**Instructions to install and run  
the OSLC Simulink Adapter**

**(for Matlab R2013b, 2014a, 2014b)**

By Axel Reichwein ([axel.reichwein@koneksys.com](mailto:axel.reichwein@koneksys.com)) January 25, 2016

# Installing OSLC4J

Follow the instructions in the document named “Instructions to install Eclipse Lyo”. The document also contains instructions on how to use a proxy server with Maven and Eclipse.

# Download, import, and build edu.gatech.mbsec.subversion.client and adapter.subversion projects

Clone **subversion-client** repository

1. Open the Git Repositories View (Window -> Show View -> type “Git Repositories” in the search field)
2. Click on the Clone Repository icon 
3. In the URI field, paste the following URL: <https://github.com/ld4mbse/subversion-client.git>
4. The Host and Repository fields will autofill. Enter your credentials in the Username and Password fields.
5. Click Next, only select the **master** branch
6. Click Next until Finish.

Clone **edu.gatech.mbsec.adapter.subversion** repository

1. Open the Git Repositories View (Window -> Show View -> type “Git Repositories” in the search field)
2. Click on the Clone Repository icon 
3. In the URI field, paste the following URL: <https://github.com/ld4mbse/oslc-adapter-subversion.git>
4. The Host and Repository fields will autofill. Enter your credentials in the Username and Password fields.
5. Click Next, only select the **master** branch
6. Click Next until Finish.

Import **edu.gatech.mbsec.subversion.client project** into Eclipse workspace and build project

1. In the Git repositories view, right-click **edu.gatech.mbsec.subversion.client** and select “Import Projects”. Click Next until Finish
2. The **edu.gatech.mbsec.subversion.client** project is in the Eclipse workspace
3. In Eclipse, open the Project Explorer view. (Window → Show View → Project Explorer)
4. Expand the **edu.gatech.mbsec.subversion.client** project
5. Right click pom.xml -> Run As -> Maven clean
6. Right click pom.xml -> Run As -> Maven install

Import **edu.gatech.mbsec.adapter.subversion** into Eclipse workspace and build project

1. In the Git repositories view, right-click **edu.gatech.mbsec.adapter.subversion** and select “Import Projects”. Click Next until Finish
2. The **edu.gatech.mbsec.adapter.subversion** project is in the Eclipse workspace
3. In Eclipse, open the Project Explorer view. (Window → Show View → Project Explorer)
4. Expand the **edu.gatech.mbsec.adapter.subversion** project
5. Right click pom.xml -> Run As -> Maven clean
6. Right click pom.xml -> Run As -> Maven install

# Downloading edu.gatech.mbsec.adapter.simulink repository

1. Open the Git Repositories View (Window -> Show View -> type “Git Repositories” in the search field)
2. Click on the Clone Repository icon 
3. In the URI field, paste the following URL: <https://github.com/ld4mbse/oslc-adapter-simulink.git>
4. The Host and Repository fields will autofill. Enter your credentials in the Username and Password fields.
5. Click Next, only select the **master** branch
6. Click Next until Finish.

# Importing projects into the Eclipse workspace

1. In the Git repositories view, right-click edu.gatech.mbsec.adapter.simulink and select “Import Projects”. Click Next until Finish
2. The 3 projects are in the Eclipse workspace

# Building the edu.gatech.mbsec.adapter.simulink projects

1. In Eclipse, open the Project Explorer view. (Window → Show View → Project Explorer)
2. Expand the **edu.gatech.mbsec.adapter.simulink.ecore** project
3. Right click pom.xml -> Run As -> Maven clean
4. Right click pom.xml -> Run As -> Maven install
5. Expand the **edu.gatech.mbsec.adapter.simulink.resources** project
6. Right click pom.xml -> Run As -> Maven clean
7. Right click pom.xml -> Run As -> Maven install
8. Expand the **edu.gatech.mbsec.adapter.simulink**  project
9. Right click pom.xml -> Run As -> Maven clean
10. Right click pom.xml -> Run As -> Maven install

If there is no error mark next to any project, you can skip the next steps.

1. If there is a red error mark next to any project, select the project. Right-click->Maven->Update Project… and click OK
2. Make sure that the Eclipse projects displayed in the project explorer view do not contain any error icons displayed next to the project names as for example displayed below.



If a project has an error icon, then select the project and open its properties view (Project->right click->Properties). Under the Projects Facet tab, make sure that 1.8 is selected.

If a project still shows an error, then change its **JDK compliance** to 1.8. Select the project, right-click -> Properties. Select Java Compiler and select **1.8** in the drop down menu next to the JDK compliance setting as highlighted below.

