



AUTODESK Platform Services **APS Online Training – Simple Viewer**

伊勢崎 俊明

Developer **A**dvocacy & **S**upport、Autodesk Developer Network

アジェンダ – ご紹介する内容

- はじめに – Simple Viewer
- デベロッパキーの取得
- セットアップ
- 認証
- データと変換
- Viewer と UI



はじめに

インターネットで各社提供 API への接続が可能な時代

- 業界標準 & オープンソース API テクノロジ

<HTML>
HTML



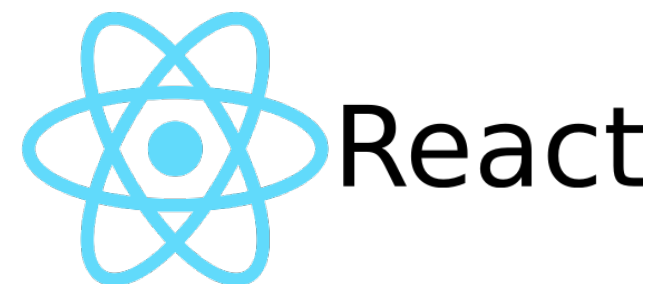
{CSS}
CSS



(JS)
JavaScript



RESTful API
GET PUT POST DELETE

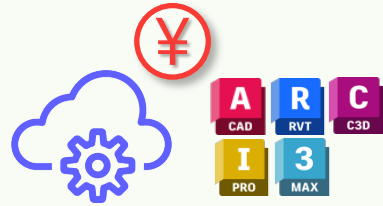


Autodesk Platform Services – 主要 API

Core APIs



Model
Derivative



Design
Automation



Viewer



Authentication



Webhooks

Data APIs



AEC
Data Model



Manufacturing
Data Model



Data
Management



Data
Exchange

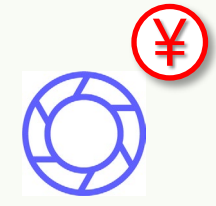
Product APIs



Forma



Fusion



Reality
Capture



ACC

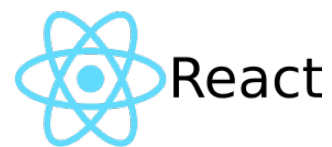


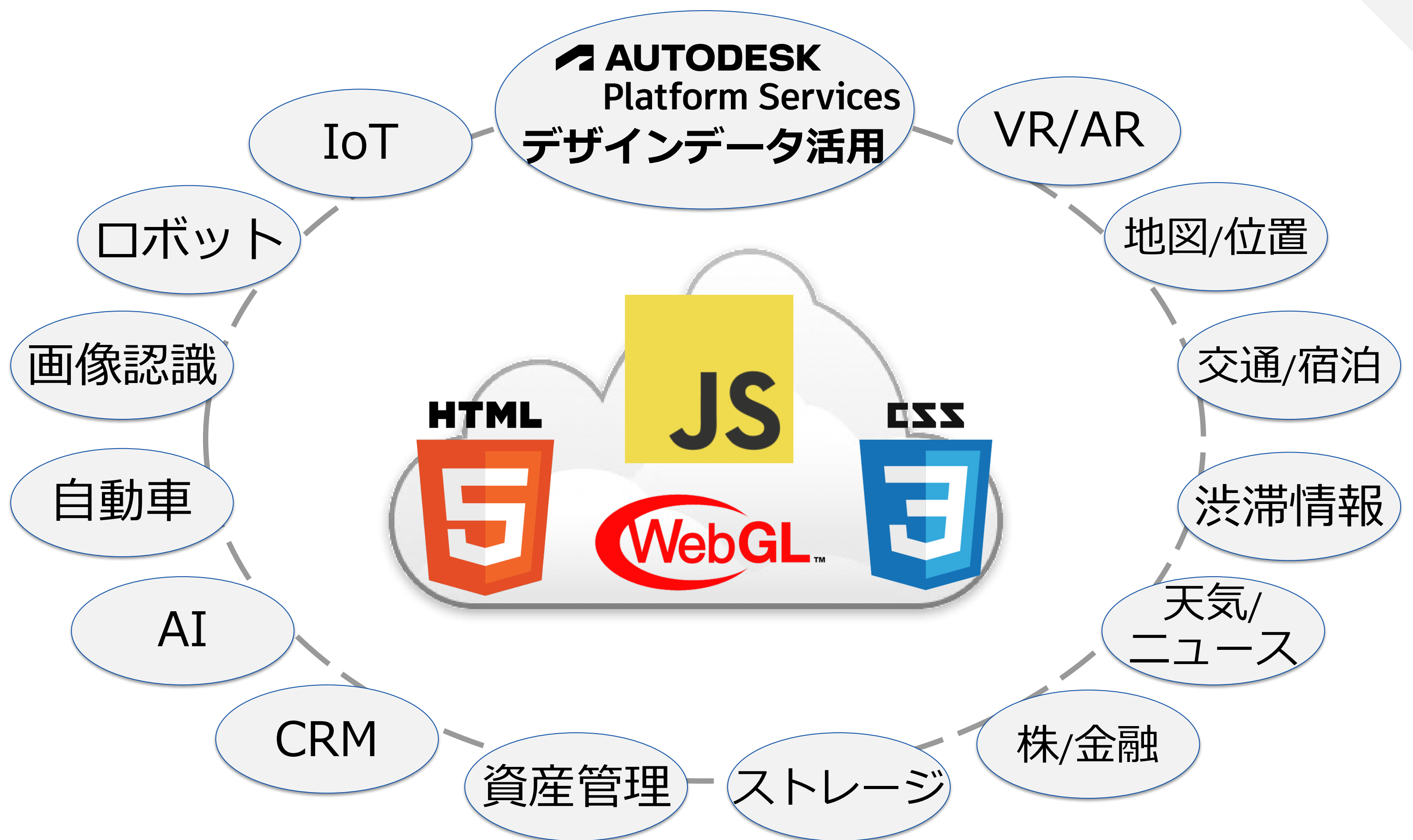
BIM 360



Tandem

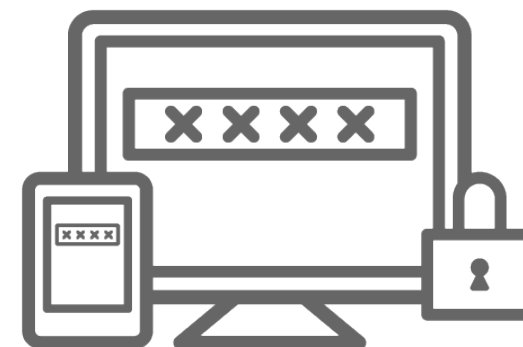
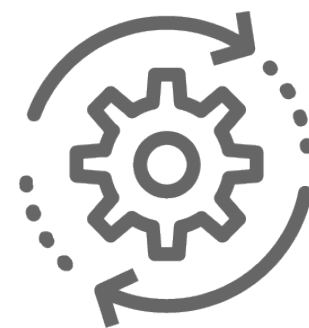
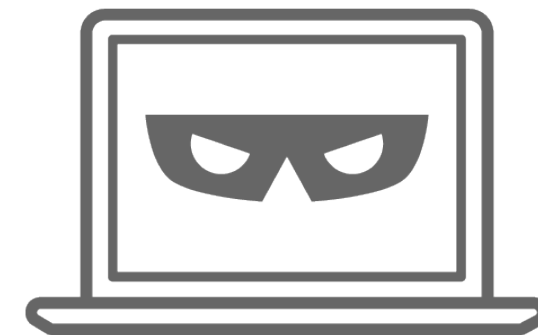
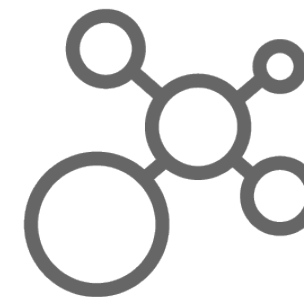
RESTful API
GET PUT POST DELETE





デスクトップ製品のアドイン開発者の方へ

- デスクトップ開発と Web 開発の違いをご理解ください
 - Web アプリの特性 – セキュリティ視点
 - 通信経路
 - 呼び出し数制限
 - 非同期処理
 - 仮想環境
 - オンプレミスとの違い

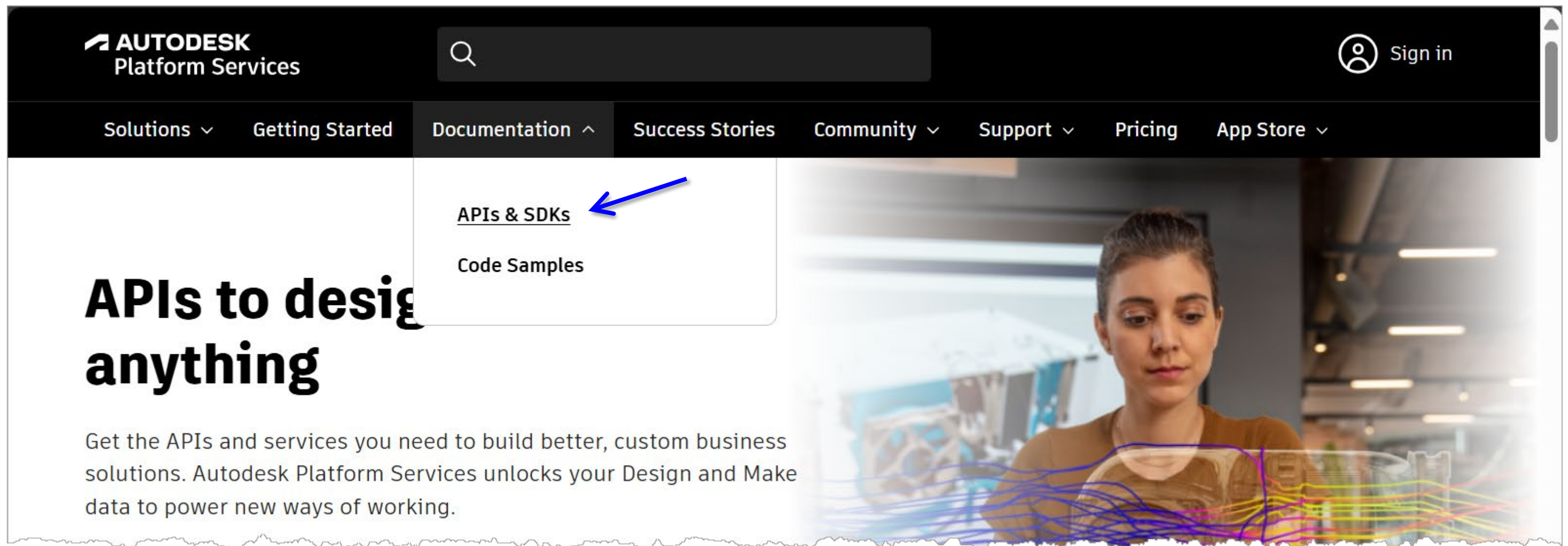


関連ブログ記事

公式ドキュメント（英語のみ）

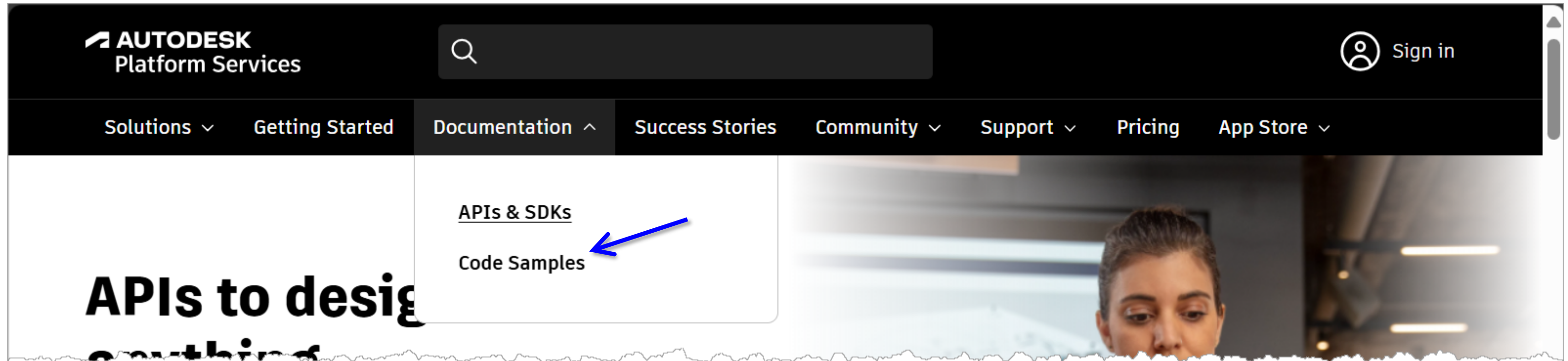
<https://aps.autodesk.com/developer/documentation>

- APS ポータル (<https://aps.autodesk.com/>) に記載



APS サンプル – GitHub リポジトリ

<https://aps.autodesk.com/code-samples>



AUTODESK	Solutions	Documentation	Resources	About
Company overview	AEC Data Model	AEC Data Model	Get Help	About APS
Careers	Autodesk Construction Cloud	Authentication	API Status	Pricing
Investor relations	Autodesk Fusion	Autodesk Construction Cloud	Blog	Success Stories
Newsroom	BIM 360	BIM 360	FAQ	Certified Partners
	BuildingConnected API	Data Exchange	Code Samples	Partnerships
			ADN Member Sign-in	

学習リソース : Learn APS Tutorial (英語のみ)

<https://tutorials.autodesk.io/>

Get the APIs and services you need to build better, custom business solutions. Autodesk Platform Services unlocks your Design and Make data to power new ways of working.

Sign up to try Autodesk Platform Services



APS Tuesdays

Join a free weekly webinar to get to know APS



Learn APS Tutorial

Step by step guide on GitHub

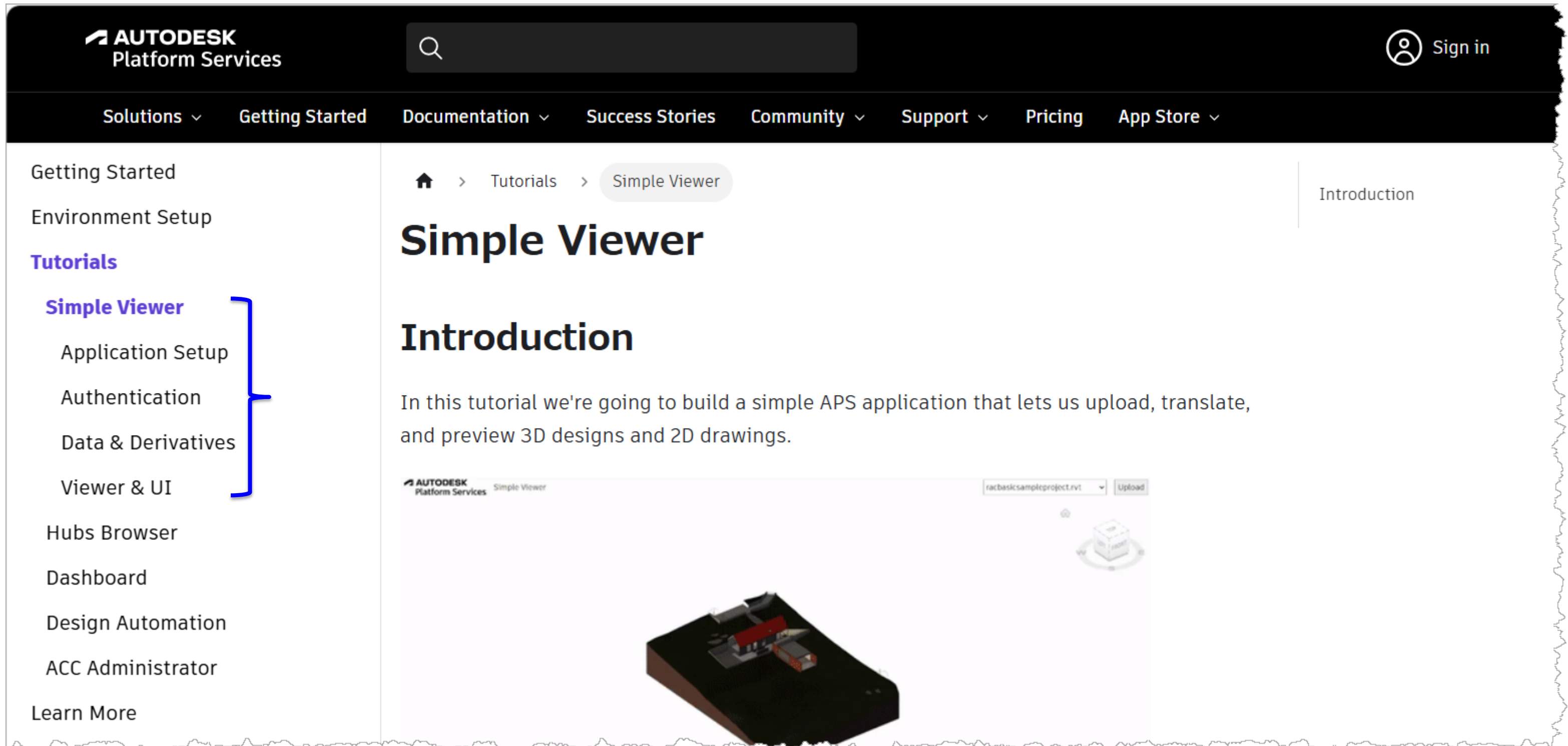


Developer Newsletter

Stay up to date with the developer community

学習リソース : Learn APS Tutorial

<https://get-started.aps.autodesk.com/>



AUTODESK
Platform Services

Search

Sign in

Solutions ▾ Getting Started Documentation ▾ Success Stories Community ▾ Support ▾ Pricing App Store ▾

Getting Started
Environment Setup
Tutorials
 Simple Viewer
 Application Setup
 Authentication
 Data & Derivatives
 Viewer & UI
Hubs Browser
Dashboard
Design Automation
ACC Administrator
Learn More

