File Downloads

In this lesson, we will look at how to download the response to a file using the Rest Assured library.

We'll cover the following

What is file download?

There are instances where we need to save the response body message as a file. In this example, we will make a `GET` *request* that returns a `JSON` string which will be saved as a file.

Example: Download JSON file

- HTTP method: **GET**
- Target URL: `http://ezifyautomationlabs.com:6565`
- Resource path: `/educative-rest/students`

Take a look at the code below:

```
import static org.testng.Assert.assertEquals;
import static org.testng.Assert.assertTrue;

import java.io.File;
import java.io.IOException;
import java.nio.file.Files;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.testng.annotations.Test;

import io.restassured.RestAssured;
import io.restassured.response.Response;

public class APIDemo {

        private static final Logger LOG = LoggerFactory.getLogger(APIDemo.class);

        @Test
        public void testDownload() throws IOException {
```

```
                String url = "http://ezifyautomationlabs.com:6565/educative-rest/students";


        // making API call
                Response response = RestAssured.given()
                                .log().all(true)
                                .get(url)
                                .andReturn();

        // validating http status code
                assertEquals(response.getStatusCode(), 200, "http status code");

        // reading the response boody as byte[]
                byte[] bytes = response.getBody().asByteArray();
        // validating that the response content length > 0
                assertTrue(bytes.length > 0, "response content length is 0");

        // writing the byte[] to file
                File file = new File("students.json");
                Files.write(file.toPath(), bytes);

        // validating the existence and size of file
                assertTrue(file.exists(), "file " + file + " does not exist");
                assertTrue(file.length() > 0, "file size is 0");
        assertEquals(bytes.length, file.length(), "file size and response content");

                String content = new String(Files.readAllBytes(file.toPath()));
                LOG.info("printing content of the file => {}", content);
        }

}
```




The output of this code returns the following response:


```
[{
    "id": 100,
    "first_name": "John",
    "last_name": "Doe",
    "gender": "male"
},
{
    "id": 101,
    "first_name": "Kelly",
    "last_name": "Flower",
    "gender": "female"
},
{
    "id": 102,
    "first_name": "Json",
    "last_name": "Ray",
    "gender": "male"
```

```
}]
```

Note: The content in the code above snippet might be different from the results of the actual code.

Let's understand the example code above.

The code uses the TestNG and Rest Assured libraries to automate the HTTP GET to fetch the `Student`.

- **Step 1** – we make the `GET` API call to fetch the `Student` using Rest Assured constructs like the following:

```
// making API call
Response response = RestAssured.given()
    .log().all(true)
    .get(url)
    .andReturn();
```

- **Step 2** – we verify the HTTP status code to ensure the creation of student is successful.

```
assertEquals(response.getStatusCode(), 200, "http status code");
```

- **Step 3** – we read the response body as `byte[]` and validate the content length.

```
// reading the response body as byte[]
byte[] bytes = response.getBody().asByteArray();
// validating that the response content length > 0
assertTrue(bytes.length > 0, "response content length is 0");
```

- **Step 4** – we write the `byte[]` to a file with an appropriate file extension. In our case, it is a JSON string.

```
// writing the byte[] to file
File file = new File("students.json");
Files.write(file.toPath(), bytes);
```

- **Step 5** – we validate the existence of the created file and its size.

```
// validating the existence and size of file
assertTrue(file.exists(), "file " + file + " does not exist");
```

```
assertTrue(file.length() > 0, "file size is 0");

assertEquals(bytes.length, file.length(), "file size and response content"
);
```

In the next lesson, we will learn how to call APIs that need authorization.