03

Refatorando todas as acoes

Transcrição

Olá! Agora vamos ajustar as outras ações do nosso código. Se você já fez isso sozinho, pode até pular esse vídeo, já que não faremos nada diferente do que fizemos no vídeo anterior.

Cada ação precisa devolver um nome:

```
} else if(paramAcao.equals("MostraEmpresa")) {

    MostraEmpresa acao = new MostraEmpresa();
    nome = acao.executa(request, response);

} else if (paramAcao.equals("AlteraEmpresa")) {

    AlteraEmpresa acao = new AlteraEmpresa();
    nome = acao.executa(request, response);

} else if (paramAcao.equals("NovaEmpresa")) {

    NovaEmpresa acao = new NovaEmpresa();
    nome = acao.executa(request, response);
}
```

COPIAR CÓDIGO

Esse código não irá compilar, pois ainda precisaremos alterar as respectivas classes.

Agora a classe `AlteraEmpresa` precisa nos devolver uma `String`. Essa `String` deve nos definir, nesse caso, um `redirect` para o endereço `entrada?acao=ListaEmpresas`:

```
public class AlteraEmpresa {

    public String executa(HttpServletRequest request, HttpServ

        System.out.println("Alterando empresa");

        String nomeEmpresa = request.getParameter("nome");
        String paramDataEmpresa = request.getParameter("data")
        String paramId = request.getParameter("id");
        Integer id = Integer.valueOf(paramId);

        Date dataAbertura = null;
        try {
            SimpleDateFormat sdf = new SimpleDateFormat("dd/MM,
            dataAbertura = sdf.parse(paramDataEmpresa);
        } catch (ParseException e) {
            throw new ServletException(e);
        }

        System.out.println(id);

        Banco banco = new Banco();
        Empresa empresa = banco.buscaEmpresaPelaId(id);
        empresa.setNome(nomeEmpresa);
        empresa.setDataAbertura(dataAbertura);

        return "redirect:entrada?acao=ListaEmpresas";
    }
}
```

COPIAR CÓDIGO

Até poderíamos criar um objeto específico com um tipo de redirecionamento que definisse os dois atributos. Porém, esse tipo de `String` também é utilizado em projetos reais. Salvando essas alterações, nossa ação `AlteraEmpresa` já deve passar a funcionar.

Repetiremos o mesmo processo para `NovaEmpresa`:

```
public class NovaEmpresa {

    public String executa(HttpServletRequest request, HttpServ

System.out.println("Cadastrando nova empresa");

        String nomeEmpresa = request.getParameter("nome");
        String paramDataEmpresa = request.getParameter("data")

        Date dataAbertura = null;
        try {
            SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/
            dataAbertura = sdf.parse(paramDataEmpresa);
        } catch (ParseException e) {
            throw new ServletException(e);
        }

        Empresa empresa = new Empresa();
        empresa.setNome(nomeEmpresa);
        empresa.setDataAbertura(dataAbertura);

        Banco banco = new Banco();
        banco.adiciona(empresa);

        request.setAttribute("empresa", empresa.getNome());

        return "redirect:entrada?acao=ListaEmpresas";
    }
}
```

COPIAR CÓDIGO

E também em `MostraEmpresa`:

```
public class MostraEmpresa {

    public String executa(HttpServletRequest request, HttpServ

        System.out.println("mostrando dados da empresa");

        String paramId = request.getParameter("id");
        Integer id = Integer.valueOf(paramId);

        Banco banco = new Banco();

        Empresa empresa = banco.buscaEmpresaPelaId(id);

        System.out.println(empresa.getNome());

        request.setAttribute("empresa", empresa);

        return "forward:/formAlteraEmpresa.jsp";
    }
}
```

COPIAR CÓDIGO

Nesse caso, colocamos uma barra (`/`) em `forward:/formAlteraEmpresa.jsp`, mas aparentemente isso não é necessário, pois nosso código já estava funcionando sem ela.

Nesse ponto, poderemos reiniciar o Tomcat e testar cada uma de nossas ações no navegador, e todas elas estarão funcionando. Mas por que fizemos essa refatoração do código?

No início, cada uma de nossas ações era praticamente um Servlet. Elas continuam parecidas, mas agora temos uma vantagem: não precisamos mais trabalhar com `response.sendRedirect()` nem com `forward`, somente com o padrão `return: "redirect:entrada?acao=ListaEmpresas"`, que é mais fácil e mais simples.

Existem controladores (entradas) ainda mais espertos, cujos códigos simplificam ainda mais as ações, sem ser necessário ler o parâmetro, fazer o parsing, etc. Alguns exemplos são o **Spring MVC** e o **VRaptor**, que são controladores usados no mercado. Porém, para entendermos de que forma eles fazem isso, vamos melhorar mais um pouco nosso controlador.

No próximo vídeo, faremos mais uma melhoria relacionada com nossos JSPs. Até lá!