

[Travaux Pratique]

[JAVA JEE]

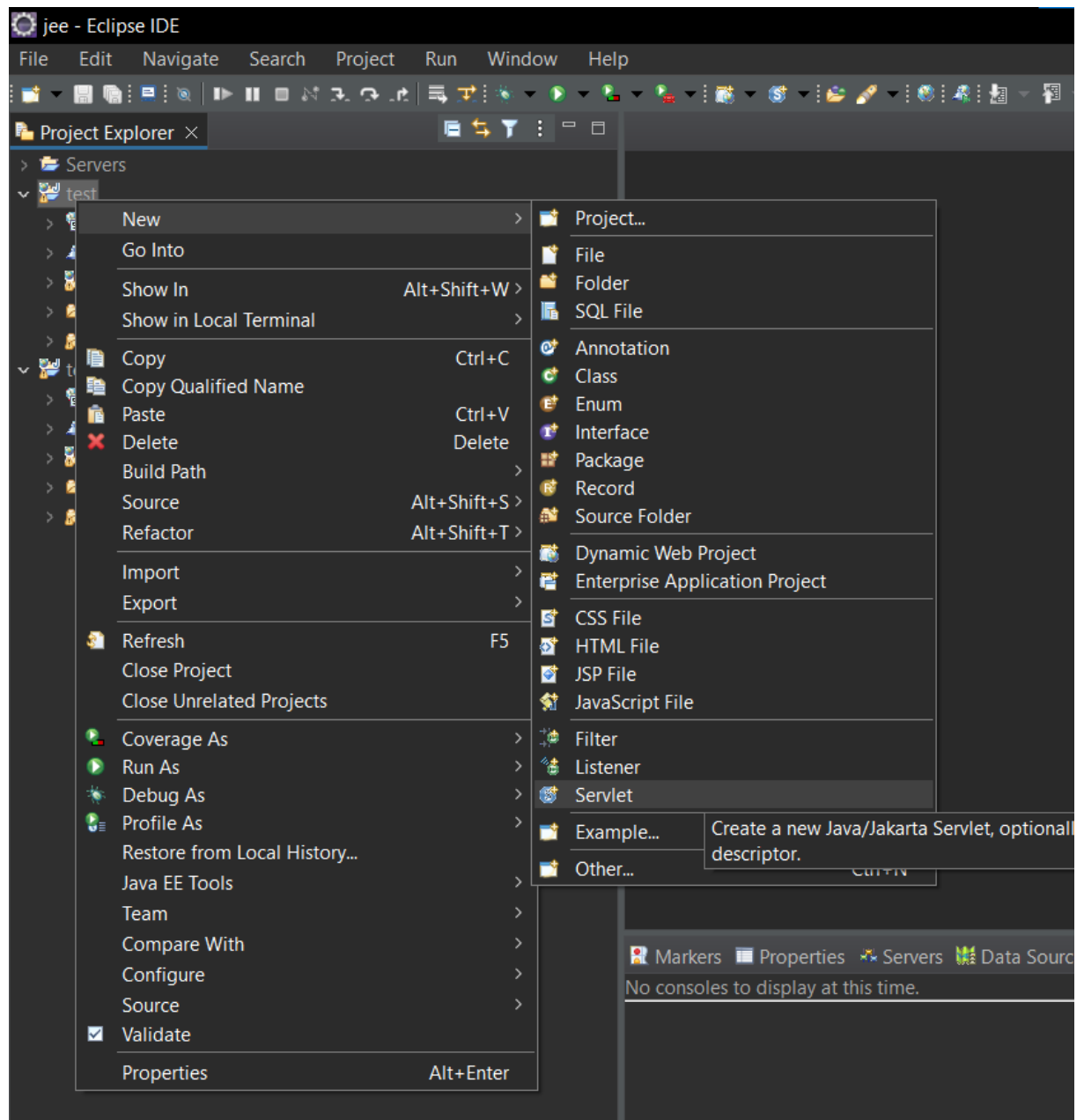
MOHAMED SALEM AHMEDOU	C14703
EL MOUSTAPHA SIDI BOUYA	C14481
FETOUH SADEGH	C14738
SELMA AHMED ZEYAD	C15785

Partie 1 :

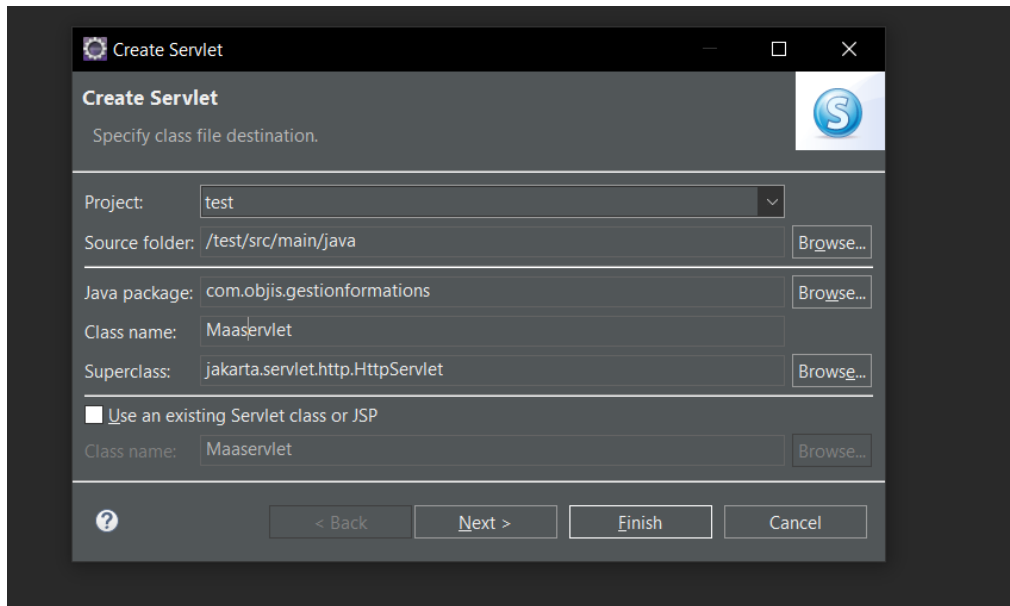
Création de notre première servlet en suivant les instructions dans le lien www.objis.com/votre-premiere-servlet/

