133 – tomcat

Rapport personnel

Version 1 du 31.03.23

**Schwander Nicolas**

Module du 09.08.2019 au 19.11.2019

Table des matières

[1. Introduction 3](#_Toc131075990)

[1.1 Objectifs du module 3](#_Toc131075991)

[2. TestsTechno 3](#_Toc131075992)

[2.1 Déploiement Tomcat 3](#_Toc131075993)

[2.1.1 Théorie 3](#_Toc131075994)

[2.1.2 Tizoo Cpanel 4](#_Toc131075995)

[2.1.3 Local 4](#_Toc131075996)

[2.2 JSP 5](#_Toc131075997)

[2.2.1 Exercice 2 5](#_Toc131075998)

[2.2.2 Exercice 3 5](#_Toc131075999)

[2.2.3 Exercice 4 5](#_Toc131076000)

[2.3 JDBC 6](#_Toc131076001)

[2.3.1 Exercice 5 6](#_Toc131076002)

[2.4 Bean with scope 7](#_Toc131076003)

[2.5 Servlet 7](#_Toc131076004)

[2.5.1 Exercice 7 7](#_Toc131076005)

[2.6 Session varibles/beans 8](#_Toc131076006)

[2.6.1 Exercice 8 8](#_Toc131076007)

[3. Conclusion 9](#_Toc131076008)

[3.1 Ce que j’ai appris 9](#_Toc131076009)

[3.2 Ce que j’ai aimé 9](#_Toc131076010)

[3.3 Ce que j’ai moins aimé 9](#_Toc131076011)

1. Introduction
   1. Objectifs du module

* Côté client

La liaison au métier par protocole http (XML et JSON)

HTML5-JS

Pattern MVC

Contrôle des évènements produits par l’utilisateur

* Côté serveur

Métier en JAVA (servlets et JSP)

Création et utilisation de WebServices

L’application Web en JSP

Pattern MVC

Contrôle des évènements produits par les appels des clients

Connexion au serveur MySQL

* En global

Hébergement APACHE et TOMCAT

Documentation et JAVADoc

1. TestsTechno
   1. Déploiement Tomcat
      1. Théorie

Tomcat est un serveur web principalement utilisé pour exécuter des applications web Java, telles que des sites web dynamiques et des applications d'entreprise.

Tomcat fonctionne en tant que conteneur de servlets, il peut gérer les requêtes et les réponses pour les applications web Java. Il utilise le protocole HTTP pour communiquer avec les clients et il peut être configuré pour prendre en charge HTTPS pour les connexions sécurisées.

* + 1. Tizoo Cpanel

Une version de tomcate était préalablement installé sur nos comptes Tizoo, il fallait simplement mofier un fichier de config.

Une image contenant texte

Description générée automatiquement

Cette variable permet a tomcat de créer automatiquement un repertoire avec le war unpacké. Ce qui rend la page utilisable.

* + 1. Local

Il a fallu simplement installer la version la plus recente de tomcat avec quelques modification notamment :

On a ajouté le rôle manager-script

Une image contenant texte

Description générée automatiquement

Ensuite j’ai exécuté le script de startup de tomcat par le cmd et j’ai ajouté le server tomcat a netbeans :

Une image contenant texte

Description générée automatiquement

* 1. JSP

Les JSP sont une technologie de développement web basée sur Java qui permet aux développeurs de créer des pages web dynamiques en combinant des fragments de code Java avec des modèles HTML.

Lorsqu'un navigateur demande une page JSP à un serveur web Tomcat, le serveur interprète la page JSP et génère du code HTML qui est renvoyé au navigateur. Le code Java inclus dans la page JSP est exécuté côté serveur pour effectuer des traitements et des calculs.

* + 1. Exercice 2

Dans cet exercice il suffisait simplement de lancer une jsp qui contenait une architecture html et un hello world en titre.

Une image contenant texte

Description générée automatiquement

* + 1. Exercice 3

Dans cet exercice on fait un formulaire html qui pointe vers une jsp qui effectue simplement un if sur le mot de passe et nom d’utilisateur

* + 1. Exercice 4

Dans cet exercice il a fallu utiliser un petit webservice, pour ce faire j’ai créé une petite jsp.

Une image contenant texte

Description générée automatiquement

Celle-ci est appelé par un formulaire dans l’index.html

* 1. JDBC
     1. Exercice 5

Le but dans cet exercice était de créer un petite application web java qui lis des données dans une DB.

Pour cela on a un workerDB qui permet d’ouvrir une connexion et de récupérer les pays dedans.

Du coté de la jsp il fallait simplement iterer a travers les pays et les afficher :

Une image contenant texte

Description générée automatiquement

* 1. Bean with scope

Il existe en java web des manières de faire transiter des beans avec un scope qui définit jusqu’à ou il est atteignable

<jsp:useBean id="beanInfo" scope="session" class="beans.BeanInfo"/>

<jsp:useBean id="beanError" scope="session" class="beans.BeanError"/>

Ici le scope est définit comme « session » ce qui veut dire que c’est le même partout dans la même session.

Ensuite on set ce Bean :

beanInfo.setNom(username);

beanInfo.setPrenom(password);

et pour le réutiliser au sein d’une même session,

Une image contenant texte

Description générée automatiquement



* 1. Servlet
     1. Exercice 7

Dans cet exercice nous avons découvert les servlet, il suffisait de faire un requête post avec un formulaire en html sur le servlet et on renvoyai les données traitées.

Une image contenant texte

Description générée automatiquement



* 1. Session varibles/beans
     1. Exercice 8

En java web il existe des variables de session, dans l’exercice suivant il suffisait de faire transiter quelques donnes dans un bean stocké dans la session.

J’ai fait ceci de la manière suivante dans le servlet :

Une image contenant texte

Description générée automatiquement

Ensuite pour récupérer les variables coté client :

Une image contenant texte

Description générée automatiquement



* 1. Rest

1. Conclusion
   1. Ce que j’ai appris
   2. Ce que j’ai aimé
   3. Ce que j’ai moins aimé

Raccourci clavier pour P\_titre 1 : ctrl + f1