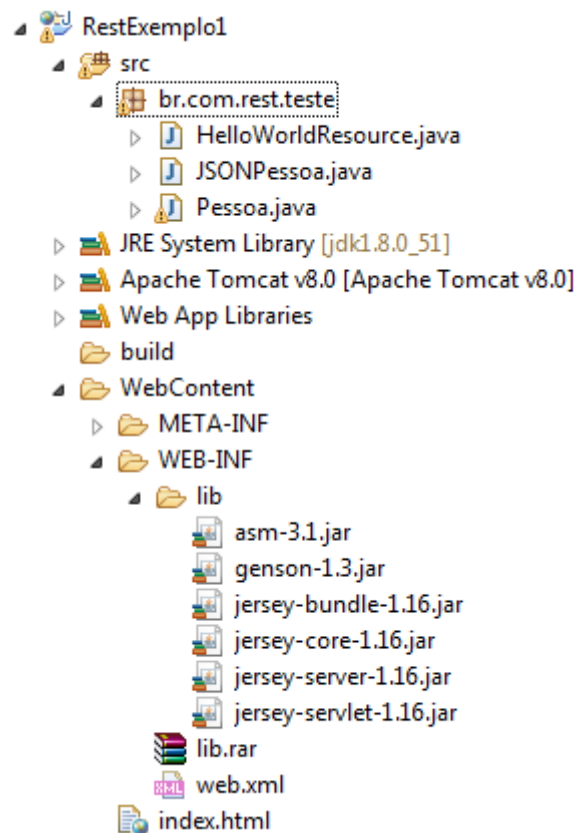


## REST - Utilização de Objetos



### Classes criadas:

- JSONPessoa.java
- Pessoa.java

### Libs inseridas:

- genson-1.3.jar
- Jersey-bundle-1.16

## IMPLEMENTAÇÃO:

- JSONPessoa.java

```
2
3 import javax.ws.rs.Consumes;
4 import javax.ws.rs.GET;
5 import javax.ws.rs.POST;
6 import javax.ws.rs.Path;
7 import javax.ws.rs.Produces;
8 import javax.ws.rs.core.Response;
9
10 @Path("/json")
11 public class JSONPessoa {
12
13     @GET
14     @Path("/get")
15     @Produces("application/json")
16     public Pessoa getProductInJSON() {
17
18         Pessoa p = new Pessoa();
19         p.setCodigo(1);
20         p.setNome("Jose Silva");
21
22         return p;
23     }
24
25     @POST
26     @Path("/post")
27     @Consumes("application/json")
28     public Response createPessoaInJSON(Pessoa pessoa) {
29
30         String result = "Product created : " + pessoa;
31         return Response.status(201).entity(result).build();
32     }
33 }
34
35
```

- **@Path:** Especifica um caminho relativo para um determinado recurso. Também identifica em qual URI um recurso será disponibilizado para receber requisições
- **@GET:** Especifica que um método (recurso) processará apenas requisições do tipo GET. Quando algum cliente faz uma requisição do tipo GET o método anotado para manipular a requisição será acionado.
- **@POST:** Tem a mesma função do @Get, porém, atende apenas requisições do tipo POST.
- **@Produces:** Especifica os tipos MIME que o método produzirá como resposta para o cliente.
- **@Consumes:** Especifica os tipos MIME que o método pode receber do cliente.

**\*MIME (Internet media type ou MIME type - Tipo de mídia da internet)** -> O tipo MIME identifica o tipo de representação contida no corpo de um pedido HTTP ou de resposta utilizando o cabeçalho Content-Type.

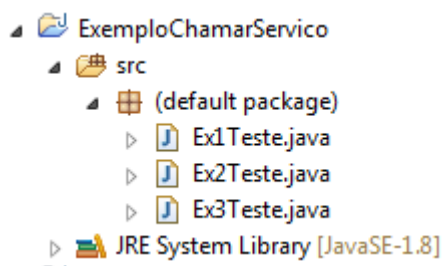
- Pessoa.java

```

27
26 }
25
24 }
23     return "Pessoa [codigo=" + codigo + ", nome=" + nome + "];"
22     public String toString() {
21     @Override
20 }
19 }
18     this.nome = nome;
17     public void setNome(String nome) {
16 }
15     return nome;
14     public String getNome() {
13 }
12     this.codigo = codigo;
11     public void setCodigo(int codigo) {
10 }
9     return codigo;
8     public int getCodigo() {
7 }
6     private String nome;
5     private int codigo;
4 }
3 public class Pessoa {
2
1

