

Handbook of 3D City Models

3D都市モデル導入のためのガイドブック



実証環境構築マニュアル

3D City Model Demonstration Manual

series
No. 09

はじめに

- Project PLATEAUでは、2020年度に3D都市モデルの実証環境構築のための実証調査を実施した。本実証調査は、3D都市モデル及びこれを活用したユースケース開発のための可視化環境である「PLATEAU VIEW」を開発することで、3D都市モデルがもたらすソリューションの価値を検証することを目的としている。
- 2021年度には、クリッピング機能や日射シミュレーション機能など、Web上で実施可能な解析機能を追加「PLATEAU VIEW 1.1」をリリースした。また、「第2章 実証環境の構築手順」の内容を大幅に拡充し、PLATEAU VIEWを構築するためのチュートリアルを充実させるとともに、Project PLATEAU GitHub上でTerria.js用カタログ生成アプリとそのチュートリアルを公開した。
- 2022年度には、データセットの可視化機能に限定されていた「PLATEAU VIEW 1.1」を発展させ、データ登録・管理・配信機能及び機能追加を行った「PLATEAU VIEW 2.0」を開発し、GitHub上でソーススクリプトを公開した。さらに、PLATEAU VIEWを利用するための技術チュートリアルを公式ウェブサイト上で公開した。
- 実証環境の構築を検討するに当たっては、Project PLATEAUのオープンデータの思想に基づき、出来るだけオープンソースのフレームワークを利用することで、ベンダーロックを回避する形でシステム構成を行うことを心掛けた。また、オープンソースの利用には、地方公共団体等が低コストで類似のシステムを構築できることや、技術者コミュニティとの連携によるシステムの持続的な発展が期待されること等のメリットもある。
- 本マニュアルは、PLATEAU VIEWの開発により得られた成果をもとに、その機能、システム環境、仕様、構築手法等を解説することで、地方公共団体や民間企業、技術者コミュニティ等に所属する多様なプレイヤーが3D都市モデルの可視化環境を構築する際に参照できる知見を提供し、3D都市モデルの整備及びこれを活用したユースケース開発への参画のすそ野を広げることを目的とするものである。
- 地方公共団体や民間企業、技術者等の多くの方に本マニュアルを参照していただき、Project PLATEAUの技術コミュニティがさらに発展することを期待する。

アップデートノート

2023.3.22 「実証環境構築マニュアル 第3.0版」

2022年度の調査結果を踏まえ、以下の項目を改訂した。

- 「第1編 PLATEAU VIEW 2.0」を新たにリリースしたPLATEAU VIEW 2.0編として拡充
- 「第2編 PLATEAU VIEW 1.1」をPLATEAU VIEW 1.1編として改訂

2022.3.25 「実証環境構築マニュアル 第2.0版」

2021年度の調査結果を踏まえ、以下の項目を改訂した。

- 「1.4 ソフトウェア構成」を新たにリリースしたPLATEAU VIEW 1.1の構成にアップデート
- 「第2章 実証環境の構築手順」の内容を大幅に拡充
- 「第3章 ユーザーマニュアル」をPLATEAU VIEW 1.1に合わせて改訂

2021.3.26 「実証環境構築マニュアル 第1.0版」

■ 目次

第1編 PLATEAU VIEW 2.0

第1章 実証環境の構成	5
1.1 本マニュアルの目的	6
1.2 全体構成	7
1.3 PLATEAU CMS	10
1.4 FMEサーバ	19
1.5 PLATEAU Editor・PLATEAU VIEW	20
1.6 PLATEAU SDK for Unity/Unreal	28
第2章 実証環境の構築手順	36
2.1 PLATEAU CMS・PLATEAU Editorの環境構築	37
2.2 FMEサーバ	65
第3章 ユーザーマニュアル	68
3.1 PLATEAU CMSの利用方法	69
3.2 PLATEAU Editorの利用方法	112
3.3 PLATEAU VIEWの利用方法	147
3.4 PLATEAU SDK for Unity/Unrealの利用方法	152

第2編 PLATEAU VIEW 1.1

第1章 実証環境の構成	162
1.1 PLATEAU VIEW 1.1 について	163
1.2 サーバ環境	165
1.3 サーバ構成	166
1.4 ソフトウェア構成	167
1.5 実証環境のデータ構成	175
第2章 実証環境の構築手順	180
2.1 実証環境構築の前提	181
2.2 インフラ構築とデプロイ	186
2.3 Dockerを用いた環境構築	188
2.4 データ変換の方法境構築	189
2.5 カタログの編集	191
第3章 ユーザーマニュアル	192
3.1 PLATEAU VIEWのUI機能の利用方法	193

第1編 PLATEAU VIEW 2.0

第1章 実証環境の構成

1.1 本マニュアルの目的

3D都市モデルの実証環境とは、3D都市モデル及びこれを活用したユースケース開発のために可視化環境を提供するプログラム、サーバ、データ等の一連のシステムをいう。具体的な機能としては、3D都市モデルそれ自体を可視化することに加え、3D都市モデルと共に分析やシミュレーション等に用いられる各種データの可視化も行う。これにより、3D都市モデルの提供価値を検証することができる。

2020年度のProject PLATEAUでは、実証環境としてウェブ上で閲覧可能なビューア「PLATEAU VIEW 1.0」を開発し、ウェブサイト「PLATEAU」上で公開した。続く、2021年度には、PLATEAU VIEW 1.0に機能追加を行った「PLATEAU VIEW 1.1」を開発し、アップデートを行った。

そして、2022年度では、PLATEAU VIEW 1.1を発展させ、データ登録・管理・配信機能及び機能追加を行った「PLATEAU VIEW 2.0」を開発し、アップデートを行った。

PLATEAU VIEW 2.0は、3D都市モデルやユースケースのデータ登録・管理・配信を可能とするコンテンツ管理システム（PLATEAU CMS/Editor）及び全国約100都市以上で整備された3D都市モデルを可視化すると共に、都市計画決定情報、人流データ、都市アセットデータ等の各種のデータを3D都市モデルに重畳して軽量・高速に表示することが可能なWebGIS（PLATEAU VIEW）を一般向け・行政向けに提供するものである。

本マニュアルでは、PLATEAU VIEW 2.0及びPLATEAU VIEW 1.1の機能、システム環境、仕様、構築手法等を解説することで、地方公共団体や民間企業等が3D都市モデルの可視化環境を構築する際に参照できる知見を提供し、3D都市モデルの整備及びこれを活用したユースケース開発を促進することを目的とするものである。

なお、PLATEAU VIEW 2.0及びPLATEAU VIEW 1.1のソースコードについては、Project PLATEAUのGitHubにおいてオープンソースとして公開しているので、参考にして頂きたい。

PLATEAU VIEW 2.0:

Project PLATEAU GitHub : <https://github.com/Project-PLATEAU/PLATEAU-VIEW-2.0>

PLATEAU VIEW 1.1:

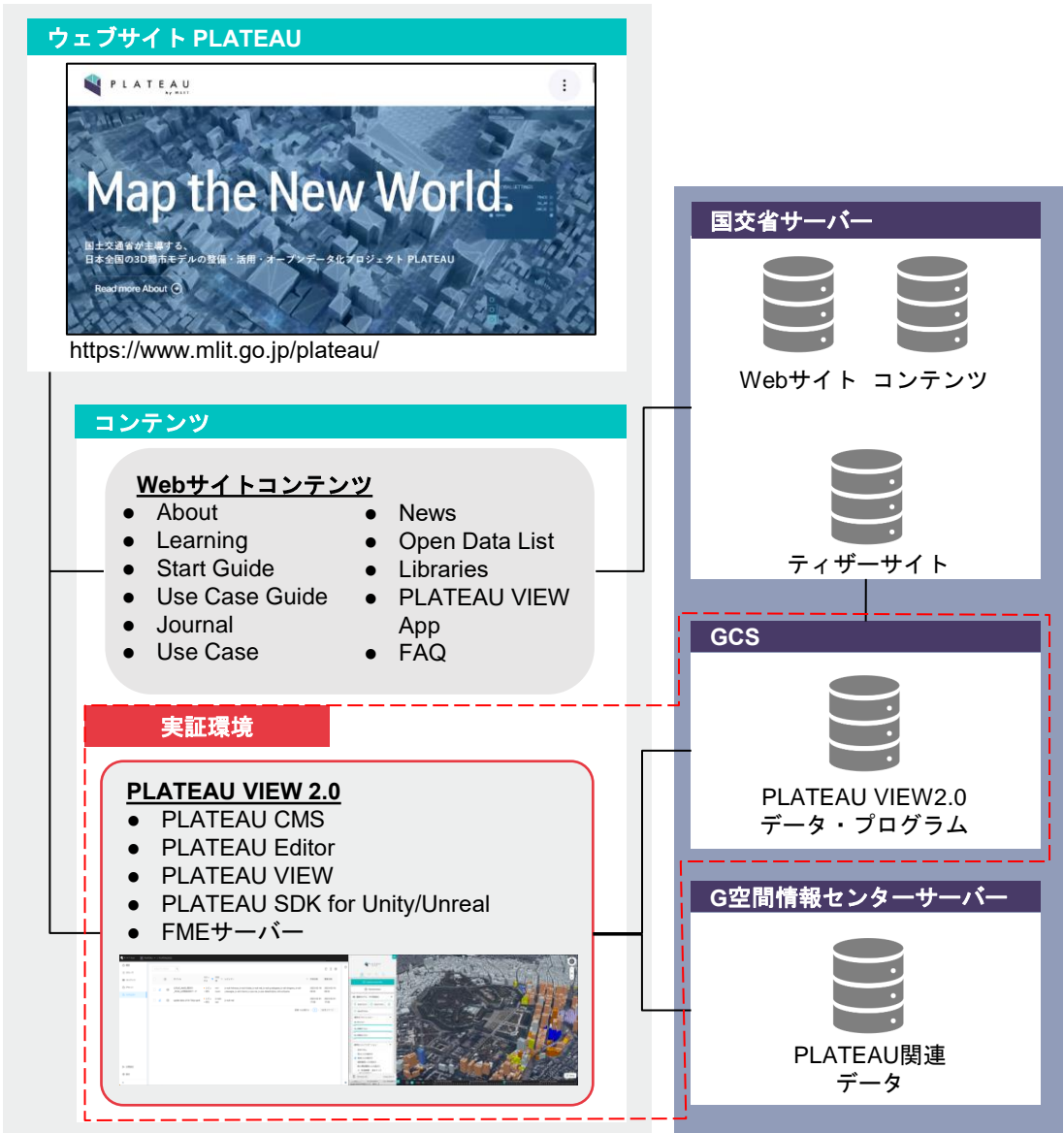
Project PLATEAU GitHub : <https://github.com/Project-PLATEAU/PLATEAU-VIEW-1.1>

1.2 全体構成

1.2.1 Project PLATEAU ウェブサイトの構成

Project PLATEAUを構成するウェブサイトの校正について、以下に全体図を示す。

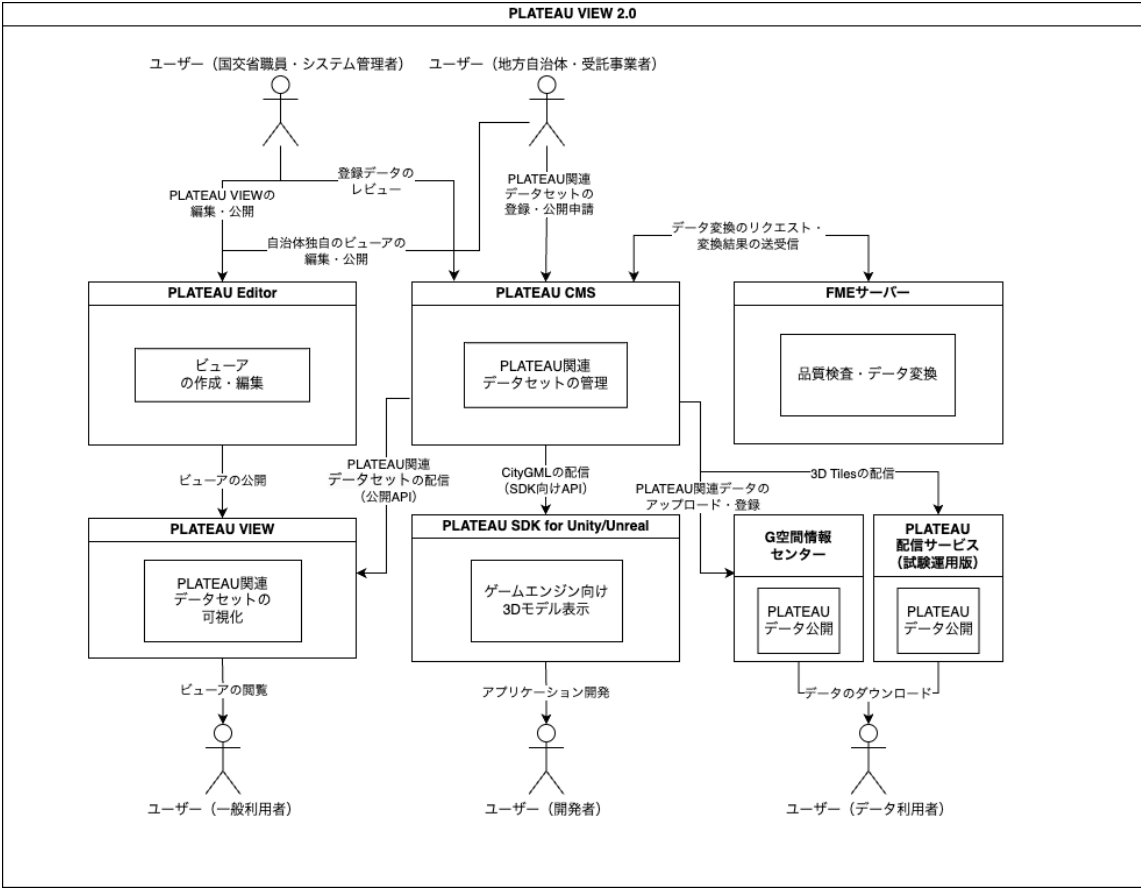
表 Project PLATEAU ウェブサイトの構成



1.2.2 システム全体構成

PLATEAU VIEW 2.0を構成するシステム及び想定ユーザーを解説する。PLATEAU VIEW 2.0は複数のシステムで構成されており、以下に全体図を示す。

表 PLATEAU VIEW 2.0 システム全体構成



1.2.3 用語集

PLATEAU VIEW 2.0において独自に使用される用語をまとめる。

表 用語集：ユーザー

用語	説明
国交省職員・システム管理者	PLATEAU CMS上での登録データのレビュー、PLATEAU Editor上での公開ページの編集等を行う。
地方自治体・受託事業者	PLATEAU CMS上でPLATEAU関連データセットの登録を行う。
データ利用者	G空間情報センターへ登録されたPLATEAUデータを利用する。
一般利用者	PLATEAU VIEWを利用し、PLATEAU関連データセットの閲覧を行う。
アプリ開発者	PLATEAU SDK for Unity/Unrealを利用し、PLATEAUデータを利用したアプリケーション開発を行う。

表 用語集：システム

用語	説明
PLATEAU CMS	PLATEAU関連データセットを管理する。
PLATEAU Editor	PLATEAU VIEW・自治体独自のビューアを編集・公開する。
PLATEAU VIEW	PLATEAU関連データセットを可視化する。
FMEサーバー	PLATEAUデータの品質検査・データ変換等を行う。
PLATEAU SDK for Unity/Unreal	Unity・Unreal Engine向けSDK。PLATEAUデータをゲームエンジン上で描画する。
G空間情報センター	PLATEAUデータをオープンデータとして公開する。

表 用語集：データ種別

用語	説明
PLATEAUデータ	都市局が定める「3D都市モデル標準製品仕様書」に準拠して作成された3D都市モデルのデータ。CityGML2.0形式で作成される。
ユースケースデータ	PLATEAUデータに重畳して表示することを目的として作成されたユースケースに関するデータ。3D浸水想定区域図や動的データ、シミュレーションデータなど。
その他データ	PLATEAU VIEWで閲覧可能な、PLATEAUデータおよびユースケースデータ以外のデータ。避難施設、ランドマーク、行政界など。
PLATEAU関連データセット	PLATEAUデータ、ユースケースデータ、その他データの総称。

1.3 PLATEAU CMS

1.3.1 PLATEAU CMSとは

1.2のシステム全体構成で述べたように、PLATEAU VIEW 2.0のシステムの機能は、大きく「データの管理」と「データの可視化」に分かれる。PLATEAU CMSは、このうち「データの管理」を担うシステムのことを指す。

CMSとは「コンテンツ管理システム（Content Management System）」の略で、一般的な概要は以下の通りである。

コンテンツ管理システム（CMS）は、企業がデジタルコンテンツを管理するのに役立ちます。チーム全体がこれらのシステムを使って、コンテンツの作成、編集、整理、公開を行うことができます。コンテンツを保存する単一の場所として機能し、組み込み（または設計された）ワークフローを使用して、共同デジタルコンテンツ管理および作成のための自動化されたプロセスを提供します。役割に応じて、個人にはさまざまな特権と責任が与えられます。例えば、著者は作品を投稿し保存することができますが、編集者は作品を修正し公開することができます。管理者は、こうした作業をすべて行えるだけでなく、組織内の他の人にコンテンツの更新や改訂の許可を与えることもできます。

CMSでは、最小限の技術コストでWebサイトやWebサイトのコンテンツを作成・管理できるため、プロジェクト・マネージャーやトラフィック・マネージャーのような役割を果たす必要なしに、より優れたコンテンツの作成に集中することができます。CMSは、コンテンツ管理のための簡単で費用対効果の高いソリューションを提供します。これにより、企業は専任のコンテンツ開発チームに投資しなくても、コンテンツを管理・配信することができます。

引用元 : <https://www.oracle.com/jp/content-management/what-is-cms/>

この中でもPLATEAU CMSは、管理対象のコンテンツをAPIを通じて提供することを前提とした「ヘッドレスCMS」に位置付けられる。管理者だけでなく、地方自治体・受託事業者やユースケースデータの登録者など様々な事業者がPLATEAU VIEW 2.0で公開するPLATEAU関連データセットを一元管理し、APIとして公開することができるシステムである。

(1) 使用ソフトウェア・サービス

上記システムを構築するために、オープンソースソフトウェア（OSS）と、有償のクラウドサービスを組み合わせて利用している。クラウドサービスの詳細については次項の各コンポーネントの説明にて述べる。

表 使用ソフトウェア・サービス一覧

項目	項目	説明
Re:Earth CMS	OSS	PLATEAU CMSのナレッジを元に、Eukarya社がOSSとして提供するプロダクト。様々なデータの管理を行える汎用的なCMS。ワークスペースの作成・プロジェクトの作成・スキーマの定義・コンテンツの作成・アセットのアップロード・公開申請・公開API・インテグレーションAPIによる連携などの機能を有する。
FME	有償	データの品質検査と変換を行うアプリケーション。詳しい説明は1.4を参照。
G空間情報センター	オープンデータ・ストレージサービス	官民問わず様々な主体により整備・提供される多様な地理空間情報を集約し、利用者がワンストップで検索・ダウンロードし利用できる、産学官の地理空間情報を扱うプラットフォーム。OSSのCKANを用いて構築されている。
Auth0	有償クラウドサービス	Auth0社が提供するクラウドサービス。アカウントの管理・認証・認可を行うIDプロバイダを提供する。Auth0を使用することで、開発者はアプリケーションに安全で使いやすく信頼性の高い認証・認可機能を組み込むことができる。PLATEAU EditorもAuth0を使用して認証・認可機能を実現している。
SendGrid	有償クラウドサービス	Twilio社が提供するクラウドサービス。企業や開発者がアプリケーションやWebサイトから大量のメールを配信するために使用される。PLATEAU CMSではご意見ご要望のメール送信で使用している。
Google Cloud Platform (GCP)	有償クラウドサービス	Google社が提供するクラウドコンピューティングプラットフォーム。PLATEAU CMSを動作させるためのサーバーや、ファイルを保存するためのサーバーをGCP上に構築している。
MongoDB Atlas	有償クラウドサービス	MongoDB社が提供するクラウドサービス。保守運用が自動化されたマネージドなMongoDBを提供している。MongoDBとは、ドキュメント指向のNoSQLデータベースで、データの柔軟性と拡張性が特徴。PLATEAU CMSはデータベースとしてMongoDBを使用している。

PLATEAU CMSはその他様々な技術を組み合わせて構築されている。

PLATEAU CMSで内部的に利用されている技術・ライブラリ等

項目	説明
Go	プログラミング言語の1つで、Googleによって開発された。構文がシンプルであり、かつ処理が高速な言語であり、主にバックエンド開発に用いられる。PLATEAU CMSのバックエンド実装に利用されている。
TypeScript	プログラミング言語の1つで、Microsoftによって開発された。JavaScriptに静的型付けを加えたスーパーセットであり、大規模システムの開発に用いられる。PLATEAU CMSではフロントエンド向け開発に利用されている。
React	Meta(旧Facebook)によって開発されたUI構築のためのJavaScriptライブラリ。特に大規模かつ複雑なUI実装において利用される。PLATEAU CMSではフロントエンド向け開発に利用されている。
CesiumJS	デジタル3D地球儀上に様々な情報を描画することができる地図エンジン。Webブラウザ上で動作し、WebGLを用いて描画を行うため、PCやスマートフォンで閲覧することができる。PLATEAU CMS上では、データプレビューで使用している。
Resium	React上でCesiumJSを手軽に利用可能にするコンポーネントを提供するライブラリ。Eukarya開発。PLATEAU CMS上では、データプレビューで使用している。
GraphQL	API向けに作られたクエリ言語およびランタイムを指す。WebAPIの開発において、RESTなどの方式と比較して、より柔軟かつ効率的なAPIの提供を可能にする。PLATEAU CMSではバックエンドとフロントエンド間の通信に利用されている。

(2) PLATEAU CMSにおける用語

表 PLATEAU CMSにおける用語集

用語	意味	身近な例
アカウント	管理者、地方自治体・受託事業者、ユースケースデータの登録者それぞれに1つずつ与えられるユーザーアカウントを指す。	--
ワークスペース	複数のユーザーが同じワークスペースで作業できる場所のことを指す。PLATEAU VIEW 2.0プロジェクトでは、管理者、地方自治体・受託事業者、ユースケースデータの登録者が全員同じワークスペース「PLATEAU」で操作する。	Slackのワークスペース・GitHubのオーガナイズーション
プロジェクト	ワークスペースに複数作成可能で、データ管理の目的に応じて作成する。管理者、地方自治体・受託事業者、ユースケースデータの登録者が全員同じプロジェクトで操作をする。	GitHubのレポジトリ
モデル	データを管理する単位で、 管理者のみが 、管理するデータのスキーマを設定できる。プロジェクトの複数作成が可能。	ExcelやRDBにおける表
スキーマ	モデルのデータ構造を指す。どんなフィールドがどんな型のデータを持つかや、制約条件などを定義する。 管理者のみが 定義可能で、地方自治体・受託事業者、ユースケースデータの登録者はこのスキーマに沿ってデータを入力する。	--
アイテム	管理するデータの最小単位で、スキーマに沿って実際のデータをアイテムとして登録する。3D都市モデルデータ登録者、ユースケースデータの登録者は対象データをアイテムとして入力する。	ExcelやRDBにおける行
フィールド	管理するデータの属性を指す。整数値、文字列などのデータ型を 管理者のみが 設定する。	ExcelやRDBにおける列
アセット	アップロードされたファイルを指す。サイズ・作成日時・URL・種類などのメタデータを持つ。複数ファイルの集まりをまとめて1つのアセットとして扱うことができる。管理者、地方自治体・受託事業者、ユースケースデータの登録者はPLATEAU関連データセットをアセットとしてアップロードする。	--
公開API	外部からアクセスされるPLATEAU CMSの認証不要のAPIを指す。登録したアイテムを公開リクエストで承認されるとそのアイテムが公開APIから配信されるようになる。PLATEAU VIEW 2.0 ではこの公開APIから配信されたデータがPLATEAU VIEWで使用され、自動的にデータカタログに掲載される。	--
公開リクエスト	登録したアイテムはすぐにはAPIとして配信されない。これを公開するためには管理者の承認を得ることが必要である。PLATEAU CMSでは、作成・更新したアイテムに対して、PLATEAU CMS上で公開の申請を出すことができる。この申請が承認されるとそのアイテムのデータが公開され、公開API経由で配信される。アイテムごとに個別に公開状況が管理されており、それぞれのアイテムごとに公開リクエストが必要である。	GitHubのプルリクエスト
インテグレーション	PLATEAU CMS以外の外部アプリケーションとの連携機能を指す。PLATEAU CMSのワークスペースにインストールすることで、インテグレーションAPIを利用してデータの取得や変更を行ったり、任意のイベント発生時にWebhookを利用して外部アプリケーションと連携を行うことができる。	Slack App

（３）PLATEAU CMSの主な機能

PLATEAU CMSでは主に以下の機能が利用可能である。詳しい使い方は、3.1を参照されたい。

- ワークスペースの作成・ユーザーの招待
- プロジェクトの作成
- スキーマの定義
- コンテンツの登録・編集
- アセットの登録・Zipファイルの解凍・プレビュー
- コンテンツの公開リクエスト
- コンテンツの公開（公開APIとしてコンテンツを公開可能）
- インテグレーションの作成・インストール（外部システムとの連携が可能で、本システムではFMEサーバーとの連携等で利用。）

（４）対応データフォーマット等

1. アセットの対応データフォーマット

PLATEAU CMSでは、全てのフォーマットの静的ファイルをアップロードが可能であるが、特に以下のファイルフォーマットに関してはプレビュー機能をサポートしている。

- 画像データ
 - PNG
 - JPEG
 - SVG
 - GIF
- GISデータ
 - GeoJSON
 - CZML
 - KML
 - Mapbox Vector Tiles (MVT)
 - 3D Tiles

2. スキーマのフィールド型

スキーマのフィールド型としては、以下のデータ型をサポートしている。画像データ

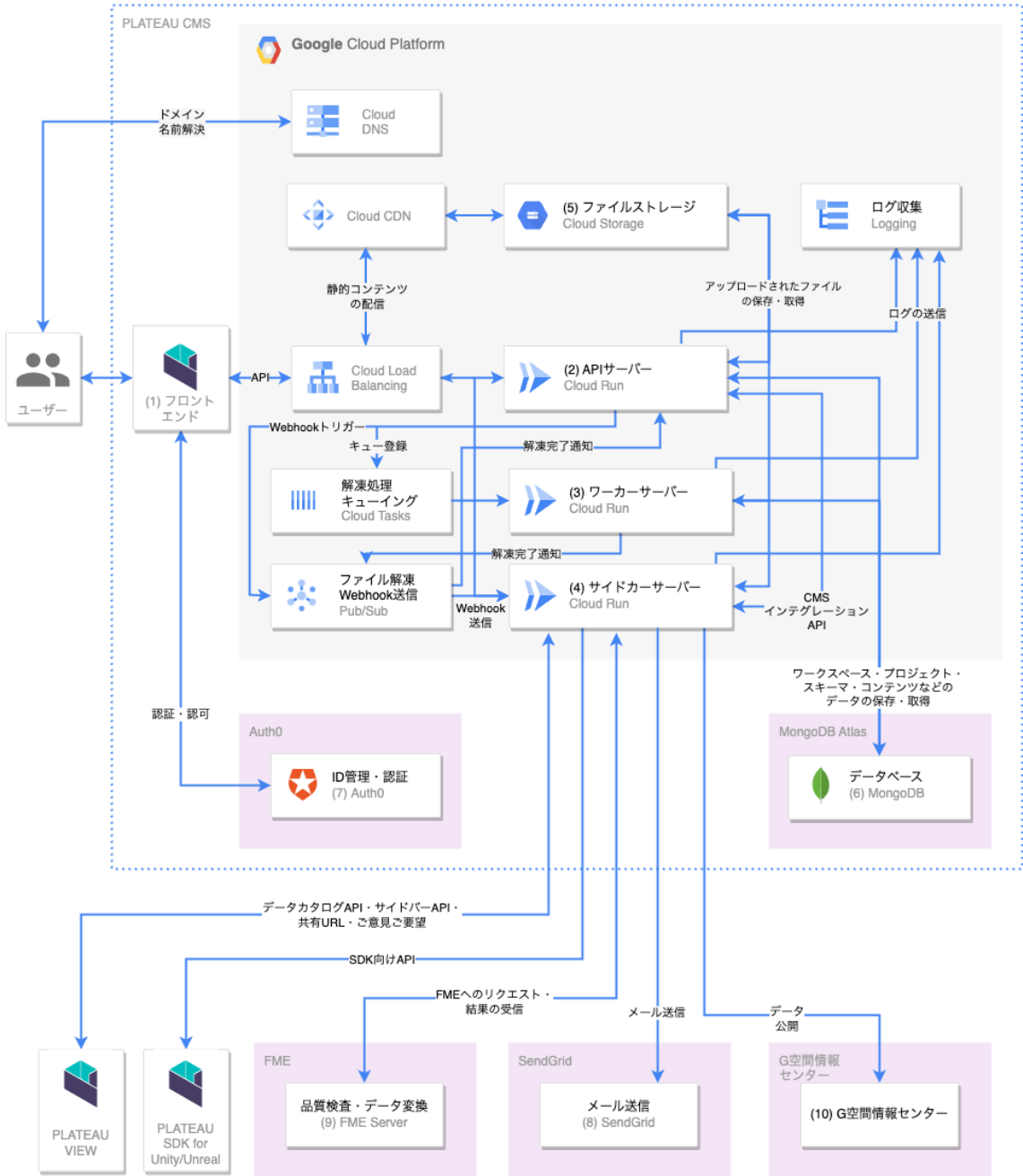
- テキスト: 短文向けのフィールド
- テキストエリア: 長文向けのフィールド
- マークダウン: マークダウンのフィールド
- アセット: アセットをリンクするためのフィールド
- 選択: 選択式のフィールド
- 整数値: 数値のフィールド
- URL: URLのフィールド

また、解凍処理が可能なアセットのファイルサイズはZipファイルで5GB程度である。また、アップロード可能なファイルサイズは、60分以内にアップロード可能なファイルサイズとなる。（インフラストラクチャの制約のため）

1.3.2 システム構成の全体像

PLATEAU CMSのシステム構成について説明する。

表 PLATEAU CMSのシステム構成



(1) フロントエンド

Webブラウザ上で動作する、(HTML・CSS・JavaScriptによる) フロントエンドアプリケーション。APIサーバー・Auth0・その他各種サーバーとの通信を行い、UIや、プレビュー機能による地図の表示を行う。

（2）APIサーバー（Cloud Run）

PLATEAU CMSのAPIサーバーであり、HTTPサーバーとして外部からのリクエストを受信している。MongoDBやGoogle Cloud Storageと連携して、アイテムの保存やプロジェクトの管理・公開などの、様々なビジネスロジックを実行する。

APIサーバーは、Cloud Run 上で動作する。Cloud Run とは、GCPで利用可能なサーバーレス（CaaS）プラットフォームであり、Dockerコンテナをデプロイすることで、サーバーの保守管理の手間なしに、アプリケーションをクラウド上で動作させることができる。同時接続リクエスト数が規定数以上に達すると自動的にコンテナが増加し、より多くのトラフィックを自動的に分散処理することができる。

Cloud Runは、デフォルト設定では、HTTPリクエストを受信して処理している間のみCPUが動作し、リクエストを処理していない時は動作を停止するため、HTTPリクエストを実際に受信し処理するために動作したCPU時間分のみが課金対象となる。この点が、常時稼働が前提となることが多いAWSのEC2やGCPのGCEとは異なる。

（2023年4月現在）PLATEAU VIEW 2.0のPLATEAU CMSのAPIサーバーは、メモリ4GB・CPU2コアの設定で動作している。

（3）ワーカーサーバー（Cloud Run）

PLATEAU CMSにおける非同期バックグラウンド処理を行うサーバーであり、HTTPサーバーとしてAPIサーバーからのリクエストを受信している。zipファイル等の解凍処理やwebhookの送信などをCloud TasksやCloud PubSubと連携しながら行っている。

ワーカーサーバーは、APIサーバーと同じく、Cloud Run 上で動作する。

（2023年4月現在）PLATEAU VIEW 2.0のPLATEAU CMSのWorkerサーバーは、メモリ32GB・CPU8コアの設定で動作している。

（4）サイドカーサーバー（Cloud Run）

PLATEAU CMSと連携しPLATEAU CMSを補助する形で動作するサーバー。外部サービスとの連携や、PLATEAU SDK for Unity/Unreal向けのデータ配信、PLATEAU Editor・PLATEAU VIEWで凡例等の表示を行うサイドバーのデータ等を扱う。（サイドバーについては、3.2.7で詳細を解説）以下の機能を持つ。

- FMEへの品質検査・データ変換処理のリクエスト・結果の受信と保存
- G空間情報センターへのデータ登録処理・カタログ検査
- PLATEAU SDK for Unity/Unreal向けCityGMLデータ配信API
- PLATEAU VIEWの建築検索機能向け検索インデックスの構築
- その他データのランドマーク・鉄道駅・行政界データのCZMLへの変換
- PLATEAU Editor・PLATEAU VIEW向けAPI
 - データカタログAPI
 - サイドバー設定・共有URLデータの保存
 - ご意見ご要望を受け取りSendGridと連携してメール送信

サイドカーサーバーは、APIサーバーと同じく、Cloud Run 上で動作する。

（2023年4月現在）PLATEAU VIEW 2.0のPLATEAU CMSのPLATEAU VIEWサーバーは、メモリ16GB・CPU4コアの設定で動作している。

（５）ファイルストレージ（Google Cloud Storage）

フロントエンドのアプリケーションのソースコードや画像、ユーザーによってアップロードされたアセットファイルを保存する、オブジェクトストレージサーバー。

Google Cloud Storage（GCS）を使用しており、容量は無制限であり、自動的にスケーリングし、バックアップも自動的に行われ、データは複数拠点に分散配置される。巨大なファイルを格納・配信することが可能。

PLATEAU CMSでは、PLATEAU関連データセット等の静的ファイルをGCSに保存している。

（６）MongoDB

MongoDBとは、ドキュメント指向のNoSQLデータベースで、データの柔軟性と拡張性が特徴。PLATEAU CMSはデータベースとしてMongoDBを使用している。

PLATEAU CMSでは、MongoDB社が提供するクラウドサービス MongoDB Atlas を利用している。保守運用が自動化されたマネージドなMongoDBが利用可能で、自動的に3台以上のサーバーから成るクラスタを構成し、データベース内のデータは自動的に各サーバーに複製され、リクエストは分散処理されるようになっている。M0からM30まで、様々なマシンスペックのサーバーによる可用性の高いクラスタを構築することができ、マシンスペックによって料金が変わる。

（2023年4月現在）PLATEAU VIEW 2.0のPLATEAU CMS向けには、M10クラスタを運用しており、PLATEAU Editorと同じクラスタを使用している。

MongoDBには、ユーザーやワークスペース、スキーマ、コンテンツ（データカタログに表示される説明文等）に関する情報が保存される。

（７）Auth0

Auth0社が提供するクラウドサービス。アカウントの管理・認証・認可を行うIDプロバイダを提供する。Auth0を使用することで、開発者はアプリケーションに安全で使いやすく信頼性の高い認証認可機能を組み込むことができる。PLATEAU EditorもAuth0を使用して認証認可機能を実現している。なお、PLATEAU CMSとPLATEAU Editorでは同じAuth0テナントを利用している。

（2023年4月現在）テナントに対して登録されているユーザー数に応じて課金されるが、7000ユーザーまでは無料となっている。

（８）SendGrid

Twilio社が提供するクラウドサービス。クラウドベースのメール配信プラットフォームで、企業や開発者がアプリケーションやWebサイトから大量のメールを配信するために使用される。PLATEAU CMSではご意見ご要望のメール送信で使用している。

（９）FME Server

Safe Software Inc.（カナダ）が開発したデータ変換エンジンで、データの品質検査や、CityGMLから3D TilesやMapbox Vector Tilesへのデータ変換などを行う。詳細は「1.3 FMEサーバ」を参照されたい。

（１０）G空間情報センター

官民間問わず様々な主体により整備・提供される多様な地理空間情報を集約し、利用者がワンストップで検索・ダウンロードし利用できる、産学官の地理空間情報を扱うプラットフォーム。

PLATEAU CMSでは、PLATEAUデータの公開時（公開申請承認時）に自動的にG空間情報センターへ登録を行う。

(11) その他のコンポーネント

表 その他のコンポーネント

コンポーネント名	説明
Cloud CDN	GCPのCDN（Content Deliver Network = ウェブコンテンツをインターネット経由で配信するために最適化されたネットワーク）。GCSなどと組み合わせて使用することで、リクエスト元から地理的に近いサーバーにコンテンツのキャッシュを自動的に配置し、コンテンツ配信を高速化・効率化させることができる。
Cloud DNS	GCPのDNS。PLATEAU VIEW 2.0のPLATEAU Editorで使用しているドメインに対応するレコードは全てCloud DNSにて管理されている。
Cloud Load Balancing	GCPのマネージドなロードバランサ（負荷分散システム）。PLATEAU VIEW 2.0のPLATEAU Editorで使用されるドメインのIPアドレスはすべてCloud Load Balancingに向いており、リクエストのホスト（ドメイン）に応じてAPIサーバーやストレージサーバーに自動的にルーティングされる。
Cloud Logging	GCPのログ収集サービス。Cloud Runなどから出力されるログを閲覧可能。システムのトラブルシューティング時に役立つ。
Cloud PubSub	GCPのメッセージングサービス。アプリケーション間の連携を行うために利用される。PLATEAU CMSでは、APIサーバーとワーカーサーバー間の通信に使用している。
Cloud Tasks	GCPのキューイングサービス。大量の分散タスクの実行を管理できる。PLATEAU CMSではワーカーサーバーで行われる解凍処理のキューイング及び再試行処理制御のために使用している。

1.4 FMEサーバ

FMEサーバでは、PLATEAU CMSに登録された3D都市モデル（CityGML形式）データ（標準製品仕様書第2.x版（以下「製品仕様書」という）に基づいて整備）について、製品仕様書に規定される品質要求のうち全数・自動検査を行うべきとされる事項を中心とした検査を行う。また、本検査に合格したデータをPLATEAU VIEWで表示できる形式（3DTilesおよびMVT形式）に変換する。なお、FMEサーバにはSafe Software Inc.の製品、FME Serverを用いた。

1.4.1 サーバ構成

FMEサーバでは、AWSのサービスであるAmazon EC2（Amazon Elastic Compute Cloud）を利用して、FME Serverを稼働させた。

表 AWS上のサーバ環境

項目	内容
サービス名	Amazon EC2（Amazon Elastic Compute Cloud）
OS	Linux（Ubuntu Server 20.04 LTS (HVM)）
インスタンスタイプ	t2.large
メモリ	64.0 GiB
ストレージ	500GB
リージョン	ap-northeast-1（東京）

1.4.2 データ構成

CityGML品質検査では、標準製品仕様書の要件に沿って圧縮（ZIP）されたファイルをFMEサーバにアップロードして処理を実行し、検査結果（JSON）を出力する。

3DTiles変換では、上記の品質検査で合格したCityGMLファイルを読み込んで処理を実行し、変換後のファイル（3DTiles、MVT）を出力する。

表 変換フォーマット

#	変換対象フィーチャータイプ	変換先データフォーマット
1	建築物	3D Tiles
2	道路（交通領域、交通補助領域を含む）	MVT (LOD1, LOD2) , 3D Tiles (LOD3)
3	都市設備	3D Tiles
4	植生	3D Tiles
5	浸水想定区域（洪水、津波、高潮、内水）	3D Tiles
6	土地利用	MVT
7	都市計画決定情報	MVT
8	土砂災害警戒区域	MVT

1.5 PLATEAU Editor ・ PLATEAU VIEW

1.5.1 PLATEAU Editor ・ PLATEAU VIEWとは

1.1のシステム全体構成で述べたように、PLATEAU VIEW 2.0のシステムの機能は、大きく「データの管理」と「データの可視化」に分かれる。

PLATEAU EditorとPLATEAU VIEWは、このうち「データの可視化」を担うシステムのことを指す。具体的にはこの可視化システムは、更に以下のように分かれる。

PLATEAU VIEW

- ユーザーが、PLATEAUデータを初めとする様々なGISデータの可視化を、Web上のデジタル3D地球儀上で行うことができる、Webアプリケーション。下記に述べるPLATEAU Editorによって自動的に管理・運用されている。

PLATEAU Editor

- 上記のPLATEAU VIEWそのものを作成および公開するための、Webアプリケーション。目的に応じて様々なWebアプリケーションを作成・公開することが可能であるが、PLATEAU VIEW 2.0では、1つのWebアプリケーションをPLAYEAU Editor経由で一般向けに公開し、それをPLATEAU VIEWと呼称している。

(1) 使用ソフトウェア・サービス

上記システムを構築するために、オープンソースソフトウェア（OSS）と、有償のクラウドサービスを組み合わせて利用している。クラウドサービスの詳細については次項の各コンポーネントの説明にて述べる。

表 使用ソフトウェア・サービス一覧

項目	項目	説明
Re:Earth	OSS	Eukarya社が開発・公開しているOSS。CesiumJSを内包しており、ノーコードで地図やデジタル地球儀を使用したWebアプリケーションを作成・公開することができるWebアプリケーション。プラグインによる機能拡張にも対応。
Cesium Ion	有償クラウドサービス	Cesium社が提供するクラウドサービス。地球儀上の日本における地表面を表現する3Dのデータ（テラインデータ）を配信するために使用している。
Auth0	有償クラウドサービス	Auth0社が提供するクラウドサービス。アカウントの管理・認証・認可を行うIDプロバイダを提供する。Auth0を使用することで、開発者はアプリケーションに安全で使いやすく信頼性の高い認証・認可機能を組み込むことができる。PLATEAU EditorもAuth0を使用して認証・認可機能を実現している。
Google Cloud Platform (GCP)	有償クラウドサービス	Google社が提供するクラウドコンピューティングプラットフォーム。PLATEAU CMSを動作させるためのサーバーや、ファイルを保存するためのサーバーをGCP上に構築している。
MongoDB Atlas	有償クラウドサービス	MongoDB社が提供するクラウドサービス。保守運用が自動化されたマネージドなMongoDBを提供している。MongoDBとは、ドキュメント指向のNoSQLデータベースで、データの柔軟性と拡張性が特徴。PLATEAU CMSはデータベースとしてMongoDBを使用している。

PLATEAU Editorはその他様々な技術を組み合わせて構築されている。

PLATEAU Editorで内部的に利用されている技術・ライブラリ等

項目	説明
Go	プログラミング言語の1つで、Googleによって開発された。構文がシンプルであり、かつ処理が高速な言語で、主にバックエンド開発に用いられる。PLATEAU Editorのサーバー実装に利用されている。
TypeScript	プログラミング言語の1つで、Microsoftによって開発された。JavaScriptに静的型付けを加えたスーパーセットであり、大規模システムの開発に用いられる。PLATEAU Editorではフロントエンド向け開発に利用されている。
React	Meta(旧Facebook)によって開発されたUI構築のためのJavaScriptライブラリ。特に大規模かつ複雑なUI実装において利用される。PLATEAU Editorではフロントエンド向け開発に利用されている。
CesiumJS	デジタル3D地球儀上に様々な情報を描画することができる地図エンジン。Webブラウザ上で動作し、WebGLを用いて描画を行うため、PCやスマートフォンで閲覧することができる。PLATEAU Editor上でデータを表示するために使用している。
Resium	React上でCesiumJSを手軽に利用可能にするコンポーネントを提供するライブラリ。Eukarya開発。PLATEAU Editor上でCesiumJSと共に使用している。
GraphQL	API向けに作られたクエリ言語およびランタイムを指す。WebAPIの開発において、RESTなどの方式と比較して、より柔軟かつ効率的なAPIの提供を可能にする。PLATEAU Editorではバックエンドとフロントエンド間の通信に利用されている。

（２）PLATEAU Editorの主な機能

PLATEAU Editorでは主に以下の機能が利用可能である。詳しい使い方は、3.3を参照されたい。

- ワークスペースの作成・ユーザーの招待
- プロジェクトの作成
- レイヤーの配置・スタイルの設定
- GISデータの読み込み・表示
- シーンの設定変更（ベースマップ・テライン・カメラなどの各種設定）
- インフォボックス（レイヤーの詳細を表示する画面領域）の作成・編集
- ウィジェット（地球儀の上に表示される様々なUI）の配置・編集
- プラグインのインストール（ウィジェットなどを機能拡張可能）
- プロジェクトの公開

（３）PLATEAU VIEWの主な機能

PLATEAU VIEWでは主に以下の機能が利用可能である。詳しい使い方は、3.4を参照されたい。

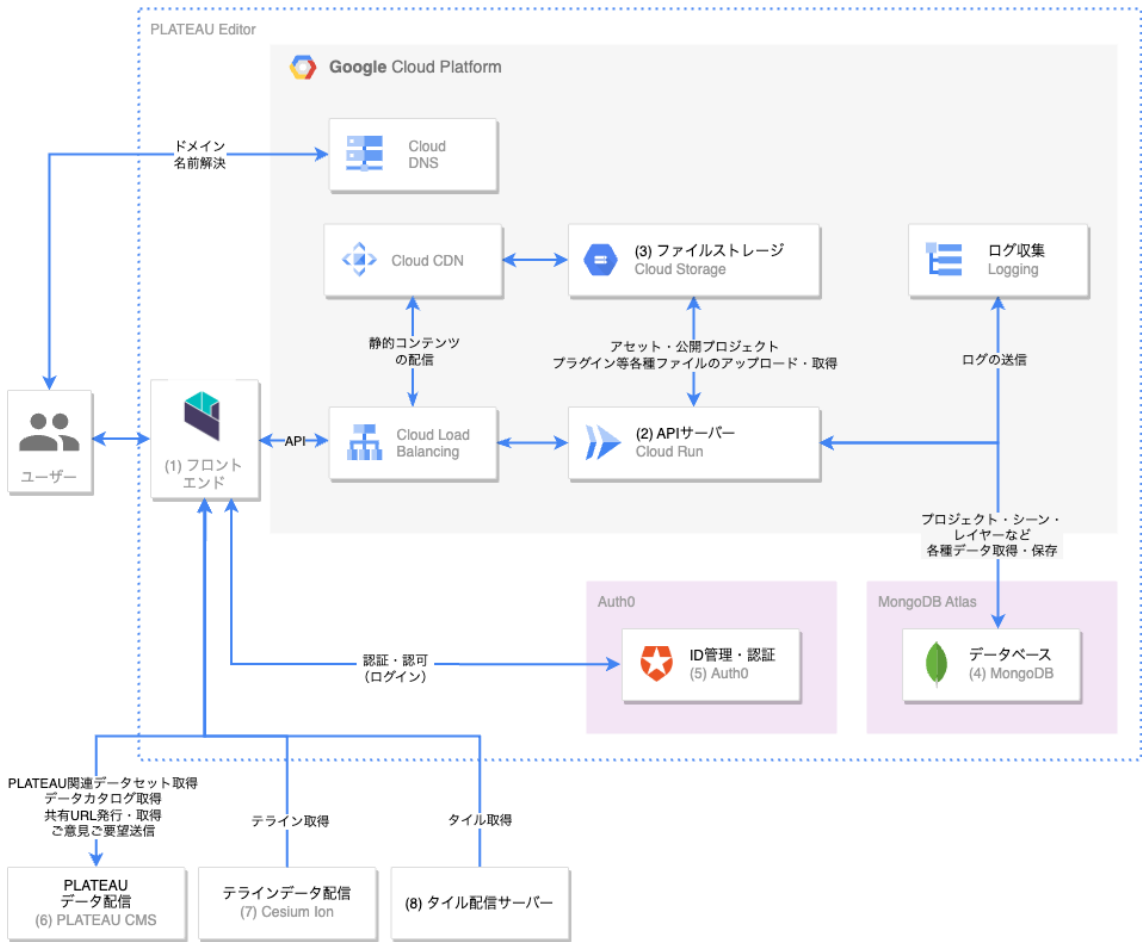
- データカタログ
 - データセットの一覧の閲覧
 - データセットの地図上への追加
 - Myデータ（ファイルやURLからデータの追加）
- データセットの凡例の表示・絞り込み表示などの操作・地物の属性表示
- 地図の設定の変更（3D/2Dの切り替え・ベースマップの切り替え）
- タイムラインによる時系列データの表示
- ストーリーテリングの表示と編集
- ナビゲーターによるカメラ操作・歩行者モード・現在地の表示
- 建築物検索機能・建築物クリップ機能
- 共有URLの発行・画面の印刷
- ご意見ご要望

1.5.2 システム構成の全体像

PLATEAU VIEW 2.0におけるPLATEAU Editorのシステム構成について説明する。

なお、PLATEAU VIEWとは、PLATEAU Editorによって作成・公開され、PLATEAU Editor上で動作するWebアプリケーションであるため、以下の図には記載されていないことに注意されたい。

表 システム構成



以下、上記構成図の各要素についてそれぞれ説明する。

なお、各コンポーネントは、保存データ量、ネットワーク転送量、CPU時間、マシンスペックなどに基づいて料金が発生する。

(1) フロントエンド

Webブラウザ上で動作する、(HTML・CSS・JavaScriptによる) フロントエンドアプリケーション。APIサーバー・Auth0・その他各種サーバーとの通信を行い、UIや地図を表示する。

（2）APIサーバー（Cloud Run）

PLATEAU EditorのAPIサーバーであり、HTTPサーバーとして外部からのリクエストを受信している。MongoDBやファイルストレージと連携して、レイヤーの保存やプロジェクトの管理・公開などの、様々なビジネスロジックを実行する。

APIサーバーは、Cloud Run 上で動作する。Cloud Run とは、GCPで利用可能なサーバーレス（CaaS）プラットフォームであり、Dockerコンテナをデプロイすることで、サーバーの保守管理の手間なしに、アプリケーションをクラウド上で動作させることができる。同時接続リクエスト数が規定数以上に達すると自動的にコンテナが増加し、より多くのトラフィックを自動的に分散処理することができる。

Cloud Runは、デフォルト設定では、HTTPリクエストを受信して処理している間のみCPUが動作し、リクエストを処理していない時は動作を停止するため、HTTPリクエストを実際に受信し処理するために動作したCPU時間分のみが課金対象となる。この点が、常時稼働が前提となることが多いAWSのEC2やGCPのGCEとは異なる。

（2023年4月現在）PLATEAU VIEW 2.0のPLATEAU EditorのAPIサーバーは、メモリ1GB・CPU2コアの設定で動作している。

（3）ファイルストレージ（Goole Cloud Storage）

フロントエンドのアプリケーションのソースコードや画像、ユーザーによってアップロードされたアセットファイルや、プロジェクト公開時にビルドされる情報、インストールされたプラグインのファイルを保存する、オブジェクトストレージサーバー。

Google Cloud Storage（GCS）というサービスを使用しており、容量は無制限であり、自動的にスケーリングし、バックアップも自動的に行われ、データは複数拠点に分散配置される。巨大なファイルを格納・配信することが可能。

なお、PLATEAU VIEW 2.0のPLATEAU関連データセットなどのGISデータは、主にPLATEAU CMSのストレージサーバーにて保存・配信されており、PLATEAU Editorで保存されているファイルの量はそれに比較してさほど大きくない。PLATEAU VIEW上で可視化されるGISデータは主にPLATEAU CMSから配信されるデータを使用している。

（4）MongoDB

MongoDBとは、ドキュメント指向のNoSQLデータベースで、データの柔軟性と拡張性が特徴。PLATEAU EditorはデータベースとしてMongoDBを使用している。

PLATEAU Editorでは、MongoDB社が提供するクラウドサービス MongoDB Atlas を利用している。保守運用が自動化されたマネージドなMongoDBが利用可能で、自動的に3台以上のサーバーから成るクラスタを構成し、データベース内のデータは自動的に各サーバーに複製され、リクエストは分散処理されるようになっている。M0からM30まで、様々なマシンスペックのサーバーによる可用性の高いクラスタを構築することができ、マシンスペックによって料金が変わる。

（2023年4月現在）PLATEAU VIEW 2.0のPLATEAU Editor向けには、M10クラスタを運用している。

(5) Auth0

Auth0社が提供するクラウドサービス。アカウントの管理・認証・認可を行うIDプロバイダを提供する。Auth0を使用することで、開発者はアプリケーションに安全で使いやすく信頼性の高い認証認可機能を組み込むことができる。PLATEAU EditorもAuth0を使用して認証認可機能を実現している。

(2023年4月現在) テナントに対して登録されているユーザー数に応じて課金されるが、7000ユーザーまでは無料となっている。

(6) PLATEAU CMS

PLATEAU VIEWで利用可能なデータカタログや、PLATEAUデータをはじめとする各種GISデータ等を配信しているシステム。詳しくは1.2を参照されたい。

(7) Cesium Ion

Cesium社が提供するクラウドサービス。PLATEAU VIEW 向けに独自に作成された、地球儀上の日本における地表面を表現する3Dのデータ（テラインデータ）を配信するために使用している。

(8) タイル配信サーバー

PLATEAU VIEWの地図の設定で選択可能なベースマップについて、それぞれ以下のサーバーから配信されるタイルデータを使用している。

タイルデータとは、ユーザーからリクエストされた地図表示範囲に対して、予めタイル状に分割された画像データのことであり、それらをサーバーから配信するサービスを使用することでタイルデータを取得してCesium上で描画している。

- 全国最新写真 (シームレス) : PLATEAU VIEW 向けに独自に作成されたタイルデータ。PLATEAU VIEW 1.1向けに構築されたAWSのS3から配信されている。
- 空中写真 (Bing) : Bing Map
- 地理院地図: 国土地理院サーバーから配信される地理院タイル (国土地理院が配信するタイルデータ)。 <http://maps.gsi.go.jp/development/siyou.html>
- Dark Matter : Carto社から提供されているタイル

(9) その他のコンポーネント

表 その他のコンポーネント

コンポーネント名	説明
Cloud CDN	GCPのCDN（Content Deliver Network = ウェブコンテンツをインターネット経由で配信するために最適化されたネットワーク）。GCSなどと組み合わせて使用することで、リクエスト元から地理的に近いサーバーにコンテンツのキャッシュを自動的に配置し、コンテンツ配信を高速化・効率化させることができる。
Cloud DNS	GCPのDNS。PLATEAU Editorで使用しているドメインに対応するレコードは全てCloud DNSにて管理されている。
Cloud Load Balancing	GCPのマネージドなロードバランサ（負荷分散システム）。PLATEAU VIEW 2.0のPLATEAU Editorで使用されるドメインのIPアドレスはすべてCloud Load Balancingに向いており、リクエストのホスト（ドメイン）に応じてAPIサーバーやストレージサーバーに自動的にルーティングされる。
Cloud Logging	GCPのログ収集サービス。Cloud Runなどから出力されるログを閲覧可能。システムのトラブルシューティング時に役立つ。

1.6 PLATEAU SDK for Unity/Unreal

1.6.1 PLATEAU SDK for Unity/Unrealの概要

PLATEAU SDK for Unity及びPLATEAU SDK for Unrealについて解説する。SDKとは、Software Development Kit (ソフトウェア開発キット)の略であり、両SDKはUnityおよびUnreal Engine上でPLATEAUの3D都市モデルデータを利用するためのツールである。Unity および Unreal Engine とはゲームエンジンの一種であり、ゲーム制作、映像制作、シミュレーションなどの目的で広く利用されているツールである。

このSDKを使用することで、実世界を舞台にしたアプリケーションの開発やPLATEAUの豊富なデータを活用した都市シミュレーションを開発できる。

図 PLATEAU SDK for Unity/Unrealで読み込んだ3D都市モデル

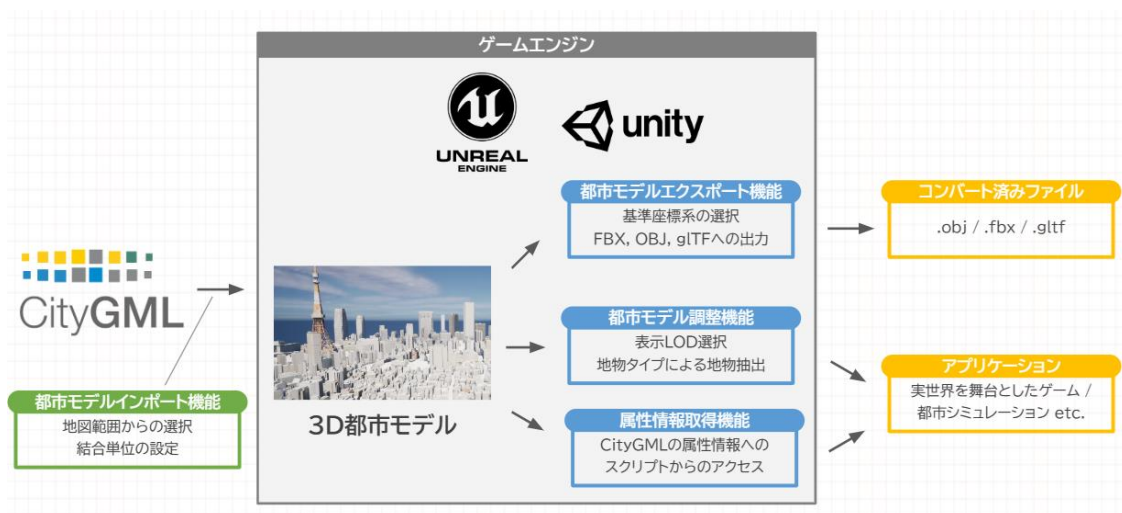


1.6.2 PLATEAU SDK for Unity/Unrealの機能

PLATEAU SDK for Unity/Unrealが提供する機能について解説する。PLATEAU SDK for Unity/Unrealは主に以下の機能を提供している。

