

Utveckling av desktopapplikationer

Föreläsning 10

VT2017

NAZILA H.



Innehåll

➤ JAX-RS Client

JAX-RS Client API

- JAX-RS Client API används för att skriva klientsidan av en Restful web service.
- Package: `javax.ws.rs.client`
- Vi behöver skriva en Klient-applikation som kan kommunicera med servern.
- Vi väljer JavaFX men det kan också vara andra typer av klient-applikationer.

Att lägga till jar-filer

Vi behöver lägga till följande jar-filer/library i projektet:

- Javax.ws.rs-api
- Jersey library

Client Request

- Interfacet 'Client' i paketet.
- Objekt av typen Client.
- Specificera destinationsadressen
- Specificera formatet (Skapa en Request)
- Anropa en av http-metoderna (get, put, post, ...)

```
Client client = ClientBuilder.newClient();
```

```
String name = client.target(" http://localhost:8080/FotballServer/webapi/teams" )  
    .request(MediaType.TEXT_PLAIN)  
    .get(String.class);
```

Client

Client-objekt ska stängas efter att vi är klara med det.

```
Client client = ClientBuilder.newClient();
```

```
...
```

```
client.close();
```

Client Target

➤ `javax.ws.rs.client`

➤ metoden `path("newPath ")` används för att lägga till en sub-path.

```
Client client = ClientBuilder.newClient();
```

```
WebTarget base = client.target("http://localhost:8080/FotballServer/webapi");
```

```
// WebTarget för http://localhost:8080/FotballServer/webapi/teams
```

```
WebTarget teams= base.path("teams");
```

```
// WebTarget för http://localhost:8080/FotballServer/webapi/teams/1
```

```
WebTarget teamId= teams.path("1");
```

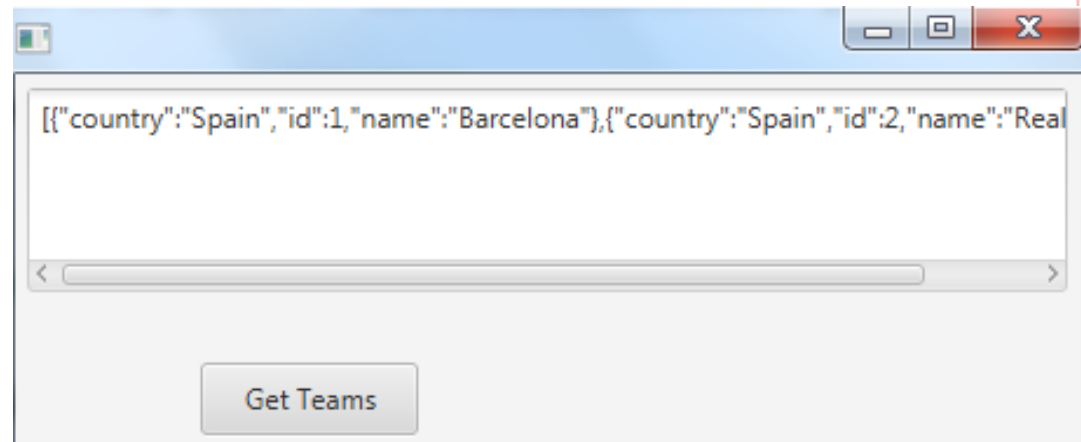
Exempel – Get Team as String

```
@FXML
private void handleJasonButtonAction(ActionEvent event) {

String c = client.target("http://localhost:8080/FotballServer/webapi/teams")
    .request(MediaType.APPLICATION_JSON)
    .get(String.class);
text.appendText(c);

}
```

En textArea
på scenen



Get a Team as an Object

```
@FXML
private void handleGetButtonAction(ActionEvent event) {
    Team team=client
        .target("http://localhost:8080/BackendDemoFootball/webapi/teams/1")
        .request(MediaType.APPLICATION_JSON)
        .get(Team.class) ;
}
```

Get All Teams as a List of Objects

```
List<Team> team=client
    .target("http://localhost:8080/BackendDemoFootball/webapi/teams")
    .request(MediaType.APPLICATION_JSON)
    .get(new GenericType<List<Team>>() {});
```

Update Team

@FXML

```
private void handlePutButtonAction(ActionEvent event) {  
  
    Team t=new Team();  
    t.setId(2);  
    t.setName("A New Name");  
  
    Response r= client  
        .target("http://localhost:8080/BackendDemoFootball/webapi/teams/2")  
        .request()  
        .put(Entity.entity(t, MediaType.APPLICATION_JSON));  
  
    text.appendText(r.toString());  
}
```

