OM2M Java API 0.6.0

About OM2M

OM2M provides an open source service platform for M2M interoperability based on the ETSI-M2M standard. OM2M follows a RESTful approach with open interfaces to enable developing services and applications independently of the underlying network. It proposes a modular architecture running on top of an OSGi layer, making it highly extensible via plugins. It supports multiple protocol bindings such as HTTP and CoAP. Various interworking proxies are provided to enable seamless communication with vendor-specific technologies such as Zigbee and Phidgets devices. (from http://www.eclipse.org/om2m/)

See more about OM2M Project : http://www.eclipse.org/om2m/.

About OM2M Java API

For a Java developer, develop solutions based on “Open M2M” server while enjoying all its features and services, is another concern in addition to business concerns. It is from this realization that the need for an API to communicate with the server “Open M2M” by object was born.

Implemented (user)functions in this version

Manage Applications

- Create an application.
- Delete an application.
- Retrieve an application.
- Retrieve all applications.
- Verify the existence of an application.

Manage containers

- Create a container.
- Delete a container.
- Retrieve a container.
- Retrieve all containers.
- Verify the existence of a container.

Manage Content Instances

- Create a Content Instance.
- Retrieve all Content Instances.

How to use this API

Befor any manipulation with this API, verify if your OM2M plateform is running well (http://wiki.eclipse.org/OM2M/REST_API)

Requirements

- Local or distante OM2M plateform : host, port and Basic Authorization header. See http://wiki.eclipse.org/OM2M/REST_API
- JAVA 1.7 or higher version
- If you wante to build OM2M-java-client-api from sources, you need Apaceh Maven 3. Else download the API from here.
- Include the API in your projet

Quick Start

> **Each piece of code of the examples that follow, can be inserted anywhere in your classes method. Do not forget the requirements :).**

Manage applications

How to create an application

```
Application myApp = new Application("MyAppID");

SearchStrings searchStrings = new SearchStrings();
searchStrings.getSearchString().add("keyWord1");
searchStrings.getSearchString().add("keyWord2");
myApp.setSearchStrings(searchStrings);

Om2mManagersFactorty.configure("http://127.0.0.1:8080", "Basic YWRtaW46YWRtaW4=");
Om2mManager<Application> om2mManager = Om2mManagersFactorty.getManager(Om2mManagersFactorty.APP_
MANAGER);
int httpCodeResp = om2mManager.create(myApp);
```

How to retreive an application

```
Om2mManagersFactorty.configure("http://127.0.0.1:8080", "Basic YWRtaW46YWRtaW4=");
Om2mManager<Application> om2mManager = Om2mManagersFactorty.getManager(Om2mManagersFactorty.APP_
MANAGER);

Application app = om2mManager.get("MyAppID");
```

How to retreive all applications

```
Om2mManagersFactorty.configure("http://127.0.0.1:8080", "Basic YWRtaW46YWRtaW4=");
Om2mManager<Application> om2mManager = Om2mManagersFactorty.getManager(Om2mManagersFactorty.APP_
MANAGER);
List<Application> apps = om2mManager.getAll(null);

System.out.println("List of application >");
for (Application application : apps) {
   System.out.println("app >>> " + application.getAppId());
}
```

How to delete an application

```
Om2mManagersFactorty.configure("http://127.0.0.1:8080", "Basic YWRtaW46YWRtaW4=");
Om2mManager<Application> om2mManager = Om2mManagersFactorty.getManager(Om2mManagersFactorty.APP_
MANAGER);

Application myApp = new Application("MyAppID");
int httpCodeResp = om2mManager.delete(myApp);
```

How to verify if an application exist

```
Om2mManagersFactorty.configure("http://127.0.0.1:8080", "Basic YWRtaW46YWRtaW4=");
Om2mManager<Application> om2mManager = Om2mManagersFactorty.getManager(Om2mManagersFactorty.APP_
MANAGER);

Application myApp = new Application("MyAppID");
boolean appExist = om2mManager.exist(myApp)
```

Manage containers

How to create a container

```
Container container = new Container("MyContainerId1");
// Specify the application in which is contained the "container".
Application myApp = new Application("MyAppID");
container.setApplication(myApp);

Om2mManagersFactorty.configure("http://127.0.0.1:8080", "Basic YWRtaW46YWRtaW4=");
Om2mManager<Container> containerManager = Om2mManagersFactorty.getManager(Om2mManagersFactorty.CON
TAINER_MANAGER);
int httpCodeResp = containerManager.create(container);
```

