

Bruno Stuelp Filho – bstuelp@gmail.com

Project GitHub: <https://github.com/bstuelp/ANDDigital>

1. Technologies

- a. Java + JAX-RS + Jersey
- b. Gson
- c. Tomcat 9
- d. Postman

2. Project Structure

package com.andDigital.entity;
Entities Customer and Phones

package com.andDigital.services;
Rest Services Methods and Interface

package com.andDigital.test;
JUnit tests

3. API – URLs e Spec

- GET - <http://localhost:8080/ANDDigital/rest/customer/getAll>
Responses

- 200 – get all phone numbers
Response body: application/json: The registers

- GET - <http://localhost:8080/ANDDigital/rest/customer/getPhone/{id}>
Responses

- 200 – get all phone numbers of a single customer
Response body: application/json: The register
- 400 – Customer not found
Response body: application/json: Error message

- PUT - <http://localhost:8080/ANDDigital/rest/customer/activate>
Responses

- 200 – activate a phone number
Response body: application/json: Success message
- 400 – phone not found
Response body: application/json: Error message

Params – Json

```
{  
    "number": "664512235"  
}
```

4. Calling endpoints

- GET - `http://localhost:8080/ANDDigital/rest/customer/getAll`

The screenshot shows a REST client interface with a GET request to `http://localhost:8080/ANDDigital/rest/customer/getAll`. The response is a JSON array of two customer objects. The first object has an id of 1 and a list of four phone numbers. The second object has an id of 2 and a list of three phone numbers.

```
1 [
2   {
3     "id": 1,
4     "listPhones": [
5       {
6         "phone": "197654321"
7       },
8       {
9         "phone": "293456789"
10      },
11      {
12        "phone": "394987312"
13      },
14      {
15        "phone": "493789456"
16      }
17    ]
18  },
19  {
20    "id": 2,
21    "listPhones": [
22      {
23        "phone": "597258369"
24      },
25      {
26        "phone": "691852963"
27      },
28      {
29        "phone": "798741369"
30      }
31    ]
32  }
33 ]
```

`http://localhost:8080/ANDDigital/rest/customer/getPhone/2 -> 200`

The screenshot shows a REST client interface with a GET request to `http://localhost:8080/ANDDigital/rest/customer/getPhone/2`. The response is a JSON object representing a customer with id 2 and a list of three phone numbers. The status bar indicates a 200 OK response with a time of 23 ms and a size of 197 B.

Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body

```
1 {
2   "id": 2,
3   "listPhones": [
4     {
5       "phone": "597258369"
6     },
7     {
8       "phone": "691852963"
9     },
10    {
11      "phone": "798741369"
12    }
13  ]
14 }
```

Status: 200 OK Time: 23 ms Size: 197 B

http://localhost:8080/ANDDigital/rest/customer/getPhone/22 - > Return 400

GET http://localhost:8080/ANDDigital

http://localhost:8080/ANDDigital/rest/customer/getPhone/22

GET http://localhost:8080/ANDDigital/rest/customer/getPhone/22

Send Save

Params Authorization Headers Body Pre-request Script Tests Cookies Code Comments (0)

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 31 ms Size: 169 B Download

Pretty Raw Preview JSON

```
1 {
2   "Error": "Customer not found!"
3 }
```

http://localhost:8080/ANDDigital/rest/customer/activate - 200 - Number activated

```
{
  "number": "664512235"
}
```

GET http://localhost:8080/ANDDigital

PUT http://localhost:8080/ANDDigital

http://localhost:8080/ANDDigital/rest/customer/activate

PUT http://localhost:8080/ANDDigital/rest/customer/activate

Send Save

Params Authorization Headers (1) Body Pre-request Script Tests Cookies Code Comments (0)

none form-data x-www-form-urlencoded raw binary JSON (application/json)

