

Construindo uma API REST com Java, Spring Boot, Spring Data JPA, Spring Web, Lombok e Validation.

O objetivo desse documento é informar ao leitor de como está estruturado a construção dessa API REST. Ademais, o usuário poderá consumir os seguintes endpoints:

EndPoints	Verbo	Descrição
/encomenda	GET	Lista todas as encomendas
/encomenda/{id}	GET	Retorna uma encomenda especifica
/encomenda	POST	Adiciona uma nova encomenda
/encomenda/{id}	PUT	Atualiza uma encomenda
/encomenda/{id}	DELETE	Exclui uma encomenda

A API consome uma outra API que é a ViaCEP sendo utilizada para pesquisar as informações relacionadas ao cep fornecido.

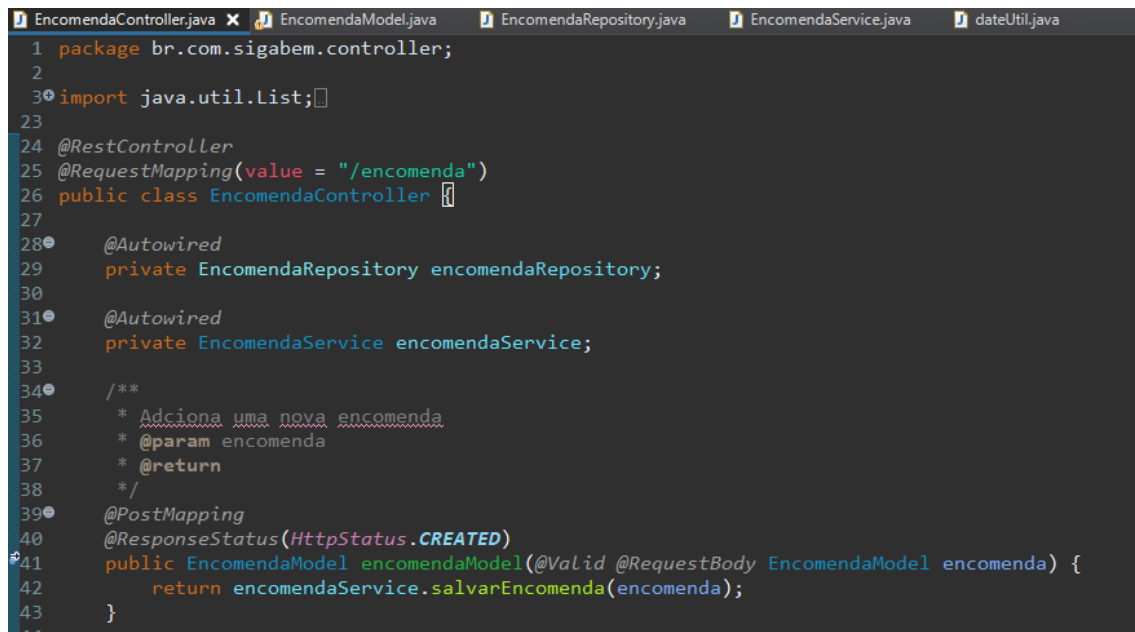
1. Tecnologias Aplicadas

1.1. Spring Boot

Facilita na configuração do ambiente de desenvolvimento tendo como base o padrão MVC, em outras palavras, aumenta a produtividade.