- 3D都市モデルインポート機能
- 3D都市モデル調整機能
- 3D都市モデルエクスポート機能
- 属性情報取得機能

図 PLATEAU SDK for Unity/Unrealの機能



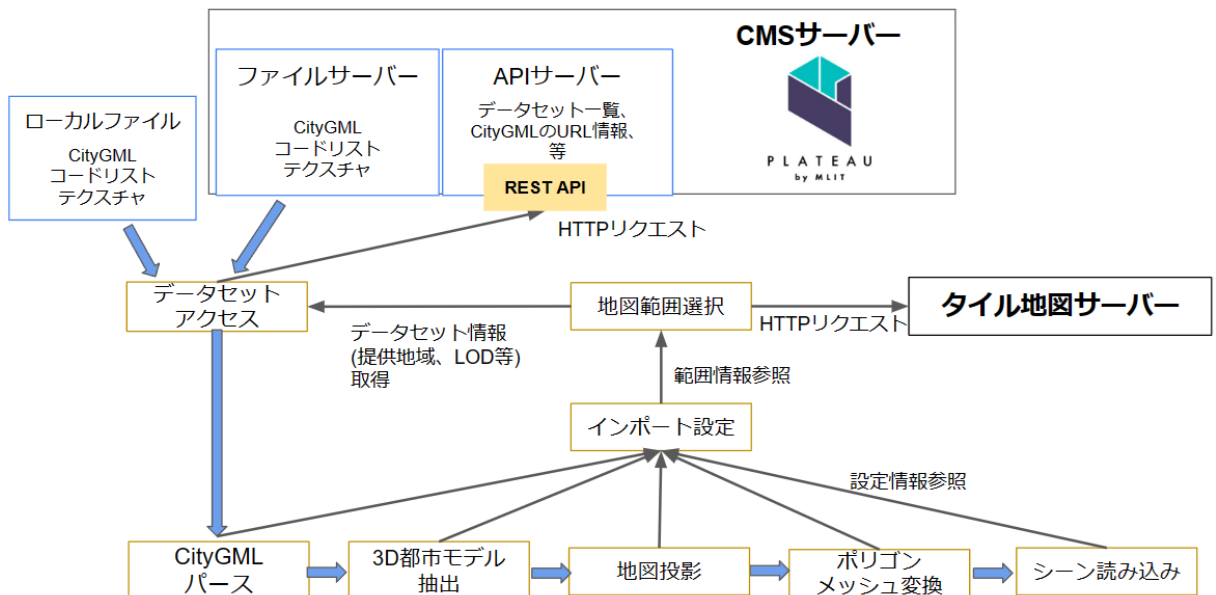
3D都市モデルインポート機能

3D都市モデルインポート機能では、PLATEAUの[3D都市モデル標準製品仕様書](#)（第2.xシリーズ）に準拠しているすべての3D都市モデルデータを入力としてゲームエンジンにインポートできる。

インポート処理は以下の流れで行われる。一連の処理は非同期化されており、ユーザーの操作を妨げない設計になっている。

1. データセットへのアクセス
2. データセットで提供されているCityGML、コードリスト、テクスチャファイルの情報やCityGMLの提供地域、LOD、地物種類の情報を取得する。ローカルからのインポートの場合はファイル検索によって情報が取得され、サーバーからのインポートの場合はAPIサーバー経由で情報が取得される。
3. インポート設定
インポート設定はPLATEAU SDKのUIから指定される。UI上ではインポートする地物種類の選択、LODの選択、テクスチャの有無、範囲の選択が可能である。範囲の選択には専用のGUIが提供されており、画面内ではデータセット情報とタイル地図サーバーから取得された平面地図が表示される。以下の処理ではここで設定された内容がパラメータとして使用される。
4. 3D都市モデル抽出
インポート時に設定された座標範囲内の3D都市モデルを抽出する。範囲選択画面で指定された座標最小値・最大値がパラメータとして入力され、その範囲内の地物が抽出される。
5. 平面直角座標への座標投影
CityGMLの各座標は緯度経度で保持されている。ゲームエンジン内では直交座標で座標値を扱う必要があるため、各座標は平面直角座標に変換される。また、この際に各地物がゲームエンジンのシーン内で原点近くに表示されるようにするため、選択範囲の中心にオフセットされる。このオフセット値はインポート設定UIで自由なオフセット値で上書きすることもできる
6. CityGMLの形状データのポリゴンメッシュへの変換
CityGMLの形状データはそのままの形式ではゲームエンジンで可視化ができないため、インポート時にポリゴンメッシュに変換される。ポリゴンメッシュに変換する際の結合単位は主要地物単位（建築物、道路等）、最小地物単位（壁面、屋根面等）、地域単位（100mx100mの範囲の地物をすべて結合）から選択できる
7. シーン読み込み
ポリゴンメッシュをゲームエンジン内のMeshオブジェクトとして可視化する。

図 インポート処理の概要



3D都市モデル調整機能

3D都市モデル調整機能では、インポート機能でシーンにインポートされた3D都市モデルの見た目の調整が行える。具体的には、以下の設定によって各地物の表示・非表示の切り替えができる。

- LOD (CityGMLでの地物の形状の詳細度)
 - 各地物について最大のLODのみを表示するか全てのLODを表示するかを選択も可能
- 地物タイプ (建築物、道路、起伏等)

各地物のLOD情報はゲームエンジンのシーン内でのオブジェクト構造から取得している。インポートされた3D都市モデル内の各地物は、[LOD0]、[LOD1]等の名前のオブジェクトを親として持つため、親オブジェクトを参照することでLODの値を取得できる。地物タイプの取得にはインポート元となったCityGMLファイルを参照している。CityGMLファイル名もLODと同様、ゲームエンジン内のオブジェクトの名前に情報が格納されており、LODのオブジェクトの更に親オブジェクトを辿ることで参照が可能になっている。

3D都市モデルエクスポート機能

3D都市モデルエクスポート機能ではインポートされた3D都市モデルを3Dファイル形式でエクスポートできる。以下の出力設定ができる。

- ファイル形式
 - OBJ, FBX, glTFに対応している。
- テクスチャ
 - 出力するメッシュにテクスチャを含めるかどうかを選択する。
- 非アクティブオブジェクトを含める
 - 3D都市モデル調整機能で非表示に設定されたオブジェクトを含めるかどうかを選択する。
- 座標変換
 - 出力座標系をLocal (Unityでの座標) とPlane Cartesian (平面直角座標) から選択する。
- 座標軸
 - 出力されるメッシュの座標軸を設定する。

属性情報取得機能

属性情報取得機能ではCityGMLに含まれるすべての属性情報にアクセスできるAPIを提供している。インポートされた3D都市モデルのオブジェクト構成にはCityGMLファイル名と地物ID情報が名前として含まれている。地物に対応するオブジェクトの属性情報を取得したい場合、まずCityGMLファイル名を入力としてCityGMLをパースするAPIを呼び出す。さらに得られた3D都市モデルの情報と地物IDを入力として地物を検索するAPIを呼び出すことで付随する属性情報を取得できる。

属性情報取得機能の詳細な情報については各PLATEAU SDKのマニュアルを参照されたい。

<https://project-plateau.github.io/PLATEAU-SDK-for-Unity/manual/AccessCityObject.html>

<https://project-plateau.github.io/PLATEAU-SDK-for-Unreal/manual/AccessCityObject.html>

図 CityGMLとUnityでのオブジェクトの対応

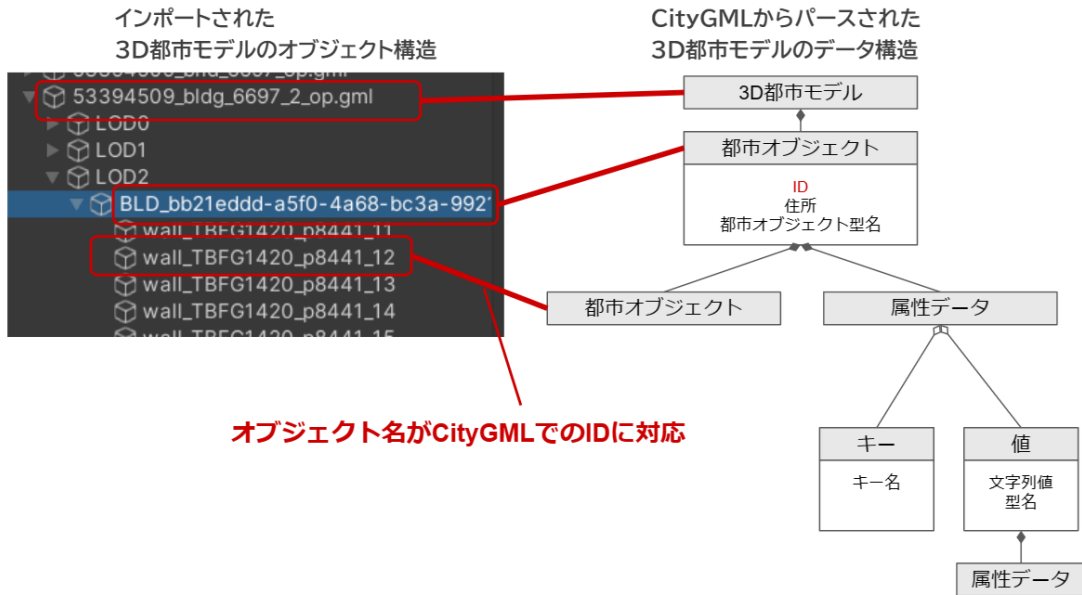
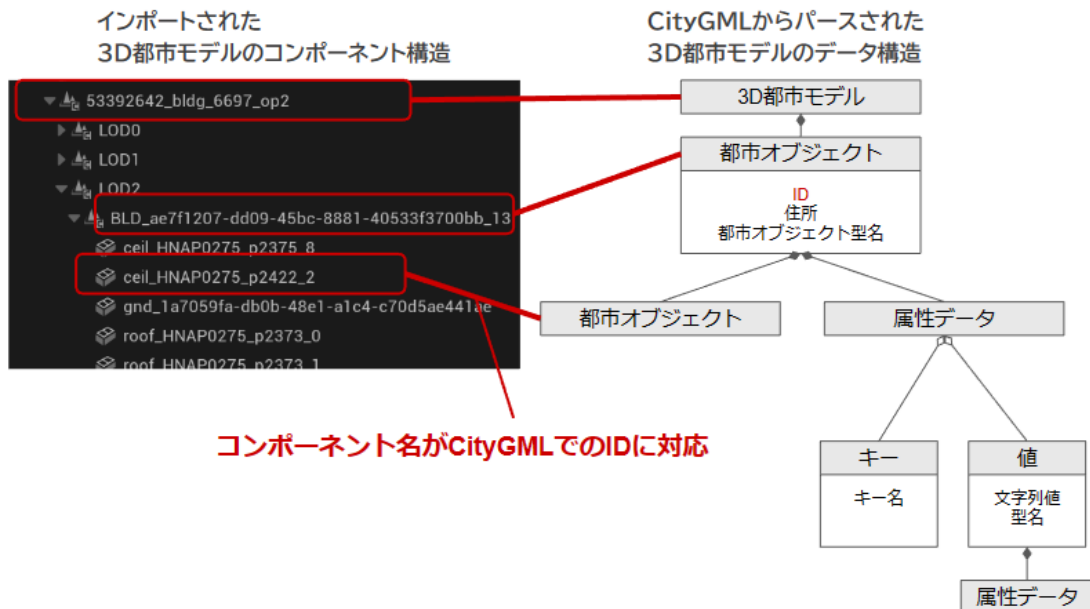


図 CityGMLとUnreal Engineでのオブジェクトの対応



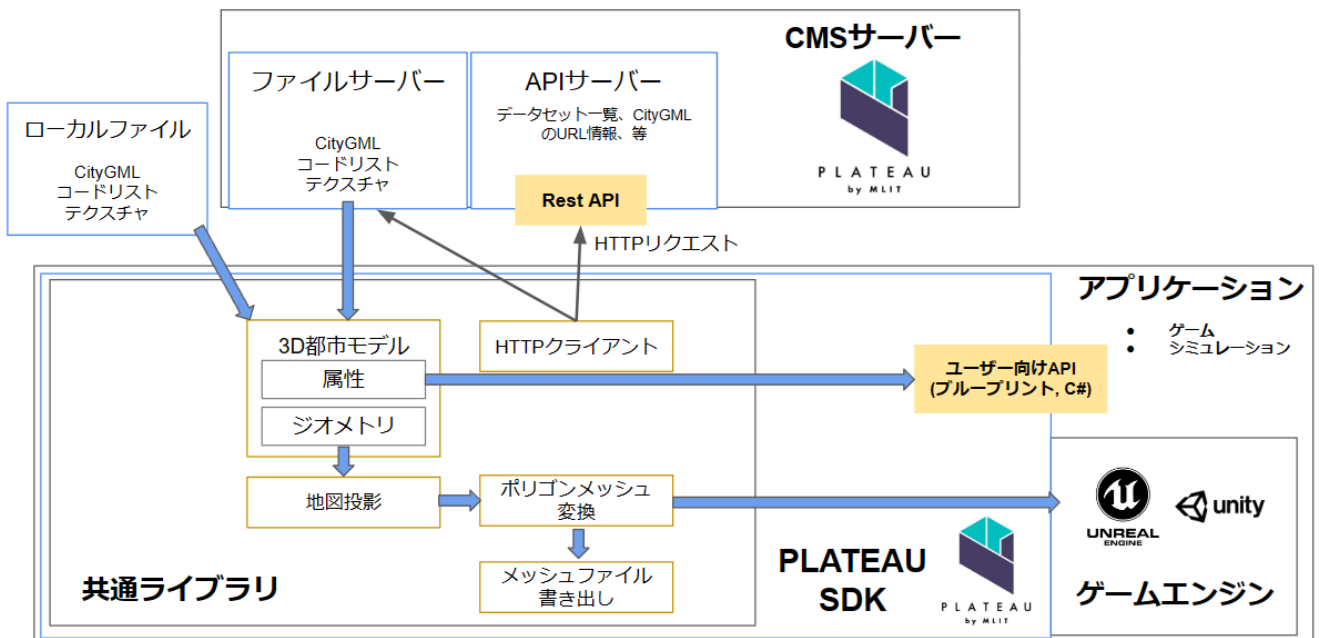
1.6.3 PLATEAU SDK for Unity/Unrealのアーキテクチャ

PLATEAU SDK for Unity/Unrealは以下のコンポーネントによって構成されている。

- PLATEAU SDK for Unity
 - SDKのUnity向けプラグイン
- PLATEAU SDK for Unreal
 - SDKのUnreal Engine向けプラグイン
- 共通ライブラリ
 - Unity、Unreal向けSDKが共通で利用するロジックを提供するライブラリ
 - 内部実装はC++で記述されているが、Unity向けにC#のラッパー実装も提供している。

本PLATEAU SDK for Unity/Unrealに固有のロジックは共通ライブラリで実装されており、Unity、Unreal向けSDKでは主にゲームエンジンとのインテグレーション処理が実装されている。

図 PLATEAU SDK for Unity/Unrealのアーキテクチャ



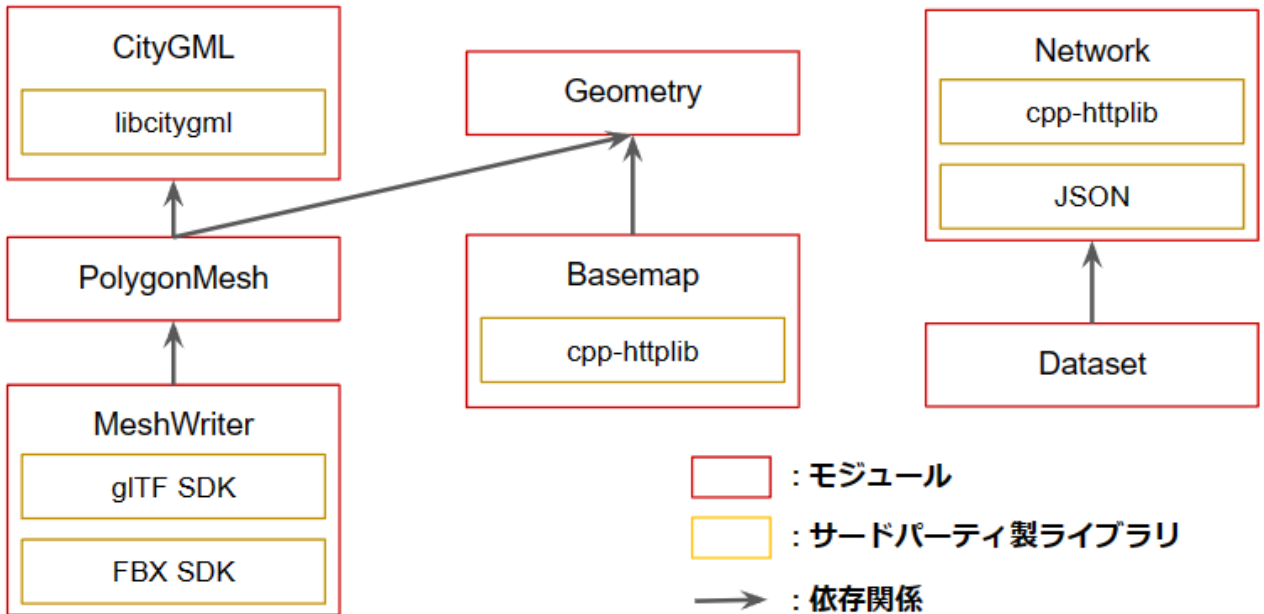
(1) 共通ライブラリのモジュール構成

共通ライブラリは以下のモジュールによって構成されている。

表 共通ライブラリのモジュール

モジュール名	説明
CityGML	CityGMLのパーズ処理・パーズ結果へのアクセスを行うAPIを提供する。
PolygonMesh	パーズされたCityGMLに含まれるジオメトリ情報のポリゴンメッシュ化・生成されたポリゴンメッシュへのアクセス・ポリゴンメッシュ化する際のマージ単位や抽出範囲などの設定を行うAPIを提供する。
Geometry	ポリゴンメッシュを入力として、FBX、glTF、OBJ形式へのエクスポート処理・エクスポートする際の座標変換やテクスチャの有無等の設定を行うAPIを提供する。
Basemap	XYZタイル形式の地図の読み込みを行うAPIを提供する。
Network	REST APIを利用してPLATEAU CMSサーバーにアクセスするAPIを提供する。
Dataset	PLATEAUの3D都市モデル標準製品仕様書（第2.xシリーズ）に準拠するデータセットへのアクセス（PLATEAU CMSサーバーに存在するデータセットの一覧の取得、提供されるLOD・地物の種類の取得、CityGMLのダウンロード等）を行うAPIを提供する。

図 共通ライブラリのモジュール構成



(2) 共通ライブラリが利用するサードパーティ製ライブラリ

共通ライブラリでは以下のサードパーティ製ライブラリを利用している。

表 共通ライブラリが利用するサードパーティ製ライブラリ

ライブラリ名	説明
libcitygml	https://github.com/jklimke/libcitygml CityGMLのパーズ、可視化のためのオープンソースのライブラリ。CityGML 2.0に準拠している。共通ライブラリではPLATEAUでのCityGMLの拡張仕様に 対応するためにソースコードを改変して利用している。
cpp-httpplib	https://github.com/yhirose/cpp-httpplib HTTP通信のためのオープンソースのライブラリ。XYZタイルへのアクセスと PLATEAU CMSサーバーへのアクセスに利用している。
JSON	https://github.com/nlohmann/json JSONのシリアライズ、デシリアライズのためのオープンソースのライブラリ。 PLATEAU CMSサーバーから得られたHTTPレスポンスのパーズに利用している。
glTF SDK	https://github.com/microsoft/glTF-SDK Microsoft社が提供するglTFのシリアライズ、デシリアライズのためのオープン ソースのライブラリ。glTF形式でのエクスポートに利用している。
FBX SDK	https://www.autodesk.com/developer-network/platform-technologies/fbx-sdk-2020-3-1 Autodesk社が提供するFBXのシリアライズ、デシリアライズのためのライブラ リ。FBX形式でのエクスポートに利用している。再配布が禁止されているため、 共通ライブラリ本体に静的リンクする形で組み込まれている。

第1編 PLATEAU VIEW 2.0

第2章 実証環境の構築手順

2.1 PLATEAU CMS・PLATEAU Editorの環境構築

2.1.1 システム構成の全体像

本章では、システム構築を担当するエンジニアを想定読者として、PLATEAU CMSおよびPLATEAU Editorのシステムの構築を主にGCP上で行うための手順を提供する。

なお、（2023年3月現在）PLATEAU CMSおよびPLATEAU EditorはオンプレミスやAWSでの運用には対応していない（PLATEAU EditorおよびPLATEAU CMSがPubSubやCloud Taskに依存・S3をサポートしていないため）。

本章で構築するシステム構成の全体像は、1.2・1.4を参照。ただし、MongoDB Atlas、Auth0、Cloud DNSは両者で同じものを共用する。

2.1.2 各種サービスのセットアップ

前項で示したシステムを構築するためのセットアップ手順を示す。事前に以下の準備が必要である。

- クレジットカード（GCPなどで使用）
- 本システムで使用するドメイン（管理画面にアクセスしてネームサーバーの変更ができること）

本項で示す各種コマンドは、特に注記がない限り、Windows（PowerShell）とMacOS/Linuxの両方に対応している。注記がある場合はそれに従うこと。

（1）Auth0のセットアップ

Auth0（<https://auth0.com/jp>）にアクセスしてAuth0のアカウントを作成し、テナントを開設する。ドメイン名・リージョンは問わない。なお（2023年3月現在）東京リージョンを選ぶことができる。

次に、公式のQuick Startを参考に、Auth0 Management APIのセットアップを行う（「Configure the Provider」の項の前まで）。

<https://github.com/auth0/terraform-provider-auth0/blob/main/docs/guides/quickstart.md>

注意：PLATEAU Editorなどへのログインに失敗する原因になるため、今この時点ではまだ自身のユーザーをAuth0のテナント内に作成しないこと。

後のステップで必要となる情報は以下の通りなので確認しておくこと。

- ドメイン
- クライアントID
- クライアントシークレット

後で使用するので以下のコマンドを実行して、クライアントシークレットを変数に出力しておく。

```
export AUTH0_CLIENT_SECRET="<クライアントシークレット>"
```

(2) MongoDB Atlas のセットアップ

MongoDB Atlas (<https://www.mongodb.com/ja-jp/atlas>) にアクセスしてアカウントを作成し、プロジェクト及びクラスタを作成する。なお有料のクラスタを作成する場合は、クレジットカードの登録が必要である。名前やリージョンなどは自由。




プロジェクトの「Database Access」の設定を確認し、データベースのユーザーが作成されていることを必ず確認する。作成されていない場合は、「ADD NEW DATABASE USER」ボタンをクリックして、任意のパスワードなどの認証方式でユーザーを作成する。そのユーザーに対して、少なくとも読み書き（readWrite）の権限を許可するロール（Read and write to any database・Atlas Admin など）を設定する必要がある。

Database Access

Database Users

Custom Roles

+ ADD NEW DATABASE USER

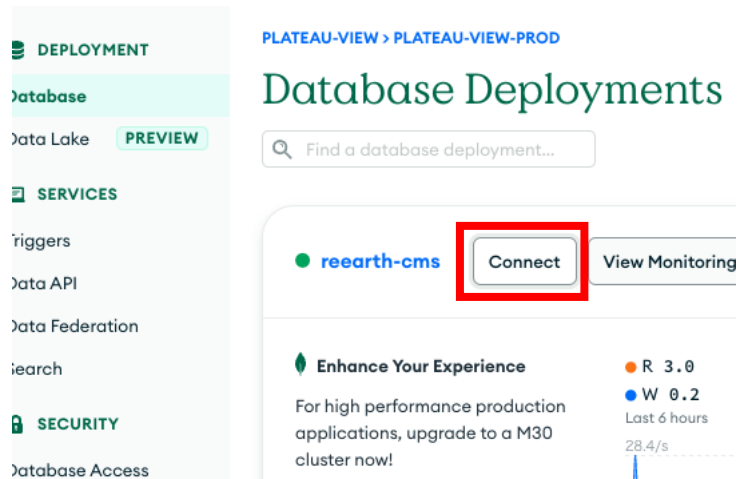
User Name ↕	Authentication Method ▲	MongoDB Roles	Resources	Actions
 reearth-plateau-dev	SCRAM	readWriteAnyDatabase@admin	All Resources	<div> EDIT</div> <div> DELETE</div>

次に、プロジェクトの「Network Access」の設定を確認し、全てのIPアドレスからの接続が許可されていることを必ず確認する。許可されていない場合は、「ADD IP ADDRESS」ボタンをクリックし、Access List Entryで「0.0.0.0/0」と入力してエントリーを追加する。

Network Access

IP Access List		Peering	Private Endpoint
+ ADD IP ADDRESS			
You will only be able to connect to your cluster from the following list of IP Addresses:			
IP Address	Comment	Status	Actions
0.0.0.0/0 (includes your current IP address)		● Active	EDIT DELETE

クラスタを作成後、クラスタ名の右にある「Connect」ボタンをクリックし、「Connect your application」からデータベース接続URLを取得する。なおデータベース接続URLは慎重に扱い、他人には共有しないこと。



後で使用するので以下のコマンドを実行して、データベース接続URLを変数に出力しておく。URL中の <password> を先ほど作成したユーザーのパスワードに置き換えることを忘れないよう注意。

```
export REEARTH_DB="<データベース接続URL>"
```

(3) Google Cloud Platform のセットアップ

Google Cloud Platform (<https://console.cloud.google.com/?hl=JA>) にアクセスし、Googleアカウントでログインし、プロジェクトを作成する。プロジェクトには予めクレジットカードを登録し紐付けておく必要がある。

後で使用するので以下のコマンドを実行して、プロジェクトIDを変数に出力しておく。

```
export PROJECT_ID="<GCPプロジェクトID>"
```

(4) ドメイン・プレフィックス名の決定

今回構築するシステムで使いたいドメインを変数に出力しておく。後のステップでドメインのDNS設定を変更するため、予めドメインの確保を済ませておくこと。

例：plateauview.example.com

```
export DOMAIN="<ドメイン>"
```

次に今回構築するシステムで使用するプレフィックス名を決め、変数に出力しておく。

プレフィックス名とは、GCP内に様々なリソースを作成するときの名称の先頭に付加される任意の文字列である。20文字以内で半角英数ハイフンが使用可能。例えばステージング環境と本番環境で変えるなどの用途がある。管理者にしか見えないので自由に決めて良く、GCPプロジェクトIDと同じにしても問題ない。

例：plateauview-test

```
export SERVICE_PREFIX="<プレフィックス名>"
```

2.1.3 ツールのインストール

(1) gcloud コマンド

以下の説明に従ってgcloudコマンドのインストールを行う。

<https://cloud.google.com/sdk/docs/install?hl=ja>

インストール後、ターミナル上でgcloudコマンドを使用してGoogleアカウントでログインを行い、初期設定を済ませる。

```
gcloud components update
gcloud config configurations create ${SERVICE_PREFIX}
gcloud config set project ${PROJECT_ID}
gcloud auth login

# GCPのAPIの有効化。中には完了まで少し時間を要するものもある。
gcloud services enable certificatemanager.googleapis.com
gcloud services enable secretmanager.googleapis.com
gcloud services enable cloudbuild.googleapis.com
gcloud services enable cloudresourcemanager.googleapis.com
gcloud services enable cloudtasks.googleapis.com
gcloud services enable compute.googleapis.com
gcloud services enable dns.googleapis.com
gcloud services enable iam.googleapis.com
gcloud services enable run.googleapis.com
gcloud services enable sts.googleapis.com

gcloud config set compute/region asia-northeast1
gcloud auth application-default login
```

(2) Terraform コマンド

以下の手順に従い、Terraformコマンドをインストールする。

<https://developer.hashicorp.com/terraform/tutorials/aws-get-started/install-cli>

2.1.4 DNSのセットアップ

(1) Cloud DNSのセットアップ

GCPのCloud DNS上にゾーンを作成する。

```
gcloud dns managed-zones create ${SERVICE_PREFIX} --dns-name ${DOMAIN} --description
"${SERVICE_PREFIX}"
gcloud dns record-sets list --zone ${ZONE_NAME}
```

以下のような出力が得られる。

```
NAME                TYPE TTL  DATA
***** NS    21600 ns-cloud-a1.googledomains.com.,ns-cloud-
a2.googledomains.com.,ns-cloud-a3.googledomains.com.,ns-cloud-a4.googledomains.com.
***** SOA    21600 ns-cloud-a1.googledomains.com. cloud-dns-hostmaster.google.com.
1 21600 3600 259200 300
```

(2) NSレコードの設定

今回セットアップしたいドメインをCloud DNSでホスティングできるように、レジストラの設定を変更する。具体的には、上記のように出力されたDNSサーバーの完全修飾ドメイン名を、使用したいドメインのNSレコードに設定する。手順は使用しているレジストラの手順を参照。

例：上記例の場合、NSレコードに4件の完全修飾ドメイン名（FQDN）を設定することが必要。

- ns-cloud-a1.googledomains.com.
- ns-cloud-a2.googledomains.com.
- ns-cloud-a3.googledomains.com.
- ns-cloud-a4.googledomains.com.

2.1.5 Terraformの準備と初回実行

(1) Terraformファイルのダウンロード

GitHubの以下のURLからZipファイルをダウンロードして解凍する。

<https://github.com/Project-PLATEAU/PLATEAU-VIEW-2.0/archive/refs/heads/main.zip>

または、MacOS・Linuxの場合は以下のコマンドでダウンロードと解凍が行える。

```
curl -O https://github.com/Project-PLATEAU/PLATEAU-VIEW-2.0/archive/refs/heads/main.zip
unzip main.zip
cd PLATEAU-VIEW-main/terraform
```

gitコマンドが使える環境の場合は、以下のようにgitコマンドを使っても良い。

```
git clone https://github.com/Project-PLATEAU/PLATEAU-VIEW-2.0.git
cd PLATEAU-VIEW/terraform
```

Windows (PowerShell) の場合は以下のコマンドが使用できる。

```
wget https://github.com/Project-PLATEAU/PLATEAU-VIEW-2.0/archive/refs/heads/main.zip -
OutFile main.zip
Expand-Archive main.zip
cd PLATEAU-VIEW-main¥terraform
```


(2) 環境設定ファイルの作成

今回設定したい環境の設定ファイル（tfvars）を準備する。コメントを参考に、実行環境に合わせて必要な情報のセットアップを行う。

terraformフォルダ内のexample.tfvarsファイルを直接編集し、編集を行い、必要な設定を追記する（別名としてコピーして編集しても良い）。

```
base_domain      = "" # 先ほど決定したドメイン
gcp_project_name = "" # 先ほど取得したGCPプロジェクトID
service_prefix   = "" # 先ほど決めたプレフィックス名
dns_managed_zone_name = "" # Cloud DNS ゾーン名（service_prefixと同じ値で良い）
gcp_region       = "asia-northeast1" # 東京リージョン

auth0 = {
  domain = "" # Auth0のドメイン
}

auth0_provider = {
  client_id = "" # Auth0のクライアントID
  domain    = "" # Auth0のドメイン（上記Auth0ドメイン設定と同じ値で良い）
}

# 以下は任意設定（このままでも先に進める）
plateauview = {
  fme_baseurl      = ""
  fme_skip_quality_check = false
  ckan_base_url    = ""
  ckan_org          = ""
  option_to        = ""
  option_from      = ""
  cms_plateau_project = "plateau-2022"
  cms_system_project  = "plateauview"
}
```

(3) バックエンドの作成

Terraformの状態 (tfstate) を保存する場所のことをバックエンドと呼ぶ。ここでは先ほど作成したGCPプロジェクトにGCS (Cloud Storage) バケットを作成する。

```
gcloud storage buckets create gs://$(SERVICE_PREFIX)-terraform-tfstate
```

terraform.tfファイルのbackendのバケットを設定する。\$(SERVICE_PREFIX)は上記で決めたプレフィックス名に置き換えること。

```
backend "gcs" {  
  bucket = "$(SERVICE_PREFIX)-terraform-tfstate"  
}
```

(4) Terraformの初回実行

```
terraform init
```

次に、以下のコマンドを実行する（tfvarsファイルを別名で作成した場合はそのファイル名にすること）。

```
terraform apply -var-file=env/example.tfvars
```

途中でyesを入力して実行を進めること。しばらく時間を要するが、以下のような出力が得られれば成功。

```
Apply complete! Resources: * added, * changed, * destroyed.
```

Outputs:

```
plateauview_cms_url = "*****"
plateauview_cms_webhook_secret = "*****"
plateauview_cms_webhook_url = "*****"
plateauview_reearth_url = "*****"
plateauview_sdk_token = "*****"
plateauview_sidebar_token = "*****"
plateauview_sidecar_url = "*****"
```

これらのoutputsは後で使う。もう一度outputsを表示したいときは *terraform output* コマンドで表示可能。

- **plateauview_cms_url**: PLATEAU CMSのURL
- **plateauview_cms_webhook_secret**: 下記「2.1.9 PLATEAU CMSのセットアップ」で使用
- **plateauview_cms_webhook_url**: 下記「2.1.9 PLATEAU CMSのセットアップ」で使用
- **plateauview_reearth_url**: PLATEAU EditorのURL
- **plateauview_sdk_token**: PLATEAU SDK for Unity/Unreal用のトークン。PLATEAU SDK for Unity/UnrealのUIで設定する（詳しくは3.4を参照）。
- **plateauview_sidebar_token**: PLATEAU VIEWのサイドバー用のAPIトークン。PLATEAU Editor上でサイドバーウィジェットの設定から設定する（詳しくは「3.3.6 ウィジェットの設定」を参照）。
- **plateauview_sidecar_url**: サイドカーサーバーのURL。PLATEAU Editor上でサイドバーウィジェットの設定から設定する（詳しくは「3.3.6 ウィジェットの設定」を参照）。

(5) シークレットの設定

GCPのSecret Managerとは、データベースの接続URLなど、特に機密性の高い情報（シークレット）をGCP上に安全に保存するためのサービスである。

Terraformの初回実行後、以下のコマンドでシークレットを設定する。

```
echo -n "${REEARTH_DB}" | gcloud secrets versions add reearth-api-REEARTH_DB --data-file=-
echo -n "${REEARTH_DB}" | gcloud secrets versions add reearth-cms-REEARTH_CMS_WORKER_DB --data-file=-
echo -n "${REEARTH_DB}" | gcloud secrets versions add reearth-cms-REEARTH_CMS_DB --data-file=-
```

また、任意で以下のシークレットの設定も行うことができる。それぞれ \${...} は変更して実行すること。設定しなくても次に進むことは可能。

```
# FMEのトークン
echo -n "${REEARTH_PLATEAUVIEW_FME_TOKEN}" | gcloud secrets versions add reearth-cms-REEARTH_PLATEAUVIEW_FME_TOKEN --data-file=-
# G空間情報センターのAPIトークン
echo -n "${REEARTH_PLATEAUVIEW_CKAN_TOKEN}" | gcloud secrets versions add reearth-cms-REEARTH_PLATEAUVIEW_CKAN_TOKEN --data-file=-
# SendGridのAPIキー
echo -n "${REEARTH_PLATEAUVIEW_SENDGRID_APIKEY}" | gcloud secrets versions add reearth-cms-REEARTH_PLATEAUVIEW_SENDGRID_APIKEY --data-file=-
```

(6) Cloud Runのデプロイ

初回時のみ手動でCloud Runのデプロイを行う。

```
gcloud run deploy reearth-api ¥
--image eukarya/plateauview2-reearth:latest ¥
--region asia-northeast1 ¥
--platform managed ¥
--quiet

gcloud run deploy reearth-cms-api ¥
--image eukarya/plateauview2-reearth-cms:latest ¥
--region asia-northeast1 ¥
--platform managed ¥
--quiet

gcloud run deploy reearth-cms-worker ¥
--image eukarya/plateauview2-reearth-cms-worker:latest ¥
--region asia-northeast1 ¥
--platform managed ¥
--quiet

gcloud run deploy plateauview-api ¥
--image eukarya/plateauview2-sidecar:latest ¥
--region asia-northeast1 ¥
--platform managed ¥
--quiet
```

2.1.6 各種デプロイ完了まで待機

以下のコマンドを時々試行し "pong"が返ってくるまで待つ。

```
curl https://api.${DOMAIN}/ping
```

2.1.7 Auth0ユーザーの作成

注意：必ず上記ステップでデプロイ完了を確認してから、Auth0のユーザーを作成すること。そうでないと正常にPLATEAU EditorやPLATEAU CMSにログインできなくなる。

先ほど作成したAuth0のテナント内に、PLATEAU CMSとPLATEAU Editorにログインするためのユーザーを作成する。

<https://manage.auth0.com/dashboard/> にアクセスし、テナントのダッシュボードにログイン後、サイドバーメニューより[User Management]-[Users]を開き、[Create User]をクリックし、登録したいユーザーのメールアドレス・パスワード・ユーザー名を入力する。

この時、Connectionは、テナントに既定で作成されているコネクションである `Username-Password-Authentication` を指定する。

ユーザーを作成すると、メールアドレス宛にメールアドレスの確認を行うためのメールが送信される。必ずそのメールを開いて、メールアドレスの認証を済ませてから、次のステップに進むこと。

なお、他人のユーザーを作成する際は、ユーザー作成後にパスワードリセット（ログイン画面から行える）を依頼することをお勧めする。

2.1.8 PLATEAU CMSの動作確認

以下のURLにログインし、ログイン画面が表示されることを確認する。<ドメイン>は先ほど決めたドメインに置き換えること。HTTPではなくHTTPSであることに注意。

`https://cms.<ドメイン>`

例：ドメインが `plateau-test.example.com` の場合、`https://cms.plateau-test.example.com`

PLATEAU CMSへのログインを行い、ログイン後PLATEAU CMSの画面へ無事遷移すれば、PLATEAU CMSは正常に動作している。

ログインできない場合は、各種デプロイ完了後にAuth0にユーザーを作成済みであること、ユーザーのメールアドレスの認証が済んでいることを確認する。なお、ログインしてもすぐにログイン画面に戻される場合は、Auth0にユーザーは存在するが、PLATEAU EditorおよびPLATEAU CMSのデータベース内にユーザーを作成する処理に失敗していることが考えられる。

次に、PLATEAU CMS連携機能を担うサイドカーサーバーが正しく動作しているか確認する。以下のURLにアクセスし、`pong`と表示されたら成功である。

`https://api.<ドメイン>/ping`

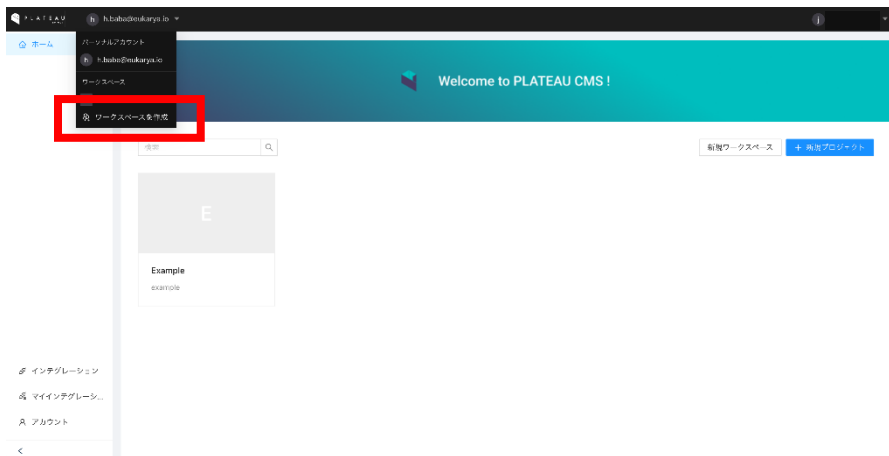
2.1.9 PLATEAU CMSのセットアップ

PLATEAU CMS内でのプロジェクトの初期設定及びインテグレーションの設定に移る。

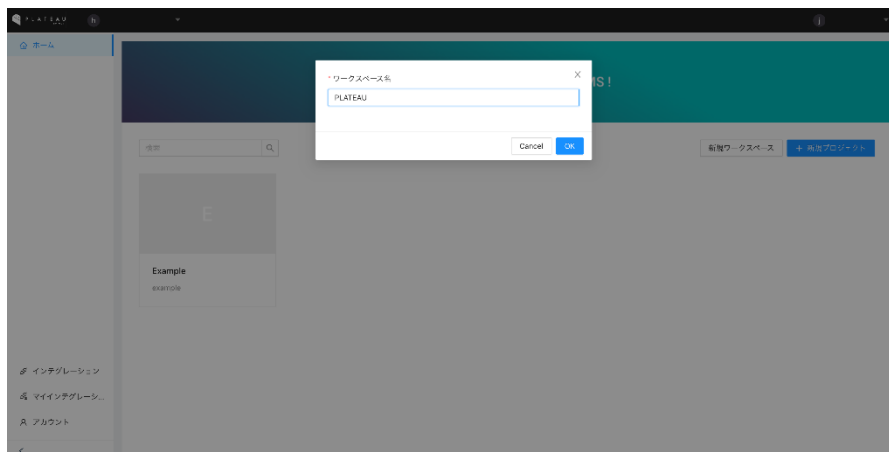
(1) ワークスペースの作成

個人ワークスペースでは他のユーザーを招待できないため、新たにワークスペースを作成する。

1. PLATEAU CMSにログインし、ホーム画面に遷移
2. 新規ワークスペース作成ボタンを押下
 - a. ヘッダー > ワークスペースを作成
 - b. ホーム > 新規ワークスペース



c. ワークスペース名を入力

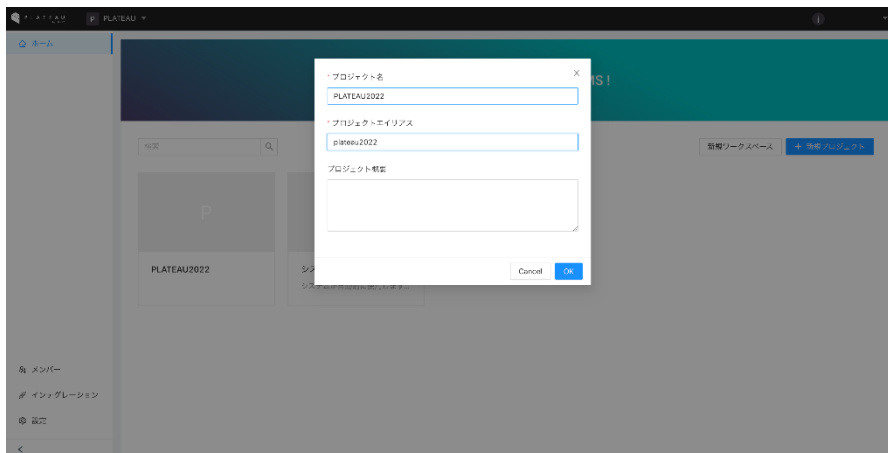


(2) プロジェクトの作成

2つのプロジェクトを作成する。

1つ目は、PLATEAU関連データセットを管理するためのプロジェクト（プロジェクトエイリアス：plateau-2022）であり、もう1つ目は、インテグレーションが自動的に動作する際に内部的に情報を保存するためのプロジェクト（プロジェクトエイリアス：plateauview）。

なお、プロジェクトのエイリアスとは、PLATEAU CMSから公開されるAPIの別名で、公開APIのURLの中で使われる。必須項目であり、5文字以上32文字以下の半角英数字と一部記号が使用可能（_または-）。他プロジェクトが同じエイリアスを使うとエラーとなる。



(3) モデルの作成

1つ目のPLATEAUデータ用プロジェクトに、以下の通りスキーマを3つ作成する。

1. PLATEAUデータ（キー：plateau）

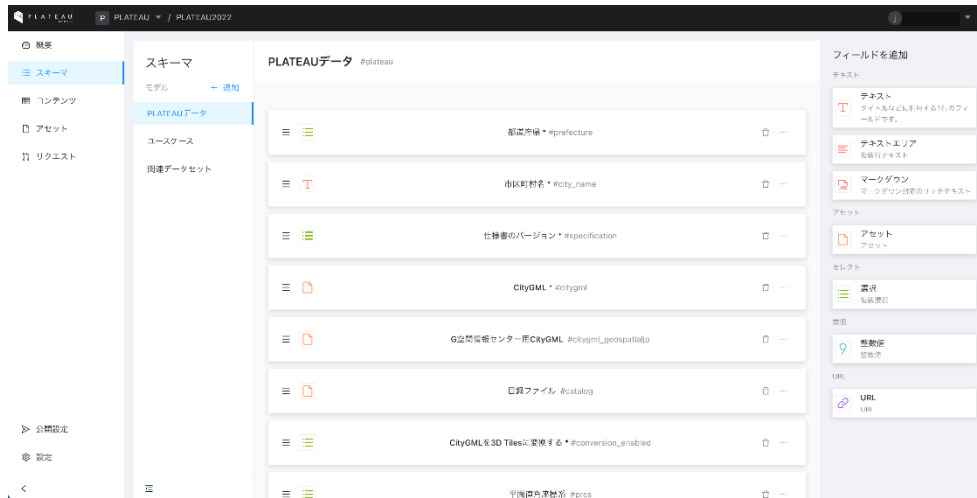


表 PLATEAUデータ スキーマ一覧（1/3）

名前	キー	種類	複数値	必須	説明
都道府県	prefecture	選択		<input type="radio"/>	47都道府県から選択（例：東京都）
市区町村名	city_name	テキスト		<input type="radio"/>	東京都23区のみ「東京都23区」、それ以外は「札幌市」「茂原市」のように入力。区の場合は「東京都23区/千代田区」のようにスラッシュを挟む。
仕様書のバージョン	specification	選択		<input type="radio"/>	登録データが準拠する標準製品仕様書のバージョン。令和4年度事業では「第2.3版」を選択
CityGML	citygml	アセット		<input type="radio"/>	CityGMLを含むzipファイルをアップロード
G空間情報センター用CityGML	citygml_geospatialjpn	アセット			G空間情報センターで公開されるCityGMLのzipファイル。ない場合は上のCityGMLフィールドが公開される
目録ファイル	catalog	アセット			G空間情報センターで公開される目録のxlsxファイル G空間情報センター用メタデータシートも含む
オープンデータURL	opendata_url	URL			PLATEAU VIEWからアクセス可能なオープンデータURL。通常、G空間情報センターの各都市のページのURL。
CityGMLを3D Tilesに変換する	conversion_enabled	選択		<input type="radio"/>	アセットとしてアップロードしたCityGMLデータを3D Tilesに自動変換するかどうかを選択します。ご自身でFMEを利用して変換済みの3D Tiles形式のデータをお持ちの場合は「変換しない」を、自動変換を希望する場合は「変換する」を選択してください。
平面直角座標系	prcs	選択			3D Tilesへの「変換を行う」に設定した場合は、品質検査も自動で行われるため、必ず適切なJGD2011の平面直角座標系を「第1系」～「第16系」から選択。
品質検査パラメータ	quality_check_params	アセット			3D Tilesへの「変換を行う」に設定した場合に行われる品質検査に使用されるパラメータを記述したxlsxファイルが必要であればアップロードすることができます。

表 PLATEAUデータ スキーマ一覧 (2/3)

名前	キー	種類	複数値	必須	説明
政令指定都市で3D Tilesを区に分割	conversion_division_by_ward	選択			「分割する」「分割しない」から選択。政令指定都市の場合に3D Tilesへの変換時に区ごとにデータを分割するかどうか。特に理由がない限り「分割する」を選択。
説明文 建築物モデル	description_bldg	テキストエリア			
説明文 道路モデル	description_tran	テキストエリア			
説明文 都市設備モデル	description_frn	テキストエリア			
説明文 植生モデル	description_veg	テキストエリア			
説明文 土地利用モデル	description_luse	テキストエリア			
説明文 土砂災害警戒区域モデル	description_lslid	テキストエリア			
説明文 都市計画決定情報モデル	description_urf	テキストエリア	○		説明文に対応するアセットのzipファイル名を一行目に記述すること
説明文 洪水浸水想定区域モデル	description_fld	テキストエリア	○		説明文に対応するアセットのzipファイル名を一行目に記述すること
説明文 津波浸水想定区域モデル	description_tnm	テキストエリア	○		説明文に対応するアセットのzipファイル名を一行目に記述すること
説明文 高潮浸水想定区域モデル	description_htd	テキストエリア	○		説明文に対応するアセットのzipファイル名を一行目に記述すること
説明文 内水浸水想定区域モデル	description_ifld	テキストエリア	○		説明文に対応するアセットのzipファイル名を一行目に記述すること
建築物モデル	bldg	アセット	○		CityGMLを3D Tilesに「変換しない」場合のみ手動で3D Tiles形式の建築物モデルデータをアップロードします。それ以外の場合はシステムによって自動的に設定されます。
道路モデル	tran	アセット	○		CityGMLを3D Tilesに「変換しない」場合のみ手動でMVT形式 (LOD1,2) ・3D Tiles形式 (LOD3) の道路モデルデータをアップロードします。それ以外の場合はシステムによって自動的に設定されます。
都市設備モデル	frn	アセット	○		CityGMLを3D Tilesに「変換しない」場合のみ手動で3D Tiles形式の都市設備モデルデータをアップロードします。それ以外の場合はシステムによって自動的に設定されます。
植生モデル	veg	アセット	○		CityGMLを3D Tilesに「変換しない」場合のみ手動で3D Tiles形式の植生モデルデータをアップロードします。それ以外の場合はシステムによって自動的に設定されます。
土地利用モデル	luse	アセット	○		CityGMLを3D Tilesに「変換しない」場合のみ手動でMVT形式の土地利用モデルデータをアップロードします。それ以外の場合はシステムによって自動的に設定されます。
土砂災害警戒区域モデル	lslid	アセット	○		CityGMLを3D Tilesに「変換しない」場合のみ手動でMVT形式の土砂災害警戒区域モデルデータをアップロードします。それ以外の場合はシステムによって自動的に設定されます。
都市計画決定情報モデル	urf	アセット	○		CityGMLを3D Tilesに「変換しない」場合のみ手動でMVT形式の都市計画決定情報モデルデータをアップロードします。それ以外の場合はシステムによって自動的に設定されます。

表 PLATEAUデータ スキーマ一覧 (3/3)

名前	キー	種類	複数値	必須	説明
洪水浸水想定 区域モデル	fld	アセット	○		CityGMLを3D Tilesに「変換しない」場合のみ手動で3D Tiles形式の洪水浸水想定区域モデルデータをアップロードします。それ以外の場合はシステムによって自動的に設定されます。
津波浸水想定 区域モデル	tnm	アセット	○		CityGMLを3D Tilesに「変換しない」場合のみ手動で3D Tiles形式の津波浸水想定区域モデルデータをアップロードします。それ以外の場合はシステムによって自動的に設定されます。
高潮浸水想定 区域モデル	htd	アセット	○		CityGMLを3D Tilesに「変換しない」場合のみ手動で3D Tiles形式の高潮浸水想定区域モデルデータをアップロードします。それ以外の場合はシステムによって自動的に設定されます。
内水浸水想定 区域モデル	ifld	アセット	○		CityGMLを3D Tilesに「変換しない」場合のみ手動で3D Tiles形式の内水浸水想定区域モデルデータをアップロードします。それ以外の場合はシステムによって自動的に設定されます。
全データ	all	アセット			CityGMLを3D Tilesに「変換しない」場合のみ手動で3D TilesおよびMVT形式の全データをまとめたzipファイルをアップロードします。それ以外の場合はシステムによって自動的に設定されます。
辞書データ	dictionary	アセット			システムが自動的に設定するフィールドのため、手動で変更しないこと。
辞書データ (JSON)	dic	テキストエリア			システムが自動的に設定するフィールドのため、手動で変更しないこと。
品質検査・3D Tiles変換 ステータス	conversion_status	選択			「未実行」「実行中」「完了」「エラー」から選択。システムが自動変更するので手動で変更しないこと。
目録ファイル 検査ステータス	catalog_status	選択			「未実行」「実行中」「完了」「エラー」から選択。システムが自動変更するので手動で変更しないこと。
最大LODデータ	max_lod	アセット			システムが自動的に設定するフィールドのため、手動で変更しないこと。
最大LODデータ ステータス	max_lod_status	選択			「未実行」「実行中」「完了」「エラー」から選択。システムが自動変更するので手動で変更しないこと。
検索インデックス	search_index	アセット	○		システムが自動的に設定するフィールドのため、手動で変更しないこと。
SDKに公開	sdk_publication	選択			「公開する」「公開しない」から選択。PLATEAU SDKの公開状況。システムにより自動的に設定が変更されるので（非公開にしたい場合以外は）手動で変更しないこと。

2. ユースケースデータ（キー：usecase）

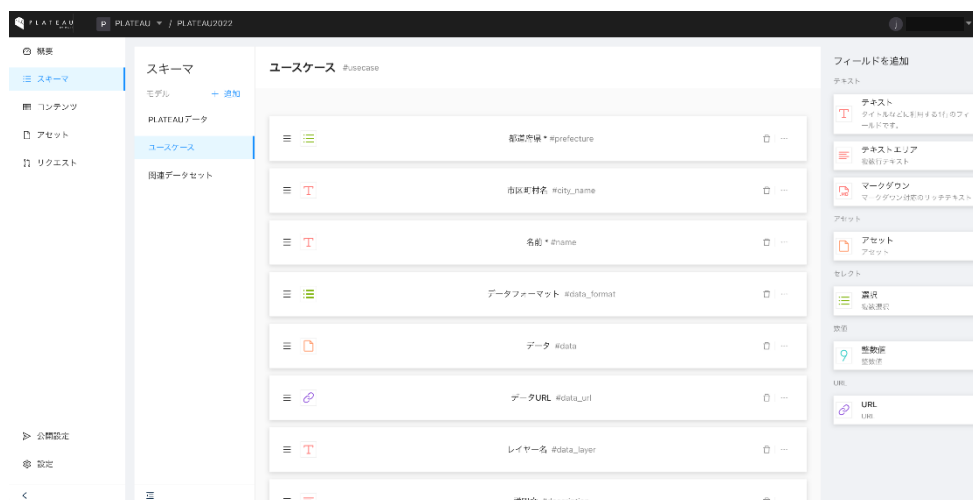


表 ユースケースデータ スキーマ一覧

名前	キー	種類	複数値	必須	説明
名前	name	テキスト		<input type="radio"/>	PLATEAU VIEWのデータカタログで表示される名前。 例：「丸の内ストリートパーク人流データ」「羽田イノベーションシティ」
都道府県	prefecture	選択		<input type="radio"/>	47都道府県と「全球データ」から選択（例：東京都）
市区町村名	city_name	テキスト		<input type="radio"/>	東京都23区のみ「東京都23区」、それ以外は「札幌市」「茂原市」のように入力。区の場合は「東京都23区/千代田区」のようにスラッシュを挟む。
種類	data_type	選択			「避難施設」「ランドマーク」「鉄道駅」「公園」「鉄道」「緊急輸送用道路」「行政界」から選択
データ	data	アセット			
データURL	data_url	URL			外部サーバーに保存されたデータを利用する場合のみ設定
データフォーマット	data_format	選択			「MVT」「GeoJSON」「CZML」「CSV」「WMS」「3D Tiles」「GTFS」から選択
レイヤー名	data_layers	テキスト			MVTもしくはWMSを利用する場合のみ、掲載対象のレイヤー名を入力
説明文	description	テキストエリア			データカタログで表示される
オープンデータURL	opendata_url	URL			PLATEAU VIEWからアクセス可能なオープンデータURL
作成年度	year	選択			「令和4年度」「令和3年度以前」から選択
順序	order	整数値			数字を指定するとデータカタログで同じフォルダの中にあるデータセットの中で順序を変えることができる。指定しない場合または0以下の場合はデフォルトの挙動になる。 例：1, 10, 11, 100
その他設定	config	テキストエリア			管理者以外の人は設定しないでください

