発ハンズオン概要

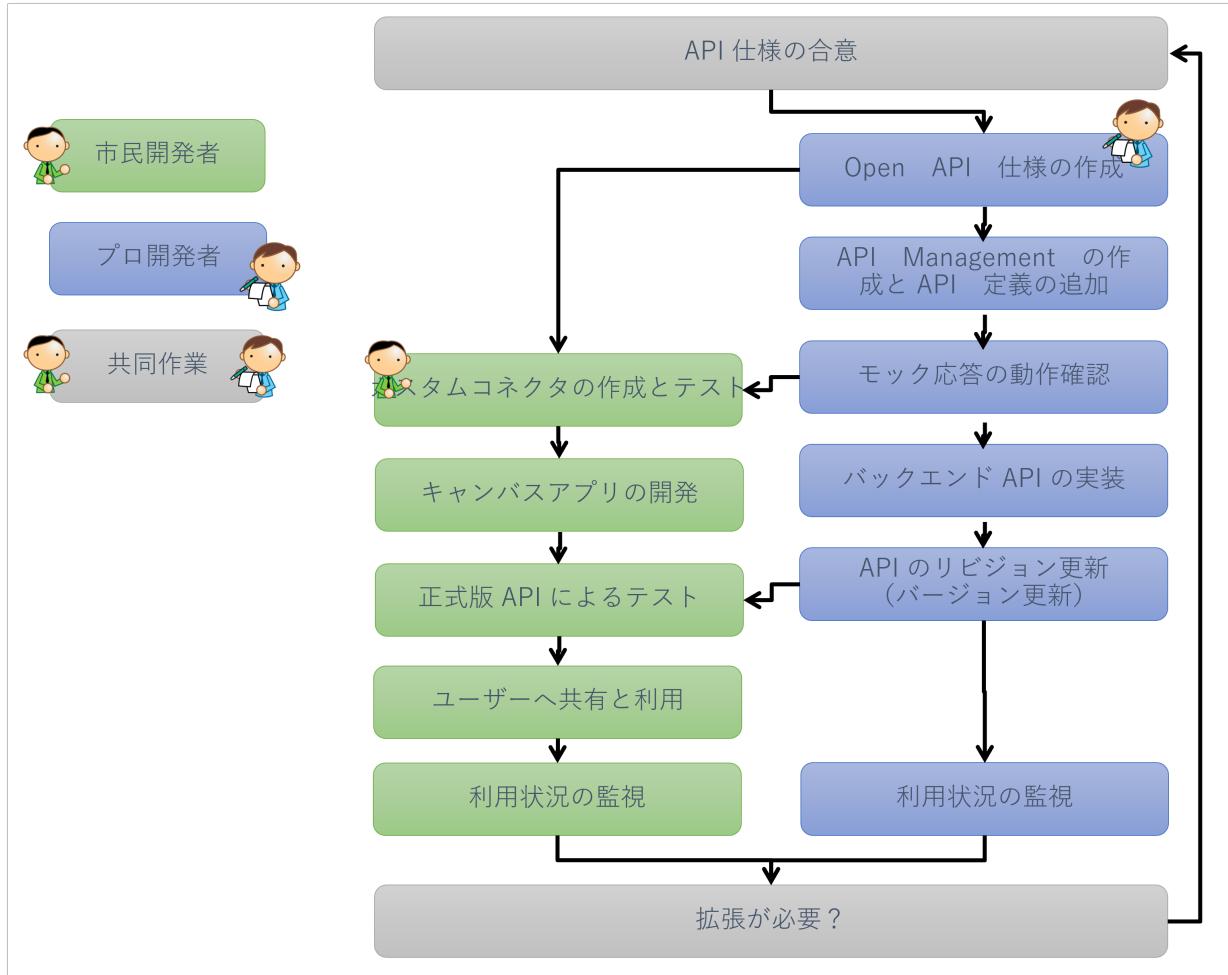
ハンズオンで構築する内容

本ハンズオンでは Power Apps キャンバスアプリから Azure API Management を介して、Azure App Service 上でホストされたカスタム API を呼び出すアプリケーションを構築します。



ハンズオンの作業フロー

ハンズオンでは下記のような構築作業を行います。各参加者が一連の作業を全て実施するため、実業務のような役割分担は行いません。

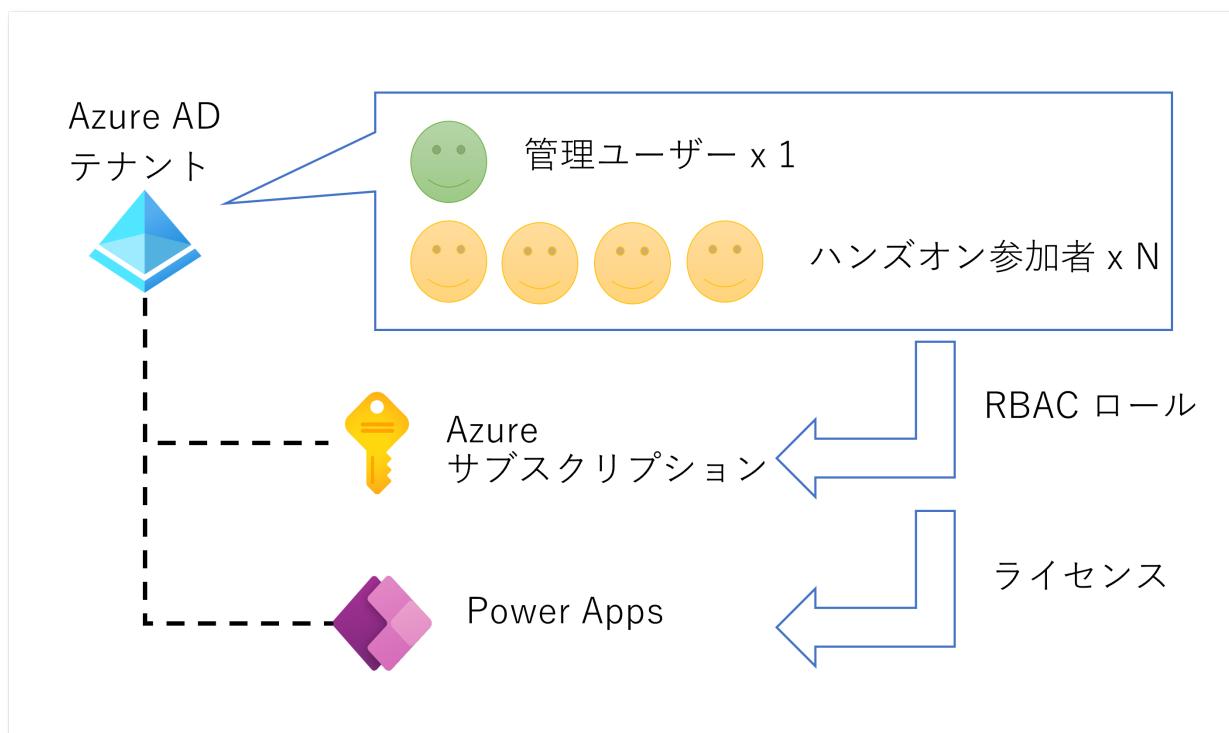


Section 00：ハンズオンの環境確認

ハンズオン環境

ハンズオンを実施するためには下記の環境が必要になります。

- Azure Active Directory テナント
 - ハンズオン参加者の人数分のユーザー
 - ユーザーの登録・管理権限を持った管理ユーザー
- 上記テナントに関連付けられた Azure サブスクリプション
 - 管理ユーザーはサブスクリプションの所有者(Owner)ロールに割り当てる
 - ハンズオン参加者はサブスクリプションの共同作成者(Contributor)ロールに割り当てる
 - 無償評価版等も可
- 上記テナントに関連付けられた Power Apps
 - 管理ユーザーおよびハンズオン参加者ユーザーは Power Apps per user プランに割り当てる
 - 開発者プランや評価版でも可



各ユーザーへのライセンス割り当て状況は [Microsoft 365 管理センター](#)で、Azure RBAC ロールの割り当て状況は [Azure ポータル](#) で確認出来る。

The screenshot shows two side-by-side views in the Microsoft 365 Admin Center.

Left View (Azure AD Users):

- Header: Microsoft 365 管理センター
- Breadcrumb: ホーム > アクティブなユーザー
- Title: アクティブなユーザー
- Section: 推奨処置 (2)
- Table:
 | ユーザー名 | ライセンス |
| --- | --- |
| Fusion開発環境構築 | Power Apps per user plan |
| Sample User01 | Power Apps per user plan |
| Sample User02 | Power Apps per user plan |
| Sample User03 | Power Apps per user plan |

Right View (Azure IAM Roles):

- Header: Microsoft 365 管理センター | ロール割り当て (IAM) | ロールの割り当て
- Section: このサブスクリプションにおけるロール割り当ての数: 36
- Table:
 | 名前 | 種類 | 役割 |
| --- | --- | --- |
| Sample User01 | ユーザー | 共同作成者 |
| Sample User02 | ユーザー | 共同作成者 |
| Sample User03 | ユーザー | 共同作成者 |
| System Administrator | 所有者 | 所有者 |

Prefix の確認

本ハンズオンでは複数の参加者が共通のAzure サブスクリプションや PowerApps 環境を使用して作業を行います。このため各参加者がユニークになる **Prefix** を決めておいてください。自身の名前などをつけると分かりやすくなります。ハンズオンの手順の中で `${Prefix}` の記載があった場合には、ここで決めた Prefix の文字列に読み替えて作業を行ってください。

Section 01 : API 仕様の合意

ここでは Todo リストを管理する API を設計します。

データの定義

まず 1 つの Todo アイテムを表すデータは以下のようになるものとします。

- 一意に識別するための `id`
- 題名を表す `title`
- 具体的な作業内容を表す `task`
- 有効、完了等の状態を表す `status`
- 締め切りを表す `duedate`
- 最終更新日時を表す `updatedOn`

json

```
{  
    "id": 100,  
    "title": "Task1",  
    "task": "経費を清算する",  
    "status": "Active",  
    "duedate": "2023-03-20T08:14:05.0872495+00:00",  
    "updatedOn": "2023-03-14T08:14:05.0872065+00:00"  
}
```

操作の定義

API では上記で定義した Todo を扱うために必要な以下の操作を提供するものとします。

Path	Method	意味
/todo/v1	GET	登録されている全ての Todo アイテムを取得する
/todo/v1	POST	Todo アイテム新規に登録する
/todo/v1/{id}	GET	指定した id に該当する Todo を 1 件取得する
/todo/v1/{id}	PUT	指定した id に該当する Todo の内容を変更する
/todo/v1/{id}	DELETE	指定した id に該当する Todo を削除する

Power Apps キャンバスアプリはこれらの操作を使用してユーザーインターフェースを提供します。

Section 02 : Azure API Management の作成

この後で構築する API を管理するための Azure API Management を作成します。

!

