

→ **JAX-RS: Java API for RESTful Web Services**

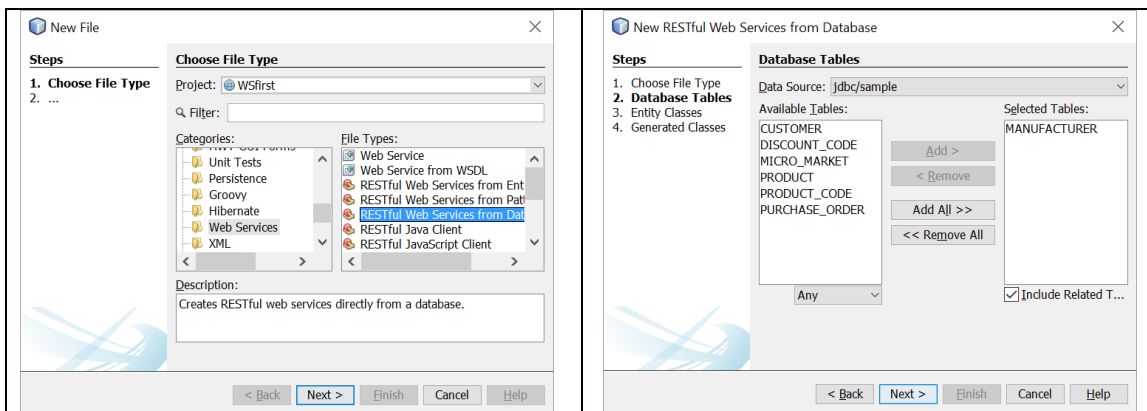
**A - Creating RESTful Web Services from a Database**

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:

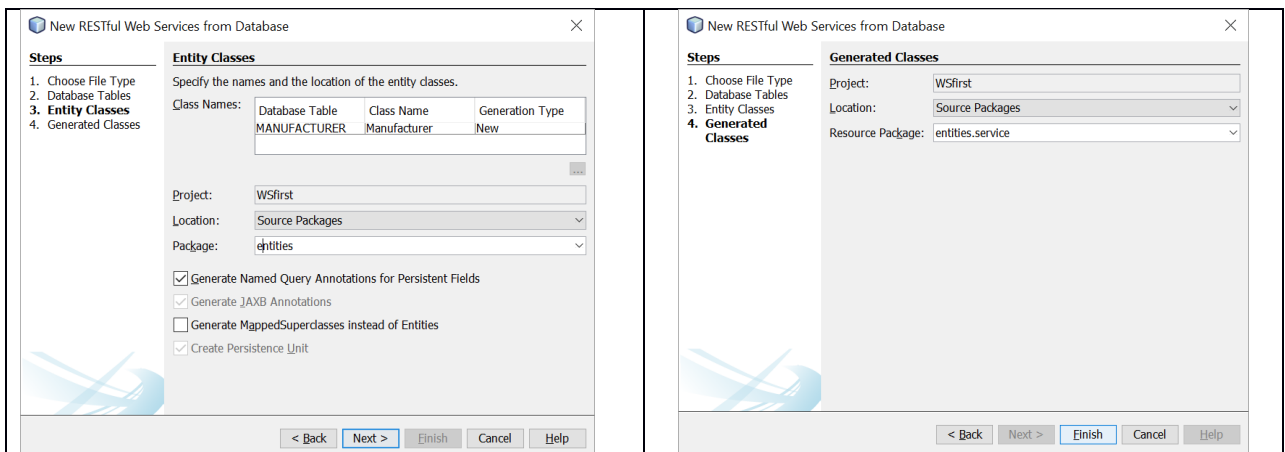
- Clicando com o botão direito do rato sobre o nó do projecto seleccione: New /Other /Web Services / RESTful WebServices from Database. Após Next, o “New RESTful Web Service wizard” abre no painel das tabelas da base de dados.

- Em Data Source seleccione jdbc/sample e aparecerão as tabelas da base de dados exemplo com que já trabalhou. Seleccione 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).



- 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.

