**ASSIGNMENT 3  
API AUTOMATION - REST ASSURED**

**I/ Yêu cầu thực hành**

1. Học viên biết cách tạo một project Maven, sử dụng pom.xml file để cài đặt thư viện
2. Học viên đã có kiến thức cơ bản về API, Rest Assured, TestNG
3. Thực hành đưa giải pháp vào viết code, thực thi và sửa lỗi code nếu có
4. Lưu lại code.

**III. Thực hành**

Thực hiện automation luồng cơ bản sau và kiểm chứng thông tin API.

* Đăng nhập hệ thống với thông tin tài khoản cho trước
* Tạo một thông tin công việc bất kì

*Chi tiết:*1. Đăng nhập hệ thống với tài khoản sau:

**account**: testerFunix  
 **password**: Abc13579

| **Endpoint** | http://13.228.39.1:5000/api/users/login |
| --- | --- |
| **Method** | **POST** |
| **Body** | {  "account": "…...", // tên tài khoản : String.  "password": "…." // Mật khẩu : String  } |
| **Response** | **200 OK**  {  "message": "Đăng nhập thành công",  "token": "………...”  "user": {  "id": ...,  "name": "……..",  "email": "……..",  "account": "…….",  "password": "…….,  "type": "….",  "avatar": “….",  "phone": "…..",  ………………….  "createdAt": "…...",  "updatedAt": "…."  }  } |

**Kiểm chứng thông tin sau:**

* Đảm bảo đăng nhập thành công với HTTP status code = 200 ok.
* Đảm bảo trong Response trả về có trường **message** và nội dung của   
   message là “Đăng nhập thành công”
* Đảm bảo trong Response trả về có chứa trường **token**
* Đảm bảo Response trả về có thông tin user. Thông tin user phải chứa:

- account/password trùng với thông tin account/password đăng nhập.  
 - type là "**UNGVIEN**"

**2.** Tạo một thông tin công việc:

| **Endpoint** | http://13.228.39.1:5000/api/work-user/createWork-user |
| --- | --- |
| **Method** | **POST** |
| **Authen Header** | Thêm header token với thông tin token có được từ bước 1.  “token”: “……..” |
| **Body** | {  "nameWork": "…...", //Tên công việc  "experience": "……..", // Kinh nghiệm: 1 năm, 2 năm...  "education": "……" // Trình độ: đại học, THPT, ..  } |
| **Response** | **201 Created** {  "id": ...,  "nameWork": "….",  "experience": "…..",  "education": "…….",  "UserWorkAccount": …..,  "updatedAt": "………….",  "createdAt": "………….."  } |

**Kiểm chứng thông tin sau:**

- Đảm bảo tạo thành công và nhận về HTTP status code 201 Created.

- Đảm bảo response trả về có chứa trường **id** - định danh công việc.

- Đảm bảo Response chứa đầy đủ các trường tương ứng với lúc tạo. Giá trị của các  
 trường này phải hoàn toàn khớp với lúc tạo

***Một số yêu cầu và gợi ý làm bài.***

1. Tạo project có tên là **ASSIGNMENT-3.**

**Tạo cấu trúc project:**

**1.1** Tạo thư mục có tên là **configuration** chứa 2 properties file :

#**configs.properties :** Khai báo các tham số theo cấu hình như mẫu sau.

| baseurl=http://13.228.39.1:5000  loginPath=/api/users/login  createWorkPath=/……………...  updateWorkPath=/…………………….  DetailWorkPath=/……………………….  account=………..  password=…………... |
| --- |

#**token.properties :** lưu thông tin **token** từ kết quả trả về sau khi đăng nhập thành  
 công. token này dùng để add vào header cho các API sau. (file sẽ tạm thời để trống.)  
  
 **1.2** Trong src/main/java , tạo một package có tên : **com.api.auto.utils**  : package  
 này sẽ chứa các class tiện ích như đọc/ghi properties file.  
 **1.3** Trong src/test/java , tạo một package có tên com.api.auto.testcase : package  
 này sẽ chứa các class để chúng ta viết testcase.  
  
2. Trong package : **com.api.auto.utils,** tạo class có tên **PropertiesFileUtils**

Nhiệm vụ của class này sẽ bao gồm:

- Đọc các giá trị property từ file **configs.properties**

- Ghi giá trị token vào file **token.properties** để các API khác có thể sử dụng.