Voici les capture d'écran nécessaire :

1. Création d'un nouveau servlet dans notre projet web dynamique



2. Ajouton le package *com.objis.gestionformations* ainsi que la classe *MaServlet*



3. En cliquant sur le bouton Finish, Eclipse génère le code suivant la ou on va modifier la méthode « doGet » de la manière suivant :

```
MaServlet.java x
import jakarta.servlet.ServletException;
9
10 /**
11  * Servlet implementation class MaServlet
12  */
13 public class MaServlet extends HttpServlet {
14     private static final long serialVersionUID = 1L;
15
16     /**
17      * @see HttpServlet#HttpServlet()
18      */
19     public MaServlet() {
20         super();
21         // TODO Auto-generated constructor stub
22     }
23
24     /**
25      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
26      */
27     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
28         // TODO Auto-generated method stub
29         //response.getWriter().append("Served at: ").append(request.getContextPath());
30         System.out.println("welcome to my servlet");
31     }
32
33     /**
34      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
35      */
36     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
37         // TODO Auto-generated method stub
38         doGet(request, response);
39     }
40 }
41
42
Markers Properties Servers Data Source Explorer Snippets Terminal Console x
No consoles to display at this time.
Writable Smart Insert 1:1:0
```

4. Déploiement de la servlet

The screenshot displays a web browser window at the top with the address bar showing `localhost:8080/test1/MaServlet`. Below the browser is the Eclipse IDE interface. The **Project Explorer** on the left shows the project structure: `test` > `Deployment Descriptor: test` > `JAX-WS Web Services` > `Java Resources` > `build` > `src` > `test1` > `Deployment Descriptor: test1` > `JAX-WS Web Services` > `Java Resources` > `src/main/java` > `com.objjis.gestioninformations` > `MaServlet.java`. The **MaServlet.java** file is open in the editor, showing the following code:

```
24  /**
25   * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
26   */
27  protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
28      // TODO Auto-generated method stub
29      //response.getWriter().append("Served at: ").append(request.getContextPath());
30      System.out.println("welcome to my servlet");
31  }
32
33  /**
34   * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
35   */
36  protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
```

The **Console** window at the bottom shows the output of the Tomcat v10.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (27 oct. 2022):

```
oct. 27, 2022 11:39:15 AM org.apache.catalina.startup.Catalina load
INFO: L'initialisation du serveur a pris [1513] millisecondes
oct. 27, 2022 11:39:15 AM org.apache.catalina.core.StandardService startInternal
INFO: Démarrage du service [Catalina]
oct. 27, 2022 11:39:15 AM org.apache.catalina.core.StandardEngine startInternal
INFO: Démarrage du moteur de Servlets : [Apache Tomcat/10.0.27]
oct. 27, 2022 11:39:16 AM org.apache.coyote.AbstractProtocol start
INFO: Démarrage du gestionnaire de protocole ["http-nio-8080"]
oct. 27, 2022 11:39:16 AM org.apache.catalina.startup.Catalina start
INFO: Le démarrage du serveur a pris [1165] millisecondes
welcome to my servlet
```

5. Modifiez la configuration et le code de la Servlet afin d'avoir le résultat suivant

The screenshot displays a web browser window at the top and the Eclipse IDE at the bottom. The browser shows the URL `localhost:8080/test1/MaServlet` and the output `welcome to my servlet`. The Eclipse IDE shows the `MaServlet.java` file with the following code:

