

**Projet AlloDocteur-Integration Web.**

**El Hadji Gaye - AlizNet**

**Avec Netbeans.**

---

**Auteur** El Hadji Gaye

**Pour** Ecole

**Date** 04/02/2018

---

**Objet** Projet AlloDocteur-Integration Graphique.

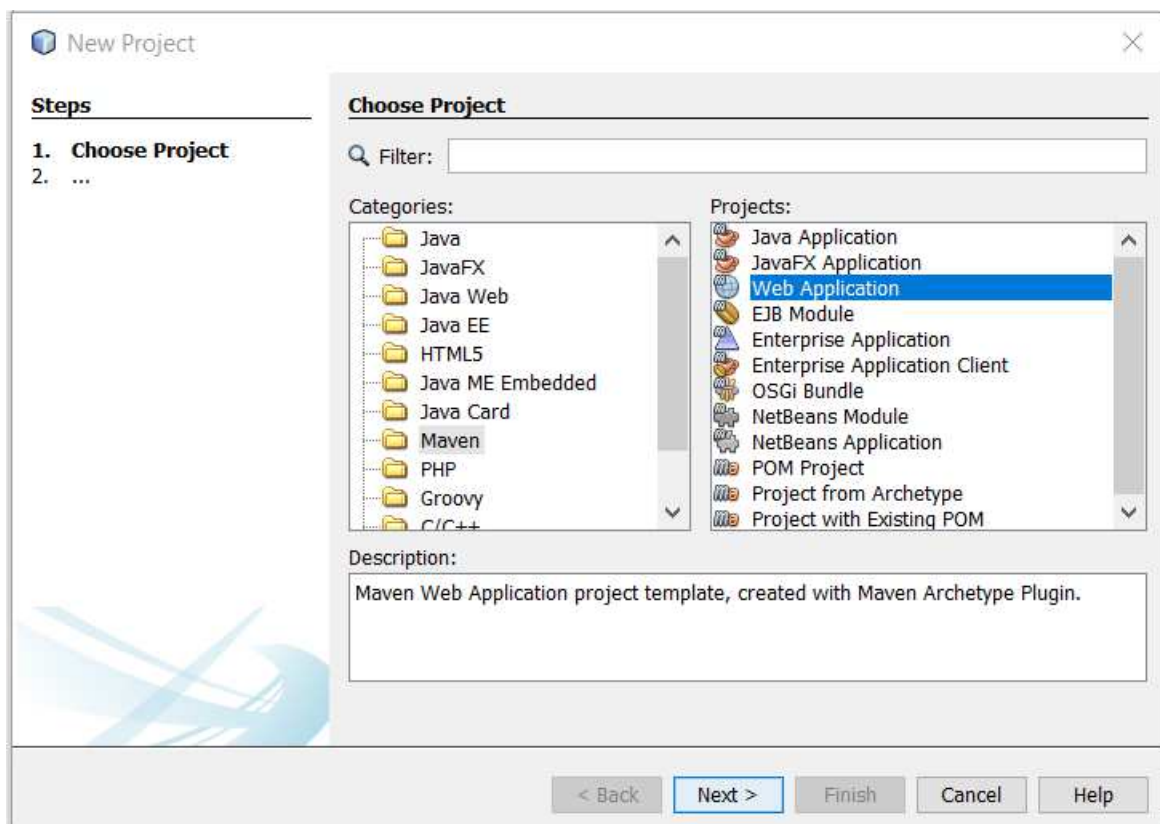
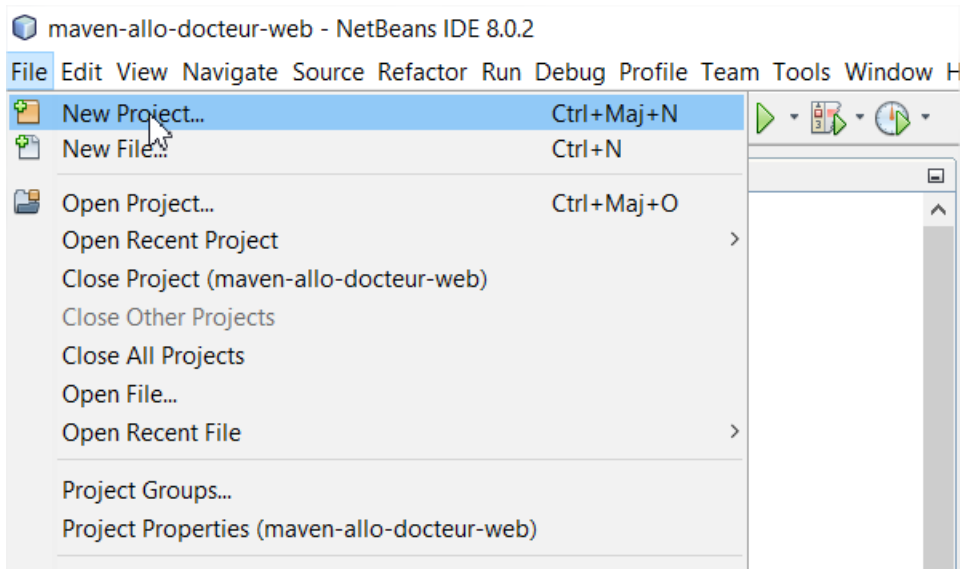
---

I)	Integration graphique d'un projet Web HttpServlet Standard .....	3
II)	Integration de Spring-Hibernate dans un projet Web.....	13

## I) Integration graphique d'un projet Web HttpServlet Standard

Le but de ce document est de vous expliquer comment utiliser l'integration graphique du projet AlloDocteur.

