

公共交通情報流通連携基盤システム

プログラミングマニュアル

2013年3月29日

株式会社横須賀テレコムリサーチパーク

1. システム構成

公共交通情報流通連携基盤システム

本システムは、Linux OS (Kernel 2.6.26 以上) を搭載したクラウドサーバ上で動作する。具体的なシステム構成は以下の通りである。

◆基盤FrontEndサーバー

基盤フロントエンドサーバーは、外部に情報配信を行うデータを扱うサーバーである。Web Application FrameworkとしてRailsを使用し、REST APIなどをこのサーバーに実装する。この環境は、AWS環境にてAutoScalingすることを前提として構築を行う。したがって、実際のDB本体はこの中では稼働せず、全て後述するDBサーバーを参照するように実装される。

本サーバーのソフトウェアの構成を以下に示す。

- ・ Ubuntu Server 12.04 LTS 64bit
DHCPにてIP取得
ファイヤーウォール設定無し
- ・ nginx PPA
- ・ rbenv
JRuby > v1.7.2
Ruby on Rails > v3.2.2

◆基盤MongoDBサーバー

基盤MongoDBサーバーは、情報流通連携基盤にて扱うデータをストアするためのDBサーバーである。MongoDBが稼働しており、情報流通連携基盤で扱うデータ形式でデータを保持する。

本サーバーのソフトウェアの構成を以下に示す。

- ・ Ubuntu Server 12.04 LTS 64bit
DHCPにてIP取得
ファイヤーウォール設定無し
- ・ MongoDB PPA

◆基盤PostGISサーバー

基盤PostGISサーバー(バックエンドサーバー)は、内部的に用いるデータをストアするために用いる。PostgreSQLとPostGISが稼働しており、下記に示す国土地理院発行の国土数値情報をストアする。

本サーバーのソフトウェアの構成を以下に示す。

- ・ Ubuntu Server 12.04 LTS 64bit
 - DHCPにてIP取得
 - ファイヤーウォール設定無し
- ・ APT
 - PostgreSQL
 - PostGIS
 - SSH Server

【PostgreSQLに設定するDB】

DB:gsi				概要
public	busroutes	table	postgres	バス経路
public	busroutes_gid_seq	sequence	postgres	
public	busstops	table	postgres	バス停
public	busstops_gid_seq	sequence	postgres	
public	coast	table	postgres	日本海岸線
public	coast_gid_seq	sequence	postgres	
public	dline	table	postgres	自治境界線
public	dline_gid_seq	sequence	postgres	
public	geography_columns	view	postgres	鉄道路線
public	geometry_columns	view	postgres	
public	railroads	table	postgres	
public	railroads_gid_seq	sequence	postgres	駅
public	spatial_ref_sys	table	postgres	
public	stations	table	postgres	
public	stations_gid_seq	sequence	postgres	

2. システム機能の概要

公共交通情報流通連携基盤システム

本システムでは、利用者プログラム(アプリケーション)に対して、以下の機能を提供する。

2-1. フロントエンドAPI機能

フロントエンドAPI機能は、外部からの要求を受け付け、仕様に基づく処理を行い、応答メッセージを作成する機能である。フロントエンドAPI機能は、以下に挙げる9種類の小機能からなる。

(1) SPARQL-Based Command

SPARQL 仕様に基づき、情報流通連携基盤に格納されたオープンデータを検索・登録・閲覧・削除する機能。

(2) Traceability/RealtimeData Command

トレーサビリティ情報やイベントログ、リアルタイムデータに代表される、時系列データ処理を伴うオープンデータ操作を行う機能。

(3) Geographical Data Management Command

地理情報演算を伴うオープンデータ操作を行う機能。

(4) Security Management Command

ユーザ・グループ管理、アクセス制御に代表される、セキュリティ操作を行う機能。

(5) Vocabulary Management Command

ボキャブラリ管理を行う機能。

(6)Notification Management Command

オープンデータの登録・更新に呼応して、情報流通連携基盤システムが利用者プログラムに通知する機能。

(7)Triple Management Command

センサやスマートメータのような小型機器を効率的に扱うために、RDFトリプルを簡便に操作する機能。

(8)Identification Resolution Command

ucode からオープンデータの格納先を解決する機能。

(9)Traffic Extension (交通実証拡張)

列車・バスに関する運行情報を効率的に取得する機能。

2-2. データベース機能

データベース機能は、公共交通情報流通連携基盤システムが保持するデータを管理する機能を提供する。本機能は、以下に挙げる3つの小機能からなる。

(1)Semantic DB Server

RDF ベースのデータを格納・検索する機能を提供する。

(2)RDF/JSON Server

交通実証で頻繁に利用される情報をDocument Store型データとして管理し、データの格納・検索機能を提供する。

(3)Database Adapter

前項の(1)(2)に挙げたデータベースを同期・連携させる機能を提供する。

2-3. 共通規定事項

以下、本システムにおける標準APIに対する共通規定事項を説明する。

(1) 準拠するプロトコル・認可方式

標準APIは、HTTP/1.1 プロトコル上で規定する。また、アプリケーションの認可を行う場合の手法は、OAuth 2.0に準拠する。

(2) HTTPメソッドと機能の対応

標準APIのHTTPメソッド名と機能との対応は、基本的に一般的なRESTful APIに従い、以下の通りとする。

HTTP メソッド	機能
GET	取得・検索
POST	新規登録
PUT	更新
DELETE	削除

(3) HTTPステータスコード

標準 APIを受け付けた情報流通連携基盤システムが返すステータスコードとその意味は、以下の通りである。

ステータス コード	意味
200	OK. 正常完了。
201	Created. 対象リソースを作成した。
204	No Content. 正常完了（レスポンスメッセージがない場合）
400	Bad Request. パラメータエラー。
401	Not Authenticated. 未認証状態または認証失敗
403	Forbidden. 権限エラー。
404	Not Found. 対象となる情報が存在しない、定義されていない
409	Conflict. 既登録情報と重複したため登録に失敗した。
413	Request Entity Too Large. リクエストが許容範囲を超過
500	Internal Error. システム内部のエラー。
501	Not Implemented. 実装されていない機能

(4) クエリパラメータとレスポンス形式

クエリパラメータおよびレスポンスの形式は、JSON形式またはXML形式とする。「3.1 SPARQL-based Command」を除いて、両者の区別方法は以下の2通りであるが、原則として前者を使用する。デフォルトのレスポンス形式はJSONとする。

- ・呼び出しURLの末尾に、jsonまたは.xmlを追加
- ・呼び出しURLのクエリパラメータとして、format=jsonまたはformat=xmlを追加

POSTまたはPUTメソッドで「JSONまたはXML形式で格納する」と規定する構造化データを送る場合は、要求するレスポンス形式と同じフォーマットで記述する。たとえば、XML形式のレスポンスを要求する場合は、構造化データをXML形式で記述する。

情報流通連携基盤システムは、受け付けたリクエストを正しく処理できた場合、ステータスコード200、201、または204を返す。レスポンスメッセージがある（ステータスコードが200である）場合は、JSON形式またはXML形式のテキストデータをメッセージボディに載せる。XML形式のデータのルート要素は<api_response>である。

情報流通連携基盤システムは、受け付けたリクエストを正しく処理できなかった場合、200番台以外のステータスコードを返す。このとき、パラメータ名がmsg、パラメータ値がtext/plainであるデータをJSON形式またはXML形式にしたテキストデータを、エラーメッセージとしてメッセージボディに載せる。XML形式のエラーメッセージのルート要素は<error_response>である。

(5) リクエストパラメータの代替表記

「3.1 SPARQL-based Command」を除き、リクエストパラメータにWell-KnownなURIを指定する場合は、URLエンコーディングを回避するために、以下の代替表記を行う。

- ・ リクエストパラメータのURIがucode URNである場合は、ucode値に「ucode」という接頭辞を追加した文字列を代替表記とする。たとえば「ucode_00001C000000000000001000000010000」の代わりに「ucode_00001C000000000000001000000010000」と表記する。
- ・ リクエストパラメータのURIが指定のボキャブラリである場合は、alias URI欄に記載したQNameとローカル名をアンダーバー“_”で接続した文字列を代替表記とする。たとえば「http://purl.org/dc/elements/1.1/title」の代わりに「dc_title」と表記する。

(6) データ型

本システムでは、XML Schema が規定するデータ型に加えて、以下に列記したデータ型も利用する。

型名	説明
struct	構造体
RDF	RDF/XMLまたはRDF/JSON
(型名) []	(型名)の配列

「場所情報の新規登録」「地図情報の新規登録」「ルールの新規登録」「ユーザの新規登録」「グループの新規登録」「オープンデータの新規作成」の各コマンドにおいてパラメータ名にrdfを指定した場合、以下のURIを指定することにより、ucodeの自動発行を行う。ただし、<val>は英文字で始まる英数字の列である。

ucode_?<val>

このとき、同じURIには同じucodeを割り当てる。

(7) Streams API

「Traceability/RealTimeData Management Command」および「Triple Management Command」の検索・閲覧コマンドにパラメータ “ stream” を指定した場合、Stream APIに基づき、コネクションを継続し、値が更新されるごとにその結果を返す。

なお、streamパラメータの値が0である場合、サーバが許容する最大時間を指定したものとする。
また、Stream APIに基づき接続を継続できる最大時間は、実装依存とする。

(8) その他の共通規定

上記に加えて、以下の共通規定を設けている。

- ・ POST、PUT、DELETEメソッドにおいて、JSON形式で記述するメッセージボディ内で標準APIのパラメータ値としてURIを指定するときは、その値を<>で括弧。ただし、RDF/XMLやRDF/JSONに従う場合はその限りでない。
- ・ HTTP、URL等の規約にしたがったエンコーディングを行う。なお、本資料で示す例では、API仕様の可読性を優先するため、これらのエンコーディングを行わない場合がある。
- ・ JSON形式で多バイト文字を含むデータを返す場合、JSONの仕様にしたがったエンコーディングを行う。
- ・ 認証を行う際に利用する認証キーについては、別途規定する。

3. ユースケース

公共交通情報流通連携基盤システム

本章では、各APIの具体的な利用例を説明する。

3-1. 現在位置の情報からJR五反田駅を選択して時刻表を調べる

＜クエリ＞

/api/v1/place?lat=35.6260&lon=139.7236&radius=50&rdf_type=ug_station

＜レスポンス＞

```
{
  "places": {
    "ucode_GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG0": {
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#type": [{"value": "http://uidcenter.org/vocab/ucr/ug#Station", "type": "uri"}],
      "http://purl.org/dc/elements/1.1/title": [{"value": "JR五反田駅", "type": "literal"}],
      "http://uidcenter.org/vocab/ucr/ug#Region": [{"value": "GEOM", "type": "literal"}],
      "http://www.w3.org/2003/01/geo/wgs84_pos#lat": [{"value": "35.62644", "type": "literal"}],
      "http://www.w3.org/2003/01/geo/wgs84_pos#long": [{"value": "139.72346", "type": "literal"}]},
    "ucode_GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG1": {
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#type": [{"value": "http://uidcenter.org/vocab/ucr/ug#Station", "type": "uri"}],
      "http://purl.org/dc/elements/1.1/title": [{"value": "東急五反田駅", "type": "literal"}],
      "http://uidcenter.org/vocab/ucr/ug#Region": [{"value": "GEOM", "type": "literal"}],
      "http://www.w3.org/2003/01/geo/wgs84_pos#lat": [{"value": "35.62513", "type": "literal"}],
      "http://www.w3.org/2003/01/geo/wgs84_pos#long": [{"value": "139.72411", "type": "literal"}]},
    "ucode_GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG2": {
```



```

        {"value": "urn:ucode:_zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz", "type": "uri"}}
      ]
    }
  }
}

```

3-2. 山手線が関係する運行情報のリアルタイム配信を受信する

<クエリ>

/api/v1/stats?rdf_type=pt_TrainInfo&pt_TrainInfoLine=山手線&stream=0

運行情報が配信され次第、セッションを確立したクライアントに対してPush配信を行う。
この配信はW3C EventStreamに準拠した形で通信を行う。
一回の配信毎にJSONオブジェクトを配信する。

<レスポンス>

```

{
  "results": {
    "urn:ucode:_xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx0": {
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#type": [{"value": "value": "http://opendata.ubin.jp/puti#TrainInfo", "type": "uri"}],
      "http://purl.org/dc/elements/1.1/time": [{"value": "2013-01-13T20:06:00+9", "type": "literal"}],
      "http://opendata.ubin.jp/puti#TrainInfoTime": [{"value": "2013-01-21T11:21", "type": "literal"}],
      "http://opendata.ubin.jp/puti#TrainInfoLine": [{"value": "山手線", "type": "literal"}],
      "http://opendata.ubin.jp/puti#TrainInfoFrom": [{"value": "大崎", "type": "literal"}],
      "http://opendata.ubin.jp/puti#TrainInfoTo": [{"value": "五反田", "type": "literal"}],
      "http://opendata.ubin.jp/puti#TrainInfoRange": [{"value": "品川-渋谷", "type": "literal"}],
      "http://opendata.ubin.jp/puti#TrainInfoDirection": [{"value": "内回り", "type": "literal"}],
      "http://opendata.ubin.jp/puti#TrainInfoCause": [{"value": "信号トラブル", "type": "literal"}],
      "http://opendata.ubin.jp/puti#TrainInfoStatus": [{"value": "運転見合わせ", "type": "literal"}],
      "http://opendata.ubin.jp/puti#TrainInfoText": [{"value": "山手線は、大崎駅で発生した信号トラブルの影響で、内回りで運転を見合わせています。", "type": "literal"}]
    },
    "urn:ucode:_xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx1": {
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#type": [{"value": "value": "http://opendata.ubin.jp/puti#TrainInfo", "type": "uri"}],
      "http://purl.org/dc/elements/1.1/time": [{"value": "2013-01-13T20:06:00+

```

```

    9", "type": "literal"]],
    "http://opendata.ubin.jp/puti#TrainInfoTime": [{"value": "2013-01-21T09:
85", "type": "literal"}]],
    "http://opendata.ubin.jp/puti#TrainInfoLine": [{"value": "山手線", "type": "
literal"}]],
    "http://opendata.ubin.jp/puti#TrainInfoFrom": [{"value": "大崎", "type": "
literal"}]],
    "http://opendata.ubin.jp/puti#TrainInfoTo": [{"value": "五反田", "type": "
literal"}]],
    "http://opendata.ubin.jp/puti#TrainInfoRange": [{"value": "品川-渋谷", "
type": "literal"}]],
    "http://opendata.ubin.jp/puti#TrainInfoDirection": [{"value": "内回り", "
type": "literal"}]],
    "http://opendata.ubin.jp/puti#TrainInfoCause": [{"value": "混雑", "type": "
literal"}]],
    "http://opendata.ubin.jp/puti#TrainInfoStatus": [{"value": "遅延", "type": "
literal"}]],
    "http://opendata.ubin.jp/puti#TrainInfoText": [{"value": "山手線は、大崎
駅で発生した混雑の影響で、内回りで遅れが出ています。", "type": "literal"}]]
  }
}
}

```

3-3. 東京駅に設置されたセンサーから気温・湿度のリアルタイム配信を受信する

<クエリ>

```
/api/v1/places?lat=35.6815&lon=139.7662&radius=50&rdf_type=puti_Sensor&stream=0
```

運行情報が配信され次第、セッションを確立したクライアントに対してPush配信を行う。
この配信はW3C EventStreamに準拠した形で通信を行う。一回の配信毎にJSONオブジェクトを配信する。

<レスポンス>

```

{
  "results": {
    "urn:ucode:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx": {
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#type": [{"value": "http://
opendata.ubin.jp/puti#Sensor", "type": "uri"}],
      "http://purl.org/dc/elements/1.1/time": [{"value": "2013-01-13T20:06:00+
9", "type": "literal"}]],
      "http://opendata.ubin.jp/uc#temperature": [{"value": "23.9", "type": "
literal"}]],
      "http://opendata.ubin.jp/uc#humidity": [{"value": "30", "type": "literal"}]]
    },
    "urn:ucode:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx": {
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#type": [{"value": "http://
opendata.ubin.jp/puti#Sensor", "type": "uri"}],

```

```

    "http://purl.org/dc/elements/1.1/time": [{"value": "2013-01-13T20:06:00+
    9", "type": "literal"}],
    "http://opendata.ubin.jp/puti#frequency": [{"value": "60", "type": "
    literal"}],
    "http://opendata.ubin.jp/puti#pollen": [{"value": "100", "type": "literal"}]
  }
}

```

以下の各章では、本システムで提供する標準APIについて、具体的に説明する。

4. SPARQL-based Command

公共交通情報流通連携基盤システム

SPARQL-based Commandは、SPARQL 1.1プロトコルに準拠した、オープンデータの登録・更新・削除・閲覧・検索機能を提供する。
本コマンドの一覧を以下の表に示す。

[表4-1] SPARQL-based Command の一覧

URLパス	HTTPメソッド	意味
/api/v1/sparql/	GET	SPARQL 1.1準拠のクエリを発行する
/api/v1/sparql/	POST	SPARQL 1.1準拠のクエリを発行する
/api/v1/rdf-graph-store	GET	RDFグラフを閲覧する
/api/v1/rdf-graph-store	POST	RDFグラフを追加する
/api/v1/rdf-graph-store	PUT	RDFグラフを更新する
/api/v1/rdf-graph-store	DELETE	RDFグラフを削除する
/api/v1/rawdata	GET	Raw Data を閲覧する
/api/v1/rawdata	POST	Raw Data を追加する
/api/v1/rawdata	PUT	Raw Data を更新する
/api/v1/rawdata	DELETE	Raw Data を削除する

URLパスが/api/v1/rdf-graph-store であるAPIは、SPARQL 1.1 Graph Store HTTP Protocolに従ったAPIである。しかし、このAPIはRDFグラフ単位での操作のみサポートしているため、RDFデータのトリプル(またはその集合となる部分グラフ)単位で操作できるAPIを追加する。後者のURLパスは/api/v1/rawdata である。

Named Graphを利用する場合は、本節に記載したAPIのgraph パラメータを利用してRDFグラフを識別する。Named Graph を使用しない場合はgraph パラメータを使用しないか、defaultを利用する(詳細はコマンドごとに説明する)。

本節に記載したAPIを利用する場合、識別子は自動発行されない。

——[4-1: SPARQL1.1 準拠のクエリ発行 (GET メソッド)]——

<機能概要>

HTTP GET メソッドを利用して、SPARQL1.1 準拠のクエリを発行する。

<メソッド>

GET

<URLパス>

/api/v1/sparql?query=<query>

<制約条件>

なし。誰でもリクエストできる。

<リクエストパラメータ>

リクエストパラメータは、GETメソッドのURL部に格納する(以下の表)。

[表4-2] SPARQL1.1 準拠のクエリ発行 (GET メソッド) のリクエストパラメータ

パラメータ名	デフォルト値	説明
query	xsd:string	検索対象パラメータ名

<リクエストヘッダ>

レスポンスの形式を、リクエストヘッダのAcceptで設定する。SPARQL 1.1のSELECTオペレーションを発行する場合に指定できるパラメータ、CONSTRUCT またはDESCRIBEオペレーションを発行する場合に指定できるパラメータ、ASK オペレーションを発行する場合に指定できるパラメータは以下の表の通りである。

[表4-3] SELECTオペレーションのレスポンス形式を指定するAcceptヘッダ値

パラメータ値	説明
application/sparql-results+xml	SPARQL Query Results XML Formatに基づくレスポンス
application/sparql-results+json	SPARQL Query Results JSON Formatに基づくレスポンス

[表4-4] レスポンスのRDFグラフ表現形式を指定するAcceptヘッダ値

パラメータ値	説明
application/rdf+xml	RDF/XML
text/plain	N-Triples
text/rdf+n3	Notation3

[表4-5] レスポンスのバイナリ値形式を指定するAcceptヘッダ値

パラメータ値	説明
application/sparql-results+xml	SPARQL Query Results XML Formatに基づくレスポンス
text/boolean	テキスト表現 (true/false)

<ステータスコード>

ステータスコードは以下の表の通りである。

[表4-6] SPARQL1.1準拠のクエリ発行(GET メソッド)のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	クエリが正しくない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

レスポンスは以下の通りである。

- ・ SELECTオペレーションのレスポンスは、acceptヘッダ値に基づき、以下のいずれかである。
 - SPARQL Query Results JSON Format に基づくレスポンス
 - SPARQL Query Results XML Format に基づくレスポンス
- ・ CONSTRUCT、DESCRIBE オペレーションのレスポンスは、RDFグラフデータである。このフォーマットは、acceptヘッダ値で指定した通りである。
- ・ ASK オペレーションのレスポンスは、acceptヘッダ値に基づき、以下のいずれかである。
 - SPARQL Query Results XML Format に基づくレスポンス
 - true またはfalse の文字列

【APIの利用例】

<クエリ>

```
GET /api/v1/sparql/?query=PREFIX%20dc%3A%20%3Chttp%3A%2F%2Furl.org%2Fdc%2F
elements%2F1.1%2F%3E%20%0ASELECT%20%3Fbook%20%3Fwho%20%0AWHERE%20%7B%20%3F
book%20dc%3Acreator%20%3Fwho%20%7D%0A HTTP/1.1
Host: www.example.org
Accept: application/sparql-results+xml
```


<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Content-Type: application/sparql-results+xml
<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="book"/>
    <variable name="who"/>
  </head>
  <results>
    <result>
      <binding name="book">
        <uri>http://www.example/book/book5</uri>
      </binding>
      <binding name="who">
        <bnode>r29392923r2922</bnode>
      </binding>
    </result>
    ...
  </sparql>
```

――[4-2: SPARQL1.1 準拠のクエリ発行 (POST メソッド)]――

<機能概要>

HTTP POST メソッドを利用して、SPARQL1.1 準拠のクエリを発行する。

<メソッド>

POST

<URL パス>

/api/v1/sparql/

<制約条件>

なし。誰でもリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、メッセージボディ部に格納する。

[表4-7] SPARQL1.1準拠のクエリ発行(POSTメソッド)のリクエストパラメータ

パラメータ名	デフォルト値	説明
query	xsd:string	SPARQLクエリ文をURLエンコーディングしたもの

<リクエストヘッダ>

レスポンスの形式を、リクエストヘッダのAcceptで設定する。その方法は、「3.1.1 SPARQL1.1準拠のクエリ発行(GET メソッド)」と同じである。(以下の表を参照のこと)

<ステータスコード>

ステータスコードは以下の表の通りである。

[表4-8] SPARQL1.1準拠のクエリ発行(POSTメソッド)のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	クエリが正しくない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

レスポンスは以下の通りである。

- ・ SELECTオペレーションのレスポンスは、acceptヘッダ値に基づき、以下のいずれかである。
 - SPARQL Query Results JSON Format に基づくレスポンス
 - SPARQL Query Results XML Format に基づくレスポンス
- ・ CONSTRUCT、DESCRIBE オペレーションのレスポンスは、RDFグラフデータである。このフォーマットは、acceptヘッダ値で指定した通りである。
- ・ ASK オペレーションのレスポンスは、acceptヘッダ値に基づき、以下のいずれかである。
 - SPARQL Query Results XML Format に基づくレスポンス
 - true またはfalse の文字列

【APIの利用例】

<クエリ>

```
POST /api/v1/sparql HTTP/1.1
Host: www.example.org
Accept: application/sparql-results+xml
Content-Type: application/x-www-form-urlencoded
Content-Length: xxx
query=PREFIX%20dc%3A%20%3Chttp%3A%2F%2Furl.org%2Fdc%2Felements%2F1.1%2F%3E%20%0ASELECT%20%3Fbook%20%3Fwho%20%0AWHERE%20%7B%20%3Fbook%20dc%3Acreator%20%3Fwho%20%7D%0A
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Content-Type: application/sparql-results+xml
<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="book"/>
    <variable name="who"/>
  </head>
  <results>
    <result>
      <binding name="book"><uri>http://www.example/book/book5</uri></binding>
      <binding name="who"><bnode>r29392923r2922</bnode></binding>
    </result>
    ...
  </sparql>
```

[4-3: RDFグラフの閲覧]

<機能概要>

RDFグラフを閲覧する。

<メソッド>

GET

<URLパス>

/api/v1/rdf-graph-store?graph=<graph>
/api/v1/rdf-graph-store?default

<制約条件>

なし。誰でもリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する

[表4-9] RDFグラフの閲覧のリクエストパラメータ

パラメータ名	デフォルト値	説明
graph	(指定なし)	閲覧するRDFグラフの識別URI

<リクエストヘッダ>

レスポンスの形式を、リクエストヘッダのAcceptで設定する。その方法は、「3.1.1 SPARQL1.1準拠のクエリ発行(GETメソッド)」のCONSTRUCTオペレーションと同じである。(別表を参照のこと)

<ステータスコード>

ステータスコードは以下の表の通りである。

[表4-10] RDFグラフの閲覧のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	パラメータ値が正しくない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

Acceptヘッダで指定した形式でエンコードされた、RDFグラフ表現。

【APIの利用例】

<クエリ>

```
GET /api/v1/rdf-graph-store?default HTTP/1.1
Host: www.example.org
Accept: application/rdf+xml
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/rdf+xml; charset=utf-8
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:books="http://www.example/book/"
xmlns:dc="http://purl.org/dc/elements/1.1/" >
<rdf:Description rdf:about="http://www.example/book/book6">
<dc:title>Example Book #6 </dc:title>
</rdf:Description>
</rdf:RDF>
```

――[4-4: RDF グラフの追加]――

<機能概要>

RDF グラフを追加する。

<メソッド>

POST

<URLパス>

/api/v1/rdf-graph-store?graph=<graph>

/api/v1/rdf-graph-store?default

<制約条件>

任意のRDFグラフに対して更新権限を有するユーザがリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、POSTメソッドのURL部に格納する。追加するRDFグラフは、メッセージボディ部に格納する。

[表4-11] RDFグラフの閲覧のリクエストパラメータ

パラメータ名	デフォルト値	説明
graph	(指定なし)	追加対象のRDFグラフを識別するURI

<リクエストヘッダ>

追加するRDFグラフの形式を、リクエストヘッダのContent-type で設定する。指定できるパラメータ値とその意味は、「3.1.1 SPARQL1.1準拠のクエリ発行(GET メソッド)」内の別表を参照のこと。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表4-12] RDFグラフの追加のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータ値が正しくない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
POST /api/v1/rdf-graph-store?default HTTP/1.1
Host: www.example.org
Accept: application/rdf+xml
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:books="http://www.example/book/"
  xmlns:dc="http://purl.org/dc/elements/1.1/" >
  <rdf:Description rdf:about="http://www.example/book/book6">
    <dc:title>Example Book #6 </dc:title>
  </rdf:Description>
</rdf:RDF>
```

<レスポンス>

```
HTTP/1.1 204 No Content
```

—[4-5: RDFグラフの更新]—

<機能概要>

RDFグラフを更新する。本クエリの操作完了後に登録されているRDFグラフは、本クエリで指定したものである。本クエリに含まれないRDFグラフは削除される。

<メソッド>

```
PUT
```

<URLパス>

```
/api/v1/rdf-graph-store?graph=<graph>
/api/v1/rdf-graph-store?default
```

<制約条件>

任意のRDFグラフに対して更新権限を有するユーザがリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、PUTメソッドのURL部に格納する。更新するRDFグラフは、メッセージボディ部に格納する。

[表4-13] RDFグラフの更新のリクエストパラメータ

パラメータ名	デフォルト値	説明
graph	(指定なし)	更新対象のRDFグラフを識別するURI

<リクエストヘッダ>

更新するRDFグラフの形式を、リクエストヘッダのContent-type で設定する。指定できるパラメータ値とその意味は、「3.1.1 SPARQL1.1準拠のクエリ発行(GET メソッド)」内の別表を参照のこと。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表4-14] RDFグラフの更新のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータ値が正しくない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
PUT /api/v1/rdf-graph-store?default HTTP/1.1
Host: www.example.org
Accept: application/rdf+xml
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:books="http://www.example/book/"
xmlns:dc="http://purl.org/dc/elements/1.1/" >
<rdf:Description rdf:about="http://www.example/book/book6">
<dc:title>Example Book #6 </dc:title>
</rdf:Description>
</rdf:RDF>
```

<レスポンス>

```
HTTP/1.1 204 No Content
```

――[4-6: RDF グラフの削除]――

<機能概要>

RDF グラフを削除する。本クエリの実施後、RDF グラフは空になる。

<メソッド>

DELETE

<URLパス>

/api/v1/rdf-graph-store?graph=<graph>

/api/v1/rdf-graph-store?default

<制約条件>

任意のRDF グラフに対して更新権限を有するユーザがリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、メッセージボディ部に格納する。

[表4-15] RDF グラフの削除のリクエストパラメータ

パラメータ名	デフォルト値	説明
graph	(指定なし)	削除対象のRDF グラフを識別するURI

<ステータスコード>

ステータスコードは以下の表の通りである。

[表4-16] RDF グラフの削除のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータ値が正しくない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【API の利用例】

<クエリ>

DELETE /api/v1/rdf-graph-store?default HTTP/1.1

<レスポンス>

HTTP/1.1 204 No Content

——[4-7: Triple の閲覧]——

<機能概要>

指定された主語・述語・目的語をもつTriple を閲覧する。

<メソッド>

GET

<URLパス>

/api/v1/rawdata?subject=<subject>&predicate=<predicate>&object=<object>&graph=<graph>

<制約条件>

なし。誰でもリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。

[表4-17] Triple の閲覧のリクエストパラメータ

パラメータ名	デフォルト値	説明
subject	(指定なし)	出力対象のTriple の主語
predicate	(指定なし)	出力対象のTriple の述語
object	(指定なし)	出力対象のTriple の目的語
graph	default	閲覧するRDFグラフを識別するURI

<リクエストヘッダ>

レスポンスの形式を、リクエストヘッダのAcceptで設定する。その方法は、「3.1.1 SPARQL1.1準拠のクエリ発行(GETメソッド)」のCONSTRUCTオペレーションと同じである。(別表を参照のこと)

<ステータスコード>

ステータスコードは以下の表の通りである。

[表4-18] Triple の閲覧のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	パラメータ値が正しくない。
404 Not Found	条件を満たすTripleがない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

subject、predicate、object で指定されたリソースをもつTriple に対するRDFグラフ表現。出力されるグラフの形式は、Acceptヘッダで指定されたものである。

【APIの利用例】

<クエリ>

