

# Test Technique

En préambule je m'excuse pour le temps que j'ai pris, j'ai notamment pris du temps à réinstaller un environnement de code potable sur mon PC personnel, sur lequel je n'ai jamais vraiment développé.

## *I - What bugs did you find ? How did you solve these and why ?*

### 1) Le nom de l'utilisateur qui ne reste pas

J'ai adopté une stratégie simple, j'ai suivi les directions qui m'étaient données, et j'ai corrigé les bugs qui me tombaient dans les bras, que je savais corriger.

J'ai fait ainsi puisque je ne connais pas vraiment cette technologie, et donc je préférais découvrir les bugs par le fonctionnel qu'en me baladant au hasard dans le code.

J'arrive donc au niveau de la connexion/inscription, je m'inscris à base de "test", je note au passage que le mot de passe est caché mais il est impossible d'y jeter un oeil (j'étais pas sûr d'avoir bien écrit "motdepasstest"), mais bon, ce n'est pas un bug, juste une petite optimisation à éventuellement ajouter.

J'arrive à m'inscrire, on me demande d'aller dans **People** créer deux autres utilisateurs et là, premier bug, leur nom n'apparaît ni sur ce que j'appellerais leur "usercard", ni dans leurs informations.

Je cherche donc comment trouver ce bug dans le code, en partant du drawer.js, là où je sais que je pourrais descendre dans l'architecture à partir de **People**.

Je descend, trouve les inputs du "username", "email" et "password". Comme les deux derniers se retrouvaient dans les informations, je me doutais que quelque chose clochait seulement et uniquement avec le username. Bingo, la définition du "username" n'existe pas dans le modèle de "user", mais un certain "name" y est, juste à vérifier si c'est plutôt le modèle ou l'input qui diffère. J'ai fini par opter pour un changement côté input. Je crois constater que le terme "username" est surtout utilisé pour le user actif, pas pour les membres ajoutés par ce dernier (et donc plus présent sur les script auth.js, signin.js par exemple).

Ça marche, premier bug résolu !

## 2) La mise à jour des informations qui ne se fait pas

En essayant de modifier les informations de mes nouveaux collaborateurs “name1” et “name2”, je me rends compte que le bouton **Update** ne répond pas. Voilà mon deuxième bug.

Comme je me suis un peu baladé, je sais où trouver ce bouton “update”, ou plutôt ce “LoadingButton”. Je me concentre donc sur le script view.js des users.

J’ai mis un peu de temps à comprendre ce qui clochait hier, mais je savais que c’était ce bouton, car au click il ne faisait rien. J’ai fini par trouver en comparant avec le “LoadingButton” **Save** utilisé auparavant pour créer ces utilisateurs.

Le problème était finalement sous mes yeux depuis le début, “onChange” au lieu de “onClick”.

Et voilà, deuxième bug réglé.

## 3) L’ouverture du projet

Bon, il est temps d’aller créer un projet, je crée donc mon sobre “project1”. Parfait, plus qu’à le modifier ... et hop callStack d’erreur. Très bien, c’est donc mon troisième bug à résoudre.

Bon, comme je n’ai pas forcément l’habitude d’utiliser ces stackTrace, je préfère ouvrir la console web, je l’analyse un temps et vois un problème avec l’appel d’un “toString()” apparemment non défini (ce qui m’aurait jamais choqué, venant du java).

Je cherche donc naturellement dans le script “view.js” du dossier “project”, trouve ce fameux “toString()”, l’enlève, et hop, problème résolu.

## 4) La modification du projet

On accède au projet (je note déjà que son nom ne s’affiche pas non plus).

J’essaie alors de l’éditer et je me retrouve dans un loading infini, avec une erreur 500 en fond.

Après beeeaucoup de console.log, je fini par comprendre le soucis: l’id du projet n’est pas transmis, et donc la valeur de “project” est **null**.

Après encore plus de log, je fini par comprendre pourquoi elle n’est pas transmise: le `${project?._id}` ne permet pas sa transmission, project étant une liste . Je le remplace par `${project[0]._id}`, récupéré à ce niveau là auparavant, fait des tests pour vérifier que le bon id est récupéré après transmission, tout est bon. La variable project n’est donc plus nulle, et le chargement s’effectue.

J'en ai profité pour faire afficher le nom du projet dans la liste des projets et une fois ce dernier sélectionné, mais je ne suis pas allé plus loin.

Au passage j'ai conscience que la solution d'utiliser `project[0]` n'est pas la meilleure, mais faute de temps, j'ai préféré m'en contenter.

J'ai aussi choisi de m'arrêter là pour la résolution de bugs.

## *II - Which feature did you develop and why ?*

Bon, c'est là où ma faible connaissance de la technologie pose problème, j'ai bien fais deux trois tentatives de petit ajout, mais rien de concluant.

J'ai laissé en commentaire ma dernière idée: un envoi de mail automatique à l'ajout d'un partenaire, puisque nous pouvons fournir son adresse mail, c'était l'occasion de le prévenir et de lui permettre de rejoindre l'application.

Mes autres idées était de l'elasticSearch dans les barres de recherche, ou de permettre au description d'accueillir des images.

## *III - Do you have any feedback about the code / architecture of the project and what was the difficulty you encountered while doing it ?*

Contrairement à ce dont j'avais peur au début, j'ai pas eu de soucis pour me déplacer dans le code applicatif, tout me semblait assez naturel et bien rangé, et une fois que je me suis fais à la logique du projet, je savais où aller pour trouver ce que je voulais.

Mes problèmes sont ici plus techniques, je n'ai aucun problème à m'inspirer de ce qui est déjà fait quelque part pour résoudre un bug proche ou mettre en place une fonctionnalité similaire, mais repartir de zéro est pour l'instant un peu trop ambitieux pour moi. En outre, les soucis sont plus sur ma personne que sur le code lui-même.