3. その他データ（キー：dataset）

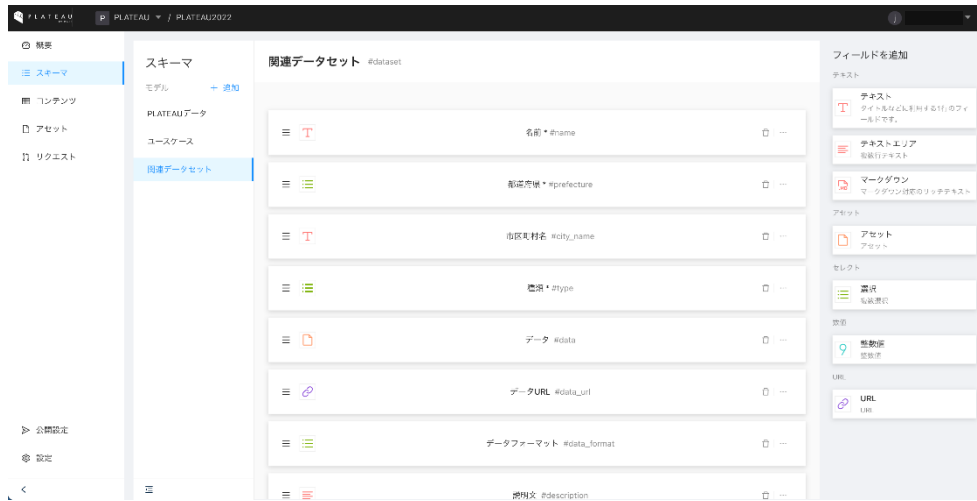


表 関連データ スキーマ一覧

名前	キー	種類	複数値	必須	説明
名前	name	テキスト		<input type="radio"/>	PLATEAU VIEWのデータカタログで表示される名前。例：「避難施設（千代田区）」
都道府県	prefecture	選択		<input type="radio"/>	47都道府県から選択（例：東京都）
市区町村名	city_name	テキスト		<input type="radio"/>	東京都23区のみ「東京都23区」、それ以外は「札幌市」「茂原市」のように入力。区の場合は「東京都23区/千代田区」のようにスラッシュを挟む。
種類	data_type	選択			「避難施設」「ランドマーク」「鉄道駅」「公園」「鉄道」「緊急輸送用道路」「行政界」から選択
データ	data	アセット			
データURL	data_url	URL			外部サーバーに保存されたデータを利用する場合のみ設定
データフォーマット	data_format	選択			「MVT」「CSV」「GeoJSON」「CZML」から選択
レイヤー名	data_layers	テキスト			MVTもしくはWMSを利用する場合のみ、掲載対象のレイヤー名を入力
説明文	description	テキストエリア			データカタログで表示される
オープンデータURL	opendata_url	URL			PLATEAU VIEWからアクセス可能なオープンデータURL
作成年度	year	選択			「令和4年度」「令和3年度以前」から選択
順序	order	整数値			数字を指定するとデータカタログで同じフォルダの中にあるデータセットの中で順序を変えることができる。指定しない場合または0以下の場合はデフォルトの挙動になる。 例：1, 10, 11, 100
その他設定	config	テキストエリア			管理者以外の方は設定しないでください

次に、2つ目のPLATEAU関連データセット用プロジェクトに、以下の通りにモデルを4つ作成する。

名前は自由につけて良いが、連携機能の動作に必要なためキーは必ず指定の文字列を設定すること。
なお必須フィールドは存在しない。

1. itemasset

- キー: itemasset
- フィールド

名前	フィールドタイプ
item	テキスト
asset	アセット

2. share

- キー: share
- フィールド

名前	フィールドタイプ
data	テキストエリア

3. sidebar-data

- キー: sidebar-data
- フィールド

名前	フィールドタイプ
data	テキストエリア

4. sidebar-template

- キー: sidebar-template
- フィールド

名前	フィールドタイプ
data	テキストエリア

(4) プロジェクトの公開設定

1. 1つ目のプロジェクト（PLATEAU関連データセット用）の公開設定タブに遷移。
2. 公開を「公開」に変更し、全てのモデルの切り替えをONにし、アセットをONにしてから、保存ボタンを押下する。

公開設定

公開

非公開

プロジェクトの公開範囲を選択してください。
この操作は以下のすべてのモデルに影響します。

プロジェクトエイリアス

モデル

切り替え

アセット

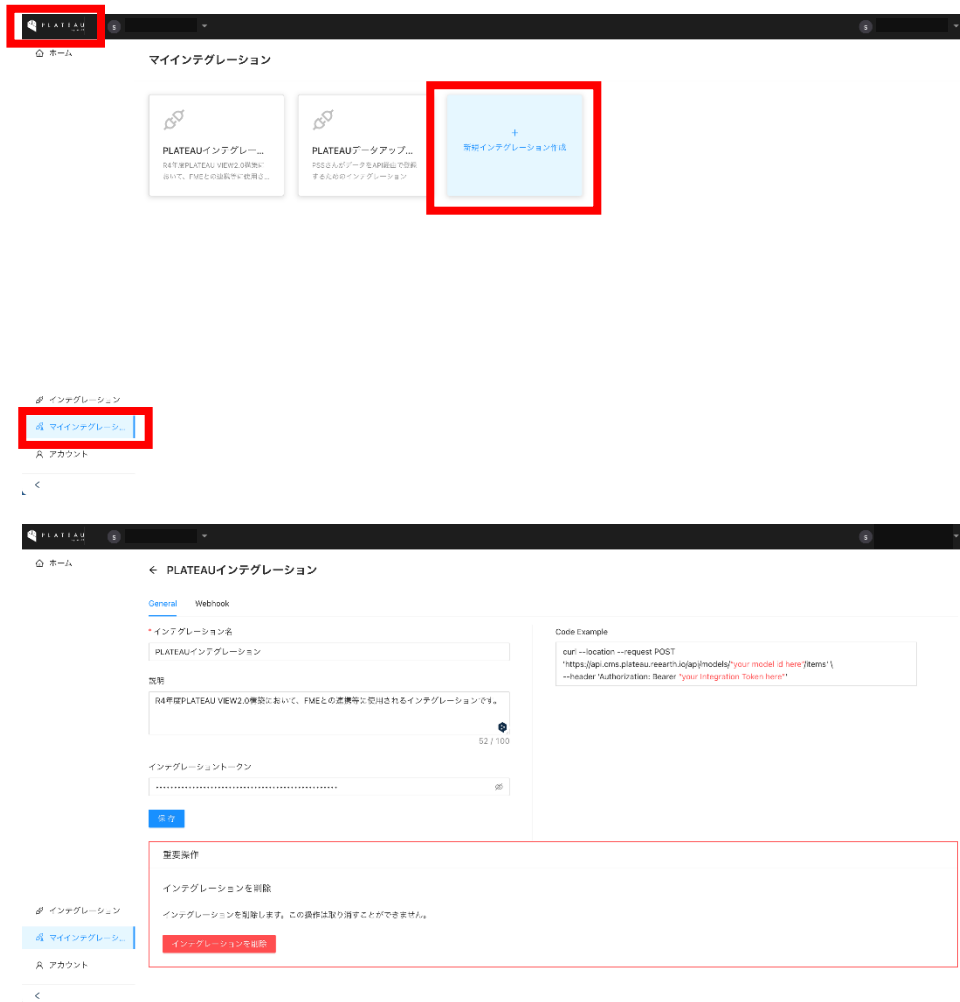
変更を保存

> 公開設定

(5) インテグレーションの作成

インテグレーションとは、PLATEAU CMSと外部アプリケーションとの連携を行うAPIなどの仕組みである。PLATEAU CMSのワークスペースにインストールすることで、PLATEAU CMSの管理APIを利用してデータの取得や変更を行ったり、任意のイベント発生時に外部アプリケーションと連携を行ったりすることができる。

左上のPLATEAU CMSのロゴをクリックしてトップページ（プロジェクト選択画面）に戻り、左下の「マイインテグレーション」から、インテグレーションを作成する。



インテグレーション作成後、以下のような設定を行う。

- Webhook：以下の通りWebhookを設定する。
注意：作成後Webhookの有効化を忘れないこと。
 - 名前：任意
 - URL：イベント発行時のリクエスト先URL。サーバーのURLを入力する。terraform outputsの「plateauview_cms_webhook_url」の値をそのまま指定する。
 - シークレット：Webhookのリクエストが正当なリクエストかどうかをクライアント側で検証するためのシークレット。terraform outputsの「plateauview_cms_webhook_secret」の値をそのまま指定する。
 - イベント：Webhookが発行されるイベント。全て選択する。

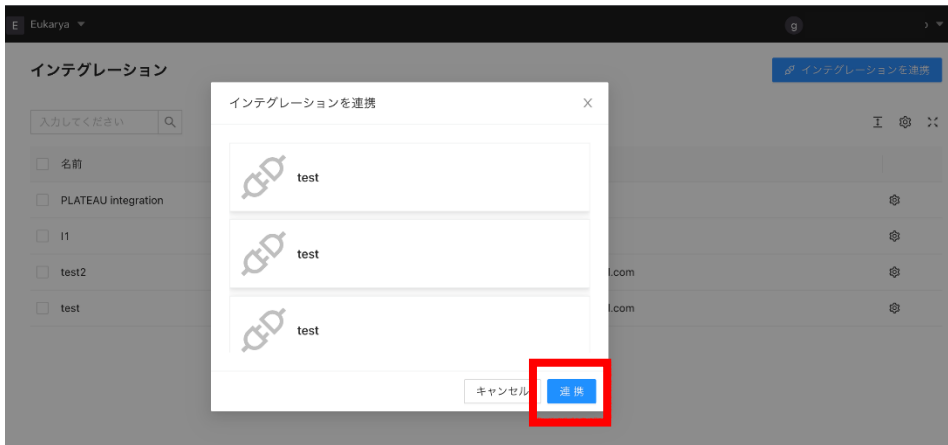
The screenshot shows the 'PLATEAU インテグレーション' (PLATEAU Integration) settings page. The 'Webhook' tab is selected. The form includes the following fields and options:

- 名前 (Name):** A text input field containing 'PLATEAU webhook'.
- URL:** A text input field with a placeholder indicating it should be the URL from terraform outputs.
- 選択 (Select):** A dropdown menu with a note: 'このシークレットはWebhookのリクエストを署名する時に使用されます。' (This secret is used to sign the requests from the Webhook).
- イベント (Events):** A section with checkboxes for '作成' (Create), 'アップデート' (Update), '削除' (Delete), '公開' (Publish), and '非公開' (Unpublish). The '作成' and 'アップデート' checkboxes are checked.
- アセット (Assets):** A section with checkboxes for 'アップロード' (Upload), '圧縮' (Compress), and '削除' (Delete). The 'アップロード' checkbox is checked.

At the bottom left, there is a sidebar with links for 'インテグレーション' (Integrations), 'マイインテグレーション' (My Integrations), and 'アカウント' (Account). A '保存' (Save) button is located at the bottom of the main form area.

(6) インテグレーションの招待

該当プロジェクトに遷移後、インテグレーションタブにて、先ほど作成したインテグレーションを連携する。その後、インテグレーションの権限を「オーナー」に変更する。



最後に、サイドカーサーバーにインテグレーションのトークンを設定する。

先ほど作成したインテグレーションの詳細画面で「インテグレーショントークン」をコピーし、以下の ``${REEARTH_PLATEAUVIEW_CMS_TOKEN}`` の部分に貼り付けてコマンドを実行する。

```
echo -n "${REEARTH_PLATEAUVIEW_CMS_TOKEN}" | gcloud secrets versions add reearth-cms-REEARTH_PLATEAUVIEW_CMS_TOKEN --data-file=-
```

環境変数の変更を適用するため、もう一度サイドカーサーバーをCloud Runにデプロイする。

```
gcloud run deploy plateauview-api ¥
--image eukarya/plateauview2-sidecar:latest ¥
--region asia-northeast1 ¥
--platform managed ¥
--quiet
```

2.1.10 PLATEAU Editorの動作確認

以下のURLにログインし、ログイン画面が表示されることを確認する。

`https://reearth.<ドメイン>`

ログインを行い、ログイン後PLATEAU Editorのダッシュボード画面に遷移すれば、構築は完了である。

ログインできない場合は、各種デプロイ完了後にAuth0にユーザーを作成済みであること、ユーザーのメールアドレスの認証が済んでいることを確認する。なお、ログインしてもすぐにログイン画面に戻される場合は、Auth0にユーザーは存在するが、Re:EarthおよびPLATEAU CMSのデータベース内にユーザーを作成する処理に失敗していることが考えられる。

次にPLATEAU VIEWの構築に移る。PLATEAU VIEWの作成・公開方法については3.4を参照。

2.1.11 参考 : Terraformの実行（2回目以降）

後からtfvarsの内容を変更する場合、変更後に再度Terraformを実行し、インフラに最新の設定を反映する必要がある。

まずtfvarsの編集を行い、次に以下のコマンドを実行し、差分を確認して問題がなければyesと入力して実行を行い、インフラに設定を反映させる（tfvarsファイルを別名で作成し場合は適宜ファイル名を変更）。なお、事前に変数AUTH0_CLIENT_SECRETの設定が必要。

```
terraform apply -var-file=env/example.tfvars
```

2.3 FMEサーバ

実証環境におけるFMEサーバの構築では、Safe Software Inc.のFME Serverを稼働させるサーバー環境とFME Serverのライセンスの取得が必要となる。ここでは、FME Server稼働のための推奨環境について解説する。

2.3.1 環境の準備

FME ServerはWindowsあるいはLinux上で動作するアプリケーションである。動作確認されているOSはWindows（11、10、Server2022、Server2019、Server2016、Server2012R2）、Linux（Ubuntu 20.04 LTS、Ubuntu 18.04 LTS、Debian 11、Debian 10、Red Hat Enterprise Linux 8（[要EPEL8リポジトリ](#)）、Rocky Linux 8（[要EPEL8リポジトリ](#)）、Oracle Linux 8（[要EPEL8リポジトリ](#)）、Red Hat Enterprise Linux 7（[要EPEL7リポジトリ](#)）、CentOS 7（[要EPEL7リポジトリ](#)）、Oracle Linux 7（[要EPEL7リポジトリ](#)））となる。最新の情報はSafe社HPの[ドキュメント](#)を参照されたい。

インストールにはPentium4、あるいはAMD Opteron以降のCPU、8GB以上のRAM、20GB以上のディスクスペースが必要である。ただし、PLATEAU VIEWで使用する3D都市モデルの変換の実行には、64GB以上のRAMと500GB以上のディスクスペースの確保が望ましい。

2.3.2 構築の手順

以下にWindowsおよびLinuxへのインストール手順を示す（より詳細な設定についてはSafe社の[ドキュメント](#)を参照されたい）。

（1）Windows

Safe社のFME Downloadsページからインストーラを入手し、管理者権限で実行する。実行に必要な権限はインストールするディレクトリへの書き込み権限、マシンへのLog on as a serviceの権限である。インストールはインストーラのダイアログに従って、下記の通り実行する。

- ・ Choose Setup Type : Expressを選択
- ・ Destination Folder : アプリケーション、リポジトリとリソースのディレクトリを指定（デフォルトで実行）
- ・ FME Server Hostname : サーバーのHostnameを指定
- ・ Web Application Server Port : 80番、あるいは8080番が推奨
- ・ Database User : インストール時にPostgreSQL データベースに作成されるFME Server Databaseのユーザー名とパスワードを設定

（2）Linux

Safe社のFME Downloadsページからインストーラを入手し、以下のコマンドを実行する。

（インストーラと同ディレクトリでroot userとして実行）

```
chmod +x fme-server-b18205-linux-x64~ubuntu.16.04.run
./fme-server-b18205-linux-x64~ubuntu.16.04.run
```

プロンプトで指定する内容はWindowsの項を参照されたい。

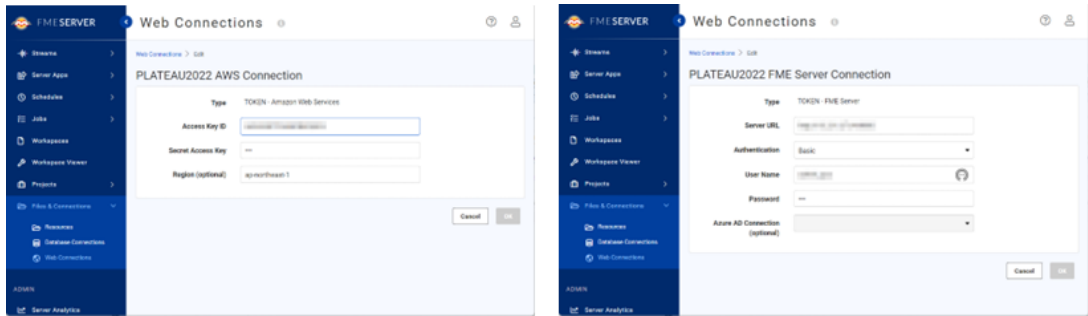
2.3.3 FME Serverの設定

FME Serverの設定は以下の手順で行う。

(1) ウェブ接続の作成

以下の2つの名前付きウェブ接続を作成する。

- AWS Services “PLATEAU2022 AWS Connection”
- FME Server “PLATEAU2022 FME Server Connection”



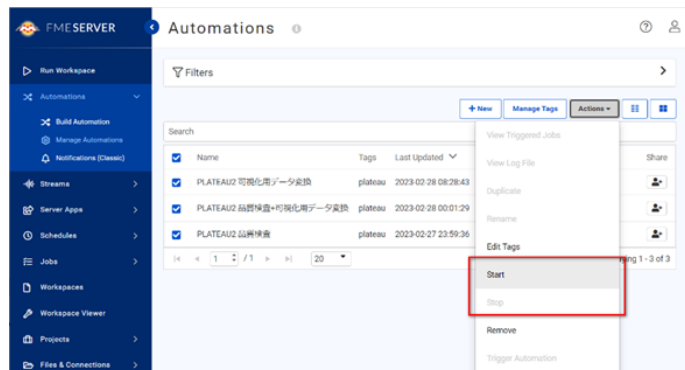
(2) プロジェクトファイルのインポート

PLATEAU VIEW 2.0に登録されたCityGMLの品質検査と可視化のための3DTiles等への変換を行うワークスペースをFME Serverで実行するためのプロジェクトファイル（fsproject）をインポートする。

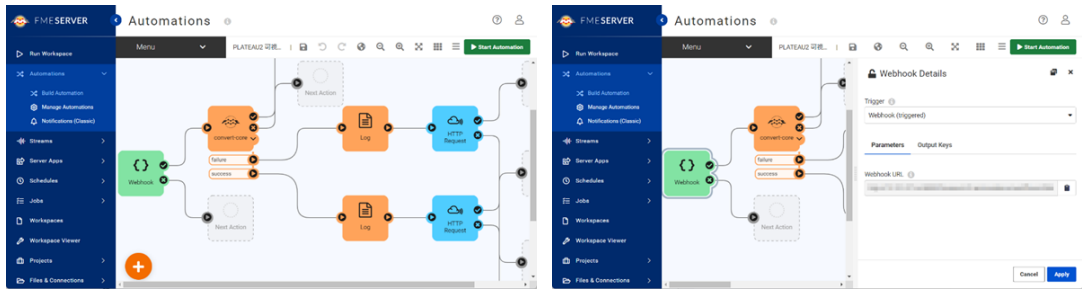
plateau-2022-quality-check-and-convert-all1.0.0_2023-2-26-T151828_b22782.fsproject

(3) ウェブフックURLの取得

Automations管理画面で、全てのAutomationが停止している（STOP）ことを確認してから次の処理を行う。



Webhookトリガーをクリックし、右側に現れるパラメータ設定画面からWebhook URL（FME Serverが自動生成する）をコピーする。



(4) パラメータファイル (JSON) の作成

以下のJSONファイル (common_parameters.json) を作成する。

```
{
  "S3_BUCKET_NAME": "S3 バケット名",
  "WEBHOOK_URLS": {
    "CONVERT_ALL": "自動処理「PLATEAU2 可視化用データ変換」のWebhook Triggerから取得したURL",
    "QUALITY_CHECK": "自動処理「PLATEAU2 品質検査」",
    "QUALITY_CHECK_AND_CONVERT_ALL": "自動処理「PLATEAU2 品質検査+可視化用データ変換」"
  }
}
```

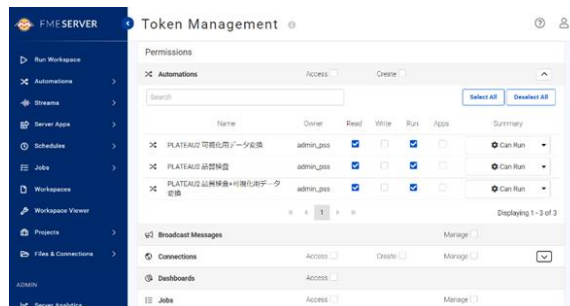
(5) パラメータファイルの登録

上記パラメーターファイルをFME ServerウェブUIから Resources > Data > plateau_2022 > resources フォルダにアップロード (resources フォルダは初回に作成する)。

(6) APIトークンの取得

PLATEAU CMS連携に必要なAPIトークンを取得する。APIトークンの取得はFME Serverのウェブインターフェースから実行する。なお、トークン文字列は作成時の1回しか取得できないので、取得しそこなった場合は再作成する。必要な権限は以下の通り。

- Automations : PLATEAU2 可視化用データ変換、品質検査、品質検査+可視化用データ変換の Read、Run
- Connections : AWS, FME Server ウェブ接続フルアクセス
- Repositories : plateau2022-cms リポジトリ Read、Run
- Services : Job Submitter フルアクセス



第1編 PLATEAU VIEW 2.0

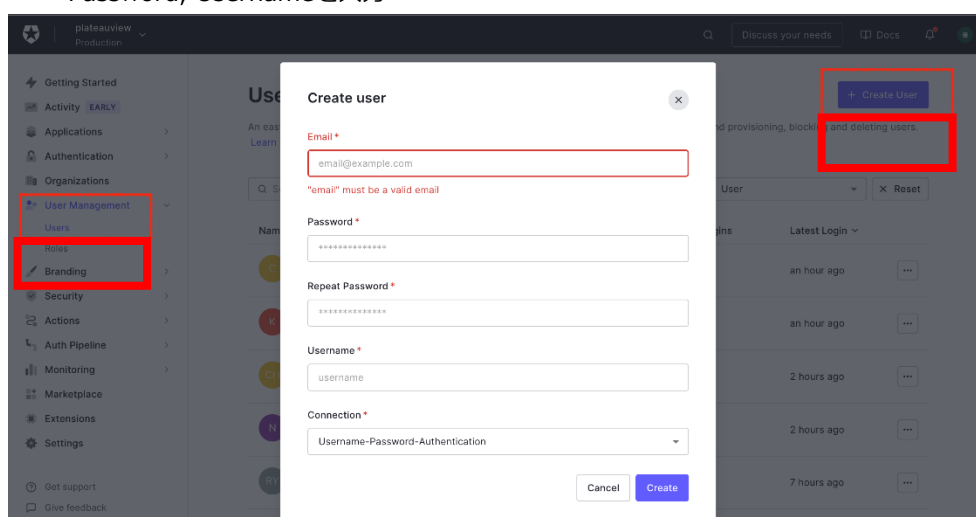
第3章 ユーザーマニュアル

3.1 PLATEAU CMSの利用方法

3.1.1 管理者向け利用方法

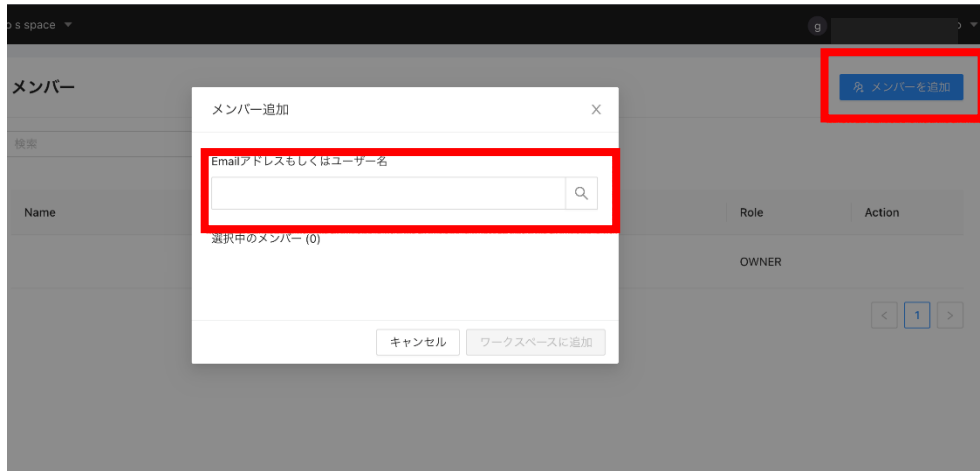
(1) ワークスペースへのメンバーの招待

1. ワークスペースへの追加対象者の情報を準備する。必要となる情報は次の通り。
 - 所属会社
 - メールアドレス
 - 対象者氏名
2. Auth0のテナントにて、招待するメンバーを登録する。
 - a. 「User Management」タブ → 「Create user」ボタンより、Email, Password, Repeat Password, Usernameを入力



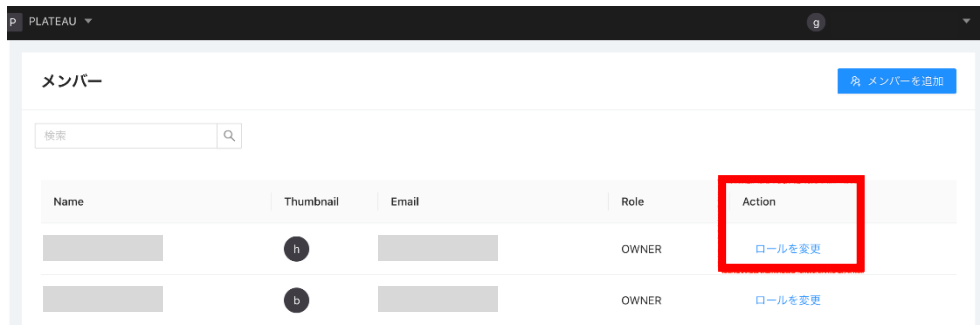
- b. ユーザーの登録が出来たら、対象ユーザーにメールが送信されるので、メール内容に従ってメールアドレスの認証操作を行う。
- c. Auth0にて、招待するメンバーのEmailがverifiedされている事を確認する。
- d. 招待するメンバーに対し、下記情報にてPLATEAU CMSにログインできることを伝える。
 URL: <https://cms.plateauview.mlit.go.jp/>
 ID: 個人のメールアドレス
 パスワード: 入力したパスワード

3. OWNER権限のアカウントでPLATEAU CMSにログインする
4. 対象ワークスペースにて、招待するメンバーを追加する。



(2) ワークスペースへのメンバーの権限の変更

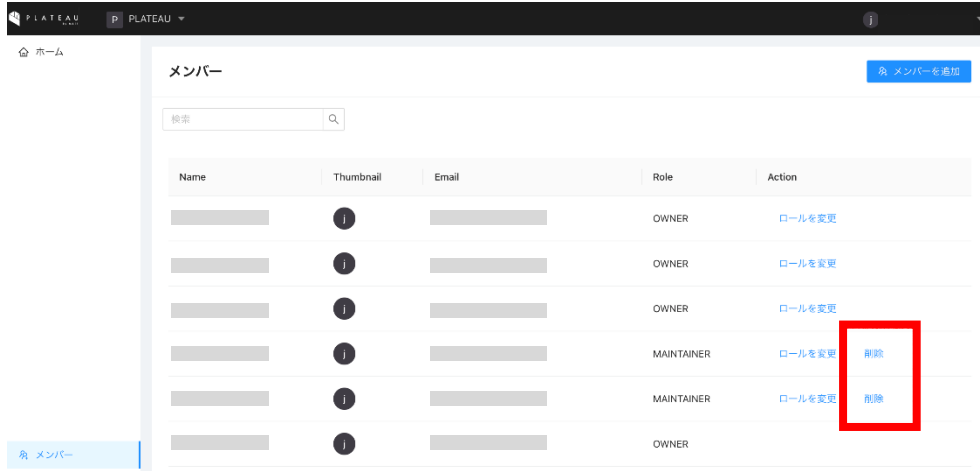
1. OWNER権限のアカウントでPLATEAU CMSにログイン
2. メンバータブの該当ユーザーのActionの項目より、ロールを変更を選択



3. 希望するロールを選択
 - a. オーナー: ワークスペースへのメンバーの招待や削除含めて全ての操作が可能なユーザー
 - b. メンテイナー: ワークスペースへのメンバーの招待や削除以外の操作が可能なユーザー
 - c. 編集者: コンテンツ・アセット・リクエスト・コメントの作成・編集が可能なユーザー
 - d. 閲覧者: 閲覧権限のみ付与されたユーザー

(3) ワークスペースからのメンバーの削除

1. OWNER権限のアカウントでPLATEAU CMSにログイン
2. メンバータブの該当ユーザーのActionの項目より、削除を選択



(4) 公開リクエストの承認

リクエスト承認とは

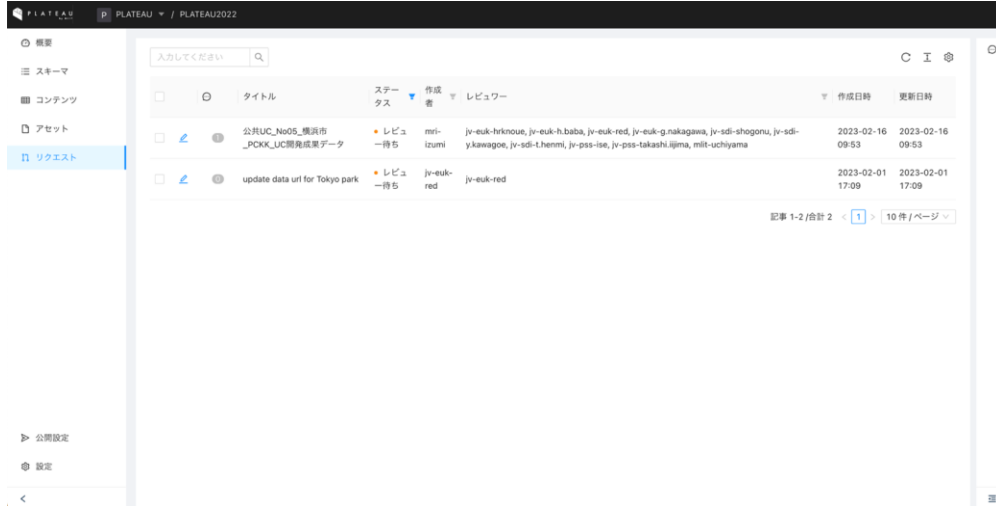
おおまかな流れ

1. OWNER権限のアカウントでPLATEAU CMSにログイン
2. 該当プロジェクトのリクエストタブから、レビュー対象のリクエストの鉛筆ボタンを選択
3. 自身のアカウントがレビュワーに指定されているか確認
4. リクエストの内容の確認
 - a. 承認する場合: 承認ボタンを選択
 - b. コメントする場合: コメントを入力してコメントボタンを選択

(4) 公開リクエストの承認

公開リクエストを承認すると、承認されたリクエストに属するアイテムが公開APIを経由して一般公開される。

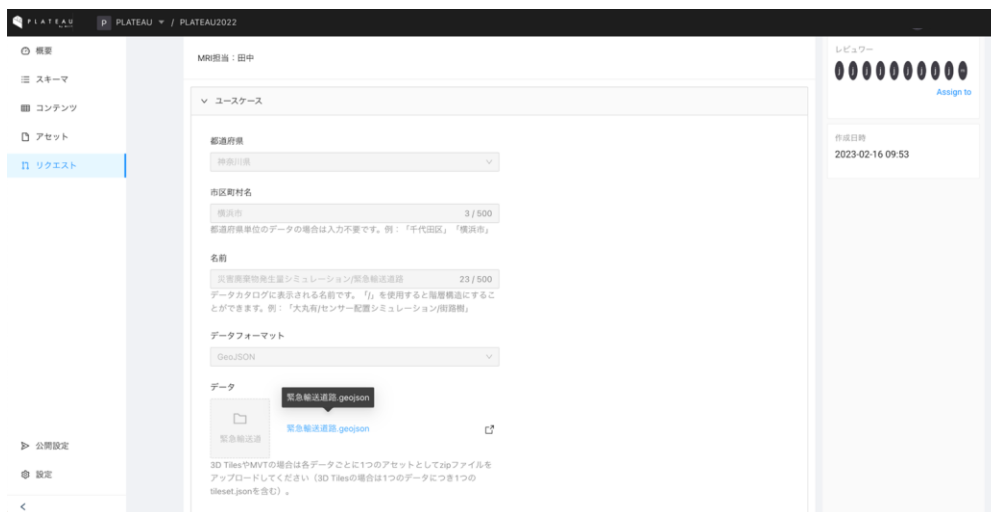
1. 「リクエスト」のタブからレビュー待ちのリクエストを表示する



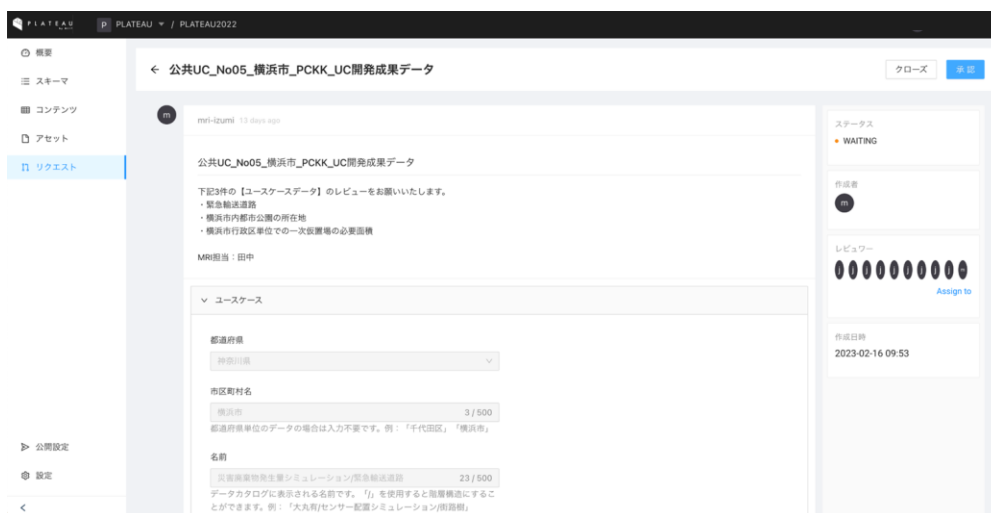
2. 対象のリクエストのペンアイコンをクリックし、公開リクエストの詳細ページへ移動する。



3. 対象のリクエストに属するアイテムの値が正しいかそれぞれを確認する。アセットがリンクされている場合は、そのアセットページへ移動し、地図上での確認も行う。



4. 問題なければ、画面右上の「承認」ボタンを選択する



3.1.2 PLATEAUデータデータ登録方法

PLATEAUデータ（標準製品仕様書に基づいて整備を行なったCityGMLなどのデータ）の整備事業者の方はこちらを参照。

（1）登録のおおまかな流れ

1. 各フィールドを入力し、アイテムを作成（フィールドについては、下記「フィールドについて」を参照）
2. PLATEAUデータの3D Tilesへの変換状況を確認
3. PLATEAUデータ（3D Tiles）をプレビューで確認
4. 公開リクエストを作成し、3D都市モデルデータの公開を申請

（2）フィールドについて

表 データ登録 フィールド一覧（1/3）

名前	キー	種類	複数値	必須	説明
都道府県	prefecture	選択		○	47都道府県から選択（例：東京都）
市区町村名	city_name	テキスト		○	東京都23区のみ「東京都23区」、それ以外は「札幌市」「茂原市」のように入力。区の場合は「東京都23区/千代田区」のようにスラッシュを挟む。
仕様書のバージョン	specification	選択		○	登録データが準拠する標準製品仕様書のバージョン。令和4年度事業では「第2.3版」を選択
CityGML	citygml	アセット		○	CityGMLを含むzipファイルをアップロード
G空間情報センター用CityGML	citygml_geospatial.jp	アセット			G空間情報センターで公開されるCityGMLのzipファイル。ない場合は上のCityGMLフィールドが公開される
目録ファイル	catalog	アセット			G空間情報センターで公開される目録のxlsxファイル。G空間情報センター用メタデータシートも含む
オープンデータURL	opendata_url	URL			PLATEAU VIEWからアクセス可能なオープンデータURL。通常、G空間情報センターの各都市のページのURL。
CityGMLを3D Tilesに変換する	conversion_enabled	選択		○	アセットとしてアップロードしたCityGMLデータを3D Tilesに自動変換するかどうかを選択します。ご自身でFMEを利用して変換済みの3D Tiles形式のデータをお持ちの場合は「変換しない」を、自動変換を希望する場合は「変換する」を選択してください。
平面直角座標系	prcs	選択			3D Tilesへの「変換を行う」に設定した場合は、品質検査も自動で行われるため、必ず適切なJGD2011の平面直角座標系を「第1系」～「第16系」から選択。
品質検査パラメータ	quality_check_params	アセット			3D Tilesへの「変換を行う」に設定した場合に行われる品質検査に使用されるパラメータを記述したxlsxファイルが必要であればアップロードすることができます。
政令指定都市で3D Tilesを区に分割	conversion_division_by_ward	選択			「分割する」「分割しない」から選択。政令指定都市の場合に3D Tilesへの変換時に区ごとにデータを分割するかどうか。特に理由がない限り「分割する」を選択。
説明文 建築物モデル	description_bldg	テキストエリア			
説明文 道路モデル	description_tran	テキストエリア			
説明文 都市設備モデル	description_frn	テキストエリア			
説明文 植生モデル	description_veg	テキストエリア			
説明文 土地利用モデル	description_luse	テキストエリア			
説明文 土砂災害警戒区域モデル	description_lsls	テキストエリア			

表 データ登録 フィールド一覧 (2/3)

名前	キー	種類	複数値	必須	説明
説明文 土砂災害警戒区域モデル	description_lslid	テキストエリア			
説明文 都市計画決定情報モデル	description_urf	テキストエリア	○		説明文に対応するアセットのzipファイル名を一行目に記述すること
説明文 洪水浸水想定区域モデル	description_fld	テキストエリア	○		説明文に対応するアセットのzipファイル名を一行目に記述すること
説明文 津波浸水想定区域モデル	description_tnm	テキストエリア	○		説明文に対応するアセットのzipファイル名を一行目に記述すること
説明文 高潮浸水想定区域モデル	description_htd	テキストエリア	○		説明文に対応するアセットのzipファイル名を一行目に記述すること
説明文 内水浸水想定区域モデル	description_ifld	テキストエリア	○		説明文に対応するアセットのzipファイル名を一行目に記述すること
建築物モデル	bldg	アセット	○		CityGMLを3D Tilesに「変換しない」場合のみ手動で3D Tiles形式の建築物モデルデータをアップロードします。それ以外の場合はシステムによって自動的に設定されます。
道路モデル	tran	アセット	○		CityGMLを3D Tilesに「変換しない」場合のみ手動でMVT形式 (LOD1,2) ・3D Tiles形式 (LOD3) の道路モデルデータをアップロードします。それ以外の場合はシステムによって自動的に設定されます。
都市設備モデル	frn	アセット	○		CityGMLを3D Tilesに「変換しない」場合のみ手動で3D Tiles形式の都市設備モデルデータをアップロードします。それ以外の場合はシステムによって自動的に設定されます。
植生モデル	veg	アセット	○		CityGMLを3D Tilesに「変換しない」場合のみ手動で3D Tiles形式の植生モデルデータをアップロードします。それ以外の場合はシステムによって自動的に設定されます。
土地利用モデル	luse	アセット	○		CityGMLを3D Tilesに「変換しない」場合のみ手動でMVT形式の土地利用モデルデータをアップロードします。それ以外の場合はシステムによって自動的に設定されます。
土砂災害警戒区域モデル	lslid	アセット	○		CityGMLを3D Tilesに「変換しない」場合のみ手動でMVT形式の土砂災害警戒区域モデルデータをアップロードします。それ以外の場合はシステムによって自動的に設定されます。
都市計画決定情報モデル	urf	アセット	○		CityGMLを3D Tilesに「変換しない」場合のみ手動でMVT形式の都市計画決定情報モデルをアップロードします。それ以外の場合はシステムによって自動的に設定されます。
洪水浸水想定区域モデル	fld	アセット	○		CityGMLを3D Tilesに「変換しない」場合のみ手動で3D Tiles形式の洪水浸水想定区域モデルデータをアップロードします。それ以外の場合はシステムによって自動的に設定されます。
津波浸水想定区域モデル	tnm	アセット	○		CityGMLを3D Tilesに「変換しない」場合のみ手動で3D Tiles形式の津波浸水想定区域モデルデータをアップロードします。それ以外の場合はシステムによって自動的に設定されます。
高潮浸水想定区域モデル	htd	アセット	○		CityGMLを3D Tilesに「変換しない」場合のみ手動で3D Tiles形式の高潮浸水想定区域モデルデータをアップロードします。それ以外の場合はシステムによって自動的に設定されます。
内水浸水想定区域モデル	ifld	アセット	○		CityGMLを3D Tilesに「変換しない」場合のみ手動で3D Tiles形式の内水浸水想定区域モデルデータをアップロードします。それ以外の場合はシステムによって自動的に設定されます。

表 データ登録 フィールド一覧 (3/3)

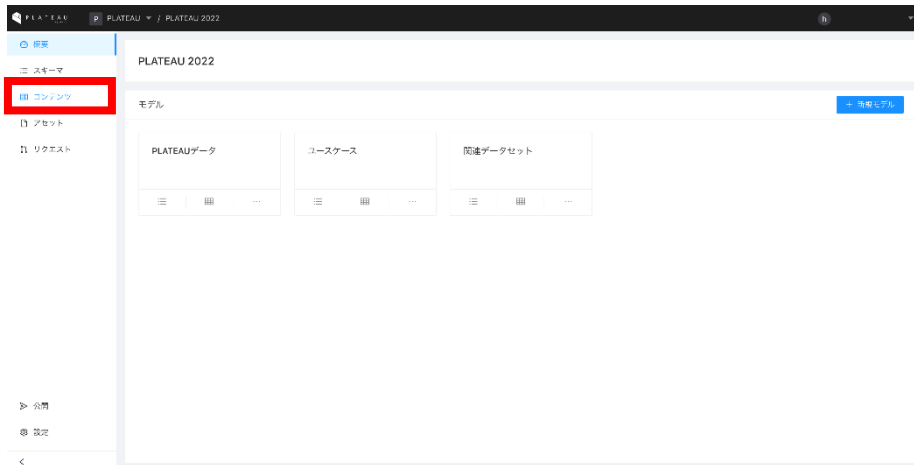
名前	キー	種類	複数値	必須	説明
全データ	all	アセット			CityGMLを3D Tilesに「変換しない」場合のみ手動で3D TilesおよびMVT形式の全データをまとめたzipファイルをアップロードします。それ以外の場合はシステムによって自動的に設定されます。
辞書データ	dictionary	アセット			システムが自動的に設定するフィールドのため、手動で変更しないこと。
辞書データ (JSON)	dic	テキストエリア			システムが自動的に設定するフィールドのため、手動で変更しないこと。
品質検査・3D Tiles変換 ステータス	conversion_status	選択			「未実行」「実行中」「完了」「エラー」から選択。システムが自動変更するので手動で変更しないこと。
目録ファイル 検査ステータス	catalog_status	選択			「未実行」「実行中」「完了」「エラー」から選択。システムが自動変更するので手動で変更しないこと。
最大LODデータ	max_lod	アセット			システムが自動的に設定するフィールドのため、手動で変更しないこと。
最大LODデータ ステータス	max_lod_status	選択			「未実行」「実行中」「完了」「エラー」から選択。システムが自動変更するので手動で変更しないこと。
検索インデックス	search_index	アセット	○		システムが自動的に設定するフィールドのため、手動で変更しないこと。
検索インデックス ステータス	search_index_status	選択			「未実行」「実行中」「完了」「エラー」から選択。システムが自動変更するので手動で変更しないこと。
SDKに公開	sdk_publication	選択			「公開する」「公開しない」から選択。PLATEAU SDKの公開状況。システムにより自動的に設定が変更されるので（非公開にしたい場合以外は）手動で変更しないこと。

（3）PLATEAUデータの登録手順

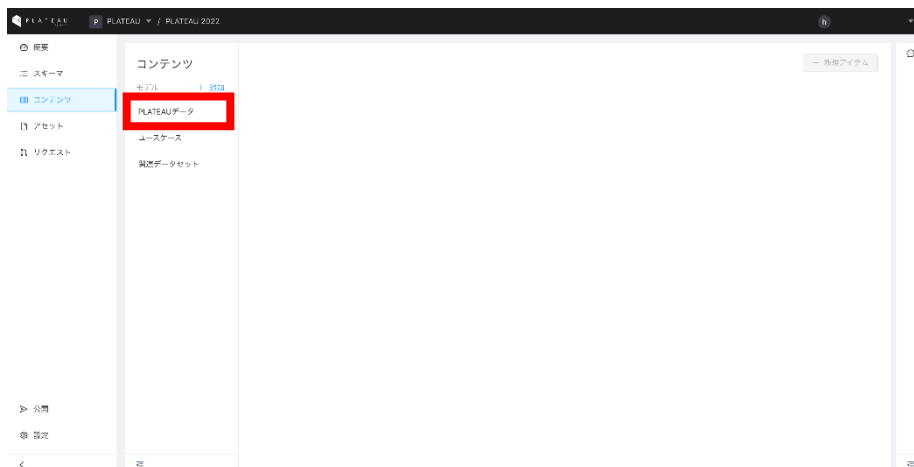
アイテム画面へ切り替え（必須）

プロジェクトを「PLATEAU2022」に切り替え後、概要画面が表示される。PLATEAUデータの登録をするため、画面左のサイドバーから「コンテンツ」を選択し、アイテムの作成ページへ移動する。

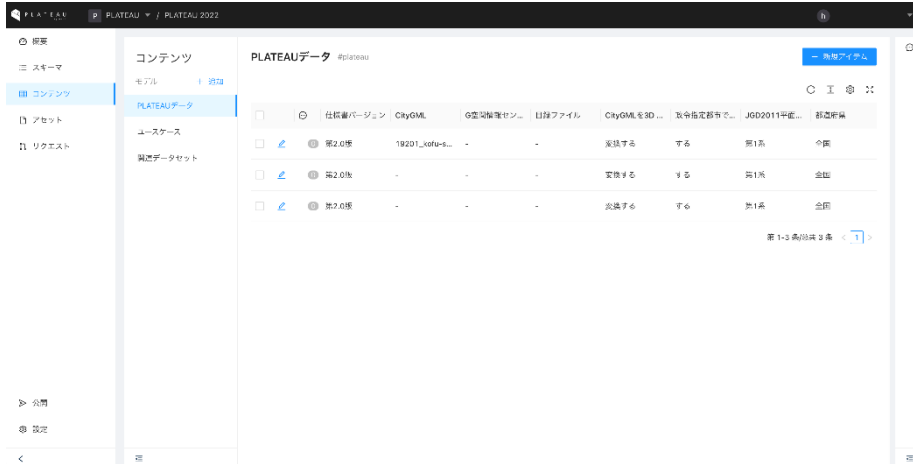
1. 「コンテンツ」をクリック



2. 「コンテンツ」を選択すると、複数のモデルが表示される。この中から、「PLATEAUデータ」を選択する。



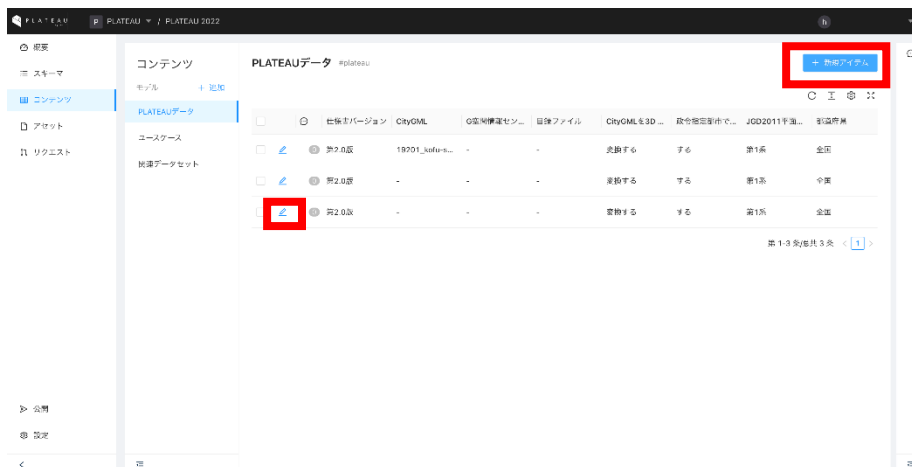
3. 「PLATEAUデータ」を選択すると、すでに登録済みのアイテムが表示される。アイテムが確認できたらアイテム画面への切り替えは成功である。



PLATEAUデータの登録（必須）

標準製品仕様書準拠のPLATEAUデータの登録を行う。

1. 画面右上の「+新規アイテム」を選択する
（既存アイテムを変更・更新する場合、アイテムのリストに表示されたペンアイコンをクリックし操作する）



2. 新規アイテム登録の画面に移動する



3. 各フィールドを入力する。

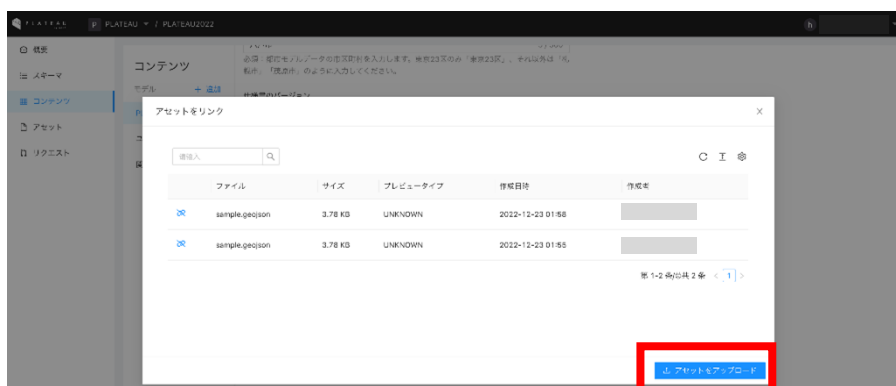


2. PLATEAUデータを設定します。

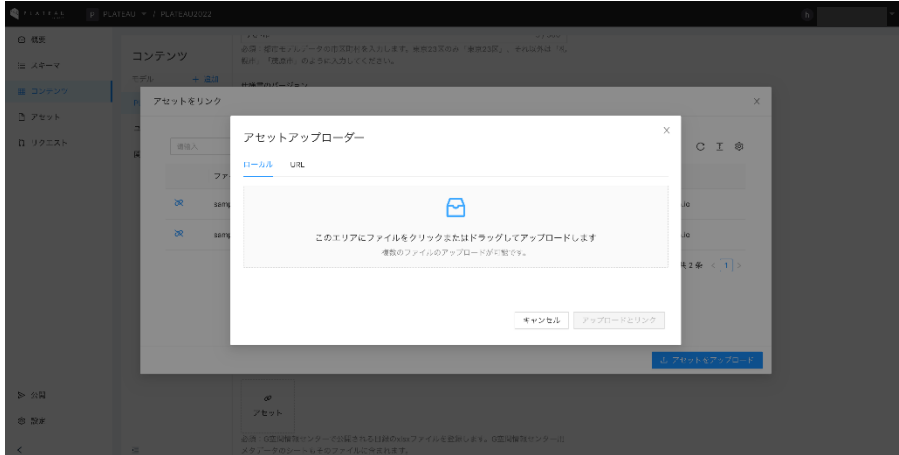
a. 「アセット」をクリックする。



- b. アセットを選択するモーダルが表示される。すでにアセットとしてPLATEAUデータをアップロード済みの場合はこちらから選択する。新規にアップロードする場合は、「アセットをアップロード」をクリックする。
PLATEAUデータはデータサイズが非常に大きく、アップロードに時間がかかる。あらかじめアセットとしてアップロードし、この画面で選択することを推奨する。

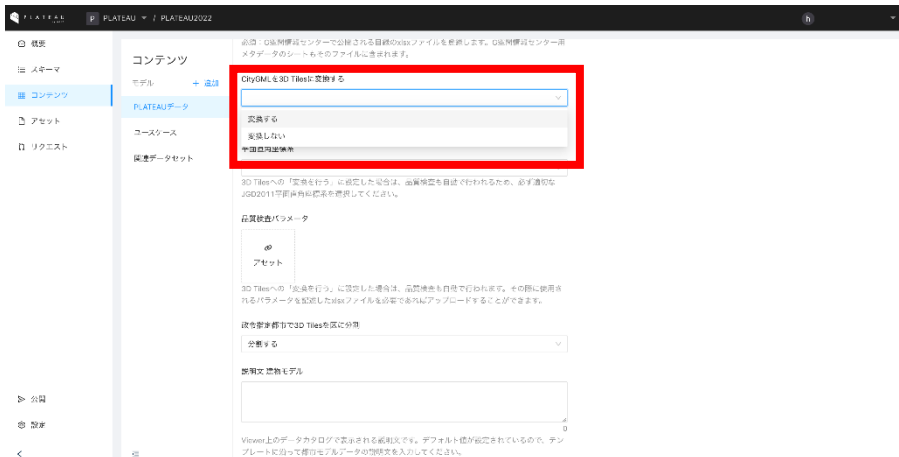


- c. 新規にアップロードする場合、「アセットをアップロード」をクリックすると、PLATEAUデータをアップロードするためのモーダルが表示される。こちらからPLATEAUデータをアップロードする。



- d. PLATEAUデータ選択後「アップロードとリンク」をクリックする。
e. CityGML形式のデータはデータフォーマットの特性上PLATEAU CMSでプレビューすることができない。

3. CityGMLを3D Tilesに変換するかどうかを選択する。すでにFMEなどのツールを利用して、3D Tiles形式へ変換済みのデータを持っている場合は、「変換しない」を選択する。

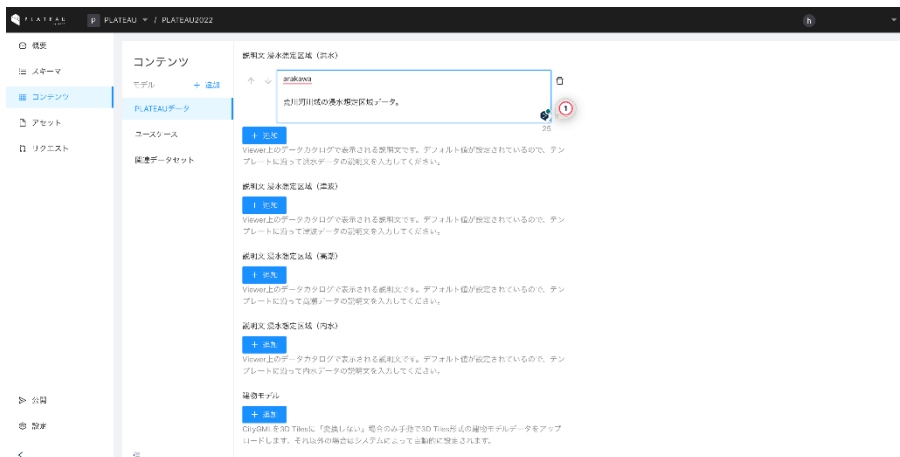


4. その他、「平面直角座標系」などのフィールドを入力する。

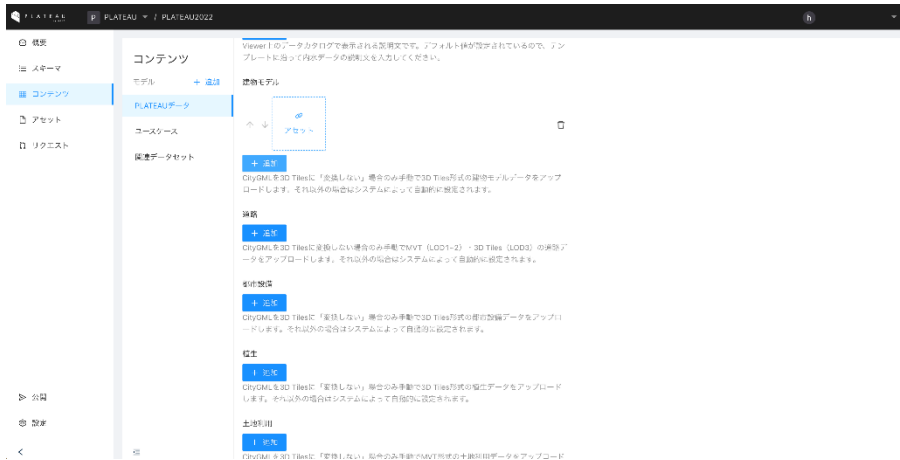
5. PLATEAU VIEWのデータカタログに表示される説明文を記述する。デフォルトでテンプレート文が設定されている。テンプレートに沿って書き換えを行う。



6. 浸水想定区域データはPLATEAUデータごとに任意の数のデータが存在するため、可変長のリスト形式で説明文を記述する。
 浸水想定区域データでは、1行目に変換後のファイル名(e.g. 19201_kofu-shi_2022_citygml_1_fld_pref_fujigawa_nigorigawa_1.zip)を入力する。(前ページの3で「変換する」を選択した方は、FMEでの変換が完了するまではファイル名はわからないため、変換が完了してから再度このフィールドを入力する)