Créer le projet Maven Web « **maven-allo-docteur-web** ».



New Web Application

**Steps**

1. Choose Project
2. **Name and Location**
3. Settings

**Name and Location**

Project Name:

Project Location:

Project Folder:

Artifact Id:

Group Id:

Version:

Package:  (Optional)

< Back **Next >** Finish Cancel Help

Vous pouvez par exemple utiliser le serveur d'application **GlassFish** ou **TomCat**.

New Web Application

**Steps**

1. Choose Project
2. Name and Location
3. **Settings**

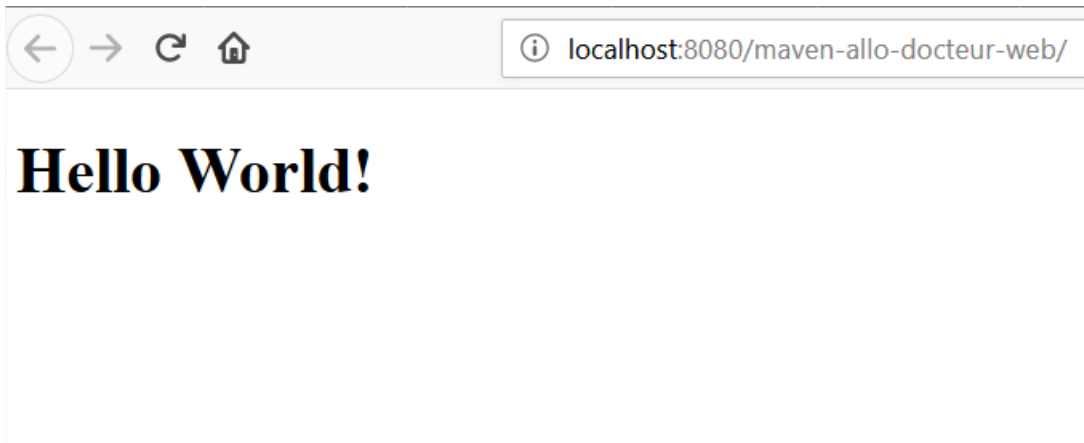
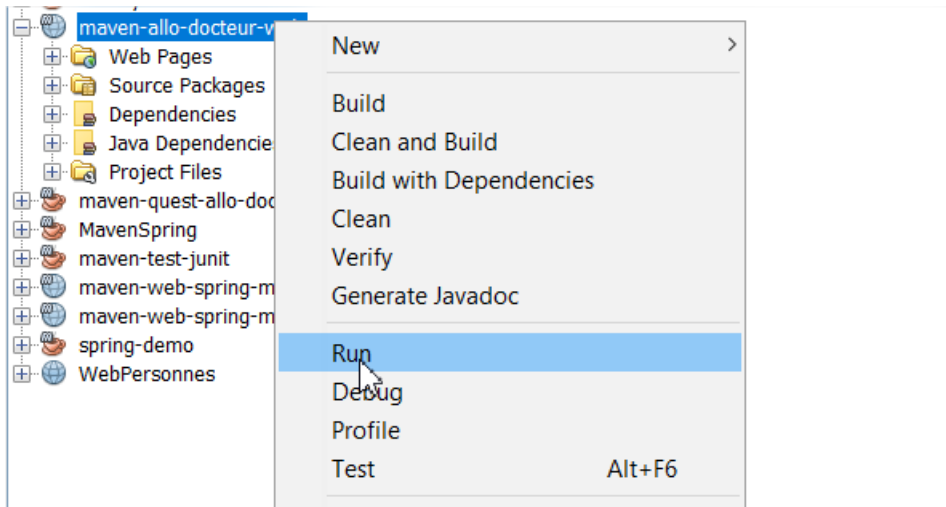
**Settings**

Server:

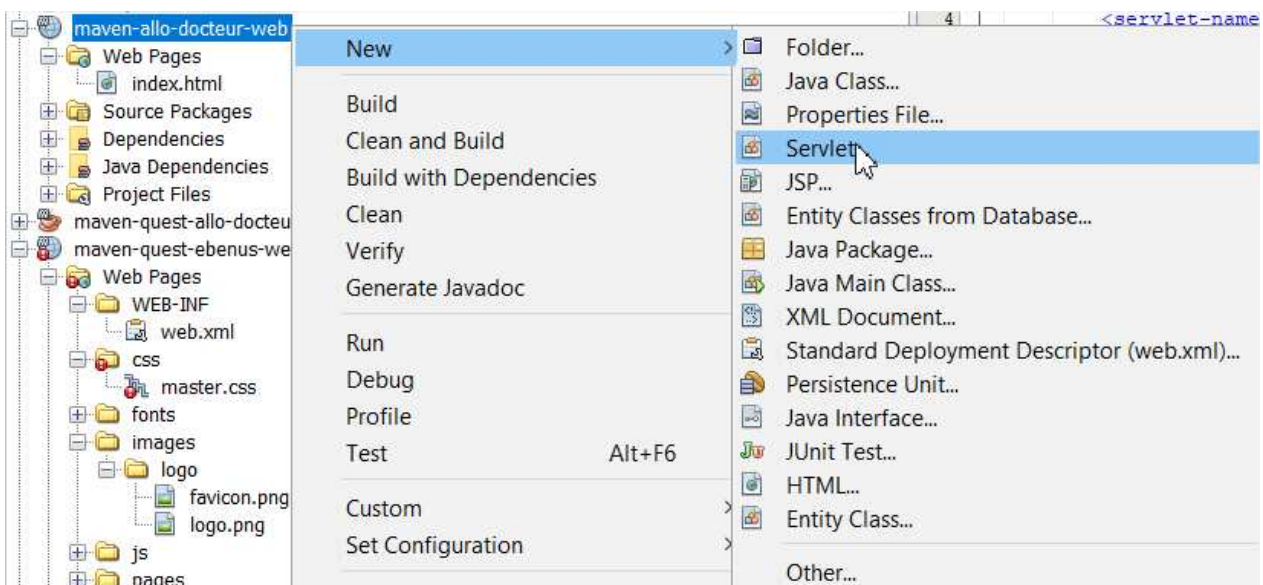
Java EE Version:

