
Repository-RI Documentation

Release stable

September 25, 2015

1	Index	3
1.1	Installation and Administration Guide	3
1.2	User and Programmer Guide	11

The Repository GE is a core enabler of the FIWARE Business Framework (together with the Store GE and the Marketplace GE). The Repository GE and its reference implementation provide a consistent uniform API to access USDL service descriptions and associated media files for applications of the business framework. A service provider can use the Repository GE to publish the description of various aspects of the service according to a unified description language.

Moreover, the Repository provides a uniform API to perform smart searches over the published descriptions. This API is oriented to service descriptions and media files serialized in RDF formats (e.g a Linked-USDL document) and allows the usage of SPARQL as query language.

This project is part of [FIWARE](#).

Installation and Administration Guide The guide for Repository-RI maintainers that explains how to install it.

User and Programmer Guide The guide for user and programmers that explains how to browse resources and collections using a web browser, and how to integrate them using the existing APIs.

1.1 Installation and Administration Guide

1.1.1 Introduction

This Installation and Administration Guide covers Repository-RI version 4.4.3. Any feedback on this document is highly welcomed, including bugs, typos or things you think should be included but aren't. Please send it to the "Contact Person" email that appears in the [Catalogue page for this GEI](#).

1.1.2 System Requirements

Hardware Requirements

The following table contains the minimum resource requirements for running the Repository:

- CPU: 1-2 cores with at least 2.4 GHZ
- Physical RAM: 1G-2GB
- Disk Space: 25GB The actual disk space depends on the amount of data being stored within the Repositories NoSQL database System.

Operating System Support

The Repository has been tested in the following Operating Systems:

- Ubuntu 12.04, 14.04
- CentOS 6.3, 6.5, 7.0
- Debian 7

Software Requirements

In order to have the Repository running, the following software is needed. However, these dependencies are not meant to be installed manually in this step, as they will be installed throughout the documentation:

- MongoDB 2.x - mandatory
- Java 1.8.x - mandatory
- Virtuoso 7.x - mandatory
- Application Server, Apache Tomcat 8.x - mandatory
- Repository Software - mandatory
- Mongo Shell - optional (JavaScript shell that allows you to execute commands on the internal data store of the Repository from the command line)

To install the required version of Virtuoso, it is possible to download a compiled version for the supported Operative Systems or it is possible to compile it from the source code. In this way, the installation of Virtuoso has some extra requirements, that will be also installed throughout this document.

- autoconf
- automake
- libtool
- flex
- bison
- gperf
- gawk
- m4
- make
- openssl
- openssl-devel

1.1.3 Software Installation

Getting the Repository software

The packaged version of the Repository software can be downloaded from:

- [The FIWARE Files page](#)
- [The FIWARE catalogue](#).

This package contains the war file of the Repository as well as the intallation scripts used in this document.

Alternatively, it is possible to install the Repository from the sources published in GitHub. To clone the repository, the git package is needed:

```
# Ubuntu/Debian
$ apt-get install git

# CentOS
$ yum -y install git
```


To download the source code using git, execute the following command:

```
$ git clone https://github.com/conwetlab/Repository-RI.git
```

Installing the Repository using scripts

In order to facilitate the installation of the Repository, the script *install.sh* has been provided. This script installs all needed dependencies, configures the repository and deploys it.

Note that the installation script installs dependencies such as Java 1.8 or Tomcat. If you are installing the Repository in a system that is already in use, you may want to have more control over what dependencies are installed. In this case have a look at section *Manually Installing the Repository*.

To use the installation script execute the following command:

```
$ ./install.sh
```

The installation script also optionally resolves the extra dependencies that are needed for the installation of Virtuoso.

Some packages are needed for installing Virtuoso: autoconf, automake, libtool, flex, bison, gperf, gawk, m4, make, openssl, openssl-devel Do you want to install them? Y/N

Finally, the installation script allows to configure the OAuth2 user authentication.

```
Do you want to activate OAuth2 authentication in the Repository? Y/N
Y

The default OAuth2 endpoint is http://account.lab.fiware.org
Do you want to provide a different idm endpoint? Y/N
n

What is your FIWARE Client id?
[client id]

What is your FIWARE Client Secret?
[client secret]

What is your Callback URL?
http://[host]:[port]/FiwareRepository/v2/callback
```

Note: The existing scripts try to install tomcat 8.0.26, in case this version would not be available it will be needed to update the version in the script *installTomcat8.sh*

Manually installing the Repository

Debian

All the mandatory dependencies can be easily installed on a debian based Linux distribution using different scripts:

```
$ export INSPWD=$PWD
$ export REPO_OS=debian
$ ./scripts/installTools.sh
$ ./scripts/installJava8.sh
$ ./scripts/installTomcat8.sh
$ ./scripts/installMongoDB.sh
$ ./scripts/installVirtuoso7.sh
```

To install Virtuoso from the source code, it is also provided an script:

```
$ export REPO_OS=""
$ ./scripts/installVirtuoso7.sh
```

The variable `INSPWD` contains the path where the repository (Virtuoso, and Tomcat) has been installed.

Ubuntu

All the mandatory dependencies can be easily installed on a debian based Linux distribution using diferent scripts, and replacing “XX.XX” by Ubuntu version.

```
$ export INSPWD=$PWD
$ export REPO_OS=ubuntuXX.XX
$ ./scripts/installTools.sh
$ ./scripts/installJava8.sh
$ ./scripts/installTomcat8.sh
$ ./scripts/installMongoDB.sh
$ ./scripts/installVirtuoso7.sh
```

The variable `INSPWD` contains the path where the repository (Virtuoso, and Tomcat) has been installed.

CentOS/RedHat

Similarly, the different dependencies can be installed in CentOS/RedHat, and replacing “X” by Centos version.

```
$ export INSPWD=$PWD
$ export REPO_OS=centosX
$ ./scripts/installTools.sh
$ ./scripts/installJava8.sh
$ ./scripts/installTomcat8.sh
$ ./scripts/installMongoDB.sh
$ ./scripts/installVirtuoso7.sh
```

To install Virtuoso from the source code, it is also provided an script:

```
$ export REPO_OS=""
$ ./scripts/installVirtuoso7.sh
```

The variable `INSPWD` contains the path where the repository (Virtuoso, and Tomcat) has been installed.

1.1.4 Configuration

This configuration section assumes that the enviroment variable `INSPWD` exists, this variable is created during the installation process. If it does not exists execute the following command from the directory where the repository have been installed:

```
$ export INSPWD=$PWD
```

Please note that if you have used the script *install.sh* you can skip *Virtuoso 7 Configuration* and *Tomcat 8 Configuration* sections, since the specified actions are performed by the script.

Virtuoso 7 Configuration

The first step is to create and configure the Virtuoso database to store RDF content. You may need to have root rights to do that.

```
$ cd $INSPWD/virtuoso7/var/lib/virtuoso/db/
$ $INSPWD/virtuoso7/bin/virtuoso-t -f &
$ cd $INSPWD
```

This allows you to start the Virtuoso database. To make advanced configuration you can edit the file `$INSPWD/virtuoso7/var/lib/virtuoso/db/virtuoso.ini` by your own.

