JAX-RS
Injection Annotations

Sang Shin
"Code with Passion!"

What are JAX-RS injection annotations?

- By using JAX-RS provided injection annotations, you are asking JAX-RS runtime to inject some useful information into your code
- @PathParam, @MatrixParam, @QueryParam, @FormParam, @BeanParam
- @HeaderParam, @CookieParam
- @Context

URI Path Template and @PathParam, @QueryParam, @DefaultValue, @MatrixParam, @FormParam, @BeanParam

What is URI Path Template?

- URI path templates are “URIs with variables embedded”
- The variable captures client request value

```
// Will respond to http://example.com/users/SangShin
@Path("/users/{username}")
public class UserResource {

    @GET
    @Produces("text/xml")
    //  In order to obtain the value of the username variable, @PathParam
    //  is used on a method parameter
    public String getUser(@PathParam("username") String myUserName) {
        // “myUserName” variable has value of “SangShin”
    }
}
```

@PathParam, @QueryParam

- Annotated method parameters extract client request data
  - `@PathParam` extracts information from the request URI
    - http://host/catalog/items/123
  - `@QueryParam` extracts information from the request URI query parameters
    - http://host/catalog/items/?start=0

Example: @PathParam, @QueryParam, @DefaultValue

```
@Path("/items/")
@Consumes("application/xml")
public class ItemsResource {

   // Example request: http://host/catalog/items/123
   @Path("{id}/")
   ItemResource getItemResource(@PathParam("id")Long id){
   ...
   }

   // Example request: http://host/catalog/items/?start=0
   @GET
   ItemsConverter get(@QueryParam("start")int start,
                      @QueryParam("size")
                      @DefaultValue("3")int size {
        ...
   }

}
```

URI Path template variable with Reg. Expression pattern

- @Path("users/{username: [a-zA-Z][a-zA-Z_0-9]+}")
- *username* variable will only match user names that begin with one upper or lower case letter and zero or more alpha numeric characters and the underscore character.
- If a user name does not match, then a 404 (Not Found) response will be sent

@MatrixParam

- Matrix parameters are a set of “name=value” in URI path

```
// Will respond to http://example.com/users;userName="Sang Shin";age="20"
@Path("/users")
public class UsersResource {

   @GET
   public Response getUser(
         @MatrixParam("userName") String userName,
         @MatrixParam("age") int  age) {
      String output = "User's name is " + userName + " and age is " + age;
      return Response.status(Response.Status.OK)
                  .entity(output)
                  .build();
   }
```

@FormParam

- Binds the value(s) of a form parameter contained within a request entity body to a resource method parameter