# Manual configuration

TheOSLC Simulink adapter currently supports the retrieval of Simulink models within a specific directory. The location of the directory containing the Simulink projects is currently hard coded in a configuration file. Several Simulink models are already located in the ***simulinkmodels*** folder in the **edu.gatech.mbsec.adapter.simulink** project.

TheOSLC Simulink adapter currently supports the retrieval of Simulink models within

1. **a specific local directory – “Local mode”**

Specify the **location of the folder containing Simulink** **models** which will be considered by the OSLC Simulink adapter in the config.properties file under *edu.gatech.mbsec.adapter.simulink/configuration*. As an example displayed below, the location of the folder containing Simulink models for the OSLC adapter is specified to **C:/Users/…/git/oslc4jsimulink/edu.gatech.mbsec.adapter.simulink/Simulink Models/**

Note: The file path can contain backslashes

**Warning**: Do not put quotes around the file path!

1. **or from a subversion repository – “SVN Repository mode”**

Set the value of **syncWithSvnRepo** to **true**

Specify the **Subversion repository URL containing Simulink models** which will be considered by the OSLC Simulink adapter in the config.properties file under *edu.gatech.mbsec.adapter.simulink/configuration*. As an example displayed below, the **Subversion repository URL** is specified to be **https://mysvnrepos.com/svn/simulinkrepository/**

**Warning**: Do not put quotes around the file path!

Set the time period in seconds at which the adapter will poll the Subversion repository for updates. Example: **delayInSecondsBetweenDataRefresh** = none **or**

**delayInSecondsBetweenDataRefresh** = 90

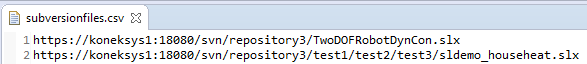
Specify your Subversion credentials through the **svnUserName** and **svnPassword** fields

Specify the **location of the folder containing Simulink models** where the Subversion files will be saved locally in config.properties file under *edu.gatech.mbsec.adapter.simulink/configuration*. As an example displayed below, the location of the folder containing Simulink models for the OSLC adapter is specified to **C:/Users/…/git/oslc4jsimulink/edu.gatech.mbsec.adapter.simulink/localworkingdirs**

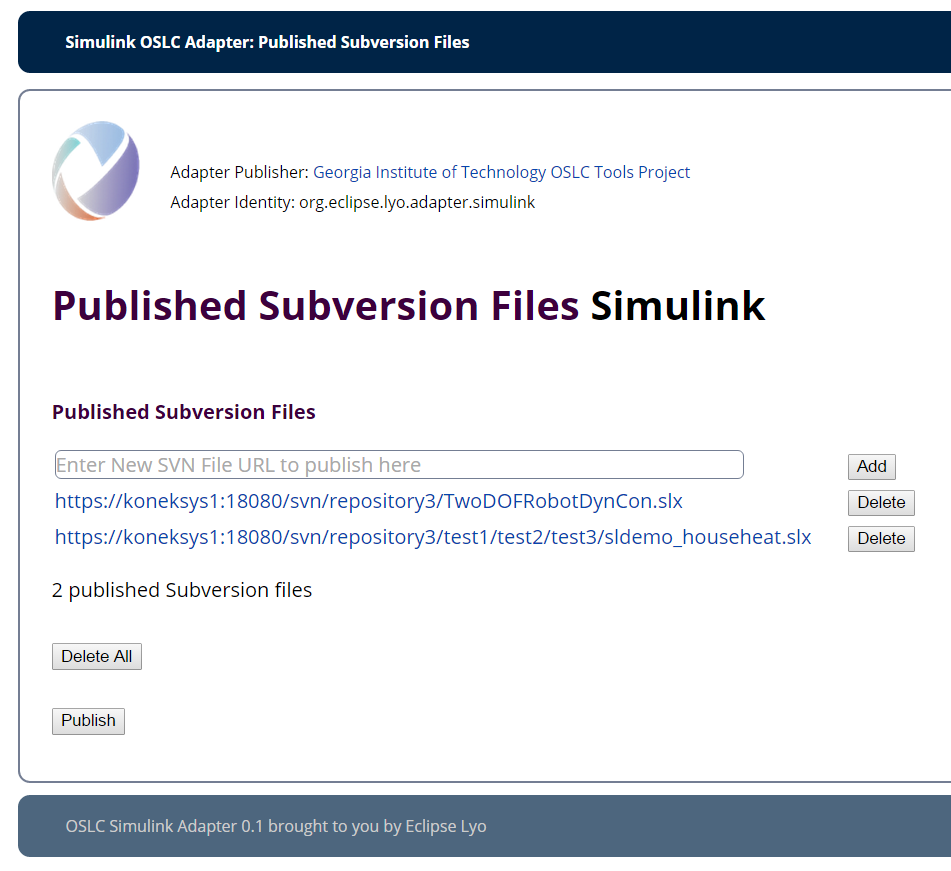
1. **or from individual Subversion-hosted files - “Individual SVN files mode”**

