Example with CSRF and validation protection

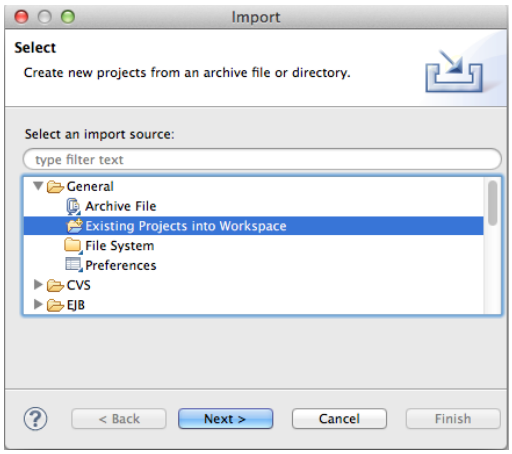
1. Presentation

This second example is a very simple shop. Users can register and login to a simple shop. They can buy products by clicking on the 'buy' button. The user needs to choose the quantity of products he want and then to enter his credit card information. To complete the order, the credit card number needs to be valid.

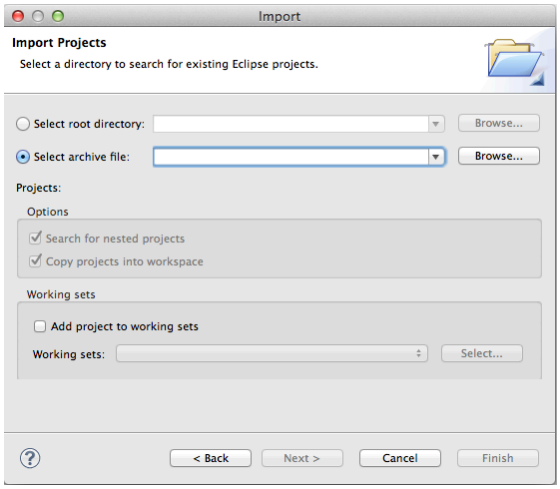
2. 7 steps installation

1. Download the auction_example eclipse project in the 'example' section.

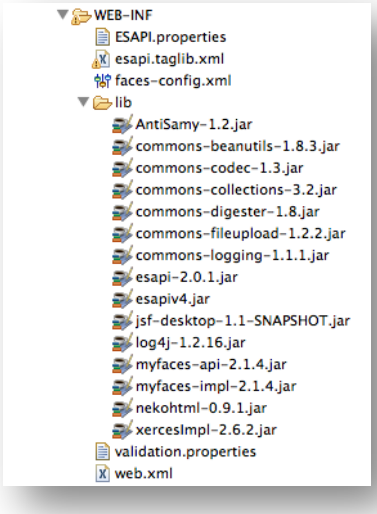
2. Import the project in your Eclipse environment by selecting File -> import.

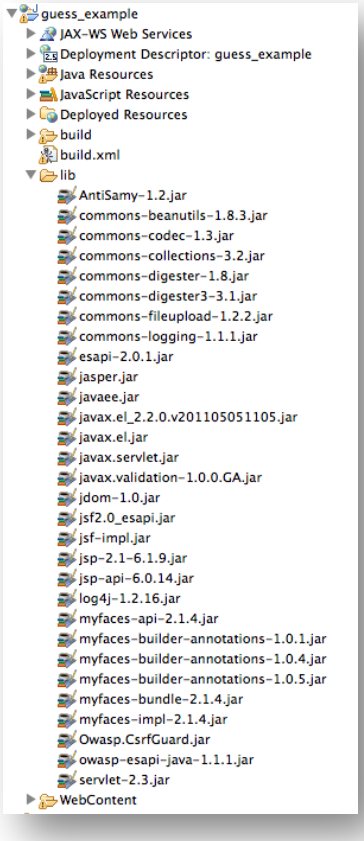


Choose 'Existing Projects into Workspace', then select the zip downloaded and press 'Finish'.



3. All libraries specified in the 'librairies' section of the OWASP EJSF Project need to be present in the build path. Some librairies need to be copied in the lib directory located in the WEB-INF folder of the project. The libraries are normally already there.





4. Download the esapi.jar file in the 'installation' section and you need to copy this file in the WEB-INF/lib directory.

5. The next step is necessary for the login works! The users.txt file is available in the 'Downloads' section and need to be copied in this path:

- Mac and Linux users: \users\your_name\esapi\
- Windows users: C:\Documents and Settings\your_user\.esapi\

6. copy all config files int the WEB-INF folder (ESAPI.properties, esapi.taglib.xml, faces-config.xml, users.txt, validation.properties)

7. in the web.xml file you need to add some code (the web.xml is available on the 'Download' section too) :

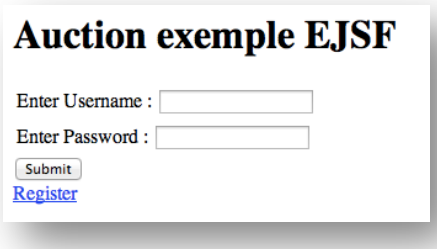
```
<filter>
   <filter-name>CsrfFilter</filter-name>
   <filter-class>ch.security4web.esapi.csrf.OwaspCSRFSessionFilter</filter-class>
  </filter>
  <filter>
   <filter-name>AuthorizationFilter</filter-name>
   <filter-class>ch.security4web.esapi.authentication.EsapiAuthorizationFilter</filter-class>
  </filter>
<filter>
```