```
GET /api/v1/rdf-graph-store?subject=%3Curn%3Aucode%3A%5F
00001C00000000000000001000000100123%3C HTTP/1.1
Host: www.example.org
Accept: application/rdf+xml
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/rdf+xml; charset=utf-8
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.1/" >
<rdf:Description rdf:about="urn:ucode:_ 00001C00000000000000001000000100123">
<dc:title>Example Book #6</dc:title>
</rdf:Description>
</rdf:RDF>
```

——[4-8: Triple の追加]——

<機能概要>

Triple を追加する。情報流通連携基盤システムに既に登録されているTriple と主語、述語、目的語のすべてが一致するTriple を、追加登録することはできない。

<メソッド>

POST

<URLパス>

/api/v1/rawdata?graph=<graph>

<制約条件>

指定したTriple を更新できるユーザがリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。追加するTripleは、リクエストボディに記述する。

[表4-19] Triple の追加のリクエストパラメータ

パラメータ名	デフォルト値	説明
graph	default	追加対象のRDFグラフを識別するURI

<リクエストヘッダ>

追加するTriple を表現する形式を、リクエストヘッダのContent-type で設定する。指定できるパラメータ値とその意味は、「3.1.1 SPARQL1.1準拠のクエリ発行(GET メソッド)」内の別表を参照のこと。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表4-20] Triple の追加のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータ値が正しくない。
403 Forbidden	アクセス権限がない。
409 Conflict	情報流通連携基盤システムの中に、指定されたTripleと主語・述語・目的語の完全一致するものが含まれている。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

- ・すべてのTriple が登録された場合に正常終了となる。エラー終了となった場合、指定されたTriple はどれも登録されていない。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
POST /api/v1/rawdata HTTP/1.1
Host: www.example.org
Accept: application/rdf+xml
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:books="http://www.example/book/"
xmlns:dc="http://purl.org/dc/elements/1.1/" >
<rdf:Description rdf:about="urn:ucode:_00001C000000000000001000000100123">
<dc:title>Example Book #6</dc:title>
</rdf:Description>
</rdf:RDF>
```

<レスポンス>

HTTP/1.1 204 No Content

—[4-9: Triple の更新]—

<機能概要>

Triple を更新する。

<メソッド>

PUT

<URLパス>

/api/v1/rawdata?graph=<graph>

<制約条件>

指定したTriple に対して更新権限を有するユーザがリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、PUTメソッドのURL部に格納する。更新するtripleは、メッセージボディに格納する。

[表4-21] Triple の更新のリクエストパラメータ

パラメータ名	デフォルト値	説明
graph	(指定なし)	更新対象のRDFグラフを識別するURI

更新は、削除+追加の順に実施する。更新するTriple は、この順にリクエストボディに記述する。削除するTriple と追加するTriple の間に1行以上の空行を入れること。更新するTripleの表現形式がRDF/XMLでない場合、削除するTriple、追加するTriple それぞれが1行以上の空行を含んではならない。

<リクエストヘッダ>

更新するTriple の表現形式を、リクエストヘッダのContent-type で設定する。指定できるパラメータ値とその意味は、「3.1.1 SPARQL1.1準拠のクエリ発行(GETメソッド)」内の別表を参照のこと。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表4-22] Triple の更新のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータ値が正しくない。
403 Forbidden	アクセス権限がない。
409 Conflict	登録時、情報流通連携基盤システムの中に、指定されたTripleと主語・述語・目的語の完全一致するものが含まれていた。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

- ・すべてのTriple が登録された場合に正常終了となる。エラー終了となった場合、指定されたTriple はどれも登録されていない。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
POST /api/v1/rawdata HTTP/1.1
Host: www.example.org
Accept: application/rdf+xml
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:books="http://www.example/book/"
xmlns:dc="http://purl.org/dc/elements/1.1/" >
<rdf:Description rdf:about="http://www.example/book/book6">
<dc:title>Example Book #6</dc:title>
</rdf:Description>
</rdf:RDF>
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:books="http://www.example/book/"
xmlns:dc="http://purl.org/dc/elements/1.1/" >
<rdf:Description rdf:about="http://www.example/book/book6">
<dc:title>Example Book #7</dc:title>
</rdf:Description>
</rdf:RDF>
```

<レスポンス>

HTTP/1.1 204 No Content

[4-10: Triple の削除]

<機能概要>

指定したTriple を削除する。

<メソッド>

DELETE

<URLパス>

/api/v1/rawdata?graph=<graph>

<制約条件>

指定したTriple に対する更新権限を有するユーザがリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、DELETEメソッドのURL部に格納する。

[表4-23] Triple の削除のリクエストパラメータ

パラメータ名	デフォルト値	説明
graph	(指定なし)	削除対象のRDFグラフを識別するURI

<ステータスコード>

ステータスコードは以下の表の通りである。

[表4-24] Triple の削除のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータ値が正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	指定されたTripleの中に、情報流通連携基盤システムに格納されたTriple
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
DELETE /api/v1/rawdata HTTP/1.1
Host: www.example.org
Accept: application/rdf+xml
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

```

```
xmlns:books="http://www.example/book/"  
xmlns:dc="http://purl.org/dc/elements/1.1/" >  
<rdf:Description rdf:about="http://www.example/book/book6">  
<dc:title>Example Book #7</dc:title>  
</rdf:Description>  
</rdf:RDF>
```

<レスポンス>

HTTP/1.1 204 No Content

5. Traceability/RealTimeData Mngt Command

公共交通情報流通連携基盤システム

Traceability/RealTimeData Management Commandは、トレーサビリティに代表されるイベント管理に必要な機能を提供する。トレーサビリティ管理の対象となる事象をイベント(event)と呼び、それを基本的にucodeで識別する。

本コマンドの一覧を以下の表に示す。

[表5-1] Traceability/RealTimeData Management Command の一覧

URLパス	HTTPメソッド	意味
/api/v1/events	GET	イベントを検索する
/api/v1/events	POST	イベントを登録する
/api/v1/events/<target>	GET	イベントを閲覧する
/api/v1/events/<target>/<property>	GET	イベントを閲覧する
/api/v1/events/<target>	PUT	イベントを更新する
/api/v1/events/<target>/<property>	PUT	イベントを更新する
/api/v1/events/<target>	DELETE	イベントを削除する
/api/v1/events/<target>/<property>	DELETE	イベントを削除する
/api/v1/trace/<ucode>	GET	トレースフォワード/トレースバックを実施する

——[5-1: イベントの検索]——

<機能概要>

イベントを検索する。

<メソッド>

GET

<URLパス>

/api/v1/events?<param1>=<value1>&<param2>=<value2>

<制約条件>

なし。誰でもリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GET メソッドのURL部に格納する。複数のパラメータを指定した場合、AND 検索となる。

[表5-2] イベントの検索のリクエストパラメータ

パラメータ名	デフォルト値	説明
paramN	(指定なし)	検索対象パラメータ名
valueN	(指定なし)	検索対象パラメータ値

- ・ <paramN>、<valueN> の組を1個以上指定しなければならない。
- ・ <paramN> はイベント発生源のオープンデータ属性を示すプロパティURI である。ただし、に記したパラメータも利用できる。URI 値に含まれるカンマは、URLエンコードすること。パラメータ値が複数である場合、それぞれをカンマで区切ること。

offsetとlimitを指定したクエリは、検索結果をイベント発生時刻(ev:date)の新しい順に並べたときのoffset番目からlimit個分を要求することを意味する。ただし、レスポンスがイベント発生時刻(ev:date)順に並んでいることは保証しない。

[表5-3] イベントの検索のリクエストパラメータ

パラメータ名	型	説明
target	xsd:anyURI []	イベント対象(ev:target、ev:source、ev:destination)の識別子
source	xsd:anyURI []	イベント発生源(ev:source)の識別子
destination	xsd:anyURI []	イベント発生の結果生成された(ev:destination)識別子
owner	xsd:anyURI []	イベント発生者(ev:owner、ev:startOwner、ev:endOwner)の識別子
after	xsd:datetime	イベント発生時刻(ev:date)がこの値より後である
before	xsd:datetime	イベント発生時刻(ev:date)がこの値より前である
place	xsd:anyURI []	イベント発行場所(ev:place)の識別子
description	xsd:string	イベント説明文(ev:description/部分一致検索)
stream	xsd:integer	Stream API に基づくコネクションを指定された秒数継続する
offset	xsd:integer	検索結果のオフセット値。省略時は最初から返す。
limit	xsd:integer	検索結果の返却数。省略時は情報流通基盤システムが設定する限界数。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表5-4] イベントの検索のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<param1>、<value1>の組がない。<paramN>が正しくない。
404 Not Found	検索条件を満たすイベントが情報流通連携基盤システムに登録されていない。
413 Request Entity Too Large	limit値が大きすぎる。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

レスポンスは、以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表5-5] イベントの検索のレスポンスパラメータ

パラメータ名	型	説明
events	RDF	イベントのリスト。レスポンス形式にXMLを指定した場合、各イベント情報はRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各イベント情報はRDF/JSONで表現される。
remains	xsd:boolean	条件にマッチするが、レスポンスに含まれないイベントがある場合はtrue、ない場合はfalse。

【APIの利用例】

<クエリ>

```
GET /api/v1/events?source=ucode_00001C000000000000001000000100123 HTTP/1.1
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"events": {
  "ucode_00001C000000000000001000000100800": {
    "http://uidcenter.org/ucr/vocab/event#type": [
      { "value": "urn:ucode:_OFFFDE0000000000000000000000001234567",
        "type": "uri" } ],
    "http://uidcenter.org/ucr/vocab/event#source": [
      { "value": "ucode_00001C000000000000001000000100123",
        "type": "uri" } ],
    "http://uidcenter.org/ucr/vocab/event#destination": [
      { "value": "urn:ucode:_00001C000000000000001000000100125",
        "type": "uri" } ],
    "http://uidcenter.org/ucr/vocab/event#date": [
      { "value": "2012-03-07T12:00:00+0900",
```

```

    "type": "literal" } ] ,
    "http://uidcenter.org/ucr/vocab/event#place": [
      { "value": "urn:ucode:_00001C00000000000000001000000100A00",
        "type": "uri" } ] },
    "urn:ucode:_00001C00000000000000001000000100801": {
      "http://uidcenter.org/ucr/vocab/event#type": [
        { "value": "urn:ucode:_OFFFDE00000000000000000000000000001234567",
          "type": "uri" } ] ,
      "http://uidcenter.org/ucr/vocab/event#source": [
        { "value": "urn:ucode:_00001C00000000000000001000000100123",
          "type": "uri" } ] ,
      "http://uidcenter.org/ucr/vocab/event#destination": [
        { "value": "urn:ucode:_00001C00000000000000001000000100126",
          "type": "uri" } ] ,
        { "value": "urn:ucode:_00001C00000000000000001000000100127",
          "type": "uri" } ] ] ,
      "http://uidcenter.org/ucr/vocab/event#date": [
        { "value": "2012-03-07T13:00:00+0900",
          "type": "literal" } ] ] ,
      "http://uidcenter.org/ucr/vocab/event#place": [
        { "value": "urn:ucode:_00001C00000000000000001000000100A01",
          "type": "uri" } ] ] } } },
    "remains": false
  }
}

```

[5-2: イベントの新規登録]

<機能概要>

イベントを新規登録する。発生日時が指定されていない場合、現時刻を発生日時とする。

<メソッド>

POST

<URLパス>

/api/v1/events

<制約条件>

イベントの発生源のucodeに対するイベント登録を許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、POSTメソッドのボディ部に、JSONまたはXML形式で格納する。

(A)か(B)のどちらか1種類のみを指定できるが、(A)を推奨する。

[表5-6] イベントの新規登録のリクエストパラメータ (A)

パラメータ名	型	説明
rdf	RDF	RDF/XMLまたはRDF/JSONで記述されたRDFデータ。urn:ucode: _?<val>という形のURIを含めることにより、ucodeの自動発行を要求できる。

[表5-7] イベントの新規登録のリクエストパラメータ (B)

パラメータ名	型	説明
target	xsd:anyURI	登録するイベント識別子
params	struct[]	キーが登録パラメータ名、値が登録値であるリスト。
numofDestination	xsd:integer	イベント発生の結果発行されるucodeの数

・ <rdf> と <params> のどちらか一方は指定されていなければならない。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表5-8] イベントの新規登録のステータスコード

ステータスコード	意味
201 Created	正常終了。
400 Bad Request	パラメータに rdf と params のどちらも指定されていない。 パラメータに rdf と params, target, num のいずれかが指定されている。 <params>のキーが正しくない。
409 Conflict	パラメータに target と num の両方を指定した。 明記したイベント識別子が、すでに情報流通連携基盤システムに登録されている
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

クエリパラメータが(A)の形式であった場合のレスポンスは、以下の表(A)に示す構造データをJSONまたはXML形式で表現したものである。

クエリパラメータが(B)の形であった場合のレスポンスは、以下の表(B)に示す構造データをJSONまたはXML形式で表現したものである。

[表5-9] イベントの新規登録のレスポンスパラメータ (A)

パラメータ名	型	説明
ucode	struct[]	キーが指定された変数名、値が発行されたucodeであるリスト。

[表5-10] イベントの新規登録のレスポンスパラメータ (B)

パラメータ名	型	説明
ucode	xsd:anyURI	新規登録されたイベントのucode。
destination	xsd:anyURI []	このイベントにより発行された対象物のucode。

【APIの利用例】

<クエリ1>

```
POST /api/v1/events HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{"rdf":
{ "ucode_e": {
    "http://uidcenter.org/vocab/ev#source" : [
      { "value": "urn:ucode:_00001C000000000000001000000100124",
        "type": "uri" } ],
    "http://uidcenter.org/vocab/ev#date" : [
      { "value": "2012-03--07T13:00:00+0900", "type": "literal" } ],
    "http://uidcenter.org/vocab/ev#place": [
      { "value": "urn:ucode:_00001C000000000000001000000100A01",
        "type": "uri" } ],
    "http://uidcenter.org/vocab/ev#destination" : [
      { "value": "ucode_d1", "type": "uri" },
      { "value": "ucode_d2", "type": "uri" },
      { "value": "ucode_d3", "type": "uri" } ] } }
```

<レスポンス1>

```
HTTP/1.1 201 Created
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"ucode": {
  "ucode_e": "urn:ucode:_00001C000000000000001000000100801",
  "ucode_d1": "urn:ucode:_00001C000000000000001000000100125",
  "ucode_d2": "urn:ucode:_00001C000000000000001000000100126",
  "ucode_d3": "urn:ucode:_00001C000000000000001000000100127" }
```

<クエリ2>

```
POST /api/v1/events HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{"params":{
  "<ev:type>":"<urn:ucode:_0FFFDE00000000000000000000000001234567>",
  "<ev:source>":"<urn:ucode:_00001C00000000000000000000000001000000100124>",
  "numofDestination":3,
  "<ev:date>":"2012-03-07T13:00:00+0900",
  "<ev:place>":"<urn:ucode:_00001C00000000000000000000000001000000100A01>"}}
```

<レスポンス2>

```
HTTP/1.1 201 Created
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"ucode":"<urn:ucode:_00001C00000000000000000000000001000000100801>",
  "destination":[
    "<urn:ucode:_00001C00000000000000000000000001000000100125>",
    "<urn:ucode:_00001C00000000000000000000000001000000100126>",
    "<urn:ucode:_00001C00000000000000000000000001000000100127>" ] }
```

[5-3: イベントの閲覧]

<機能概要>

イベントを閲覧する。

<メソッド>

GET

<URLパス>

/api/v1/events/<target>

<制約条件>

<target>が指定するイベントに対する情報閲覧を許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。また、クエリパラメータとして?streamを指定できる。このときは、Stream APIに基づくコネクションを指定された秒数継続する。

[表5-11] イベントの閲覧のリクエストパラメータ

パラメータ名	型	説明
target	xsd:anyURI []	イベントの識別子

<ステータスコード>

ステータスコードは以下の表の通りである。

[表5-12] イベントの閲覧のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<target>が指定されていない。
404 Not Found	対象のイベントが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

レスポンスは、以下の表の構造データをJSONまたはXML形式で表現したものである。

[表5-13] イベントの閲覧のレスポンスパラメータ

パラメータ名	型	説明
events	RDF	イベントのリスト。レスポンス形式にXMLを指定した場合、各イベント情報はRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各イベント情報はRDF/JSONで表現される。

【APIの利用例】

<クエリ>

```
GET /api/v1/events/ucode_00001C000000000000001000000100800 HTTP/1.1
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"events": {
  "urn:ucode:_00001C000000000000001000000100800": {
    "http://uidcenter.org/ucr/vocab/event#type": [
      { "value": "urn:ucode:_0FFFDE0000000000000000000000001234567",
        "type": "uri" } ],
    "http://uidcenter.org/ucr/vocab/event#source": [
      { "value": "urn:ucode:_00001C000000000000001000000100123",
        "type": "uri" } ],
```

```

"http://uidcenter.org/ucr/vocab/event#destination": [
  { "value": "urn:ucode:_00001C000000000000001000000100125",
    "type": "uri" } ] ,
"http://uidcenter.org/ucr/vocab/event#date": [
  { "value": "2012-03-07T12:00:00+0900",
    "type": "literal" } ] ,
"http://uidcenter.org/ucr/vocab/event#place": [
  { "value": "urn:ucode:_00001C000000000000001000000100A00",
    "type": "uri" } ] ] ] }

```

――[5-4: イベントの閲覧(プロパティ指定)]――

<機能概要>

プロパティ値を指定してイベントを閲覧する。

<メソッド>

GET

<URLパス>

/api/v1/events/<target>/<property>

<制約条件>

<target>が指定するイベントに対する情報閲覧を許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。また、クエリパラメータとして?stream を指定できる。このときは、Stream API に基づくコネクションを指定された秒数継続する。

[表5-14] イベントの閲覧(プロパティ指定)のリクエストパラメータ

パラメータ名	型	説明
target	xsd:anyURI []	イベントの識別子
property	xsd:anyURI []	プロパティの識別子

<ステータスコード>

ステータスコードは以下の表の通りである。

[表5-15] イベントの閲覧(プロパティ指定)のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<ucode>、<target>が指定されていない。
404 Not Found	対象のイベントが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

指定されたイベントに結びついている属性。イベントの識別子もプロパティの識別子も1 種類である場合は、該当する属性値のリストを返す。そうでない場合は、以下の表に示す構造データをJSON またはXML形式で表現したものである。

[表5-16] イベントの閲覧(プロパティ指定)のレスポンスパラメータ

パラメータ名	型	説明
events	RDF	イベントのリスト。レスポンス形式にXMLを指定した場合、各イベント情報はRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各イベント情報はRDF/JSONで表現される。

【APIの利用例】

<クエリ1>

```
GET /api/v1/events/ucode_00001C000000000000001000000100800/ev_date HTTP/1.1
Host: www.example.org
```

<レスポンス1>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
[ "2012-03-07T13:00:00+0900" ]
```

<クエリ2>

```
GET /api/v1/events/ucode_00001C000000000000001000000100800,
ucode_00001C000000000000001000000100801/ev_date HTTP/1.1
Host: www.example.org
```

<レスポンス2>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: text/plain; charset=utf-8
{"events": {
  "urn:ucode:_00001C000000000000001000000100800": {
    "http://uidcenter.org/ucr/vocab/event#date": [
      { "value": "2012-03-07T12:00:00+0900",
```

```

    "type": "literal" } ] },
    "urn:ucode:_00001C000000000000001000000100801": {
      "http://uidcenter.org/ucr/vocab/event#date": [
        { "value": "2012-03-07T13:00:00+0900",
          "type": "literal" } ] ] ] }

```

――[5-5: イベントの更新]――

<機能概要>

イベントを更新する。

<メソッド>

PUT

<URLパス>

/api/v1/events/<target>

<制約条件>

<target> に対するイベントを更新することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータのうち、更新対象の識別子はURI 部分に指定する。それ以外のパラメータ(以下の表)は、PUTメソッドのボディ部に、JSONまたはXML形式で格納する。

[表5-17] イベントの更新のリクエストパラメータ

パラメータ名	型	説明
rdf	RDF	RDF/XMLまたはRDF/JSON形式で記した更新情報。更新情報のsubject は、<target> と一致していること。コマンド終了後、更新情報に含まれるpredicate に対する値は、個数を含めてリクエストパラメータが指定した情報と完全に一致する。更新情報に含まれないpredicate に関する値は変化しない。
params	struct[]	キーが登録パラメータ名、値が登録値であるリスト。コマンド終了後、キーに指定されたプロパティに対する値は、個数を含めてリクエストパラメータが指定した情報と完全に一致する。リクエストパラメータのキーにないプロパティに関する値は変化しない。

- ・ params またはrdf のいずれか一方のみが指定されなければならない。rdf を推奨する。
- ・ params のキーは、イベントの属性を示すプロパティ値である。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表5-18] イベントの更新のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータ値が正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	対象のイベント識別子が情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ1>

```
PUT /api/v1/events/ucode_00001C000000000000001000000100800 HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{"rdf": {
  "urn:ucode:_00001C000000000000001000000100800": {
    "http://uidcenter.org/ucr/vocab/ug#region": [
      { "value": "urn:ucode:_00001C000000000000001000000100A01",
        "type": "uri" } ] } }
```

<レスポンス1>

```
HTTP/1.1 204 No Content
Connection: close
```

<クエリ2>

```
PUT /api/v1/events/ucode_00001C000000000000001000000100800 HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{"params": {"<ev:place>": "<urn:ucode:_00001C000000000000001000000100A01>"}}
```

<レスポンス2>

```
HTTP/1.1 204 No Content
Connection: close
```

――[5-6: イベントの更新(プロパティ指定)]――

<機能概要>

プロパティ値を指定してイベントを更新する。

<メソッド>

PUT

<URLパス>

/api/v1/events/<target>/<property>

<制約条件>

<target> に対するイベントを更新することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータのうち、更新対象の識別子とプロパティ値はURL 部分に指定する。更新するパラメータ値は、PUTメソッドのボディ部に格納する。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表5-19] イベントの更新(プロパティ指定)のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータ値が正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	対象のイベント識別子が情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
PUT /api/v1/events/ucode_00001C000000000000001000000100800/ev_place HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
[ "<urn:ucode:_00001C000000000000001000000100A01>" ]
```

<レスポンス>

```
HTTP/1.1 204 No Content
Connection: close
```

——[5-7: イベントの削除]——

<機能概要>

イベントを削除する。

<メソッド>

DELETE

<URLパス>

/api/v1/events/<target>

<制約条件>

<target> に対するイベントを削除することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>

削除対象のイベント識別子はURI 部分に指定する。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表5-20] イベントの削除のステータスコード

ステータスコード	意味
204 No Content	正常終了。
403 Forbidden	アクセス権限がない。
404 Not Found	対象のイベント識別子が情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

DELETE /api/v1/events/ucode_00001C000000000000001000000100800 HTTP/1.1
Host: www.example.org
Content-Length: xxx

<レスポンス>

HTTP/1.1 204 No Content
Connection: close

――[5-8: イベントの削除(プロパティ指定)]――

<機能概要>

プロパティ値を指定してイベントを削除する。指定したプロパティ以外のイベント情報は残る。

<メソッド>

DELETE

<URLパス>

/api/v1/events/<target>/<property>

<制約条件>

targetに対するイベントを削除することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>

削除対象のイベント識別子とプロパティはURI 部分に指定する。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表5-21] イベントの削除(プロパティ指定)のステータスコード

ステータスコード	意味
204 No Content	正常終了。
403 Forbidden	アクセス権限がない。
404 Not Found	対象のイベント識別子が情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
DELETE /api/v1/events/ucode_00001C00000000000001000000100800/ev_date
HTTP/1.1
Host: www.example.org
Content-Length: xxx
```

<レスポンス>

```
HTTP/1.1 204 No Content
Connection: close
```

＜機能概要＞

指定したtarget を起点とするトレースフォワード/トレースバックを実施し、その結果得られたイベントのリストを返す。

＜メソッド＞

GET

＜URLパス＞

/api/v1/trace/<target>?direction=<direction>&limit=<limit>

＜制約条件＞

<target>が指定するオープンデータまたはイベントに対する情報閲覧を許可されているユーザによるアクセスであること。

＜リクエストパラメータ＞

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。

[表5-22] トレースの実施のリクエストパラメータ

パラメータ名	型	説明
target	xsd:anyURI	トレースの起点となるucode。(イベントまたはイベント対象物ucode)
direction	xsd:string	トレースのパラメータ。forward:(トレースフォワード)またはback:(トレースバック)の値をとる。省略時はforward。
limit	xsd:integer	トレースする階層数。省略時は1階層。

＜ステータスコード＞

ステータスコードは以下の表の通りである。

[表5-23] トレースの実施のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<target>が指定されていない。
404 Not Found	対象のオープンデータまたはイベントが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

＜レスポンス＞

レスポンスは、以下の表に示す構造データをJSONまたはXML形式で表現したものである。

【APIの利用例】

<クエリ>

```
GET /api/v1/trace/ucode_00001C0000000000000001000000100800?  
direction=forward&limit=2 HTTP/1.1  
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK  
Content-Length: xxx  
Connection: close  
Content-Type: application/json; charset=utf-8  
{  
  "events": [  
    {  
      "urn:ucode:_00001C0000000000000001000000100800": {  
        "http://uidcenter.org/ucr/vocab/event#type": [  
          { "value": "urn:ucode:_OFFFDE000000000000000000000001234567",  
            "type": "uri" } ],  
        "http://uidcenter.org/ucr/vocab/event#source": [  
          { "value": "urn:ucode:_00001C0000000000000001000000100123",  
            "type": "uri" } ],  
        "http://uidcenter.org/ucr/vocab/event#destination": [  
          { "value": "urn:ucode:_00001C0000000000000001000000100124",  
            "type": "uri" },  
          { "value": "urn:ucode:_00001C0000000000000001000000100125",  
            "type": "uri" } ] ,  
        "http://uidcenter.org/ucr/vocab/event#date": [  
          { "value": "2012-03-07T12:00:00+0900",  
            "type": "literal" } ] ,  
        "http://uidcenter.org/ucr/vocab/event#place": [  
          { "value": "urn:ucode:_00001C0000000000000001000000100A00",  
            "type": "uri" } ] ] },  
    {  
      "urn:ucode:_00001C0000000000000001000000100801": {  
        "http://uidcenter.org/ucr/vocab/event#type": [  
          { "value": "urn:ucode:_OFFFDE000000000000000000000001234567",  
            "type": "uri" } ],  
        "http://uidcenter.org/ucr/vocab/event#source": [  
          { "value": "urn:ucode:_00001C0000000000000001000000100125",  
            "type": "uri" } ],  
        "http://uidcenter.org/ucr/vocab/event#destination": [  
          { "value": "urn:ucode:_00001C0000000000000001000000100126",  
            "type": "uri" },  
          { "value": "urn:ucode:_00001C0000000000000001000000100127",  
            "type": "uri" } ] ,  
        "http://uidcenter.org/ucr/vocab/event#date": [  
          { "value": "2012-03-07T13:00:00+0900",  
            "type": "literal" } ] ,  
        "http://uidcenter.org/ucr/vocab/event#place": [  
          { "value": "urn:ucode:_00001C0000000000000001000000100A01",  
            "type": "uri" } ] ] }  
  ]  
}
```


6. Geographical Data Management Command

公共交通情報流通連携基盤システム

Geographical Data Management Commandは、GIS 等の地理情報処理に必要な機能を提供するコマンドである。場所や地図は、基本的にucodeで識別する。
場所や地図が示す基本的な領域(幾何データ)は、それを識別するucodeとプロパティug:region を結びつけて表現する。このプロパティの値は、以下の表のいずれかとする。データタイプ指定を省略した場合は、Well Known Text が指定されたものと見なす。

[表6-1] 幾何データの記述形式

データ形式	指定するデータタイプ
Well_Known_Text	ogc:wktLiteral
GML	ogc:gmlLiteral
GeoJSON	ug:GeoJSONLiteral

たとえば、<urn:ucode: 00001C00000000000000001000000100800> で示されるURI の指す場所が北緯35.67 度、東経139.76度の一点であることをNotation3 形式(prefix 宣言を省略する)で記述すると以下ようになる。

- ・ 幾何データにWell Known Text 形式を利用する場合
 <urn:ucode: 00001C00000000000000001000000100800> ug:region
 "<http://www.opengis.net/def/crs/OGC/1.3/CRS84> Point(35.67
 139.76)"^^ogc:wktLiteral .
- ・ 幾何データにGML 形式を利用する場合
 <urn:ucode: 00001C00000000000000001000000100800> ug:region "<gml:Point
 srsName=¥"http://www.opengis.net/def/crs/OGC/1.3/CRS84¥">
 <gml:coordinates>139.76 35.67</gml:coordinates>
 </gml:Point> ¥"^^ogc:gmlLiteral .
- ・ 幾何データにGeoJSON形式を利用する場合
 <urn:ucode: 00001C00000000000000001000000100800> ug:region
 "{¥"type¥":¥"Point¥", ¥"coordinates¥": ¥"[139.76 35.67]¥"
 }"^^ug:geoJSONLiteral .