< Back Next > **Finish** Cancel Help

Lancez le projet « **maven-allo-docteur-web** ».



Créer la Servlet **LoginServlet**.



New Servlet

Steps

1. Choose File Type

2. **Name and Location**

3. Configure Servlet Deployment

**Name and Location**

Class Name: LoginServlet

Project: maven-allo-docteur-web

Location: Source Packages

Package: com.cours.servlet

Created File: cteur\maven-allo-docteur-web\src\main\java\com\cours\servlet>LoginServlet.java

< Back

Next >

Finish

Cancel

Help

Il faudra en même temps le fichier **web.xml**.

New Servlet

Steps

1. Choose File Type

2. Name and Location

3. **Configure Servlet Deployment**

**Configure Servlet Deployment**

Register the Servlet with the application by giving the Servlet an internal name (Servlet Name). Then specify patterns that identify the URLs that invoke the Servlet. Separate multiple patterns with commas.

☒ Add information to deployment descriptor (web.xml)

Class Name: com.cours.servlet.LoginServlet

Servlet Name: LoginServlet

URL Pattern(s): /LoginServlet

Initialization Parameters:

Name	Value

New

Edit...

Delete

< Back

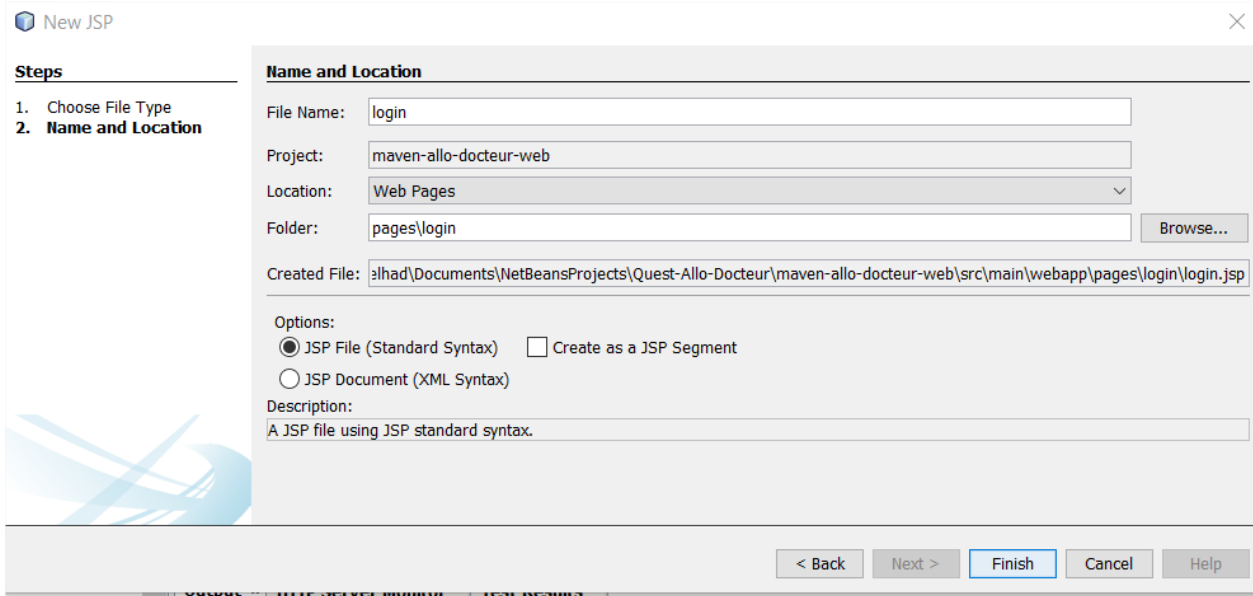
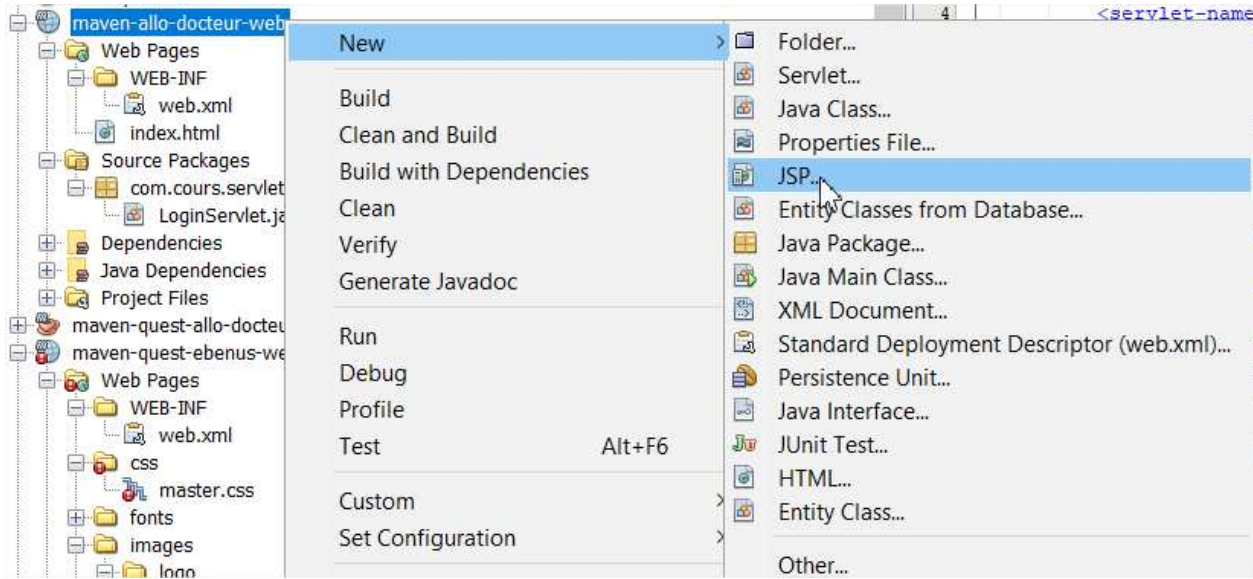
Next >