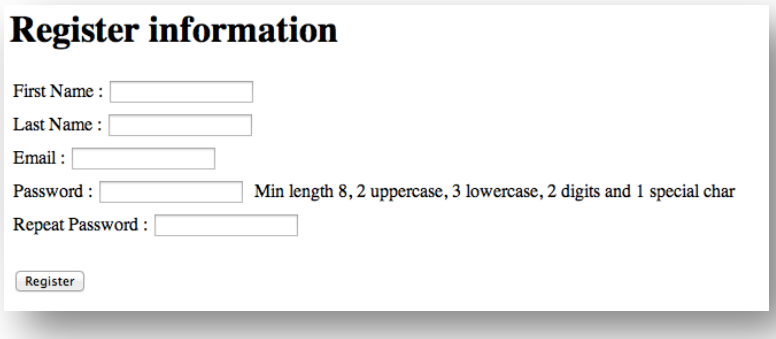
```
<context-param>
   <param-name>facelets.LIBRARIES</param-name>
   <param-value>/WEB-INF/esapi.taglib.xml</param-value>
  </context-param>
  <listener>
   <description>OwaspCSRFSessionListener</description>
   <listener-class>ch.security4web.esapi.csrf.OwaspCSRFSessionListener</listener-class>
  </listener>
</context-param>
```

3. A simple example

3.1 Register

First you need to register. You need to enter your first name; this is your username, your last name, a valid address email and a secure password (Minimum 8 chars length with two uppercase chars and three lowercase chars with two digits and one special char).
If all informations are valid, user will be created and added to the users.txt file.

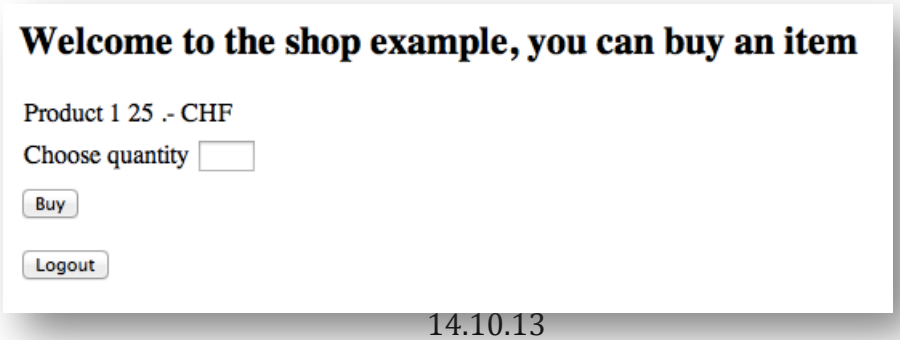




3.2 Login and buy product

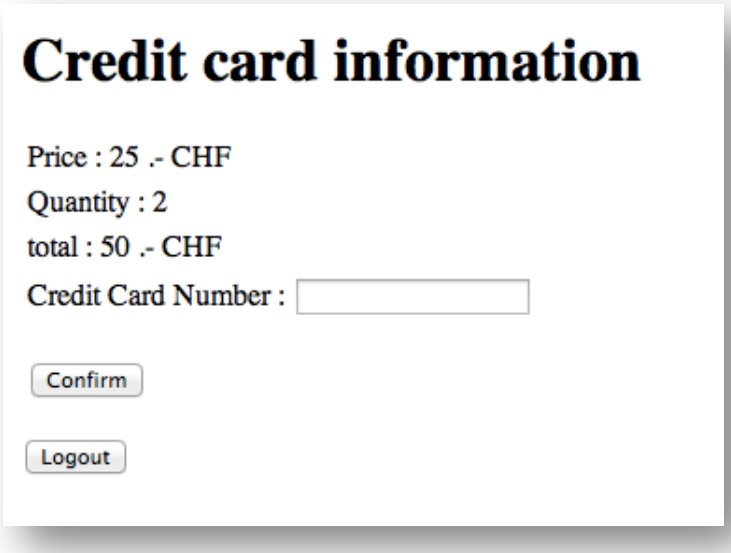
Then you need to login in the login page with your first name inserted and with your password.

You can buy the product 1 in this page, you need to enter the quantity you want and then press the ‘buy’ button.



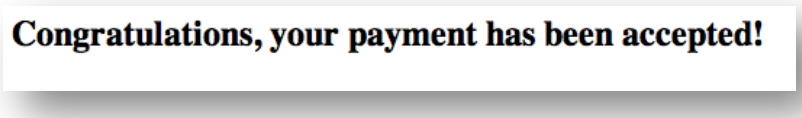
3.3 Pay the order

In the next step, you have information about your order and the total to pay. You must insert a valid credit card to confirm your order and pay it.



By inserting not valid information, you will receive a message and you need to correct it. In this example, 21231221 is not a valid credit card number.

By inserting a correct credit card number like: 4539998965627588 you will redirect to a congratulations page.



4. Protection

The first protection feature is CSRF. When you log in in the shop site, a sessionID will be assigned. If you look at the bar address you have a 'jsessionid' token defined.

http://localhost:8080/auction_example/faces/login.xhtml;jsessionid=1E4CB8CACC8A141C8657BF60B79DE701

The second feature is the input validation. In the register page, all input are validate, simple string, email validator and password validator. The quantity is also validated in the auction page when the user needs to enter the quantity he wants. Finally, the credit card number is validated too.