SYNTAX TECHNOLOGIES

API

Class 3

Agenda

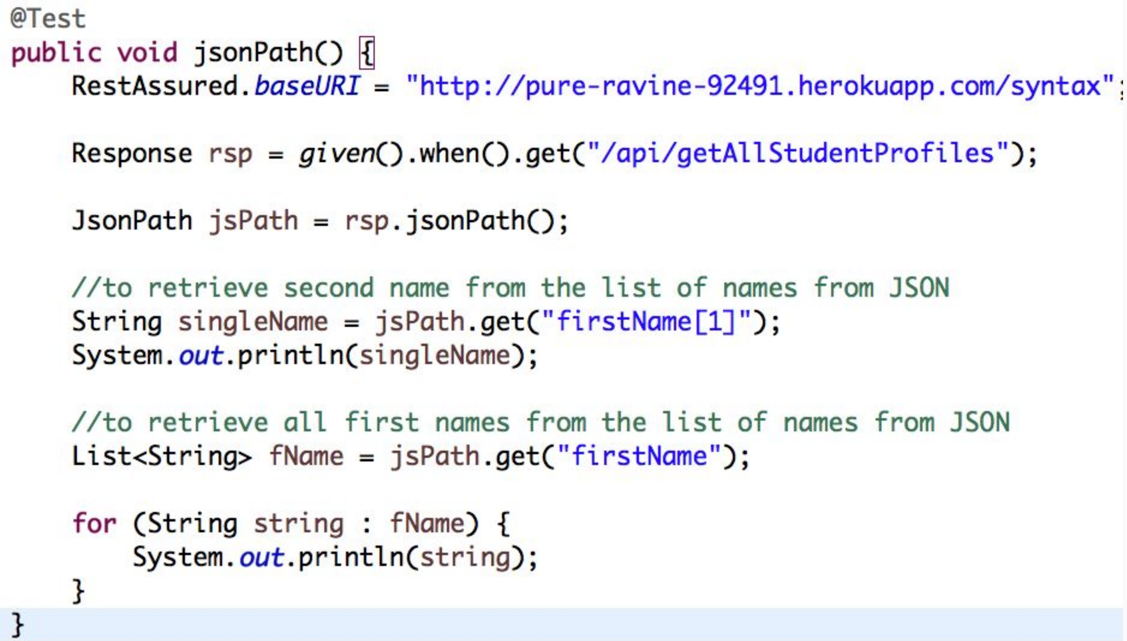
JSON Path

Headers

Serialization and DeSerialization

JSON Path

- JsonPath is an alternative to using XPath for easily getting values from a Object document.
- JsonPath is Another way to validate response body



```
@Test
public void jsonPath() {
    RestAssured.baseURI = "http://pure-ravine-92491.herokuapp.com/syntax";

    Response rsp = given().when().get("/api/getAllStudentProfiles");

    JsonPath jsPath = rsp.jsonPath();

    //to retrieve second name from the list of names from JSON
    String singleName = jsPath.get("firstName[1]");
    System.out.println(singleName);

    //to retrieve all first names from the list of names from JSON
    List<String> fName = jsPath.get("firstName");

    for (String string : fName) {
        System.out.println(string);
    }
}
```

Headers

HTTP Headers are an important part of the API request and response as they represent set of metadata associated with the API request and response.

Most of these headers are for management of **connections between client, server and proxies and do not require explicit validation** through testing.

Headers are mostly classified as **request headers and response headers**, know the major request and response headers.