Finish

Cancel

Help

Créer la **JSP login.jsp** dans le repertoire **maven-allo-docteur-web\src\main\webapp\pages\login**.



Recuperer le dossier **CREA-INTERFACE-GRAPHIQUE-ALLODOCTEUR.zip** de l'intra et allez dans le repertoire **CREA-INTERFACE-GRAPHIQUE-ALLODOCTEUR/app**. Copiez les dossiers **fonts,images,scripts** et **styles** dans le repertoire **maven-allo-docteur-web\src\main\webapp**.

Copier aussi **favicon.ico** et **apple-touch-icon.png** dans le repertoire **maven-allo-docteur-web\src\main\webapp**.

Le fichier **LoginServlet.java** aura pour contenu :

```
package com.cours.servlet;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author elhad
 */
// @WebServlet(name = "LoginServlet", urlPatterns = {"/LoginServlet"})
public class LoginServlet extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        this.getServletContext().getRequestDispatcher("/pages/login/login.jsp").forward(request, response);
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

    }

}
```

Le fichier **web.xml** aura pour contenu :

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-
app_3_1.xsd">
  <servlet>
    <servlet-name>LoginServlet</servlet-name>
    <servlet-class>com.cours.servlet.LoginServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>LoginServlet</servlet-name>
    <url-pattern>/LoginServlet</url-pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>
      30
    </session-timeout>
  </session-config>
  <welcome-file-list>
    <welcome-file>LoginServlet</welcome-file>
  </welcome-file-list>
</web-app>
```

Prendre le contenu du fichier **CREA-INTERFACE-GRAPHIQUE-ALLODOCTEUR/app/allo-doctor-compte.html** pour le copier dans **maven-allo-docteur-web\src\main\webapp\pages\login\login.jsp**.

Il y aura besoins de faire quelques adaptation :

```
<link rel="stylesheet" href="styles/bootstrap.css">
```

Deviendra :

```
<link rel="stylesheet" href="/maven-allo-docteur-web/styles/bootstrap.css">
```

```
<script src="/maven-allo-docteur-web/scripts/jquery.js"></script>
```

Deviendra :

