

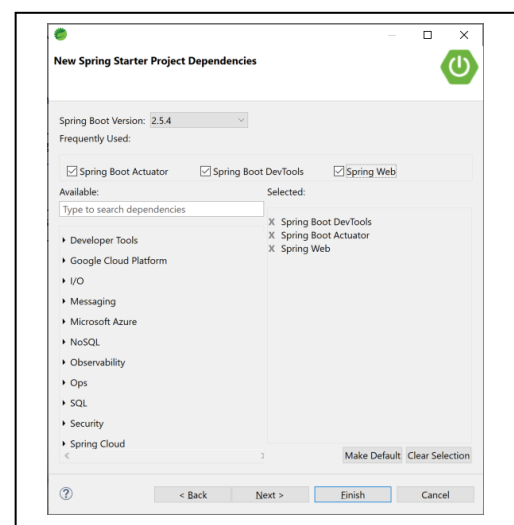
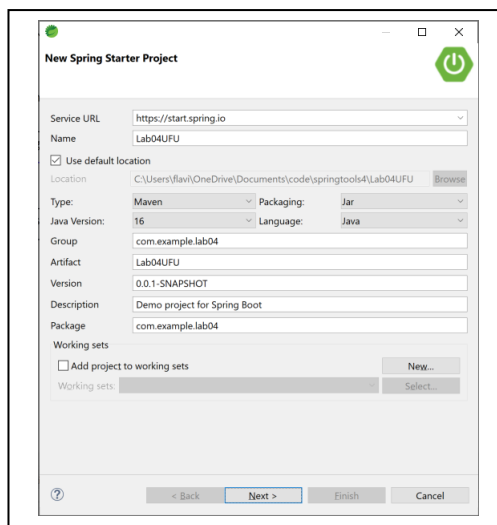
Programação para Internet

Bacharelado em Ciência da Computação – FACOM – UFU

Prof. Flávio de Oliveira Silva, Ph.D.

O objetivo deste laboratório é utilizar o Spring Boot para criar o lado servidor de uma aplicação web. O foco deste laboratório é o componente “Controller”.

1. Inicialmente é necessário preparar o ambiente de desenvolvimento:
 - a. Java JDK (<https://jdk.java.net/>). Instalar a versão JDK 17 ou 19.
 - b. Spring Tools for Eclipse (<https://spring.io/tools>). A última versão disponível do Spring Tool for Eclipse é a 4.17
2. Após instalar o Spring Tools, iniciar um novo projeto utilizando o “Spring Starter Project”, conforme indicado abaixo. Selecionar os componentes “DevTools”, “Web” e “Actuator”



3. Adicionar uma classe do tipo “Controller” que será responsável pelas tratar as requisições HTTP. Esta classe deverá ser chamada **com.example.lab04.LabController**. Adicionar na classe a anotação para *controller* (`@Controller`) e a anotação - `@RequestMapping(path="/lab4")` - para indicar este como o caminho relativo a todas as requisições que serão tratadas por este “controller”.

```
package com.example.lab04;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.ResponseBody;

@Controller
@RequestMapping(path="/lab4")
public class LabController {

}
```

```

package com.example.lab04;

public class User {
    private Long id;
    private String name;
    private String email;
    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    @Override
    public String toString() {
        return "User [id=" + id + ", name=" + name + ", email=" + email + ",
toString()=" + super.toString() + "];"
    }
}

```

4. Crie uma classe que representa um conceito do domínio da aplicação, por exemplo **User**

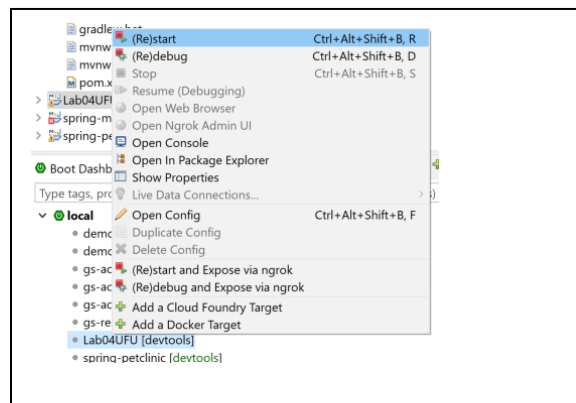
5. Adicionar na classe “Controller” um método um método **addUser**.

```
@GetMapping(path="/add") //
public @ResponseBody String addUser (@RequestParam String name, @RequestParam String
email) {
    User n = new User();
    n.setName(name);
    n.setEmail(email);
    return n.toString();
}
```

6. Na pasta **src/main/resources/static** adicione um arquivo HTML com o conteúdo abaixo:

```
<!DOCTYPE html>
<html>
<head>
    <title>LAB04/BCC/FACOM/UFU - Index</title>
</head>
<body>
<form action="/lab4/add" method="GET" id="nameForm">
    <div>
        <label for="nameField">Input User Data</label></br>
        <input name="name" id="nameField"></br>
        <input name="email" id="emailField">
        <button>Add User</button>
    </div>
</form>
</body>
</html>
```

7. Utilizando o Boot Dashboard, inicie a aplicação – (Re)Start - como mostrado na figura abaixo:



8. Em seu navegador verifique se o servidor iniciado no passo anterior está funcionando corretamente. Para isto informe a seguinte URL de serviço:

[http://localhost:8080/lab4/add?name="student"&email="student@ufu.br"](http://localhost:8080/lab4/add?name=)

Observe e anote a resposta.

9. O que acontece caso o atributo “action” do formulário por alterado para `action="/add"`? Altere o formulário HTML criado no passo 6 e anote o comportamento.
10. Proponha uma nova classe que represente um conceito do seu domínio de aplicação. Vamos considerar que o nome desta classe é: **CDomainClass**. Como feito no passo 4, crie a definição desta classe no pacote. Fique livre para definir os atributos desta classe da maneira mais conveniente.
11. Da mesma forma que feito no passo 5, crie um método chamado **addDomainObject** na classe “Controller”. Este método deve ser mapeado para a url **"/addDomainObject"**
12. **Crie um formulário HTML para informar os dados de um objeto qualquer da Classe CDomainClass. No formulário, no elemento <form> indique que o método que deve ser invocado é o addDomainObject criado no passo anterior.**
13. Realize o teste deste novo formulário e anote o resultado obtidos.
14. Altere o formulário HTML criado no passo 6. No elemento <form> indique que o método que deve ser invocado é: `action="/lab4/addDomainObject"`. Utilizando o formulário do Passo 6 alterado, clique no botão “Add User” e veja o que ocorre. Anote a resposta e explique a razão do comportamento ocorrido.
15. Exporte o projeto para um arquivo .zip e prepare um documento com todas as respostas solicitadas. Para exportar utilize a opção: **File → Export ... → General - Archive File**
16. Faça o upload deste arquivo .ZIP para uma pasta no **OneDrive** ou **GoogleDrive**. E gere um link de compartilhamento.
17. Envie sua resposta na Atividade do TEAMS – **LAB04-SpringBoot** e na resposta coloque o link da pasta compartilhada que foi criada e que contém o resultado das atividades práticas. A tarefa deve ser encerrada até o final do próximo dia conforme indicado no MS TEAMS.