7. CityGMLを3D Tilesに「変換しない」と設定した場合のみ手動でこれらのフィールドを設定する。設定の手順は上記「PLATEAUデータを設定」と同じ手順である。



8. 「操作厳禁」と記載のあるフィールドは操作しない。



9. フィールドの入力が終了したら、「保存」ボタンをクリックし、アイテムを保存する。

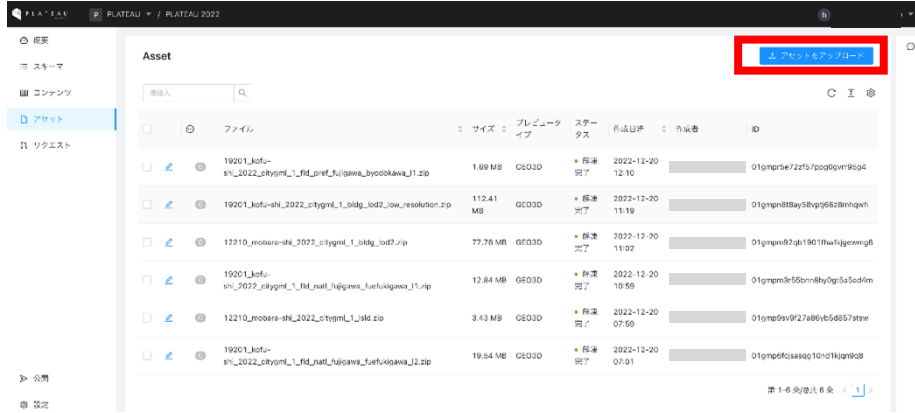


(4) アセットの作成方法

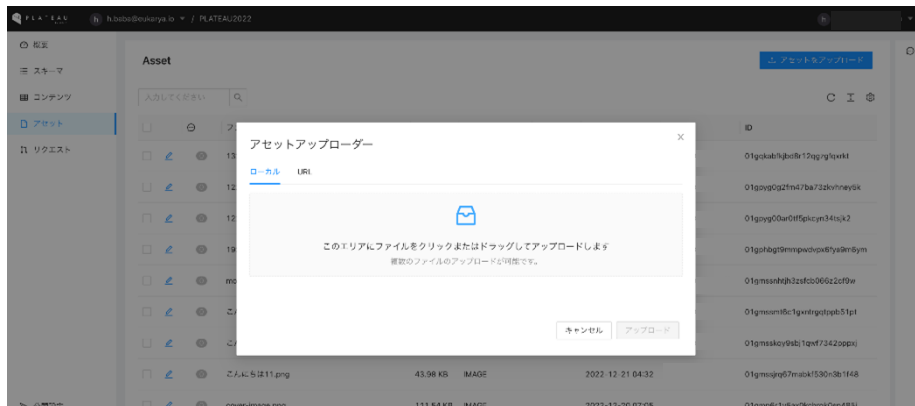
アセットをアップロード

アセット一覧ページから「アセットをアップロード」をクリックする。

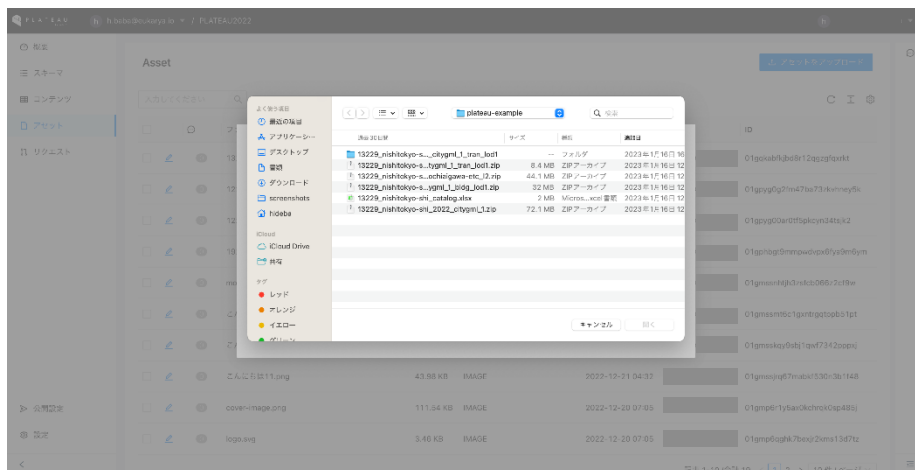
1. アセットの一覧から、ペンアイコンをクリックし自身のアセットを選択する。



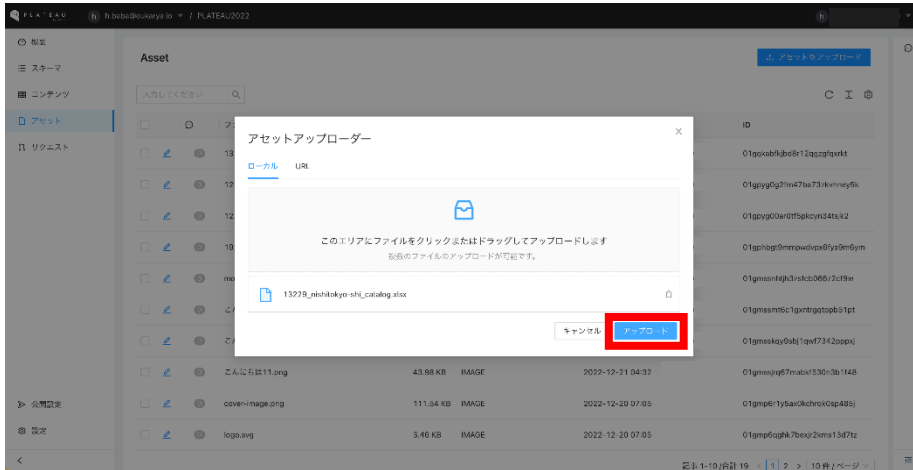
2. アセットをアップロードするモーダルが表示される。



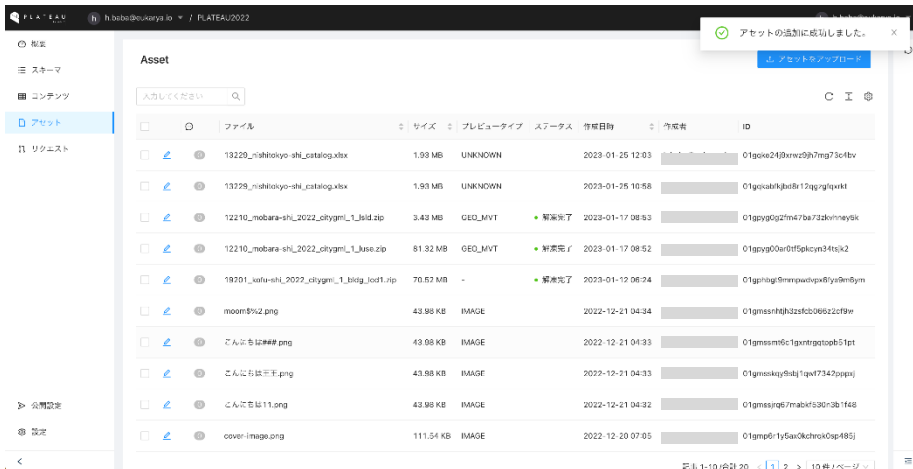
3. ファイルを選択する



4. ファイルを選択すると、選択したファイルが表示される。その後、「アップロード」ボタンを押す。（CityGMLなどサイズの大きいファイルはアップロードに時間がかかることがある。ブラウザのタブを閉じず、アップロードが完了するまで待機する）



5. ファイルのアップロードが完了すると、アセットの一覧に表示される。



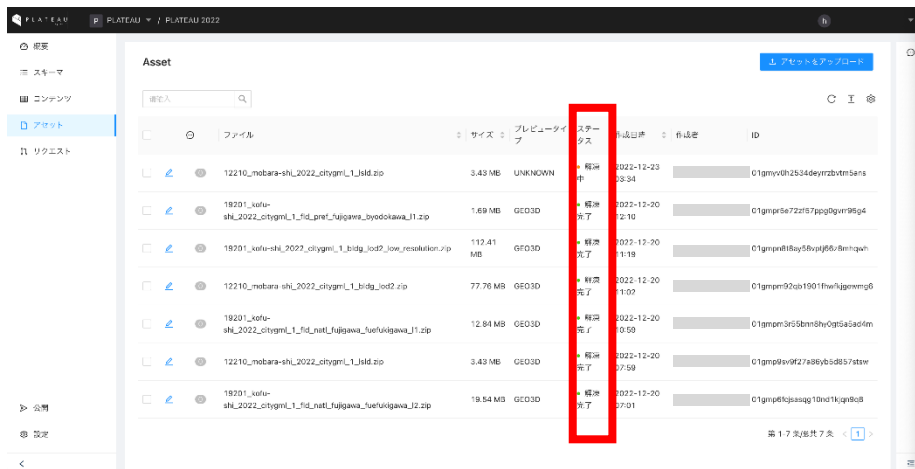
(5) アセットの解凍状況の確認 (必須)

Zipファイル形式でアップロードされたデータはPLATEAU CMSによって自動的に解凍処理が実行される。PLATEAUデータなどはファイルサイズが大きいため、解凍処理に非常に時間がかかる(1時間以上等)。アセットのプレビュー機能などは解凍が全て完了してから利用可能になる。以下の手順に則って、アセットの解凍状況を適宜確認する。

1. アセット一覧ページへ移動

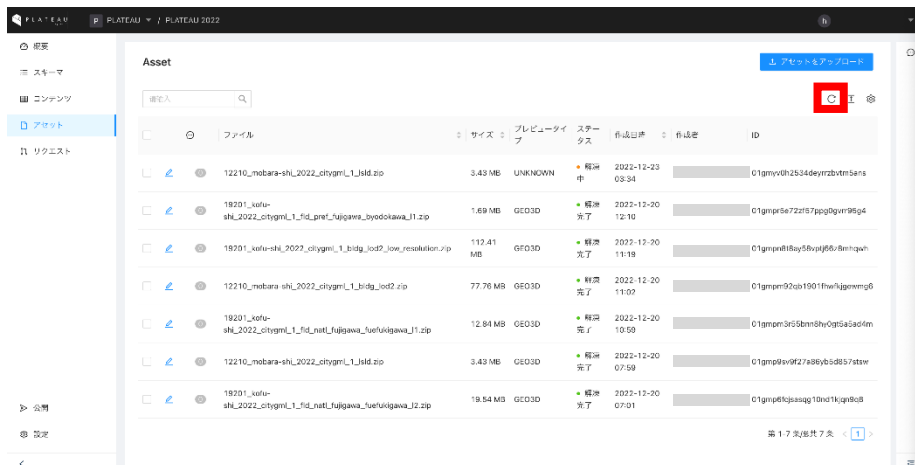
アセット一覧ページへ移動し、「ステータス」を確認する。

- ・ 解凍中：解凍処理が実行中です。
- ・ 解凍完了：解凍処理が完了しました。プレビュー機能が利用可能です。
- ・ 失敗：不正なファイルや圧縮形式の問題で解凍処理が失敗しました。ファイルを削除し、再度適切なファイルをアップロードしなおしてください。



2. ステータスを更新

解凍ステータスが更新されない場合は、画面右上の更新アイコンをクリックしてステータスを更新する



(6) 3D Tilesへの変換処理状況の確認

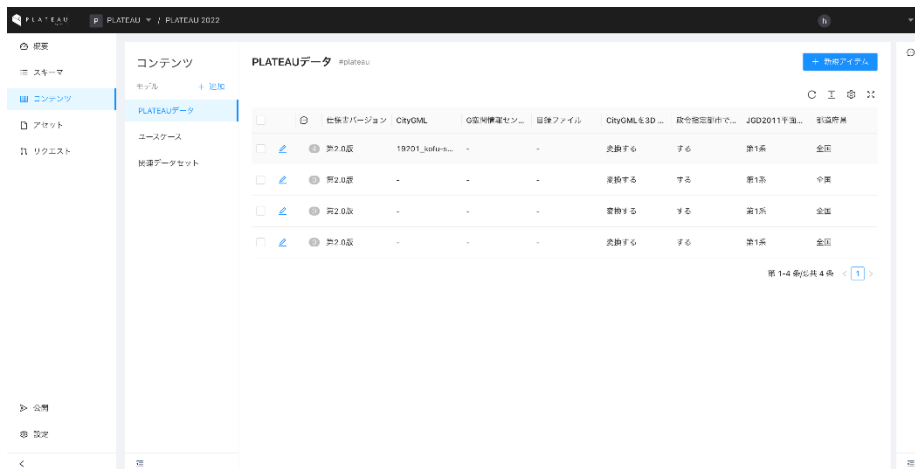
上記フィールドの入力において、CityGMLを3D Tilesに「変換する」と設定した場合はFMEによって自動的に3D Tilesへの変換処理が実行される。この変換処理には非常に時間がかかる場合がある（データによっては24時間程度）。こちらの変換状況は、アイテムのコメント機能を通して確認することが可能である。変換時に発生したエラー等も同様にコメントとして投稿される。

(7) コメント機能

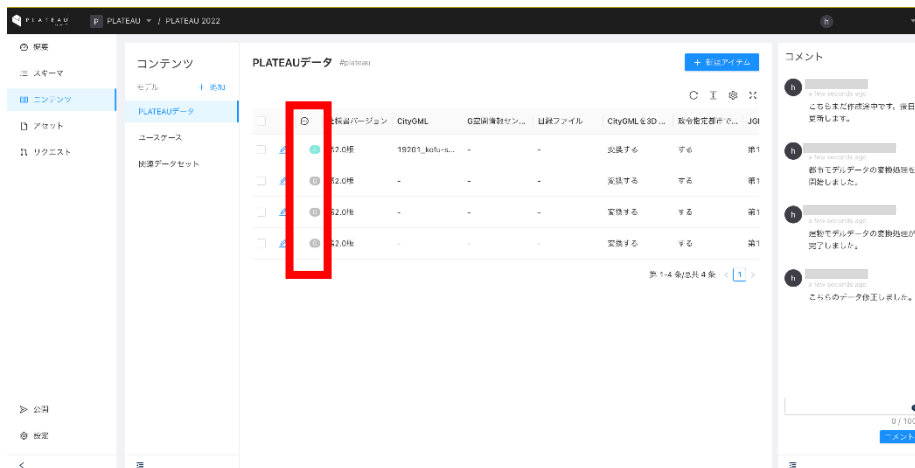
PLATEAU CMSにはコメント機能が存在し、「アイテム」「アセット」「公開リクエスト」それぞれにコメントを投稿し、ユーザー同士でコミュニケーションを取ることができる。また、PLATEAU CMS自身や外部システムが本コメント機能を利用してユーザーにメッセージを送信する手段としても利用される。

1. コメント確認方法

- 「アイテム」、「アセット」もしくは「公開リクエスト」のページを開く。

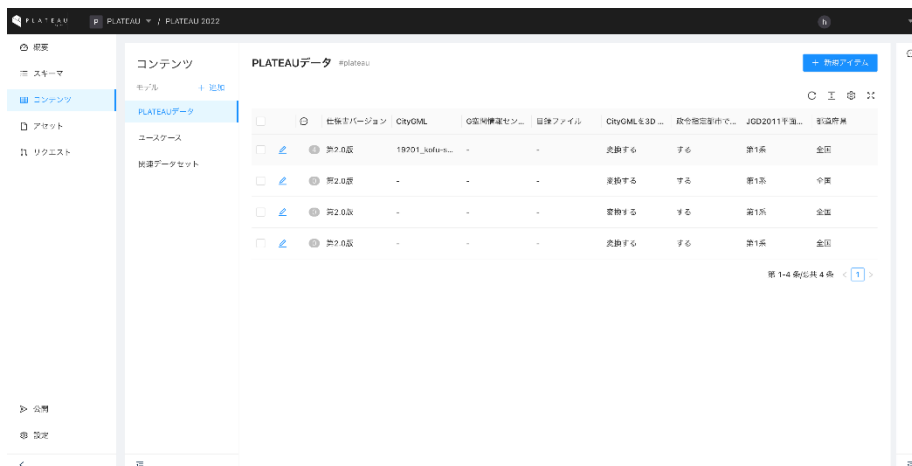


- 対象アイテムの吹き出しアイコンにメッセージの件数が表示される。吹き出しアイコンをクリックしメッセージを確認する。

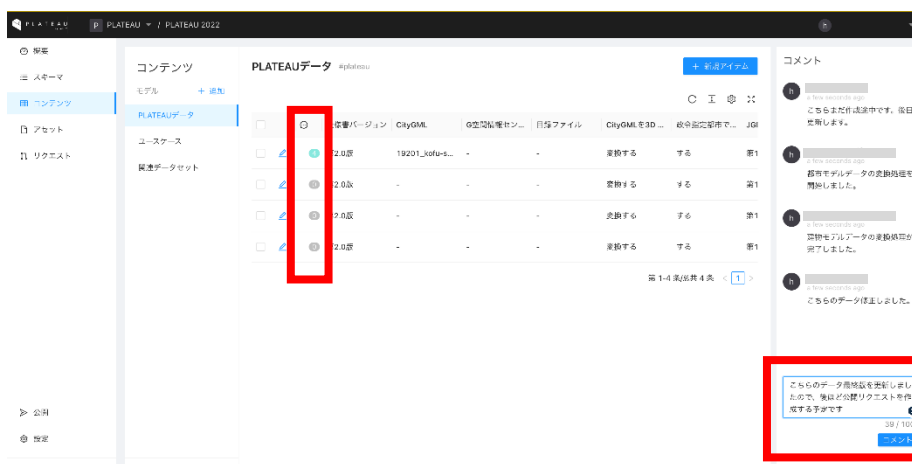


2. コメント投稿方法

- a. 「アイテム」、「アセット」もしくは「公開リクエスト」のページを開く。



- b. 画面下部のテキストボックスからコメントを投稿する。



(8) 公開リクエストの作成 (必須)

公開リクエストを作成し、入力したユースケースデータの確認を管理者に依頼する。

公開リクエスト作成前に必ず以下の項目が全て完了していることを確認してから公開リクエストを作成する。

1. 公開リクエスト作成手順

a. 「新規リクエスト」をクリック



b. 新規リクエスト作成用のモーダルが表示される。



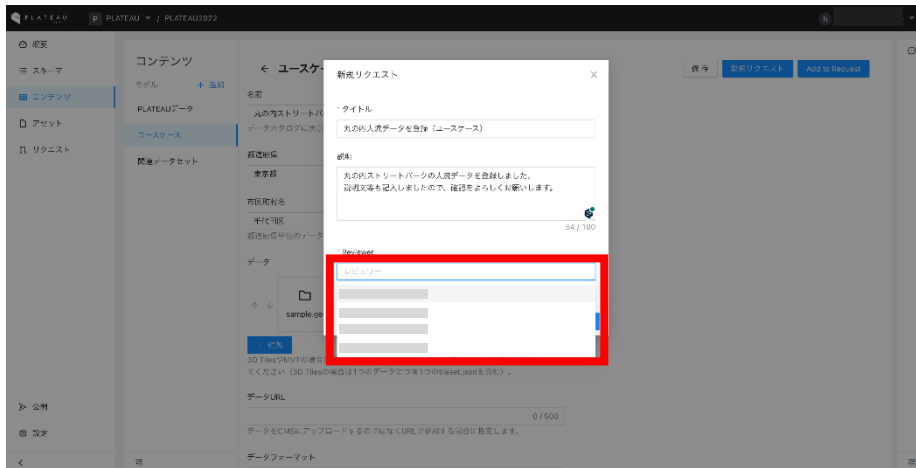
c. リクエストのタイトルを記入

「ユースケース」「PLATEAUデータ」などを記入すると、レビューがわかりやすくなる。



d. レビューに登録データについて説明を簡単に入力する。

- e. レビューーを選択します。ドロップダウンをクリックし、レビュー権限のあるユーザーを表示する。



- f. レビューーを選択し、OKをクリックする。

2. 公開リクエスト作成前のチェックリスト

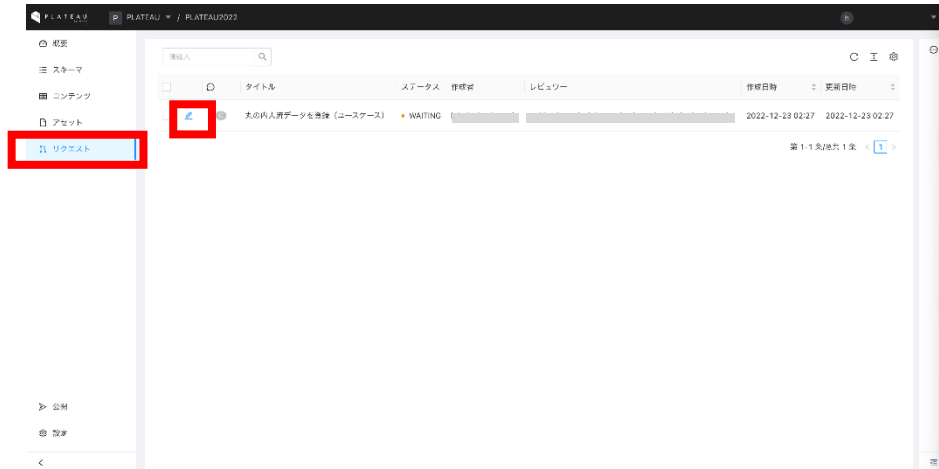
表 チェックリスト

項目1	項目2	確認
全て入力が完了している	都道府県	<input type="checkbox"/>
	市区町村名	<input type="checkbox"/>
	仕様書のバージョン	<input type="checkbox"/>
	CityGML	<input type="checkbox"/>
	目録ファイル	<input type="checkbox"/>
	CityGMLを3D Tilesに変換する	<input type="checkbox"/>
	平面直角座標系	<input type="checkbox"/>
	政令指定都市で3D Tilesを区に分割	<input type="checkbox"/>
	説明文 建築物モデル	<input type="checkbox"/>
	説明文 道路	<input type="checkbox"/>
	説明文 都市設備	<input type="checkbox"/>
	説明文 植生	<input type="checkbox"/>
	説明文 土地利用	<input type="checkbox"/>
	説明文 土砂災害警戒区域	<input type="checkbox"/>
	説明文 都市計画決定情報	<input type="checkbox"/>
	説明文 浸水想定区域（洪水）	<input type="checkbox"/>
	説明文 浸水想定区域（津波）	<input type="checkbox"/>
	説明文 浸水想定区域（高潮）	<input type="checkbox"/>
	説明文 浸水想定区域（内水）	<input type="checkbox"/>
	CityGMLデータの変換が無事完了し、入力項目を満たしている	<input type="checkbox"/>
	「品質検査・3D Tiles 変換 ステータス」が「完了」になっている	<input type="checkbox"/>
すべてのフィールドにFMEでの 変換後のファイルが設定されて いる	建築物モデル	<input type="checkbox"/>
	道路	<input type="checkbox"/>
	都市設備	<input type="checkbox"/>
	植生	<input type="checkbox"/>
	土地利用	<input type="checkbox"/>
	土砂災害警戒区域	<input type="checkbox"/>
	都市計画決定情報	<input type="checkbox"/>
	浸水想定区域（洪水）	<input type="checkbox"/>
	浸水想定区域（津波）	<input type="checkbox"/>
	浸水想定区域（高潮）	<input type="checkbox"/>
	浸水想定区域（内水）	<input type="checkbox"/>
上記ファイル全てをプレビューし、ジオメトリに問題がないか、属性に問題がないかを確認している		<input type="checkbox"/>
TODO:目録ファイル		<input type="checkbox"/>
それぞれのフィールドに	説明文 浸水想定区域（洪水）	<input type="checkbox"/>
	説明文 浸水想定区域（津波）	<input type="checkbox"/>
	説明文 浸水想定区域（高潮）	<input type="checkbox"/>
	説明文 浸水想定区域（内水）	<input type="checkbox"/>

(9) その他状況に応じて必要な操作

1. 公開リクエストのステータスを確認

- a. 画面左サイドバーの「リクエスト」を選択し、公開リクエスト一覧を表示する。



- b. 閲覧するリクエストのペンアイコンをクリックし、公開リクエストの詳細ページへ移動する。



- c. 公開リクエスト詳細ページでは、コメント機能を利用して公開リクエストに対するコメントを閲覧・投稿できる。



- d. 変更リクエストがレビューによって承認された場合、画面右側に「APPROVED」と表示される。

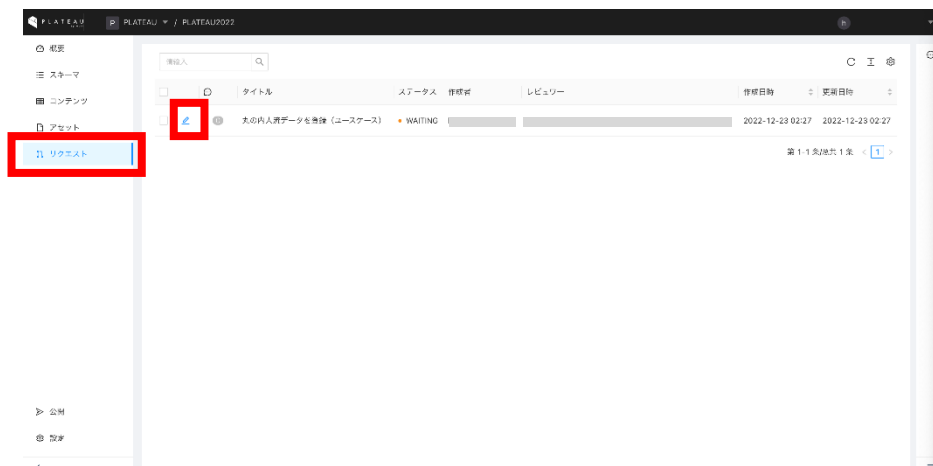


- e. ステータスが「APPROVED」公開リクエストに紐づいたアイテムは自動的に公開される。

2. 公開リクエストのクローズ

誤って公開リクエストを作成してしまった場合、公開リクエストをクローズする。

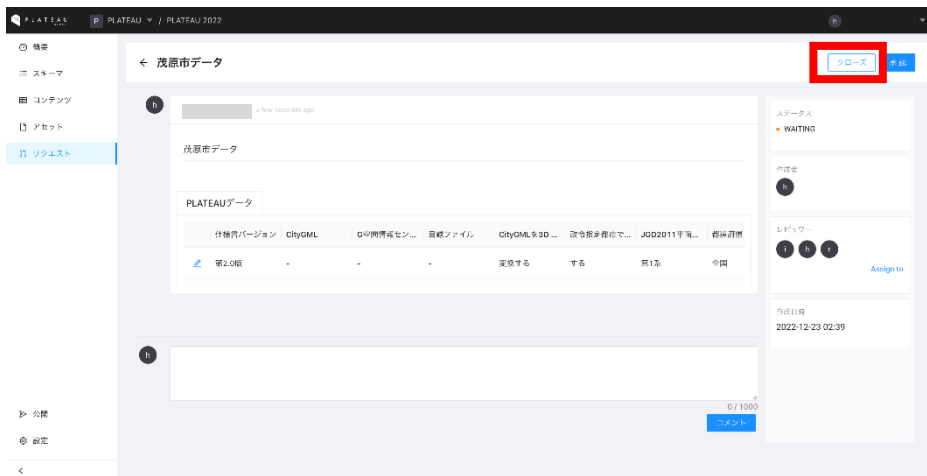
a. 画面左サイドバーの「リクエスト」を選択し、公開リクエスト一覧を表示する。



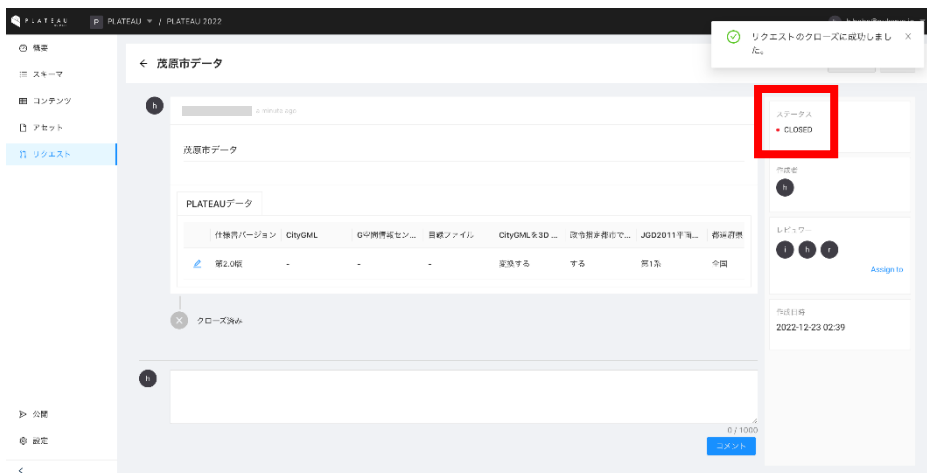
b. 閲覧するリクエストのペンアイコンをクリックし、公開リクエストの詳細ページへ移動する。



- c. 画面右上の「クローズ」ボタンをクリックし、公開リクエストをクローズする。



- d. 公開リクエストのステータスが「CLOSED」になっていることを確認する。



2. 公開リクエスト作成前のチェックリスト

表 チェックリスト

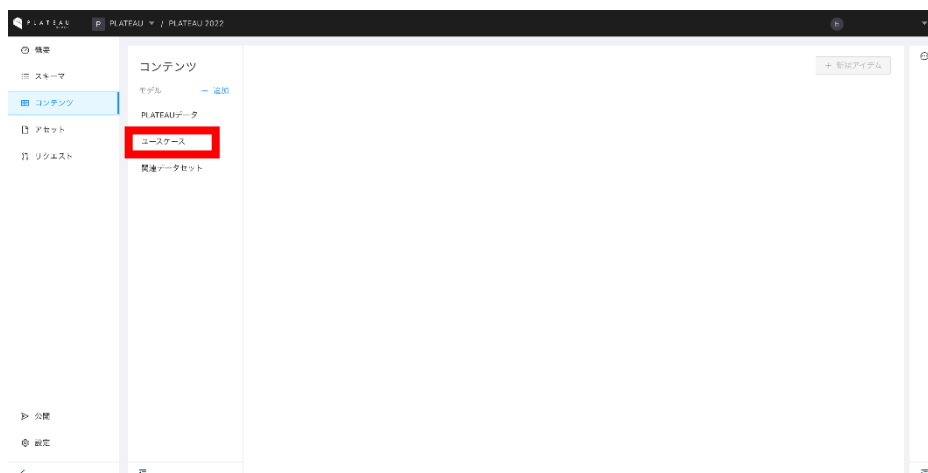
項目1	項目2	確認
全て入力が完了している	都道府県	<input type="checkbox"/>
	市区町村名	<input type="checkbox"/>
	仕様書のバージョン	<input type="checkbox"/>
	CityGML	<input type="checkbox"/>
	目録ファイル	<input type="checkbox"/>
	CityGMLを3D Tilesに変換する	<input type="checkbox"/>
	平面直角座標系	<input type="checkbox"/>
	政令指定都市で3D Tilesを区に分割	<input type="checkbox"/>
	説明文 建築物モデル	<input type="checkbox"/>
	説明文 道路	<input type="checkbox"/>
	説明文 都市設備	<input type="checkbox"/>
	説明文 植生	<input type="checkbox"/>
	説明文 土地利用	<input type="checkbox"/>
	説明文 土砂災害警戒区域	<input type="checkbox"/>
	説明文 都市計画決定情報	<input type="checkbox"/>
	説明文 浸水想定区域（洪水）	<input type="checkbox"/>
	説明文 浸水想定区域（津波）	<input type="checkbox"/>
	説明文 浸水想定区域（高潮）	<input type="checkbox"/>
	説明文 浸水想定区域（内水）	<input type="checkbox"/>
	CityGMLデータの変換が無事完了し、入力項目を満たしている	<input type="checkbox"/>
	「品質検査・3D Tiles 変換 ステータス」が「完了」になっている	<input type="checkbox"/>
すべてのフィールドにFMEでの 変換後のファイルが設定されて いる	建築物モデル	<input type="checkbox"/>
	道路	<input type="checkbox"/>
	都市設備	<input type="checkbox"/>
	植生	<input type="checkbox"/>
	土地利用	<input type="checkbox"/>
	土砂災害警戒区域	<input type="checkbox"/>
	都市計画決定情報	<input type="checkbox"/>
	浸水想定区域（洪水）	<input type="checkbox"/>
	浸水想定区域（津波）	<input type="checkbox"/>
	浸水想定区域（高潮）	<input type="checkbox"/>
	浸水想定区域（内水）	<input type="checkbox"/>
上記ファイル全てをプレビューし、ジオメトリに問題がないか、属性に問題がないかを確認している		<input type="checkbox"/>
TODO:目録ファイル		<input type="checkbox"/>
それぞれのフィールドに	説明文 浸水想定区域（洪水）	<input type="checkbox"/>
	説明文 浸水想定区域（津波）	<input type="checkbox"/>
	説明文 浸水想定区域（高潮）	<input type="checkbox"/>
	説明文 浸水想定区域（内水）	<input type="checkbox"/>

(3) ユースケースデータの登録手順

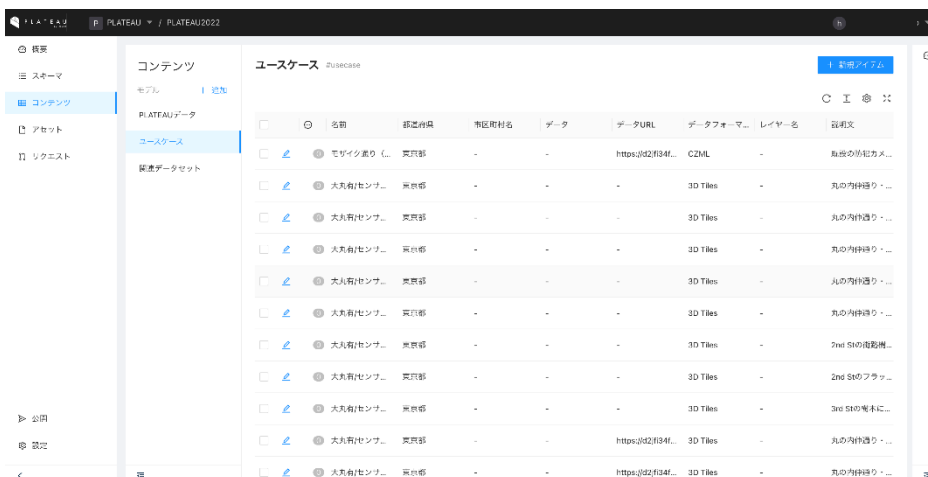
1. アイテム画面へ切り替え（必須）

プロジェクトを「PLATEAU2022」に切り替え後、概要画面が表示される。ユースケースデータの登録をするため、画面左のサイドバーから「コンテンツ」を選択し、アイテムの作成ページへ移動する。

- a. 「コンテンツ」をクリックすると、複数のモデルが表示されます。この中から、「ユースケースデータ」を選択する。



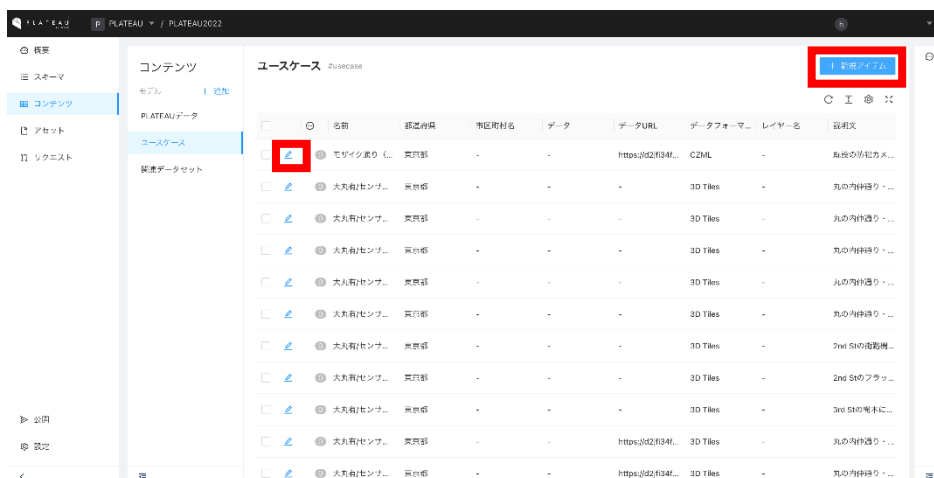
- b. 「ユースケース」を選択すると、すでに登録済みのアイテムが表示される。アイテムが確認できたらアイテム画面への切り替えは成功である。



2. ユースケースデータの登録（必須）

a. 画面右上の「+新規アイテム」を選択する。

既存アイテムを変更・更新する場合、アイテムのリストに表示されたペンアイコンをクリックし操作する。



b. 新規アイテム登録の画面に移動する。



c. 各フィールドを入力する。

ユースケース

名前
次の内ストリートパーク人混みデータ 10 / 500
データURLに指定される名前です。

選択作業
東京都
神奈川県
新潟県
富山県
石川県
福井県
山梨県
データURL
0 / 500
データもCMSにアップロードするのではなくURLで参照する場合は指定します。

データフォーマット
GeoJSON

レイヤー名
0 / 500

d. データの「追加」ボタンをクリックする。

ユースケース

名前
次の内ストリートパーク人混みデータ 10 / 500
データURLに指定される名前です。

選択作業
東京都

市町村名
千代田区 4 / 500
都道府県単位でのデータの指定は入力不要です。例：「千代田区」「横浜市」

データ
追加
3D TilesやMBTの場合は各データセットに1つのアセットとしてzipファイルをアップロードしてください。3D Tilesの場合は1つのデータにつき1つのdataset.jsonを含む。

データURL
https://example.com 10 / 500
データもCMSにアップロードするのではなくURLで参照する場合は指定します。

データフォーマット
GeoJSON

レイヤー名

e. 「アセット」をクリックする。

ユースケース

名前
次の内ストリートパーク人混みデータ 10 / 500
データURLに指定される名前です。

選択作業
東京都

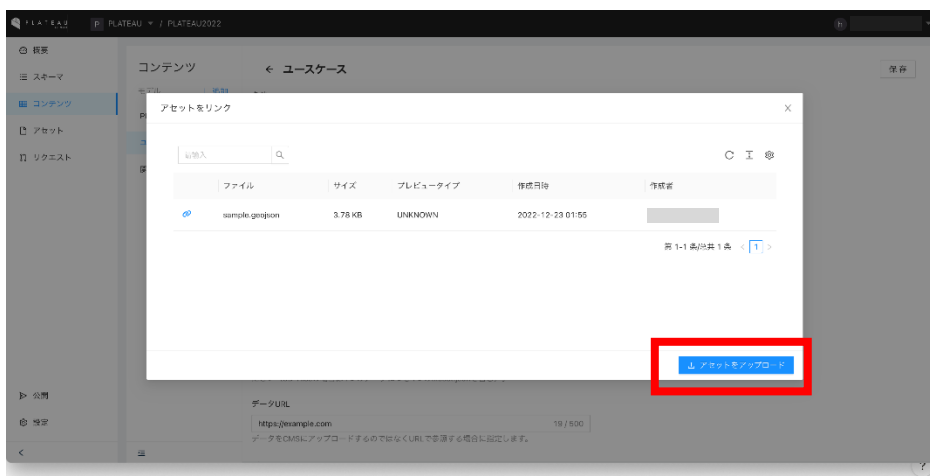
市町村名
千代田区 4 / 500
都道府県単位でのデータの指定は入力不要です。例：「千代田区」「横浜市」

データ
アセット
3D TilesやMBTの場合は各データセットに1つのアセットとしてzipファイルをアップロードしてください。3D Tilesの場合は1つのデータにつき1つのdataset.jsonを含む。

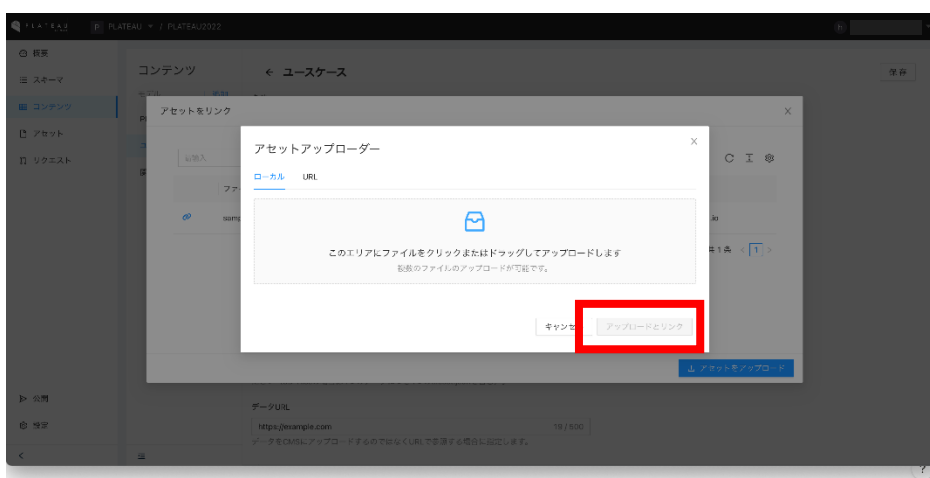
データURL
https://example.com 10 / 500
データもCMSにアップロードするのではなくURLで参照する場合は指定します。

レイヤー名

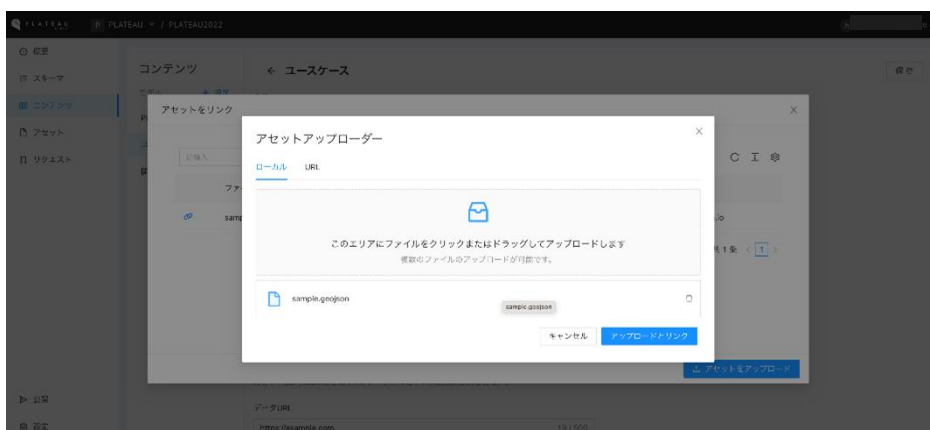
- f. アセットを選択するモーダルが表示される。すでにアセットとしてユースケースデータをアップロード済みの場合はこちらから選択する。新規にアップロードする場合は、「アセットをアップロード」をクリックする。



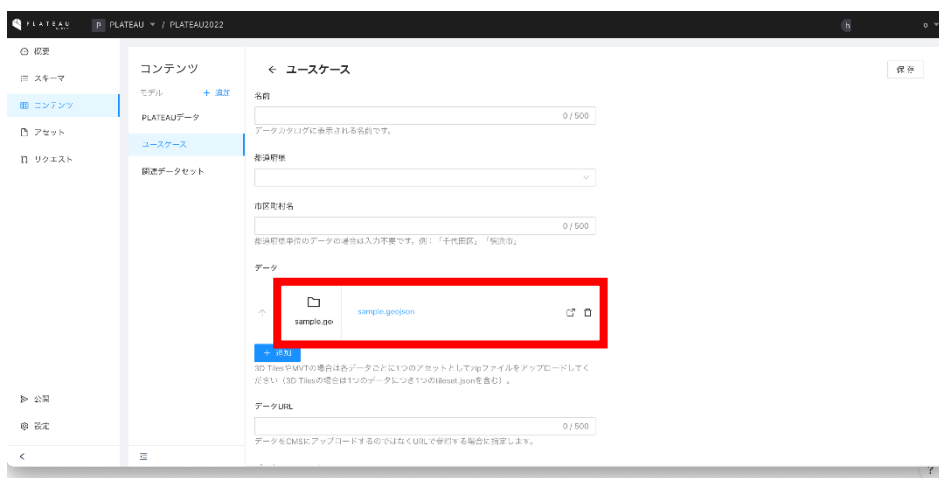
- g. 新規にアップロードする場合、「アセットをアップロード」をクリックすると、ユースケースデータをアップロードするためのモーダルがさらに表示される。こちらからユースケースデータをアップロードする。



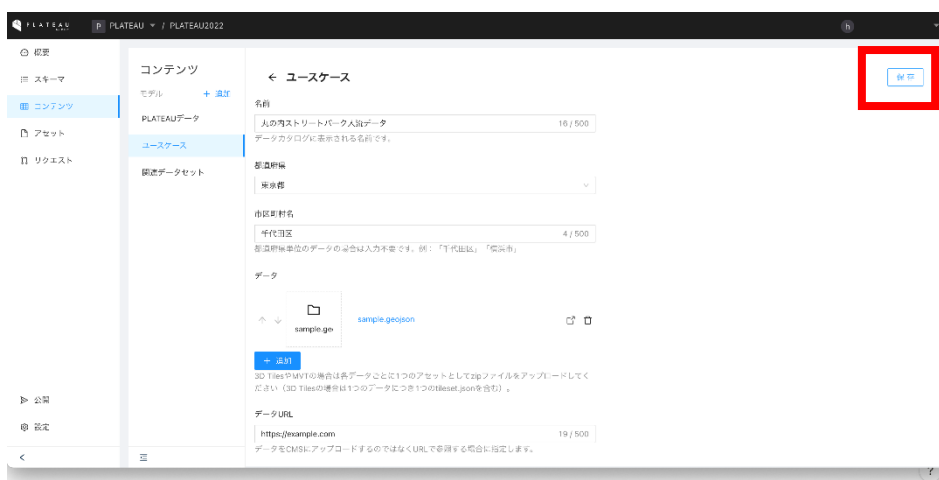
- h. ユースケースデータを選択後「アップロードとリンク」をクリックする。



- i. アセット名を選択し、アセット詳細ページへ遷移する。その後、アップロードしたユースケースデータをプレビューする。アセットのプレビュー方法は次ページを参照されたい。



- j. そのほかのフィールドも同様にを入力をし、「保存」ボタンをクリックしてアイテムを保存する。

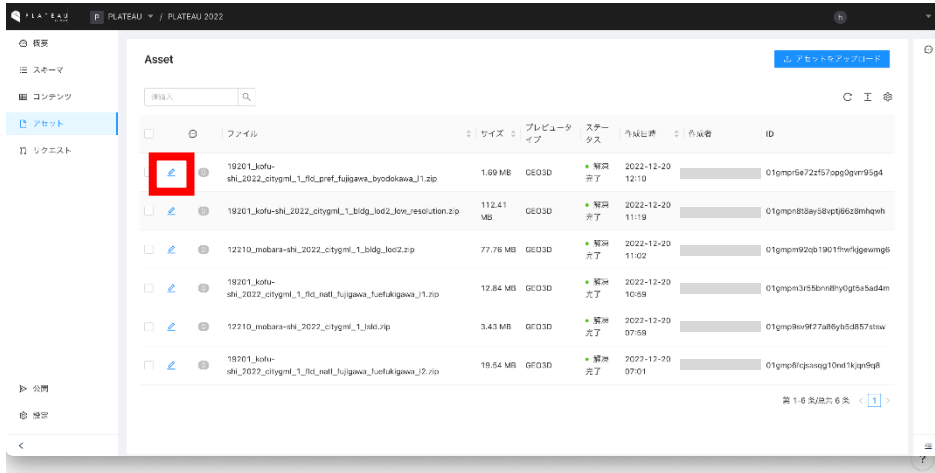


(4) アセットのプレビュー方法

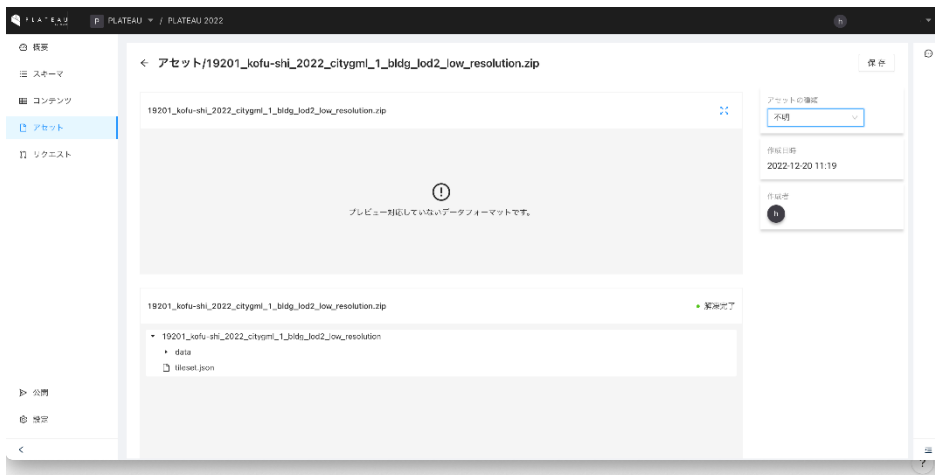
1. アセットを選択

すでにデータをアップロードし、アセットを作成している場合、アセット一覧ページから対象のアセットを選択する。アイテム作成ページからアセットの詳細画面へ移動する場合、本ステップは省略する。

a. アセットの一覧から、ペンアイコンをクリックし自身のアセットを選択する。



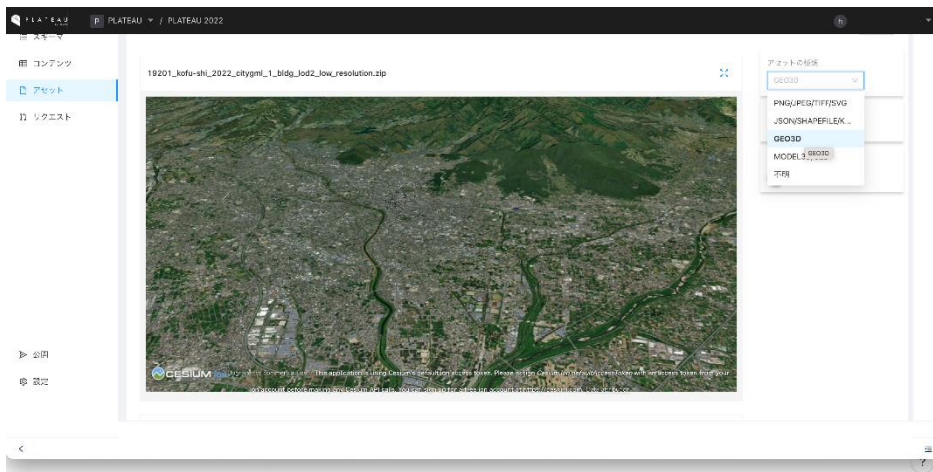
b. アセットを選択すると、アセットの詳細・プレビュー画面へ遷移する。



c. アセットの種類を選択する。



d. アセットの種類を選択すると、種類に応じたプレビューが表示される。



e. 地図上に表示された地物を選択し、属性情報を確認する。



f. PLATEAUデータ登録者の方は以下のルールに沿ってプレビューする。

表 アセットの種類 対応表

名前	アセットの種類
建築物（ファイル名に、`bldg`がついているもの）	3D Tiles
道路（ファイル名に、`tran`がついているもの）	<ul style="list-style-type: none"> • LOD1 : MVT • LOD2 : MVT • LOD3 : 3D Tiles
都市設備（ファイル名に、`fm`がついているもの）	3D Tiles
植生（ファイル名に、`veg`がついているもの）	3D Tiles
浸水想定区域（洪水、津波、高潮、内水）（ファイル名に`fld`、`tnm`、`htd`、`htd`などがついているもの）	3D Tiles
土地利用（ファイル名に`luse`がついているもの）	MVT
都市計画決定情報（ファイル名に`urf`がついているもの）	MVT
土砂災害警戒区域（ファイル名に`sls`がついているもの）	MVT

(5) 公開リクエストの作成 (必須)

公開リクエストを作成し、入力したユースケースデータの確認を管理者に依頼する。

公開リクエスト作成前に必ず以下の項目が全て完了していることを確認してから公開リクエストを作成する。

1. 公開リクエスト作成手順

a. 「新規リクエスト」をクリック



b. 新規リクエスト作成用のモーダルが表示される。



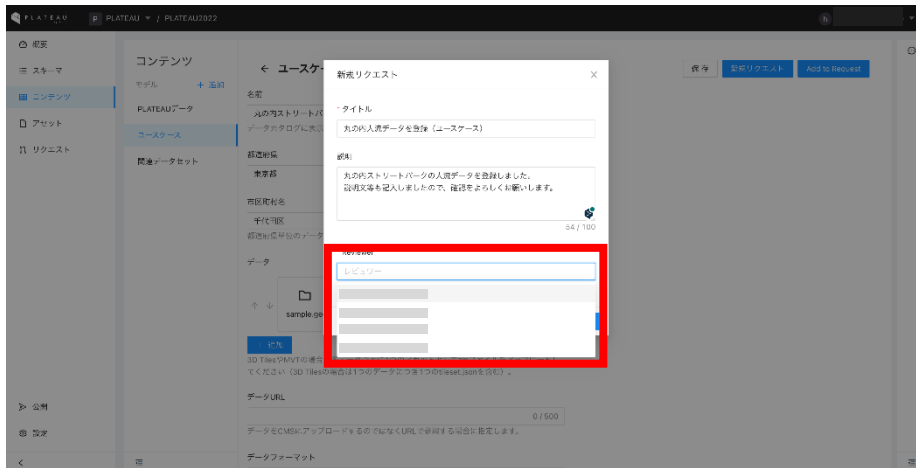
c. リクエストのタイトルを記入

「ユースケース」「PLATEAUデータ」などを記入すると、レビューがわかりやすくなる。



d. レビューに登録データについて説明を簡単に入力する。

- e. レビューーを選択します。ドロップダウンをクリックし、レビューー権限のあるユーザーを表示する。



- f. レビューーを選択し、OKをクリックする。

2. 公開リクエスト作成前のチェックリスト

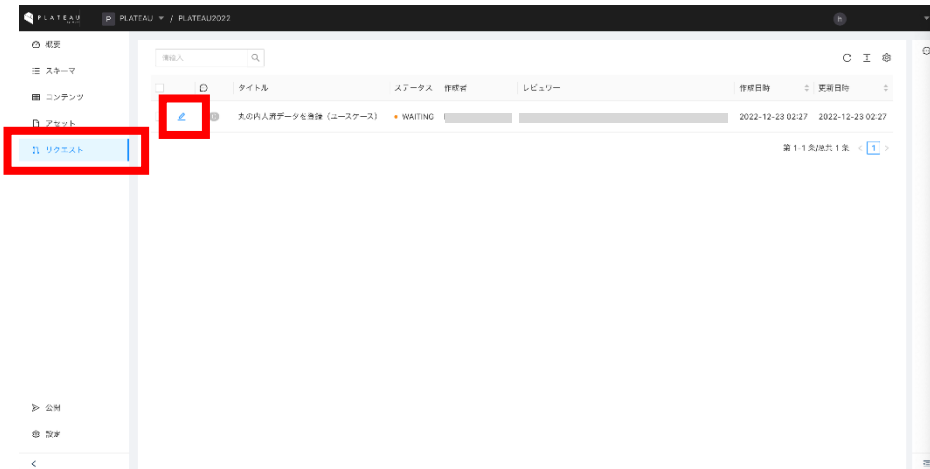
表 チェックリスト

項目1	項目2	確認
全て入力が完了している	都道府県	<input type="checkbox"/>
	市区町村名	<input type="checkbox"/>
	仕様書のバージョン	<input type="checkbox"/>
	CityGML	<input type="checkbox"/>
	目録ファイル	<input type="checkbox"/>
	CityGMLを3D Tilesに変換する	<input type="checkbox"/>
	平面直角座標系	<input type="checkbox"/>
	政令指定都市で3D Tilesを区に分割	<input type="checkbox"/>
	説明文 建築物モデル	<input type="checkbox"/>
	説明文 道路	<input type="checkbox"/>
	説明文 都市設備	<input type="checkbox"/>
	説明文 植生	<input type="checkbox"/>
	説明文 土地利用	<input type="checkbox"/>
	説明文 土砂災害警戒区域	<input type="checkbox"/>
	説明文 都市計画決定情報	<input type="checkbox"/>
	説明文 浸水想定区域（洪水）	<input type="checkbox"/>
	説明文 浸水想定区域（津波）	<input type="checkbox"/>
	説明文 浸水想定区域（高潮）	<input type="checkbox"/>
	説明文 浸水想定区域（内水）	<input type="checkbox"/>
	CityGMLデータの変換が無事完了し、入力項目を満たしている	<input type="checkbox"/>
	「品質検査・3D Tiles 変換 ステータス」が「完了」になっている	<input type="checkbox"/>
すべてのフィールドにFMEでの 変換後のファイルが設定されて いる	建築物モデル	<input type="checkbox"/>
	道路	<input type="checkbox"/>
	都市設備	<input type="checkbox"/>
	植生	<input type="checkbox"/>
	土地利用	<input type="checkbox"/>
	土砂災害警戒区域	<input type="checkbox"/>
	都市計画決定情報	<input type="checkbox"/>
	浸水想定区域（洪水）	<input type="checkbox"/>
	浸水想定区域（津波）	<input type="checkbox"/>
	浸水想定区域（高潮）	<input type="checkbox"/>
	浸水想定区域（内水）	<input type="checkbox"/>
上記ファイル全てをプレビューし、ジオメトリに問題がないか、属性に問題がないかを確認している		<input type="checkbox"/>
TODO:目録ファイル		<input type="checkbox"/>
それぞれのフィールドに	説明文 浸水想定区域（洪水）	<input type="checkbox"/>
	説明文 浸水想定区域（津波）	<input type="checkbox"/>
	説明文 浸水想定区域（高潮）	<input type="checkbox"/>
	説明文 浸水想定区域（内水）	<input type="checkbox"/>

