**Laborator 1 – Tema Servlet**

Pavaluca Tudor – Ioan

Tema Servlet presupune instalarea pe calculatoarele propria a unor servere de aplicatii (AS): containerele Tomcat, Jetty şi serverele WildFly, GlassFish.

**Aplicatie**

GuessingNumber este o aplicatie web in care utilizatorul trebuie sa ghiceasca un numar dintr-un anumit interval. El are un numar limitat de incercari. Limita minima, limita maxima si numarul maxim de incercari for fi alese de catre utilizator.

Aceasta aplicatie contine 2 servleturi, 3 pagini JSP si un fisier CSS. Primul Servlet(StartServlet) valideaza cele trei valori trimise de catre user (limita minima, limita maxima si numarul maxim de incercari) si ajuta la pornirea jocului, punand pe sesiune un obiect de tip Game. Al doilea Servlet(GameServlet), se ocupa de afisarea si logica jocului. Cele 3 pagini JSP au urmatoarele roluri:

* index.jsp – afisarea paginii de start a jocului si trimiterea catre Servlet a datelor introduse de utilizator.
* result.jsp – afisarea rezultatului (Daca a castigat sau daca a pierdut)
* error.jsp – afisarea unui mesaj de eroare in cazul in care utilizatorul a introdus date gresite.

Aplicatia mai contine clasa Game si o clasa Util. Aceasta din urma contine metodele de validare folosite in aplicatie. Pentru mapare s-a folosit adnotarea @WebServlet.

**Configurare si modalitati de deploy:**

1. **Tomcat WAR**

* Creare war folosind Gradle
* Copierea fisierului war in fisierul webapps din **$CATALINA\_HOME**
* Pornire server Tomcat folosind comanda **startup** in folderul **bin**
* Aplicatia web va fi disponibila la adresa <http://localhost:8081/ServletLab>

1. **Tomcat Context Extern**

* Vom crea un director extern ce va contine fisierele corespunzatoare aplicatiei (in acest caz, directorul extern va fi **d:\Master\Semestrul 1\TPAJD\Teme\ContextExtern**)
* Se creaza fisierul “**context.xml**” in **CATALINA\_HOME/conf/Catalina/localhost/**
* Fisierul are urmatorul continut:

<Context docBase=" d:\Master\Semestrul 1\TPAJD\Teme\ContextExtern"

path="" reloadable="true" />

* Se porneste tomcat folosind **startup** in **bin**
* Se acceseaza aplicatia <http://localhost:8081/context>

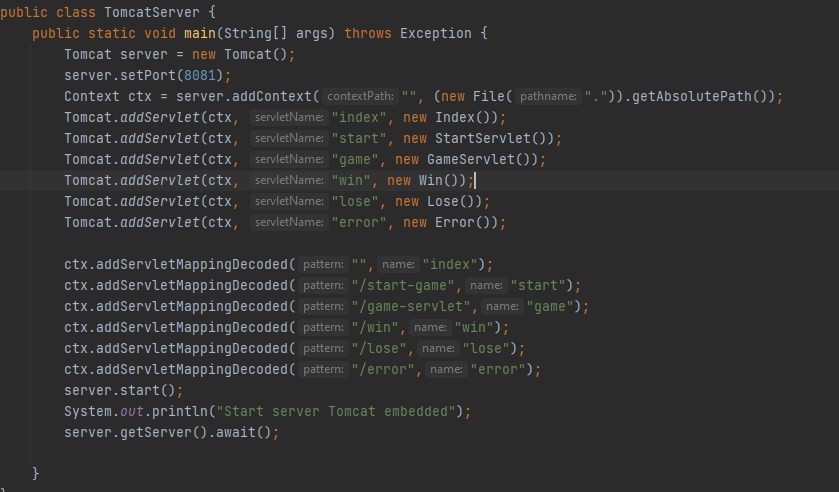
1. **Tomcat Embedded**

* Adaugarea unor dependinte suplimentare in fisierul build.gradle

compile 'org.apache.tomcat.embed:tomcat-embed-core:8.5.23', 'org.apache.tomcat.embed:tomcat-embed-jasper:8.5.23',

'org.apache.tomcat:tomcat-jasper:8.5.23'

* Daca avem fisiere JSP, este recomandat sa le transformam in Servlet-uri
* Implementarea unei noi clase Java



* Arhiva jar se construieste prin intermediul comenzii **gradle clean build**.
* Pentru a rula aplicatia trebuie sa rulam comanda **java -jar build/libs/TomcatEmbedded.jar** din directorul aplicatiei.
* Aplicatia va fi disponibila la adresa <http://localhost:8081>

1. **Jetty WAR**

* Creare war folosind Gradle
* Copierea fisierului war in **$JETTY\_HOME/webapps**
* Executare comanda **java -jar start.jar -Djetty.port=8082** in **%JETTY\_HOME**
* Se acceseaza aplicatia la <http://localhost:8082/ServletLab/>

1. **Jetty Context extern**

* Vom crea un director extern ce va contine fisierele corespunzatoare aplicatiei (in acest caz, directorul extern va fi **d:\Master\Semestrul 1\TPAJD\Teme\ContextExtern**)
* Se creaza fisierul “**context.xml**” in **$JETTY\_HOME/webapps**
* Fisierul are urmatorul continut:

<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE Configure PUBLIC "-//Mort Bay Consulting//DTD Configure//EN"

"http://jetty.mortbay.org/configure.dtd">

<Configure class="org.eclipse.jetty.webapp.WebAppContext">

<Set name="contextPath">/aliasJetty</Set>

<Set name="resourceBase"> d:\Master\Semestrul 1\TPAJD\Teme\ContextExtern</Set>

</Configure>

* Executare comanda java -jar start.jar -Djetty.port=8082 in %JETTY\_HOME
* Se acceseaza aplicatia la adresa <http://localhost:8082/aliasJetty/>

1. **Jetty Embedded**

* Adaugarea unor dependinte suplimentare in fisierul build.gradle

compile 'org.eclipse.jetty:jetty-server:9.4.3.v20170317',

'org.eclipse.jetty:jetty-servlet:9.4.3.v20170317'

* Daca avem fisiere JSP, este recomandat sa le transformam in Servlet-uri
* Implementarea unei noi clase Java



* Arhiva jar se construieste prin intermediul comenzii **gradle clean build**.
* Pentru a rula aplicatia trebuie sa rulam comanda **java -jar build/libs/JettyEmbedded.jar** din directorul aplicatiei.
* Aplicatia va fi disponibila la adresa [http://localhost:8082](http://localhost:80821)

1. **Wildfly WAR**

* Creare war folosind Gradle
* Copiere fisier war in **$JBOSS\_HOME/standalone/deployments**
* Din **$JBOSS\_HOME/bin** rulam comanda **standalone.bat -Djboss.http.port=8083**
* Se acceseaza aplicatia la adresa <http://localhost:8083/Servlet/>

1. **Glassfish WAR**

* Creare war folosind Gradle
* Copiere fisier war in **$GF\_HOME/glassfish/domains/domain1/autodeploy/**
* Din **$GF\_HOME/bin** executam comanda **asadmin** iar apoi **start-domain**
* Pentru oprirea serverului se executa comanda **stop-domain**
* Se acceseaza aplicatia la adresa <http://localhost:8084/Servlet/>