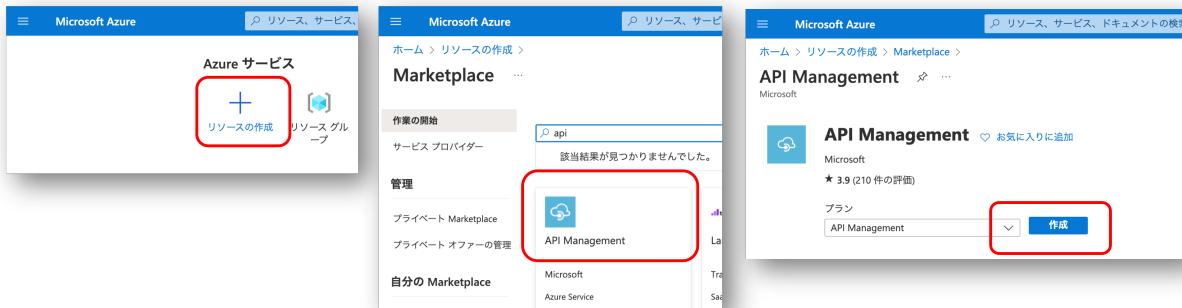
本セクションの作業内容は代表者が 1 回だけ行ってください。

Azure サインイン

まず [Azure ポータル](#) を開きます。必要に応じて管理者ユーザー名とパスワード等を使用してサインインしてください。

API Management の作成

- リソースの作成 をクリック
- API を検索し、API Management を選択
- API Management を作成



- 基本 タブ
 - リソースグループは 新規作成 を選択し rg-fusiondev-shared を入力
 - リージョンは (Asia Pacific) Japan East を選択
 - リソース名は apim-fusiondev-shared-MMDD を入力
 - MMDD はハンズオン実施日を指定してください
 - Organization name は フュージョン開発ハンズオン MMDD を入力
 - 管理者のメールアドレスを入力
 - 価格レベルは Developer (SLA なし) を選択
 - 次へ：監視 を選択
- 監視 タブでは特に何も入力せずに 次へ を選択(画面キャプチャ省略)
- スケーリング タブでは特に何も入力せずに 次へ を選択(画面キャプチャ省略)
- マネージド ID タブでは特に何も入力せずに 次へ を選択(画面キャプチャ省略)
- 仮想ネットワーク タブでは特に何も入力せずに 次へ を選択(画面キャプチャ省略)
- プロトコル設定 タブでは特に何も入力せずに 次へ を選択(画面キャプチャ省略)
- タグ タブでは特に何も入力せずに 次へ を選択(画面キャプチャ省略)
- タグ タブでは特に何も入力せずに 次へ を選択(画面キャプチャ省略)
- 確認とインストール タブ
 - 作成 を選択

API Management ゲートウェイのインストール

基本

プロジェクトの詳細
デプロイされたリソースとコストを管理するためのサブスクリプションを選択します。フォルダーなどのリソース グループを使用すると、すべてのリソースを整理して管理することができます。

サブスクリプション * ① Visual Studio Enterprise サブスクリプション

リソース グループ * ① fusiondev0118-rg
新規作成

インスタンスの詳細
リージョン * ① (Asia Pacific) Japan East

リソース名 * fusiondev-apim-0118

ワークスペース名 * ① workspace1

管理者のメール アドレス * ① ayumu.inaba@live.com

価格レベル
API Management の価格レベルは、ユニットごとのコンピューティング容量と提供される機能セット(たとえば、仮想ネットワーク、複数リージョンのデプロイ、セルフホスティッド ゲートウェイのサポートなど)によって異なります。さらに多くの API 要求に対応するには、代わりに API Management サービス ユニットを追加することをご検討ください。[詳細情報](#)

⚠️ Developer レベルの API Management には SLA が含まれていないため、運用目的で使用しないでください。たとえば アップグレード中などに、サービスの停止が断続的に発生する場合があります。[詳細情報](#)

価格レベル ① Developer (SLAなし)

Create API Management service

確認とインストール

使用条件
[作成] をクリックすることで、お客様は (a) 上記の Marketplace のオファリングに関する法律条項とプライバシーに関する声明に同意し、(b) Microsoft および、そのオファリングに関連する料金が、現在の支払い方法に対して Azure サブスクリプションと同じ請求頻度で請求されることを認め、かつ、(c) Microsoft がお支払の連絡先情報を、使用量情報、取引に関する情報を、サポート、請求、その他の取引上のアクティビティのために、オファリングのプロバイダーと共に共有する可能性があることに同意するものとします。Microsoft は、カードデータのオファリングに対する権利は提供しません。その他の詳細については [Azure Marketplace 使用条件](#) をご覧ください。

基本

サブスクリプション	ME-MngEnvMCAP784488-ainaba-1
リソース グループ	rg-fusiondev-shared
リージョン	Japan East
リソース名	apim-fusiondev-shared-0324
Organization name	フュージョン開発ハンズオン 0324
管理者のメール アドレス	ayumu.inaba@live.com
価格レベル	Developer

監視

Application Insights	無効
Application Insights インスタンス	-

スケーリング

ユニット	1
------	---

マネージド ID

ID の種類	なし
--------	----

仮想ネットワーク

作成 < 前へ 次へ > オートメーション用のテンプレートをダウンロードする

デプロイが進行中です という画面が表示されたら完了までしばらく待ちます。

Section 03 : Todo API の定義

以降では共有の API Management に対して、ハンズオン参加者が個別に API を定義していきます。他の参加者が作成中の API を壊さないように注意してください。

API Management の確認

ハンズオン参加者は各自に割り当てられたユーザーと初期パスワードを使用して [Azure ポータル](#) にサインインしてください。初めてサインインするときには初期パスワードからの変更を求められますので、新しいパスワードを入力します。設定した新しいパスワードを忘れないようにしてください。

- 画面左上のメニューから **リソースグループ** を選択
- 管理者が代表で作成した **rg-fusiondev-shared** という名前のリソースグループを選択
- 管理者が代表で作成した **apim-fusiondev-shared-MMDD** という名前の API Management を選択



The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo, a search bar, and various icons for account management. Below the header, the URL is [641d55625d46a6b9bd30bee4 | 概要](#). The main content area is titled "rg-fusiondev-shared" and shows a list of resources under the "リソース グループ" section. On the left, there's a sidebar with links for Activity Log, Access Control (IAM), Tags, Resource Visualizer, Events, Deployment, and Metrics. The main table lists one item: "apim-fusiondev-shared-0324" with columns for Name, Type, and Location (Japan East). Filter and sorting options are available at the bottom of the table.

API の新規作成

- **API** メニューを選択
- **+ Add API** を選択
- **Define a new API** セクションの **HTTP** を選択

The screenshot shows the Microsoft Azure API Management API designer interface. On the left, there's a sidebar with 'All APIs' and 'Todo API ainaba'. Under 'Todo API ainaba', 'v1' is selected. The main area shows a 'Frontend' configuration for an operation. The 'Display name' is 'List All Todo Items', 'Name' is 'list-all-todo-items', 'URL' is 'GET /', and 'Description' is empty. The 'Tags' field contains 'e.g. Booking'. Below this, tabs for 'Template', 'Query', 'Headers', 'Request', and 'Responses' are shown, with 'Responses' being the active tab. Under 'Responses', a '200 OK' response is defined with an empty 'Description'. The 'Representations' section shows a table with one row for 'application/json', which has a sample value of '[{"id": 100, "title": "Task1", "task": "経費を清算する", "status": "P"}]'. The 'Headers' section is empty. At the bottom, there are 'Operations' and 'Definitions' tabs, and buttons for 'Save' and 'Discard'.

- Basic を Full に切替
- Display Name 欄に Todo API \${Prefix} を入力
- Name 欄が自動的に todo-api-\${prefix} と入力される
- Web service URL 欄は空のまま入力しない
- URL scheme 欄は HTTPS を選択
- API URL suffix 欄に todo-api-\${prefix} を入力
- Tags 欄は空のまま入力しない
- Products 欄は Starter を選択
- Gateways 欄は Managed が選択されているのでそのまま何もしない
- Version this API ? 欄を チェック する
- Version identifier 欄に v1 を入力
- Versioning scheme 欄は Path を選択
- Usage example 欄が https://apim-fusiondev-shared-MMDD.azure-api.net/todo-api-\${prefix}/v1 となっていることを確認
- Create を選択

Create an HTTP API

Basic Full

* Display name	Todo API ainaba
* Name	todo-api-ainaba
Description	
Web service URL	e.g. http://httpbin.org
URL scheme	<input type="radio"/> HTTP <input checked="" type="radio"/> HTTPS <input type="radio"/> Both
API URL suffix	todo-api-ainaba
Base URL	https://fdsample0317-apim.azure-api.net/todo-api-ainaba
Tags	e.g. Booking
Products	Starter
Gateways	Managed x
Version this API?	<input checked="" type="checkbox"/>
* Version identifier	v1
* Versioning scheme	Path
Usage example	https://fdsample0317-apim.azure-api.net/todo-api-ainaba/v1

Create Cancel

Todo アイテム データ構造の定義

Todo アイテム 1 件分のデータ構造を定義します。

- 作成時に Display Name に入力した Todo API \${prefix} のバージョン v1 が選択されていることを確認
- Design タブを選択
- Definitions に切替
- + Add definition を選択
- Definition Name に TodoItem を入力
- Samle (JSON) 欄に下記のサンプルデータを入力

```
{
  "id": 1,
  "title": "Task1",
  "task": "Do Task 1",
  "status": "Active",
  "duedate": "2023-03-20T08:14:05.0872495+00:00",
  "updatedOn": "2023-03-14T08:14:05.0872065+00:00"
}
```

- 自動生成された Payload 欄を修正
 - duedate プロパティに `"format": "date-time"`, を追加
 - updatedOn プロパティに `"format": "date-time"`, を追加
- Create definition** を選択
- Example 欄に上記のサンプルデータを入力
- Save を選択

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

- Left Sidebar:** Shows 'All APIs' with 'Todo API ainaba' selected. Below it, 'v1' is highlighted.
- Top Bar:** Shows 'REVISION 1' and 'CREATED Mar 24, 2023, 5:13:31 PM'.
- Central Area:**
 - Create new definition:** A modal window titled 'Create new definition' is open. It has a 'Definition for' dropdown set to 'Todo API ainaba > New definition' and a 'Name' input field set to 'Todoitem'. A 'Sample (JSON)' code editor contains the JSON provided at the top of the page.
 - Definition Body:** A separate modal window titled 'Definition' is open. It has a 'Name' input field set to 'Todoitem' and a 'Schema' code editor containing the same JSON sample.

Todo 一覧 データ構造の定義

Todo アイテム 1 件分のデータ構造を参照して、Todo 一覧のデータ構造を定義します。

- + Add definition を選択
- Definition Name に TodoList を入力
- Samle (JSON) 欄は空のまま何も入力しない
- Payload 欄には下記を入力 (OpenAPI 3.0 形式)

json

```
{  
    "type": "array",  
    "items": {  
        "$ref": "#/components/schemas/TodoItem"  
    }  
}
```

- Create definition を選択
- Example 欄に上記のサンプルデータを入力
- Save を選択

```
[
  {
    "id": 1,
    "title": "Task1",
    "task": "Do Task 1",
    "status": "Active",
    "duedate": "2023-03-20T08:14:05.0872495+00:00",
    "updatedOn": "2023-03-14T08:14:05.0872065+00:00"
  },
  {
    "id": 2,
    "title": "Task2",
    "task": "Do Task 2",
    "status": "Done",
    "duedate": "2023-03-19T08:14:05.0872875+00:00",
    "updatedOn": "2023-03-14T08:14:05.0872868+00:00"
  },
  {
    "id": 3,
    "title": "Task3",
    "task": "Do Task 3",
    "status": "Active",
    "duedate": "2023-03-20T08:14:05.0872885+00:00",
    "updatedOn": "2023-03-13T08:14:05.0872879+00:00"
  }
]
```

The screenshot shows the 'Create new definition' interface for a 'Todo API alnaba > New definition'. It displays two panels: 'Definition' and 'Definition body'.

Definition Panel:

- Name:** TodoList
- Definition body:** A JSON schema for 'TodoList'.

```
{
  "type": "array",
  "items": {
    "$ref": "#/definitions/TodoItem"
  }
}
```

Definition Body Panel:

- Name:** TodoList
- Definition body:** A detailed JSON schema for 'TodoList'.

```

{
  "id": 1,
  "title": "Task1",
  "task": "Do Task 1",
  "status": "Active",
  "duedate": "2023-03-20T08:14:05.0872495+00:00",
  "updatedOn": "2023-03-14T08:14:05.0872065+00:00"
},
{
  "id": 2,
  "title": "Task2",
  "task": "Do Task 2",
  "status": "Done",
  "duedate": "2023-03-19T08:14:05.0872875+00:00",
  "updatedOn": "2023-03-14T08:14:05.0872868+00:00"
},
{
  "id": 3,
  "title": "Task3",
  "task": "Do Task 3",
  "status": "Active",
  "duedate": "2023-03-20T08:14:05.0872885+00:00",
  "updatedOn": "2023-03-13T08:14:05.0872879+00:00"
}

```

API 仕様の定義

ここでは Todo アイテムの一覧を取得する API 操作を作成します。

- 作成時に Display Name に入力した Todo API \${prefix} のバージョン v1 が選択されていることを確認
- Design タブを選択
- Operations に切替
- + Add operation を選択
- Display Name に List All Todo Items を入力
- Name は自動的に list-all-doto-items が入力される
- URL は GET を選択、パスはスラッシュ / のみを入力
- Responses タブを選択
- + Add response では 200 OK を選択
- + Add representation では application/json を選択
- Sample 欄には上記の Todo 一覧のサンプルデータを入力
- DEFINITION 欄で TodoList を選択
- Save を選択

The screenshot shows the API design interface for creating a new operation. On the left, the sidebar shows the API structure: Todo API ainaba > v1. The 'Operations' tab is selected. In the main area, under 'Frontend', the 'Add operation' button is highlighted. The 'Display name' field contains 'List All Todo Items', 'Name' contains 'list-all-doto-items', and 'URL' is set to 'GET /'. The 'Responses' tab is active, showing a '200 OK' entry with an empty 'Description' field. Under 'Representations', there is one entry for 'application/json' with 'TodoList' selected in the 'DEFINITION' dropdown. The 'Headers' section is empty. At the bottom, the 'Save' button is highlighted.