(6) その他状況に応じて必要な操作

1. 公開リクエストのステータスを確認

- a. 画面左サイドバーの「リクエスト」を選択し、公開リクエスト一覧を表示する。



- b. 閲覧するリクエストのペンアイコンをクリックし、公開リクエストの詳細ページへ移動する。



- c. 公開リクエスト詳細ページでは、コメント機能を利用して公開リクエストに対するコメントを閲覧・投稿できる。



- d. 変更リクエストがレビューによって承認された場合、画面右側に「APPROVED」と表示される。

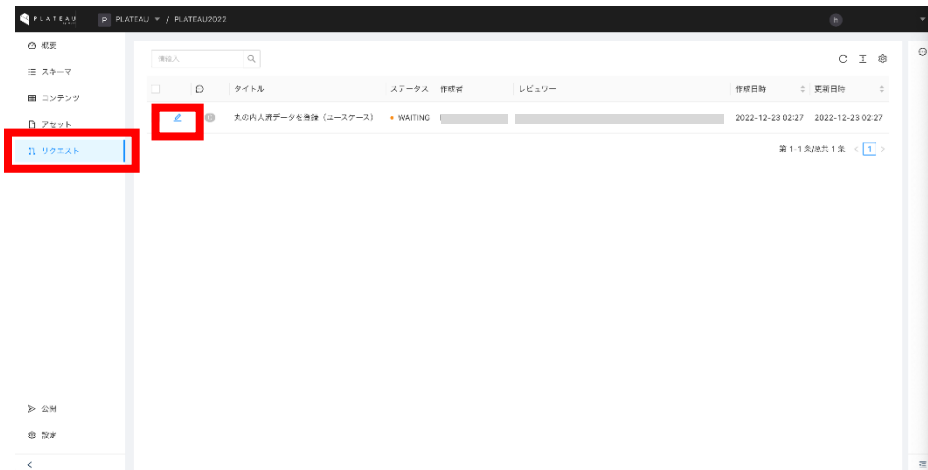


- e. ステータスが「APPROVED」公開リクエストに紐づいたアイテムは自動的に公開される。

2. 公開リクエストのクローズ

誤って公開リクエストを作成してしまった場合、公開リクエストをクローズする。

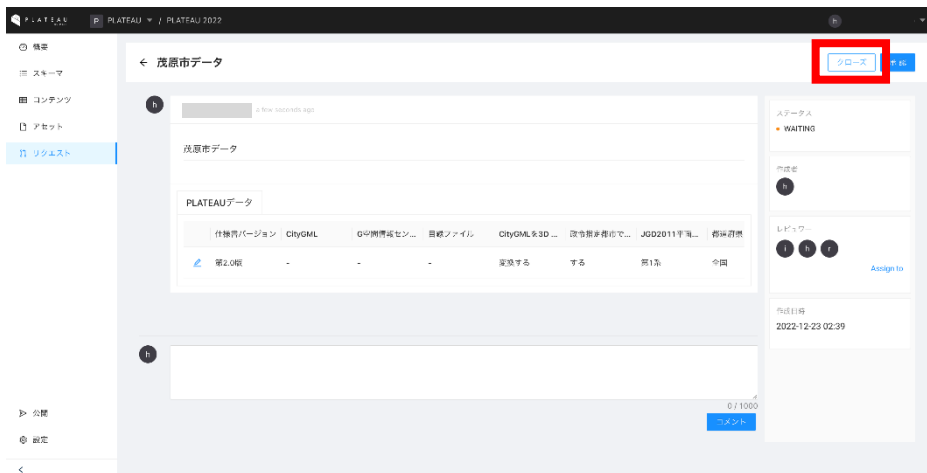
- 画面左サイドバーの「リクエスト」を選択し、公開リクエスト一覧を表示する。



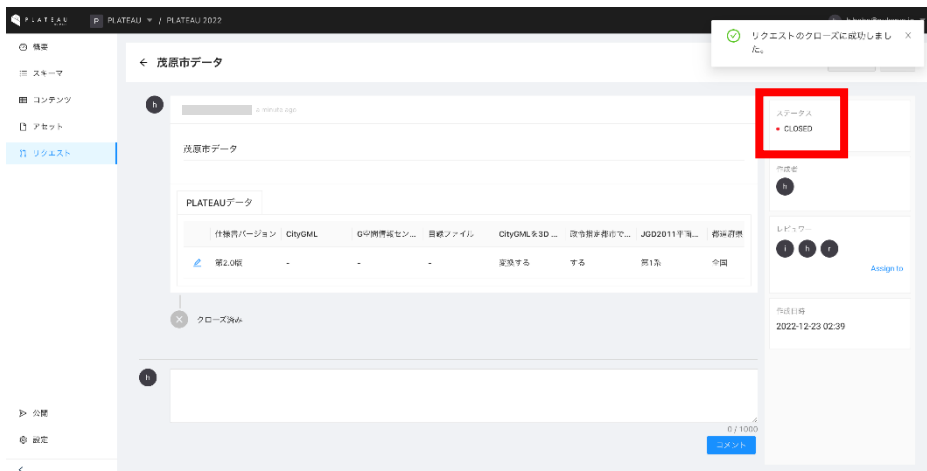
- 閲覧するリクエストのペンアイコンをクリックし、公開リクエストの詳細ページへ移動する。



- c. 画面右上の「クローズ」ボタンをクリックし、公開リクエストをクローズする。



- d. 公開リクエストのステータスが「CLOSED」になっていることを確認する。



3.2 PLATEAU Editorの利用方法

ここではPLATEAU Editorの使用方法について述べながら、PLATEAU VIEW 2.0におけるPLATEAU VIEWをPLATEAU Editor上で作成・公開する手順を解説する。なお、PLATEAU Editorで採用されているOSSであるRe:Earthは非常に多くの機能を持っているが、ここではPLATEAU VIEWを構築するために関係する項目のみを説明する。

それ以外のRe:Earthの全ての機能をここで紹介することはできないので、必要に応じて以下のURLも併せて参照されたい。<https://docs.reearth.io/ja/>

3.2.1 ワークスペース

(1) ワークスペースの概要

PLATEAU Editorのワークスペースは複数のユーザーが共同でプロジェクトを管理することができる。ワークスペースの設定では、共有メンバーの招待や削除、役割の選択、プロジェクトの管理、ワークスペースに関連する素材（アセット）の保存などを行うことができる。

(2) ワークスペースの種類

PLATEAU Editorでは、2種類のワークスペースが用意されている。主な違いは以下の表の通り。

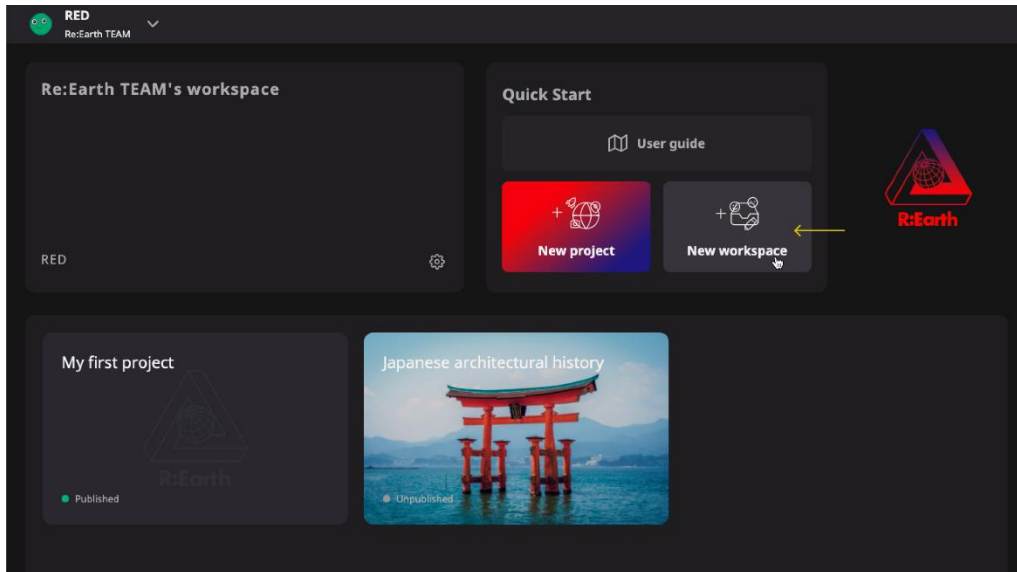
表 PLATEAU Editorのワークスペースの種類

項目	個人のワークスペース	共同ワークスペース
作成	PLATEAU Editorのアカウントが作成された際に自動で作成される。	ダッシュボードや設定から手動で作成する必要がある。
数	1ユーザーにつき1つ。	いくつでも作成することができる。
削除	ユーザーのアカウントが削除された際に自動で削除される。	ワークスペースの所有者は、管理しているプロジェクトがない場合ワークスペースを削除できる。
メンバーの追加	メンバーの追加はできない。	複数人メンバーを追加することができる。

（3）ワークスペースの作成

PLATEAU VIEWを構築する場合、複数人で共同作業することが前提となるため、個人ワークスペースではなく、共同ワークスペースで作業することを推奨する。そのため、まず最初にワークスペースを作成することから始める。

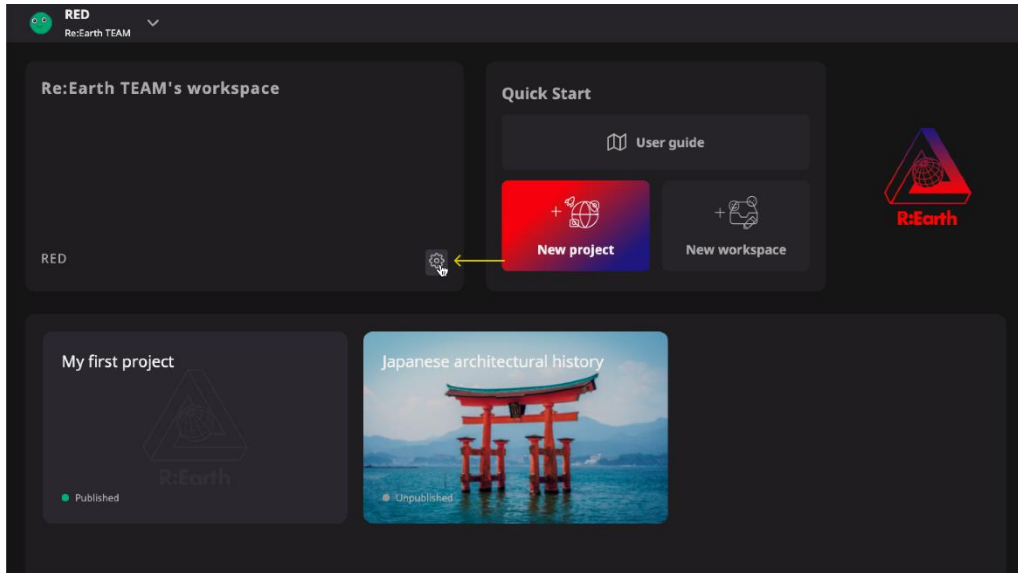
ダッシュボードの「クイックスタート」→「新規ワークスペース作成」からワークスペースの作成が行える。



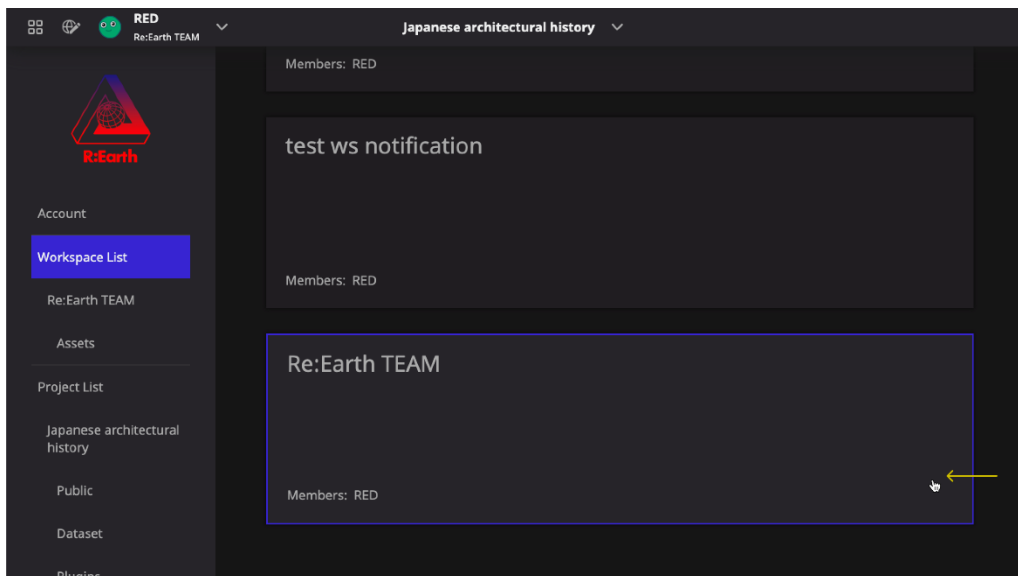
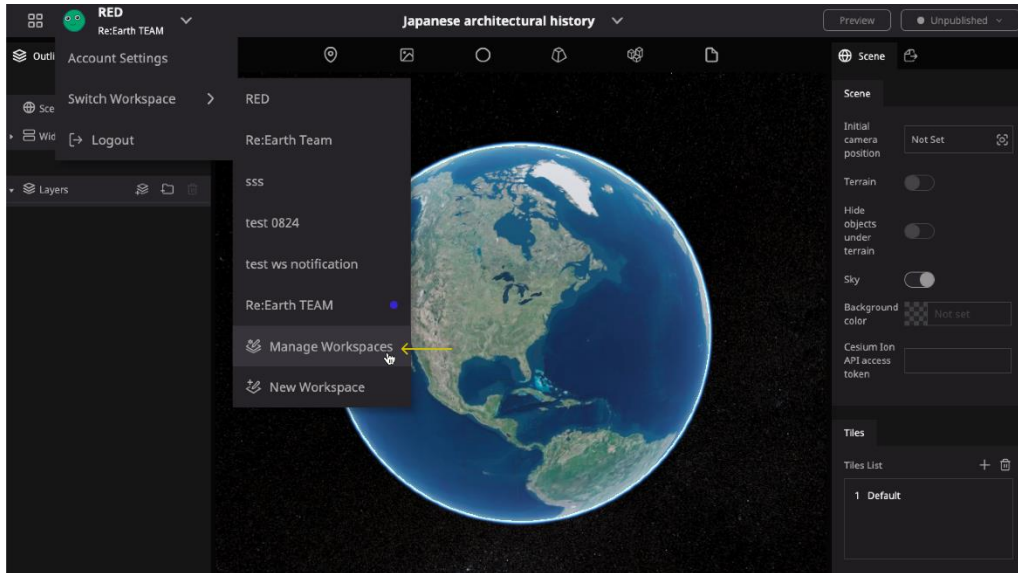
(4) ワークスペースの設定

ワークスペースの設定画面では、さまざまな設定の確認・変更をすることができます。
設定ページにアクセスする方法は2通りある。

1. ダッシュボード → ワークスペースブロック → ⚙️ ボタン

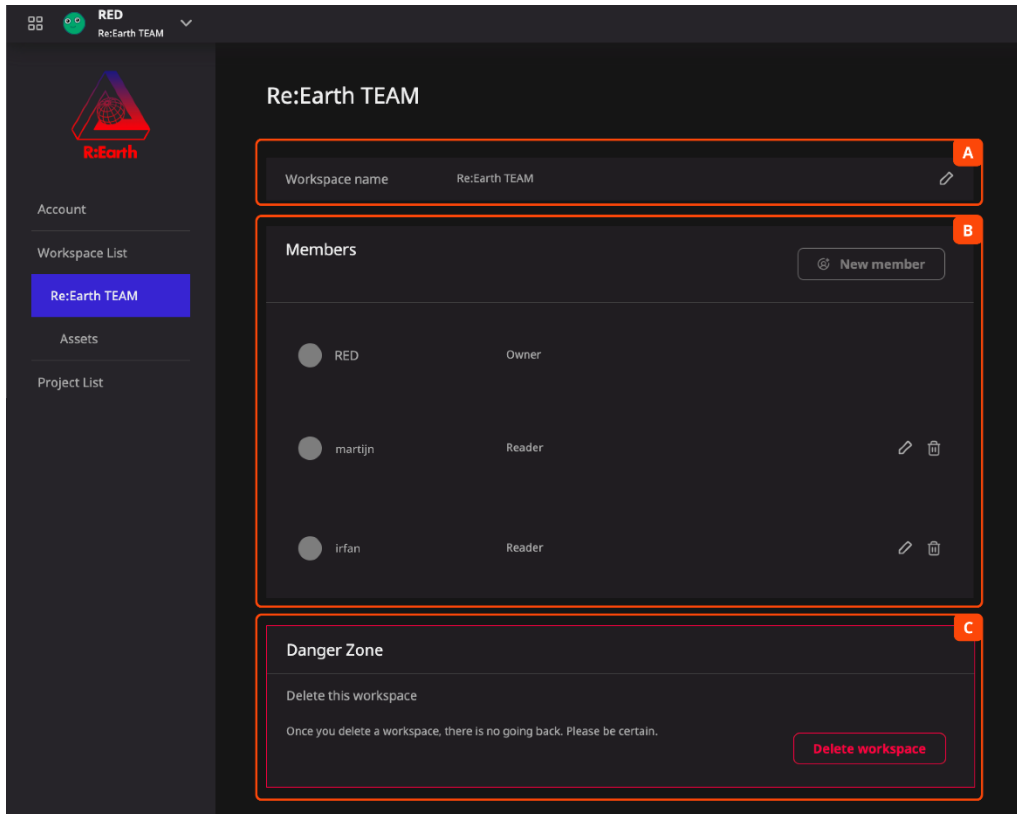


2. ヘッダーのユーザーメニュー → ワークスペース → ワークスペースの管理 → ワークスペース一覧
→ ワークスペースを選択



ワークスペースの設定ページでは、ワークスペースの情報の編集、メンバーの管理、ワークスペースの削除を行うことができる。

- A) ワークスペース基本情報はPLATEAU Editor内で表示されるワークスペースの基本情報である。現在はワークスペース名だけだが、将来的に多くの編集可能プロパティをサポートする予定である。
- B) メンバーではワークスペースに招待するメンバーの管理を行う。
- C) 危険項目では、ワークスペースへのアクセスに関する項目が含まれます。ワークスペースを削除するにはここから操作する。



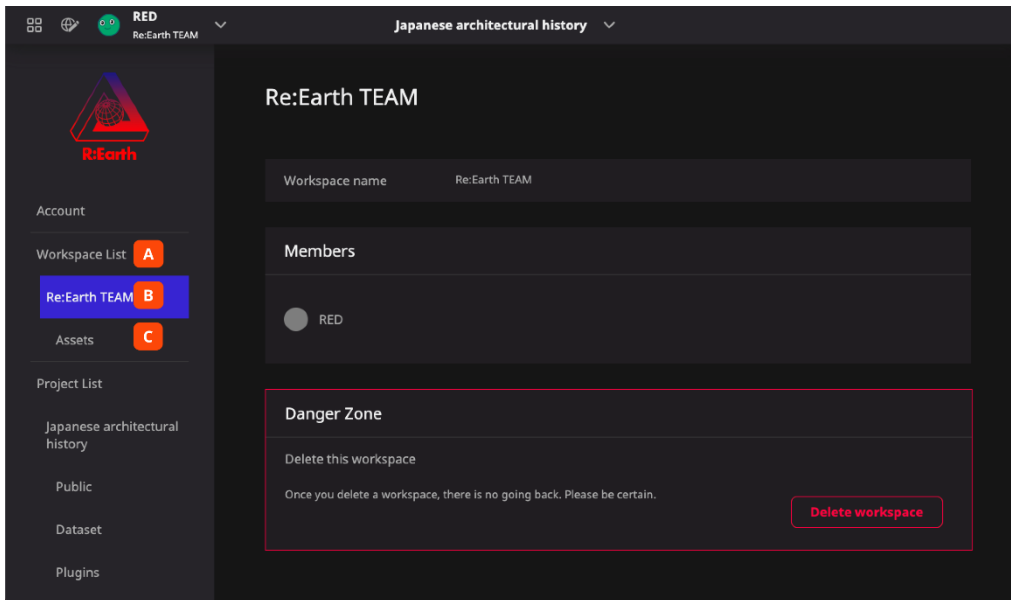
左側のナビゲーションバーでは、現在のワークスペースに関する設定をしたり、全てのワークスペースの管理をすることができます。

- A) ワークスペース一覧では自分が管理しているすべてのワークスペースの管理や切り替えを行うことが可能である。
- B) 現在のワークスペースではメンバーの招待(※1)やワークスペース(※2)の削除を行うことができる。
- C) アセットでは現在のワークスペースに登録されているアセット(※3)を管理することができる。

※1：マニュアル <https://docs.reearth.io/ja/user-manual/project-and-workspace/workspace/managing-members>

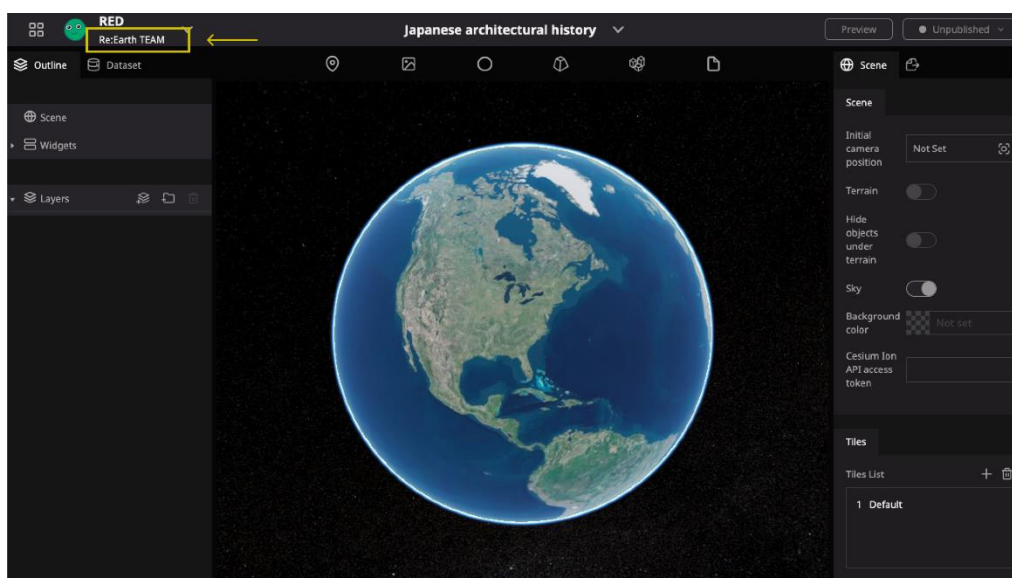
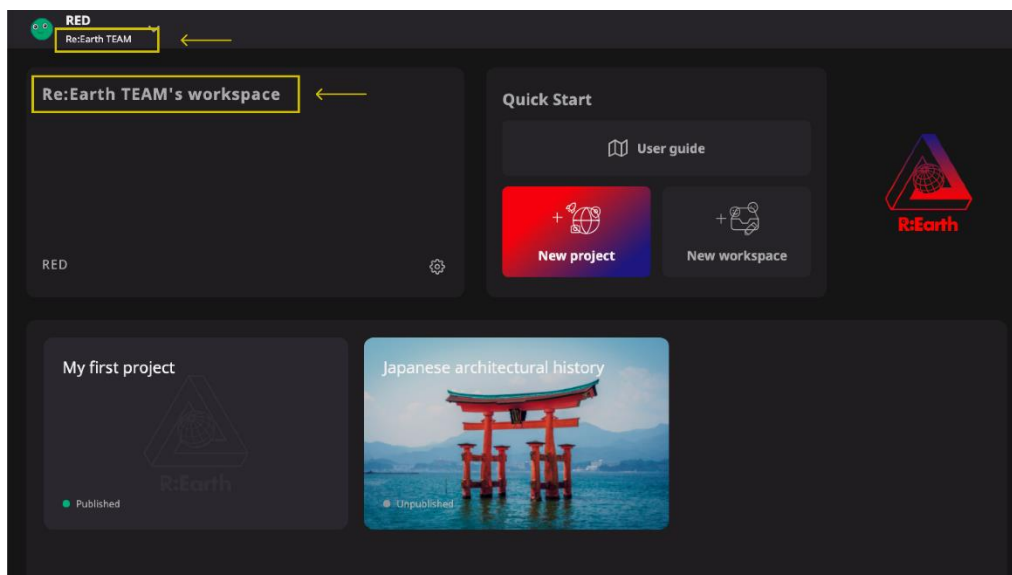
※2：マニュアル <https://docs.reearth.io/ja/user-manual/project-and-workspace/workspace/settings-and-deleting#delete-team-workspace>

※3：マニュアル <https://docs.reearth.io/ja/user-manual/project-and-workspace/workspace/assets-library>



(5) 作業中のワークスペース

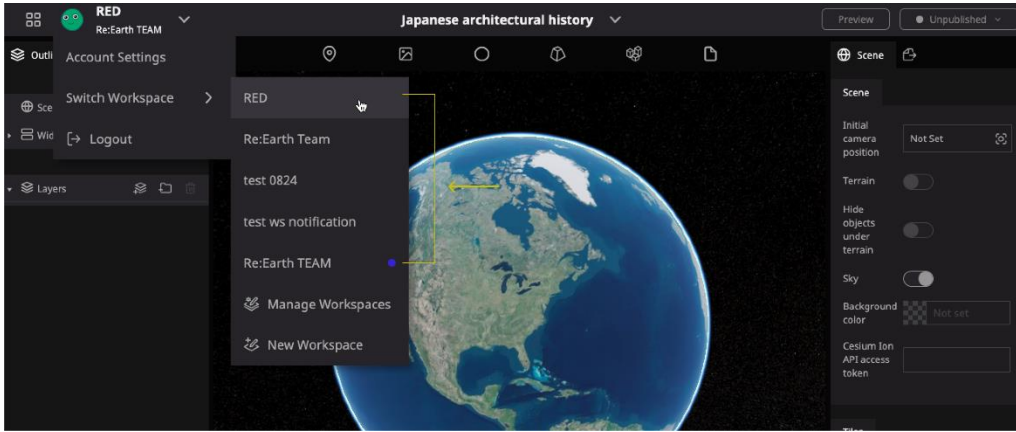
プロジェクトはワークスペース毎に管理しているため、異なるワークスペースに切り替えると異なるプロジェクトが表示される。作業したいプロジェクトを見失ったとしても慌てずに、自分が目的のワークスペースにいるかどうかを確認する。現在のワークスペースは、ヘッダーの左上で確認することができる。



(6) ワークスペースの切り替え

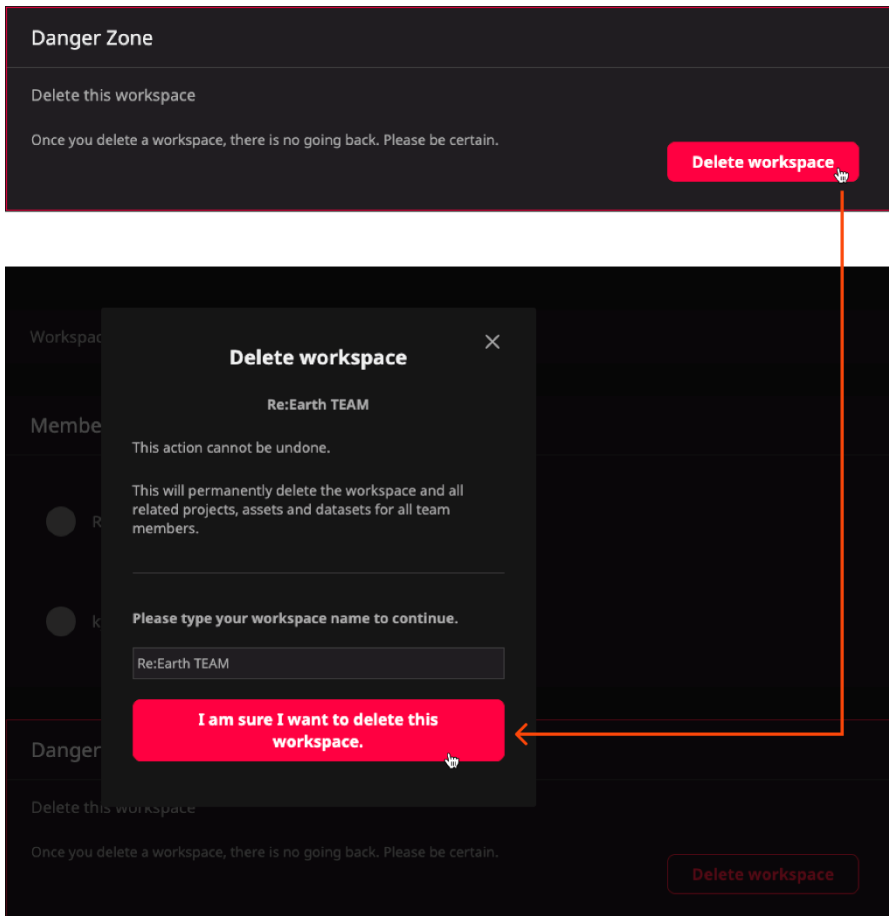
作業したいワークスペースにいない場合や、他のメンバーと共同作業する場合にヘッダーメニューのワークスペースから別のワークスペースに切り替えることができる。

ヘッダーメニュー → ワークスペース → 目的のワークスペースを選択



(7) ワークスペースの削除

意図せずにプロジェクトを削除しないように、ワークスペースの削除はワークスペース内にプロジェクトがない場合（全て削除済み）のみ可能である。また、ワークスペースの管理者のみが削除することができる。



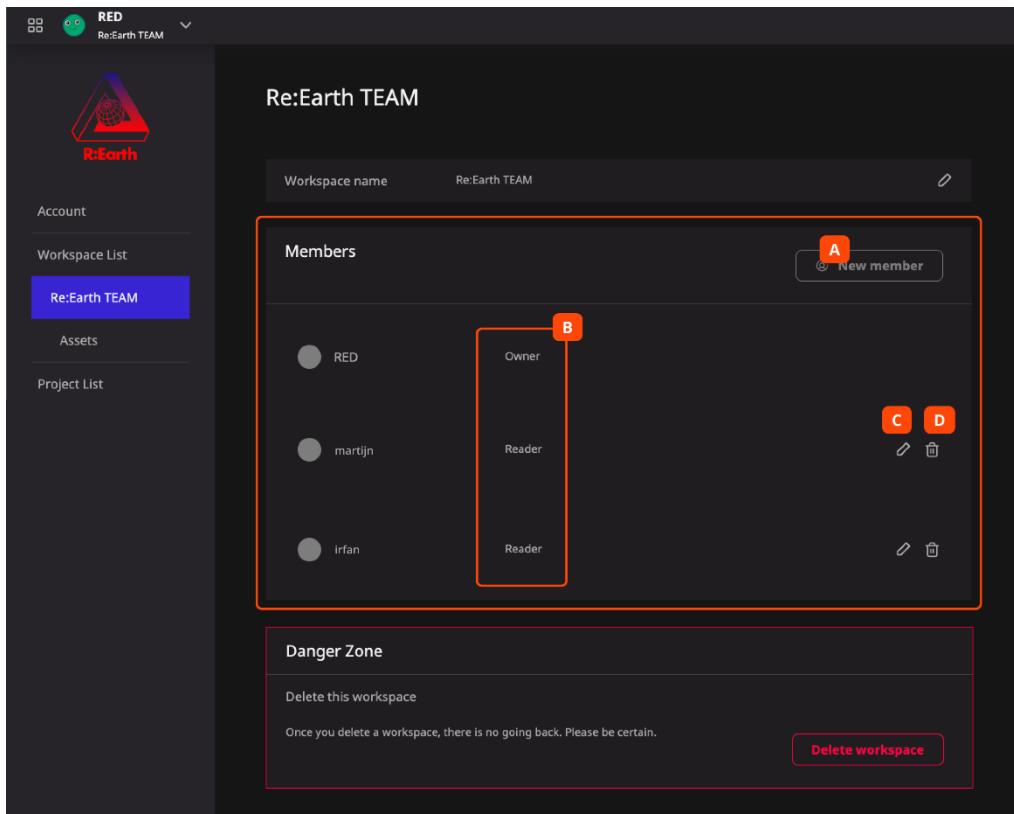
3.2.2 メンバーの招待

PLATEAU Editorではワークスペースにメンバーを招待することで、チームでプロジェクトを編集することができる。同じワークスペースにいるメンバーは同じプロジェクトを編集することができる。

(1) メンバーの概要

ワークスペースの設定ページで、ワークスペースのメンバーの管理をすることができる。

- A) 新規メンバーボタンで、メンバーの追加を行う。
- B) メンバーの権限ボタンでは、共有中のメンバーの権限を確認することができる。
- C) 権限変更ボタンでは、ワークスペース内でのメンバーの権限を変更することができる。
- D) メンバーの削除ボタンをクリックすると、メンバーをワークスペースから削除することができる。



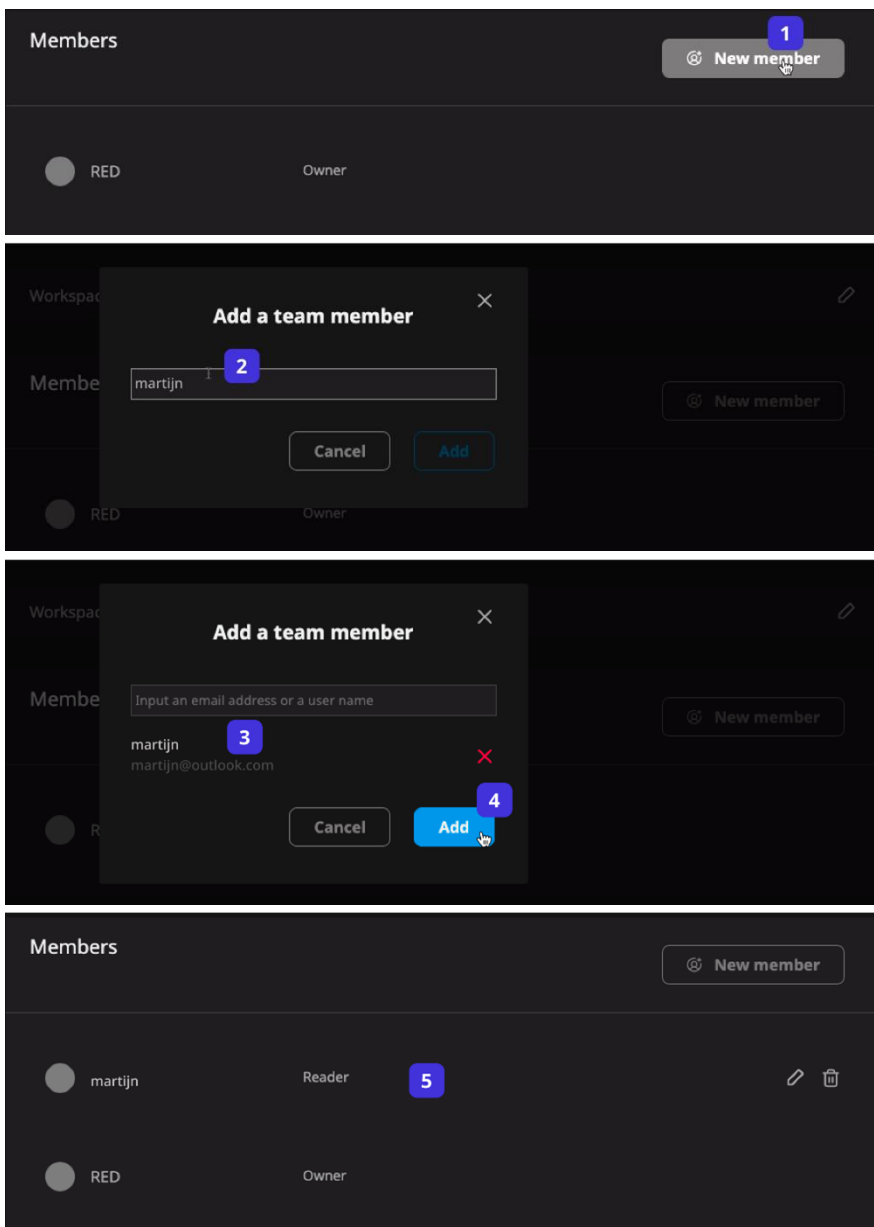
(2) メンバーの招待手順

以下の手順でメンバーをワークスペースに招待することができます。

1. 新規メンバーをクリックすると、メンバーを追加モーダルが表示される。
2. 招待したいユーザーのユーザー名もしくはメールアドレスを入力する。
3. 招待したいユーザーが正しく表示されているか確認する。
4. ユーザーが正しいことを確認したら追加ボタンをクリックする。
5. メンバーリストにメンバーが追加される。

※ チームワークスペースに追加できるのは、PLATEAU Editorアカウントを持っているユーザーだけである。

※ CAUTION: ユーザーネームは大文字と小文字を区別しますので、正確に入力する



(3) 権限の種類

メンバーの権限は以下の3種類存在する。

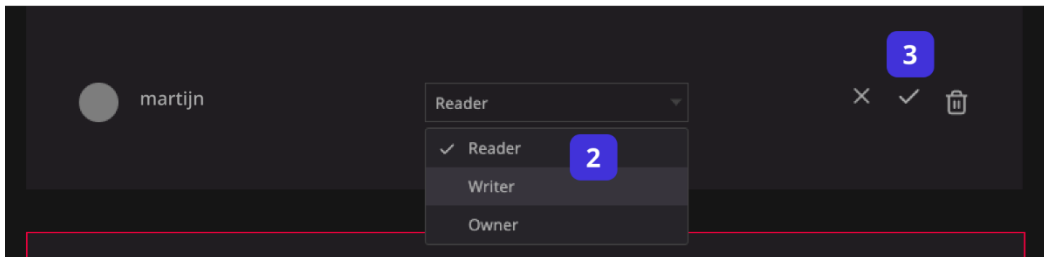
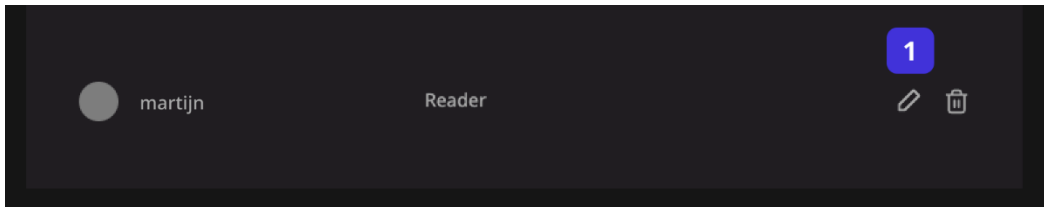
- **オーナー**：ワークスペースの最高権限を持ち、ワークスペースに関する全ての設定を操作することができる。
- **編集**：ワークスペース内のプロジェクトの閲覧・編集・公開をすることができ、ワークスペース内にメンバーを追加することもできる。
- **閲覧**：全てのプロジェクトを閲覧することができるが、編集及び設定を変更することはできない。

(4) メンバーの権限の変更

各メンバーに対して、特定の操作を制限するためにそれぞれに権限を与えることができる。メンバーの権限を変更するには次の操作を行う。

1. メンバーリストの ペンボタンをクリックする。
2. ドロップダウンリストから、付与する権限を選択する。
3. ☒ ボタンをクリックして確定する。

※ 権限の変更ボタンが表示されていない場合、自分の現在の権限では編集できないことを示している。自身の権限を変更してもらうためにワークスペースのオーナーに連絡する。



3.2.3 プロジェクトの作成

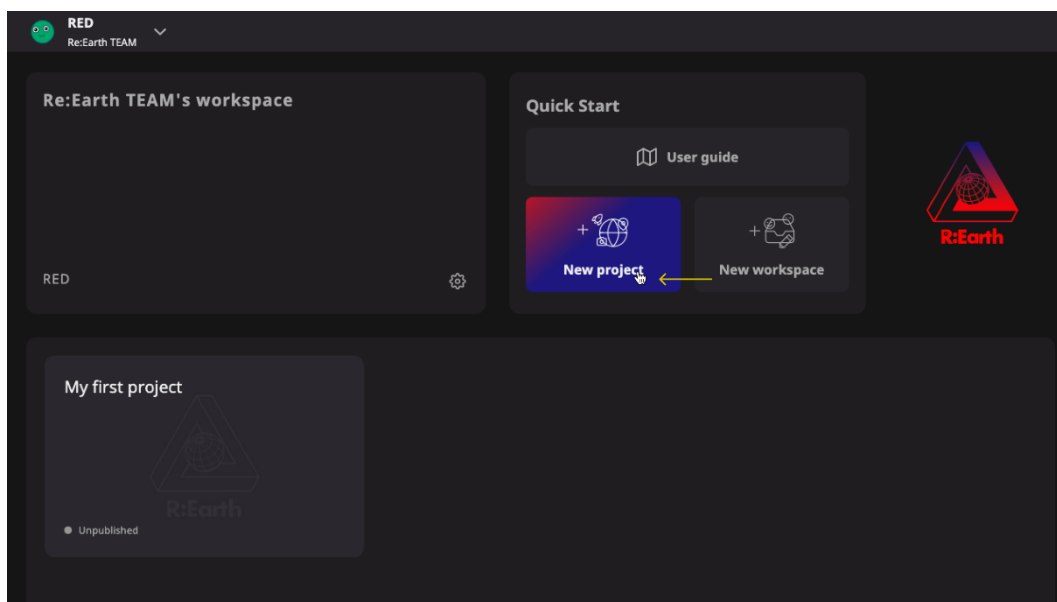
(1) プロジェクトの概要

プロジェクトでは、編集・公開・データのインポートなどが可能である。プロジェクトを作成すると、編集画面を開くことが可能。編集画面にはデジタル地球儀が表示されます。PLATEAU Editorではこれをシーンと呼ぶ。

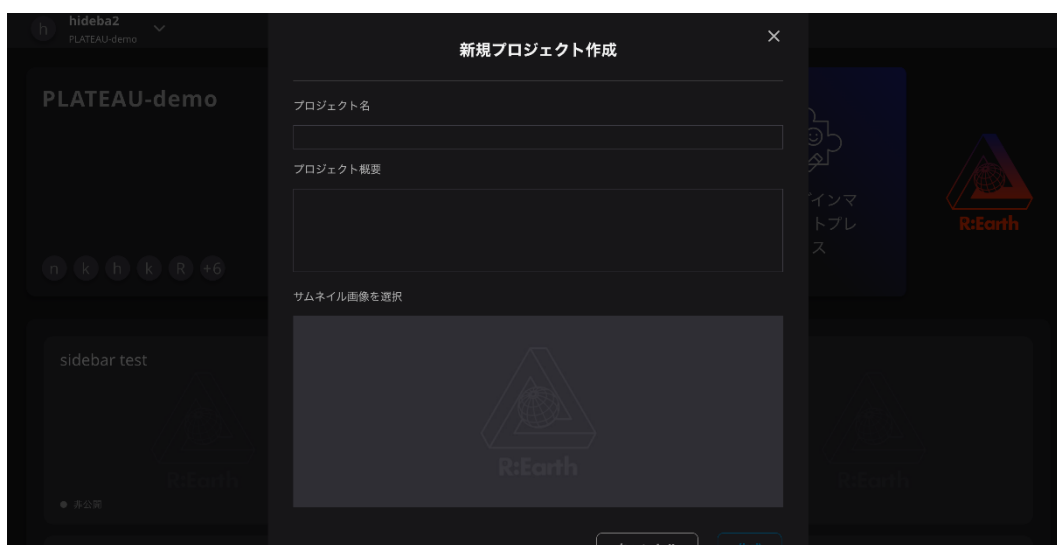
(2) プロジェクトの作成

新規プロジェクトを作成する方法は以下の通りである

- ダッシュボード → クイックスタート → 新規プロジェクト作成



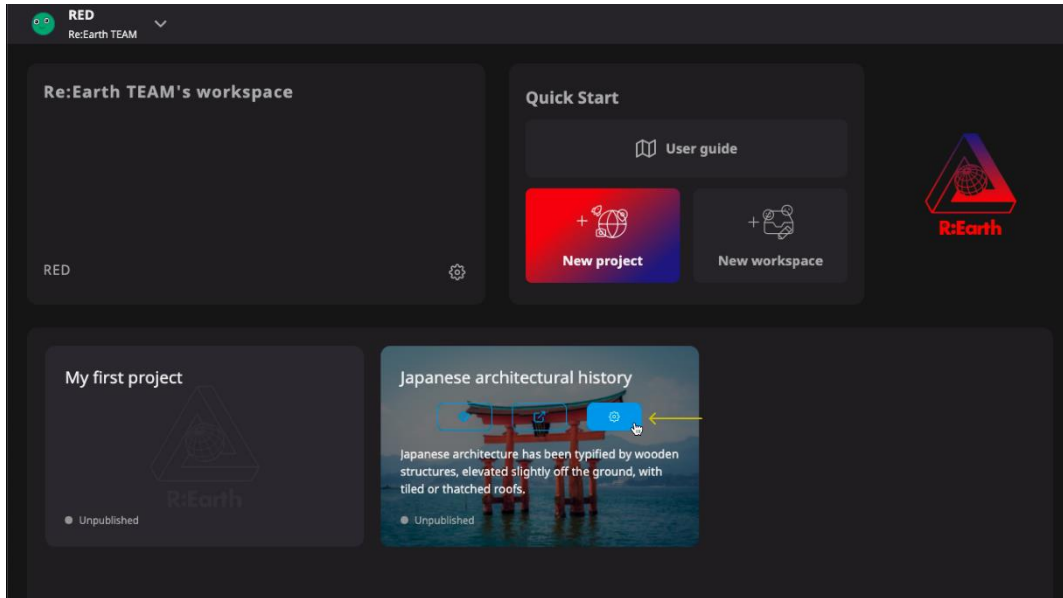
- モーダルに情報を記入して作成完了



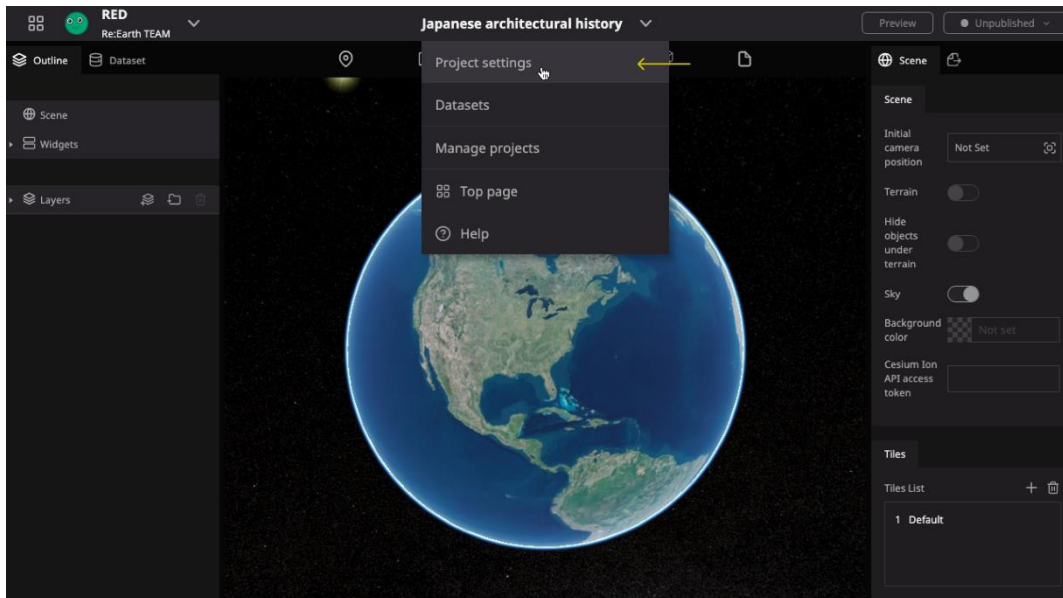
(3) プロジェクトの設定

設定へアクセスする方法は2つある。

1. ダッシュボード → プロジェクトカード → ⚙️ ボタン

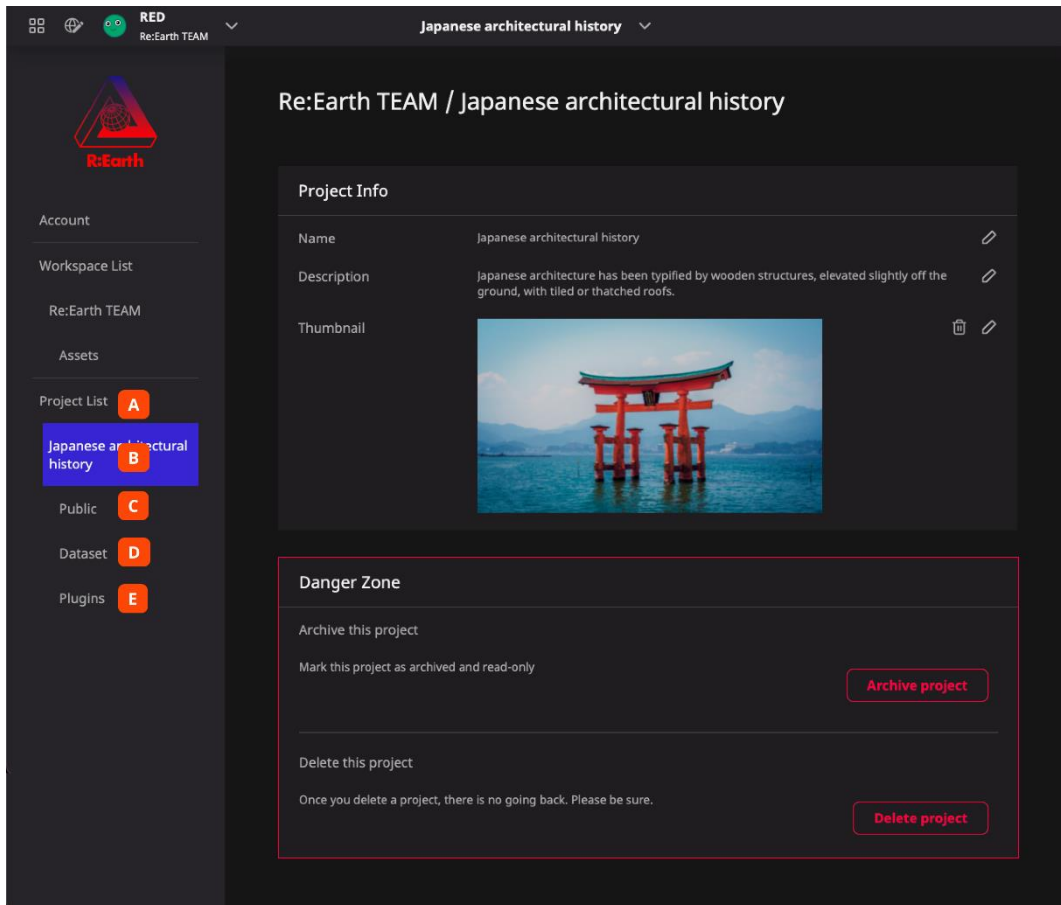


2. ヘッダー → プロジェクトメニュー → プロジェクト設定



左側のナビゲーションバーでは、現在のプロジェクトに関する様々な設定をしたり、すべてのプロジェクトを管理することができる。

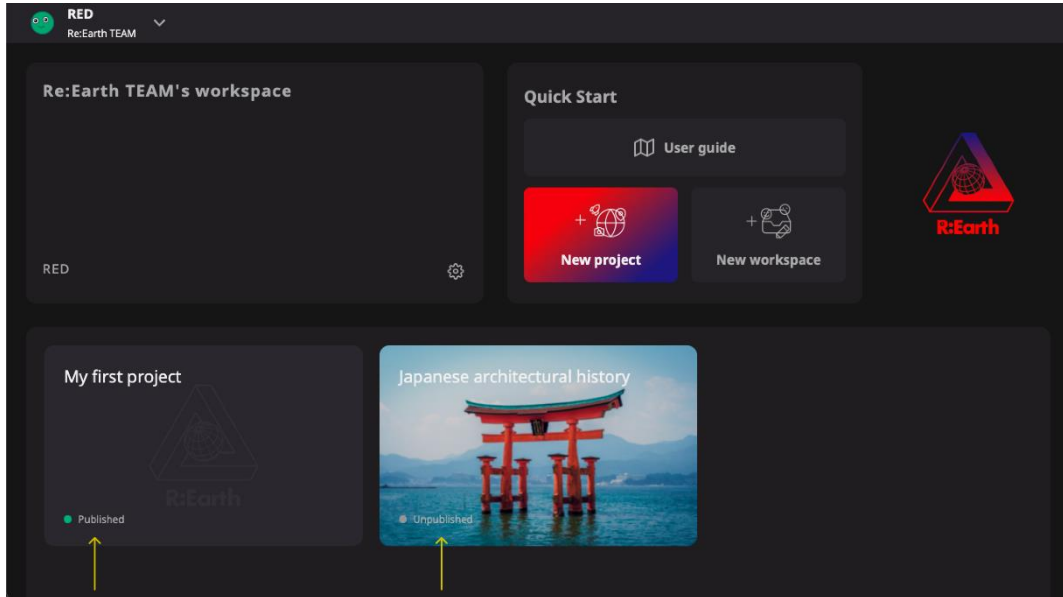
- A) プロジェクトリスト：現在のワークスペースにあるすべてのプロジェクトを表示する。このページでは、さまざまなプロジェクトを選択して、それぞれの設定にアクセスできる。
- B) 現在のプロジェクト：プロジェクトの基本情報の変更や、プロジェクトの非編集化や削除を行うことができる。詳しくは、設定・非編集化・削除を参照されたい。
- C) 公開設定：プロジェクトの公開に関する設定を行うことができます。詳しくは、公開設定を参照されたい。
- D) プラグイン：現在のプロジェクトにインポートされているプラグインを、管理することができる。詳しくは、プラグインライブラリ および [プラグインセクション]を参照されたい。



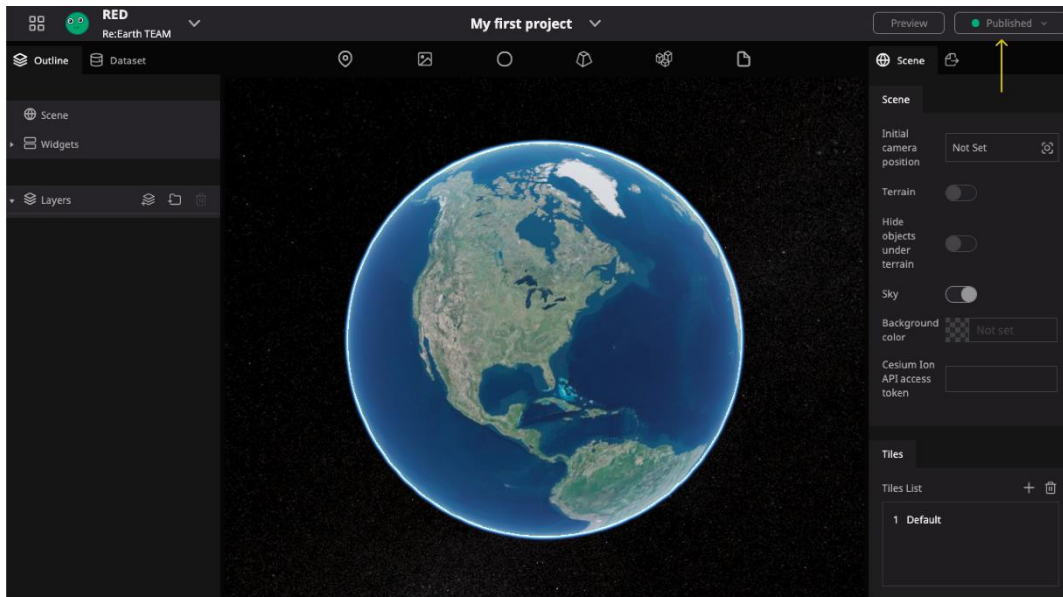
(4) プロジェクトの公開状況

プロジェクトが完成したらプロジェクトをインターネット上に公開することができる。プロジェクトの公開状況は、2つの方法で確認することができる。

ダッシュボードの、各プロジェクトカードで公開状況を確認することができる。



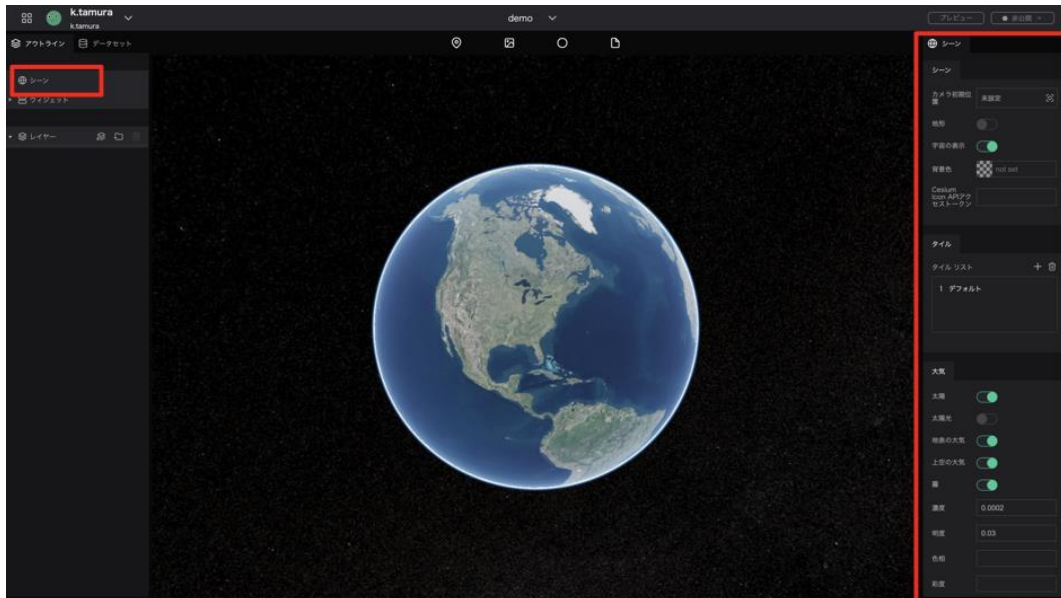
編集画面の右上に、公開状況が表示されており、いつでも確認・変更することが可能である。



