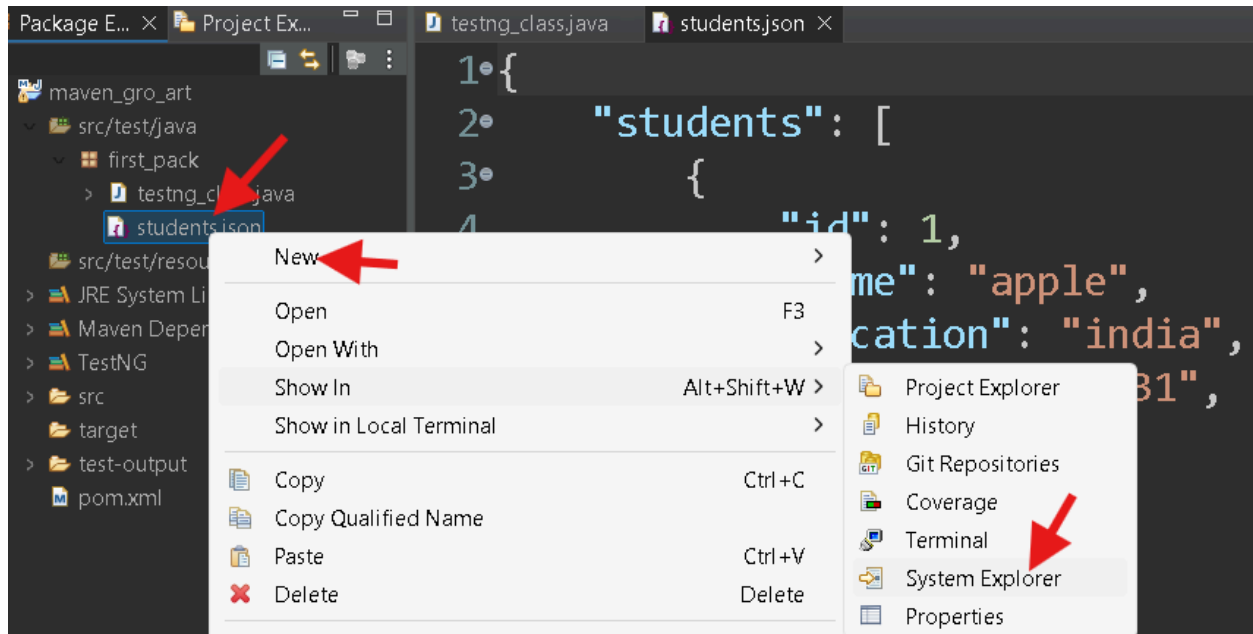


## How many ways we create request body:

1. Hashmap
2. Using org.json
3. Using POJO(Plain Old Java Object)
4. Using external json file

## Create student.json file:

To open it:



It has:

```
{
  "students": [
    {
      "id": 1,
      "name": "apple",
      "location": "india",
      "phone": "1231231",
      "courses": [
        "python",
        "java",
        "selenium"
      ]
    },
    {
      "id": 2,
      "name": "banana",
      "location": "india",
```

```

        "phone": "1237567",
        "courses": [
            "java",
            "selenium"
        ]
    },
    {
        "id": 3,
        "name": "gari",
        "location": "bhutan",
        "phone": "126788787",
        "courses": [
            "javascript",
            "playwright"
        ]
    }
]
}

```

## To run the json server:

In command prompt:

json-server students.json

```

C:\Users\kiran\OneDrive\Desktop\RestAssured\RestAssured_Eclipse\maven_gro_art\src\test\java\first_pack>json-server students.json
JSON Server started on PORT :3000
Press CTRL-C to stop
Watching students.json...

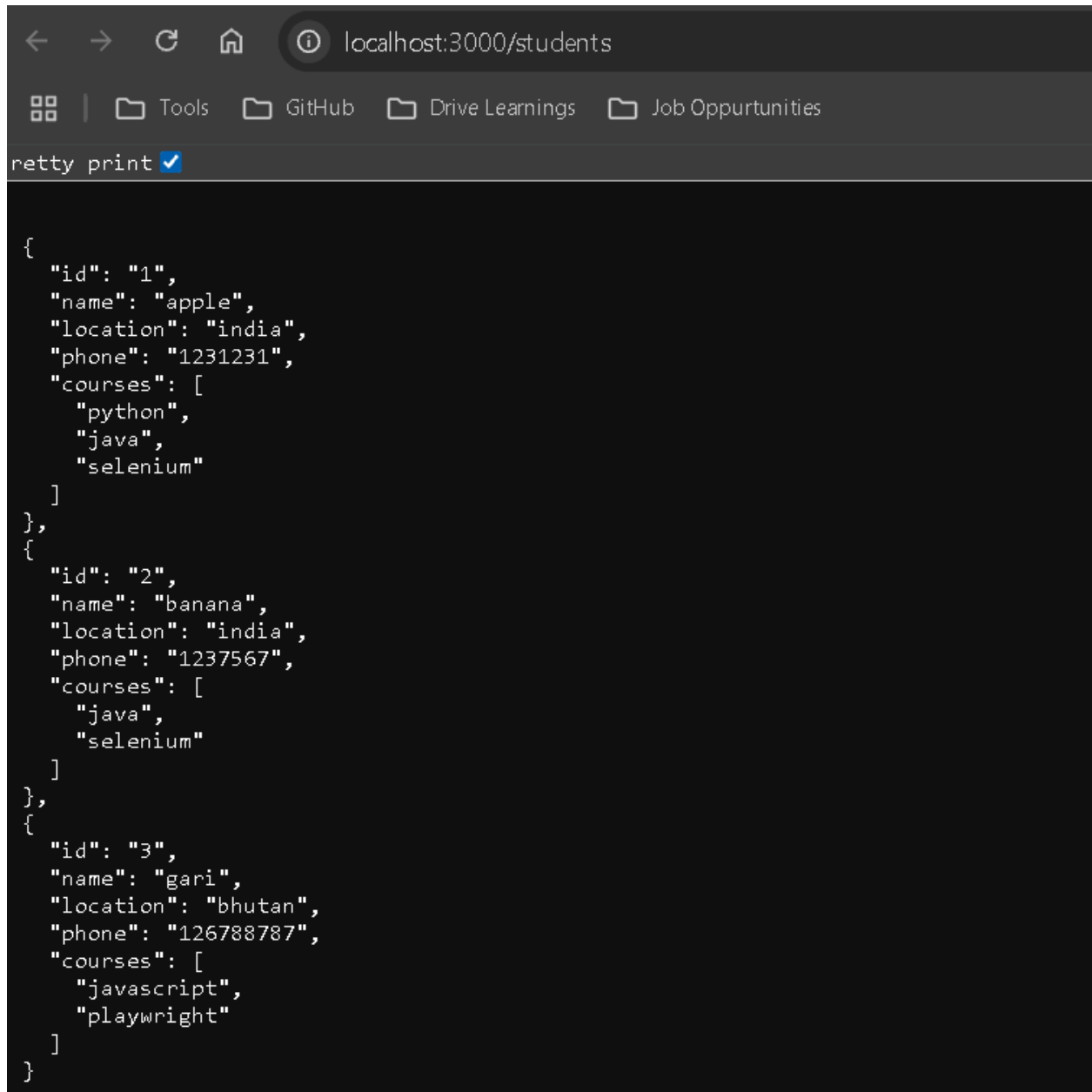
(.^ ^.^)

Index:
http://localhost:3000/

Static files:
Serving ./public directory if it exists

Endpoints:
http://localhost:3000/students

```



```
{
  "id": "1",
  "name": "apple",
  "location": "india",
  "phone": "1231231",
  "courses": [
    "python",
    "java",
    "selenium"
  ]
},
{
  "id": "2",
  "name": "banana",
  "location": "india",
  "phone": "1237567",
  "courses": [
    "java",
    "selenium"
  ]
},
{
  "id": "3",
  "name": "gari",
  "location": "bhutan",
  "phone": "126788787",
  "courses": [
    "javascript",
    "playwright"
  ]
}
```

## Code:

**Thats add user to the json server and after 5 seconds delay deletes the user**

```
package _02_Creating_Post_Request_Payloads_in_Multiple_Ways;
```

```
import static io.restassured.RestAssured.given;
```

```
import static org.hamcrest.Matchers.equalTo;
```

```
import java.util.HashMap;
```

```
import org.testng.annotations.Test;
```

```
public class Part01 {
```

```
    String id;
```

```
//    Post request body using Hashmap
```

```

@Test(priority= 1)
void testPostUsingHashMap() {
    HashMap<String, Object> data = new HashMap<>();
    data.put("name", "scott");
    data.put("location", "france");
    data.put("phone", "145234");
    String lang[] = {"javascript", "playwright"};
    data.put("courses", lang);

    id = given()
        .contentType("application/json")
        .body(data)
    .when()
        .post("http://localhost:3000/students")
        .jsonPath().getString("id");
//    .then()
//        .statusCode(201)
//        .body("name", equalTo("scott"))
//        .body("location", equalTo("france"))
//        .body("phone", equalTo("145234"))
//        .body("courses[0]", equalTo("javascript"))
//        .body("courses[1]", equalTo("playwright"))
//        .header("Content-Type", "application/json")
//        .log().all()
//        .jsonPath().getInt("id");
}
// Get request
@Test(priority = 2)
void testGet(){
    given()

    .when()
        .get("http://localhost:3000/students/" + id)

    .then()
        .statusCode(200)
        .body("name", equalTo("scott"))
        .body("location", equalTo("france"))
        .body("phone", equalTo("145234"))
        .body("courses[0]", equalTo("javascript"))
        .body("courses[1]", equalTo("playwright"))
        .header("Content-Type", "application/json")
        .log().all();
}

