

# Simple REST

Example `rest-xml-json` can be browsed at <https://github.com/apache/tomee/tree/master/examples/rest-xml-json>

Defining a REST service is pretty easy, simply add `@Path` annotation to a class then define on methods the HTTP method to use (`@GET`, `@POST`, ...).

## The Code

### The REST service: `@Path`, `@Produces`, `@Consumes`

Here we see a bean that uses the Bean-Managed Concurrency option as well as the `@Startup` annotation which causes the bean to be instantiated by the container when the application starts. Singleton beans with `@ConcurrencyManagement(BEAN)` are responsible for their own thread-safety. The bean shown is a simple properties "registry" and provides a place where options could be set and retrieved by all beans in the application.

Actually lines:

```
@Produces({ MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON })
@Consumes({ MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON })
```

are optional since it is the `default` configuration. And these lines can be configured by method too if you need to be more precise.

```
@Path("/greeting")
@Produces({ MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON })
@Consumes({ MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON })
public class GreetingService {
    @GET
    public Response message() {
        return new Response("Hi REST!");
    }

    @POST
    public Response lowerCase(final Request message) {
        return new Response(message.getValue().toLowerCase());
    }
}
```

# Testing

## Test for the JAXRS service

The test uses the OpenEJB ApplicationComposer to make it trivial.

The idea is first to activate the jaxrs services. This is done using `@EnableServices` annotation.

Then we create on the fly the application simply returning an object representing the web.xml. Here we simply use it to define the context root but you can use it to define your REST Application too. And to complete the application definition we add `@Classes` annotation to define the set of classes to use in this app.

Finally to test it we use cxf client API to call the REST service in `get()` and `post()` methods.

Side note: to show we use JSON or XML depending on the test method we activated on `EnableServices` the attribute `httpDebug` which prints the http messages in the logs.

```
package org.superbiz.rest;

import org.apache.cxf.jaxrs.client.WebClient;
import org.apache.openejb.jee.WebApp;
import org.apache.openejb.junit.ApplicationComposer;
import org.apache.openejb.junit.Classes;
import org.apache.openejb.junit.EnableServices;
import org.apache.openejb.junit.Module;
import org.junit.Test;
import org.junit.runner.RunWith;

import javax.ws.rs.core.MediaType;
import java.io.IOException;

import static org.junit.Assert.assertEquals;

@EnableServices(value = "jaxrs", httpDebug = true)
@RunWith(ApplicationComposer.class)
public class GreetingServiceTest {
    @Module
    @Classes(GreetingService.class)
    public WebApp app() {
        return new WebApp().contextRoot("test");
    }

    @Test
    public void getXml() throws IOException {
        final String message = WebClient.create("http://localhost:4204").path(
            "/test/greeting/")
            .accept(MediaType.APPLICATION_XML_TYPE)
```

```

        .get(Response.class).getValue();
        assertEquals("Hi REST!", message);
    }

    @Test
    public void postXml() throws IOException {
        final String message = WebClient.create("http://localhost:4204").path(
            "/test/greeting/")
            .accept(MediaType.APPLICATION_XML_TYPE)
            .post(new Request("Hi REST!"), Response.class).getValue();
        assertEquals("hi rest!", message);
    }

    @Test
    public void getJson() throws IOException {
        final String message = WebClient.create("http://localhost:4204").path(
            "/test/greeting/")
            .accept(MediaType.APPLICATION_JSON_TYPE)
            .get(Response.class).getValue();
        assertEquals("Hi REST!", message);
    }

    @Test
    public void postJson() throws IOException {
        final String message = WebClient.create("http://localhost:4204").path(
            "/test/greeting/")
            .accept(MediaType.APPLICATION_JSON_TYPE)
            .post(new Request("Hi REST!"), Response.class).getValue();
        assertEquals("hi rest!", message);
    }
}

```

## Running

Running the example is fairly simple. In the "rest-xml-json" directory run:

```
$ mvn clean install
```

