

Introdução ao AngularJS

Professor Anderson Henrique



Alterando o (ChamadoController)

No método **listChamados**, vamos **apagar** todos os chamados (**fakes**) que criamos manualmente, e vamos alterar a implementação desse método utilizando o **ChamadoDAO**

```
25  @GET
26  @Produces(MediaType.APPLICATION_JSON)
27  @Path("/")
28  public List<Chamado> listChamados() {
29      try {
30          ChamadoDAO chamadoDAO = new ChamadoDAO();
31          return chamadoDAO.listarTudo();
32      } catch (SQLException | ClassNotFoundException e) {
33          Logger.getLogger(ChamadoController.class.getName())
34              .log(Level.SEVERE, null, e);
35          throw new WebApplicationException(Response.Status.INTERNAL_SERVER_ERROR);
36      }
37  }
```

Tente criar uma instância da classe **ChamadoDAO** e retornar uma **lista** de **chamados** (lembrando que o construtor dessa classe já faz uma conexão com o banco de dados), CASO ocorra exceção gere um **log** e armazene no servidor Tomcat e nosso **jersey** vai lançar uma **WebApplicationException** como **resposta** a essa exceção

No método **getChamado** vamos apagar o chamado (**fake**) que criamos manualmente e vamos implementar utilizando a classe **ChamadoDAO**

```

38     @GET
39     @Produces(MediaType.APPLICATION_JSON)
40     @Path("{id}/")
41     public Chamado getChamado(@PathParam("id") long id) {
42         try {
43             ChamadoDAO chamadoDAO = new ChamadoDAO();
44             return chamadoDAO.listar(id);
45         } catch (SQLException | ClassNotFoundException e) {
46             Logger.getLogger(ChamadoController.class.getName())
47                 .log(Level.SEVERE, null, e);
48             throw new WebApplicationException(Response.Status.INTERNAL_SERVER_ERROR);
49         }
50     }

```

No método **create** vamos apagar nosso return (fake) e utilizar o **ChamadoDAO**

```

53 @POST
54 @Consumes(MediaType.APPLICATION_JSON)
55 @Path("/")
56 public Response create(Chamado chamado) {
57     try {
58         chamado.setStatus(Status.NOVO);
59         ChamadoDAO chamadoDAO = new ChamadoDAO();
60         chamadoDAO.inserir(chamado);
61         return Response.status(Response.Status.OK).build();
62     } catch (SQLException | ClassNotFoundException e) {
63         Logger.getLogger(ChamadoController.class.getName())
64             .log(Level.SEVERE, null, e);
65         throw new WebApplicationException(Response.Status.INTERNAL_SERVER_ERROR);
66     }
67 }

```

No método **update** vamos apagar o nosso código de resposta (**fake**) e vamos implementar de acordo com o modelo **DAO**

```

69  @PUT
70  @Consumes(MediaType.APPLICATION_JSON)
71  @Path("/")
72  public Response update(Chamado chamado) {
73      try {
74          chamado.setStatus(Status.PENDENTE);
75          ChamadoDAO chamadoDAO = new ChamadoDAO();
76          chamadoDAO.alterar(chamado);
77          return Response.status(Response.Status.OK).build();
78      } catch (SQLException | ClassNotFoundException e) {
79          Logger.getLogger(ChamadoController.class.getName())
80              .log(Level.SEVERE, null, e);
81          throw new WebApplicationException(Response.Status.INTERNAL_SERVER_ERROR);
82      }
83  }

```

No método **delete** vamos apagar o nosso código de resposta (**fake**) e vamos implementar de acordo com o modelo **DAO**

```

85 @DELETE
86 @Path("/{id}/")
87 public Response delete(@PathParam("id") long id) {
88     try {
89         ChamadoDAO chamadoDAO = new ChamadoDAO();
90         chamadoDAO.excluir(id);
91         return Response.status(Response.Status.OK).build();
92     } catch (SQLException | ClassNotFoundException e) {
93         Logger.getLogger(ChamadoController.class.getName())
94             .log(Level.SEVERE, null, e);
95         throw new WebApplicationException(Response.Status.INTERNAL_SERVER_ERROR);
96     }
97 }

```

Pronto! Concluimos todos os métodos? (Ainda falta implementação do método **concluir()**)

```

99      @PUT
100      @Path("/{id}/")
101      public Response concluir(@PathParam("id") long id) {
102          try {
103              ChamadoDAO chamadoDAO = new ChamadoDAO();
104
105              Chamado c = chamadoDAO.listar(id);
106              c.setStatus(Status.FECHADO);
107
108              chamadoDAO.alterar(c);
109              return Response.status(Response.Status.OK).build();
110          } catch (SQLException | ClassNotFoundException e) {
111              Logger.getLogger(ChamadoController.class.getName())
112                  .log(Level.SEVERE, null, e);
113              throw new WebApplicationException(Response.Status.INTERNAL_SERVER_ERROR);
114          }
115      }

```

Com esse método alteramos o status do chamado para FECHADO e atualizamos na tabela

Agora só precisamos trabalhar com o Controller do Angularjs e na nossa Tela **index.html**

Vamos adicionar uma **coluna** na nossa tabela que é o status, ele deverá ser mostrado para o cliente, e o botão **concluir** só deverá ser **mostrado** SE o **status** do chamado não for **FECHADO**

```
<table border="1">
  <tr>
    <th>ID</th>
    <th>Assunto</th>
    <th>Mensagem</th>
    <th>Status</th>
    <th>Opções</th>
  </tr>
  <tr ng-repeat="chamado in controller.chamados">
    <td>{{chamado.id}}</td>
    <td>{{chamado.assunto}}</td>
    <td>{{chamado.mensagem}}</td>
    <td>{{chamado.status}}</td>
```

```
50 <button type="button"  
51     ng-click="controller.concluir(chamado)"  
52     ng-show="chamado.status !== 'FECHADO'">Concluir</button>
```

E também precisamos mudar no nosso controller do angularjs, no arquivo **api.js** a implementação do método **concluir**

```
52 self.concluir = function(chamado) {
53     self.chamado = chamado;
54
55     $http({
56         method: 'PUT',
57         url: urlBase + 'chamados/' + self.chamado.id + '/'
58     }).then(function successCallback(response) {
59         self.atualizarTabela();
60     }, function errorCallback(response) {
61         self.ocorreuErro();
62     });
63 }
```

Pronto! Agora vamos testar o nosso projeto inicializando o servidor Tomcat (Limpar e Construir) o projeto e depois executar

Dúvidas?

Professor Anderson Henrique



Obrigado pela atenção de todos vocês...

Professor Anderson Henrique

