

# eコミマップ

## API リファレンス

Ver. 2.0

2016/08/01

国立研究開発法人 防災科学技術研究所



## 目次

1	e コミマップ API リクエスト仕様	1
1.1	リクエストの形式	1
1.1.1.	既定のリクエストパラメータ	1
1.2	登録・更新・削除用のトークン	2
1.3	レスポンスの形式	2
1.3.1.	レスポンスコード	2
1.3.2.	既定のレスポンス	3
1.3.3.	エラー時のレスポンス	3
1.4	認証方式	4
1.4.1.	OAuth 2.0 認証方式による認証	4
2	e コミマップ API	10
2.1	ユーザ情報 API	10
2.2	グループ情報 API	15
2.3	マップ情報 API	19
2.4	レイヤ情報 API	24
2.5	マップレイヤ情報 API	24
2.6	描画設定 API	35
2.7	描画設定アイコン API	43
2.8	登録情報 API	45
2.9	フィーチャファイル情報 API	52
2.10	コンテンツファイルアップロード API	55
2.11	アイコンファイルアップロード API	56
3	API フィルター	56
3.1	ユーザ認証フィルター	57
3.2	メソッドフィルター	57

# 1 e コミマップ API リクエスト仕様

## 1.1 リクエストの形式

APIのリクエストは、APIリクエストパスに対してメソッドとパラメータを指定して送信する。メソッドはHTTPメソッドとして指定（GET/POST/PUT/DELETE）する方法のほか、リクエストのパラメータ `_method` に同様の文字列を指定する方法がある。

パラメータを伴うリクエストおよびレスポンスの形式を以下に示す。

### GET リクエスト（情報取得で利用）

リクエスト： `GET /path/to/process/value1/?param1=option1&param2=option2`

レスポンス： 情報 JSON形式またはXML形式（複数または単一）

### POST リクエスト（新規登録で利用）

リクエスト： `POST /path/to/process/value1/`

POSTデータ：XMLやJSON等の文字列

ファイルのアップロードデータもPOSTで送信する

レスポンス： 登録処理結果 JSON文字列

### PUT リクエスト（更新で利用）

リクエスト： `PUT /path/to/process/value1/`

PUTデータ：XMLやJSON等の文字列

レスポンス： 更新処理結果 JSON文字列

### DELETE リクエスト（削除で利用）

リクエスト： `DELETE /path/to/process/value1/?param1=option1&param2=option2`

レスポンス： 削除処理結果 JSON文字列

#### 1.1.1. 既定のリクエストパラメータ

<code>_method</code>	PUT、DELETEメソッドに対応しないクライアントプログラムからのリクエストでは、リクエストパラメータによりメソッドを指定する。POST、GETリクエストに <code>_method</code> で指定することで対応するメソッドの処理が実行される。
<code>_timestamp</code>	PUT、DELETEメソッドを送信する際に、このパラメータか <code>_force</code> のどちらかを必ず設定する必要がある。データ取得時刻の整数値を設定する。

_force	PUT、DELETE メソッドを送信する際に指定することができる。 _timestamp を無視して情報を更新する。 衝突時も強制的に更新されるため通常は使用しない。
sort	GET メソッドで複数の処理結果を返す場合に、ソートを行うカラム ID。先頭に-をつけた場合は降順とする
limit	GET メソッドで複数の処理結果を返す場合に、返却する件数。
offset	GET メソッドで複数の処理結果を返す場合に、offset 件目の結果以降を返却する。この値が全結果数よりも大きくなる場合はエラーを返す。

## 1.2 登録・更新・削除用のトークン

登録更新削除を行う場合は、トークンとしてセッション ID を設定する必要がある。

HTTP ヘッダに設定する場合：

X-CSRF-Token
--------------

設定例(doj):

<pre>dojo.xhrGet({   url: url,   headers: {"X-CSRF-Token":dojo.cookie("JSESSIONID")},   handleAs: "text",   handle: function(data, args) { ~以下省略~</pre>
---

POST のフォームに hidden で設定する場合：

<input type="hidden" name="session_token" value="セッションID"/>
---

## 1.3 レスポンスの形式

出力形式は HTTP レスポンスとして取得される。実行結果を示すレスポンスコードを含む HTTP ヘッダが出力される。

### 1.3.1 レスポンスコード

200 OK	成功
400 Bad Request	リクエストパラメータが不正である
403 Forbidden	閲覧、登録、更新、削除の権限がない処理を実行した
404 Not Found	存在しない API のパスの実行、または指定したリソースが見えない

409 Conflict	更新時に衝突が検知された
500 Internal Server Error	そのほかの理由で処理が失敗した

### 1.3.2. 既定のレスポンス

timestamp	データ取得時はサーバ時間の整数値が設定される。 PUT と DELETE リクエスト時に、パラメータ_timestamp に設定することで衝突の検知を行うことができる。
result	処理結果の JSON オブジェクトを格納する
results	複数の値を取得する GET メソッドの結果を JSON オブジェクトの配列として格納する
total	複数の値を取得する場合は、データの総数が格納される

### 1.3.3. エラー時のレスポンス

error	エラーメッセージ
result	エラーの詳細が必要な場合は JSON オブジェクトを格納
results	エラーの詳細が複数必要な場合は JSON 配列を格納

## 1.4 認証方式

API 処理を任意のユーザーの権限で実行するために、以下のユーザー認証方式に対応する。なお、下記方式による認証パラメーターを付加しない場合はゲストユーザーとしての権限で動作する。

### 1.4.1. OAuth 2.0 認証方式による認証

#### 1.4.1.1. はじめに

OAuth2.0 に準拠した認証に対応しており、一般的な OAuth2.0 認証ライブラリを使って認証を行うことができる。

ただし、OAuth2.0 は接続プロトコルに HTTPS を利用することを前提としているため、OAuth 2.0 を利用する場合は OAuth 認証処理、API 処理では HTTPS で通信できるサーバ環境を構築すること。

OAuth によるユーザ認証で API を利用するまでの流れ

1. OAuth 認証するための事前設定として
  - 「1.4.1.2 OAuth クライアントアプリケーション登録」を行う。
2. 「1.4.1.3 アクセストークンの取得」でユーザの承諾後アクセストークンを取得する。
3. 「1.4.1.4

OAuth 認証情報のリクエスト」の形式で  
アクセストークンを付加して API にリクエストする。

#### 1.4.1.2. OAuth クライアントアプリケーション登録

OAuth 認証を受け付けてよいか確認のために OAuth クライアント認証が行われる。  
その OAuth クライアント認証に必要なキーは、下記のセクションで生成できる。

「サーバ設定」画面 「アプリケーション OAuth キーの設定」セクション
Consumer Key : アプリケーション識別用
Consumer Secret Key : アプリケーション秘密鍵

このキー(とくに Consumer Secret Key)は認証連携を行うアプリケーション開発者が保管し、外部に漏れないよう配慮する必要がある。万が一、Consumer Key と Consumer Secret Key が漏れた場合は、アプリケーションを偽装される恐れがあるため、同画面からキーの更新を行うこと。

#### 1.4.1.3. アクセストークンの取得

下記のいずれかの認証フローを利用することで、API リクエストで使用できるアクセストークン(有効期限は無期限)を取得することができる。連携アプリケーションにあった認証フローを選択すること。

IPA では「Authorization Code グラント種別」が最も無難な認証フローといわれている。

参考情報 : [IPA セキュリティトークンと OAuth 2.0](#)