Which should create output like the following.

```

/opt/softs/java/jdk1.6.0_30/bin/java -ea -Didea.launcher.port=7534
-Didea.launcher.bin.path=/opt/softs/idea/bin -Dfile.encoding=UTF-8 -classpath
/opt/softs/idea/lib/idea_rt.jar:/opt/softs/idea/plugins/junit/lib/junit-
rt.jar:/opt/softs/java/jdk1.6.0_30/jre/lib/plugin.jar:/opt/softs/java/jdk1.6.0_30/jre/
lib/javaws.jar:/opt/softs/java/jdk1.6.0_30/jre/lib/jce.jar:/opt/softs/java/jdk1.6.0_30
/jre/lib/charsets.jar:/opt/softs/java/jdk1.6.0_30/jre/lib/resources.jar:/opt/softs/jav
a/jdk1.6.0_30/jre/lib/deploy.jar:/opt/softs/java/jdk1.6.0_30/jre/lib/management-

```

agent.jar:/opt/softs/java/jdk1.6.0\_30/jre/lib/jsse.jar:/opt/softs/java/jdk1.6.0\_30/jre/lib/rt.jar:/opt/softs/java/jdk1.6.0\_30/jre/lib/ext/localedata.jar:/opt/softs/java/jdk1.6.0\_30/jre/lib/ext/sunjce\_provider.jar:/opt/softs/java/jdk1.6.0\_30/jre/lib/ext/sunpkcs11.jar:/opt/softs/java/jdk1.6.0\_30/jre/lib/ext/dnsns.jar:/opt/dev/openejb/openejb-trunk/examples/rest-xml-json/target/test-classes:/opt/dev/openejb/openejb-trunk/examples/rest-xml-json/target/classes:/home/rmannibucau/.m2/repository/org/apache/openejb/javaee-api/6.0-4/javaee-api-6.0-4.jar:/home/rmannibucau/.m2/repository/junit/junit/4.10/junit-4.10.jar:/home/rmannibucau/.m2/repository/org/hamcrest/hamcrest-core/1.1/hamcrest-core-1.1.jar:/home/rmannibucau/.m2/repository/org/apache/openejb/openejb-cxf-rs/4.5.1/openejb-cxf-rs-4.5.1.jar:/home/rmannibucau/.m2/repository/org/apache/openejb/openejb-http/4.5.1/openejb-http-4.5.1.jar:/home/rmannibucau/.m2/repository/org/apache/openejb/openejb-core/4.5.1/openejb-core-4.5.1.jar:/home/rmannibucau/.m2/repository/org/apache/openejb/mbean-annotation-api/4.5.1/mbean-annotation-api-4.5.1.jar:/home/rmannibucau/.m2/repository/org/apache/openejb/openejb-jpa-integration/4.5.1/openejb-jpa-integration-4.5.1.jar:/home/rmannibucau/.m2/repository/org/apache/commons/commons-lang3/3.1/commons-lang3-3.1.jar:/home/rmannibucau/.m2/repository/org/apache/openejb/openejb-api/4.5.1/openejb-api-4.5.1.jar:/home/rmannibucau/.m2/repository/org/apache/openejb/openejb-loader/4.5.1/openejb-loader-4.5.1.jar:/home/rmannibucau/.m2/repository/org/apache/openejb/openejb-javaagent/4.5.1/openejb-javaagent-4.5.1.jar:/home/rmannibucau/.m2/repository/org/apache/openejb/openejb-jee/4.5.1/openejb-jee-4.5.1.jar:/home/rmannibucau/.m2/repository/com/sun/xml/bind/jaxb-impl/2.1.13/jaxb-impl-2.1.13.jar:/home/rmannibucau/.m2/repository/commons-cli/commons-cli/1.2/commons-cli-1.2.jar:/home/rmannibucau/.m2/repository/org/apache/activemq/activemq-ra/5.7.0/activemq-ra-5.7.0.jar:/home/rmannibucau/.m2/repository/org/apache/activemq/activemq-core/5.7.0/activemq-core-5.7.0.jar:/home/rmannibucau/.m2/repository/org/slf4j/slf4j-api/1.7.2/slf4j-api-1.7.2.jar:/home/rmannibucau/.m2/repository/org/apache/activemq/kahadb/5.7.0/kahadb-5.7.0.jar:/home/rmannibucau/.m2/repository/org/apache/activemq/protobuf/activemq-protobuf/1.1/activemq-protobuf-1.1.jar:/home/rmannibucau/.m2/repository/commons-logging/commons-logging/1.1.1/commons-logging-1.1.1.jar:/home/rmannibucau/.m2/repository/commons-net/commons-net/3.1/commons-net-3.1.jar:/home/rmannibucau/.m2/repository/org/apache/geronimo/components/geronimo-connector/3.1.1/geronimo-connector-3.1.1.jar:/home/rmannibucau/.m2/repository/org/apache/geronimo/components/geronimo-transaction/3.1.1/geronimo-transaction-3.1.1.jar:/home/rmannibucau/.m2/repository/org/apache/geronimo/specs/geronimo-j2ee-connector\_1.6\_spec/1.0/geronimo-j2ee-connector\_1.6\_spec-1.0.jar:/home/rmannibucau/.m2/repository/org/objectweb/howl/howl/1.0.1-1/howl-1.0.1-1.jar:/home/rmannibucau/.m2/repository/org/apache/geronimo/javamail/geronimo-javamail\_1.4\_mail/1.8.2/geronimo-javamail\_1.4\_mail-1.8.2.jar:/home/rmannibucau/.m2/repository/org/apache/xbean/xbean-asm-