補足

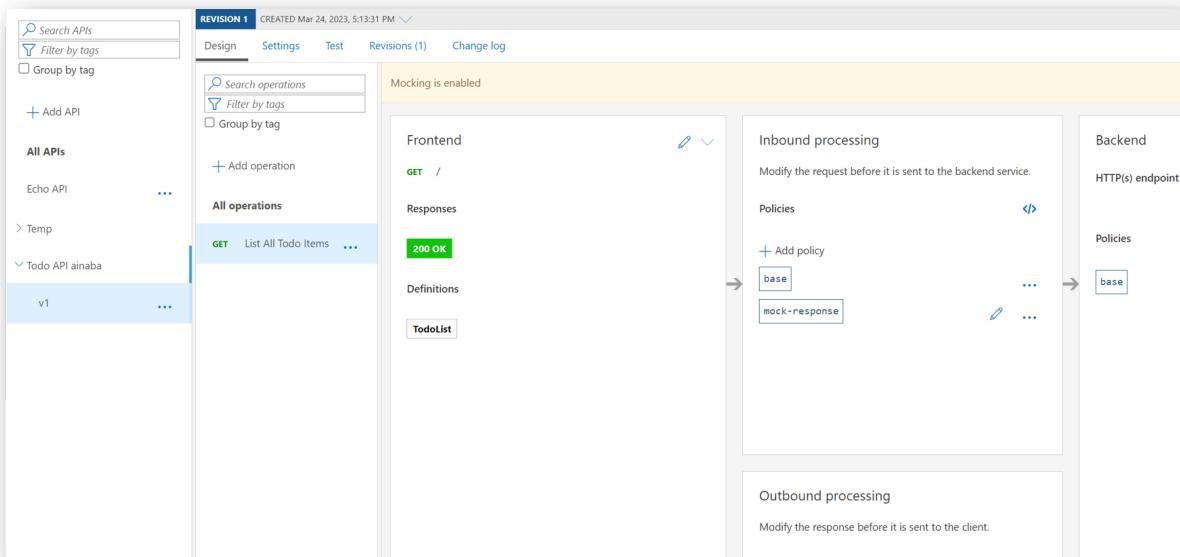
ここでは单一の操作しか定義していませんが、後の手順で [Section 01](#) で設計したその他の操作も作成します。

Section 04：モック応答の有効化とテスト実行

まだ API Management のバックエンドで動作する API を実装していませんので、このままでは API が動作しません。API 実装には時間がかかるため、ここではモック応答を有効にしてダミーのデータを返すように構成しておくことで、キャンバスアプリの開発に着手することが出来るようにしておきます。

モック応答の有効化

- 先ほど作成した `List All Todo Items` が選択されていることを確認
- Inbound processing 欄で `+ Add policy` を選択
- `Mock responses` ポリシーを選択
- API Management response 欄で `200 OK, application/json` を選択
- `Save` を選択
- `Mocking is enabled` が表示されることを確認



API のテスト実行

- モック応答を有効にした `List All Todo Items` を選択します
- `Test` タブに切り替え
- `Send` ボタンをクリックすると、サンプルデータが返却されることを確認

The screenshot shows the Microsoft Azure API Management Test interface. On the left, there's a sidebar with search and filter options, and a tree view of APIs: Echo API, Temp, and Todo API ainaba (selected). Below that is a v1 section. At the top right, it says "REVISION 1 CREATED Mar 24, 2023, 5:13:31 PM". The main area has tabs for Design, Settings, Test, Revisions (1), and Change log. The Test tab is selected. It shows a "Todo API ainaba > List All Todo Items > Console" path. A note says "Mocking is enabled". Under "HTTP request", a GET method is selected with the URL "GET https://fdsample0317-apim.azure-api.net/todo-api-ainaba/v1/" and "Host: fdsample0317-apim.azure-api.net". Under "HTTP response", the status is "HTTP/1.1 200 OK". The response body is a JSON array of three tasks:

```
[{"id": 1, "title": "Task1", "task": "Do Task 1", "status": "Active", "dueDate": "2023-03-20T08:14:05.0872495+00:00", "updatedOn": "2023-03-14T08:14:05.0872065+00:00"}, {"id": 2, "title": "Task2", "task": "Do Task 2", "status": "Done", "dueDate": "2023-03-19T08:14:05.0872875+00:00", "updatedOn": "2023-03-14T08:14:05.0872868+00:00"}, {"id": 3, "title": "Task3", "task": "Do Task 3", "status": "Active", "dueDate": "2023-03-20T08:14:05.0872885+00:00", "updatedOn": "2023-03-13T08:14:05.0872879+00:00"}]
```

At the bottom, there are buttons for Send, Trace, and CORS proxy settings.

補足

テスト実行結果画面に表示されているように、下記の URL に対する HTTP GET 操作で Todo アイテムの一覧を取得することができました。

`https://apim-fusiondev-shared-MMDD.azure-api.net/todo-api-${prefix}/v1/`

これは以下の要素の組み合わせになっていることが分かります。

要素	値	説明
Host	apim-fusiondev-shared-MMDD	API Managementn インスタンス単位(共有)
API	todo-api-\${prefix}	API 定義を作成時に指定した API URL Suffix(ハンズオン参加者単位で作成)
Version	v1	API 定義を作成時に指定した Version Identifier
Method	GET	List All Todo Items 作成時に指定した HTTP GET
Path	/	List All Todo Items 作成時に指定したパス(ルート)

API Managementn インスタンスは共有していますが、バージョン単位で独立した API がホストされ、Method と Path で操作を切り替えていくことになります。

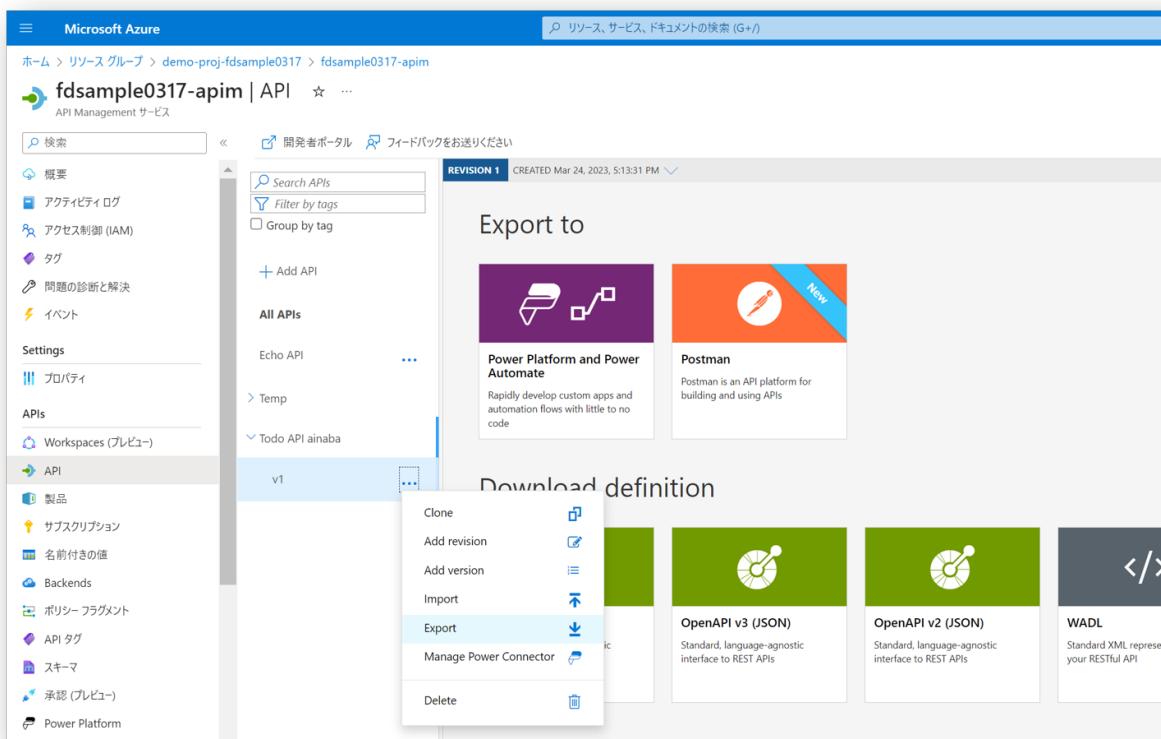
Section 05：カスタムコネクタの作成

作成した Todo API を呼び出すためのカスタムコネクタを作成します。

OpenAPI 仕様書をダウンロード

まず API の仕様書を入手します。

- 作成した Todo API \${prefix} API の v1 を選択
- コンテキストメニュー(….)の Export を選択
- OpenAPI v2 (JSON) を選択、ローカルPCにダウンロードしておく



API キーの作成と取得

API を呼び出すためには API キーが必要になりますので、そちらも作成・取得します。

- API Management の サブスクリプション メニューを選択
- + サブスクリプションの追加 を選択
- 名前に todoapi-{\$prefix}-subsc を入力
- 表示名に \${prefix}'s subscription を入力
- スコープは API を選択
- API は自身が作成した Todo API \${prefix} v1 を選択
- ユーザーは空欄のまま
- 作成 を選択

ホーム > リソース グループ > rg-fusiondev-shared > apim-fusiondev-shared-0324

apim-fusiondev-shared-0324 | サブスクリプション ☆ ...

検索

+ サブスクリプションの追加 列 最新の情報に更新

API コンシューマーは、製品を登録して API の使用を開始できます。詳細情報

検索

状態 すべて 承認待ち

表示名 主キー 2 次キー スコープ 状態

表示名	主キー	2 次キー	スコープ	状態
(todoapi-ainaba-subsc)	*****	*****	製品: Starter	アクティブ
(ainaba subscription)	*****	*****	製品: Unlimited	アクティブ
Built-in all-access su...	*****	*****	サービス	アクティブ

新しいサブスクリプション

サブスクリプション

名前 *

todoapi-ainaba-subsc

表示名

ainaba subscription

トレースを許可

スコープ

API

API

Todo API ainaba v1

製品

ユーザー

通知の送信の対象

開発者ポータル

- 製品: Starter スコープのサブスクリプションの一番右にあるメニュー (….) を選択
- キーの表示/非表示 を選択してキーを表示
- 主キー に表示され値をコピーして控えておく