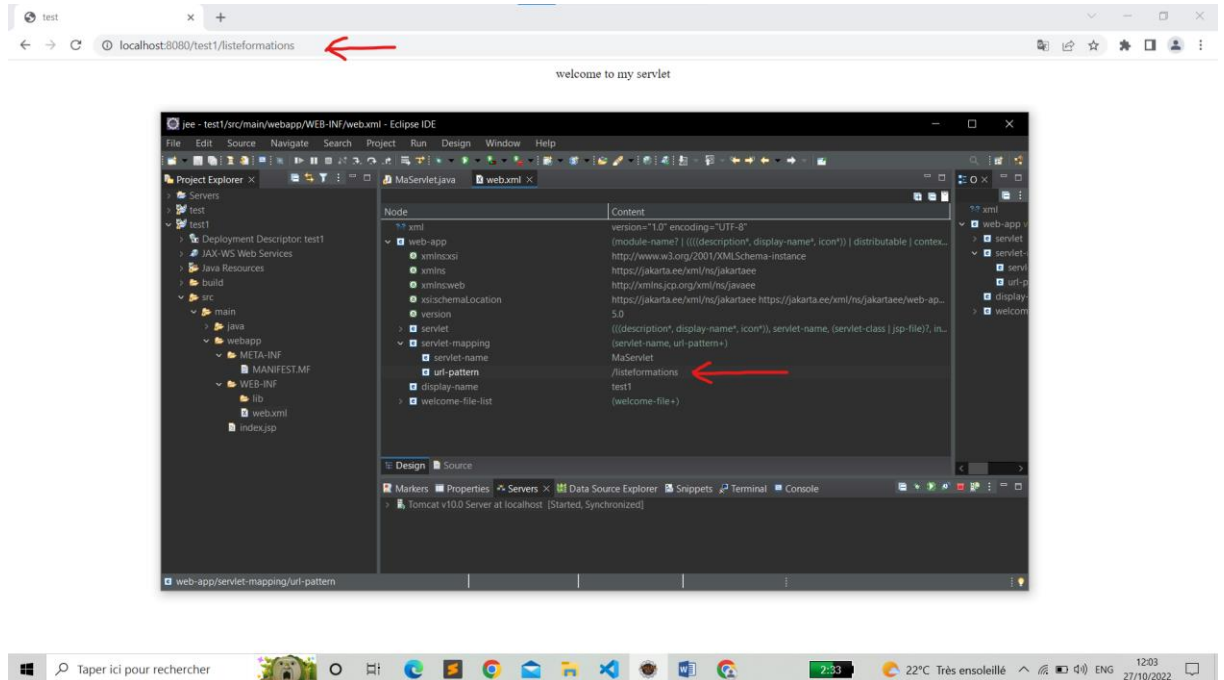
```
26 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
27 */
28 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
29     // TODO Auto-generated method stub
30     //response.getWriter().append("Served at: ").append(request.getContextPath());
31     //System.out.println("welcome to my servlet");
32     response.setContentType("text/html");
33     PrintWriter out = response.getWriter();
34     out.println("<html>");
35     out.println("<head> <title> test </title> </head>");
36     out.println("<body> <center> welcome to my servlet </center>");
37     out.println("</body>");
38     out.println("</html>");
39 }
40
41 /**
42 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
43 */
44 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
45     // TODO Auto-generated method stub
46     doGet(request, response);
47 }
```

The IDE's console window shows the following output:

```
Tomcat v10.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (27 oct. 2022)
oct. 27, 2022 11:53:11 AM org.apache.catalina.startup.Catalina start
INFO: Le démarrage du serveur a pris [1232] millisecondes
oct. 27, 2022 11:54:11 AM org.apache.catalina.core.StandardContext reload
<
```

The Windows taskbar at the bottom shows the search bar with the text "Taper ici pour rechercher" and several application icons.

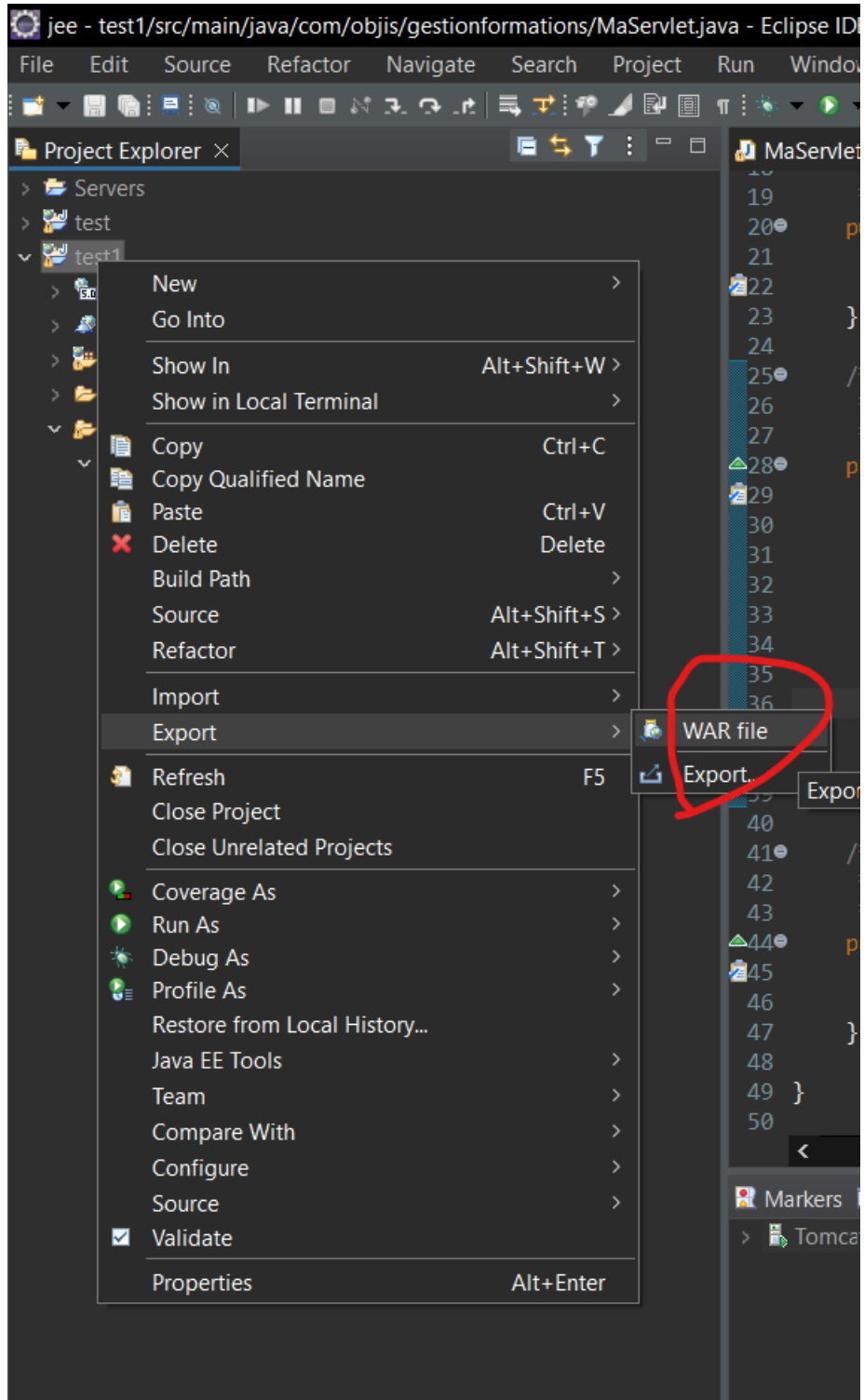
6. Configuration de la servlet à travers le fichier web.xml en changeant l'url pattern de la servlet de la manière suivante :