MongoDB Configuration

By default the Database saves its data in `/var/lib/mongodb`. Since all the Resources you upload to the Repository are stored there, the size of this folder can grow rapidly. If you want to relocate that folder, you have to edit `/etc/mongodb.conf`

```
# mongodb.conf

# Where to store the data.
dbpath=/var/lib/mongodb
```

Tomcat 8 Configuration

To continue, the next step is to start and to configure Tomcat 8. You may need to have root rights to do that.

```
$ cd $INSPWD/apache-tomcat/bin/
$ ./shutdown.sh
$ ./startup.sh
$ cd
```

To start Apache Tomcat 8 is necessary to have some variables well configured like `CATALINA_HOME`, `JAVA_HOME`. Maybe you will need configure them if you make a manual installation.

It is possible to use the Apache Tomcat Application server as is, that is, without any further configuration. However, it is recommended to allow incoming connections to the Repository only through HTTPS. This can be achieved by using a front-end HTTPS server that will proxy all requests to Repository, or by configuring the Application Server in order to accept only HTTPS/SSL connection, please refer to <http://tomcat.apache.org/tomcat-8.0-doc/ssl-howto.html> for more information.

Repository Configuration

If you have installed the Repository manually, you have to deploy the Repository software to your Application Server. For that you have to copy the Repository WAR package into the “webapp” folder of Apache Tomcat. To install it on other Java Application Servers (e.g. JBoss), please refer to the specific application server guidelines.

Also, you have to create a properties file located at `/etc/default/Repository-RI.properties` with the configuration of the repository. To create the properties file with basic configuration it is possible to use the script `repositorySettings.sh`.

The repository can use OAuth2 authentication with FIWARE Lab accounts. If you have used the automatic installation script you have been already asked to choose whether you want to use this authentication mechanism and to provide OAuth2 credentials in that case.

Before enabling OAuth2 authentication in the Repository, it is needed to have registered it on the corresponding idM (KeyRock) instance.

It is needed to provide: * A name for the application * A description * The URL of the Repository * The callback URL of the Repository: [http://\[host\]:\[port\]/FiwareRepository/v2/callback?client_name=FIWAREClient](http://[host]:[port]/FiwareRepository/v2/callback?client_name=FIWAREClient)

The OAuth2 authentication can be enabled and disabled modifying the file `web.xml` located at `WEB-INF/web.xml`.

To enable OAuth2 include `securityOAuth2.xml`

```
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>
    /WEB-INF/securityOAuth2.xml
  </param-value>
</context-param>
```

To disable OAuth2 include `noSecurity.xml`

```
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>
    /WEB-INF/noSecurity.xml
  </param-value>
</context-param>
```

You can modify OAuth2 credentials in the `Repository-RI.properties` file located at `/etc/default/Repository-RI.properties`

```
oauth2.server=https://account.lab.fiware.org
oauth2.key=[Client id]
oauth2.secret=[Client secret]
oauth2.callbackURL=http://[host]/FiwareRepository/v2/callback
```

Note: If you have decided to use OAuth2 authentication you will need to modify `oauth2.callbackURL` property to include the host where the Repository is going to run.

Finally, you can configure the MongoDB and Virtuoso instances the Repository is going to use in `Repository-RI.properties`, which contains the following values by default.

```
#MongoDb Database
mongodb.host=127.0.0.1
mongodb.db=test
mongodb.port=27017

#Virtuoso Database
virtuoso.host=jdbc:virtuoso://localhost:
virtuoso.port=1111
virtuoso.user=dba
virtuoso.password=dba
```

1.1.5 Sanity check procedures

The Sanity Check Procedures are those activities that a System Administrator has to perform to verify that an installation is ready to be tested. Therefore there is a preliminary set of tests to ensure that obvious or basic malfunctioning is fixed before proceeding to unit tests, integration tests and user validation.

End to End testing

Although one End to End testing must be associated to the Integration Test, we can show here a quick testing to check that everything is up and running. The first test step involves creating a new resource as well as the implicit creation of a collection. The second test step checks if meta information in different file formats can be obtained.

Step 1 - Create the Resource

Create a file named resource.xml with resource content like this.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<resource>
  <creator>Yo</creator>
  <creationDate></creationDate>
  <modificationDate></modificationDate>
  <name>Resource Example</name>
  <contentUrl>http://localhost:8080/FiwareRepository/v2/collec/collectionA/collectionB/ResourceExample</contentUrl>
  <contentFileName>http://whereistheresource.com/ResourceExample</contentFileName>
</resource>
```

Send the request:

```
curl -v -H "Content-Type: application/xml" -X POST --data "@resource.xml" http://[SERVER_URL]:8080/FiwareRepository/v2/collec/collectionA/collectionB/ResourceExample
```

You should receive a HTTP/1.1 201 as status code

Create a file named resourceContent.txt with arbitrary content.

```
curl -v -H "Content-Type: text/plain" -X PUT --data "@resourceContent.txt" http://localhost:8080/FiwareRepository/v2/collec/collectionA/collectionB/ResourceExample
```

You should receive a HTTP/1.1 200 as status code

Step 2 - Retrieve meta information

Test HTML Response:

Open `http://[SERVER_URL]:8080/FiwareRepository/v2/collec/collectionA/` in your web browser. You should receive meta information about the implicit created collection in HTML format.

Test Text Response:

```
curl -v -H "Content-Type: text/plain" -X GET http://[SERVER_URL]:8080/FiwareRepository/v2/collec/collectionA/
```

You should receive meta information about the implicit created collection in text format. You may use curl to also test the other supported content types (application/json, application/rdf+xml, text/turtle, text/n3, text/html, text/plain, application/xml)

List of Running Processes

You can execute the command `ps -ax | grep 'tomcat\|mongo\|virtuoso'` to check that the Tomcat web server, the Mongo database, and Virtuoso Triple Store are running. It should show a message text similar to the following:

```
1048 ?          Ssl      0:51 /usr/bin/mongod --config /etc/mongodb.conf
1112 pts/1      SNl      0:01 virtuoso-t -f
1152 ?          Sl       0:03 /usr/lib/jvm/java-8-oracle/bin/java -Djava.util.logging.config.file=/home/...
2031 pts/1      S+       0:00 grep --color=auto --exclude-dir=.bzip --exclude-dir=.cvs --exclude-dir=.git
```

Network interfaces Up & Open

To check whether the ports in use are listening, execute the command `netstat -ntpl`. The expected results must be somehow similar to the following:

tcp	0	0	127.0.0.1:28017	0.0.0.0:*	ESCUCHAR	-
tcp	0	0	127.0.1.1:53	0.0.0.0:*	ESCUCHAR	-
tcp	0	0	0.0.0.0:1111	0.0.0.0:*	ESCUCHAR	11271/virtuoso-t
tcp	0	0	127.0.0.1:631	0.0.0.0:*	ESCUCHAR	-
tcp	0	0	0.0.0.0:8890	0.0.0.0:*	ESCUCHAR	11271/virtuoso-t
tcp	0	0	127.0.0.1:27017	0.0.0.0:*	ESCUCHAR	-
tcp6	0	0	:::8080	:::*	ESCUCHAR	11286/java
tcp6	0	0	:::1:631	:::*	ESCUCHAR	-
tcp6	0	0	127.0.0.1:8005	:::*	ESCUCHAR	11286/java
tcp6	0	0	:::8009	:::*	ESCUCHAR	11286/java

Databases

The last step in the sanity check (once that we have identified the processes and ports) is to check the databases that has to be up and accept queries. For that, we execute the following commands:

- MongoDB