Set the value of **useIndividualSubversionFiles** to **true**

1. Specify the **Subversion file URLs representing Simulink models** which will be published by the adapter at startup in the subversionfiles.csv file under *edu.gatech.mbsec.adapter.simulink/configuration*. As an example displayed below, the **Subversion file URLs** are specified to be



1. **During adapter runtime, you can change the Subversion files** to be published through the web app **at** [**http://localhost:8080/oslc4jsimulink/services/svnfilepublisher**](http://localhost:8080/oslc4jsimulink/services/svnfilepublisher)



By clicking on **Publish**, the adapter will retrieve the latest version of the Subversion files and publish them

1. Specify your Subversion credentials through the **svnUserName** and **svnPassword** fields
2. Specify the **location of the folder containing Simulink models** where the Subversion files will be saved locally in config.properties file under *edu.gatech.mbsec.adapter.simulink/configuration*. As an example displayed below, the location of the folder containing Simulink models for the OSLC adapter is specified to **C:/Users/…/git/oslc4jsimulink/edu.gatech.mbsec.adapter.simulink/localworkingdirs**

The contents of this folder will be deleted when adapter starts in individualSubversionFile mode.

**Warning**: Do not choose as local Subversion file storage the same folder as the one containing all sample Simulink models, nor the one containing the local Simulink models models without Subversion info.

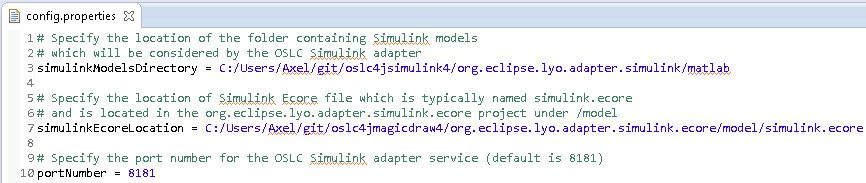
Several example Simulink models are located in the s***imulinkmodels*** folder in the **edu.gatech.mbsec.adapter.simulink** project. Following steps are also necessary to configure the adapter:

1. Specify the **location of Simulink Ecore file** in the config.properties file under *edu.gatech.mbsec.adapter.simulink/configuration*. The location of the Simulink ecore file named simulink.ecore is in the **edu.gatech.mbsec.adapter.simulink.ecore** project under /model/simulink.ecore. As an example displayed below, the location of the simulink.ecore file is specified to **C:/Users/…/git/oslc4jsimulink/edu.gatech.mbsec.adapter.simulink.ecore /model/simulink.ecore**

Note: The file path can contain backslashes

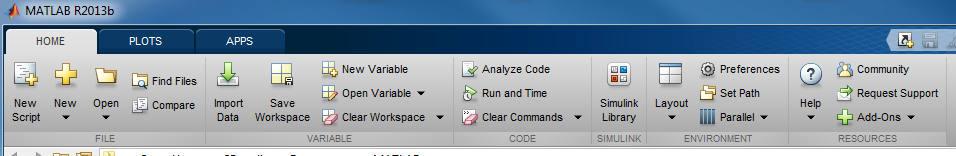
**Warning**: Do not put quotes around the file path and add nothing at the end!

1. Specify the **port number** of the OSLC Simulink adapter service of in the config.properties file under *edu.gatech.mbsec.adapter.simulink/configuration*. By default, port 8181 will be used. As an example displayed below, the port number is set to **8181**.



# Setting up the Matlab workspace

1. Launch Matlab
2. From the File menu, select *Set Path…(*if your Matlab version has a ribbon-based user interface such as inR2013, choose *Set Path* in the *Home* ribbon in the *Environment* section*)*



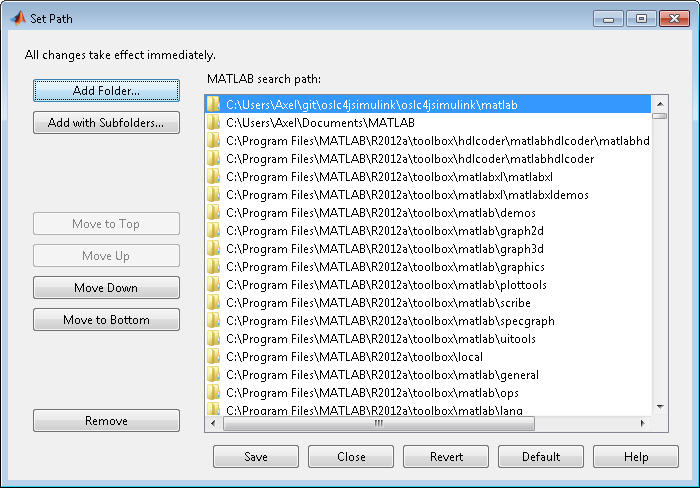
1. Use the Add Folder… command to add the ***matlab*** folder of the *oslc4jsimulink* project to the Matlab search path as shown below based on the location of your local git repository

