



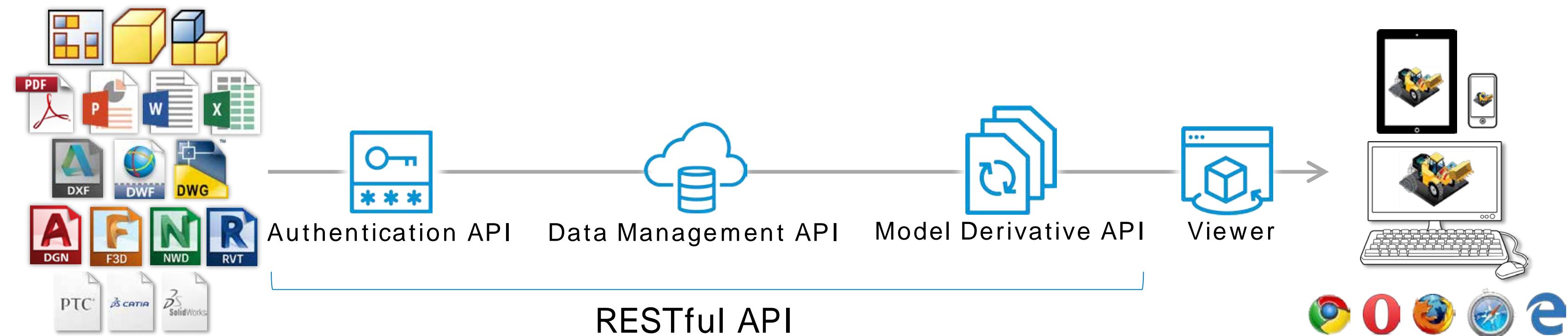
Forge Online

# OSS Bucket の利用： Postman でのアクセス トークンの取得

伊勢崎 俊明

オートデスク 株式会社

# API の視点で見た Forge Viewer ソリューション



まず、デベロッパキーを取得して  
次に、アクセス トークンを取得

# デベロッパキーとアクセストークン

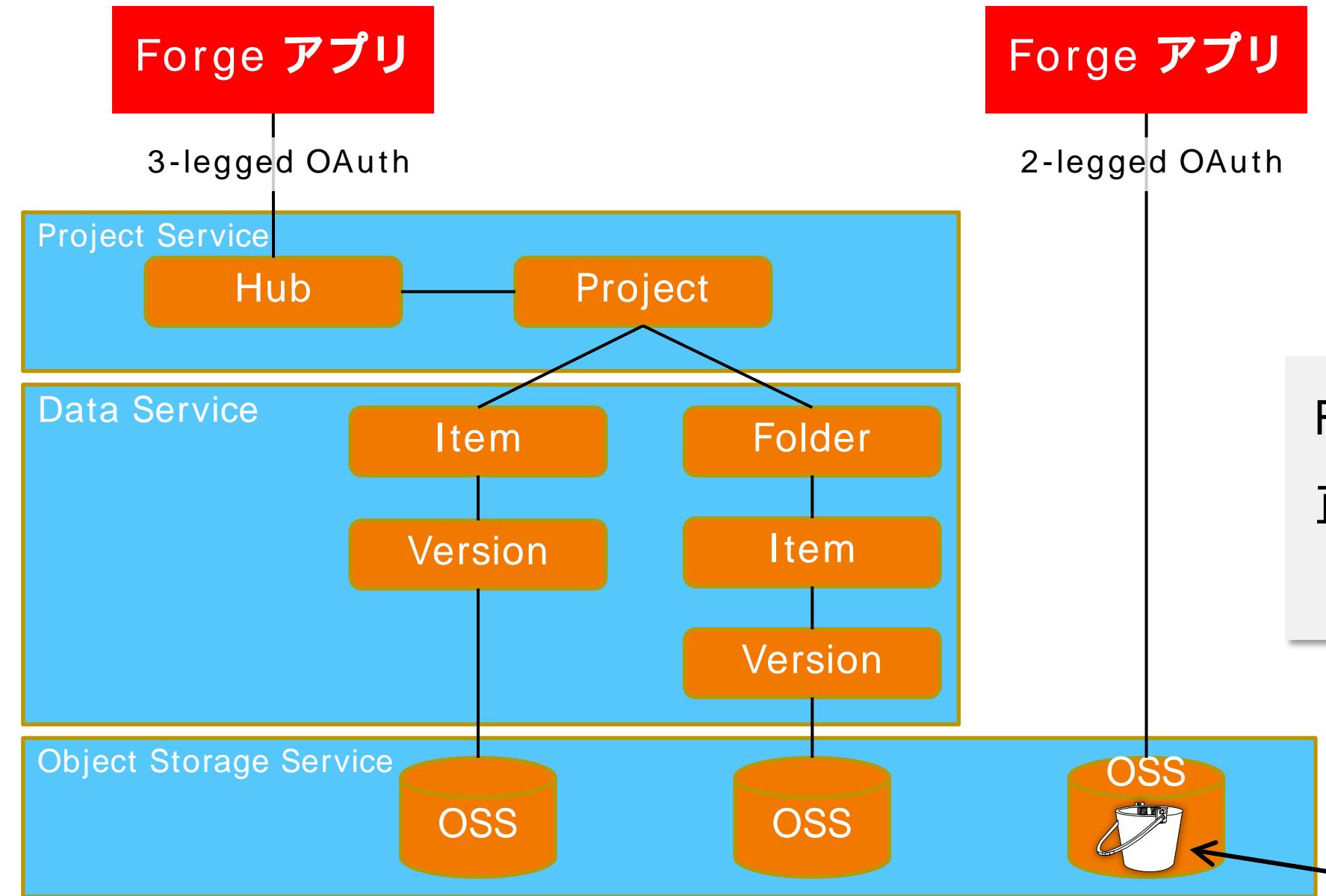
## Forge ポータル

([forge.autodesk.com](https://forge.autodesk.com))



# シナリオ / 目的に応じた認証

Forge アプリが A360 の  
ユーザ データ領域に直接  
アクセスして運用する方法  
- 3-Legged OAuth -



Forge アプリがデータを  
直接管理・運用する方法  
- 2-Legged OAuth -

Bucket

# Access Token（アクセス トークン）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

§ Web クライアントからの呼出は CORS エラーになる（仕様）

§ 生成に必要なもの

§ デベロッパキー

§ Client ID と Client Secret のペア

§ Forge ポータルで App（アプリ）作成時に取得可能

§ 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	プロファイル (Autodesk ID) の表示	
user:read	プロファイル (Autodesk ID) の読み取り	
user:write	プロファイル (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)	2-leggedで利用
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 Endpoint 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	<b>GET</b> <a href="https://developer.api.autodesk.com/oss/v2/buckets">https://developer.api.autodesk.com/oss/v2/buckets</a>
Authentication Context	app-only
Required OAuth Scopes	<b>bucket :read</b>
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	<b>POST</b> <a href="https://developer.api.autodesk.com/oss/v2/buckets">https://developer.api.autodesk.com/oss/v2/buckets</a>
Authentication Context	app-only
Required OAuth Scopes	<b>bucket :create</b>
Data Format	JSON

bucket:create

# 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

The screenshot shows the official website for Postman. At the top, there's a navigation bar with links for PRODUCT, SOLUTIONS, PLANS & PRICING, DOCS, API NETWORK, and a red SIGN IN button. To the left, there's a logo icon of a pen inside a circle. The main headline reads "Postman Makes API Development Simple." Below it, a subtext says "Developers use Postman to build modern software for the API-first world." There are two calls-to-action: a red "Download the App" button and a link "Want to improve your Postman skills? Take the next step!". The background features a dark space-themed illustration with a rocket ship launching, a satellite, and a planet with a ring.

# 2-legged OAuth での Viewer 利用手順の理解

## § API リファレンスの見方

§ Forge ポータル [forge.autodesk.com](https://forge.autodesk.com) で Documents を選択

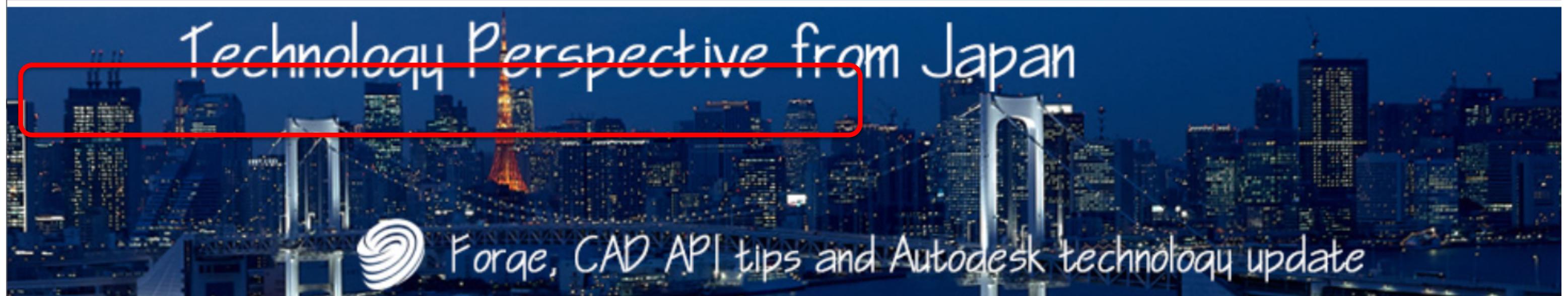
§ API Reference で必要な endpoint をチェック

The image shows two screenshots of the Autodesk Forge Documentation website.

**Left Screenshot:** Shows the main documentation page. A large blue header says "FORGE DOCUMENTATION". Below it, under "APIs & Services", there are two main sections: "Authentication" and "BIM 360". Each section has a "Developer's guide" and an "API reference" link. A circled "1" with an arrow points to the "Documentation" menu item at the top of the page.

**Right Screenshot:** Shows the "Authentication (OAuth) v2" API Reference page. It includes sections for "Developer's Guide", "Step-by-Step Tutorials", and "API Reference". Under "API Reference", there is a "HTTP Specification" section with "Two-Legged Context" and "Three-Legged Context" subsections. A circled "2" with an arrow points to the "API reference" link in the Authentication section of the left screenshot. A circled "3" with an arrow points to the "Two-Legged Context" link in the right screenshot.

