REST API

Representational State Transfer (REST) is a software architecture style consisting of guidelines and best practices for creating scalable web services. REST is a simpler alternative to SOAP and WSDL-based Web services.

REST APIs communicate over the Hypertext Transfer Protocol (HTTP) (GET, POST, PUT, DELETE) used by web browsers.

CRUD

Simple transaction methods can be mapped to HTTP messages.

- **C**REATE - POST
- **R**EAD - GET
- **U**PDATE - PUT
- **D**ELETE - DELETE

JBOSS RestEasy

http://docs.jboss.org/resteasy/docs/3.0.7.Final/userguide/html_single/index.html

JSON (JavaScript Object Notation)

Simple XML example of a data:

```
<menu id="file" value="File">
  <popup>
    <menuitem value="New" onclick="CreateNewDoc()" />
    <menuitem value="Open" onclick="OpenDoc()" />
    <menuitem value="Close" onclick="CloseDoc()" />
  </popup>
</menu>
```

The same in JSON format:

```
{"menu": {
  "id": "file",
  "value": "File",
  "popup": {
  "menuitem": [
    {"value": "New", "onclick": "CreateNewDoc()"},
    {"value": "Open", "onclick": "OpenDoc()"},
    {"value": "Close", "onclick": "CloseDoc()"}
  ]
 }
```

```
}}
```

Rest Api Example

1.) Create a dynamic web project: REST_API

2.) Copy the following class:

```
package restapi;

import java.util.HashSet;
import java.util.Set;

import javax.ws.rs.ApplicationPath;
import javax.ws.rs.core.Application;

import org.ait.rest.Restapi;

@ApplicationPath("/rest")
public class RestApplication extends Application {
public Set<Class<?>> getClasses() {
    System.out.println("Restapi.class");
    Set<Class<?>> classes = new HashSet<Class<?>>();
    classes.add(Restapi.class);
    return classes;
}
}
```

3.) Create an other file (org.ait.rest.Restapi) with the following content:

```
package org.ait.rest;

import javax.enterprise.context.RequestScoped;
import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.PathParam;
import javax.ws.rs.Produces;
import javax.xml.bind.annotation.XmlElement;
import javax.xml.bind.annotation.XmlRootElement;

@RequestScoped
@Path("restapi/{name}")
public class Restapi {
    @GET
    @Produces("application/json")
    public Person testApi(@PathParam("name") String name) {
        Person me = new Person();
        me.name = name;
        me.age = 18;
```

```
        return me;
    }
}

@XmlRootElement(name = "Person")
class Person {
    String name;
    @XmlElement
    public String getName() {
        return name;
    }
    @XmlElement
    public int getAge() {
        return age;
    }
    int age;
}
```

4.) Browse: e.g.: http://localhost:8080/REST_API/rest/restapi/karcsi