

Découvrez un bug complexe

7–8 minutes

Félicitations, vous voici arrivé au terme de ce cours. Maintenant que vous savez comment déboguer des bugs d'intégration et de JavaScript, il est temps de tout combiner.

Donnez du contexte à un bug

C'est-à-dire ? Tout combiner ? On va traiter plusieurs bugs à la fois ?

Tout à fait !

En effet, quand vous rencontrez une erreur que vous devez déboguer, il y a deux types de situation :

- **soit le bug est “tout seul”**. Autrement dit, il y a un bug, d'affichage par exemple, qui n'influe que sur l'affichage du projet sur une page spécifique. Dans ce type de cas, vous pouvez appliquer les méthodes de débogage que nous avons vues ensemble dans les parties précédentes.
- **soit le bug engendre un autre bug** qui engendre un autre bug (qui engendre... bref, je pense que vous avez compris 😊). Et dans la plupart des cas, c'est ce qui se produit. En effet, dans le cas d'applications complexes réalisant beaucoup de traitements, comme des appels API, de l'affichage dynamique sur le DOM, l'erreur qui apparaît, dans votre console par exemple, ne peut être que la face visible de l'iceberg.

Il est aussi important de préciser que le bug peut venir de chez vous mais pas que !

Euhhh, c'est-à-dire ? Le bug pourrait venir d'où ?

Il peut aussi venir du **serveur**, l'API peut être boguée, les données que vous demandez à la base de données peuvent être incorrectes, etc. Le plus souvent, c'est l'analyse du code qui va vous permettre de voir d'où il peut venir.

J'ai récemment travaillé sur un projet Python où je devais insérer des données d'une certaine façon dans une table. C'est la donnée la plus récente qui devait être entrée mais il s'avérait que ce n'était pas tout le temps le cas. Il m'a bien fallu une journée pour comprendre d'où pouvait provenir le bug : les données étaient modifiées à plusieurs endroits de l'application. Bref, ce n'était pas très drôle.



Lors de la rédaction de ce cours, j'ai eu la chance de pouvoir travailler avec **Quentin Laurent**, développeur tout comme moi et mentor : il m'a aidé à la relecture de ce cours et m'a aussi donné quelques bonnes idées pour le projet fil rouge. Il a aussi dû faire face à des bugs marquants, et en voici un en particulier :

Je me souviens d'un bug particulièrement bloquant lié à Firebase. Pour faire simple, Firebase permet, entre autres, de gérer la partie authentification d'un projet. Quand on travaille côté web, on doit l'utiliser en ajoutant un script côté code et c'est ce que je faisais. Seulement voilà, ça ne marchait pas ! L'objet Firebase était bien présent dans ma console mais pas il refusait de fonctionner. En cherchant un peu, j'ai découvert que j'avais des problèmes de traitement asynchrone en cascade dans mon programme : c'était un véritable enfer à déboguer !

Bref, comme vous pouvez le constater, quel que soit votre niveau et votre expérience, nous rencontrons tous des bugs qui peuvent parfois être difficiles à résoudre. C'est l'un des éléments les plus challengeants de votre travail de développeur. Cela dit, pas de panique, vous serez rarement seul à travailler dessus.



Analysez la situation

Tirez parti des méthodes vues dans les chapitres précédents pour résoudre le bug

Est-ce que vous vous souvenez de notre méthodologie de débogage vue dans le premier chapitre du cours ?

Euhhhh....

Si vous ne vous en souvenez plus, ces étapes étaient :

- Observer le ou les bugs ;
- Écrire un test répétable ;
- Élaborer et tester votre théorie ;
- Résoudre le bug (ou réitérer les étapes précédentes).

Il est important que vous essayiez de suivre ces étapes au début de votre apprentissage du code et du débogage. Après quelques années d'expérience, vous le ferez de façon instinctive et n'y penserez même plus.

N'hésitez d'ailleurs pas à prendre une feuille de papier et un crayon pour réfléchir à votre bug. Je me servais de cette méthode il y a quelques années quand j'ai appris déboguer. Elle me permettait de poser mes idées avant de passer par la phase code.

Apprenez à isoler un bug majeur

Un de mes amis a dernièrement eu des problèmes d'envoi d'e-mails. Il utilisait un site sous WordPress pour un comparateur de prix. Quand ses prospects remplissaient leur formulaire, il souhaitait être prévenu par mail qu'un client venait de saisir ses coordonnées. Sauf que les e-mails ne partaient pas et il n'arrivait pas à voir d'où pouvait venir le problème. Je lui ai donc demandé de me fournir l'ensemble des informations qu'il avait, à savoir :

- le message d'erreur ;

- le plugin utilisé ou au moins le code Php relatif à ce plugin ;
- l'URL du WordPress ;

J'ai assez rapidement pu voir que la librairie qui était utilisée pour envoyer les e-mails était PhpMailer. Avec l'aide d'un script Php, j'ai pu m'envoyer un e-mail de test avec la librairie PhpMailer via mon compte gmail. Cela m'a permis d'avoir un script qui marchait.

J'ai ensuite changé les informations de connexion pour remplacer mes identifiants par les siens et là, patatra, ça ne marchait pas. Cela dit, ça m'a permis de savoir que c'était possiblement un problème de configuration de l'email et non un problème dû à la librairie ou au WordPress : j'ai donc isolé le problème.

En cherchant un peu sur Internet, j'ai vu que le serveur Smtп était mal configuré, je l'ai donc modifié et le problème a été réglé. J'ai pu ensuite appliquer la solution sur son WordPress.

La morale de cette histoire est qu'il est important, face à un bug complexe, d'isoler le problème : il ne faut surtout pas essayer de tout déboguer à la fois, c'est le meilleur moyen de perdre du temps et de ne pas comprendre d'où le problème vient. En y réfléchissant bien, le problème de mon ami aura pu venir de plein d'endroits : de la configuration du WordPress, de son serveur, du serveur Smtп, des identifiants de connexion, de la librairie, etc.

À vous de jouer !



Bon, c'est l'heure pour vous d'affronter le boss final ! Le projet Façadia semble avoir un bug bien gênant et vous allez devoir l'analyser grâce à la méthodologie de débogage.

Comme pour les exemples précédents, voici :

- l'issue du bug : <https://github.com/tdimnet/debuggez-l-interface-de-votre-site/issues/5>
- la branche contenant le bug : [partie-5/chapitre-1/bug-complexe](#)

Comme je le disais un peu plus haut, n'hésitez pas à prendre une feuille de papier et un crayon et notez vos idées. Essayez d'observer et de créer un test répétable pour le bug.

Une fois que vous pensez avoir identifié le bug, vous pouvez me rejoindre dans la vidéo ci-dessous.



En résumé

- Un bug peut soit arriver tout seul, soit arriver en chaîne. Essayez toujours d'analyser la situation avant d'écrire votre code.
- N'oubliez pas la méthode que nous avons vue dans le premier chapitre pour résoudre les bugs : observez le ou les bugs, puis écrivez un test répétable, élaborez et testez votre théorie puis enfin résolvez le bug (ou réitérez les étapes précédentes).
- Quand un bug gênant se produit, essayez toujours de l'isoler pour comprendre sa source.

Maintenant que vous avez vu comment analyser et répliquer un bug complexe, vous pouvez vous intéresser à sa résolution.