```

```
//      For Deleting student record
@Test(priority = 3)
void testDelete() throws InterruptedException {
    Thread.sleep(5000);
    given()

        .when()
            .delete("http://localhost:3000/students/" + id)

        .then()
            .statusCode(200);
}
}
```

## New code:

```
package _02_Creating_Post_Request_Payloads_in_Multiple_Ways;
import static io.restassured.RestAssured.given;
import static org.hamcrest.Matchers.equalTo;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.util.HashMap;

import org.json.JSONObject;
import org.json.JSONTokener;
import org.testng.annotations.Test;

public class Part01 {
    String id;

    @Test(priority= 1)
    void testPostUsingDifferentWays() throws FileNotFoundException {
//        Using HashMap
//        HashMap<String, Object> data = new HashMap<>();
//        data.put("name", "scott");
//        data.put("location", "france");
//        data.put("phone", "145234");
//        String lang[] = {"javascript", "playwright"};
//        data.put("courses", lang);

//        Using JsonObject
//        JSONObject data = new JSONObject();
```

```

//      data.put("name", "scott");
//      data.put("location", "france");
//      data.put("phone", "145234");
//      String lang[] = {"javascript", "playwright"};
//      data.put("courses", lang);

//      Using POJO(Plain Old Java Object)
//      Its like making getter and setter form other java file and getting its value after
setting it

//      Using external file
File f = new
File("C:\\Users\\kiran\\OneDrive\\Desktop\\RestAssured\\RestAssured_Eclipse\\maven_gro_art\\
src\\test\\java\\_02_Creating_Post_Request_Payloads_in_Multiple_Ways\\new_user.json");
FileReader fr = new FileReader(f);
JSONTokener jk = new JSONTokener(fr);
JSONObject data = new JSONObject(jk);
System.out.println(data);

id = given()
    .contentType("application/json")
    .body(data.toMap())
    .when()
        .post("http://localhost:3000/students")
        .jsonPath().getString("id");
//      .then()
//          .statusCode(201)
//          .body("name", equalTo("scott"))
//          .body("location", equalTo("france"))
//          .body("phone", equalTo("145234"))
//          .body("courses[0]", equalTo("javascript"))
//          .body("courses[1]", equalTo("playwright"))
//          .header("Content-Type", "application/json")
//          .log().all()
//          .jsonPath().getInt("id");
    }
//      Get request
@Test(priority = 2)
void testGet(){
    given()

        .when()
            .get("http://localhost:3000/students/" + id)

```

```

        .then()
            .statusCode(200)
            .body("name", equalTo("scott"))
            .body("location", equalTo("france"))
            .body("phone", equalTo("145234"))
            .body("courses[0]", equalTo("javascript"))
            .body("courses[1]", equalTo("playwright"))
            .header("Content-Type", "application/json")
            .log().all();
    }

//    For Deleting student record
@Test(priority = 3)
void testDelete() throws InterruptedException {
    Thread.sleep(5000);
    given()

        .when()
            .delete("http://localhost:3000/students/" + id)

        .then()
            .statusCode(200);
    }
}

```

## Output:

```

[RemoteTestNG] detected TestNG version 7.11.0
SLF4J(W): No SLF4J providers were found.
SLF4J(W): Defaulting to no-operation (NOP) logger implementation
SLF4J(W): See https://www.slf4j.org/codes.html#noProviders for further details.
{"courses":["javascript","playwright"],"phone":"145234","name":"scott","location":"france"}
HTTP/1.1 200 OK
X-Powered-By: tinyhttp
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, HEAD, PUT, PATCH, POST, DELETE
Access-Control-Allow-Headers: content-type
Content-Type: application/json
Date: Wed, 16 Jul 2025 13:54:52 GMT
Connection: keep-alive
Keep-Alive: timeout=5
Content-Length: 137

```

```

{
  "id": "7c37",

```

```
"courses": [  
  "javascript",  
  "playwright"  
],  
"phone": "145234",  
"name": "scott",  
"location": "france"  
}
```

PASSED:

\_02\_Creating\_Post\_Request\_Payloads\_in\_Multiple\_Ways.Part01.testPostUsingDifferentWays

PASSED: \_02\_Creating\_Post\_Request\_Payloads\_in\_Multiple\_Ways.Part01.testDelete

PASSED: \_02\_Creating\_Post\_Request\_Payloads\_in\_Multiple\_Ways.Part01.testGet

=====

Default test

Tests run: 3, Failures: 0, Skips: 0

=====

=====

Default suite

Total tests run: 3, Passes: 3, Failures: 0, Skips: 0

=====