

UNIVERSIDADE DE COIMBRA

LICENCIATURA ENGENHARIA INFORMÁTICA
2014-2015

Sistemas Distribuídos - Projeto 2

Autores:

Bruno MARTINS - Nº2007183389

8 de Dezembro de 2014

Conteúdo

1	Introdução	2
2	Web-Based Internal Architecture	3
3	How Struts (webserver) is integrated with RMIServer	5
4	WebSockets integration with Struts	6
5	Struts/RMIServer integration with Google Calendar	7
6	Description of tests (Table with Pass/Fail)	8
6.1	Teste de Registo via web	8
6.2	Teste de Login via web	9
6.3	Teste de criação de meeting via web	10
6.4	Teste de tentativa de acesso directo a actions	11
6.5	Teste de Associação da conta com Google Calendar	11
6.6	Teste de login com Google Calendar	11
7	Manual de Instalação	12

1 Introdução

O presente relatório destina-se a apresentar a descrição do projecto 2 da cadeira de Sistemas Distribuídos intitulado *Meeto – Collaboration and Social Networking*.

Reuniões são necessárias quando diversas pessoas trabalham em conjunto. Alunos reúnem-se para coordenar o seu esforço nos trabalhos práticos; trabalhadores reúnem-se para discutir o estado dos projectos e gestores reúnem-se para tomar decisões chave relacionadas com as suas organizações e por aí fora. É necessário um sistema que impeça as pessoas de gastarem tempo desnecessário em reuniões. Afinal de contas, tempo é o nosso recurso mais valioso.

Este projecto consiste então numa implementação de um sistema de gestão de reuniões, onde os utilizadores podem criar reuniões, convidar participantes, ter uma sala de chat para cada tópico da reunião e no final adicionar decisões chave a cada reunião. É também possível relacionar pessoas a certas tarefas - uma vez que uma reunião sem atribuição de tarefas concretas aos seus elementos de pouco serve.

Nesta segunda fase do projecto, pretende-se criar uma aplicação web-based, permitindo que qualquer utilizador possa aceder à aplicação, independentemente do local onde se encontra, sem que para isso seja necessário instalar qualquer tipo de software específico. Desta forma, a aplicação web irá ter uma conexão ao servidor base de dados, através de JavaBeans, que serão despoltados através de Actions de Struts2.

A informação principal será automaticamente actualizada, através da utilização dos WebSockets. Assim, o utilizador será automaticamente notificado caso haja alguma meeting que o envolva directamente.

Finalmente, a aplicação web será integrada com o Google Calendar. Significa isto que os utilizadores poderão associar a sua conta do Google com a do Meeto. Desta forma, sempre que um utilizador cria uma nova meeting, a mesma será criada do seu calendário Google.

2 Web-Based Internal Architecture

A arquitectura interna da aplicação teve como base o esquema fornecido no enunciado do projecto. Nesta meta 2, para além da arquitectura já patente na meta 1, foram adicionados aspectos web-friendly de modo a que a aplicação seja acessível online.

Para tal, o utilizador pode aceder através de qualquer browser à página onde está alojada a nossa aplicação. A partir daí, é estabelecida a comunicação ao **Tomcat WebServer**, através de uma ligação http. Todas as funcionalidades a que o utilizador acede na web são tratadas pela framework MVC Struts2. Esta encarrega-se de executar todas as acções presentes nas diversas partes da aplicação, facilitando o acesso a dados dos JSP's (através de getters e setters) e encaminhando cada *Action* para ficheiros Java correspondentes que por sua vez chamam Beans que são os responsáveis pela ligação RMI à base de dados. Depois de ter realizado o pedido à base de dados, as variáveis das Actions são acedidas no JSP através das tags do Struts2 (graças, mais uma vez aos getters e setters - para isso as variáveis num lado e no outro têm de possuir, naturalmente, o mesmo nome).

Todas as *Actions* de Struts2 são também apanhadas por um interceptor criado de raiz (interceptor esse que foi definido como default para quase todas as actions). No fundo, este componente intercepta todas as actions a decorrer na rede e realiza as seguintes verificações: se apanhar uma action em que a sua sessão não tem qualquer tipo de user associado (esta associação de users à sessão é feita logo após o processo de login) então encaminha directamente para o login.jsp, protegendo assim a aplicação de acessos directos não autorizados a Actions por parte de utilizadores que não estão logados. De seguida, realiza outra verificação para controlar se o utilizador, apesar de logado, está a aceder directamente por URL a acções de criação (e.g.: criar uma meeting). Se a Action passar por estas verificações, então é porque o user está a tentar realizar um pedido de forma válida, e segue para o seu destino.

Finalmente foi implementada uma ligação **REST** over http ao API do Google, de forma a assegurar a integração de ambas as plataformas tal como foi referido no capítulo da introdução.