OAuth2.0 に準拠した認証フローのため、一般的な OAuth 認証ライブラリを使って認証することができる。

OAuth 認証に使用する URL は以下のものである。

Authorization URL	oauth2/authorize
Token URL	oauth2/token

連携アプリが認可コードやアクセストークンを受け取るためのリダイレクト先を指定する場合でも、接続プロトコルに HTTPS を用いて漏洩しないように配慮する必要がある。

#### ● Authorization Code グラント種別

連携アプリケーションは WEB アプリケーションを想定している。

##### 1. 連携アプリがユーザに

「Authorization URL+Consumer Key」にアクセスするように促す。

リダイレクト先には連携アプリが認可コードを受け取るための URL を指定する。サイト ID(オプション)は e コミマップのサイトを指定する。



Authorization URL 例: GET

https://**サーバ名**/map/oauth2/authorize?cid=**サイト ID**

&response\_type=code&client\_id=**Consumer Key**&redirect\_uri=**リダイレクト先**

2. ユーザが認証画面で連携アプリを許可すると認可コードが発行される。  
※ログインしていない場合はログイン画面が表示される。  
※Authorization URL の cid パラメータで指定したサイトのログイン画面となる。  
cid パラメータを省略した場合はデフォルトで cid=1 となる。



認証画面

※ログイン画面、認証画面は Authorization URL で指定したサイトが表示される。

リダイレクト例: GET

リダイレクト先?code=**認可コード**

3. 連携アプリが「Token URL」に  
OAuth クライアント認証用「Consumer Key+Consumer Secret Key」と  
連携アプリがアクセストークンを受け取るためのリダイレクト先(必須)と  
認可コードを付加してリクエストするとアクセストークンが取得できる。

連携アプリからのリクエスト例: POST

```
curl -v -L -X POST ¥
```

```
-d "grant_type=authorization_code" ¥
```

```
-d "code=認可コード" ¥
```

```
-d "client_id=Consumer Key" -d "client_secret=Consumer Secret Key" ¥
```

```
-d "redirect_uri=リダイレクト先" ¥
```

```
https://サーバ名 /map/oauth2/token
```

レスポンス結果: JSON

```
{"access_token": "アクセストークン"}
```

- Implicit グラント種別

連携アプリケーションは Consumer Secret Key を隠すことのできない

JavaScriptにより連携するWEBアプリケーションを想定している。

1. 連携アプリがユーザに

「Authorization URL+Consumer Key」と  
連携アプリがアクセストークンを受け取るためのリダイレクト先(必須)を指定した  
URLにアクセスするように促す。  
サイトID(オプション)はeコママップのサイトを指定する。

Authorization URL 例: GET  
https://**サーバ名**/map/oauth2/authorize?cid=**サイトID**  
&response\_type=token&client\_id=**Consumer Key**&redirect\_uri=**リダイレクト先**

2. ユーザが認証画面で連携アプリを許可するとアクセストークンが発行される。

※ログインしていない場合はログイン画面が表示される。



認証画面

※ログイン画面、認証画面は Authorization URL で指定したサイトが表示される。

リダイレクト例: GET  
リダイレクト先#access\_token=**アクセストークン**

● Resource Owner Password Credentials グラント種別

連携アプリケーションはブラウザを用いないアプリケーションプログラムが想定される。

1. 連携アプリがユーザIDとパスワードをユーザに求める。

※得られるアクセストークンは有効期限が無期限のため、  
連携アプリがユーザのIDとパスワードを保持しておく必要はない。

2. 連携アプリが「Token URL」に

OAuth クライアント認証用「Consumer Key+Consumer Secret Key」と  
ユーザの「ID+パスワード」を付加してリクエストすると  
アクセストークンが取得できる。

連携アプリからのリクエスト例: POST  
curl -v -L -X POST ¥  
-d "grant\_type=password" ¥

```
-d "username=ユーザ ID" -d "password=パスワード" ¥  
-d "client_id=Consumer Key" -d "client_secret=Consumer Secret Key" ¥  
https://サーバ名 /map/oauth2/token
```

レスポンス結果: JSON

```
{"access_token":"アクセストークン"}
```

- Client Credentials グラント種別

認証していない状態だとゲストユーザとしての権限で動作するため、この認証フローはサポートしていない。

#### 1.4.1.4. OAuth 認証情報のリクエスト

- リクエストヘッダに付加する方法

[RFC6750](#) で規定されている「Bearer Token Usage」の形式でリクエストする。

リクエストヘッダの **Authorization** ヘッダに下記のような形式でアクセストークンを指定する。

Authorization: Bearer アクセストークン
--------------------------------

※アクセストークンを生で指定するため HTTPS 通信が前提

- リクエストパラメータに付加する方法

リクエストパラメータの `oauth_token` にアクセストークンを指定する。

## 2 e コミマップ API

### 2.1 ユーザ情報 API

リソースクラス

jp.ecom-plat.map.rest.resources.UserInfoResource.java

#### GET

ユーザ情報を取得する

リクエストパス

/rest/user/[cid]/[groupId]/[userId]

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
groupId	必須	グループ ID 0 ならサイト全体
userId	必須	ユーザ ID

レスポンス (コード: 200)

ユーザ情報の JSON 形式データ

ID	データ型	説明
userId	整数	ユーザ ID
authId	文字列	ログイン時に入力する ID
name	文字列	
level	整数	サイト内の権限レベル
email	文字列	
mobile	文字列	

レスポンスエンティティクラス

jp.ecom-plat.map.rest.entity.UserInfoEntity.java

エラー

404	ユーザが存在しない
-----	-----------

#### GET

ユーザの一覧を取得する

#### リクエストパス

/rest/user/[cid]/[groupId]
----------------------------

#### リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
groupId	必須	グループ ID 0 ならサイト全体

#### レスポンス (コード:200)

ユーザ情報の JSON 配列形式データ

ID	データ型	説明
total	整数	結果の総数
results	配列	ユーザ情報の JSON 形式データの配列

ユーザ情報の JSON 形式データ

ID	データ型	説明
userId	整数	ユーザ ID
authId	文字列	ログイン時に入力する ID
name	文字列	
level	整数	サイト内の権限レベル
email	文字列	
mobile	文字列	

レスポンスエンティティクラス

jp.ecom-plat.map.rest.entity.UserInfoEntityList.java

#### エラー

404	グループが存在しない
-----	------------

## POST

ユーザを新規に登録する

#### リクエストパス

/rest/user/[cid]/[groupId]
----------------------------

### リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
groupId	必須	グループ ID 0 ならサイト全体

### リクエストデータ

ユーザ情報の JSON 形式データ

ID	データ型	説明
authId	文字列	ログイン時に入力する ID
name	文字列	
level	整数	サイト内の権限レベル
email	文字列	
mobile	文字列	

リクエスト例：

```
["authId":"user1","email":"user1.sample@sample.com","level":10,"mobile":"user1.mobile@sample.com", "name":"ユーザ"]
```

### レスポンス (コード : 200)

ID	データ型	説明
timestamp	整数	データ取得サーバ時間
result	JSON	JSON 形式データ
userId	整数	追加されたユーザの ID

レスポンス例：

```
["timestamp":12345678, "result":{"userId":1}]
```

レスポンスエンティティクラス

jp.ecom.plat.map.rest.entity.UserInfoEntity.java

### エラー

404	グループが存在しない
409	登録時に衝突を検知した

## PUT

ユーザを更新する

リクエストパス

/rest/user/[cid]/[groupId]/[userId]

#### リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
groupId	必須	グループ ID 0 ならサイト全体
userId	必須	ユーザ ID

#### リクエストデータ

ユーザ情報の JSON 形式データ

ID	データ型	説明
authId	文字列	ログイン時に入力する ID
name	文字列	
level	整数	サイト内の権限レベル
email	文字列	
mobile	文字列	