Home > Tutorials > Simple Viewer

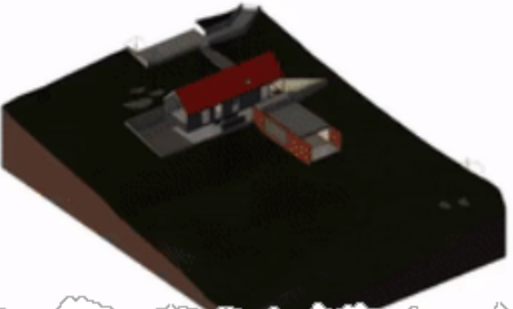
Simple Viewer

Introduction

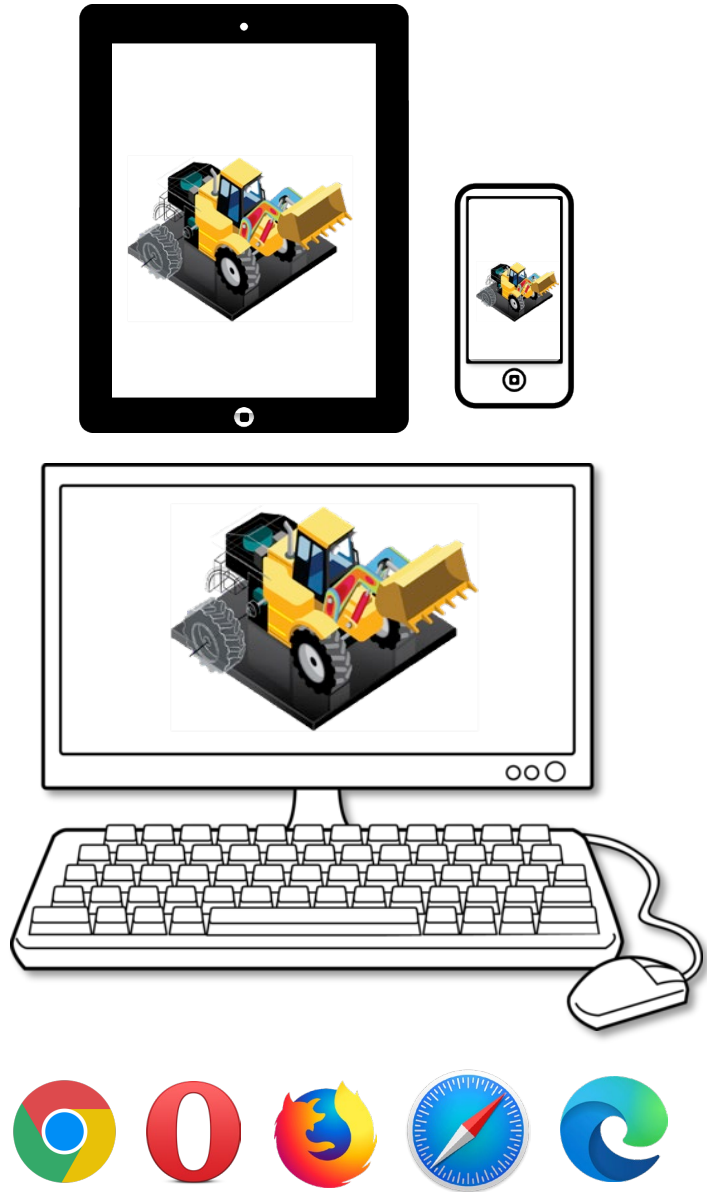
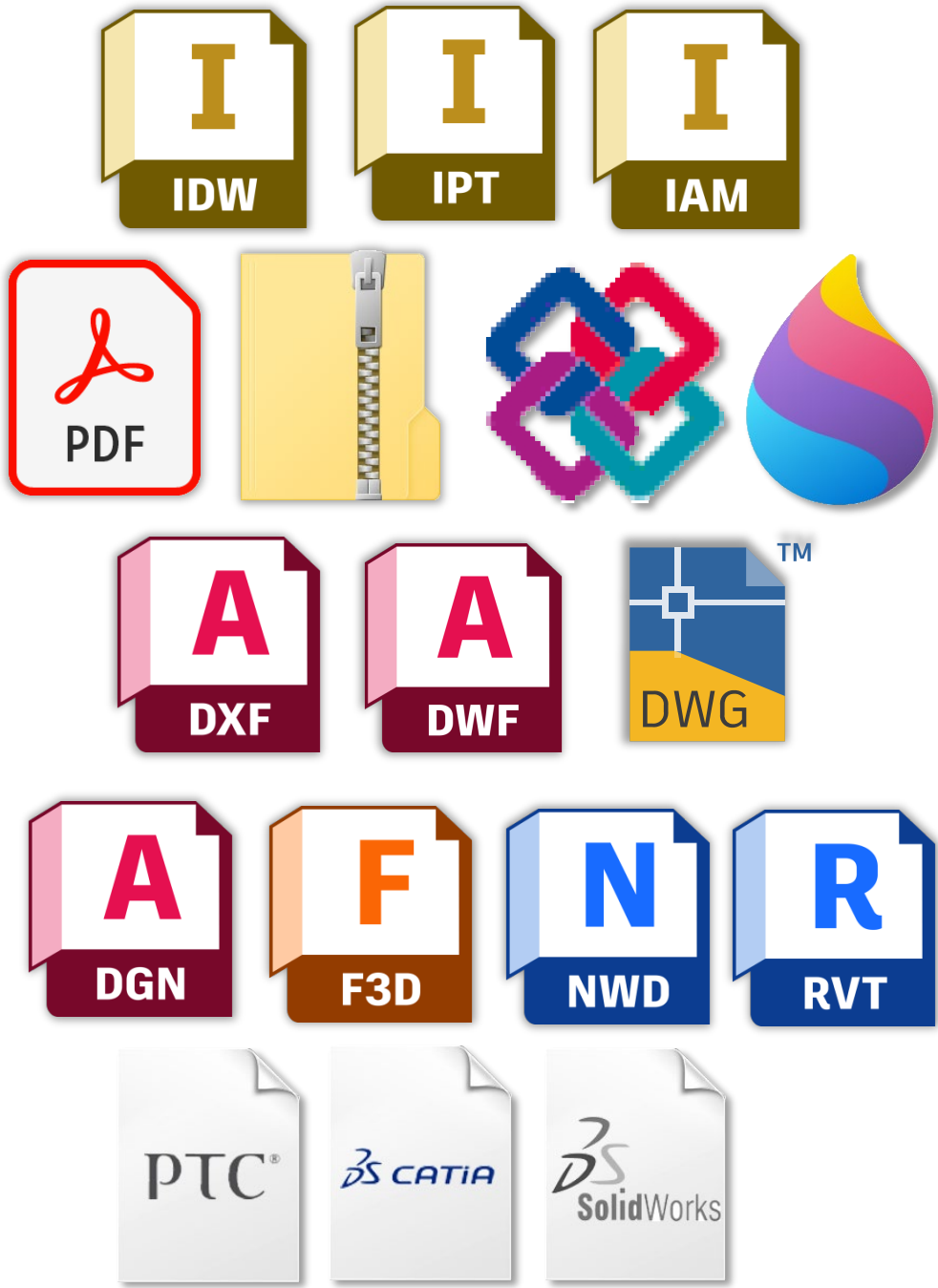
In this tutorial we're going to build a simple APS application that lets us upload, translate, and preview 3D designs and 2D drawings.

AUTODESK Platform Services Simple Viewer

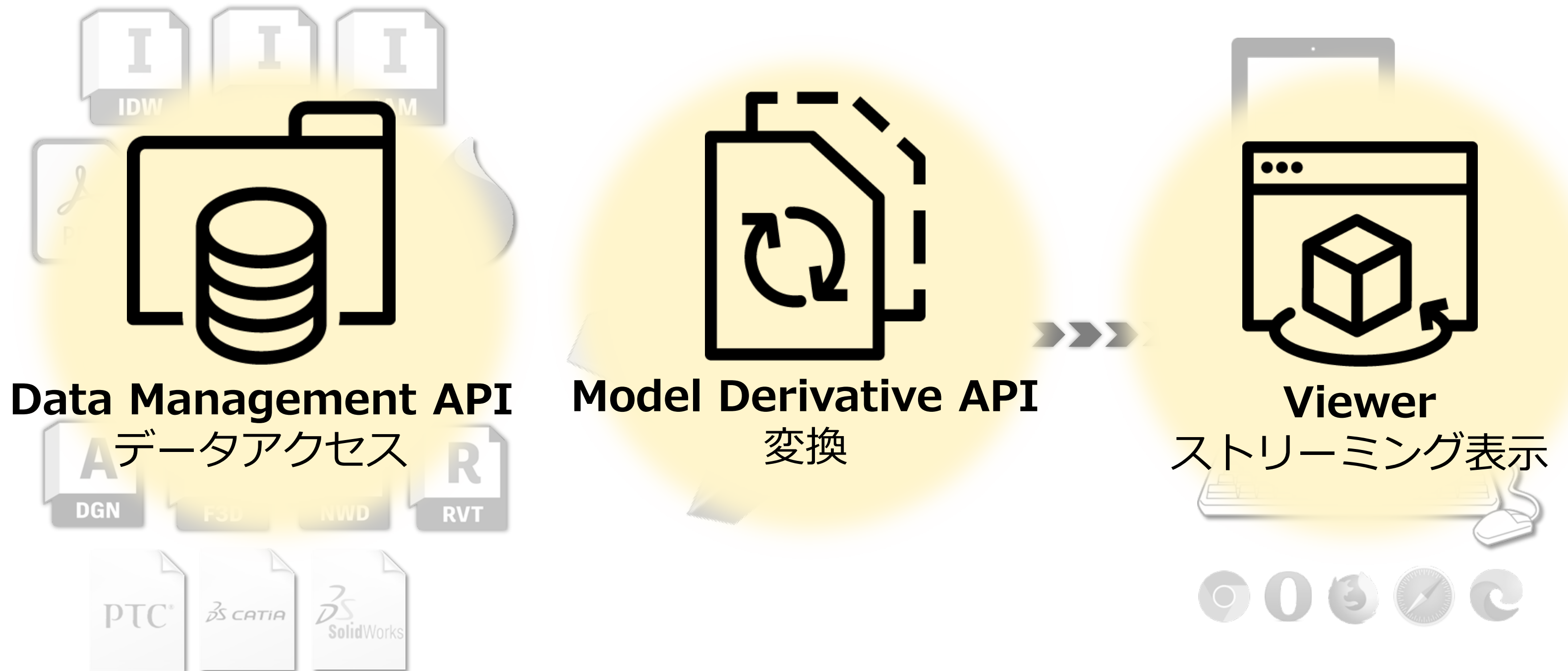
racbasicsampleproject.rvt Upload



Viewer ソリューション

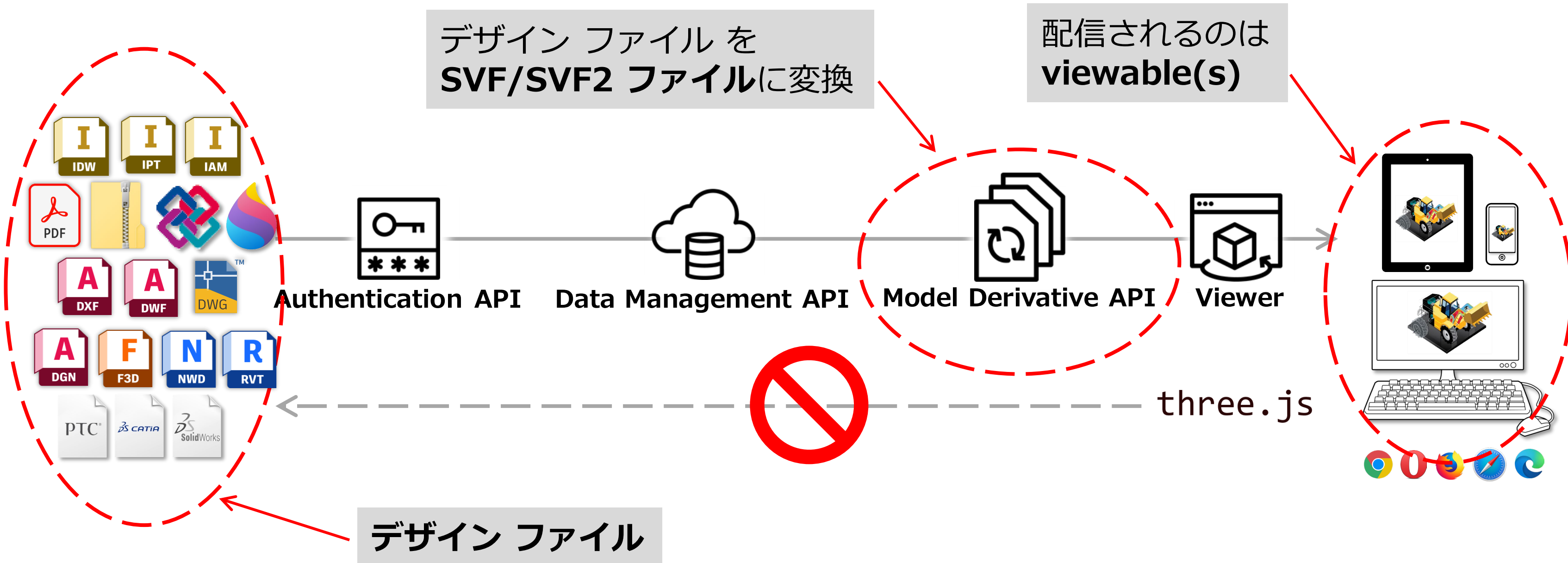


Viewer ソリューション

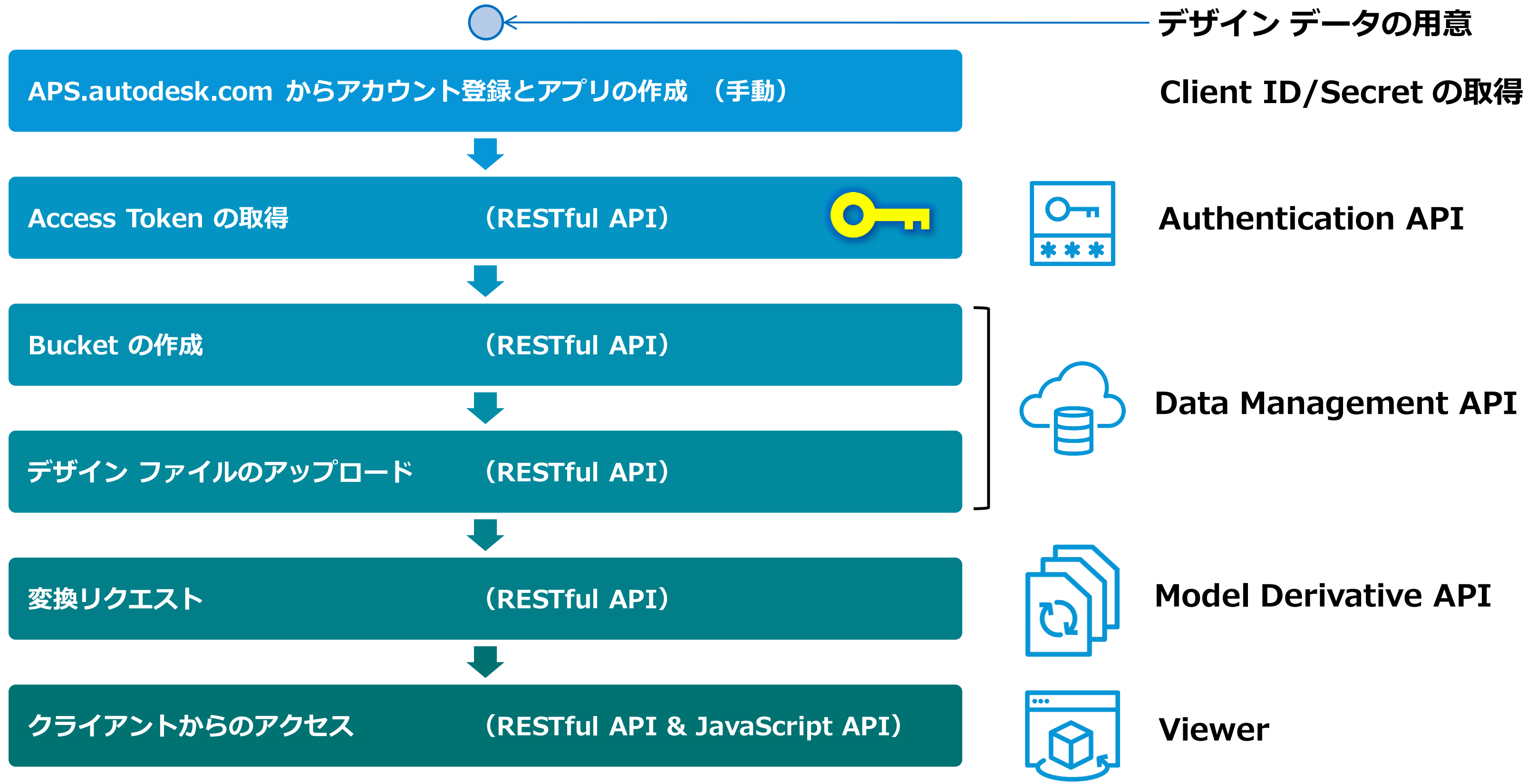


APS Viewer ソリューションのながれ (OSS)