Use the **Add with Subfolders…** command to add the **folder** in the *oslc4jsimulink* project **containing Simulink models, or local working copies of Subversion files**, to the Matlab search path. For example, based on the settings in Step #5, the concerned folder would have as path

* + In “local mode”: “**C:/Users/…/git/oslc4jsimulink/edu.gatech.mbsec.adapter.simulink/Simulink Models/**”
  + In “SVN repository mode” and “Individual SVN files mode”:

“**C:/Users/…/git/oslc4jsimulink/edu.gatech.mbsec.adapter.simulink/localworkingdirs**”

1. Click *Save* and then *Close*



# Installing Apache Tomcat

**The Steps #7 to #10 are only necessary if you want to use a specific Tomcat instance instead of the Tomcat instance embedded in Eclipse and downloaded by the Maven Tomcat plugin.** Steps #7 to #10 are **only** necessary if you want:

* to deploy the Simulink adapter on a specific Tomcat instance (possibly with specific configurations)
* to deploy the Simulink adapter such that it accepts HTTP PUT requests, necessary for updating Simulink parameters

1. Download Tomcat 8 by going to this page:

<https://tomcat.apache.org/download-80.cgi>

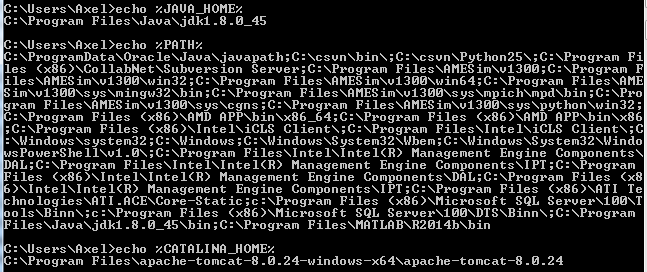
1. Download the zip distribution for your operating system. Note: do not use the windows installer as it doesn’t install all Tomcat scripts.
2. Unzip the Tomcat 8 distribution in a folder where your user account has read/write permission. Note: Windows disables direct file access to programs folder for normal users per default. Note: Installation of XAMPP can mess up previous installations of Tomcat and may need to be removed
3. Make sure that the /bin folder in your Tomcat installation directory contains the catalina.bat batch file

# Configuring Java for Tomcat

1. Make sure that you have JDK 8 installed on your machine. OSLC Adapters are now currently being build with Java 8 compilers. So Tomcat should also run with Java 8.
2. On Windows, verify your installed Java version by typing in the command prompt **java –version**



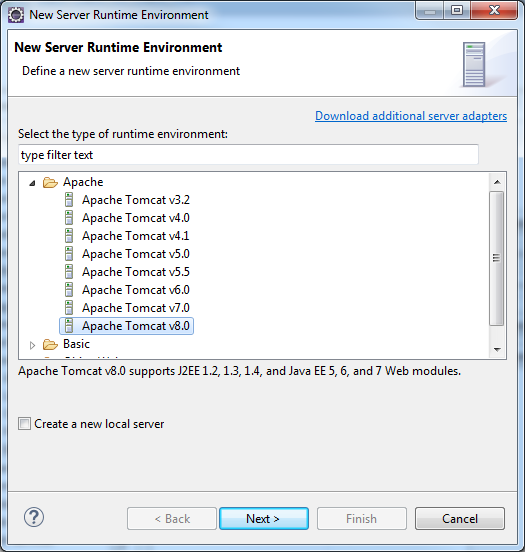
1. Test that environment variables JAVA\_HOME and PATH respectively point to JDK and JDK/bin. Verify this on Windows by typing in the command prompt **echo %JAVA\_HOME%** and **echo %PATH%**. If necessary set the envornoment variable in the command prompt using the **set** command (Example: **set** variable=string). After having set the environment variables, open a new ommand prompt to verify the values of the environment variables.