本コマンドの一覧を以下の表に示す。

[表6-2] Geographical Data Management Command の一覧

URLパス	HTTPメソッド	意味
/api/v1/places	GET	場所情報を検索する
/api/v1/places	POST	場所情報を登録する
/api/v1/places/<target>	GET	場所情報を閲覧する
/api/v1/places/<target>/<property>	GET	場所情報を閲覧する
/api/v1/places/<target>	PUT	場所情報を更新する
/api/v1/places/<target>/<property>	PUT	場所情報を更新する
/api/v1/places/<target>	DELETE	場所情報を削除する
/api/v1/places/<target>/<property>	DELETE	場所情報を削除する
/api/v1/places/<target>/ug_consistsOf	PUT	場所情報の包含関係を移設する
/api/v1/maps	GET	地図情報を検索する
/api/v1/maps	POST	地図情報を登録する
/api/v1/maps/<target>	GET	地図情報を閲覧する
/api/v1/maps/<target>/<property>	GET	地図情報を閲覧する
/api/v1/maps/<target>	PUT	地図情報を更新する
/api/v1/maps/<target>/<property>	PUT	地図情報を更新する
/api/v1/maps/<target>	DELETE	地図情報を削除する
/api/v1/maps/<target>/<property>	DELETE	地図情報を削除する

[6-1: 場所情報の検索]

<機能概要>

場所情報を検索する。

<メソッド>

GET

<URLパス>

/api/v1/places?<param1>=<value1>&<param2>=<value2>

<制約条件>

なし。誰でもリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GET メソッドのURL部に格納する。複数のパラメータを指定した場合、AND 検索となる。

[表6-3] 場所情報の検索のリクエストパラメータ

パラメータ名	デフォルト値	説明
paramN	(指定なし)	検索対象パラメータ名
valueN	(指定なし)	検索対象パラメータ値

・ <paramN>、<valueN> の組を1個以上指定しなければならない。<paramN> は以下のいずれかである。複数指定した場合は、AND 検索となる。

1. target。検索対象の場所の識別子であり、パラメータ値の型はxsd:anyURI[] である。URI内のカンマはURLエンコードすること。target が複数ある場合は、カンマで区切る。
2. predicate。4. または5. の値を目的語に持つ述語を指定する場合に使用する。パラメータ値の型はxsd:anyURI である。省略時はug:region が指定されたと見なす。
3. offset, limit。パラメータ値はxsd:integer である。検索結果のうちoffset番目からlimit個分を要求することを意味する。次項4 のパラメータを指定した場合は、指定した1点から近い順に並べたときのoffset番目からlimit個分を要求することを意味する。
4. 1点と半径を指定し、それが示す円内に含まれる場所の識別子を検索する場合に使用する。パラメータは以下の表の組み合わせである。
5. ポリゴン等の形状を指定し、それに含まれる/含む/重なる場所の識別子を検索する場合に使用する。パラメータは以下の表のうち1 つである。パラメータ値はOpen Geospatial Consortiumが規定するWell Known Text(WKT)であること。
6. geo format。幾何データのデータタイプを指定する。このパラメータを省略した場合のデータタイプはogc:wktLiteral(Well Known Text 形式)である。
7. 地物属性を示すプロパティURI。

[表6-4] 場所情報の検索のリクエストパラメータ

パラメータ名	型	説明
lat	xsd:double	WGS84での緯度。省略できない。
lon	xsd:double	WGS84での経度。省略できない。
floor	xsd:double[]	階数。下限値と上限値をカンマで区切って指定する。下限値と上限値が等しい場合は省略した場合は指定なし。(altとどちらか1つのみ指定)
alt	xsd:double[]	高度[m]。下限値と上限値をカンマで区切って指定する。省略した場合は指定なし。(floor とどちらか1つのみ指定)
radius	xsd:double	検索半径[m]。省略できない。

[表6-5] 場所情報の検索のリクエストパラメータ

パラメータ名	型	説明
intersect	xsd:string(WKT)	パラメータ値が指定する領域と重なりを持つ
within	xsd:string(WKT)	パラメータ値が指定する領域に完全に含まれる
contains	xsd:string(WKT)	パラメータ値が指定する領域を完全に含む

<ステータスコード>

ステータスコードは以下の表の通りである。

[表6-6] 場所情報の検索のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<param1>、<value1>の組がない。<paramN>が正しくない。
404 Not Found	検索条件を満たす場所情報が情報流通連携基盤システムに登録されていない。
413 Request Entity Too Large	limit値が大きすぎる。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

レスポンスは、以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表6-7] 場所情報の検索のレスポンスパラメータ

パラメータ名	型	説明
place	RDF	場所のリスト。レスポンス形式にXMLを指定した場合、各場所情報はRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各場所情報はRDF/JSONで表現される。
remains	xsd:boolean	条件にマッチするが、レスポンスに含まれない場所がある場合はtrue、ない場合はfalse。

【APIの利用例】

<クエリ>

```
GET /api/v1/places?within=POLYGON((0 0) (2 0) (2 2) (0 2)) HTTP/1.1
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{ "places":
  { "urn:ucode:_00001C000000000000000010000000100800" : {
```

——[6-2: 場所情報の新規登録]——

場所情報を新規登録する。

POST

```
/api/v1/places
```

なし。誰でもリクエストできる。

リクエストパラメータ(以下の表)は、POSTメソッドのボディ部に、JSONまたはXML形式で格納する。
(A)か(B)のどちらか1種類のみを指定できるが、(A)を推奨する。

パラメータ名	型	説明
rdf	RDF	RDF/XMLまたはRDF/JSONで記述されたRDFデータ。urn:ucode:_?<val>という形のURIを含めることにより、ucodeの自動発行を要求できる。

[表6-9] 場所情報の新規登録のリクエストパラメータ (B)

パラメータ名	型	説明
target	xsd:anyURI []	場所情報の識別子
params	struct []	キーがプロパティのURI、値が登録値であるリスト。
num	xsd:integer	新規発行するucodeの個数。省略時は1。

- ・ <target> または<num> の一方のみを指定できる。
- ・ <rdf> と<params> のどちらか一方は指定されていなければならない。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表6-10] 場所情報の新規登録のステータスコード

ステータスコード	意味
201 Created	正常終了。
400 Bad Request	パラメータに rdf と params のどちらも指定されていない。 パラメータに rdf と params, target, num のいずれかが指定されている。 <params>のキーが正しくない。 パラメータに target と num の両方を指定した。
409 Conflict	指定した場所情報の識別子は、すでに情報流通連携基盤システムに登録されている。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

クエリパラメータが(A)の形式であった場合のレスポンスは、以下の表(A)に示す構造データをJSONまたはXML形式で表現したものである。

クエリパラメータが(B)の形であった場合のレスポンスは、以下の表(B)に示す構造データをJSONまたはXML形式で表現したものである。

[表6-11] 場所情報の新規登録のレスポンスパラメータ (A)

パラメータ名	型	説明
ucode	struct	キーが指定された変数名、値が発行されたucodeのURI表現であるリスト。

[表6-12] 場所情報の新規登録のレスポンスパラメータ (B)

パラメータ名	型	説明
ucode	xsd:anyURI []	作成された場所情報の識別ucode。

【APIの利用例】

<クエリ1>

```
POST /api/v1/places HTTP/1.1
Content-Length: xxx
Content-Type: application/json; charset=utf-8
Host: www.example.org
{"rdf":
{ "ucode_x": {
"http://uidcenter.org/vocab/ug#region" : [
{ "value": "POINT(1 1)", "type": "literal" } ] } } }
```

<レスポンス1>

```
HTTP/1.1 201 Created
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"ucode": {
"ucode_x": "urn:ucode:_00001C00000000000000001000000100801" } }
```

<クエリ2>

```
POST /api/v1/places HTTP/1.1
Content-Length: xxx
Content-Type: application/json; charset=utf-8
Host: www.example.org
{"params": {
"<ug:type>": "<urn:ucode:_0FFFDE00000000000000000001234567>",
"<ug:region>": "POINT(1 1)" }}
```

<レスポンス2>

```
HTTP/1.1 201 Created
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"ucode": ["<urn:ucode:_00001C00000000000000001000000100801>"]}
```

——[6-3: 場所情報の閲覧]——

＜機能概要＞

場所情報を閲覧する。

＜メソッド＞

GET

＜URLパス＞

/api/v1/places/<target>

＜制約条件＞

<target>が指定する場所情報の閲覧を許可されているユーザによるアクセスであること。

＜リクエストパラメータ＞

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。

[表6-13] 場所情報の閲覧のリクエストパラメータ

パラメータ名	型	説明
target	xsd:anyURI []	場所情報の識別子

＜ステータスコード＞

ステータスコードは以下の表の通りである。

[表6-14] 場所情報の閲覧のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<target>が指定されていない。
404 Not Found	対象の場所情報が見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

＜レスポンス＞

以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表6-15] 場所情報の閲覧のレスポンスパラメータ

パラメータ名	型	説明
place	RDF	指定された場所情報のリスト。レスポンス形式にXMLを指定した場合、各場所情報はRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各場所情報はRDF/JSONで表現される。

【APIの利用例】

<クエリ>

```
GET /api/v1/places/ucode_00001C000000000000001000000100800 HTTP/1.1
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"places": {
  "urn:ucode:_00001C000000000000001000000100800" : {
    "http://uidcenter.org/vocab/ug#type" : [
      { "value" : "urn:ucode:_0FFFDE00000000000000000000000001234567",
        "type" : "uri" } ],
    "http://uidcenter.org/vocab/ug#region" : [
      { "value" : "POINT(1 1)", "type" : "literal" } ] } ] }
```

――[6-4: 場所情報の閲覧(プロパティ指定)]――

<機能概要>

プロパティを指定して、場所情報を閲覧する。

<メソッド>

GET

<URLパス>

/api/v1/places/<target>/<property>

<制約条件>

<target>が指定する場所情報の閲覧を許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。

[表6-16] 場所情報の閲覧(プロパティ指定)のリクエストパラメータ

パラメータ名	型	説明
target	xsd:anyURI []	場所情報の識別子
property	xsd:anyURI []	プロパティ値

<ステータスコード>

ステータスコードは以下の表の通りである。

[表6-17] 場所情報の閲覧(プロパティ指定)のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<target>、<property>が指定されていない。
404 Not Found	対象の場所情報が見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

指定されたプロパティをもつ目的語値。場所の識別子もプロパティの識別子も1種類である場合は、該当する目的語値のリストを返す。そうでない場合は、以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表6-18] 場所情報の閲覧(プロパティ指定)のレスポンスパラメータ

パラメータ名	型	説明
place	RDF	指定された場所情報のリスト。レスポンス形式にXMLを指定した場合、各場所情報はRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各場所情報はRDF/JSONで表現される。

【APIの利用例】

<クエリ1>

```
GET /api/v1/places/ucode_00001C000000000000001000000100800/ug_region
HTTP/1.1
Host: www.example.org
```

<レスポンス1>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
[ "POINT(1 1)" ]
```

<クエリ2>

```
GET /api/v1/places/ucode_00001C000000000000001000000100800,
ucode_00001C000000000000001000000100801/ug_region HTTP/1.1
Host: www.example.org
```

<レスポンス2>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"places": {
  "urn:ucode:_00001C000000000000001000000100800": {
    "http://uidcenter.org/vocab/ug#region": [
      { "value": "POINT(1 1) ", "type": "literal" } ] },
  "urn:ucode:_00001C000000000000001000000100801": {
    "http://uidcenter.org/vocab/ug#region": [
      { "value": "POINT(1.5 1.5) ", "type": "literal" } ]}}
```

[6-5: 場所情報の更新]

<機能概要>

場所情報を更新する。

<メソッド>

PUT

<URLパス>

/api/v1/places/<target>

<制約条件>

<target> に対する場所情報を更新することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータ(以下の表)のうち、更新対象の識別子はURI 部分に指定する。それ以外のパラメータは、PUTメソッドのボディ部に、JSONまたはXML形式で格納する。

[表6-19] 場所情報の更新のリクエストパラメータ

パラメータ名	型	説明
rdf	RDF	RDF/XMLまたはRDF/JSON形式で記した更新情報。更新情報のsubject は、<target> と一致していること。コマンド終了後、リクエストパラメータに含まれるpredicateに対する値は、個数を含めてリクエストパラメータが指定した情報と完全に一致する。
params	struct[]	キーが登録パラメータ名、値が登録値であるリスト。コマンド終了後、キーに指定されたプロパティに対する値は、個数を含めてリクエストパラメータが指定した情報と完全に一致する。リクエストパラメータのキーにないプロパティに関する値は変化しない。

- ・ params または rdf のいずれか一方のみが指定されなければならない。 rdf を推奨する。
- ・ params のキーは、場所情報の属性を示すプロパティ値である。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表6-20] 場所情報の更新のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータ値が正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	対象の場所情報識別子が情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ1>

```
PUT /api/v1/places/ucode_00001C000000000000001000000100800 HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{"rdf": {
  "urn:ucode:_00001C000000000000001000000100800": {
    "http://uidcenter.org/ucr/vocab/ug#region": [
      { "value": "POINT(1 1)",
        "type": "uri" } ] ] }
```

<レスポンス1>

HTTP/1.1 204 No Content
Connection: close

<クエリ2>

PUT /api/v1/places/ucode_00001C00000000000001000000100800 HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{"params":{"<ug:region>":"POINT(1 1)"}}

<レスポンス2>

HTTP/1.1 204 No Content
Connection: close

[6-6: 場所情報の更新(プロパティ指定)]

<機能概要>

プロパティを指定して、場所情報を更新する。

<メソッド>

PUT

<URLパス>

/api/v1/places/<target>/<property>

<制約条件>

<target> に対する場所情報を更新することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータのうち、更新対象の識別子とプロパティはURI 部分に指定する。
更新する値は、PUTメソッドのボディ部に、JSONまたはXML形式で格納する。
本コマンド終了後、更新対象の識別子URI とプロパティをもつオブジェクトは、リクエストパラメータに指定した値のみとなる。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表6-21] 場所情報の更新(プロパティ指定)のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータ値が正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	対象の場所情報識別子が情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
PUT /api/v1/places/ucode_00001C000000000000001000000100800/ug_region
HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
[ "POINT(1 1)" ]
```

<レスポンス>

```
HTTP/1.1 204 No Content
Connection: close
Content-Type: application/json; charset=utf-8
```

――[6-7: 場所情報の削除]――

<機能概要>

場所情報を削除する。

<メソッド>

DELETE

<URLパス>

/api/v1/places/<target>

<制約条件>

<target> に対する場所情報を削除することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>

削除対象の識別子はURI 部分に指定する。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表6-22] 場所情報の削除のステータスコード

ステータスコード	意味
204 No Content	正常終了。
403 Forbidden	アクセス権限がない。
404 Not Found	対象の場所情報識別子が情報流通連携基盤システムに登録されていない。
409 Conflict	この場所情報識別子を包含・同値・隣接等の対象にしている場所情報が登録されているため、削除できない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

DELETE /api/v1/places/ucode_00001C000000000000001000000100800 HTTP/1.1

Host: www.example.org

Content-Length: xxx

<レスポンス>

HTTP/1.1 204 No Content

Connection: close

――[6-8: 場所情報の属性削除]――

<機能概要>

プロパティを指定して、場所情報の属性を削除する。指定したプロパティ以外の場所情報は残る。

<メソッド>

DELETE

<URLパス>

/api/v1/places/<target>/<property>

<制約条件>

<target> に対する場所情報を更新することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>

削除対象の識別子とプロパティはURI 部分に指定する。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表6-23] 場所情報の属性削除のステータスコード

ステータスコード	意味
204 No Content	正常終了。
403 Forbidden	アクセス権限がない。
404 Not Found	対象の場所情報識別子が情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
DELETE /api/v1/places/ucode_00001C000000000000001000000100800/ug_region
HTTP/1.1
Host: www.example.org
Content-Length: xxx
```

<レスポンス>

```
HTTP/1.1 204 No Content
Connection: close
```

――[6-9: 場所情報の包含関係移設]――

<機能概要>

場所情報の包含関係を移設する。これは、「3.3.6 場所情報の更新(プロパティ指定)」の特殊ケースである。

<メソッド>

PUT

<URLパス>

/api/v1/places/<target>/ug_consistsOf

<制約条件>

<target> に対する場所情報を更新することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータのうち、更新対象の識別子はURI 部分に指定する。

更新する値は、PUTメソッドのボディ部に格納する。

本コマンドの終了後、<target> に含まれる(ug:consistsOf の関係をもつ)場所の識別子は、個数を含めてリクエストパラメータに指定した値と完全に一致する。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表6-24] 場所情報の包含関係移設のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータ値が正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	対象の場所情報識別子が情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
PUT /api/v1/places/ucode_00001C000000000000001000000100800/ug_conconsistsOf
HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
[ "<urn:ucode:_00001C000000000000001000000100900>",
  "<urn:ucode:_00001C000000000000001000000100901>" ]
```

<レスポンス>

```
HTTP/1.1 204 No Content
Connection: close
```

——[6-10: 地図情報の検索]——

<機能概要>

地図情報を検索する。

<メソッド>

GET

<URLパス>

/api/v1/maps?<param1>=<value1>&<param2>=<value2>

<制約条件>

なし。誰でもリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。複数のパラメータを指定した場合、AND 検索となる。

[表6-25] 地図情報の検索のリクエストパラメータ

パラメータ名	デフォルト値	説明
paramN	(指定なし)	検索対象パラメータ名
valueN	(指定なし)	検索対象パラメータ値

・ <paramN>、<valueN> の組を1個以上指定しなければならない。<paramN> は以下のいずれかである。複数指定した場合は、AND 検索となる。

1. target。検索対象の場所の識別子であり、パラメータ値の型はxsd:anyURI[] である。URI内のカンマはURLエンコードすること。target が複数ある場合は、カンマで区切る。
2. predicate。4. または5. の値を目的語に持つ述語を指定する場合に使用する。パラメータ値の型はxsd:anyURI である。省略時はug:region が指定されたと見なす。
3. offset, limit。パラメータ値はxsd:integer である。検索結果のうちoffset番目からlimit個分を要求することを意味する。次項4 のパラメータを指定した場合は、指定した1点から近い順に並べたときのoffset番目からlimit個分を要求することを意味する。
4. 1点と半径を指定し、それが示す円内に含まれる場所の識別子を検索する場合に使用する。パラメータは以下の表の組み合わせである。
5. ポリゴン等の形状を指定し、それに含まれる/含む/重なる場所の識別子を検索する場合に使用する。パラメータは以下の表のうち1 つである。パラメータ値はOpen Geospatial Consortium が規定するWell Known Text(WKT)であること。
6. geo format。幾何データのデータタイプを指定する。このパラメータを省略した場合のデータタイプはogc:wktLiteral(Well Known Text 形式)である。
7. 地物属性を示すプロパティURI。

[表6-26] 地図情報の検索のリクエストパラメータ

パラメータ名	型	説明
lat	xsd:double	WGS84での緯度。省略できない。
lon	xsd:double	WGS84での経度。省略できない。
floor	xsd:double[]	階数。下限値と上限値をカンマで区切って指定する。下限値と上限値が等しい場合は省略した場合は指定なし。(altとどちらか1つのみ指定)
alt	xsd:double[]	高度[m]。下限値と上限値をカンマで区切って指定する。省略した場合は指定なし。(floor とどちらか1つのみ指定)
radius	xsd:double	検索半径[m]。省略できない。

[表6-27] 地図情報の検索のリクエストパラメータ

パラメータ名	型	説明
intersect	xsd:string(WKT)	パラメータ値が指定する領域と重なりを持つ
within	xsd:string(WKT)	パラメータ値が指定する領域に完全に含まれる
contains	xsd:string(WKT)	パラメータ値が指定する領域を完全に含む

<ステータスコード>

ステータスコードは以下の表の通りである。

[表6-28] 地図情報の検索のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<param1>、<value1>の組がない。<paramN>が正しくない。
404 Not Found	検索条件を満たす地図情報が情報流通連携基盤システムに登録されていない。
413 Request Entity Too Large	limit値が大きすぎる。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

レスポンスは、以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表6-29] 地図情報の検索のレスポンスパラメータ

パラメータ名	型	説明
maps	RDF	地図のリスト。レスポンス形式にXMLを指定した場合、各地図情報はRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各地図情報はRDF/JSONで表現される。
remains	xsd:boolean	条件にマッチするが、レスポンスに含まれない地図がある場合はtrue、ない場合はfalse。

【APIの利用例】

<クエリ>

```
GET /api/v1/maps?within=POLYGON((0 0) (2 0) (2 2) (0 2)) HTTP/1.1
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"maps": {
  "urn:ucode:_00001C000000000000001000000100800" : {
```

——[6-11: 地図情報の新規登録]——

地図情報を新規登録する。

POST

```
/api/v1/maps
```

なし。誰でもリクエストできる。

リクエストパラメータ(以下の表)は、POSTメソッドのボディ部に、JSONまたはXML形式で格納する。
(A)か(B)のどちらか1種類のみを指定できるが、(A)を推奨する。

パラメータ名	型	説明
rdf	RDF	RDF/XMLまたはRDF/JSONで記述されたRDFデータ。urn:ucode:_?<val>という形のURIを含めることにより、ucodeの自動発行を要求できる。

[表6-31] 地図情報の新規登録のリクエストパラメータ (B)

パラメータ名	型	説明
target	xsd:anyURI []	地図情報の識別子
params	struct []	キーがプロパティのURI、値が登録値であるリスト。
num	xsd:integer	新規発行するucodeの個数。省略時は1。

- ・ <target> または<num> の一方のみを指定できる。
- ・ <rdf> と<params> のどちらか一方は指定されていなければならない。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表6-32] 地図情報の新規登録のステータスコード

ステータスコード	意味
201 Created	正常終了。
400 Bad Request	パラメータに rdf と params のどちらも指定されていない。 パラメータに rdf と params, target, num のいずれかが指定されている。 <params>のキーが正しくない。
409 Conflict	パラメータに target と num の両方を指定した。 指定した地図情報の識別子は、すでに情報流通連携基盤システムに登録されている。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

クエリパラメータが(A)の形式であった場合のレスポンスは、以下の表(A)に示す構造データをJSONまたはXML形式で表現したものである。

クエリパラメータが(B)の形であった場合のレスポンスは、以下の表(B)に示す構造データをJSONまたはXML形式で表現したものである。

[表6-33] 地図情報の新規登録のレスポンスパラメータ (A)

パラメータ名	型	説明
ucode	struct	キーが指定された変数名、値が発行されたucodeのURI表現であるリスト。

[表6-34] 地図情報の新規登録のレスポンスパラメータ (B)

パラメータ名	型	説明
ucode	xsd:anyURI []	作成された地図情報の識別ucode。

【APIの利用例】

<クエリ1>

```
POST /api/v1/maps HTTP/1.1
Content-Length: xxx
Content-Type: application/json; charset=utf-8
Host: www.example.org
{"rdf": {
  "ucode_?x": {
    "http://uidcenter.org/vocab/ug#region" : [
      { "value": "POINT(1 1)", "type": "literal" } ] } }
```

<レスポンス1>

```
HTTP/1.1 201 Created
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"ucode":{
  "ucode_?x":"urn:ucode:_00001C0000000000000001000000100801"}}
```

<クエリ2>

```
POST /api/v1/maps HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{"params":{
  "<ug:type>":"<urn:ucode:_0FFFDE00000000000000000000000001234567>",
  "<ug:region>":"POINT(1 1)" }}
```

<レスポンス2>

```
HTTP/1.1 201 Created
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"ucode": ["<urn:ucode:_00001C0000000000000001000000100801>"]}
```

——[6-12: 地図情報の閲覧]——

<機能概要>

地図情報を閲覧する。

<メソッド>

GET

<URLパス>

/api/v1/maps/<target>

<制約条件>

<target>が指定する地図情報の閲覧を許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。

[表6-35] 地図情報の閲覧のリクエストパラメータ

パラメータ名	型	説明
target	xsd:anyURI []	地図情報の識別子

<ステータスコード>

ステータスコードは以下の表の通りである。

[表6-36] 地図情報の閲覧のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<target>が指定されていない。
404 Not Found	対象の地図情報が見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表6-37] 地図情報の閲覧のレスポンスパラメータ

パラメータ名	型	説明
maps	RDF	指定された地図情報のリスト。レスポンス形式にXMLを指定した場合、各地図情報はRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各地図情報はRDF/JSONで表現される。

【APIの利用例】

<クエリ>

```
GET /api/v1/maps/ucode_00001C0000000000000001000000100800 HTTP/1.1
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"maps": {
  "urn:ucode:_00001C0000000000000001000000100800" : {
    "http://uidcenter.org/vocab/ug#type" : [
      { "value" : "urn:ucode:_OFFFDE0000000000000000000000000001234567",
        "type" : "uri" } ],
    "http://uidcenter.org/vocab/ug#region" : [
      { "value" : "POINT(1 1)", "type" : "literal" } ] } ] }
```

――[6-13: 地図情報の閲覧(プロパティ指定)]――

<機能概要>

プロパティを指定して、地図情報を閲覧する。

<メソッド>

GET

<URLパス>

/api/v1/maps/<target>/<property>

<制約条件>

<target>が指定する地図情報の閲覧を許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。

[表6-38] 地図情報の閲覧(プロパティ指定)のリクエストパラメータ

パラメータ名	型	説明
target property	xsd:anyURI [] xsd:anyURI []	地図情報の識別子 プロパティ値

<ステータスコード>

ステータスコードは以下の表の通りである。

[表6-39] 地図情報の閲覧(プロパティ指定)のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<target>、<property>が指定されていない。
404 Not Found	対象の地図情報が見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

指定されたプロパティをもつ目的語値。場所の識別子もプロパティの識別子も1種類である場合は、該当する目的語値のリストを返す。そうでない場合は、以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表6-40] 地図情報の閲覧(プロパティ指定)のレスポンスパラメータ

パラメータ名	型	説明
places	RDF	指定された地図情報のリスト。レスポンス形式にXMLを指定した場合、各地図情報はRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各地図情報はRDF/JSONで表現される。

【APIの利用例】

<クエリ1>

```
GET /api/v1/maps/ucode_00001C000000000000001000000100800/ug:region HTTP/1.1
Host: www.example.org
```

<レスポンス1>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{[ "POINT(1 1)" ]}
```

<クエリ2>

```
GET /api/v1/maps/ucode_00001C00000000000000001000000100800,  
ucode_00001C00000000000000001000000100801/ug_region HTTP/1.1  
Host: www.example.org
```

<レスポンス2>

```
HTTP/1.1 200 OK  
Content-Length: xxx  
Connection: close  
Content-Type: application/json; charset=utf-8  
{  
  "maps": {  
    "urn:ucode:_00001C00000000000000001000000100800": {  
      "http://uidcenter.org/vocab/ug#region" : [  
        { "value" : "POINT(1 1)", "type" : "literal" } ] },  
    "urn:ucode:_00001C00000000000000001000000100801": {  
      "http://uidcenter.org/vocab/ug#region" : [  
        { "value" : "POINT(1.5 1.5)", "type" : "literal" } ] }  
  }  
}
```

[6-14: 地図情報の更新]

<機能概要>

地図情報を更新する。

<メソッド>

PUT

<URLパス>

/api/v1/maps/<target>

<制約条件>

<target> に対する地図情報を更新することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータのうち、更新対象の識別子はURI 部分に指定する。それ以外のパラメータは、PUTメソッドのボディ部に、JSONまたはXML形式で格納する。

[表6-41] 地図情報の更新のリクエストパラメータ

パラメータ名	型	説明
rdf	RDF	RDF/XMLまたはRDF/JSON形式で記した更新情報。更新情報のsubject は、<target> と一致していること。コマンド終了後、更新情報に含まれるpredicate に対する値は、個数を含めてリクエストパラメータが指定した情報と完全に一致する。更新情報に含まれないpredicate に関する値は変化しない。
params	struct[]	キーが登録パラメータ名、値が登録値であるリスト。コマンド終了後、キーに指定されたプロパティに対する値は、個数を含めてリクエストパラメータが指定した情報と完全に一致する。リクエストパラメータのキーにないプロパティに関する値は変化しない。

- ・ rdf またはparams のどちらか1つのみが指定されなければならない。rdf を推奨する。
- ・ params のキーは、地図情報の属性を示すプロパティ値である。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表6-42] 地図情報の更新のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータ値が正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	対象の地図情報識別子が情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ1>

```
PUT /api/v1/places/ucode_00001C000000000000001000000100800 HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{"rdf": {
  "urn:ucode:_00001C000000000000001000000100800": {
    "http://uidcenter.org/ucr/vocab/ug#region": [
      { "value": "POINT(1 1)",
        "type": "uri" } ] ] }
```

