



Certified Tech
Developer

The Ultimate Degree

Spring MCV com Spring Boot + Thymeleaf

Como vimos anteriormente, o objetivo do padrão MVC é separar a view, da lógica da aplicação, para facilitar a reutilização de código e simplificar o desenvolvimento de ambas as camadas separadamente, a fim de permitir alterar a view ou criar aplicações com múltiplas views.

Criar uma aplicação Spring MVC com Spring Boot

Para criar um projeto **Spring Boot** com o objetivo de usar o **Spring MVC**, simplesmente temos que adicionar o **spring-boot-starter-web** como dependência em nosso projeto e a dependência para o motor de telas que desejamos utilizar, pois não queremos fazer uso das típicas jsp. Neste exemplo, vamos utilizar o **Thymeleaf (spring-boot-starter-thymeleaf)** na parte da visualização (View), o que nos permite ter uma integração completa com **Spring** e uma sintaxe mais próxima do **HTML**, porque em vez de adicionar novas tags, o que ele faz é adicionar novos atributos as tags **HTML5**.

Nos métodos a seguir, podemos dar um exemplo de como são utilizados e verificar se todos produzirão o mesmo resultado: enviar para a view << list >> o nome da aplicação e uma lista de produtos.

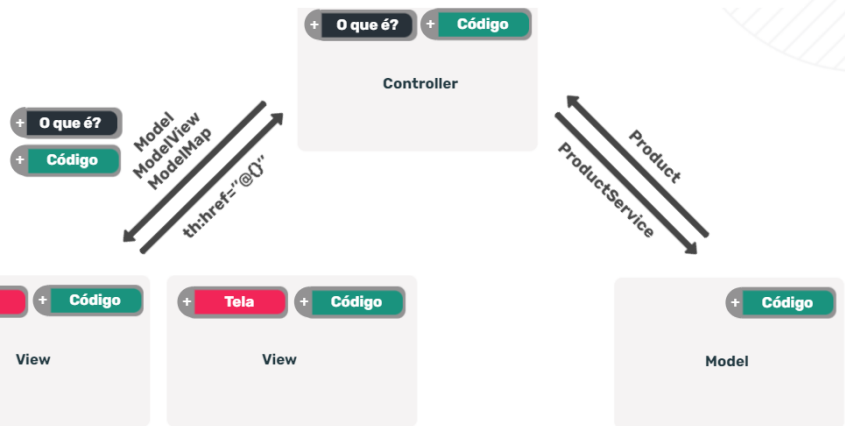
Implementação do Spring Boot MVC com Thymeleaf

Nosso exemplo de controller tem duas anotações na classe: `@Controller`, para que o Spring saiba que é um controller, e `@RequestMapping("product")`. Esta segunda anotação é a recomendada, pois todas as URLs do mesmo controller começam igual, e precisamos identificar qual controller estamos chamando. Desta forma, evitamos ter que colocar essa primeira parte da URL em cada um dos métodos, podendo ser colocada em uma anotação do controller. Portanto, a URL para o método que irá exibir a lista de produtos seria <http://localhost:9876/myapp/products/list>.

Para enviar informações do controller para a view.

o Spring nos fornece o **Model** (uma interface),
ModelMap (uma classe) e

o **ModelAndView** (que reúne em um único
objeto o Model e o ModelMap).



Representa la lista de productos en una tabla. En el código podemos observar los atributos de Thymeleaf, como `th:replace`, `th:each` y `th:text`.

Lista de productos			
Id	Código	Nome	Preço

[Adicionar Produto](#)

Novo Produto

Código

Nome

Preço

Model ModelAndView:

```
@GetMapping(value = "/list")
public String list(Model model) {
    model.addAttribute("title", applicationName);
    model.addAttribute("products", productService.getAllProducts());
    return "list";
}

@GetMapping(value = "/listModelMap")
public String listModelMap(ModelMap model) {
    model.addAttribute("title", applicationName);
    model.addAttribute("products", productService.getAllProducts());
    return "list";
}

@GetMapping(value = "/listModelAndView")
public ModelAndView listModelAndView() {
    ModelAndView mav = new ModelAndView();
    mav.addObject("title", applicationName);
    mav.addObject("products", productService.getAllProducts());
    mav.setViewName("list");
    return mav;
}
```