Beautify

```
1 {
2   "number": "664512235"
3 }
```

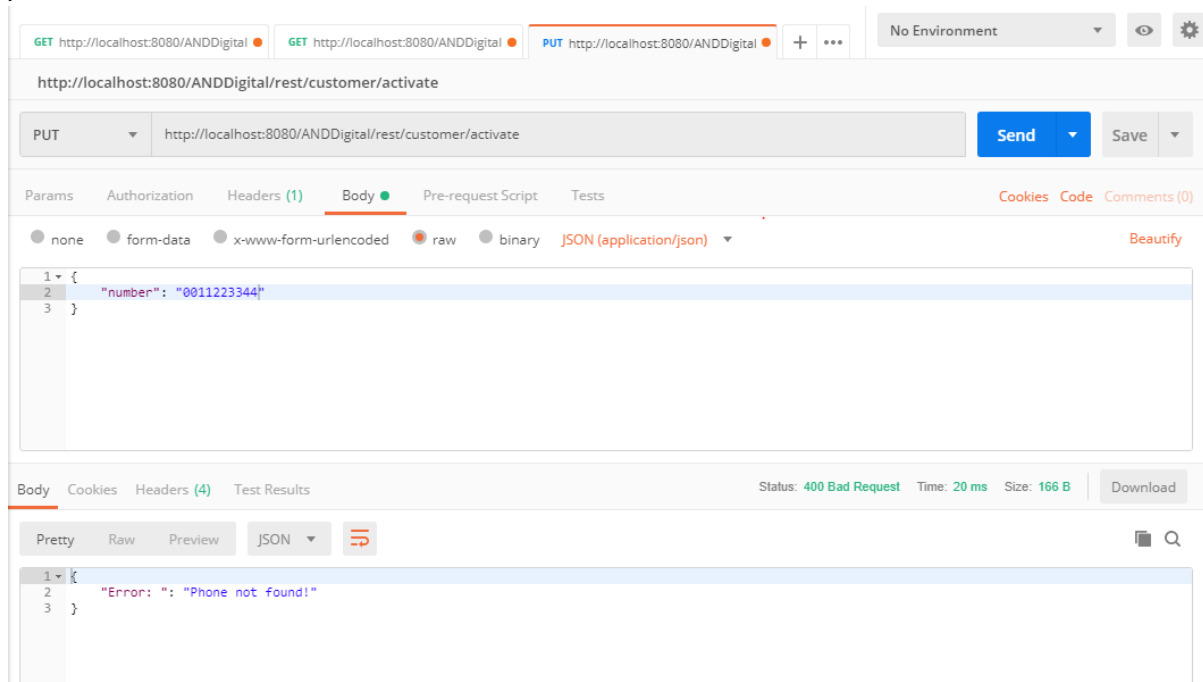
Body Cookies Headers (3) Test Results Status: 200 OK Time: 36 ms Size: 140 B Download

Pretty Raw Preview JSON

```
1 {
2   "Success": "Phone activated."
3 }
```

http://localhost:8080/ANDDigital/rest/customer/activate - 400 - Number not found.

```
{
  "number": "0011223344"
}
```



5. Unit tests and Code coverage

a. Start

The screenshot shows an IDE with two tabs: `AndDigitalTest.java` and `JUnitTest.java`. The `Services.java` file is open, showing a REST service implementation. The code includes a `Services` class that implements `Serializable` and `IServices`. It has two endpoints: `getAllPhones()` and `getPhoneById()`. The `getAllPhones()` method returns a `Response` object with a status of `OK` and a body containing a list of customers. The `getPhoneById()` method returns a `Response` object with a status of `OK` and a body containing a customer object.

Below the code, the `Coverage` tab is visible, showing a table with the following data:

Element	Coverage	Covered Instruction...	Missed Instructions	Total Instructions
ANDDigital	0,0 %	0	441	441
src	0,0 %	0	441	441
com.andDigital.services	0,0 %	0	384	384
com.andDigital.entity	0,0 %	0	40	40

b. After implementation

```
@Test
public void testGetAllPhones() {
    assertEquals(200, classUnderTest.getAllPhones().getStatus());
}
```

AndDigitalTest.java JunitTest.java Services.java

```

1 package com.andDigital.services;
2
3 import java.io.Serializable;
4
20
21 @Path("/customer")
22 public class Services implements Serializable, IServices {
23     private static final long serialVersionUID = 1L;
24     private List<Customer> listCustomer = new ArrayList<Customer>();
25     boolean activated = false;
26
27     @GET
28     @Path("/getAll")
29     @Produces(MediaType.APPLICATION_JSON)
30     @Override
31     public Response getAllPhones() {
32         listCustomer = this.createObject();
33
34         return (Response.status(Response.Status.OK).entity(listCustomer).build());
35     }
36
37     @GET
38     @Path("/getPhone/{id}")
39     @Produces(MediaType.APPLICATION_JSON)
40     @Override
41     public Response getPhoneById(@PathParam("id") int id) {
42         listCustomer = this.createObject();