<レスポンス1>

HTTP/1.1 204 No Content
Connection: close

<クエリ2>

PUT /api/v1/maps/ucode_00001C000000000000001000000100800 HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{"params":{"<ug:region>":"POINT(1 1)"}}

<レスポンス2>

HTTP/1.1 204 No Content
Connection: close
Content-Type: application/json; charset=utf-8

——[6-15: 地図情報の更新(プロパティ指定)]——

<機能概要>

プロパティを指定して、地図情報を更新する。

<メソッド>

PUT

<URLパス>

/api/v1/maps/<target>/<property>

<制約条件>

<target> に対する地図情報を更新することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータのうち、更新対象の識別子とプロパティはURI 部分に指定する。更新する値は、PUTメソッドのボディ部に格納する。

本コマンド終了後、更新対象の識別子URI とプロパティをもつオブジェクトは、リクエストパラメータに指定した値のみとなる。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表6-43] 地図情報の更新(プロパティ指定)のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータ値が正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	対象の地図報識別子が情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
PUT /api/v1/maps/ucode_00001C0000000000000001000000100800/ug¥_region HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
[ "POINT(1 1)" ]
```

<レスポンス>

```
HTTP/1.1 204 No Content
Connection: close
Content-Type: application/json; charset=utf-8
```

――[6-16: 地図情報の削除]――

<機能概要>

地図情報を削除する。

<メソッド>

DELETE

<URLパス>

/api/v1/maps/<target>

<制約条件>

<target> に対する地図情報を削除することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>

削除対象の識別子はURI 部分に指定する。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表6-44] 地図情報の削除のステータスコード

ステータスコード	意味
204 No Content	正常終了。
403 Forbidden	アクセス権限がない。
404 Not Found	対象の地図情報識別子が情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
DELETE /api/v1/maps/ucode_00001C000000000000001000000100800 HTTP/1.1
Host: www.example.org
Content-Length: xxx
```

<レスポンス>

```
HTTP/1.1 204 No Content
Connection: close
```

——[6-17: 地図情報の属性削除]——

<機能概要>

プロパティを指定して、地図情報の属性を削除する。指定したプロパティ以外の地図情報は残る。

<メソッド>

DELETE

<URLパス>

/api/v1/maps/<target>/<property>

<制約条件>

<target> に対する地図情報を更新することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>

削除対象の識別子とプロパティはURI 部分に指定する。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表6-45] 地図情報の属性削除のステータスコード

ステータスコード	意味
204 No Content	正常終了。
403 Forbidden	アクセス権限がない。
404 Not Found	対象の地図情報識別子が情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
DELETE /api/v1/maps/ucode_00001C000000000000001000000100800/ug_region
HTTP/1.1
Host: www.example.org
Content-Length: xxx
```

<レスポンス>

```
HTTP/1.1 204 No Content
Connection: close
```

7. Security Management Command

公共交通情報流通連携基盤システム

Security Management Commandは、標準API が提供する各オペレーションに対するアクセス制御を実現するためのコマンドである。認可の仕組みとして、OAuth 2.0 が利用できる。Security Management Command が規定するルール文書は、XMLで記述する。そのXML文書には、以下の表に示すタグが含まれているものとする。

[表7-1] ルール文書に含まれるXMLタグキー

ルール	説明	[省略値の値]
rule	ルール文書のルートタグ(省略不能)	
property	制限対象のプロパティURI[すべて]	
operator	制限対象のオペレータURI[すべて]	
operation	制限対象のオペレーションURL[すべて]	
method	制限対象のHTTPメソッド[すべて]	
permission	認可情報	[deny]

認可情報では、allowで対象のオペレーションを許可することを示し、denyで対象のオペレーションを禁止することを示す。

ルール文書それぞれにucodeを付与する。これをアクセス制御のucodeとする。このucodeとルールのXML文書とを、dc:description のプロパティで接続する。また、任意のURIリソースとアクセス制御ルールのucodeをdeterms:accessRightsプロパティで接続することにより、そのURIリソースに関する情報に対するアクセス制御を定義する。

本コマンドの一覧を以下の表に示す。

[表7-2] Security Management Command の一覧

URLパス	HTTPメソッド	意味
/api/v1/rules	GET	ルールを検索する
/api/v1/rules	POST	ルールを新規登録する
/api/v1/rules/<target>	GET	ルールを閲覧する
/api/v1/rules/<target>/<property>	GET	ルールを閲覧する
/api/v1/rules/<target>	PUT	ルールを更新する
/api/v1/rules/<target>/<property>	PUT	ルールを更新する
/api/v1/rules/<target>	DELETE	ルールを削除する
/api/v1/rules/<target>/<property>	DELETE	ルールを削除する
/api/v1/users	GET	ユーザを検索する
/api/v1/users	POST	ユーザを新規登録する
/api/v1/users/<target>	GET	ユーザを閲覧する
/api/v1/users/<target>/<property>	GET	ユーザを閲覧する
/api/v1/users/<target>	PUT	ユーザを更新する
/api/v1/users/<target>/<property>	PUT	ユーザを更新する
/api/v1/users/<target>	DELETE	ユーザを削除する
/api/v1/users/<target>/<property>	DELETE	ユーザを削除する
/api/v1/groups	GET	グループを検索する
/api/v1/groups	POST	グループを新規登録する
/api/v1/groups/<target>	GET	グループを閲覧する
/api/v1/groups/<target>/<property>	GET	グループを閲覧する
/api/v1/groups/<target>	PUT	グループを更新する
/api/v1/groups/<target>/<property>	PUT	グループを更新する
/api/v1/groups/<target>	DELETE	グループを削除する
/api/v1/groups/<target>/<property>	DELETE	グループを削除する

[7-1: ルールの検索]

<機能概要>

ルールを検索する。

<メソッド>

GET

<URLパス>

/api/v1/rules?<param1 >=<value1 >¶m2=<value2 >

<制約条件>

なし。誰でもリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。複数のパラメータを指定した場合、AND 検索となる。

[表7-3] ルールの検索のリクエストパラメータ

パラメータ名	デフォルト値	説明
paramN	(指定なし)	検索対象パラメータ名
valueN	(指定なし)	検索対象パラメータ値

- ・ <paramN>、<valueN> の組を1個以上指定しなければならない。
- ・ <paramN> は以下のいずれかである。
 - － ルール属性を示すプロパティURI。
 - － target。検索対象とするルールの識別子であり、パラメータ値の型はxsd:anyURI [] である。URI がカンマを含むとき、それをURLエンコードすること。target が複数ある場合は、それらをカンマで区切ること。
 - － offset, limit。パラメータ値はxsd:integer である。検索結果のうちoffset番目からlimit個分を要求することを意味する。
- ・ ルール文書内のXMLタグ名。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表7-4] ルールの検索のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<param1>、<value1>の組がない。<paramN>が正しくない。
404 Not Found	検索条件を満たすルールが情報流通連携基盤システムに登録されていない。
413 Request Entity Too Large	limit値が大きすぎる。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

レスポンスは、以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表7-5] ルールの検索のレスポンスパラメータ

パラメータ名	型	説明
rules	RDF	ルールのリスト。レスポンス形式にXMLを指定した場合、各ルール情報はRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各ルール情報はRDF/JSONで表現される。
remains	xsd:boolean	条件にマッチするが、レスポンスに含まれないルールがある場合はtrue、ない場合はfalse。

【APIの利用例】

<クエリ>

GET /api/v1/rules?permission=allow HTTP/1.1
Host: www.example.org

<レスポンス>

HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{
 "rules": [
 {"urn:ucode_00001C0000000000000001000000100800" : {
 "http://purl.org/dc/elements/1.1/description" : [
 { "value" : "<?XML version='1.0' "><rule> ...
 <permission>allow</permission></rule>",
 "type" : "literal"]}],
 "remains":false
 }
]
}

[7-2: ルールの新規登録]

<機能概要>

ルールを新規登録する。

<メソッド>

POST

<URLパス>

/api/v1/rules

<制約条件>

なし。誰でもリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、POSTメソッドのボディ部に、JSONまたはXML形式で格納する。
(A)か(B)のどちらか1種類のみを指定できるが、(A)を推奨する。

[表7-6] ルールの新規登録のリクエストパラメータ (A)

パラメータ名	型	説明
rdf	RDF	RDF/XMLまたはRDF/JSONで記述されたRDFデータ。urn:ucode: _?<val>という形のURIを含めることにより、ucodeの自動発行を要求できる。

[表7-7] ルールの新規登録のリクエストパラメータ (B)

パラメータ名	型	説明
target	xsd:anyURI []	ルールの識別子
params	struct[]	キーがプロパティのURI、値が登録値であるリスト。
num	xsd:integer	新規発行するucodeの個数。省略時は1。

- ・ targetまたはnumの一方のみを指定できる。
- ・ rdfとparams のどちらか一方は指定されていなければならない。また、params にはdc:descriptionを含まなければならない。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表7-8] ルールの新規登録のステータスコード

ステータスコード	意味
201 Created	正常終了。
400 Bad Request	パラメータに rdf と params のどちらも指定されていない。 パラメータに rdf と params, target, num のいずれかが指定されている。 <params>のキーが正しくない。
409 Conflict	パラメータに target と num の両方を指定した。 指定したルールの識別子は、すでに情報流通連携基盤システムに登録されている。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

クエリパラメータが(A)の形式であった場合のレスポンスは、以下の表(A)に示す構造データをJSONまたはXML形式で表現したものである。

クエリパラメータが(B)の形であった場合のレスポンスは、以下の表(B)に示す構造データをJSONまたはXML形式で表現したものである。

[表7-9] ルールの新規登録のレスポンスパラメータ (A)

パラメータ名	型	説明
ucode	struct	キーが指定された変数名、値が発行されたucodeのURI表現であるリスト。

[表7-10] ルールの新規登録のレスポンスパラメータ (B)

パラメータ名	型	説明
ucode	xsd:anyURI []	作成されたルールの識別ucode。

【APIの利用例】

<クエリ1>

```
POST /api/v1/rules HTTP/1.1
Content-Length: xxx
Content-Type: application/json; charset=utf-8
Host: www.example.org
{"rdf": [
  { "ucode": "_?x": {
    "http://purl.org/dc/elements/1.1/description" : [
      { "value": "<?XML version='1.0'">
<rule> ... <permission>allow</permission></rule>",
      "type": "literal" } ] } } ] }
```

<レスポンス1>

```
HTTP/1.1 201 Created
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"ucode": {"ucode": "_?x": "urn:ucode:_00001C00000000000001000000100801"}}
```

<クエリ2>

```
POST /api/v1/rules HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{"params": {"<dc:description>": "<?XML version='1.0'"><rule> ...
<permission>allow</permission></rule>"}}
```

<レスポンス2>

```
HTTP/1.1 201 Created
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"ucode": ["<urn:ucode:_00001C00000000000001000000100801>"]}
```

—[7-3: ルールの閲覧]—

<機能概要>

ルールを閲覧する。

<メソッド>

GET

<URLパス>

/api/v1/rules/<target>

<制約条件>

<target>が指定するルールの閲覧を許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。

[表7-11] ルールの閲覧のリクエストパラメータ

パラメータ名	型	説明
target	xsd:anyURI []	ルールの識別子

<ステータスコード>

ステータスコードは以下の表の通りである。

[表7-12] ルールの閲覧のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<target>が指定されていない。
404 Not Found	対象のルールが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表7-13] ルールの閲覧のレスポンスパラメータ

パラメータ名	型	説明
rules	RDF	指定されたルール情報のリスト。レスポンス形式にXMLを指定した場合、各ルール情報はRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各ルール情報はRDF/JSONで表現される。

【APIの利用例】

<クエリ>

```
GET /api/v1/rules/ucode_00001C000000000000001000000100800 HTTP/1.1
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"rules": {
  "urn:ucode:_00001C000000000000001000000100800": {
    "http://purl.org/dc/elements/1.1/description": [
      { "value": "<?XML version='1.0'><rule> ...
<permission>allow</permission></rule>",
        "type": "literal" } ] },
    "remains": false
  }
}
```

――[7-4: ルールの閲覧(プロパティ指定)]――

<機能概要>

プロパティを指定して、ルールを閲覧する。

<メソッド>

GET

<URLパス>

/api/v1/rules/<target>/<property>

<制約条件>

<target>が指定するルールの閲覧を許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。property パラメータとして、ルール文書XMLのキーを指定してもよい。

[表7-14] ルールの閲覧(プロパティ指定)のリクエストパラメータ

パラメータ名	型	説明
target	xsd:anyURI []	ルール情報の識別子
property	xsd:anyURI []	プロパティ値

<ステータスコード>

ステータスコードは以下の表の通りである。

[表7-15] ルールの閲覧(プロパティ指定)のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<target>、<property>が指定されていない。
404 Not Found	対象のルールが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

指定されたプロパティをもつ目的語値。

ルールの識別子もプロパティの識別子も1種類である場合は、該当する目的語値のリストを返す。

そうでない場合は、以下の表の構造データをJSONまたはXML形式で表現したものである。

[表7-16] ルールの閲覧(プロパティ指定)のレスポンスパラメータ

パラメータ名	型	説明
place	RDF	指定されたルール情報のリスト。レスポンス形式にXMLを指定した場合、各ルール情報はRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各ルール情報はRDF/JSONで表現される。

【APIの利用例】

<クエリ1>

GET /api/v1/rules/ucode_00001C000000000000001000000100800/permission

HTTP/1.1

Host: www.example.org

<レスポンス1>

HTTP/1.1 200 OK

Content-Length: xxx

Connection: close

Content-Type: application/json; charset=utf-8

["allow"]

<クエリ2>

```
GET /api/v1/rules/ucode_00001C000000000000001000000100800,  
ucode_00001C000000000000001000000100801/permission HTTP/1.1  
Host: www.example.org
```

<レスポンス2>

```
HTTP/1.1 200 OK  
Content-Length: xxx  
Connection: close  
Content-Type: application/json; charset=utf-8  
{  
  "rules": {  
    "urn:ucode:_00001C000000000000001000000100800": {  
      "permission": [  
        { "value": "allow", "type": "literal" } ] },  
    "urn:ucode:_00001C000000000000001000000100801": {  
      "permission": [  
        { "value": "deny", "type": "literal" } ] }  
    }  
  }  
}
```

[7-5: ルールの更新]

<機能概要>

ルールを更新する。

<メソッド>

PUT

<URLパス>

/api/v1/rules/<target>

<制約条件>

<target>に対するルールを更新することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータ(以下の表)のうち、更新対象の識別子はURI 部分に指定する。それ以外のパラメータは、PUTメソッドのボディ部に、JSONまたはXML形式で格納する。

[表7-17] ルールの更新のリクエストパラメータ

パラメータ名	型	説明
rdf	RDF	RDF/XMLまたはRDF/JSON形式で記した更新情報。更新情報のsubject は、<target> と一致していること。コマンド終了後、リクエストパラメータに含まれるpredicateに対する値は、個数を含めてリクエストパラメータが指定した情報と完全に一致する。
rdf	RDF	RDF/XMLまたはRDF/JSON形式で記した更新情報。更新情報のsubject は、<target> と一致していること。コマンド終了後、更新情報に含まれるpredicate に対する値は、個数を含めてリクエストパラメータが指定した情報と完全に一致する。更新情報に含まれないpredicate に関する値は変化しない。

- ・ params またはrdf のいずれか一方のみが指定されなければならない。rdf を推奨する。
- ・ <params> のキーは、ルールの属性を示すプロパティ値である。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表7-18] ルールの更新のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータ値が正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	対象のルール識別子が情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ1>

```
PUT /api/v1/rules/ucode_00001C00000000000000010000000100800 HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{"rdf": {
  "urn:ucode:_00001C00000000000000010000000100800": {
    "http://uidcenter.org/ucr/vocab/ug#region": [
      { "value": "<?XML version='1.0'><rule>
... <permission>deny</permission></rule>",
      "type": "uri" } ] } } }
```

<レスポンス1>

HTTP/1.1 204 No Content
Connection: close
Content-Type: application/json; charset=utf-8

<クエリ2>

PUT /api/v1/rules/ucode_00001C0000000000000001000000100800 HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{
 "params": {
 "<dc:description>": "<?XML version='1.0'><rule>
 ... <permission>deny</permission></rule>"
 }
}

<レスポンス2>

HTTP/1.1 204 No Content
Connection: close
Content-Type: application/json; charset=utf-8

――[7-6: ルールの更新(プロパティ指定)]――

<機能概要>

プロパティを指定して、ルールを更新する。

<メソッド>

PUT

<URLパス>

/api/v1/rules/<target>/<property>

<制約条件>

<target>に対するルールを更新することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータのうち、更新対象の識別子とプロパティはURI 部分に指定する。プロパティ名として、ルールXML文書のキーを指定してもよい。更新する値は、PUTメソッドのボディ部に、JSONまたはXML形式で格納する。

本コマンド終了後、更新対象の識別子URI とプロパティをもつオブジェクトは、リクエストパラメータに指定した値のみとなる。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表7-19] ルールの更新(プロパティ指定)のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータ値が正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	対象のルール識別子が情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
PUT /api/v1/rules/ucode_00001C0000000000000001000000100800/permission
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
[ "deny" ]
```

<レスポンス>

```
HTTP/1.1 204 No Content
Connection: close
Content-Type: application/json; charset=utf-8
```

――[7-7: ルールの削除]――

<機能概要>

ルールを削除する。

<メソッド>

DELETE

<URLパス>

/api/v1/rules/<target>

<制約条件>

<target>に対するルールを削除することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>

削除対象の識別子はURI 部分に指定する。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表7-20] ルールの削除のステータスコード

ステータスコード	意味
204 No Content	正常終了。
403 Forbidden	アクセス権限がない。
404 Not Found	対象のルール識別子が情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
DELETE /api/v1/rules/ucode_00001C00000000000001000000100800 HTTP/1.1
Host: www.example.org
Content-Length: xxx
```

<レスポンス>

```
HTTP/1.1 204 No Content
Connection: close
```

——[7-8: ルール属性の削除]——

<機能概要>

プロパティを指定して、ルールの属性情報を削除する。指定したプロパティ以外のルール属性情報は残る。

<メソッド>

DELETE

<URLパス>

/api/v1/rules/<target>/<property>

<制約条件>

<target>に対するルールを更新することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>

削除対象の識別子とプロパティはURL 部分に指定する。プロパティとして、ルールXML文書のキーを指定してもよい。ただしpermission は指定できない。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表7-21] ルール属性の削除のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータが正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	対象のルール識別子が情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

DELETE /api/v1/rules/ucode_00001C00000000000001000000100800/operator HTTP/1.1

Host: www.example.org

Content-Length: xxx

<レスポンス>

HTTP/1.1 204 No Content

Connection: close

――[7-9: ルールの適用先閲覧]――

<機能概要>

ルールの適用先を閲覧する。

<メソッド>

GET

<URLパス>

/api/v1/rules/<target>/applied

<制約条件>

なし。誰でもリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。

[表7-22] ルールの適用先閲覧のリクエストパラメータ

パラメータ名	型	説明
target	xsd:anyURI []	ルールの識別子

<ステータスコード>

ステータスコードは以下の表の通りである。

[表7-23] ルールの適用先閲覧のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<target>が指定されていない。
404 Not Found	対象のルールが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

このルールを適用されているオープンデータの識別子のリストを、XMLまたはJSON形式で返す。

【APIの利用例】

<クエリ>

```
GET /api/v1/rules/ucode_00001C00000000000000010000000100800/applied
HTTP/1.1
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
["<urn:ucode:_00001C00000000000000010000000100700>",
"<urn:ucode:_00001C00000000000000010000000100701>",
"<urn:ucode:_00001C00000000000000010000000100702>"]
```

<機能概要>

ルールをオープンデータに適用する。これと同じことは、オープンデータの属性値として `dcterms:accessRights` を追加するオペレーションを利用しても実現できる。

<メソッド>

PUT

<URLパス>

`/api/v1/rules/<target>/applied`

<制約条件>

適用先オープンデータに対するルール属性を更新することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータのうち、更新対象の識別子はURI 部分に指定する。適用先のオープンデータ識別子のリストは、PUTメソッドのボディ部に、JSONまたはXML形式で格納する。
本コマンド終了後、適用先のオープンデータ識別子のリストは、リクエストパラメータに指定した値のみとなる。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表7-24] ルールの適用のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータ値が正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	対象のルール識別子が情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
PUT /api/v1/rules/ucode_00001C000000000000001000000100800/applied
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
["<urn:ucode:_00001C0000000000000001000000100700>",
"<urn:ucode:_00001C0000000000000001000000100701>",
"<urn:ucode:_00001C0000000000000001000000100702>"]
```


<レスポンス>

HTTP/1.1 204 No Content
Connection: close
Content-Type: application/json; charset=utf-8

――[7-11: ルールの適用解除]――

<機能概要>

指定したオープンデータを、このルールの適用対象から外す。これと同じことは、オープンデータの属性値としてdterms:accessRights を更新するオペレーションを利用しても実現できる。

<メソッド>

DELETE

<URLパス>

/api/v1/rules/<target>/applied

<制約条件>

適用先オープンデータに対するルール属性を更新することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>

ルールの識別子はURI 部分に指定する。適用解除対象のオープンデータの識別子は、PUTメソッドのボディ部に、JSONまたはXML形式で格納する。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表7-25] ルールの適用解除のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータが正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	対象のルール識別子が情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

DELETE /api/v1/rules/ucode_00001C00000000000001000000100800/applied
HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8

```
[ "<urn:ucode:_00001C000000000000001000000100700>",  
  "<urn:ucode:_00001C000000000000001000000100701>",  
  "<urn:ucode:_00001C000000000000001000000100702>" ]
```

<レスポンス>

HTTP/1.1 204 No Content
Connection: close

――[7-12: ユーザの検索]――

<機能概要>

ユーザを検索する。

<メソッド>

GET

<URLパス>

/api/v1/users?<param1 >=<value1 >¶m2=<value2 >

<制約条件>

なし。誰でもリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。複数のパラメータを指定した場合、AND 検索となる。

[表7-26] ユーザの検索のリクエストパラメータ

パラメータ名	デフォルト値	説明
paramN	(指定なし)	検索対象パラメータ名
valueN	(指定なし)	検索対象パラメータ値

・ <paramN>、<valueN> の組を1個以上指定しなければならない。<paramN> は以下のいずれかである。

- ユーザ属性を示すプロパティURI
- target。ユーザの識別子であり、パラメータ値の型はxsd:anyURI[] である。URI がカンマを含むとき、それをURLエンコードすること。target が複数ある場合は、それらをカンマで区切ること。
- offset, limit。パラメータ値はxsd:integerである。検索結果のうちoffset番目からlimit個分を要求することを意味する。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表7-27] ユーザの検索のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<param1>、<value1>の組がない。<paramN>が正しくない。
404 Not Found	検索条件を満たすユーザが情報流通連携基盤システムに登録されていない。
413 Request Entity Too Large	limit値が大きすぎる。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

レスポンスは、以下の表の構造データをJSONまたはXML形式で表現したものである。

[表7-28] ユーザの検索のレスポンスパラメータ

パラメータ名	型	説明
place	RDF	ユーザのリスト。レスポンス形式にXMLを指定した場合、各ユーザ情報はRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各ユーザ情報はRDF/JSONで表現される。
remains	xsd:boolean	条件にマッチするが、レスポンスに含まれないユーザがある場合はtrue、ない場合はfalse。

【APIの利用例】

<クエリ>

```
GET /api/v1/users?foaf_name=Alice HTTP/1.1
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"users": {
  "urn:ucode:_00001C000000000000001000000100800" : {
    "http://xmlns.com/foaf/0.1/name" : [
      { "value" : "Alice", "type" : "literal" } ],
    "http://xmlns.com/foaf/0.1/age" : [
      { "value" : "35", "type" : "literal" } ]}],
  "remains":false
}
```

<機能概要>

ユーザを新規登録する。

<メソッド>

POST

<URLパス>

/api/v1/users

<制約条件>

なし。誰でもリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、POSTメソッドのボディ部に、JSONまたはXML形式で格納する。
(A)か(B)のどちらか1種類のみを指定できるが、(A)を推奨する。

[表7-29] ユーザの新規登録のリクエストパラメータ (A)

パラメータ名	型	説明
rdf	RDF	RDF/XMLまたはRDF/JSONで記述されたRDFデータ。urn:ucode: _?<val>という形のURIを含めることにより、ucodeの自動発行を要求できる。

[表7-30] ユーザの新規登録のリクエストパラメータ (B)

パラメータ名	型	説明
target params num	xsd:anyURI [] struct [] xsd:integer	ユーザの識別子 キーがプロパティのURI、値が登録値であるリスト。 新規発行するucodeの個数。省略時は1。

- ・ target またはnumの一方のみを指定できる。
- ・ <rdf> と<params>のどちらか一方は指定されていないなければならない。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表7-31] ユーザの新規登録のステータスコード

ステータスコード	意味
201 Created	正常終了。
400 Bad Request	パラメータに rdf と params のどちらも指定されていない。 パラメータに rdf と params, target, num のいずれかが指定されている。 <params>のキーが正しくない。
409 Conflict	パラメータに target と num の両方を指定した。 指定したユーザの識別子は、すでに情報流通連携基盤システムに登録されている。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

クエリパラメータが(A)の形式であった場合のレスポンスは、以下の表(A)に示す構造データをJSONまたはXML形式で表現したものである。

クエリパラメータが(B)の形であった場合のレスポンスは、以下の表(B)に示す構造データをJSONまたはXML形式で表現したものである。

[表7-32] ユーザの新規登録のレスポンスパラメータ (A)

パラメータ名	型	説明
ucode	struct	キーが指定された変数名、値が発行されたucodeのURI表現であるリスト。

[表7-33] ユーザの新規登録のレスポンスパラメータ (B)

パラメータ名	型	説明
ucode	xsd:anyURI []	作成されたユーザの識別ucode。

【APIの利用例】

<クエリ1>

```
POST /api/v1/users HTTP/1.1
Content-Length: xxx
Content-Type: application/json; charset=utf-8
Host: www.example.org
{"rdf": {
  "ucode_x": {
    "http://xmlns.com/foaf/0.1/name" : [
      { "value": "Alice", "type": "literal" } ],
    "http://xmlns.com/foaf/0.1/age" : [
      { "value": "35", "type": "literal" } ] } } }
```

<レスポンス1>

```
HTTP/1.1 201 Created
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"uicode":{"
"uicode_?x":"urn:uicode:_00001C000000000000001000000100801" } }
```

<クエリ2>

```
POST /api/v1/users HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{"params":{"<foaf:name>":"Alice","<foaf:age>":35}}
```

<レスポンス2>

```
HTTP/1.1 201 Created
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"uicode":["<urn:uicode:_00001C000000000000001000000100801>"]}
```

――[7-14: ユーザ情報の閲覧]――

<機能概要>

ユーザ情報を閲覧する。

<メソッド>

GET

<URLパス>

/api/v1/users/<target>

<制約条件>

<target>が指定するユーザ情報の閲覧を許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。

[表7-34] ユーザ情報の閲覧のリクエストパラメータ

パラメータ名	型	説明
target	xsd:anyURI []	ユーザの識別子

<ステータスコード>

ステータスコードは以下の表の通りである。

[表7-35] ユーザ情報の閲覧のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<target>が指定されていない。
404 Not Found	対象のユーザが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

レスポンスは、以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表7-36] ユーザ情報の閲覧のレスポンスパラメータ

パラメータ名	型	説明
users	RDF	指定されたユーザのリスト。レスポンス形式にXMLを指定した場合、各ユーザ情報はRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各ユーザ情報はRDF/JSONで表現される。

【APIの利用例】

<クエリ>

```
GET /api/v1/users/ucode_00001C00000000000000010000000100800 HTTP/1.1
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"users": [{
  "urn:ucode:_00001C00000000000000010000000100800": {
    "http://xmlns.com/foaf/0.1/name" : [
      { "value" : "Alice", "type" : "literal" } ],
    "http://xmlns.com/foaf/0.1/age" : [
      { "value" : "35", "type" : "literal" } ]}]}
```

――[7-15: ユーザ情報の閲覧(プロパティ指定)]――

<機能概要>

プロパティを指定して、ユーザ情報を閲覧する。

<メソッド>

GET

<URLパス>

/api/v1/users/<target>/<property>

<制約条件>

<target>が指定するユーザの閲覧を許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。

[表7-37] ユーザ情報の閲覧(プロパティ指定)のリクエストパラメータ

パラメータ名	型	説明
target	xsd:anyURI []	ユーザの識別子
property	xsd:anyURI []	プロパティ値

<ステータスコード>

ステータスコードは以下の表の通りである。

[表7-38] ユーザ情報の閲覧(プロパティ指定)のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<target>、<property>が指定されていない。
404 Not Found	対象のユーザが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

指定されたプロパティをもつ目的語値。ユーザの識別子もプロパティの識別子も1種類である場合は、該当する目的語値のリストを返す。そうでない場合は、以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表7-39] ユーザ情報の閲覧(プロパティ指定)のレスポンスパラメータ

パラメータ名	型	説明
users	RDF	指定されたユーザのリスト。レスポンス形式にXMLを指定した場合、各ユーザ情報はRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各ユーザ情報はRDF/JSONで表現される。

【APIの利用例】

<クエリ1>

```
GET /api/v1/users/ucode_00001C0000000000000010000000100800/foaf_age HTTP/1.1
Host: www.example.org
```

