

```
1 package br.com.padon.application.controllers;
2
3 import br.com.padon.application.models.*;
4 import br.com.padon.application.repositorys.
    ConterRepository;
5 import br.com.padon.application.repositorys.LojaRepository
    ;
6 import br.com.padon.application.repositorys.
    TrabalhaRepository;
7 import com.fasterxml.jackson.databind.JsonNode;
8 import org.springframework.beans.factory.annotation.
    Autowired;
9 import org.springframework.web.bind.annotation.*;
10
11 import java.util.List;
12 import java.util.Optional;
13
14 @RestController
15 @CrossOrigin
16 @RequestMapping("/loja")
17 public class LojaController {
18
19     @Autowired
20     private LojaRepository loja;
21     @Autowired
22     private TrabalhaRepository trabalha;
23     @Autowired
24     private ConterRepository conter;
25
26     @PostMapping("/create")
27     public Loja createLoja(@RequestBody JsonNode node) {
28         return loja.save(new Loja(
29             node.get("nome").asText(),
30             node.get("endereco").asText()
31         ));
32     }
33
34     @GetMapping("/get")
35     public List<Loja> getAllLojas() {
36         return loja.findAll();
37     }
38
39     @PostMapping("/save")
40     public Loja saveLoja(@RequestBody JsonNode node) {
```

```

41         Loja lojaById = loja.findById(node.get("id").asInt
    ()).orElseThrow();
42         lojaById.setNome(node.get("nome").asText());
43         lojaById.setEndereco(node.get("endereco").asText
    ());
44         return loja.save(lojaById);
45     }
46
47     @PostMapping("/byid")
48     public Optional<Loja> getLoja(@RequestBody JsonNode
    node) {
49         return loja.findById(node.get("id").asInt());
50     }
51
52     @PostMapping("/delete")
53     public boolean deleteLoja(@RequestBody JsonNode node
    ) {
54         loja.deleteById(node.get("id").asInt());
55         return true;
56     }
57
58     @PostMapping("/addfuncionario")
59     public Trabalha addFuncionario(@RequestBody JsonNode
    node) {
60         return trabalha.save(new Trabalha(node.get("lojaId
    ").asInt(), node.get("cpf").asText()));
61     }
62
63     @PostMapping("/removefuncionario")
64     public boolean removeFuncionario(@RequestBody JsonNode
    node) {
65         Trabalha result = trabalha.getTrabalha(node.get("
    lojaId").asInt(), node.get("cpf").asText());
66         if (result == null) {
67             return false;
68         }
69
70         trabalha.delete(result);
71         return true;
72     }
73
74     @PostMapping("/addproduto")
75     public Conter addProduto(@RequestBody JsonNode node) {
76         return conter.save(new Conter(

```

```
77         node.get("produtoId").asInt(),
78         node.get("lojaId").asInt(),
79         node.get("estoque").asInt(),
80         node.get("quantidadeMinima").asInt()
81     ));
82     }
83
84     @PostMapping("/removeproduto")
85     public boolean removeProduto(@RequestBody JsonNode
86     node) {
87         Conter result = conter.getConter(node.get("
88         produtoId").asInt(), node.get("lojaId").asInt());
89         if (result == null) {
90             return false;
91         }
92         conter.delete(result);
93         return true;
94     }
95 }
```