3.2.4 シーンの設定

シーンでは、デジタルアースのベースマップやカメラの初期位置など、プロジェクト全体の設定等を行うことができる。またレイヤーやインフォボックスの配置などを行える。

基本的な考え方として、画面左のパネル（アウトライン）または地球儀上で編集対象を選択し、画面右のパネルで設定を変更するという操作が行える。

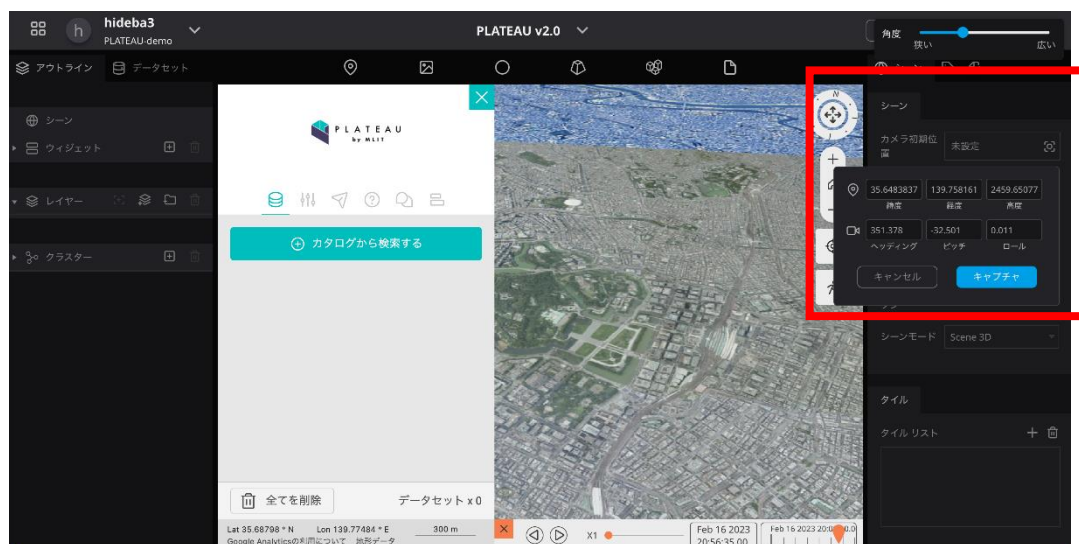


ここではPLATEAU VIEW 2.0におけるシーン全体の設定を行う。左サイドバー上部の「シーン」を選択すると、右パネルにシーンの設定が表示される。

1. 初期カメラ位置の設定

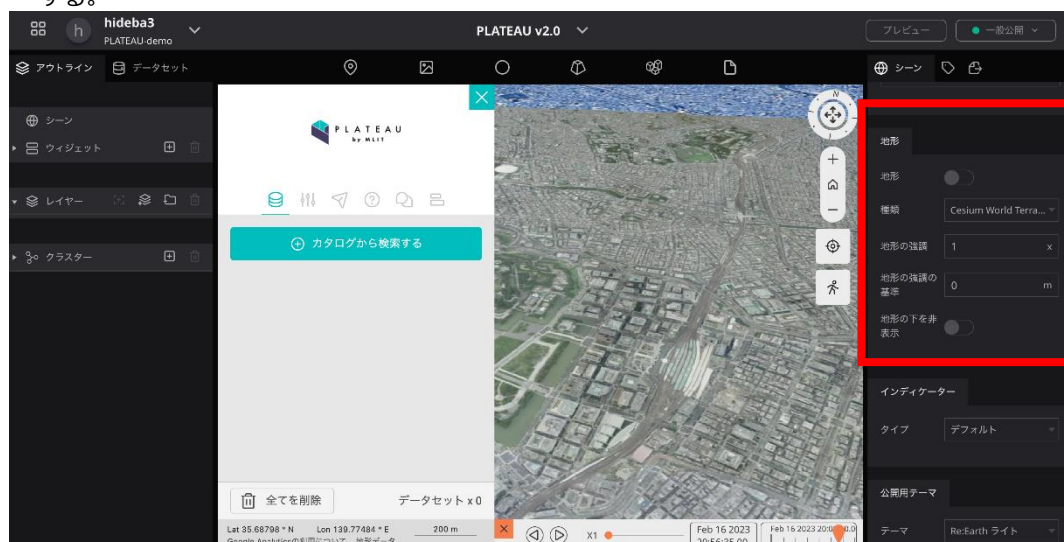
デフォルトでは、ページロード時に北アメリカ全土が表示される設定になっている。

「カメラ初期位置の設定」では、ページロード後、最初に表示されるカメラの位置を設定できる。「キャプチャ」をクリックすると、その位置と画角がカメラ初期位置に設定される。



2. タイル・地形の設定

タイルとは地表のベースマップの画像のことである。また地形とは地表の標高などに基づく三次元的形状のことである。デフォルトでは、デフォルト（Cesiumのタイル）のタイルと、Cesium World Terrainを使用しているため、PLATEAU独自のTerrainのURLを設定する。



3. インディケーターの設定

インディケーターとは、地物選択時に表示されるカーソルの画像のことである。デフォルトでは緑色のインディケーターが表示されるが、PLATEAU VIEW 2.0では「十字線」に変更している。

4. 高度な設定

「先進的機能を使用」をONにする。

3.2.5 プラグインのインストール

VIEW2.0における「サイドバー」「ナビゲーター」「ストーリー」「歩行者モード」などのウィジェットは全てPLATEAU Editorのプラグインによって実装されている。ここでは、プラグインをインストールし、ウィジェットとして配置する。

プラグインは以下のURLで公開されているため、Zipファイルをダウンロードする。

<https://github.com/Project-PLATEAU/PLATEAU-VIEW-2.0/releases>

編集画面のメニューにある「プラグイン」からプラグインの設定へ移動する。



左側のメニューにある「プラグイン」を選択し、「個人インストール済み」タブに移動する。「PCからZipファイルをアップロード」より、先ほどダウンロードしたZipファイルを選択し、プラグインをアップロード・インストールする。



「PLATEAU Plugin」がインストール済みであることを確認する。

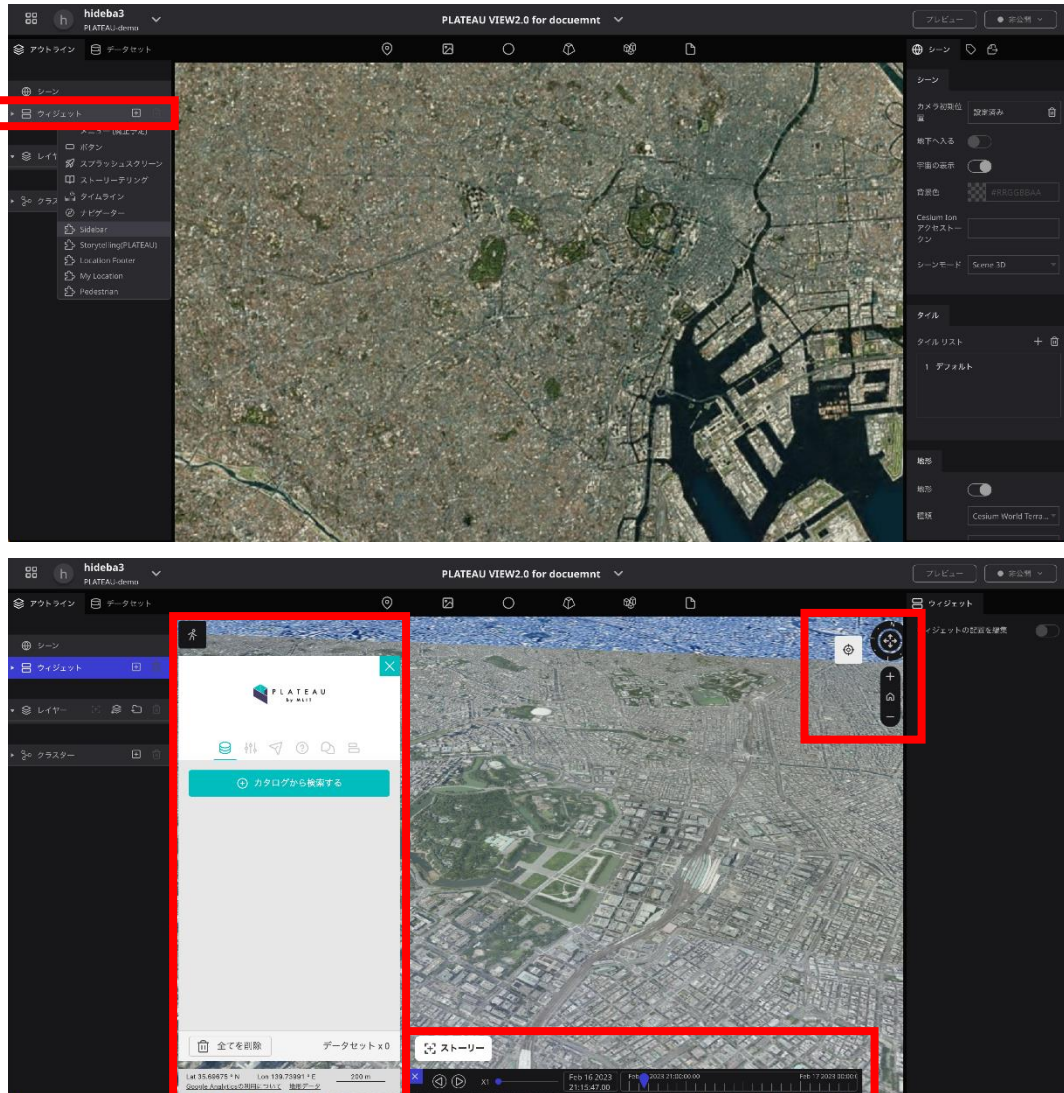


3.2.6 ウィジェットの設定

ウィジェットでは、ウインドウ上に配置される機能の設定が可能である。また、ウィジェット配置システムによって、それらのウィジェットの配置を自由に設定することができる。

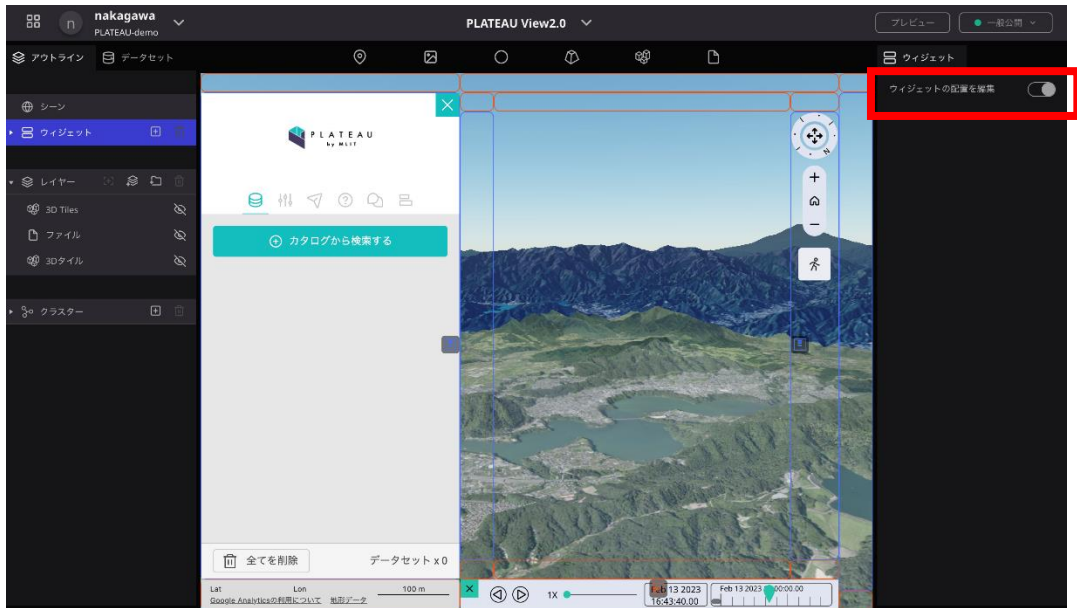
(1) ウィジェットの追加

左パネルより、インストール済みのウィジェットをシーンへ追加する。



(2) ウィジェットの配置

ウィジェット配置モードをONにすることで、自由にウィジェットの配置を調整できる。サイドバーを左側に、ナビゲーター等を右側に、タイムライン等を下方に配置しPLATEAU VIEWを構成する。

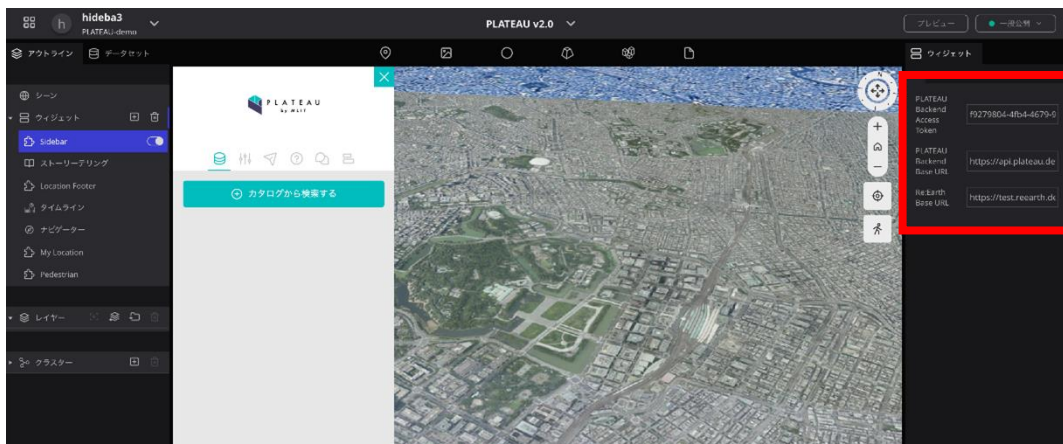


(3) ウィジェットの各種設定

それぞれのウィジェットは設定項目を有しており、様々な設定を行うことができる。左パネル上部でウィジェットを選択すると、右パネルに選択中のウィジェットの設定項目が表示される。

PLATEAU VIEW 2.0では、特に以下のウィジェットで設定を行う。

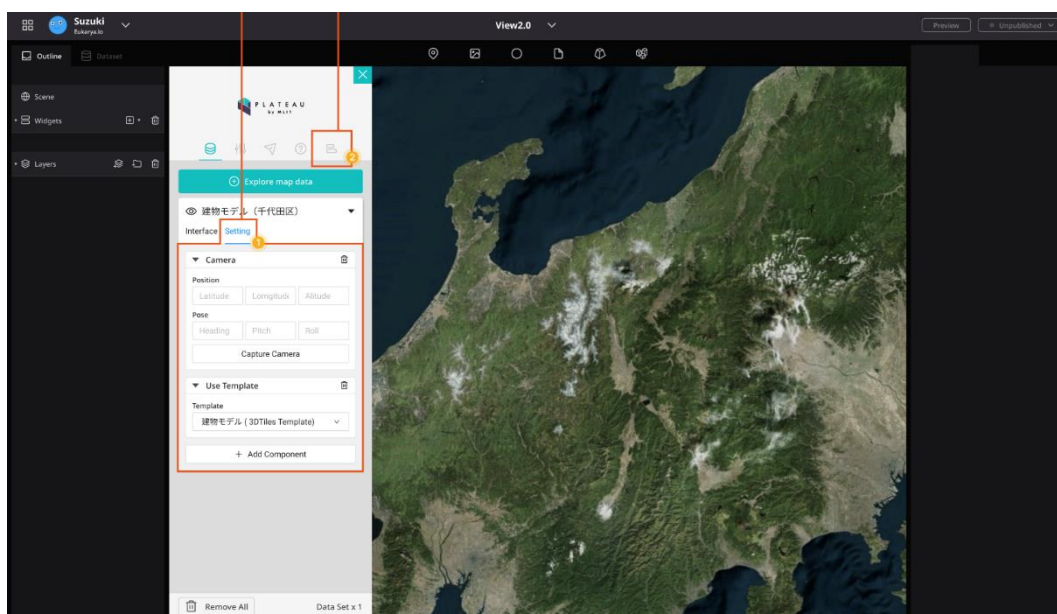
- サイドバー
 - **Backend URL:** サイドカーサーバーのエンドポイント。2.1のTerraform実行結果として得た「plateauview_sidecar_url」の値を入力。
 - **Backend Access Token:** PLATEAU Viewサーバーの認証に使用されるアクセストークン。2.1のTerraformの実行結果として得た「plateauview_sidebar_token」の値を入力する。
 - **Project Name:** 共有機能などのデータを保存するための、PLATEAU CMS上のプロジェクトエイリアス名。「2.1.9 PLATEAU CMSのセットアップ」で作成したPLATEAU CMSのPLATEAU VIEW用プロジェクトのプロジェクトエイリアスを入力する。
 - **Data Catalog URL:** データカタログ用サイドカーサーバーのエンドポイント。2.1のTerraform実行結果として得た「plateauview_sidecar_url」の値を入力する。通常、PLATEAU Backend Base URLの設定と同じ値になる。
 - **Data Catalog Project Name:** データカタログの参照元となるPLATEAU CMS上のプロジェクトのエイリアス名。「2.1.9 PLATEAU CMSのセットアップ」で作成したPLATEAU CMSのPLATEAU関連データセット用プロジェクトのプロジェクトエイリアスを入力する。
 - **Re:Earth Project Public URL:** 共有URL機能を利用した際のURL。2.1のTerraformの実行結果として得た「plateauview_reearth_url」にプロジェクトエイリアスをサブドメインとして付加したURL。サブドメインは下記「3.3.9 プロジェクトの公開」にて設定するプロジェクトエイリアス。例：
「plateauview_reearth_url」が「<https://reearth.example.com>」かつ、プロジェクトエイリアスが「plateauview」ならば
「<https://plateauview.reearth.example.com>」となる。
 - **Enable Publication:** データカタログでのデータセット公開時（公開ボタン押下時）にG空間情報センターにデータを公開したい場合は、ONにする。



3.2.7 サイドバーの設定

サイドバーを使用することで、データの追加や凡例の閲覧が可能。PLATEAU EditorではPLATEAU VIEW-のサイドバーとほぼ同様に使用することができる。そのためサイドバーの基本的な使用方法是3.2を参照。ここでは、PLATEAU Editorでのみアクセス可能な、管理者向けの機能について解説する。管理者向けに追加されている、PLATEAU Editorでのみアクセス可能な機能は以下の通り。

1. データのコンポーネントを編集するためのタブ
2. テンプレートを編集するためのタブ



PLATEAU Editorでは、データカタログから追加可能なそれぞれのデータセットに対し、UI操作によって凡例の表示や地図上のデータのスタイルを細かく設定することができる。それを可能にしているのが「コンポーネント機能」と「テンプレート機能」である。

そこで、ここではコンポーネント機能やテンプレート機能に共通する概念や使用方法について解説する。なお、データカタログからのデータの追加方法は「3.3.3 PLATEAU VIEW-の利用方法」を、PLATEAU VIEWにおける対応データフォーマットや、個別のコンポーネントに関するより詳細な説明は、「3.2.10 データの可視化とコンポーネント」を参照。

(1) コンポーネントとテンプレート

コンポーネントとは、データセットに対し「凡例の表示」「スタイルの変更」など、PLATEAU VIEWにおける追加の機能を付加する仕組みである。データカタログで追加可能なそれぞれのデータセットに対し、複数のコンポーネントを紐づけることができる。

それぞれのコンポーネントは設定項目を持ち、PLATEAU VIEWではその設定項目を編集することはできないが、PLATEAU Editorでは「データのコンポーネントを編集するためのタブ」にて編集することができる。

テンプレートとは、複数のデータセットに対し同じ設定のコンポーネントを一括で登録するための機能。データの種類に応じたテンプレートを作成することができ、データカタログからデータセットを追加すると、自動的にテンプレートで設定されたコンポーネントが反映される。

コンポーネントは複数の種類を同時に組み合わせることができ、どのような効果が発生するかはコンポーネントごとに異なる。例えばサイドバー上に凡例を表示するための「凡例コンポーネント」は、地図上のデータのスタイルには何も影響を与えない。それに対し「ポイント>色コンポーネント」を使うと、サイドバーには特にUIは表示されないが、データの地図上での見た目を変更することができる。こうした異なる機能を持つコンポーネントを組み合わせることで、データセットに対し様々な表示のカスタマイズが行える仕組みとなっている。

(2) コンポーネントの作成・編集・削除

コンポーネントを追加するには、予めデータセットがデータカタログから追加されている必要がある（3.3.3を参照）。サイドバーの追加されたデータセット上で、上記の図の1番の「設定」タブをクリックすることで、コンポーネント設定タブが開く。



コンポーネント設定タブでは以下の操作が可能。

- 各種フィールド（コンポーネント）の設定を編集
- フィールドの追加：「+ フィールドを追加」ボタンをクリック
- フィールドの削除：各フィールド右上のゴミ箱のアイコンをクリック
- フィールドのグループ管理：各フィールド右上の4つの小さな○が並ぶアイコンのボタンをクリックすると、グループの作成・削除・割り当てが可能

注意点として、最下部の「保存」ボタンをクリックしないと変更が保存されないため、クリックし忘れないように注意する。なお保存せずとも「公開」タブをクリックすることで、現在のコンポーネントのプレビューが可能。

【参考】デフォルトテンプレートがそのデータセットに対して適用されている場合、自動的に「テンプレートコンポーネント」が設定される。これを削除することでそのデータセットに対してテンプレートを無効化することが可能。テンプレートを適用しながら同時に個別にコンポーネントを追加して適用したい場合は、テンプレートコンポーネントを残したまま、他のコンポーネントを追加する。

【参考】グループとは、コンポーネントをグループにまとめることができる機能。「グループ切り替え」コンポーネントを配置することで現在表示中のグループの変更が可能になるが、グループを切り替えることで現在のグループ以外のコンポーネントは適用（表示）されなくなる。これにより例えば、グループを切り替えることで、同じデータに対し属性別にデータを表示するといった設定が可能。

(3) テンプレートの作成・編集・削除

テンプレートに関する設定を行うには、サイドバーで上記の図の2番のタブをクリックすることで、テンプレートパネルを表示することができる。



新しいテンプレートを追加するには、最上部の「+ テンプレートを追加」ボタンをクリックすることでテンプレート編集画面に遷移する。

テンプレートを編集するには、作成済みのテンプレート名が書かれたボタンをクリックすることでテンプレート編集画面に遷移する。

テンプレートを削除するには、作成済みのテンプレート名が書かれたボタンの右端にあるゴミ箱のアイコンをクリックする。

テンプレート編集画面では、上記コンポーネント編集タブと同様の操作が可能だが、更に以下の操作が追加で可能。

- テンプレート名の変更：上部の鉛筆アイコンをクリックすることで名前の変更が可能

注意点として、最下部の「保存」ボタンをクリックしないと変更が保存されないため、クリックし忘れないように注意する。

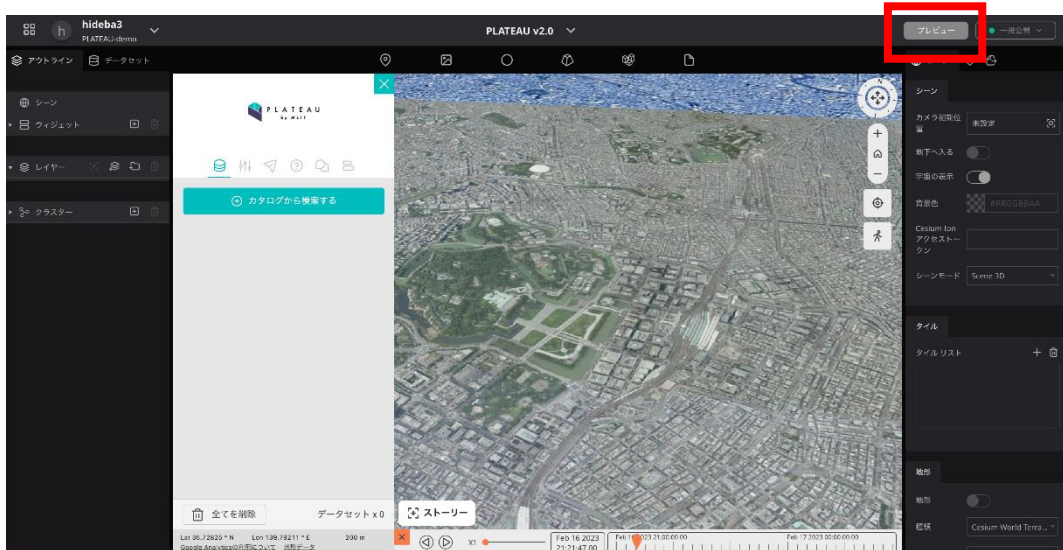
【参考】テンプレート名は、データカタログの種類の名前にすることで、種類が一致するデータセットに対し反映される。例えば「建築物モデル」「津波浸水想定区域モデル」「避難施設情報」のような名前にすることで、その種類のデータに対してテンプレートが適用される。ただし個別のデータセットで「テンプレートコンポーネント」を使用することで、任意のテンプレートを適用することもできる。



3.2.8 プロジェクトのプレビュー

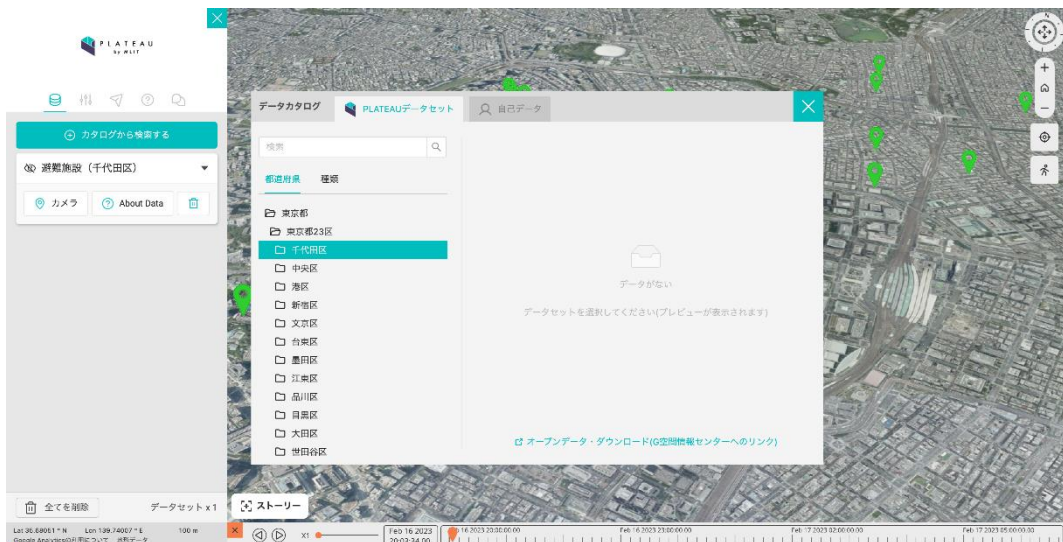
プレビューでは、現在編集中のデジタルアースが公開後どのように表示されるかを確認することができる。プレビューを確認するには画面右上の「プレビュー」をクリックする。プレビューではそれまで編集した内容が反映される。

なお、既に公開しているプロジェクトについては再度公開を実行しない限り、編集内容は反映されない。



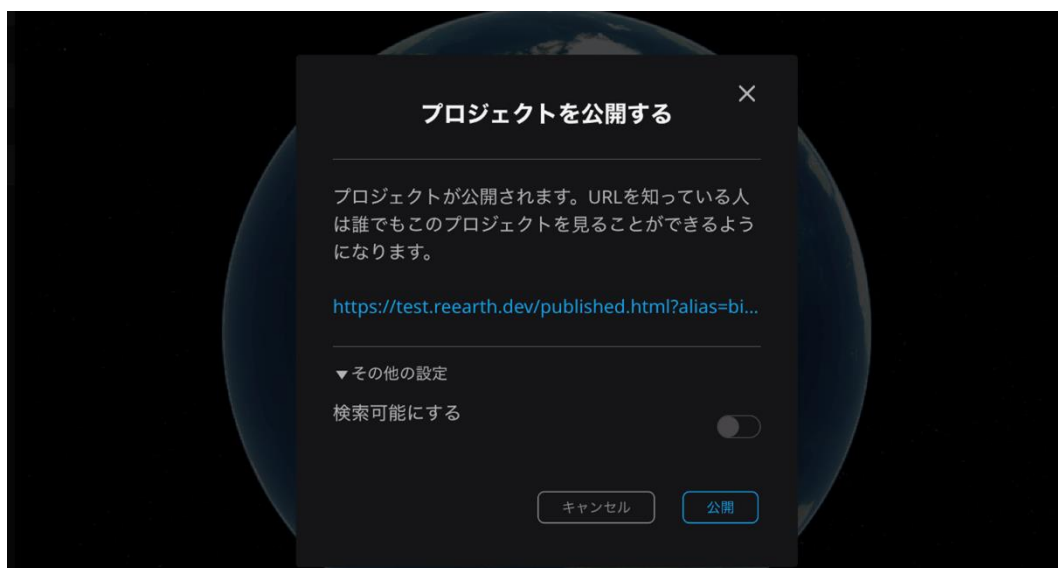
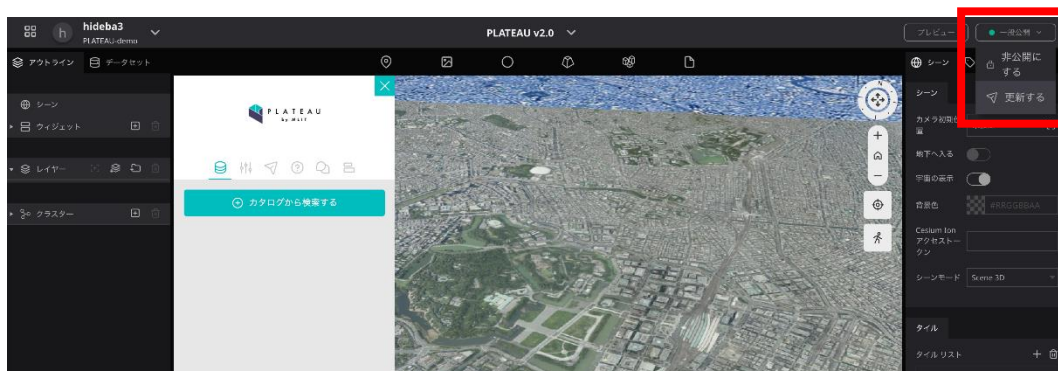
プレビューはブラウザの新規タブで表示される。編集中のPLATEAU VIEWが想定通りに動作するかをプレビューページで確認する。

なお、データカタログでデータセットが表示されるには、PLATEAU CMSで各種データが登録されており、公開申請を承認されてPLATEAU CMSから公開されている必要がある。

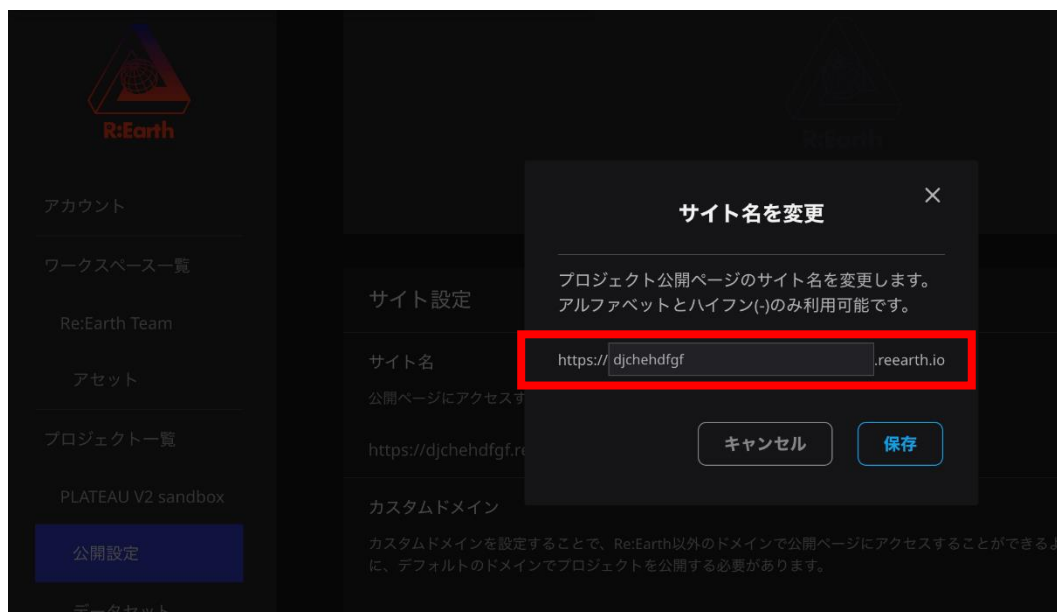


3.2.9 プロジェクトの公開

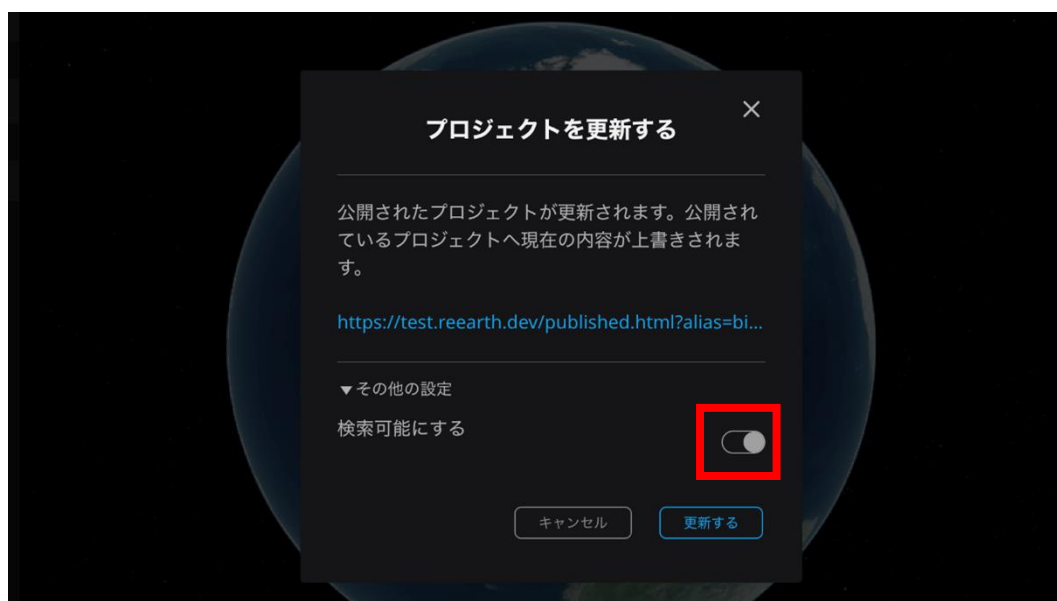
プロジェクト編集画面の右上の「公開」からプロジェクトを公開することができます。



- 公開URL：リンクを知っているインターネット上の全員が閲覧できるURLである。サブドメインとしてURLに使われる文字列は、任意に変更可能である。サブドメインとは、画像の例では <https://test.reearth.dev/...> の **test** の部分になる。他者にとってわかりやすいサブドメインに変更することを推奨する。
- サブドメインを変更するには、プロジェクト一覧 → 公開設定 → サイト設定 → 編集で変更できる。



- 埋め込み用コード：その他のwebサイトでプロジェクトを埋め込む場合は、埋め込み用コードを使用する。
- プロジェクトをGoogleの検索結果に表示させるためには、公開時の確認画面で、その他の設定から、検索可能にするのトグルボタンをオン（緑色の状態）にすることで、Googleの検索結果への表示が可能になる。なお、Google検索に実際に公開ページがインデックスされるには、他のWebサイトからの被リンクが必要であり、数日以上かかることがある。



3.2.10 データの可視化とコンポーネント

PLATEAU VIEWで表示可能なデータフォーマットとそれらの表現方法のパターン、およびコンポーネントを概説する。

(1) 主な対応データフォーマット

パフォーマンスの観点からWebで表示するのに適したフォーマットの使用を推奨する。ShapefileはWeb用に最適化されたフォーマットではないため、表示が遅いなどの理由で非推奨。

表 主な対応データフォーマット

フォーマット	説明	MyDataでローカルファイルから追加	MyDataでURLから追加	表示レイヤーの指定	対応する地物
GeoJSON	JSONで記述される、2Dの空間データを扱うためのフォーマット。ドキュメント： https://geojson.org/	☑	☑	不要	ポイント・ポリライン・ポリゴン
KML	空間データをXMLで記述するフォーマット。ドキュメント： https://developers.google.com/kml/documentation/?hl=ja	☑	☑	不要	ポイント・ポリライン・ポリゴン
CZML	JSONで記述される、CesiumJSでの空間データを表現するのに適したフォーマット。時系列データも表現可能。ドキュメント： https://github.com/AnalyticalGraphicsInc/czmlwriter/wiki/CZML-Structure	☑	☑	不要	ポイント・ポリライン・ポリゴン
CSV	テキストファイルのフォーマットの1つ。PLATEAU VIEWでは経度・緯度・高さの情報を持ったポイントデータとして取り込むことができる。読み込むためには少なくとも経度・緯度を表す列が必要。	☑	☑	不要	ポイント・ポリライン・ポリゴン
3D Tiles	Cesiumによって開発された、Web上で3D地理空間コンテンツを効率的にストリーミングするためのデータ形式。ドキュメント： https://github.com/CesiumGS/3d-tiles	✕	☑	不要	3D Tiles
MVT	Mapbox Vector Tileの略称。Mapbox社が開発。Web上で高速かつ効率的なベクタータイルの表示を実現。ドキュメント： https://docs.mapbox.com/data/tilesets/guides/vector-tiles-standards/	✕	☑	必須	ポリゴン
glTF	3Dモデルのデータフォーマットの1つ。3DモデルをWeb上で簡単に共有・表示するための標準形式として、Khronos Groupによって策定。ドキュメント： https://github.com/KhronosGroup/glTF	☑	☑	不要	ポイント※
WMS	Webマップサービスの1つであり、OGCによって定義された標準規格。地理情報をリクエストし、画像として返す。ドキュメント： https://www.ogc.org/standard/wms/	✕	☑	必須	ラスター

※表示には別途ポイントデータが必要なためCSVなど他のデータと組み合わせることを推奨

(2) コンポーネントの種類

PLATEAU VIEWでは、コンポーネントを組み合わせることで凡例などの表示やデータのスタイリングを行うことができる。コンポーネントの概念に関する説明や操作方法是「3.2.7 サイドバーの設定」を参照。

ここでは主なコンポーネントを解説する。画像が上下に並んでいる場合、上側が実際の見た目、下側が編集画面での表示である。

※開発中の画面であり実際の表示と異なる場合がある。

1. 一般>カメラ

「カメラ」ボタンで移動するカメラの位置を設定することができる。このコンポーネントがない場合は自動的にカメラ位置が計算される（MVTやWMSなどカメラ位置の自動計算が行われないデータフォーマットもある）。地図上のデータの表示には影響を与えない。

2. 一般>説明

説明を表示する。マークダウンも対応しており、URL指定で画像の表示も可能。地図上のデータの表示には影響を与えない。

3. 一般>凡例

凡例を表示する。スタイルは四角・丸・線・アイコンから選択でき、アイコンの場合は画像URLを指定することができる。地図上のデータの表示には影響を与えない。

4. 一般>リアルタイム

指定した時間が経過する度に、データをリフレッシュ（同じURLから取得し直す）する。同じURLでリアルタイムに内容が変化するようなデータ（例：バスロケなどのGTFSデータ）に向く。

5. 一般>タイムライン

CSVやGeoJSONなどで時刻の属性が埋め込まれている場合に使用可能。フィールド名を指定するとその時刻を使用して時系列データを表示する。時系列データは、PLATEAU VIEW画面下部のタイムラインを操作することで、自動的に時刻に連動した表示が行われる。

なおCZMLは、時系列データを含む場合、このコンポーネントがなくても自動的にタイムラインに連動してデータが表示されるため、このコンポーネントは不要。

6. 一般>ストーリー

ストーリー（カメラ移動と文章表示を行う一連の流れ）を登録することで、ユーザーは登録されたストーリーを再生が可能。ストーリーウィジェットと連動して編集可能。複数のストーリーを登録可能。地図上のデータの表示には影響を与えない。

「編集」ボタンをクリックすると保存されているストーリーがストーリーウィジェットに展開され、「編集を終了する」ボタンをクリックすることで現在のストーリーウィジェットの内容が保存される。

7. 一般>リンクボタン

クリックすると指定URLのページが開くボタンを配置するコンポーネント。地図上のデータの表示には影響を与えない。

8. 一般>スイッチグループ（グループの切り替え）

現在のグループを切り替えることが可能。グループを切り替えると、そのグループに属しているコンポーネントのみが適用（表示）される。例えば、同じデータでも属性別で塗り分けを選べるようにしたい場合に使用（グループの数だけ色コンポーネントを追加してそれぞれグループに割り当てる）。

9. 一般>スイッチデータセット（データセットの切り替え）


1つのデータセットの中に更に複数のデータセットを含む場合のみ有効。グループの切り替えとは異なり、データセットを切り替えてもコンポーネントは変化しない。

10. ポイント・ポリゴン・ポリライン・3D Tiles>色・ストローク・可視

地図上のデータの色（マーカーの色・ポリゴンとポリラインの塗り色）、ストロークの色や太さ、可視（表示非表示）を上書き可能。式を書くことで属性に応じた色の塗り分けを行うことが可能。式が複数ある場合は、式が上から順に評価され、マッチした式のうち最も下の式の設定が使用される。

11. ポイント> サイズ

ポイントのサイズを変更可能。

▼ サイズ 

サイズ

12. ポイント> アイコン

ポイントのアイコンを設定可能。サイズは倍率。アイコンURLはCORSの設定に注意。

▼ アイコン 

アイコンURL

サイズ

13. ポイント> ラベル

ポイントのアイコンを設定可能。サイズは倍率。アイコンURLはCORSの設定に注意。

▼ ラベル 

Text

Font size px

Font color  #FFFFFF

Extruded ☐

Use Background ☐

14. ポイント> モデル

ポイントのモデルを設定可能。例：バスロケのバス走行位置にバスの3Dモデルを配置

▼ モデル 

モデルURL

目盛

15. ポイント> ポイントに変換 (CSV)

CSVデータのみ、ポイントデータに変換することができる。緯度と経度のカラム名指定が最低限必要。

▼ ポイントに変換 (CSV) 

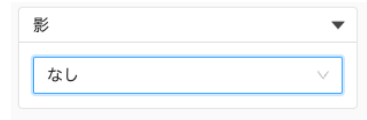
経度フィールド名

緯度フィールド名

高さフィールド名

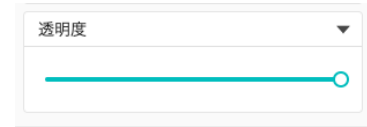
16. 3D Tiles> 影

3D Tilesの影の設定を変更可能にする。



17. 3D Tiles> 透明度

3D Tilesの透明度を変更可能にする。



18. 3D Tiles> クリッピング

3D Tilesのクリッピング機能（マウスで操作可能な箱で建築物などをくり抜いて表示する）を有効化する。



(3) 補遺

- 3D データの高さ方向の座標値について
PLATEAU VIEWで表示する3Dデータを作成する場合、z座標は楕円体高（ジオイド高 + 標高）とする必要がある。
- PLATEAU VIEW 1.1からの変更点
1.1では内部システムにTerriaJSが採用されており、データカタログを記述したJSONを作成することで、凡例などのカスタマイズを可能にしていた。
2.0では内部システムがRe:Earthに置き換えられ、UI操作のみでコンポーネントを組み合わせることで凡例やスタイルのカスタマイズがノーコードで可能になった。
これに伴い、1.1で使用されていたツール「plateau-catalog-generator」は使用する必要がない。また、それを使用して生成したJSONは2.0では使用できないことに注意。

3.3 PLATEAU VIEWの利用方法

ここではでは、PLATEAU VIEWの概要及びその利用方法について解説する。

3.3.1 主な機能一覧



表 PLATEAU VIEWの主なUI機能一覧

UI機能	概要
データカタログ	データの一覧表示、閲覧データの詳細説明表示、閲覧データの選択等
My Data	ローカルおよびWeb上のデータの表示
サイドバー	データカタログで選択したデータの凡例や表示設定
地図設定	背景地図の選択
共有・プリント	URLを発行し操作中のPLATEAU VIEWの共有、印刷
ストーリー	ストーリーの作成、編集及び閲覧
ナビゲーター	PLATEAU VIEWの回転、傾き及び初期位置の表示
ズーム	ズームレベルの制御
歩行者モード	歩行者モードを有効にし、一人称視点でのPLATEAU VIEW閲覧
ご意見ご要望	コメントまたは質問の送信
使い方説明	PLATEAU VIEWの使い方が緯度
タイムライン	時系列データの制御

3.3.2 PLATEAU VIEW 1.1からの変更点

表 主な変更点对照表

PLATEAU VIEW 1.1	PLATEAU VIEW 2.0
共有機能を利用してPLATEAU VIEWを第三者に共有する際、設定情報が全てURL内にエンコードされていたため、非常に長いURLになっていた。	共有機能を利用してPLATEAU VIEWを第三者に共有する際、設定情報をPLATEAU VIEWサーバーへ保存することで、より短いURLで共有が可能になった。
「地図設定」「共有」「ストーリー」「凡例表示」などが画面上部から左部に点在しており、ユーザーにとって利用しにくいUIだった。	「地図設定」「共有」「ストーリー」「凡例表示」などを全てサイドバーに集約することで地図画面をより大きく見せ、すっきりとしたUIへ改善した。
データカタログ内のフォルダ構成が複雑で、対象データを見つけることが難しかった。	データカタログ内のフォルダ構成を見直すとともに、都道府県もしくはデータの種類によってフォルダ構成を変更できるようにした。データ件数が多くなっても目的のデータを容易に見つけられるようになった。

3.3.3 PLATEAU VIEWの利用方法

基本的な利用方法として、各UIパースの概要や、データの選択・表示・重畳に関連する一連の操作について説明する。

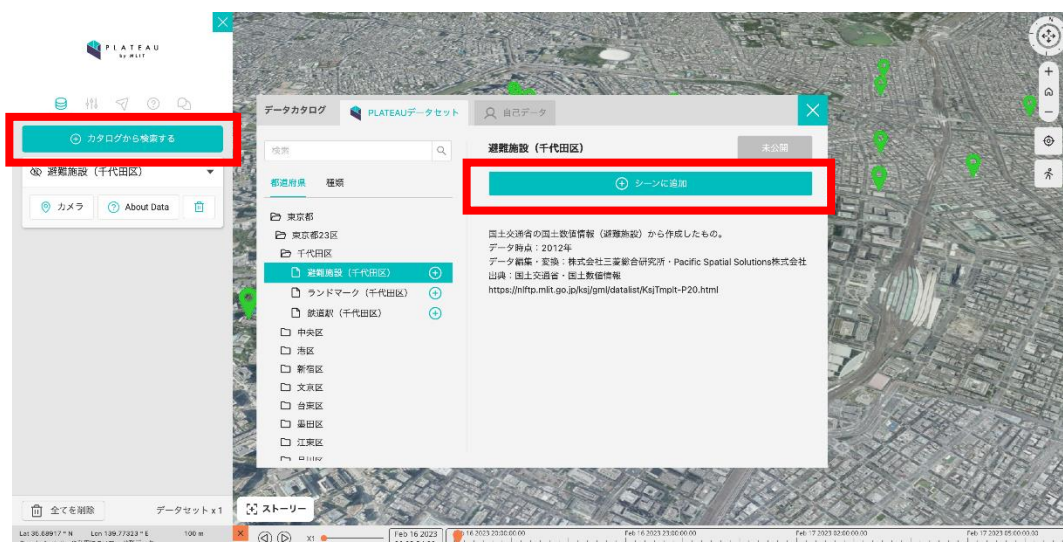
(1) 地図設定

サイドバーのタブを切り替えることで地図設定を以下の中から設定が変更できる。

- ・ マップビュー
 - ・ 地図を3D地形で表示
 - ・ 地図を3D地形なしで表示
 - ・ 地図を2Dで表示
- ・ 地下を隠す
- ・ 地下に入る
- ・ ベースマップ
 - ・ 空中写真
 - ・ 空中写真 (Bing)
 - ・ 地理院地図
 - ・ Dark Matter

(2) データの閲覧

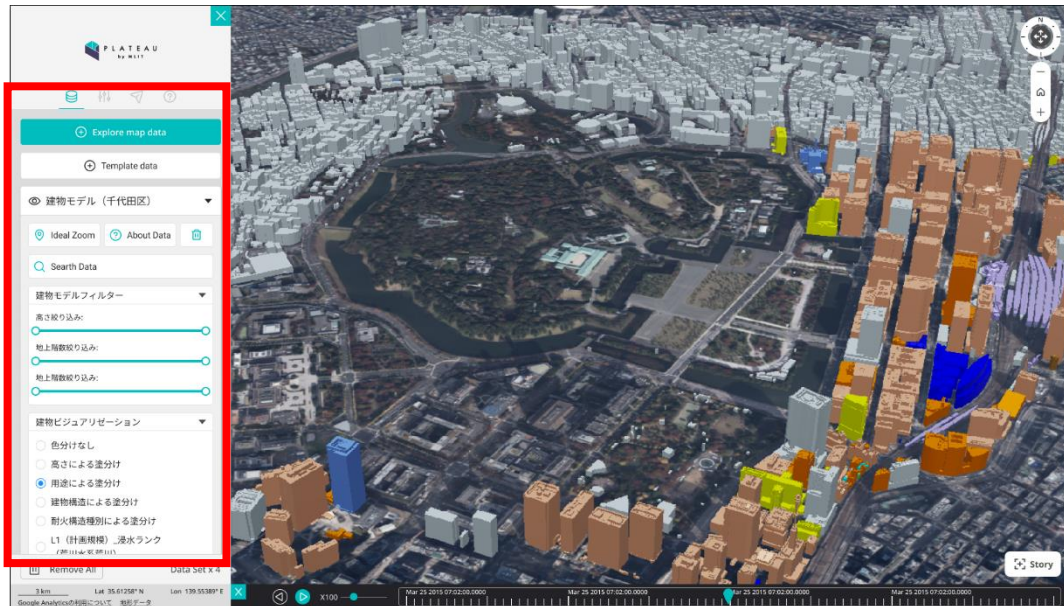
サイドバーの「カタログから検索する」を選択しデータカタログを表示する。データカタログから閲覧したいデータを選択し、「シーンに追加」をクリックする。



なお、「種類」のタブを選択することでデータの種類別にフォルダ分けしてデータを選択することができる。

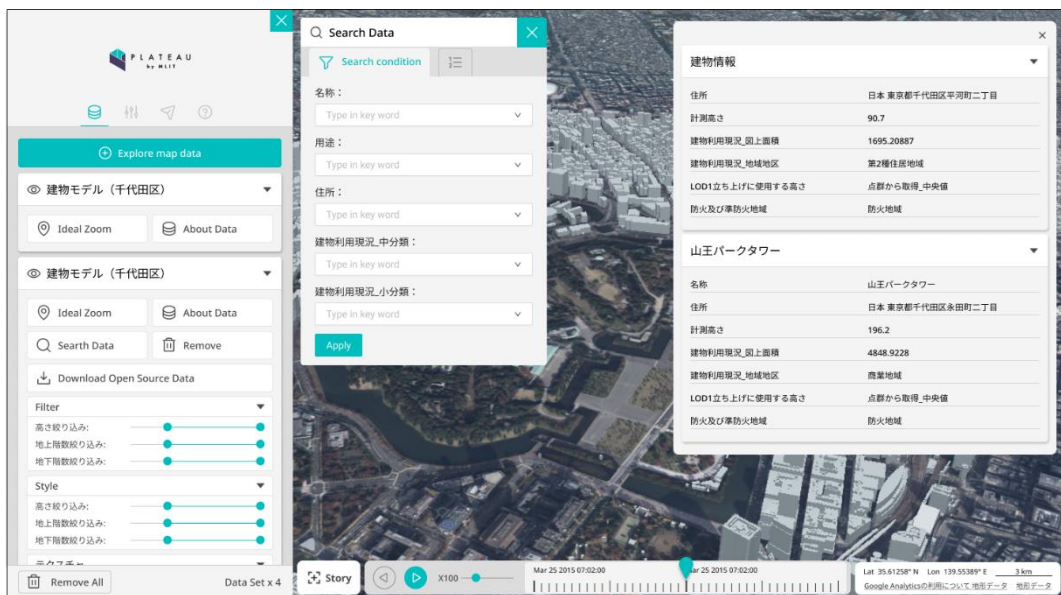


サイドバーから凡例の確認や、データのフィルター、色分け等を行うことができる。「用途による塗り分け」を選択する。



地物をクリックすることで、選択された地物の属性情報をインフォボックスで閲覧することができる。

また、属性検索機能を利用することで、名称や用途などで地物を検索することができる。



(3) PLATEAU VIEWの共有

地図上へのデータの追加、ストーリーの作成、地図設定などの変更をした後、共有機能を利用することで第三者へ閲覧中のPLATEAU VIEWを共有することができる。

サイドバーから「共有」ボタンをクリックすることで、現在閲覧中のPLATEAU VIEW情報が保存され、共有用URLが発行される。

また、閲覧中のPLATEAU VIEWのスクリーンショットをダウンロードや印刷することができる。



(4) その他閲覧上の操作方法

その他、マウスの操作方法は以下の通りである。また、ナビゲーターでは地図の角度等を調節が可能である。

図 マウス操作方法

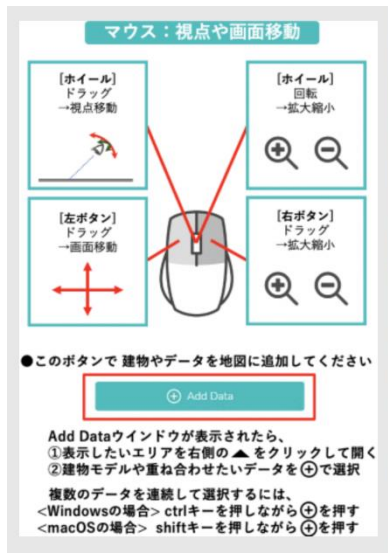
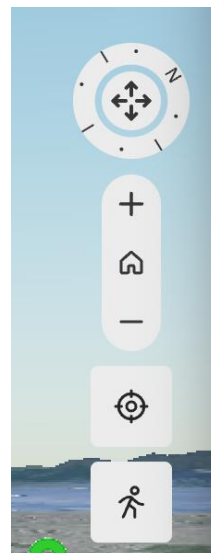


図 ナビゲーター



3.4 SDKの利用方法

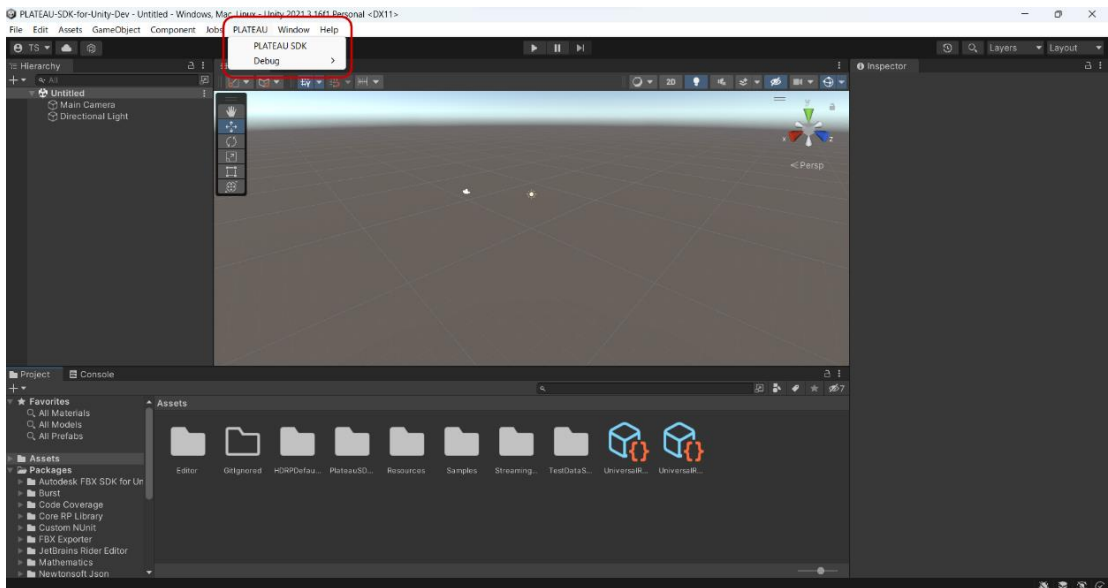
3.4.1 PLATEAU SDK for Unityの利用方法

PLATEAU SDK for Unityの基本的な利用方法について解説する。PLATEAU SDK for Unityのより詳細なマニュアルは以下のURLから参照されたい。

<https://prproject-plateau.github.io/PLATEAU-SDK-for-Unity/index.html>

PLATEAU SDK for Unity画面の表示・各機能へのアクセス

Unityのメニューバーから PLATEAU → PLATEAU SDK を選択してPLATEAU SDK画面を開く。PLATEAU SDK画面の上部タブから選択することで各機能にアクセスする。



3D都市モデルのインポート

PLATEAU SDK画面の上部タブから［インポート］を押下すると、都市モデルインポート画面が表示される。

(1)インポート元の選択

インポートはローカルとサーバーの2種類から選択できる。ローカルからインポートする場合、まずは3D都市モデルデータを用意する。PLATEAUの3D都市モデルデータはポータルサイトで公開されている。ポータルサイトから地域を選び、CityGMLデータをダウンロードし、展開する。

<https://www.geospatial.jp/ckan/dataset/plateau>



データ



データ目録

東京都23区の提供データ目録です。

詳細



図郭マップ

利用するデータの地図範囲を確認いただけます。

詳細



CityGML

CityGML形式のデータで、次のデータが格納されています。建築物 橋梁 道路 土地利用・公園 地形 都市設備 洪水浸水想定区域（国管理、都管理）...

