



*Ministère de l'Enseignement Supérieur et de la recherche scientifique
Direction Générale des Etudes Technologiques
Institut Supérieur des Etudes Technologiques de Sfax*

compte rendu

Nom et Prénom :

Ameny Ayedi

Groupe :

INF-DSI22 G1

N° CIN / Passeport :

11179217

Année Universitaire : 2024/2025

Table des matieres

1 Introduction	2
1.1 Objectifs du TP	2
1.2 Contexte et outils utilis�es	2
2 Pr�paration de l'environnement de d�veloppement	2
2.1 Creation des dossiers de travail	2
2.2 Installation de JDK 8	2
2.3 Installation d'Eclipse	2
2.4 Configuration d'Apache Tomcat	2
3 Cr�eation d'une application web simple	2
3.1 Creation d'un projet web dynamique	2
3.2 Cr�eation d'une page web statique	3
3.3 Cr�eation d'une Servlet	3
3.4 Ex�cution de la Servlet	3
4 Mapping URL/Servlet via le fichier web.xml	3
4.1 Utilisation du fichier web.xml	3
4.2 Ex�cution avec le nouveau mapping	3
5 G�n�ration de code HTML via une Servlet	3
5.1 Creation d'une Servlet g�n�rant du HTML	3
5.2 Appel d'un fichier HTML depuis une Servlet	3
6 Passage de param�tres via une Servlet	4
6.1 R�cup�ration de param�tres en mode GET	4
6.2 Cr�eation d'un formulaire HTML	4
6.3 Passage de param�tres en mode POST	4
7 Conclusion	4
7.1 Bilan du TP	4
7.2 Points �a am�liorer	4
8 Annexes	4
8.1 Code source	4
8.2 URLs de test	4

1 Introduction

1.1 Objectifs du TP

Ce TP a pour objectif de mettre en place un environnement de développement JEE (Java Enterprise Edition) et de créer une application web simple en utilisant Eclipse, Apache Tomcat, et les Servlets. L'objectif principal est de comprendre comment configurer un environnement de développement pour les applications web en Java, ainsi que de créer et exécuter des Servlets pour gérer les requêtes HTTP.

1.2 Contexte et outils utilisés

Les outils utilisés dans ce TP sont :

- JDK 8 (Java Development Kit)
- Eclipse IDE (Environnement de Développement Intégré)
- Apache Tomcat 9 (Serveur web)
- Servlets (Composants web Java)

2 Préparation de l'environnement de développement

2.1 Création des dossiers de travail

Un dossier nommé Atelier JEE a été créé sous C : \ pour organiser les fichiers et les projets.

2.2 Installation de JDK 8

Le JDK 8 (version 64 bits pour Windows) a été téléchargé depuis le site officiel d'Oracle.



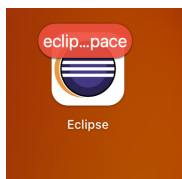
La variable d'environnement JAVA_HOME a été configurée pour pointer vers le répertoire d'installation du JDK. Le chemin des exécutables de JDK (%JAVA_HOME%\bin) a été ajouté au début de la variable d'environnement PATH.

2.3 Installation d'Eclipse

L'EDI (Environnement de Développement Intégré) Eclipse (version 2019-12 R) a été téléchargé et installé dans le dossier D : \Atelier JEE\EDI. Le workspace (espace de travail) a été configuré pour pointer vers D : \Atelier JEE\workspace.

2.4 Configuration d'Apache Tomcat

Le serveur web Apache Tomcat (version 9) a été téléchargé et extrait dans le dossier D : \Atelier JEE\Servers. Eclipse a été configuré pour utiliser Tomcat en tant que serveur



d'exécution.

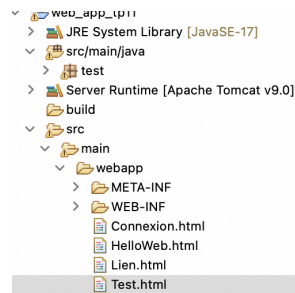
3 Création d'une application web simple

3.1 Création d'un projet web dynamique

Un projet web dynamique nommé web app tp01 a été créé dans Eclipse. Le projet a été configuré pour utiliser Apache Tomcat v9.0 comme serveur d'exécution.