#### リクエスト例 :

```
{"authId": "user1", "email": "user1.sample@sample.com", "level": 10, "mobile": "user1.mobile@sample.com", "name": "ユーザ"}
```

#### レスポンス (コード : 200)

レスポンスデータなし

#### エラー

404	ユーザが存在しない
409	更新時に衝突を検知した

## DELETE

ユーザを削除する

#### リクエストパス

/rest/user/[cid]/[groupId]/[userId]

#### リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID



groupId	必須	グループ ID 0 ならサイト全体
userid	必須	ユーザ ID

レスポンス (コード : 200)

レスポンスデータなし

エラー

404	ユーザが存在しない
409	削除時に衝突を検知した

## 2.2 グループ情報 API

リソースクラス

jp.ecom-plat.map.rest.resources.GroupInfoResource.java

### GET

グループ情報を取得する

リクエストパス

/rest/group/[cid]/[groupId]

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
groupId	必須	グループ ID

レスポンス (コード: 200)

グループ情報の JSON 形式データ

ID	データ型	説明
groupId	整数	グループ ID
name	文字列	グループ名
status	整数	グループの有効・無効の状態

レスポンスエンティティクラス

jp.ecom-plat.map.rest.entity.GroupInfoEntity.java

エラー

404	グループが存在しない
-----	------------

### GET

グループの一覧を取得する

リクエストパス

/rest/group/[cid]

### リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID

### レスポンス (コード:200)

グループ情報の JSON 配列形式データ

ID	データ型	説明
total	整数	結果の総数
results	配列	ユーザ情報の JSON 形式データの配列

グループ情報の JSON 形式データ

ID	データ型	説明
groupId	整数	グループ ID
name	文字列	グループ名
status	整数	グループの有効・無効の状態

レスポンスエンティティクラス

jp.ecom-plat.map.rest.entity.GroupInfoEntityList.java

### エラー

404	グループが存在しない
-----	------------

## POST

グループを新規に登録する

### リクエストパス

/rest/user/[cid]/[groupId]
----------------------------

### リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID

### リクエストデータ

グループ情報の JSON 形式データ

ID	データ型	説明
name	文字列	グループ名
status	整数	グループの有効・無効の状態

リクエスト例：

```
["name":"デモ用グループページ","status":1]
```

レスポンス (コード : 200)

ID	データ型	説明
timestamp	整数	データ取得サーバ時間
result	JSON	JSON 形式データ
groupId	整数	追加されたグループの ID

レスポンス例：

```
{"timestamp":12345678, "result":{"groupId":1}}
```

レスポンスエンティティクラス

jp.ecom.plat.map.rest.entity.GroupInfoEntity.java

エラー

404	グループが存在しない
409	登録時に衝突を検知した

## PUT

グループを更新する

リクエストパス

```
/rest/group/[cid]/[groupId]
```

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
groupId	必須	グループ ID

リクエストデータ

グループ情報の JSON 形式データ

ID	データ型	説明
name	文字列	グループ名
status	整数	グループの有効・無効の状態

リクエスト例：

```
{"name":"デモ用グループページ","status":1}
```

レスポンス (コード:200)

レスポンスデータなし

エラー

404	グループが存在しない
409	更新時に衝突を検知した

## DELETE

グループを削除する

リクエストパス

/rest/group/[cid]/[groupId]

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
groupId	必須	グループ ID

レスポンス (コード:200)

レスポンスデータなし

エラー

404	グループが存在しない
409	削除時に衝突を検知した

## 2.3 マップ情報 API

リソースクラス

jp.ecom-plat.map.rest.resources.MapInfoResource.java

### GET

マップ情報を取得する

リクエストパス

```
/rest/map/[cid]/[groupId]/[mapId]
```

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
groupId	必須	グループ ID 0 ならサイト全体
mapId	必須	マップ ID

レスポンス (コード:200)

マップ情報の JSON 形式データ

ID	データ型	説明
mapId	整数	マップ ID
title	文字列	マップ名
description	文字列	説明
extent	JSON 配列	範囲 左、下、右、左の緯度経度度単位の JSON 配列
status	整数	マップ情報種別 0:地図作成途中,1:編集中,10:公開中,100:確定,-100:削除済み
type	整数	マップ情報状態 0:通常,10:テンプレートレイヤ格納用
date	文字列	マップ情報作成日時
modified	文字列	マップ情報更新日時

レスポンスエンティティクラス

jp.ecom-plat.map.rest.entity.MapInfoEntity.java

エラー

404	マップが存在しない
-----	-----------

## GET

マップの一覧を取得する

リクエストパス

/rest/user/[cid]/[groupId]
----------------------------

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
groupId	必須	グループ ID 0 ならサイト全体

レスポンス (コード:200)

マップ情報の JSON 配列形式データ

ID	データ型	説明
total	整数	結果の総数
results	配列	マップ情報の JSON 形式データの配列

マップ情報の JSON 形式データ

ID	データ型	説明
mapId	整数	マップ ID
title	文字列	マップ名
description	文字列	説明
extent	JSON 配列	範囲 左、下、右、左の緯度経度度単位の JSON 配列
status	整数	マップ情報種別 0:地図作成途中,1:編集中,10:公開中,100:確定,-100:削除済み
type	整数	マップ情報状態 0:通常,10:テンプレートレイヤ格納用
date	文字列	マップ情報作成日時
modified	文字列	マップ情報更新日時

レスポンスエンティティクラス

jp.ecom-plat.map.rest.entity.MapInfoEntityList.java

## エラー

404	グループが存在しない
-----	------------

## POST

マップを新規に登録する

### リクエストパス

/rest/map/[cid]/[groupId]
---------------------------

### リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
groupId	必須	グループ ID 0 ならサイト全体

### リクエストデータ

マップ情報の JSON 形式データ

ID	データ型	説明
title	文字列	マップ名
description	文字列	説明
extent	JSON 配列	範囲 左、下、右、左の緯度経度度単位の JSON 配列

### リクエスト例 :

{"description": "地図の説明", "extent": [120.199, 22.93, 151.35, 46.7799], "title": "マップ1"}
--

### レスポンス (コード : 200)

ID	データ型	説明
timestamp	整数	データ取得サーバ時間
result	JSON	JSON 形式データ
mapId	整数	追加されたマップの ID

### レスポンス例 :

{"timestamp": 12345678, "result": {"mapId": 1}}
---

### レスポンスエンティティクラス

jp.ecom.plat.map.rest.entity.MapInfoResult.java

## エラー

404	グループが存在しない
-----	------------



409	登録時に衝突を検知した
-----	-------------

## PUT

マップを更新する

リクエストパス

```
/rest/map/[cid]/[groupId]/[mapId]
```

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
groupId	必須	グループ ID 0 ならサイト全体
mapId	必須	マップ ID

リクエストデータ

マップ情報の JSON 形式データ

ID	データ型	説明
title	文字列	マップ名
description	文字列	説明
extent	JSON 配列	範囲 左、下、右、左の緯度経度度単位の JSON 配列
status	整数	マップ情報種別 0:地図作成途中,1:編集中,10:公開中,100:確定,-100:削除済み
type	整数	マップ情報状態 0:通常,10:テンプレートレイヤ格納用

リクエスト例 :

```
{"description": "地図の説明", "extent": [120. 199, 22. 93, 151. 35, 46. 7799], "title": "マップ 1", "status": 1, "type": 0}
```

レスポンス (コード : 200)

