

Session 13

RESTful Services Part 2

1

Lecture Objectives

- Understand how to pass parameters to a Web services
- Understand how to return values from a Web service

© Robert Kelly, 2018

2

Reading & References

■ Reading

Be careful - other JAX-RS documentation assumes knowledge of other Java EE technologies (e.g., JPA)

■ Tutorials

https://javabrainz.io/courses/javaee_jaxrs/docs.oracle.com/javaee/7/tutorial/webservices-intro.htm#GIJTI
(Chapters 29.2.4 - 29.4)

■ Reference

Session material follows

■ Java EE API

Java EE 7 Tutorial text

docs.oracle.com/javaee/7/api/javax/ws/rs/package-summary.html

■ Book

RESTful Java Web Services, 3rd Edition,
<https://www.amazon.com/RESTful-Java-Web-Services-pragmatic/dp/1788294041>

How Do You Pass Parameters to a RESTful Service?

- Without using the servlet parameters (request and response) directly, we need a different way to pass parameters from client to server

■ Use the URL

- URI components become an argument to the method responding to the request

<http://example.com/users/myname>

Acts as a parameter

■ Use the query string (form data set)

- Parameters are mapped to arguments in the method signature

Other Data Passed to the Service

- You can also obtain the following items in your service
- Query
- URI path
- Form
- Cookie
- Header
- Matrix

© Robert Kelly, 2018

5

Extracting Query Parameters - URL Query String

- Your web service can extract parameters contained in form dataset

```
@Path("smooth")      Remember the form dataset is      Instantiated
@GET                  contained in the URL for a GET      with the
public Response smooth(                                     user-defined
    @DefaultValue("2") @QueryParam("step") int step,      class
    @DefaultValue("true") @QueryParam("min-m") boolean hasMin,
    @DefaultValue("true") @QueryParam("max-m") boolean hasMax,
    @DefaultValue("true") @QueryParam("last-m") boolean hasLast,
    @DefaultValue("blue") @QueryParam("min-color") ColorParam minColor,
    @DefaultValue("green") @QueryParam("max-color") ColorParam maxColor,
    @DefaultValue("red") @QueryParam("last-color") ColorParam lastColor
) { ... }
```

Notice that parameters are parsed into Java types

Missing parameters assume default value

A 400 error code is returned if parameter cannot be parsed

© Robert Kelly, 2018

6

Are We On Track?

- Download the html file - `CSE336-Track-Services.html`
- Write the Web service that will respond to a submit (get specified in the form tag) from the download file, and if the card# is 123, return an html page that says "Welcome back Alonzo" Otherwise, the return page will say "Would you like to apply for a library card?"

Library Card Application

Complete this application and click the Submit button. You may pick up your new card at any branch library or Central Library or have the card mailed to you. In order to activate your card, you must visit your library in person and provide acceptable ID.

* Required

Library Card

Card Number (3 digits)

Name

* Card Type:

Young Adults (Ages 13 - 16)

Adult (Ages 17 and over)

Seniors (Ages 62 and over)

© Robert Kelly, 2018

7

Were We On Track ...

- ApplicationConfig** `<form method="get" action="http://localhost:8080/CSE336-Services/library/librarycards" ...>`

```
@javax.ws.rs.ApplicationPath("library")
public class ApplicationConfig extends Application {

    @Override
    public Set<Class<?>> getClasses() {
        Set<Class<?>> resources = new java.util.HashSet<>();
        addRestResourceClasses(resources);
        return resources;
    }

    private void addRestResourceClasses(Set<Class<?>> resources) {
        resources.add(lectures.HelloWorld.class);
        resources.add(lectures.LibrarycardsResource.class);
    }
}
```

© Robert Kelly, 2018


8

... Were We On Track?

■ GET request handler

```
@Path("librarycards")
public class LibrarycardsResource {
    ...
    @GET
    @Produces(MediaType.TEXT_HTML)
    public String getText(@QueryParam("cnum") int cardNumber) {
        String s1 = "<html><body><h1>";
        String s2 = "</h1></body></html>";
        String message = "";
        if (cardNumber==123){
            message="Welcome back Alonzo";
            return s1+message+s2; }
        else {
            return s1 + "Would you like to apply for a library
card?" + s2; } } }
```

Note that the *cardnumber* parameter was an *int* (not a *String*)



© Robert Kelly, 2018

9

Extracting Form Parameters from a POST Request

- Remember that form parameters in a POST request are not contained in the URL (they are in the HTTP body)

```
@POST
@Consumes("application/x-www-form-urlencoded")
public void post(@FormParam("name") String name) {
    // Store the message
}
```

Other annotation exists to extract a Map of name-value pairs

© Robert Kelly, 2018

10

Have You Achieved the Lecture Objectives?

- Understand how to pass parameters to a Web services
- Understand how to return values from a Web service