```
$ mongo
MongoDB shell version: 2.4.9
connecting to: test
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
http://docs.mongodb.org/
Questions? Try the support group
http://groups.google.com/group/mongodb-user
> db
```

It should show a message text similar to the following:

```
test
```

- Virtuoso

```
$isql
OpenLink Interactive SQL (Virtuoso), version 0.9849b.
Type HELP; for help and EXIT; to exit.
SQL> SPARQL SELECT DISTINCT ?g WHERE { GRAPH ?g { ?s ?q ?l } };
```

It should show a message text similar to the following:

```
g
LONG VARCHAR

http://www.openlinksw.com/schemas/virttrdf#
http://www.w3.org/ns/ldp#
http://localhost:8890/sparql
http://localhost:8890/DAV/
http://www.w3.org/2002/07/owl#

5 Rows. -- 90 msec.
```

1.1.6 Diagnosis Procedures

The Diagnosis Procedures are the first steps that a System Administrator has to take to locate the source of an error in a GE. Once the nature of the error is identified by these tests, the system admin can resort to more concrete and specific testing to pinpoint the exact point of error and a possible solution.

The resource load of the Repository-RI strongly depends on the number of concurrent requests received as well as on the free main memory and disk space:

- Minimum available main memory: 1 GB
- Minimum available hard disk space: 2 GB

Resource availability

State the amount of available resources in terms of RAM and hard disk that are necessary to have a healthy enabler. This means that below these thresholds the enabler is likely to experience problems or bad performance.

Resource consumption

Resource consumption strongly depends on the load, especially on the number of concurrent requests.

The main memory consumption of the Tomcat application server should be between 48MB and 1024MB. These numbers can vary significantly if you use a different application server.

I/O flows

The only expected I/O flow is of type HTTP or HTTPS, on ports defined in Apache Tomcat configuration files, inbound and outbound. Requests interactivity should be low.

1.2 User and Programmer Guide

1.2.1 Introduction

This document describes the basic concepts regarding the Repository GEri and explains the necessary steps to develop applications which makes use of the Repository backend functionality. The Repository API is based on REST principles and generally returns XML or JSON encoded responses. Since REST is independent of a concrete programming language, you just have to know how to make an HTTP request in the programming language of your choice.

Despite the Repository is a Generic Enabler which provides pure backend functionality to other applications (e.g. Generic Enablers or end user facing applications), it is possible to browse between the existing collections and resources using the basic web interface provided.

To give you a feeling of how the Repository works and how you can interact with the system let us take a look at some examples, realized with the command line tool cURL. 'cURL' is a command which can be used to perform any kind of HTTP operation - and therefore is also usable for the Repository. The library libcurl enables the integration in C programs as well. Other easy way to interact with the Repository is using any REST client like some browsers extensions provide to the user. We describe the API REST to show you what operations is able to do, and what responses are given by the Repository.

1.2.2 User Guide

The Repository allows the client to choose the appropriate data format for retrieving meta information about a resource or a collection. One of this data format is `text/html` which provides a user friendly interface to see the different collection and resources stored in the repository.

Browsing Collections

It is possible to visualize the existing collections registered inside a given collection in the Repository by accessing the URL `http://[SERVER_HOST]/FiwareRepository/v2/collec/{collection}`

Collection: collectionA

Creation Date: Wed May 27 16:05:09 CEST 2015

Collections:

Collection Id	Creation Date
collectionA/collectionB	Wed May 27 16:05:09 CEST 2015
collectionA/collectionC	Thu May 28 15:40:29 CEST 2015
collectionA/collectionD	Thu May 28 15:40:34 CEST 2015

Resources:

Resource Id	Resource Meta Information	Creation Date	Modification Date	Filename	Mime Type
-------------	---------------------------	---------------	-------------------	----------	-----------

It is possible to visualize the existing resources contained in a given collection by accessing the URL `http://[SERVER_HOST]/FiwareRepository/v2/collec/{collectionA}/{collectionB}/`

Collection: collectionA/collectionB

Creation Date: Wed May 27 16:05:09 CEST 2015

Collections:

Collection Id	Creation Date
---------------	---------------

Resources:

Resource Id	Resource Meta Information	Creation Date	Modification Date	Filename	Mime Type
collectionA/collectionB/ResourceExample	Meta Information	Wed May 27 16:05:09 CEST 2015	Wed May 27 16:06:51 CEST 2015	http://whereistheresource.com/ResourceExample	text/plain
collectionA/collectionB/resource1	Meta Information	Thu May 28 15:39:24 CEST 2015	Thu May 28 15:39:24 CEST 2015	resourceFileName	
collectionA/collectionB/resource	Meta Information	Thu May 28 15:39:51 CEST 2015	Thu May 28 15:39:51 CEST 2015	resourceFileName	
collectionA/collectionB/resource2	Meta Information	Thu May 28 15:40:17 CEST 2015	Thu May 28 15:40:17 CEST 2015	resourceFileName	
collectionA/collectionB/resource3	Meta Information	Thu May 28 15:40:21 CEST 2015	Thu May 28 15:40:21 CEST 2015	resourceFileName	

Browsing Resources

It is possible to visualize the meta information of a given resource by accessing the URL
`http://[SERVER_HOST]/FiwareRepository/v2/collec/{collectionA}/{collectionB}/{resource}`

Resource:
collectionA/collectionB/ResourceExample

Resource Id	Creation Date	Modification Date	Filename	Mime Type
collectionA/collectionB/ResourceExample	Wed May 27 16:05:09 CEST 2015	Wed May 27 16:06:51 CEST 2015	http://whereistheresource.com/ResourceExample	text/plain

Note: Depending on the configuration of the Repository RI instance, you may be redirected to the FIWARE Lab login page in order to be authenticated.

1.2.3 Programmer Guide

The Repository offers pure backend functionality; in this way, the current section gives an overview of the operations that can be made with the API in order to integrate the functionality provided by the Repository with an existing solution.

This section contains an overview of the existing REST API. You can find detailed documentation of the API in [Apiary](#) and in [GitHub Pages](#).

Managing Collections

Following you can find the description of the operations that can be performed over collections:

- GET [http://\[SERVER_HOST\]/FiwareRepository/v2/collec/{collection}](http://[SERVER_HOST]/FiwareRepository/v2/collec/{collection}) Get a collection.
- POST [http://\[SERVER_HOST\]/FiwareRepository/v2/collec/](http://[SERVER_HOST]/FiwareRepository/v2/collec/) Create a collection.
- DELETE [http://\[SERVER_HOST\]/FiwareRepository/v2/collec/{collection}](http://[SERVER_HOST]/FiwareRepository/v2/collec/{collection})/ Delete a resource

Note: Every collection name must comply the regular expression "[a-zA-Z0-9_-]+".

Getting collections

- Request
 - Verb: GET

- URI: `http://[SERVER_HOST]/FiwareRepository/v2/collec/{collection}`
- Content-type: `application/json`, `application/xml`, `text/plain`

- Responses

HTTP Code	Type	Description
200	OK	Your request has been completed properly.
404	Not Found	The collection with the given path has not been found.
406	Not Acceptable	The collection can not be generated in the given format.
500	Internal Server Error	There was an internal error in the system so your request cannot be completed.

Creating collections

- Request

- Verb: POST
- URI: `http://[SERVER_HOST]/FiwareRepository/v2/collec/[collection]`
- Content-Type: `application/json`, `application/xml`
- Body:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<collection xmlns:atom="http://www.w3.org/2005/Atom">
  <creator>CreatornameUpdate</creator>
  <name>TestCollection</name>
  <collections/>
  <resources/>
</collection>
```