# Postman コレクションの利用



ホーム アーカイブ オートデスク ADN オープン ADN よくある質問 Forge ポータル

« AutoCAD 360 外部ストレージ接続のサポート停止について | メイン | 新しい Forge 利用料 - 米国、ヨーロッパのみ »

2016/09/28

## Postman による Viewer 利用手順の理解 - 2 legged 認証



<本記事は2019年12月10日に Forge Viewer v7 にあわせて一部改定しています。>

先日の [ブログ記事](#) で Postman による RESTful API のテストについてご紹介しました。今回は、Forge Viewer で 3D モデルを表示するまでの処理を、Postman を使ってトレースしてみたいと思います。ここでは、最もシンプルな **2-legged 認証** を利用して、任意に作成した Bucket にファイルをアップロード、変換して Viewer

2020年5月

日	月	火	水	木	金	土
			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					

最近の記事

参考ブログ記事：[http://adndevblog.typepad.com/technology\\_perspective/2016/09/understanding-steps-to-use-viewer-on-postman.html](http://adndevblog.typepad.com/technology_perspective/2016/09/understanding-steps-to-use-viewer-on-postman.html)



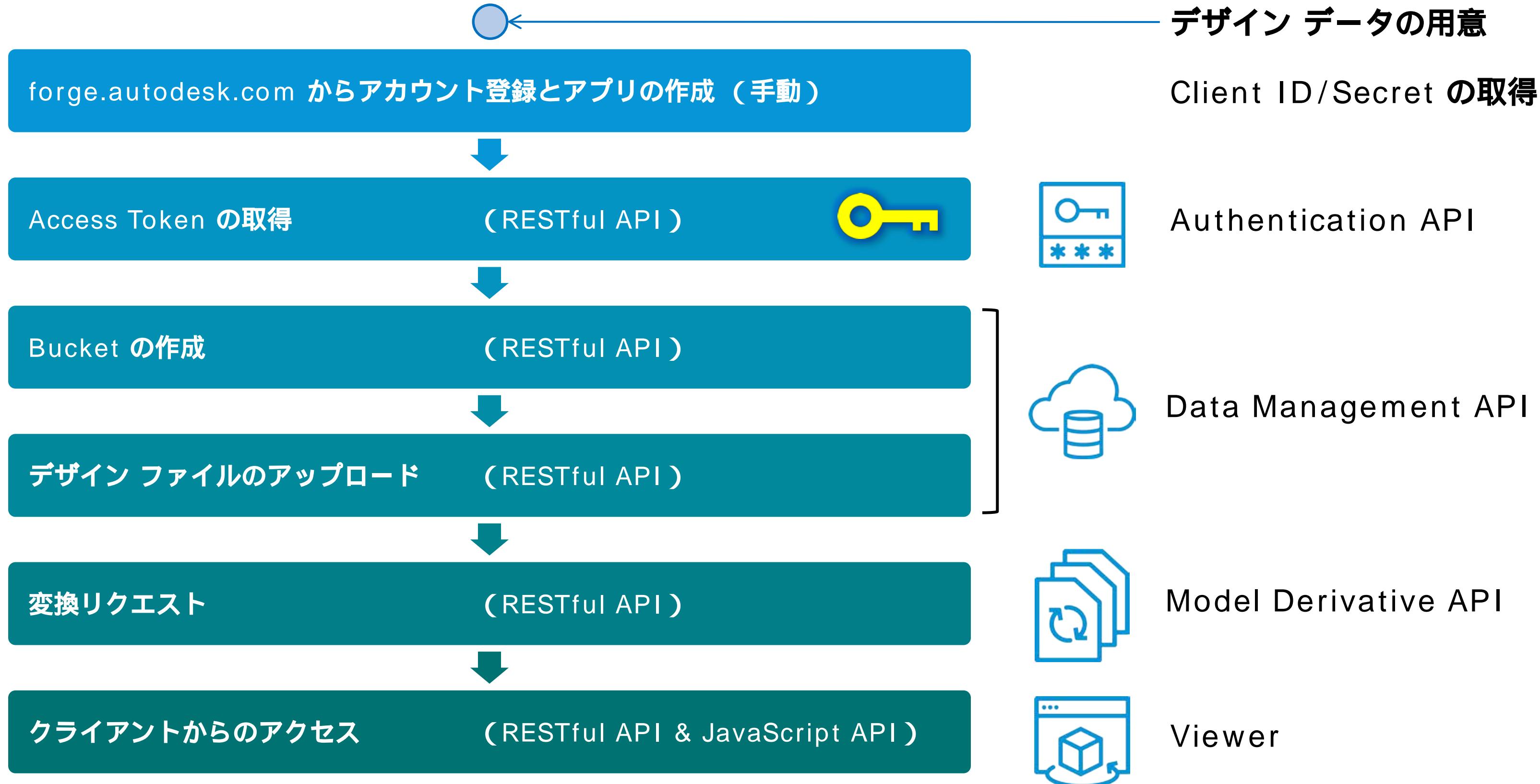
Forge Online

# OSS Bucket の利用： OSS Bucket の利用と SVF 変換

伊勢崎 俊明

オートデスク 株式会社

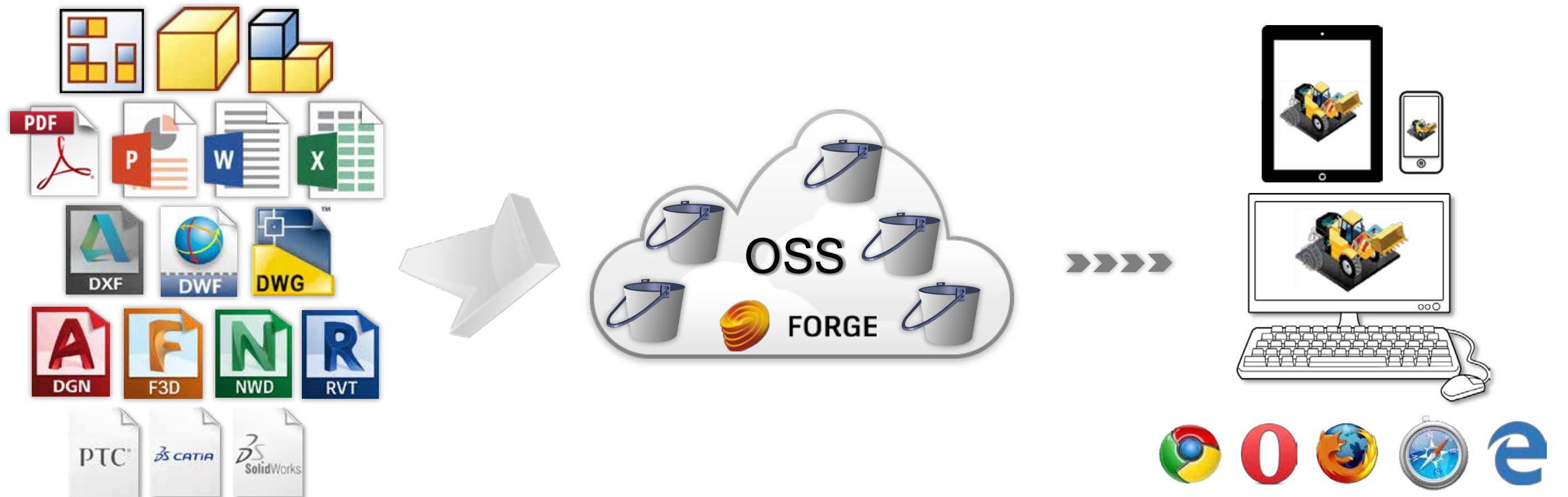
# 2-legged OAuth



# Bucket とは

- § Bucket は Forge 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(- \_ .)