+ サブスクリプションの追加 列 最新の情報に更新

API コンシューマーは、製品を登録して API の使用を開始できます。詳細情報

検索

状態 すべて 承認待ち

スコープ すべて

表示名 主キー 2 次キー スコープ 状態 所有者 トレースを許可

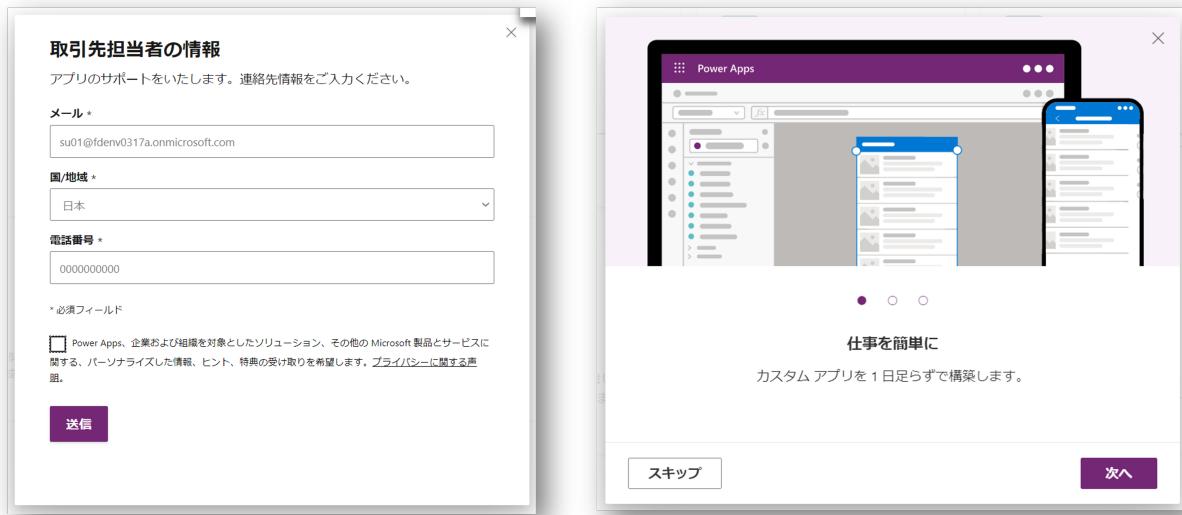
表示名	主キー	2 次キー	スコープ	状態	所有者	トレースを許可
(todoapi-ainaba-subsc)	*****	*****	製品: Starter	アクティブ	Administrator	X ...
(ainaba subscription)	*****	*****	製品: Unlimited	アクティブ	Administrator	X ...
Built-in all-access subscr...	*****	*****	サービス	アクティブ		X ...
ainaba subscription	74b2b4dff7344188... [Copy]	8f97444e5fb7439d... [Copy]	API: Todo API ainaba v1	アクティブ		キーの表示/非表示

サブスクリプションのアクティベーション

Power Apps にサインイン

ハンズオン参加者は各自に割り当てられたユーザーを使用して [Power Apps](#) にサインインしてください。初めて Power Apps を利用する場合にはユーザー情報の入力等が求められます。

- ・メール欄はそのまま
- ・国/地域では 日本 を選択
- ・電話番号はダミーの値(000-0000-0000等)を入力



その後で表示されるチュートリアルは **スキップ** して構いません。

カスタムコネクタ用のメニューの表示

初期状態では左側のメニューにカスタムコネクタが表示されていないため、表示するように設定します。

- ・ 詳細 メニュー内の **すべて検出** を選択
- ・ データ セクション内にある **カスタム コネクタ** と **接続** をピン止め

The screenshot shows the Power Apps interface. On the left, there's a navigation bar with options like 'ホーム' (Home), '作成' (Create), '学習' (Learn), 'アプリ' (App), 'テーブル' (Table), 'フロー' (Flow), 'ソリューション' (Solution), '詳細' (Details), and '検出' (Discovery). The '検出' option is currently selected. Below it, there's a 'Power Platform' section. A central modal window titled '検出' (Discovery) is open, showing a list of items: 'テーブル', 'フロー', 'チャットボット', 'AI モデル', 'ソリューション', 'カード', '選択肢', '接続', and 'データフロー'. At the bottom of this list is a 'すべて検出' (Discover All) button. To the right of the modal, the main Power Apps interface shows the 'データ' (Data) section with sections for '選択肢' (Choices), '接続' (Connections), and 'カスタム コネクタ' (Custom Connector). The 'カスタム コネクタ' section has a sub-section for creating a new connection.

カスタムコネクタの作成

- メニューに追加された カスタム コネクタ を選択
- Power Automate へようこそ画面が表示されたら 開始する を選択
- + カスタム コネクタの新規作成 を選択
- OpenAPI ファイルをインポートします を選択
- コネクタ名には TodoApi_\${Prefix}_v1 と入力
- 先ほどダウンロードしておいた仕様書を インポート する
- 続行 を選択

カスタム コネクタはアドオンで作成できます。既存の REST API が 10 以上の認証方法で、その API に接続するためにカスタム

OpenAPI ファイルをインポートします

Todo API ainaba.swagger.json

Import Run Cancel

- 全般 画面は特に何も入力せず セキュリティ → を選択
- セキュリティ 画面
 - 認証タイプは API キー を選択
 - 定義 → を選択

1. 全般

コネクタ名 TodoApi_Ainaba_v1

Swagger エディター コネクタの作成 キャンセル

全般情報

アイコンと簡単な説明をカスタム コネクタに追加します。ホストとベース URL が Swagger フィルトから自動的に生成されます。

アイコンの背景色

アイコンの背後に表示する色 (例: #007ee5)

説明

カスタム コネクタに簡単な説明を追加します

スキーা*

HTTPS HTTP

ホスト*

fdsample0317-apim.azure-api.net

ベース URL

/todo-api-ainaba/v1

セキュリティ →

2. セキュリティ

コネクタ名 TodoApi_Ainaba_v1

Swagger エディター コネクタの作成 キャンセル

セキュリティ

認証の種類を選択し、必須フィールドに入力して、カスタム コネクタのセキュリティを設定します。

詳細情報

APIによって実装される認証の種類を選びます。

API キー

ユーザーは接続の作成時に API キーを入力する必要があります
パラメーターのラベル

API キー

パラメーター名*

Ocp-Apim-Subscription-Key

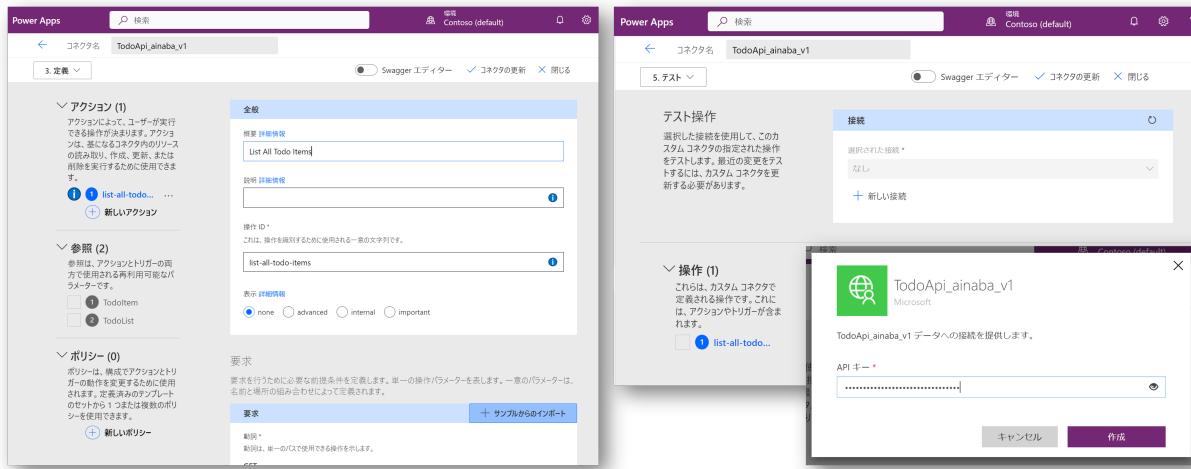
パラメーターの値*

ヘッダー

編集

← 全般 → 定義

- 定義 画面
 - アクションに `List All Todo Items` が定義されていることを確認
 - 参照に `TodoItem` と `TodoList` が定義されていることを確認
 - エラー等が表示されていないことを確認
 - `コネクタの作成` を選択
 - `カスタムコネクタが正常に作成されました` と表示されるまで待つ
- コード (プレビュー) 画面は特に何も入力せず `テスト →` を選択
- テスト 画面
 - `+ 新しい接続` を選択
 - API キーに先ほど控えておいた `主キー` を入力
 - `作成` を選択



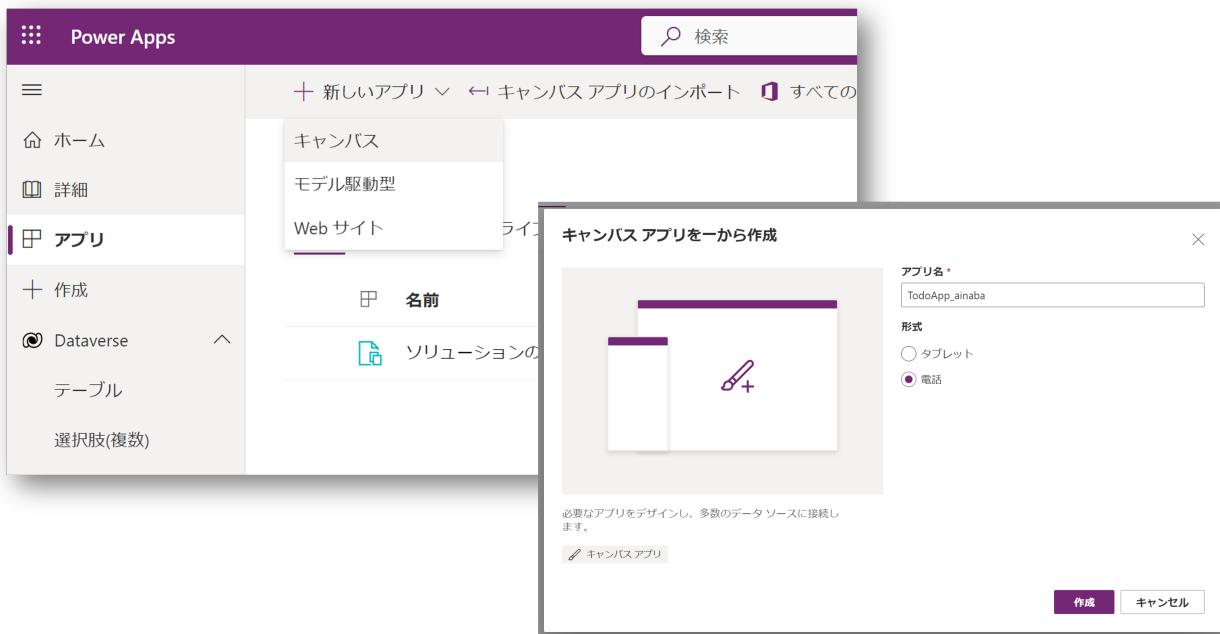
- 接続 メニューが選択されている状態になるため、`カスタム コネクタ` メニューを選択
- 作成したコネクタ `TodoApi_${prefix}_v1` を編集 (鉛筆アイコン)
- テスト 画面まで進める
- 選択された接続欄で先ほど作成した `接続` を選択
- `list-all-todo-items` の `テスト操作` を選択
- モック応答データが返却されることを確認
- 閉じる を選択