```

Problems Search Console Servers Coverage

AndDigitalTest (20/02/2019 21:32:41)

Element	Coverage	Covered Instruction...	Missed Instructions	Total Instructions
ANDDigital	72,6 %	320	121	441
src	72,6 %	320	121	441
com.andDigital.services	73,7 %	283	101	384
com.andDigital.entity	50,0 %	20	20	40
com.andDigital.test	100,0 %	17	0	17

c. After implementation

```

@Test
public void testGetPhoneById() {
    assertEquals(200, classUnderTest.getPhoneById(2).getStatus());

    assertEquals(400, classUnderTest.getPhoneById(22).getStatus());
}

```

*AndDigitalTest.java Services.java

```

34         return (Response.status(Response.Status.OK).entity(listCustomer).build());
35     }
36
37     @GET
38     @Path("/getPhone/{id}")
39     @Produces(MediaType.APPLICATION_JSON)
40     @Override
41     public Response getPhoneById(@PathParam("id") int id) {
42         listCustomer = this.createObject();
43
44         Customer data = listCustomer.stream().filter(customer -> id == customer.getId()).findAny().orElse(null);
45
46         if (data == null) {
47             Map<String, String> mapError = new HashMap<String, String>();
48             mapError.put("Error: ", "Customer not found!");
49
50             return (Response.status(Response.Status.BAD_REQUEST).entity(mapError).build());
51         }
52
53         return (Response.status(Response.Status.OK).entity(data).build());
54     }
55
56     @PUT
57     @Path("/activate/{number}")
58     @Consumes(MediaType.APPLICATION_JSON)
59     @Produces(MediaType.APPLICATION_JSON)

```

Problems Search Console Servers Coverage

AndDigitalTest (20/02/2019 21:35:54)

Element	Coverage	Covered Instruction...	Missed Instructions	Total Instructions
ANDDigital	80,7 %	363	87	450
src	80,7 %	363	87	450
com.andDigital.services	81,8 %	314	70	384
com.andDigital.entity	57,5 %	23	17	40
com.andDigital.test	100,0 %	26	0	26

AndDigitalTest.java Services.java

```

34 return (Response.status(Response.Status.OK).entity(listCustomer).build());
35 }
36
37 @GET
38 @Path("/getPhone/{id}")
39 @Produces(MediaType.APPLICATION_JSON)
40 @Override
41 public Response getPhoneById(@PathParam("id") int id) {
42     listCustomer = this.createObject();
43
44     Customer data = listCustomer.stream().filter(customer -> id == customer.getId()).findAny().orElse(null);
45
46     if (data == null) {
47         Map<String, String> mapError = new HashMap<String, String>();
48         mapError.put("Error: ", "Customer not found!");
49
50         return (Response.status(Response.Status.BAD_REQUEST).entity(mapError).build());
51     }
52
53     return (Response.status(Response.Status.OK).entity(data).build());
54 }
55
56 @PUT
57 @Path("/activate/{number}")
58 @Consumes(MediaType.APPLICATION_JSON)
59 @Produces(MediaType.APPLICATION_JSON)

```

Problems Search Console Servers Coverage

AndDigitalTest (20/02/2019 21:36:51)

Element	Coverage	Covered Instruction...	Missed Instructions	Total Instructions
ANDDigital	84,3 %	386	72	458
src	84,3 %	386	72	458
com.andDigital.services	85,7 %	329	55	384
com.andDigital.entity	57,5 %	23	17	40
com.andDigital.test	100,0 %	34	0	34

d. After implementation

```

@Test
public void testActivatedNumber() {
    assertEquals(200, classUnderTest.activatedNumber("493789456").getStatus());
    assertEquals(400, classUnderTest.activatedNumber("000000000").getStatus());
}

