

TECH MISC

“Talk is cheap, show me the code”

–Linus Torvalds

HTTP — Status codes

- * 2xx: success
- * 3xx: redirection
- * 4xx: client error
- * 5xx: server error

HTTP 2xx

- * 200 OK —> everything was fine
- * 201 Created —> OK + resource created
- * 202 Accepted —> long request accepted
- * 204 No Content —> OK, no response

HTTP 3xx

- * 301 Moved Permanently —> browser should not hit this URL anymore in favor of the redirection
- * 303 See Other —> The response can be found at the redirection URL via GET
- * 307 Temporary Redirect —> browser should redirect but this URL will be back as the main one

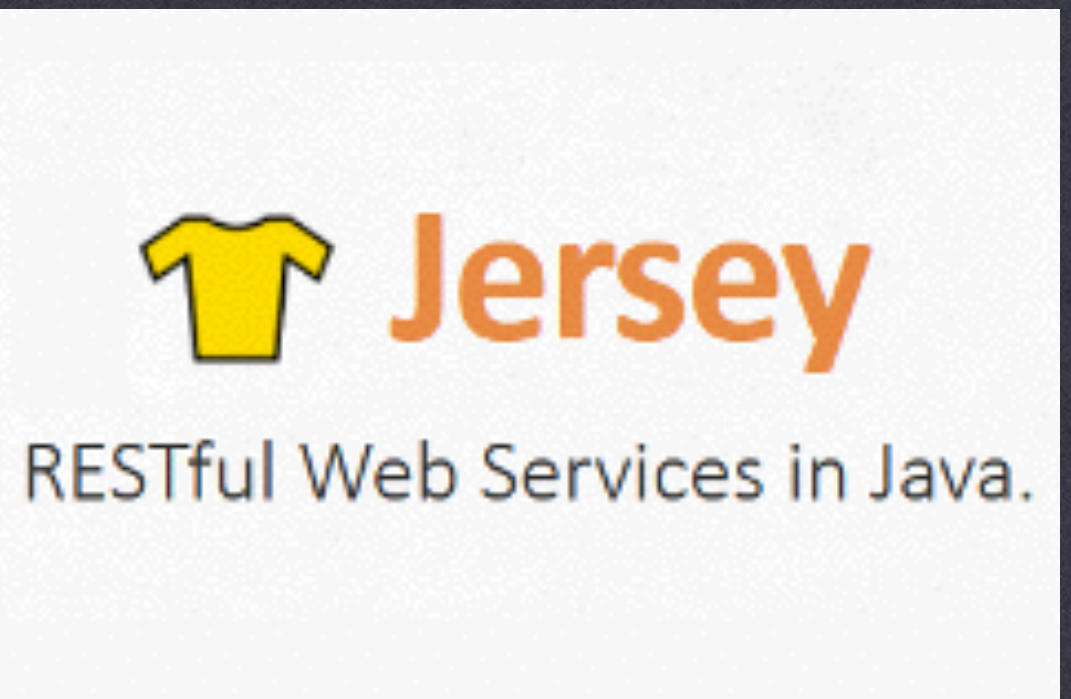
HTTP 4xx

- * 400 Bad Request —> malformed client request
- * 401 Unauthorized —> unauthenticated
- * 403 Forbidden —> authenticated but not authorized
- * 404 Not Found —> resource does not exist
- * 405 Method Not Allowed

HTTP 5xx

- * 500 Internal Server Error —> generic server error
- * 501 Not Implemented —> Method not implemented (in the future plans)
- * 503 Service Unavailable —> server temporarily unavailable (load, maintenance)

JAX-RS



JAX-RS

- * aka JSR 311 & JSR 339
- * Jersey is the reference implementation
- * Annotation-based framework to specify REST services

Useful Annotations

```
@GET
public Response getHTML() {
    ...
}
```

```
@GET
@Produces("application/json")
public Response getJSON() {
    ...
}
```

```
@GET
@Produces("application/xml")
@Path("xml/{firstName}")
public Response getXML() {
    ...
}
```

```
@GET
@Produces("application/json")
@Path("json/{firstName}")
public Response getJSON(@PathParam("firstName") String firstName) {
    ...
}
```


Useful Annotations

```
@GET
@Produces("application/json")
@Path("json/companyList")
public Response get(
    @QueryParam("start") int start,
    @QueryParam("limit") int limit
)
{...}
```

```
@POST
@Consumes("application/json")
@Produces("application/json")
public Response create(Contact contact) {
    ...
}
```

```
@PUT
@Consumes("application/json")
@Produces("application/json")
@Path("{contactId}")
public Response update(Contact contact) {
    ...
}
```

```
@DELETE
@Produces("application/json")
@Path("{contactId}")
public Response delete(@PathParam("contactId") int contactId) {
    ...
}
```


Response builder

- * `Response.ok("response").build()` — 200
- * `Response.serverError().build()` — 500
- * `Response.created(uri).build()` — 201
- * ...
- * `Response.status(Response.Status.BAD_REQUEST).entity("msg").build()`

And now some code