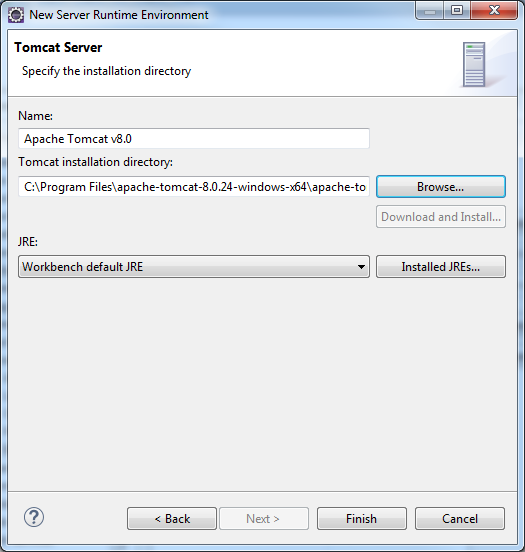
1. Make sure that JAVA\_HOME and JRE\_HOME both point to the same Java version, for example Java 8.
2. Make sure that CATALINA\_HOME points to the correct installation directory of your Tomcat distribution.

# Adding Server Runtime Environment in Eclipse

1. In Eclipse. Open **Window -> Preferences -> Server -> Runtime Environments** to create a Tomcat installed runtime.
2. Click on **Add...** to open the New Server Runtime dialog,
3. From the drop down menu, select Tomcat 8.0 as shown below. Click Next.



1. Enter the **Tomcat 8.0 installation directory** (not the Apache installation directory!) as highlighted below.
2. Click on Finish.



# Enabling PUT on Apache Tomcat

Tomcat by default is not enabled for HTTP PUT command. But, it can easily be configured to support it.

1. In your Apache Tomcat 8 installation directory, open **/conf/web.xml**
2. Add the ***readonly* init param** to the web.xml file as shown below and save the file

**<init-param>**

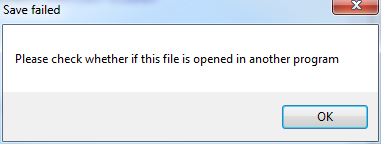
**<param-name>readonly</param-name>**

**<param-value>false</param-value>**

**</init-param>**

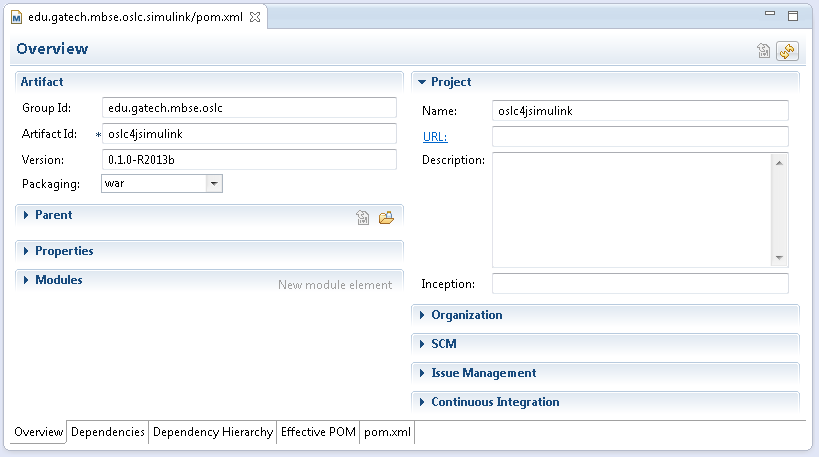


**Note**: If you get the warning shown below while trying to save the file, then copy the web.xml file into another location, modify it, and then replace the original web.xml file by the modified web.xml file.



# Setting the Apache Tomcat server port

1. By default, the OSLC Simulink adapter service will run on port 8181. Change the port of the oslc4jsimulink service only if you need to avoid a conflict with another service already running on port 8181. Skip the next steps if you do not need to change the port.
2. In Eclipse, open the Project Explorer view. (Window → Show View → Project Explorer)
3. Expand the **edu.gatech.mbsec.adapter.simulink** project
4. Select and open the maven **pom.xml** file through double-click
5. The pom.xml file contains several tabs. By default, the overview tab will be displayed. The various available tabs are displayed at the bottom of the editor window. Click on the **pom.xml tab** of the pom.xml file as highlighted below.



1. In the pom.xml tab of the pom.xml file, specify the **port of the OSLC Simulink adapter service** in theMaventomcat plugin configuration found at the **bottom of the pom.xml** tab of the pom.xml file as highlighted below. Enter the port number in the configuration section as shown below.



