**Information About project :**

It’s a simple API project to manage users and tokens. The source code was given with all instruction to assure the test.

We have 2 modules to test : TokenController and UsersController

**Environment for test**

The test will be executed on local using docker.

We have already a default admin user to permit token generation and to create new user and all metrics to test all situation

To release the test we are using Junit

**The goals :**

We are testing only end to end API and only Token controller and User controller. The other controller and test are not include into the perimeter

We are waiting for status code from each request.

The following requirements must be met

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Description | criticality | Category |
| TOK\_01 | Token creation/refresh - Getting 201 status using correct user/password | Get 201 status if request with correct user/Password is used - Use json playload with format : { "username": "<user name>", "password": "<password>" } - We get a http 201 status  - We get response with json format : {"token": "String", "creationDate": "yyyy-MM-dd HH:mm:ss", "expirationDate": "yyyy-MM-dd HH:mm:ss"}  Remarque : An user can create token if the user/password are correct | Critical | non-functional test |
| TOK\_02 | Token creation/refresh - Getting 200 status using correct user/password which not expired | Get 200 status if request with correct user/Password is used and token is not expired - Use json playload with format : { "username": "<user name>", "password": "<password>" } - We get a http 200 status after performing request - We get response with json format : {"token": "String", "creationDate": "yyyy-MM-dd HH:mm:ss", "expirationDate": "yyyy-MM-dd HH:mm:ss"}  - The tokn don't change, but the expiration date will be extend  Remarque : An user can create token if the user/password are correct | Critical | non-functional test |
| TOK\_03 | Token creation/refresh - Getting 401 status using incorrect user/password | Get 401 status if request with incorrect user/Password is used  No token is created | Critical | non-functional test |
| TOK\_04 | User creation - Getting 201 statut using correct or updated token | Get 201 statut if  User used correct or updated token Authorization: bearer <valid token> the user name and password are filled and json format { "username": "<user name>", "password": "<password>" } | Critical | non-functional test |
| TOK\_05 | User creation - Getting 409 statut if attempting to create a duplicate user | Get 401 statut if  User used correct or updated token Authorization: bearer <valid token> But the user name and password are existing | Critical | non-functional test |
| TOK\_06 | User creation - Getting 401 statut using an expired valid token | Get 401 statut if  User used correct or updated token Authorization: bearer <valid token> who is expired But the user name and password are existing | Critical | non-functional test |
| TOK\_07 | User creation - Getting 403 statut using a non admin unexpired token | Get 403 statut if  User used correct or updated token Authorization: bearer <valid token> who is not a admin token But the user name and password are existing | Critical | non-functional test |

**Test strategie :**

We are using Robot framework to performe the test

Each test should be tested on Postman befor to be implemented, if there are some issue, it should be write inside the tools management used by the client

We are testing :

* first : All test case about Token creation/Refresh by verify the status codes
* Second User creation by verify the status code
* Write end to end test if all case are correcte by using a generated token and verify the status code

**Stop condition :**

* Negative : We stop the test if there is an blicking error who block all next test
* Positif : We stop test if all tests pass successful and attend all goals

**Prerequiste :**

The micro service application

Crédential for admin

Excepted result for each test

**Deliverable**

* Git repository of the project and test

Project : <https://github.com/njartiana/IZICAP>

TA : <https://github.com/njartiana/Test_auto_izicap>

* Test plan and all related documents
* Test book and requirements

Requirements and book test.xlsx

* A readme to explain how to run the code