- or

```
{
  "type": "collection",
  "creator": "Creator",
  "creationDate": "",
  "modificationDate": "",
  "name": "TestCollection"
}
```

- Responses

HTTP Code	Type	Description
201	OK	The collection has been created. The <i>location</i> header will contain the final URL that users can use to access this collection.
400	Bad Request	The collection is not well formed or collection name is not valid.
409	Conflict	The collection with the path and the name given already exist.
415	Unsoported Media Type	The request entity has a media type which a collection does not support.
500	Internal Server Error	There was an internal error in the system so your request cannot be completed.

Removing collections

- Request
 - Verb: DELETE
 - URI: [http://\[SERVER_HOST\]FiwareRepository/v2/collec/{collection}](http://[SERVER_HOST]FiwareRepository/v2/collec/{collection})
- Responses

HTTP Code	Type	Description
204	No Content	The collection and all its content has been deleted.
404	Not Found	The collection with the given path has not been found.
500	Internal Server Error	There was an internal error in the system so your request cannot be completed.

Managing Resources

Following you can find the description of the operations that can be performed over resources:

- GET [http://\[SERVER_HOST\]FiwareRepository/v2/collec/{collection}/{resource}.meta](http://[SERVER_HOST]FiwareRepository/v2/collec/{collection}/{resource}.meta) Get metadata of a resource
- GET [http://\[SERVER_HOST\]FiwareRepository/v2/collec/{collection}/{resource}](http://[SERVER_HOST]FiwareRepository/v2/collec/{collection}/{resource}) Get content of a resource.
- POST [http://\[SERVER_HOST\]FiwareRepository/v2/collec/](http://[SERVER_HOST]FiwareRepository/v2/collec/) Create a resource with specified metadata.
- PUT [http://\[SERVER_HOST\]FiwareRepository/v2/collec/{collection}/{resource}.meta](http://[SERVER_HOST]FiwareRepository/v2/collec/{collection}/{resource}.meta) Replace metadata of a resource.
- PUT [http://\[SERVER_HOST\]FiwareRepository/v2/collec/{collection}/{resource}](http://[SERVER_HOST]FiwareRepository/v2/collec/{collection}/{resource}) Replace content of a resource.
- DELETE [http://\[SERVER_HOST\]FiwareRepository/v2/collec/{collection}/{resource}](http://[SERVER_HOST]FiwareRepository/v2/collec/{collection}/{resource}) Delete a resource

Note: Every resource name must comply the regular expresion "[a-zA-Z0-9._-] +".

Getting resources metadata

- Request
 - Verb: GET
 - URI: [http://\[SERVER_HOST\]FiwareRepository/v2/collec/{collection}/{resource}.meta](http://[SERVER_HOST]FiwareRepository/v2/collec/{collection}/{resource}.meta)
 - Accept: application/json, application/xml, text/plain
- Responses

HTTP Code	Type	Description
200	OK	Your request has been completed properly.
404	Not Found	The resource with the given path has not been found.
406	Not Acceptable	The resource can not be generated in the given format.
500	Internal Server Error	There was an internal error in the system so your request cannot be completed.

Getting resources

- Request
 - Verb: GET
 - URI: `http://[SERVER_HOST]/FiwareRepository/v2/collec/{collection}/{resource}`
 - Accept: application/json, application/xml, application/rdf+xml, text/turtle, application/x-turtle, text/n3, text/rdf+n3, text/n-triples, text/plain
- Responses

HTTP Code	Type	Description
200	OK	Your request has been completed properly.
204	No Content	Your request has been processed, but the resource has not got content.
404	Not Found	The resource with the given path has not been found.
406	Not Acceptable	The resource content can not be generated in the given format.
500	Internal Server Error	There was an internal error in the system so your request cannot be completed.

Creating resources

- Request
 - Verb: POST
 - URI: `http://[SERVER_HOST]/FiwareRepository/v2/collec/{collection}`
 - Content-Type: application/json, application/xml
 - Body:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<resource>
  <creator>Creator</creator>
  <creationDate></creationDate>
  <modificationDate></modificationDate>
  <name>TestResource</name>
  <contentUrl>http://testresourceurl.com/resource</contentUrl>
  <contentFileName>resourceFileName</contentFileName>
</resource>
```

– or

```
{
  "type": "resource",
  "creator": "Creator",
  "creationDate": "",
  "modificationDate": "",
  "name": "TestResource",
  "contentUrl": "http://testresourceurl.com/resource",
  "contentFileName": "resourceFileName"
}
```

- Responses

HTTP Code	Type	Description
201	Created	The resource has been created. The <i>location</i> header will contain the final URL that users can use to access this collection.
400	Bad Request	The resource is not well formed or collection name is not valid.
409	Conflict	The resource with the path and the name given already exist.
415	Unsoported Media Type	The request entity has a media type which a resource does not support.
500	Internal Server Error	There was an internal error in the system so your request cannot be completed.

Updating resources metadata

- Request

- Verb: PUT
- URI: `http://[SERVER_HOST]/FiwareRepository/v2/collec/{collection}/{resource}.meta`
- Content-Type: `application/json`, `application/xml`
- **Body:** :: `<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <resource id="collectionA/collectionB/resource">`
`<creator>Creator</creator> <creationDate></creationDate> <modification-`
`Date></modificationDate> <name>Test resource</name> <contentFile-`
`Name>resourceFileName</contentFileName>`
`</resource>`
- or