# Installing the Chrome/Firefox Postman plugin (or any REST client)

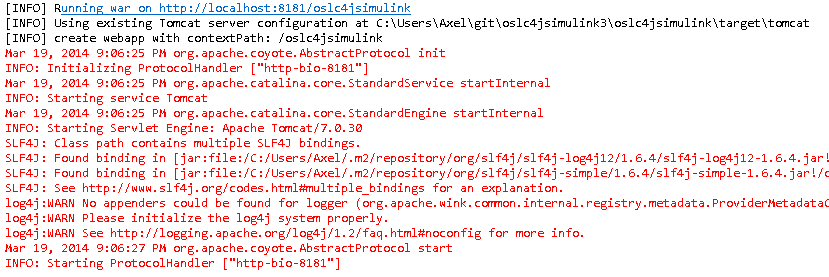
1. For Google Chrome, add the Postman REST client to your browser: <https://chrome.google.com/webstore/detail/postman-rest-client/fdmmgilgnpjigdojojpjoooidkmcomcm?hl=en>
2. And the Postman launcher: <https://chrome.google.com/webstore/detail/postman-launcher/igofndmniooofoabmmpfonmdnhgchoka?hl=en>

# Launching oslc4jsimulink (OSLC Simulink Adapter)

Select the oslc4jsimulink launch configuration (Run -> Run Configurations… and select in the Maven build category the launch configuration named **oslc adapter for simulink**) and click Run.

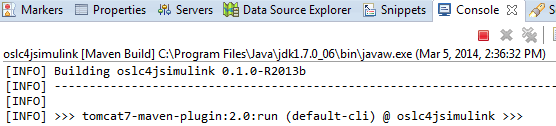
In the console window, several logging related exceptions will appear (SLF4J and log4j). This is not critical.

The OSLC Simulink adapter is running and following statements can be seen in the Eclipse Console windows displayed below

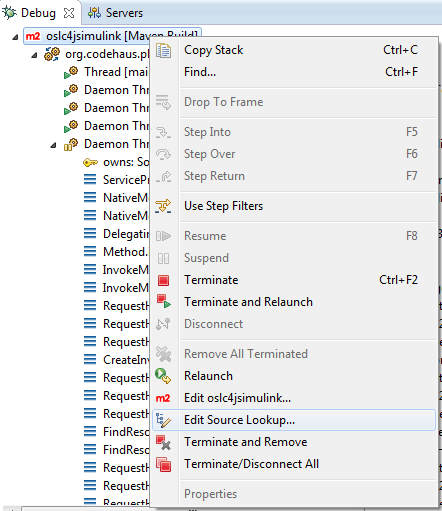


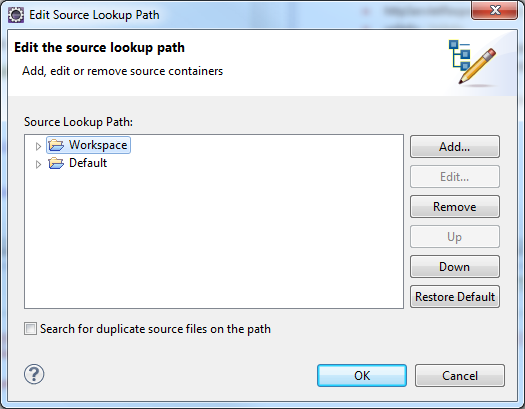
**Warning**: If the OSLC Simulink adapter service fails to launch due to a **java.net.BindException**, a different port for the OSLC Simulink adapter needs to be used since there is a conflict with another service using the same port. By default, the OSLC Simulink adapter uses port 8181. A **java.net.BindException** means that a different service is already using this port. Go back to Steps #5 and #10 to change the port number.

**Note**: In order to stop a running oslc4jsimulink web application, click Terminate in the Console window, as shown below, or in the toolbar of the debug perspective.



**Note**: If you launch the Maven launch configuration (OSLC Simulink adapter) in **debug mode**, and do not see the Java code when the application hits a breakpoint, then you need to add the Eclipse workspace to the source lookup path. In the Debug view, right click on the running thread (in threads tab), or on the application as shown below and select **Edit Source Lookup**, and add the workspace. Re-launch the Maven launch configuration and the code should be visible in the editor when the application hits a breakpoint.