詳細



3D Tiles, GeoJson, MVT, Shape

3D Tiles形式は、次の項目が格納されています。建物モデル 橋梁モデル 洪水浸水想定区域（国管理河川、県管理河川）...

より多くの情報
ダウンロード

詳細

フォルダパスには、展開したCityGMLデータの最上位フォルダ（「udx」などの1つ上の階層のフォルダ）を設定する。

ローカルのデータではなくサーバーから直接シーンに3D都市モデルをインポートしたい場合、インポート元として【サーバー】を選択する。新規に構築されたサーバーに接続したい場合、接続先設定を開き、サーバーURLと認証トークンを入力する。

接続先サーバーにはhttps://api.<ドメイン>を入力する。<ドメイン>は本マニュアルの2.1.2（4）で設定された文字列である。

認証トークンには、本マニュアルの2.1.5（4）で

```
terraform apply -var-file=env/example.tfvars
```

を実行した際の出力の内、

```
plateauview_sdk_token = "*****"
```

の""内の文字列を入力する。

正常に接続できた場合サーバー上に存在するデータセットの一覧が表示されるので、取得したいデータセットを選択する。

(2) 基準座標系の選択

リストのうち、データが存在する場所として近いものを選択する。基準座標系が離れすぎている場合、インポートした3D都市モデルに歪みが生じてしまうため注意する。



(3)インポート範囲の選択

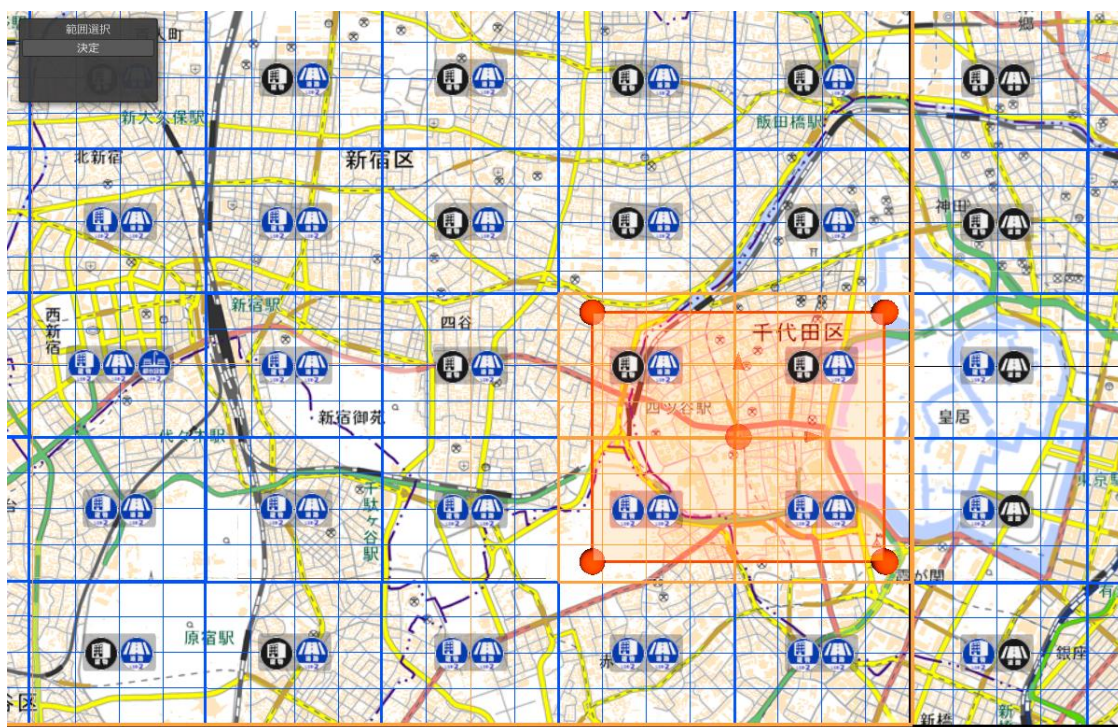
範囲選択ボタンを押す。現在のUnityシーンに変更がある場合、変更を保存するかどうかを尋ねるダイアログが表示されるので [Save] または [Don't Save] を押下する。シーンビューの表示が切り替わり、範囲選択画面になる。

範囲選択画面の操作方法：

- ・ マウスホイールを上下に回してズームアウト、ズームイン
- ・ マウスホイールを押し込んだままドラッグしてカメラ移動
- ・ オレンジ色の球体をドラッグして範囲を選択
- ・ シーンビュー左上の [決定] ボタンを押して範囲を確定

画面の見方：

- ・ 青色の線は利用可能な地域
- ・ 地域ごとに利用可能なGML種別と対応LODがアイコン形式で表示
- ・ 地図は国土地理院のサーバーから自動でダウンロードして表示されるため、インターネットへの接続が必要



(4)地物別設定

地物の種類ごとにインポートに関する設定を行う。

設定項目について：

- ・ インポートする
 - ・ チェックが付いている地物タイプのみインポートする。
- ・ Mesh Collider をセットする
 - ・ チェックが付いている場合、各モデルに Mesh Collider が追加される。
- ・ テクスチャを含める
 - ・ チェックが付いていて、かつ地物にテクスチャがある場合はそれを含めてインポートする。
- ・ LOD描画設定
 - ・ 複数のLODを利用可能な地物タイプで表示される。
 - ・ バーの左端と右端をドラッグして、インポートするLODの範囲を指定する。
- ・ モデル結合
 - ・ 主要地物単位（建築物,道路等）
 - ・ モデルのメッシュは建物ごとに結合されて出力される。建物ごとに地物データを取得したい場合はこれを選択する。
 - ・ 最小地物単位（壁面,屋根面等）
 - ・ 屋根、壁単位など非常に細かくオブジェクトを分けたい場合はこれを選択する。
 - ・ 地域単位
 - ・ モデルのメッシュは地域ごとに結合されて出力される。オブジェクト数を削減して軽量化できるが、建物ごとの地物データは取得不可になる。

(5)インポートの実行

〔モデルをインポート〕 ボタンを押すとインポートが開始される。

ウィンドウを下にスクロールすると、インポート処理の進捗が表示される。処理が進むと、都市のオブジェクトが順次シーンに配置される。進捗表示がすべて「完了」になったらインポート処理が終了している。



3D都市モデルのエクスポート

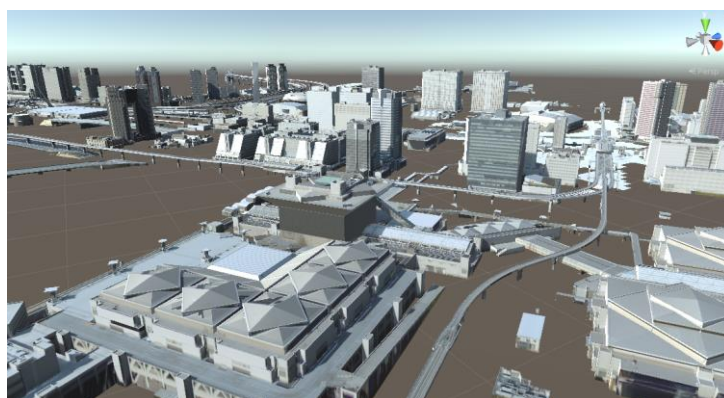
PLATEAU SDK for Unity画面の上部タブから「エクスポート」を押下すると、都市モデルエクスポート画面が表示される。



次の設定が可能である。

- 出力形式のOBJ、FBX、glTFからの選択
- テクスチャを含めるかどうか
- 非アクティブオブジェクトを含めるかどうか
- 座標変換

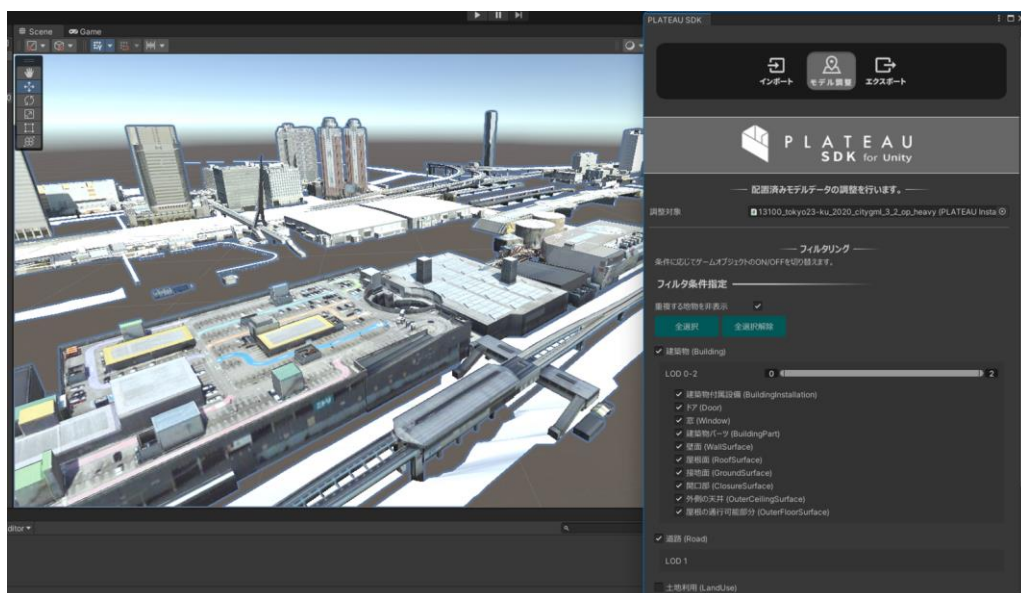
「エクスポート」ボタンを押すと、指定のフォルダに3Dモデルファイルが出力される。



3D都市モデルの調整

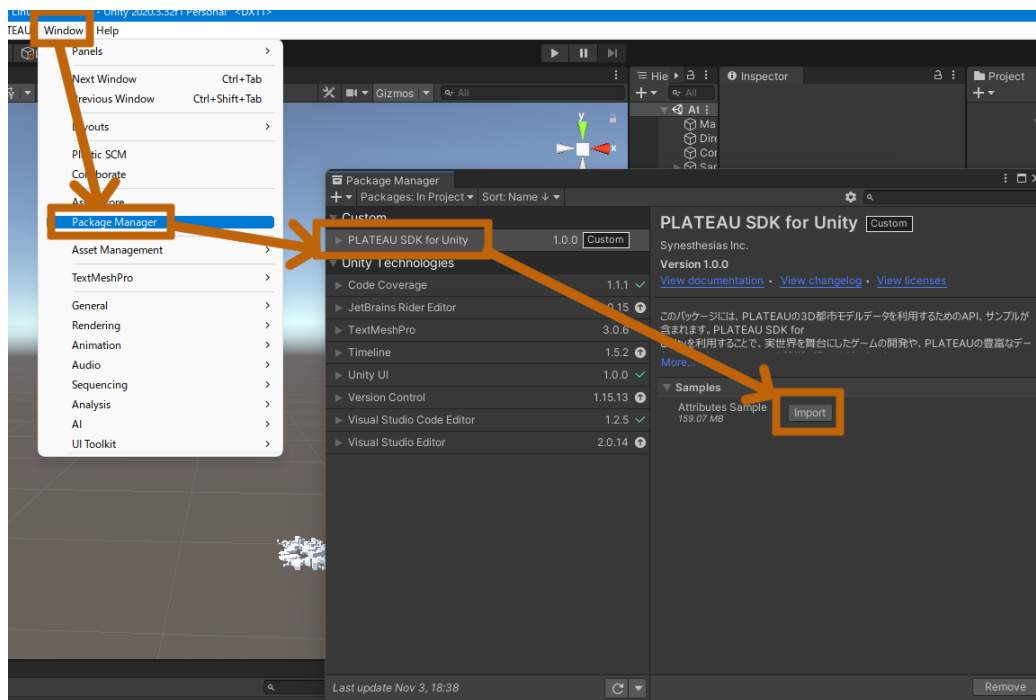
PLATEAU SDK for Unity画面の上部タブから「モデル調整」を押下すると、都市モデル調整画面が表示される。都市モデル調整画面では、インポートされた3D都市モデルについて以下の条件を指定して表示・非表示を切り替える。

- ・ 地物タイプ（建築物、道路等）
- ・ 重複する地物を表示するか
- ・ LODの範囲

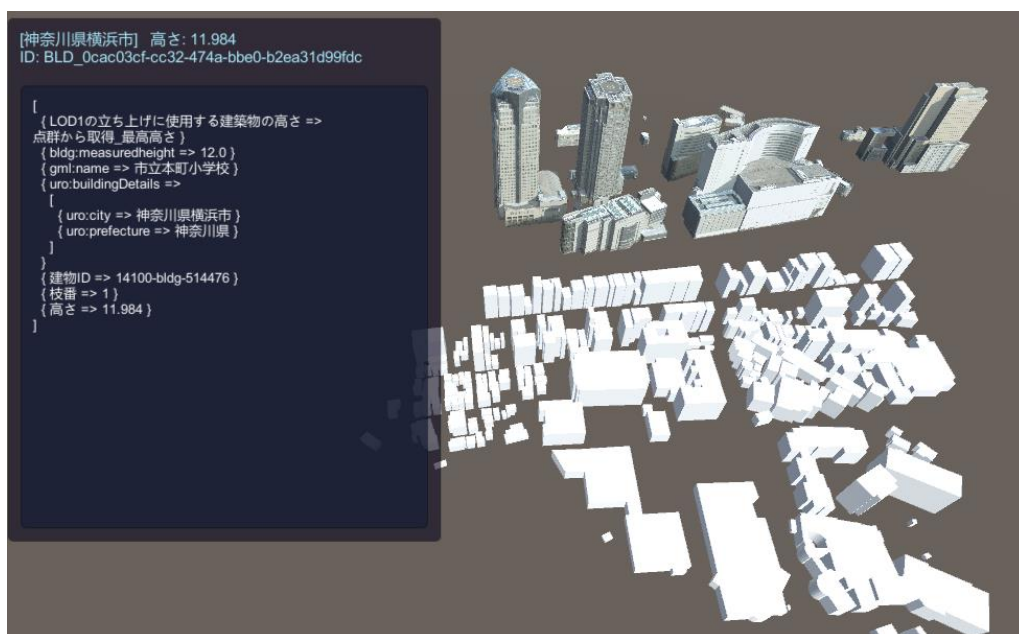


属性情報へのアクセス

PLATEAU SDK for UnityではCityGMLの属性情報を取得して活用するためのC#向けAPIを提供している。APIの利用方法については、[SDKのユーザーマニュアル](#)に詳細を記載している。また、APIの実際の利用方法を知るためのサンプルを提供している。UnityのPackage Manager からサンプルシーンをインポートして、シーン「AttributesSample」を開くことでサンプルが確認できる。



サンプルではクリックした建物の属性情報を表示できる。



3.4 PLATEAU SDK for Unrealの利用方法

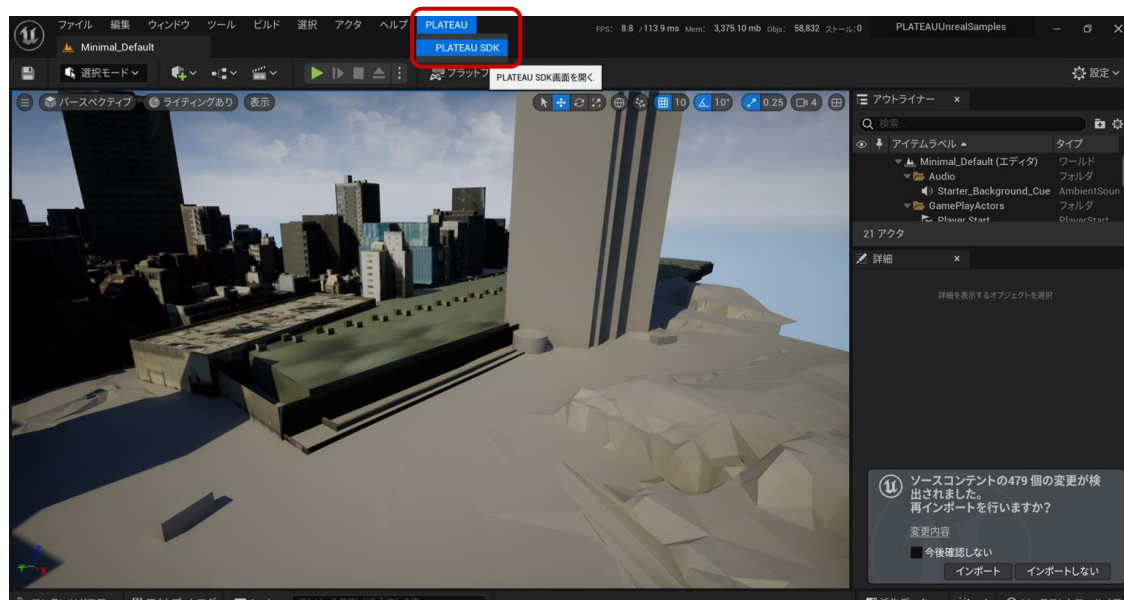
3.4.1 PLATEAU SDK for Unrealの利用方法

PLATEAU SDK for Unrealの基本的な利用方法について解説する。PLATEAU SDK for Unrealのより詳細なマニュアルは以下のURLから参照されたい。

<https://prpjct-plateau.github.io/PLATEAU-SDK-for-Unreal/index.html>

SDK画面の表示・各機能へのアクセス

Unreal Engineのメニューバーから PLATEAU → PLATEAU SDK を選択してSDK画面を開く。SDK画面の上部タブから選択することで各機能にアクセスする。



都市モデルのインポート

PLATEAU SDK for Unreal画面の上部タブから「インポート」を押下すると、都市モデルインポート画面が表示される。

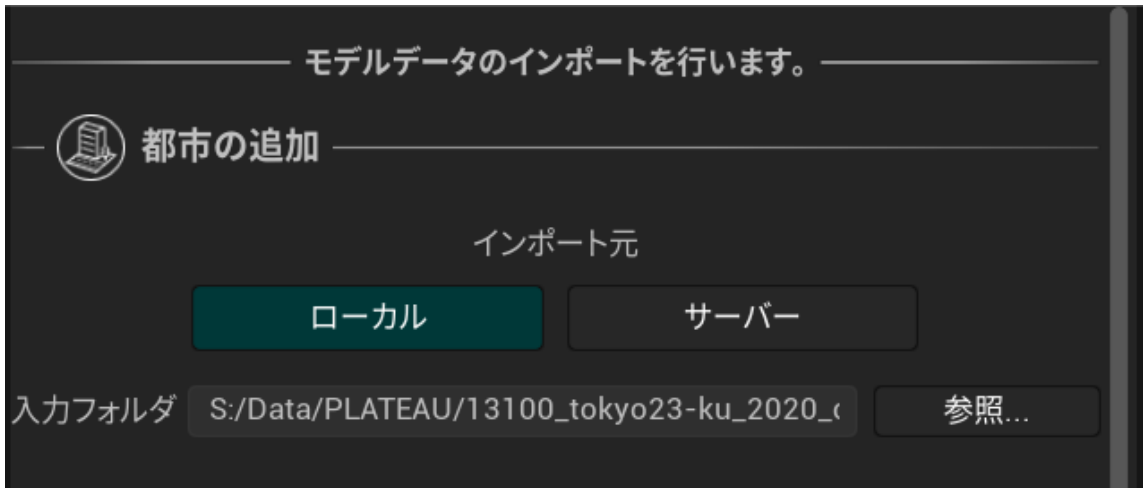
(1)インポート元の選択

インポートはローカルとサーバーの2種類から選択できる。ローカルからインポートする場合、まずは3D都市モデルデータを用意する。PLATEAUの3D都市モデルデータはポータルサイトで公開されている。ポータルサイトから地域を選び、CityGMLデータをダウンロードし、展開する。

<https://www.geospatial.jp/ckan/dataset/plateau>



フォルダパスには、展開したCityGMLデータの最上位フォルダ（「udx」などの1つ上の階層のフォルダ）を設定する。



ローカルのデータではなくサーバーから直接シーンに3D都市モデルをインポートしたい場合、インポート元として「サーバー」を選択する。新規に構築されたサーバーに接続したい場合、接続先上書設定を開き、サーバーURLと認証トークンを入力する。

接続先サーバーにはhttps://api.<ドメイン>を入力する。<ドメイン>は本マニュアルの2.1.2（4）で設定された文字列である。

認証トークンには、本マニュアルの2.1.5（4）で

```
terraform apply -var-file=env/example.tfvars
```

を実行した際の出力の内、

```
plateauview_sdk_token = "*****"
```

の""内の文字列を入力する。

正常に接続できた場合サーバー上に存在するデータセットの一覧が表示されるので、取得したいデータセットを選択する。



(2)基準座標系の選択

リストのうち、データが存在する場所として近いものを選択する。基準座標系が離れすぎている場合、インポートした3D都市モデルに歪みが生じてしまうため注意する。

モデルデータの配置を行います。

1 基準座標系の選択

基準座標系

09: 東京(本州), 福島, 栃木, 茨城, 埼玉, 千葉, 群馬, 神奈川

2 マップ範囲選択

範囲

3 地物別設定

建築物	
インポートする	<input checked="" type="checkbox"/>
テクスチャをインポートする	<input checked="" type="checkbox"/>
MeshColliderをセットする	<input checked="" type="checkbox"/>
Min Lod	2
Max Lod	3
モデル結合	主要地物
道路	
インポートする	<input type="checkbox"/>
テクスチャをインポートする	<input checked="" type="checkbox"/>

01: 長崎, 鹿児島(南西部)

02: 福岡, 佐賀, 熊本, 大分, 宮崎, 鹿児島(北東部)

03: 山口, 島根, 広島

04: 香川, 愛媛, 徳島, 高知

05: 兵庫, 鳥取, 岡山

06: 京都, 大阪, 福井, 滋賀, 三重, 奈良, 和歌山

07: 石川, 富山, 岐阜, 愛知

08: 新潟, 長野, 山梨, 静岡

09: 東京(本州), 福島, 栃木, 茨城, 埼玉, 千葉, 群馬, 神奈川

10: 青森, 秋田, 山形, 岩手, 宮城

11: 北海道(西部)

12: 北海道(中央部)

13: 北海道(東部)

14: 諸島(東京南部)

15: 沖縄

16: 諸島(沖縄西部)

17: 諸島(沖縄東部)

18: 小笠原諸島

19: 南鳥島

(3)インポート範囲の選択

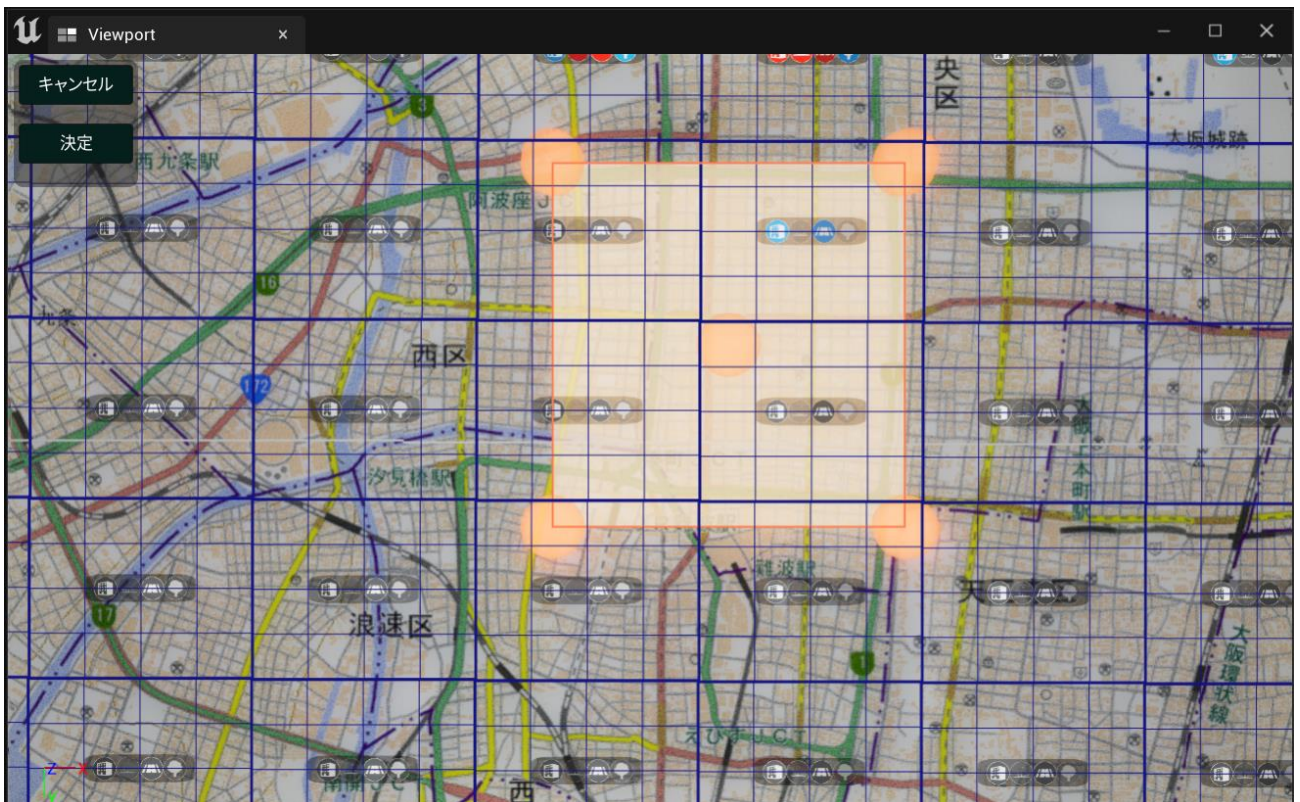
範囲選択ボタンを押すと範囲選択画面が開く。

範囲選択画面の操作方法：

- ・ 左クリックしながらドラッグしてカメラ移動
- ・ オレンジ色の球体をドラッグして範囲を選択
- ・ シーンビュー左上の〔決定〕ボタンを押して範囲を確定

画面の見方：

- ・ 青色の線は利用可能な地域
- ・ 地域ごとに利用可能なGML種別と対応LODがアイコン形式で表示
- ・ 地図は国土地理院のサーバーから自動でダウンロードして表示されるため、インターネットへの接続が必要



(4)地物別設定

地物の種類ごとにインポートに関する設定を行う。

設定項目について:

- インポートする
 - チェックが付いている地物タイプのみインポートする。
- テクスチャを含める
 - チェックが付いていて、かつ地物にテクスチャがある場合はそれを含めてインポートする。
- LOD描画設定
 - 複数のLODを利用可能な地物タイプで表示される。
 - バーの左端と右端をドラッグして、インポートするLODの範囲を指定する。
- モデル結合
 - 主要地物単位（建築物,道路等）
 - モデルのメッシュは建物ごとに結合されて出力される。建物ごとに地物データを取得したい場合はこれを選択する。
 - 最小地物単位（壁面,屋根面等）
 - 屋根、壁単位など非常に細かくオブジェクトを分けたい場合はこれを選択する。
 - 地域単位
 - モデルのメッシュは地域ごとに結合されて出力される。オブジェクト数を削減して軽量化できるが、建物ごとの地物データは取得不可になる。

(5)インポートの実行

「モデルをインポート」ボタンを押すとインポートが開始される。

PLATEAUCityModelLoaderアクターの詳細画面を開くと、インポート処理の進捗が表示される。処理が進むと、都市のオブジェクトが順次レベルに配置される。進捗表示が「完了」になったらインポート処理が終了している。



都市モデルのエクスポート

SDK画面の上部タブから「エクスポート」を押下すると、都市モデルエクスポート画面が表示される。

インポート モデル調整 エクスポート

PLATEAU
SDK for Unreal

モデルデータのエクスポートを行います。

— 選択オブジェクト —

アウトライナーからアクターを選択してください

— 出力形式 —

出力形式 OBJ ▼

▼ オプション

テクスチャを出力する	<input checked="" type="checkbox"/>
座標設定	ローカル座標 ▼
非表示モデルを出力する	<input checked="" type="checkbox"/>

出力先フォルダを選択

参照...

..../Users/kazsa/Synesthesias/PLATEAU-

モデルをエクスポート

次の設定が可能である。

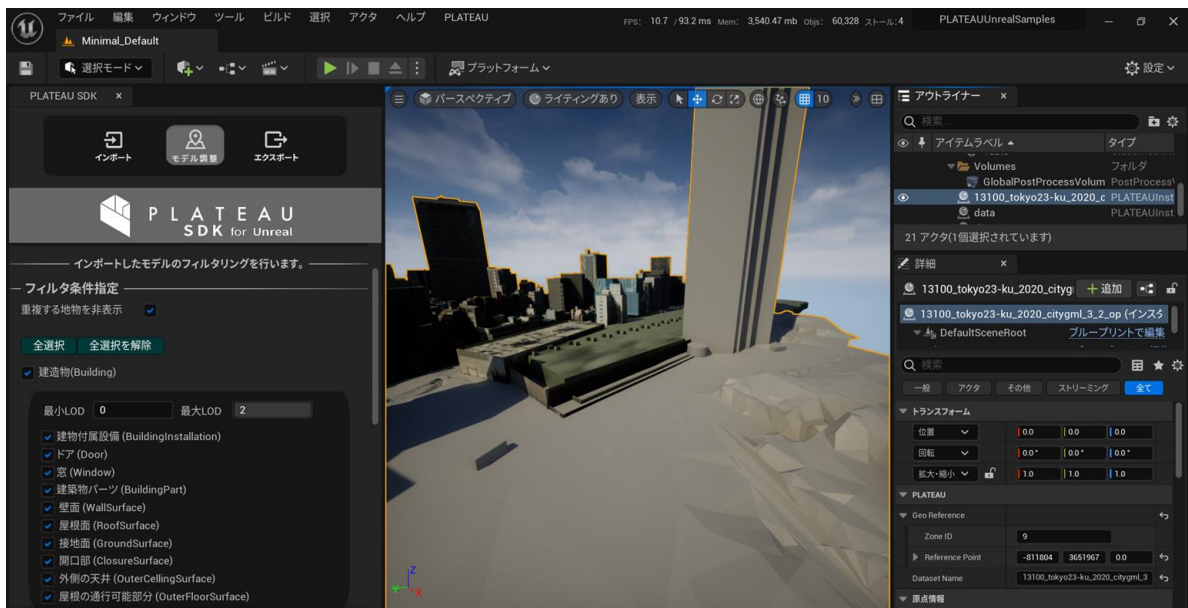
- 出力形式のOBJ、FBX、glTFからの選択
- テクスチャを含めるかどうか
- 非アクティブオブジェクトを含めるかどうか
- 座標変換

「エクスポート」ボタンを押すと、指定のフォルダに3Dモデルファイルが出力される。

都市モデルの調整

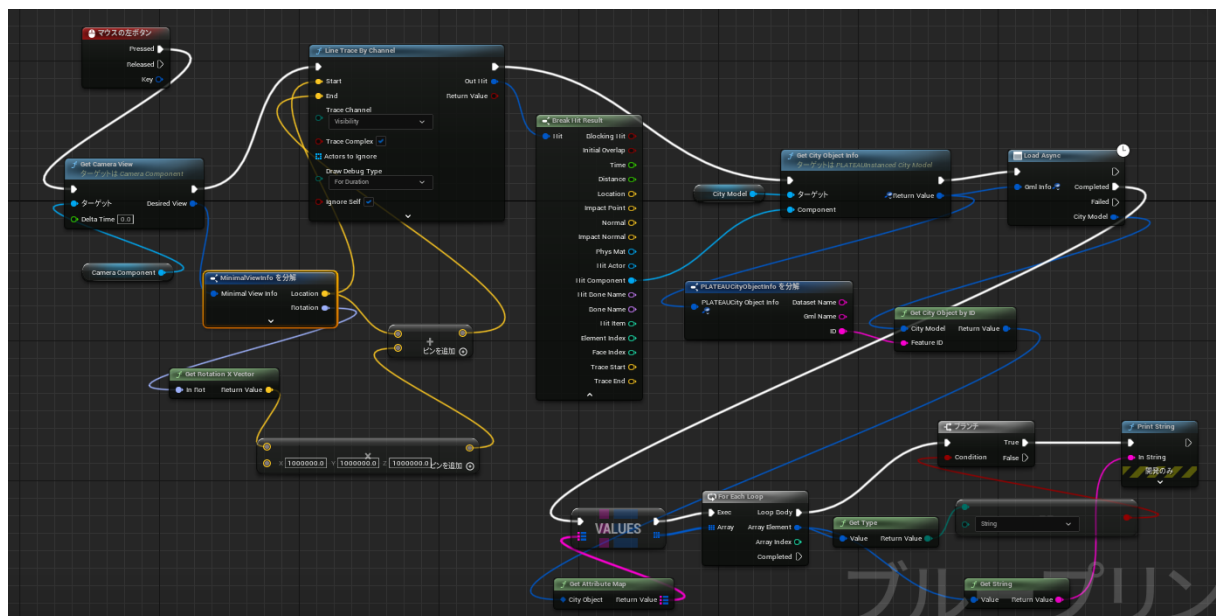
PLATEAU SDK for Unreal画面の上部タブから「モデル調整」を押下すると、都市モデル調整画面が表示される。都市モデル調整画面では、インポートされた3D都市モデルについて以下の条件を指定して表示・非表示を切り替える。

- ・ 地物タイプ（建築物、道路等）
- ・ 重複する地物を表示するか
- ・ LODの範囲



属性情報へのアクセス

PLATEAU SDK for UnrealではCityGMLの属性情報を取得して活用するためのC#向けAPIを提供している。APIの利用方法については、[SDKのユーザーマニュアル](#)に詳細を記載している。また、APIの実際の利用方法を知るためのサンプルとしてブループリント（Plugins/PLATEAU-SDK-for-Unreal/Content/GetObjectInfo.uasset）を提供している。



サンプルの動作方法についても[ユーザーマニュアル](#)を参照されたい。サンプルを動作させると、クリックした建物の属性情報を表示できる。



第2編 PLATEAU VIEW 1.1

第1章 実証環境の構成

1.2 サーバ環境

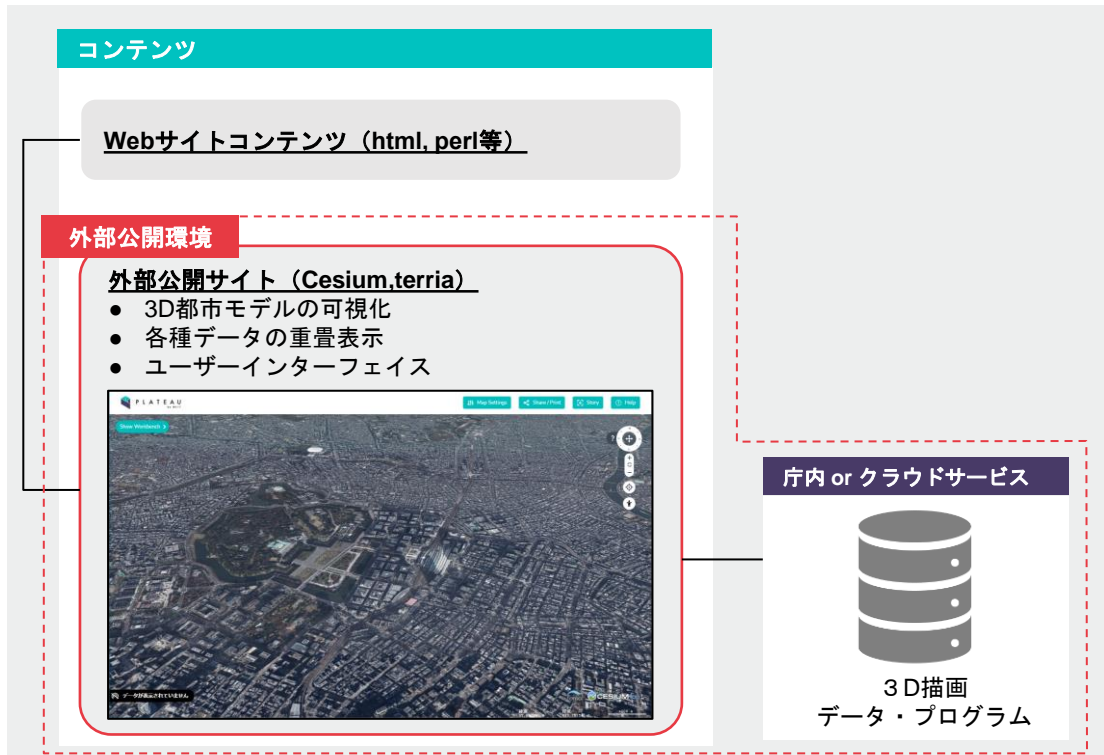
PLATEAU VIEW関連データセットや関連プログラムは、レンダリングデータやオーバーレイデータと共にAWS（Amazon Web Services）上のサーバに格納されており、ユーザからの要求に応じてデータを読み出す仕組みである。PLATEAU VIEW 1.1では、全国約50都市分の巨大なデータを配信する必要があったことから、スケーラビリティに優れた外部クラウドサービスであるAWSを採用している。

他方、地方公共団体において実証環境を構築する場合は、必ずしも外部クラウドサービスを用いる必要はなく、庁内のサーバ（オンプレミス）を利用する方法も可能である。この二つの方法のメリット・デメリットを整理すると以下ようになる。

表 サービス提供方法の比較

	庁内サーバの利用（オンプレミス）	外部クラウドサービスの利用
利用方法	庁内の既存サーバ、新規調達する物理的なサーバを利用	民間企業が提供する外部クラウドサービスを利用
メリット	既存サーバを利用できれば、初期費用を抑えることが可能	搭載するデータ量の増加、利用者の増加への対応（スケールアップ）が容易に対応可能
デメリット	一度購入したサーバのスケールアップ（容量追加、台数追加）は調達上煩雑	インターネット回線の利用が前提であり、庁内からの接続可能な端末が限定される（自治体のセキュリティポリシーに依存）

図 地方公共団体独自サイトの構築例



1.3 サーバ構成

PLATEAU VIEW 1.1では、AWSのサービスであるAmazon EC2（Amazon Elastic Compute Cloud）を利用して、データやプログラムを格納するサーバを構築した。全国約50都市の大規模なデータをリアルタイムに処理する必要があることから、ELB（Elastic Load Balancing）により5台の仮想サーバでトラフィックを自動的に分散している。可視化やUIに関するデータ類の容量は約550GBであり、ストレージはAmazon S3（ストレージ容量が無制限）を利用している。下表に現在のPLATEAU VIEW 1.1を運用しているサーバ構成を示すが、利用者数やデータサイズ、求められるサービスレベルに応じて任意のサイズで構築されたい。

なお、AWSサーバの構築・運用では、サーバ、ストレージ、データ転送について料金が発生する。

表 AWS上のサーバ環境（全国版の事例）

項目	内容
サービス名	Amazon EC2（Amazon Elastic Compute Cloud）
OS	Linux（ubuntu）
プロセッサ	Intel Xeon Platinum 8000
インスタンスタイプ	t3.small
vCPU（仮想CPU）	2
メモリ	2.0 GiB
仮想化タイプ	ハードウェア仮想マシン（hvm）
ロードバランサー（負荷分散装置）	ELB（5台構成）
ストレージ	Amazon S3（Simple Storage Service） （実際の使用容量は約550GB）
リージョン	ap-northeast-1（東京）

自治体向け留意点

各自治体が自らの都市を対象とした環境を構築する際は、PLATEAU VIEW 1.1に比べて扱うデータの規模が小さくなるため、比較的小規模なサーバ構成で足りるものと考えられる。庁内サーバ又は外部クラウドサービスのサーバ環境を構築する際のデータ規模の目安は以下のとおりであり、上述のAmazon EC2を利用する場合は、ストレージは1/50程度、ロードバランサーは1台(1/5)、CPU機能はXeon E5系プロセッサ程度スペックで足りると見込まれる。

- ・建物データの目安：政令市・テクスチャ付き 2GB
基礎自治体・テクスチャなし 0.5～1GB
- ・地形データの目安：0.1～1.5GB（取得精度に依存）
- ・オルソデータの目安：1～50GB（面積に依存）
- ・災害リスクデータの目安：1～2GB（ハザード数に依存）

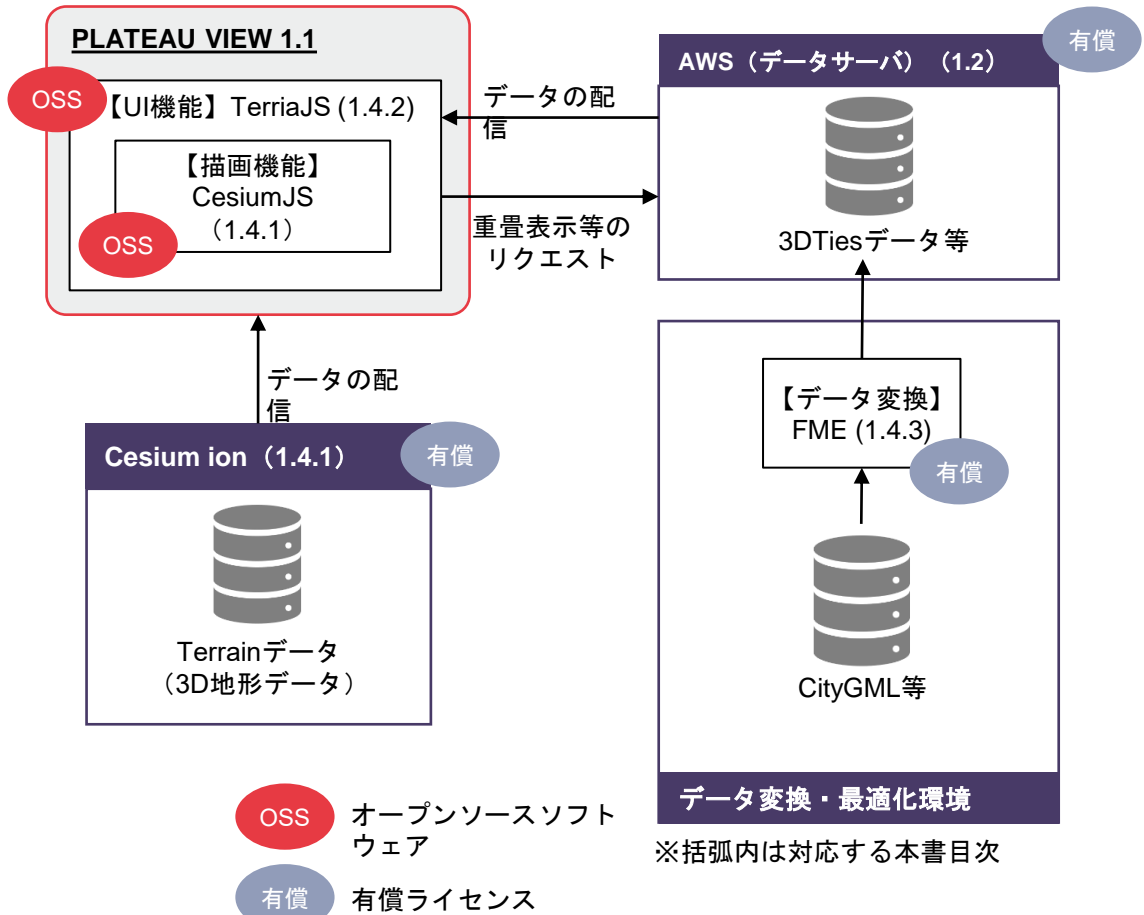
1.4 ソフトウェア構成

PLATEAU VIEW 1.1で使用している主要なソフトウェア類とその機能を解説する。ソフトウェアは、オープンソースソフトウェア（OSS）と有償ライセンスのソフトウェアの両方を組み合わせて利用している。「PLATEAU VIEW 1.1」ではこれらのソフトウェアが有する基本的な機能に加えて、独自に追加・拡張した機能も使用している。

【CesiumJSとTerriaJSの関係】

CesiumJSはデータの描画、TerriaJSはUI（ユーザーインターフェイス）の提供及びUIを介してCesiumJSの描画機能を制御する役割となる。実証環境が有する様々な機能は、TerriaJSを中心に構築されている。なお、各種3Dデータを配置する地表面を表現するデータ（Terrainデータ）については、Cesium社が提供しているクラウドサービスであるCesium ionを使用して全国の地形データの配信を行っている。

図 全国版・実証環境を構成する主要なソフトウェア類の構成



1.4.1 CesiumJS

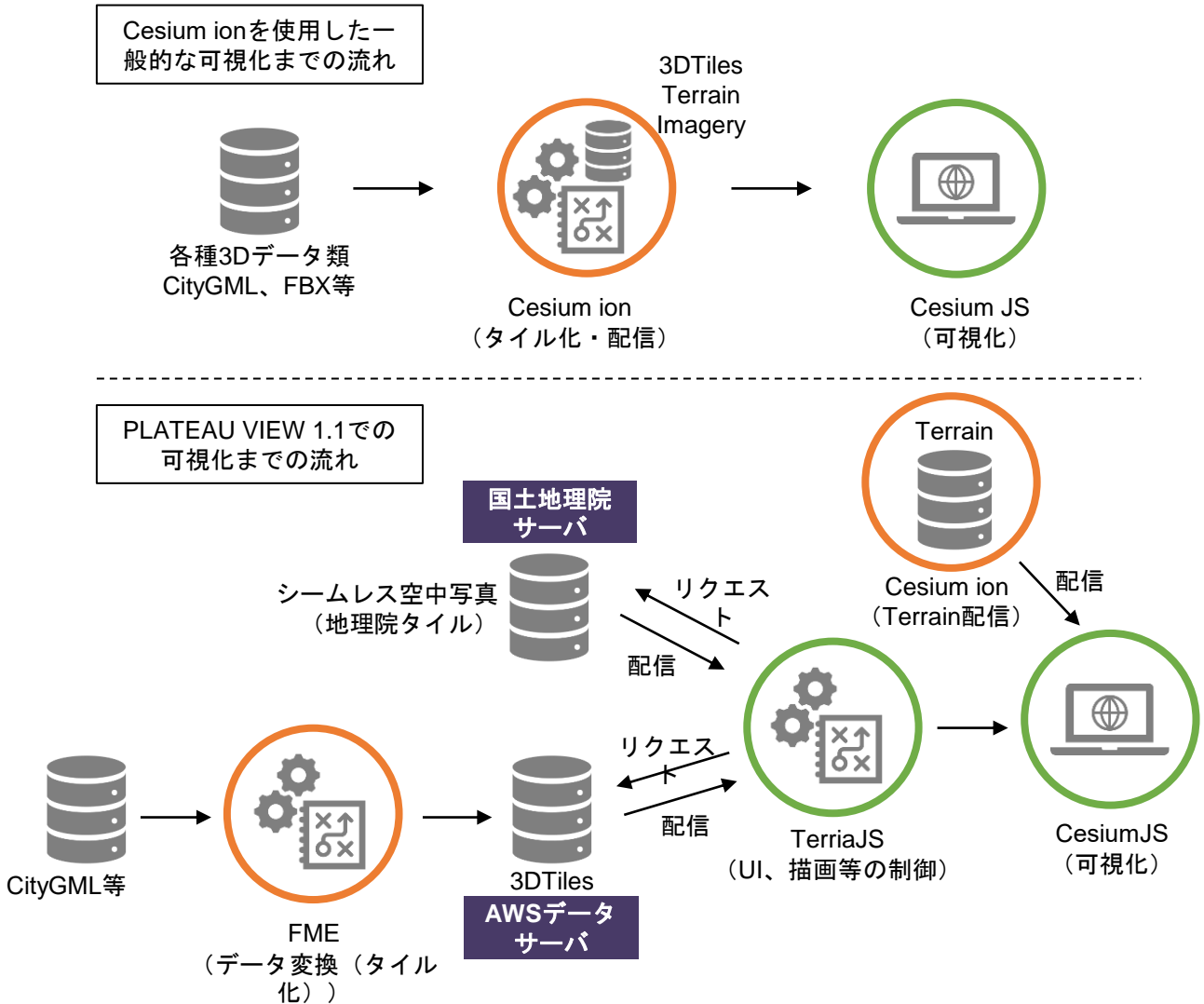
CesiumJSは、3次元位置情報を活用するために設計されたソフトウェアの基盤となるオープンプラットフォームである。WebGLを利用しており、デスクトップ用あるいはスマートフォン用ブラウザで動作する。ブラウザ上で3Dマップを表示するためのJava Scriptライブラリ（CesiumJS）と、様々なファイル形式のデータをストリーミング用に最適化（タイル化）し、配信するサービスとしての機能（Cesium ion）で構成されている。なお、CesiumJSは、オープンソースコードとしてGithubにて公開されている。<https://github.com/CesiumGS/cesium>

CesiumJSでは、多様な形式かつ膨大なサイズのデータをブラウザ上でストレスなくストリーミングおよびレンダリングすることを可能とするために、3DTiles、Terrain（地形）、Imagery（イメージ）という独自の描画機構により軽量化・最適化を図ったファイル形式を採用している。Cesium ionは、様々なファイル形式に対応しており、Cesium社が提供するプラットフォームを介して、ユーザーによるデータのアップロード、管理、タイリング、配信等を行うことができる。

PLATEAU VIEW 1.1では、FME（1.4.3参照）を用いてCityGMLから3DTiles形式への変換を行っており、描画機能としてCesiumJSを使用している（Cesium ionは使用していない）。

なお、Cesium社はCesium ionというクラウドサービスも展開しており、Cesiumがサポートする3Dデータ（p.12）をアップロードすると3D地図として表現するのに適した形式である3DTilesに変換したうえで、ホスティングするサービスを提供している。このサービスを利用すれば、独自にデータの変換（2.4参照）をせずに、参照するURLを取得し、3Dモデルを表示することができる。本事業では、3Dデータの配信には使用していないが、3Dデータを配置する際の基盤となる地形データ（Terrain）の配信に使用している。

図 CesiumJSを使用した可視化までの流れ



【地理院タイル等の配信】

PLATEAU VIEW 1.1では、背景図として選択可能な「空中写真」は、国土地理院サーバーから配信される地理院タイルを取得してCesium上で描画している。地理院タイルとは、国土地理院が配信するタイル状の地図データである。国土地理院では、ユーザーからリクエストされた地図表示範囲に対して、予めタイル状に分割したデータを国土地理院サーバーから配信するサービスを提供している。

<http://maps.gsi.go.jp/development/siyou.html>

また、同じく背景図として選択可能な「空中写真 (Bing)」についてはBingMAP、「地理院地図」についてはOSM (OpenStreetMap) から配信されるデータを描画している。

<https://openstreetmap.jp/>

<https://www.microsoft.com/maps/ja-JP/choose-your-bing-maps-API.aspx#1>

表 変換後にCesiumJSで利用できるファイル形式

描画機構	対応ファイル形式	拡張子	備考
3D Tiles	CityGML	.citygml,.xml,.gml	3次元地理空間データ
	KML/KMZ	.kml,.kmz	KML/KMZ
	LASeR	.las,.laz	点群データ(LASファイル)
	COLLADA	.dae	3D-CG間の交換フォーマット
	Wavefront OBJ	.obj	3Dジオメトリのみの表現形式
Terrain (地形)	Floating Point Raster	.flt	
	Arc/Info ASCII Grid	.asc	
	Source Map	.src	
	Geo TIFF	.tiff,.tif	
	Erdas Imagine	.img	
	USGS ASCII DEM /CDED	.dem	
	Cesium Terrain Database	.terraindb	
Imagery (イメージ)	Floating Point Raster	.flt	
	Arc/Info ASCII Grid	.asc	
	Source Map	.src	
	Geo TIFF	.tiff,.tif	
	Erdas Imagine	.img	
	USGS ASCII DEM /CDED	.dem	
	Cesium Terrain Database	.terraindb	
	JPEG	.jpg,.jpeg	
	PNG	.png	

表 TerriaJSがサポートするサービス

描画機構	区分	内容
グラフィックオブジェクト	ポイント	2D/3D空間上に一時的なグラフィックスを描画
	ライン	
	ポリゴン	
	ラベル	
	マーカー	
外部サービス (高解像度画像)	WMS	背景地図を表示
	TMS	
	WMTS	
	ArcGIS	
	Bing Maps	
	Google Earth	
	Mapbox	
	Open Steet Map	
外部サービス (地形データ)	Open Steet Map Buildings	海、湖、川の地球規模での高解像度の地形を表示
	Cesium World Terrain	
	Google Earth Enterprise	
ベクターデータ	VT MAK VR- TheWorldサーバ	直接参照可能なファイルフォーマット
	GeoJSON	
	TopoJSON	
	CZML	
	KML,KMZ	

1.4.2 TerriaJS

TerriaJSは、CesiumJS向けにUIを付加したり、様々な形式のデータを読み込めるようにするオープンソースのフレームワークである。<https://terria.io/>

CesiumJSの描画機能をコントロールする役割を持っており、PLATEAU VIEW 1.1では、TerriaJSのUI機能を使用して、表示データの選択・追加・削除（Add Data、データカタログ、ワークベンチ）、背景図の選択（Map Settings）、URL生成／印刷（Share/Print）、ストーリーの設定（Story）、Help、自分のデータを追加等を提供している。

【TerriaJSの拡張】

TerriaJSは機能を追加することができる。PLATEAU VIEW 1.1では、CityGMLで作成した建物モデルに付与された属性情報を検索する機能や歩行者ビュー機能等を追加で開発した（開発言語：JavaScript）。



UI機能	内容
Add Data	Data Catalogue画面の起動
Data Catalogue	データの選択
My Data	ローカル及びWeb上のデータをアップロード
ワークベンチ	Data Catalogueで選択したデータの表示
Map Settings	背景地図の選択・地図機能の設定
Share / Print	URL生成と印刷
Story	ストーリーエディタの起動（複数のポイントでシーン設定をして保存、ポイントを移動しながら再生ができる）
Help	ヘルプ（マウス操作）の起動
ジャイロスコープ	ビューの回転・傾け、初期位置の表示
ズーム	ズームイン／ズームアウト
現在地表示	自位置に地図を寄せる
歩行者モード	歩行者モードの開始

表 TerriaJSによって実装された主な機能一覧（1／2）

機能名称	操作	内容	機能区分		
			基本	追加	
				ver1	Ver1.1
3次元視点移動	マウス移動	任意の位置に視点を移動	○		
	エリア選択移動	選択した都市への視点移動	○		
	UC選択移動	選択したUC（ユースケース）への視点移動	○		
属性情報の表示	地物選択	【選択時に表示される属性情報の例】 建物：高さ、階数、住所、都市計画区域、用途地域、建物用途など 鉄道駅：路線名、運営会社、駅名など 避難施設：施設名、住所、避難施設の種類など	○		
属性情報の色分け表示	属性選択（排他制御）	特定のコードを持つ地物に色を割り当てて表示		○	
属性情報のピックアップ表示	属性選択（範囲選択）	特定の属性項目に属するコードにしきい値を設定し、しきい値以上のコードを持つ地物のみを表示など、属性値に基づいた絞り込み		○	
検索機能	キーワード選択・入力	特定の属性項目に属するコードを検索して表示		○	
オーバーレイデータの表示	地図選択 [Map Settings]	3D地形データ（DEM）の表示	○		
		2D表示	○		
		空中写真（国土地理院）の表示		○	
		空中写真（Bing）の表示	○		
		地理院地図（淡色）の表示	○		
		Dark Matter地図の表示	○		
	静的情報表示	建物関係、都市インフラ、都市計画、災害系情報、国土数値情報（駅、避難施設など）の表示	○		
	動的情報表示	人流、バス走行位置、コミュニティサイクル走行状況（基本） リアルタイムデータ（GTFS-RT、CSVなど）の表示（ver2）	○		○
日影表示	日影表示	受光、投影、受光＋投影の3パターンから選択して日影を表示、あるいは非表示の選択	○		

