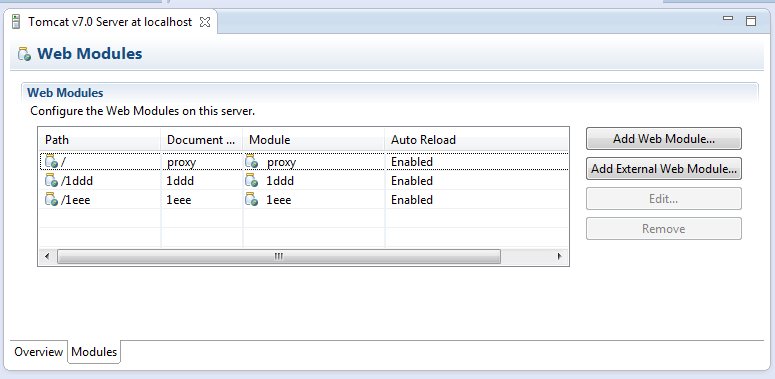
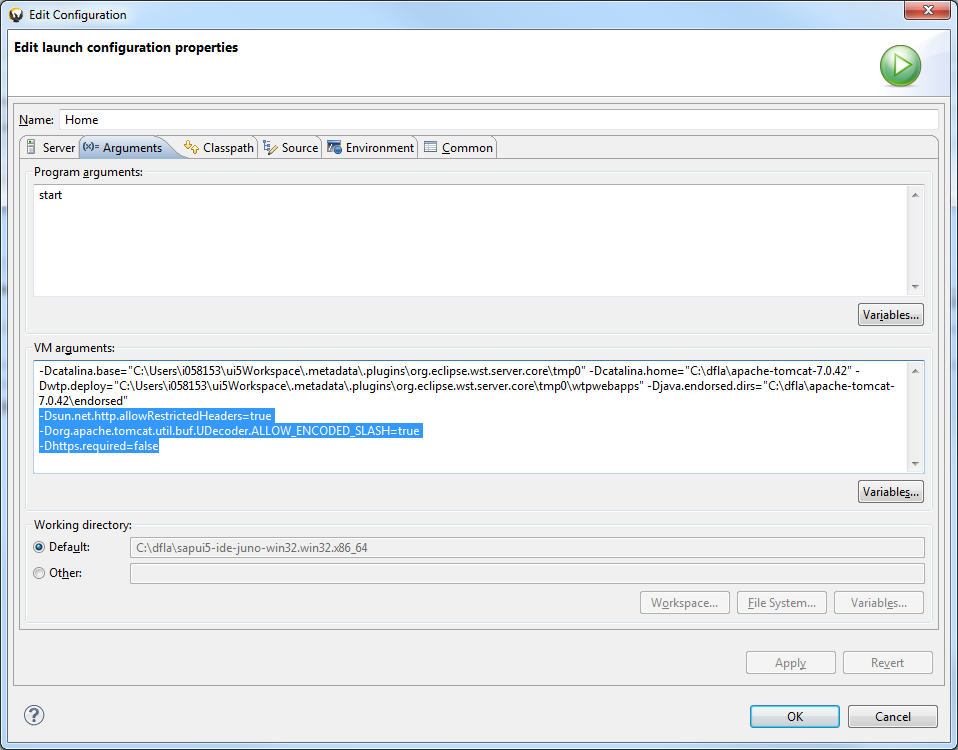
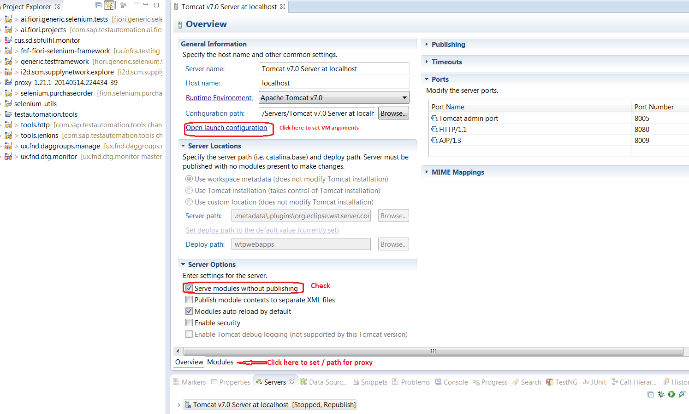
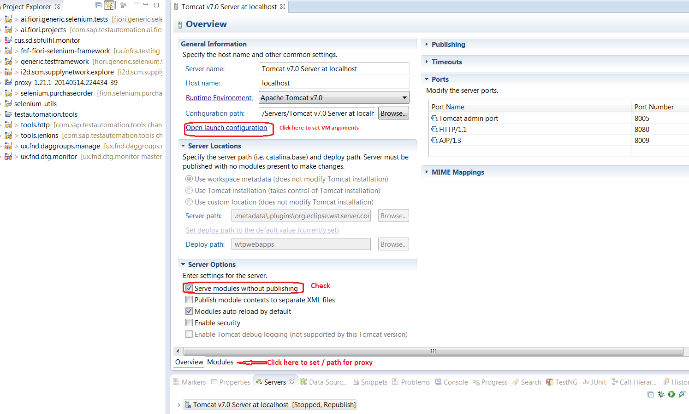
1. Download the war file from [here](https://nexus.wdf.sap.corp:8443/nexus/service/local/artifact/maven/redirect?r=deploy.releases&g=com.sap.ui2.srvc&a=proxy&v=1.26.3&e=war)and import in your Eclipse through File->Import-> Web->War File.
2. Afterwards extract the downloaded war file in your windows explorer and copy the folder **WEB-INF\classes**to the  **WebContent\WEB-INF\classes** folder of the proxy Eclipse project. This is necessary because Eclipse does not extract the files during import.
3. Deploy the proxy servlet on your tomcat instance under the root path '/' as shown below: press "Add Web Module..." button and select the downloaded war-file. Ensure that you enter "/" as path. The result has to look like this (the first row of the table is relevant!!!):   
     
    
4. Change the launch configuration of your local Tomcat for the proxy servlet to work:  You have to set two system properties. Open the Tomcat server configuration where you just added the proxy module and switch back to the Overview tab. Here click the link to open the launch configuration. In the popup select the tab "Arguments" and add the following VM arguments. Afterwards select "OK".  
   -Dsun.net.http.allowRestrictedHeaders=true  
   -Dorg.apache.tomcat.util.buf.UDecoder.ALLOW\_ENCODED\_SLASH=true  
   If you want to work with tomcat in non HTTPS mode (most likely the case), you have to add the following parameter  
   -Dhttps.required=false  
    
5. Ensure that the "Serve Modules without Publishing" is checked.   
      
   
6. Maintain the file WebContent\WEB-INF\forward.properties in your proxy servlet project and set the gateway server to the one you are using.  
   Example:

|  |
| --- |
| gm6\_https = https://ldcigm6.wdf.sap.corp:44333    ...    gateway = ${gm6\_https} |

If you need more information about the proxy servlet please refer to the ProxyServlet wiki page  [Proxy Servlet](https://wiki.wdf.sap.corp/wiki/display/UICEI/Proxy+Servlet)

#### 6. Ensure that the "Serve Modules without Publishing" flag is checked on your tomcat!



#----------------------------------------------------------------------------

# sf

#----------------------------------------------------------------------------

#internet\_proxy = proxy:8080

sfodata=${sf\_https}

sf\_https=https://apisalesdemo4.successfactors.com/odata/v2

sfodata.pattern = /sfserver

sfodata.target = ${sf\_https}

sfodata.options = nopreservehost

#sfodata.proxy = ${internet\_proxy}