

Définitions des tests

EPIC ROAD TRIP Application

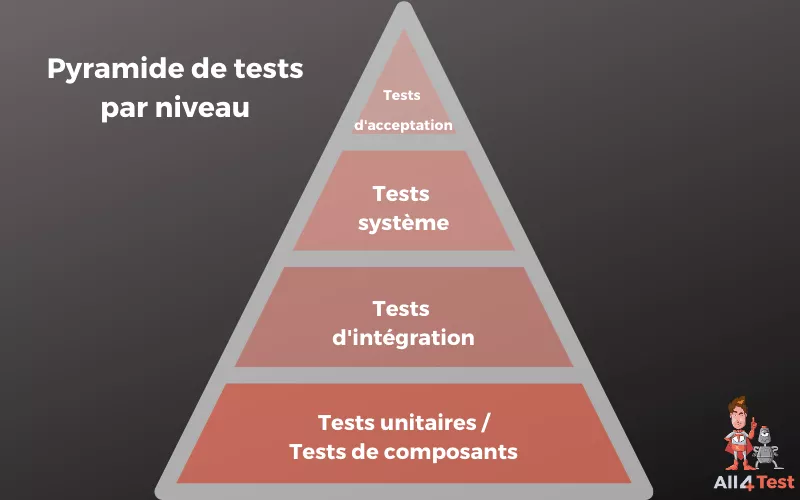
Teams :

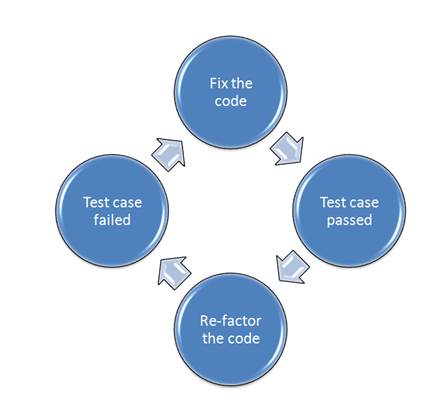
* Aurélie Poirier
* Lilian Garrido
* Tom Léal
* Loïc Thomas

1. **INTRODUCTION**

Nous allons mettre en place des tests pour évaluer automatiquement la qualité et la robustesse de notre code à chaque push sur la branche “development”. Cependant nous n'avons utilisé que trois niveaux de test : les tests d’acceptations, les tests systèmes, les tests d’intégrations et les tests unitaires.

1. **ARBORESCENCE**





1. **DÉTAILS**
   1. **API**

|  |  |  |
| --- | --- | --- |
| **Type** | **Titre** | **Description** |
| TI - 1 | Places | Test routes places on API |
| TI - 1.1 | Get places by name | * should respond with a 401 status code when token is not exist |
| TI - 1.2 | Get details place | * should respond with a 200 status code when search Paris like city |
| TI - 2 | Users | Test routes users on API |
| TI - 2.1 | Create USERS only ADMIN | * should respond with a 401 status code because Unauthorized without admin's token * should respond with a 200 status code because have the admin's token |
| TI - 2.2 | Get USERS only ADMIN | * should respond with a 200 status code with admin account |
| TI – 2.3 | Get USER | * should respond with a 200 status code with admin account |
| TI – 2.4 | Patch USERS | * should respond with a 404 status code because id is missing * should respond with a 200 status code with admin account |
| TI – 2.5 | Delete USERS only ADMIN | * should respond with a 404 status code because id is missing * should respond with a 200 status code with admin account |
| TI - 3 | Register / Login | Test routes Register /Login on API |
| TI – 3.1 | Register when the data is missing | * should respond with a 400 status code because username is missing * should respond with a 400 status code because email is missing * should respond with a 400 status code because password is missing |
| TI – 3.2 | Register by given all data | * should respond with a 200 status code * should respond with a 409 status code because user email is already exist |
| TI - 3.3 | Login user after register | * should respond with a 200 status code |
| TI - 3.4 | Delete user after login but without token | * should respond with a 401 status code |
| TI – 3.5 | Delete user after login and get token | * should respond with a 200 status code |
| TI - 4 | Trip | Test routes trip on API |
| TI – 4.1 | Post trip with token | * should respond with a 200 status code with object response |
| TI – 4.2 | Post trip without token | * should respond with a 401 status code because no token |
| TI – 4.3 | Get trips by user id | * should respond with a 200 status code with object response |
| TU - 1 | Generate | Test function for generate route |
| TU – 1.0 | Create USERS only ADMIN | * should respond with a 401 status code because Unauthorized without admin's token * should respond with a 200 status code because have the admin's token |
| TU – 1.1 | Generate report road trip | * Should respond with a 401 because no token existing * should respond with a 200 status code because have the user's token |
| TU – 1.2 | Get error when key report is not defined from road trip body | * Should respond with a 500 because body doesn't have ek key activities for 'to' |
| TU – 1.3 | Get error when key report is not defined on report road trip body | * should respond with a 500 status code because any value in key 'to' exists |
| TU – 2 | Security | Test middleware function |
| TU – 2.1 | Get error on protected route with fake token | * should respond with a 200 status code because have the admin's token |
| TU – 2.2 | Get error on protected route | * Should respond with a 401 because user connected is unauthorized because the token no correspond to id |
| TU – 2.3 | Get error on protected route | * should respond with a 404 status code because header request not set |

* 1. **WEB interface**

|  |  |  |
| --- | --- | --- |
| **Type** | **Titre** | **Description** |
| TC - 1 | App | Test App component |
| TC – 1.1 | App | * should renders without crashing |
| TC - 2 | Home | Test Home component |
| TC – 2.1 | Home | * should renders without crashing |
| TC - 3 | Maps | Test Maps component |
| TC – 3.1 | Maps | * should renders without crashing |
| TC - 4 | SignIn | Test SignIn component |
| TC – 4.1 | Render | * should renders without crashing |
| TU – 4.2 | Form | * should login |
| TC - 5 | SignUp | Test SignUp component |
| TC – 5.1 | Render | * should renders without crashing |
| TU – 5.2 | Form | * should register |
| TC - 6 | Layout | Test Layout component |
| TC – 6.1 | Layout | * should renders without children * should renders with children |