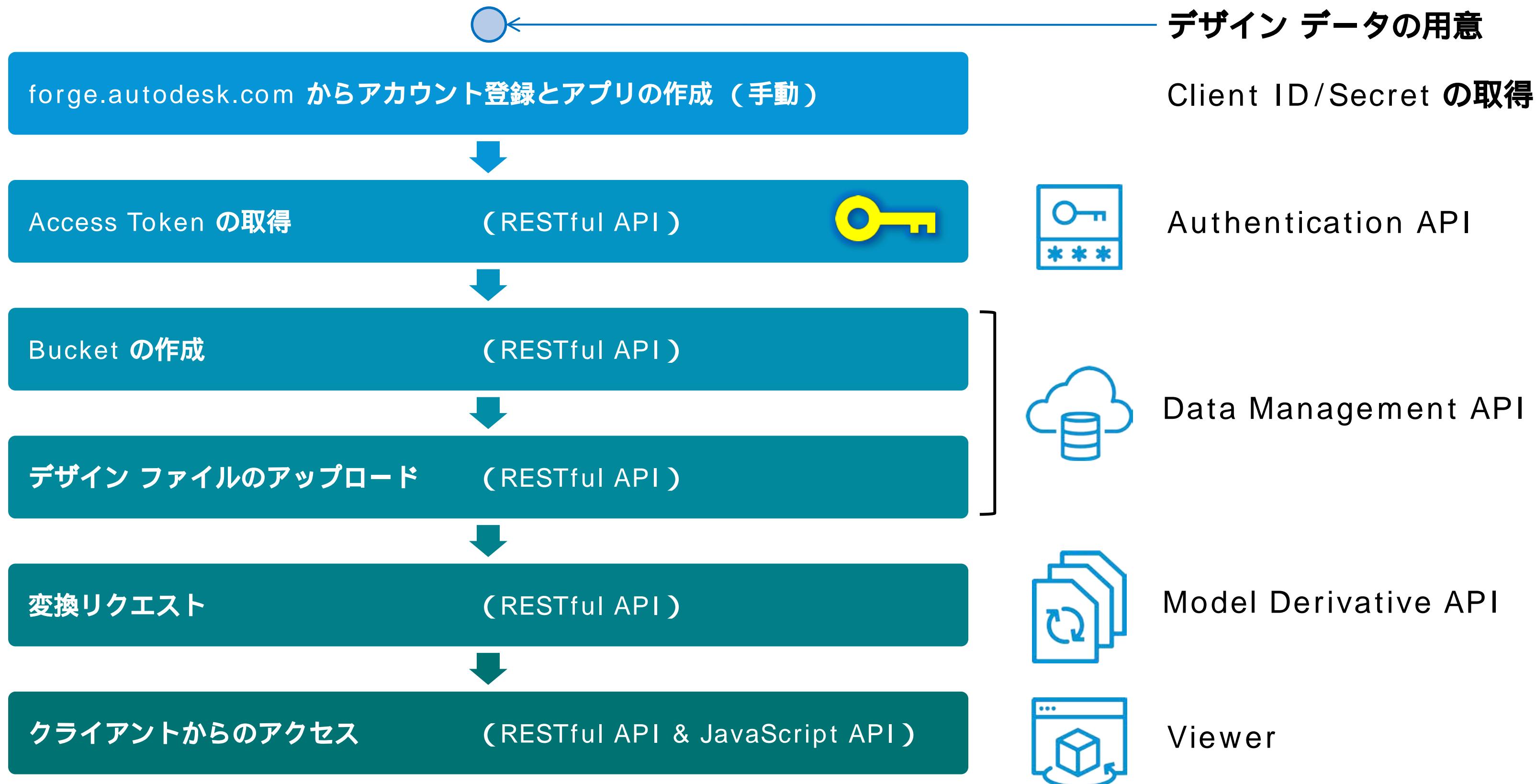
Client ID 利用を推奨



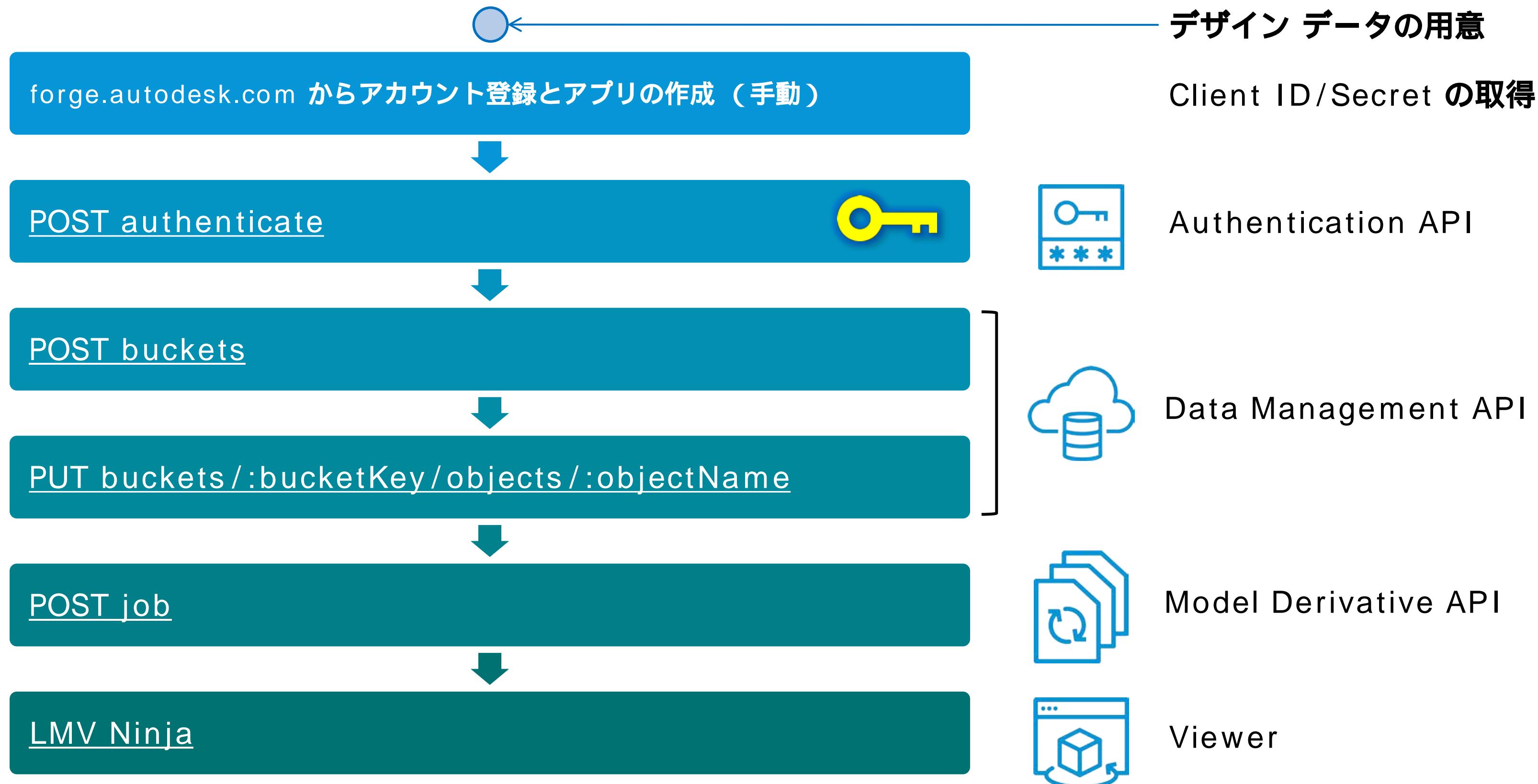
# Bucket ポリシー

- § 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される
- § Bucket 作成後のポリシー変更は不可
- § Bucket への永続的なアクセス条件
  - § 作成時と同じ Client ID を使用する必要あり
  - § 異なる Client ID を持つアプリはアクセス不可

# Viewer 表示までのステップ



# Viewer 表示までのステップと endpoint



# Postman コレクションの利用



## 3. バケットの作成

2-Legged 認証では OSS 上に一意な名前で Bucket を作成して、ファイルをアップロード、変換を経て、Viewer への表示を準備します。Bucket の作成とデザインファイルのアップロードには、**Data Management API** を使用します。Bucket 作成時には、名称とともに Bucket Policy を指定する必要があります。詳細は、過去のブログ記事 [Bucket に関するサマリー](#) をご確認ください。

Bucket の作成に使用する endpoint は <https://developer.api.autodesk.com/oss/v2/buckets>、メソッドは **POST** となります。

既に登録されている Bucket について、特定の Bucket が存在するかを知るために、endpoint <https://developer.api.autodesk.com/oss/v2/buckets/:bucketKey/details>、メソッド **GET** を利用することができます。上記 URI 中の `:bucketKey` 部は照会したい Bucket 名で置き換えてください。この endpoint は、Bucket 内のデザイン ファイルの列挙にも利用することが可能です。

### 3. バケットの作成.postman\_collectionをダウンロード

ダウンロードした Postman コレクションを使って Bucket を作成したり、有無をチェックしたりする方法を次の動画で確認することができます。

# Bucket 作成で使用する endpoint

The screenshot shows the Autodesk Forge Data Management API v2 documentation. On the left, there's a sidebar with links like 'Developer's Guide', 'Step-by-Step Tutorials', 'API Reference' (which is expanded), 'HTTP Specification', 'Data Management', 'Commands', 'OSS' (expanded), 'Buckets' (under OSS), 'POST buckets' (under Buckets), 'GET buckets', 'GET buckets/:bucketKey/details', 'DELETE buckets/:bucketKey', and 'Objects'. The main content area has a title 'POST buckets' with a description: '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.' Below this is a section titled 'Resource Information' with tables for 'Method and URI', 'Authentication Context', 'Required OAuth Scopes', and 'Data Format'. A red arrow points to the 'JSON' entry under 'Data Format'.

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 作成 : POST buckets のリクエスト ボディ

## § JSON 指定

```
{  
  "bucketKey": "<Your Bucket Name>",  
  "policyKey": "<Your Bucket Policy>"  
}
```

The screenshot shows the Postman interface for creating a bucket. The top bar includes a 'Create a bucket' button, a cursor icon, and 'Comments 0' and 'Examples 0' buttons. The main area shows a POST request to <https://developer.api.autodesk.com/oss/v2/buckets>. The 'Body' tab is selected, showing the JSON payload:

```
1 {  
2   "bucketKey": "<Your Bucket Name>",  
3   "policyKey": "<Your Bucket Policy>"  
4 }
```

Below the body, there are tabs for 'Params', 'Authorization', 'Headers (12)', 'Pre-request Script', 'Tests', and 'Settings'. A 'JSON' dropdown is also visible. On the right side, there is a large orange pencil icon with the word 'Cool' written above it.

# ファイルアップロードで使用する endpoint

The screenshot shows the Autodesk Forge Data Management API v2 documentation. On the left, there's a sidebar with links like 'Developer's Guide', 'Step-by-Step Tutorials', and 'API Reference' (which is expanded). Under 'API Reference', there are sections for 'HTTP Specification' (with 'Data Management' and 'Commands'), 'OSS' (with 'Buckets' and 'Objects'), and a detailed view of the 'PUT buckets/:bucketKey/objects/:objectName' endpoint.

**PUT buckets/:bucketKey/objects/:objectName**

Upload an object. If the specified object name already exists in the bucket, the uploaded content will overwrite the existing content for the bucket name/object name combination.

Note that for objects larger than 100 MB we recommend using the [PUT buckets/:bucketKey/objects/:objectName/resumable](#) endpoint.

## Resource Information

Method and URI	PUT <a href="https://developer.api.autodesk.com/oss/v2/buckets/:bucketKey/objects/:objectName">https://developer.api.autodesk.com/oss/v2/buckets/:bucketKey/objects/:objectName</a>
Authentication Context	app-only
Required OAuth Scopes	<code>data:write</code> or <code>data:create</code> ( <code>data:write</code> allows overwriting exist
Data Format	JSON

# ご参考：Params タブでの URI パラメータ指定

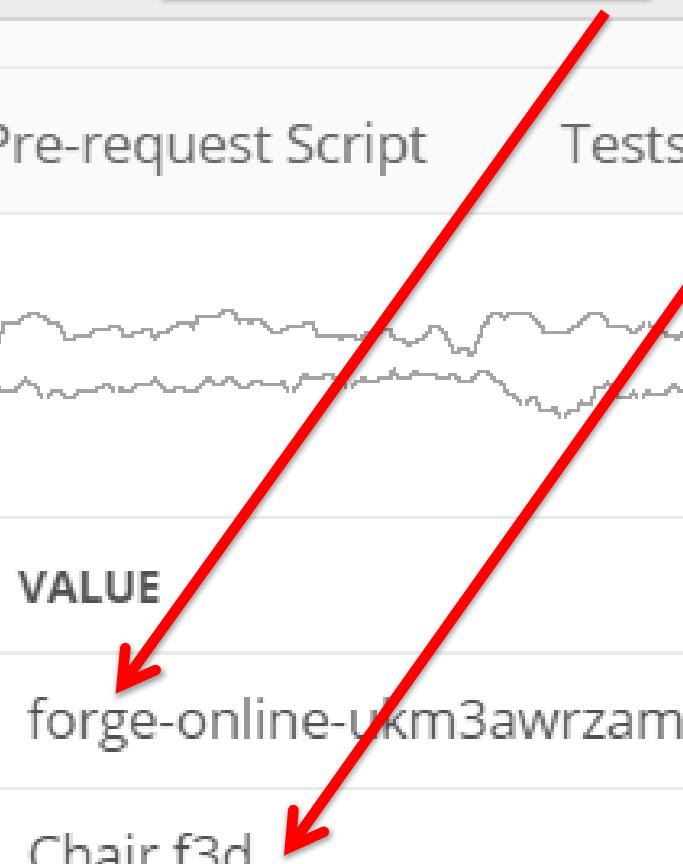
PUT https://developer.api.autodesk.com/oss/v2/buckets/:Your Bucket Name/objects/:Your Design File

Params ● Authorization Headers (12) Body ● Pre-request Script Tests ● Settings

Query Params

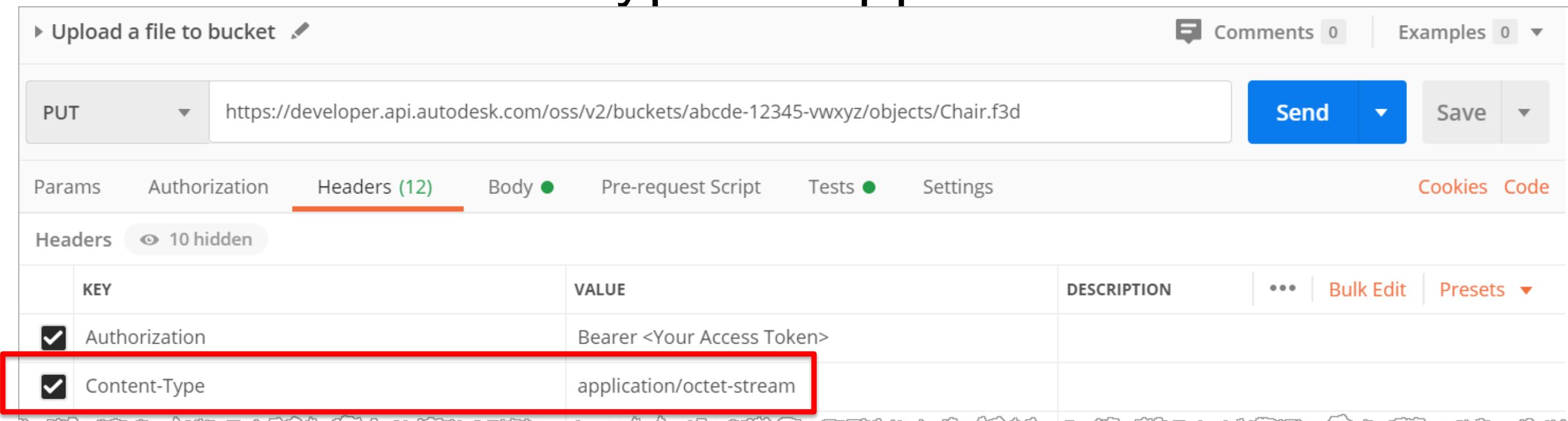
Path Variables

KEY	VALUE
Your Bucket Name	forge-online-ukm3awrzampanfubjvy2rngsgxln
Your Design File	Chair.f3d