```
<script src="/maven-allo-docteur-web/scripts/jquery.js"></script>
```

Le fichier login.jsp devient :

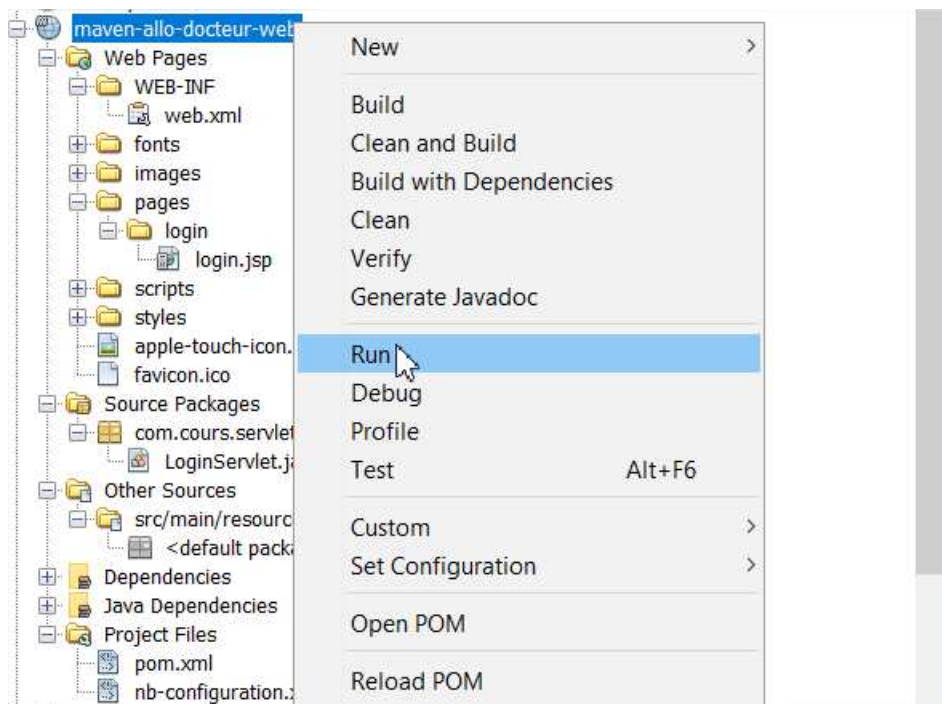
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!doctype html>
<html class="no-js" lang="">
  <head>
    <meta charset="utf-8">
    <meta name="description" content="">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Allo Docteur</title>
    <link rel="apple-touch-icon" href="/maven-allo-docteur-web/apple-touch-icon.png">
    <link rel="stylesheet" href="/maven-allo-docteur-web/styles/bootstrap.css">
    <link rel="stylesheet" href="/maven-allo-docteur-web/styles/main.css">
    <link rel="stylesheet" href="/maven-allo-docteur-web/styles/rwd.css">
  </head>
  <body class="clearfix">
    <!--[if IE]>
      <p class="browserupgrade">Vous utilisez un <strong>obsolète</strong> navigateur. Merci de se <a
href="http://browsehappy.com/">mettre à jour</a> pour améliorer votre experience.</p>
    <![endif]-->
    <div class="container-fluid no-padding">
      <div class="container-fluid clearfix no-padding header-container">
        <div class="container header-content">
          <a href="#" class="logo">
            
          </a>
          <ul class="loginMenu">
            <li><a href="#">Création compte</a></li>
            <li><a class="active" href="#">Connexion</a></li>
          </ul>
          <ul class="nav">
            <li><a href="#">Patient</a></li>
            <li><a href="#">Médecin</a></li>
            <li><a href="#">Prendre rendez vous</a></li>
            <li><a href="#">Informations médecins</a></li>
            <li><a href="#">Informations patient</a></li>
            <li><a href="#">Recherche patient</a></li>
            <li><a href="#">Présence rendez-vous</a></li>
            <li><a href="#">Selection créneau</a></li>
          </ul>
        </div>
      </div>
      <div class="container-fluid clearfix">
        <div class="container content-outer connexion">
          <div class="content-container connexion-container clearfix">
            <div class="content">
              <div class="righ-block">
                <h3>SE CONNECTER</h3>
```

```

<form class="login-form">
  <input type="text" placeholder="LOGIN">
  <input type="password" placeholder="PASSWORD">
  <button type="submit">se connecter</button>
  <div class="clearfix"> </div>
  <a href="#" class="forgot-pass-link clearfix">Mot de passe oublié?</a>
</form>
</div>
</div>
</div>
</div>
</div>
<script src="/maven-allo-docteur-web/scripts/jquery.js"></script>
<script src="/maven-allo-docteur-web/scripts/jquery.slicknav.js"></script>
<script src="/maven-allo-docteur-web/scripts/main.js"></script>
</body>
</html>

```

Lançons maintenant l'application :



On obtient au chargement de l'Url : <http://localhost:8080/maven-allo-docteur-web/>



Vous pouvez maintenant modifier la Servlet **LoginServlet** afin de réaliser l'authentification afin d'afficher soit la page d'accueil des médecins (<http://elhadji-gaye.fr/alloDocteur/medecin.html>) dans le cas où l'utilisateur est un médecin ou la page d'accueil des patients (<http://elhadji-gaye.fr/alloDocteur/patients.html>) dans le cas où l'utilisateur est un patient.

Vous pouvez réitérer la même opération pour tous les pages html qui sont dans **CREA-INTERFACE-GRAPHIQUE-ALLODOCTEUR/app** et transformer les pages HTML en page JSP.

Vous trouverez le projet **maven-allo-docteur-web.zip** resultant dans <http://elhadji-gaye.fr/alloDocteur/integration-web/maven-allo-docteur-web.zip>, vous pouvez le telecharger mais il vaut mieux faire la procedure soit même pour bien comprendre.

## II) Integration de Spring-Hibernate dans un projet Web.

Créer le fichier **applicationContext.xml** dans **maven-allo-docteur-web\src\main\webapp\WEB-INF**.

Le contenu du fichier **applicationContext.xml** sera :

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:tx="http://www.springframework.org/schema/tx"
  xmlns:context="http://www.springframework.org/schema/context"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-4.2.xsd
    http://www.springframework.org/schema/tx
    http://www.springframework.org/schema/tx/spring-tx-4.2.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context-4.2.xsd">

  <context:component-scan base-package="com.cours.service,com.cours.dao" />

  <bean id="entityManagerFactory"
class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">
    <property name="persistenceXmlLocation" value="classpath*:META-INF/spring-persistence.xml" />
    <property name="dataSource" ref="dataSource" />
    <property name="jpaVendorAdapter">
      <bean class="org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter">
        <!-- <property name="showSql" value="true" />-->
        <property name="databasePlatform" value="org.hibernate.dialect.MySQL5InnoDBDialect" />
        <property name="generateDdl" value="true" />
      </bean>
    </property>
    <property name="loadTimeWeaver">
      <bean class="org.springframework.instrument.classloading.InstrumentationLoadTimeWeaver" />
    </property>
  </bean>

  <!-- le gestionnaire de transactions -->
  <tx:annotation-driven transaction-manager="txManager" />
  <bean id="txManager" class="org.springframework.orm.jpa.JpaTransactionManager">
    <property name="entityManagerFactory" ref="entityManagerFactory" />
  </bean>

