

RESTFul Services In Java Using Jersey

HTTP GET Request

Bryan Hansen
twitter: bh5k

<http://www.linkedin.com/in/hansenbryan>



pluralsight 
hardcore developer training

GET



myresource

- We have done a basic example of a GET already:

```
@GET
@Produces(MediaType.TEXT_PLAIN)
public String getIt() {
    return "Got it!";
}
```

- Annotated with @GET
- Mapping is defined with @Path
- The URL is defined through our context/servlet mapping/@Path
- Content returned is defined with @Produces
 - @Produces can take an Array of multiple types

ActivityResource

- Let's create a Service to retrieve all activities
 - Create Activity
 - Create ActivityRepository
 - Create ActivityResource
 - Annotate with GET
- <http://localhost:8080/exercise-services/webapi/activities>



@XmlRootElement

- **MessageBodyWriter not found for media type=application/xml**
- **Jersey doesn't know how to marshal POJOs to XML**
 - Need to define what the XmlRootElement is
 - Can also rename other elements
 - @XmlElement
 - Can specify attributes too
- **Uses JAXB to do Java-to-XML**

```
▼<activities>
  ▼<activity>
    <description>Swimming</description>
    <duration>55</duration>
  </activity>
  ▼<activity>
    <description>running</description>
    <duration>30</duration>
  </activity>
</activities>
```

JSON

- **MessageBodyWriter not found for media type=application/json**
- **JSON requires another Jar to be added to our pom.xml**
 - jersey-media-moxy
- **@Produces(MediaType.APPLICATION_JSON)**
- **A weird gotcha: JAXB converts the POJO to XML and then XML to JSON**
 - Our XML annotations affect JSON output
 - Notice desc/description

{JSON}

@Produces

- Can be used to produce multiple output formats
- @Produces can take a String or an Array of Strings
 - @Produces({MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML})
- Need to use something besides our browser to set accept header
 - Most browsers support some type of posting tool
 - Chrome - Postman
 - Firefox - Poster

Specific Objects

- **How do we retrieve a specific Activity**

- <http://localhost:8080/exercise-services/webapi/activities/1234>
- Use @PathParam to grab URL params
- @PathParam("paramName")
- Can nest @Path references

- **This applies to nested objects as well**

- <http://localhost:8080/exercise-services/webapi/activities/1234/user>

```
▼<activity>
  <desc>Cycling</desc>
  <duration>55</duration>
  ▼<user>
    <id>bryanId</id>
    <name>Bryan Hansen</name>
  </user>
</activity>
```


Summary

- Reviewed Basic GET
- Created New Service
- XmlRootElement
- JSON
- @Produces
- Specific/Nested Objects