**MINISTERUL EDUСAŢIEI ŞI СERСETĂRII AL REPUBLIСII MOLDOVA**

**UNIVERSITATEA DE STAT „ALEСU RUSSO” DIN BĂLŢI**

**FAСULTATEA DE ŞTIINŢE REALE, EСONOMIСE ȘI ALE MEDIULUI**

**СATEDRA DE MATEMATIСĂ ȘI INFORMATIСĂ**

**LUCRARE DE LABORATOR NR. 11**

**„Spring Boot”**

**Curs „Platforma JAVA Enterprise II”**

**Coordonator:**

**Olga CERBU**

dr.,conf.univ

\_\_\_\_\_\_\_\_\_\_\_\_

(semnătura)

**Autor:**

Studentul grupei AW21M

**Dumitru BAZAOСHI**

\_\_\_\_\_\_\_\_\_\_\_\_

(semnătura)

**BĂLȚI, 2023**

**CUPRINS**

[Scop. Condiția problemei 2](#_Toc149377582)

[Ce este Spring? 2](#_Toc149377583)

[Caracteristicile Spring 2](#_Toc149377584)

[Crearea programului 3](#_Toc149377585)

[Concluzii 7](#_Toc149377586)

[Referințe bibliografice 8](#_Toc149377587)

Scop. Condiția problemei

Scopul acestei lucrări de laborator legată de Spring Boot este familiarizarea cu dezvoltarea rapidă a aplicațiilor Java folosind cadru (framework) Spring Boot. Aceasta implică înțelegerea conceptelor cheie și a funcționalităților oferite de Spring Boot pentru a simplifica și accelera dezvoltarea aplicațiilor.

De asemenea această lucrare de laborator urmărește să ofere o bază solidă de cunoștințe în dezvoltarea de aplicații cu Spring Boot, ceea ce va permite dezvoltarea rapidă și eficientă a aplicațiilor Java de înaltă calitate în viitor.

Ce este Spring Boot?

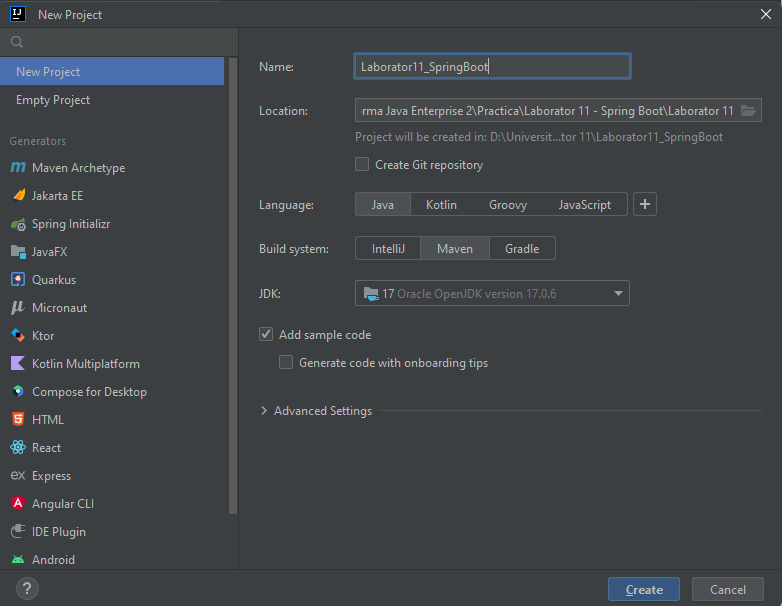
Spring Boot este un cadru (framework) pentru dezvoltarea rapidă a aplicațiilor în limbajul de programare Java. A fost dezvoltat ca o extensie a proiectului Spring Framework și își propune să simplifice dezvoltarea aplicațiilor Java, reducând la minimum configurația și configurarea manuală necesare. Spring Boot furnizează seturi predefinite de reguli și configurări care permit dezvoltatorilor să înceapă să dezvolte aplicații fără a trebui să petreacă mult timp configurând infrastructura.