Headers

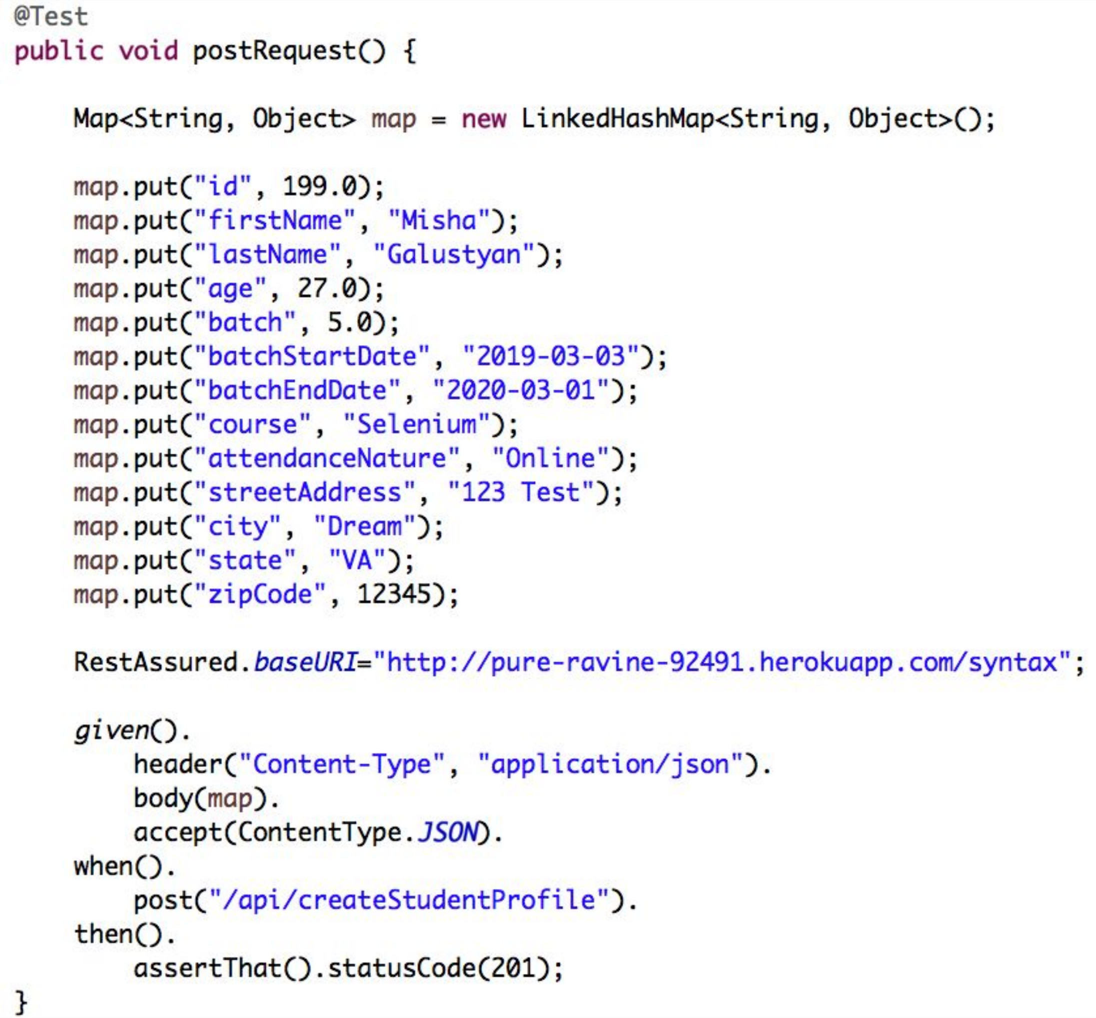
Headers are a name, followed by :, followed by the value of the header.

Example:

- **Content-Type header** tells the server that the content of this message is JSON.
- **Accept header** tells the server that the client (application sending the message) will only accept response payloads represented in JSON.

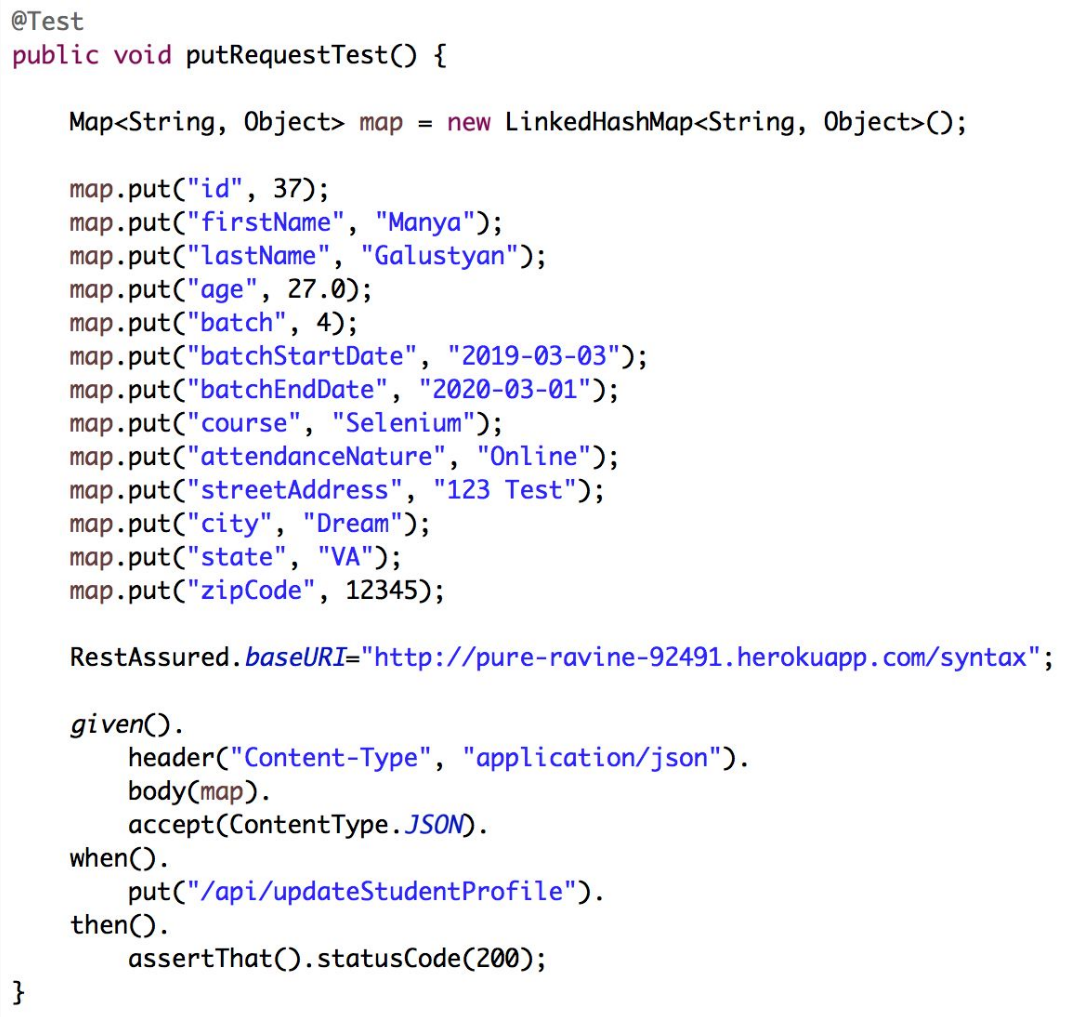
For some headers we may need to set values or set assertions to ensure that they convey the right information and everything works fine in the API.

