Basic Authentication

In this lesson, we will learn about handling basic authentication for a given API.

We'll cover the following

- What is basic authentication?
- Example

What is basic authentication?

The basic authentication scheme requires the user to send the access credentials encoded in base64 or send an authorization token. `REST Assured` provides an easy way to configure and handle the credentials/token that the request requires.

The authentication is applicable to any `HTTP` Request like `GET`, `PUT`, `POST`, `DELETE`, etc.

Example

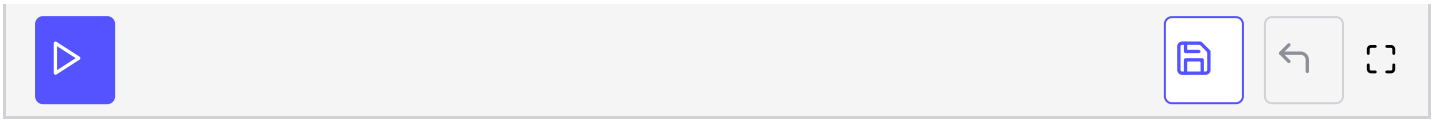
Let's understand basic authentication better with an example code snippet:

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

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

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 Logger LOG = LoggerFactory.getLogger(APIDemo.class);

    /**
     * Basic authentication using Valid username and password
```

```
 */
    @Test
    public void test_authentication_ValidCredentials() {
            String url = "http://ezifyautomationlabs.com:6565/educative-rest/auth/students";

    String valid_userName = "testuser";
    String valid_password = "testpass";

    Response response =  RestAssured
                                          .given()
                                          .auth().basic(valid_userName, valid_password)
                                    .when()
                                    .get(url)
                                    .thenReturn();

    LOG.info("It will return a valid response");
    response.getBody().prettyPrint();
    assertTrue(response.getStatusCode() == 200);
    }

/**
 * Basic authentication using In-valid username and password
 */
    @Test
    public void test_authentication_InvalidCredentials() {
            String url = "http://ezifyautomationlabs.com:6565/educative-rest/auth/students";

    String invalid_userName = "testuser1";
    String valid_password = "testpass";

    Response response =  RestAssured
                                          .given()
                                          .auth().basic(invalid_userName, valid_password)
                                    .when()
                                    .get(url)
                                    .thenReturn();

    LOG.info("It will return authorization error 401");
    response.getBody().prettyPrint();
    assertTrue(response.getStatusCode() == 401);
    }

/**
 * Basic authentication using Auth token
 */
    @Test
    public void test_authentication_AuthToken() {

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

    String authCode = "Basic dGVzdHVzZXI6dGVzdHBhc3M=";

    Response response =  RestAssured
                                           .given()
                                  .header("authorization", authCode)
                                      .when()
                                      .get(url)
                                      .thenReturn();

    LOG.info("It will return a valid response");
    response.getBody().prettyPrint();
    assertTrue(response.getStatusCode() == 200);
```

```
    }

}
```



Understanding the example code above:

- Case 1 – *Authentication using valid credentials:*

  ```
  Response response =  RestAssured
                       .given()
                       .auth().basic(valid_userName, valid_password)
                       .when()
                       .get(url)
                       .thenReturn();
  ```

  In this case, a valid username and password is sent using the `auth().basic(valid_userName, valid_password)` method and will be encoded to base64 in the request internally. This returns a valid response with status code `200`.

- Case 2 – *Authentication using invalid credentials:*

  In this case, the code remains the same. We just pass an *invalid* username and the response returned contains authorization error, status code as `401`.

- Case 3 – *Authentication using auth token:*

  ```
  Response response =  RestAssured
                      .given()
                      .header("authorization", authCode)
                      .when()
                      .get(url)
                      .thenReturn();
  ```

  In this case, an *authorization token* is passed using headers `header("authorization", authCode)`. The valid auth token will return a response with status code `200`.

To know more, please follow this link.

In the next lesson, we'll learn about handling async requests.

In the next lesson, we'll learn about handling async requests.