|  |
| --- |
| package activities; |
|  |  |
|  | import io.restassured.http.ContentType; |
|  | import io.restassured.response.Response; |
|  | import org.testng.annotations.Test; |
|  |  |
|  | import java.io.File; |
|  | import java.io.FileInputStream; |
|  | import java.io.FileWriter; |
|  | import java.io.IOException; |
|  |  |
|  | import static io.restassured.RestAssured.given; |
|  | import static org.hamcrest.Matchers.equalTo; |
|  |  |
|  | public class Activity2 { |
|  |  |
|  | final static String baseURI = "https://petstore.swagger.io/v2/user"; |
|  |  |
|  | @Test(priority = 1) |
|  | public void postUser() throws IOException { |
|  | FileInputStream inputJSON = new FileInputStream("src/test/java/activities/userInfo.json"); |
|  |  |
|  | String reqBody = new String(inputJSON.readAllBytes()); |
|  |  |
|  | Response response = given().contentType(ContentType.JSON) |
|  | .body(reqBody) |
|  | .when().post(baseURI); |
|  |  |
|  | inputJSON.close(); |
|  |  |
|  | response.then().statusCode(200); |
|  | response.then().body("message", equalTo("1234")); |
|  | } |
|  |  |
|  | @Test(priority = 2) |
|  | public void getUserInfo() { |
|  | File outputJSON = new File("src/test/java/activities.userGETResponse.json"); |
|  | Response response = given().contentType(ContentType.JSON) |
|  | .pathParam("username", "SampleTest") |
|  | .when().get(baseURI + "/{username}"); |
|  |  |
|  | String resBody = response.getBody().asPrettyString(); |
|  |  |
|  | try { |
|  |  |
|  | outputJSON.createNewFile(); |
|  | FileWriter writer = new FileWriter(outputJSON.getPath()); |
|  | writer.write(resBody); |
|  | writer.close(); |
|  | } catch (IOException excp) { |
|  | excp.printStackTrace(); |
|  | } |
|  |  |
|  | response.then().body("id", equalTo(1234)); |
|  | response.then().body("username", equalTo("SampleTest")); |
|  | response.then().body("firstName", equalTo("Sample")); |
|  | response.then().body("lastName", equalTo("Test")); |
|  | response.then().body("email", equalTo("sampletest@mail.com")); |
|  | response.then().body("password", equalTo("password1234")); |
|  | response.then().body("phone", equalTo("979797979")); |
|  | } |
|  |  |
|  | @Test(priority=3) |
|  | public void deleteUser() throws IOException { |
|  | Response response = |
|  | given().contentType(ContentType.JSON) |
|  | .pathParam("username", "SampleTest") |
|  | .when().delete(baseURI + "/{username}"); |
|  |  |
|  | response.then().statusCode(200); |
|  | response.then().body("message", equalTo("SampleTest")); |
|  | } |
|  | } |