POST



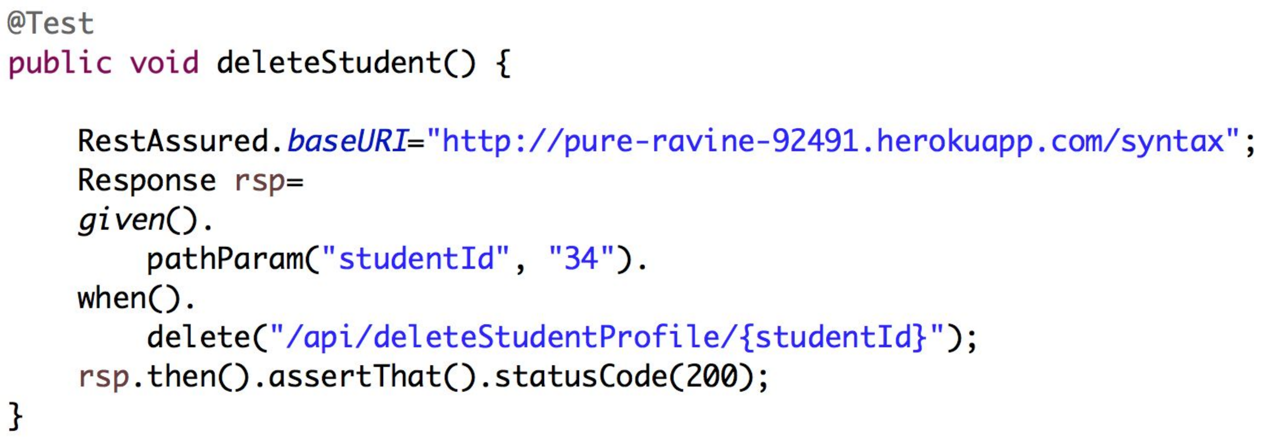
```
@Test
public void postRequest() {

    Map<String, Object> map = new LinkedHashMap<String, Object>();

    map.put("id", 199.0);
    map.put("firstName", "Misha");
    map.put("lastName", "Galustyan");
    map.put("age", 27.0);
    map.put("batch", 5.0);
    map.put("batchStartDate", "2019-03-03");
    map.put("batchEndDate", "2020-03-01");
    map.put("course", "Selenium");
    map.put("attendanceNature", "Online");
    map.put("streetAddress", "123 Test");
    map.put("city", "Dream");
    map.put("state", "VA");
    map.put("zipCode", 12345);

    RestAssured.baseURI="http://pure-ravine-92491.herokuapp.com/syntax";

    given().
        header("Content-Type", "application/json").
        body(map).
        accept(ContentType.JSON).
    when().
        post("/api/createStudentProfile").
    then().
        assertThat().statusCode(201);
}
```

Put



```java
@Test
public void putRequestTest() {

    Map<String, Object> map = new LinkedHashMap<String, Object>();

    map.put("id", 37);
    map.put("firstName", "Manya");
    map.put("lastName", "Galustyan");
    map.put("age", 27.0);
    map.put("batch", 4);
    map.put("batchStartDate", "2019-03-03");
    map.put("batchEndDate", "2020-03-01");
    map.put("course", "Selenium");
    map.put("attendanceNature", "Online");
    map.put("streetAddress", "123 Test");
    map.put("city", "Dream");
    map.put("state", "VA");
    map.put("zipCode", 12345);

    RestAssured.baseURI="http://pure-ravine-92491.herokuapp.com/syntax";

    given().
        header("Content-Type", "application/json").
        body(map).
        accept(ContentType.JSON).
    when().
        put("/api/updateStudentProfile").
    then().
        assertThat().statusCode(200);
}
```

Delete



```
@Test
public void deleteStudent() {

    RestAssured.baseURI="http://pure-ravine-92491.herokuapp.com/syntax";
    Response rsp=
    given().
        pathParam("studentId", "34").
    when().
        delete("/api/deleteStudentProfile/{studentId}");
    rsp.then().assertThat().statusCode(200);
}
```

Payload

A Payload is the body of the HTTP request or response.

Typically when working with an HTTP API we will send and receive JSON or XML payloads.

When we send request we have request payload

When we get response we have response payload