レスポンスデータなし

エラー

404	マップが存在しない
409	更新時に衝突を検知した

## DELETE

マップを削除する

リクエストパス

```
/rest/map/[cid]/[groupId]/[mapId]
```

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
groupId	必須	グループ ID 0 ならサイト全体
mapId	必須	マップ ID

レスポンス (コード:200)

レスポンスデータなし

エラー

404	マップが存在しない
409	削除時に衝突を検知した

## 2.4 マップレイヤ情報 API

リソースクラス

jp.ecom-plat.map.rest.resources.MapLayerInfoResource.java

### GET

マップ内のレイヤ情報一覧を取得する

リクエストパス

```
/rest/maplayer/[cid]/[groupId]/[mapId]
```

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
groupId	必須	グループ ID 0 ならサイト全体
mapId	必須	地図 ID

レスポンス (コード:200)

レイヤ情報の JSON 配列形式データ

ID	データ型	説明
total	整数	結果の総数
results	配列	レイヤ情報の JSON 形式データの配列

レイヤ情報の JSON 形式データ

ID	データ型	説明
layerId	文字列	レイヤ ID
name	文字列	レイヤ名称
parent	文字列	親レイヤ ID
visible	倫理	初期表示状態
radio	文字列	Radio ボタンで排他表示するグループ名
closed	倫理	初期展開状態
opacity	数値	初期不透明度
scale	整数	レイヤ標準表示縮尺分母
maxScale	整数	レイヤ最大表示縮尺分母
attrNameId	文字列	名前として仕様する属性
searchType	整数	地図からの検索種別

		-1:検索しない,1:クリック時に検索,255:常に検索
reload	JSON 形式の文字列	主題図項目の再読み込み設定
created	文字列	属性作成日時
modified	文字列	属性更新日時

レスポンスエンティティクラス

jp.ecom.plat.map.rest.entity.MapLayerInfoEntityList.java

エラー

404	マップが存在しない
-----	-----------

## POST

マップにレイヤを追加する

リクエストパス

/rest/maplayer/[cid]/[groupId]/[mapId]
--

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
groupId	必須	グループ ID 0 ならサイト全体
mapId	必須	マップ ID

リクエストデータ

マップレイヤ情報の JSON 形式データ

ID	データ型	説明
layerId	文字列	レイヤ ID
name	文字列	レイヤ名称
parent	文字列	親レイヤ ID
visible	倫理	初期表示状態
radio	文字列	Radio ボタンで排他表示するグループ名
closed	倫理	初期展開状態
opacity	数値	初期不透明度
scale	整数	レイヤ標準表示縮尺分母

maxScale	整数	レイヤ最大表示縮尺分母
attrNameId	文字列	名前として仕様する属性
searchType	整数	地図からの検索種別 -1:検索しない,1:クリック時に検索,255:常に検索
reload	JSON 形式の文字列	主題図項目の再読み込み設定 {¥"caps¥":true,¥"sec¥":30} caps:true,false 再読み込み時にレイヤ情報を更新 sec:整数 再読み込み間隔 (秒)

リクエスト例 :

```
{ "layerId": "c1", "maxScale": 0, "name": "名称", "scale": 0, "visible": false, "closed": false, "opacity": 0.5, "attrNameId": "attr1" }
```

レスポンス (コード : 200)

ID	データ型	説明
timestamp	整数	データ取得サーバ時間
result	JSON	JSON 形式データ
layerId	整数	追加されたレイヤの ID

レスポンス例 :

```
{ "timestamp": 12345678, "result": { "layerId": "c1" } }
```

レスポンスエンティティクラス

jp.ecom-plat.map.rest.entity.LayerInfoResult.java

エラー

404	マップが存在しない
409	登録時に衝突を検知した

## DELETE

マップからレイヤを除外する

リクエストパス

```
/rest/maplayer/{cid}/{groupId}/{mapId}/{layerId}
```

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID

groupId	必須	グループ ID 0 ならサイト全体
mapId	必須	マップ ID
layerId	必須	レイヤ ID

**レスポンス (コード : 200)**

レスポンスデータなし

**エラー**

404	マップが存在しない レイヤが存在しない
409	削除時に衝突を検知した

## 2.5 レイヤ情報 API

### リソースクラス

jp.ecom-plat.map.rest.resources.LayerInfoResource.java

#### GET

レイヤ情報を取得する

#### リクエストパス

```
/rest/layer/[cid]/[groupId]/[layerId]
```

#### リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
groupId	必須	グループ ID 0 ならサイト全体
layerId	必須	レイヤ ID

#### レスポンス (コード:200)

レイヤ情報の JSON 形式データ

ID	データ型	説明
layerId	文字列	レイヤ ID
name	文字列	レイヤ名称
description	文字列	レイヤの説明
status	整数	レイヤ編集状態 -1:削除済み,0:編集中(未公開),1:編集中(公開),50:形状編集不可,70:検索非表示,100:編集不可
type	整数	レイヤ種別 1:登録情報項目, 59:外部参照地図の WMS のレイヤを子に持つレイヤ,100:WMS レイヤ,1100:WMS ベースレイヤ
attribution	文字列	レイヤの著作権者情報
geometryType	文字列	登録情報項目レイヤ登録時のジオメトリ型指定 POINT:点,LINestring:線, POLYGON:面, MULTIPPOINT:点(複数), MULTILINestring 線(複数), MULTIPOLYGON:面(複数)

featuretypeId	文字列	WMS リクエスト用の ID
wfsURL	文字列	WFS 検索用 URL
wmsCapabilitiesURL	文字列	WMS Capabilities 用 URL
wmsURL	文字列	WMS 地図画像取得 URL
wmsFormat	文字列	WMS 地図を取得する画像フォーマット image/png image/gif image/jpeg WMS ベースレイヤのみ利用可
wmsLegendURL	文字列	WMS 凡例画像取得 URL
wmsFeatureURL	文字列	WMS 地図情報取得 URL
scale	整数	最小表示縮尺
maxScale	整数	最大表示縮尺
created	文字列	レイヤの作成日時
modified	文字列	レイヤの更新日時
parent	文字列	親レイヤ ID
visible	倫理	初期表示状態
timeSeriesType	整数	時系列種別 0:時系列無し,10:時点,11:期間,20:履歴管理

レスポンスエンティティクラス

jp.ecom-plat.map.rest.entity.LayerInfoEntity.java

エラー

404	レイヤが存在しない
-----	-----------

## GET

サイト内またはグループ内のレイヤの一覧を取得する

リクエストパス

/rest/layer/[cid]/[groupId]
-----------------------------

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
groupId	必須	グループ ID 0 ならサイト全体



レスポンス (コード:200)

レイヤ情報の JSON 配列形式データ

ID	データ型	説明
total	整数	結果の総数
results	配列	レイヤ情報の JSON 形式データの配列

レイヤ情報の JSON 形式データ

ID	データ型	説明
layerId	文字列	レイヤ ID
name	文字列	レイヤ名称
description	文字列	レイヤの説明
status	整数	レイヤ編集状態 -1:削除済み,0:編集中(未公開),1:編集中(公開),50:形状編集不可,70:検索非表示,100:編集不可
type	整数	レイヤ種別 1:登録情報項目, 59:外部参照地図の WMS のレイヤを子に持つレイヤ,100:WMS レイヤ,1100:WMS ベースレイヤ
attribution	文字列	レイヤの著作権者情報
geometryType	文字列	登録情報項目レイヤ登録時のジオメトリ型指定 POINT:点,LINESTRING:線, POLYGON:面, MULTIPPOINT:点(複数), MULTILINESTRING 線(複数), MULTIPOLYGON:面(複数)
featuretypeId	文字列	WMS リクエスト用の ID
wfsURL	文字列	WFS 検索用 URL
wmsCapabilitiesURL	文字列	WMS Capabilities 用 URL
wmsURL	文字列	WMS 地図画像取得 URL
wmsFormat	文字列	WMS 地図を取得する画像フォーマット image/png image/gif image/jpeg WMS ベースレイヤのみ利用可
wmsLegendURL	文字列	WMS 凡例画像取得 URL
wmsFeatureURL	文字列	WMS 地図情報取得 URL
scale	整数	最小表示縮尺
maxScale	整数	最大表示縮尺

created	文字列	レイヤの作成日時
modified	文字列	レイヤの更新日時
parent	文字列	親レイヤ ID
visible	倫理	初期表示状態
timeSeriesType	整数	時系列種別 0: 時系列無し,10:時点,11:期間,20:履歴管理

