RESTEasy

What does REST mean?

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

It is style of architecture which says how the resources should be *adressed*, *manipulated* and *represented*.

REST(ful) principles

- Every resource (object) has an URI
- The URI is unique - `http://myturtle.com/turtle/111`
- Resource manipulation is done with 4 HTTP methods:

GET, **PUT**, **POST**, **DELETE** (read, update, create and delete)

- Data representation is negotiated between client and server
- Client says what it wants (Accepts* headers), server says what it gave (Content-type header)

JAX-RS

- In java we use *Java Api for RESTful Web Services (JAX-RS)*
- JAX-RS is defined in JSR-311
- RESTEasy is implementation of JAX-RS 2.0
- Other implementatios - Jersey, Apache CXF

Simple JAX-RS resource

```
@Path("/rest")
public class HelloResource {

    @GET
    @Path("/hello")
        public String doGet() {
            return "Hello World";
        }
    }
}
```

This resource matches the uri

http://myturtle.com/rest/hello

Uri parameters

```
@Path("/rest")
public class HelloResource {

    @GET
    @Path("/hello")
        public String doGet(@QueryParam("name") String name)) {
            return "Hello" + name;
        }
    }
}
```

This resource matches the uri:

http://myturtle.com/rest/hello?name=Katka

Other annotations for URI parameters:

@PathParam, @MatrixParam, @HeaderParam ...

Response data format - XML example

```
@Path("/rest")
public class HelloResource {

    @GET
    @Path("/hello")
    @Produces("application/xml")
        public HelloEntity doGet(@QueryParam("name") String name)) {
            return new HelloEntity(name);
        }
    }
}

@XmlRootElement
public class HelloEntity {

    private String name;

    @XmlElement
    // getter name
}
```

How it works? - Entity providers!

- The Entity providers are objects which know how to read/write data from/to the request/response
- RESTEasy has built-in JAXB entity provider to convert Objects <-> XML
- The format of the data in the Response is determined by **@Produces/@Consumes** annotations
- RESTEasy supports also other providers JSON, File, Atom, Yaml, Plain Text, Html ...
- And you can write your custom entity provider as well

What else JAX-RS offers?

- Client - send http requests programatically!
- Client/Server request/response filters - modify request/response http headers!
- Reader/Writer interceptors - modify body message before/after processing by the resource!
- Integration with other technologies - EJB, CDI, bean validation

Demo application

References

- https://docs.oracle.com/javaee/7/tutorial/jaxrs.htm
- Book RESTfull Java with JAX-RS 2.0 from Bill Burke
- https://www.slideshare.net/mojavelinux/cdi-seam-resteasy-you-havent-seen-rest-yet