- Đọc giá trị token từ file **token.properties**  Các phương thức tương ứng:

* **String getProperty(String key)**
* **void saveToken(String token)**
* **String getToken()**

Các bạn có thể tham khảo file :<REF6\_ Properties files.docx>

#Template

| **public** **class** PropertiesFileUtils {  // Đường dẫn đến properties files trong folder configuration  **private** **static** String *CONFIG\_PATH* = <đường dẫn đến configs.properties>;  **private** **static** String *TOKEN\_PATH* = <đường dẫn đến token.properties>;  // 1 Lấy ra giá trị property từ config file bất kỳ theo key.  **public static String getProperty(String key) {**  Properties properties = **new** Properties();  String value = **null**;  FileInputStream fileInputStream = **null**;  // bat exception  **try** {  fileInputStream = **new** FileInputStream(*CONFIG\_PATH*);  ……….//input code here…….  } **catch** (Exception ex) {  System.***out***.println("Xảy ra lỗi khi đọc giá trị của " + key);  ex.printStackTrace();  } **finally** {  // luôn nhảy vào đây dù có xảy ra exception hay không.  // thực hiện đóng luồng đọc  ……….//input code here…….  }  **return** value;  }    // 2.save token vào file token.properties với key là **“token”**  **public** **static** **void** saveToken(String token) {  // Khái báo properties, biến cần thiết  ……….//input code here…….  FileOutputStream fileOutputStream = **null**;  **try** {  fileOutputStream = **new** FileOutputStream(*TOKEN\_PATH*);  //  ……….//input code here……  } **catch** (IOException ex) {  ex.printStackTrace();  } **finally** {  // Luôn nhảy vào đây dù có xảy ra exception hay không.  // thực hiện đóng luồng  ……….//input code here…….  }  }  // 3 lấy ra token  **public** **static** String getToken() {  // Khái báo properties, biến, value  ……….//input code here…….  FileInputStream fileInputStream = **null**;  // bat exception  **try** {  fileInputStream = **new** FileInputStream(*TOKEN\_PATH*);  ……….//input code here……  } **catch** (Exception ex) {  System.***out***.println("Xảy ra lỗi khi đọc giá trị token ");  ex.printStackTrace();  } **finally** {  // Luôn nhảy vào đây dù có xảy ra exception hay không.  // thực hiện đóng luồng  ……….//input code here…….  }  **return** value;  }  } |
| --- |

3. Trong package **com.api.auto.testcase**

Tạo 2 class test có tên tương ứng sau:

| TC\_API\_Login | Thực hiện login và kiểm chứng thông tin API |
| --- | --- |
| TC\_API\_CreateWork | Thực hiện tạo mới công việc và kiểm chứng thông tin API |

*Chú ý.*

* Với mỗi class test, các bạn sẽ đọc các cấu hình cần thiết từ properties file.
* Với API login, các bạn cần lưu lại token để sử dụng cho các API còn lại.
* Mỗi một testcase trong class sẽ được đánh mức độ ưu tiên để đảm bảo thứ tự thực thi. Và chúng sẽ tương ứng với một/một số thuộc tính cần kiểm chứng theo yêu cầu của đề bài.
* Với API như POST, PUT thay vì sử dụng body dưới dạng String nhìn rắc rối, chúng ta có thể thay thế bằng việc sử dụng hashMap < key-value.>

*Ví dụ như sau:*

*#String*

| String email = "abc@gmail.com";  String password = "admin"; …...  String account = "{\"email\": \"" + email + "\",\"password\": \"" + password + "\"}";  RequestSpecification requestSpecification = RestAssured.*given*()  .contentType(ContentType.***JSON***)  .body(account); |
| --- |

*#HashMap*

| String email = "abc@gmail.com";  String password = "admin"; …………………..  *//HashMap*  Map<String, Object> account = **new** HashMap<String, Object>();  body.put("email", email);  body.put("password", password);  RequestSpecification requestSpecification = RestAssured.*given*()  .contentType(ContentType.***JSON***)  .body(account); |
| --- |