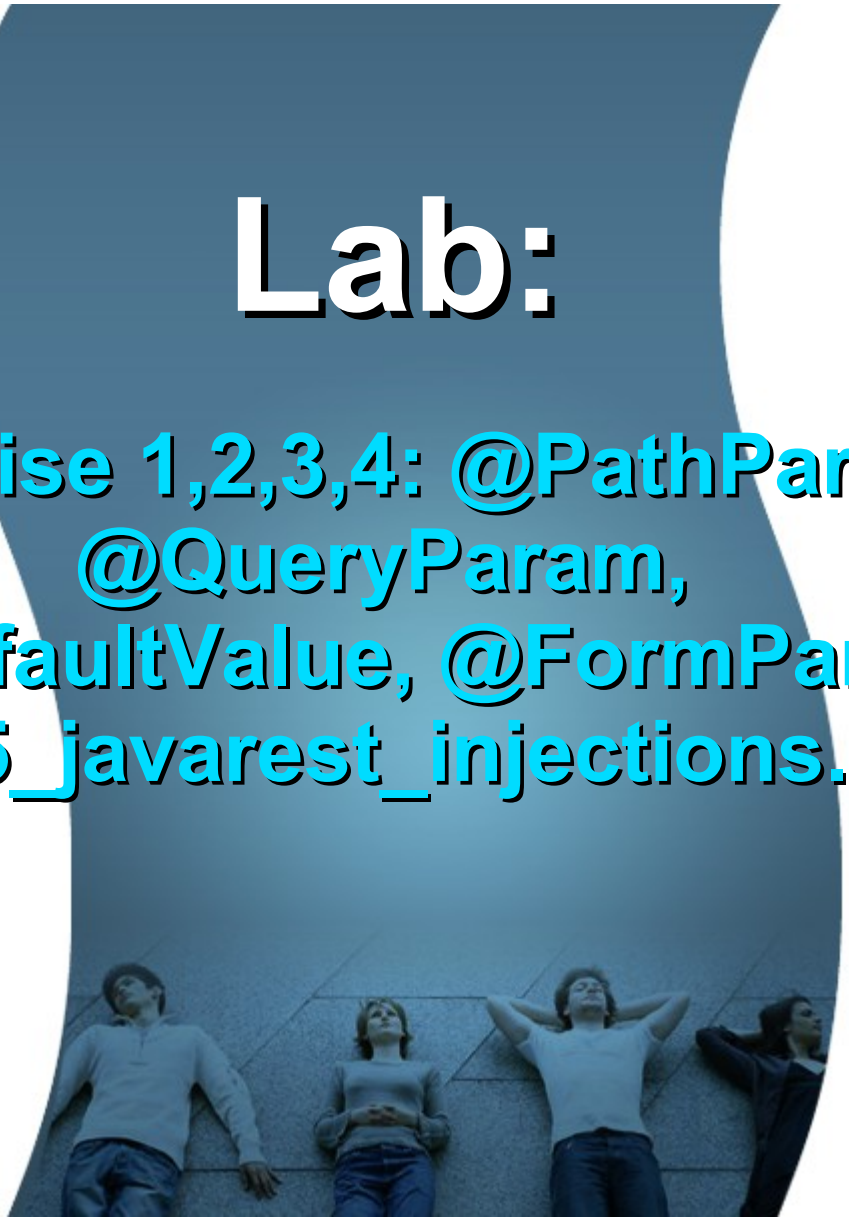
```
@POST
public Response addUser(
        @FormParam("name") String name,
        @FormParam("age") int age) {
    String output = "A User is added: " + name + ", age : " + age;
    return Response.status(200).entity(output).build();
}
```

```
<form action="resources/users" method="post">
  <p>
    Name : <input type="text" name="name" />
    Age : <input type="text" name="age" />
  </p>
  <input type="submit" value="Add User" />
</form>
```

Lab:

Exercise 1,2,3,4: @PathParam, @QueryParam, @DefaultValue, @FormParam

4365_javarest_injections.zip



@HeaderParam
@CookieParam

@HeaderParam

- Binds the value(s) of a HTTP header to a resource method parameter, resource class field, or resource class bean property

```
@GET
@Path("/get")
public Response addUser(@HeaderParam("accept") String accept) {
    String output = "addUser is called, accept header: " + accept;
    return Response.status(200).entity(output).build();

}
```

@CookieParam

- Allows you to inject cookies sent by a client request into your JAX-RS resource methods

```
@Path("/myservice")
public class MyService {

  @GET
  @Produces("text/html")
  public String get(@CookieParam("customerId") Cookie custId) {
    ...
  }
}
```

Lab:

Exercise 5: @HeaderParam, @CookieParam
4365_javarest_injections.zip

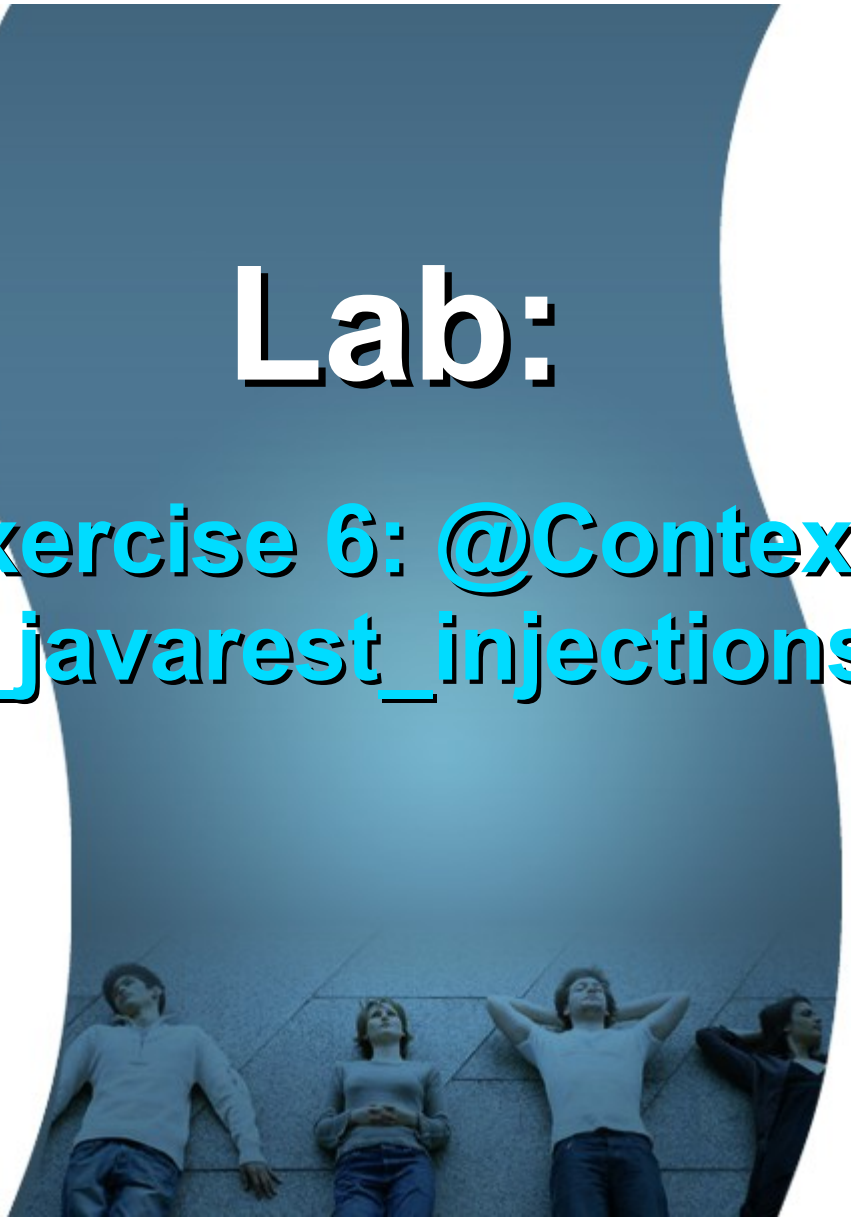
@Context

Context Object Injections via @Context

```
@Path("/users")
public class HelloWorldResource {

    @Context
    UriInfo uriInfo;

    @Context
    HttpHeaders httpHeaders;

    @Context
    SecurityContext securityContext;

    @Context
    Request request;

    @Context
    Providers providers;

...
```

Lab:

Exercise 6: @Context
4365_javarest_injections.zip



Validation

Validation

- JAX-RS supports the Bean Validation to verify JAX-RS resource classes
  - Adding constraint annotations to resource method parameters
  - Ensuring entity data is valid when the entity is passed in as a parameter

Bean validation applied to method parameters

```
// Bean Validation constraint annotations may be applied to parameters for a resource.
// The server will validate the parameters and either pass or throw a
// javax.validation.ValidationException.
@POST
@Path("/createUser")
@Consumes(MediaType.APPLICATION_FORM_URLENCODED)
public void createUser(@NotNull @FormParam("username") String username, //@NotNull is built-in constraint
                       @NotNull @FormParam("firstName") String firstName,
                       @NotNull @FormParam("lastName") String lastName,
                       @Email @FormParam("email") String email) {  // @Email is user defined constraint
...
}
```

Bean validation applied to method parameters

```
// Define @Email constraint
@Documented
@Constraint(validatedBy = EmailValidator.class)
@Size(min = 5, message = "{org.javaee7.jaxrs.resource_validation.min_size}")
@NotNull(message = "{org.javaee7.jaxrs.resource_validation.cannot_be_null}")
public @interface Email {
    String message() default "{org.javaee7.jaxrs.resource_validation.invalid_email}";
    Class<?>[] groups() default {};
    Class<? extends Payload>[] payload() default {};
}
```

Bean validation applied to fields

```
@Path("/createUser")
public class CreateUserResource {

    @NotNull
    @FormParam("username")
    private String username;

    @NotNull
    @FormParam("firstName")
    private String firstName;

    @NotNull
    @FormParam("lastName")
    private String lastName;

    @Email
    @FormParam("email")
    private String email;
     ...
}
```

Code with Passion!
JPassion.com