# PUT buckets/:bucketKey/objects/:objectName

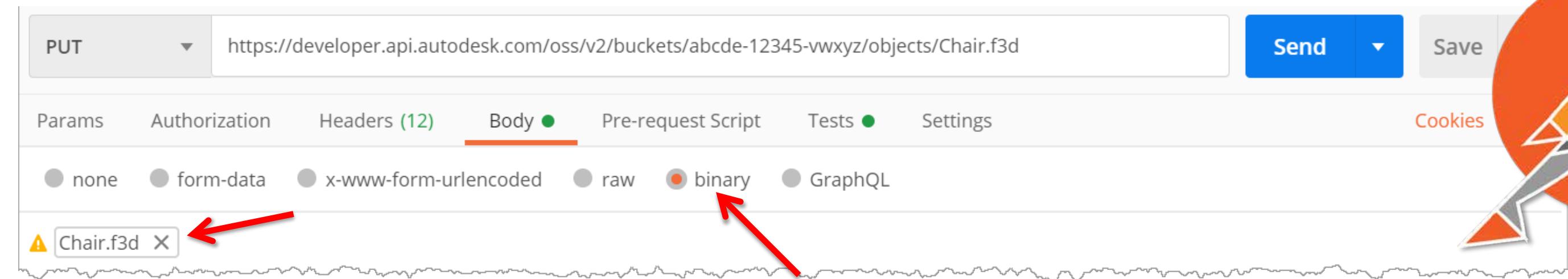
## § ヘッダー : Content-Type は application/octet-stream



The screenshot shows the Postman interface for a PUT request. The URL is `https://developer.api.autodesk.com/oss/v2/buckets/abcde-12345-vwxyz/objects/Chair.f3d`. The 'Headers' tab is selected, showing two entries: 'Authorization' with value 'Bearer <Your Access Token>' and 'Content-Type' with value 'application/octet-stream'. Both entries have a red border around them.

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
Authorization	Bearer <Your Access Token>				
Content-Type	application/octet-stream				

## § ボディ : Binary でアップロードするファイルを指定



The screenshot shows the Postman interface for a PUT request. The URL is `https://developer.api.autodesk.com/oss/v2/buckets/abcde-12345-vwxyz/objects/Chair.f3d`. The 'Body' tab is selected, showing a file input field containing 'Chair.f3d'. A large orange arrow points to this file input field. Below the input field are several radio button options: 'none', 'form-data', 'x-www-form-urlencoded', 'raw', 'binary' (which is selected), and 'GraphQL'.



# 変換処理を開始する endpoint

The screenshot shows the Autodesk Forge Model Derivative API v2 documentation. On the left, there's a sidebar with links like 'Developer's Guide', 'Step-by-Step Tutorials', 'Code Samples', 'API Reference' (which is expanded), and 'HTTP Specification'. Under 'HTTP Specification', 'Derivatives' is expanded, showing 'GET formats', 'POST job' (which is highlighted in blue), 'POST references', 'GET :urn/thumbnaill', 'GET :urn/manifest', and 'DELETE :urn/manifest'. The main content area has a title 'Derivatives' and a large button labeled 'POST job'. Below it, a list of uses includes translating designs, extracting geometries, and generating thumbnails. It also says to follow step-by-step tutorials for instructions. A note at the bottom says the endpoint is asynchronous and provides a link to check for completion.

AUTODESK<sup>®</sup>  
FORGE

Success Stories   Solutions ▾   Getting Started   Documentation   Community ▾   Support ▾   Pricing     **SIGN IN**

Model Derivative API v2

> Developer's Guide

> Step-by-Step Tutorials

> Code Samples

▼ API Reference

▼ HTTP Specification

- Derivatives
  - GET formats
  - POST job**
  - POST references
  - GET :urn/thumbnaill
  - GET :urn/manifest
  - DELETE :urn/manifest

Derivatives

**POST job**

You can use this endpoint to do the following:

- Translate a design from one format to another format.
- Extract selected parts of a design and export the set of geometries in OBJ format.
- Generate different-sized thumbnails.

Follow the [step-by-step tutorials](#) for detailed instructions on how to use this endpoint to translate files, extract geometries, and generate thumbnails.

Derivatives are stored in a manifest that is updated each time this endpoint is used on a source file.

Note that this endpoint is asynchronous and initiates a process that runs in the background. You need not keep the HTTP connection open until completion. Use the [GET :urn/manifest](#) endpoint to check for the job's completion status.

# 変換処理を開始：POST job のリクエスト ボディ

## § JSON 指定

```
{  
  "input": {  
    "urn": "<Your Encoded URN>"  
  },  
  "output": {  
    "formats": [  
      {"  
        "type": "svf",  
        "views": [  
          "2d",  
          "3d"  
        ]  
      }  
    ]  
  }  
}
```

Base64 エンコードした  
ドキュメント ID が必須

<https://www.base64encode.org/>



# 2-legged OAuth 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)

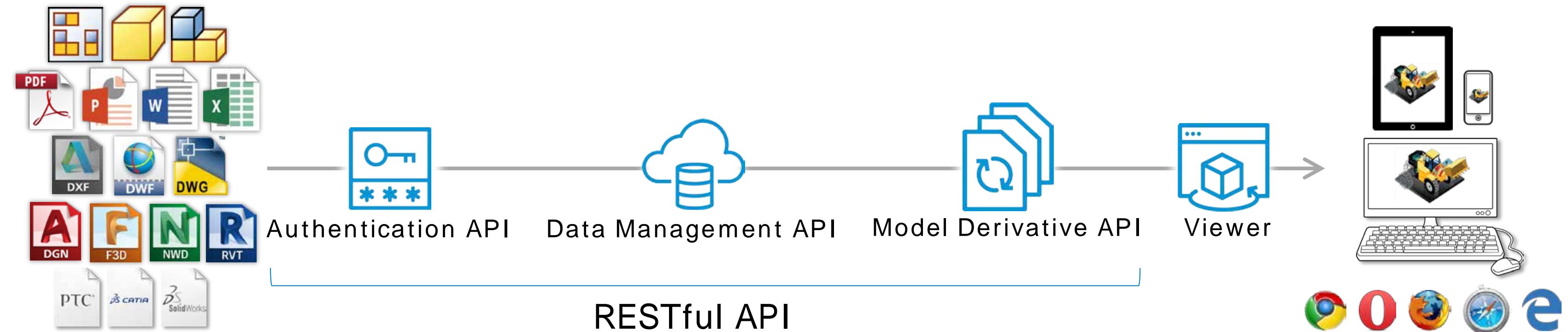
Bucket の作成 (RESTful API)

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)

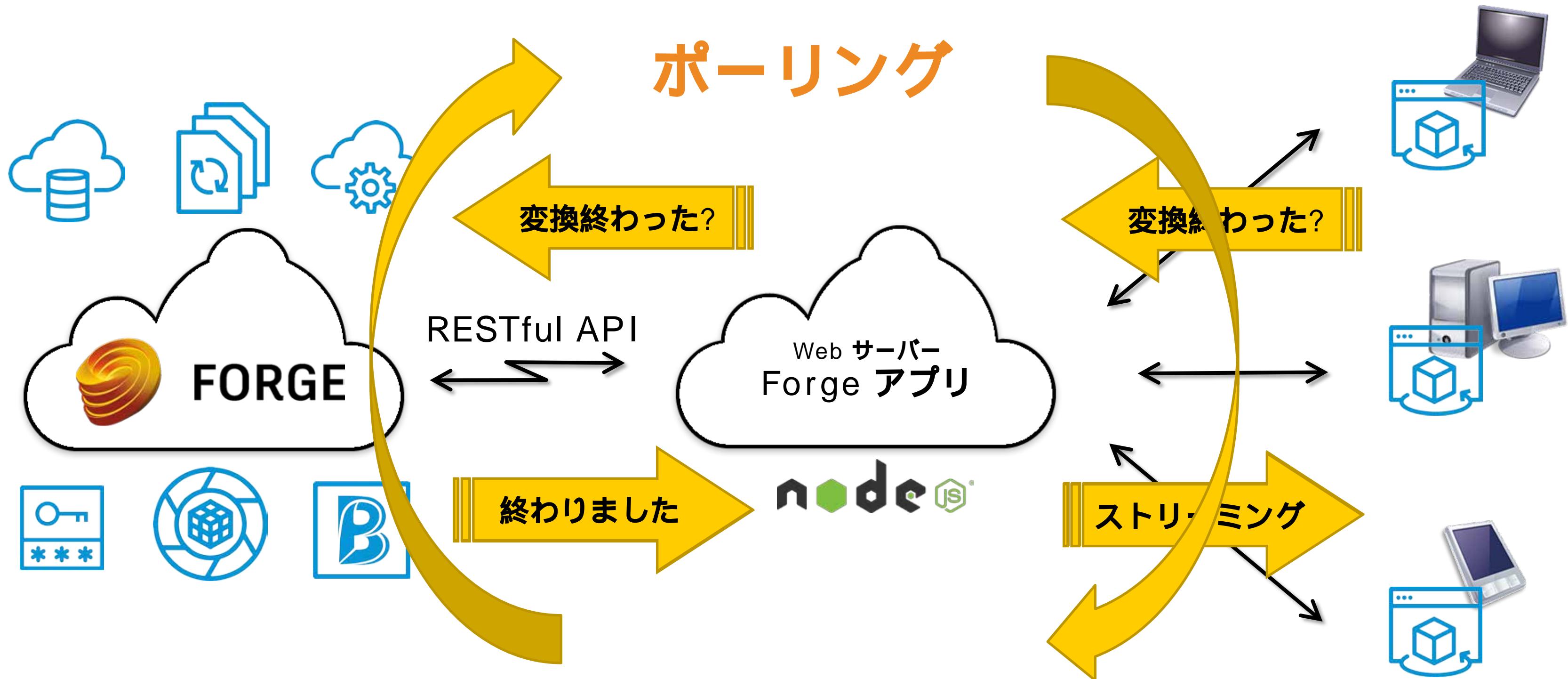
クライアントからのアクセス (RESTful API & JavaScript API)

# API の視点で見た Forge Viewer ソリューション



Model Derivative API 変換はすぐには終わらない

# 変換終了の検出





## Model Derivative API v2

> [Developer's Guide](#)

> [Step-by-Step Tutorials](#)

> [Code Samples](#)

▽ [API Reference](#)

▽ [HTTP Specification](#)

▽ [Derivatives](#)

[GET formats](#)

[POST job](#)

[POST references](#)

[GET :urn/thumbnail](#)

[GET :urn/manifest](#)

[DELETE :urn/manifest](#)

[GET :urn/manifest/:derivativeurn](#)

[HEAD :urn/manifest/:derivativeurn](#)

[GET :urn/metadata](#)

[GET :urn/metadata/:guid](#)

### Derivatives

#### **GET :urn/manifest**

Returns information about derivatives that correspond to a specific source file, including derivative URNs and statuses.