Caracteristicile Spring Boot

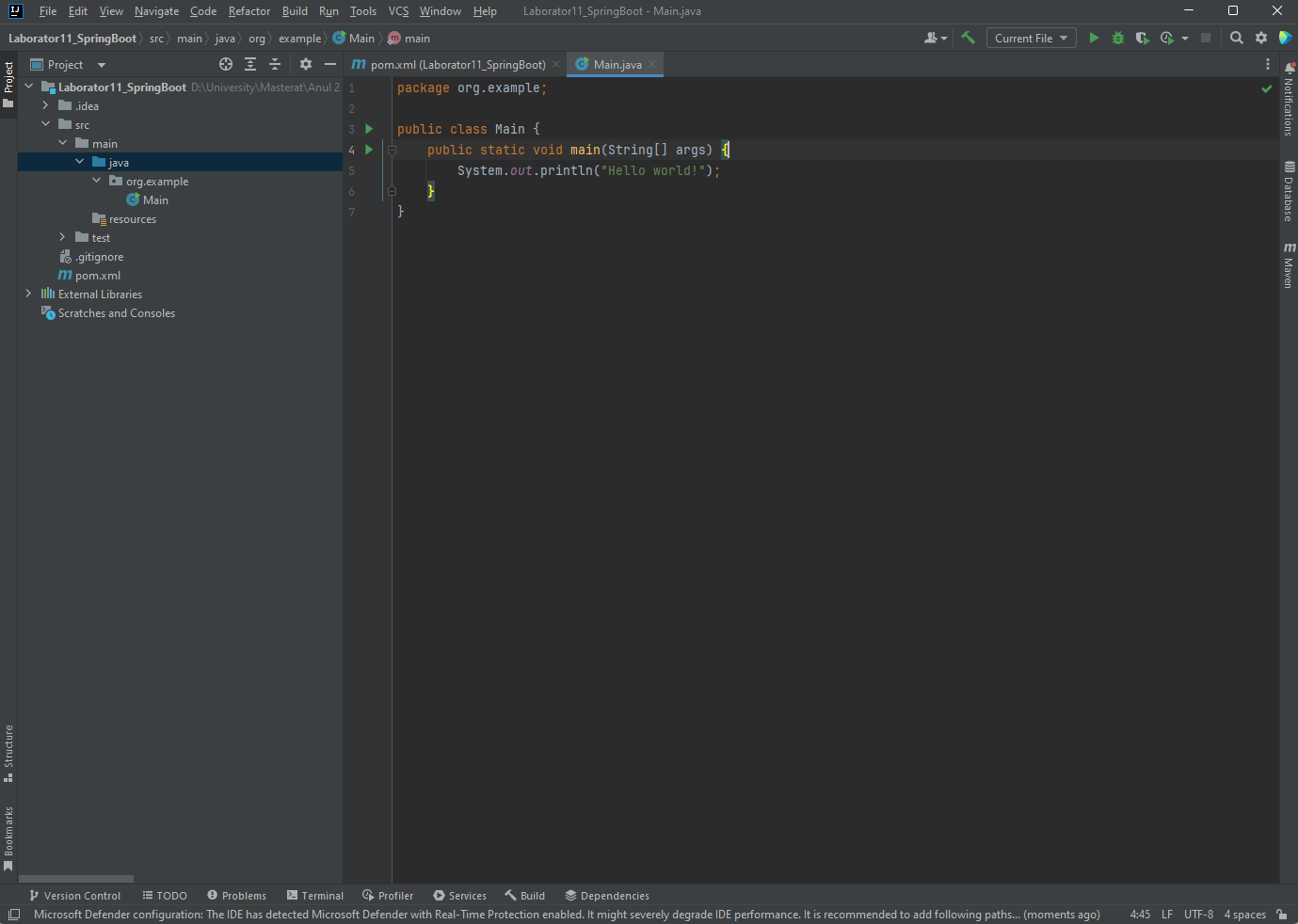
* **Opinii raționale**: Spring Boot promovează seturi de reguli și convenții pentru dezvoltare, dar permite și personalizarea acestora. Aceasta facilitează dezvoltatorilor să ia decizii în cunoștință de cauză, dar oferă și soluții implicite pentru o gamă largă de probleme.
* **Starter-uri** (Starters): Spring Boot oferă "starter-uri" care sunt pachete preconfigurate pentru diverse scopuri, cum ar fi dezvoltarea web, gestionarea datelor, securitatea și multe altele. Aceste starter-uri conțin dependințe și configurări necesare pentru a începe dezvoltarea rapid.
* **Autoconfigurare**: Spring Boot are capacitatea de a autoconfigura aplicația în funcție de bibliotecile și resursele disponibile în proiect. Cu alte cuvinte, mulți aspecte ale aplicației sunt configurate automat, astfel încât dezvoltatorii se pot concentra pe dezvoltarea funcționalității aplicației în loc de configurare.
* **Sistem încorporat de server**: Spring Boot include un server încorporat (de exemplu, Tomcat, Jetty sau Undertow), ceea ce înseamnă că nu este nevoie să instalați și să configurați manual un server web. Aplicația dvs. Java Spring Boot poate fi ambalată ca un fișier JAR independent care include serverul încorporat.
* **Suport pentru profile**: Spring Boot permite definirea de profile diferite pentru diverse medii de execuție (cum ar fi dezvoltare, testare sau producție) și gestionarea configurațiilor specifice fiecărui mediu.
* **Securitate**: Spring Boot oferă facilități pentru securitate, inclusiv autentificare și autorizare, prin intermediul modulului Spring Security.
* **Gestionarea dependințelor**: Spring Boot integrează gestionarea dependințelor folosind Apache Maven sau Gradle, ceea ce face ușor gestionarea bibliotecilor utilizate în aplicație.