- APS Viewer からのオリジナルデータへの反映は不可

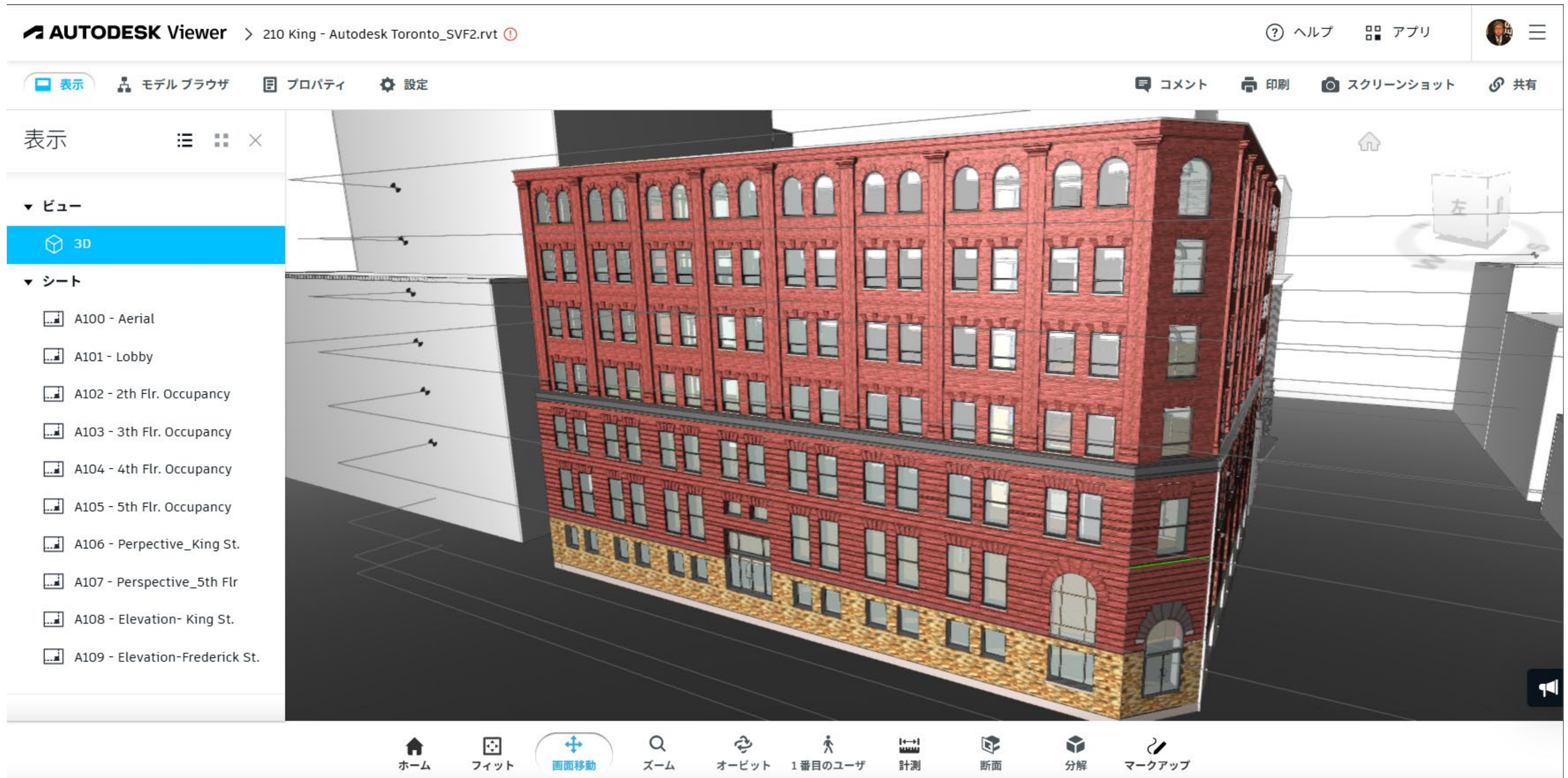


2-legged OAuthでの Viewer 利用手順の理解



Autodesk Viewer

<https://viewer.autodesk.com/>



APS SDK : Simple Viewer でも利用

- エンドポイントをラップするサーバー用途パッケージ
 - エンドポイントをラップ
 - SDKManager と API 毎に分割したパッケージで提供
 - .NET と Node.js (TypeScript) を用意
- **.NET SDK**
 - NuGet パッケージとして提供
 - <https://www.nuget.org/profiles/AutodeskPlatformServices.SDK>
- **Node.js SDK (TypeScript SDK)**
 - Node.js パッケージとして提供
 - <https://www.npmjs.com/~aps.sdk>





Developer's Guide

Overview

- API Basics
 - GDPR Compliance
 - Webhooks
 - Field Guide
 - Supported Translations
- Rate Limits

How-to Guide

Code Samples

Reference Guide

- REST API Reference
- .NET SDK Reference
- TypeScript SDK Reference

Change History

[Documentation](#) / [Model Derivative API](#) / [Developer's Guide](#)

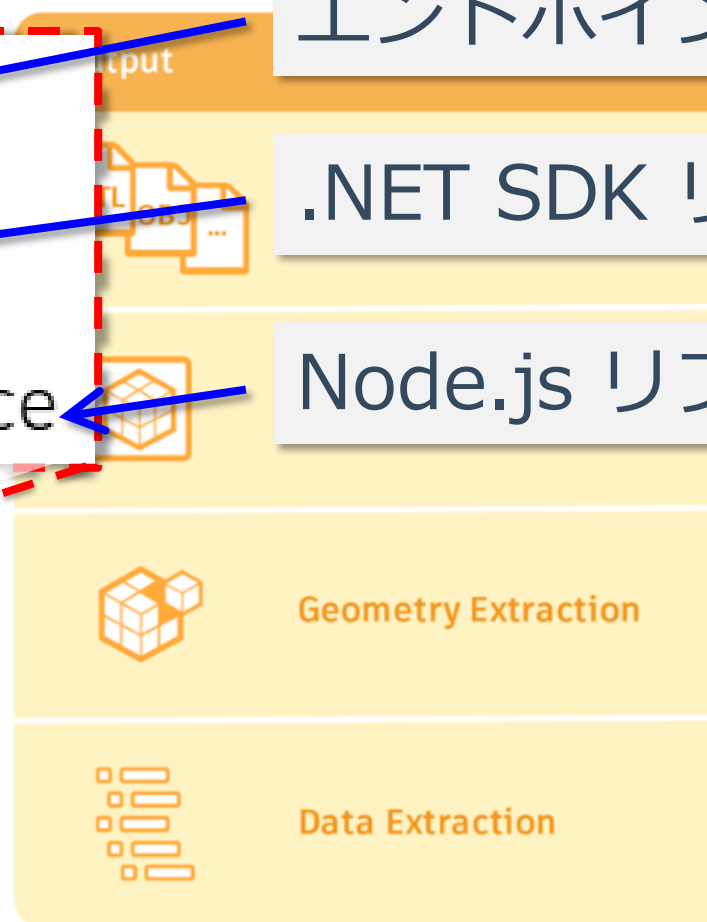
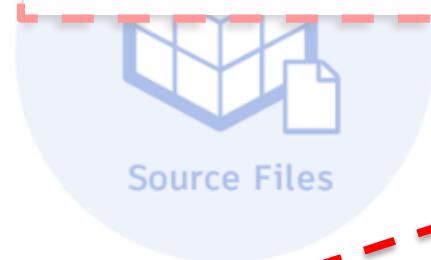
About the Model Derivative API

- REST API Reference
- .NET SDK Reference
- TypeScript SDK Reference

エンドポイント リファレンス

.NET SDK リファレンス

Node.js リファレンス





デベロッパキーの取得

APS を始めるには?

- まずは <https://aps.autodesk.com/> へ

1



2



3

Autodesk ID で
サインイン

APS アプリ作成

開発

- 目的は**デベロッパ**キーの取得
 - **Client ID** (別名 : Consumer Key)
 - **Client Secret** (別名 : Consumer Secret)

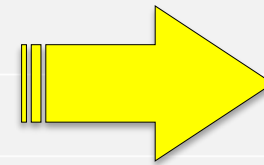
My applications からのアプリ登録

どのAPIを使うか指定

API Access

These are the APIs your app is allowed to access. It can be changed anytime.

- ☒ AEC Data Model API
- ☒ Application Management API
- ☒ Autodesk Construction Cloud API
- ☒ BIM 360 API
- ☒ BuildingConnected API
- ☒ Data Exchange API
- ☒ Data Management API
- ☒ Design Automation API
- ☒ Flow Graph Engine API
- ☒ Manufacturing Data Model API
- ☒ Model Derivative API
- ☒ Parameters API
- ☒ Premium Reporting API
- ☒ Reality Capture API
- ☒ Tandem Data API
- ☒ Token Flex Usage Data API
- ☒ Webhooks API



My applications からのアプリ登録

課金対象のチーム

Team Assignment

Assign an [Autodesk team](#) to pay for any charges incurred by this application

Toshiaki Isezaki - 2013



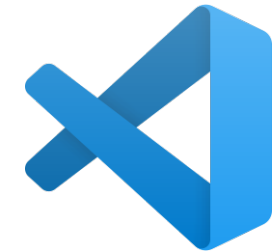
Assign

アカウントで複数のチームを編成している場合には、プライマリ管理者、または、セカンダリ管理者に適切なチームを確認してください。通常はアカウントに 1 つのチームしか関連付けられていないので、表示されるチームを割り当て（Assign）ます。

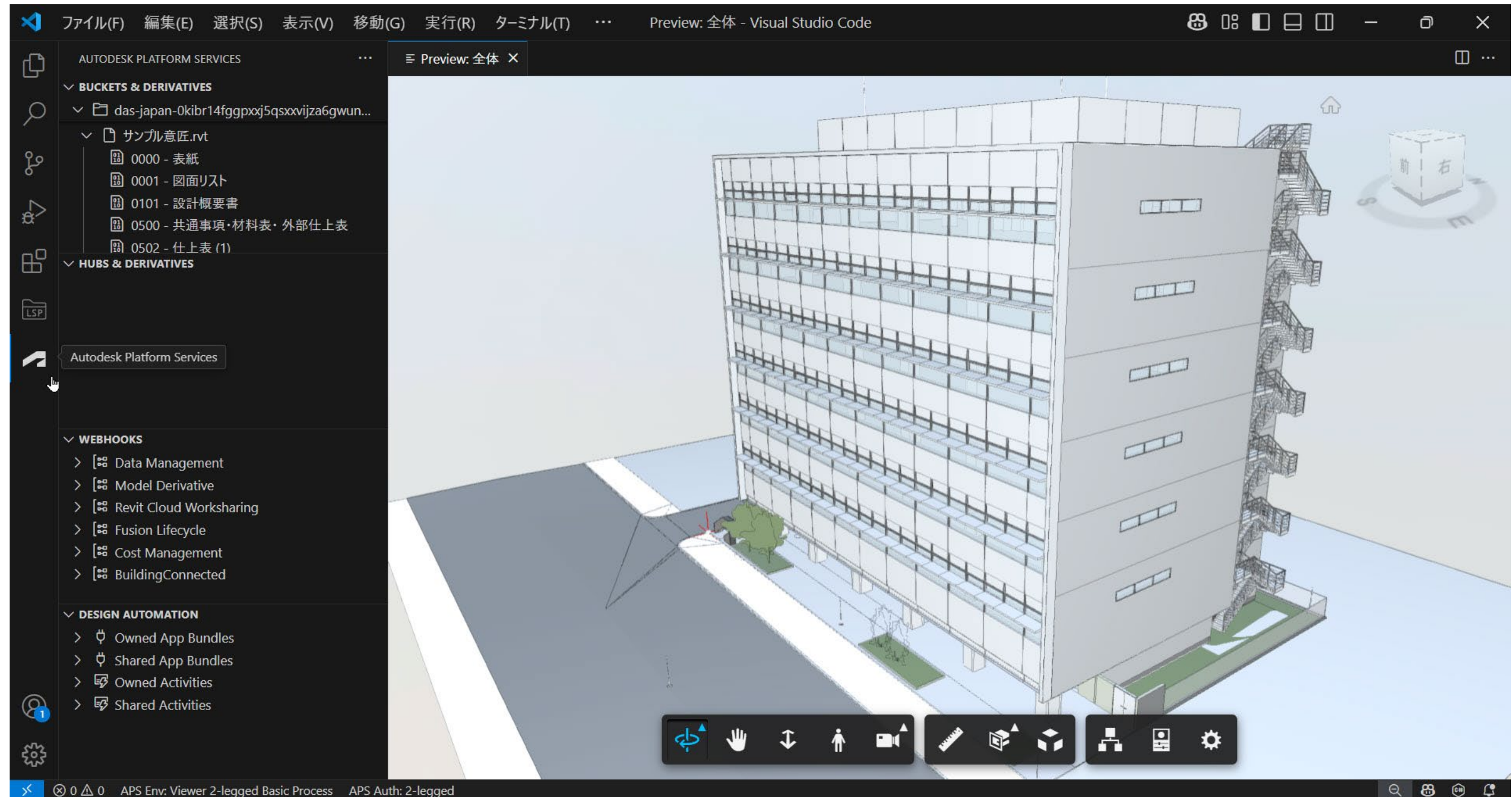
今回使用する開発環境

これに限定するものではありません

- Web ブラウザ : **Google Chrome**
- HTML/JavaScript エディタ : **VS Code**
- Web サーバー実装 : **Node.js**
- テスト ツール : **Postman**
- リポジトリ ユーティリティ : **git for Windows**



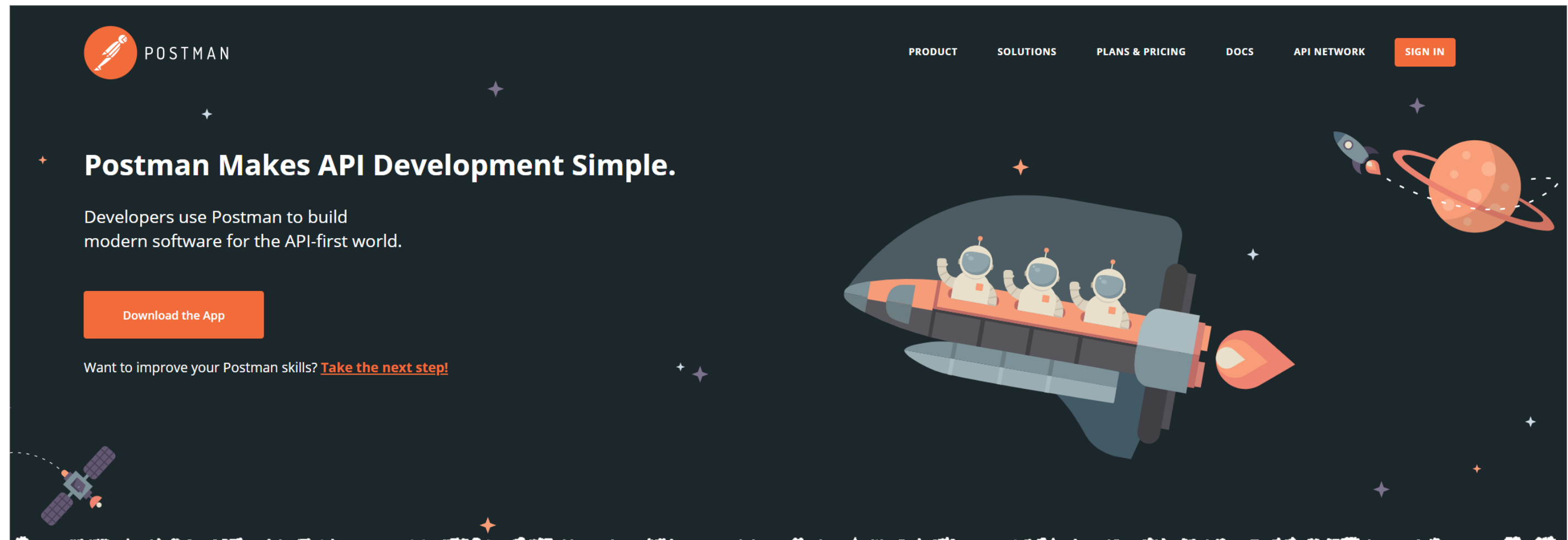
VS Code - APS エクステンション



RESTful API のテストツール



- Postman
 - <https://www.getpostman.com/>
 - コードを書かずに RESTful API のテストが可能





セッティング

 **AUTODESK**
Platform Services



 Sign in

Solutions ▾

Getting Started

Documentation ▾

Success Stories

Community ▾

Support ▾

Pricing

App Store ▾

Getting Started

Environment Setup

Tutorials

Simple Viewer

Application Setup

Authentication

Data & Derivatives

Viewer & UI

Hubs Browser

Dashboard

Design Automation

ACC Administrator

Learn More

 > Tutorials > Simple Viewer > Application Setup

Application Setup

Let's start by creating a new project, installing required dependencies, and setting up a basic server application.

