Sistemas Distribuídos - 2015/2016 Curso: Engª Informática

Folha 11 - 1

→ JAX-RS: Java API for RESTful Web Services

A - Creating RESTful Web Services from a Database

1- Comece por criar um projeto do tipo "Java Web application", como fez nos exercícios das fichas anteriores. No exemplo que ilustra este exercício o nome do projecto foi wsFirst.

2 – Gerar entity classes e RESTful Web Services:

- Clicando com o botão direito do rato sobre o nó do projecto seleccione: New /Other /Web Services / RESTful WebServices from Database. Após Next, o "New RESTful Web Service wizard" abre no painel das tabelas da base de dados.

- Em Data Source seleccione jdbc/sample e aparecerão as tabelas da base de dados exemplo com que já trabalhou. Selecione a tabela MANUFACTURER.



- Após Next, atribua um nome ao package onde irão ser colocadas as *entities* a criar (por exemplo, entities).

- Após Next, poderá ver uma janela com o nome do package onde irão ser colocadas as classes do serviço a criar (por exemplo services).

New RESTful Web	Services from Database	ie		×		🕡 New RESTful Web Se	ervices from Databa	ase		×
Steps	Entity Classes			_	Steps	Generated Class	25			
 Choose File Type Database Tables Entity Classes Generated Classes 	Specify the names and Class Names: Datab MANU	d the location of th ibase Table C UFACTURER Ma	he entity classes. Class Name lanufacturer	Generation Type New		 Choose File Type Database Tables Entity Classes Generated Classes 	<u>P</u> roject: Location: Resource Pac <u>k</u> age:	WSfirst Source Packages entities.service		~
	Project: WSfirs Location: Sourc Package: ehtitie	rst rce Packages les Query Annotations	s for Persistent Fi	elds						
	Generate JAXB Ann Generate MappedS Create Persistence	nnotations Superclasses instea e <u>U</u> nit	ad of Entities							
		< <u>B</u> ack Next	t > <u>F</u> inish	Cancel <u>H</u> elp				< <u>B</u> ack Next >	Einish Cancel	Help

Paula Prata, Departamento de Informática da UBI

Sistemas Distribuídos - 2015/2016 Curso: Engª Informática

Folha 11 - 2

- Após Finish, o editor irá criar as *entities* (neste caso apenas uma), tal como já tinha feito em exercícios anteriores. Pode observar a entitiy criada no package entities.

- No package services pode observar que para cada entity foi criado um stateless Session Bean. Neste caso foi criado 0 bean ManufacturerFacadeREST, que contém as operações base para manipular os dados da tabela correspondente.

Observe que esta classe é subclasse da classe abstracta AbstractFacade. Esta classe é uma classe genérica que pode ser parametrizada com diferentes tipos de dados. Neste caso com Manufacturer, consoante a tabela que queremos manipular.

- Observe ainda que é criada uma classe de configuração ApplicationConfig.java.

B - Testar o RESTful Web Service localmente.

Antes de testar o web service vamos permitir que ele possa ser acedido por outras máquinas. Selecionar o projecto, clicar no botão direito e seleccionar:

new/ other / Web services/ Cross origin Resource Sharing; após Next, dar um nome ao package (util por exemplo), Finish.

🜍 New File	×	🗊 New Cross-Origin R	esource Sharing Filter X
Steps	Choose File Type	Steps	Name and Location
1. Choose File Type 2	Project: WSfirst File Types: Categories: File Types: File Types: Categories: File Types: File Type: F	1. Choose File Type 2. Configure Jersey Filter	Class Name: NewCrossOriginResourceSharingFilter Project: WSfirst Location: Source Packages Package: util Created File: \src\java\util\NewCrossOriginResourceSharingFilter.java Fill HTTP Access Control Headers: Access-Control-Allow-Qrigin: * Access-Control-Allow-Methods: GET, POST, PUT, DELETE Sele Access-Control-Allow-Headers: Content-Type
	< Back Next > Einish Cancel Help		< Back Next > Einish Cancel Help

Agora pode testar o web service: Clicar com o botão direito do rato sobre o nó do projecto, seleccionar a opção "Test RESTful Web Services". Seleccione a opção, Locally generated Test Client ... e após "ok", o editor gerará o ficheiro test-resbeans.html (ver default package) faz o deploy da aplicação e inicia a execução.

