

JAX-RS extension

Introduction

This Restlet Extension implements the Java Specification [JAX-RS: Java API for RESTful Web Services](#)¹. Note that this specification is not final yet, and its implementation also, of course.

Description

To run this example, you need the Restlet libraries. Download a 2.0 version from www.restlet.org/downloads/². (For a general Restlet example take a look at [the first steps examples](#)³).

Now create a new Java Project, and add the following jars (resp. projects) to the classpath (right click on project, Properties, Java Build Path, Libraries (resp. Projects), Add):

- org.restlet (the core Restlet API)
- org.restlet.ext.jaxrs (the JAX-RS Runtime)
- javax.ws.rs (the JAX-RS API and also the specification)

Depending of your needs you have to add the following:

- if you want to use the provider for javax.xml.transform.DataSource: add javax.activation and javax.mail
- if you want to use the provider for JAXB: add javax.xml.bind and javax.xml.stream
- if you want to use the provider for JSON: add org.json

Click "Ok" twice. Now you are ready to start. - First we will create an example root resource class and then show how to get it running by the Restlet JAX-RS extension.

For additional details, please consult the [Javadocs](#)⁴.

Create JAX-RS example

Create a new package, e.g. test.restlet.jaxrs

Create a root resource class

First create an easy root resource class: Create a new java class named **EasyRootResource** in the previously created package and insert the following source code:

```
package test.restlet.jaxrs;

import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.Produces;

@Path("easy")
public class EasyRootResource {

    @GET
    @Produces("text/html")
```

-
1. <https://jsr311.dev.java.net/>
 2. <http://www.restlet.org/downloads/>
 3. daisy:318-restlet (First steps)
 4. <http://www.restlet.org/documentation/2.0/jse/ext/org/restlet/ext/jaxrs/package-summary.html>

```

public String getHtml() {
    return "<html><head></head><body>\n"
        + "This is an easy resource (as html text).\n"
        + "</body></html>";
}

@GET
@Produces("text/plain")
public String getPlain() {
    return "This is an easy resource (as plain text)";
}
}

```

Create Application

To provide a collection of root resource classes (and others) for a JAX-RS runtime you integrate these classes to an Application. Create a new class **ExampleApplication** in the same package with the following content:

```

package test.restlet.jaxrs;

import java.util.HashSet;
import java.util.Set;
import javax.ws.rs.core.Application;

public class ExampleApplication extends Application {

    public Set<Class<?>> getClasses() {
        Set<Class<?>> rrcs = new HashSet<Class<?>>();
        rrcs.add(EasyRootResource.class);
        return rrcs;
    }
}

```

The root resource class and the Application is specified by the JAX-RS specification. It can be used in any JAX-RS runtime environment.

Now create a runtime environment instance and pass the Application instance to it. This is runtime environment specific. Below you see this for the Restlet JAX-RS environment:

Set up a JAX-RS server

A JAX-RS server using the Restlet JAX-RS extension is set up like any Restlet server. Create a third class in the same package, named **ExampleServer**:

```

package test.restlet.jaxrs;

import org.restlet.Component;
import org.restlet.Server;
import org.restlet.data.Protocol;
import org.restlet.ext.jaxrs.JaxRsApplication;

public class ExampleServer {

    public static void main(String[] args) throws Exception {
        // create Component (as ever for Restlet)
        Component comp = new Component();
        Server server = comp.getServers().add(Protocol.HTTP, 8182);

        // create JAX-RS runtime environment
        JaxRsApplication application = new JaxRsApplication(comp.getContext());
    }
}

```

```

// attach Application
application.add(new ExampleApplication());

// Attach the application to the component and start it
comp.getDefaultHost().attach(application);
comp.start();

System.out.println("Server started on port " + server.getPort());
System.out.println("Press key to stop server");
System.in.read();
System.out.println("Stopping server");
comp.stop();
System.out.println("Server stopped");
}
}

```

Start this class, open a browser and request <http://localhost:8182/easy>. Now you see the HTML representation. If you request the same URI with accepted media type "text/plain", you get a plain text representation.

This example (a little bit extended) is available in the project `org.restlet.example`. See package `org.restlet.test.jaxrs`. There is another root resource class with a reachable resource class and also an example with user authentication.

A lot of more resource classes are available in the test project (`org.restlet.test`, packages starting with `org.restlet.test.jaxrs`). They are implemented for testing, and most of them do not do intelligent things ... :-). But they show the actual status of development of this JAX-RS runtime environment.

This runtime environment is still under development, and I'm very busy continuing it ...

Run in a Servlet Container

If you want to run the JAX-RS Application in a Servlet Container, create a subclass of the `JaxRsApplication`. In the constructor you could attach the Application and sets the Guard and the RoleChecker (if needed).

```

public class MyJaxRsApplication extends JaxRsApplication {

    public MyJaxRsApplication(Context context) {
        super(context);
        this.add(new ExampleApplication());
        this.setGuard(...); // if needed
        this.setRoleChecker(...); // if needed
    }
}

```

For details to run this Application in a Servlet Container take a look at [Restlet FAQ](#)⁶.

You could use this subclass also in the example above:

```

// create JAX-RS runtime environment
Application application = new MyJaxRsApplication(comp.getContext());

// if you use this kind, you don't need to attach the Application again.

```

Comments are welcome to the [Restlet mailing list](#)⁷ or directly to Stephan.Koops@web.de !

6. daisy:333-restlet (FAQ)

7. <http://www.restlet.org/community/lists>

This extension is the result of a (german) [master thesis](#)⁸.

8. <http://users.informatik.haw-hamburg.de/%7Eubicomp/arbeiten/master/koops.pdf>