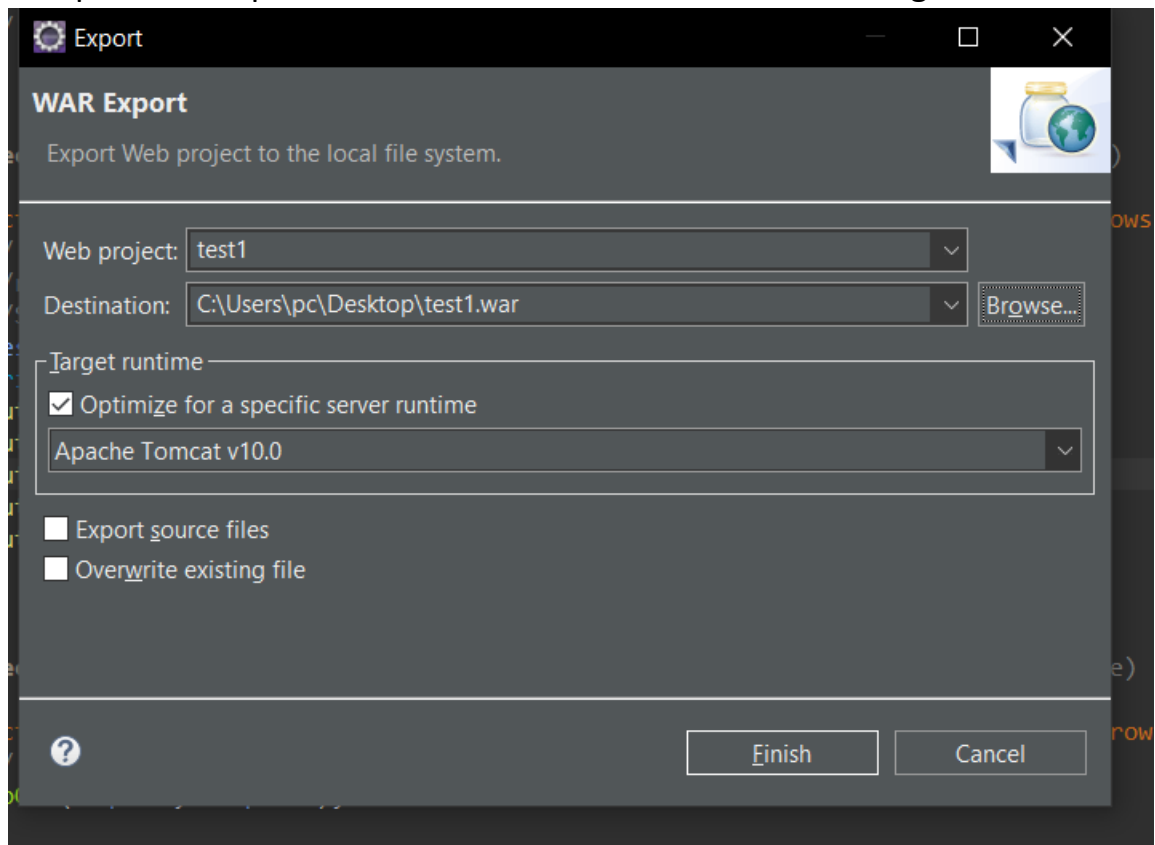
Partie 2 :

Création de fichier WAR de notre projet web dans eclipse :