shaded/3.12/xbean-asm-shaded-  
3.12.jar:/home/rmannibucau/.m2/repository/org/apache/xbean/xbean-finder-  
shaded/3.12/xbean-finder-shaded-  
3.12.jar:/home/rmannibucau/.m2/repository/org/apache/xbean/xbean-reflect/3.12/xbean-  
reflect-3.12.jar:/home/rmannibucau/.m2/repository/org/apache/xbean/xbean-  
naming/3.12/xbean-naming-  
3.12.jar:/home/rmannibucau/.m2/repository/org/apache/xbean/xbean-  
bundleutils/3.12/xbean-bundleutils-  
3.12.jar:/home/rmannibucau/.m2/repository/org/hsqldb/hsqldb/2.2.8/hsqldb-  
2.2.8.jar:/home/rmannibucau/.m2/repository/commons-dbcp/commons-dbcp/1.4/commons-dbcp-  
1.4.jar:/home/rmannibucau/.m2/repository/commons-pool/commons-pool/1.5.7/commons-pool-  
1.5.7.jar:/home/rmannibucau/.m2/repository/org/codehaus/swizzle/swizzle-  
stream/1.6.1/swizzle-stream-  
1.6.1.jar:/home/rmannibucau/.m2/repository/wsd14j/wsd14j/1.6.2/wsd14j-  
1.6.2.jar:/home/rmannibucau/.m2/repository/org/quartz-scheduler/quartz/2.1.6/quartz-  
2.1.6.jar:/home/rmannibucau/.m2/repository/org/slf4j/slf4j-jdk14/1.7.2/slf4j-jdk14-  
1.7.2.jar:/home/rmannibucau/.m2/repository/org/apache/openwebbeans/openwebbeans-  
impl/1.1.6/openwebbeans-impl-  
1.1.6.jar:/home/rmannibucau/.m2/repository/org/apache/openwebbeans/openwebbeans-  
spi/1.1.6/openwebbeans-spi-  
1.1.6.jar:/home/rmannibucau/.m2/repository/org/apache/openwebbeans/openwebbeans-  
ejb/1.1.6/openwebbeans-ejb-  
1.1.6.jar:/home/rmannibucau/.m2/repository/org/apache/openwebbeans/openwebbeans-  
ee/1.1.6/openwebbeans-ee-  
1.1.6.jar:/home/rmannibucau/.m2/repository/org/apache/openwebbeans/openwebbeans-ee-  
common/1.1.6/openwebbeans-ee-common-  
1.1.6.jar:/home/rmannibucau/.m2/repository/org/apache/openwebbeans/openwebbeans-  
web/1.1.6/openwebbeans-web-  
1.1.6.jar:/home/rmannibucau/.m2/repository/org/javassist/javassist/3.15.0-  
GA/javassist-3.15.0-  
GA.jar:/home/rmannibucau/.m2/repository/org/apache/openjpa/openjpa/2.2.0/openjpa-  
2.2.0.jar:/home/rmannibucau/.m2/repository/commons-lang/commons-lang/2.4/commons-lang-  
2.4.jar:/home/rmannibucau/.m2/repository/commons-collections/commons-  
collections/3.2.1/commons-collections-  
3.2.1.jar:/home/rmannibucau/.m2/repository/net/sourceforge/serp/serp/1.13.1/serp-  
1.13.1.jar:/home/rmannibucau/.m2/repository/asm/asm/3.2/asm-  
3.2.jar:/home/rmannibucau/.m2/repository/org/apache/bval/bval-core/0.5/bval-core-  
0.5.jar:/home/rmannibucau/.m2/repository/commons-beanutils/commons-beanutils-  
core/1.8.3/commons-beanutils-core-  
1.8.3.jar:/home/rmannibucau/.m2/repository/org/apache/bval/bval-jsr303/0.5/bval-  
jsr303-0.5.jar:/home/rmannibucau/.m2/repository/org/fusesource/jansi/jansi/1.8/jansi-  
1.8.jar:/home/rmannibucau/.m2/repository/org/apache/openejb/openejb-  
server/4.5.1/openejb-server-  
4.5.1.jar:/home/rmannibucau/.m2/repository/org/apache/openejb/openejb-  
client/4.5.1/openejb-client-  
4.5.1.jar:/home/rmannibucau/.m2/repository/org/apache/openejb/openejb-  