New project

Node.js & VSCode

.NET & VSCode

.NET & VS2022

Create a new folder for your project, navigate to it in the command line, and initialize a new Node.js project:

```
npm init -y
```

Next, install all the Node.js dependencies we're going to use. In this case it will be

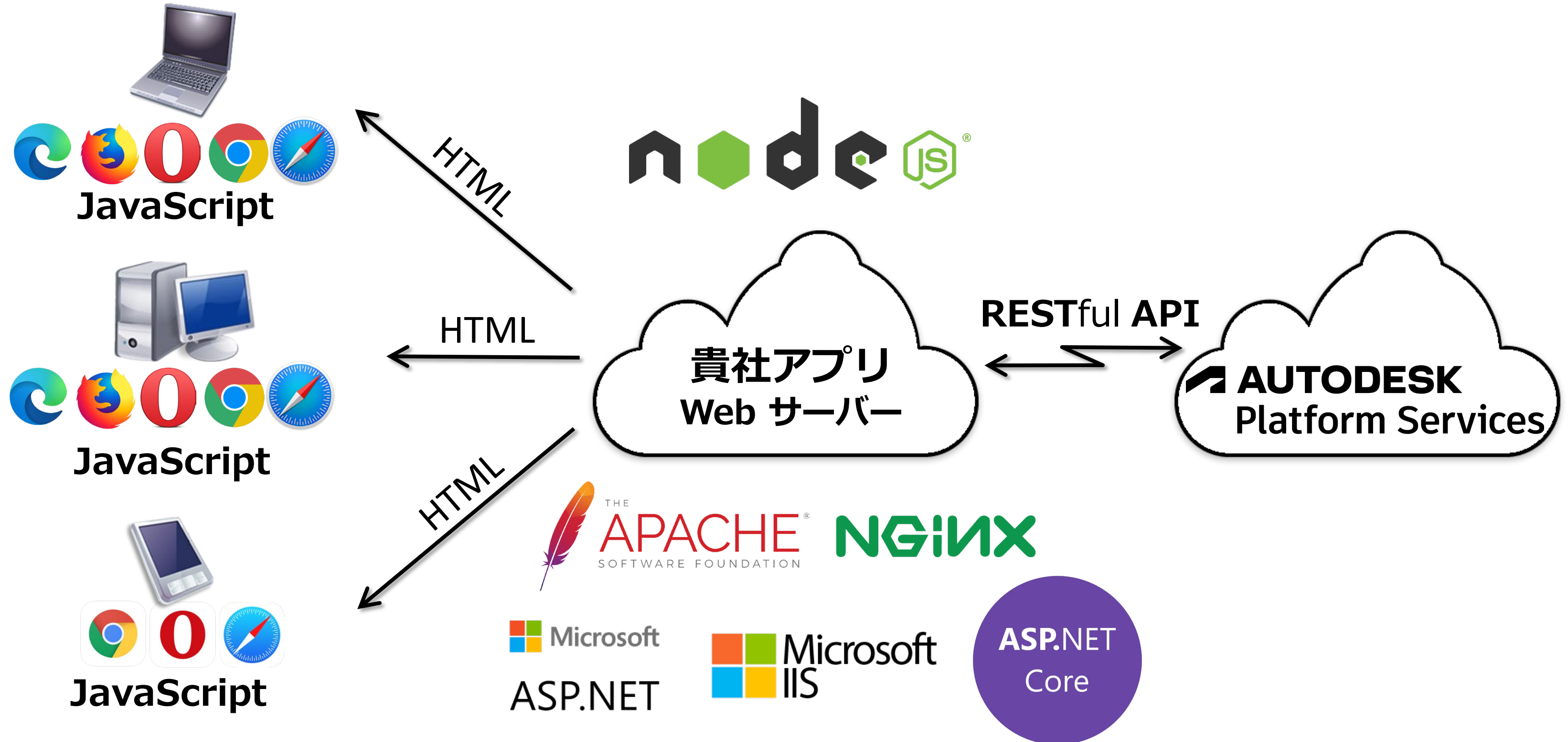
New project

Application config

Basic server

Try it out

一般的な Web アプリの構成



Node.js とは

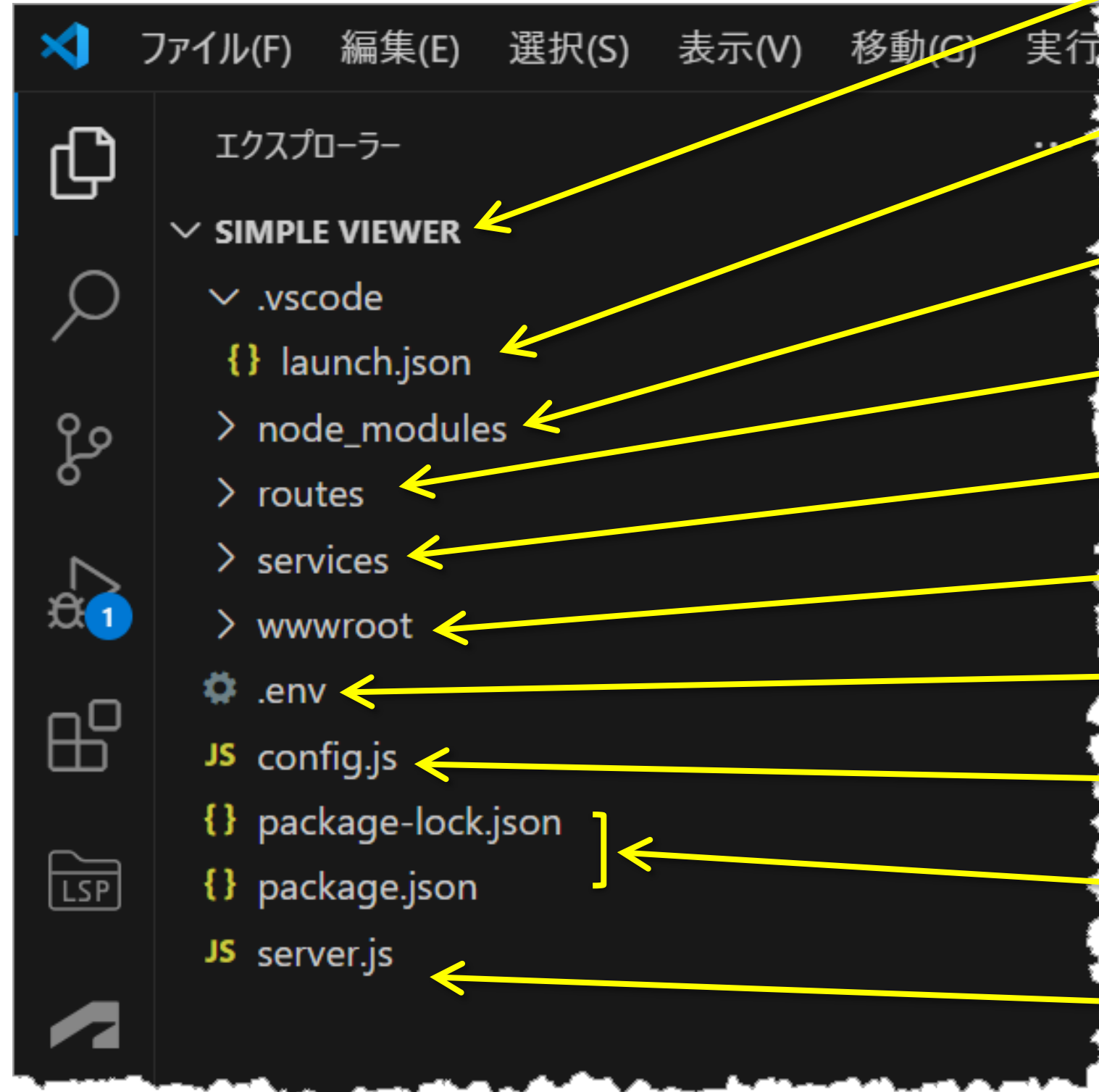


- オープンソース
- JavaScript をサーバー上で実行するための環境
- Google V8 JavaScript Engine が使用
- Node Package Manager で拡張可能
- 今回使用しているパッケージ(ミドルウェア) :
 - @aps_sdk/...
 - @types/forged-viewer
 - dotenv
 - express
 - express-formidable

VS Code 上の操作手順

1. プロジェクト フォルダの作成、VS Code でフォルダを開く
2. VS Code のターミナルで Node.js の初期化
3. ターミナルでAPS Node.js SDK パッケージのインストール
4. **package.json** ファイル内 “**dependencies**” の記述追記
5. **wwwroot**、**routes**、**services** サブフォルダの作成
6. **config.js** ファイルの作成、実装
7. **.env** ファイルの作成と Client ID/Secret の指定
8. **.vscode** フォルダ下に **launch.json** ファイルを作成、実装
9. **server.js** ファイルの作成とサーバー実装記述の追加
10. **package.json** ファイル内の “**scripts**” の記述変更
11. デバッグ実行 >> [**localhost:8080**](http://localhost:8080)

ここまでのプロジェクト構成



プロジェクト フォルダを
Simple Viewer とした場合

VS Code デバッグで使用

依存関係からインストール
された Node.js パッケージ

ルーティング実装一式

サーバー実装一式

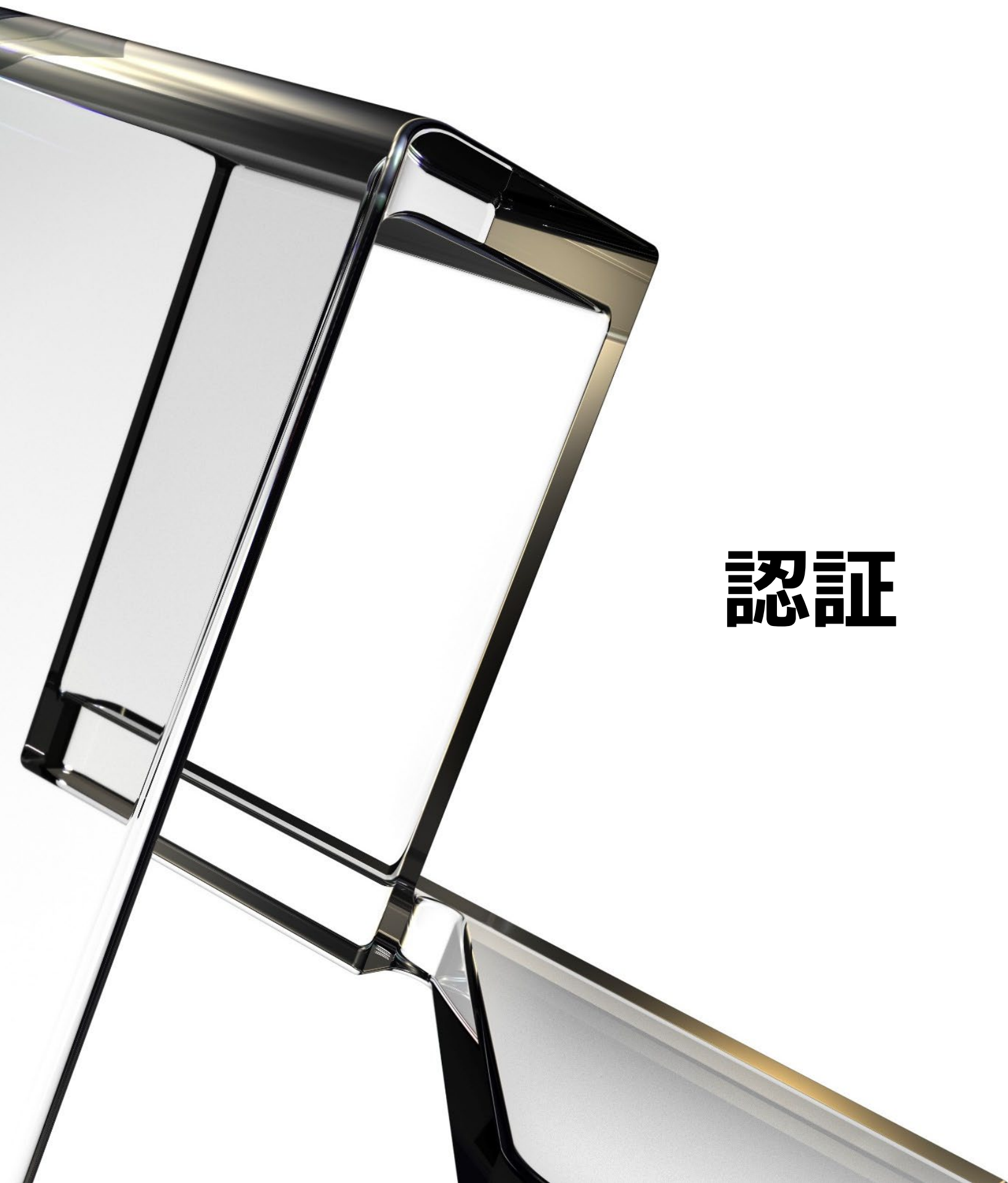
クライアント構成一式

Client ID、Secret を記述

Node.js の起動時情報指定

Node.js パッケージ依存関係

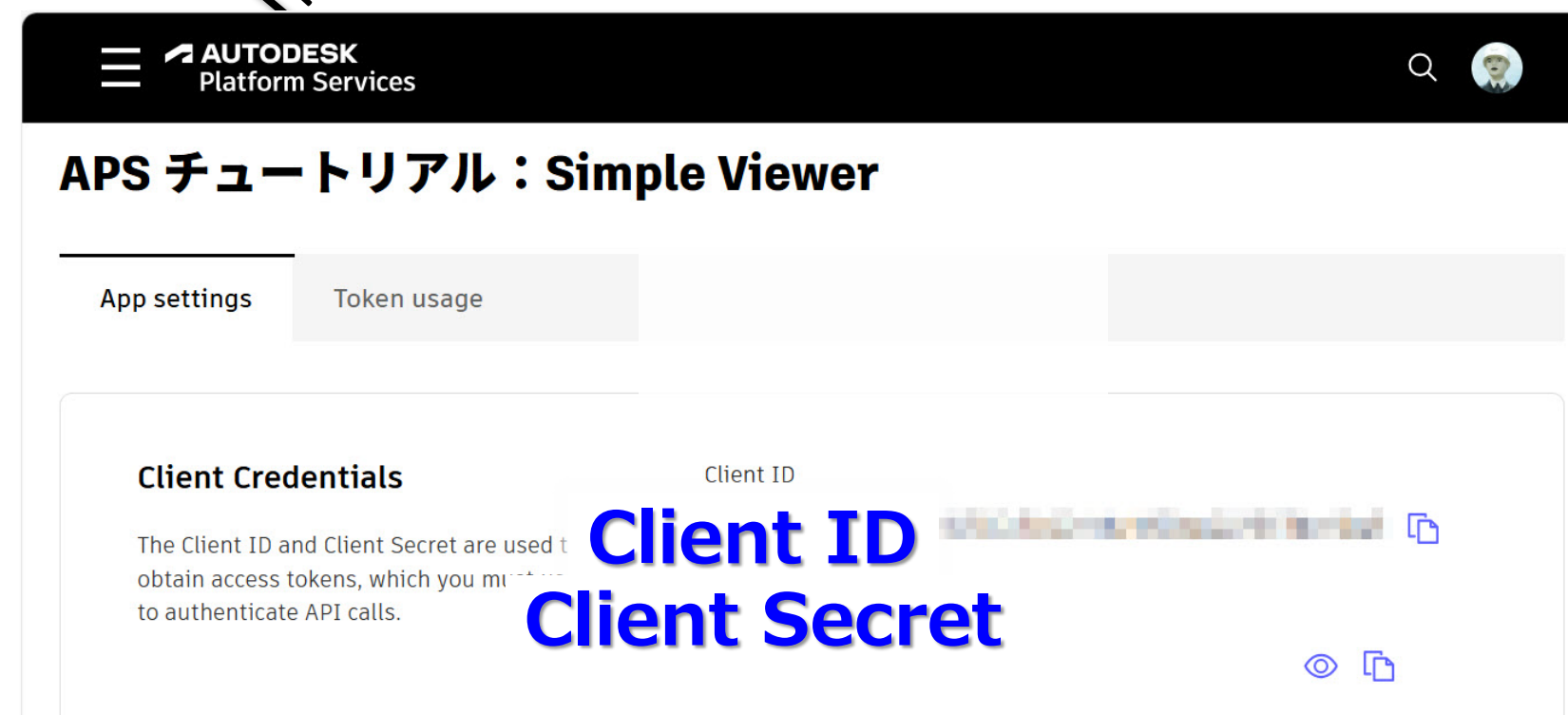
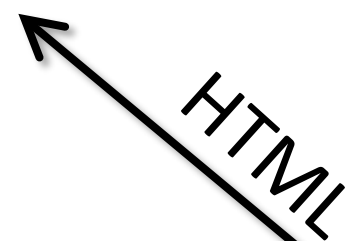
Node.js サーバーの起動構成



認証

デベロッパ キーとアクセス トークン

APS ポータル (aps.autodesk.com)



API

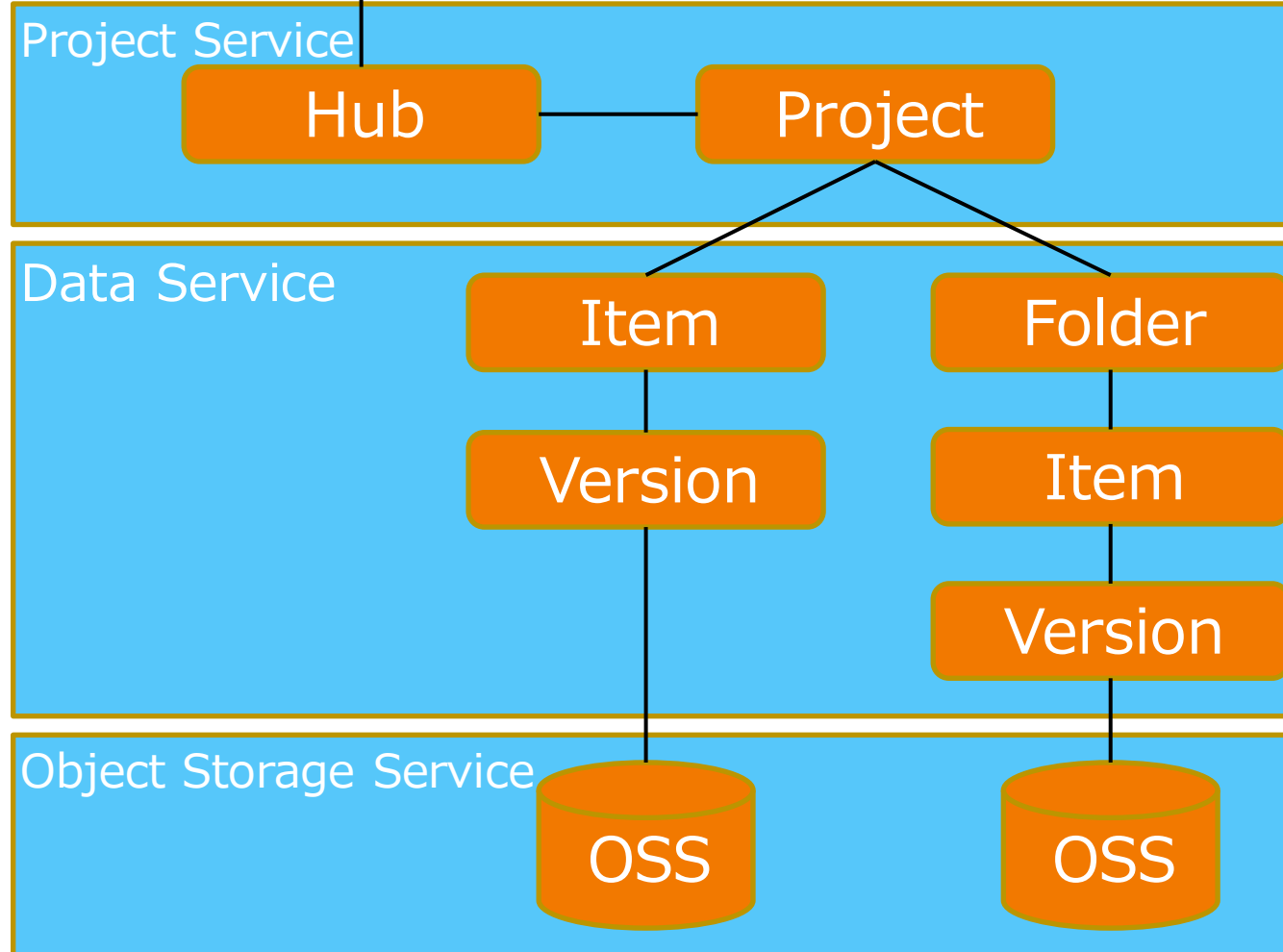


シナリオ/目的に応じた認証フロー

Data Management

APS アプリ

3-legged OAuth

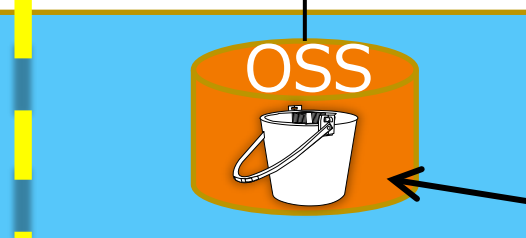


APS アプリが Fusion Team、ACC のユーザデータ領域に直接アクセスして運用
- 3 - Legged OAuth -

OSS

APS アプリ

2-legged OAuth



APS アプリが Bucket でデータを直接管理・運用
- 2 - Legged OAuth -

Bucket

アクセス トークン (Access Token) とは

- クラウド リソース へのアクセス権限をチェックする仕組み
 - 有効期限が設定される
- 生成に必要なもの
 - デベロッパキー
 - Client ID と Client Secret のペア
 - APS ポータルでアプリ (App) 作成時に取得可能
 - スコープ (Scope)
 - リソースへのアクセス権限を指定
 - 使用するリソースによって適切に使い分けが必要
 - 複数の Scope 文字列を半角スペースで結合して指定