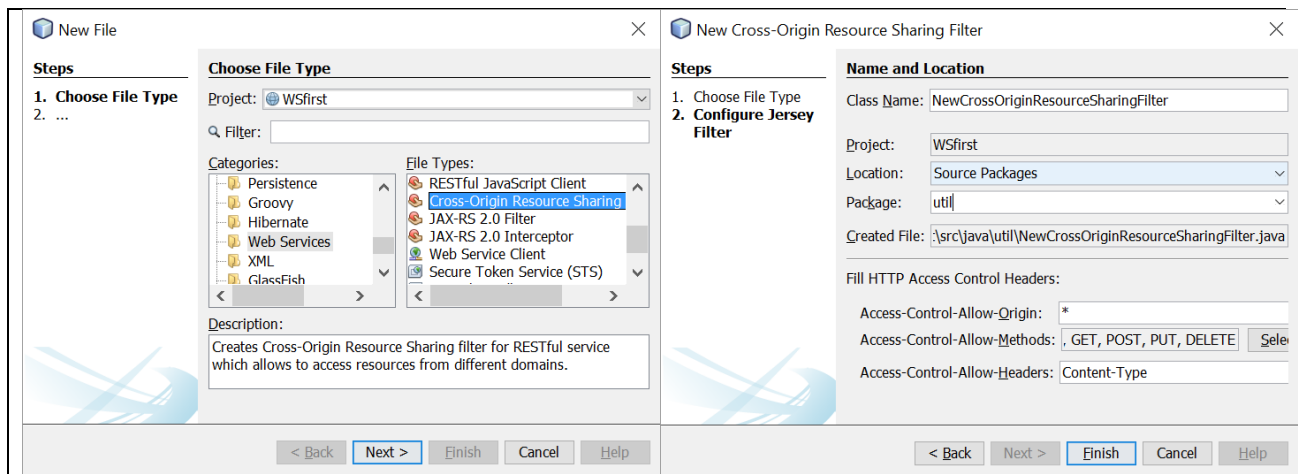
Observe que esta classe é subclasse da classe abstracta AbstractFacade. Esta classe é uma classe genérica que pode ser parametrizada com diferentes tipos de dados. Neste caso com Manufacturer, consoante a tabela que queremos manipular.

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

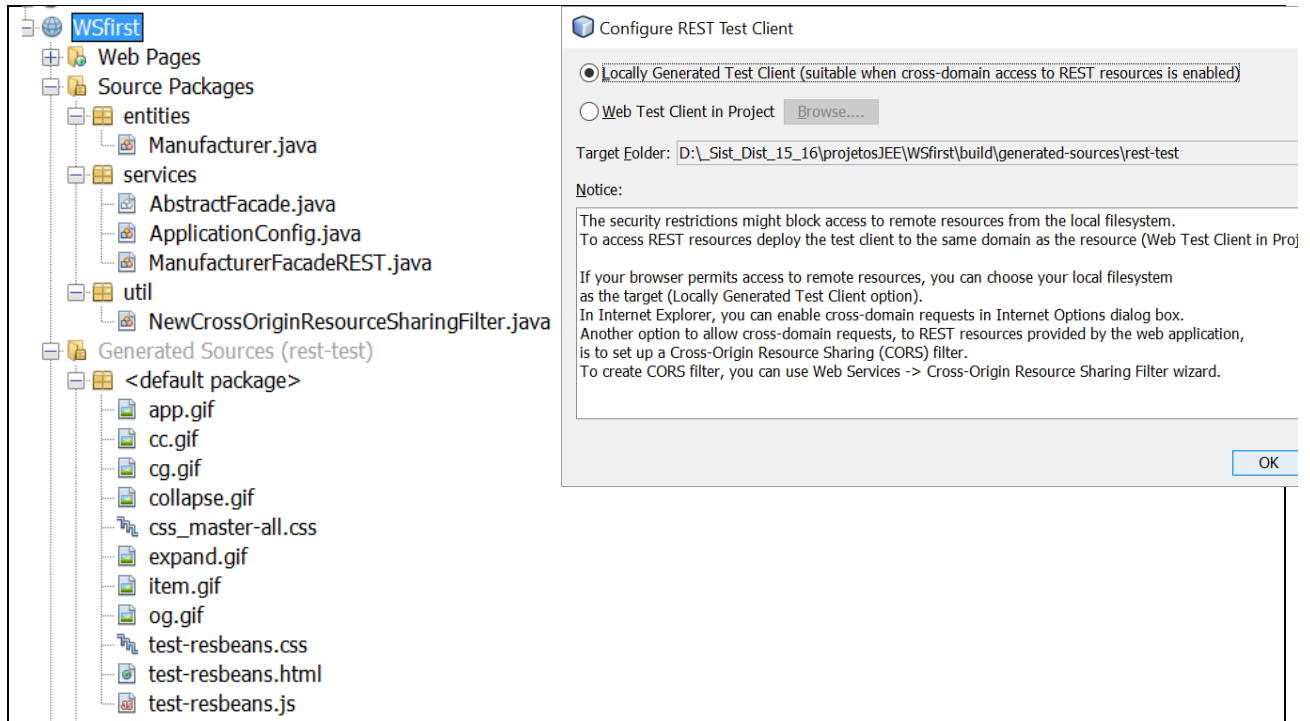
## B - Testar o RESTful Web Service localmente.