  <!-- traduction des exceptions -->
  <bean class="org.springframework.dao.annotation.PersistenceExceptionTranslationPostProcessor" />
  <!-- persistence -->
  <bean class="org.springframework.orm.jpa.support.PersistenceAnnotationBeanPostProcessor" />

  <!-- Simple implementation of the standard JDBC DataSource interface,
  configuring the plain old JDBC DriverManager via bean properties -->
  <bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
```

```

    <property name="driverClassName" value="com.mysql.jdbc.Driver" />
    <property name="url" value="jdbc:mysql://localhost:3306/base_allo_docteur" />
    <property name="username" value="application" />
    <property name="password" value="passw0rd" />
  </bean>
</beans>

```

Le fichier **web.xml** devient :

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-
app_3_1.xsd">
  <servlet>
    <servlet-name>LoginServlet</servlet-name>
    <servlet-class>com.cours.servlet.LoginServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>LoginServlet</servlet-name>
    <url-pattern>/LoginServlet</url-pattern>
  </servlet-mapping>
  <!--
  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/applicationContext.xml</param-value>
  </context-param>
  <listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
  </listener>
  -->
  <session-config>
    <session-timeout>
      30
    </session-timeout>
  </session-config>
  <welcome-file-list>
    <welcome-file>LoginServlet</welcome-file>
  </welcome-file-list>
</web-app>

```

Attention vous ne decommenterez l'inclusion avec **applicationContext.xml** que quand vous aurez configurez votre **Spring-Hibernate** proprement.

La Servlet **LoginServlet** devient :

```
package com.cours.servlet;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.springframework.web.context.WebApplicationContext;
import org.springframework.web.context.support.WebApplicationContextUtils;

/**
 *
 * @author elhad
 */
// @WebServlet(name = "LoginServlet", urlPatterns = {"/LoginServlet"})
public class LoginServlet extends HttpServlet {

    private static final Log log = LogFactory.getLog(LoginServlet.class);
    // private IServiceFacade serviceFacade = null;

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        //WebApplicationContext context =
        WebApplicationContextUtils.getRequiredWebApplicationContext(getServletContext());
        //serviceFacade = (IServiceFacade) context.getBean("serviceFacade");
        //log.debug("allAdresses: " + serviceFacade.getAdresseDao().findAllAdresses());
        this.getServletContext().getRequestDispatcher("/pages/login/login.jsp").forward(request, response);
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

    }
}
```

Le fichier **pom.xml** devient :

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-
4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.cours</groupId>
  <artifactId>maven-allo-docteur-web</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>war</packaging>
  <name>maven-allo-docteur-web</name>
  <properties>
    <endorsed.dir>${project.build.directory}/endorsed</endorsed.dir>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>
  <dependencies>
    <dependency>
      <groupId>com.cours</groupId>
      <artifactId>maven-allo-docteur-dao</artifactId>
      <version>1.0-SNAPSHOT</version>
    </dependency>
    <dependency>
      <groupId>com.cours</groupId>
      <artifactId>maven-allo-docteur-service</artifactId>
      <version>1.0-SNAPSHOT</version>
    </dependency>
    <dependency>
      <groupId>javax</groupId>
      <artifactId>javaee-web-api</artifactId>
      <version>7.0</version>
      <scope>provided</scope>
    </dependency>
    <!-- Pour les utilisateur du serveur d'application Tomcat ajouter la dependance org.glassfish.web -->
    <dependency>
      <groupId>org.glassfish.web</groupId>
      <artifactId>jstl-impl</artifactId>
      <version>1.2</version>
      <exclusions>
        <exclusion>
          <artifactId>servlet-api</artifactId>
          <groupId>javax.servlet</groupId>
        </exclusion>
        <exclusion>
          <artifactId>jsp-api</artifactId>
          <groupId>javax.servlet.jsp</groupId>
        </exclusion>
      </exclusions>
    </dependency>
  </dependencies>

```

```

<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-web</artifactId>
  <version>4.2.1.RELEASE</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-orm</artifactId>
  <version>4.2.1.RELEASE</version>
</dependency>
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-core</artifactId>
  <version>4.0.1.Final</version>
</dependency>
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-entitymanager</artifactId>
  <version>4.0.1.Final</version>
</dependency>
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>5.1.41</version>
</dependency>
<dependency>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
  <version>1.2.17</version>
</dependency>
<dependency>
  <groupId>log4j</groupId>
  <artifactId>apache-log4j-extras</artifactId>
  <version>1.2.17</version>
</dependency>
<dependency>
  <groupId>commons-logging</groupId>
  <artifactId>commons-logging</artifactId>
  <version>1.2</version>
  <type>jar</type>
</dependency>
</dependencies>
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>3.1</version>
      <configuration>

```

```

    <source>1.7</source>
    <target>1.7</target>
    <compilerArguments>
      <endorseddirs>${endorsed.dir}</endorseddirs>
    </compilerArguments>
  </configuration>
</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-war-plugin</artifactId>
  <version>2.3</version>
  <configuration>
    <failOnMissingWebXml>>false</failOnMissingWebXml>
  </configuration>
</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-dependency-plugin</artifactId>
  <version>2.6</version>
  <executions>
    <execution>
      <phase>validate</phase>
      <goals>
        <goal>copy</goal>
      </goals>
      <configuration>
        <outputDirectory>${endorsed.dir}</outputDirectory>
        <silent>>true</silent>
        <artifactItems>
          <artifactItem>
            <groupId>javax</groupId>
            <artifactId>javaee-endorsed-api</artifactId>
            <version>7.0</version>
            <type>jar</type>
          </artifactItem>
        </artifactItems>
      </configuration>
    </execution>
  </executions>
</plugin>
</plugins>
</build>
</project>