<レスポンス1>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
[ "35" ]
```

<クエリ2>

```
GET /api/v1/users/ucode_00001C0000000000000010000000100800,
ucode_00001C0000000000000010000000100801/foaf_age HTTP/1.1
Host: www.example.org
```

<レスポンス2>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"users": {
  "urn:ucode:_00001C0000000000000010000000100800": {
    "http://xmlns.com/foaf/1.1/age" : [
      { "value" : "35", "type" : "literal" } ] },
  "urn:ucode:_00001C0000000000000010000000100801": {
    "http://xmlns.com/foaf/1.1/age" : [
      { "value" : "40", "type" : "literal" } ] ]}}
```

<機能概要>

ユーザを更新する。

<メソッド>

PUT

<URLパス>

/api/v1/users/<target>

<制約条件>

<target>に対するユーザを更新することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータ(以下の表)のうち、更新対象の識別子はURI 部分に指定する。それ以外のパラメータは、PUTメソッドのボディ部に、JSONまたはXML形式で格納する。

[表7-40] ユーザの更新のリクエストパラメータ

パラメータ名	型	説明
rdf	RDF	RDF/XMLまたはRDF/JSON形式で記した更新情報。更新情報のsubject は、<target> と一致していること。コマンド終了後、更新情報に含まれるpredicate に対する値は、個数を含めてリクエストパラメータが指定した情報と完全に一致する。更新情報に含まれないpredicate に関する値は変化しない。
params	struct[]	キーが登録パラメータ名、値が登録値であるリスト。コマンド終了後、キーに指定されたプロパティに対する値は、個数を含めてリクエストパラメータが指定した情報と完全に一致する。リクエストパラメータのキーにないプロパティに関する値は変化しない。

- ・ params またはrdf のいずれか一方のみが指定されなければならない。rdf を推奨する。
- ・ params のキーは、ユーザの属性を示すプロパティ値である。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表7-41] ユーザの更新のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータ値が正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	対象のユーザが情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ1>

```
PUT /api/v1/users/urn:_00001C000000000000001000000100800 HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{"rdf": {
  "urn:ucode:_00001C000000000000001000000100800": {
    "http://xmlns.com/foaf/0.1/name": [
      { "value": "Alice", "type": "literal" } ],
    "http://xmlns.com/foaf/0.1/age": [
      { "value": "35", "type": "literal" } ] } }
```

<レスポンス1>

```
HTTP/1.1 204 No Content
Connection: close
```

<クエリ2>

```
PUT /api/v1/users/ucode_00001C000000000000001000000100800 HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{"params":{"<foaf:name>":"Alice","<foaf:age>","35}}
```

<レスポンス2>

```
HTTP/1.1 204 No Content
Connection: close
```

――[7-17: ユーザの更新(プロパティ指定)]――

<機能概要>

プロパティを指定して、ユーザを更新する。

<メソッド>

PUT

<URLパス>

/api/v1/users/<target>/<property>

<制約条件>

<target>に対するユーザを更新することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータのうち、更新対象の識別子とプロパティはURI 部分に指定する。更新する値は、PUTメソッドのボディ部に、JSONまたはXML形式で格納する。

本コマンド終了後、更新対象の識別子URI とプロパティをもつオブジェクトは、リクエストパラメータに指定した値のみとなる。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表7-42] ユーザの更新(プロパティ指定)のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータ値が正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	対象のユーザ識別子が情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
PUT /api/v1/users/ucode_00001C000000000000000010000000100800/foaf_age HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
[ "35" ]
```

<レスポンス>
HTTP/1.1 204 No Content
Connection: close

[7-18: ユーザの削除]

<機能概要>
ユーザを削除する。

<メソッド>
DELETE

<URLパス>
/api/v1/users/<target>

<制約条件>
<target>に対するユーザを削除することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>
削除対象の識別子はURI 部分に指定する。

<ステータスコード>
ステータスコードは以下の表の通りである。

[表7-43] ユーザの削除のステータスコード

ステータスコード	意味
204 No Content	正常終了。
403 Forbidden	アクセス権限がない。
404 Not Found	対象のユーザ識別子が情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>
成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>
DELETE /api/v1/users/ucode_00001C000000000000001000000100800 HTTP/1.1
Host: www.example.org
Content-Length: xxx

<レスポンス>
HTTP/1.1 204 No Content
Connection: close

<機能概要>

プロパティを指定して、ユーザの属性情報を削除する。指定したプロパティ以外のユーザ情報は残る。

<メソッド>

DELETE

<URLパス>

/api/v1/users/<target>/<property>

<制約条件>

<target> に対するユーザ属性を更新することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>

削除対象の識別子とプロパティはURI 部分に指定する。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表7-44] ユーザの属性情報削除のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータが正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	対象のユーザが情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

DELETE /api/v1/users/ucode_00001C00000000000001000000100800/foaf_age
HTTP/1.1
Host: www.example.org
Content-Length: xxx

<レスポンス>

HTTP/1.1 204 No Content
Connection: close

<機能概要>

グループを検索する。

<メソッド>

GET

<URLパス>

/api/v1/groups?<param1 >=<value1 >¶m2=<value2 >

<制約条件>

なし。誰でもリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。複数のパラメータを指定した場合、AND 検索となる。

[表7-45] グループの検索のリクエストパラメータ

パラメータ名	デフォルト値	説明
paramN	(指定なし)	検索対象パラメータ名
valueN	(指定なし)	検索対象パラメータ値

- ・ <paramN>、<valueN> の組を1個以上指定しなければならない。
- ・ <paramN> は以下のいずれかである。
 - グループ属性を示すプロパティURI。
 - target。グループの識別子であり、パラメータ値の型はxsd:anyURI[] である。URI がカンマを含むとき、それをURLエンコードすること。target が複数ある場合は、それらをカンマで区切ること。
 - offset, limit。パラメータ値はxsd:integer である。検索結果のうちoffset番目からlimit個分を要求することを意味する。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表7-46] グループの検索のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<param1>、<value1>の組がない。<paramN>が正しくない。
404 Not Found	検索条件を満たすグループが情報流通連携基盤システムに登録されていない。
413 Request Entity Too Large	limit値が大きすぎる。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

レスポンスは、以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表7-47] グループの検索のレスポンスパラメータ

パラメータ名	型	説明
groups	RDF	グループのリスト。レスポンス形式にXMLを指定した場合、各グループ情報はRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各グループ情報はRDF/JSONで表現される。
remains	xsd:boolean	条件にマッチするが、レスポンスに含まれない場所がある場合はtrue、ない場合はfalse。

【APIの利用例】

<クエリ>

```
GET /api/v1/groups?foaf_name=Alice HTTP/1.1
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"groups": {
  "urn:ucode:_00001C000000000000001000000100800": {
    "http://xmlns.com/foaf/0.1/name" : [
      { "value" : "ABC Group", "type" : "literal" } ] },
  "remains":false
}
```


——[7-21: グループの新規登録]——

＜機能概要＞

グループを新規登録する。

＜メソッド＞

POST

＜URLパス＞

/api/v1/groups

＜制約条件＞

なし。誰でもリクエストできる。

＜リクエストパラメータ＞

リクエストパラメータ(以下の表)は、POSTメソッドのボディ部に、JSONまたはXML形式で格納する。
(A)か(B)のどちらか1種類のみを指定できるが、(A)を推奨する。

[表7-48] グループの新規登録のリクエストパラメータ (A)

パラメータ名	型	説明
rdf	RDF	RDF/XMLまたはRDF/JSONで記述されたRDFデータ。urn:ucode: _?<val>という形のURIを含めることにより、ucodeの自動発行を要求できる。

[表7-49] グループの新規登録のリクエストパラメータ (B)

パラメータ名	型	説明
target params num	xsd:anyURI [] struct [] xsd:integer	グループの識別子 キーがプロパティのURI、値が登録値であるリスト。 新規発行するucodeの個数。省略時は1。

- ・ <target>または<num>の一方のみを指定できる。
- ・ <rdf>と<params>のどちらか一方は指定されていなければならない。

＜ステータスコード＞

ステータスコードは以下の表の通りである。

[表7-50] グループの新規登録のステータスコード

ステータスコード	意味
201 Created	正常終了。
400 Bad Request	パラメータに rdf と params のどちらも指定されていない。 パラメータに rdf と params, target, num のいずれかが指定されている。 <params>のキーが正しくない。
409 Conflict	パラメータに target と num の両方を指定した。 指定したグループの識別子は、すでに情報流通連携基盤システムに登録されている。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

クエリパラメータが(A)の形式であった場合のレスポンスは、以下の表(A)に示す構造データをJSONまたはXML形式で表現したものである。

クエリパラメータが(B)の形であった場合のレスポンスは、以下の表(B)に示す構造データをJSONまたはXML形式で表現したものである。

[表7-51] グループの新規登録のレスポンスパラメータ (A)

パラメータ名	型	説明
ucode	struct	キーが指定された変数名、値が発行されたucodeのURI表現であるリスト。

[表7-52] グループの新規登録のレスポンスパラメータ (B)

パラメータ名	型	説明
ucode	xsd:anyURI []	作成されたグループの識別ucode。

【APIの利用例】

<クエリ1>

```
POST /api/v1/groups HTTP/1.1
Content-Length: xxx
Content-Type: application/json; charset=utf-8
Host: www.example.org
{"rdf": {
  "ucode_x": {
    "http://xmlns.com/foaf/0.1/name" : [
      { "value": "ABC Group", "type": "literal" } ] } } }
```

<レスポンス1>

```
HTTP/1.1 201 Created
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"uicode":{
  "uicode_?x":"urn:uicode:_00001C000000000000001000000100801" } }
```

<クエリ2>

```
POST /api/v1/groups HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{"params":{"<foaf:name>":"ABC Group"}}
```

<レスポンス2>

```
HTTP/1.1 201 Created
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"uicode":["<urn:uicode:_00001C000000000000001000000100801>"]}
```

――[7-22: グループ情報の閲覧]――

<機能概要>

グループ情報を閲覧する。

<メソッド>

GET

<URLパス>

/api/v1/groups/<target>

<制約条件>

<target>が指定するグループ情報の閲覧を許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。

[表7-53] グループ情報の閲覧のリクエストパラメータ

パラメータ名	型	説明
target	xsd:anyURI []	グループの識別子

<ステータスコード>

ステータスコードは以下の表の通りである。

[表7-54] グループ情報の閲覧のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<target>が指定されていない。
404 Not Found	対象のグループが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

レスポンスは、以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表7-55] グループ情報の閲覧のレスポンスパラメータ

パラメータ名	型	説明
groups	RDF	指定されたグループのリスト。レスポンス形式にXMLを指定した場合、各グループ情報はRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各グループ情報はRDF/JSONで表現される。

【APIの利用例】

<クエリ>

```
GET /api/v1/groups/ucode_00001C000000000000001000000100800 HTTP/1.1
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"groups": {
  "urn:ucode:_00001C000000000000001000000100800": {
    "http://xmlns.com/foaf/0.1/name" : [
      { "value" : "ABC Group", "type" : "literal" } ] ] }
```

——[7-23: グループ情報の閲覧(プロパティ指定)]——

<機能概要>

プロパティを指定して、グループを閲覧する。

<メソッド>

GET

<URLパス>

/api/v1/groups/<target>/<property>

<制約条件>

<target>が指定するグループの閲覧を許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。

[表7-56] グループ情報の閲覧(プロパティ指定)のリクエストパラメータ

パラメータ名	型	説明
target	xsd:anyURI []	グループの識別子
property	xsd:anyURI []	プロパティ値

<ステータスコード>

ステータスコードは以下の表の通りである。

[表7-57] グループ情報の閲覧(プロパティ指定)のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<target>、<property>が指定されていない。
404 Not Found	対象のグループが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

指定されたプロパティをもつ目的語値。グループの識別子もプロパティの識別子も1種類である場合は、該当する目的語値のリストを返す。そうでない場合は、以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表7-58] グループ情報の閲覧(プロパティ指定)のレスポンスパラメータ

パラメータ名	型	説明
groups	RDF	指定されたグループのリスト。レスポンス形式にXMLを指定した場合、各グループ情報はRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各グループ情報はRDF/JSONで表現される。

【APIの利用例】

<クエリ1>

```
GET /api/v1/groups/ucode_00001C000000000000001000000100800/foaf_name
HTTP/1.1
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
[ "ABC Group" ]
```

<クエリ2>

```
GET /api/v1/groups/ucode_00001C000000000000001000000100800,
ucode_00001C000000000000001000000100801/foaf_name HTTP/1.1
Host: www.example.org
```

<レスポンス2>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"groups": {
  "urn:ucode:_00001C000000000000001000000100800": {
    "http://xmlns.com/foaf/1.1/name" : [
      { "value" : "ABC Group", "type" : "literal" } ] },
  "urn:ucode:_00001C000000000000001000000100801": {
    "http://xmlns.com/foaf/1.1/name" : [
      { "value" : "XYZ Group", "type" : "literal" } ] ]}}
```

<機能概要>

グループを更新する。

<メソッド>

PUT

<URLパス>

/api/v1/groups/<target>

<制約条件>

<target> に対するグループを更新することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータ(以下の表)のうち、更新対象の識別子はURI 部分に指定する。それ以外のパラメータは、PUTメソッドのボディ部に、JSONまたはXML形式で格納する。

[表7-59] グループの更新のリクエストパラメータ

パラメータ名	型	説明
rdf	RDF	RDF/XMLまたはRDF/JSON形式で記した更新情報。更新情報のsubject は、<target> と一致していること。コマンド終了後、更新情報に含まれるpredicate に対する値は、個数を含めてリクエストパラメータが指定した情報と完全に一致する。更新情報に含まれないpredicate に関する値は変化しない。
params	struct[]	キーが登録パラメータ名、値が登録値であるリスト。コマンド終了後、キーに指定されたプロパティに対する値は、個数を含めてリクエストパラメータが指定した情報と完全に一致する。リクエストパラメータのキーにないプロパティに関する値は変化しない。

- ・ params またはrdf のいずれか一方のみが指定されなければならない。rdf を推奨する。
- ・ params のキーは、グループの属性を示すプロパティ値である。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表7-60] グループの更新のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータ値が正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	対象のグループが情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ1>

```
PUT /api/v1/groups/ucode_00001C000000000000001000000100800 HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{"rdf": {
  "urn:ucode:_00001C000000000000001000000100800": {
    "http://xmlns.com/foaf/0.1/name": [
      { "value": "ABC Group", "type": "literal" } ] } } }
```

<レスポンス1>

```
HTTP/1.1 204 No Content
Connection: close
```

<クエリ2>

```
PUT /api/v1/groups/ucode_00001C000000000000001000000100800 HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{"params":{"<foaf:name>":"ABC Group"}}
```

<レスポンス2>

```
HTTP/1.1 204 No Content
Connection: close
```


――[7-25: グループの更新(プロパティ指定)]――

<機能概要>

プロパティを指定して、グループを更新する。

<メソッド>

PUT

<URLパス>

/api/v1/groups/<target>/<property>

<制約条件>

<target> に対するグループを更新することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータのうち、更新対象の識別子とプロパティはURI 部分に指定する。更新する値は、PUTメソッドのボディ部に、JSONまたはXML形式で格納する。

本コマンド終了後、更新対象の識別子URI とプロパティをもつオブジェクトは、リクエストパラメータに指定した値のみとなる。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表7-61] グループの更新(プロパティ指定)のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータ値が正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	対象のグループが情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
PUT /api/v1/groups/ucode_00001C000000000000001000000100800/foaf_name
HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
[ "ABC Group" ]
```

<レスポンス>
HTTP/1.1 204 No Content
Connection: close

――[7-26: グループの削除]――

<機能概要>
グループを削除する。

<メソッド>
DELETE

<URLパス>
/api/v1/groups/<target>

<制約条件>
<target> に対するグループを削除することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>
削除対象の識別子はURI 部分に指定する。

<ステータスコード>
ステータスコードは以下の表の通りである。

[表7-62] グループの削除のステータスコード

ステータスコード	意味
204 No Content	正常終了。
403 Forbidden	アクセス権限がない。
404 Not Found	対象のグループ識別子が情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>
成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>
DELETE /api/v1/groups/ucode_00001C000000000000001000000100800 HTTP/1.1
Host: www.example.org
Content-Length: xxx

<レスポンス>

HTTP/1.1 204 No Content
Connection: close

――[7-27: グループの属性情報削除]――

<機能概要>

プロパティを指定して、グループの属性情報を削除する。指定したプロパティ以外のユーザ情報は残る。

<メソッド>

DELETE

<URLパス>

/api/v1/groups/<target>/<property>

<制約条件>

<target> に対するグループ属性を更新することが許可されているユーザによるアクセスであること。

<リクエストパラメータ>

削除対象の識別子とプロパティはURI 部分に指定する。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表7-63] グループの属性情報削除のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータが正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	対象のグループが情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

DELETE /api/v1/groups/ucode_00001C000000000000001000000100800/foaf_name
HTTP/1.1
Host: www.example.org
Content-Length: xxx

<レスポンス>

HTTP/1.1 204 No Content

Connection: close

8. Notification Management Command

公共交通情報流通連携基盤システム

Notification とは、対象としているオープンデータが登録・更新され、指定した条件を満たした場合にユーザプログラムにコールバックする仕組みである。コールバック先は、URLで指定する。コールバックURLに“%U”が含まれる場合、情報流通連携基盤システムは、この部分を対象の識別子のURI 表現で置き替える。指定できるNotification 条件を、以下の表に列挙する。

[表8-1] Notification の条件リスト

条件名	意味
any	条件なし(つねにコールバックする)
gt	指定した値より大きいgte指定した値以上
lt	指定した値より小さいlte指定した値以下
eq	指定した値と等しい

Notification Management Commandは、このNotification 機能を実現するためのコマンドである。本コマンドの一覧を、以下の表に示す。

[表8-2] Notification Management Command の一覧

URLパス	HTTPメソッド	意味
/api/v1/triggers	GET	Notification を検索する
/api/v1/triggers	POST	Notification を登録する
/api/v1/triggers/<target>	GET	Notification 情報を閲覧する
/api/v1/triggers/<target>	PUT	Notification 情報を更新する
/api/v1/triggers/<target>	DELETE	Notification 情報を削除する
/api/v1/triggers/<target>/run	PUT	Notification を起動・再開する
/api/v1/triggers/<target>/run	DELETE	Notification を停止する

——[8-1: Notification の検索]——

＜機能概要＞

Notification を検索する。

ただし、閲覧権限のないNotification は検索できない。

＜メソッド＞

GET

＜制約条件＞

なし。誰でもリクエストできる。

＜URLパス＞

/api/v1/triggers?<param1 >=<value1 >&<param2 >=<value2 >

＜リクエストパラメータ＞

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。

[表8-3] Notification の検索のリクエストパラメータ

パラメータ名	デフォルト値	説明
paramN	(指定なし)	検索対象パラメータ名
valueN	(指定なし)	検索対象パラメータ値

・ <paramN>、<valueN> の組を1個以上指定しなければならない。<paramN> はNotification 属性を示すプロパティである。ただし、下記も利用できる。

- name: Notification 名。
- target: 発動対象のオープンデータ識別子。複数ある場合はカンマで区切る。
- type: パラメータの発動条件。詳細は別表を参照のこと。
- value: 発動条件の閾値。
- callback: コールバックURL (URLエンコードすること)。
- offset, limit: パラメータ値はxsd:integer である。検索結果のうちoffset番目からlimit個分を要求することを意味する。

＜ステータスコード＞

ステータスコードは以下の表の通りである。

[表8-4] Notification の検索のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<param1>、<value1>の組がない。<paramN>が正しくない。
404 Not Found	検索対象のNotificationが見つからない。
413 Request Entity Too Large	limit値が大きすぎる。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

レスポンスは、以下の表の構造データをJSONまたはXML形式で表現したものである。

[表8-5] Notification の検索のレスポンスパラメータ

パラメータ名	型	説明
triggers	struct[]	検索の結果得られたNotification。
— ucode	xsd:anyURI	Notification のucode。
— name	xsd:string	Notification 名。
— target	xsd:anyURI []	発動対象のオープンデータ識別子
— type	xsd:string	Notification の発動条件。詳細は別表を参照のこと。
— value	xsd:string	発動条件の閾値。
— callback	xsd:string	コールバックURL。
— run	xsd:boolean	稼働中ならばtrue、そうでなければfalse
— (*)	(*)	以下、該当するNotificationに結びついている属性を返す。パラメータ名はNotification属性を示すプロパティである。
remains	xsd:boolean	検索条件にマッチするが、レスポンスに含めていないNotificationがあればtrue、なければfalse

【APIの利用例】

<クエリ>

GET /api/v1/triggers?name=TriggerA HTTP/1.1
Host: www.example.org

<レスポンス1>

HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{
 "triggers": [
 {"ucode": "<urn:ucode:_00001C000000000000001000000100126>",
 "name": "TriggerA",
 "target": [
 "<urn:ucode:_00001C000000000000001000000100123>",
 "<urn:ucode:_00001C000000000000001000000100124>"],
 "type": "any",
 "callback": "http://www.example.org/?ucode=%U",

```
"run": "false" }],  
  "remains": false}
```

――[8-2: Notification の新規作成]――

<機能概要>

Notification を新規作成する。

<メソッド>

POST

<URLパス>

/api/v1/triggers

<制約条件>

なし。誰でもリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、PUTメソッドのボディ部に、JSONまたはXML形式で格納する。

[表8-6] Notification の新規作成のリクエストパラメータ

パラメータ名	型	説明
params	struct[]	キーが登録パラメータ名、値が登録値であるリスト。

- ・ <params> が空であってはならない。また、必ずグループ名を指定しなければならない。
- ・ <params> のキーは、グループの属性を示すプロパティである。ただし、キーとして、下記も利用できる。ただし、target と num はどちらか一方のみ指定できる。
 - name: グループ名
 - type: パラメータの発動条件。詳細は別表を参照のこと。
 - target: 発動対象のオープンデータ識別子。
 - value: 発動条件の閾値。
 - callback: コールバックURL (URLエンコードすること)。
 - num: 発行するucodeの個数。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表8-7] Notification の新規作成のステータスコード

ステータスコード	意味
201 Created	正常終了。
400 Bad Request	<params>が空である。<params>のキーが正しくない。targetとnumを両方指定した。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

レスポンスは、以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表8-8] Notification の新規作成のレスポンスパラメータ

パラメータ名	型	説明
ucode	xsd:anyURI []	作成されたNotification のucode。

【APIの利用例】

<クエリ>

```
POST /api/v1/triggers HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{ "name": "TriggerA",
  "target": [ "<urn:ucode:_00001C000000000000001000000100123>",
    "<urn:ucode:_00001C000000000000001000000100124>" ],
    "type": "any",
    "callback": "http://www.example.org/?ucode=%U" } }
```

<レスポンス1>

```
HTTP/1.1 201 Created
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"ucode": [ "<urn:ucode:_00001C000000000000001000000100127>" ]}
```

――[8-3: Notification 情報の閲覧]――

<機能概要>

Notification 情報を閲覧する。

<メソッド>

GET

<URLパス>

/api/v1/triggers/<ucode>

<制約条件>

<ucode>が指定するNotification に対する情報閲覧を許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。

[表8-9] Notification 情報の閲覧のリクエストパラメータ

パラメータ名	型	説明
ucode	xsd:anyURI []	Notification の識別子

<ステータスコード>

ステータスコードは以下の表の通りである。

[表8-10] Notification 情報の閲覧のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<ucode>が指定されていない。
404 Not Found	検索対象のNotificationが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

レスポンスは、以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表8-11] Notification 情報の閲覧のレスポンスパラメータ

パラメータ名	型	説明
target	xsd:anyURI []	発動対象のオープンデータ識別子
type	xsd:string	Notification の発動条件。詳細は別表を参照のこと。
value	xsd:string	発動条件の閾値。
callback	xsd:string	コールバックURL。
run (*)	xsd:boolean (*)	稼働していればtrue、そうでなければfalse 以下該当するNotificationに結びついている属性を返す。パラメータ名はNotification属性を示すプロパティである。

【APIの利用例】

<クエリ>

```
GET /api/v1/triggers/ucode_00001C000000000000001000000100126 HTTP/1.1
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{ "name": "TriggerA",
  "target": ["<urn:ucode:_00001C000000000000001000000100123>",
    "<urn:ucode:_00001C000000000000001000000100124>"],
    "type": "any",
    "callback": "http://www.example.org/?ucode=%U",
    "run": false }
```

——[8-4: Notification 情報の更新]——

<機能概要>

Notification 情報を更新する。

<メソッド>

PUT

<URLパス>

/api/v1/triggers/<ucode>

<制約条件>

<ucode>が指定するNotification に対する情報更新を許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータ(以下の表)のうち、更新対象Notification のucodeはURL 部分に指定する。それ以外の下記パラメータは、PUTメソッドのボディ部に、JSONまたはXML形式で格納する。

[表8-12] Notification 情報の更新のリクエストパラメータ

パラメータ名	型	説明
params	struct[]	キーがパラメータ名、値が登録値であるリスト。

- ・ <params>が空であってはならない。<params>のキーは、Notification の属性を示すプロパティである。ただし、キーとして、下記も利用できる。XML形式の場合、ucode値はucodeURL表現すること。
 - name: Notification 名
 - target: 発動対象のオープンデータucode
 - type: パラメータの発動条件。詳細は別表を参照のこと。
 - value: 発動条件の閾値。
 - callback: コールバックURL (URLエンコードすること)。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表8-13] Notification 情報の更新のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	<param1>、<value1>の組がない。<paramN>が正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	検索対象のNotificationが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
PUT /api/v1/triggers/ucode_00001C00000000000001000000100126 HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{ "name": "TriggerA",
  "target": ["<urn:ucode;_00001C00000000000001000000100123>",
    "<urn:ucode:_00001C00000000000001000000100124>"],
    "type": "any",
    "callback": "http://www.example.org/?ucode=%U" } }
```

<レスポンス>
HTTP/1.1 204 No Content
Connection: close

[8-5: Notification の削除]

<機能概要>
Notification を削除する。

<メソッド>
DELETE

<URLパス>
/api/v1/triggers/<ucode>

<制約条件>
<ucode>が指定するNotification に対する削除を許可されているユーザによるアクセスであること。

<リクエストパラメータ>
削除対象Notification のucodeはURI 部分に指定する。

<ステータスコード>
ステータスコードは以下の表の通りである。

[表8-14] Notification の削除のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	<ucode>が指定されていないか、正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	検索対象のNotificationが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>
成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>
DELETE /api/v1/triggers/ucode_00001C0000000000000001000000100126 HTTP/1.1
Host: www.example.org
Content-Length: 0
Content-Type: application/json; charset=utf-8

<レスポンス>

HTTP/1.1 204 No Content
Connection: close

――[8-6: Notification の起動・再開]――

<機能概要>

Notification を起動または再開する。

<メソッド>

PUT

<URLパス>

/api/v1/triggers/<ucode>/run

<制約条件>

<ucode>が指定するNotification に対する情報更新を許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータのうち、更新対象Notification のucodeはURL 部分に指定する。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表8-15] Notification の起動・再開のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	<param1>、<value1>の組がない。<paramN>が正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	検索対象のNotificationが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

PUT /api/v1/triggers/ucode_00001C00000000000001000000100126/run HTTP/1.1
Host: www.example.org
Content-Length: 0

<レスポンス>
HTTP/1.1 204 No Content
Connection: close

[8-7: Notification の停止]

<機能概要>
Notification を停止する。

<メソッド>
DELETE

<URLパス>
/api/v1/triggers/<ucode>/run

<制約条件>
<ucode>が指定するNotification に対する情報更新を許可されているユーザによるアクセスであること。

<リクエストパラメータ>
削除対象Notification のucodeはURI 部分に指定する。

<ステータスコード>
ステータスコードは以下の表の通りである。

[表8-16] Notification の停止のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	<ucode>が指定されていないか、正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	検索対象のNotificationが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>
成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>
DELETE /api/v1/triggers/ucode_00001C0000000000000001000000100126/run HTTP/1.1
Host: www.example.org
Content-Length: 0

<レスポンス>

HTTP/1.1 204 No Content

Connection: close

9. Vocabulary Management Command

公共交通情報流通連携基盤システム

Vocabulary Management Commandは、ボキャブラリ管理機能を実現するためのコマンドである。ボキャブラリは、RDF Schema 形式に基づいて入出力する。
本コマンドの一覧を、以下の表に示す。

[表9-1] Vocabulary Management Command の一覧

URLパス	HTTPメソッド	意味
/api/v1/vocabularies	GET	ボキャブラリを検索する
/api/v1/vocabularies	POST	ボキャブラリを登録する
/api/v1/vocabularies/<target>	GET	ボキャブラリを閲覧する
/api/v1/vocabularies/<target>/<property>	GET	ボキャブラリを閲覧する
/api/v1/vocabularies/<target>	PUT	ボキャブラリを更新する
/api/v1/vocabularies/<target>/<property>	PUT	ボキャブラリを更新する
/api/v1/vocabularies/<target>	DELETE	ボキャブラリを削除する
/api/v1/vocabularies/<target>/synonyms	GET	同義語を閲覧する
/api/v1/vocabularies/<target>/synonyms	PUT	同義語を更新する
/api/v1/vocabularies/<target>/parents	GET	親ボキャブラリを閲覧する
/api/v1/vocabularies/<target>/parents	PUT	親ボキャブラリを更新する
/api/v1/vocabularies/<target>/children	GET	子ボキャブラリを閲覧する

