Cliente REST em Java

1. package cliente;
2. import java.net.URI;
3. import javax.ws.rs.core.\*;
4. import com.sun.jersey.api.client.\*;
5. import com.sun.jersey.api.client.config.\*;
6. import com.sun.jersey.api.representation.\*;
7. import br.edu.utfpr.jersey.model.Livro;
8. public class Tester {
9. public static void main(String[] args) {
10. ClientConfig config = new DefaultClientConfig();
11. Client client = Client.create(config);
13. WebResource service = client.resource(getBaseURI());
15. // Criando um Livro
16. System.out.println(">> Criando um Livro");
17. Livro todo = new Livro("3", "RESTful Web Services", "Autor 3", "3333");
18. ClientResponse response = service.path("rest").path("biblioteca")
19. .path(todo.getId()).accept(MediaType.APPLICATION\_XML)
20. .put(ClientResponse.class, todo);
22. // O código de retorno deve ser 201 == recurso criado
23. System.out.println("Código de retorno: " + response.getStatus());
25. // Obtendo a biblioteca
26. System.out.println("\n>> Obtendo a biblioteca em XML");
27. System.out.println(service.path("rest").path("biblioteca")
28. .accept(MediaType.TEXT\_XML).get(String.class));
30. // Obter JSON para aplicação
31. System.out.println("\n>> Obtendo a biblioteca em JSON para aplicação");
32. System.out.println(service.path("rest").path("biblioteca")
33. .accept(MediaType.APPLICATION\_JSON).get(String.class));
35. // Obtendo o XML para a aplicação
36. System.out.println("\n>> Obtendo a biblioteca em XML para a aplicação");
37. System.out.println(service.path("rest").path("biblioteca")
38. .accept(MediaType.APPLICATION\_XML).get(String.class));
39. // Obtendo o Livro com id 1
40. System.out.println("\n>> Obtendo o Livro com ID 1");
41. System.out.println(service.path("rest").path("biblioteca/1")
42. .accept(MediaType.APPLICATION\_XML).get(String.class));
44. // Deletando o Livro com id 1
45. System.out.println("\n>> Deletando o Livro com ID 1");
46. service.path("rest").path("biblioteca/1").delete();
48. // Obtendo todos os livros, o id 1 deve ser deletado
49. System.out.println("\n>> Obtendo a biblioteca em XML com o livro 1 deletado");
50. System.out.println(service.path("rest").path("biblioteca")
51. .accept(MediaType.APPLICATION\_XML).get(String.class));
52. // Criando um Livro
53. System.out.println("\n>> Criando um Livro com Form");
54. Form form = new Form();
55. form.add("id", "4");
56. form.add("titulo", "Java: como programar");
57. form.add("autor", "Deitel");
58. form.add("isbn", "1234");
59. response = service.path("rest").path("biblioteca")
60. .type(MediaType.APPLICATION\_FORM\_URLENCODED)
61. .post(ClientResponse.class, form);
62. System.out.println("Form response " + response.getEntity(String.class));
64. // Obtendo todos os Livros, o id 4 deve ser criado
65. System.out.println("\n>> Obtendo a biblioteca em XML com o livro 4 criado");
66. System.out.println(service.path("rest").path("biblioteca")
67. .accept(MediaType.APPLICATION\_XML).get(String.class));
68. }
69. private static URI getBaseURI() {
70. return UriBuilder.fromUri("http://localhost:8080/RestLivro").build();
71. }
72. }