Section 06：キャンバスアプリの開発

カスタムコネクタおよび接続を利用して API を呼び出すキャンバスアプリを開発します。

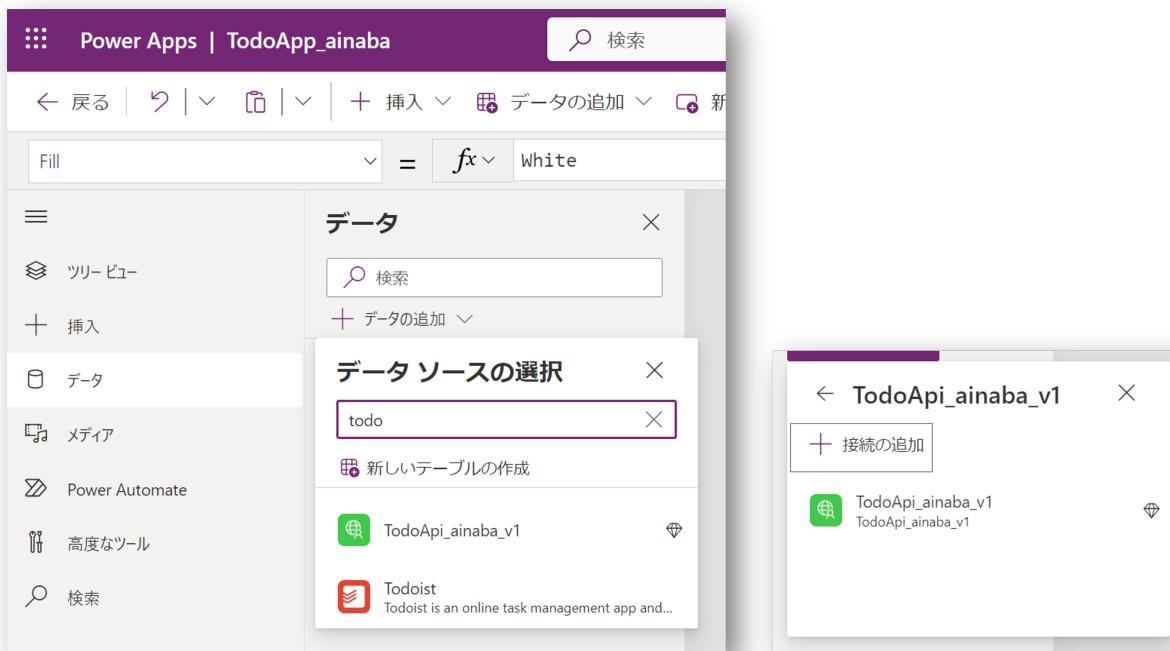
キャンバスアプリの新規作成

- アプリ メニューを選択
- + 新しいアプリ から キャンバス を選択
- アプリ名に TodoApp_\${prefix} を入力
- 作成 を選択



データの追加

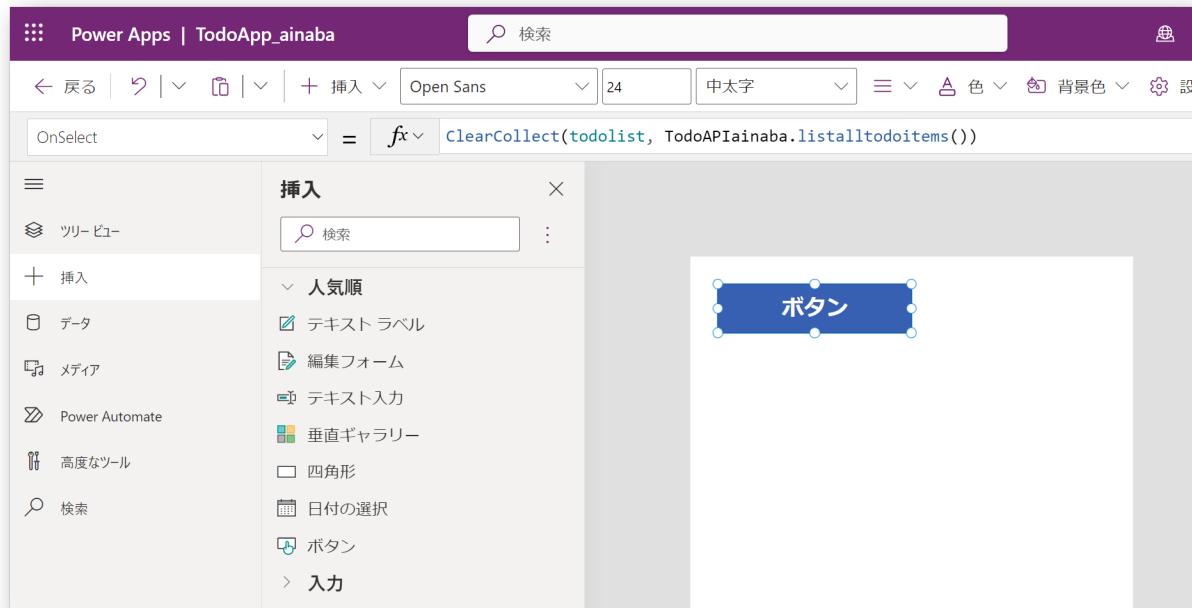
- データ メニューを開く
- + データの追加 を選択
- データソースの選択で TodoApi_\${prefix}_v1 を検索して選択
- 作成した接続を選択(既定ではコネクタと同名)



Todo アイテム一覧を取得するボタン

- 挿入 メニューから ボタン を選択
- キャンバスに追加されたボタンを選択
- OnSelect イベントに下記の式を入力

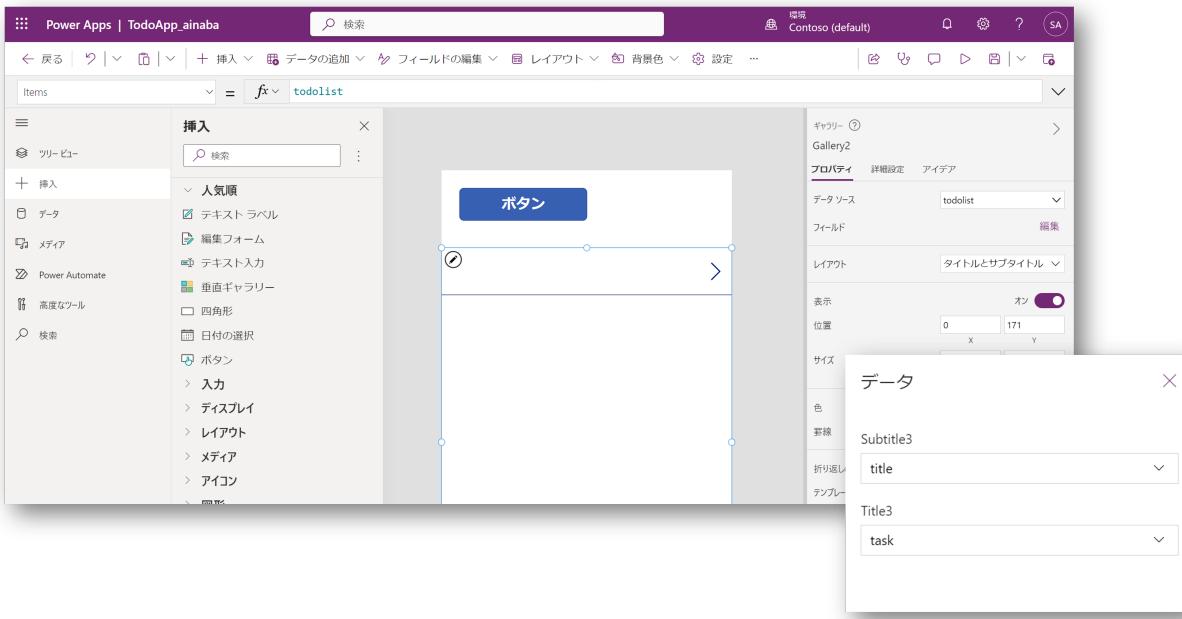
```
ClearCollect(todolist, TodoAPIainaba.listalltodoitems())
```



このボタンを押下すると `todolist` という変数に API の呼び出し結果が格納されます。

一覧表示用のギャラリー

- ・ 插入 メニューから 垂直ギャラリー を選択
- ・ キャンバスに追加された垂直ギャラリーを選択
- ・ レイアウトで タイトルとサブタイトル を選択
- ・ フィールドの 編集 を選択
- ・ 表示するデータ項目として title と task を選択



動作確認

- ・ ALT キーを押しながらボタンをクリック
- ・ ギャラリーにモック応答が表示されることを確認

ボタン

Do Task 1

Task1



Do Task 2

Task2



Do Task 3

Task3



Section 07：バックエンド API の実装

今度は API Management で定義した API の使用を満たす実際の Todo API を実装します。Azure Web Apps は参加者個別に作成します。

実装済み Todo API の取得

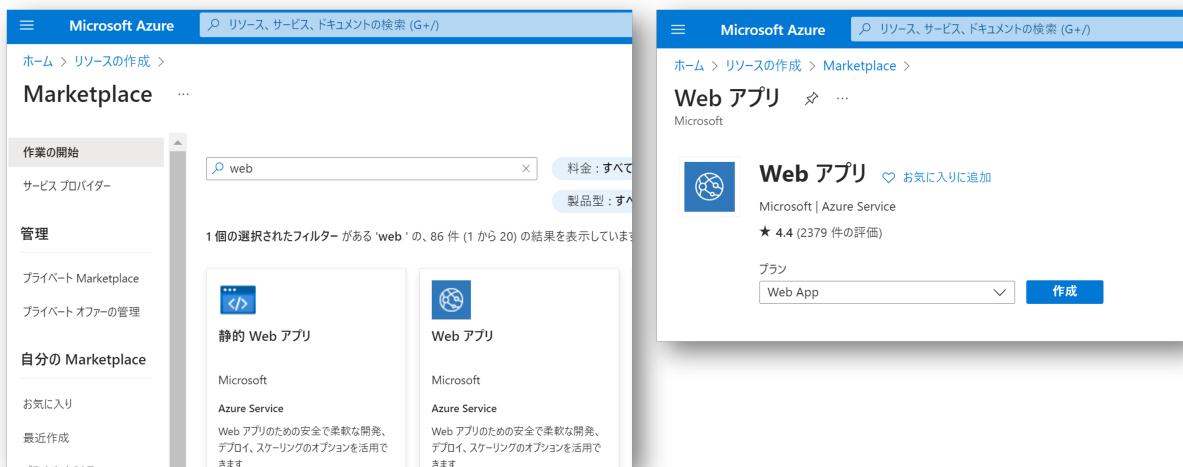
このハンズオンでは時間の関係から ASP.NET で実装済みのアプリケーションを Azure Web Apps 上にデプロイするのみとします。サンプルアプリが保存されている [GitHub Release](#) を開き、Assets の中から `default.zip` を選択してローカル PC にダウンロードしておいてください。

Azure へのサインイン

ハンズオン参加者は各自に割り当てられたユーザーを使用して [Azure ポータル](#) にサインインしてください。

Web Apps の作成

- リソースの作成 をクリック
- Web アプリ を検索し、Web アプリ を選択
- Web アプリ を作成



- リソースグループは新規作成を選択し、`rg-fusiondev-${prefix}` と入力
- 名前には `web-fusiondev-${prefix}-MMDD` と入力
- ランタイムスタックは `.NET 7 (STS)` を選択
- オペレーティングシステムは `Windows` を選択
- 地域は `Japan East` を選択
- その他の選択肢は既定のままで `確認および作成` をクリック
- 確認および作成画面でエラーがないことを確認して `作成` ボタンをクリック
- デプロイの完了を待ちます(おおよそ 1 分程度)

- `リソースに移動` を選択して Web App の画面を開く
- `概要メニュー` を選択
- `参照` を選択して初期ページが参照できることを確認

Todo API のデプロイ

- Web App の 構成 メニューを選択
- + 新しいアプリケーション設定 を選択
- 名前に ASPNETCORE_ENVIRONMENT を入力
- 値に Development を入力
- OK を選択して アプリケーション設定の追加と編集画面を閉じる
- 構成画面に戻ったら 保存 を選択

The screenshot shows the Azure portal's 'Application settings' blade. A modal window is open for editing a new application setting. The 'Name' field is set to 'ASPNETCORE_ENVIRONMENT' and the 'Value' field is set to 'Development'. The modal has a checkbox for 'Deployment slot settings' which is unchecked.

- Web App の 高度なツール メニューを選択して 移動 をクリック
- Kudu サイトが開いたら Tools メニューから Zip Push Deploy を選択
- 初期ページ(hostingstart.html) が表示されているあたりに、ダウンロードしておいた default.zip をドラッグアンドドロップする

The screenshot shows the Kudu site for the web app. The 'Tools' menu is open, with 'Zip Push Deploy' selected. The REST API blade shows the contents of the '/wwwroot' directory, which contains a single file named 'Hostingstart.html'. Below the directory listing is a large blue area with the placeholder text 'Drop here to deploy your ZIP'.