É também importante referir que, para facilitar a transição entre páginas e colocação de JSP's dentro de JSP's, é usado JQuery para controlo do DOM. Esta

aplicação utiliza também a framework Bootstrap para seu "embelezamento".

3 How Struts (webserver) is integrated with RMI-Server

Tal como referido em cima, a integração do Struts2 com o RMI Server é feita através de JavaBens que são acedidos através de ficheiros "JavaAction" correspondentes. Os JSP's não possuem qualquer tipo de JavaScript. As forms e buttons relevantes são criados através com tags de Struts2 e possuem o campo "action". É neste que são colocadas as acções definidas no struts.xml. Qualquer JavaScript que possa existir está presente em ficheiros JS (juntamente com o JQuery) e importados no final das páginas JSP, consoante necessário.

Por exemplo, para criar uma meeting, o utilizador coloca as informações relevantes na form de criação e carrega no botão "Submit" que possui uma action "createMeeting" que, graças ao struts.xml, sabe que método java chamar (createMeeting()) e onde este se encontra (pt.dei.uc.actions.MeetingAction.java). Este MeetingAction vai criar objecto do tipo MeetingBean, onde está localizado o método para a ligação à base de dados e execução do pedido de criação em si. Depois de chamado o método e retornado o resultado para uma variável outcome, é verificado esse mesmo resultado: se for igual a true então vai chamar o método addActionMessages do Struts2 com "Success". Caso contrário, chama o método addErrorMessage com "Error". Estas mensagens de sucesso ou de erro são então apresentadas com `<s:if test="hasActionMessages()"` ou `<s:elseif test="hasActionErrors()"` no main.jsp (ficheiro de retorno para a action createMeeting e resultado Success).

4 WebSockets integration with Struts

A integração de Websockets com o Struts2 foi necessária de forma a que o Struts2 permitisse a passagem dos pedidos para os endpoints de websocket. Para isso teve que ser colocado a seguinte constante no ficheiro struts.xml : `<constant name="struts.mapper.action.prefix.enabled" value="true"/>` . Foi também necessário definir excluir manualmente o interceptor default acima referido, para que as acções de websockets fossem executadas.

5 Struts/RMIServer integration with Google Calendar

Através da integração com a Google pretende-se que o utilizador possa associar a sua conta Google à conta Meeto com o objetivo de facilitar o processo de login na aplicação e de poder ver todas as reuniões agendadas na sua conta Google.

Para a implementação deste mecanismo foi necessário registar uma aplicação na Google para assim obter uma *apiKey* e *apiSecret*. Com estas duas chaves, a aplicação Meeto pode fazer chamadas à API da Google e obter todos os dados necessários da conta do utilizador.

Antes de o utilizador poder fazer login com a sua conta Google, este precisa de ter uma conta Meeto e fazer a associação entre estas duas contas. Sempre que o utilizador faz login, a aplicação verifica se o campo na base de dados referente ao id da conta Google está preenchido ou não. Se não estiver, é apresentado um link no canto superior direito do ecrã que dá início ao processo de associação das contas.

Ao carregar em “Associate with Google” o utilizador é encaminhado para uma página que informa o utilizador que a aplicação pretende obter acesso aos seus dados básicos de perfil e calendário. Uma vez que o utilizador aceita estas permissões, a aplicação recolhe o id do utilizador na Google, o seu email e guarda também o token associado ao pedido para futuras operações tais como adicionar eventos no calendário. Logo de seguida, a aplicação gera um novo calendário de nome Meeto no Google Calendar que servirá para adicionar todas as reuniões do utilizador.

No processo de login a aplicação obtém o id Google do utilizador e faz uma procura por esse id na base de utilizadores. Se encontrar o utilizador inicia a sessão com o username existente nesse registo e atualiza o token de acesso gerado pela Google. Caso contrário o login falha.

6 Description of tests (Table with Pass/Fail)

Foram feitos diversos testes nas diversas funcionalidades da aplicação, de modo a assegurar a robustez da aplicação garantindo, assim, um sistema seguro e eficiente, procurámos testá-lo minuciosamente.

6.1 Teste de Registo via web

Test Name: Registar				
Description: Registo do cliente na BD da aplicação				
Prerequisites: O cliente tem de entrar na aplicação e escolhe a opção REGISTAR				
Setup: O cliente cria uma nova conta				
Step	Operator Action	Expected Results	Observed Results	Pass / Fail
1	Escolhe a opção Registar	Sistema abre os campos de registo	São os esperados	PASS
2	Inserir username, password e email	Sistema aceita dados e guarda na BD ou rejeita caso o username escolhido já esteja a ser utilizado	São os esperados	PASS