The URNs of the derivatives are used to download the generated derivatives when calling the [GET :urn/manifest/:derivativeurn](#) endpoint.

The statuses are used to verify whether the translation of requested output files is complete.

Note that different output files might complete their translation processes at different times, and therefore may have different `status` values.

When translating a source file a second time, the previously created manifest is not deleted; it appends the information (only new translations) to the manifest.

### Resource Information

---

**Method and URI** **GET** <https://developer.api.autodesk.com/modelderivative/v2/designdata/:urn/> (US)

---

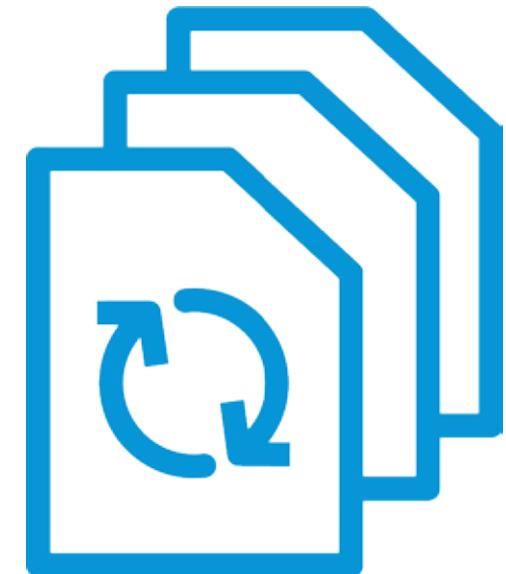
**Method and URI** **GET** <https://developer.api.autodesk.com/modelderivative/v2/regions/eu/designdata/:urn/> (EMEA)

# ポーリングで使用する endpoint のレスポンス ボディ

```
{  
  "type": "manifest",  
  "hasThumbnail": "true",  
  "status": "success",  
  "progress": "complete",  
  "region": "US",  
  "urn":  
    "dxJuOmFkc2sub2JqZWN0czpvcy5vYmplY3Q6ZnBkLWphcGFuLWF2cGc1ZmdyaDVxYnBv  
    OGhrMTVsc3p6ZzhkcmZrbnJvdXdt2QwcDhsbxNlMzJwN29qb3h6NXB6b251dGktN2cvQ  
    OFNMi5mM2Q",  
  "version": "1.0",  
  "derivatives": [  
    {
```

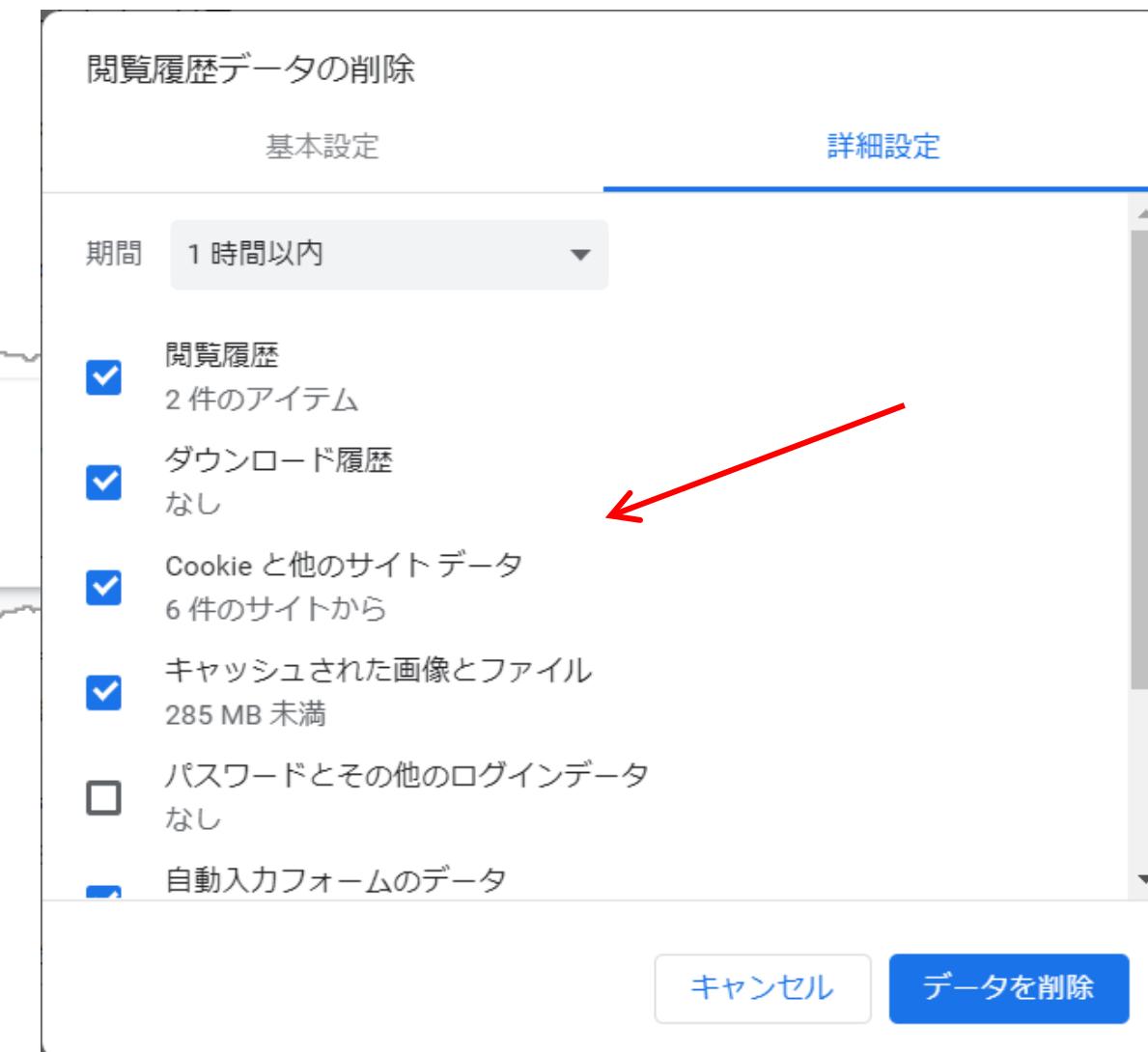
# 変換済ファイルの再変換について

- § 每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（確実）
    - § DELETE :urn / manifest endpoint
  2. 変換ジョブで再変換を指定
    - § POST job endpoint 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!!

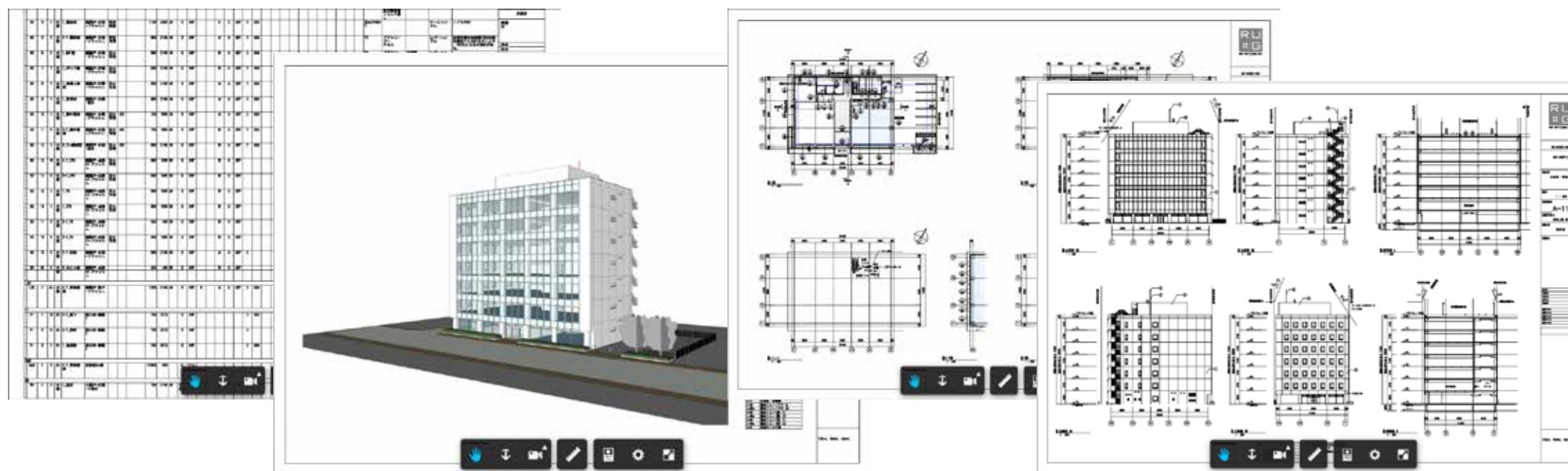


# Forge Viewer カスタマイズでの注意点

- § 配信された SVF はブラウザにキャッシュされます
  - § CAD モデルの変更が Model Derivative 変換後も反映されない
  - § 同一ファイル名/URNでの表示試行錯誤で陥りがち
  - § ブラウザ キャッシュのクリアを推奨



# 2D シート一覧の取得



```
function onDocumentLoadSuccess(viewerDocument) {
    var viewables = viewerDocument.getRoot().search({
        'type': 'geometry',
        'role': '2d'
    });

    var index = 0;
    viewables.forEach(function (value) {
        $('#sheets').append($('').val(index).text(value.data.name));
        index = index + 1;
    });
}
```

The screenshot shows a dropdown menu titled 'A-9 - 配置図' (Configuration Drawing) which is currently selected. Other options in the list include:

- A-7 - 仕上表 A1
- A-10 - 平面図
- A-41 - 法規チェック図
- A-0 - 表紙 図面リスト**
- A-32 - 建具リスト(ドア) A1
- A-33 - 建具リスト(窓) A1
- A-8 - 仕上表 A2
- A-32a - 建具リスト(ドア) A2
- A-33a - 建具リスト(窓) A2
- A-34 - 建具リスト(CW) A1
- A-34a - 建具リスト(CW) A2
- A-11 - 立面図・断面図
- A-9 - 配置図



Forge Online

# OSS Bucket の利用： Forge SDK を使った Viewer 実装

伊勢崎 俊明

オートデスク 株式会社

# Forge サンプル

## § Forge ポータル下部 Code Samples からのアクセス

The screenshot shows the Autodesk Forge website's footer navigation bar. The bar is divided into five main sections: FOLLOW FORGE, SOLUTIONS, DOCUMENTATION, RESOURCES, and ABOUT. The RESOURCES section contains links such as Get Help, API Status, Blog, FAQ, and Code Samples. A red arrow points to the 'Code Samples' link. The footer also includes links for Privacy/Cookies, Privacy Settings, Terms of Service, Legal Notices & Trademarks, Report Noncompliance, and a copyright notice for © 2019 Autodesk Inc.

