<epam>

REST API testing

Ákos Osvald
Budapest 2018

About REST

- Representational state transfer (Roy Fielding, 2000)
- Architectural approach for developing web services
- Builds upon existing systems and features of the internet's HTTP. Not a new „standard”!
- Description of how a well-designed Web application should behave
- Main restrictions of REST:
    - Use of a uniform interface
    - Client-server-based
    - Stateless operations
    - RESTful resource caching
- Example host with endpoint and query parameter:
  GET http://www.an-example-application.com/api/v2/user/names?name_id=123456

REST API testing

WHAT IS IT?

- Part of integration testing
- A functional testing operation
- Testing of direct application business logic interface
- Independent testing from presentation layer

HOW TO DO?

- External tools (Postman, Rest-Assured, SoapUI, Katalon, etc.)
- Calling user endpoints with proper method and payload
- Verify the received result, its status code and content

Why do we need it?

BENEFITS

- Provides immediate feedback of backend (changes)
- Quick, much more faster than GUI testing
- Independent from representation layer
- Less brittle and easier to maintain than GUI tests
- Involves reusable and standardized methods
- Easy to automatize
- Widely usable besides functional testing (e.g.. Performance, security, db, etc.)
- Testing negative scenarios as well as user journeys
- Can redound test driven development in case of early contract

Few words about Rest-Assured and JSON

Rest-Assured

- Testing REST services in Java
- Free and open source
- Built-in assertions to verify the response on the fly
- Works through static methods
- Supports BDD
- Contains JSONPath and XMLPath over response body
- http://rest-assured.io/

JSON

- JavaScript Object Notation
- Open-standard file format that uses human-readable text
- Language-independent data format.

Sportsbetting example

REQUEST CONTRACT

- Host: localhost:8080
- Method: POST
- Endpoint: /sportsbetting-web/registration
- Request body:

```
{
  "accountNumber": "1234"
  "balance": "16000"
  "currency": "HUF"
  "dateOfBirth": null
  "name": "testName"
  "password": "Secret1234"
  "userName": "testName-1234"
}
```

EXECUTION

- Create a model for request body
- Create a request with the method and path
- Perform the request
- Analize the response

Implementation overview

Set a base Uri as a global value

```
RestAssured.baseURI = "http://localhost:8080/sportsbetting-web/";
```

Build the request body

```
RegistrationRequest.builder().withAccountNumber(”1234”)...build();
```

The actual requesting

```
Response registrationResponse =

given()

    .contentType(ContentType.JSON)
    .body(request)
    .post(”registration”)

.then()

    .statusCode(200)
    .extract()
    .response();
```

Validate the received response response

```
assertNotNull(registrationResponse.jsonPath().getString(ID));
```

JsonPath and JsonAssert

JsonPath

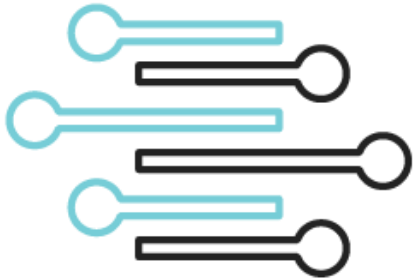
- Java DSL for reading JSON documents
- Dependency of Rest-Assured
- Convenient way of working with result bodies
- Get result as generic list

```
response.jsonPath().getList("path", genericType.class);
```

JSONAssert

- Comparison tool for JSON
- User friendly error message output
- Wide range of built-in asserts
- Customisable Comparator
- Example:

```
JSONAssert.assertEquals(expectedJSON, response.asString(), isStrict());
```



THANK YOU FOR YOUR ATTENTION!