Controller:

```
@Controller
@RequestMapping("products")
public class ProductController {

    public static final String VIEW_LIST = "list";
    public static final String VIEW_FORM = "formulario";

    @Value("${application.name}")
    private String applicationName;

    @Autowired
    private ProductService productService;

    @GetMapping(value = "/list")
    public String list(Model model) {
        model.addAttribute("title", applicationName);
        model.addAttribute("products", productService.getAllProducts());
        return VIEW_LIST;
    }

    @GetMapping(value = "/listModelMap")
    public String listModelMap(ModelMap model) {
        model.addAttribute("title", applicationName);
        model.addAttribute("products", productService.getAllProducts());
        return VIEW_LIST;
    }

    @GetMapping("/listModelAndView")
    public ModelAndView listModelAndView() {
        ModelAndView mav = new ModelAndView();
        mav.addObject("title", applicationName);
        mav.addObject("products", productService.getAllProducts());
        mav.setViewName(VIEW_LIST);
        return mav;
    }

    @GetMapping("/create")
    public String create(Map model) {
        Product product = new Product();
        model.put("product", product);
        model.put("title", applicationName);
        return VIEW_FORM;
    }

    @PostMapping("/save")
    public String save(Product product) {
```

```

        productService.save(product);
        return "redirect:" + VIEW_LIST;
    }

    @GetMapping("/remove/{id}")
    public String remove(@PathVariable(value="id") Integer idProduct) {
        productService.remove(idProduct);
        return "redirect:../" + VIEW_LIST;
    }
}

```

View:

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head th:replace="layout/layout::head">
    <meta charset="UTF-8">
</head>
<body>
    <header th:replace="layout/layout::header"></header>

    <div class="container">
        <div class="card bg-dark">
            <div class="card-header">
                <h3>Lista de produtos</h3>
            </div>
            <div class="card-body">
                <table class="table table-striped">
                    <thead>
                        <tr>
                            <th>Id</th>
                            <th>Código</th>
                            <th>Nome</th>
                            <th>Preço</th>
                            <th></th>
                        </tr>
                    </thead>
                    <tbody>
                        <tr th:each="product: ${products}">
                            <td th:text="${product.id}"></td>
                            <td th:text="${product.code}"></td>
                            <td th:text="${product.name}"></td>
                            <td th:text="${product.price}"></td>
                            <td><a th:href="@{remove/} +
${product.id}" th:text="'Eliminar'" class="btn btn-outline-
danger"></a></td>
                        </tr>
                    </tbody>
                </table>
            </div>
        </div>
    </div>

```

```

        <tfoot>
        <tr>
            <td colspan="5">
                <a th:href="@{create}" class="btn btn-
outline-primary btn-block">Adicionar Produto</a>
            </td>
        </tr>
        </tfoot>
    </table>
</div>
</div>
</div>

<footer th:replace="layout/layout::footer" class="bg-dark"></footer>
</body>
</html>

```

View:

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head th:replace="layout/layout::head">
    <meta charset="UTF-8">
</head>
<body>
    <header th:replace="layout/layout::header"></header>

    <div class="container">
        <div class="card bg-dark">
            <div class="card-header">
                <h3>Novo produto</h3>
            </div>
            <div class="card-body">
                <form th:action="@{save}" th:object="${produto}" method="post">
                    <!-- <input type="hidden" th:field="${product.id}" -->
                    <input type="hidden" th:field="**{id}" />

                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Código</label>
                        <div class="col-sm-6">
                            <!-- <input type="text" th:field="${product.name}" -->
                            <input type="text" th:field="**{name}">
                        </div>
                    </div>
                    <div class="form-group row">
                        <label class="col-sm-2 col-form-label">Nome</label>
                        <div class="col-sm-6">

```

```

        <!-- <input type="text"
th:field="${produto.codigo}"> -->
        <input type="text" th:field="*{codigo}">
    </div>
</div>
<div class="form-group row">
    <label class="col-sm-2 col-form-label">Preço</label>
    <div class="col-sm-6">
        <!-- <input type="text" th:field="${product.price}">
-->
        <input type="text" th:field="*{price}">
    </div>
</div>
<div class="form-group">
    <div class="col-sm-6">
        <input type="submit" value="Adicionar
produto" class="btn btn-outline-primary">
        <a th:href="@{list}" class="btn btn-outline-
danger">Cancelar</a>
    </div>
</div>
</form>
</div>
</div>
</div>
    <footer th:replace="layout/layout::footer" class="bg-dark"></footer>
</body>
</html>

```

Product Model:

```
public class Product{
    private int id;
    private String code;
    private String name;
    private double price;

    public Product(int id, String code, String name, double price){
        this.id = id;
        this.code = code;
        this.name = name;
        this.price = price;
    }

    public int getId(){
        return id;
    }
    public String getCode(){
        return code;
    }
    public String getName(){
        return name;
    }
    public double getPrice(){
        return price;
    }
    public void setId(int id){
        this.id = id;
    }
    public void setCode(String code){
        this.code = code;
    }
    public void setName(String name){
        this.name = name;
    }
    public void setPrice(double price){
        this.price = price;
    }
}
```