ejbd/4.5.1/openejb-ejbd-  
4.5.1.jar:/home/rmannibucau/.m2/repository/org/apache/openejb/openejb-  
rest/4.5.1/openejb-rest-  
4.5.1.jar:/home/rmannibucau/.m2/repository/org/apache/openejb/openejb-cxf-  
transport/4.5.1/openejb-cxf-transport-

```

4.5.1.jar:/home/rmannibucau/.m2/repository/org/apache/cxf/cxf-rt-transport-
http/2.7.0/cxf-rt-transport-http-
2.7.0.jar:/home/rmannibucau/.m2/repository/org/apache/cxf/cxf-api/2.7.0/cxf-api-
2.7.0.jar:/home/rmannibucau/.m2/repository/org/apache/ws/xmlschema/xmlschema-
core/2.0.3/xmlschema-core-
2.0.3.jar:/home/rmannibucau/.m2/repository/org/apache/cxf/cxf-rt-core/2.7.0/cxf-rt-
core-2.7.0.jar:/home/rmannibucau/.m2/repository/org/apache/cxf/cxf-rt-frontend-
jaxrs/2.7.0/cxf-rt-frontend-jaxrs-
2.7.0.jar:/home/rmannibucau/.m2/repository/javax/ws/rs/javax.ws.rs-api/2.0-
m10/javax.ws.rs-api-2.0-m10.jar:/home/rmannibucau/.m2/repository/org/apache/cxf/cxf-
rt-bindings-xml/2.7.0/cxf-rt-bindings-xml-
2.7.0.jar:/home/rmannibucau/.m2/repository/org/apache/cxf/cxf-rt-rs-extension-
providers/2.7.0/cxf-rt-rs-extension-providers-
2.7.0.jar:/home/rmannibucau/.m2/repository/org/apache/cxf/cxf-rt-rs-extension-
search/2.7.0/cxf-rt-rs-extension-search-
2.7.0.jar:/home/rmannibucau/.m2/repository/org/apache/cxf/cxf-rt-rs-security-
cors/2.7.0/cxf-rt-rs-security-cors-
2.7.0.jar:/home/rmannibucau/.m2/repository/org/apache/cxf/cxf-rt-rs-security-
oauth2/2.7.0/cxf-rt-rs-security-oauth2-
2.7.0.jar:/home/rmannibucau/.m2/repository/org/codehaus/jettison/jettison/1.3/jettison-
1.3.jar:/home/rmannibucau/.m2/repository/stax/stax-api/1.0.1/stax-api-1.0.1.jar
com.intellij.rt.execution.application.AppMain
com.intellij.rt.execution.junit.JUnitStarter -ideVersion5
org.superbiz.rest.GreetingServiceTest
INFO - Cannot find the configuration file [conf/openejb.xml]. Will attempt to create
one for the beans deployed.
INFO - Configuring Service(id=Default Security Service, type=SecurityService,
provider-id=Default Security Service)
INFO - Configuring Service(id=Default Transaction Manager, type=TransactionManager,
provider-id=Default Transaction Manager)
INFO - Creating TransactionManager(id=Default Transaction Manager)
INFO - Creating SecurityService(id=Default Security Service)
INFO - Initializing network services
INFO - Creating ServerService(id=httpjbd)
INFO - Using 'print=true'
INFO - Using 'indent.xml=true'
INFO - Creating ServerService(id=cxf-rs)
INFO - Initializing network services
INFO - Starting service httpjbd
INFO - Started service httpjbd
INFO - Starting service cxf-rs
INFO - Started service cxf-rs
INFO - ** Bound Services **
INFO - NAME IP PORT
INFO - httpjbd 127.0.0.1 4204
INFO - -----
INFO - Ready!
INFO - Configuring enterprise application: /opt/dev/openejb/openejb-
trunk/examples/GreetingServiceTest
INFO - Configuring Service(id=Default Managed Container, type=Container, provider-
id=Default Managed Container)

