REST

REpresentational **S**tate **T**ransfer

What is REST?

- REST is an architectural style based on Web Standards and HTTP protocols. Everything in REST is a Resource.
- Every resource supports HTTP common methods identified by global IDs (typically URIs)
- The REST arch is Client driven. A REST server waits for REST clients to request access for information.
- REST Allows for resources to have different Representations: String, XML, JSON, HTML etc. (MIMEs:Multipurpose Internet Mail Extensions)

How Do We Use REST?

- **JAX-RS** defines REST support via Java Specification Request (JAX-RS) which uses Annotations to define REST relevance of Java Classes
- **Jersey** provides Restful web services in Java Servlet container. Also provides client library to communicate with RESTful web service. (Obtainable through Maven)
- **GlassFish** Jersey is now included in the GlassFish Java EE server.

HTTP Methods

Basic HTTP operations/methods are PUT, GET, POST DELETE and are used by REST Services.

- **GET:** defines reading access of the resource. IE: FROM (HQL)
- **PUT:** Creates new Resource. IE: INSERT INTO (HQL)
- **DELETE:** Removes resource. IE: DELETE (HQL)
- **POST:** Updates an existing resource. IE: UPDATE (HQL)

Jersey Annotations

Table 1. JAX-RS annotations

Annotation	Description
@PATH(your_path)	Sets the path to base URL + /your_path. The base URL is based on your application name, the servlet and the URL pattern from the `web.xml` configuration file.
@POST	Indicates that the following method will answer to an HTTP POST request.
@GET	Indicates that the following method will answer to an HTTP GET request.
@PUT	Indicates that the following method will answer to an HTTP PUT request.
@DELETE	Indicates that the following method will answer to an HTTP DELETE request.
@Produces(MediaType.TEXT_PLAIN[, more-types])	@Produces defines which MIME type is delivered by a method annotated with @GET. In the example text ("text/plain") is produced. Other examples would be "application/xml" or "application/json".
@Consumes(type[, more-types])	@Consumes defines which MIME type is consumed by this method.
@PathParam	Used to inject values from the URL into a method parameter. This way you inject, for example, the ID of a resource into the method to get the correct object.

@Consumes

- Used when accepting a resource from a client marshal and sets the representational MIME.
- The server CONSUMES the MIME sent from the Client
- Typically used with a methods denoted with a @POST or @PUT

@Produces

- The server marshals the resource into a MIME upon client request
- Typically used with methods denoted with a @GET

DEMO

Quick Visual Demonstration