レスポンスエンティティクラス

jp.ecom-plat.map.rest.entity.LayerInfoEntityList.java

エラー

404	グループが存在しない
-----	------------

## POST

レイヤを新規に登録する

リクエストパス

/rest/layer/[cid]/[groupId]
-----------------------------

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
groupId	必須	グループ ID 0 ならサイト全体

リクエストデータ

レイヤ情報の JSON 形式データ

ID	データ型	説明
name	文字列	レイヤ名称
description	文字列	レイヤの説明
status	整数	レイヤ編集状態 -1:削除済み,0:編集中(未公開),1:編集中(公開),50:形状編集不可,70:検索非表示,100:編集不可
type	整数	レイヤ種別 1:登録情報項目, 51: 外部 WMS の参照用コンテンツレイヤ,59: 外部 WMS の参照レイヤを子に持つレイヤ,100:WMS レイヤ,1100:WMS ベースレイヤ

attribution	文字列	レイヤの著作権者情報
geometryType	文字列	登録情報項目レイヤ登録時のジオメトリ型指定 POINT:点,LINestring:線, POLYGON:面, MULTIPPOINT:点(複数), MULTILINestring 線(複数), MULTIPOLYGON:面(複数)
featuretypeId	文字列	WMS リクエスト用の ID
wfsURL	文字列	WFS 検索用 URL
wmsCapabilitiesURL	文字列	WMS Capabilities 用 URL
wmsURL	文字列	WMS 地図画像取得 URL
wmsFormat	文字列	WMS 地図を取得する画像フォーマット image/png image/gif image/jpeg WMS ベースレイヤのみ利用可
wmsLegendURL	文字列	WMS 凡例画像取得 URL
wmsFeatureURL	文字列	WMS 地図情報取得 URL
scale	整数	最小表示縮尺
maxScale	整数	最大表示縮尺
parent	文字列	親レイヤ ID
visible	論理	初期表示状態

登録情報項目リクエスト例：

```
{"attribution":"","description":"","geometryType":"MULTIPOLYGON","maxScale":0,"name":"名称","scale":0,"status":1,"type":1,"visible":true}
```

主題図項目リクエスト例：

```
{"attribution":"","description":"","maxScale":0,"name":"WMSマップ","scale":0,"status":0,"type":59,"visible":false,"wmscapabilitiesurl":"http://wmssample.com/wms?SERVICE=wms&&REQUEST=GetCapabilities&VERSION=1.1.1","wmsformat":"image/png","featuretypeId":"layer_1,layer_2","wmslegendurl":"http://wmssample.com/map/wms?&SERVICE=WMS&","wmsURL":"http://wmssample.com/map/wms?&SERVICE=WMS&"}
```

レスポンス (コード : 200)

ID	データ型	説明
timestamp	整数	データ取得サーバ時間
result	JSON	JSON 形式データ
layerId	文字列	追加されたレイヤの ID

レスポンス例：

```
{"timestamp":12345678,"result":{"layerId":ref1}}
```

レスポンスエンティティクラス

jp.ecom-plat.map.rest.entity.LayerInfoResult.java

#### エラー

404	グループが存在しない
409	登録時に衝突を検知した

## PUT

レイヤを更新する

#### リクエストパス

/rest/layer/[cid]/[groupId]/[layerId]

#### リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
groupId	必須	グループ ID 0 ならサイト全体
layerId	必須	レイヤ ID

#### リクエストデータ

レイヤ情報の JSON 形式データ

ID	データ型	説明
name	文字列	レイヤ名称
description	文字列	レイヤの説明
attribution	文字列	レイヤの著作権者情報
featuretypeId	文字列	WMS リクエスト用の ID
wfsURL	文字列	WFS 検索用 URL
wmsCapabilitiesURL	文字列	WMS Capabilities 用 URL
wmsURL	文字列	WMS 地図画像取得 URL
wmsFormat	文字列	WMS 地図を取得する画像フォーマット image/png image/gif image/jpeg WMS ベースレイヤのみ利用可
wmsLegendURL	文字列	WMS 凡例画像取得 URL
wmsFeatureURL	文字列	WMS 地図情報取得 URL
scale	整数	最小表示縮尺

maxScale	整数	最大表示縮尺
----------	----	--------

リクエスト例：

```
{"attribution":"","description":"","maxScale":0,"name":"名称","scale":0 }
```

リクエスト例：

```
{"attribution":"","description":"","maxScale":0,"name":"WMSマップ","scale":0,"visible":false,"wms capabilitiesurl":"http://wmssample.com/wms?SERVICE=wms&REQUEST=GetCapabilities&VERSION=1.1.1","wms format":"image/png","wms legendurl":"http://wmssample.com/map/wms?&SERVICE=WMS&","wmsURL":"http://wmssample.com/map/wms?&SERVICE=WMS&"}
```

レスポンス (コード：200)

レスポンスデータなし

エラー

404	レイヤが存在しない
409	更新時に衝突を検知した

## DELETE

レイヤを削除する

リクエストパス

```
/rest/layer/[cid]/[groupId]/[layerId]
```

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
groupId	必須	グループ ID 0 ならサイト全体
layerId	必須	レイヤ ID

レスポンス (コード：200)

レスポンスデータなし

エラー

404	レイヤが存在しない
409	削除時に衝突を検知した

## 2.6 属性情報 API

### リソースクラス

jp.ecom-plat.map.rest.resources.AttrInfoResource.java

#### GET

登録情報項目の属性情報を 1 件取得する

#### リクエストパス

```
/rest/attr/[cid]/[layerId]/[attrId]
```

#### リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
layerId	必須	レイヤ ID
attrId	必須	属性 ID

#### レスポンス (コード:200)

属性情報の JSON 形式データ

ID	データ型	説明
attrId	文字列	属性 ID
name	文字列	属性名称
dataType	整数	データ型 12:文字列,4:整数,6:数値,91:日付,92:日付と時間,93:時間, 10010:URL,10020:HTML を含む文書,11000:選択,11020:チェックボックス
dataExp	文字列	チェック値 選択項目であればカンマ区切りの文字列 チェックボックスであれば選択時に設定する文字列
status	整数	状態 0:編集可,1:編集不可,-1:検索時に表示しない
length	整数	表示文字数
nullable	倫理	必須入力 true:null 値が許可される

レスポンスエンティティクラス

jp.ecom-plat.map.rest.entity.AttrInfoEntity.java

エラー

404	レイヤまたは属性が存在しない
-----	----------------

## GET

登録情報項目の属性情報一覧を取得する

リクエストパス

/rest/attr/[cid]/[layerId]/
-----------------------------

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
layerId	必須	レイヤ ID

レスポンス (コード:200)

