

公共交通情報流通連携基盤システム

インストールマニュアル

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. インストール方法

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とインストール方法は以下の通りである。

【サーバソフトウェアのシステム構成】

- o Ubuntu Server 12.04 LTS 64bit
- o nginx PPA
- o JRuby 1.7.2 以上
- o Ruby on Rails 3.2.2 以上

【インストール方法】

1. JRuby / Ruby on Rails の動作環境を構築する
2. 基盤システムのパッケージを展開する。ここではホームディレクトリに展開し、~/opendata が作成されたものとする。
3. ~/opendata/java/sesame.jar を Java の外部jar ファイルディレクトリ (Ubuntuだと/usr/java/packages/lib/ext)に設置する。
4. 同様に、以下のjarファイルも外部jar ファイルディレクトリに設置する。
 - openrdf-sesame-2.6.10-onejar.jar
from <http://sourceforge.net/projects/sesame/files/Sesame%202/2.6.10/>
 - commons-httpclient-3.1.jar
from <http://archive.apache.org/dist/httpcomponents/commons-httpclient/binary/>
 - log4j-1.2.14.jar
from <http://www.java2s.com/Code/Jar/l/log4j.htm>
 - slf4j-api-1.7.2.jar
 - slf4j-nop-1.7.2.jar
from <http://www.slf4j.org/download.html>

- commons-logging-1.1.1.jar
from http://commons.apache.org/logging/download_logging.cgi
- commons-codec-1.7.jar
from http://commons.apache.org/codec/download_codec.cgi

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>は英数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、接続を継続し、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<>で括る。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。