Create a new Team

```
@FXML
private void handlePostButtonAction(ActionEvent event) {
    Team t=new Team();
    t.setName("My New Team"); //Id will be set by MYSQL (Auto Increment)

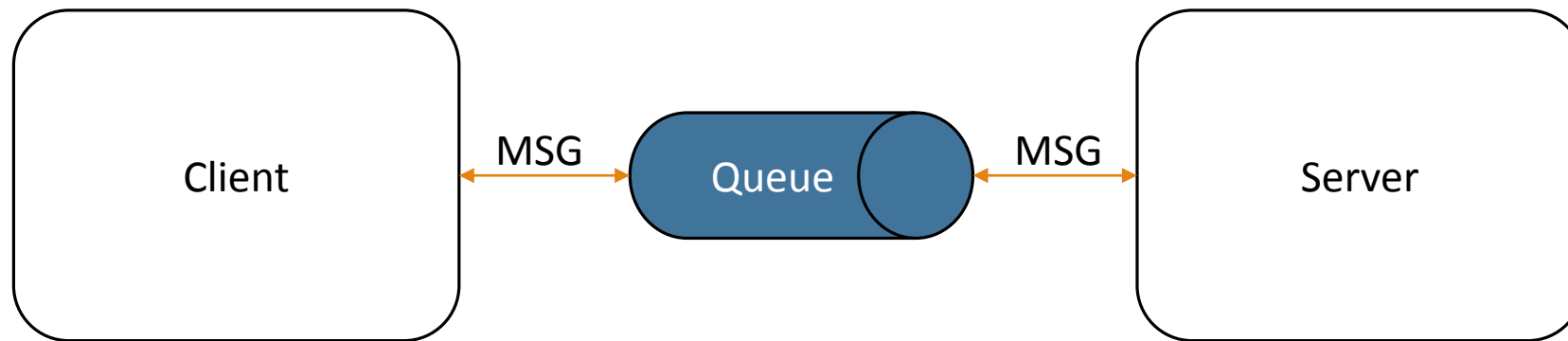
    Team team= client
        .target("http://localhost:8080/BackendDemoFootball/webapi/teams")
        .request(MediaType.APPLICATION_JSON)
        .post(Entity.json(t), Team.class);

    //Just for checking
    text.appendText("OK" + team.getName());
}
```

Delete a Team

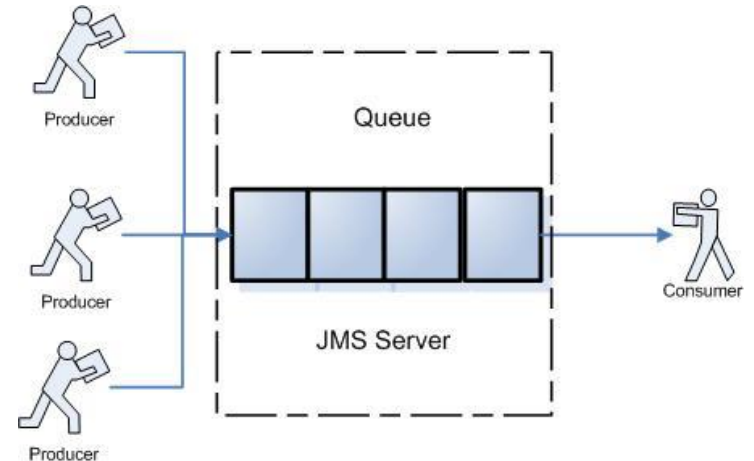
```
@FXML
private void handleDeleteButtonAction(ActionEvent event) {
    Response r= client
        .target("http://localhost:8080/BackendDemoFootball/webapi/teams/2")
        .request(MediaType.APPLICATION_JSON)
        .delete();
    text.appendText(r.toString());
}
```

JMS - Java Message Service



Asynkron kommunikation i java

- JMS API (Specifikation)
- Java Message Service
- Möjliggör en pålitlig och flexible service för asynkron kommunikation mellan klient och server.



JMS Provider

- Klienten och servern kommunicerar genom en JMS Provider.
- Kommunikationen är asynkron vilket betyder att klient inte behöver vänta på svar från servern utan fortsätter med exekveringsflödet.
- Samma som att två människor skickar epost till varandra men JMS fungerar mellan applikationer.
- Vid användning av JMS räknas båda applikationerna som klient då båda använder JMS Server.