```
{
  "type": "resource",
  "creator": "Creator",
  "creationDate": "",
  "modificationDate": "",
  "name": "TestResource",
  "contentUrl": "http://testresourceurl.com/resource",
  "contentFileName": "resourceFileName"
}
```

- Responses

HTTP Code	Type	Description
200	OK	Your request has been completed properly.
400	Bad Request	The request is not well formed or collection name is not valid.
403	Forbidden	The request was a valid request, but the server is refusing to respond to it.
409	Conflict	The resource with the path and the name given already exist.
415	Unsoported Media Type	The request entity has a media type which a resource does not support.
500	Internal Server Error	There was an internal error in the system so your request cannot be completed.

Updating resources content

- Request
 - Verb: PUT
 - URI: [http://\[SERVER_HOST\]/FiwareRepository/v2/collec/{collection}/{resource}](http://[SERVER_HOST]/FiwareRepository/v2/collec/{collection}/{resource})
 - Content-Type: application/json, application/xml, application/rdf+xml, text/turtle, application/x-turtle, text/n3, text/rdf+n3, text/n-triples, text/plain
- Responses

HTTP Code	Type	Description
200	OK	Your request has been completed properly.
400	Bad Request	The resource content is not well formed.
404	Not Found	The resource with the given path has not been found.
415	Unsoported Media Type	The request entity has a media type which the resource content does not support.
500	Internal Server Error	There was an internal error in the system so your request cannot be completed.

Deleting resources

- Request
 - Verb: PUT
 - URI: [http://\[SERVER_HOST\]/FiwareRepository/v2/collec/{collection}/{resource}](http://[SERVER_HOST]/FiwareRepository/v2/collec/{collection}/{resource})
- Responses

HTTP Code	Type	Description
204	No Content	The resource and all its content has been deleted.
404	Not Found	The resource with the given path has not been found.
500	Internal Server Error	There was an internal error in the system so your request cannot be completed.

Making SPARQL queries

Following you can find the quereng operations that can be executed.

- GET [http://\[SERVER_HOST\]/FiwareRepository/v2/services/query/{contentUrl}](http://[SERVER_HOST]/FiwareRepository/v2/services/query/{contentUrl})
- GET [http://\[SERVER_HOST\]/FiwareRepository/v2/services/query?query=\[Query\]](http://[SERVER_HOST]/FiwareRepository/v2/services/query?query=[Query]) Execute a query in the triple store.
- POST [http://\[SERVER_HOST\]/FiwareRepository/v2/services/query](http://[SERVER_HOST]/FiwareRepository/v2/services/query) | Execute a long query in the triple store.

Getting a resource by Url Content

- Request
 - Verb: GET
 - URI: [http://\[SERVER_HOST\]/FiwareRepository/v2/services/query/{contentUrl}](http://[SERVER_HOST]/FiwareRepository/v2/services/query/{contentUrl})

- Responses

- Verb: GET

- Responses

- Verb: GET

- Accept: application/json, application/xml, application/rdf+xml, text/turtle, application/x-turtle, text/n3, text/rdf+n3, text/n-triples, text/plain

- **Body:** "SELECT ?s ?p ?o WHERE { ?s ?p ?o }"

- | HTTP Code | Type | Description |
|-----------|-----------------------|---|
| 200 | OK | Your request has been completed properly. |
| 400 | Bad Request | The query is not well formed. |
| 406 | Not Acceptable | The query response can not be generated in the given format. |
| 415 | Unsoported Media Type | The request entity has a media type which a query does not support. |

Creating a resource

- Create a message body and save it to a file named `resource.xml`.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<resource>
  <creator>Creator</creator>
  <creationDate></creationDate>
  <modificationDate></modificationDate>
  <name>resource</name>
  <contentUrl>http://testresourceurl.com/resource</contentUrl>
  <contentFileName>resourceFileName</contentFileName>
</resource>
```

- Send the request to the server.

```
curl -v -H "Content-Type: application/xml" -X POST --data "@resource.xml" http://[SERVER_HOST]/F
```

- Response

```
* Hostname was NOT found in DNS cache
*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 8080 (#0)
> POST /FiwareRepository/v2/collec HTTP/1.1
> User-Agent: curl/7.35.0
> Host: localhost:8080
> Accept: */*
> Content-Type: application/xml
> Content-Length: 349
>
* upload completely sent off: 349 out of 349 bytes
< HTTP/1.1 201 Created
* Server Apache-Coyote/1.1 is not blacklisted
< Server: Apache-Coyote/1.1
< Content-Location: http://testresourceurl.com/resource
< Content-Length: 0
< Date: Thu, 28 May 2015 12:12:23 GMT
<
* Connection #0 to host localhost left intact
```

Updating the content of a resource

- Create a message body and save it to a file named `resourceContent.xml`.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:app="http://www.app.fake/app#">

  <rdf:Description
    rdf:about="http://www.app.fake/app/App1">
    <app:name>App1</app:name>
    <app:country>USA</app:country>
    <app:company>Company1</app:company>
    <app:price>0.99</app:price>
    <app:year>2010</app:year>
  </rdf:Description>

  <rdf:Description
```

```
rdflib:about="http://www.app.fake/app/App2">
  <app:name>App2</app:name>
  <app:country>Spain</app:country>
  <app:company>Company2</app:company>
  <app:price>0.99</app:price>
  <app:year>2010</app:year>
</rdflib:Description>

</rdflib:RDF>
```

- Send the request to the server.

```
curl -v -H "Content-Type: application/rdf+xml" -X PUT --data-binary "@resourceContent.xml" http:
```

- Response:

```
* Hostname was NOT found in DNS cache
*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 8080 (#0)
> PUT /FiwareRepository/v2/collec/collectionA/collectionB/resource HTTP/1.1
> User-Agent: curl/7.35.0
> Host: localhost:8080
> Accept: */*
> Content-Type: application/rdf+xml
> Content-Length: 645
>
* upload completely sent off: 645 out of 645 bytes
< HTTP/1.1 200 OK
* Server Apache-Coyote/1.1 is not blacklisted
< Server: Apache-Coyote/1.1
< Content-Length: 0
< Date: Thu, 28 May 2015 12:41:21 GMT
<
* Connection #0 to host localhost left intact
```

Getting a resource content

- Send the request to the server.

```
curl -v -H "Accept: application/rdf+xml" -X GET http://localhost:8080/FiwareRepository/v2/collec
```

- Response:

```
* Hostname was NOT found in DNS cache
*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 8080 (#0)
> GET /FiwareRepository/v2/collec/collectionA/collectionB/resource HTTP/1.1
> User-Agent: curl/7.35.0
> Host: localhost:8080
> Accept: application/rdf+xml
>
< HTTP/1.1 200 OK
* Server Apache-Coyote/1.1 is not blacklisted
< Server: Apache-Coyote/1.1
< Content-Type: application/rdf+xml
< Content-Length: 645
< Date: Thu, 28 May 2015 12:43:23 GMT
<
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:app="http://www.app.fake/app#">

  <rdf:Description
    rdf:about="http://www.app.fake/app/App1">
    <app:name>App1</app:name>
    <app:country>USA</app:country>
    <app:company>Company1</app:company>
    <app:price>0.99</app:price>
    <app:year>2010</app:year>
  </rdf:Description>

  <rdf:Description
    rdf:about="http://www.app.fake/app/App2">
    <app:name>App2</app:name>
    <app:country>Spain</app:country>
    <app:company>Company2</app:company>
    <app:price>0.99</app:price>
    <app:year>2010</app:year>
  </rdf:Description>

</rdf:RDF>
* Connection #0 to host localhost left intact
```

Deleting a resource

- Send the request to the server.

```
curl -v -X DELETE http://localhost:8080/FiwareRepository/v2/collec/collectionA/collectionB/resource
```

- Response:

```
* Hostname was NOT found in DNS cache
*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 8080 (#0)
> DELETE /FiwareRepository/v2/collec/collectionA/collectionB/resource HTTP/1.1
> User-Agent: curl/7.35.0
> Host: localhost:8080
> Accept: */*
>
< HTTP/1.1 204 No Content
* Server Apache-Coyote/1.1 is not blacklisted
< Server: Apache-Coyote/1.1
< Content-Length: 0
< Date: Thu, 28 May 2015 12:46:33 GMT
<
* Connection #0 to host localhost left intact
```

Executing a Query

- Send the request to the server.

