

Configurazione Portale:

Installazione JAVA JDK 7:

Installare jdk 7 dal sito di Oracle come da guida al seguente link:

<http://www.webupd8.org/2012/01/install-oracle-java-jdk-7-in-ubuntu-via.html>

```
sudo add-apt-repository ppa:webupd8team/java
sudo apt-get update
sudo apt-get install oracle-java7-installer
```

Accettare le licenze oracle.

Una volta installato verificarne il risultato e la versione con il comando:

```
java -version
```

Prendere la path di dove è stata effettuata l'installazione java:

solitamente sotto jvm folder ...esempio: `/usr/lib/jvm/java-7-oracle`

Copy the path from your preferred installation and then edit the file `/etc/environment`:

Using your text editor like nano or vi:

```
vi nano /etc/environment
```

In this file, add the following line (replacing YOUR_PATH by the just copied path):

```
JAVA_HOME="YOUR_PATH"
```

That should be enough to set the environment variable. Now reload this file:

```
source /etc/environment
```

Test it by executing:

```
echo $JAVA_HOME
```

If it returns the just set path, the environment variable has been set successfully. If it doesn't, please make sure you followed all steps correctly.

Configurare JCE_Policy

- Download the version that matches your installed JVM E.g. [UnlimitedJCEPolicyJDK7.zip](http://www.oracle.com/technetwork/java/javase/downloads/jce-7-download-432124.html) from the address: <http://www.oracle.com/technetwork/java/javase/downloads/jce-7-download-432124.html>
- Unzip the downloaded zip
- Copy local_policy.jar and US_export_policy.jar to the \$JAVA_HOME/jre/lib/security (Note: these jars will be already there so you have to overwrite them)

Installazione Tomcat Application Server:

Utilizzare la versione 7, versioni di tipo 7.0.5x scaricabili dal seguente link <http://tomcat.apache.org/>

Portare il pacchetto scaricato sotto la folder /opt del proprio server.

Si può scegliere di configurare la console di amministrazione di tomcat da interfaccia web, tramite la configurazione del file *tomcat_users.xml* in cui si inseriranno gli utenti che accederanno alle webui di tomcat per uplodare le proprie applicazioni. Configurare a proprio piacimento.

Esempio:

tomcat-users.xml in the server/tomcat/conf/ folder :

Esempio di utilizzo:

```
<user username="admin" password="Mypwd " roles="manager-gui"/>
```

Configurazione saml:

Utilizzare una folder a piacere sul file system del server. Oppure seguire le folder come suggerite sotto:

Inserire la folder saml, sotto la directory root

Ossia /root/saml

Inserire le keys del portale sotto anche qui :

/root/keys qui ci saranno i files (ad esempio il pfx file)

Successivamente configurare tomcat per i vari connector; il file è situato in :

install_dir/conf/server.xml:

Vedi paragrafo successivo.

Modifica della porta in LISTENING

Per cambiare la porta su cui ascolta il server, editare *install_dir/conf/server.xml* e cambiare gli attributi della porta Connector element da 8080 alla porta che si desidera (es: 80).

```
<Connector port="80" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" />
```

Configurazione HTTPS con certificato per il dominio, configurare le impostazioni del certificato corrette. Vedi esempio generico sottostante da completare con le proprie informazioni:

```
<Connector port="443" protocol="org.apache.coyote.http11.Http11Protocol"
    maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
    clientAuth="false" sslProtocol="TLS"
    keystoreFile="XXXXXXXXXXXXXXXXXX" → path al keystoreFile
    keystorePass="XXXXXXX" → Password della keystore
    keystoreType="PKCS12" → type of keystore
    keyAlias="XXXXXXXXXXXXXXXXXXXXXXX" → Alias della key
    keyPass="XXXXXXXXXX" /> → Password della key
```

In fondo allo stesso file server.xml, ricordarsi di inserire le seguenti impostazioni per i vari contesti, all'interno del tag HOST:

```
<Context path="" docBase="<nomeWARsenzaEstensione>">
    <!-- Default set of monitored resources -->
    <WatchedResource>WEB-INF/web.xml</WatchedResource>
</Context>
<Context path="/ROOT" docBase="/ROOT">
    <!-- Default set of monitored resources -->
    <WatchedResource>WEB-INF/web.xml</WatchedResource>
</Context>
```