**FOLLOW FORGE**

- [Twitter](#)
- [Facebook](#)

**SOLUTIONS**

- [BIM 360](#)
- [Data Management](#)
- [Design Automation](#)
- [Model Derivative](#)
- [Reality Capture](#)
- [Viewer](#)
- [Webhooks](#)

**DOCUMENTATION**

- [Authentication](#)
- [BIM 360](#)
- [Data Management](#)
- [Design Automation](#)
- [Model Derivative](#)
- [Reality Capture](#)
- [Token Flex](#)
- [Viewer](#)
- [Webhooks](#)

**RESOURCES**

- [Get Help](#)
- [API Status](#)
- [Blog](#)
- [FAQ](#)
- [Code Samples](#)
- [Videos & Slides](#)

**ABOUT**

- [About Forge](#)
- [Pricing](#)
- [Success Stories](#)
- [Partners](#)
- [Forge Fund](#)
- [DevCon 2018](#)
- [DevCon 2017](#)
- [DevCon 2016](#)

Privacy/Cookies | Privacy Settings | Terms of Service | Legal Notices & Trademarks | Report Noncompliance | © 2019 Autodesk Inc. All rights reserved.

# Forge SDK

## § RESTful API をラップするサーバー実装用ユーティリティ

The screenshot shows a GitHub repository page for Autodesk Forge. At the top, there are two cards: one for the 'Forge Java SDK' and one for the 'Node.js boilerplate examples collection'. Below these are two larger cards: one for the 'Forge Node.js SDK' and one for the 'Forge Boilerplate Examples Collection'. The Java and Node.js cards each have a star rating and a brief description. The Node.js card also includes a screenshot of a 3D engine model.

Forge Java SDK	Node.js boilerplate examples collection	Forge Node.js SDK
 ★ 13 Provides Java SDK to help you easily integrate Forge REST APIs into the application	 ★ 41 Demonstrates Forge NPM packages	 ★ 33 Provides Node.js SDK to help you easily integrate Forge REST APIs into the application

# Forge SDK と Forge SDK のサンプル

## § 実際に使用するページ

§ <https://github.com/Autodesk-Forge/forge-api-nodejs-client>

§ forge-api-nodejs-client リポジトリ内のサンプルを参照

The screenshot shows the GitHub repository page for 'Autodesk-Forge / forge-api-nodejs-client'. The page includes the GitHub logo, navigation links for Features, Business, Explore, and Pricing, and a search bar. It displays the repository name, a brief description 'Easily integrate Forge REST APIs into your application', and statistics like 70 commits, 2 branches, 13 releases, 6 contributors, and Apache-2.0 license. A yellow bar highlights the repository's activity. At the bottom, it shows recent commits from 'spark-developer'.

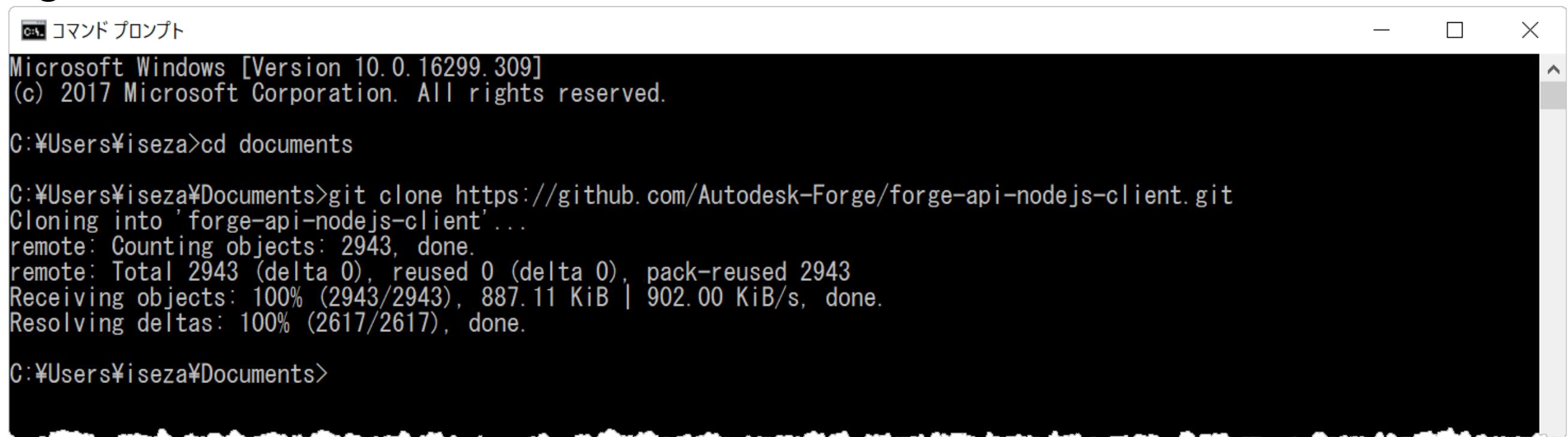
Commit	Message	Date
spark-developer	Remove console log	Latest commit 987d62c on 14 Apr
docs	added- (#16)	2 months ago
samples	Added basic DM samples	3 months ago
src	Remove console log	a month ago
test	Adds `state` argument to generateAuthUrl (#20)	a month ago



# チュートリアル環境のセットアップ

## 1. コマンドプロンプトでリポジトリ内容をコピー

§ `git clone <リポジトリ URL>` コマンド



The screenshot shows a Windows Command Prompt window titled "コマンドプロンプト". The window displays the following text:

```
Microsoft Windows [Version 10.0.16299.309]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\iseza>cd documents

C:\Users\iseza\Documents>git clone https://github.com/Autodesk-Forge/forge-api-nodejs-client.git
Cloning into 'forge-api-nodejs-client'...
remote: Counting objects: 2943, done.
remote: Total 2943 (delta 0), reused 0 (delta 0), pack-reused 2943
Receiving objects: 100% (2943/2943), 887.11 KiB | 902.00 KiB/s, done.
Resolving deltas: 100% (2617/2617), done.

C:\Users\iseza\Documents>
```

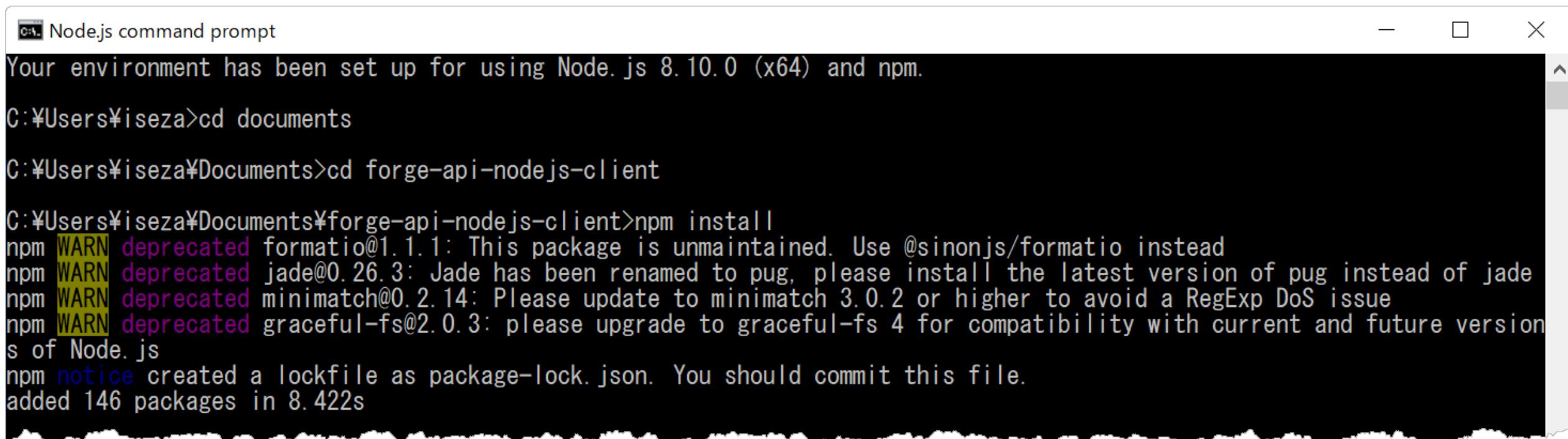
## 2. コピーされたリポジトリの場所を確認

§ `C:\Users\<Windows ユーザ名\Documents` フォルダ

§ `forge-api-nodejs-client` フォルダ

# チュートリアル環境のセットアップ ~ 続き

## 3. Node.js command prompt でパッケージをインストール § npm install



```
C:\ Node.js command prompt
Your environment has been set up for using Node.js 8.10.0 (x64) and npm.

C:\$Users\$iseza>cd documents

C:\$Users\$iseza\$Documents>cd forge-api-nodejs-client

C:\$Users\$iseza\$Documents\$forge-api-nodejs-client>npm install
npm WARN deprecated formatio@1.1.1: This package is unmaintained. Use @sinonjs/formatio instead
npm WARN deprecated jade@0.26.3: Jade has been renamed to pug, please install the latest version of pug instead of jade
npm WARN deprecated minimatch@0.2.14: Please update to minimatch 3.0.2 or higher to avoid a RegExp DoS issue
npm WARN deprecated graceful-fs@2.0.3: please upgrade to graceful-fs 4 for compatibility with current and future versions of Node.js
npm notice created a lockfile as package-lock.json. You should commit this file.
added 146 packages in 8.422s
```

## 4. Forge ポータルで Client ID/Client Secret を取得 § <https://forge.autodesk.com> でアプリ登録

# Node.js のローカル Web サーバーを設定する

5. .¥samples¥dmSample.js を Brackets で開く

- § FORGE\_CLIENT\_ID/FORGE\_CLIENT\_SECRET に4.を代入
- § BUCKET\_KEY に一意な Bucket 名を代入
- § FILE\_NAME にアップロードするファイル名を代入
- § FILE\_PATH にアップロードするファイルのフルパスを代入