```
curl -v -H "Accept: application/json" -H "Content-Type: text/plain" -X POST --data "SELECT ?s ?p"
```

- Response:

```
* Hostname was NOT found in DNS cache
* Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 8080 (#0)
> POST /FiwareRepository/v2/services/query HTTP/1.1
> User-Agent: curl/7.35.0
> Host: localhost:8080
> Accept: application/json
> Content-Type: text/plain
> Content-Length: 41
>
* upload completely sent off: 41 out of 41 bytes
< HTTP/1.1 200 OK
* Server Apache-Coyote/1.1 is not blacklisted
< Server: Apache-Coyote/1.1
< Content-Type: application/json
< Transfer-Encoding: chunked
< Date: Thu, 28 May 2015 13:33:32 GMT
<
{
  vars: 3
  columns: [3]
  0: {
    values: [10]
    0: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
    1: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
    2: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
    3: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
    4: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
    5: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
    6: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
    7: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
    8: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
    9: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
    -
    name: "p"
  }-
  1: {
    values: [10]
    0: "http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat"
    1: "http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat"
    2: "http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat"
    3: "http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat"
    4: "http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat"
    5: "http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat"
    6: "http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat"
    7: "http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat"
    8: "http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat"
    9: "http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat"
    -
    name: "o"
  }-
  2: {
    values: [10]
    0: "http://www.openlinksw.com/virtrdf-data-formats#default-iid"
    1: "http://www.openlinksw.com/virtrdf-data-formats#default-iid-nullable"
    2: "http://www.openlinksw.com/virtrdf-data-formats#default-iid-nonblank"
    3: "http://www.openlinksw.com/virtrdf-data-formats#default-iid-nonblank-nullable"
```

```

4:  "http://www.openlinksw.com/virtrdf-data-formats#default"
5:  "http://www.openlinksw.com/virtrdf-data-formats#default-nullable"
6:  "http://www.openlinksw.com/virtrdf-data-formats#sql-varchar"
7:  "http://www.openlinksw.com/virtrdf-data-formats#sql-varchar-nullable"
8:  "http://www.openlinksw.com/virtrdf-data-formats#sql-varchar-dt"
9:  "http://www.openlinksw.com/virtrdf-data-formats#sql-varchar-dt-nullable"
-
name: "s"
}-
-
}

```

HTTP content negotiation allows the client to choose the appropriate data format for retrieving meta information about a resource or a collection. Besides XML and JSON the Repository also supports human readable output formats using HTML rendering ('text/html' accept header) including hyperlinked representation and formatted text.

Text Representation

- Request URL: [http://\[REPOSITORY_URL\]/v2/collec/{collection}](http://[REPOSITORY_URL]/v2/collec/{collection})
- Accept Header: text/plain
- Result:

```

Collection: testCollection
Creation Date: Thu Mar 21 10:46:39 CET 2013

```

Collections:

```

+++++
++  Collection Id          +  Creation Date          ++
+++++
++  testCollection/collectionA  +  Thu Mar 21 10:47:53 CET 2013  ++
++  testCollection/collectionB  +  Thu Mar 21 10:47:53 CET 2013  ++
++  testCollection/           +  Thu Mar 21 10:47:53 CET 2013  ++
+++++

```

Resources:

```

+++++
++  Resource Id          +  Creation Date          +  Modification Date
+++++
++  testCollection/testResource1  +  Thu Mar 21 10:46:39 CET 2013  +  Thu Mar 21 10:46:39 CE
++  testCollection/testResource2  +  Thu Mar 21 10:47:53 CET 2013  +  Thu Mar 21 10:47:53 CE
++  testCollection/testResource3  +  Thu Mar 21 10:47:53 CET 2013  +  Thu Mar 21 10:47:53 CE
++  testCollection/testResource4  +  Thu Mar 21 10:47:53 CET 2013  +  Thu Mar 21 10:47:53 CE
+++++

```

JSON Representation

- Request URL: [http://\[REPOSITORY_URL\]/v2/collec/{collection}](http://[REPOSITORY_URL]/v2/collec/{collection})
- Accept Header: application/json
- Result:

```

{
  "resources": [
    {

```

```
    "type": "resource",
    "name": "testResource1",
    "content": null,
    "collection": null,
    "contentType": "application\/rdf+xml",
    "contentFileName": "filename",
    "contentUrl": "",
    "id": "testCollection\/testResource1",
    "creationDate": 1363859199839,
    "creator": "",
    "modificationDate": 1363859199839
  },
  {
    "type": "resource",
    "name": "testResource2",
    "content": null,
    "collection": null,
    "contentType": "plain\/text",
    "contentFileName": "filename",
    "contentUrl": "",
    "id": "testCollection\/testResource2",
    "creationDate": 1363859273515,
    "creator": "",
    "modificationDate": 1363859273515
  },
  {
    "type": "resource",
    "name": "testResource3",
    "content": null,
    "collection": null,
    "contentType": "text\/turtle",
    "contentFileName": "filename",
    "contentUrl": "",
    "id": "testCollection\/testResource3",
    "creationDate": 1363859273535,
    "creator": "",
    "modificationDate": 1363859273535
  },
  {
    "type": "resource",
    "name": "testResource4",
    "content": null,
    "collection": null,
    "contentType": "application\/rdf+xml",
    "contentFileName": "filename",
    "contentUrl": "",
    "id": "testCollection\/testResource4",
    "creationDate": 1363859273545,
    "creator": "",
    "modificationDate": 1363859273545
  }
],
"collections": [
  {
    "resources": [

    ],
    "collections": [
```

```

    ],
    "type": "collection",
    "id": "testCollection\\collectionA",
    "name": "collectionA",
    "creationDate": 1363859273552,
    "creator": "",
    "modificationDate": null
  },
  {
    "resources": [

    ],
    "collections": [

    ],
    "type": "collection"
    "id": "testCollection\\collectionB",
    "name": "collectionB",
    "creationDate": 1363859273566,
    "creator": "",
    "modificationDate": null
  },
  {
    "resources": [

    ],
    "collections": [

    ],
    "type": "collection"
    "id": "testCollection\\",
    "name": "testCollection",
    "creationDate": 1363859273575,
    "creator": "",
    "modificationDate": null
  }
],
"type": "collection"
"id": "testCollection",
"creationDate": 1363859199837,
"creator": "",
"modificationDate": null
}

```

XML Representation

- Request URL: [http://\[REPOSITORY_URL\]/v2/collec/{collection}](http://[REPOSITORY_URL]/v2/collec/{collection})
- Accept Header: application/json
- Result:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<collection id="testCollection" xmlns:atom="http://www.w3.org/2005/Atom">
  <creationDate>2013-03-21T10:46:39.837+01:00</creationDate>
  <creator/>
  <collections>