6.2 Teste de Login via web

Test Name: Login				
Description: Identificação do cliente				
Prerequisites: Cliente tem de ter uma conta criada na aplicação				
Setup: O cliente insere a sua pass e username para usar a sua conta				
Step	Operator Action	Expected Results	Observed Results	Pass / Fail
1	Inserir username, password	Sistema aceita dados se essa conta existir	São os esperados	PASS

6.3 Teste de criação de meeting via web

Test Name: Teste de Submissão de Meeting				
Description: É testada a funcionalidade de introduzir meeting				
Prerequisites: O utilizador tem que já estar registado				
Setup: O servidor irá fazer handle das meetings				
Step	Operator Action	Expected Results	Observed Results	P / F
1	logar User	Login efectuado com sucesso!	Nada apontar	PASS
2	User entra na aplicação e acede à opção "Create Meeting" do menu lateral	aparece o passo seguinte	-	PASS
3	Introduz os detalhes da meeting	-	-	PASS
4	Escolhe vários utilizadores para a meeting	Output "Meeting successfully created"!	-	PASS
5	Escolhe a opção "List meetings" do menu	A meeting acaba de criar encontra-se na lista	nada a apontar	PASS

6.4 Teste de tentativa de acesso directo a actions

Test Name: Teste de tentativa de acesso directo a actions				
Description: Tentativa de acesso inválido a action por url				
Prerequisites:-				
Setup: Acesso normal à sua conta				
Step	Operator Action	Expected Results	Observed Results	Pass / Fail
1	Sem estar logado, tentar aceder a uma action (e.g. home.action)	Sistema reencaminha o user para a página de login	-	PASS
2	Estando logado, tenta aceder a uma action de criação (e.g. createAction)	O sistema reencaminha para a página principal e é mostrado uma mensagem de erro a comunicar que o utilizador não pode realizar aquela acção	PASS	

6.5 Teste de Associação da conta com Google Calendar

6.6 Teste de login com Google Calendar

Test Name: Associar conta Google				
Description: Associar uma conta Meeto a uma conta Google				
Prerequisites: O cliente tem de ter uma conta Meeto e uma conta Google.				
Setup: As duas contas ficam associadas				
Step	Operator Action	Expected Results	Observed Results	Pass / Fail
1	Carregar em "Associate with Google"	O cliente é encaminhado para o ecrã de permissões da Google	São os esperados	PASS
2	Aceitar as condições	A aplicação recolhe os dados necessários e encaminha o utilizador para o ecrã inicial	São os esperados	PASS

Test Name: Login com conta Google				
Description: Login na aplicação Meeto com uma conta Google				
Prerequisites: O cliente tem de ter uma conta Meeto e uma conta Google. As duas contam tem de estar associadas.				
Setup: Acesso normal à sua conta				
Step	Operator Action	Expected Results	Observed Results	Pass / Fail
1	Carregar em Google Calendar	Sistema procura por dados existentes na conta na base de dados. Actualiza o token caso encontre os dados e encaminha para o menu principal.	São os esperados	PASS

7 Manual de Instalação

Para correr a aplicação irá necessitar dos seguintes requisitos:

- Java 8
- Servidor com MySQL com a respectiva base de dados (o script de criação da base de dados encontra-se na pasta support, ficheiro *meetosqlscript.sql*)
- Tomcat 8

Após reunidos todos os requisitos necessários, é necessário que o MySQL esteja a correr dentro da própria (localhost), ou no caso que esteja se encontre noutra máquina, que seja definido o respectivo IP e porta. Como foi mencionado anteriormente, deve-se correr o script *meetosqlscript.sql* que contem todas as tabelas criadas bem como alguns dados. Por fim, deve-se executar-se por esta ordem a execução de servidores da máquina para obter o comportamento necessário da meta 1:

- 1 - RMIServer.java
- 2 - TCPServer.java (duas vezes para simular o comportamento de servidor primário e secundário)
- 3 - TCPClient.java (quantas vezes necessário para criar vários clientes)

Relativamente ao IP de máquina onde os servidores e o cliente estão a correr, existe um ficheiro intitulado *property* (support/property) onde se pode alterar os endereços e as portas onde estes vão estar a correr. Possui o seguinte conteúdo:

```
tcpip1=10.0.0.1
tcpip2=10.0.0.2
tcpServerPort=7000
tcpServerPortAux=7001
tcpBackupServerPort=7002
tcpBackupserverPortAux=7003
udpPort=5432
rmiServerip=10.0.0.1
rmiServerPort1=6001
rmiServerPort2=6002
```

No que diz respeito à segunda meta do projecto, e depois de correr os passos acima mencionados, esta poderá ser validada possuindo e executando o Tomcat e colocando o ficheiro "property" e "policy.all" na sua pasta bin. De seguida, basta abrir um browser e aceder ao seguinte link:

10.0.0.1:8080/meeto. Substituir "10.0.0.1" pelos ips acima definidos, se necessário.