——[9-1: ボキャブラリの検索]——

<機能概要>

ボキャブラリを検索する。

<メソッド>

GET

<制約条件>

なし。誰でもリクエストできる。

<URLパス>

/api/v1/vocabularies?<param1 >=<value1 >&<param2 >=<value2 >

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。

[表9-2] ボキャブラリの検索のリクエストパラメータ

パラメータ名	デフォルト値	説明
paramN	(指定なし)	検索対象パラメータ名
valueN	(指定なし)	検索対象パラメータ値

- ・ <paramN>、<valueN> の組を1個以上指定しなければならない。<paramN> は以下のいずれかである。
 - RDF Schema で利用されるプロパティURI。
 - target。検索対象とするボキャブラリの識別子であり、型はxsd:anyURI [] である。URIがカンマを含むとき、それをURLエンコードすること。target が複数ある場合は、それらをカンマで区切ること。
 - offset, limit。パラメータ値はxsd:integer である。検索結果のうちoffset番目からlimit個分を要求することを意味する。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表9-3] ボキャブラリの検索のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<param1>、<value1>の組がない。<paramN>が正しくない。
404 Not Found	検索対象のボキャブラリが見つからない。
413 Request Entity Too Large	limit値が大きすぎる。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

レスポンスは、以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表9-4] ボキャブラリの検索のレスポンスパラメータ

パラメータ名	型	説明
vocabularies	RDF	RDF Schema 形式に基づくボキャブラリのリスト。レスポンス形式にXMLを指定した場合、各ボキャブラリ情報はRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各ボキャブラリ情報はRDF/JSONで表現される。
remains	xsd:boolean	条件にマッチするが、レスポンスに含まれないボキャブラリがある場合はtrue、ない場合はfalse。

【API の利用例】

＜クエリ＞

```
GET /api/v1/vocabularies?rdfs_label=Title HTTP/1.1
Host: www.example.org
```

＜レスポンス＞

[illegible]

```

    "http://www.w3.org/2000/01/rdf-schema#subPropertyOf": [
      { "value": "http://purl.org/dc/elements/1.1/title",
        "type": "uri" } ] } },
    "remains": false
  }

```

--- [9-2: ボキャブラリの新規作成] ---

<機能概要>

ボキャブラリを新規作成する。

<メソッド>

POST

<URLパス>

/api/v1/vocabularies

<制約条件>

なし。誰でもリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、POSTメソッドのボディ部に、JSONまたはXML形式で格納する。
(A)か(B)のどちらか1種類のみを指定できるが、(A)を推奨する。

[表9-5] ボキャブラリの新規作成のリクエストパラメータ (A)

パラメータ名	型	説明
rdf	RDF	RDF/XMLまたはRDF/JSONで記述されたRDFデータ。urn:ucode:_?<val>という形のURIを含めることにより、ucodeの自動発行を要求できる。

[表9-6] ボキャブラリの新規作成のリクエストパラメータ (B)

パラメータ名	型	説明
target	xsd:anyURI []	ボキャブラリの識別子
schema	xsd:string	RDF schema で記述されたボキャブラリ定義文書。
params	struct []	キーがプロパティのURI、値が登録値であるリスト。
num	xsd:integer	新規発行するucodeの個数。省略時は1。

- ・ <target>または<num>の一方のみを指定できる。
- ・ <rdf>、<params>、<schema>のどれか一つが指定されていなければならない。
- ・ <params> のキーは、RDF Schema のpredicateである。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表9-7] ボキャブラリの新規作成のステータスコード

ステータスコード	意味
201 Created	正常終了。
400 Bad Request	パラメータに rdf と params のどちらも指定されていない。 パラメータに rdf と params, target, num のいずれかが指定されている。 <params>のキーが正しくない。 パラメータに target と num の両方を指定した。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

クエリパラメータが(A)の形式であった場合のレスポンスは、以下の表(A)に示す構造データをJSONまたはXML形式で表現したものである。

クエリパラメータが(B)の形であった場合のレスポンスは、以下の表(B)に示す構造データをJSONまたはXML形式で表現したものである。

[表9-8] ボキャブラリの新規作成のレスポンスパラメータ (A)

パラメータ名	型	説明
ucode	struct	キーが指定された変数名、値が発行されたucodeのURI表現であるリスト。

[表9-9] ボキャブラリの新規作成のレスポンスパラメータ (B)

パラメータ名	型	説明
ucode	xsd:anyURI []	作成されたボキャブラリの識別ucode。

【APIの利用例】

<クエリ1>

POST /api/v1/vocabularies HTTP/1.1

Content-Length: xxx

Content-Type: application/json; charset=utf-8

Host: www.example.org

{ "rdf": [{

"ucode?_x": {

"http://www.w3.org/2002/07/owl#sameAs": [

{ "value": "http://purl.org/dc/terms/title",

"type": "uri" }],

"http://www.w3.org/2000/01/rdf-schema#label": [

{ "value": "Title", "type": "literal", "lang": "en-US" }],

"http://www.w3.org/2000/01/rdf-schema#comment": [

```

    {"value": "A name given to the resource.", "type": "literal",
    "lang": "en-US"} ],
    "http://www.w3.org/2000/01/rdf-schema#isDefinedBy": [
      {"value": "http://purl.org/dc/terms/", "type": "uri"} ],
    "http://purl.org/dc/terms/issued": [
      {"value": "2008-01-14", "type": "literal"} ],
    "http://purl.org/dc/terms/modified": [
      {"value": "2010-10-11", "type": "literal"} ],
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#type": [
      {"value":
        "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property",
        "type": "uri"} ],
    "http://purl.org/dc/terms/hasVersion": [
      {"value":
        "http://dublincore.org/usage/terms/history/#titleT-002",
        "type": "uri"} ],
    "http://www.w3.org/2000/01/rdf-schema#range": [
      {"value": "http://www.w3.org/2000/01/rdf-schema#Literal",
        "type": "uri"} ],
    "http://www.w3.org/2000/01/rdf-schema#subPropertyOf": [
      {"value": "http://purl.org/dc/elements/1.1/title",
        "type": "uri"} ] ] ] ] }

```

<レスポンス1>

```

HTTP/1.1 201 Created
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"unicode": {
  "unicode_?x": "urn:unicode:_OFFFDE0000000000000000000038035" } }

```

<クエリ2>

```

POST /api/v1/vocabularies HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{"schema": "<rdf:Description rdf:about=¥\"http://purl.org/dc/terms/title¥\">
<rdfs:label xml:lang=¥\"en-US¥\">Title</rdfs:label>
<rdfs:comment xml:lang=¥\"en-US¥\">
A name given to the resource.</rdfs:comment>
<rdfs:isDefinedBy rdf:resource=¥\"http://purl.org/dc/terms/¥\" />
<dcterms:issued>2008-01-14</dcterms:issued>
<dcterms:modified>2010-10-11</dcterms:modified>
<rdf:type rdf:resource=¥\"
http://www.w3.org/1999/02/22-rdf-syntax-ns#Property¥\" />
<dcterms:hasVersion rdf:resource=¥\"
http://dublincore.org/usage/terms/history/#titleT-002¥\" />
<rdfs:range rdf:resource=¥\"
http://www.w3.org/2000/01/rdf-schema#Literal¥\" />
<rdfs:subPropertyOf rdf:resource=¥\"

```

```
http://purl.org/dc/elements/1.1/title¥" />
</rdf:Description>"}}
```

<レスポンス2>

HTTP/1.1 201 Created

Content-Length: xxx

Connection: close

Content-Type: application/json; charset=utf-8

```
{"unicode":["<urn:unicode:_OFFFDE000000000000000000000000038035>"]}
```

[9-3: ボキャブラリの閲覧]

<機能概要>

ボキャブラリ情報を閲覧する。

<メソッド>

GET

<URLパス>

/api/v1/vocabularies/<target>

<制約条件>

なし。誰でもリクエストできる。

<リクエストパラメータ>

リクエストパラメータは、GETメソッドのURL部に格納する。

[表9-10] ボキャブラリの閲覧のリクエストパラメータ

パラメータ名	型	説明
target	xsd:anyURI []	ボキャブラリの識別子

<ステータスコード>

ステータスコードは以下の表の通りである。

〔表9-11〕 ボキャブラリの閲覧のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<target>が指定されていない。
404 Not Found	対象のボキャブラリが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

＜レスポンス＞

レスポンスは、以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表9-12] ボキャブラリの閲覧のレスポンスパラメータ

パラメータ名	型	説明
vocabularies	RDF	RDF Schema に基づく、指定されたボキャブラリのリスト。レスポンス形式にXMLを指定した場合、各ボキャブラリ情報はRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各ボキャブラリ情報はRDF/JSONで表現される。

【API の利用例】

＜クエリ＞

```
GET /api/v1/vocabularies/ucode_0FFFDE0000000000000000000038035 HTTP/1.1  
Host: www.example.org
```

＜レスポンス＞

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"vocabularies": {
    "urn:ucode:_OFFFDE000000000000000000000038035": {
        "http://www.w3.org/2002/07/owl#sameAs": [
            {"value":"http://purl.org/dc/terms/title",
             "type":"uri"} ],
        "http://www.w3.org/2000/01/rdf-schema#label": [
            {"value":"Title", "type":"literal", "lang":"en-US"} ],
        "http://www.w3.org/2000/01/rdf-schema#comment": [
            {"value":"A name given to the resource.", "type":"literal",
             "lang":"en-US"} ],
        "http://www.w3.org/2000/01/rdf-schema#isDefinedBy": [
            {"value":"http://purl.org/dc/terms/", "type":"uri"} ],
        "http://purl.org/dc/terms/issued": [
            {"value":"2008-01-14", "type":"literal"} ],
        "http://purl.org/dc/terms/modified": [
            {"value":"2010-10-11", "type":"literal"} ],
        "http://www.w3.org/1999/02/22-rdf-syntax-ns#type": [
```



```

    {"value":
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property",
    "type":"uri"} ],
    "http://purl.org/dc/terms/hasVersion": [
    {"value":
    "http://dublincore.org/usage/terms/history/#titleT-002",
    "type":"uri"} ],
    "http://www.w3.org/2000/01/rdf-schema#range": [
    {"value":"http://www.w3.org/2000/01/rdf-schema#Literal",
    "type":"uri"} ],
    "http://www.w3.org/2000/01/rdf-schema#subPropertyOf": [
    {"value":"http://purl.org/dc/elements/1.1/title",
    "type":"uri"} ] } } }

```

――[9-4: ボキャブラリ情報の閲覧(プロパティ指定)]――

<機能概要>

ボキャブラリ情報を閲覧する。

<メソッド>

GET

<URLパス>

/api/v1/vocabularies/<target>/<property>

<制約条件>

なし。誰でもリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。

[表9-13] ボキャブラリ情報の閲覧(プロパティ指定)のリクエストパラメータ

パラメータ名	型	説明
target	xsd:anyURI []	ボキャブラリの識別子
property	xsd:anyURI []	プロパティ値

<ステータスコード>

ステータスコードは以下の表の通りである。

[表9-14] ボキャブラリ情報の閲覧(プロパティ指定)のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<target>、<property>が指定されていない。
404 Not Found	対象のボキャブラリが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

指定されたプロパティをもつ目的語値。ボキャブラリの識別子もRDF schema のpredicateも1種類である場合は、該当する目的語値のリストを返す。そうでない場合は、以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表9-15] ボキャブラリ情報の閲覧(プロパティ指定)のレスポンスパラメータ

パラメータ名	型	説明
vocabularies	RDF	指定されたボキャブラリ情報のリスト。レスポンス形式にXMLを指定した場合、各ボキャブラリ情報はRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各ボキャブラリ情報はRDF/JSONで表現される。

【APIの利用例】

<クエリ1>

```
GET /api/v1/vocabularies/ucode_OFFFDE000000000000000000000038035/rdfs_label
HTTP/1.1
Host: www.example.org
```

<レスポンス1>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
[ "Title" ]
```

<クエリ2>

```
GET /api/v1/vocabularies/ucode_OFFFDE000000000000000000000038035,
ucode_OFFFDE000000000000000000000038036/rdfs_label HTTP/1.1
Host: www.example.org
```

<レスポンス2>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"vocabularies": [
{"urn:ucode_OFFFDE000000000000000000000038035" : {
```

—[9-5: ボキャブラリ情報の更新]—

ボキャブラリ情報を更新する。

PUT

```
/api/v1/vocabularies/<target>
```

<target>が指定するボキャブラリに対する情報更新を許可されているユーザによるアクセスであること。

リクエストパラメータのうち、更新対象ボキャブラリ<target> はURI 部分に指定する。それ以外のパラメータ(以下の表)は、PUTメソッドのボディ部に、JSONまたはXML形式で格納する。

パラメータ名	型	説明
rdf	RDF	RDF/XMLまたはRDF/JSON形式で記した更新情報。更新情報のsubject は、<target> と一致していること。コマンド終了後、更新情報に含まれるpredicate に対する値は、個数を含めてリクエストパラメータが指定した情報と完全に一致する。更新情報に含まれないpredicate に関する値は変化しない。
params	struct[]	キーが登録パラメータ名、値が登録値であるリスト。コマンド終了後、キーに指定されたpredicate に対する値は、個数を含めてリクエストパラメータが指定した情報と完全に一致する。リクエストパラメータのキーにないpredicate に関する値は変化しない。
schema	xsd:string	RDF schema で記述されたボキャブラリ定義文書。コマンド終了後のボキャブラリ情報は、指定したボキャブラリ定義文書と完全に一致する。指定したボキャブラリ定義文書にないpredicate に関する値は削除される。

- 2013年3月29日

＜ステータスコード＞

ステータスコードは以下の表の通りである。

「表9-17」 ボキャブラリ情報の更新のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータが正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	対象のボキャブラリが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

＜レスポンス＞

成功時、レスポンスボディは空である。

【API の利用例】

<クエリ1>

```
PUT /api/v1/vocabularies/ucode_0FFFDE000000000000000000000000000000000000000000000000000000000000 HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{"rdf": {
    "urn:ucode:_0FFFDE000000000000000000000000000000000000000000000000000000000000": {
        "http://purl.org/dc/terms/hasVersion": [
            { "value": "http://dublincore.org/usages/terms/history/#titleT-002",
              "type": "uri" } ] ] ] }
```

＜レスポンス1＞

```
HTTP/1.1 204 No Content
Connection: close
```

<クエリ?>

```
PUT /api/v1/vocabularies/ucode_OFFFDE000000000000000000000038035 HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{"schema":
  "<rdf:Description rdf:about='\"http://purl.org/dc/terms/title\"">
  <rdfs:label xml:lang='\"en-US\"">Title</rdfs:label>
  <rdfs:comment xml:lang='\"en-US\"">
  A name given to the resource.</rdfs:comment>
  <rdfs:isDefinedBy rdf:resource='\"http://purl.org/dc/terms/\"' />
  <dcterms:issued>2008-01-14</dcterms:issued>
  <dcterms:modified>2010-10-11</dcterms:modified>
  <rdf:type rdf:resource='\"
  http://www.w3.org/1999/02/22-rdf-syntax-ns#Property\"' />
  <dcterms:hasVersion rdf:resource='\"
  http://dublincore.org/usage/terms/history/#titleT-002\"' />
```

```
<rdfs:range rdf:resource=¥"
http://www.w3.org/2000/01/rdf-schema#Literal¥" />
<rdfs:subPropertyOf rdf:resource=¥"
http://purl.org/dc/elements/1.1/title¥" />
</rdf:Description>"}

```

<レスポンス2>

HTTP/1.1 204 No Content
Connection: close

――[9-6: ボキャブラリ情報の更新(プロパティ指定)]――

<機能概要>

プロパティを指定して、ボキャブラリ情報を更新する。

<メソッド>

PUT

<URLパス>

/api/v1/vocabularies/<target>/<property>

<制約条件>

<target>が指定するボキャブラリに対する情報更新を許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータのうち、更新対象ボキャブラリ<target>とプロパティ<property>はURI 部分に指定する。更新する値を、PUTメソッドのボディ部に、JSONまたはXML形式で格納する。本コマンド終了後、更新対象ボキャブラリ<target>とプロパティ<property>をもつオブジェクトは、リクエストパラメータに指定した値のみとなる。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表9-18] ボキャブラリ情報の更新(プロパティ指定)のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	<target>、<property>が正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	検索対象のボキャブラリが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

＜レスポンス＞

成功時、レスポンスボディは空である。

【API の利用例】

＜クエリ＞

```
PUT /api/v1/vocabularies/ucode_0FFFDE000000000000000000000000000000000000000000000000000000000000/rdfs_label
HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
[ "title" ]
```

＜レスポンス＞

```
HTTP/1.1 204 No Content
Connection: close
```

——[9-7: ボキャブラリの削除]——

＜機能概要＞

ボキャブラリを削除する。

＜メソッド＞

DELETE

<URLパス>

```
/api/v1/vocabularies/<target>
```

＜制約条件＞

<target>が指定するボキャブラリに対する削除を許可されているユーザによるアクセスであること。

＜リクエストパラメータ＞

削除対象ボキャブラリはURI 部分に指定する。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表9-19] ボキャブラリの削除のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	<target>が指定されていないか、正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	検索対象のボキャブラリが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
DELETE /api/v1/vocabularies/ucode_0FFFDE000000000000000000000000000038035 HTTP/1.1
Host: www.example.org
Content-Length: 0
```

<レスポンス>

```
HTTP/1.1 204 No Content
Content-Length: 0
Connection: close
```

――[9-8: 同意語の検索]――

<機能概要>

指定したボキャブラリの同意語(owl:sameAs で結ばれているボキャブラリ)を検索する。

<メソッド>

GET

<URLパス>

/api/v1/vocabularies/<target>/synonyms

<制約条件>

なし。誰でもリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。

〔表9-20〕 同意語の検索のリクエストパラメータ

パラメータ名	型	説明
target	xsd:anyURI	ボキャブラリの識別子

＜ステータスコード＞

ステータスコードは以下の表の通りである。

「表9-21」 同意語の検索のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<target>が指定されていない。
404 Not Found	対象のボキャブラリが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

＜レスポンス＞

レスポンスは、以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表9-22] 同意語の検索のレスポンスパラメータ

パラメータ名	型	説明
vocabularies	xsd:anyURI []	指定したボキャブラリの同意語のリスト。

【API の利用例】

＜クエリ＞

```
GET /api/v1/vocabularies/ucode_0FFFDE00000000000000000000000038035/synonyms  
HTTP/1.1  
Host: www.example.org
```

＜レスポンス1＞

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"vocabularies":["<urn:ucode: 0FFFDE00000000000000000000000000002800E>"]}
```


——[9-9: 同意語情報の更新]——

<機能概要>

ボキャブラリの同意語情報を更新する。

<メソッド>

PUT

<URLパス>

/api/v1/vocabularies/<target>/synonyms

<制約条件>

<target>が指定するボキャブラリに対する情報更新を許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータのうち、更新対象ボキャブラリ<target> はURI 部分に指定する。それ以外のパラメータ(以下の表)は、PUTメソッドのボディ部に、JSONまたはXML形式で格納する。

[表9-23] 同意語情報の更新のリクエストパラメータ

パラメータ名	型	説明
synonyms	xsd:anyURI []	指定したボキャブラリの同意語ucodeのリスト。

- ・本コマンド終了後、更新対象ボキャブラリの同義語は、リクエストパラメータに指定した値のみとなる。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表9-24] 同意語情報の更新のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	<synonyms>がない。<synonyms>が正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	検索対象のボキャブラリが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
PUT /api/v1/vocabularies/ucode_OFFFDE000000000000000000000038035/synonyms
HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{"synonyms":["<urn:ucode:_OFFFDE000000000000000000002800E>"]}
```

<レスポンス>

```
HTTP/1.1 204 No Content
Connection: close
```

――[9-10: 親ボキャブラリの検索]――

<機能概要>

指定したボキャブラリの親ボキャブラリ(rdfs:subPropertyOf, rdfs:subClassOf の先にあるリソース)を検索する。

<メソッド>

GET

<URLパス>

/api/v1/vocabularies/<target>/parents

<制約条件>

なし。誰でもリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。

[表9-25] 親ボキャブラリの検索のリクエストパラメータ

パラメータ名	型	説明
target	xsd:anyURI	ボキャブラリの識別子。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表9-26] 親ボキャブラリの検索のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<target>が指定されていない。
404 Not Found	対象のボキャブラリが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

＜レスポンス＞

レスポンスは、以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表9-27] 親ボキャブラリの検索のレスポンスパラメータ

パラメータ名	型	説明
vocabularies	xsd:anyURI []	指定したボキャブラリの親ボキャブラリのリスト。

【API の利用例】

＜クエリ＞

[illegible]

＜レスポンス＞

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"vocabularies":["<urn:ucode: 0FFFDE00000000000000000000000000002800E>"]}

```

—[9-11: 親ボキャブラリ情報の更新]—

＜機能概要＞

ボキャブラリの親ボキャブラリ (rdfs:subPropertyOf, rdfs:subClassOf の先にあるリソース) を更新する。

＜メソッド＞

PUT

<URLパス>

```
/api/v1/vocabularies/<target>/parents
```

＜制約条件＞

<target>が指定するボキャブラリに対する情報更新を許可されているユーザによるアクセスであること。

＜リクエストパラメータ＞

リクエストパラメータのうち、更新対象ボキャブラリ<target> はURI 部分に指定する。親ボキャブラリは、PUTメソッドのボディ部に、JSONまたはXML形式で格納する。

本コマンド終了後、更新対象ボキャブラリの親ボキャブラリは、リクエストパラメータに指定した値のみとなる。

＜ステータスコード＞

ステータスコードは以下の表の通りである。

「表9-28」 親ボキャブラリ情報の更新のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータが正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	検索対象のボキャブラリが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

＜レスポンス＞

成功時、レスポンスボディは空である。

【API の利用例】

＜クエリ＞

[illegible]

HTTP/1.1

Host: www.example.org

Content-Length: xxx

Content-Type: application/json; charset=utf-8

```
["<urn:ucode: 0FFFDE000000000000000000000000002800E>"]
```

＜レスポンス＞

HTTP/1.1 204 No Content

Connection: close

——[9-12: 子ボキャブラリの検索]——

<機能概要>

指定したボキャブラリの子ボキャブラリ (rdfs:subPropertyOf, rdfs:subClassOf の前にあるリソース) を検索する。

<メソッド>

GET

<URLパス>

/api/v1/vocabularies/<target>/children

<制約条件>

なし。誰でもリクエストできる。

<リクエストパラメータ>

リクエストパラメータ (以下の表) は、GETメソッドのURL部に格納する。

[表9-29] 子ボキャブラリの検索のリクエストパラメータ

パラメータ名	型	説明
target	xsd:anyURI	ボキャブラリの識別子。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表9-30] 子ボキャブラリの検索のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<target>が指定されていない。
404 Not Found	対象のボキャブラリが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

レスポンスは、以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表9-31] 子ボキャブラリの検索のレスポンスパラメータ

パラメータ名	型	説明
vocabularies	xsd:anyURI []	指定したボキャブラリの子ボキャブラリのリスト。

【APIの利用例】

<クエリ>

```
GET /api/v1/vocabularies/ucode_OFFFDE0000000000000000000002800E/children
HTTP/1.1
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"vocabularies":["<urn:ucode:_OFFFDE00000000000000000000038035>"]}
```

10. Triple Management Command

公共交通情報流通連携基盤システム

Triple Management Commandは、センサやスマートメータなどの小型機器がRDFモデルの主語・述語・目的語からなるTriple を効率的に扱うために、利用者プログラムが「標準データ規格」を簡素化したオープンデータ操作を行うためのコマンドである。

本コマンドの一覧を、以下の表に示す。

[表10-1] Triple Management Command の一覧

URLパス	HTTPメソッド	意味
/api/v1/datapoints	GET	オープンデータを検索する
/api/v1/datapoints	POST	オープンデータを登録する
/api/v1/datapoints/<target>	GET	オープンデータを閲覧する
/api/v1/datapoints/<target>/<property>	GET	オープンデータを閲覧する
/api/v1/datapoints/<target>	PUT	オープンデータを更新する
/api/v1/datapoints/<target>/<property>	PUT	オープンデータを更新する
/api/v1/datapoints/<target>	DELETE	オープンデータを削除する
/api/v1/datapoints/<target>/<property>	DELETE	オープンデータの属性値を削除する

——[10-1: オープンデータ検索]——

<機能概要>

オープンデータを検索する。

<メソッド>

GET

<制約条件>

なし。誰でもリクエストできる。

<URLパス>

/api/v1/datapoints?<param1 >=<value1 >&<param2 >=<value2 >

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。

[表10-2] オープンデータ検索のリクエストパラメータ

パラメータ名	デフォルト値	説明
paramN	(指定なし)	検索対象パラメータ名
valueN	(指定なし)	検索対象パラメータ値

- ・ <paramN>、<valueN> の組を1個以上指定しなければならない。<paramN> は以下のいずれかである。
 - オープンデータ属性を示すプロパティURI。
 - target。検索対象とするオープンデータの識別子であり、パラメータ値の型はxsd:anyURI [] である。URI がカンマを含むとき、それをURLエンコードすること。target が複数ある場合は、それらをカンマで区切ること。
 - stream。Stream API に基づくコネクションを、パラメータ値で指定する秒数継続する。
 - offset, limit。パラメータ値はxsd:integer である。検索結果のうちoffset番目からlimit個分を要求することを意味する。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表10-3] オープンデータ検索のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<param1>、<value1>の組がない。<paramN>が正しくない。
404 Not Found	検索対象のオープンデータが見つからない。
413 Request Entity Too Large	limit値が大きすぎる。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

レスポンスは、以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表10-4] オープンデータ検索のレスポンスパラメータ

パラメータ名	型	説明
results	RDF	検索の結果得られたオープンデータ。レスポンス形式にXMLを指定した場合、データはRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、データはRDF/JSONで表現される。
remains	xsd:boolean	条件にマッチするが、レスポンスに含まれない場所がある場合はtrue、ない場合はfalse。

【API の利用例】

＜クエリ＞

```
GET /api/v1/datapoints?dc_title=ABC%20meter HTTP/1.1
Host: www.example.org
```

＜レスポンス＞

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"results": {
    "urn:ucode:_00001C000000000000001000000100124": {
        "http://purl.org/dc/elements/1.1/title" : [
            { "value" : "ABC Meter", "type" : "literal" } ] ],
    "remains":false
}
```

—[10-2: オープンデータの新規作成]—

＜機能概要＞

オープンデータを新規作成する。

＜メソッド＞

POST

<URLパス>

```
/api/v1/datapoints
```

＜制約条件＞

なし。誰でもリクエストできる。

＜リクエストパラメータ＞

リクエストパラメータ(以下の表)は、POSTメソッドのボディ部に、JSONまたはXML形式で格納する。
(A)か(B)のどちらか1種類のみを指定できるが、(A)を推奨する。

[表10-5] オープンデータの新規作成のリクエストパラメータ (A)

パラメータ名	型	説明
rdf	RDF	RDF/XMLまたはRDF/JSONで記述されたRDFデータ。urn:ucode: _?<val>という形のURIを含めることにより、ucodeの自動発行を要求できる。

[表10-6] オープンデータの新規作成のリクエストパラメータ (B)

パラメータ名	型	説明
target	xsd:anyURI []	オープンデータの識別子
params	struct[]	キーがプロパティのURI、値が登録値であるリスト。
num	xsd:integer	新規発行するucodeの個数。省略時は1。

- ・ <target>または<num> の一方のみを指定できる。
- ・ <rdf>と<params> のどちらか一方は指定されていなければならない。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表10-7] オープンデータの新規作成のステータスコード

ステータスコード	意味
201 Created	正常終了。
400 Bad Request	パラメータに rdf と params のどちらも指定されていない。 パラメータに rdf と params, target, num のいずれかが指定されている。 <params>のキーが正しくない。
409 Conflict	パラメータに target と num の両方を指定した。 指定したオープンデータの識別子は、すでに情報流通連携基盤システムに登録されている。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

クエリパラメータが(A)の形式であった場合のレスポンスは、以下の表(A)に示す構造データをJSONまたはXML形式で表現したものである。

クエリパラメータが(B)の形であった場合のレスポンスは、以下の表(B)に示す構造データをJSONまたはXML形式で表現したものである。

[表10-8] オープンデータの新規作成のレスポンスパラメータ (A)

パラメータ名	型	説明
ucode	struct	キーが指定された変数名、値が発行されたucodeのURI表現であるリスト。

[表10-9] オープンデータの新規作成のレスポンスパラメータ (B)

パラメータ名	型	説明
ucode	xsd:anyURI []	作成されたオープンデータの識別ucode。

【APIの利用例】

<クエリ1>

```
POST /api/v1/datapoints HTTP/1.1
Content-Length: xxx
Content-Type: application/json; charset=utf-8
Host: www.example.org
{"rdf": {
  "<urn:ucode:_?x>": {
    "http://purl.org/dc/elements/1.1/title": [
      { "value": "ABC Meter", "type": "literal" } ] } }
```

<レスポンス1>

```
HTTP/1.1 201 Created
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"ucode ": {
  "ucode:_?x": "urn:ucode:_00001C00000000000001000000100125"}
```

<クエリ2>

```
POST /api/v1/datapoints HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{"params": {"<dc:title>": "ABC meter"}}
```