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

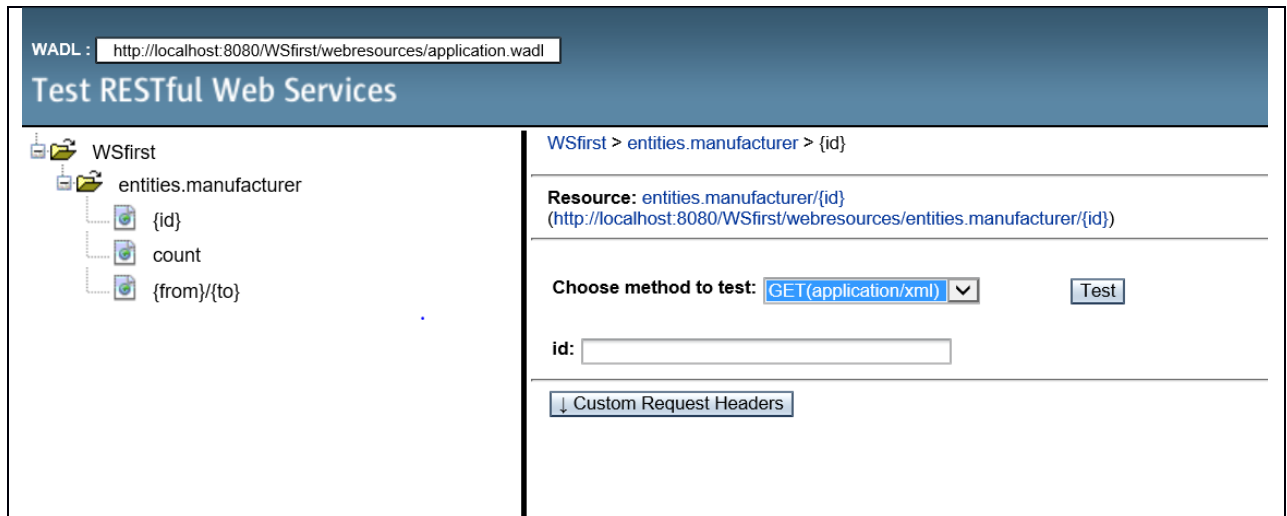
new/ other / Web services/ Cross origin Resource Sharing; após Next, dar um nome ao package (util por exemplo), Finish.



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”. Selecione 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.