```

```
INFO - Auto-creating a container for bean org.superbiz.rest.GreetingServiceTest:
Container(type=MANAGED, id=Default Managed Container)
INFO - Creating Container(id=Default Managed Container)
INFO - Using directory /tmp for stateful session passivation
INFO - Enterprise application "/opt/dev/openejb/openejb-trunk/examples/GreetingServiceTest" loaded.
INFO - Assembling app: /opt/dev/openejb/openejb-trunk/examples/GreetingServiceTest
INFO - Existing thread singleton service in SystemInstance() null
INFO - Created new singletonService
org.apache.openejb.cdi.ThreadSingletonServiceImpl@54128635
INFO - Succeeded in installing singleton service
INFO - OpenWebBeans Container is starting...
INFO - Adding OpenWebBeansPlugin : [CdiPlugin]
INFO - All injection points are validated successfully.
INFO - OpenWebBeans Container has started, it took 102 ms.
INFO - Deployed Application(path=/opt/dev/openejb/openejb-trunk/examples/GreetingServiceTest)
INFO - Setting the server's publish address to be http://127.0.0.1:4204/test
INFO - REST Service: http://127.0.0.1:4204/test/greeting/.* -> Pojo
org.superbiz.rest.GreetingService
FINE - ***** REQUEST *****
GET http://localhost:4204/test/greeting/
Host=localhost:4204
User-Agent=Apache CXF 2.7.0
Connection=keep-alive
Accept=application/xml
Content-Type= */*
Pragma=no-cache
Cache-Control=no-cache
```

```
*****
```

```
FINE - HTTP/1.1 200 OK
Date: Fri, 09 Nov 2012 11:59:00 GMT
Content-Length: 44
Set-Cookie: EJBSESSIONID=fc5037fa-641c-495d-95ca-0755cfa50beb; Path=/
Content-Type: application/xml
Connection: close
Server: OpenEJB/4.5.1 Linux/3.2.0-23-generic (amd64)
```

```
<response><value>Hi REST!</value></response>
INFO - Undeploying app: /opt/dev/openejb/openejb-trunk/examples/GreetingServiceTest
INFO - Stopping network services
INFO - Stopping server services
INFO - Cannot find the configuration file [conf/openejb.xml]. Will attempt to create
one for the beans deployed.
INFO - Configuring Service(id=Default Security Service, type=SecurityService,
```



```

provider-id=Default Security Service)
INFO - Configuring Service(id=Default Transaction Manager, type=TransactionManager,
provider-id=Default Transaction Manager)
INFO - Creating TransactionManager(id=Default Transaction Manager)
INFO - Creating SecurityService(id=Default Security Service)
INFO - Initializing network services
INFO - Creating ServerService(id=httpjbd)
INFO - Using 'print=true'
INFO - Using 'indent.xml=true'
INFO - Creating ServerService(id=cxf-rs)
INFO - Initializing network services
INFO - Starting service httpjbd
INFO - Started service httpjbd
INFO - Starting service cxf-rs
INFO - Started service cxf-rs
INFO - ** Bound Services **
INFO -      NAME                IP                PORT
INFO -      httpjbd             127.0.0.1        4204
INFO - -----
INFO - Ready!
INFO - Configuring enterprise application: /opt/dev/openejb/openejb-
trunk/examples/GreetingServiceTest
INFO - Configuring Service(id=Default Managed Container, type=Container, provider-
id=Default Managed Container)
INFO - Auto-creating a container for bean org.superbiz.rest.GreetingServiceTest:
Container(type=MANAGED, id=Default Managed Container)
INFO - Creating Container(id=Default Managed Container)
INFO - Using directory /tmp for stateful session passivation
INFO - Enterprise application "/opt/dev/openejb/openejb-
trunk/examples/GreetingServiceTest" loaded.
INFO - Assembling app: /opt/dev/openejb/openejb-trunk/examples/GreetingServiceTest
INFO - Existing thread singleton service in SystemInstance() null
INFO - Created new singletonService
org.apache.openejb.cdi.ThreadSingletonServiceImpl@54128635
INFO - Succeeded in installing singleton service
INFO - OpenWebBeans Container is starting...
INFO - Adding OpenWebBeansPlugin : [CdiPlugin]
INFO - All injection points are validated successfully.
INFO - OpenWebBeans Container has started, it took 11 ms.
INFO - Deployed Application(path=/opt/dev/openejb/openejb-
trunk/examples/GreetingServiceTest)
INFO - Setting the server's publish address to be http://127.0.0.1:4204/test
INFO - REST Service: http://127.0.0.1:4204/test/greeting/.* -> Pojo
org.superbiz.rest.GreetingService
FINE - ***** REQUEST *****
POST http://localhost:4204/test/greeting/
Host=localhost:4204
Content-Length=97
User-Agent=Apache CXF 2.7.0
Connection=keep-alive
Accept=application/xml

