



FACULDADE DE CIÊNCIAS E
TECNOLOGIA DA
UNIVERSIDADE DE COIMBRA
DEPARTAMENTO DE
ENGENHARIA INFORMÁTICA

SISTEMAS DISTRIBUIDOS META2

Elementos do grupo:

Christopher Jin Liu, 2013150914

Introdução

Este projeto foi realizado no âmbito da cadeira de Sistemas distribuídos, e tem como objectivo aplicar os conhecimentos adquiridos sobre um sistema RMI de modo a criar uma camada de persistência de dados, numa arquitetura cliente-servidor. As ligações entre os cliente e os servidor são feitas através do protocolo TCP/IP sendo necessário uma ligação UDP/IP entre servidores RMI para verificação de atividade. Sendo que o acesso ao servidor RMI é feito através de métodos remotos.

Com a adição de que nesta 2ª parte do projeto deverá ser criado um frontend acessível pela web para o que foi previamente realizado na fase 1 do projeto utilizando Struts2 e criar novas funcionalidades como ser possível fazer login com o Facebook através de uma REST API.

Arquitetura do projeto e integração do Struts2 com a meta 1

O projeto está dividido em várias packages como foi feito na fase 1 em que cada package têm uma parte do sistema, as packages existentes na meta 1 estão inalteradas tirando a parte do RMI onde os servidores RMI foram postos a correr em localhost devido ao facto de se estar a usar ficheiros e não uma base de dados. Para consultar a arquitetura do RMI, Consola, TCP Server estão no relatório da meta1.

Sendo que foi criada uma nova package webconsola que contém o código referente à meta 2 que será descrito nos parágrafos seguintes.

Servlets

Nesta fase para cada funcionalidade da consola web existe uma servlet com a sua Action e Bean correspondente. E em que cada acção contém um jsp relacionado para o qual o utilizador a partir do menu é redireccionado para realizar o que pretende.

Struts2

A integração da framework Struts2 com o RMI da meta 1 é feita através da Servlet referida no ponto acima em que a framework serve como um filtro para os pedidos realizados nas páginas, que depois de serem recebidos são direccionados para a Action correspondente através do ficheiros struts.xml e da framework que por sua vez chama o método execute() ou seja, os requests de um cliente são enviados sob a forma de uma Action, se o controller recebe um request chama a Action que interage com o model correspondente. O model por sua vez retorna a string a dizer ao controller qual a página de output a enviar ao cliente de acordo com o que está no ficheiro struts.xml A informação é passada entre o model e a view sob a forma de JavaBeans e as tags definidas nos inputs no permitem à view ler e escrever o conteúdo desses beans, da camada de apresentação sem a necessidade de implementar código Java para o fazer.

Será agora explicado a parte funcional do projeto de notar que cada servlet contém a action como por exemplo alterareleicaoAction onde está o execute que será chamado pelo struts.xml, contém também como já foi referido a sua Bean por

exemplo `alterareleicaoBean` que faz o lookup do servidor RMI e chama as funções remotas do RMI.

Alterareleicao

Para alterar uma eleição é necessário fornecer os dados pedidos na página que depois irão ser buscados pela framework que serão enviados para o RMI através da Bean, e onde será visto se a eleição que a pessoa pretende alterar ainda existe através do campo "ID" caso exista é alterada para ficar como os novos dados.

Alterarpessoa

Faz o mesmo que a servlet anterior só com a diferença que neste caso é primeiro visto se o departamento que a pessoa inseriu existe caso exista é enviado para o RMI para ver se o BI que a pessoa introduziu existe caso exista os dados são alterados.

Consultardetalhes

Neste caso no jsp referente existe apenas uma form com um botão submit que depois de ser "triggered" vai chamar a função `execute` na Action que por sua vez irá chamar uma função na Bean que depois irá pedir ao servidor RMI para retornar uma `CopyOnWriteArrayList<EleicaoInfo>` que contém todas as eleições e seus detalhes para se poder fazer display para o utilizador.

Criareleicao

Para criar uma eleição é necessário introduzir todos os dados pedidos na página para serem enviados para o RMI sendo que antes disso na Bean é feito um check como anteriormente para ver se o departamento ou faculdade que a pessoa inseriu existe.

Criarlista

Semelhante a anterior com a exceção de que será realizado na Bean vários checks como se as 3 pessoas adicionadas existem, se a eleição existe, etc, todos checks feitos estão comentados no código da Bean.

Criarmesa

Semelhante à criarlista.

Criarpessoas

Semelhante a alterarpessoas com a diferença que não é feito um check se o BI existe mas apenas se o departamento que a pessoa inseriu existe.

Resultados

O processo aqui é idêntico ao consultardetalhes com a diferença que existe um campo na form ID que é o ID relativo à eleição que a pessoa pretende consultar os resultados.

Verlocaleleicao

Semelhante ao anterior com a diferença de em vez mostrar os resultados mostra onde cada eleitor votou na eleição com o ID introduzido.

Login

Como o intuito de proteger as páginas contra acesso a ações sem o utilizador estar logado é pedido o BI da pessoa e a password correspondente, que depois na Bean serão comparados com todos os utilizadores existentes retornados pelo servidor RMI e vê se as credenciais coincidem e se o tipo do utilizador pode acessar a webconsola, caso possa será posto na session o BI do utilizador com a key “BI”.

Logout

Neste caso não é necessário um Bean pois não é necessário comunicação com o RMI existindo apenas uma Action que no método execute ao ser chamado retira o BI da session e através do struts.xml redireciona o utilizador para a página de

login. Sendo que caso não fosse feito a operação de o BI ser retirado as páginas irão poder ser acessada.

AuthorizationInterceptor

Esta é uma classe à parte das outras tirando o login e logout sendo o único propósito dela controlar o interceptor criado em struts.xml para quando é feito um submit numa página em que na session não contenha a key “BI” ou esta seja null ou vazia vai retornar uma string que por sua vez na struts.xml vai redirecionar para a página de login evitando assim o acesso a acções sem login.

Toda a parte dos interceptors foi baseado através de:

<https://stackoverflow.com/a/24401457>

Descrição dos testes feitos à plataforma

Teste	Resultado
Inputs indesejados	Fail
Proteção de páginas	Pass