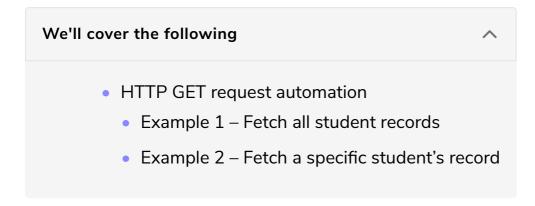
GET Request

In this lesson, we will learn to automate an HTTP GET request using Rest Assured.



HTTP GET request automation

Rest Assured uses given/when/then syntax from behavior-driven development as it makes it easy to read and write tests.

Example 1 – Fetch all student records

- HTTP Method: GET
- Target URL: http://ezifyautomationlabs.com:6565/
- Resource path: /educative-rest/students
- Take a look at the code below:

```
Response response = RestAssured.given().get(url).andReturn();

LOG.info("Step - 2 : Print the JSON response body");
response.getBody().prettyPrint();

LOG.info("Step - 3 : Assert StatusCode = 200");
assertEquals(response.getStatusCode(), 200, "http status code");

LOG.info("Step - 4 : Verify that the response contains id = 101");
LOG.info("list of Student's Id " +response.getBody().jsonPath().getList("id"));
assertTrue(response.getBody().jsonPath().getList("id").contains(101));
}
```







[]

For logging, we are creating an instance of the *Logger* interface for *GETRequestTest* class which internally uses *logback* for the concrete implementation.

Let's understand the example code above.

The code above uses the TestNG and RestAssured libraries for automating HTTP GET request.

• Step 1 – makes a GET request and returns a Response object

```
Response response = RestAssured.given().get(url).andReturn();
```

• **Step 2** – prints the response body in the JSON format

```
response.getBody().prettyPrint();
```

• Step 3 – asserts the response status code as 200

```
assertTrue(response.getStatusCode()==200);
```

• **Step 4** – gets the list of all Student Ids from the response body and assert that one of the *Id* is *101*

```
LOG.info("list of Student's Id " +response.getBody().jsonPath().getList("i
d"));
assertTrue(response.getBody().jsonPath().getList("id").contains(101));
```

Example 2 – Fetch a specific student's record

- HTTP Method: GET
- Target URL: http://ezifyautomationlabs.com:6565
- Resource path for making GET request with path param: /educativerest/students/{id}
- Resource path for making GET request with query params: /educativerest/students?first_name={x}&last_name={y}
- Take a look at the code below:

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import static org.hamcrest.Matchers.equalTo;
import org.testng.annotations.Test;
import static org.testng.Assert.assertEquals;
import static org.testng.Assert.assertTrue;
import io.restassured.response.Response;
import io.restassured.RestAssured;
public class GETRequestTest {
        private static Logger LOG = LoggerFactory.getLogger(GETRequestTest.class);
        @Test
        public void testAStudentRecord() {
         String url = "http://ezifyautomationlabs.com:6565/educative-rest/students";
                 LOG.info("Step - 1 Get a Student's record with a specific Id.");
                 String url1 = url + "/" + "101";
                 Response response1 = RestAssured.given()
                     .get(url1)
                         .andReturn();
                 LOG.info(response1.getStatusLine());
                 assertTrue(response1.getStatusCode()==200);
                 Long id = response1.getBody().jsonPath().getLong("id");
                 assertTrue(id==101);
                 LOG.info("Step - 2 Get a Students record using matching gender");
                 Response response = RestAssured
                     .given()
                     .queryParam("gender", "male")
                     .get(url)
                         .andReturn();
                 response.getBody().prettyPrint();
        }
```

Let's understand the example code above.

The code above uses TestNG and RestAssured libraries for automating the HTTP GET request.

• **Step 1** – it makes a GET request using {*id*} in the *Path* param to get the Response from the server. It then verifies that the response's body contains the requested {*id*}

• **Step 2** – it makes a **GET** request using *Query Params* like Student's *gender* to get the matching Student's data from the server.

```
Response response = RestAssured
    .given()
    .queryParam("gender", "male")
    .get(url).andReturn();
response.getBody().prettyPrint();
```

In the next lesson, we will learn about automating a POST request.