

Jersey

RESTful Web Services in Java.

What is Jersey?

Jersey is Java (JAX-RS: JSR 311 & JSR 339).

It takes the pain out of writing RESTful web services by providing a set of simple annotations to describe features and functions of your application.

RESTful Service

```
1 package path.where.your.service.classes.are;
2
3 @Path("/my/something")
4 @Produces({ "application/json" })
5 public class SomethingService extends AuthenticatedService
6 {
7     private static Logger logger = LoggerFactory.getLogger( SomethingService.class );
8
9     public SomethingService( @Context HttpServletRequest request )
10    {
11        super( request );
12    }
13
14    @POST
15    @Path("/create/something")
16    public Response createSomethingFromCompound( String json )
17    {
18        // do something with the request body
19
20        return Response.ok().build();
21    }
22 }
23
```

What did that do?

The class itself is quickly tagged with the base URL that this class responds for
`@Path("/my/something")`

It also specifies the return content-type for the whole class
`@Produces({"application/json"})`

Then, individual methods can be annotated with calling methods
`@POST`

And even further depth to the class level `@PATH`
`@Path("/create/something")`

To have the webserver invoke `createSomethingFromCompund()` the user would POST
data to `/my/something/create/something`

Benefits

No more:

- * Setting content-type
- * External mapping of URLs
- * Action Decision Tree (if POST then X)

Inputs

So, we have seen how it can direct requests and simplify responses, but it also simplifies the inputs.

@Consumes - set the expected request content-type

@Param - parse query string parameters

`@Consumes`

`@Consumes("text/plain")`

Methods can be annotated with an expected request content type to ensure your method is getting the right input.

@Param

```
public Response requestHandler(  
    @DefaultValue("2") @QueryParam("step") int step,  
    @DefaultValue("true") @QueryParam("max-m") boolean hasMax,  
    @DefaultValue("green") @QueryParam("max-color") ColorParam maxColor) {}
```

Define expectation and defaults in the method declaration.

Custom Java types can be used. They just need a constructor that accepts a String

Running it

Java 6+ HTTP Server

Servlet 2.x/3.x Containers (Tomcat, Jetty, etc)

J2EE

OSGi

Any Java cloud service

Client API

In addition to simplifying the creation of web services...

Jersey offers a suite of tools for writing clients that interact with web services.

Client Features

Simplified HTTP Client

Filters for processing responses

Support for Reactive Client (avoiding: call back hell)

Scratching the Surface

This is just the beginning...

Learn more

[Jersey User's Guide](#)

[Jersey Examples](#)