```

Content-Type=application/xml

Pragma=no-cache

Cache-Control=no-cache

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><request><value>Hi  
REST!</value></request>
```

\*\*\*\*\*

FINE - HTTP/1.1 200 OK

Date: Fri, 09 Nov 2012 11:59:00 GMT

Content-Length: 44

Set-Cookie: EJBSESSIONID=7cb2246d-5738-4a85-aac5-c0fb5340d36a; Path=/

Content-Type: application/xml

Connection: close

Server: OpenEJB/4.5.1 Linux/3.2.0-23-generic (amd64)

```
<response><value>hi rest!</value></response>
```

INFO - Undeploying app: /opt/dev/openejb/openejb-trunk/examples/GreetingServiceTest

INFO - Stopping network services

INFO - Stopping server services

INFO - Cannot find the configuration file [conf/openejb.xml]. Will attempt to create one for the beans deployed.

INFO - Configuring Service(id=Default Security Service, type=SecurityService, provider-id=Default Security Service)

INFO - Configuring Service(id=Default Transaction Manager, type=TransactionManager, provider-id=Default Transaction Manager)

INFO - Creating TransactionManager(id=Default Transaction Manager)

INFO - Creating SecurityService(id=Default Security Service)

INFO - Initializing network services

INFO - Creating ServerService(id=httpejbd)

INFO - Using 'print=true'

INFO - Using 'indent.xml=true'

INFO - Creating ServerService(id=cxf-rs)

INFO - Initializing network services

INFO - Starting service httpejbd

INFO - Started service httpejbd

INFO - Starting service cxf-rs

INFO - Started service cxf-rs

INFO - \*\* Bound Services \*\*

NAME	IP	PORT
------	----	------

httpejbd	127.0.0.1	4204
----------	-----------	------

INFO - -----

INFO - Ready!

INFO - Configuring enterprise application: /opt/dev/openejb/openejb-trunk/examples/GreetingServiceTest

INFO - Configuring Service(id=Default Managed Container, type=Container, provider-id=Default Managed Container)

INFO - Auto-creating a container for bean org.superbiz.rest.GreetingServiceTest: Container(type=MANAGED, id=Default Managed Container)

INFO - Creating Container(id=Default Managed Container)

INFO - Using directory /tmp for stateful session passivation

```

INFO - Enterprise application "/opt/dev/openejb/openejb-
trunk/examples/GreetingServiceTest" loaded.
INFO - Assembling app: /opt/dev/openejb/openejb-trunk/examples/GreetingServiceTest
INFO - Existing thread singleton service in SystemInstance() null
INFO - Created new singletonService
org.apache.openejb.cdi.ThreadSingletonServiceImpl@54128635
INFO - Succeeded in installing singleton service
INFO - OpenWebBeans Container is starting...
INFO - Adding OpenWebBeansPlugin : [CdiPlugin]
INFO - All injection points are validated successfully.
INFO - OpenWebBeans Container has started, it took 10 ms.
INFO - Deployed Application(path=/opt/dev/openejb/openejb-
trunk/examples/GreetingServiceTest)
INFO - Setting the server's publish address to be http://127.0.0.1:4204/test
INFO - REST Service: http://127.0.0.1:4204/test/greeting/.* -> Pojo
org.superbiz.rest.GreetingService
FINE - ***** REQUEST *****
GET http://localhost:4204/test/greeting/
Host=localhost:4204
User-Agent=Apache CXF 2.7.0
Connection=keep-alive
Accept=application/json
Content-Type=/*/*
Pragma=no-cache
Cache-Control=no-cache