In alcuni casi si dovrà mettere al primo Context path="/" ed al secondo Context path="/root"

Endpoints:

RICORDARSI successivamente di modificare opportunamente il file application.properties, con i dati del database corretti ed accessi e tutti gli endpoints, fra cui anche l'IP address del business layer

JAVA HEAP (In order to prevent Java Heap error)

Per casi di errore tipo : Tomcat – java.lang.OutOfMemoryError: PermGen space

Open {\$tomcat-folder}\bin\catalina.bat

Search JAVA_OPTS

Settare od incrementare a seconda delle impostazioni del proprio server. Esempio di incremento:

```
JAVA_OPTS= -Xms128m -Xmx1024m -XX:PermSize=64m -XX:MaxPermSize=256m
```

Oppure

```
JAVA_OPTS="-XX:MaxPermSize=512m -Xms256m -Xmx512m"
```

WIKI : Deployment del portale PRISMA e configurazione SSO

In merito alle informazioni per il deployment del portale PRISMA e configurazione SSO :

Deployment portale su testbed generico:

- Service Provider Entity ID: (SP_ENTITY_ID) : XXXXXXXXXXXXx
- Password del keystore di applicazione (KEYSTORE_PWD) : XXXXXXXXXXXXXXX
- Nome della chiave per la crittografia del metadati lato SP (SAML_KEY_NAME) : XXXXXXXXXXXXXXX
- Password della chiave per la crittografia del metadati lato SP (SAML_KEY_PWD) : XXXXXXXXXXXXXXX
- Dominio di pubblicazione (DOMAIN) : XXXXXXXXX

Caricamento dei metadata degli Identity Provider associati:

I metadata relativi agli Identity Providers associati alla piattaforma vengono caricati da file. E' quindi necessario scaricare i metadata di un dato IdP in formato XML e caricare quest'ultimi sull'Application Server (Tomcat 7) che ospita il deployment del portale, precisamente nella directory `/root/saml/metadata/{nome-idp}/`.

Nota: questi dati vengono letti in fase di bootstrap dell'applicazione dalla classe `it.prisma.presentationlayer.webui.configs.WebSecurityConfig`.

Pubblicazione dei metadata associati al Service Provider PRISMA:

A valle del deployment, i metadata associati al Service Provider PRISMA vengono automaticamente pubblicati all'indirizzo: `https://{prisma-uri}/saml/metadata/defaultAlias`.

Riferimenti:

Qualora fosse necessario effettuare configurazioni avanzate relative al SSO, si faccia riferimento alla documentazione ufficiale del modulo Spring SAML (<http://projects.spring.io/spring-security-saml/>).

Avvio e test del Server

Eseguire `install_dir/bin/startup.sh` (su Unix/Linux) per avviare Tomcat.

Per testare se funziona aprire un browser locale e andare all' URL `http://localhost:8080/` (si presume che non si sia modificata la porta di default 8080)

Per stoppare il Server: `install_dir/bin/shutdown.sh`.