Scope（スコープ）とは

Scope 文字列	意味
user-profile:read	プロフィール（Autodesk ID）の表示
user:read	プロフィール（Autodesk ID）の読み取り
user:write	プロフィール（Autodesk ID）の書き込み
viewables:read	変換後のデザインデータ（SVF）の読み取り（表示）
data:read	ストレージデータの読み取り
data:write	ストレージデータの書き込み（編集）
data:create	ストレージデータの作成
data:search	ストレージデータの検索
bucket:create	新しい Bucket の作成
bucket:read	Bucket の読み取り
bucket:update	Bucket の更新
bucket:delete	Bucket の削除
code:all	コードの生成または実行（Design Automation API）
account:read	アプリやサービス アカウントの読み取り
account:write	アプリやサービス アカウントの書き込み

適切な OAuth スコープ

- リファレンスで エンドポイント 毎に必要なスコープをチェック！

GET buckets

This endpoint will return the buckets owned by the application. This endpoint supports pagination.

Resource Information

Method and URI	GET https://developer.api.autodesk.com/oss/v2/buckets
Authentication Context	app-only
Required OAuth Scopes	bucket :read
Data Format	JSON

bucket:read

POST buckets

Creates a bucket. Buckets are arbitrary spaces that are created by applications and are used to store objects for later retrieval. A bucket is owned by the application that creates it.

Resource Information

Method and URI	POST https://developer.api.autodesk.com/oss/v2/buckets
Authentication Context	app-only
Required OAuth Scopes	bucket :create
Data Format	JSON

bucket:create

AUTODESK
Platform Services

 Sign in

Solutions ▾

Getting Started

Documentation ▾

Success Stories

Community ▾

Support ▾

Pricing

App Store ▾

Getting Started

Environment Setup

Tutorials

Simple Viewer

Authentication

Data & Derivatives

Viewer & UI

Hubs Browser

Dashboard

Design Automation

ACC Administrator

Learn More

 > Tutorials > Simple Viewer > Authentication

Authentication

In this step we're going to extend the server implementation so that it can authenticate itself to the APS platform, and generate access tokens for different use cases.

 TIP

It is a good practice to generate an "internal" token with more capabilities (for example, allowing you to create or delete files in the Data Management service) that will only be used by the server, and a "public" token with fewer capabilities that can be safely shared with the client-side logic.

Access tokens

Node.js & VSCode

.NET & VSCode

.NET & VS2022

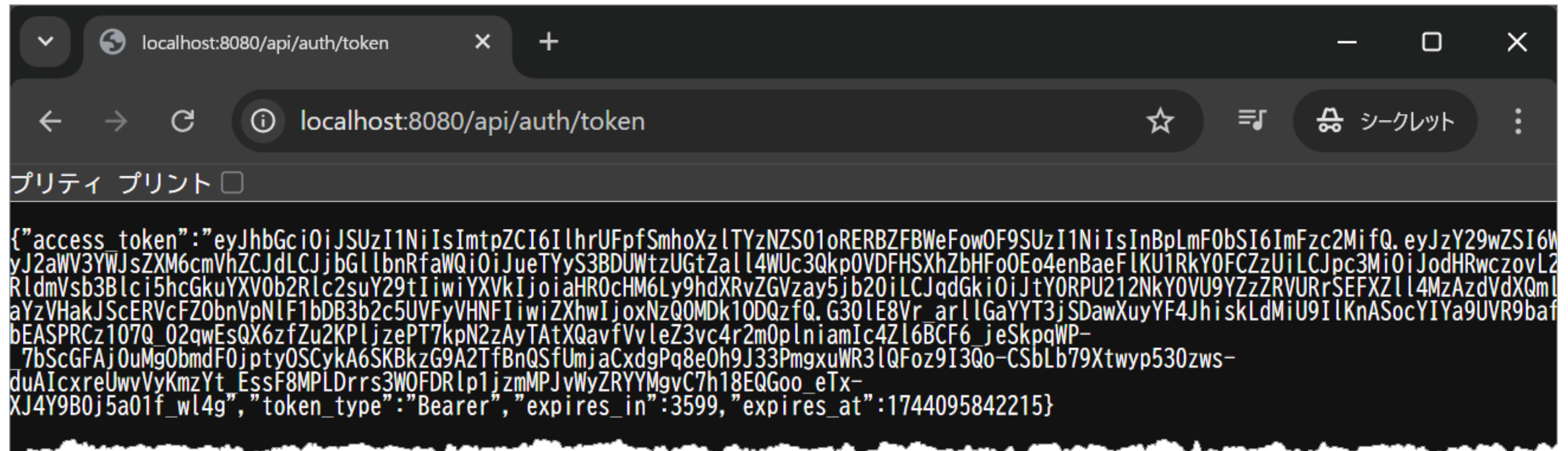
Access tokens

Server endpoints

Try it out

VS Code 上の操作手順

1. **services** フォルダ下に **aps.js** ファイルを作成して実装
2. **routes** フォルダ下に **auth.js** ファイルを作成して実装
3. **server.js** ファイルに **auth.js** をマウント
4. デバッグ実行 >> localhost:8080/api/auth/token



Authentication | Autodesk Platform Services

get-started.aps.autodesk.com/tutorials/simple-viewer/auth

本人確認を行ってください

AUTODESK
Platform Services

Solutions

Getting Started

Documentation

Success Stories

Community

Support

Pricing

App Store

Getting Started

Environment Setup

Tutorials

Simple Viewer

Authentication

Data & Derivatives

Viewer & UI

Hubs Browser

Dashboard

Design Automation

ACC Administrator

Learn More

Home

Tutorials

Simple Viewer

Authentication

Authentication

In this step we're going to extend the server implementation so that it can authenticate itself to the APS platform, and generate access tokens for different use cases.

TIP

It is a good practice to generate an "internal" token with more capabilities (for example, allowing you to create or delete files in the Data Management service) that will only be used by the server, and a "public" token with fewer capabilities that can be safely shared with the client-side logic.

Access tokens

Node.js & VSCode

.NET & VSCode

.NET & VS2022

Access tokens

Server endpoints

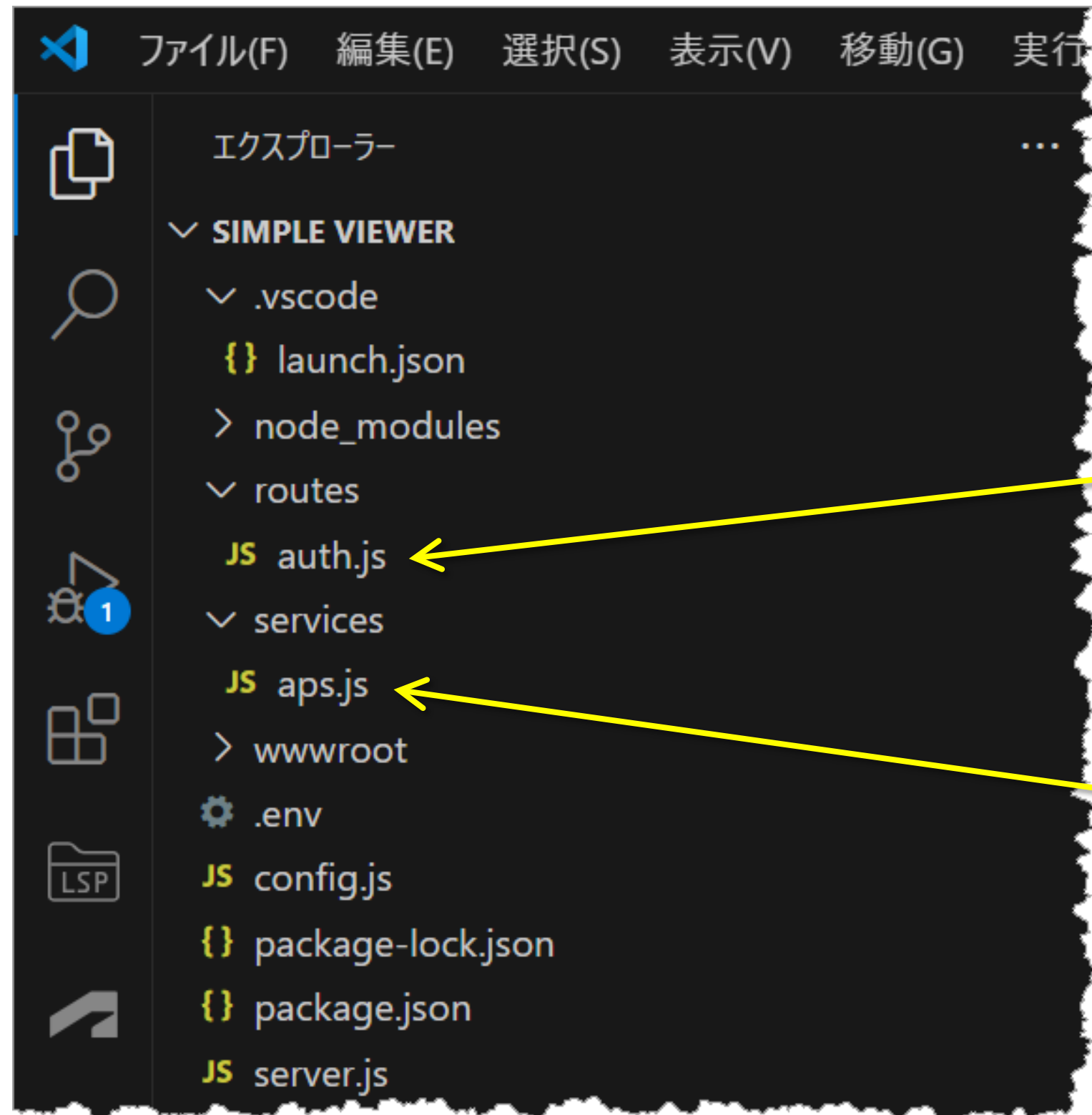
Try it out

6

検索

11:00
2025/04/11

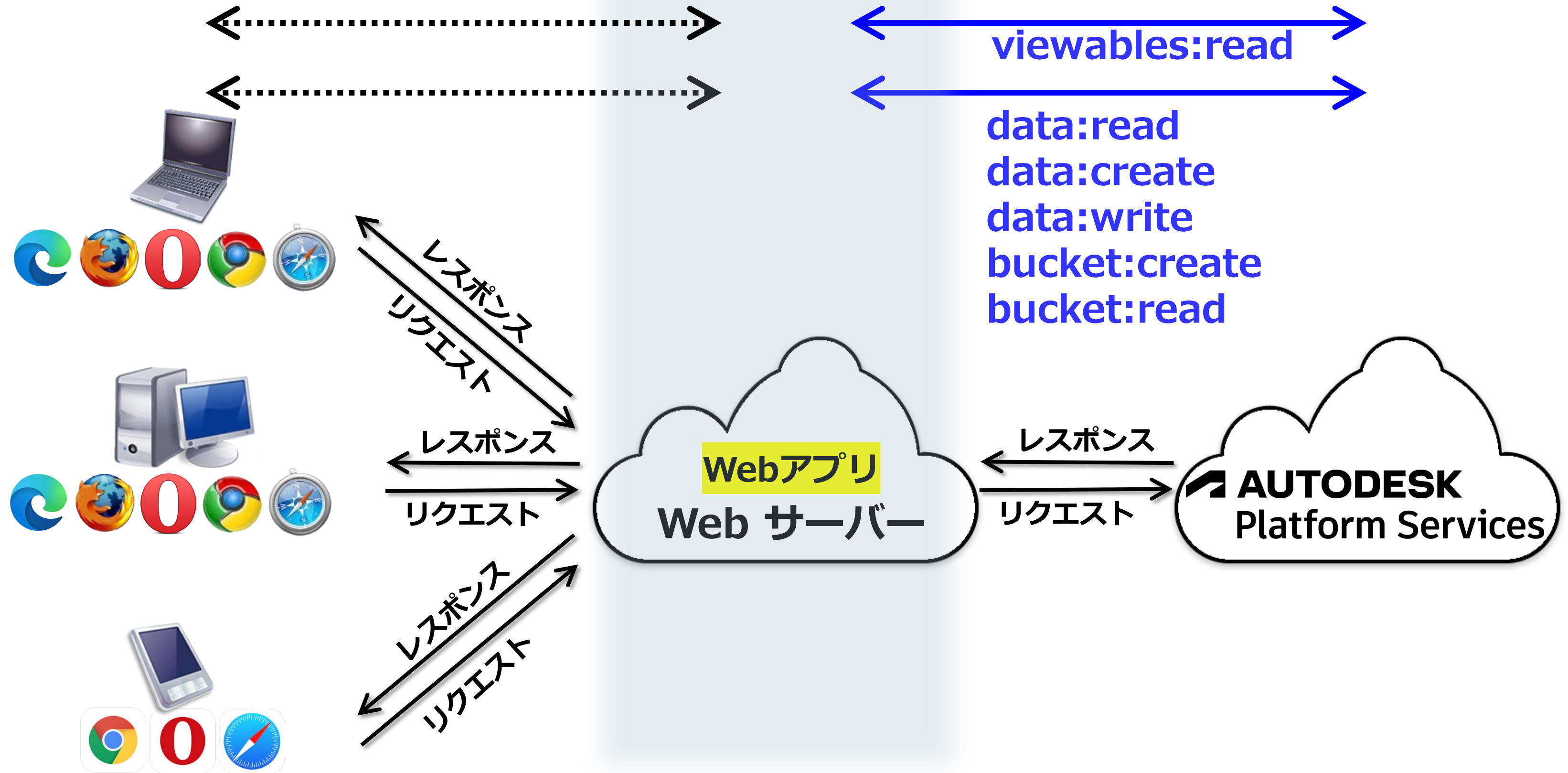
ここまでのプロジェクト構成



クライアントが呼ぶ出す
エンドポイントを実装
(Client ID/Secret を隠蔽し
てアクセストークンを取得)

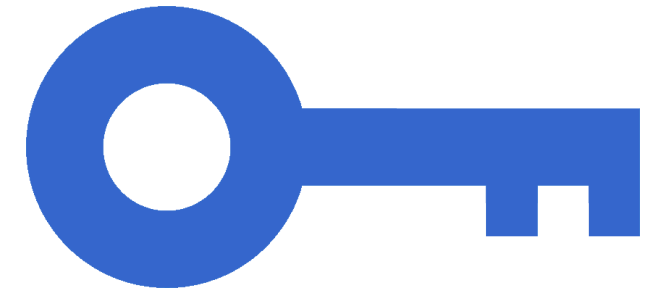
サーバー実装からアクセス
トークンを取得する関数定義

スコープによるトークンを使い分け



その他、2-legged 認証単体の評価

- アクセストークンの取得
- Postman
 - Postman を使った 2-legged 認証(OAuth v2)
- VS Code
 - 新Node.js SDKとVS Codeで2-leggedアクセストークンを取得



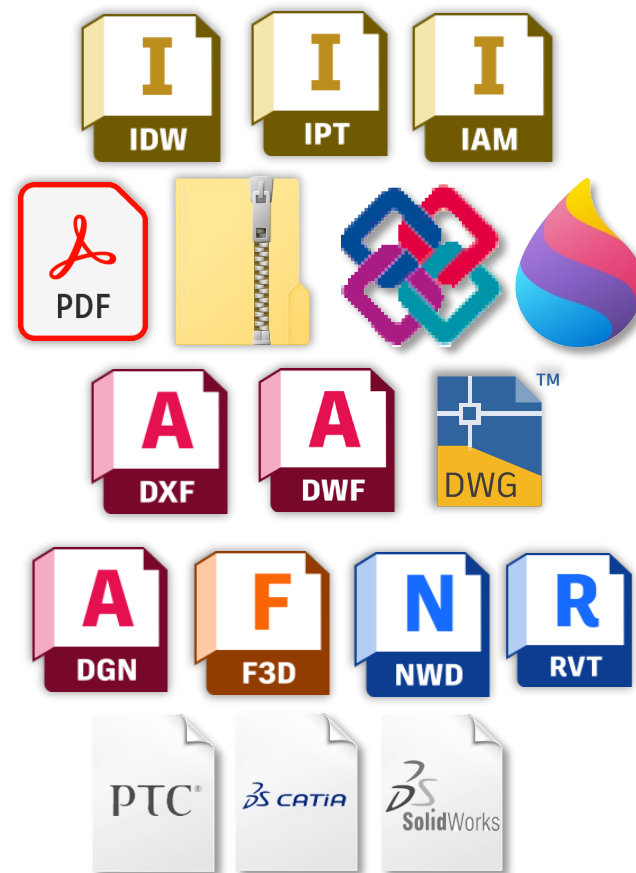
An abstract, three-dimensional geometric structure composed of metallic, reflective surfaces and white, translucent planes. The structure is tilted and viewed from a low angle, creating a sense of depth and complexity. It features sharp edges and smooth curves, with light reflecting off the metallic surfaces.