属性情報一覧の JSON 形式データ

ID	データ型	説明
total	整数	結果の総数
results	配列	属性情報の JSON 形式データの配列

属性情報の JSON 形式データ

ID	データ型	説明
attrId	文字列	属性 ID
name	文字列	属性名称
dataType	整数	データ型 12:文字列,4:整数,6:数値,91:日付,92:日付と時間,93:時間, 10010:URL,10020:HTML を含む文書,11000:選択,11020:チェックボックス
dataExp	文字列	チェック値 選択項目であればカンマ区切りの文字列 チェックボックスであれば選択時に設定する文字列
status	整数	状態

		0:編集可,1:編集不可,-1:検索時に表示しない
length	整数	表示文字数
maxLength	整数	属性最大文字数 指定がなければ 0
nullable	倫理	必須入力 true:null 値が許可される

レスポンス例 :

```
{
  "results": [
    {
      "attrId": "attr0",
      "dataExp": "",
      "dataType": 12,
      "length": 20,
      "maxLength": 0,
      "name": "名称",
      "nullable": true,
      "status": 0
    },
    {
      "attrId": "attr1",
      "dataExp": "O",
      "dataType": 11020,
      "length": 20,
      "maxLength": 0,
      "name": "公開",
      "nullable": true,
      "status": 0
    }
  ],
  "total": 2
}
```

レスポンスエンティティクラス

jp.ecom.plat.map.rest.entity.AttrInfoList.java

## POST

登録情報項目に属性情報を新規に登録する

リクエストパス

```
/rest/attr/[cid]/[layerId]
```

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
layerId	必須	レイヤ ID

リクエストデータ

属性情報の JSON 形式データ

ID	データ型	説明
name	文字列	属性名称
dataType	整数	データ型 12:文字列,4:整数,6:数値,91:日付,92:日付と時間,93:時間, 10010:URL,10020:HTML を含む文書,11000:選択,11020:チェックボックス
dataExp	文字列	チェック値 データ型が選択項目であればカンマ区切りの文字列、 チェックボックスであれば選択時に設定する文字列
status	整数	状態 0:編集可,1:編集不可,-1:検索時に表示しない



length	整数	表示文字数
nullable	倫理	必須入力 true:null 値が許可される

リクエスト例：

```
["dataExp":"","dataType":12,"length":20,"name":"名称","nullable":true,"status":0]
```

レスポンス (コード：200)

ID	データ型	説明
timestamp	整数	データ取得サーバ時間
result	JSON	登録結果の JSON 形式データ
attrId	整数	追加された属性 ID

レスポンス例：

```
{"timestamp":12345678, "result":{"attrId":"attr1"}}
```

エラー

404	レイヤが存在しない
-----	-----------

## PUT

属性情報を更新する

リクエストパス

```
/rest/attr/[cid]/[layerId]/[attrId]
```

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
layerId	必須	レイヤ ID
attrId	必須	属性 ID

リクエストデータ

属性情報の JSON 形式データ

ID	データ型	説明
name	文字列	属性名称
dataType	整数	データ型 12:文字列,4:整数,6:数値,91:日付,92:日付と時間,93:時間,

		10010:URL,10020:HTML を含む文書,11000:選択,11020:チェックボックス
dataExp	文字列	チェック値 データ型が選択項目であればカンマ区切りの文字列、 チェックボックスであれば選択時に設定する文字列
status	整数	状態 0:編集可,1:編集不可,-1:検索時に表示しない
length	整数	表示文字数
nullable	倫理	必須入力 true:null 値が許可される

リクエスト例：

```
["dataExp":"","dataType":12,"length":20,"name":"名称","nullable":true,"status":0]
```

レスポンス (コード：200)

レスポンスデータなし

エラー

404	レイヤまたは属性情報が存在しない
-----	------------------

## DELETE

属性情報を削除する

リクエストパス

```
/rest/attr/[cid]/[layerId]/[attrId]
```

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
layerId	必須	レイヤ ID
attrId	必須	削除する属性 ID

レスポンス (コード：200)

レスポンスデータなし

エラー

404	レイヤまたは属性情報が存在しない
-----	------------------

## 2.7 描画設定 API

リソースクラス

jp.ecom-plat.map.rest.resources.SldResource.java

### GET

SLD を取得する

リクエストパス

/rest/sld/[cid]/[mapId]/[layerId]

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
mapId	必須	地図 ID 0 なら共通描画設定を取得する。 地図 ID を指定すると地図に対して設定された描画設定を取得するが、地図に描画設定がない場合は、共通描画設定を返却する。 共通描画設定も無い場合は、デフォルトの描画設定が返却される。 何が返却されたかは、 NamedLayer → UserStyle → Name の名称で判断可能。
layerId	必須	レイヤ ID

レスポンス (コード : 200)

描画設定 SLD フォーマットの XML 文字列

以下のタグの値で、デフォルトの SLD か共通の SLD か地図固有の SLD かを区別する

NamedLayer → UserStyle → Name の内容が

"layer style :"+レイヤ ID

の場合は共通スタイル

"map style : "+地図 ID+"/"+レイヤ ID

の場合は、地図専用のスタイルとなる。

レスポンス例 :

```

<?xml version="1.0" encoding="UTF-8"?>
<sld:StyledLayerDescriptor xmlns="http://www.opengis.net/sld" xmlns:sld="http://www.opengis.net/sld" xmlns:ogc="http://www.opengis.net/ogc" xmlns:gml="http://www.opengis.net/gml" version="1.0.0"
>
<sld:NamedLayer>
<sld:Name>map:c1227</sld:Name>
<sld:UserStyle>
<sld:Name>layer style : c1227</sld:Name>
<sld:Title/>
<sld:FeatureTypeStyle>
<sld:Name>name</sld:Name>
<sld:Rule>

~中略~

</sld:Rule>
</sld:FeatureTypeStyle>
</sld:UserStyle>
</sld:NamedLayer>
</sld:StyledLayerDescriptor>

```

#### エラー

404	レイヤが存在しない
-----	-----------

## POST / PUT

SLD を登録・更新する

POST と PUT で同一の処理を行う

#### リクエストパス

```
/rest/sld/[cid]/[mapId]/[layerId]
```

#### リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
mapId	必須	地図 ID 0 なら共通描画設定を登録・更新する。 地図 ID を指定すると地図に対する描画設定を登録・更新する。
layerId	必須	レイヤ ID

レスポンス (コード:200)

レスポンスデータなし

エラー

404	レイヤが存在しない
409	更新時に衝突を検知した

## DELETE

SLD を削除する

地図に設定された SLD を削除して、共通描画設定を利用する場合に利用

リクエストパス

/rest/sld/[cid]/[mapId]/[layerId]

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
mapId	必須	地図 ID 地図 ID を指定すると地図に対する描画設定を登録・更新する。
layerId	必須	レイヤ ID

レスポンス (コード:200)

レスポンスデータなし

エラー

404	レイヤが存在しない
404	SLD が存在しない

## 2.8 描画設定アイコン API

リソースクラス

jp.ecom-plat.map.rest.resources.SldIconResource.java

### GET

描画設定用のアイコン一覧を取得する

リクエストパス

/rest/sldicon/[cid]/[groupId]

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
groupId	無し or 0 以上	グループ ID 0 より大きい値が設定された場合は、そのグループに登録されたアイコン一覧を <code>groupIcon</code> に格納する

レスポンス (コード:200)

アイコン一覧の JSON 文字列

`defaultIcon`、`groupIcon`、`bosaiIcon` にカテゴリ分けされたアイコン一覧が格納される

アイコン一覧

ID	データ型	説明
category	文字列	アイコンカテゴリ名称
icons	JSON 配列	アイコン情報の JSON 配列
name	文字列	アイコン名称
url	文字列	アイコンのパス /map/files/icons から記載