# Node.js のローカル Web サーバーを設定する

```
var fs = require('fs');

var ForgeSDK = require('../src/index');

// TODO - insert your CLIENT_ID and CLIENT_SECRET
var FORGE_CLIENT_ID = 'デベロッパ ポータルで取得した Client ID',
    FORGE_CLIENT_SECRET = 'デベロッパ ポータルで取得した Client Secret';

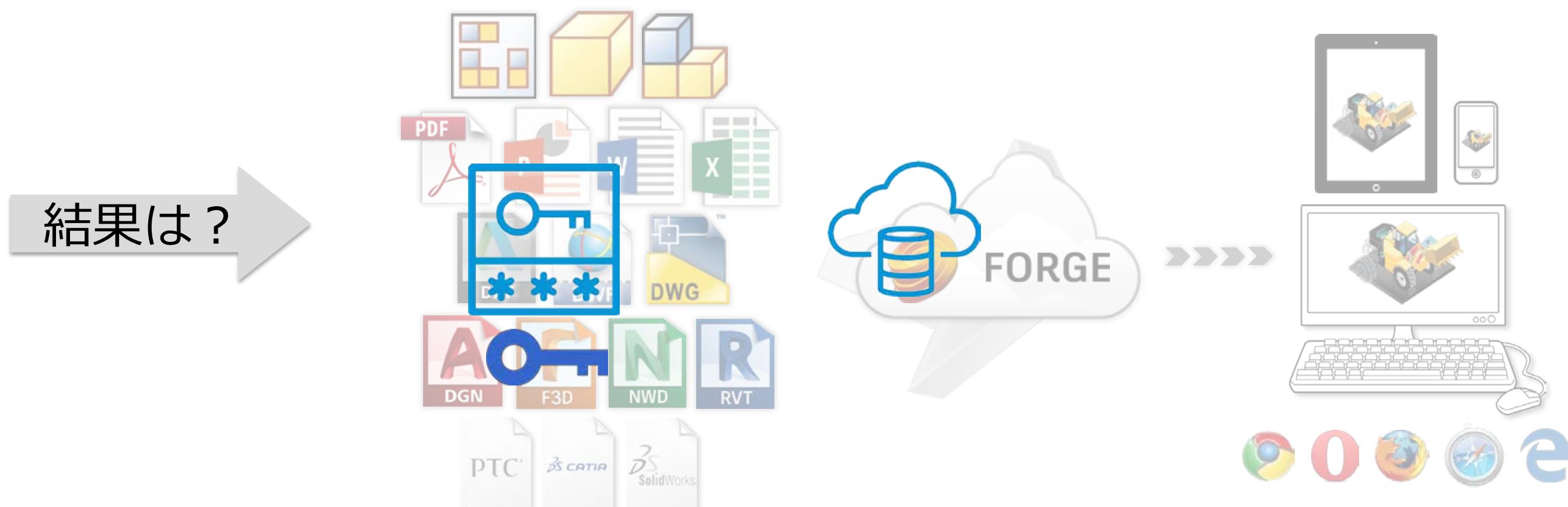
// TODO - Choose a bucket key - a unique name to assign to a bucket. It must be globally unique across all applications and
// regions, otherwise the call will fail. Possible values: -_.a-z0-9 (between 3-128 characters in
// length). Note that you cannot change a bucket key.
var BUCKET_KEY = '一意なBucket名にするための任意文字列' + FORGE_CLIENT_ID.toLowerCase();

// TODO - Choose a filename - a key for the uploaded object
var FILE_NAME = 'デザイン ファイル名.拡張子';

// TODO - specify the full filename and path
var FILE_PATH = 'デザイン ファイルのフルパス';
:
```

# Node.js でローカル Web サーバーを起動する

1. Node.js command prompt を起動
2. forge-api-nodejs-client¥samples フォルダに移動
3. 'node dmSample.js' と入力して実行



# Node.js でローカル Web サーバーを起動する

# 変換処理の実装を追加

1. samples¥dmSample.js を Brackets で再度開く
2. Forge SDK から Derivatives Client をインスタンス化

```
:  
var bucketsApi = new ForgeSDK.BucketsApi(), // Buckets Client  
objectsApi = new ForgeSDK.ObjectsApi(), // Objects Client  
derivativesApi = new ForgeSDK.DerivativesApi(); // Derivatives Client  
:
```

3. base64encode() 関数を getBuckets() 関数の後に追記  
§ ドキュメント IDを Base64 エンコード

```
function base64encode(str) {  
  return new Buffer(str).toString('base64');  
}
```

# 変換処理の実装を追加 ~ 続き

## 4. translateFile() 関数を base64encode() 関数の後に追記

```
var translateFile = function(encodedURN){  
  console.log("**** Translating file derivative");  
  var postJob =  
  {  
    input: {  
      urn: encodedURN  
    },  
    output: {  
      formats: [  
        {  
          type: "svf",  
          views: ["2d", "3d"]  
        }  
      ]  
    }  
  };  
  return new Promise(function(resolve, reject) {  
    derivativesApi.translate(postJob, {}, oAuth2TwoLegged, oAuth2TwoLegged.getCredentials()).then(  
      function(res){  
        resolve(res);  
      },function(err){  
        reject(err);  
      }  
    );  
  });  
};
```

# 変換処理の実装を追加 ~ 続き

## 5. manifestFile() 関数を translateFile() 関数の後に追記

```
var manifestFile = function (encodedURN) {
  console.log("**** Getting File Manifest Status");
  return new Promise(function(resolve, reject) {
    derivativesApi.getManifest(encodedURN, {}, oAuth2TwoLegged, oAuth2TwoLegged.getCredentials()).then(
      function(res){
        if (res.body.progress != "complete"){
          console.log("The status of your file is " + res.body.status);
          console.log(" Please wait while we finish Translating your file");
        }else{
          console.log("**** " + res.body.status);
          console.log("**** " + res.body.progress);
          resolve(res);
        }
      },function(err){
        reject(err);
      }
    );
  });
}
```

# 変換処理の実装を追加 ~ 続き

## 6. 変換と変換終了を判定する処理を追記

```
uploadFile(BUCKET_KEY, FILE_PATH, FILE_NAME).then(function(uploadRes){  
  
    console.log("**** Upload file response:", uploadRes.body);  
    deleteFile(BUCKET_KEY, FILE_NAME).then(function(deleteRes) {  
        console.log("**** Delete file response status code:", deleteRes.statusCode);  
    }, defaultHandleError);  
    var objectId = uploadRes.body.objectId;  
    console.log("objectId:", objectId);  
    var urn = base64encode(objectId);  
    console.log("urn:", urn);  
    translateFile(urn).then(function(translateRes){  
        manifestFile(urn).then(function(){  
            console.log("**** Your File is ready for viewing");  
        }, defaultHandleError)  
    }, defaultHandleError);  
  
}, defaultHandleError);
```

# Viewer での表示

Node.js command prompt

```
C:\Users\isezaki\Documents\forge-api-nodejs-client\samples>node dmSample.js
**** Got Credentials { access_token: 'eyJhbGciOiJIUzI1NiIsImtpZC16Imp3dF9zeW1tZXRyaWNfa2V5In0.eyJjbGIlbnRfaWQiOjJWMlpQcnNaR242eWVqTXRSNzZHM2IMR2pxRngyamcycCisImV4cCI6MTUyMjA1NjE0NywiI2NvcGUiOlsizGF0YTp3cmI0ZSIsImRhGE6cmVhZCIsImJ1Y2tlDpyZFkIwiYnVja2V0OnVwZGF0ZSIsImJ1Y2tlDpjcmVhdGUiXSwiYXVkJoiiaHR0cHM6Ly9hdXRvZGVzay5jb20vYXVkJL2p3dGV4cDYwiwiianRpIjoiU2JjYM5dUhCN2Ywa3ZSOU5DdINzNTdnWHpxTzJNVFBdjFkNWgzOFhENFNFXF2NlhSDixaDVqNGFVdmRMMCJ9. yB9q49y8cbk1Bq90wTOBwoHpLGYyVm1dc8rxQ33G3U',
  token_type: 'Bearer',
  expires_in: 3599,
  expires_at: 2018-03-26T09:22:26.296Z }
**** Creating bucket if not exist : forge_sample_v2zprczgn6yejmtr76g3ilgjqfx2jg2p
**** Getting bucket details : forge_sample_v2zprczgn6yejmtr76g3ilgjqfx2jg2p
**** Create bucket if not exist response: { bucketKey: 'forge_sample_v2zprczgn6yejmtr76g3ilgjqfx2jg2p',
  bucketOwner: 'V2ZPrcZGn6yejMtR76G3iLGjqFx2jg2p',
  createdDate: 1522051630150,
  permissions:
    [ { authId: 'V2ZPrcZGn6yejMtR76G3iLGjqFx2jg2p', access: 'full' } ],
  policyKey: 'temporary' }
**** Getting all buckets
**** Uploading file. bucket:forge_sample_v2zprczgn6yejmtr76g3ilgjqfx2jg2p filePath:C:/Models/Chair.f3d
**** Get all buckets response:
forge_sample_v2zprczgn6yejmtr76g3ilgjqfx2jg2p
v2zprczgn6yejmtr76g3ilgjqfx2jg2p-transient
**** Upload file response: { bucketKey: 'forge_sample_v2zprczgn6yejmtr76g3ilgjqfx2jg2p',
  objectId: 'urn:adsk.objects:os.object:forge_sample_v2zprczgn6yejmtr76g3ilgjqfx2jg2p/Chair.f3d',
  objectKey: 'Chair.f3d',
  sha1: 'dee7dbbcff880de4ca854c17e6ba830110d1cdf',
  size: 1501003,
  contentType: 'application/octet-stream',
  location: 'https://developer.api.autodesk.com/oss/v2/buckets/forge_sample_v2zprczgn6yejmtr76g3ilgjqfx2jg2p/objects/Chair.f3d' }
objectId: urn:adsk.objects:os.object:forge_sample_v2zprczgn6yejmtr76g3ilgjqfx2jg2p/Chair.f3d
urn: dXJuOmFkc2sub2JqZWN0czpvcy5vYmp1Y3Q6Zm9yZ2Vfc2FtcGxIX3YyenByY3pnbjZ5ZWptdHI3NmcaWxnanFmeDJqZzJwLONoYWiyLmYzZA==
**** Translating file derivative
**** Getting File Manifest Status
****success
****complete
**** Your File is ready for viewing
```