```

```
<collections id="testCollection/collectionA">
  <creationDate>2013-03-21T10:47:53.552+01:00</creationDate>
  <name>collectionA</name>
  <creator/>
  <collections/>
  <resources/>
</collections>
<collections id="testCollection/collectionB">
  <creationDate>2013-03-21T10:47:53.566+01:00</creationDate>
  <name>collectionB</name>
  <creator/>
  <collections/>
  <resources/>
</collections>
<collections id="testCollection/">
  <creationDate>2013-03-21T10:47:53.575+01:00</creationDate>
  <name>testCollection</name>
  <creator/>
  <collections/>
  <resources/>
</collections>
</collections>
<resources>
  <resources id="testCollection/testResource1">
    <creationDate>2013-03-21T10:46:39.839+01:00</creationDate>
    <creator/>
    <modificationDate>2013-03-21T10:46:39.839+01:00</modificationDate>
    <contentFileName>filename</contentFileName>
    <contentType>application/rdf+xml</contentType>
    <contentUrl/>
    <name>testResource1</name>
  </resources>
  <resources id="testCollection/testResource2">
    <creationDate>2013-03-21T10:47:53.515+01:00</creationDate>
    <creator/>
    <modificationDate>2013-03-21T10:47:53.515+01:00</modificationDate>
    <contentFileName>filename</contentFileName>
    <contentType>plain/text</contentType>
    <contentUrl/>
    <name>testResource2</name>
  </resources>
  <resources id="testCollection/testResource3">
    <creationDate>2013-03-21T10:47:53.535+01:00</creationDate>
    <creator/>
    <modificationDate>2013-03-21T10:47:53.535+01:00</modificationDate>
    <contentFileName>filename</contentFileName>
    <contentType>text/turtle</contentType>
    <contentUrl/>
    <name>testResource3</name>
  </resources>
  <resources id="testCollection/testResource4">
    <creationDate>2013-03-21T10:47:53.545+01:00</creationDate>
    <creator/>
    <modificationDate>2013-03-21T10:47:53.545+01:00</modificationDate>
    <contentFileName>filename</contentFileName>
    <contentType>application/rdf+xml</contentType>
    <contentUrl/>
    <name>testResource4</name>
```



```

    </resources>
  </resources>
</collection>

```

When you insert some resource content in any of the accepted RDF formats, it is also possible to retrieve that content in any of the allowed RDF formats. This section explains how to retrieve resources in the different supported formats.

XML+RDF Representation

- Request URL: [http://\[REPOSITORY_URL\]/v2/collec/{collection}/{resource}](http://[REPOSITORY_URL]/v2/collec/{collection}/{resource})
- Accept Header: application/rdf+xml
- Result:

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:app="http://www.app.fake/app#">

  <rdf:Description
    rdf:about="http://www.app.fake/app/App1">
    <app:name>App1</app:name>
    <app:country>USA</app:country>
    <app:company>Company1</app:company>
    <app:price>0.99</app:price>
    <app:year>2010</app:year>
  </rdf:Description>

  <rdf:Description
    rdf:about="http://www.app.fake/app/App2">
    <app:name>App2</app:name>
    <app:country>Spain</app:country>
    <app:company>Company2</app:company>
    <app:price>0.99</app:price>
    <app:year>2010</app:year>
  </rdf:Description>
</rdf:RDF>

```

Turtle Representation

- Request URL: [http://\[REPOSITORY_URL\]/v2/collec/{collection}/{resource}](http://[REPOSITORY_URL]/v2/collec/{collection}/{resource})
- Accept Header: "text/turtle"
- Result:

```

@prefix rdfdf: <http://www.openlinksw.com/virtrdf-data-formats#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix fn: <http://www.w3.org/2005/xpath-functions/#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix yago: <http://dbpedia.org/class/yago/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix ogc: <http://www.opengis.net/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ogcgsf: <http://www.opengis.net/def/function/geosparql/> .
@prefix protseq: <http://purl.org/science/protein/bysequence/> .
@prefix xsld: <http://www.w3.org/TR/WD-xsl> .
@prefix sql: <sql:> .

```

```
@prefix geo:    <http://www.w3.org/2003/01/geo/wgs84_pos#> .
@prefix sc:     <http://purl.org/science/owl/sciencecommons/> .
@prefix sd:     <http://www.w3.org/ns/sparql-service-description#> .
@prefix ogcsf:  <http://www.opengis.net/ont/sf#> .
@prefix nci:    <http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#> .
@prefix virtrdf: <http://www.openlinksw.com/schemas/virtrdf#> .
@prefix xml:    <http://www.w3.org/XML/1998/namespace> .
@prefix ogcgm1: <http://www.opengis.net/ont/gml#> .
@prefix ogcgsr: <http://www.opengis.net/def/rule/geosparql/> .
@prefix rdfa:   <http://www.w3.org/ns/rdfa#> .
@prefix mf:     <http://www.w3.org/2001/sw/DataAccess/tests/test-manifest#> .
@prefix scovo:  <http://purl.org/NET/scovo#> .
@prefix virtcxml: <http://www.openlinksw.com/schemas/virtcxml#> .
@prefix dbpprop: <http://dbpedia.org/property/> .
@prefix foaf:   <http://xmlns.com/foaf/0.1/> .
@prefix mesh:   <http://purl.org/commons/record/mesh/> .
@prefix sioc:   <http://rdfs.org/sioc/ns#> .
@prefix xsl10:  <http://www.w3.org/XSL/Transform/1.0> .
@prefix product: <http://www.buy.com/rss/module/productV2/> .
@prefix void:   <http://rdfs.org/ns/void#> .
@prefix dawgt:  <http://www.w3.org/2001/sw/DataAccess/tests/test-dawg#> .
@prefix go:     <http://purl.org/obo/owl/GO#> .
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix vcard:  <http://www.w3.org/2001/vcard-rdf/3.0#> .
@prefix xsl1999: <http://www.w3.org/1999/XSL/Transform> .
@prefix xf:     <http://www.w3.org/2004/07/xpath-functions> .
@prefix rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix vcard2006: <http://www.w3.org/2006/vcard/ns#> .
@prefix ldp:    <http://www.w3.org/ns/ldp#> .
@prefix math:   <http://www.w3.org/2000/10/swap/math#> .
@prefix ogcgs:  <http://www.opengis.net/ont/geosparql#> .
@prefix obo:    <http://www.geneontology.org/formats/oboInOwl#> .
@prefix bif:    <bif:> .
@prefix dc:     <http://purl.org/dc/elements/1.1/> .

<http://www.app.fake/app/App1>
  <http://www.app.fake/app#company>
    "Company1" ;
  <http://www.app.fake/app#country>
    "USA" ;
  <http://www.app.fake/app#name>  "App1" ;
  <http://www.app.fake/app#price>
    "0.99" ;
  <http://www.app.fake/app#year>  "2010" .