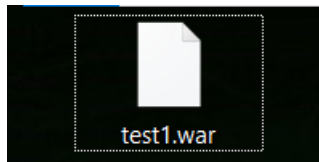
1. Sélectionner **File > Export**



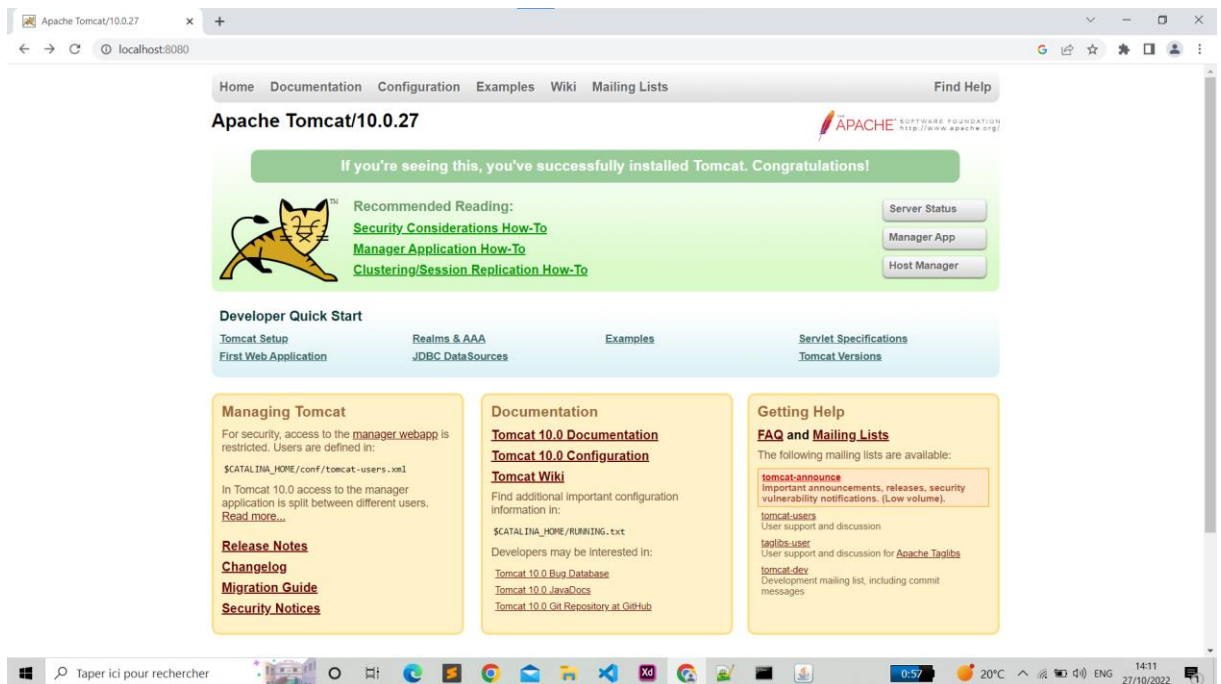
2. On spécifie l'emplacement la ou le fichier WAR va être enregistré



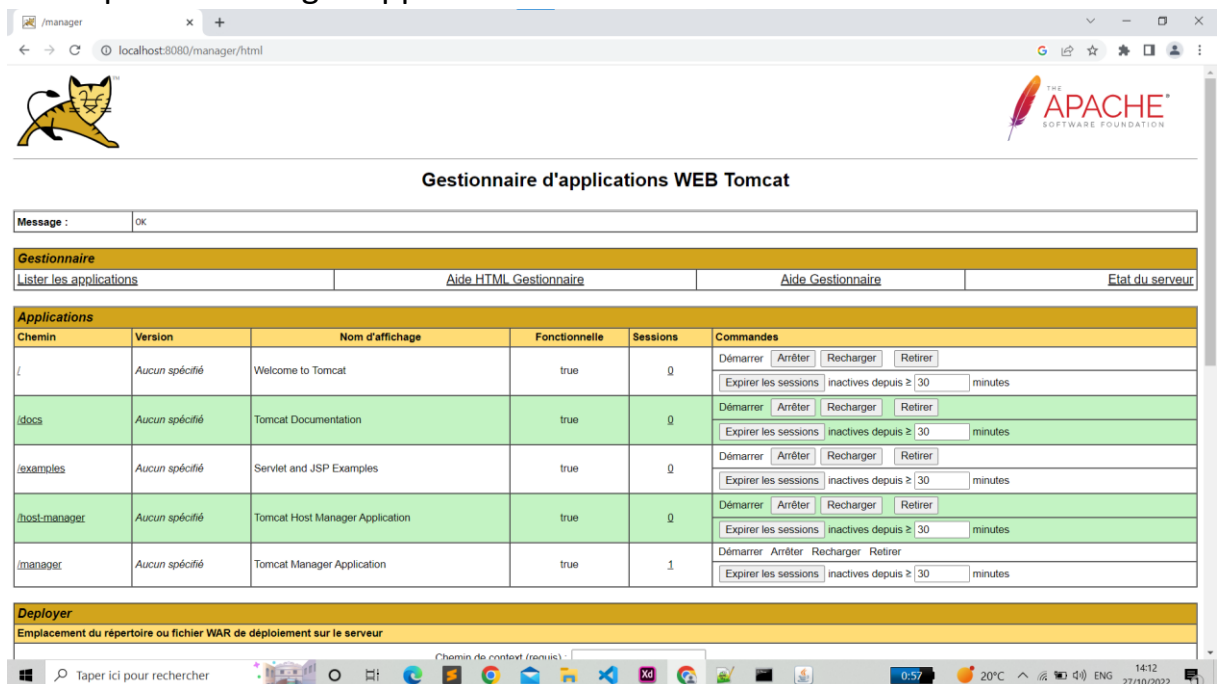
3. On clique sur finish est voici notre fichier WAR pré a être déploie



4. On démarre notre server tomcat



5. On clique sur Manager App



Chemin	Version	Nom d'affichage	Fonctionnelle	Sessions	Commandes
/	Aucun spécifié	Welcome to Tomcat	true	0	Démarrer Arrêter Recharger Retirer Expire les sessions inactives depuis ≥ 30 minutes
/docs	Aucun spécifié	Tomcat Documentation	true	0	Démarrer Arrêter Recharger Retirer Expire les sessions inactives depuis ≥ 30 minutes
/examples	Aucun spécifié	Servlet and JSP Examples	true	0	Démarrer Arrêter Recharger Retirer Expire les sessions inactives depuis ≥ 30 minutes
/host-manager	Aucun spécifié	Tomcat Host Manager Application	true	0	Démarrer Arrêter Recharger Retirer Expire les sessions inactives depuis ≥ 30 minutes
/manager	Aucun spécifié	Tomcat Manager Application	true	1	Démarrer Arrêter Recharger Retirer Expire les sessions inactives depuis ≥ 30 minutes