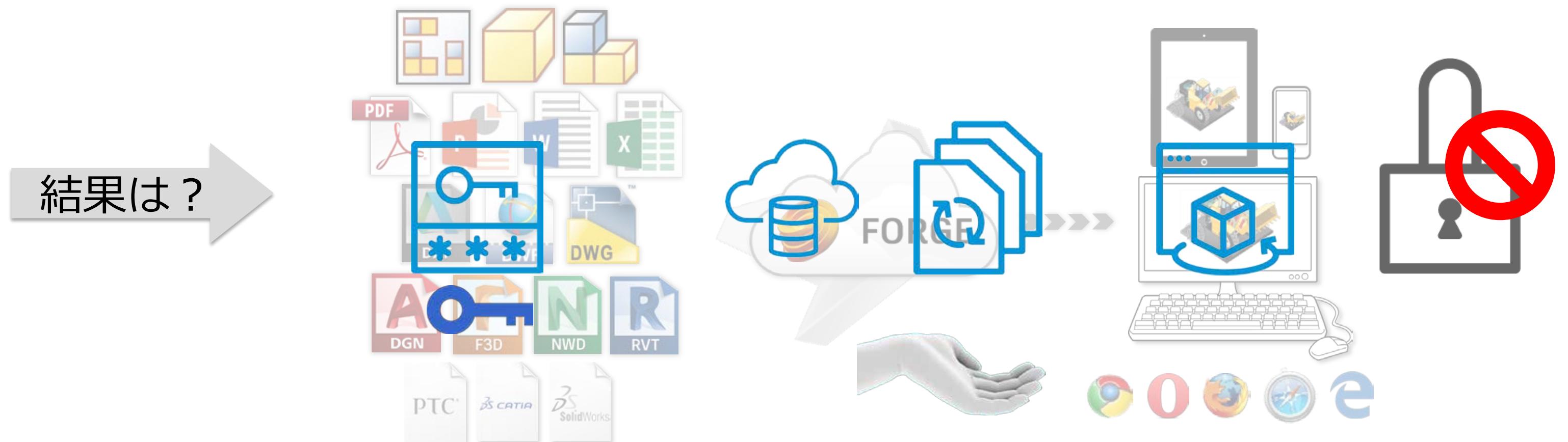
**Access Token**

**エンコードされた URN**

# ここまで実装

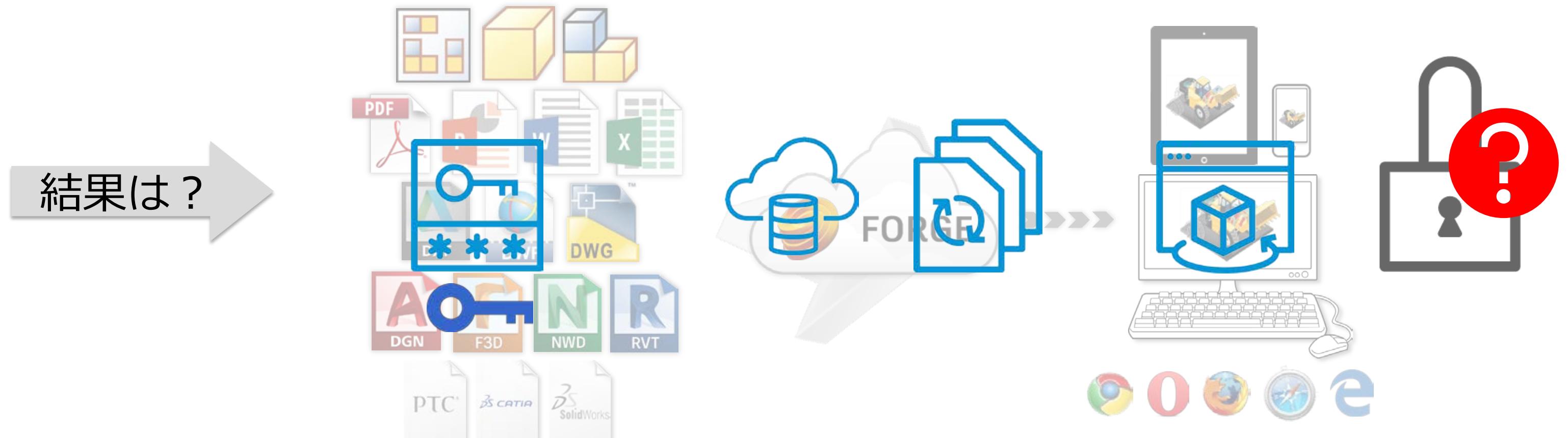
§ Viewer 表示が手動

§ Access Token と URN は URL パラメータ渡し



# Viewer 表示部分の自動化

- § ここで例：Node.js 実装による Web サーバー処理
- § 本来、Web サーバー実装に依存
- § さまざまな方法が考えられる

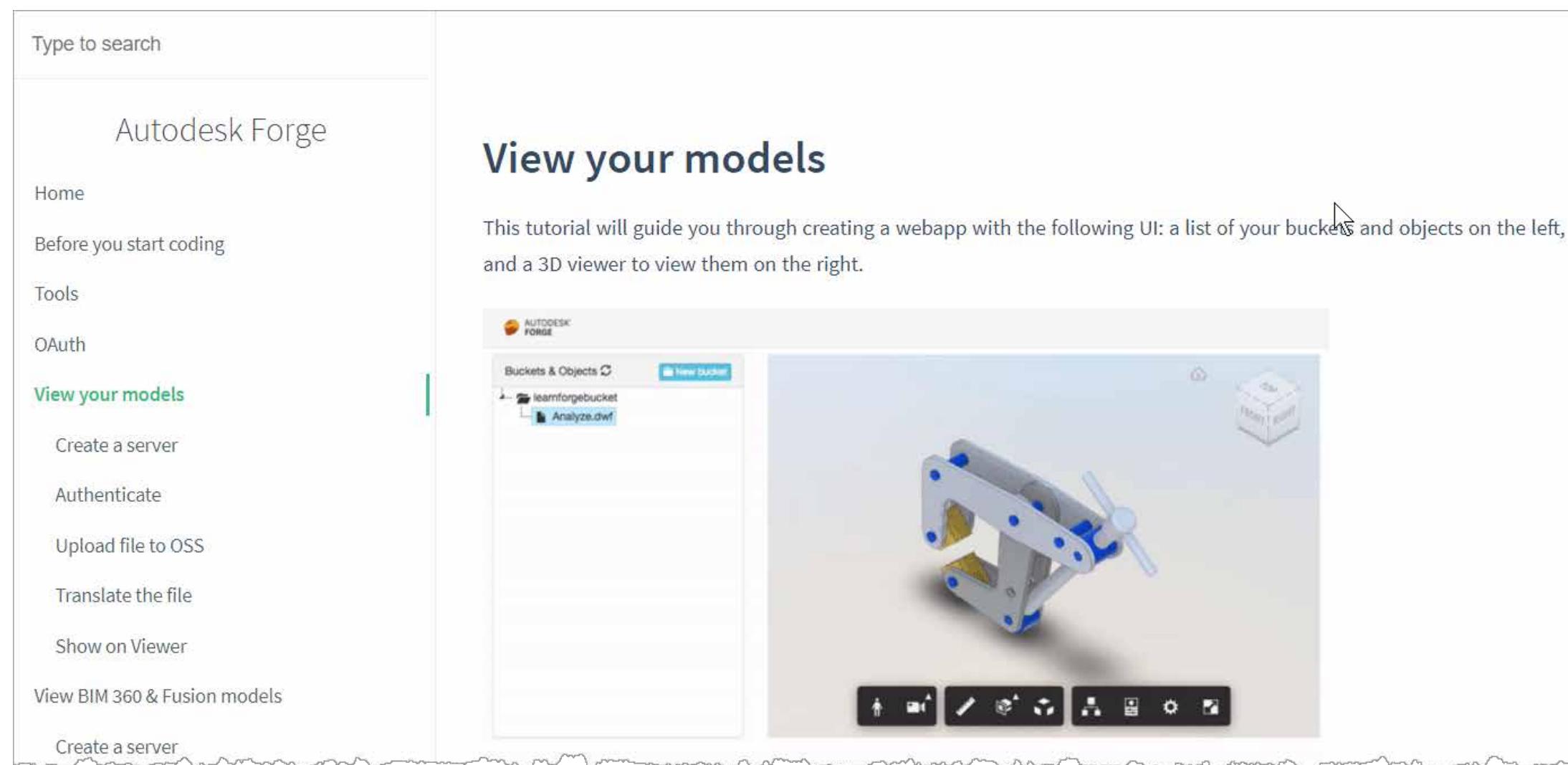


# 次のステップ：Learn Forge

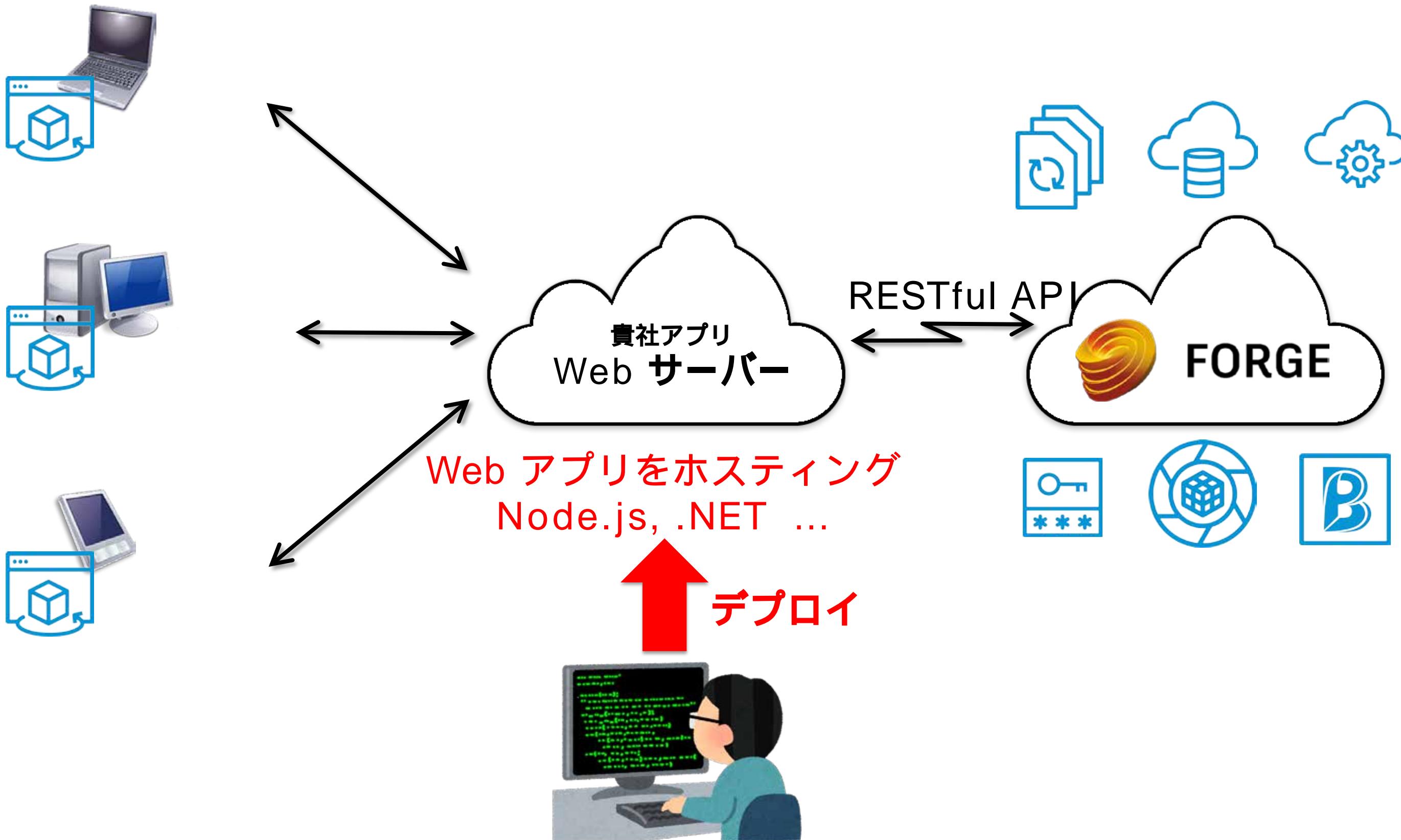
## § Forge 全般の学習サイト（英語）

§ <https://learnforge.autodesk.io/#/tutorials/viewmodels>

## § さまざまな Web サーバーテクノロジ別に手順を解説



# ローカル開発環境からサイト公開へ



# ご参考：デプロイ作業について

- § Web アプリのデプロイ：GitHub Desktop を用了 GitHub へのプッシュ
  - § [https://adndevblog.typepad.com/technology\\_perspective/2019/05/push-web-app-to-github-using-github-desktop.html](https://adndevblog.typepad.com/technology_perspective/2019/05/push-web-app-to-github-using-github-desktop.html)
- § Web アプリのデプロイ：Heroku へのデプロイ
  - § [https://adndevblog.typepad.com/technology\\_perspective/2019/05/deploy-web-app-to-heroku.html](https://adndevblog.typepad.com/technology_perspective/2019/05/deploy-web-app-to-heroku.html)
- § Web アプリのデプロイ：Azure へのデプロイ
  - § [https://adndevblog.typepad.com/technology\\_perspective/2019/05/deploy-web-app-to-azure.html](https://adndevblog.typepad.com/technology_perspective/2019/05/deploy-web-app-to-azure.html)
- § Web アプリのデプロイ：Visual Studio から Azure への直接デプロイ
  - § [https://adndevblog.typepad.com/technology\\_perspective/2019/05/deploy-web-app-from-visual-studio.html](https://adndevblog.typepad.com/technology_perspective/2019/05/deploy-web-app-from-visual-studio.html)



AUTODESK®

Make anything.