How to get a container

```
Container container = new Container("MyContainerId1");
// Specify the application in which is contained the "container".
Application myApp = new Application("MyAppID");
container.setApplication(myApp);

Om2mManagersFactorty.configure("http://127.0.0.1:8080", "Basic YWRtaW46YWRtaW4=");
Om2mManager<Container> containerManager = Om2mManagersFactorty.getManager(Om2mManagersFactorty.CON
TAINER_MANAGER);
Container containerResp = containerManager.get(container);
```

How to get all containers

```
Om2mManagersFactorty.configure("http://127.0.0.1:8080", "Basic YWRtaW46YWRtaW4=");
Om2mManager<Container> containerManager = Om2mManagersFactorty.getManager(Om2mManagersFactorty.CON
TAINER_MANAGER);
List<Container> containers = containerManager.getAll(new Application("MyApp" + APP_ID_TEST));

System.out.println("List of contentInstance >");
for (Container container : containers) {
    System.out.println("container> " + container.getId());
}
```

How to delete a container

```
Container container = new Container("MyContainerId1");
// Specify the application in which is contained the "container".
Application myApp = new Application("MyAppID");
container.setApplication(myApp);

Om2mManagersFactorty.configure("http://127.0.0.1:8080", "Basic YWRtaW46YWRtaW4=");
Om2mManager<Container> om2mManager = Om2mManagersFactorty.getManager(Om2mManagersFactorty.CONTA
INER_MANAGER);
int httpCodeResp = om2mManager.delete(container);
```

How to verify if a container exist

```
Container container = new Container("MyContainerId1");
// Specify the application in which is contained the "container".
Application myApp = new Application("MyAppID");
container.setApplication(myApp);
```

```
Om2mManagersFactorty.configure("http://127.0.0.1:8080", "Basic YWRtaW46YWRtaW4=");
Om2mManager<Container> containerManager = Om2mManagersFactorty.getManager(Om2mManagersFactorty.CON
TAINER_MANAGER);
boolean exist = containerManager.exist(container);
```

Manage ContentIncances

How to create a contentInstence

```
// parent container of the future contentInstance
Container container = new Container("MyContainerId2");
// Parent application of the container
Application myApp = new Application("MyAppID");
container.setApplication(myApp);

Om2mManagersFactorty.configure("http://127.0.0.1:8080", "Basic YWRtaW46YWRtaW4=");
ContentInstanceManager contentInstanceManager = Om2mManagersFactorty.getManager(Om2mManagersFactorty.
CONTENT_INSTANCE_MANAGER);

for (int i = 0; i < 10; i++) {
   ContentInstance contentInstance = new ContentInstance("contentInstanceId" + i);
   contentInstance.setContainer(container);
   Base64Binary base64Binary = new Base64Binary();
   base64Binary.setContentType("String");

   base64Binary.setValue(("dataInstance_" + container.getId() + "_" + myApp.getAppId() + "-" + i).toString().getBytes());
   contentInstance.setContent(base64Binary);

   int httpCodeResp = contentInstanceManager.create(contentInstance);
   System.out.println("httpCodeResp : " + httpCodeResp);
}
```

How to get all contentInstances

```
// parent container of the future contentInstance
Container container = new Container("MyContainerId2");
// Parent application of the container
Application myApp = new Application("MyAppID");
container.setApplication(myApp);

Om2mManagersFactorty.configure("http://127.0.0.1:8080", "Basic YWRtaW46YWRtaW4=");
ContentInstanceManager contentInstanceManager = Om2mManagersFactorty.getManager(Om2mManagersFactorty.
CONTENT_INSTANCE_MANAGER);

List<ContentInstance> contentInstances = contentInstanceManager.getAll(container);


System.out.println("List of contentInstance >");
for (ContentInstance ci : contentInstances) {
   System.out.println("CI> " + ci.getValueAsString() + " // " + ci.toString());
}
```

By Belili Fahem - belili.fahem@gmail.com - last modification : Sunday, 18. October 2015
/ @LinkedIn / @Twiter