6. On navigue vers le bas vers la place de déploiement des application de type WAR

The screenshot shows the Tomcat Manager application running in a web browser at localhost:8080/manager/html. The interface is divided into several sections:

- Deployer**: This section contains fields for deployment configuration. It includes a table with columns for application name, status, and actions. Below the table, there are input fields for "Chemin de contexte (requis)", "Version (pour les déploiements en parallèle)", "URL du fichier XML de configuration", and "URL vers WAR ou répertoire". A "Deployer" button is located at the bottom of this section.
- Fichier WAR à déployer**: This section has a label "Choisir le fichier WAR à téléverser" and a "Choisir un fichier" button. Below it, there is a "Déployer" button.
- Configuration**: This section is titled "Relisant les fichiers de configuration TLS" and contains a "Nom d'hôte TLS (optionnel)" input field and a "Retire" button.
- Diagnostics**: This section is titled "Vérifiez si une application web a causé une fuite de mémoire lors de son arrêt, rechargement ou déchargement". It includes a "Trouver des fuites" button and a description of the diagnostic process. Below this, there is a "Diagnostics de configuration TLS du connecteur" section with three rows: "Chiffres" (listing virtual TLS hosts and keys), "Certificats" (listing virtual TLS hosts and certificate chains), and "Certificats de confiance" (listing virtual TLS hosts and trust certificates).

7. On sélectionne notre application WAR

This screenshot shows the same Tomcat Manager interface as the previous one, but with a Windows file selection dialog box open over it. The dialog box is titled "Ouvrir" and shows the contents of the "C:\Bureau" directory. The files listed include "rufus-3-11.exe", "servlet.pdf", "test1.war", "testServlet.war", "tp2jee.docx", "tp2jee.pdf", "Vysor", and "XAMPP Control Panel". The "testServlet.war" file is selected. The dialog box has a "Nom du fichier" field containing "testServlet.war" and a "Tous les fichiers (*.*)" filter. The "Ouvrir" button is highlighted.

8. Est finalement on le déploie

localhost:8080/manager/html

/manager	Aucun spécifié	Tomcat Manager Application	true	1	Démarrer Arrêter Recharger Retirer Expirer les sessions inactives depuis ≥ 30 minutes
----------	----------------	----------------------------	------	---	--

Deployer

Emplacement du répertoire ou fichier WAR de déploiement sur le serveur

Chemin de contexte (requis) :

Version (pour les déploiements en parallèle) :

URL du fichier XML de configuration :

URL vers WAR ou répertoire :

Deployer

Fichier WAR à déployer

Choisir le fichier WAR à téléverser Choisir un fichier testServlet.war

Deployer

Configuration

Relisant les fichiers de configuration TLS

Nom d'hôte TLS (optionnel)

Retire

Diagnostics

Vérifiez si une application web a causé une fuite de mémoire lors de son arrêt, rechargement ou déchargement

Trouver des fuites Le diagnostic doit démarrer une collecte complète de la mémoire, utilisez-le avec précaution dans un environnement de production

Diagnostics de configuration TLS du connecteur



Chiffres Lister les hôtes virtuels TLS configurés et les chiffres utilisés par chacun

Certificats Entrez la liste des hôtes virtuels TLS et la chaîne de certificats pour chacun.

Certificats de confiance Lister les hôtes virtuels TLS configurés et les certificats de confiance utilisés par chacun

9. Voici notre application a été déployée

localhost:8080/manager/html/upload?org.apache.catalina.filters.CSRF_NONCE=BEC8BFC178C7FCE49D2AEC99F486145