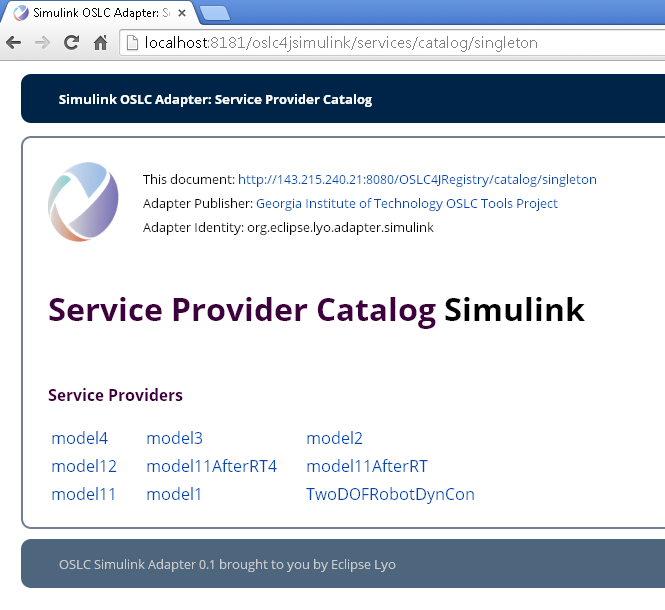


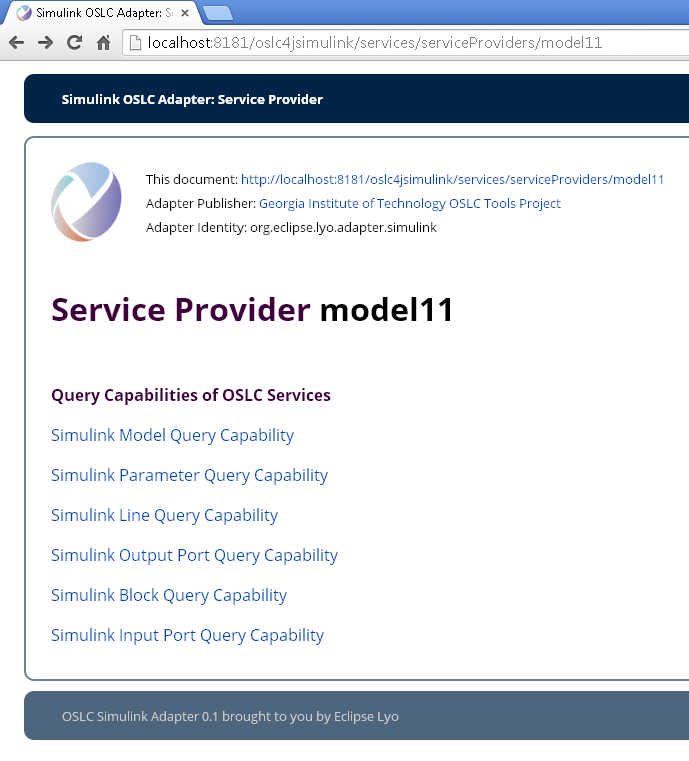


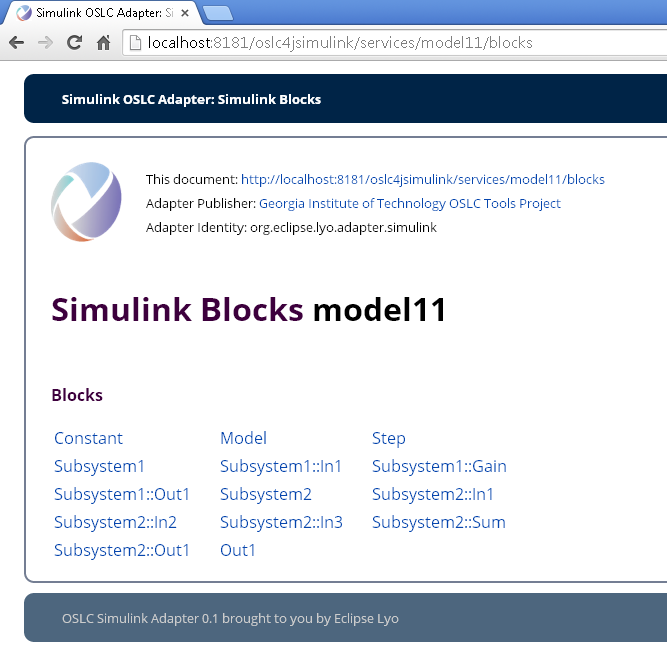
# Testing the OSLC Simulink Adapter

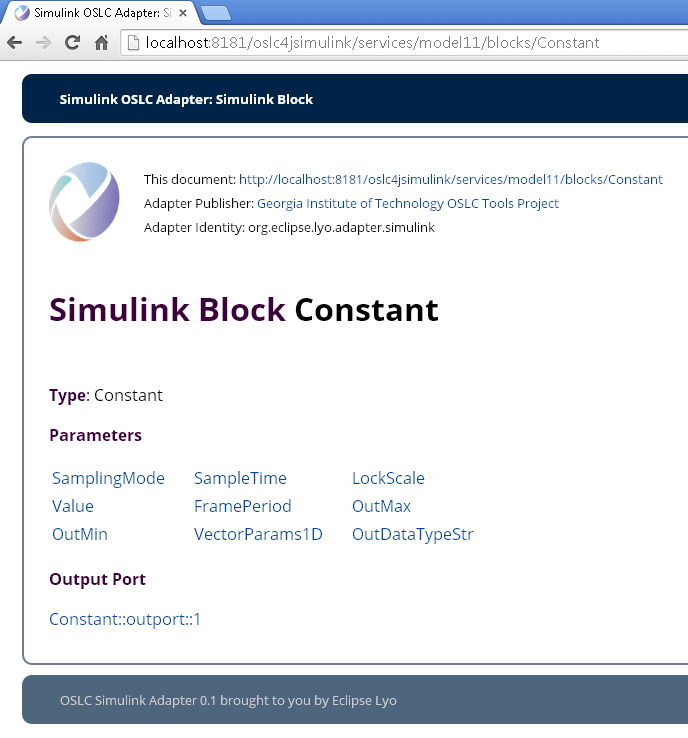
**Testing the retrieval of OSLC resources in HTML**