`./startup.sh`

`./shutdown.sh`

Scaricare il progetto

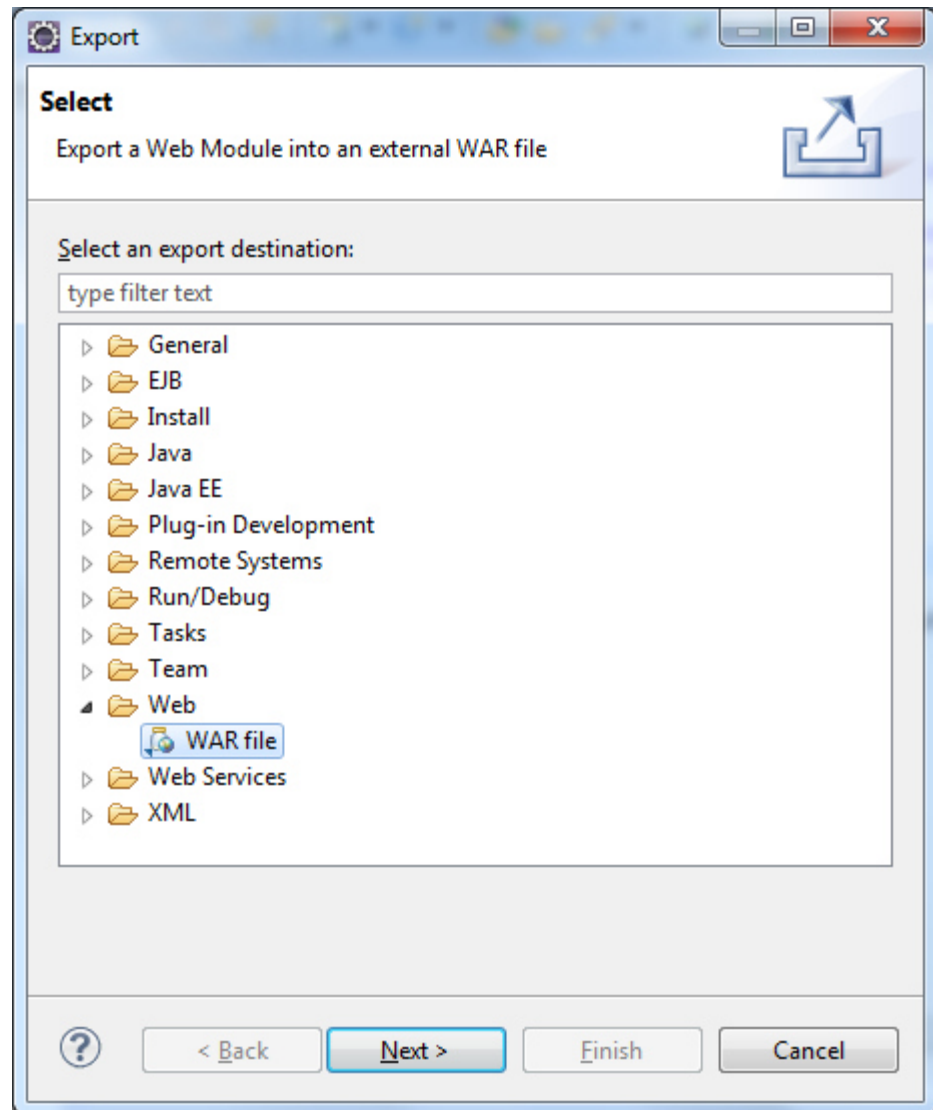
Effettuare il download del .war file della UI del portale.

Di seguito esempio di export di un .war da Eclipse..

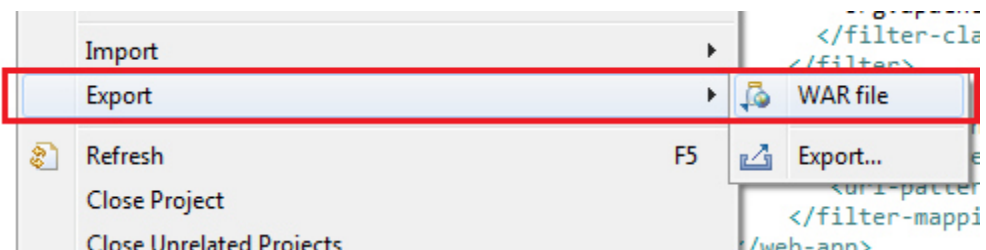
Example export WAR file from Eclipse:

- Following are the steps to create WAR file for a Java dynamic web project in Eclipse:
 - Select project name in the Project explorer view, we can access the export WAR function by either, Select in our case MIUR---WEBUI.. project:

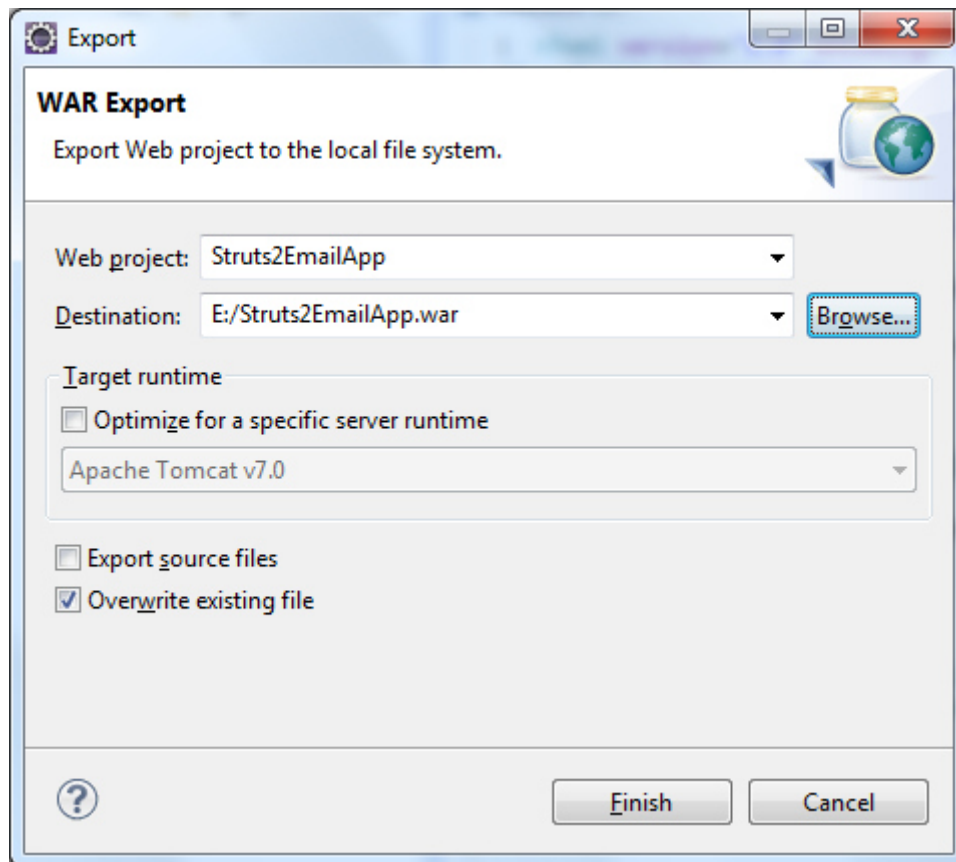
- Click File > Export... from main menu (shortcut: Alt + F + O), then select WAR file in the Export dialog like in the following screenshot:



- Or right click on the project name, select Export > WAR file from context menu:



- The WAR Export dialog appears, we have to specify two required information:
 - Web Project: Select the Java dynamic web project to export. The currently selected project is displayed as default.
 - Destination: Specify path of the WAR file to be exported. The file must end with ".war" extension.



- In addition, there are three optional checkboxes:
 - Optimize for a specific server runtime: Let Eclipse optimize the generated WAR file for a server runtime environment. Don't check if you are not sure which server on which the WAR file will be deployed.
 - Export source files: include Java sources file in the generated WAR file. This is usually left unchecked.
 - Overwrite existing file: overwrites if the specified WAR file already exists. This option is checked by default.

Now click Finish then Eclipse will create the desired WAR file in our destination folder.

Deploy APP:

Prima di trasferire il file .war sull'applicazione Server (Tomcat folder webapp), ricordarsi prima di cancellare i vecchi file e cartelle sul server, se già deployato in precedenza:

Posizionarsi nella folder /webapp/ e qui cancellare il vecchio MIUR---WEBUI.war (in caso salvarsene una copia sul proprio file system)
Cancellare anche la folder (se presente) che ha lo stesso nome MIUR---WEBUI

Controllare anche /work/Catalina/localhost/ MIUR---WEBUI

Uplodare il nuovo .war che si vuole deployare e poi rilanciare tomcat due volte, e l'aggiornamento/deploy è stato effettuato.