基本機能：TerriaJSに元々備わっている基本機能を利用してPLATEAU VIEW 1.1に実装した機能

追加機能：TerriaJSの機能を拡張してPLATEAU VIEW 1.1に実装した機能（実装後TerriaJSに統合された機能も含む）

※表中の[]は、P.13に示したUI機能を指す

表 TerriaJSによって実装された主な機能一覧（2／2）

機能名称	操作	内容	機能区分		
			基本	追加	
				ver1	Ver1.1
基本操作機能	開始地点に戻る	Homeボタンを配置し任意の1点に戻る	○		
	[歩行者モード]	視点と注目点を決めて、アイレベルから見たシーンを表示（ver1） 地表面だけでなく、3Dオブジェクト（建物モデルなど）の表面に沿って移動する機能（ver2）		○	○
	メニュー表示・非表示（折りたたみ）	画面上に表示される各種情報を折りたたむことができる機能	○		
	印刷機能 [Share/Print]	印刷、PDFファイル出力	○		
	クリップ機能	3Dモデルを任意の直方体で切り取り表示する機能			○
現在地表示	[現在地表示]	GPSで取得した現在地を表示	○		
スマホ対応		スマートフォン上に表示	○		
データカタログ作成UI	[自分のデータを追加]	ユーザーがデータをアップロードして、データカタログに新たなレイヤとして登録	○		
データカタログからの選択	[データカタログ]	データカタログに登録してあるデータレイヤを選択して、重ね合わせる	○		
ストーリー作成と保存	[Story]	ルートを設定し動画を作成	○		
パブリッシュ設定	[Share/Print]	外部向けの公開サイトにURL等を発行	○		
タイムライン表示	[Map Settings]	タイムラインの表示・非表示切り替え			○

基本機能：TerriaJSに元々備わっている基本機能を利用してPLATEAU VIEW 1.1に実装した機能

追加機能：TerriaJSの機能を拡張してPLATEAU VIEW 1.1に実装した機能（実装後TerriaJSに統合された機能も含む）

※表中の[]は、P.13に示したUI機能を指す

1.4.3 FME

FMEは、位置情報データ変換の世界標準ソフトであり、スマートシティ向けのシステム構築でも利用されている。CityGML、IFC等400種類以上のフォーマットに対応し、プログラミング不要の画面操作だけで、データ、サーバー、外部APIデータに接続し、データ変換、各種ビューアへの取り込みができる。トランスフォーマーと呼ばれるデータ変換ロジックを有したパーツを組み込んだワークスペースを作成することにより、多種多様な形式のデータ構造・内容を自由に変換することができる。また、作成したワークスペースをFME Serverに登録することにより、データの変換作業を自動化することも可能である。

FMEは、多様なデータ形式に対応したトランスフォーマー（データ変換ロジック）をライブラリとして用意しており、PLATEAU VIEW 1.1では、CityGML形式等から3DTiles形式への変換を行っている。

FMEは以下の製品・サービスから構成され、有料ライセンスソフトウェアである。

FME Desktop: ワークスペース（データ変換のフロー）を作成、実行するためのツールセット

FME Server: ワークスペースに基づくデータ変換サービスを提供するためのサーバーソフトウェア

FME Cloud: クラウド上でFME Serverの機能を提供するサービス(時間課金または年課金)

<https://pacificspatial.com/fme/>

なお、PLATEAU VIEW 1.1で用いたFMEの変換テンプレートは以下のURLで公開している。

<https://github.com/Project-PLATEAU/FMEScript-CityGML-to-3DTiles>

図 FMEによるデータ変換処理のイメージ

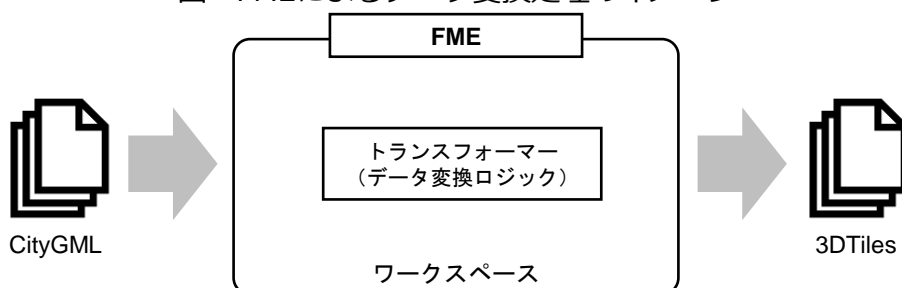
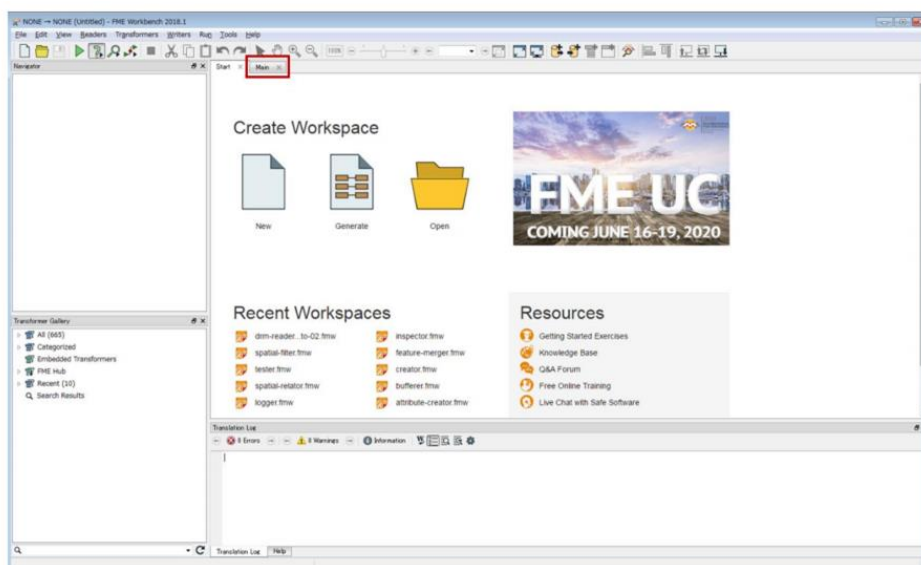


図 FMEの画面 (Workbench)



1.5 実証環境のデータ構成

1.5.1 データの種類

「PLATEAU VIEW 1.1」で表示可能なデータセットの種類の例と出典を示す。

表 東京都の主なデータセット（1／2）

データ区分/データ名		出典
建物関連	建物モデル ※LOD1及びLOD2の3D都市モデル	建物図形：土地利用現況・建物現況調査（東京都）、航空写真測量（国際航業） 計測高さ：LOD1 航空レーザー測量（国際航業）、LOD2 航空写真測量（国際航業） 建物テクスチャ：航空写真（国際航業、八王子市） 建物現況：建物現況調査（東京都） 都市計画情報：東京都の都市計画地理情報システムデータ 土砂災害警戒区域：国土数値情報 津波浸水想定：東京都津波浸水想定 洪水浸水想定区域：浸水想定区域図（国土交通省）、浸水予想区域図（東京都市型水害対策連絡会）
	避難施設	国土数値情報（国土交通省）
	ランドマーク	建物の属性（名称）：国土地理院発行の数値地図（国土基本情報）（測量法に基づく国土地理院長承認（使用）R 2JHs 844」を加工して作成。
	鉄道駅	国土数値情報（国土交通省）
都市アセット	道路	土地利用現況（土地利用現況・建物現況調査の結果や航空写真等）
	公園	土地利用現況（土地利用現況・建物現況調査の結果や航空写真等）
	橋梁モデル ※橋梁、道路高架部、鉄道高架部、歩道橋、ペディストリアンデッキ	航空写真（国際航業、八王子市）
交通	鉄道	国土数値情報（国土交通省）
	緊急輸送道路 ※第1～3次緊急輸送道路	国土数値情報（国土交通省）
都市計画	都市計画道路	都市計画道路（東京都）2都市基街都第247号
	都市高速鉄道	都市高速鉄道（東京都）2都市基交都第56号
	地区計画	都市計画地理情報システムデータ（東京都）
	地域地区	都市計画決定情報GISデータ
	用途地域	都市計画決定情報GISデータ
	土地利用 ※LOD1及びLOD2の3D都市モデル	土地利用現況・建物現況調査（東京都）
	行政界	国土数値情報（国土交通省）

表 東京都の主なデータセット（2 / 2）

データ区分/データ名		出典	
都市活動	高速道路、一般道 ※車の移動速度から求めた渋滞状況	スマートフォンアプリを通じて許諾を取得したアプリユーザーの位置情報を、個人特定を回避する形で加工したデータ（フリックテック株式会社）	
	地価バリューマップ	国土交通省国土政策局 国土数値情報 「地価公示」、「都道府県地価調査」をもとに株式会社日建設計総合研究所が編集・加工	
	東日本大震災時の帰宅困難者 ※東京駅周辺、池袋駅周辺	国土交通省都市局都市安全課、株式会社日建設計総合研究所	
	（比較用）平常時の滞留者 ※東京駅周辺	国土交通省都市局都市安全課、株式会社日建設計総合研究所	
	経済センサス 事業所数 平成28年度	総務省統計局 「平成28年度経済センサス・基礎調査」	
	経済センサス 従業者数 平成28年度	総務省統計局 「平成28年度経済センサス・基礎調査」	
洪水浸水想定区域図			
	洪水浸水想定区域図 ※荒川、江戸川、利根川、多摩川、城南地区、江東内部河川、石神井川および白子川、神田川、野川	河川事務所、東京都が作成した浸水想定区域図	
ユースケース例	大丸有	丸の内ストリートパーク人流データ	大丸有エリアマネジメント協会、大手町・丸の内・有楽町地区まちづくり協議会、三菱地所株式会社
		丸の内シャトル：バス走行位置、バス停、走向ルート	日の丸自動車興業株式会社
		千代田区コミュニティサイクル：利用状況、サイクルポート	株式会社ドコモ・バイクシェア
	渋谷区	渋谷人流データ	許諾に基づきスマートフォンアプリを通じてユーザーの位置情報を取得し、個人特定を回避する形に処理したうえで株式会社ブログウォッチャーが提供するデータを編集・加工
	羽田	羽田イノベーションシティ BIMデータ	鹿島建設株式会社
	竹芝	東京ポートシティ竹芝 BIMデータ	鹿島建設株式会社
	リアルタイム	東京都交通局 バスロケーション情報	東京公共交通オープンデータチャレンジ
		西武バス バス関連リアルタイム情報	東京公共交通オープンデータチャレンジ
		関東地方整備局 河川ライブカメラ	関東地方地方整備局京浜河川事務所、荒川下流河川事務所

表 東京都以外の主なデータセット（例：横浜市）

データ区分/データ名		出典
建物関連	建物モデル ※LOD1あるいはLOD1及びLOD2の3D都市モデル	建物図形：都市計画基本図、都市計画基礎調査（各自治体） 計測高さ：航空写真測量（各自治体） 建物現況：建物現況調査（各自治体） 土地利用現況：土地利用現況調査（各自治体） 都市計画情報：都市計画区域（各自治体） 土砂災害警戒区域：国土数値情報（国土交通省） 洪水浸水想定区域：浸水想定区域図（国土交通省） 津波浸水想定区域：津波浸水想定図（各自治体） 建物の属性：名称は、数値地図（国土基本情報）「国土地理院」を加工。 「測量法に基づく国土地理院長承認（使用）R 2JHs 809」 建物の属性：名称（駅名）は、「国土数値情報（鉄道データ）」「国土交通省」を加工。
	避難施設	国土数値情報（国土交通省）
	鉄道駅	国土数値情報（国土交通省）
都市アセット	道路	都市計画基本図、基盤地図情報
	公園	国土数値情報（国土交通省）
交通	鉄道	国土数値情報（国土交通省）
	緊急輸送道路 ※第1～3次緊急輸送道路	国土数値情報（国土交通省）
都市計画	区域区分	都市計画情報（各自治体）
	都市計画区域	都市計画情報（各自治体）
	用途地域	都市計画情報、土地利用現況図（各自治体）
	土地利用	土地利用現況図（各自治体）、国土数値情報（国土交通省）
	行政界	国土数値情報（国土交通省）
洪水浸水想定区域		河川事務所、各自治体が作成した浸水想定区域図
土砂災害想定区域		各地方自治体が作成した土砂災害想定区域
津波浸水想定区域図		各地方自治体が作成した津波浸水想定区域図
ユースケース例	人流データ（赤外線センサー）（横浜市）	赤外線センサーを用いて人数を計測（株式会社日建設計総合研究所）
	みなとみらい クイーンズスクエア横浜（地下鉄改札～地上階・共用部分） ※LOD4の建物モデル	3DshapeデータをCityGML形式に変換（株式会社日建設計総合研究所）
	リアルタイムデータ	横浜市交通局 バス関連リアルタイム情報（東京公共交通オープンデータチャレンジ）

1.5.2 データの形式

実証環境で扱った主なデータ形式及びデータ変換パターンの例を示す。

表 データ形式の一覧

データ形式	概要
CityGML (City Geography Markup Language) ※1、2	<ul style="list-style-type: none"> OGC (Open Geospatial Consortium)が策定した3D都市モデルのためのオープンデータモデル及びデータ形式の国際標準を指す。 建築物や道路、橋梁などの様々な地物（オブジェクト）について定義し、これに名称や用途、建築年、行政計画といった都市活動に関する情報（セマンティクス）を付与することで、都市空間の意味や地物間の関係性を再現したジオメトリ（幾何形状）とセマンティクスを統合したモデルである。 地物や地物属性を追加できる拡張機能やADE（Application Domain Extension）と呼ばれる、CityGMLの仕様自体を拡張し、地物や属性の応用スキーマを新たに定義する機能も有している。 建物等の地物の表現に関して、LOD（Level of Details）と呼ばれる概念を採用することで、同じ地物に関する詳細度の異なる様々な情報を統合的に管理・蓄積・利用することを可能としている。
3D Tiles※3	<ul style="list-style-type: none"> 3D建物、BIM/CAD、点群データ（ポイントクラウド）、地形データ、写真測量などの3次元地理空間コンテンツをストリーミング及びレンダリングするための空間データの構造とタイルフォーマットを定義した形式である。OGC標準としても採用されている。 軽量・最適化されたデータ形式のため、3DデータでもスムーズなWeb配信が可能であり、Cesiumで採用されている。
CZML※4	<ul style="list-style-type: none"> 主にCesiumで用いられるJSON形式を基本としたデータ形式である。 3Dデータ及び時間データの表現に適しており、時間の経過と共に値を変更するプロパティを正確に記述することができる（例えば、ある時間帯は赤色の線、ある時間帯は青色の線を表示する等）
Moving Feature Jsn※5、6	<ul style="list-style-type: none"> 3次元形状の物体の移動データを簡潔に記述する形式として開発され、OGCが移動体データ形式の国際標準として採択している。 移動体（人や自動車など）の動的な空間情報を一体的に記録する。
GeoJSON※7、8	<ul style="list-style-type: none"> JSONを使用して様々な地理データ構造をエンコードするためのフォーマットである。 GeoJSONオブジェクトは、GeometryとPropertyを含むFeature、または、複数のFeatureのリストから構成される。Geometryには、Point、LineString、Polygon、等が含まれ、プロパティ（属性）として、オブジェクトのタイトルや説明、ポイントの大きさ・色、線の太さ・色などの情報を持つことができる。
Shapefile※9	<ul style="list-style-type: none"> トポロジー構造をもたない空間データの位置と形に関する情報と、その属性情報を格納するためのデータ形式である。位置と形に関する情報は、（座標値のベクトルで構成される）シェープとして格納される。 地理空間上の要素をベクター形式であるポイント、ライン、ポリゴンで表現し、各要素に任意の属性を付与できる。
FBX※10	<ul style="list-style-type: none"> モデル ジオメトリ、マテリアル テクスチャ、照明、およびアニメーション シーケンスに関連するデータが格納されている。 動く3Dオブジェクトの記述に向いている。
MVT (Mapbox Vector Tile)※11	<ul style="list-style-type: none"> 画像データをブラウザ上で高速に表示するために開発されたマップタイル形式の考え方をベクターデータ（点、線、面）について適用し、これらのデータの高速配信、クライアントでの高速描画を実現したデータ形式で、個々のオブジェクトへの属性の付与やアプリケーション側での表示スタイル（色）の設定が可能である。 Mapbox Inc.等により開発されている。

※1 標準製品仕様書

※2 3D都市モデルの導入ガイド

※3 <https://github.com/CesiumGS/3d-tiles/blob/master/README.md>

※4 <https://github.com/AnalyticalGraphicsInc/czml-writer/wiki/CZML-Guide>

※5 https://www.aist.go.jp/aist_j/press_release/pr2020/pr20200602_2/pr20200602_2.html

※6 <http://docs.opengeospatial.org/is/19-045r3/19-045r3.html>

※7 <https://geojson.org/geojson-spec.html>

※8 <https://docs.geolonia.com/geojson/>

※9 <https://www.esri.com/getting-started/learn-more/shapefile/>

※10 <https://knowledge.autodesk.com/ja/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2019/JPN/3DSMax-Data-Exchange/files/GUID-26E80277-1645-4C4E-A6B2-44399376490F-htm.html>

※11 <https://github.com/mapbox/vector-tile-spec/blob/master/2.1/README.md>

表 データ変換パターンの例

データ区分		元データ形式	変換後データ形式	データ変換の留意点
建物関連	建物	CityGML	3DTiles	<ul style="list-style-type: none"> 元データ（CityGML）の品質が担保されていれば、正常の3DTilesへの変換は可能 建物属性のコード（数値）は、文字列に変換する処理を追加
	都市計画	CityGML	GeoJSON、MVT	<ul style="list-style-type: none"> ポリゴン内に中空の部分がないように分割処理が必要 分割し過ぎるとデータ量が肥大化し、動作性能が低下
	道路	CityGML	GeoJSON、MVT	<ul style="list-style-type: none"> 同上
	洪水浸水想定区域	CityGML	3DTiles	<ul style="list-style-type: none"> 元データ（CityGML）を3DTilesに変換すると描画性能が低下する場合がある 浸水域が広い場合（例：東京都の荒川）には描画性能を考慮しデータを分割するか、CityGMLを作成中に発生する3D Shapeファイルを用いて3DTilesに変換する方法もある
	土砂災害警戒区域	CityGML	GeoJSON、MVT	<ul style="list-style-type: none"> 土砂災害特別警戒区域は土砂災害警戒区域の内側に設定されているため、両者のデータを任意に選択する場合、重なり部分の土砂災害警戒区域をくり抜いて作成する必要がある
	津波浸水想定区域	CityGML	CZML	<ul style="list-style-type: none"> 洪水浸水想定区域と同様
	内水・高潮想定区域	CityGML	CZML	同上
BIM	建物	CityGML	3DTiles	<ul style="list-style-type: none"> 元データの構成部材を確認し、変換後の着色方法を事前に規定することが必要 建築物に絶対座標が付与されていない場合には、1か所基準点を定め、絶対座標を付与したのち、建物の1面の方位角を規定、PLATEAU VIEW 1.1上に位置合わせを行う
	建物、地下空間	FBX	3DTiles	同上
動的データ	人流	JSON,CSV	CZML	<ul style="list-style-type: none"> データ量が非常に多く、描画方法を事前検討した上で、必要なデータを絞り込み変換
	GPSデータ	JSON,CSV	CZML	<ul style="list-style-type: none"> 高層ビルが多いエリアでは、位置精度が低下し、異常な動き（例：バスが後退する）をすることから、データクリーニング処理が必要

第2編 PLATEAU VIEW 1.1

第2章 実証環境の構築手順

2.1 実証環境構築の前提

本章では、実証環境「PLATEAU VIEW 1.1」のクローンを構築するための手順を説明する。本手順は、2020年度から2022年度に公開されていたPLATEAU VIEW 1.1と同様にAmazon Web Service（以下、「AWS」という。）の利用を前提とした。重畳データの変更やユーザーインターフェースの変更、機能の追加などを行う場合は、Project PLATEAUのGitHub（<https://github.com/Project-PLATEAU>）のドキュメントを参照されたい。

2.1.1 作業環境の準備

作業するマシンに以下の環境を準備する。本手順ではWindows10におけるプロセスを説明する。

- Node.js
- npm
- yarn
- awscli
- Git

(1) Node.js

JavaScriptは通常ブラウザで実行するが、これをサーバ側で実行するためにNode.jsを使用する。Node.jsのインストールは下記の手順で行う。Node.jsに関する説明はNode.jsのウェブサイト（<https://nodejs.org/ja/about/>）を参照されたい。

Node.jsのウェブサイト（<https://nodejs.org/ja/download/>）からお使いの使用環境に対応したインストーラ（v10.0以降）を入手し、実行する。



インストールが完了したら、コマンドプロンプトを起動して、以下のコマンドを実行し、動作を確認する。

```
node --version
```

```
C:\Users\Yinfo>node --version
v14.17.6
```

インストールしたバージョンと同じ数字であることを確認。

(2) npm

npmは(1)でインストールしたNode.jsのモジュールを管理するツールで、Node.jsと同時にインストールされるので、ここではインストールされたnpmのバージョンを表示して動作を確認する。本手順の作業を行うためには、v6.0以降が必要である。Node.jsと同様にコマンドプロンプトを使ってバージョンを確認する。

```
npm --version
```

```
C:\Users¥info>npm --version
6.14.15
```

v6.0以降であることを確認。

(3) yarn

近年のJavaScriptアプリケーションは複数のJavaScriptファイルで構成されており、その中には再利用可能な部品 (node modules) がある。yarnや前出のnpmはこれらの部品のインストールを管理するもので、インストールすべき部品が記述されたpackages.jsonに従って、yarn installやnpm installといったコマンドでインストールを実行することができる。—また、yarnやnpmを使えば、このpackages.jsonの内容を編集することもできる。

yarnは以下のコマンドでインストールし、バージョンを表示して動作を確認する。

```
npm install -g yarn
```

```
C:\Users¥info>yarn --version
1.22.17
```

v1.19.0以降であることを確認。

(4) awscli

awscli (AWS Command Line Interface)は、AWSをコマンド入力で操作できるようにするインターフェースである。AwscliについてはAWSのドキュメント (https://docs.aws.amazon.com/ja_jp/cli/latest/userguide/cli-chap-welcome.html) を参照されたい。

Windowsには、インストーラ (<https://awscli.amazonaws.com/AWSCLIV2.msi>) が用意されているので、これを利用してインストールを実施する。上記のリンクからダウンロードしたMSIファイルをダブルクリックしてインストールを実行し、コマンドプロンプトからバージョンを表示して動作を確認する。

```
aws --version
```

```
C:\Users¥info>aws --version
aws-cli/2.4.23 Python/3.8.8 Windows/10 exe/AMD64 prompt/off
```

aws-cli/で始まるメッセージが表示されればインストールが成功。

(5) Git

Gitは今回のインストールに必要なパッケージをPLATEAU VIEW 1.1のGitHubから取得する際に使用する。GitのインストールはGitHub Desktopのホームページ（<https://desktop.github.com/>）からインストーラを入手して実施する。



これまでと同様、コマンドラインでバージョンを表示してインストールを確認する。

```
git --version
```

```
コマンドプロンプト
C:\Users\¥info>git --version
git version 2.34.1.windows.1
```

2.1.2 AWSの準備

この手順ではAWSアカウントの作成については割愛するが、参考となるAWSのドキュメントのURLを示す。<https://aws.amazon.com/jp/register-flow/>

(1) AWSプロファイルの用意

まず、上記で作成したAWSアカウントの情報を使って、AWSマネジメントコンソール(<https://aws.amazon.com/jp/console/>)にログインし、サービス一覧からIAMを選択する。さらに、IAMの画面から「ユーザー」を選択し、一覧に表示されるユーザーの中からAdministratorAccess権限を持っているユーザーを選択する（この作業のためにアカウントを作られた場合は、表示されているユーザーがAdministratorAccess権限を持つユーザーとなる）。さらに「認証情報」タブを選択すると「アクセスキーの作成」ボタンが表示されるので、これをクリックする。ここで表示されるアクセスキーIDとシークレットアクセスキーを使用して以下の設定を行う。この情報はここでしか確認できないので、手元に控えるか、「.csvファイルのダウンロード」ボタンを使ってcsvファイルとして保管する。ここで取得したアクセスキーIDとシークレットアクセスキーのセットをAWSプロファイルと呼ぶ。

次に、コマンドプロンプトを開いて、以下のコマンドを実行し、プロファイルを設定する。ここでは「plateau」というプロファイル名で作成する。

```
aws configure --profile plateau
```

AWS Access Key ID、AWS Secret Access Key、Default region name、Default output formatの入力を順次求められるので、AWS Access Key IDとAWS Secret Access Keyについては先ほど作成したものを入力し、Default region nameでは「ap-northeast-1」、Default output formatは「json」を入力する。さらに、以降の作業でこのプロファイルを自動的に読み込むため、環境変数に登録する。

```
setx AWS_PROFILE plateau
```

(2) AWS Route53

AWSコンソールを使用して、AWS Route53のパブリックホストゾーンを作成する。作成の方法は下記のAWSドキュメントを参照されたい。AWS Route53は、サーバやロードバランサをドメインに割り当てるために使用する、AWS製のDNSのようなものである。ホストゾーンとはAWS Route53の用語で、例えばplateauview.mlit.go.jpやhogehoge.mlit.go.jpというような、同じルートドメインのDNSレコードを管理するスペースのことをいう。また、ホストゾーンにはパブリックとプライベートがあり、パブリックはインターネット上に公開されるもので、プライベートは閉じられたネットワーク（VPC, Virtual Private Cloud）のDNS管理スペースとなる。

https://docs.aws.amazon.com/ja_jp/Route53/latest/DeveloperGuide/AboutHZWorkingWith.html

(3) EC2 (Amazon Elastic Compute Cloud) のキーペア

EC2は今回PLATEAU VIEW 1.1をインストールするサーバの名称で、ここにSSHでアクセスするためには公開鍵と秘密鍵のキーペアを取得する必要がある（今回のインストールではSSHでのアクセスはしないが、EC2を立ち上げる際の必須のパラメータなので取得する）。キーペアの作成方法に関するドキュメントは下記の通りだが、次のコマンドを実行して取得する。

https://docs.aws.amazon.com/ja_jp/cli/latest/userguide/cli-services-ec2-keypairs.html

```
aws ec2 create-key-pair --key-name MyKeyPair --query 'KeyMaterial' --output text >
MyKeyPair.pem
```

（４）AWS SSL Certificate

AWS Certificate Manager (<https://aws.amazon.com/jp/certificate-manager/>) を使用して画面に従って証明書をリクエストし、SSL/TLS証明書（AWS SSL Certificate）を作成する。

SSL証明書はウェブサイトをhttpsで公開するために必要で、そのサイトの公開元が実在することを証明するものである。

（５）S3バケット

AWSマネジメントコンソール (<https://aws.amazon.com/jp/console/>) からS3を選択し、「バケットを作成」ボタンをクリックし、画面に従ってAWS S3バケットを作成する。

AWS S3バケットはAWSのファイルストレージサービスで、PLATEAU VIEW 1.1のインストールの際に使用するPLATEAU VIEW 1.1のアプリケーションパッケージを圧縮したものを保管するために使用する。先ほど設定したEC2インスタンス（サーバ）がこの圧縮されたパッケージをダウンロードして展開し、PLATEAU VIEW 1.1のアプリケーションを起動する。

2.2 インフラ構築とデプロイ

2.2.1 TerriaMapのクローン

以下のコマンド（git clone）を使ってProject-PLATEAUのGitHubからTerriaMapを作業マシンにクローンする。2行目のcdは、クローンしたディレクトリ（TerriaMap）に移動するためのコマンド。

```
git clone -b plateauview git@github.com:Project-PLATEAU/PLATEAU-VIEW.git
cd TerriaMap
```

2.2.2 TerriaJSのクローン

以下のコマンド（mkdir）を使って、2.2.1でクローンした"TerriaMap"ディレクトリ内に"packages"ディレクトリを作成し、そこにProject-PLATEAUのGitHubからTerriaJSを作業マシンにクローンする。

```
mkdir packages && cd packages
git clone -b plateauview git@github.com:Project-PLATEAU/terriajs.git
cd ../../
```

2.2.3 設定ファイルの編集

（1）TerriaMap/package.jsonの編集

2.2.1, 2.2.2で作業マシンにクローンしたファイルの内、"TerriaMap"直下にあるpackage.jsonを編集する。package.jsonは2.1.2で準備したAWSの環境に関する情報を記述するファイルで、変更する各項目の内容は「表 package.jsonの項目について」に示す通り。

表 package.jsonの項目について

項目名	内容
packageName	アプリケーションのパッケージ名（任意で設定。次ページの例ではplateauview）
awsS3PackagesPath	アプリケーションのパッケージを保管するS3のパス（2.1.2（5）で作成したバケットのURL：東京リージョンで作成していれば http://s3-ap-northeast-1.amazonaws.com/ バケット名）
awsRegion	AWS のリージョンID（東京リージョンで問題なければ次ページの例と同じ：ap-northeast-1）
awsEc2InstanceType	EC2のインスタンスタイプ（この例ではt3.smallを使用（p.8参照））
awsEc2ImageID	EC2のAMIのID（Ubuntu 18.04） ※PLATEAU VIEW 1.1は上記で動作確認済み
awsKeyName	EC2のキーペア（2.1.2（3）で取得したもの。例にある“ami-00bc9b7f0e98dc134”と“my-great-key-pair”を置き換える。）

(変更例)

```
"config": {
  "packageName": "plateauview",
  "awsProfile": "terria",
  "awsS3PackagesPath": "s3://my-great-bucket/viewer",
  "awsRegion": "ap-northeast-1",
  "awsEc2InstanceType": "t3.small",
  "awsEc2ImageId": "ami-00bc9b7f0e98dc134",
  "awsKeyName": "my-great-key-pair",
```

(2) TerriaMap/deploy/aws/stack.json の編集

"TerriaMap/deploy/aws" ディレクトリにあるstack.jsonを編集する。stack.jsonは2.1.2で準備したRoute53のパブリックホストゾーンおよびSSL/TLS証明書（AWS SSL Certificate）に関する情報を記述するファイルで、変更する各項目の内容は「表 stack.jsonの項目について」に示す通り。

表 stack.jsonの項目について

項目名	内容
Parameters.HostedZoneName.Default	Route53のパブリックホストゾーン
SSLCertificateId	AWS SSL CertificateのARN

2.2.4 nodejs依存モジュールのインストール

下記のコマンドを実行し、nodejs依存モジュールをインストールする。このコマンド（yarn）を実行することで、packages.jsonのdependenciesやdevDependenciesに記述された部品がインストールされる。

```
yarn
```

2.2.5 インフラ構築とデプロイ

下記のコマンドを実行し、インフラ構築（サーバ、ロードバランサの整備とDNSレコードの構築）とデプロイを完了する。

```
export NODE_OPTIONS=--max_old_space_size=4096
yarn deploy-without-reinstall
```

これでPLATEAU VIEW 1.1のコピーが完成する。表示データを変更する場合は、2.5カタログの編集を参照されたい。

2.3 Dockerを用いた環境構築

AWSを使わずに実証環境を構築する方法として、Dockerを使用した方法を示す。Dockerファイル及び使用するソースコードはProject-PLATEAUのGitHub (<https://github.com/Project-PLATEAU/terria-docker-example>) から入手する。

2.3.1 前提条件

この手順による実証環境の構築では、構築する環境にDockerがインストールされているものとする。

2.3.2 環境構築の手順

(1) PLATEAU VIEW 1.1のソースコードのクローン

下記のコマンドを実行して、PLATEAU VIEW 1.1のソースコードをクローンする。

```
cd terria/src
git clone https://github.com/Project-PLATEAU/PLATEAU-VIEW-1.1.git TerriaMap
cd TerriaMap
mkdir packages && cd packages
git clone https://github.com/Project-PLATEAU/terriajs.git terriajs
cd ../../..
```

(2) Dockerイメージのビルド

本プロジェクトのルートディレクトリ（docker-compose.ymlがあるディレクトリ）で以下を実行する。

```
docker compose build
```

(3) PLATEAU VIEW 1.1の起動

下記を実行し、PLATEAU VIEW 1.1を起動すると、ポート80でPLATEAU VIEWが見れる。

```
docker compose up -d
docker compose exec terria /app/node_modules/.bin/pm2 start /app/ecosystem-production.config.js --update-env --env production
```

(4) PLATEAU VIEW 1.1の停止

下記を実行し、PLATEAU VIEW 1.1を停止する。

```
docker compose down
```

2.4 データ変換の方法

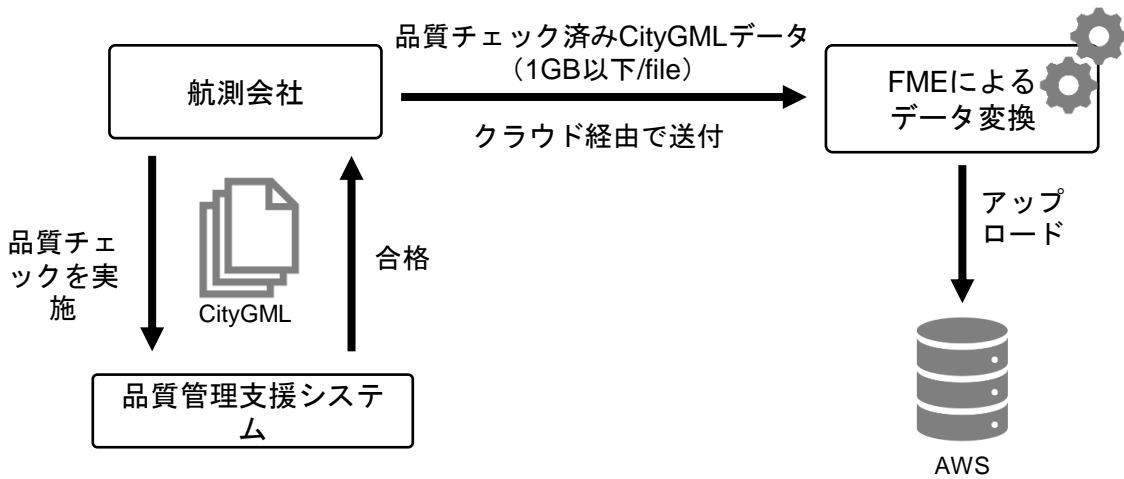
2.4.1 データ変換のワークフロー

PLATEAU VIEW 1.1におけるデータ変換のワークフローを説明する。

前述のとおり、PLATEAU VIEW 1.1では、CityGML形式の建物等の都市データをFMEを使用して3DTilesに変換し、Cesiumで表示している。データ変換に使用するFMEの変換ロジックは、CityGMLデータが「標準製品仕様書」に定められた要求品質を満足していることを前提として構築した。

データ変換の確実性を高めるため、Project PLATEAUでは、データ整備担当者が作成したCityGMLデータの品質評価を品質管理支援システムを用いて行い、合格したものをFMEにより3DTilesに変換するワークフローを構築した。

図 本事業でのデータ変換のワークフロー



地方自治体向け留意点

同じデータ形式でも内部構造が異なるデータが混在すると、FME変換における変換ロジックを一貫して適用できず、作業工程上の負担となる。このため、CityGMLを3DTilesに変換する場合等は、統一的な仕様（標準製品仕様）による品質チェックの過程をワークフローに取り入れる必要がある。

2.4.2 FMEによるデータ変換

FMEによるデータフォーマットの変換は、リーダーフィーチャータ입 (元データ) とライターフィーチャータ입 (変換後データ) の間、及びそれらの個々の属性の間の対応づけを定義し (マッピング)、ワークスペースを実行することで実現する。FMEは、CityGML、3DTiles等のデータ形式の変換ロジックのためのライブラリも有している。ユーザーは、ライブラリの中から、元データの形式と変換後データの形式を選択し、ワークスペースを実行する。

図 FMEによるデータ変換 (マッピング)

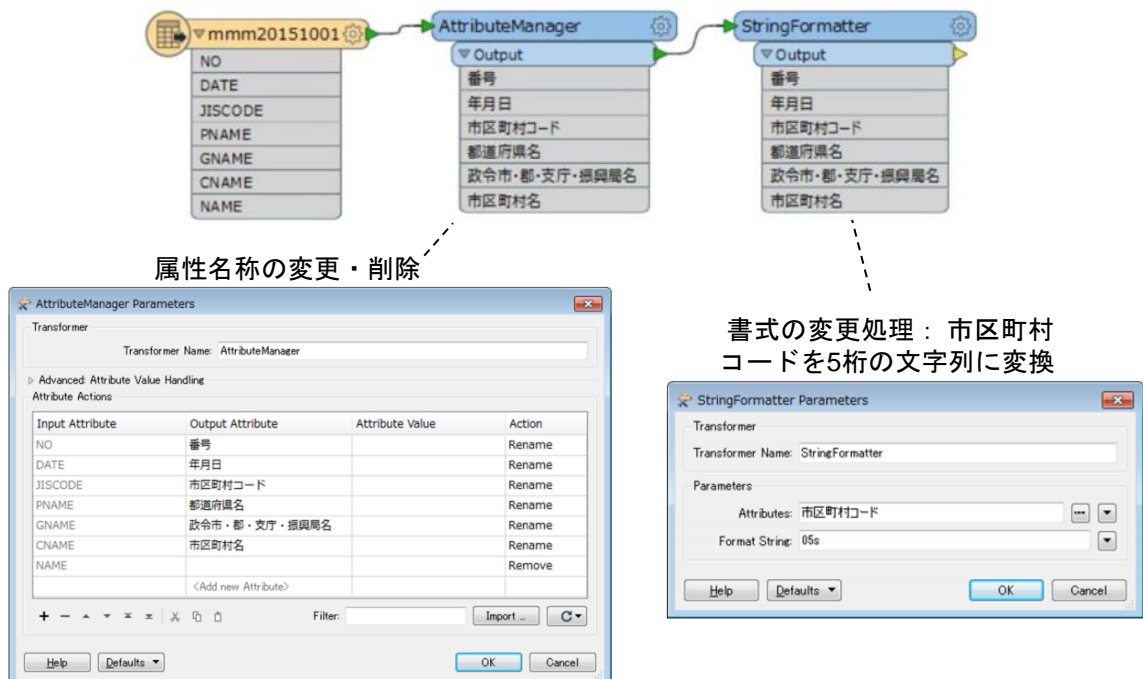


リーダー側とライター側の属性を直接マッピング可能な変換ではなく、データの読み込みから書き出しまでの間に何段階かのデータ変換を行う場合、トランスフォーマーを使用する。トランスフォーマーは、各属性のデータに対して、名称や書式等の変更処理をユーザーが記述できるものである。

なお、PLATEAU VIEW 1.1で用いたFMEの変換ロジックは以下のURLで公開している。

<https://www.geospatial.jp/ckan/dataset/plateau-tokyo23ku>

図 FMEによるデータ変換の例



2.5 カタログの編集

PLATEAU VIEW 1.1で表示するデータセットは、JSONで記述されたカタログファイルで定義する。本編に沿って構築されたPLATEAU VIEW 1.1には、公開版と同じ構成のカタログファイルが読み込まれているが（wwwroot/init/plateauview.json）、表示するデータセットの追加など、変更が必要な場合は、このカタログファイルの編集が必要となる。カタログファイルで設定できる内容は、データセットのタイトル（下図①）、説明文（下図②）、属性表、凡例の表示の他、一部のデータタイプについては、属性値による色の指定の他、地物の絞込の機能の指定なども行うことができる。



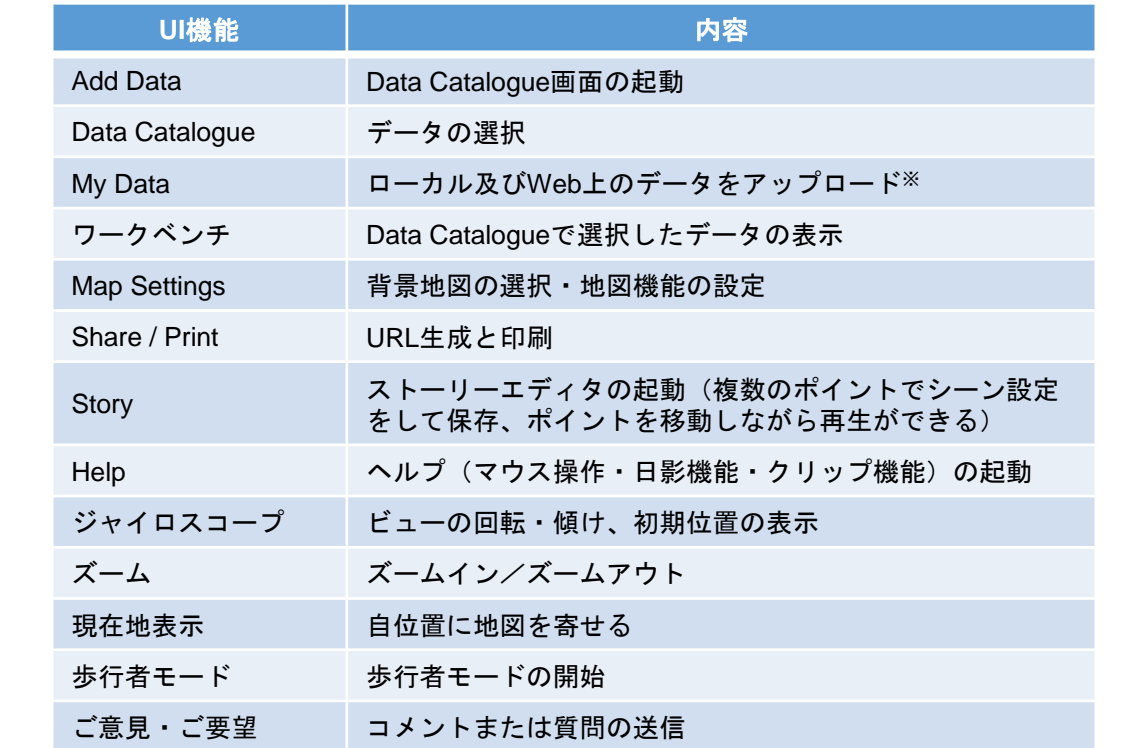
PLATEAU VIEW 1.1で使用しているカタログファイルを簡単に編集できるようにするためのツール、カタログファイルジェネレーターをProject-PLATEAUのGitHub（<https://github.com/Project-PLATEAU/plateau-catalog-generator>）で公開している。このツールを使えば、表示したいデータセットのタイトルや参照先を記入したCSVファイルから、PLATEAU VIEW 1.1で読み込み可能なJSONファイルを生成することができる。ツールの使用方法など詳細は、上記GitHubのチュートリアルを参考にする。

カタログファイルで設定できる項目は多岐にわたるため、カタログファイルジェネレーターのチュートリアルは、PLATEAU VIEW 1.1で使用しているファイルタイプと機能に絞った説明となっているが、網羅的な説明はTerriaJSのドキュメントが詳しい（<https://docs.terria.io/guide/customizing/initialization-files/>）。

第2編 PLATEAU VIEW 1.1

第3章 ユーザーマニュアル

基本的な利用方法として、地図選択、データの選択・表示、重畳に関する一連の基本操作について説明する。



PLATEAU VIEW 1.1では対応するフォーマットのデータについては、My Dataボタンを選択すると表示されるデータ登録画面にドラッグ＆ドロップすることでカタログに追加し、表示することができます。なお、このドラッグ＆ドロップによるデータの追加はPLATEAU VIEW 1.1のメインの画面上でも行える。

図 My Data登録画面（左：初期状態、右：データが登録された状態）



ドラッグ＆ドロップでの登録に対応したフォーマットとその概要は下表の通り。


表 対応フォーマット

対応フォーマット	説明
GeoJSON	GeoJSONはJSONで空間データを扱うためのフォーマットでポイント、ライン、ポリゴンを扱うことができる。 GeoJSON公式サイト： https://geojson.org/
KML/KMZ	KML/KMZは空間データをXMLで記述するフォーマットで、ポイント（マーカー）、ライン、ポリゴンを扱うことができる。 KML入門用ドキュメント： https://developers.google.com/kml/documentation/?hl=ja
CSV	CSVはテキストファイルのフォーマットの一つで、PLATEAU VIEW 1.1では緯度経度情報を持ったポイントデータとして取り込むことができる。読み込むためには少なくとも緯度、経度を表す列が必要で、それぞれの列名が緯度（latitude, lat）、経度（longitude, lon, lng）となっていれば自動的にポイントデータに変換される。
CZML	CZMLはJSON形式を基本とした、PLATEAU VIEW 1.1の基盤となっているCesiumで空間データを表現するのに適したフォーマットである。（ https://github.com/AnalyticalGraphicsInc/czml-writer/wiki/CZML-Structure ）
GPX	GPXはGPSのデータを扱うためのフォーマットで、軌跡（ライン）とウェイポイント（ポイント）を扱うことができる。 https://www.topografix.com/gpx.asp
Shape	ShapeファイルはGISデータ（2次元）の業界標準フォーマットで、shp、shx、dbfのファイルから構成される。ドラッグ＆ドロップで使用するためにはこれらのファイルをZIP圧縮する必要がある。 ESRIジャパンの解説ページ： https://www.esri.com/gis-guide/esri-dataformat/shapefile/

(1) 背景地図の選択・地図機能の設定

- 画面右上の[Map Settings]から、背景地図の選択と地図機能の設定ができる。
- [Map Settings]を押下すると、マップビュー、ベースマップの選択及び地図機能に関するチェックボックスが表示され、背景地図の選択と地図機能の設定ができる。

図 地図選択・地図機能設定画面 ([Map Settings])



マップビューは以下を選択可能

- 地図を3D地形で表示
- 地図を3D地形なしで表示
- 地図を2Dで表示

地下を隠す

- 地下のオブジェクトを表示しない

地下に入る

- 視点の移動時に地面の下に移動できるようにする

ベースマップは以下を選択可能

- 空中写真
- 空中写真 (Bing)
- 地理院地図
- Dark Matter

タイムライン

- 画面下部に表示した日の午前0時から午後12時までの24時間のタイムラインが表示される。日影表示機能と併用することで、日影を表現する時刻を任意に設定できる。なお、ワークベンチに時系列データが追加されている間は、この指定に関係なくタイムラインが表示される。

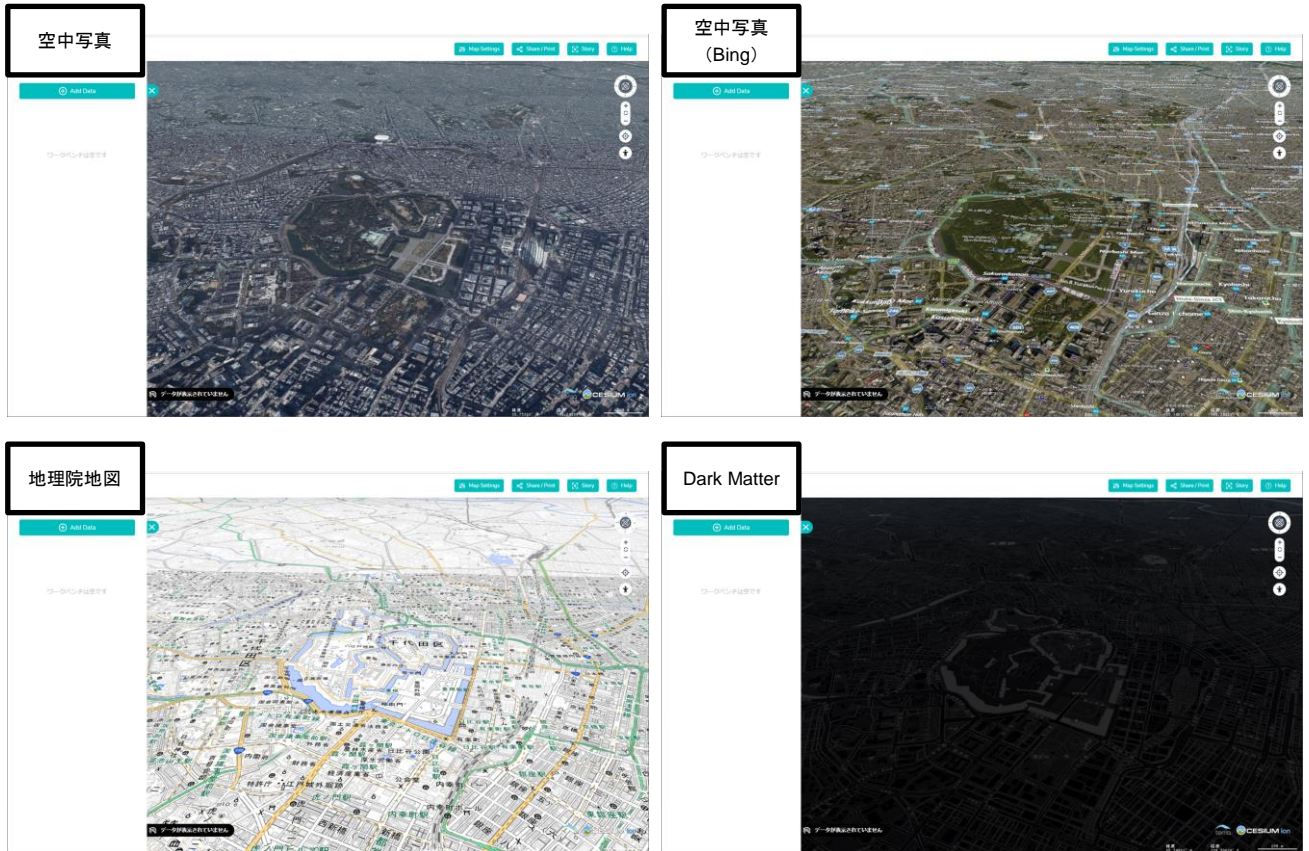
画像の最適化

- オンにすると、描画する解像度がお使いのデバイスの解像度に合わせて表現される。お使いのモニタの解像度が低い場合は、オンにすると描画の改善が期待される。

ラスターデータの質

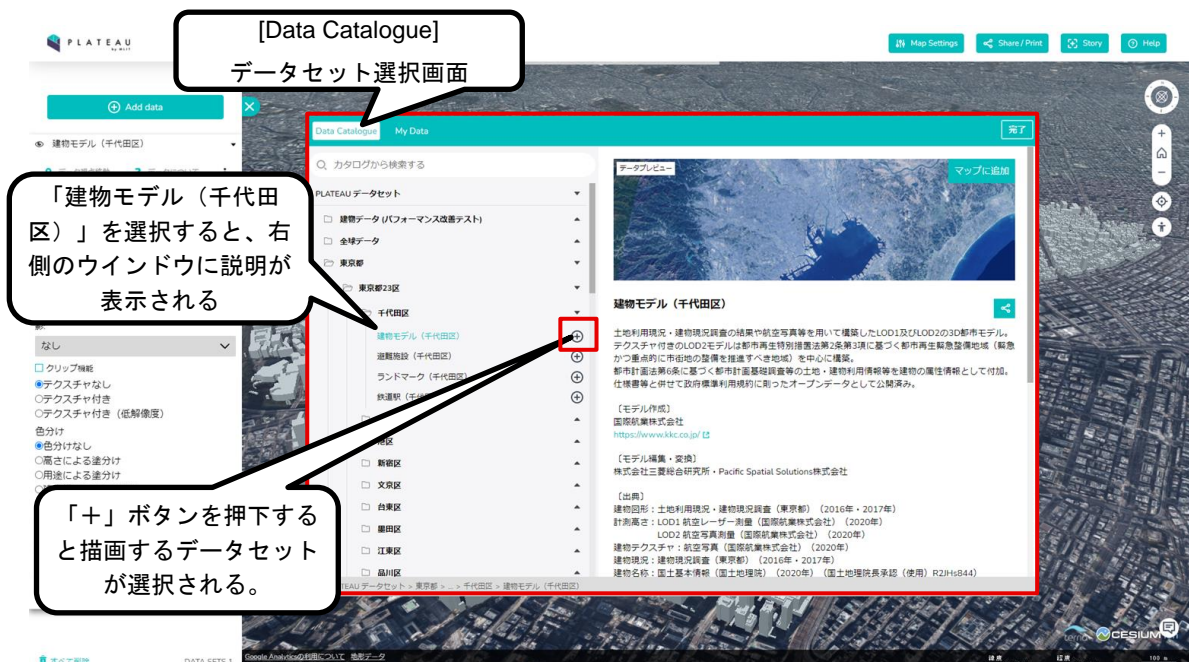
- 質の方にスライダを移動させると、ラスターデータ及びポイントクラウドデータで解像度が高くなるがマシンのスペックによっては描画速度が遅くなる。パフォーマンスの方にスライダを移動させると解像度が低くなるが描画速度は速くなる。

図 選択可能な地図



(2) データの選択・表示

- [Add Data]を押下すると、データセットの選択画面（[データカタログ]）が表示される。
- [データカタログ]においてデータ選択すると、右側のウィンドウに説明が表示される。[+]ボタンを押下することで、データを描画することができる。



- 選択したデータセットが描画領域に表示され、ワークベンチにはデータセットの情報が表示される。

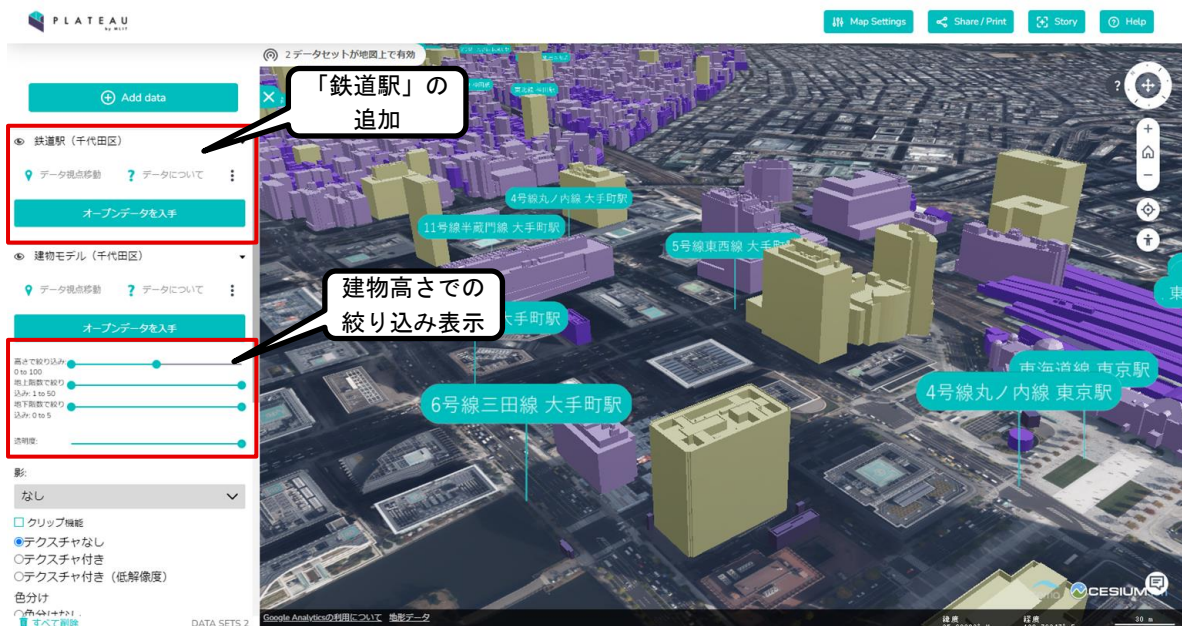


(3) データの重畳

- ・ [ワークベンチ]から重畳表示したいデータを選択する（下図では「高さによる塗分け」を選択）。
- ・ 描画領域の建物に対して、高さ別の色分けが重畳表示される。



- ・ [Add Data]から[データカタログ]を呼び出し、追加で別のデータセットを表示することも可能である（下図では「鉄道駅 (千代田区)」を追加）。
- ・ また、属性の条件を指定することで、描画対象を絞り込む等の機能も有している。
- ・ 下図は、建物高さ100m以下の建物のみを表示し、大手町駅付近を拡大した状態。



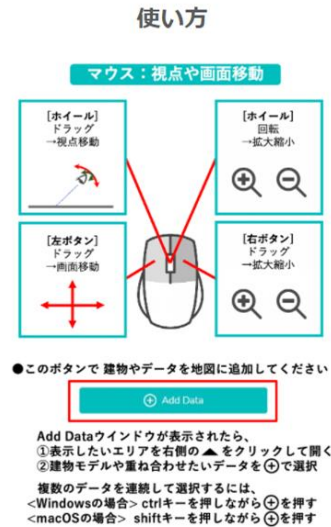
(4) 画面操作

- 描画領域の視点移動や拡大縮小等は、ボタン（画面右上）かマウスで操作することができる。なお、マウスの使い方は、画面右上の[Help]から呼び出すことができる。

図 操作ボタン（画面右上）



図 マウスの使い方（Help）



- また、地図上に描画されたオブジェクト（建物、駅など）をマウスでクリックすることで、属性情報が表示される。



実証環境構築マニュアル 第3.0版
3D City Model Demonstration Manual

令和5年3月 発行
国土交通省 都市局