初期ページを開いているブラウザの URL 欄でパスに /swagger を追加して Enter キーを押下します。URL 全体は以下のようになります。

[https://web-fusiondev-\\$\{prefix\}-MMDD.azurewebsites.net/swagger](https://web-fusiondev-$\{prefix\}-MMDD.azurewebsites.net/swagger)

Swagger の画面が開いたら動作確認を行います。

- Todo を選択して展開
- GET /todos/v1 を選択して展開
- Try it out ボタンを選択
- Execute ボタンを選択
- レスポンスコード 200 で Todo アイテム一覧が返ってくることを確認

The screenshot shows the Swagger UI interface for a project titled "My Title 1.0.0". On the left, there's a sidebar with sections for "Schemes" (set to HTTPS), "Todo", and "Models". The main area displays the "GET /todos/v1" endpoint. It includes fields for "Parameters" (empty), "Responses" (set to application/json, showing a preview of the JSON response body), and "Curl" (a command-line example). Below these, there are tabs for "Request URL" and "Server response". The "Server response" tab is active, showing a "Code" section (set to 200) and a "Details" section where the JSON response body is displayed:

```
[{"id": 100, "title": "Task1", "task": "会議を終了する", "status": "Active", "dueDate": "2023-03-27T13:47:17.2964312+00:00", "updatedOn": "2023-03-27T13:47:17.2965703+00:00", "additionalProperties": {}}, {"id": 200, "title": "Task2", "task": "会議を終了する", "status": "Done", "dueDate": "2023-03-27T13:47:17.2971902+00:00", "updatedOn": "2023-03-27T13:47:17.2971903+00:00", "additionalProperties": {}}
```

補足

このハンズオンでは API の実装に ASP.NET を採用していますが、他の言語で実装してももちろん構いません。また API をホストするサービスが Azure Web Apps である必要もありません。クラウドサービスが Azure である必要もありませんし、クラウドサービスである必要もありません。重要なのは Section 2 で合意した仕様を満たす API がネットワーク到達性のある場所で実行されていることです。

参考情報

実装済みアプリのオリジナルのソースコードは Assets の中から [Source code](#) を選択するとダウンロード出来ます。本セクションの冒頭でダウンロードした zip ファイルには、このソースコードをビルド・発行したものが含まれています。

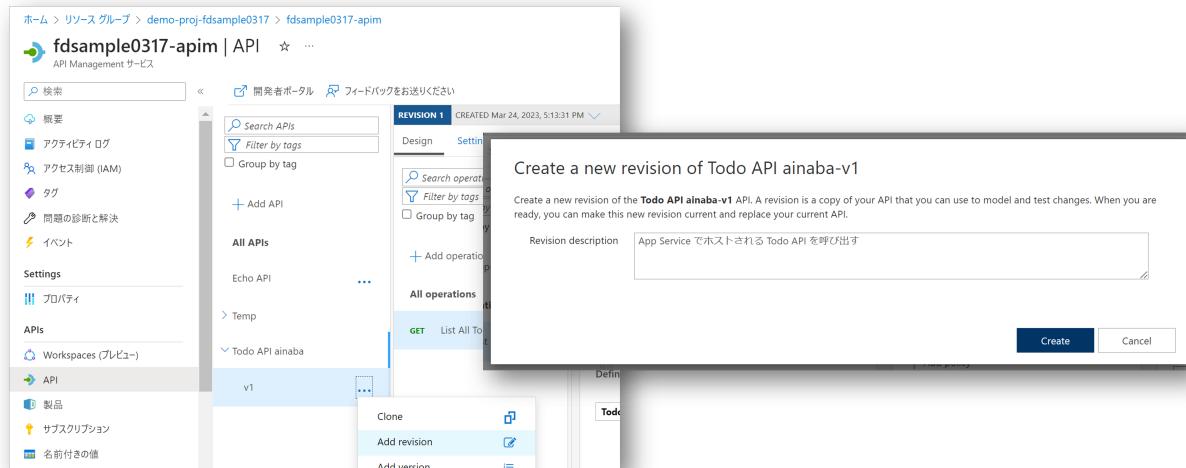
Section 08 : API リビジョンの更新

API のバックエンド側が実装できましたので、API Management のモック応答を解除し、実際の Power Apps からの呼出しがバックエンド API を呼び出すように構成を変更します。

新規リビジョンの作成

現在の設定を書き換えるのではなく、構成変更後に問題があった場合にいつでも切り戻せるように、まずは新しいリビジョンを作成します。

- [Azure Portal](#)を開く
- リソースグループ `rg-fusiondev-shared` に含まれる API Management `apim-fusiondev-shared-MMDD` を開く
- 作成した `Todo API ${prefix} API` の `v1` を選択
- メニューから `Add revision` を選択
- `Revision description` にはこのリビジョンで変更のある内容を記述
- `Create` を選択



新規リビジョンの構成変更

新しく作ったリビジョン2はこれまでの設定(リビジョン1)の内容がコピーされているので、変更部分だけを修正します。

- グレーアウトされた REVISION 2 が表示されていることを確認
- Design を選択
- List All Todo Items を選択
- Inbound processing セクションに含まれる mock-response を Delete する
- Save を選択
- Settings を選択
- Web service URL に Section 07 でテスト実行した際の Request URL をコピー
- Save を選択

URL は以下のようにになっているはずです。

<https://web-fusiondev-ainaba-0324.azurewebsites.net/todos/v1>

新規リビジョンのテスト実行

リビジョン2ではバックエンド API が呼び出されることを確認します。

- グレーアウトされた REVISION 2 が表示されていることを確認
- Test を選択
- List All Todo Items を選択
- Send を選択
- モック応答ではなくバックエンド API からの応答(HTTP 200)が表示されることを確認

Design Settings Test Revisions (2) Change log

Search operations
 Filter by tags
 Group by tag

GET List All Todo Items ...

Todo API ainaba > List All Todo Items > Console

HTTP response

Message Trace

HTTP/1.1 200 OK
content-encoding: gzip
content-type: application/json; charset=utf-8
date: Mon, 27 Mar 2023 02:07:39 GMT
request-context: appId=cid-v1:0736dbfd-441e-4a56-96a2-a5f6e37c1f7b
transfer-encoding: chunked
vary: Accept-Encoding,Origin
x-powered-by: ASP.NET

```
[{
    "id": 100,
    "title": "Task1",
    "task": "経費を清算する",
    "status": "Active",
    "duedate": "2023-03-27T13:47:17.2964312+00:00",
    "updatedOn": "2023-03-21T13:47:17.2963762+00:00",
    "additionalProperties": {}
}, {
    "id": 200,
    "title": "Task2",
    "task": "出張を申請する",
    "status": "Active",
    "duedate": "2023-03-27T13:47:17.2964312+00:00",
    "updatedOn": "2023-03-21T13:47:17.2963762+00:00"
}]
```

新規リビジョンを正式版に切り替え

- **Revisions** を選択
- Revisions の一覧に表示される Revision 2 の右端にあるメニュー (...) を選択
- **Make current** を選択
- Make revision current の画面で **Save** を選択
- 左上の REVISION 2 の背景が青くなっていることを確認
- Revisions の一覧に表示される Revision 2 の CURRENT にも チェックが入っていることを確認

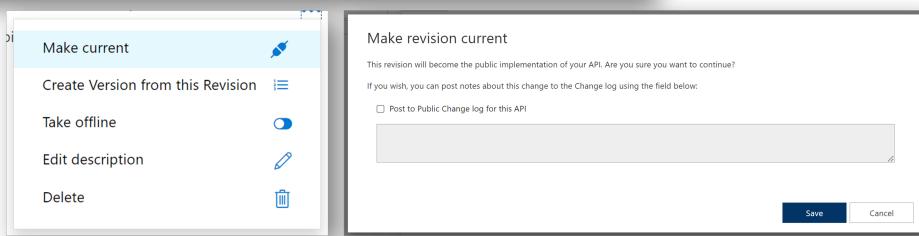
REVISION 2 CREATED Mar 27, 2023, 10:47:03 AM

Design Settings Test Revisions (2) Change log

Revisions

ID	CREATED	DESCRIPTION	URL	ONLINE	CURRENT
2	Mar 27, 2023, 10:47:03 AM	App Service でホストされる Todo API を呼び出す	/todo-api-ainaba/v1;re...	✓	...
1	Mar 24, 2023, 5:13:31 PM		/todo-api-ainaba/v1	✓	✓

+ Add revision



REVISION 2 CREATED Mar 27, 2023, 10:47:03 AM

Design Settings Test Revisions (2) Change log

Revisions

ID	CREATED	DESCRIPTION	URL	ONLINE	CURRENT
2	Mar 27, 2023, 10:47:03 AM	App Service でホストされる Todo API を呼び出す	/todo-api-ainaba/v1;re...	✓	✓
1	Mar 24, 2023, 5:13:31 PM		/todo-api-ainaba/v1	✓	...

+ Add revision

Section 09：正式版 API によるテスト

作成していたキャンバスアプリに戻り、ボタンをクリックするとバックエンド API からの応答が表示されることを確認します。手順は [Section 06](#) の動作確認と同様です。



Section 10：キャンバスアプリの共有

ここでは作成したキャンバスアプリを別のユーザーに利用してもらいます。近くの席のハンズオン参加者のユーザー名を教えてもらってください。

キャンバスアプリの保存と公開

まず作成中のアプリを利用可能な状態にします。

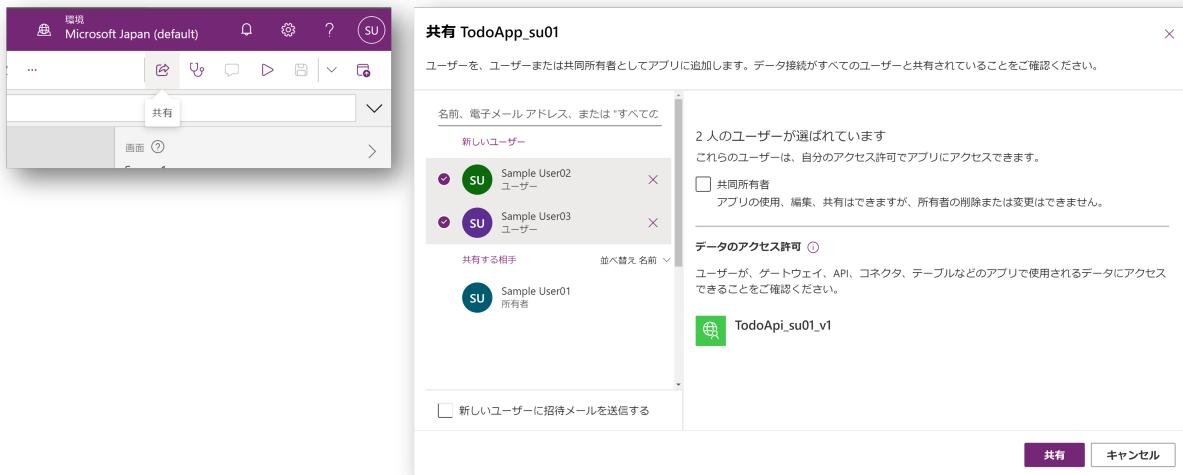
- 画面右上の **公開** を選択
- **このバージョンの公開** を選択



キャンバスアプリの共有

自分以外のユーザーが利用できるようにアプリを共有します。

- 画面右上の **共有** を選択
- 共有するユーザー（近くの席の参加者）名前を入力・選択
- 選択したユーザーの **共同所有者** のチェックが入っていないことを確認
- データのアクセス許可にカスタムコネクタ名 `TodoApi_{prefix}_v1` が表示されていることを確認
- **新しいユーザーに招待メールを送信する** のチェックを外す
- **共有** ボタンを選択



接続の共有

キャンバスアプリとともにカスタムコネクタを共有しましたが、利用者は API キーを知らないと利用できません。API キーは **接続** の作成時に入力していますので、この **接続** も共有します。

- 画面左側のメニュー内の **接続** を選択
- 作成済みの接続 **TodoApi_su01_v1** のメニューから **共有** を選択
- アプリを共有したユーザー（近くの席の参加者）の名前を入力・選択
- 権限として **使用可能** が選択されていることを確認
- 保存** ボタンを選択

名前	メール	権限
Sample User03	su03@fdenv0317a.onmicrosoft.com	使用可能
Sample User02	su02@fdenv0317a.onmicrosoft.com	使用可能
Sample User01	su01@fdenv0317a.onmicrosoft.com	所有者

共有されたアプリの利用

ここでの手順は自身が開発したアプリではなく、他の参加者が開発し共有したアプリで行います。

- 画面左側のメニュー内の **アプリ** を選択
- 共有されたアプリの名前部分をクリック
- アプリが起動する前に接続の利用を **許可** する
- アプリが表示されたら動作確認を行う