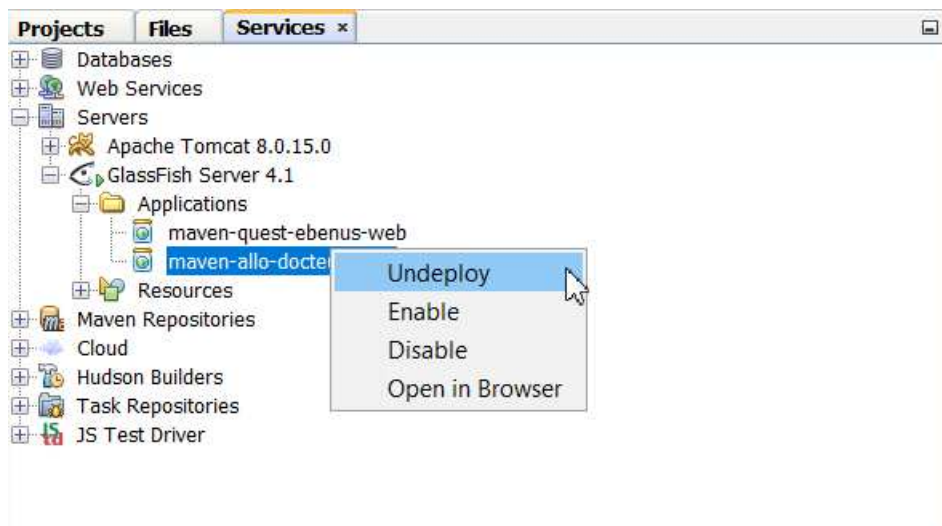
```

Allons maintenant regarder du coté du serveur GlassFish.

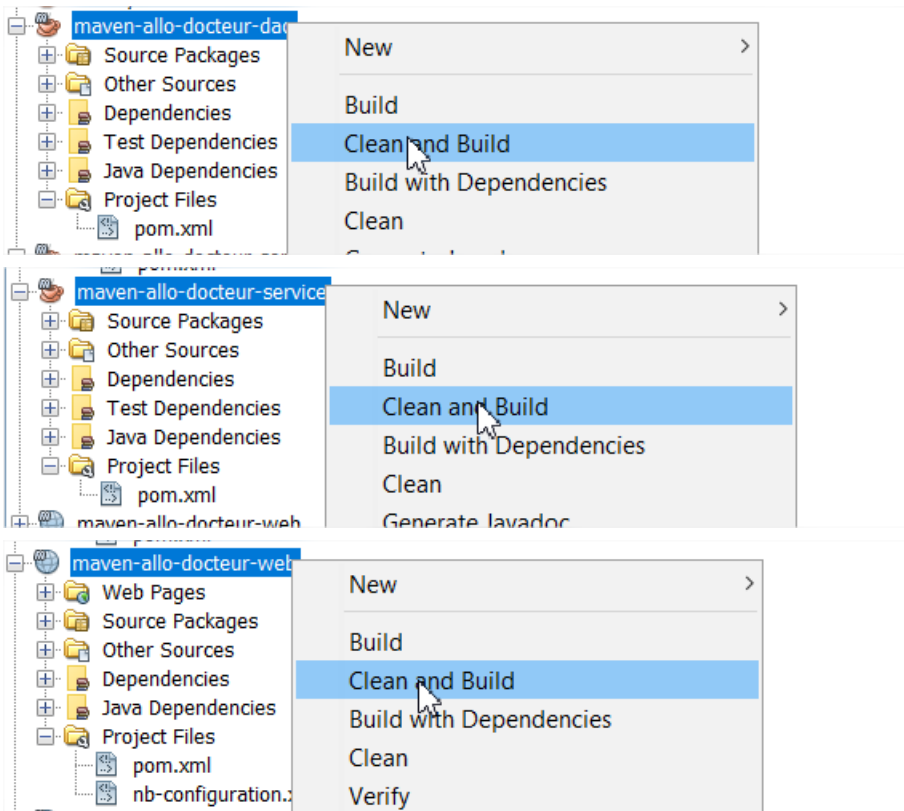


L'image ci-dessus nous montre la liste des applications qui sont déployé et nous voyons bien notre application **maven-allo-docteur-web** que nous avons déployé tout à l'heure.

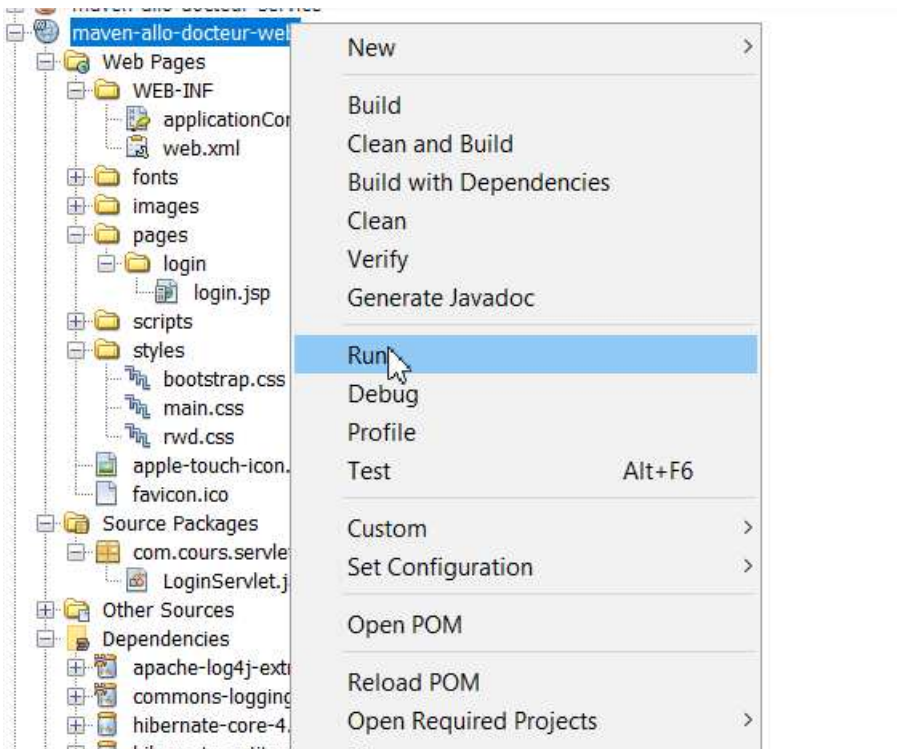
Nous allons l'enlever du serveur avec l'instruction **Undeploy**.



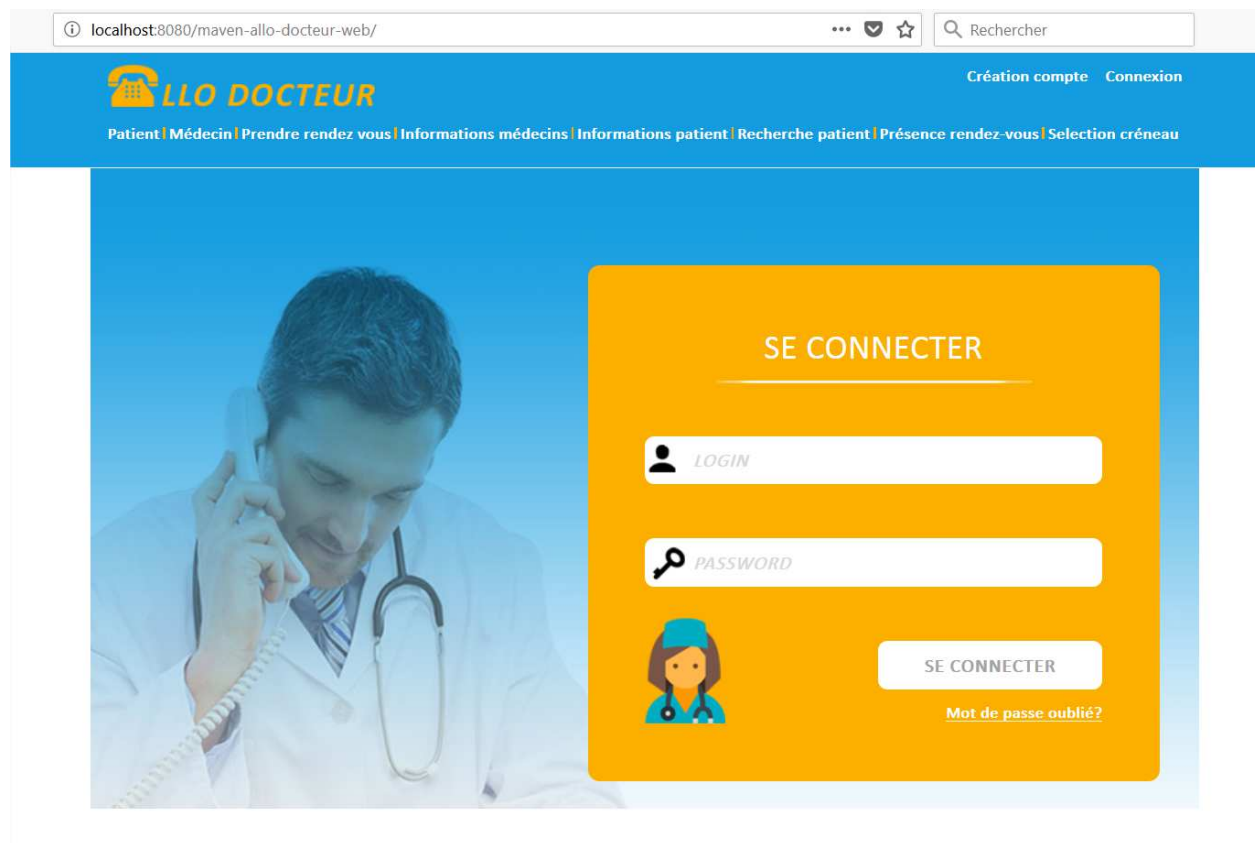
Faites un **Clean And Build** du projet **maven-allo-docteur-dao** puis du projet **maven-allo-docteur-service** et enfin **maven-allo-docteur-web**.



Lançons maintenant l'application :



On obtient au chargement de l'Url : <http://localhost:8080/maven-allo-docteur-web/>



Vous trouverez le projet **maven-allo-docteur-web.zip** resultant dans <http://elhadji-gaye.fr/alloDocteur/integration-web/maven-allo-docteur-web.zip>, vous pouvez le telecharger mais il vaut mieux faire la procedure soit même pour bien comprendre.

**Bonne chance pour la suite de votre Projet !!!**