#Template  
**TC\_API\_Login**

| **public** **class** **TC\_API\_Login** {  **private** String account;  **private** String password;  **private** Response response;  **private** ResponseBody responseBody;  **private** JsonPath jsonBody;  @BeforeClass  **public** **void** init() {  // Init data  String baseUrl = ……...;  String loginPath = ……..;  account = …………………...  password = ……………………...  RestAssured.*baseURI* = baseUrl;  // make body  Map<String, Object> body = **new** HashMap<String, Object>();  body.put…..   *…...//code here…..*  RequestSpecification request = RestAssured.*given*()  .contentType(ContentType.***JSON***)  .body(account);  response = request.post(loginPath);  responseBody = response.body();  jsonBody = responseBody.jsonPath();  System.***out***.println(" " + responseBody.asPrettyString());  }  @Test(priority = 0)  **public** **void** TC01\_Validate200Ok() {  // kiểm chứng status code  *…...//code here…..*  }  @Test(priority = 1)  **public** **void** TC02\_ValidateMessage() {  // kiểm chứng response body có chứa trường message hay không  // kiểm chứng trường message có = "Đăng nhập thành công  *…...//code here…..*  }  @Test(priority = 2)  **public** **void** TC03\_ValidateToken() {  // kiểm chứng response body có chứa trường token hay không  String token = jsonBody.getString("token");  // lưu lại token  *…...//code here…..*  }  @Test(priority = 3)  **public** **void** TC05\_ValidateUserType() {  // kiểm chứng response body có chứa thông tin user và trường type hay không  // kiểm chứng trường type có phải là “UNGVIEN”  *…...//code here…..*  }  @Test(priority = 4)  **public** **void** TC06\_ValidateAccount() {  // kiểm chứng response chứa thông tin user và trường account hay không  // Kiểm chứng trường account có khớp với account đăng nhập  // kiểm chứng response chứa thông tin user và trường password hay không  // Kiểm chứng trường password có khớp với password đăng nhập  }  } |
| --- |

**TC\_API\_CreateWork**

| **public** **class** TC\_API\_CreateWork {  **private** String token;  **private** Response response;  **private** ResponseBody responseBody;  **private** JsonPath jsonBody;    //Chúng ta có thể tự tạo data  **private** String myWork = …….  **private** String myExperience = …….  **private** String myEducation = …….    @BeforeClass  **public** **void** init() {  // Init data  String baseUrl = ……...;  String createWorkPath =…………….  String token = ……………..  RestAssured.*baseURI* = baseUrl;  // make body  Map<String, Object> body = **new** HashMap<String, Object>();  body.put(….,….);  *…...//code here….*  RequestSpecification request = RestAssured.*given*()  .contentType(ContentType.***JSON***)  .header("token", token)  .body(body);  response = request.post(createWorkPath);  responseBody = response.body();  jsonBody = responseBody.jsonPath();  System.***out***.println(" " + responseBody.asPrettyString());  }    @Test(priority = 0)  **public** **void** TC01\_Validate201Created() {  // kiểm chứng status code  *…...//code here…..*  }  @Test(priority = 1)  **public** **void** TC02\_ValidateWorkId() {  // kiểm chứng id  *…...//code here…..*  }  @Test(priority = 2)  **public** **void** TC03\_ValidateNameOfWorkMatched() {  // kiểm chứng tên công việc nhận được có giống lúc tạo  *…...//code here…..*  }  @Test(priority = 3)  **public** **void** TC03\_ValidateExperienceMatched() {  // kiểm chứng kinh nghiệm nhận được có giống lúc tạo  *…...//code here…..*  }  @Test(priority = 4)  **public** **void** TC03\_ValidateEducationMatched() {  // kiểm chứng học vấn nhận được có giống lúc tạo  *…...//code here…..*  }  } |
| --- |

4. Tạo test suite: **testng.xml** để chạy testcase theo thứ tự   
 TC\_API\_Login → TC\_API\_CreatedWork  
  
file **testng.xml** sẽ đặt trong src/test/java

Chú ý nội dung file phải đảm bảo thứ tự thực thi API

<class name=*"com.apiauto.testcase.TC\_API\_Login"*/>

<class name=*"com.apiauto.testcase.TC\_API\_CreateWork"*/>