

Projeto, Parte 2

Tolerância a Faltas

Grupo A30

Instalação e configuração do projeto

O projeto foi submetido no github(<https://github.com/tecnico-distsys/A30-SD18Proj>) com a tag SD_P2. Deve fazer o download relativo a esta entrega e extrair o ficheiro, a pasta resultante deve conter todos os quatro módulos necessários ao desenvolvimento do projeto e a pasta uddi-naming, pasta onde instalamos o nosso servidor jUDDI.

Para correr o projeto necessita:

1. Correr um servidor jUDDI local ou remoto;
2. Ir para a pasta raiz do projeto;
3. Abrir uma consola para a Station 1:
 - `cd station-ws`
 - `mvn clean compile exec:java`
4. Abrir uma consola para a Station 2:
 - `cd station-ws`
 - `mvn clean compile exec:java -Dws.i=2`(é possível definir as coordenadas da estação através do comando: `mvn clean compile exec:java -Dws.i=2 -Dstation.coords.x=80 -Dstation.coords.y=20`)
5. Abrir uma consola para a Station 3:
 - `cd station-ws`
 - `mvn clean compile exec:java -Dws.i=3`(é possível definir as coordenadas da estação através do comando: `mvn clean compile exec:java -Dws.i=3 -Dstation.coords.x=50 -Dstation.coords.y=50`)
6. Instalar o Station-ws-cli:
 - `cd station-ws-cli`
 - `mvn clean install exec:java`(nesta fase os testes de integração iram correr na fase “verify”, antes da fase “install”)
7. Abrir consola para o Binas:
 - `cd binas-ws`
 - `mvn clean compile exec:java`
8. Abrir consola para o cliente do Binas:
 - `cd binas-ws-cli`
 - `mvn clean compile exec:java`(O binas-ws-cli ao correr, irá chamar uma operação de teste chamada Ping e imprimir o seu resultado)
9. Correr os testes de integração no binas client:
 - `mvn verify`

Demonstração

Para demonstrar um exemplo de funcionamento normal

Execute os comandos anteriormente descritos até ao ponto 8. Neste momento verá que existem 3 estações ligadas, durante a execução do *Invoke ping*

```
[INFO] --- exec-maven-plugin:1.5.0:java (default-cli) @ binas-ws-cli ---
Creating client using UDDI at http://a30:HmUvRkb@uddi.sd.rnl.tecnico.ulisboa.pt:9090 for server with name A30_Binas
Invoke ping()...
Hello client from A30_Binas!
Found 3 stations on UDDI.
Ping result for station at http://localhost:8083/station-ws/endpoint:
Hello A30_Binas from A30_Station3
Ping result for station at http://localhost:8082/station-ws/endpoint:
Hello A30_Binas from A30_Station2
Ping result for station at http://localhost:8081/station-ws/endpoint:
Hello A30_Binas from A30_Station1
[INFO] -----
```

Poderá verificar que existem 3 estações introduzidas no UDDI.

Por fim executar o comando do ponto 9

```
Results :

Tests run: 29, Failures: 0, Errors: 0, Skipped: 0

[INFO] --- maven-failsafe-plugin:2.19.1:verify (default) @ binas-ws-cli ---
[INFO] -----
[INFO] BUILD SUCCESS
```

Verifica-se que todos os testes passam com sucesso.

Para demonstrar um exemplo de funcionamento com falha

Execute os comandos anteriormente descritos até ao ponto 8, mas desta vez, salte o ponto 5. O resultado da execução do ponto 8, a execução do *Invoke Ping* será a seguinte:

```
Creating client using UDDI at http://a30:HmUvRkb@uddi.sd.rnl.tecnico.ulisboa.pt:9090 for server with name A30_Binas
Invoke ping()...
Hello client from A30_Binas!
Found 2 stations on UDDI.
Ping result for station at http://localhost:8082/station-ws/endpoint:
Hello A30_Binas from A30_Station2
Ping result for station at http://localhost:8081/station-ws/endpoint:
Hello A30_Binas from A30_Station1
[INFO] -----
```

Verifica-se que desta forma, apenas 2 estações estão introduzidas no UDDI.

Por fim executar o comando do ponto 9

```
Results :

Tests in error:
  GetInfoStationIT.oneTimeSetUp:30 » BadInit_ No Station found with ID: A30_Stat...
  ListStationsIT.oneTimeSetUp:36 » BadInit_ No Station found with ID: A30_Statio...
Tests run: 22, Failures: 0, Errors: 2, Skipped: 0
```

Os erros devem-se ao facto de os testes verificarem uma 3ª station que não existe, pois não foi iniciada.