*****

FINE - HTTP/1.1 200 OK
Date: Fri, 09 Nov 2012 11:59:00 GMT
Content-Length: 33
Set-Cookie: EJBSESSIONID=7112a057-fc4c-4f52-a556-1617320d2275; Path=/
Content-Type: application/json
Connection: close
Server: OpenEJB/4.5.1 Linux/3.2.0-23-generic (amd64)

{"response":{"value":"Hi REST!"}}
INFO - Undeploying app: /opt/dev/openejb/openejb-trunk/examples/GreetingServiceTest
INFO - Stopping network services
INFO - Stopping server services
INFO - Cannot find the configuration file [conf/openejb.xml]. Will attempt to create
one for the beans deployed.
INFO - Configuring Service(id=Default Security Service, type=SecurityService,
provider-id=Default Security Service)
INFO - Configuring Service(id=Default Transaction Manager, type=TransactionManager,
provider-id=Default Transaction Manager)
INFO - Creating TransactionManager(id=Default Transaction Manager)
INFO - Creating SecurityService(id=Default Security Service)
INFO - Initializing network services
INFO - Creating ServerService(id=httpejbd)

```

```

INFO - Using 'print=true'
INFO - Using 'indent.xml=true'
INFO - Creating ServerService(id=cxf-rs)
INFO - Initializing network services
INFO - Starting service httppejbd
INFO - Started service httppejbd
INFO - Starting service cxf-rs
INFO - Started service cxf-rs
INFO - ** Bound Services **
INFO -   NAME                IP                PORT
INFO -   httppejbd           127.0.0.1        4204
INFO - -----
INFO - Ready!
INFO - Configuring enterprise application: /opt/dev/openejb/openejb-
trunk/examples/GreetingServiceTest
INFO - Configuring Service(id=Default Managed Container, type=Container, provider-
id=Default Managed Container)
INFO - Auto-creating a container for bean org.superbiz.rest.GreetingServiceTest:
Container(type=MANAGED, id=Default Managed Container)
INFO - Creating Container(id=Default Managed Container)
INFO - Using directory /tmp for stateful session passivation
INFO - Enterprise application "/opt/dev/openejb/openejb-
trunk/examples/GreetingServiceTest" loaded.
INFO - Assembling app: /opt/dev/openejb/openejb-trunk/examples/GreetingServiceTest
INFO - Existing thread singleton service in SystemInstance() null
INFO - Created new singletonService
org.apache.openejb.cdi.ThreadSingletonServiceImpl@54128635
INFO - Succeeded in installing singleton service
INFO - OpenWebBeans Container is starting...
INFO - Adding OpenWebBeansPlugin : [CdiPlugin]
INFO - All injection points are validated successfully.
INFO - OpenWebBeans Container has started, it took 10 ms.
INFO - Deployed Application(path=/opt/dev/openejb/openejb-
trunk/examples/GreetingServiceTest)
INFO - Setting the server's publish address to be http://127.0.0.1:4204/test
INFO - REST Service: http://127.0.0.1:4204/test/greeting/.* -> Pojo
org.superbiz.rest.GreetingService
FINE - ***** REQUEST *****
POST http://localhost:4204/test/greeting/
Host=localhost:4204
Content-Length=97
User-Agent=Apache CXF 2.7.0
Connection=keep-alive
Accept=application/json
Content-Type=application/xml
Pragma=no-cache
Cache-Control=no-cache

<?xml version="1.0" encoding="UTF-8" standalone="yes"?><request><value>Hi
REST!</value></request>
*****

```

```
FINE - HTTP/1.1 200 OK
Date: Fri, 09 Nov 2012 11:59:01 GMT
Content-Length: 33
Set-Cookie: EJBSSESSIONID=50cf1d2b-a940-4afb-8993-fff7f9cc6d83; Path=/
Content-Type: application/json
Connection: close
Server: OpenEJB/4.5.1 Linux/3.2.0-23-generic (amd64)

{"response":{"value":"hi rest!"}}
INFO - Undeploying app: /opt/dev/openejb/openejb-trunk/examples/GreetingServiceTest
INFO - Stopping network services
INFO - Stopping server services
```