

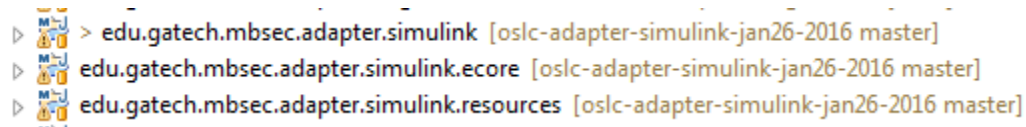
Source Code Documentation of OSLC Adapter

By Axel Reichwein, March 24, 2016

All implementations of OSLC adapters available at [github/Id4mbse](https://github.com/Id4mbse) share common features described in this document.

1. General Overview

An adapter implementation is located in 3 projects, in order to facilitate code maintenance, as shown below.



```
> edu.gatech.mbsec.adapter.simulink [oslc-adapter-simulink-jan26-2016 master]
> edu.gatech.mbsec.adapter.simulink.ecore [oslc-adapter-simulink-jan26-2016 master]
> edu.gatech.mbsec.adapter.simulink.resources [oslc-adapter-simulink-jan26-2016 master]
```

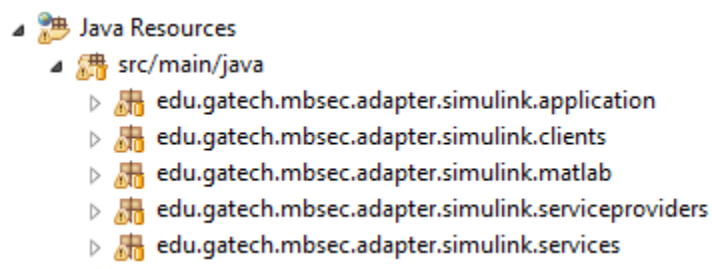
The 3 projects have following roles

- Project adapter.*: main adapter implementation
- Project adapter.*.ecore: metamodel of resources produced/consumed by adapter
- Project adapter.*.resources: OSLC-annotated Java resource classes used by clients of the OSLC adapter

2. Project adapter.*

a) Java code

The Java code is located under `src/main/java`



```
Java Resources
└─ src/main/java
   └─ edu.gatech.mbsec.adapter.simulink
      ├── application
      ├── clients
      ├── matlab
      ├── serviceproviders
      └── services
```

The Java source code includes code for:

- a) Reading/Writing information from/to the target application. The main Java class responsible for interacting with target application is called ***Manager** located in package ***edu.gatech.mbsec.adapter.*.application***.
- b) Creating OSLC resources for structuring the RESTful web services offered by the adapter (e.g. OSLC Service Provider Catalog, OSLC Service Provider). The main Java class for registering all services of each OSLC service provider is ***ServiceProviderFactory** located in package ***edu.gatech.mbsec.adapter.*.serviceproviders***. The **ServiceProviderFactory** is called by the `initServiceProvidersFromProjects()` method of the **ServiceProviderCatalogSingleton** class. Note: There is a OSLC service provider for each project/container in the target application.
- c) RESTful web services to read/write information from/to the target application, and RESTful web services publishing default OSLC resources such as an OSLC Service Provider Catalog and OSLC Service provider resources. The servlets are located in the package ***edu.gatech.mbsec.adapter.*.services***.
- d) Code to call the API of the target application if API of the target application is not in Java
- e) Examples of client code performing for instance HTTP GET or HTTP POST, or HTTP PUT calls to the OSLC adapter

b) Documentation Folder

This folder contains Installation instructions

c) *models folder

This folder contains example models of target application that were used for test purposes.

d) Web Resources

The web resources are located under `src/main/webapp`

This folder contains web resources including:

- JSP templates
- Images and css files for HTML files
- Deployment descriptor file (web.xml)

3. Project adapter.*.resources

This project contains Java code

- **OSLCJavaClassesGenerator** class which can automatically generate OSLC-annotated Java classes based on a metamodel defined in Ecore
- Generated OSLC-annotated Java classes
- The **package.info** class is specific to OSLC4J and is used to register namespace prefixes which are used to describe OSLC resources. This class for example includes several namespace prefixes used to describe OSLC resources representing SysML concepts

4. Project adapter.*.ecore

The metamodel of the resources produced/consumed by the adapter are defined in Ecore based on which Java classes with OSLC annotations are generated automatically. This allows to efficiently and consistently adapt or extend the SysML concepts which are being exposed by the OSLC adapter.