補足

上記の手順で [接続の共有](#) を行わなかった場合には、共有されたユーザーは API キーを入力する必要があります。このような場合には「アプリの利用者はどのようにしてキー入手するのか？」が問題になりますが、ここではアプリとともに API キーが埋め込まれた接続も共有することで回避しています。このため共有されたアプリを利用するだけであれば API キーを知っている必要はありません。

しかし、そもそも [Section 05](#) で各々が開発した API 固有のキーを発行して接続を作成、他のユーザーに共有しています。これが可能なのは、このハンズオンでは「キャンバスアプリの開発者が API の開発者も兼ねている」ため、Azure ポータルから API キーを作成して利用することができる権限を持っているからです。これは [フュージョン開発](#) という状況では現実的ではない可能性があります。つまり「アプリの開発者はどのようにしてキー入手するのか？」が問題になります。この問題の解決策の一つとしては後述の API Management 開発者ポータルを利用する方法が考えられます。

あるいは（このハンズオンでは実施しませんが）API キーではなくユーザー認証を利用して API を保護する方法も考えられます。この場合は Power Apps キャンバスアプリと同様に Azure AD ユーザー認証と OAuth 2.0 をベースとしたアクセス制御を実施することが可能です。

Section II : アプリと API の利用状況の監視

開発したアプリや API の利用状況を監視することで、ユーザーのニーズやアプリの改善点を把握することができます。また利用の芳しくないアプリや API を特定することで、サービスの廃止などの対応を行うことができます。本セクションでは、キャンバスアプリと API の利用状況を監視する方法を紹介します。

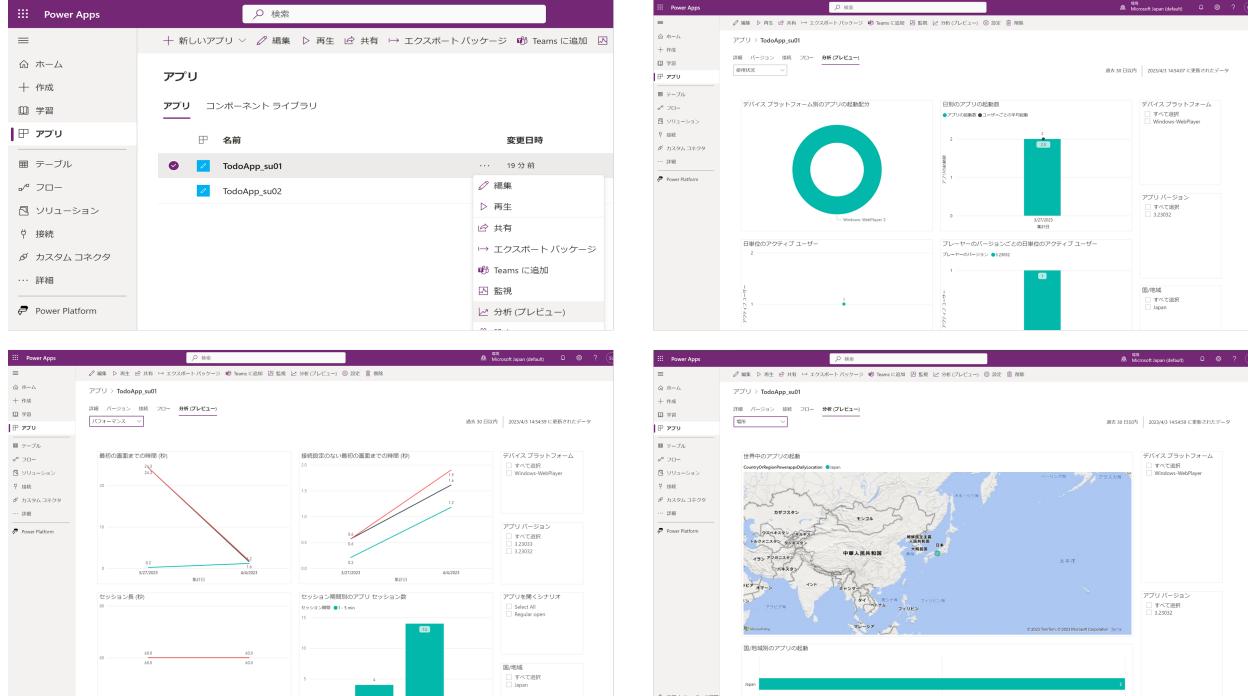
キャンバスアプリの利用状況の監視

! 以下の内容は試験中やプレビュー段階の機能を含むため、今後変更になる可能性があります。

開発・共有したキャンバスアプリの利用状況を確認します。

- Power Apps (<https://make.powerapps.com>) を開く
- 左側の **アプリ** メニューを選択
- 開発したアプリ `TodoApp_{prefix}` のメニューから分析(プレビュー)を選択
- ドロップダウンで **使用状況** を選択
- ドロップダウンで **パフォーマンス** を選択
- ドロップダウンで **場所** を選択

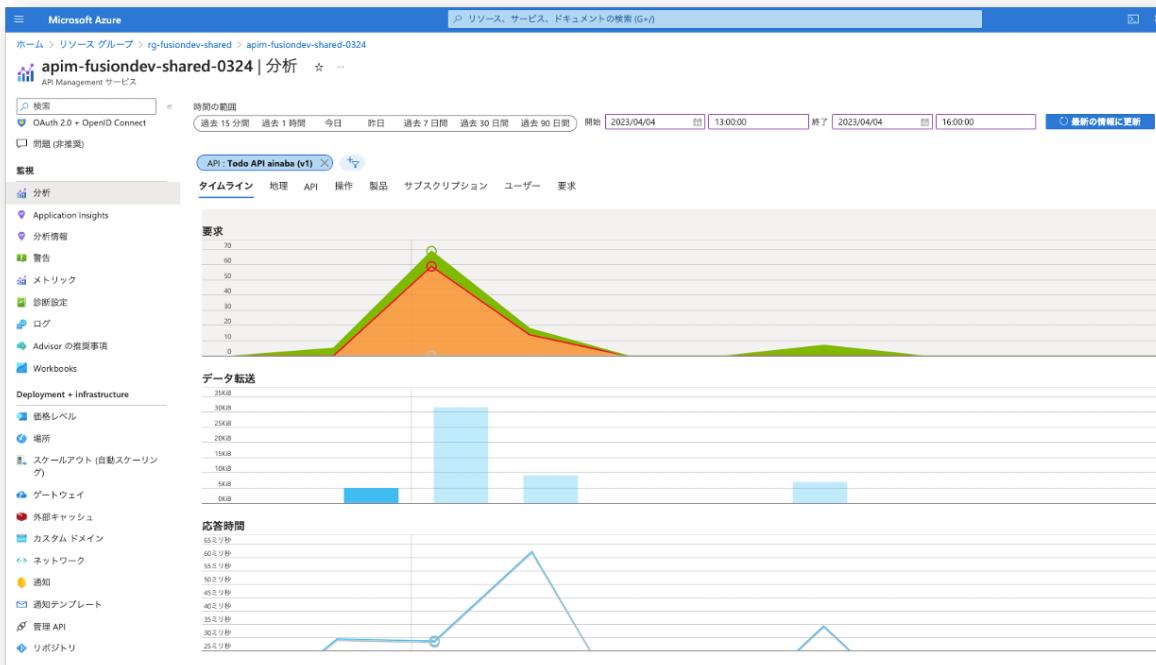
データの集計に時間がかかるためハンズオン作業中にはデータが表示されない可能性があります。
後日お時間のあるときに改めて確認してみてください。



API の利用状況の確認

開発した API の利用状況を確認します。

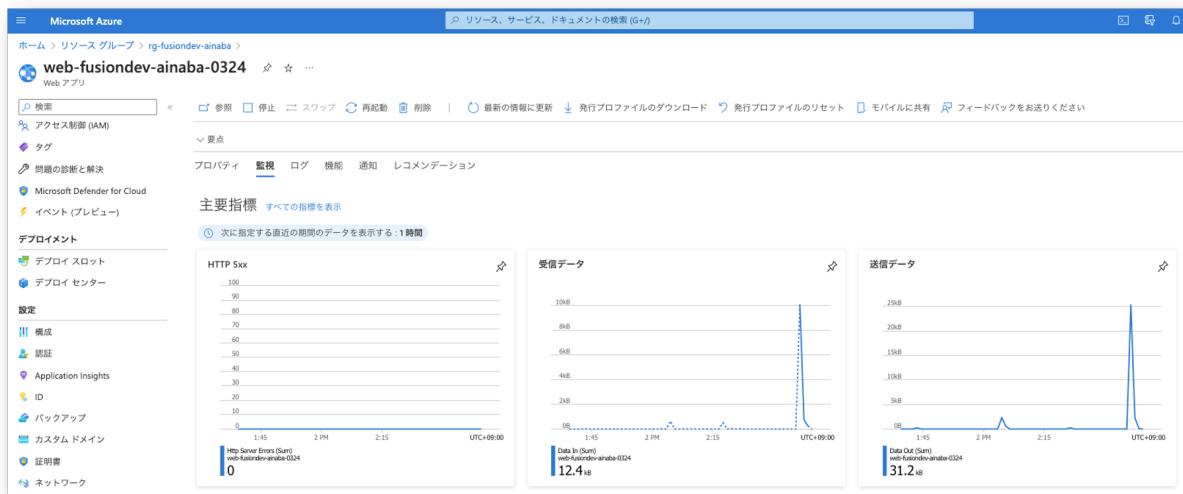
- Azure Portal (<https://portal.azure.com>) を開く
- リソースグループ rg-fusiondev-shared を開く
- API Management apim-fusiondev-shared-MMdd を開く
- 左側のメニューから **分析** を選択
- 開始日時にハンズオンの開始日時を入力
- 終了日時としてハンズオンの終了予定日時を入力
- **最新の情報に更新** をクリック
- フィルターで **API** と **Todo Api \${prefix} (v1)** を選択
- **OK** を選択
- **タイムライン**、**地理**、**API** 等のタブを選択して呼び出し状況を確認



バックエンド API の利用状況の監視

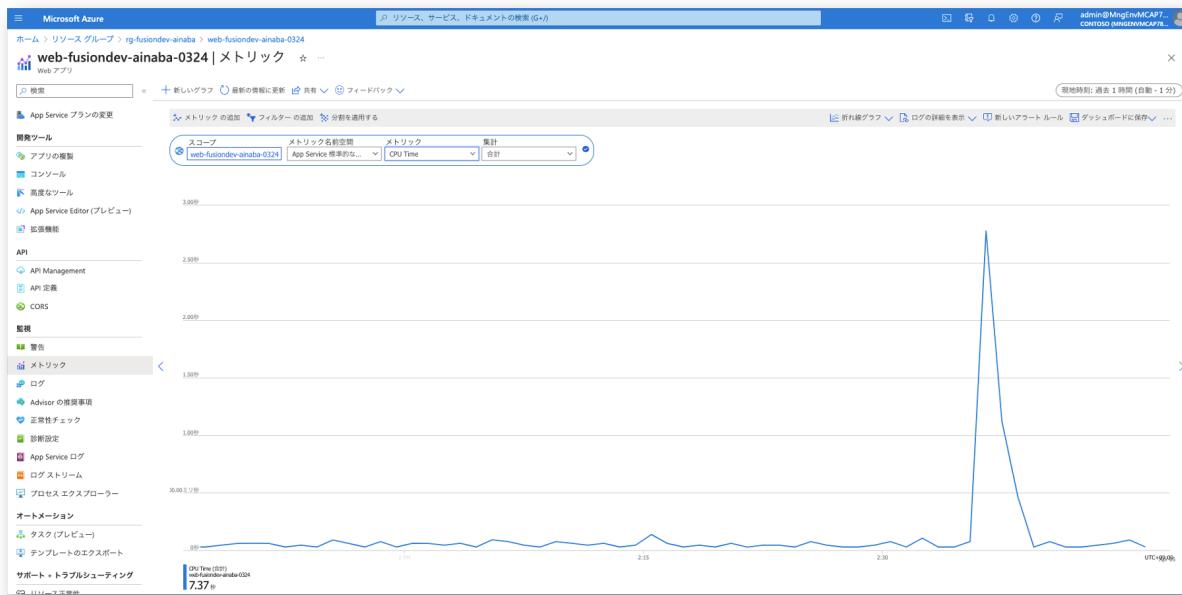
バックエンドで動作する API 実装をホストする App Services の利用状況を確認します。

- Azure Portal (<https://portal.azure.com>) を開く
- リソースグループ `rg-fusiondev-${prefix}` を開く
- App Service `web-fusiondev-${prefix}-MMdd` を開く
- 左側のメニューで **概要** を選択
- **監視** タブを選択
- HTTP 5xx のエラーや、送受信データ量、要求回数、応答時間などのグラフを確認



より詳細なメトリックデータを確認してみましょう。

- 左側のメニューで **メトリック** を選択
- スコープに **web-fusiondev-\${prefix}-MMdd** を選択
- メトリック名前空間で **App Service 標準的なメトリック** を選択
- メトリック名で **CPU Time** や **Average memory working set** を選択



Section 12：開発者ポータルの有効化

作成した API を多くの開発者に利用してもらうために、開発者ポータルを有効化します。

!