レスポンス例:

```
{
  "bosaiIcon": [{"category": "基本項目", "icons": [{"name": "集会場.png", "url": "/map/files/icons/bosai/集会場.png"}, {"name": "学校.png", "url": "/map/files/icons/bosai/学校.png"}]},
  "defaultIcon": {"icons": [{"name": "100.png", "url": "/map/files/icons/default/100.png"}, {"name": "101.png", "url": "/map/files/icons/default/101.png"}, {"name": "102.png", "url": "/map/files/icons/default/102.png"}, {"name": "103.png", "url": "/map/files/icons/default/103.png"}]},
  "groupIcon": [{"icons": [{"name": "DELアイコン32.png", "url": "/map/files/icons/c13/0/DELアイコン"}]}
```

```
32.png"}, {"name": "保健所管内別病院.png", "url": "/map/files/icons/c13/0/保健所管内別病院.png"}]
```

## 2.9 登録情報 API

リソースクラス

jp.ecom-plat.map.rest.resources.FeatureResource.java

### GET

登録情報を 1 件取得する

リクエストパス

/rest/feature/[cid]/[mapId]/[layerId]/[featureId]

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
mapId	必須	マップ ID
layerId	必須	レイヤ ID
featureId	必須	フィーチャ ID
sort		並び替え属性 ID 先頭が "-" なら降順
limit		取得件数
offset		取得位置
time		フィーチャを取得する時系列指定 日時は ISO8601 形式で記述する。タイムゾーンは UTC とする。 YYYY-MM-DDThh:mm:ss.sssZ 期間指定の場合は開始日時と終了日時を「/」で区切り記述する。

レスポンス (コード : 200)

Feature の JSON 形式データ

ID	データ型	説明
timestamp	整数	データ取得サーバ時間
result	JSON	フィーチャの JSON 形式データ

フィーチャの JSON 形式データ

ID	データ型	説明
fid	整数	フィーチャの固有 ID
wkt	WKT	WKT 方式のジオメトリ



time_from	文字列	フィーチャの期間開始日時 日時は ISO8601 形式で記述する。タイムゾーンは UTC とする。 YYYY-MM-DDThh:mm:ss.sssZ 開始日時が-infinitiy の場合の表記は、-292269054-12-02T23:00:00.000Z
time_to	文字列	フィーチャの期間終了日時 日時は ISO8601 形式で記述する。タイムゾーンは UTC とする。 YYYY-MM-DDThh:mm:ss.sssZ 終了日時が infinitiy の場合の表記は、292278994-08-16T23:00:00.000Z
properties	配列	属性コンテナ
attr?	属性の型	
files	JSON 配列	
title	文字列	ファイルタイトル
url	文字列	ファイルの URL

レスポンス例：

```
{"timestamp":1443087286434, "result":[{"fid":1, "wkt":"POINT(135.0 35.0)", "properties":{"attr1":"属性1"}, files:["title":"","url":""]}
```

レスポンスエンティティクラス

jp.ecom.plat.map.rest.entity.FeatureInfo.java

エラー

404	レイヤまたはフィーチャーが存在しない
-----	--------------------

## GET

登録情報の一覧を取得する

リクエストパス

```
/rest/feature/[cid]/[mapId]/[layerId]/
```

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID

mapId	必須	マップ ID
layerId	必須	レイヤ ID
sort		並び替え属性 ID 先頭が"-"なら降順
limit		取得件数
offset		取得位置
keywords		検索条件
time		フィーチャを取得する時系列指定 日時は ISO8601 形式で記述され、タイムゾーンは UTC とする。 YYYY-MM-DDThh:mm:ss.sssZ 期間指定の場合は開始日時と終了日時を「/」で区切り記述する。

### レスポンス (コード : 200)

#### Feature 一覧の JSON 形式データ

ID	データ型	説明
total	整数	結果の総数
results	配列	Feature の JSON 形式データの配列

#### フィーチャの JSON 形式データ

ID	データ型	説明
fid	整数	フィーチャの固有 ID
wkt	WKT	WKT 方式のジオメトリ
time_from	文字列	フィーチャの期間開始日時 日時は ISO8601 形式で記述する。タイムゾーンは UTC とする。 YYYY-MM-DDThh:mm:ss.sssZ 開始日時が-infinitiy の場合の表記は、-292269054-12-02T23:00:00.000Z
time_to	文字列	フィーチャの期間終了日時 日時は ISO8601 形式で記述する。タイムゾーンは UTC とする。 YYYY-MM-DDThh:mm:ss.sssZ 終了日時が infinitiy の場合の表記は、292278994-08-16T23:00:00.000Z
properties	JSON 配列	属性コンテナ
attr?	属性の型	
files	JSON 配列	

title	文字列	ファイルタイトル
url	文字列	ファイルの URL

レスポンス例：

```
{ "total": 123, "results": [ { "fid": 1, "wkt": "POINT(135.0 35.0)", "properties": { "attr1": "属性 1", "attr2": "属性 2" }, "files": [ { "title": "", "url": "" } ] } ] }
```

レスポンスエンティティクラス

jp.ecom-plat.map.rest.entity.FeatureInfoList.java

エラー

404	レイヤが存在しない
-----	-----------

## POST

Feature を新規に登録する

リクエストパス

```
/rest/feature/[cid]/[mapId]/[layerId]
```

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
mapId	必須	マップ ID
layerId	必須	レイヤ ID

リクエストデータ

Feature の JSON 形式データ

ID	データ型	説明
wkt	WKT	WKT 方式のジオメトリ
time_from	文字列	時系列履歴対応のフィーチャの登録日時を指定する。日時の指定が無い場合、サーバの現在時刻を登録する。 日時は ISO8601 形式で記述する。タイムゾーンは UTC とする YYYY-MM-DDThh:mm:ss.sssZ 開始日時が-infinitiy の場合の表記は、-292269054-12-02T23:00:00.000Z
properties	JSON 配列	
attr?	属性の型	

files	JSON 配列	同時にファイル情報も登録する場合に指定
title	文字列	ファイルタイトル
url	文字列	ファイルの URL

リクエスト例 :

```
{"wkt":"POINT(135.0 35.0)","properties":{"attr1":"属性1"}, files:["title":"","url":"",""]}
```

レスポンス (コード : 200)

ID	データ型	説明
timestamp	整数	データ取得サーバ時間
result	JSON	フィーチャの JSON 形式データ
fid	整数	追加されたフィーチャの ID

レスポンス例 :

```
{"timestamp":12345678, "result":{"fid":1}}
```

レスポンスエンティティクラス

jp.ecom-plat.map.rest.entity.FeatureInsert.java

エラー

404	レイヤが存在しない
409	登録時に衝突を検知した

## PUT

Feature を更新する

リクエストパス

```
/rest/feature/[cid]/[mapId]/[layerId]/[featureId]
```

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
mapId	必須	マップ ID
layerId	必須	レイヤ ID
featureId	必須	更新するフィーチャ ID

リクエストデータ

Feature の JSON 形式データ

ID	データ型	説明
fid	整数	更新するフィーチャの ID
wkt	WKT	WKT 方式のジオメトリ
time_from	文字列	時系列履歴対応のフィーチャの更新日時を指定する。日時の指定が無い場合、サーバの現在時刻を登録する。 日時は ISO8601 形式で記述され、タイムゾーンは UTC とする。 YYYY-MM-DDThh:mm:ss.sssZ 開始日時が-infinitiy の場合の表記は、-292269054-12-02T23:00:00.000Z
properties	JSON 配列	
attr?	属性の型	
files	JSON 配列	同時にファイル情報も登録する場合に指定
title	文字列	ファイルタイトル
url	文字列	ファイルの URL