Gestionnaire d'applications WEB Tomcat

Message : OK

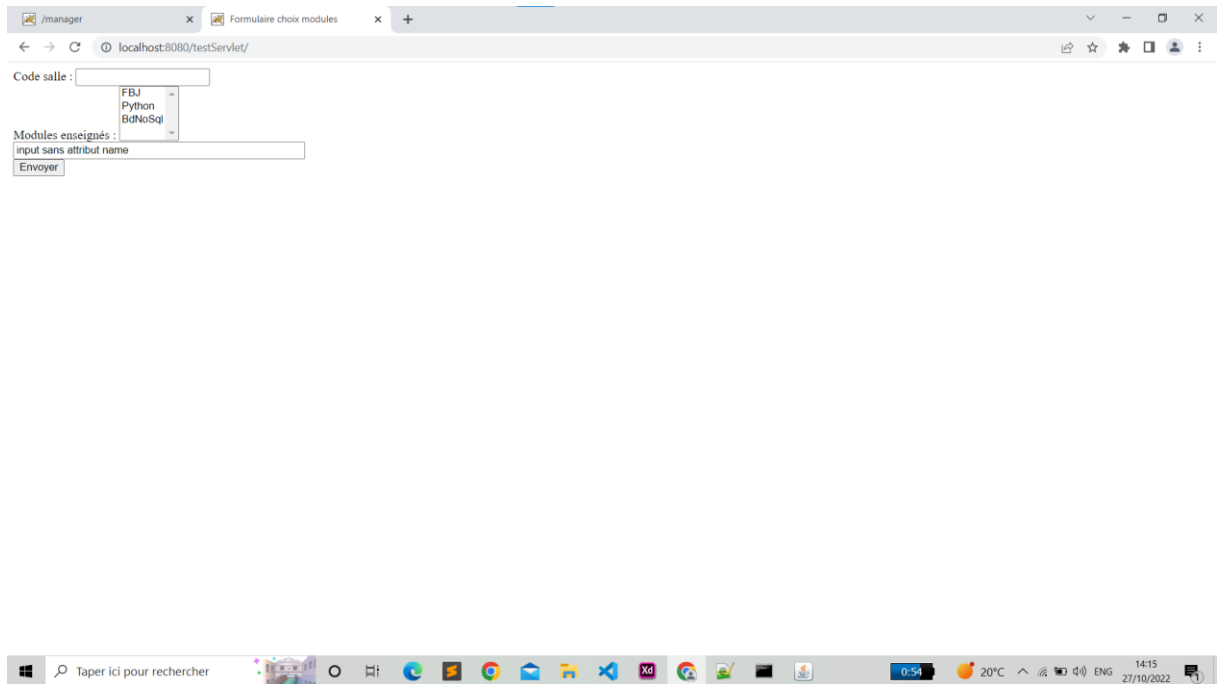
Gestionnaire

Lister les applications	Aide HTML Gestionnaire	Aide Gestionnaire	Etat du serveur
---	--	-----------------------------------	---------------------------------

Applications

Chemin	Version	Nom d'affichage	Fonctionnelle	Sessions	Commandes
/	Aucun spécifié	Welcome to Tomcat	true	0	Démarrer Arrêter Recharger Retirer Expirer les sessions inactives depuis ≥ 30 minutes
/docs	Aucun spécifié	Tomcat Documentation	true	0	Démarrer Arrêter Recharger Retirer Expirer les sessions inactives depuis ≥ 30 minutes
/examples	Aucun spécifié	Servlet and JSP Examples	true	0	Démarrer Arrêter Recharger Retirer Expirer les sessions inactives depuis ≥ 30 minutes
/host-manager	Aucun spécifié	Tomcat Host Manager Application	true	0	Démarrer Arrêter Recharger Retirer Expirer les sessions inactives depuis ≥ 30 minutes
/manager	Aucun spécifié	Tomcat Manager Application	true	1	Démarrer Arrêter Recharger Retirer Expirer les sessions inactives depuis ≥ 30 minutes
/testServlet	Aucun spécifié	testServlet	true	0	Démarrer Arrêter Recharger Retirer Expirer les sessions inactives depuis ≥ 30 minutes

10. Test de l'app (l'application cest l'application donne dans le fiche td)



Partie 3 :

Correction de l'exo dans le TP

EX1

1. a. Créer un nouveau projet web dynamique « testServlet »

The screenshot shows the 'New Dynamic Web Project' wizard in the Eclipse IDE. The window title is 'New Dynamic Web Project'. The main heading is 'Dynamic Web Project' with a subtitle: 'Create a standalone Java-based Web Application or add it to a new or existing Enterprise Application.' The wizard is divided into several sections:

- Project name:** A text field containing 'testServlet'.
- Project location:** A section with a checked checkbox 'Use default location'. Below it, a text field shows the location 'C:\Users\pc\jee\testServlet' and a 'Browse...' button.
- Target runtime:** A dropdown menu showing 'Apache Tomcat v10.0' and a 'New Runtime...' button.
- Dynamic web module version:** A dropdown menu showing '5.0'.
- Configuration:** A dropdown menu showing 'Default Configuration for Apache Tomcat v10.0' and a 'Modify...' button. Below this, a descriptive text states: 'A good starting point for working with Apache Tomcat v10.0 runtime. Additional facets can later be installed to add new functionality to the project.'
- EAR membership:** A section with an unchecked checkbox 'Add project to an EAR'. Below it, a text field shows 'EAR' and a 'New Project...' button.
- Working sets:** A section with an unchecked checkbox 'Add project to working sets' and a 'New...' button. Below this, a text field is empty and a 'Select...' button is present.

