Aula 3

Desenvolvimento *Web Back-End*

Profª Luciane Yanase Hirabara Kanashiro

1

Conversa Inicial

2

- **Nesta aula veremos o desenvolvimento Java web com Spring MVC e seus conceitos fundamentais**

3

- **Camada de Visão**
- **Thymeleaf**
- **HTML**
- **Front Controller**
- **Camada de Controle**

4

Camada de visão

5

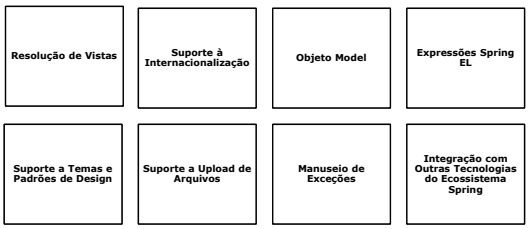
Camada de Visão

- **Interface gráfica e interação com o usuário**
- **Integração com JSP e Thymeleaf**

Fonte: https://spring.io/trademarks

6

Características



7

Thymeleaf

8

- **Código aberto criado em 2011 por Daniel Fernández**
- **Ferramenta para desenvolvimento de aplicações web Java**
- **Sintaxe amigável semelhante a HTML puro**
- **Linguagem incoporada em páginas html**
- **Compreensível por designers e desenvolvedores**

Fonte : Thymeleaf.org

9

Características e funcionalidades

10

HTML

11

- **Hypertext Markup Language: páginas web**
- **Marcações: estrutura e conteúdo da página**
  - **Texto**
  - **Imagens**
  - **Links**
  - **Formulários**
- **HTML5: versão mais atualizada**
- **Novos recursos: suporte à multimídia, novas tags e elementos**

12

```html
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Minha Página HTML</title>
</head>
<body>
  <header>
    <h1>Minha Página HTML</h1>
  </header>
  <div>
    <p>Olá, mundo! Esta é uma página HTML simples.</p>
  </div>
</body>
</html>
```

Minha Página HTML

Olá, mundo! Esta é uma página HTML simples.

13

HTML e *Thymeleaf*

- **Integração com Spring MVC**
- **Sintaxe Thymeleaf**
  - **Exemplo**

```html
<p> Olá, <span th:text="${usuario.nome}">Visitante</span> !
</p>
```

14

- **Expressões e atributos Thymeleaf**
- **Processamento no lado do servidor**
- **Ferramentas para desenvolvimento**
- **Facilidade de integração com outras tecnologias**

15

```html
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <title>Formulário de Usuário</title>
</head>
<body>

  <h2>Formulário de Usuário</h2>
  <form th:object="${usuario}" th:action="@{/salvar}" method="post">
    <label for="nome">Nome:</label>
    <input type="text" id="nome" th:field="*{nome}" /> <br/>
    <label for="email">Email:</label>
    <input type="text" id="email" th:field="*{email}" /> <br/>
    <button type="submit">Salvar</button>
  </form>

</body>
</html>
```

```java
public class Usuario {
  private String nome;
  private String email;

  // getters e setters
}
```

16

Comando e significado

Comando	Significado
th:object="${usuario}"	Define o objeto de dados (Usuario) associado ao formulário
th:action="@{/salvar}"	Especifica a URL para onde o formulário será enviado ao ser submetido Isso normalmente corresponderia a um controlador Spring
th:field="*{nome}"	Vincula o campo de entrada aos atributos nome do objeto Usuario
th:field="*{email}"	Vincula o campo de entrada aos atributos email do objeto Usuario
type="submit"	Indica que este é um botão de envio do formulário

17

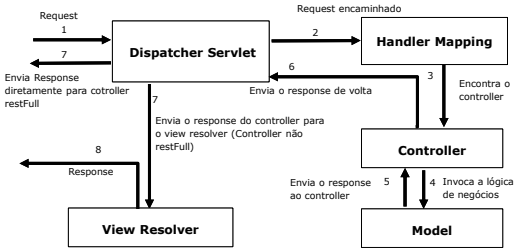
Front Controller

18

- Padrão de design arquitetural
- Componente centralizado
  - DispatcherServlet
- Responsabilidade
  - Receber requisições HTTP e encaminhá-las para os controllers
- Primeiro ponto de contato para requisições

19

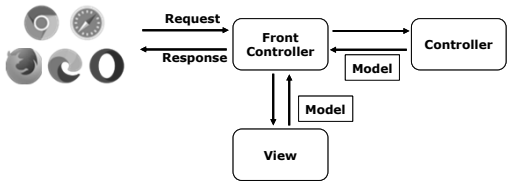
Diagrama de Fluxo – Spring MVC



20

Modelo MVC do Spring MVC

http://localhost:8080/palavra/lista ⟶ /palavra/lista



21

Camada de Controle

22

- "Controlador"
- Responsabilidade
  - Receber as requisições
  - Processar
  - Coordenar a resposta
- Separa a View e a Model

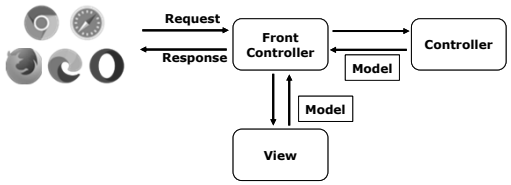
23

- Receber requisições
- Chamar serviços e lógica de negócios
- Preparar dados para a view
- Selecionar a view adequada
- Responder ao cliente

24

Modelo MVC do Spring MVC

http://localhost:8080/palavra/lista ⟶ /palavra/lista



25

Exemplo

```
@Controller
@RequestMapping("/exemplo")
public class ExemploController {

  @Autowired
  private ExemploService exemploService;

  @GetMapping("/pagina")
  public String exibirPagina(Model model) {
    List<Exemplo> exemplos = exemploService.obterTodosExemplos();
    model.addAttribute("exemplos", exemplos);
    return "pagina";
  }

  @PostMapping("/salvar")
  public String salvarExemplo(@ModelAttribute("exemplo") Exemplo exemplo) {
    exemploService.salvarExemplo(exemplo);
    return "redirect:/exemplo/pagina";
  }
}
```

26