リクエスト例：

```
{ "fid": 1, "wkt": "POINT (135.0 35.0)", "properties": { "attr1": "属性 1" }, "files": [ { "title": "", "url": "" } ] }
```

レスポンス (コード：200)

レスポンスデータなし

エラー

404	レイヤが存在しないまたは更新するフィーチャが存在しない
409	更新時に衝突を検知した

## DELETE

Feature を削除する

リクエストパス

```
/rest/feature/[cid]/[mapId]/[layerId]/[featureId]
```

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID

mapId	必須	マップ ID
layerId	必須	レイヤ ID
featureId	必須	削除するフィーチャ ID
time_to		時系列履歴対応のフィーチャーの削除日時を指定する。日時の指定が無い場合、サーバの現在時刻を利用する。 日時は ISO8601 形式で記述する。タイムゾーンは UTC とする。 YYYY-MM-DDThh:mm:ss.sssZ

#### レスポンス (コード : 200)

レスポンスデータなし

#### エラー

404	レイヤが存在しないまたは削除するフィーチャが存在しない
409	削除時に衝突を検知した

## 2.10 フィーチャファイル情報 API

リソースクラス

jp.ecom-plat.map.rest.resources.FeatureFileResource.java

### GET

フィーチャのファイル情報を取得する

リクエストパス

```
/rest/featurefile/[cid]/[layerId]/[featureId]
```

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
layerId	必須	レイヤ ID
featureId	必須	フィーチャ ID
time		フィーチャを取得する時系列指定 日時は ISO8601 形式で記述され、タイムゾーンは UTC とする。 YYYY-MM-DDThh:mm:ss.sssZ 期間指定の場合は開始日時と終了日時を「/」で区切り記述する。

レスポンス (コード : 200)

フィーチャの JSON 形式データ

ID	データ型	説明
results	JSON 配列	フィーチャのファイル情報一覧
fid	整数	フィーチャ ID
title	文字列	ファイルタイトル
url	文字列	ファイルの URL

レスポンス例 :

```
["results": [{"fid":1,"title":"","url":""}, {"fid":1,"title":"","url":""}]]
```

レスポンスエンティティクラス

jp.ecom-plat.map.rest.entity.FeatureFiles.java

エラー

404	レイヤまたはフィーチャが存在しない
-----	-------------------

## GET

レイヤに登録されたすべてのフィーチャのファイル情報を取得する

リクエストパス

```
/rest/featurefile/[cid]/[layerId]/
```

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
layerId	必須	レイヤ ID
limit		最大件数
offset		検索開始件数
center		距離順ソートの判別用の座標 カンマ区切りで地点の x 座標、y 座標を記入する。
sort		ソート種別 (距離、登録日時)
desc		降順でソートを行う
imgonly		画像のみ取得
time		フィーチャを取得する時系列指定 日時は ISO8601 形式で記述され、タイムゾーンは UTC とする。 YYYY-MM-DDThh:mm:ss.sssZ 期間指定の場合は開始日時と終了日時を「/」で区切り記述する。

レスポンス (コード : 200)

フィーチャの JSON 形式データ

ID	データ型	説明
results	JSON 配列	フィーチャのファイル情報一覧
fid	整数	フィーチャ ID
title	文字列	ファイルタイトル
url	文字列	ファイルの URL

レスポンス例 :

```
{"results":[{"fid":1,"title":"","url":""}, {"fid":1,"title":"","url":""}]}
```



## エラー

404	レイヤまたはフィーチャーが存在しない
-----	--------------------

## POST/PUT

フィーチャーのファイル情報を登録・更新する  
POST と PUT で同一の処理を行う

### リクエストパス

<code>/rest/featurefile/[cid]/[mapId]/[layerId]/[featureId]</code>
--

### リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
mapId	必須	マップ ID
layerId	必須	レイヤ ID
featureId	必須	更新するフィーチャ ID

### リクエストデータ

ファイル一覧の JSON 形式データ

ID	データ型	説明
files	JSON 配列	ファイル一覧
title	文字列	ファイルタイトル
url	文字列	ファイルの URL

### リクエスト例 :

<pre>{"files": [{"url": "/map/files/contents/1/ci/photo.png", "title": "アップロードした画像"}, {"url": "http://example.com/", "title": "外部のリンク"}]}</pre>
---

### レスポンス (コード : 200)

レスポンスデータなし

## エラー

401	編集権限がない
404	レイヤまたはフィーチャーが存在しない
500	登録に失敗した

## 2.11 コンテンツファイルアップロード API

リソースクラス

jp.ecom-plat.map.rest.resources.UploadContentsResource.java

### POST

フィーチャー関連したファイル（写真等）をアップロードする

アップロードパス

webapps/map/files/contents/[ユーザ ID]/[レイヤ ID]/

リクエストパス

/rest/upload/contents/[cid]/[layer]/

リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
layer	必須	レイヤ ID

レスポンス（コード：200）

ID	データ型	説明
results	文字列配列	フィーチャのファイル情報一覧

レスポンス例：

```
{"results":["/map/files/contents/1/c1/photo.png", "/map/files/contents/1/c1/file.zip"]}
```

レスポンスエンティティクラス

jp.ecom-plat.map.rest.entity.UploadFiles.java

エラー

401	編集権限がない
403	ファイル拡張子が許可されていない
404	レイヤが存在しない

## 2.12 アイコンファイルアップロード API

(仕様調整中 未実装)

### リソースクラス

`jp.ecom-plat.map.rest.resources.UploadIconResource.java`

## POST

描画設定でグループアイコンをアップロードする

### アップロードパス

`webapps/map/files/icons/[サイト ID]/[グループ ID]/[レイヤ ID]/`

### リクエストパス

`/rest/upload/icon/[layer]`

### リクエストパラメータ

ID	入力	説明
cid	必須	サイト ID
layer	必須	レイヤ ID

### レスポンス (コード : 200)

ID	データ型	説明
results	文字列配列	アップロードされたファイル一覧

### レスポンス例 :

```
["results":["/map/files/icons/c1/0/c100/icon.png","/map/files/icons/c1/0/c100/icon2.png "]]
```

### レスポンスエンティティクラス

`jp.ecom-plat.map.rest.entity.UploadFiles.java`

### エラー

401	編集権限がない
404	レイヤが存在しない

### 3 API フィルター

#### 3.1 ユーザ認証フィルター

API への全てのリクエストでセッションを参照してログイン情報の取得を行う。

ログイン情報からユーザ情報クラスの取得を行い、`HttpServletRequest` の `attribution` に属性名「`userInfo`」で「`UserInfo`」オブジェクトを格納する。

フィルタークラス

`jp.ecom_plat.map.rest.resources.filter.AuthorityFilter`

#### 3.2 メソッドフィルター

REST API へのリクエストの HTTP メソッドを確認しメソッドが(GET/POST)の場合、クエリパラメータを参照して「`_method`」の値を確認する。「`_method`」の値で(PUT/DELETE)が指定されている場合、`HttpServletRequest` の HTTP メソッドを指定された HTTP メソッドに変更する。

フィルタークラス

`jp.ecom_plat.map.rest.resources.filter.MethodFilter`

リクエストパラメータ

ID	説明
<code>_method</code>	PUT、DELETE メソッドに対応しないクライアントプログラムからのリクエストでは、リクエストパラメータによりメソッドを指定する。 POST、GET リクエストに <code>_method</code> で指定することで対応するメソッドの処理が実行される。