```

AndDigitalTest.java Services.java

```

60 @Override
61 public Response activatedNumber(@PathParam("number") String number)
62 {
63     listCustomer = this.createObject();
64
65     listCustomer.forEach((Customer c) -> {
66         List<Phones> data = c.getListPhones();
67         data.forEach((Phones p) -> {
68             if (number.equals(p.getPhone()))
69             {
70                 activated = true;
71                 p.setActivated(true);
72             }
73         });
74     });
75
76     if (!activated)
77     {
78         Map<String, String> mapError = new HashMap<String, String>();
79         mapError.put("Error: ", "Phone not found!");
80
81         return (Response.status(Response.Status.BAD_REQUEST).entity(mapError).build());
82     }
83
84     return (Response.status(Response.Status.OK).entity("Phone activated.").build());
85 }

```

Problems Search Console Servers Coverage

AndDigitalTest (20/02/2019 21:38:59)

Element	Coverage	Covered Instruction...	Missed Instructions	Total Instructions
ANDDigital	95,3 %	445	22	467

The screenshot shows the `Services.java` file in an IDE. The code implements a service for activating a phone number. It iterates through a list of customers and their phones, checking if a phone number matches the requested one. If it does, it sets the phone as activated. If not, it returns a 400 Bad Request response. If the phone is already activated, it returns a 200 OK response with a message.

```

65
66     listCustomer.forEach((Customer c) -> {
67         List<Phones> data = c.getListPhones();
68         data.forEach((Phones p) -> {
69             if (number.equals(p.getPhone()))
70             {
71                 activated = true;
72                 p.setActivated(true);
73             }
74         });
75     });
76
77     if (!activated)
78     {
79         Map<String, String> mapError = new HashMap<String, String>();
80         mapError.put("Error: ", "Phone not found!");
81
82         return (Response.status(Response.Status.BAD_REQUEST).entity(mapError).build());
83     }
84
85     return (Response.status(Response.Status.OK).entity("Phone activated.").build());
86 }
87
88 @Override
89 public List<Customer> createObject() {
90     /* Customer 1 */

```

The coverage report for `AndDigitalTest` (20/02/2019 21:47:33) is shown below:

Element	Coverage	Covered Instruction...	Missed Instructions	Total Instructions
ANDDigital	98,6 %	480	7	487
src	98,6 %	480	7	487
com.andDigital.entity	82,5 %	33	7	40
com.andDigital.services	100,0 %	384	0	384
com.andDigital.test	100,0 %	63	0	63

The screenshot shows the `AndDigitalTest.java` file in an IDE. The code contains unit tests for the `Services` class. It tests the `getAllPhones()`, `getPhoneById()`, and `activatedNumber()` methods using `assertEquals()` to verify the status of the phones.

```

12
13 @Before
14 public void setUp() throws Exception
15 {
16     classUnderTest = new Services();
17 }
18
19 @Test
20 public void testGetAllPhones() {
21     assertEquals(200, classUnderTest.getAllPhones().getStatus());
22 }
23
24 @Test
25 public void testGetPhoneById() {
26     assertEquals(200, classUnderTest.getPhoneById(2).getStatus());
27
28     assertEquals(400, classUnderTest.getPhoneById(22).getStatus());
29 }
30
31 @Test
32 public void testActivatedNumber() {
33     assertEquals(200, classUnderTest.activatedNumber("493789456").getStatus());
34     assertEquals(400, classUnderTest.activatedNumber("000000000").getStatus());
35 }
36 }
37

```

The coverage report for `AndDigitalTest` (20/02/2019 21:52:03) is shown below:

Element	Coverage	Covered Instruction...	Missed Instructions	Total Instructions
ANDDigital	100,0 %	477	0	477
src	100,0 %	477	0	477
com.andDigital.entity	100,0 %	33	0	33
com.andDigital.services	100,0 %	393	0	393
com.andDigital.test	100,0 %	51	0	51

6. Project decisions

- I have created two entities as if they represent the database;
- I populate these objects in the "`public List<Customer> createObject()`" method;
- In "`public Response getAllPhones()`" I just return the Object;
- In "`public Response getPhoneById(@PathParam("id") int id)`" I used Lambda function to filter customer data in object;
- In "`public Response activatedNumber(String json)`" I used again lambda functions and nested loops to set activated a phone number;