Crearea programului

1. Trecem in mediul de programare (Intellij Idea)
2. Cream un proiect nou, Schimbăm numele, locația, limbajul ramâne **Java**, sistemul de build selectăm **Maven**. Click pe butonul **Create**.



La noi s-a deschis proiectul implicit creat.



1. Pentru a lucra cu Spring Boot trebuie să adăugăm dependințele necesare pentru aceasta. Trecem în pom.xml, unde adaugăm următoarele dependințe:

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>3.1.5</version>

<type>pom</type>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

<version>3.1.5</version>

</dependency>

</dependencies>

1. Facem build la proiect pentru a inițializa biblioteca adăugată. Build -> Build Project
2. Adăugam codul în clasa **Main**

package org.example;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

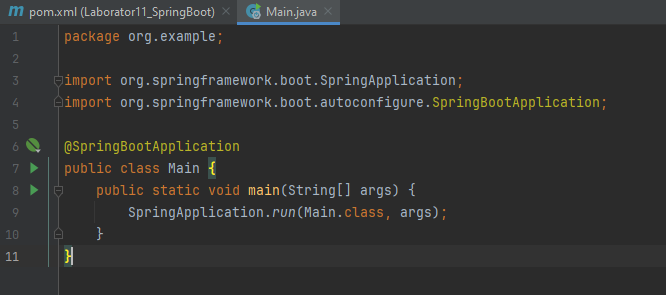
public class Main {

public static void main(String[] args) {

SpringApplication.run(Main.class, args);

}

}



1. Creăm clasa **MyController** și adăugam codul

package org.example;

import java.util.Date;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class MyController {

@RequestMapping("/")

public String home() {

return "test";

}

@RequestMapping("/hello")

public String hello() {

return "Hello";

}

@RequestMapping("numere")

public String numere() {

return "<h1>1</h1>" + "<h2>2</h2>" + "<h3>3</h3>";

}

@RequestMapping("timpul")