1. Launch Google Chrome
2. In the URL field, type for test purposes: <http://localhost:8181/oslc4jsimulink/services/catalog/singleton>. This will send a HTTP GET request to retrieve the HTML representation of the Simulink Service Provider Catalog. This will launch a Matlab command window which will close automatically. The Matlab command window may display warnings if it is an older version than R2013b.
3. You will then see an HTML page showing you the list of Service Providers. **You can browse from the Service Providers (e.g. for model11) to the Services and ultimately to the OSLC Simulink resources.**

l 



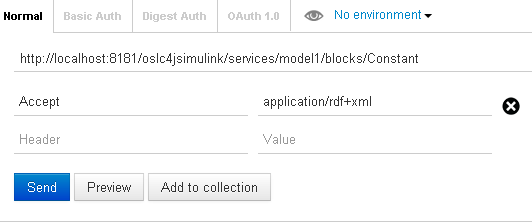






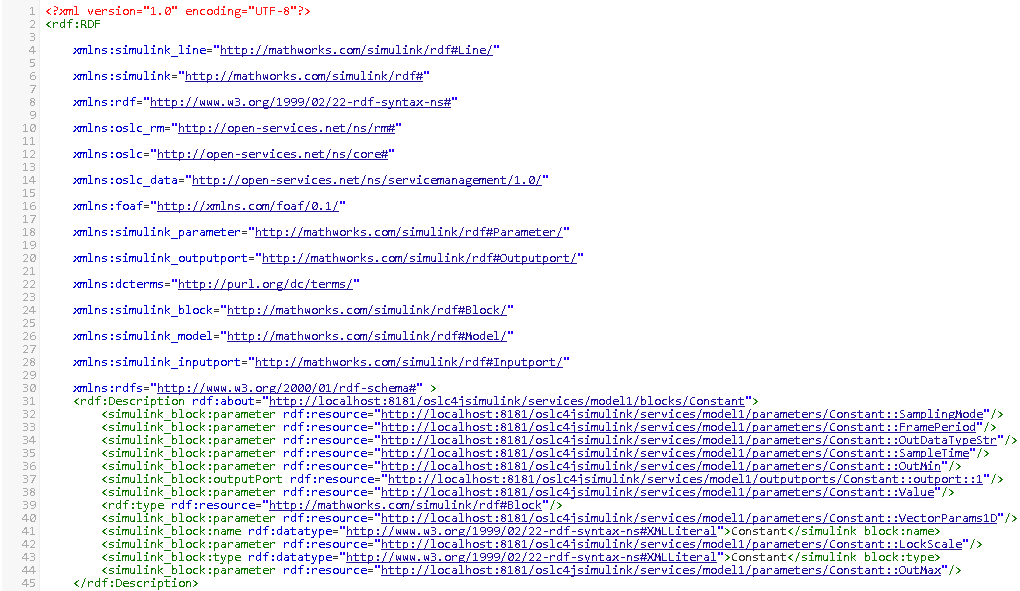
**Testing the retrieval of OSLC resources in RDF**

1. Click on the Postman icon at the top right of the Chrome browser. A new tab will open.
2. In the URL field, type for test purposes: <http://localhost:8181/oslc4jsimulink/services/model1/blocks/Constant>.
3. Click on the *Headers* field to the right of the URL field
4. Enter *Accept* in the Header field and *application/rdf+xml* in the value field as shown below
5. Click Send



1. This will send a HTTP GET request to retrieve the RDF/XML representation of the Simulink block named “Constant”.

The Postman REST client will display the RDF/XML representation of the Simulink block named “constant”. Other HTTP requests to retrieve other SysML elements can be sent.



# Testing the OSLC Simulink Adapter through example Simulink model

The edu.gatech.mbsec.adapter.simulink project contains example Simulink models containing different types of Simulink elements. The example models are located in the folder named **matlab**.

**model11** contains blocks, subsystems, model reference blocks, ports, lines, and lines with multiple target ports