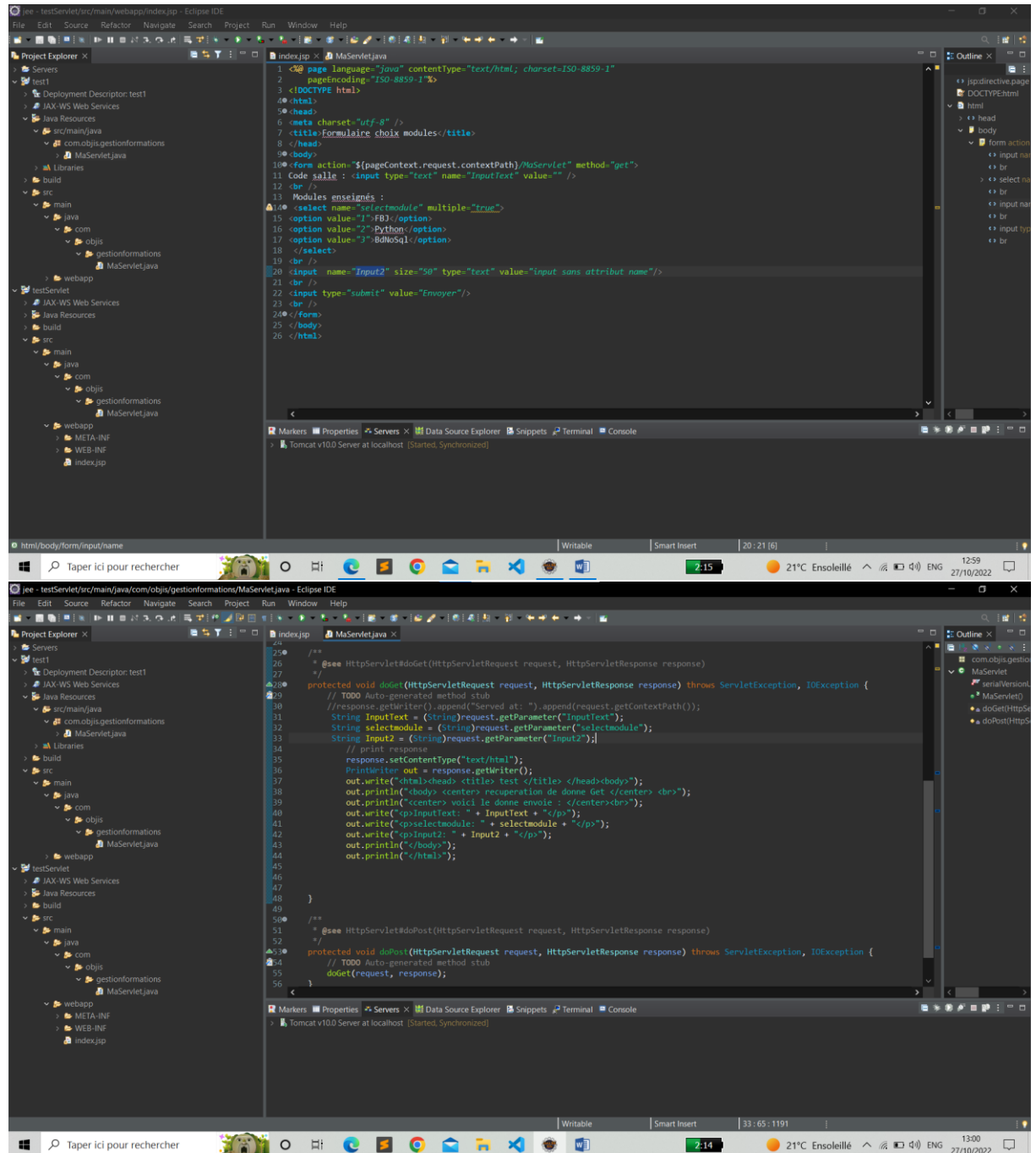
At the bottom of the wizard, there are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'. A help icon (?) is located on the far left of the bottom bar.

-
- b. ajouter le formulaire html au projet crée

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="utf-8" />
7 <title>Formulaire choix modules</title>
8 </head>
9 <body>
10 <form method="get">
11   Code salle : <input type="text" name="InputText" value="" />
12   <br />
13   Modules enseignés :
14 <select name="selectmodule" multiple="true">
15 <option value="1">FBI</option>
16 <option value="2">Python</option>
17 <option value="3">BdNoSql</option>
18 </select>
19 <br />
20 <input size="50" type="text" value="input sans attribut name"/>
21 <br />
22 <input type="submit" value="Envoyer"/>
23 <br />
24 </form>
25 </body>
26 </html>
```

2. Créer une servlet qui récupéré les données du formulaire

on doit modifier notre formulaire en précisant l'action de cette formulaire vers l'url-pattern de notre servlet et en donnant a la dernière input un nom pour qu'on puis récupérer leur valeur



Formulaire choix modules

localhost:8080/testServlet/index.jsp

Code salle :

Modules enseignés :

FBJ

Python

BdNoSql

input avec attribut name

Envoyer

test

localhost:8080/testServlet/MaServlet?InputText=123&selectmodule=2&input2=input+avec+attribut+name

recuperation de donne Get

voici le donne envoie :

InputText: 123

selectmodule: 2

Input2: input avec attribut name

3. Modifier le formulaire précédent en précisant la méthode (post) et l'action ("/testServlet/AjouterSalle") ;
Comme on a déjà crée notre servlet et on a l'appelé MaServlet on va modifier votre question comme suit
Modifier le formulaire précédent en précisant la méthode (post) et l'action ("/testServlet/MaServlet) ;

A ce moment on peut just appelle la methode doget dans la methode dopost lors quon recoit un request de type post (bien sur on peut dune autre part copier le code qu'on a dans doget dans dopost meme chose)


```

index.jsp  MaServlet.java  web.xml
28  protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
29      // TODO Auto-generated method stub
30      //response.getWriter().append("Served at: ").append(request.getContextPath());
31      String InputText = (String)request.getParameter("InputText");
32      String selectmodule = (String)request.getParameter("selectmodule");
33      String Input2 = (String)request.getParameter("Input2");
34      // print response
35      response.setContentType("text/html");
36      PrintWriter out = response.getWriter();
37      out.write("<html><head> <title> test </title> </head><body>");
38      out.println("<body> <center> recuperation de donne Get </center> <br>");
39      out.println("<center> voici le donne envoie : </center><br>");
40      out.write("<p>InputText: " + InputText + "</p>");
41      out.write("<p>selectmodule: " + selectmodule + "</p>");
42      out.write("<p>Input2: " + Input2 + "</p>");
43      out.println("</body>");
44      out.println("</html>");
45
46
47
48  }
49
50  /**
51   * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
52   */
53  protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
54      // TODO Auto-generated method stub
55      doGet(request, response);
56  }
57
58  }

```

Formulaire choix modules x +

← → ↻ localhost:8080/testServlet/index.jsp

Code salle :

Modules enseignés :

test x +

← → ↻ localhost:8080/testServlet/MaServlet

recuperation de donne Get

voici le donne envoie :

InputText: 123

selectmodule: 2

Input2: input avec attribut name

- Créer la servlet « MaServlet », qui affiche les modules enseignés associés au code saisi ;
 Ce question a été répondu par nous directement dans le question 2 et 3 précédent on a
 affiché les module associes directement dans le navigateur a la place de le printer dans le
 console de eclipse .