Sistemas Distribuídos - 2015/2016 Curso: Engª Informática

Folha 11 - 3
Configure REST Test Client
Configure field fie

No seu browser aparecerá a janela ilustrada abaixo que tem do lado esquerdo um nó para cada uma das entities criadas (neste caso para a entity manufacturer):

WADL : http://localhost:8080/WSfirst/webresources/application.v	vad
₩Sfirst	WSfirst > entities.manufacturer > {id}
ia 🌽 entities.manufacturer ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	Resource: entities.manufacturer/{id} (http://localhost:8080/WSfirst/webresources/entities.manufacturer/{id})
Count Image: second	Choose method to test: GET(application/xml)
	id:
	↓ Custom Request Headers

Expanda os nós e explore as operações disponibilizadas.

Na janela de output tem 5 abas: Em Raw View pode observar o resultado do query seleccionado.

Sistemas Distribuídos - 2015/2016 Curso: Engª Informática

	Folha 11 - 4					
WSfirst entities.manufacturer id count if (from)/{to}	Resource: entities.manufacturer/{from}}{to; (http://localhost:8080/WSfirst/webresources/entities.manufacturer/{from}}{to; Choose method to test: GET(application/xml) Test from: 0 to: 1000000					
	Response:					
	Tabular View Raw View Sub-Resource Headers Http Monitor					
	<pre></pre> <pre> </pre> <pre> </pre> <pre> </pre>					

C - Criar um cliente para o Web Service:

Criar um novo projecto (aplicação em linha de comando: Java / java Application). Neste projecto fazer: New / New Restful Java Client, dar nome à classe e seleccionar o serviço criado no projecto anterior:

New Java Applicatio	n >		🕡 New RESTful Java Cl	Client ×
Steps	Name and Location		Steps	Name and Location
 Choose Project Name and Location 	Project Name: WSfirstClient Project Location: D:_SIst_Dist_15_16\projetosJEE Project Folder: _Dist_15_16\projetosJEE\WSfirstClient		 Choose File Type Name and Location 	Class Name: manuf Project: WSfirstClient Location: Source Packages
	Use Dedicated Folder for Storing Libraries Libraries Folder: Different users and projects can	-		Package: dients <u>Created File: 15_16\projetosJEE\WSfirstClient\src\clients\manuf.java</u> Select the REST resource:
	share the same compilation libraries (see Help for details).]		Erom Project DIE Registered Browse BEST Resource Name: ManufacturerFacadeREST [entities.manufacturer]
	< Back Next > Finish Cancel Help]		Authentication: NONE SSL Subscript{Cancel_Help}

Foi gerada a classe manuf.java que contém código para aceder aos métodos do web service REST criado. Agora na classe podemos escrever o código para aceder a esses métodos.

Sistemas Distribuídos - 2015/2016 Curso: Engª Informática

Folha 11 - 5

No main que foi criado no seu projecto insira o código exemplo:

// Mostrar o número de linhas da tabela MANUFACTURER
manuf m = new manuf();
String count = m.countREST();
System.out.println(count);

// Consultar dados de um dado manufacturer
String manufacturerById = m.find_XML(String.class, "19985678");
System.out.println("Manufacturer com o ID 19985678: " + manufacturerById);

Explore e execute o projeto. Para poder visualizar os dados em XML ou em JSON terá de usar um parser para a respetiva conversão.

Para aprofundar o estudo em web services pode consultar por exemplo: http://docs.oracle.com/javaee/7/tutorial/ (capítulos 29, 30 e 31).