<レスポンス2>

```
HTTP/1.1 201 Created
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"ucode": ["<urn:ucode:_00001C00000000000001000000100125>"]}
```

――[10-3: オープンデータの閲覧]――

<機能概要>

オープンデータを閲覧する。

<メソッド>

GET

<URLパス>

/api/v1/datapoints/<target>

<制約条件>

<target>が指定するオープンデータに対する情報閲覧を許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。
また、クエリパラメータとして?stream を指定できる。このときは、Stream API に基づく接続を指定された秒数継続する(「2.8 Streams API」参照)。

[表10-10] オープンデータの閲覧のリクエストパラメータ

パラメータ名	型	説明
target	xsd:anyURI []	オープンデータの識別子

<ステータスコード>

ステータスコードは以下の表の通りである。

[表10-11] オープンデータの閲覧のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<target>が指定されていない。
404 Not Found	対象のオープンデータが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

レスポンスは、以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表10-12] オープンデータの閲覧のレスポンスパラメータ

パラメータ名	型	説明
results	RDF	指定されたデータのリスト。レスポンス形式にXMLを指定した場合、各データはRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各データはRDF/JSONで表現される。

【APIの利用例】

<クエリ>

```
GET /api/v1/datapoints/ucode_00001C000000000000001000000100124 HTTP/1.1
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"results": [
  {"urn:ucode:_00001C000000000000001000000100124" : {
    "http://purl.org/dc/elements/1.1/title" : [
      { "value" : "ABC Meter", "type" : "literal" } ] ] ] ] }
```

——[10-4: オープンデータの閲覧(プロパティ指定)]——

<機能概要>

プロパティを指定して、オープンデータを閲覧する。

<メソッド>

GET

<URLパス>

/api/v1/datapoints/<target>/<property>

<制約条件>

<target>が指定するオープンデータに対する情報閲覧を許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。また、クエリパラメータとして?stream を指定できる。このときは、Stream API に基づくコネクションを指定された秒数継続する(「2.8 Streams API」参照)。

[表10-13] オープンデータの閲覧(プロパティ指定)のリクエストパラメータ

パラメータ名	型	説明
target property	xsd:anyURI [] xsd:anyURI []	オープンデータの識別子 プロパティ値

<ステータスコード>

ステータスコードは以下の表の通りである。

[表10-14] オープンデータの閲覧(プロパティ指定)のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<target>、<property>が指定されていない。
404 Not Found	対象のオープンデータが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

指定されたプロパティをもつ目的語値。オープンデータの識別子もプロパティの識別子も1種類である場合は、該当する目的語値のリストを返す。そうでない場合は、以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表10-15] オープンデータの閲覧(プロパティ指定)のレスポンスパラメータ

パラメータ名	型	説明
results	RDF	指定されたデータのリスト。レスポンス形式にXMLを指定した場合、各データはRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各データはRDF/JSONで表現される。

【APIの利用例】

<クエリ1>

```
GET /api/v1/datapoints/ucode_00001C000000000000001000000100124/dc_title
HTTP/1.1
Host: www.example.org
```

<レスポンス1>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
["ABC meter"]
```

<クエリ2>

```
GET /api/v1/users/ucode_00001C000000000000001000000100124,  
ucode_00001C000000000000001000000100125/dc, title HTTP/1.1  
Host: www.example.org
```

<レスポンス2>

```
HTTP/1.1 200 OK  
Content-Length: xxx  
Connection: close  
Content-Type: application/json; charset=utf-8  
{ "results": [  
  {"urn:ucode_00001C000000000000001000000100124" : {  
    "http://purl.org/dc/elements/1.1/title" : [  
      { "value" : "ABC Meter", "type" : "literal" } ] } },  
  {"urn:ucode_00001C000000000000001000000100125" : {  
    "http://purl.org/dc/elements/1.1/title" : [  
      { "value" : "XYZ Meter", "type" : "literal" } ] } } ] ] }
```

[10-5: オープンデータの更新]

<機能概要>

オープンデータを更新する。

<メソッド>

PUT

<URLパス>

/api/v1/datapoints/<target>

<制約条件>

<target>が指定するオープンデータに対する情報更新を許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータのうち、更新対象オープンデータのucodeはURI 部分に指定する。それ以外のパラメータ(以下の表)は、PUTメソッドのボディ部に、JSONまたはXML形式で格納する。

[表10-16] オープンデータの更新のリクエストパラメータ

パラメータ名	型	説明
rdf	RDF	RDF/XMLまたはRDF/JSON形式で記した更新情報。更新情報のsubject は、<target> と一致していること。コマンド終了後、リクエストパラメータに含まれるpredicateに対する値は、個数を含めてリクエストパラメータが指定した情報と完全に一致する。
params	struct[]	キーが登録パラメータ名、値が登録値であるリスト。コマンド終了後、キーに指定されたプロパティに対する値は、個数を含めてリクエストパラメータが指定した情報と完全に一致する。リクエストパラメータのキーにないプロパティに関する値は変化しない。

- ・ params または rdf のいずれか一方のみが指定されなければならない。 rdf を推奨する。
- ・ params のキーは、オープンデータの属性を示すプロパティである。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表10-17] オープンデータの更新のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	<paras>が正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	検索対象のオープンデータが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ1>

```
PUT /api/v1/datapoints/ucode_00001C000000000000001000000100124 HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{"rdf": {
  "urn:ucode:_00001C000000000000001000000100124": {
    "http://purl.org/dc/elements/1.1/": [
      { "value": "ABC meter", "type": "literal" } ] ] }
```


<レスポンス1>

HTTP/1.1 204 No Content
Connection: close

<クエリ2>

PUT /api/v1/datapoints/ucode_00001C00000000000001000000100124 HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{`"<dc:title>": "ABC meter"`}

<レスポンス2>

HTTP/1.1 204 No Content
Connection: close

——[10-6: オープンデータの更新(プロパティ指定)]——

<機能概要>

プロパティを指定して、オープンデータを更新する。

<メソッド>

PUT

<URLパス>

/api/v1/datapoints/<target>/<property>

<制約条件>

<target>が指定するオープンデータに対する情報更新を許可されているユーザによるアクセスであること。

<リクエストパラメータ>

リクエストパラメータのうち、更新対象オープンデータ識別子<target> とプロパティ<property> はURI 部分に指定する。更新値は、PUTメソッドのボディ部に、JSONまたはXML形式で格納する。

本コマンド終了後、更新対象の識別子URI とプロパティをもつオブジェクトは、リクエストパラメータに指定した値のみとなる。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表10-18] オープンデータの更新(プロパティ指定)のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータが正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	検索対象のオープンデータが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
PUT /api/v1/datapoints/ucode_00001C000000000000001000000100124/dc_title
HTTP/1.1
Host: www.example.org
Content-Length: xxx
[ "ABC meter" ]
```

<レスポンス>

```
HTTP/1.1 204 No Content
Connection: close
```

——[10-7: オープンデータの削除]——

<機能概要>

オープンデータを削除する。

<メソッド>

DELETE

<URLパス>

/api/v1/datapoints/<target>

<制約条件>

<target>が指定するオープンデータに対する削除を許可されているユーザによるアクセスであること。

<リクエストパラメータ>

削除対象オープンデータ<target> はURL 部分に指定する。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表10-19] オープンデータの削除のステータスコード

ステータスコード	意味
204 No Contents	正常終了。
400 Bad Request	<target>が指定されていないか、正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	検索対象のオープンデータが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
DELETE /api/v1/datapoints/ucode_00001C000000000000001000000100124 HTTP/1.1
Host: www.example.org
Content-Length: xxx
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
```

――[10-8: オープンデータの属性削除]――

<機能概要>

オープンデータの指定した属性を削除する。

<メソッド>

DELETE

<URLパス>

/api/v1/datapoints/<target>/<property>

<制約条件>

<target>が指定するオープンデータに対する情報更新を許可されているユーザによるアクセスであること。

<リクエストパラメータ>

削除対象オープンデータのucodeと属性はURI 部分に指定する。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表10-20] オープンデータの属性削除のステータスコード

ステータスコード	意味
204 No Content	正常終了。
400 Bad Request	パラメータが正しくない。
403 Forbidden	アクセス権限がない。
404 Not Found	検索対象のオープンデータが見つからない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
DELETE /api/v1/datapoints/ucode_00001C000000000000001000000100124/dc_title
HTTP/1.1
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 204 No Content
Connection: close
```

11. Identification Resolution Command

公共交通情報流通連携基盤システム

Identification Resolution Commandは、利用者プログラムが、モノ・場所・データの識別子から、その識別子が指し示す対象に関するオープンデータが格納されているサーバを得るための、ディレクトリ型検索の機能を提供するコマンドである。
本コマンドの一覧を、以下の表に示す。

[表11-1] Identification Resolution Command の一覧

URLパス	HTTPメソッド	意味
/api/v1/rs/<ucode>	GET	簡易ucode解決を行う
/api/v1/resolve/<ucode>	GET	ucodeからオープンデータの参照先を得る
/api/v1/resolve	POST	ucodeとオープンデータの参照先との組み (ucode解決情報)を作成する
/api/v1/resolve/<ucode>	PUT	ucode解決情報を更新する
/api/v1/resolve/<ucode>	DELETE	ucode解決情報を削除する

――[11-1: 簡易ucode解決]――

<機能概要>

簡易ucode解決プロトコル に基づいたucode解決機能を提供する。すなわち、ucodeに結びついた情報の参照先を取得する。

<メソッド>

GET

<URLパス>

/api/v1/rs/<ucode>?<param1 >=<value1 >&<param2 >=<value2 >

<制約条件>

なし。誰でもリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。

[表11-2] 簡易ucode解決のリクエストパラメータ

パラメータ名	型	説明
ucode paramN	xsd:anyURI std:string	対象のucode 解決パラメータ

本コマンドが使用する具体的な値を以下の表に列記する。

[表11-3] ucode解決のパラメータ

パラメータ名	デフォルト値	説明
X-UIDC-GWMODE	resolveall	解決モード
X-UIDC-QUERYMASK	all_1	識別子解決のマスク値
X-UIDC-QUERYATTRIBUTE	UIDC_ATTR_ANONYMOUS	取得する解決情報の属性

- ・ 解決モードの値は以下の通り。

resolveall:	識別子解決(多段解決)
resolve:	識別子解決(1階層解決)
redirect:	識別子解決とHTTPリダイレクト

- ・ 取得する解決情報の属性の値は以下の通り。

UIDC_ATTR_ANONYMOUS:	指定しない
UIDC_ATTR_RS:	解決サーバ
UIDC_ATTR_IS:	情報サーバ
UIDC_ATTR_USER:	ユーザ定義情報

<ステータスコード>

ステータスコードは以下の表の通りである。

[表11-4] 簡易ucode解決のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<ucode>が指定されていない。
403 Forbidden	アクセス権限がない。
404 Not Found	対象のucodeが情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

レスポンスは、以下の表の構造データをJSONまたはXML形式で表現したものである。ただし、X-UIDC-GWMODE パラメータにredirect を指定した場合は、解決先URLにリダイレクトする。

[表11-5] 簡易ucode解決のレスポンスパラメータ

パラメータ名	型	説明
results	struct[]	解決情報のリスト。
└─X-UIDC-DATA	xsd:string	解決結果データ
└─X-UIDC-DATAVERSION	xsd:integer	解決結果データのバージョン
└─X-UIDC-DATATYPE	xsd:integer	解決結果のデータタイプ
└─X-UIDC-RETURNMASK	xsd:string	解決結果のビットマスク
└─X-UIDC-TTL	xsd:integer	解決結果の有効期限
└─X-UIDC-RESOLVEMODE	xsd:integer	解決モード

【APIの利用例】

<クエリ>

```
GET /api/v1/rs/ucode_00001C000000000000001000000100123 HTTP/1.1
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"results":[{"X-UIDC-DATA":"http://www.example.org/",
"X-UIDC-DATATYPE":17,
"X-UIDC-RETURNMASK":"FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF",
"X-UIDC-TTL":100,
"X-UIDC-RESOLVEMODE":0 } ] }
```

——[11-2: ucode解決(ucodeからオープンデータの参照先の取得)]——

<機能概要>

ucodeに結びつけられたオープンデータの参照先を取得する。

<メソッド>

GET

<URLパス>

/api/v1/resolve/<ucode>?<param1 >=<value1 >&<param2 >=<value2 >

<制約条件>

なし。誰でもリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、GETメソッドのURL部に格納する。

[表11-6] ucode解決(ucodeからオープンデータの参照先の取得)のリクエストパラメータ

パラメータ名	型	説明
ucode paramN	xsd:anyURI std:string	対象のucode 解決パラメータ

<ステータスコード>

ステータスコードは以下の表の通りである。

[表11-7] ucode解決(ucodeからオープンデータの参照先の取得)のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<ucode>が指定されていない。
403 Forbidden	アクセス権限がない。
404 Not Found	対象のucodeが情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

レスポンスは、以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表11-8] ucode解決(ucodeからオープンデータの参照先の取得)のレスポンスパラメータ

パラメータ名	型	説明
results	struct[]	解決情報のリスト。各情報は以下の構造をもつ。
X-UIDC-ATTRIBUTE	xsd:string	解決結果の属性値
X-UIDC-DATA	xsd:integer	解決結果データ
X-UIDC-RETURNMASK	xsd:string	解決結果のビットマスク
X-UIDC-RESOLVEMODE	xsd:integer	解決モード

【APIの利用例】

<クエリ>

```
GET /api/v1/resolve/ucode_00001C000000000000001000000100123?  
uc_issuer=ucode_00001C000000000000001000000100124 HTTP/1.1  
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK  
Content-Length: xxx  
Connection: close  
Content-Type: application/json; charset=utf-8  
{"results":[{"
```



```

"X-UIDC-DATA": "http://www. example. org/",
"X-UIDC-DATATYPE": 17,
"X-UIDC-RETURNMASK": "FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF",
"X-UIDC-RESOLVEMODE": 0 }, {
"X-UIDC-DATA": "http://www. example2. org/",
"X-UIDC-DATATYPE": 17,
"X-UIDC-RETURNMASK": "FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF",
"X-UIDC-RESOLVEMODE": 0 } ] }

```

――[11-3: ucode解決情報の新規作成]――

<機能概要>

ucodeに対して、オープンデータの参照先を結びつける。

<メソッド>

POST

<URLパス>

/api/v1/resolve

<制約条件>

なし。誰でもリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、POSTメソッドのボディ部に、JSONまたはXML形式で格納する。

[表11-9] ucode解決情報の新規作成のリクエストパラメータ

パラメータ名	型	説明
target params	xsd:anyURI struct[]	対象のucode キーが登録パラメータ名、値が登録値であるリスト。

・ <target>、<params>が空であってはならない。<params> のキーは、プロパティURIなどに準ずるパラメータである。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表11-10] ucode解決情報の新規作成のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<target>が指定されていない。
403 Forbidden	アクセス権限がない。
404 Not Found	対象のucodeが情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

レスポンスは、以下の表に示す構造データをJSONまたはXML形式で表現したものである。

[表11-11] ucode解決情報の新規作成のレスポンスパラメータ

パラメータ名	型	説明
ucode	xsd:anyURI []	作成されたucode解決情報を識別するucode。

【APIの利用例】

<クエリ>

```
POST /api/v1/resolve HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{"target": "<urn:ucode:_00001C0000000000000010000000100100>",
"params": {"X-UIDC-ATTRIBUTE": "17", "X-UIDC-DATA": "http://www.example.org/"}}
```

<レスポンス1>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{"ucode": ["<urn:ucode:_00001C0000000000000010000000100125>"]}
```

——[11-4: ucode解決情報の更新]——

<機能概要>

ucodeに対して、オープンデータの参照先を結びつけを更新する。

<メソッド>

PUT

<URLパス>

/api/v1/resolve/<ucode>

<制約条件>

ucode解決情報を登録したユーザがリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、PUTメソッドのボディ部に、JSONまたはXML形式で格納する。

[表11-12] ucode解決情報の更新のリクエストパラメータ

パラメータ名	型	説明
ucode	xsd:anyURI	ucode解決情報を識別するucode
target	xsd:anyURI	対象のucode
params	struct[]	キーが登録パラメータ名、値が登録値であるリスト。

・ <target>、<params>が空であってはならない。<params> のキーは、プロパティURIなどに準ずるパラメータである。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表11-13] ucode解決情報の更新のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<target>が指定されていない。
403 Forbidden	アクセス権限がない。
404 Not Found	対象のucodeが情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
PUT /api/v1/resolve/ucode_00001C0000000000000010000000100125 HTTP/1.1
Host: www.example.org
Content-Length: xxx
Content-Type: application/json; charset=utf-8
{"target": "<urn:ucode:_00001C0000000000000010000000100100>",
"params": {
  "X-UIDC-ATTRIBUTE": 17,
  "X-UIDC-DATA": "http://www.example.org/" } }
```

<レスポンス>

HTTP/1.1 204 No Content
Connection: close

――[11-5: ucode解決情報の削除]――

<機能概要>

ucodeに対して、オープンデータと参照先との結びつけを削除する。

<メソッド>

DELETE

<URLパス>

/api/v1/resolve/<ucode>

<制約条件>

ucode解決情報を登録したユーザがリクエストできる。

<リクエストパラメータ>

リクエストパラメータ(以下の表)は、PUTメソッドのボディ部に、JSONまたはXML形式で格納する。

[表11-14] ucode解決情報の削除のリクエストパラメータ

パラメータ名	型	説明
ucode	xsd:anyURI	ucode解決情報を識別するucode

<ステータスコード>

ステータスコードは以下の表の通りである。

[表11-15] ucode解決情報の削除のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	<ucode>が指定されていない。
403 Forbidden	アクセス権限がない。
404 Not Found	対象のucodeが情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

成功時、レスポンスボディは空である。

【APIの利用例】

<クエリ>

```
DELETE /api/v1/resolve/ucode_00001C00000000000001000000100125 HTTP/1.1  
Host: www.example.org  
Content-Length: xxx  
Content-Type: application/json; charset=utf-8
```

<レスポンス>

```
HTTP/1.1 204 No Content  
Connection: close
```

12. Traffic Extension（交通実証拡張）

公共交通情報流通連携基盤システム

Traffic Extension（交通実証拡張）は、公共交通に関する情報を統一した形で保持・運用するための機能である。ここでは、公共交通機関の情報の中で、静的な情報を扱う。具体的な範囲を以下に挙げる。

- ・ 駅
- ・ 鉄道路線
- ・ バス停
- ・ バス路線
- ・ 時刻表

本機能で取り扱うデータは以下の通りである。

- ・ リアルタイムなデータ
 - 走行位置データ
 - 遅延データ
 - 運休データ
 - ダイヤ変更データ
 - 臨時ダイヤデータ
 - その他緊急情報に係るデータ
 - リアルタイムな公共交通施設情報に係るデータ
- ・ 静的なデータ
 - 静的な運行情報に係るデータ
 - 路線データ
 - 駅・停留所に係るデータ
 - 公共交通関連施設データ

本規格に基づくシステムでは、情報を取得するための手段として、以下の2つの機能を提供する。

- (1) 対象（駅、バス停など）とその ucode を検索する機能
- (2) 対象（駅、バス停など）に割り当てられた ucode を解決し、それに紐付いた情報を取得する機能

一般的な使用方法としては、まず(1)の機能を使用して情報を取得したい対象のucodeを取得し、次に(2)の機能を使用してそのucodeに紐付けられた情報を取得する。

本機能のコマンドの一覧を以下の表に示す。なお、URLパスの等しい複数のコマンドは、パラメータに指定するrdf_typeの値により振り分けられる。

[表12-1] Traffic Extension（交通実証拡張）のコマンド一覧(1)

URLパス	HTTPメソッド	意味
/api/v1/places	GET	公共交通に関するエンティティを検索する

・ Geographical Data Management Command の拡張

[表12-2] Traffic Extension（交通実証拡張）のコマンド一覧(2)

URLパス	HTTPメソッド	意味
/api/v1/datapoints	GET	静的公共交通情報データを検索する
/api/v1/datapoints/<target>	GET	静的公共交通情報データを閲覧する

・ Triple Management Command の拡張

[表12-3] Traffic Extension（交通実証拡張）のコマンド一覧(3)

URLパス	HTTPメソッド	意味
/api/v1/places	GET	公共交通情報データで扱うエンティティを絞り込み検索する
/api/v1/puti/diagram/<target>	GET	時刻表データを取得する
/api/v1/datapoints	GET	走行情報・遅延情報を取得する、 在線情報を取得する、運行情報 を取得する、リアルタイムな公 共施設情報を取得する

・ PuTI (Public Transport Information) Command

——[12-1: 公共交通に関する地物情報の検索]——

<機能概要>

公共交通情報として扱うエンティティの検索を行う。ここで扱う情報はすべて地物であるため、情報流通連携基盤におけるGeographical Data Management Command に準拠した検索API を提供する。

<メソッド>

GET

<URLパス>

/api/v1/places?lat=<lat>&lon=<lon>&radius=<radius>&...

<制約条件>

なし

<リクエストパラメータ>

リクエストパラメータは、GETメソッドのURL部に格納する。複数のパラメータを指定した場合、AND検索となる。

[表12-4] 公共交通に関する地物情報の検索のリクエストパラメータ

パラメータ名	値の型	説明
lat	xsd:double	[必須] WGS84 での緯度
lon	xsd:double	[必須] WGS84 での経度
radius	xsd:double	[必須] 検索半径[m]
floor	xsd:double[]	階数、上下限値をカンマ区切り、地上(屋外)は0。
alt	xsd:double[]	高度、上下限値をカンマ区切り。
offset	xsd:integer	lat、lon で指定した地点からの距離が近い順に並べた場合にoffset番目からlimit個分のデータを要求する。
limit	xsd:integer	(前項参照)

<ステータスコード>

ステータスコードは以下の表の通りである。

[表12-5] 公共交通に関する地物情報の検索のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	パラメータ不正
404 Not Found	検索条件を満たすエンティティが情報流通連携基盤システムに登録されていない。
413 Request Entity Too Large	limit値が大きすぎる。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

レスポンスは、以下の構造データを JSONまたはXML形式で表現したものである。

[表12-6] 公共交通に関する地物情報の検索のレスポンスパラメータ

パラメータ名	型	説明
places	RDF	場所の連想配列。レスポンス形式にXMLを指定した場合、各エンティティ情報はRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各エンティティ情報はRDF/JSONで表現される。
remains	xsd:boolean	条件にマッチするが、レスポンスに含まれない場所がある場合はtrue、ない場合はfalse。

レスポンスに含まれるプロパティ (predicate) とその意味は以下の通りである。

プロパティ	値域	意味
rdf:type	rdfs:Class	公共交通データのクラス。現状ではug:Station またはug:BusStop のいずれか。
dc:title	xsd:string	地物の名称 (駅名やバス停名)
ug:region	xsd:string	鉄道・軌道 (索道は除く) 沿いに設置される駅やバス停の幾何情報 (GeoJSON形式)
w3cgeo:lat	xsd:double	駅やバス停の代表点の緯度
w3vgeo:long	xsd:double	駅やバス停の代表点の経度

<補足>

本APIは Geographical Data Management Command の拡張である。情報流通連携基盤外部仕様における Geographical Data Management Command では、返り値として地物の型 (ug:type) と地物情報 (ug:region) 以外を規定していないが、本規格では実際のアプリケーションでの使用を考慮し、その地物の一般的な名称と代表点の緯度経度を返すことを明記している。

【APIの利用例】

<クエリ>

```
GET /api/v1/places?lat=35.6260&lon=139.7236&radius=50 HTTP/1.1
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{
  "places": {
    "urn:ucode:_00001C000000000000001000000100800": {
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#type": [{
        "value": "http://uidcenter.org/ucr/vocab/ug#Station",
        "type": "uri"}],
      "http://purl.org/dc/elements/1.1/title": [{
        "value": "JR 五反田駅", "type": "literal"}],
      "http://uidcenter.org/ucr/vocab/ug#region": [{
        "value": {"type": "Point",
```

```

"coordinates": "[139. 72344 35. 62645]","",
"type": "literal"]],
"http://www. w3. org/2003/01/geo/wgs84_pos#lat": [{
"value": 35. 62645 , "type": "literal"}],
"http://www. w3. org/2003/01/geo/wgs84_pos#long": [{
"value": 139. 72344, "type": "literal"}]],
"urn:ucode:_00001C000000000000001000000100801": {
"http://www. w3. org/1999/02/22-rdf-syntax-ns#type": [{
"value": "http://uidcenter. org/ucr/vocab/ug#Station",
"type": "uri"}],
"http://purl. org/dc/elements/1. 1/title": [{
"value": "東急五反田駅", "type": "literal"}],
"http://uidcenter. org/ucr/vocab/ug#region ": [{
"value": "{ \"type\": \"Point\",
\"coordinates\": \"[139. 72344 35. 62645]\"\"",
"type": "literal"}],
"http://www. w3. org/2003/01/geo/wgs84_pos#lat": [{
"value": 35. 62645 , "type": "literal"}],
"http://www. w3. org/2003/01/geo/wgs84_pos#long": [{
"value": 139. 72344, "type": "literal"}]],
"urn:ucode:_00001C000000000000001000000100802": {
"http://www. w3. org/1999/02/22-rdf-syntax-ns#type": [{
"value": "http://uidcenter. org/ucr/vocab/ug#Station",
"type": "uri"}],
"http://purl. org/dc/elements/1. 1/title": [{
"value": "東急五反田駅", "type": "literal"}],
"http://uidcenter. org/ucr/vocab/ug#region ": [{
"value": "{ \"type\": \"Point\",
\"coordinates\": \"[139. 72344 35. 62645]\"\"",
"type": "literal"}],
"http://www. w3. org/2003/01/geo/wgs84_pos#lat": [{
"value": 35. 62645 , "type": "literal"}],
"http://www. w3. org/2003/01/geo/wgs84_pos#long": [{
"value": 139. 72344, "type": "literal"}]],
"urn:ucode:_00001C000000000000001000000100803": {
"http://www. w3. org/1999/02/22-rdf-syntax-ns#type": [{
"value": "http://uidcenter. org/ucr/vocab/ug#BusStop",
"type": "uri"}],
"http://purl. org/dc/elements/1. 1/title": [{
"value": "五反田駅5", "type": "literal"}],
"http://uidcenter. org/ucr/vocab/ug#region ": [{
"value": "{ \"type\": \"Point\",
\"coordinates\": \"[139. 72344 35. 62645]\"\"",
"type": "literal"}],
"http://www. w3. org/2003/01/geo/wgs84_pos#lat": [{
"value": 35. 62645 , "type": "literal"}],
"http://www. w3. org/2003/01/geo/wgs84_pos#long": [{
"value": 139. 72344, "type": "literal"}]],
"urn:ucode:_00001C000000000000001000000100804": {
"http://www. w3. org/1999/02/22-rdf-syntax-ns#type": [{

```

```

    "value": "http://uidcenter.org/ucr/vocab/ug# BusStop ",
    "type": "uri" ],
    "http://purl.org/dc/elements/1.1/title": [{
      "value": "五反田駅6", "type": "literal" },
    "http://uidcenter.org/ucr/vocab/ug#region ": [{
      "value": "{ \"type\": \"Point\",
        \"coordinates\": \"[139. 72344 35. 62645]\" ",
      "type": "literal" },
    "http://www.w3.org/2003/01/geo/wgs84_pos#lat": [{
      "value": 35. 62645, "type": "literal" },
    "http://www.w3.org/2003/01/geo/wgs84_pos#long": [{
      "value": 139. 72344, "type": "literal" } ] ] },
    "remains": false
  }

```

[12-2: 静的公共交通情報の検索]

<機能概要>

静的公共交通情報を検索する。

<メソッド>

GET

<URLパス>

/api/v1/datapoints?<param1>=<value1>&<param2>=<value2>...