1.2. Spring Web

Fornecer suporte aos protocolos HTTP sendo necessários para as requisições da API. Como exemplo temos a imagem 1 na linha 39 a utilização do verbo POST.



```

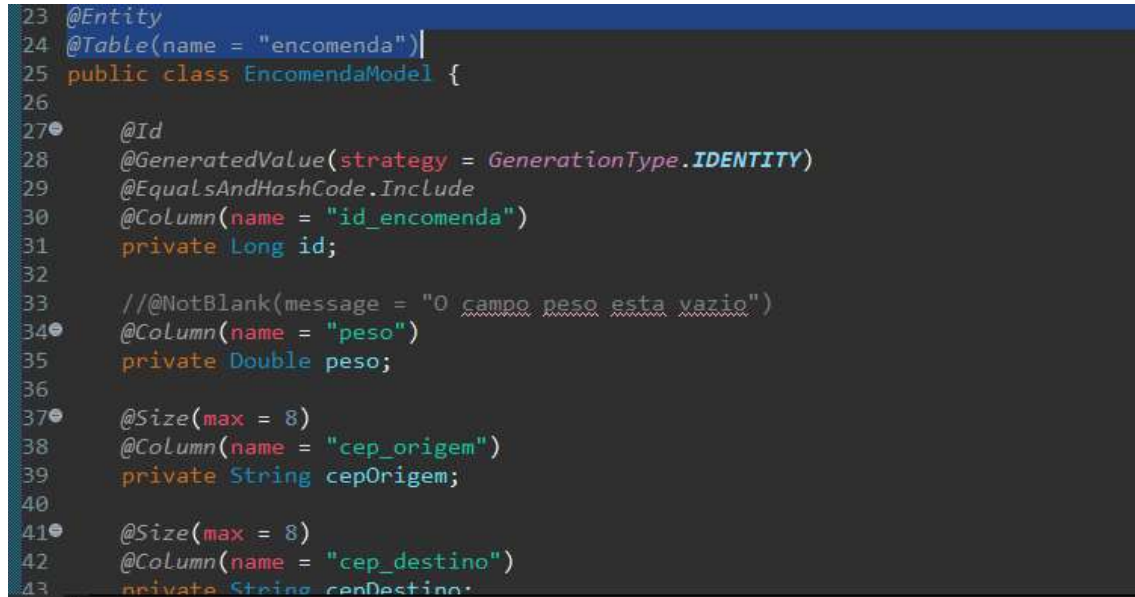
1 package br.com.sigabem.controller;
2
3 import java.util.List;
4
5 @RestController
6 @RequestMapping(value = "/encomenda")
7 public class EncomendaController {
8
9     @Autowired
10    private EncomendaRepository encomendaRepository;
11
12    @Autowired
13    private EncomendaService encomendaService;
14
15    /**
16     * Adiciona uma nova encomenda
17     * @param encomenda
18     * @return
19     */
20    @PostMapping
21    @ResponseStatus(HttpStatus.CREATED)
22    public EncomendaModel salvarEncomenda(@Valid @RequestBody EncomendaModel encomenda) {
23        return encomendaService.salvarEncomenda(encomenda);
24    }
25
26 }

```

Imagem 1: Exemplo do Spring Web;

1.3. Spring Data JPA

Poderosa ferramenta que facilita a implementação de CRUD. Segue a imagem utilizando anotação para realizar o mapeamento com o banco de dados.



```

23 @Entity
24 @Table(name = "encomenda")
25 public class EncomendaModel {
26
27     @Id
28     @GeneratedValue(strategy = GenerationType.IDENTITY)
29     @EqualsAndHashCode.Include
30     @Column(name = "id_encomenda")
31     private Long id;
32
33     // @NotBlank(message = "O campo peso esta vazio")
34     @Column(name = "peso")
35     private Double peso;
36
37     @Size(max = 8)
38     @Column(name = "cep_origem")
39     private String cepOrigem;
40
41     @Size(max = 8)
42     @Column(name = "cep_destino")
43     private String cepDestino;
44 }

```

Imagem 2: Exemplo da utilização do Spring Data JPA

1.4. MySQL

O SGBD utilizando no processo de desenvolvimento foi o MySQL por ser amplamente adotado e ser FREE.

1.5. Lombok

Ferramenta de aumento de produtividade por meio de anotações de diminuem a reprodução de linhas de código deixando o mesmo mais limpo.

```
12
13 import lombok.AllArgsConstructor;
14 import lombok.EqualsAndHashCode;
15 import lombok.Getter;
16 import lombok.NoArgsConstructor;
17 import lombok.Setter;
18
19 @Getter
20 @Setter
21 @AllArgsConstructor
22 @NoArgsConstructor
```

Imagem 3: Utilização do Lombok;

2. Estrutura do pacote

Seguindo o padrão MVC onde separamos por camadas sendo que cada uma tem responsabilidades independentes.

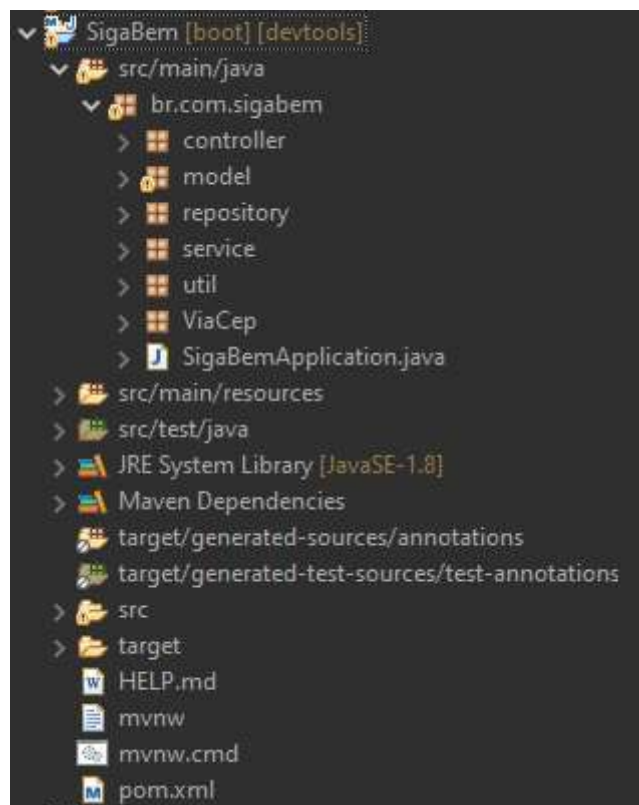


Imagem 4: Estrutura do projeto com Spring Web

3. Teste

Os testes foram realizados utilizando a ferramenta Postman. Segue algumas imagens.