public String time() {

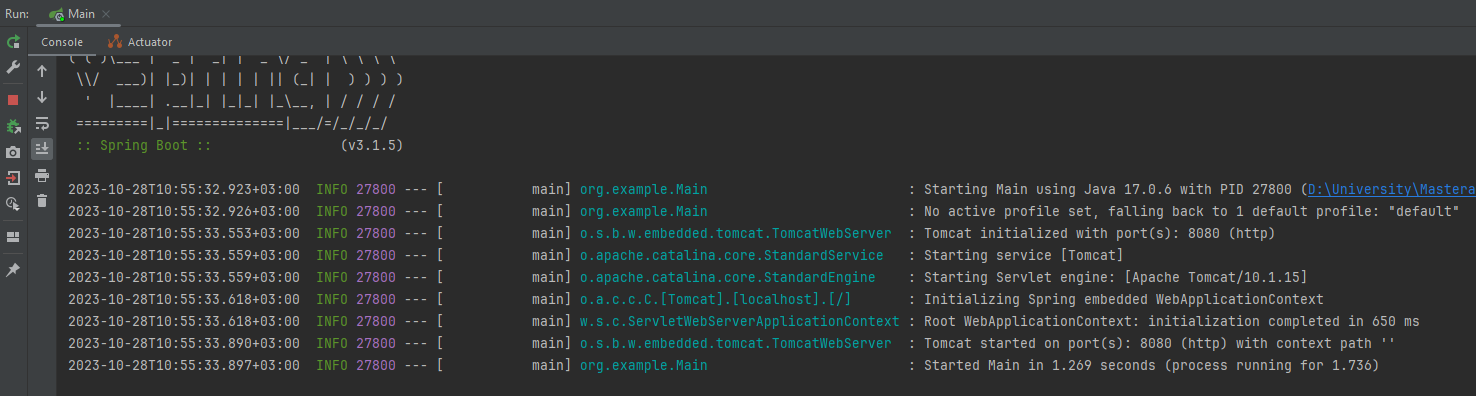
return "<h1>" + new Date() + "</h1>";

}

}

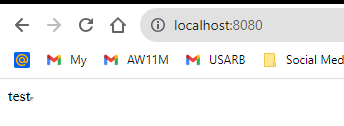


1. Rulăm proiectul.

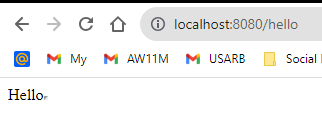


1. Observăm rezultatele conform adreselor noastre din clasa MyController în browser.

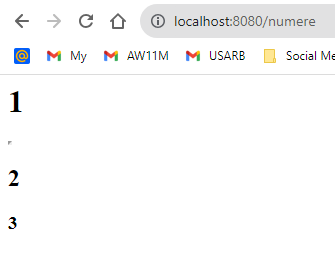
Link: <http://localhost:8080/>



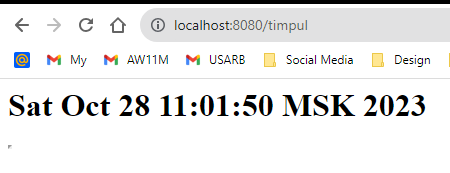
Link: <http://localhost:8080/hello>



Link: <http://localhost:8080/numere>



Link: <http://localhost:8080/timpul>



Concluzii

În concluzie, acest laborator a oferit o introducere solidă în lumea Spring Boot, un cadru puternic pentru dezvoltarea rapidă a aplicațiilor Java. Am învățat cum să folosim Spring Boot pentru a crea aplicații Java cu o configurare redusă și efort minim, permițându-ne să ne concentrăm asupra dezvoltării funcționalității și a cerințelor aplicației în sine.

De asemenea, am învățat despre autoconfigurarea Spring Boot, care face mult din configurarea grea pentru noi, astfel încât să nu mai pierdem timp prețios în această fază a dezvoltării.

În general, Spring Boot reprezintă o alegere excelentă pentru dezvoltarea de aplicații Java moderne, oferind ușurință de utilizare, performanță și un ecosistem bogat de resurse și comunitate activă. Cu cunoștințele obținute în acest laborator, suntem pregătiți să dezvoltăm aplicații Java de calitate într-un mod mai eficient și mai agil.

Referințe bibliografice

1. *Spring Framework* [online] [vizitat 27.10.2023]. Disponibil: <https://spring.io/projects/spring-framework>
2. *Core Technologies* [online] [vizitat 27.10.2023]. Disponibil: <https://docs.spring.io/spring-framework/reference/core.html>
3. *Introduction to Spring Framework* [online] [vizitat 28.10.2023]. Disponibil: <https://www.geeksforgeeks.org/introduction-to-jdbc/>
4. *Spring Boot*  [online] [vizitat 28.10.2023]. Disponibil: <https://spring.io/projects/spring-boot>