<リクエストパラメータ>

リクエストパラメータは、GETメソッドのURL部に格納する。

[表12-7] 静的公共交通情報の検索のリクエストパラメータ

パラメータ名	説明
paramN	検索対象パラメータ名
valueN	検索対象パラメータ値

- ・ <paramN>、<valueN>の組を1 個以上指定しなければならない。
- ・ <paramN>は以下のいずれかである。
 - 静的公共交通情報の属性を示すプロパティ URI。
 - offset, limit。パラメータ値はxsd:integer である。検索結果のうちoffset番目からlimit個分を要求することを意味する。

<ステータスコード>

ステータスコードは以下の表の通りである。

[表12-8] 静的公共交通情報の検索のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	パラメータ不正
404 Not Found	検索条件を満たす静的公共交通情報が情報流通連携基盤システムに登録されていない。
413 Request Entity Too Large	limit値が大きすぎる。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

レスポンスは、以下の構造データを JSONまたはXML形式で表現したものである。

[表12-9] 静的公共交通情報の検索のレスポンスパラメータ

パラメータ名	型	説明
results	RDF	静的公共交通情報の連想配列。レスポンス形式にXMLを指定した場合、各情報はRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各情報はRDF/JSONで表現される。
remains	xsd:boolean	条件にマッチするが、レスポンスに含まれない情報がある場合はtrue、ない場合はfalse。

レスポンスに含まれるプロパティ (predicate) とその意味は以下の表の通りである。

プロパティ	値域	意味
rdf:type	rdfs:Class	静的公共交通情報のクラス。現状では表2に挙げたクラスのいずれか。
dc:title	xsd:string	静的公共交通情報の名称(駅名やバス停名)
ug:region	xsd:string	駅やバス停の幾何情報(GeoJSON形式)
pt:operatorCode	xsd:string	管轄会社の識別コード
pt:nameOfRailway	xsd:string	路線名
pt:operator	xsd:string	管轄会社名
pt:yearOfTogether	xsd:integer	静的公共交通情報の対象に対する供用が開始された年
pt:yearOfBegin	xsd:integer	静的公共交通情報の対象を設置開始した年
pt:yearOfEnd	xsd:integer	静的公共交通情報の対象を設置終了した年

<補足>

本APIは Geographical Data Management Command の拡張であり、次の表に示すクラスに属する情報を対象とする。

クラス名	alias_URI	ucodeの付与対象 [例]
駅	ug:Station	管轄会社毎に付与する [JR五反田駅、東急五反田駅、都営五反田駅]
鉄道路線	ug:Railway	路線毎に付与する [山手線、浅草線、池上線]
バス停	ug:BusStop	バス停毎に付与する [五反田駅前、戸越銀座]
バス路線	ug:BusRoute	路線毎に付与する [31六本木循環]
時刻表	pg:Diagram	時刻表情報毎に付与する [「駅名, {発or 着}, 時刻」の組]

<補足>

情報流通連携基盤外部仕様における Triple Management Command では、 返り値の属性を規定していないが、本規格では実際のアプリケーションでの使用を考慮し、返り値の属性を明記している。

【APIの利用例】

<クエリ>

```
GET /api/v1/datapoints?dc_title=山手線HTTP/1.1
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{
  "places": {
    "urn:ucode:_00001C000000000000001000000100805": {
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#type": [{
        "value": "http://uidcenter.org/ucr/vocab/ug#RailWay",
        "type": "uri"}],
      "http://purl.org/dc/elements/1.1/title": [{
        "value": "山手線", "type": "literal"}],
      "http://uidcenter.org/ucr/vocab/ug#region": [{
        "value": "{ \"type\": \"Point\",
          \"coordinates\": \"[139.72344 35.62645]\" }",
        "type": "literal"}],
      "http://opendata.org/pt#operatorCode": [{
        "value": "JRJR", "type": "literal"}],
      "http://opendata.org/pt#nameOfRailway": [{
        "value": "山手線", "type": "literal"}],
      "http://opendata.org/pt#operator": [{
        "value": "JR 東日本", "type": "literal"}],
      "http://opendata.org/pt#yearOfTogether": [{
        "value": "1900", "type": "literal"}],
      "http://opendata.org/pt#yearOfBegin": [{
        "value": "0840", "type": "literal"}]}],
    "remains": false
  }
}
```

——[12-3: 静的公共交通情報の閲覧]——

<機能概要>

静的公共交通情報を閲覧する。

<メソッド>

GET

<URLパス>

/api/v1/datapoints/<target>

<リクエストパラメータ>

リクエストパラメータは、GETメソッドのURL部に格納する。

[表12-10] 静的公共交通情報の閲覧のリクエストパラメータ

パラメータ名	型	説明
target	xsd:anyURI []	静的公共交通情報の識別子

<ステータスコード>

ステータスコードは以下の表の通りである。

[表12-11] 静的公共交通情報の閲覧のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	パラメータ不正
404 Not Found	検索条件を満たす静的公共交通情報が情報流通連携基盤システムに登録されていない。
413 Request Entity Too Large	limit値が大きすぎる。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

レスポンスは、以下の構造データを JSONまたはXML形式で表現したものである。

[表12-12] 静的公共交通情報の閲覧のレスポンスパラメータ

パラメータ名	型	説明
results	RDF	静的公共交通情報の連想配列。レスポンス形式にXMLを指定した場合、各情報はRDF/XMLで表現される。レスポンス形式にJSONを指定した場合、各情報はRDF/JSONで表現される。

レスポンスに含まれるプロパティ (predicate) とその意味は、前の表の通りである。

<補足>

本APIは Geographical Data Management Command の拡張である。情報流通連携基盤外部仕様における Triple Management Command では、 返り値の属性を規定していないが、本規格では実際のアプリケーションでの使用を考慮し、返り値の属性を明記している。

【APIの利用例】

<クエリ>

```
GET /api/v1/datapoints/ucode_00001C000000000000001000000100805
HTTP/1.1
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{
  "places": {
    "urn:ucode:_00001C000000000000001000000100805": {
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#type": [{
        "value": "http://uidcenter.org/ucr/vocab/ug#RailWay",
        "type": "uri"}],
      "http://purl.org/dc/elements/1.1/title": [{
        "value": "山手線", "type": "literal"}],
      "http://uidcenter.org/ucr/vocab/ug#region": [{
        "value": "{ \"type\": \"Point\",
          \"coordinates\": \"[139.72344 35.62645]\" }",
        "type": "literal"}],
      "http://opendata.org/pt#operatorCode": [{
        "value": "JRJR", "type": "literal"}],
      "http://opendata.org/pt#nameOfRailway": [{
        "value": "山手線", "type": "literal"}],
      "http://opendata.org/pt#operator": [{
        "value": "JR 東日本", "type": "literal"}],
      "http://opendata.org/pt#yearOfTogether": [{
        "value": "1900", "type": "literal"}],
      "http://opendata.org/pt#yearOfBegin": [{
        "value": "0840", "type": "literal"}]}],
  }
```

```

    "remains":false
}

```

--- [12-4: 公共交通に関するエンティティの絞り込み検索] ---

<機能概要>

公共交通情報として扱うエンティティの検索を行う。

<メソッド>

GET

<URLパス>

/api/v1/places?lat=<lat>&lon=<lon>&radius=<radius>&...

<制約条件>

なし

<リクエストパラメータ>

リクエストパラメータは、GETメソッドのURL部に格納する。複数のパラメータを指定した場合、AND検索となる。

[表12-13] 公共交通に関するエンティティの絞り込み検索のリクエストパラメータ

パラメータ名	値の型	説明
lat	xsd:double	[必須] WGS84 での緯度
lon	xsd:double	[必須] WGS84 での経度
radius	xsd:double	[必須] 検索半径[m]
floor	xsd:double[]	階数、上下限値をカンマ区切り、地上(屋外)は0。
alt	xsd:double[]	高度、上下限値をカンマ区切り。
offset	xsd:integer	lat、lon で指定した地点からの距離が近い順に並べた場合にoffset番目からlimit個分のデータを要求する。
limit	xsd:integer	(前項参照)
hops	xsd:integer	URL をたどるホップカウント数。
rdf_type	xsd:anyURI	対象となるエンティティのタイプ(rdf:type)

rdf_type パラメータ値として指定できるクラスを以下に示す。

クラス	説明
puti:Train	電車
ug:Station	駅
ug:RailWay	線路
puti:Bus	バス
ug:BusStop	バス停
ug:BusRoute	バス路線
puti:Diagram	時刻表
puti:Sensor	センサ

<ステータスコード>

ステータスコードは以下の表の通りである。

[表12-14] 公共交通に関するエンティティの絞り込み検索のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	パラメータ不正
404 Not Found	検索条件を満たすエンティティが情報流通連携基盤システムに登録されていない。
413 Request Entity Too Large	limit値が大きすぎる。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

レスポンスは、以下の構造データを JSONまたはXML形式で表現したものである。

[表12-15] 公共交通に関するエンティティの絞り込み検索のレスポンスパラメータ

パラメータ名	型	説明
results	RDF	エンティティ情報の連想配列。RDF/JSONで表現される。
remains	xsd:boolean	条件にマッチするが、レスポンスに含まれない場所がある場合はtrue、ない場合はfalse。

<補足>

本APIはPuTI (Public Transport Information) Commandである。情報流通連携基盤外部仕様における Geographical Data Management Command では、クエリパラメータとして指定していない、hops と rdf_type をパラメータに追加している。

【APIの利用例】

<クエリ>

```
GET
/api/v1/puti/entities?lat=35.6260&lon=139.7236&radius=50&floor=-1
0,10&rdf_type=ug_railway HTTP/1.1
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{
  "results": {
    "urn:ucode:_00001C000000000000001000000100808": {
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#type": [{
        "value": "http://uidcenter.org/ucr/vocab/ug#Railway",
```

```

    "type": "uri"]],
    "http://purl.org/dc/elements/1.1/title": [{
      "value": "山手線", "type": "literal"}],
    "http://uidcenter.org/ucr/vocab/ug#region": [{
      "value": "{ \"type\": \"Point\",
        \"coordinates\": \"[139. 72344 35. 62645]\" }",
      "type": "literal"}],
    "http://opendata.org/pt#operatorCode": [{
      "value": "JRJR", "type": "literal"}],
    "http://opendata.org/pt#nameOfRailway": [{
      "value": "山手線", "type": "literal"}],
    "http://opendata.org/pt#operator": [{
      "value": "JR 東日本", "type": "literal"}],
    "http://opendata.org/pt#yearOfTogether": [{
      "value": "1900", "type": "literal"}],
    "http://opendata.org/pt#yearOfBegin": [{
      "value": "1840", "type": "literal"}]}],
    "urn:ucode: 00001C000000000000001000000100809": {
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#type": [{
        "value": "http://uidcenter.org/ucr/vocab/ug#Railway",
        "type": "uri"}],
      "http://purl.org/dc/elements/1.1/title": [{
        "value": "都営浅草線", "type": "literal"}],
      "http://uidcenter.org/ucr/vocab/ug#region": [{
        "value": "{ \"type\": \"Point\",
          \"coordinates\": \"[139. 72344 35. 62645]\" }",
        "type": "literal"}],
      "http://opendata.org/pt#operatorCode": [{
        "value": "T0EI", "type": "literal"}],
      "http://opendata.org/pt#nameOfRailway": [{
        "value": "都営浅草線", "type": "literal"}],
      "http://opendata.org/pt#operator": [{
        "value": "東京都交通局", "type": "literal"}],
      "http://opendata.org/pt#yearOfTogether": [{
        "value": "1900", "type": "literal"}],
      "http://opendata.org/pt#yearOfBegin": [{
        "value": "1840", "type": "literal"}]}]}],
    "remains": false
  }
}

```

——[12-5: 時刻表の検索]——

＜機能概要＞

時刻表のデータを取得する。

＜メソッド＞

GET

＜URLパス＞

/api/v1/puti/diagram/<ucode>

＜制約条件＞

なし

＜リクエストパラメータ＞

リクエストパラメータは、GETメソッドのURL部に格納する。

[表12-16] 時刻表の検索のリクエストパラメータ

パラメータ名	値の型	説明
ucode	xsd:anyURI	時刻表の識別子

＜ステータスコード＞

ステータスコードは以下の表の通りである。

[表12-17] 時刻表の検索のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	パラメータ不正
404 Not Found	検索条件を満たす時刻表が情報流通連携基盤システムに登録されていない。
413 Request Entity limit Too Large	limit値が大きすぎる。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

＜レスポンス＞

レスポンスは、以下の構造データからなる配列を JSONまたはXML形式で表現したものである。

[表12-18] 時刻表の検索のレスポンスパラメータ

パラメータ名	型	説明
time	Time	時刻
destination	String	行き先(オプション)
isArrival	Bool	着時刻である場合はtrue、省略時はfalse
isNonStepBus	Bool	ノンステップバスである場合はtrue、省略時はfalse

<補足>

本APIはPuTI(Public Transport Information) Commandである。

【APIの利用例】

urn:ucode:_ 00001C000000000000001000000100810 が、五反田駅・山手線内回り平日の時刻表に振られたucode であるとする。

<クエリ>

```
GET /api/v1/puti/diagram/ucode_00001C000000000000001000000100810
HTTP/1.1
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
[
  ...
  {
    "time": "13:32", "isArrival": true,
    "time": "13:35", "destination": "池袋",
    "time": "13:39",
    "time": "13:43",
    "time": "13:52",
    "time": "13:58", "destination": "池袋",
  },
  ...
]
```

——[12-6: 走行情報・遅延情報の取得]——

<機能概要>

列車・バスの走行情報・遅延情報を取得する。

<メソッド>

GET

<URLパス>

/api/v1/datapoints?rdf_type=<rdf_type>&...

<制約条件>

なし

<リクエストパラメータ>

リクエストパラメータは、GETメソッドのURL部に格納する。

[表12-19] 走行情報・遅延情報の取得のリクエストパラメータ

パラメータ名	値の型	説明
lat	xsd:double	[必須] WGS84 での緯度
lon	xsd:double	[必須] WGS84 での経度
radius	xsd:double	[必須] 検索半径[m]
floor	xsd:double[]	階数、上下限値をカンマ区切り、地上(屋外)は0。
alt	xsd:double[]	高度、上下限値をカンマ区切り。
offset	xsd:integer	lat、lon で指定した地点からの距離が近い順に並べた場合にoffset番目からlimit個分のデータを要求する。
limit	xsd:integer	(前項参照)
rdf_type	xsd:anyURI	[必須] ug_Train またはug_Bus に固定
stream	xsd:integer	0 を指定した場合、ストリーム配信を開始

<ステータスコード>

ステータスコードは以下の表の通りである。

[表12-20] 走行情報・遅延情報の取得のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	パラメータ不正
404 Not Found	検索条件を満たす情報が情報流通連携基盤システムに登録されていない。
413 Request Entity Too Large	limit値が大きすぎる。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

rdf_type にputi:Train を指定した場合のレスポンスは、以下のプロパティをもつRDF/JSONデータである。

[表12-21] 走行情報・遅延情報の取得のレスポンスパラメータ

プロパティ名	値域	意味
rdf:type	—	puti:Train に固定
dc:title	xsd:string	列車番号
puti:nameOfRailway	xsd:string	鉄道路線の名称
ug:railway	ug:Railway	鉄道路線のucode
puti:nameOfDestination	xsd:string	行き先名
puti:destination	ug:Station	行き先のucode
puti:diagrams	xsd:string	時刻表データが格納されたURL
w3cgeo:lat	xsd:double	代表点の緯度。10進数表記
w3cgeo:long	xsd:double	代表点の経度。10進数表記

rdf_type にputi:Bus を指定した場合のレスポンスは、以下のプロパティをもつRDF/JSONデータである。

プロパティ名	値域	意味
rdf:type	—	puti:Bus に固定
dc:title	xsd:string	列車番号
puti:nameOfBusRoutes	xsd:string	バス路線の名称
ug:busRoute	ug:BusRoute	バス路線ucode
puti:nameOfDestination	xsd:string	行き先名
puti:destination	ug:BusStop	行き先のucode
puti:diagrams	xsd:string	時刻表データが格納されたURL
w3cgeo:lat	xsd:double	代表点の緯度。10進数表記
w3cgeo:long	xsd:double	代表点の経度。10進数表記

<解説>

走行位置データ、遅延データは運行情報を持ち得る物体(Entity)に振られたucode に対して情報を付与した形で提供される。 具体的には列車とバスである。ここで扱うEntityに振られるクラスを以下に挙げる。

Entity	属するクラス	ucode の付与対象 [例]
列車	puti:Train	列車番号毎に付与 [3510M, 2453M]
バス車両	puti:Bus	運行番号毎に付与 [M234, M4343]

ある列車編成の遅延情報はその列車編成に振られたucode のrelation として表現される。本コマンドは、そのrelation のリストを取り出す。

<補足>

本APIはPuTI(Public Transport Information) Commandである。

【APIの利用例】

五反田駅を中心に、半径 50m 以内に存在する列車を検索する。

<クエリ>

GET

/api/v1/datapoints?rdf_type=ug_Train&lat=35.6260&lon=139.7236&radius=50 HTTP/1.1

Host: www.example.org

<レスポンス>

HTTP/1.1 200 OK

Content-Length: xxx

Connection: close

Content-Type: application/json; charset=utf-8

```
{
  "urn:ucode:_00001C000000000000001000000100811": {
    "rdf:type": [{
      "value": "http://opendata.ubin.jp/puti#Train",
      "type": "uri"
    }],
    "http://purl.org/dc/elements/1.1/title": [{
      "value": "M5014", "type": "literal"
    }],
    "http://purl.org/dc/elements/1.1/time": [{
      "value": "2013-01-13T20:06:00+9", "type": "literal"
    }],
    "http://opendata.ubin.jp/puti#NameOfRailway": [{
      "value": "山手線", "type": "literal"
    }],
    "http://uidcenter.org/ucr/vocab/ug#railway": [{
      "value": "urn:ucode:_00001C000000000000001000000100812",
      "type": "uri"
    }],
    "http://opendata.ubin.jp/puti#nameOfDestination": [{
      "value": "内回り", "type": "literal"
    }],
    "http://opendata.ubin.jp/puti#destination": [{
      "value": "urn:ucode:_00001C000000000000001000000100813",
      "type": "uri"
    }],
    "http://opendata.ubin.jp/puti#diagrams": [
      {
        "value": "urn:ucode:_00001C000000000000001000000100814",
        "type": "uri"
      },
      {
        "value": "urn:ucode:_00001C000000000000001000000100815",
        "type": "uri"
      }
    ],
    "http://www.w3.org/2003/01/geo/wgs84_pos#lat": [{
      "value": "35.6253", "type": "literal"
    }],
    "http://www.w3.org/2003/01/geo/wgs84_pos#long": [{
      "value": "139.7243", "type": "literal"
    }]
  }
}
```

——[12-7: 在線情報の取得]——

＜機能概要＞

列車・バスの在線情報を取得する。

＜メソッド＞

GET

＜URLパス＞

/api/v1/datapoints?rdf_type=<rdf_type>&...

＜制約条件＞

なし

＜リクエストパラメータ＞

リクエストパラメータは、GETメソッドのURL部に格納する。

[表12-22] 在線情報の取得のリクエストパラメータ

パラメータ名	値の型	説明
rdf_type	xsd:anyURI	[必須] 取得対象の駅・停留所・路線のクラス。 ug_Station、ug_Railway、ug_Busstop、ug_Busroute に固定。
dc_title	xsd:string	[必須] 情報を取得したい駅・停留所・路線の名称。
stream	xsd:integer	0 を指定した場合、ストリーム配信を開始

＜ステータスコード＞

ステータスコードは以下の表の通りである。

[表12-23] 在線情報の取得のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	パラメータ不正
404 Not Found	検索条件を満たす情報が情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

＜レスポンス＞

rdf_type にputi:Train を指定した場合のレスポンスは、以下のプロパティをもつRDF/JSONデータである。

[表12-24] 在線情報の取得のレスポンスパラメータ

プロパティ名	値域	意味
dc:time	xsd:datetime	その状態である時刻
puti:Trains	puti:Train[]	列車のリスト
puti:Buses	puti:Bus[]	バスのリスト

<解説>

走行位置データ、遅延データは在線情報を持ち得る物体(Entity)に振られたucode に対して情報を付与した形で提供される。具体的には駅、鉄道路線、バス停、バス路線である。ここで扱うEntityに振られるクラスを以下に挙げる。

Entity	属するクラス	ucode の付与対象 [例]
駅	puti:Train	駅と管轄会社毎に付与 [JR五反田駅、東急五反田駅、都営五反田駅]
路線	puti:Bus	路線毎に付与 [山手線、都営浅草線、東急池上線]
バス停	ug:BusStop	バス停毎に付与 [五反田駅前、戸越銀座]
バス路線	—	— [31 六本木循環]

例えば、五反田駅の情報を取得すると、五反田駅に停車していると観測された列車一覧が取得される。また山手線の路線を指定すると、その区間を走行中の列車一覧が取得される。

<補足>

本APIはPuTI (Public Transport Information) Commandである。

【APIの利用例】

五反田駅を中心に、半径 50m 以内に存在する列車を検索する。

<クエリ>

```
GET /api/v1/datapoints?rdf_type=ug_Station&dc_title=JR五反田駅
HTTP/1.1
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{
  "urn:ucode:_00001C000000000000001000000100820": {
    "rdf:type": [{
      "value": "http://uidcenter.org/ucr/vocab/uc#Station",
      "type": "uri"
    }],
    "http://purl.org/dc/elements/1.1/title": [{
      "value": "五反田駅", "type": "literal"
    }],
    "http://purl.org/dc/elements/1.1/time": [{
      "value": "2013-01-13T20:06:00+9", "type": "literal"
    }],
    "http://opendata.ubin.jp/puti#Trains": [{
```

```

        "value": "urn:ucode:_00001C000000000000001000000100821",
        "type": "uri"}, {
        "value": "urn:ucode:_00001C000000000000001000000100822",
        "type": "uri"}]]}},
    "urn:ucode:_00001C000000000000001000000100823": {
        "rdf:type": [{
            "value": "http://uidcenter.org/ucr/vocab/uc#Station",
            "type": "uri"}],
        "http://purl.org/dc/elements/1.1/title": [{
            "value": "五反田駅", "type": "literal"}],
        "http://purl.org/dc/elements/1.1/time": [{
            "value": "2013-01-13T20:06:00+9", "type": "literal"}],
        "http://opendata.ubin.jp/puti#Trains": [{
            "value": "urn:ucode:_00001C000000000000001000000100824",
            "type": "uri"}, {
            "value": "urn:ucode:_00001C000000000000001000000100825",
            "type": "uri"}]}
    }
}

```

[12-8: 運行情報の取得]

<機能概要>

列車・バスの運行情報を取得する。

<メソッド>

GET

<URLパス>

/api/v1/datapoints?rdf_type=puti_TrainInfo&puti_TrainInfoLine=<RAILWAY_NAME>

<制約条件>

なし

<リクエストパラメータ>

リクエストパラメータは、GETメソッドのURL部に格納する。

[表12-25] 運行情報の取得のリクエストパラメータ

パラメータ名	値の型	説明
rdf_type	xsd:anyURI	[必須] 運行情報のクラス。 puti_TrainInfo に固定。
puti_TrainInfoLine stream	xsd:string xsd:integer	[必須] 対象路線名。 0を指定した場合、ストリーム配信を開始

<ステータスコード>

ステータスコードは以下の表の通りである。

[表12-26] 運行情報の取得のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	パラメータ不正
404 Not Found	検索条件を満たす情報が情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

以下のプロパティをもつ RDF/JSONデータである。

[表12-27] 運行情報の取得のレスポンスパラメータ

プロパティ名	値域	意味 [例]
rdf_type	—	puti:TrainInfo に固定
dc:time	xsd:datetime	データ生成時刻 [2013-01-13T15:10:00+9]
puti:ofRailway	xsd:anyURI	発生路線ucode [urn:ucode:_00001C0000 0000000001000000012345]
puti:trainInfoTime	xsd:string	発生時刻 [15時03分頃]
puti:trainInfoLine	xsd:string	発生路線名 [中央線快速電車]
puti:trainInfoFrom	xsd:string	発生場所起点 [中野]
puti:trainInfoTo	xsd:string	発生場所終点 [高円寺]
puti:trainInfoRange	xsd:string	発生区間 [中野-高円寺]
puti:trainInfoDirection	xsd:string	行き先
puti:trainInfoCause	xsd:string	発生理由 [信号トラブル]
puti:trainInfoStatus	xsd:string	現状 [運転見合わせ]
puti:trainInfoText	xsd:string	運行情報テキスト [中央線快速電車は15 時03分頃中野～高円寺駅間での信号トラ ブルの影響で上下線で運転を見合わせて います]

<解説>

遅延データ・運休データ・ダイヤ変更データ・臨時ダイヤデータ・その他緊急情報に係るデータは、運行情報データを構成する要素とする。運行情報データは配信元毎にucodeを振り、取得順にスタックして動的RDFとして表現する。現状、運行情報配信を受けるのは鉄道に限られる。

<補足>

本APIはPuTI(Public Transport Information) Commandである。

【APIの利用例】

中央線快速電車に関する運行情報を取得する。

<クエリ>

GET /api/v1/datapoints?rdf_type=ug_TrainInfo?puti_TrainInfoLine=中央線快速電車HTTP/1.1
Host: www.example.org

<レスポンス>

HTTP/1.1 200 OK

Content-Length: xxx

Connection: close

Content-Type: application/json; charset=utf-8

```
{
  "urn:ucode:_00001C0000000000000001000000100830": {
    "rdf:type": [{
      "value": "http://opendata.ubin.jp/puti#TrainInfo",
      "type": "uri"
    }],
    "http://purl.org/dc/elements/1.1/time": [{
      "value": "2013-01-13T20:06:00+9",
      "type": "literal"
    }],
    "http://opendata.ubin.jp/puti#:OfRailway": [{
      "value": "urn:ucode:_00001C0000000000000001000000100831",
      "type": "uri"
    }],
    "http://opendata.ubin.jp/puti#:TrainInfoTime": [{
      "value": "15時03分頃",
      "type": "literal"
    }],
    "http://opendata.ubin.jp/puti#:TrainInfoLine": [{
      "value": "中央線快速電車",
      "type": "literal"
    }],
    "http://opendata.ubin.jp/puti#:TrainInfoFrom": [{
      "value": "新宿",
      "type": "literal"
    }],
    "http://opendata.ubin.jp/puti#:TrainInfoTo": [{
      "value": "荻窪",
      "type": "literal"
    }],
    "http://opendata.ubin.jp/puti#:TrainInfoRange": [{
      "value": "新宿-荻窪",
      "type": "literal"
    }],
    "http://opendata.ubin.jp/puti#:TrainInfoDirection": [{
      "value": "上下線",
      "type": "literal"
    }],
    "http://opendata.ubin.jp/puti#:TrainInfoCause": [{
      "value": "信号トラブル",
      "type": "literal"
    }],
    "http://opendata.ubin.jp/puti#:TrainInfoStatus": [{
      "value": "運転見合わせ",
      "type": "literal"
    }],
    "http://opendata.ubin.jp/puti#:TrainInfoText": [{
      "value": "中央線快速電車は、15時03分頃 中野～高円寺駅間での信号トラブルの影響で、上下線で運転を見合わせています。",
      "type": "literal"
    }
  ]
}
```

}

[12-9: リアルタイムな公共交通施設情報の取得]

<機能概要>

リアルタイムに変化する公共交通施設情報を取得する。

<メソッド>

GET

<URLパス>

/api/v1/places?rdf_type=puti_Sensor&lat=<lat>&lon=<lon>&radius=<radius>&...

<制約条件>

なし

<リクエストパラメータ>

リクエストパラメータは、GETメソッドのURL部に格納する。

[表12-28] リアルタイムな公共交通施設情報の取得のリクエストパラメータ

パラメータ名	値の型	説明
lat	xsd:double	[必須] WGS84での緯度
lon	xsd:double	[必須] WGS84での経度
radius	xsd:double	[必須] 検索半径[m]
floor	xsd:double[]	階数、上下限値をカンマ区切り、地上(屋外)は0。
alt	xsd:double[]	高度、上下限値をカンマ区切り。
offset	xsd:integer	lat、lon で指定した地点からの距離が近い順に並べた場合にoffset番目からlimit個分のデータを要求する。
limit	xsd:integer	(前項参照)
rdf_type	xsd:anyURI	[必須] puti_Sensor 固定

<ステータスコード>

ステータスコードは以下の表の通りである。

[表12-29] リアルタイムな公共交通施設情報の取得のステータスコード

ステータスコード	意味
200 OK	正常終了。
400 Bad Request	パラメータ不正
404 Not Found	検索条件を満たす情報が情報流通連携基盤システムに登録されていない。
500 Internal Error	情報流通連携基盤システム内でエラーが発生した。

<レスポンス>

以下のプロパティをもつ RDF/JSONデータである。

[表12-30] リアルタイムな公共交通施設情報の取得のレスポンスパラメータ

プロパティ名	値域	意味
rdf_type	xsd:anyURI	puti:Sensor 固定
dc:time	xsd:datetime	センサからのデータ取得時刻
uc:temperature	xsd:float	気温(セルシウス度)
uc:humidity	xsd:float	湿度(パーセント)
puti:pollen	xsd:integer	花粉飛散量(個/m3)
w3cgeo:lat	xsd:double	代表点の緯度。10進数表記
w3cgeo:long	xsd:double	代表点の経度。10進数表記

<補足>

本APIはPuTI(Public Transport Information) Commandである。

【APIの利用例】

<クエリ>

```
GET /api/v1/places?points?rdf_type=puti_Sensor&lat=35.6260&lon=139.7236&radius=50 HTTP/1.1
Host: www.example.org
```

<レスポンス>

```
HTTP/1.1 200 OK
Content-Length: xxx
Connection: close
Content-Type: application/json; charset=utf-8
{
  "urn:ucode:_00001C000000000000001000000100840": {
    "rdf:type": [{
      "value": "http://opendata.ubin.jp/puti#Sensor",
      "type": "uri"
    }],
    "http://purl.org/dc/elements/1.1/time": [{
      "value": "2013-01-13T20:06:00+9",
      "type": "literal"
    }],
    "http://uidcenter.org/ucr/vocab/uc#temperature": [{
      "value": "23.9",
      "type": "literal"
    }],
    "http://uidcenter.org/ucr/vocab/uc#humidity": [{
```

```

    "value": "30", "type": "literal"]]]},
  "urn:ucode:_00001C000000000000001000000100841": {
    "rdf:type": [{
      "value": "http://opendata.ubin.jp/puti#Sensor",
      "type": "uri"}],
    "http://purl.org/dc/elements/1.1/time": [{
      "value": "2013-01-13T20:06:00+9", "type": "literal"}],
    "http://opendata.ubin.jp/puti#frequency": [{
      "value": "60", "type": "literal"}],
    "http://opendata.ubin.jp/puti#pollen": [{
      "value": "100", "type": "literal"}]]}
  }
}

```