本セクションの設定内容は代表者が 1 回だけ行ってください。

開発者ポータルの有効化

既定の設定では開発者ポータルが無効になっていますので有効化する作業を行います。

- Azure Portal を開く
- リソースグループ **rg-fusiondev-shared** を選択
- API Management **apim-fusiondev-shared-MMDD** を選択
- 左のメニューから **開発者ポータルの概要** を選択
- 上部のメニューから **開発者ポータル** を選択
- 別のタブで開発者ポータルの編集画面が表示される

The screenshot shows two side-by-side views. On the left is the Microsoft Azure portal interface, specifically the 'Developer Portal' configuration page for the 'apim-fusiondev-shared-0324' resource group. It displays various settings like CORS and public access status. On the right is a preview of the generated developer portal for 'contoso', which features a dark blue header with 'Welcome to Contoso!', a 'Sign up' button, and some statistics about API availability and usage.

開発者ポータルの公開

- Azure Portal に戻る
- **公開** ボタンが選択可能になっているのでクリック
- 「開発者ポータルの公開」が表示されたら **はい** を選択

Microsoft Azure

ホーム > apim-fusiondev-shared-0324

apim-fusiondev-shared-0324 | ポータルの概要

開発者ポータル

概要 リビジョン

開発者ポータルは自動的に生成される、完全にカスタマイズ可能な Web サイト

従来の開発者ポータルは非推奨になったため、セキュリティ更新プログラム

ポータルの公開

ポータルを公開すると、変更とカスタマイズを閲覧者が利用できるようになります (保存されたコンテンツの変更、構成の変更、最新のソフトウェアのアップグレードなど)。

開発者ポータルがまだ公開されていません。

はい いいえ

公開

CORS を有効にする

クロスオリジンリソース共有は、Web ページにあるリソースを別のドメインに対してこれを有効にする必要があります。カスタム ドメインを追加または削除する場合は、このオプションを選択してください。

https://apim-fusiondev-shared-0324.developer.azure-api.net オリジンに対して CORS を有効にする

CORS を有効にする

手動でグローバル レベルに適用

CORS の有効化

開発者ポータルから API を呼び出すためには、CORS の設定が必要です。

- CORS を有効にする ボタンが選択可能になっているのでクリック
- 「API に対して CORS を有効にする」が表示されたら はい を選択

Microsoft Azure

ホーム > apim-fusiondev-shared-0324

apim-fusiondev-shared-0324 | ポータルの概要

開発者ポータル

概要 リビジョン

開発者ポータルは自動的に生成される、完全にカスタマイズ可能な Web サイト

従来の開発者ポータルは非推奨になったため、セキュリティ更新プログラム

ポータルの公開

ポータルを公開すると、変更とカスタマイズを閲覧者が利用できるようになります (保存されたコンテンツの変更、構成の変更、最新のソフトウェアのアップグレードなど)。

直前の発行

リビジョン 20230405153255 は 2023/4/5 15:32:55 に作成され、2023/4/5 15:32:55 に更新されました。

はい いいえ

公開

CORS を有効にする

クロスオリジンリソース共有は、Web ページに対してこれを有効にする必要があります。カスタム ドメインを追加または削除する場合は、このオプションを選択してください。

https://apim-fusiondev-shared-0324.developer.azure-api.net に対して CORS を有効にする

CORS を有効にする

手動でグローバル レベルに適用

開発者ポータル以外のクライアントの許可

上記で設定された CORS ポリシーでは、開発者ポータルからのみ API を呼び出すことができます。Power Apps キャンバスアプリから呼び出せなくなってしまっているので、その他のクライアントも許可します。

- 左のメニューから API を選択
 - 太字になっている All APIs を選択
 - Allowed origins に入力されている値を削除して アスタリスク * を入力
 - Allow credentials を No に切り替え
 - Save を選択

The screenshot shows the Azure API Management service interface for the 'apim-fusionone dev shared-0324' workspace. The left sidebar includes sections for Overview, Audit Log, IAM, Tags, Issues, Events, Settings, Workspaces, API, and Products. The main area displays an API named 'Frontend' under the 'All APIs' section. A search bar at the top allows filtering by tags. The 'Inbound processing' tab is selected, showing a description: 'Modify the request before it is sent to the backend service.' Below this is a 'Policies' section with a 'cors' policy listed. The 'Outbound processing' tab is also visible. On the right, a detailed view of the 'cors' policy settings is shown, titled 'Inbound processing'. It includes fields for Allowed origins, Allowed methods (GET, POST, PUT, DELETE), Allowed headers, Exposed headers, and Allow credentials (set to No). A note states: 'Modify the request before it is sent to the backend service.' At the bottom are 'Save' and 'Discard' buttons.

Section 13：開発者ポータルの利用

有効化した開発者ポータルを利用して、API の仕様や API キーが取得できることを確認します。

開発者ポータルの URL の取得

まずは、開発者ポータルの URL を確認します。

- Azure Portal を開く
- リソースグループ **rg-fusiondev-shared** を選択
- API Management **apim-fusiondev-shared-MMDD** を選択
- 左のメニューから **概要** を選択
- 開発者ポータルの URL を **クリップボードにコピー** する

ここでリンクをクリックして開いてしまうと、開発者ポータルの編集画面になってしまふのでご注意ください。



Microsoft Azure リソース、サービス、ドキュメントの検索 (G+) ホーム > apim-fusiondev-shared-0324 API Management サービス 検索 開発者ポータル 削除 モバイルで開く クリップボードにコピー

概要

リソース グループ (移動) : rg-fusiondev-shared
状態 : オンライン
場所 : 東日本
サブスクリプション (移動) : ME-MngEnvMCAP784488-ainaba-1
サブスクリプション ID : 3dedf418-2574-41e1-87f9-f7ea51f53648
タグ (編集) : タグを追加するにはここをクリック

開発者ポータルの URL : <https://apim-fusiondev-shared-0324.developer.azure.net>
ゲートウェイの URL : <https://apim-fusiondev-shared-0324.azure-api.net>
レベル : 開発者 (SLA なし)
仮想 IP (VIP) アドレス : バブリック: 20.44.188.82
Platform Version : stv2

開発者ポータルの表示

- 認証情報キャッシュを持たないブラウザを表示する
 - Edge の場合は **新しい InPrivate ウィンドウ** を利用
 - Chrome の場合は **新しいシークレット ウィンドウ** を利用
- コピーしておいた開発者ポータルの URL を貼り付けて開く
- 上部の **APIs** メニューを選択
- 自分や他の人が定義した API を選択して、API 仕様を確認
- API definition ドロップダウンから仕様書をダウンロードできることを確認

サインアップと API キーの取得

以下の作業は受信可能なメールアドレスが必要です。

- 上部の **Product** メニューを選択
- Starter** を選択
- 利用したい API が含まれていることを確認
- Subscribe** を選択

- Sign in 画面が表示されたら **Sign up** リンクを選択
- Sign up 画面で必要な情報を入力して **Sign up** ボタンを選択
- 入力したメールアドレスにメールが届くので、メール内のリンクをクリックしてアカウントを有効化
- Sign in 画面を表示してメールアドレスとパスワードを入力

Sign in

Not a member yet? [Sign up](#).

Email *

Password *

[Forgot your password?](#)

Powered by [Azure API Management](#).

Sign up

Already a member? [Sign in](#).

Email *

Password *

Confirm password *

First name *

Last name *

Enter the characters you see.

New | Audio

Enter the captcha here

[Sign up](#)

- 上部の **Product** メニューを選択
- Starter** を選択
- サブスクリプション名に `${prefix} subscription` と入力
- Subscribe** を選択
- User profile 画面で Primary key の **Show** を選択
- API キーが表示されることを確認

Subscription Plan: Starter

Account Details:

Email	ayumu.inaba+0405@live.com
First name	Ayumu
Last name	Inaba
Registration date	04/05/2023

Subscriptions:

Name	Description	Product	Status	Action
ainaba subscription	Primary key: feba7f14914b6419195466c76 Secondary key: XXXXXXXXXXXXXXXXXXXXXXXXX	Starter	Active	Rename Hide Regenerate

補足

開発者ポータルを用意しておくと、開発者が API を利用するために必要な情報を自分自身で取得できることができます。つまり業務上適切なタイミングで必要な API を探索、API を介して必要な機能やデータにアクセスできることになります。

留意事項

本ハンズオンでは割愛していますが、開発者ポータルは初期設定では任意のユーザーがサインアップすることが可能です。組織内でのみ利用を許可する API であれば、Azure AD 認証を有効化し、ユーザー名とパスワードによるサインアップを無効化すると良いでしょう。

Section 20 : API のメジャー バージョンアップ

本セクションは応用編です。詳細な手順は記載しておりませんが、時間のある方はチャレンジしてみてください。

ここまで Todo 一覧を取得するだけの API を公開、キャンバスアプリから利用してきました。ここでは、Todo 一覧を取得するだけではなく、Todo の追加、更新、削除も行えるように API を拡張します。

API 仕様の変更

[Section 01](#)で設計した API 仕様は例えば [こちら](#) のようになります。仕様が大きく変わりますので、メジャーバージョンを上げて新しいバージョンの API を公開すると良いでしょう。

Todo API v2 の定義

作成した API v1 のメニューから **Add version** を選択することで、新しいバージョンの API を作成できます。Version Identifier を v2 として作成すると、v1 と v2 の両方の API が動作する状態になります。作成直後は v2 は v1 と同じ仕様、同じ設定になっています。

API 仕様のインポート

v2 の API を一から作成しても良いのですが、すでに定義済みの仕様書がありますので、**Import** を選択して仕様書をインポートすることができます。

バックエンド API の実装

[Section 07](#) で App Service にデプロイした API は、一覧取得だけでなく、作成や更新といった各種操作が実装済みです。このため Web service URL に v1 と同じ URL を入力するとそのまま動作させることができます。

`https://web-fusiondev-$\{Prefix\}-MMdd.azurewebsites.net/todos/v1`

API Management で公開している API が v2 なのに、バックエンド API が v1 なのは奇妙な感じもしますが、市民開発者向けに公開する API のライフサイクルと、バックエンドシステムとして管理している API のライフサイクルは異なることがありますので、こういったケースもありえるのではないかでしょうか。

カスタムコネクタの作成

API がバージョンアップして v1 と v2 が並行稼働している状態なので、対応するカスタムコネクタも v1 とは別に v2 用のものを新規に作成します。API v1 用のカスタムコネクタを編集して仕様変更してしまうと、既にそれを利用しているキャンバスアプリに影響が出てしまう可能性があります。

キャンバスアプリの機能追加

キャンバスアプリの機能を追加して、Todo の追加、更新、削除ができるようにしてみましょう。

例えば更新操作を行うための関数は下記のような実装になります。

```
TodoAPI.updateTodoById(  
    IdInput.Text,  
    {  
        id_1:IdInput.Text,  
        title:TitleInput.Text,  
        task:TaskInput.Text,  
        status: If(StatusCheckbox.Value, "Done", "Active"),  
        dueDate: DueDatepicker.SelectedDate, updatedOn: Now()  
    }  
);
```