データと変換

Bucket とは

- Bucket は APS 共有ストレージ内の 'フォルダ'
 - Bucket はデザイン ファイルをアップロード/変換する場所
 - 他の Bucket と重複しない一意な名前が必要
 - 半角英字(小文字)、数字、 記号(- _ .)

Client ID 利用を推奨

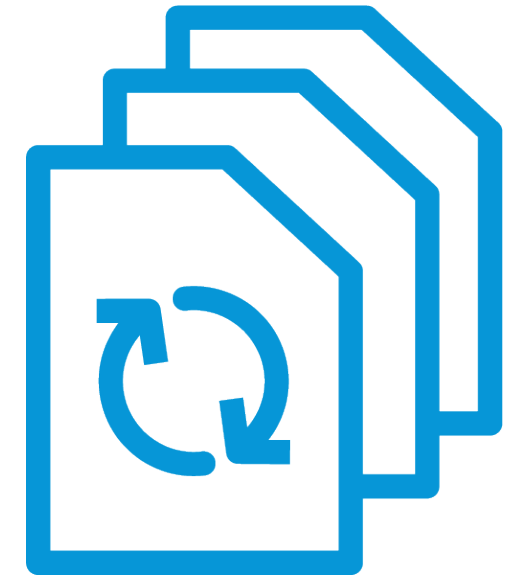


Bucket ポリシー

- Bucket 自身は永続的
- 作成時に指定する**ポリシー**で内部リソースの寿命を決定
 - **transient** : 保存されるファイルは24時間で自動消去
 - **temporary** : 保存されるファイルは30日で自動消去
 - **persistent** : 保存されるファイルは消去まで維持
- Bucket への永続的なアクセス条件
 - 作成時と同じ Client ID を使用
 - 異なる Client ID を持つアプリはアクセス不可

変換済ファイルの再変換について

- 毎回変換が実行される訳ではない
 - マニフェストが残っていると再変換はしない
- 再変換を期待する場合は
 1. 変換処理前にマニフェストを削除（确实）
 - [DELETE :urn/manifest](#) エンドポイントまたは
 2. 変換ジョブで再変換を指定
 - [POST job](#) エンドポイント の Header パラメータ : **x-ads-force**
- 注：Viewer 上での確認時にはブラウザ キャッシュを消去!!



 **AUTODESK**
Platform Services



 Sign in

Solutions ▾Getting StartedDocumentation ▾Success StoriesCommunity ▾Support ▾PricingApp Store ▾

Getting Started
Environment Setup
Tutorials
Simple Viewer
Application Setup
Authentication
Data & Derivatives
Viewer & UI
Hubs Browser
Dashboard
Design Automation
ACC Administrator
Learn More

 > Tutorials > Simple Viewer > Data & Derivatives

Data & Derivatives

In this step we will extend our server so that we can list models, upload them, and prepare them for viewing.

Data management

First, let's make sure that our application has a bucket in the Data Management service to store its files in. Typically the bucket would be created just once as part of a provisioning step but in our sample we will implement a helper function that will make sure that the bucket is available, and use it in other parts of the server app.

When creating buckets, it is required that applications set a retention policy for objects stored in the bucket. This cannot be changed at a later time. The retention policy on the bucket applies to all objects stored within. When creating a bucket, specifically set the **policyKey** to **transient**, **temporary**, or **persistent**.

Data management
Derivatives
Server endpoints
Try it out

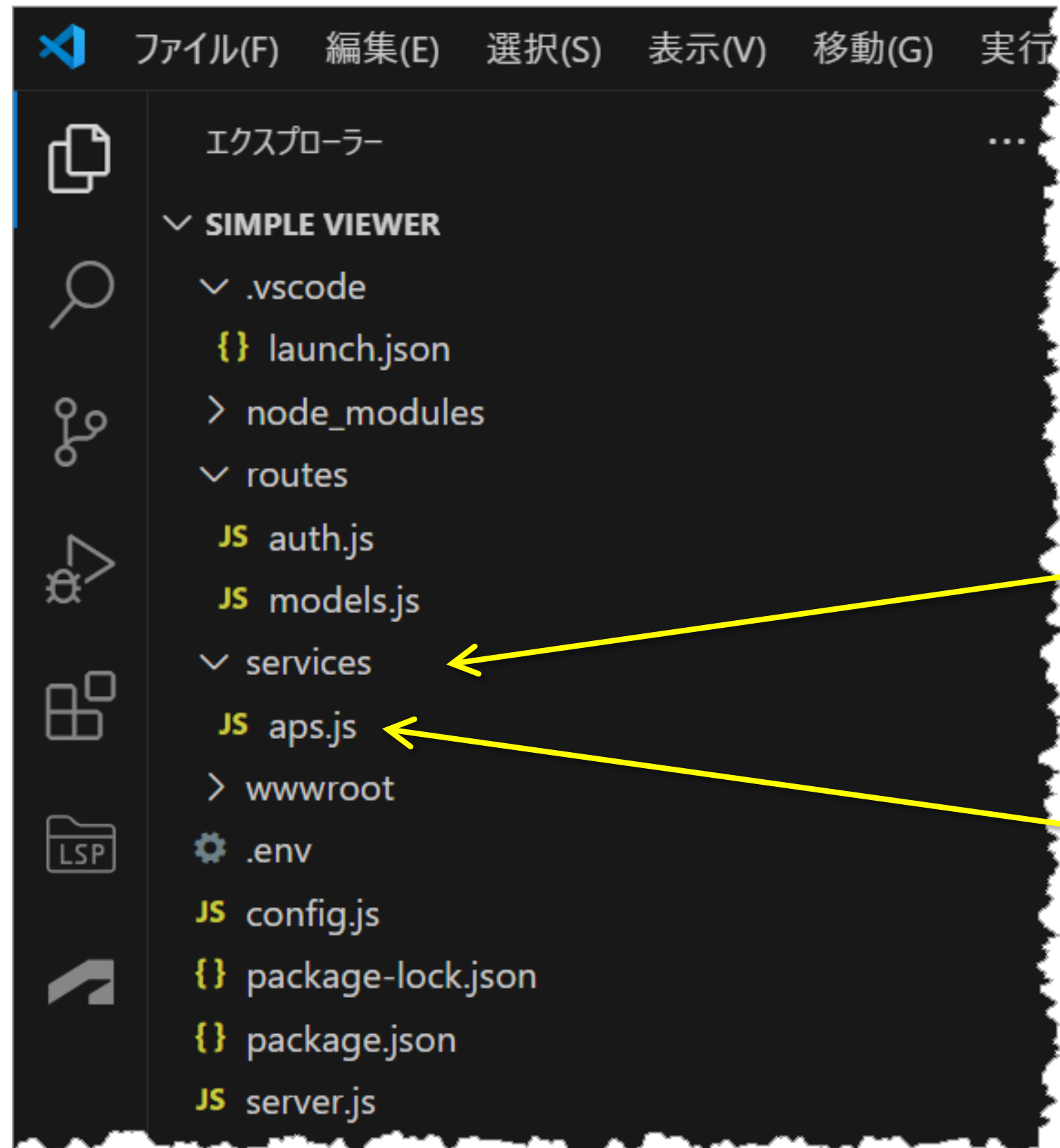


VS Code 上の操作手順

1. **services** フォルダ下の **aps.js** ファイルに実装を追加
 - Data Management API
 - Model Derivative API
2. **routes** フォルダ下に **models.js** ファイルを追加して実装
3. **server.js** ファイルに **models.js** をマウント
4. デバッグ実行 >> <http://localhost:8080/api/models>



ここまでのプロジェクト構成



クライアントが呼ぶ出す
エンドポイントを実装
(ファイルアップロードと変
換指示、変換状況の確認)

サーバー実装からアクセス
トークンを取得する関数定義

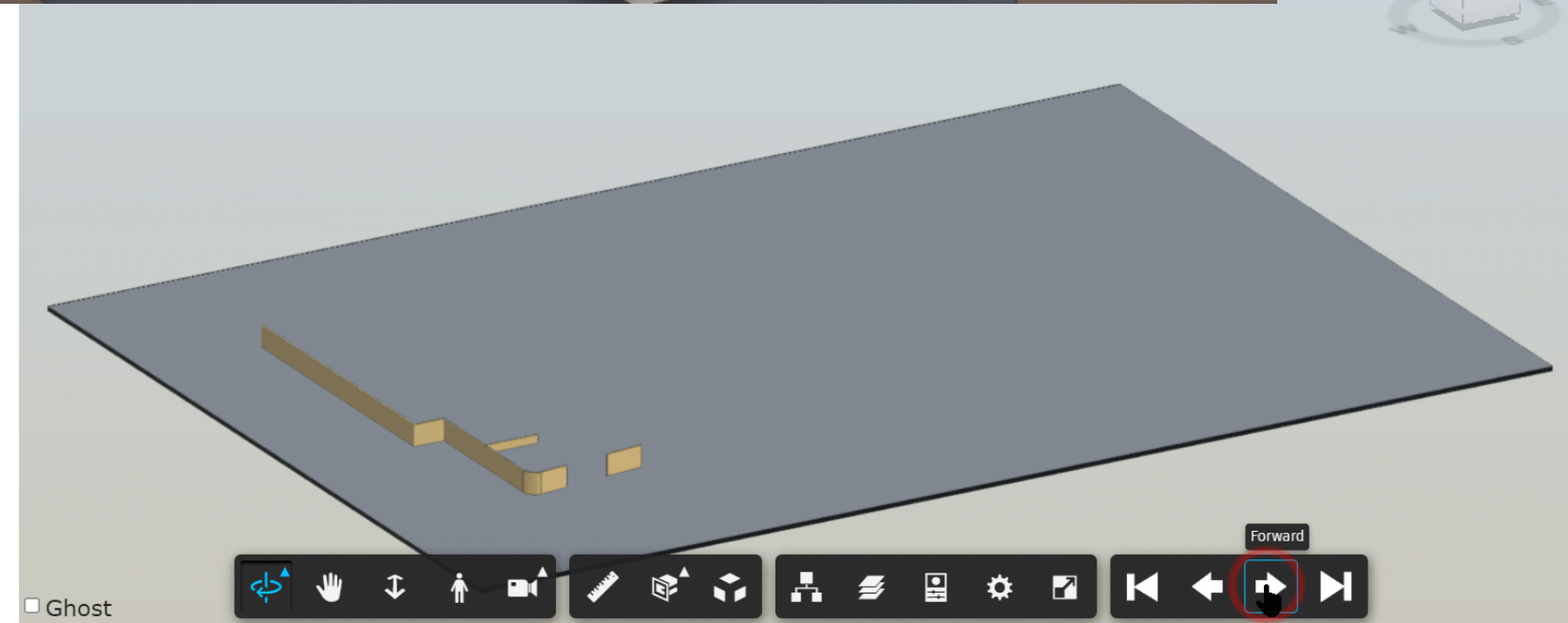
Advanced オプション

- POST job
 - advanced オプション

```
"advanced": {  
  "generateMasterViews": true  
}
```

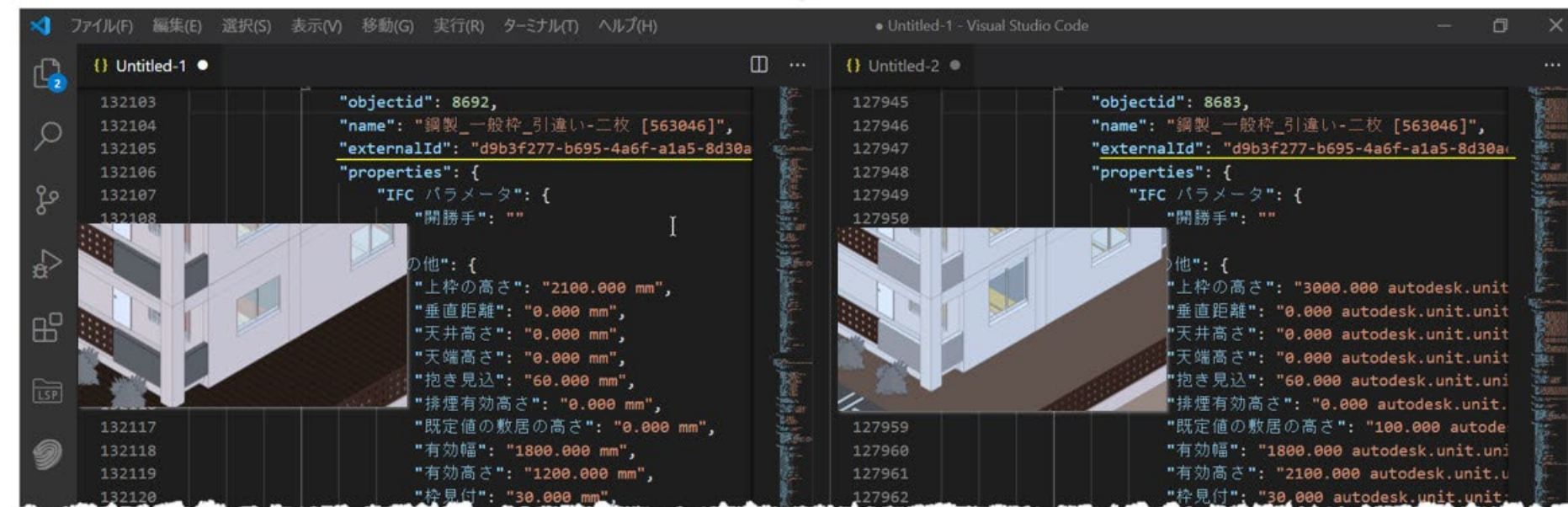
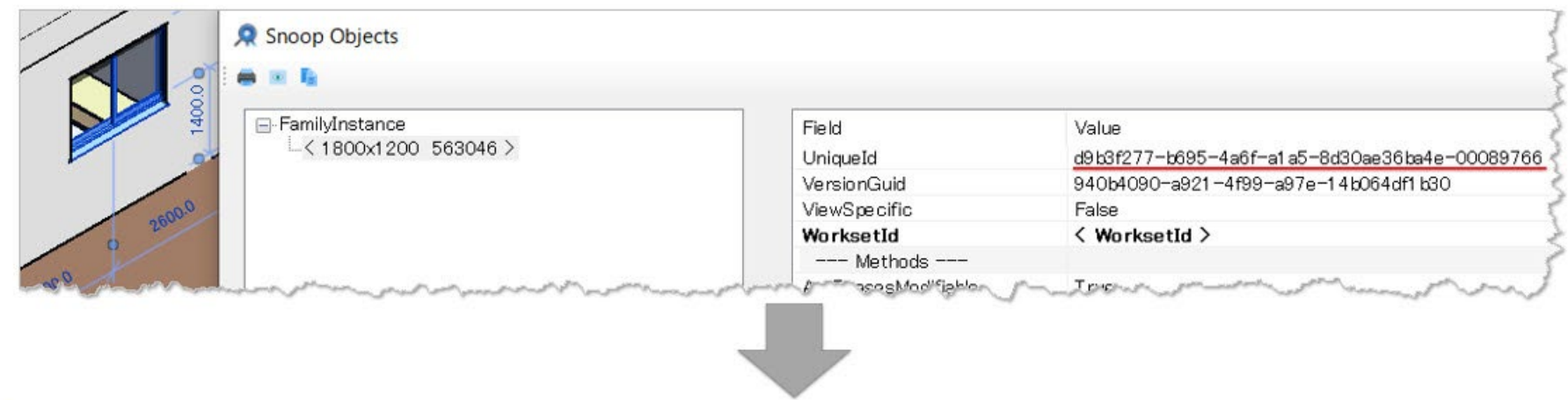
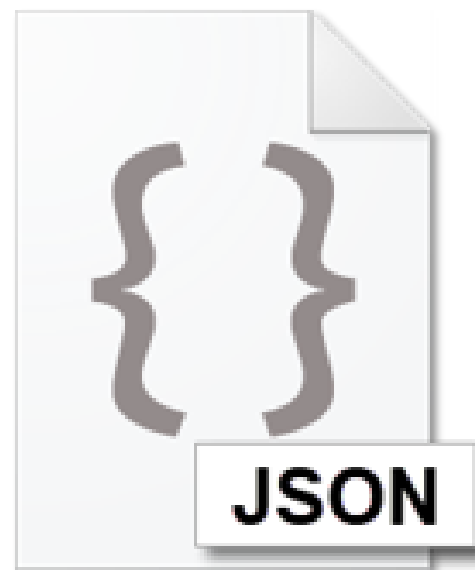


```
"advanced": {  
  "timelinerProperties": true  
}
```



メタデータの活用

- ビュー毎の情報を JSON で取得可能
 - モデル ツリー
 - オブジェクト プロパティ





Viewer と UI

Viewer と UI

56

 **AUTODESK**
Platform Services

 Sign in

Solutions ▾

Getting Started

Documentation ▾

Success Stories

Community ▾

Support ▾

Pricing

App Store ▾

Getting Started

Environment Setup

Tutorials

Simple Viewer

Viewer & UI

Hubs Browser

Dashboard

Design Automation

ACC Administrator

Learn More

 > Tutorials > Simple Viewer > Viewer & UI

Viewer & UI

Finally, we're ready to build the client-side piece of our application.

 TIP

If you're developing with Node.js, you can use TypeScript definitions for the Viewer API. Run

```
npm install --dev @types/forge-viewer
```

in your terminal to add the TypeScript definition file to your project.