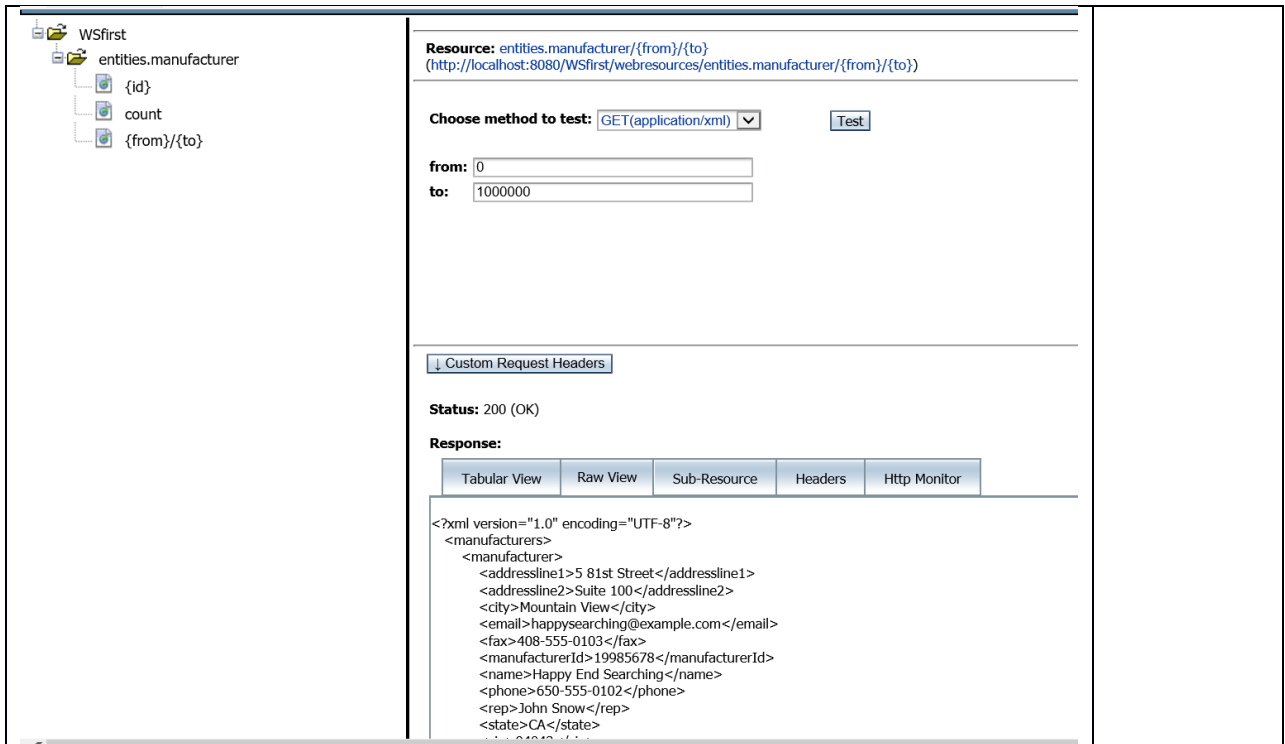


No seu browser aparecerá a janela ilustrada abaixo que tem do lado esquerdo um nó para cada uma das entities criadas (neste caso para a entity manufacturer):



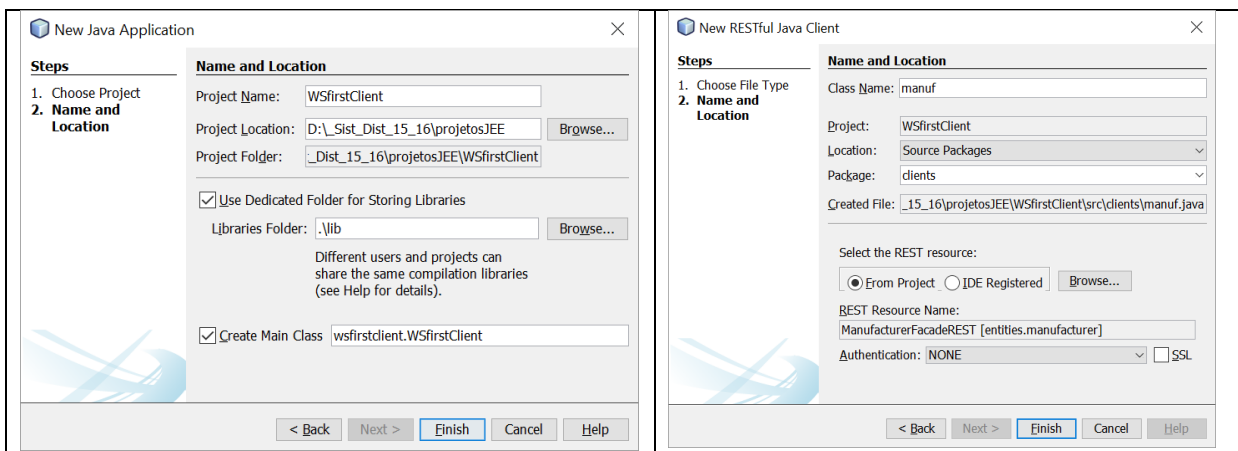
Expanda os nós e explore as operações disponibilizadas.

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



### 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:



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.

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).