<http://www.app.fake/app/App2>
  <http://www.app.fake/app#company>
    "Company2" ;
  <http://www.app.fake/app#country>
    "Spain" ;
  <http://www.app.fake/app#name>  "App2" ;
  <http://www.app.fake/app#price>
    "0.99" ;
  <http://www.app.fake/app#year>  "2010" .
```

N3 Representation

- Request URL: [http://\[REPOSITORY_URL\]/v2/collec/{collection}/{resource}](http://[REPOSITORY_URL]/v2/collec/{collection}/{resource})
- Accept Header: "text/n3"
- Result:

```
@prefix rdfdf: <http://www.openlinksw.com/virtrdf-data-formats#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix fn: <http://www.w3.org/2005/xpath-functions/#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix yago: <http://dbpedia.org/class/yago/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix ogc: <http://www.opengis.net/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ogcgsf: <http://www.opengis.net/def/function/geosparql/> .
@prefix protseq: <http://purl.org/science/protein/bysequence/> .
@prefix xslwd: <http://www.w3.org/TR/WD-xsl> .
@prefix sql: <sql:> .
@prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#> .
@prefix sc: <http://purl.org/science/owl/sciencecommons/> .
@prefix sd: <http://www.w3.org/ns/sparql-service-description#> .
@prefix ogcsf: <http://www.opengis.net/ont/sf#> .
@prefix nci: <http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#> .
@prefix virtrdf: <http://www.openlinksw.com/schemas/virtrdf#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix ogcgml: <http://www.opengis.net/ont/gml#> .
@prefix ogcgsr: <http://www.opengis.net/def/rule/geosparql/> .
@prefix rdfa: <http://www.w3.org/ns/rdfa#> .
@prefix mf: <http://www.w3.org/2001/sw/DataAccess/tests/test-manifest#> .
@prefix scovo: <http://purl.org/NET/scovo#> .
@prefix virtcxml: <http://www.openlinksw.com/schemas/virtcxml#> .
@prefix dbpprop: <http://dbpedia.org/property/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix mesh: <http://purl.org/commons/record/mesh/> .
@prefix sioc: <http://rdfs.org/sioc/ns#> .
@prefix xsl10: <http://www.w3.org/XSL/Transform/1.0> .
@prefix product: <http://www.buy.com/rss/module/productV2/> .
@prefix void: <http://rdfs.org/ns/void#> .
@prefix dawgt: <http://www.w3.org/2001/sw/DataAccess/tests/test-dawg#> .
@prefix go: <http://purl.org/obo/owl/GO#> .
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix vcard: <http://www.w3.org/2001/vcard-rdf/3.0#> .
@prefix xsl1999: <http://www.w3.org/1999/XSL/Transform> .
@prefix xf: <http://www.w3.org/2004/07/xpath-functions> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix vcard2006: <http://www.w3.org/2006/vcard/ns#> .
@prefix ldp: <http://www.w3.org/ns/ldp#> .
@prefix math: <http://www.w3.org/2000/10/swap/math#> .
@prefix ogcgs: <http://www.opengis.net/ont/geosparql#> .
@prefix obo: <http://www.geneontology.org/formats/oboInOwl#> .
@prefix bif: <bif:> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .

<http://www.app.fake/app/App1>
  <http://www.app.fake/app#company>
    "Company1" ;
  <http://www.app.fake/app#country>
```

```
        "USA" ;
    <http://www.app.fake/app#name>   "App1" ;
    <http://www.app.fake/app#price>
        "0.99" ;
    <http://www.app.fake/app#year>   "2010" .

    <http://www.app.fake/app/App2>
    <http://www.app.fake/app#company>
        "Company2" ;
    <http://www.app.fake/app#country>
        "Spain" ;
    <http://www.app.fake/app#name>   "App2" ;
    <http://www.app.fake/app#price>
        "0.99" ;
    <http://www.app.fake/app#year>   "2010" .
```

N-Triples Representation

- Request URL: [http://\[REPOSITORY_URL\]/v2/collec/{collection}/{resource}](http://[REPOSITORY_URL]/v2/collec/{collection}/{resource})
- Accept Header: “text/n-triples”
- Result:

```
<http://www.app.fake/app/App1> <http://www.app.fake/app#company> "Company1" .
<http://www.app.fake/app/App2> <http://www.app.fake/app#company> "Company2" .
<http://www.app.fake/app/App1> <http://www.app.fake/app#country> "USA" .
<http://www.app.fake/app/App2> <http://www.app.fake/app#country> "Spain" .
<http://www.app.fake/app/App1> <http://www.app.fake/app#name> "App1" .
<http://www.app.fake/app/App2> <http://www.app.fake/app#name> "App2" .
<http://www.app.fake/app/App1> <http://www.app.fake/app#price> "0.99" .
<http://www.app.fake/app/App2> <http://www.app.fake/app#price> "0.99" .
<http://www.app.fake/app/App1> <http://www.app.fake/app#year> "2010" .
<http://www.app.fake/app/App2> <http://www.app.fake/app#year> "2010" .
```

When you execute a SPARQL query in the Repository, it generates different responses depending on the type of query. If you execute a SELECT query, it returns a json object composed by n columns being n number of variables. If you execute a CONSTRUCT or a DESCRIBE query, you can specify the RDF format as described in the previous section.

Executing SELECT query

- Request URL: [http://\[REPOSITORY_URL\]/v2/collec/{collection}/{resource}](http://[REPOSITORY_URL]/v2/collec/{collection}/{resource})
- Accept Header: “application/json”
- Body: SELECT ?s ?p ?o WHERE { ?s ?p ?o }
- Result:

```
{
  vars: 3
  columns: [3]
  0: {
    values: [15]
    0: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
    1: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
    2: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
    3: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
```

```

4: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
5: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
6: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
7: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
8: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
9: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
10: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
11: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
12: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
13: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
14: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
-
name: "p"
}-
1: {
  values: [15]
  0: "http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat"
  1: "http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat"
  2: "http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat"
  3: "http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat"
  4: "http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat"
  5: "http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat"
  6: "http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat"
  7: "http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat"
  8: "http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat"
  9: "http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat"
  10: "http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat"
  11: "http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat"
  12: "http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat"
  13: "http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat"
  14: "http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat"
  name: "o"
}-
2: {
  values: [15]
  0: "http://www.openlinksw.com/virtrdf-data-formats#default-iid"
  1: "http://www.openlinksw.com/virtrdf-data-formats#default-iid-nullable"
  2: "http://www.openlinksw.com/virtrdf-data-formats#default-iid-nonblank"
  3: "http://www.openlinksw.com/virtrdf-data-formats#default-iid-nonblank-nullable"
  4: "http://www.openlinksw.com/virtrdf-data-formats#default"
  5: "http://www.openlinksw.com/virtrdf-data-formats#default-nullable"
  6: "http://www.openlinksw.com/virtrdf-data-formats#sql-varchar"
  7: "http://www.openlinksw.com/virtrdf-data-formats#sql-varchar-nullable"
  8: "http://www.openlinksw.com/virtrdf-data-formats#sql-varchar-dt"
  9: "http://www.openlinksw.com/virtrdf-data-formats#sql-varchar-dt-nullable"
  10: "http://www.openlinksw.com/virtrdf-data-formats#sql-varchar-lang"
  11: "http://www.openlinksw.com/virtrdf-data-formats#sql-varchar-lang-nullable"
  12: "http://www.openlinksw.com/virtrdf-data-formats#sql-varchar-fixedlang-x-any"
  13: "http://www.openlinksw.com/virtrdf-data-formats#sql-varchar-fixedlang-x-any-nullable"
  14: "http://www.openlinksw.com/virtrdf-data-formats#sql-varchar-uri"
  -
  name: "s"
}-
-
}

```