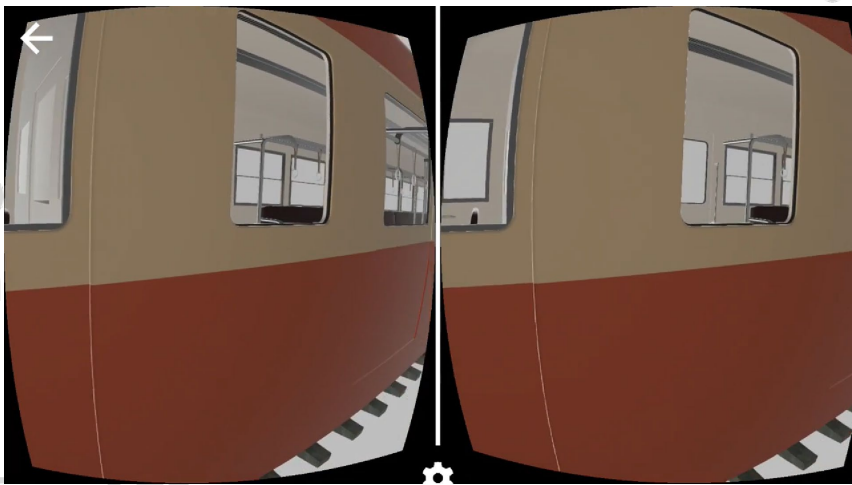
Viewer logic

Viewer logic

Application logic

User interface

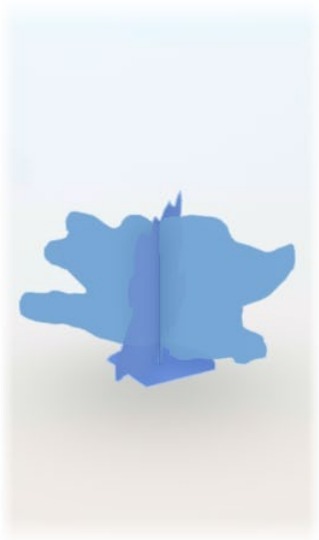
Try it out



以前のバージョン

プロパティ

- 拘束
 - レベル: レベル1
 - ホスト: レベル:レベル1
 - オフセット: 0 mm
 - 付近の要素と...: いいえ
- 識別情報
 - タイプ名: ヤマボウシ-3.0メートル
 - イメージ



バージョン 6.3

RPC 木 - 落葉樹 [214906]

- 拘束
 - レベル: レベル1
 - ホスト: レベル:レベル1
 - オフセット: 0 mm
 - 付近の要素と...: いいえ
- 識別情報
 - タイプ名: ヤマボウシ-3.0メートル
 - イメージ



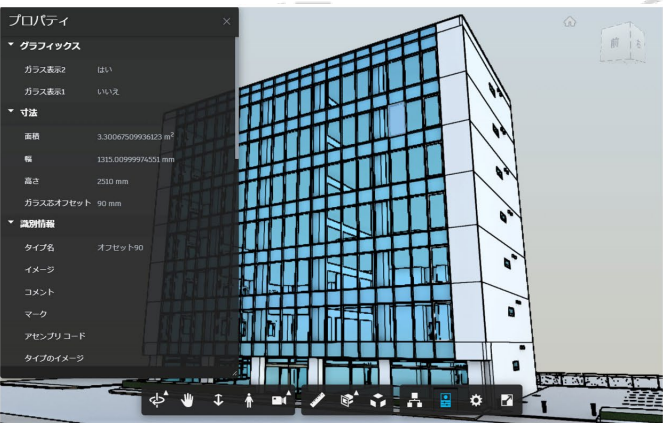
Data Management API



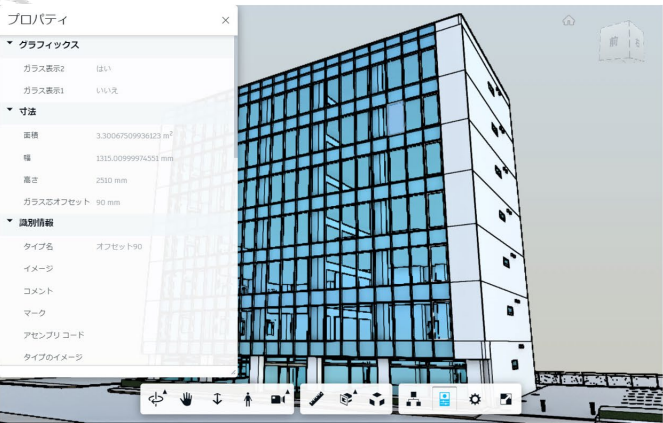
Viewer

V2.9 V3.1 V3.2 V3.3 V4.0 V5.0 V6.0 V6.1 V6.2 V6.3 V6.4 V6.5 V6.6 V7.0

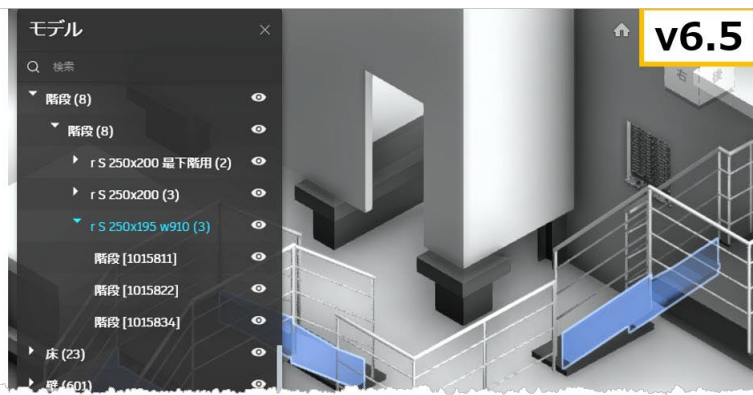
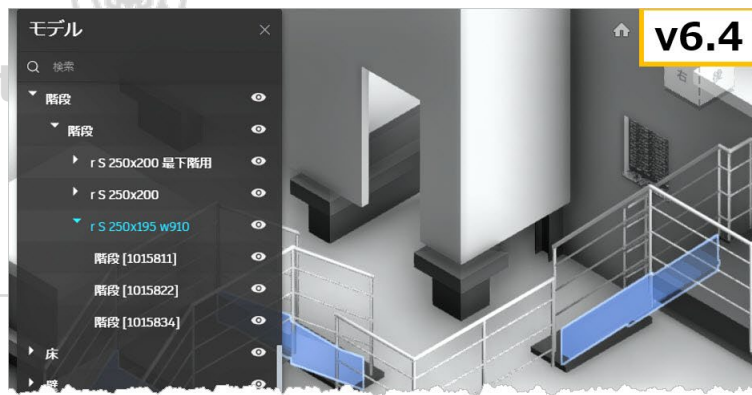
V4.2



ダークテーマ



ライトテーマ



ReCap Photo / (Beta)

Reality

表示する2D図面/3Dモデルの指定

- Model Derivative API での SVF/2 変換済ドキュメント ID
 - 指定前に Base64 エンコードが必須
 - <https://www.base64encode.org/>

例)

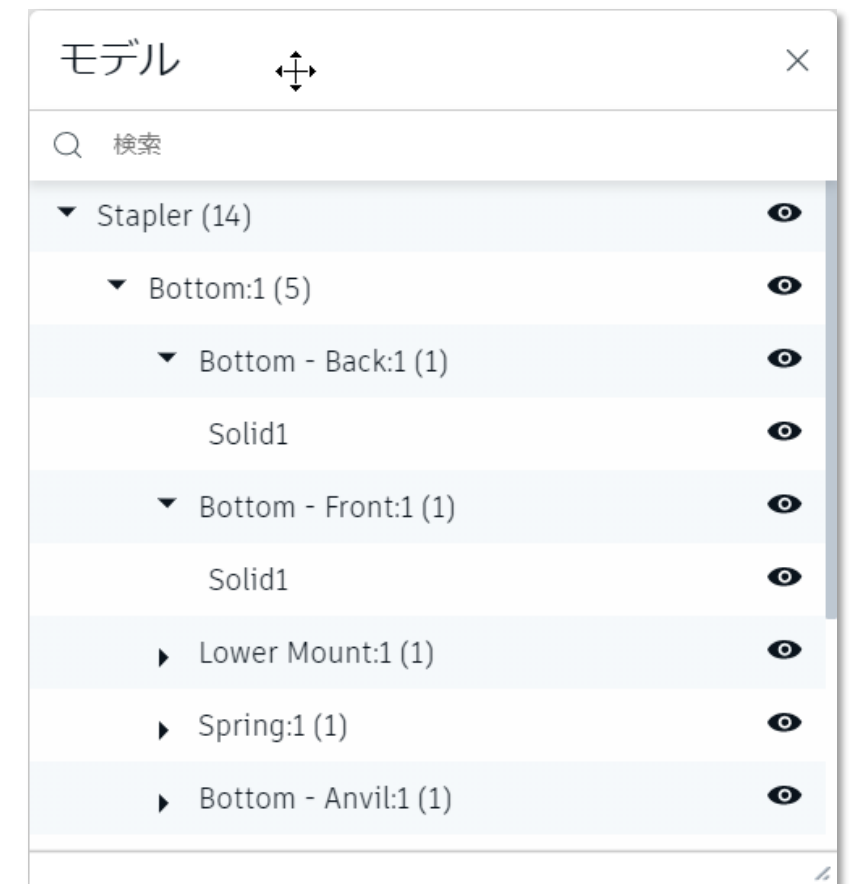
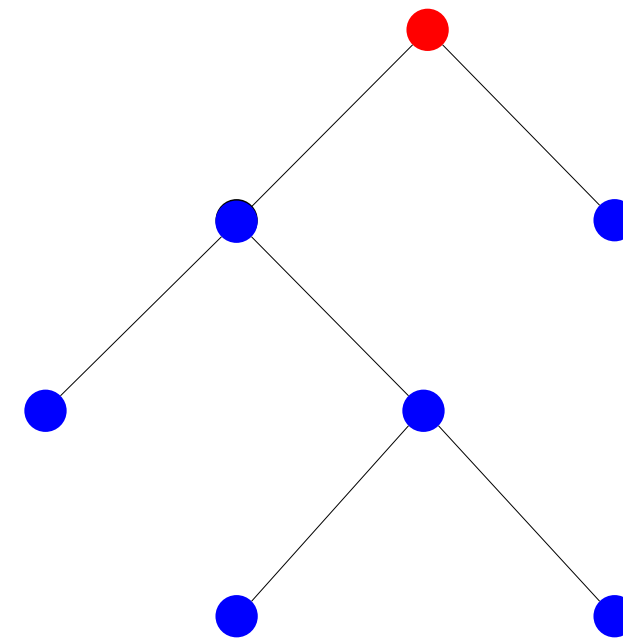
**urn:adsk.objects:os.object:fpd-japan-
avpg5fgrh5qbpo8hk15lszzg8drfknrouwmwd0p8lmse32p7ojoxz5p
zonuti-7g/Chair.f3d**

Base64 エンコード >>

**dXJuOmFkc2sub2JqZWNOczpvcy5vYmplY3Q6ZnBkLWphcGFuLWF
2cGc1ZmdyaDVxYnBvOGhrMTVsc3p6ZzhkcmZrbnJvdXdtd2QwcDh
sbXNIMzJwN29qb3h6NXB6b251dGktN2cvQ2hhaXIuZjNk**

APS Viewer 上の概念：モデル階層と識別

- オブジェクトは一意に識別される (dbId)
- オブジェクトはビュー毎にツリー構造内に発現
- ただし、モデル階層は CAD (シードファイル) によって異なる
 - Inventor モデル階層にはアセンブリを表す **ルートノード** とサブアセンブリやパーツを表す **子ノード** が存在
- 各ノードも一意に識別される
- Node Id = dbId
- dbId は変換毎に変化
- 常に不変は externalId



用語の整理

- **Name** : インスタンス識別が可能なオブジェクト名
 - 例 : Inventor のアセンブリ名やパーツ名
- **dbId** : モデル内の要素の一意的 ID (識別子)
 - ノードと dbId の間には1対1の対応があります。
- **fragId** : オブジェクト表示で内部的に使用するメッシュを参照する ID
 - 1つの dbId には複数の fragId を含めることが可能
- **Parent** : 子ノードのルートとなる親ノードの ID
- **Children** : ノード配下の子を表現する dbId 配列

VS Code 上の操作手順

1. ターミナルで Viewer 用 TypeScript 定義をインストール
2. **wwwroot** フォルダ下に **viewer.js** ファイルを作成して実装
3. **wwwroot** フォルダ下に **main.js** ファイルを作成して実装
4. **wwwroot** フォルダ下に **index.html** ファイルを作成して実装
5. **wwwroot** フォルダ下に **main.css** ファイルを作成して実装
6. デバッグ実行 >> localhost:8080

Viewer & UI | Autodesk Platform Servi

get-started.aps.autodesk.com/tutorials/simple-viewer/viewer

本人確認を行ってください

AUTODESK

Platform Services

Solutions

Getting Started

Documentation

Success Stories

Community

Support

Pricing

App Store

Getting Started

Environment Setup

Tutorials

Simple Viewer

Application Setup

Authentication

Data & Derivatives

Viewer & UI

Hubs Browser

Dashboard

Design Automation

ACC Administrator

Learn More

Tutorials

Simple Viewer

Viewer & UI

Viewer & UI

Finally, we're ready to build the client-side piece of our application.

TIP

If you're developing with Node.js, you can use TypeScript definitions for the Viewer API. Run

`npm install --dev @types/forge-viewer`

in your terminal to add the TypeScript definition file to your project.

Viewer logic

Application logic

User interface

Try it out

Viewer logic

Let's start by implementing the Viewer functionality for our application. Create a viewer.js file under the wwwroot subfolder with the following code:

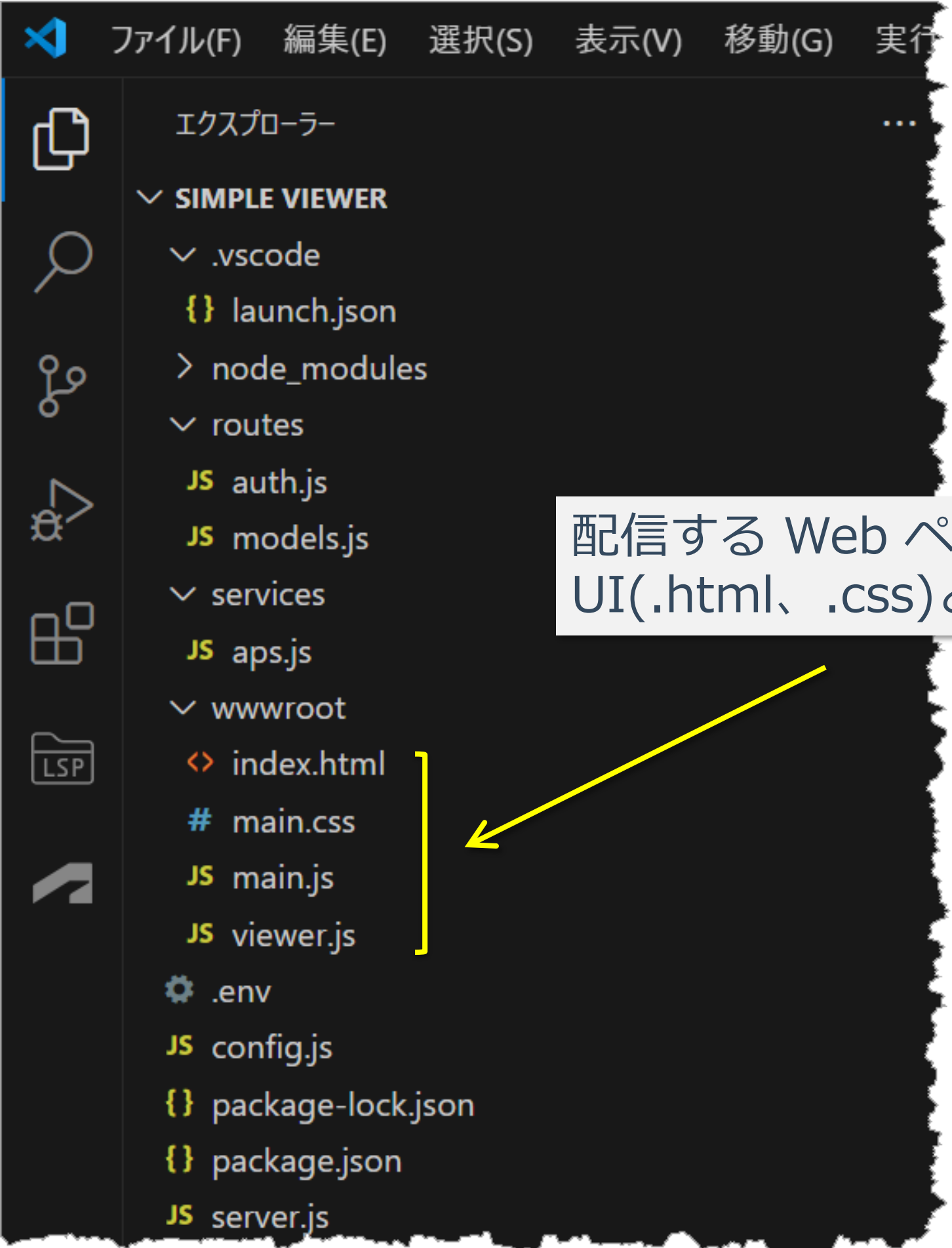
18°C
くもり

検索

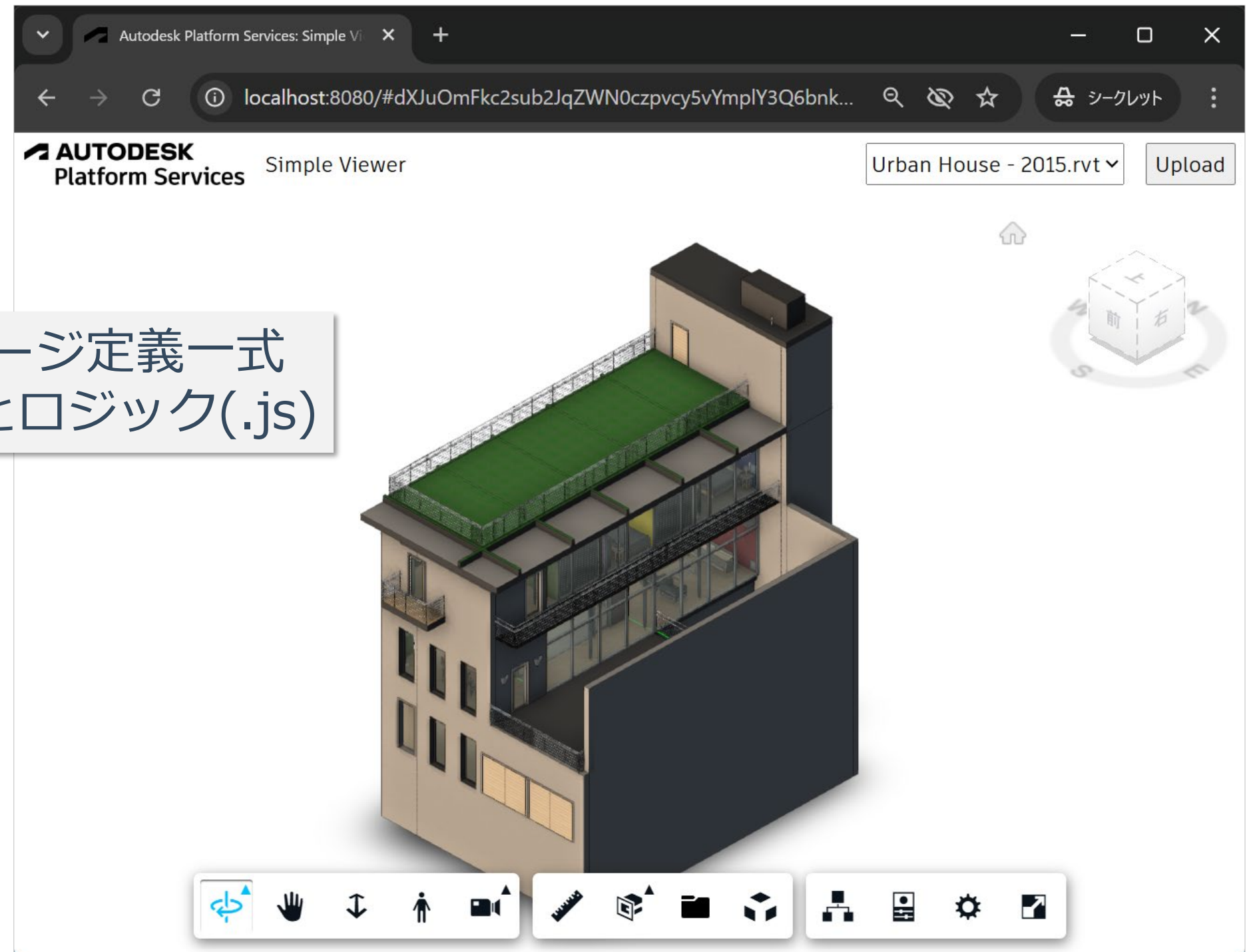
17:04

2025/04/11

ここまでのプロジェクト構成

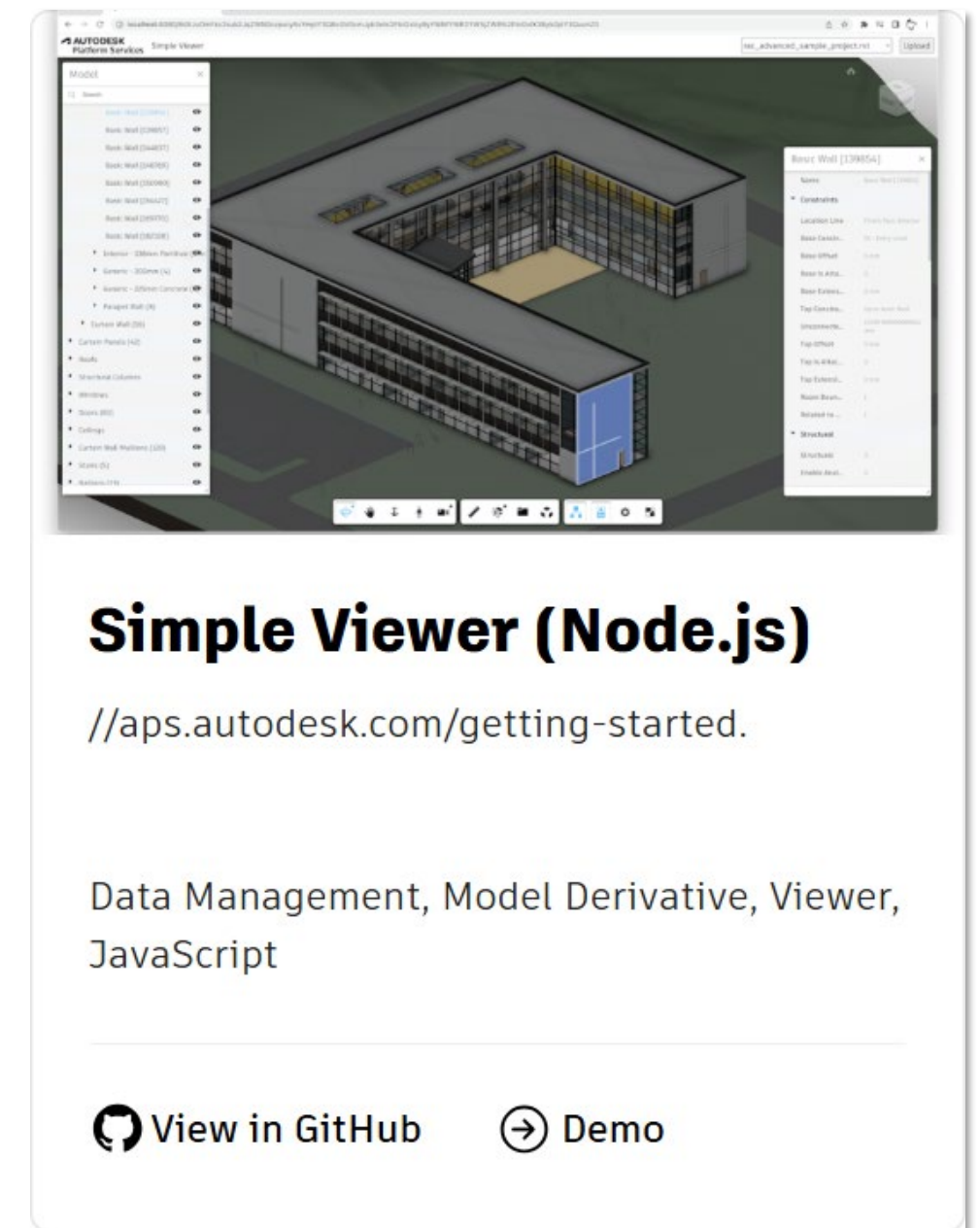


配信する Web ページ定義一式
UI(.html、.css)とロジック(.js)

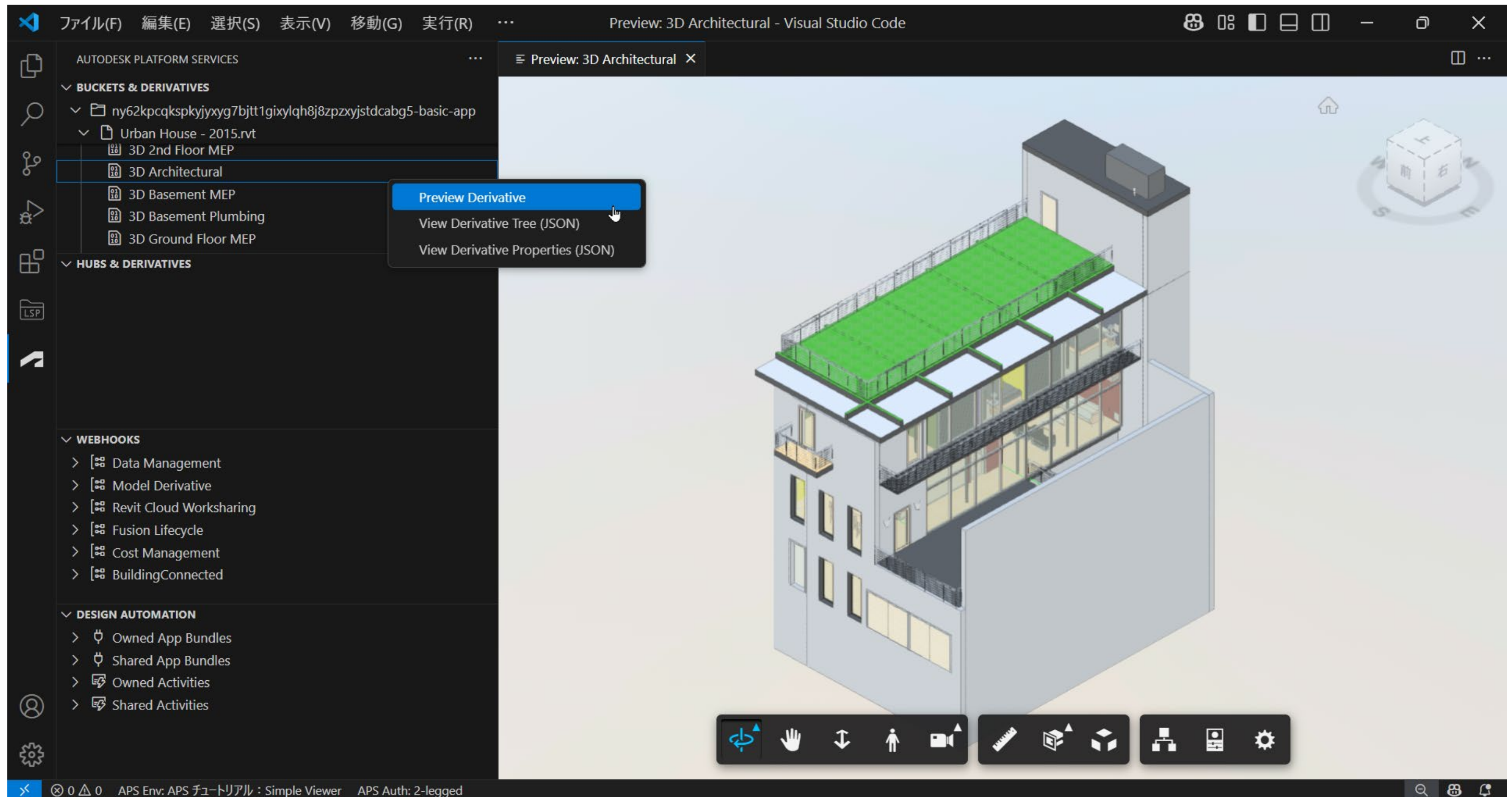


Simple Viewer GitHub リポジトリ

- Code Sample : Simple Viewer
 - Node.js ベース
 - APS SDK を使用
 - 2-legged 認証
 - Bucket の作成
 - Bucket へのアップロード
 - SVF2 変換
 - Viewer 表示



VS Code - APS エクステンションを利用した検証



2-legged 認証での Viewer 利用手順の理解

- Postman で手順検証
- RESTful API を理解
- 最終目的：
 - **URN** を取得
(ドキュメント ID)

forge.autodesk.com からアカウント登録とアプリの作成 (手動)



Access Token の取得 (RESTful API)



Bucket の作成 (RESTful API)



デザイン ファイルのアップロード (RESTful API)



変換リクエスト (RESTful API)

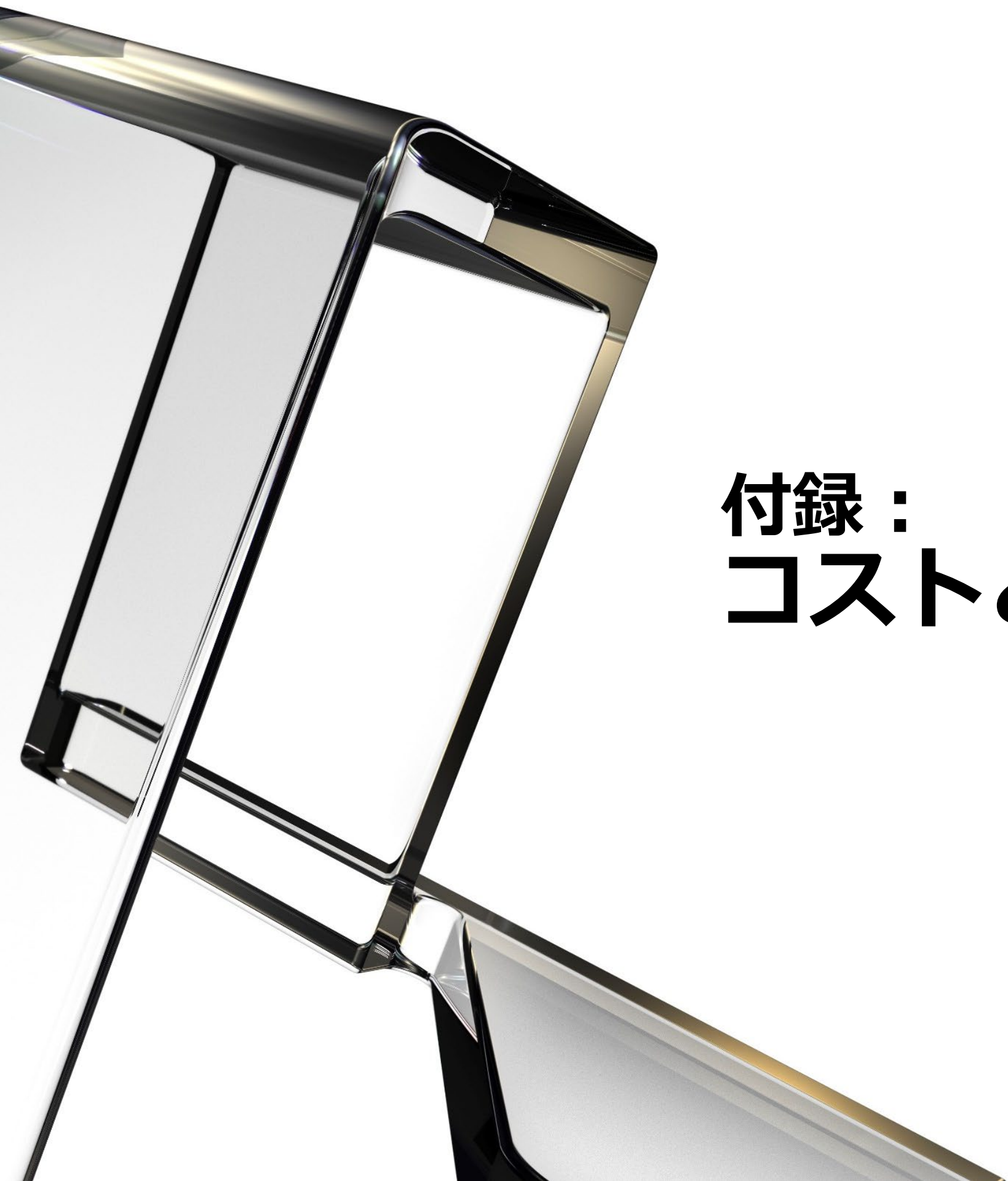


クライアントからのアクセス (RESTful API & JavaScript API)

ご注意：進化し続ける APS・開発作業に終わりはなし

クラウドサービス

- APS の機能向上によって動作/仕様が変わる可能性
 - 例) Design Automation API への OAuth スcope code:all 適用
 - APS Viewer は特定バージョン指定も可能
- クラウド開発にはメンテナンスが必須
- デスクトップ製品のアドイン開発とは異なります！
- 開発ベンダーとのサブスクリプション？契約が必要！！



付録： コストとサポート

APS 利用に対する '課金' とは？

<https://aps.autodesk.com/pricing>

- 特定の APS API 利用に **Autodesk Flex** による従量課金

プレミアムAPIとサービス

コスト 情報

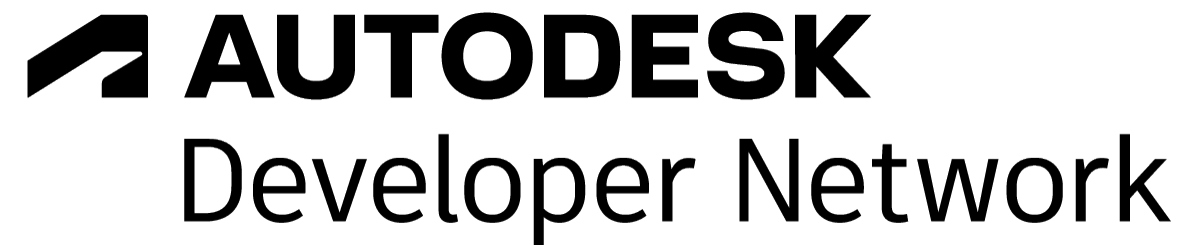
Design Automation API	2.0	Flex トークン / 処理時間
Flow Graph Engine API	1.0	Flex トークン / 処理時間
Model Derivative API	0.5 0.1	Flex トークン / コンプレックスジョブ Flex トークン / シンプルジョブ
Reality Capture API	1.0	50枚の写真毎のFlexトークン

- **Autodesk Flex とは？**

- <https://www.autodesk.com/jp/buying/flex>



Autodesk Developer Network (ADN) とは



- **年間契約の開発サポートプログラム（有償）です**
 - 新製品/バージョン、API、テクノロジーの啓蒙と流布
 - 無償ブログ公開、ウェビナー等イベント開催、その他
 - ADN 加入メンバーへの API 開発サポートの提供
 - API の問題点、不具合、要望の米国本社へのフィードバック
 - Autodesk App Store アプリの公開審査とサポート

ADN 特典

1. ADN Extranet（会員専用サイト）アクセスを提供
2. オートデスク API についてのお問合せ窓口提供
3. 開発用途の製品ライセンスを提供
4. マーケティング用ロゴの提供
5. ユーザー数に応じた Autodesk Flex の提供
6. 開発者向け有償イベントへの優待価格提供

今後のリソース

- サンプル コード : GitHub
 - <https://github.com/Developer-Autodesk>
 - <https://github.com/Autodesk-APS>
- サポート : StackOverflow
 - <https://APS.autodesk.com/en/support/get-help>
- ブログ : APS Community Blog
 - <https://APS.autodesk.com/blog>