3.2 Création d'une page web statique

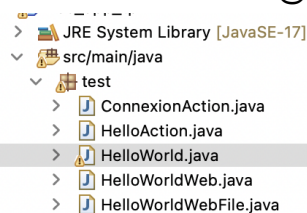
Une page HTML simple nommée Test.html a été créée dans le dossier WebContent. La page a été exécutée sur le serveur Tomcat et accessible via l'URL : `http://localhost:8080/web app tp01/Test.html`



tp01/Test.html

3.3 Création d'une Servlet

Une Servlet nommée HelloWorld a été créée dans le package test. La Servlet a été configurée pour répondre aux requêtes HTTP en mode GET et POST. Le mapping URL/Servlet a été réalisé à l'aide de l'annotation `@WebServlet` dans le code source de la Servlet.



3.4 Exécution de la Servlet

La Servlet HelloWorld a été exécutée et accessible via l'URL : `http://localhost:8080/web app tp01/HelloWorld`

Un autre mapping URL a été ajouté pour la Servlet en utilisant le pattern `*.do`, permettant d'accéder à la Servlet via : `http://localhost:8080/web app tp01/test.do`

4 Mapping URL/Servlet via le fichier web.xml

4.1 Utilisation du fichier web.xml

Le mapping URL/Servlet a également été réalisé en utilisant le fichier web.xml situé dans le dossier WebContent/WEB-INF. Le fichier web.xml a été modifié pour définir un nouveau mapping URL pour la Servlet HelloWorld avec le pattern `/HelloWeb`.

4.2 Exécution avec le nouveau mapping

La Servlet HelloWorld a été exécutée avec le nouveau mapping URL : `http://localhost:8080/web app tp01/HelloWeb`

5 Génération de code HTML via une Servlet

5.1 Création d'une Servlet générant du HTML

Une nouvelle Servlet nommée HelloWorldWeb a été créée pour générer du code HTML dynamiquement. La méthode `doGet` de la Servlet a été modifiée pour générer une réponse HTML contenant un titre HelloWorld depuis une servlet.

5.2 Appel d'un fichier HTML depuis une Servlet

Une autre Servlet nommée HelloWorldWebFile a été créée pour rediriger la requête vers un fichier HTML externe (HelloWeb.html). Le fichier HTML a été appelé à l'aide de

la méthode `request.getRequestDispatcher("HelloWeb.html").forward(request, response)` ;

3

6 Passage de paramètres via une Servlet

6.1 Récupération de paramètres en mode GET

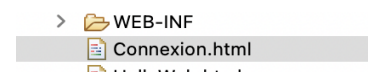
Une Servlet nommée HelloAction a été créée pour récupérer et afficher un paramètre

nommé prenom passé dans l'URL. La Servlet a été testée avec l'URL suivante : `http://localhost:8080/web app tp01/HelloAction ?prenom=Ali`

6.2 Création d'un formulaire HTML

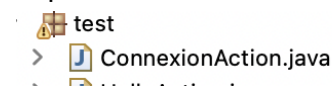
Un formulaire HTML nommé connexion.html a été créé pour saisir un nom et un mot de passe. Le formulaire a été configuré pour envoyer les données en mode GET à une Servlet

nommée ConnexionAction.



6.3 Passage de paramètres en mode POST

Le formulaire a été modifié pour envoyer les données en mode POST afin de masquer les paramètres dans l'URL. La Servlet ConnexionAction a été modifiée pour gérer les requêtes en mode POST et valider les champs du formulaire.



7 Conclusion

7.1 Bilan du TP

Ce TP a permis de prendre en main l'environnement de développement JEE en configurant Eclipse avec Apache Tomcat. Nous avons créé une application web simple en utilisant des Servlets pour gérer les requêtes HTTP, générer du contenu HTML dynamique, et passer des paramètres via des formulaires HTML. Les concepts de mapping URL/Servlet, ainsi que la gestion des requêtes en mode GET et POST, ont été explorés en détail.

7.2 Points à améliorer

— Sécurité : Pour sécuriser davantage l'application, il serait intéressant d'explorer l'utilisation de HTTPS et de valider les entrées utilisateur pour éviter les attaques par injection.

— Gestion des erreurs : Une meilleure gestion des erreurs pourrait être implémentée pour fournir des messages d'erreur plus clairs aux utilisateurs.

— Interface utilisateur : L'interface utilisateur pourrait être améliorée en utilisant des frameworks front-end comme Bootstrap pour un design plus moderne.

8 Annexes

8.1 Code source

Les fichiers sources du projet sont disponibles dans le dossier `D:\Atelier JEE\workspace\web app tp01`.

8.2 URLs de test

Les URLs suivantes ont été utilisées pour tester l'application :

— `http://localhost:8080/web app tp01/Test.html`

4

— `http://localhost:8080/web app tp01/HelloWorld`

— `http://localhost:8080/web app tp01/HelloWeb`

— `http://localhost:8080/web-app-tp01/HelloAction?prenom=Ali`