```

## PROJETO TESTE:



## IMPLEMENTAÇÃO:

- Ex1Teste.java:

```
1 import java.io.BufferedReader;
2 import java.io.IOException;
3 import java.io.InputStreamReader;
4 import java.net.HttpURLConnection;
5 import java.net.MalformedURLException;
6 import java.net.URL;
7
8 public class Ex1Teste {
9
10     public static void main(String[] args) {
11
12         imprimirTexto("Turma!");
13         somar(20, 3);
14
15     }
16
17     private static void imprimirTexto(String texto) {
18
19         try {
20             URL url = new URL(
21                 "http://localhost:8082/RestExemplo1/rest/helloworld/show="
22                 + texto);
23
24             HttpURLConnection conn = (HttpURLConnection) url.openConnection();
25             conn.setRequestMethod("GET");
26             conn.setRequestProperty("Accept", "application/json");
27
28             if (conn.getResponseCode() != 200) {
29                 throw new RuntimeException("ERRO: " + conn.getResponseCode());
30             }
31
32             BufferedReader br = new BufferedReader(new InputStreamReader(
33                 (conn.getInputStream())));
34
35             String output;
36             System.out.println("Saída: \n");
37             while ((output = br.readLine()) != null) {
38                 System.out.println(output);
39             }
40
41             conn.disconnect();
42
43         } catch (MalformedURLException e) {
44
45             e.printStackTrace();
46
47         }
```

```
46
47     } catch (IOException e) {
48
49         e.printStackTrace();
50
51     }
52 }
53
54 private static void somar(int a, int b) {
55     try {
56
57         URL url = new URL(
58             "http://localhost:8082/RestExemplo1/rest/helloworld/somar/"
59             + a + "/" + b + "");
60
61         HttpURLConnection conn = (HttpURLConnection) url.openConnection();
62         conn.setRequestMethod("GET");
63         conn.setRequestProperty("Accept", "application/json");
64
65         if (conn.getResponseCode() != 200) {
66             throw new RuntimeException("ERRO: " + conn.getResponseCode());
67         }
68
69         BufferedReader br = new BufferedReader(new InputStreamReader(
70             (conn.getInputStream())));
71
72         String output;
73         System.out.println("\nSaída: \n");
74         while ((output = br.readLine()) != null) {
75             System.out.println(output);
76         }
77
78         conn.disconnect();
79
80     } catch (MalformedURLException e) {
81
82         e.printStackTrace();
83
84     } catch (IOException e) {
85
86         e.printStackTrace();
87
88     }
89 }

```

---

```
}
```

- Ex2Teste.java:

```

2 import java.io.IOException;
3 import java.io.InputStreamReader;
4 import java.net.HttpURLConnection;
5 import java.net.MalformedURLException;
6 import java.net.URL;
7
8
9 public class Ex2Teste {
10
11     public static void main(String[] args) {
12
13         try {
14
15             URL url = new URL("http://localhost:8082/RestExemplo1/rest/json/get");
16             HttpURLConnection conn = (HttpURLConnection) url.openConnection();
17             conn.setRequestMethod("GET");
18             conn.setRequestProperty("Accept", "application/json");
19
20             if (conn.getResponseCode() != 200) {
21                 throw new RuntimeException("Failed : HTTP error code : "
22                     + conn.getResponseCode());
23             }
24
25             BufferedReader br = new BufferedReader(new InputStreamReader(
26                 (conn.getInputStream())));
27
28             String output;
29             System.out.println("Output from Server .... \n");
30             while ((output = br.readLine()) != null) {
31                 System.out.println(output);
32             }
33
34             conn.disconnect();
35
36         } catch (MalformedURLException e) {
37
38             e.printStackTrace();
39
40         } catch (IOException e) {
41
42             e.printStackTrace();
43
44         }
45
46     }
47 }

```

- Ex3Teste.java:

```

1 import java.io.BufferedReader;
2 import java.io.IOException;
3 import java.io.InputStreamReader;
4 import java.io.OutputStream;
5 import java.net.HttpURLConnection;
6 import java.net.MalformedURLException;
7 import java.net.URL;

9 public class Ex3Teste {
10
11     public static void main(String[] args) {
12
13         try {
14
15             URL url = new URL(
16                 "http://localhost:8082/RestExemplo1/rest/json/post");
17             HttpURLConnection conn = (HttpURLConnection) url.openConnection();
18             conn.setDoOutput(true);
19             conn.setRequestMethod("POST");
20             conn.setRequestProperty("Content-Type", "application/json");
21
22             String input = "{\"codigo\":2,\"nome\":\"andre\"}";
23
24             OutputStream os = conn.getOutputStream();
25             os.write(input.getBytes());
26             os.flush();
27
28             if (conn.getResponseCode() != HttpURLConnection.HTTP_CREATED) {
29                 throw new RuntimeException("Failed : HTTP error code : "
30                     + conn.getResponseCode());
31             }
32
33             BufferedReader br = new BufferedReader(new InputStreamReader(
34                 (conn.getInputStream())));
35
36             String output;
37             System.out.println("Output from Server .... \n");
38             while ((output = br.readLine()) != null) {
39
40                 System.out.println(output);
41             }
42
43             conn.disconnect();
44
45         } catch (MalformedURLException e) {
46             e.printStackTrace();
47         } catch (IOException e) {
48             e.printStackTrace();
49         }
50     }
51 }
52
53

```