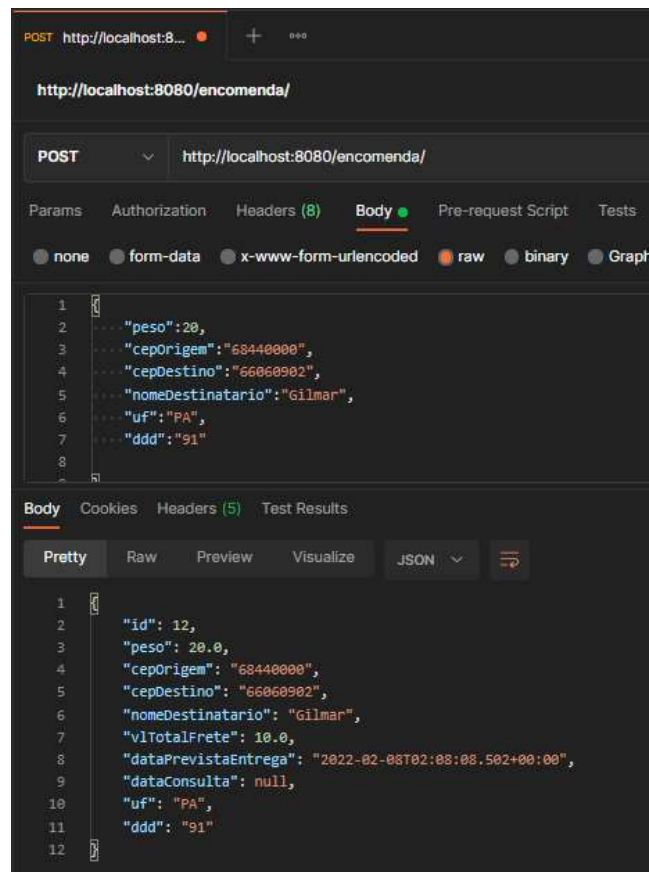


Imagem 5: Teste de criação de uma nova encomenda

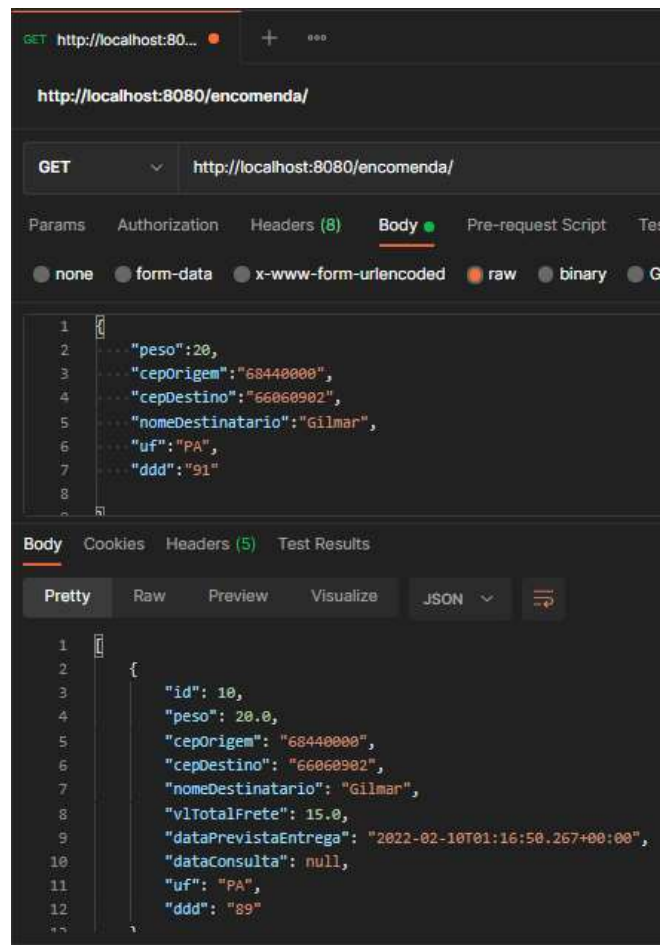


Imagem 6: Teste de busca das encomendas