

Découvrez les sources de bug potentielles

4–5 minutes

Découvrez l'origine des bugs Front-End

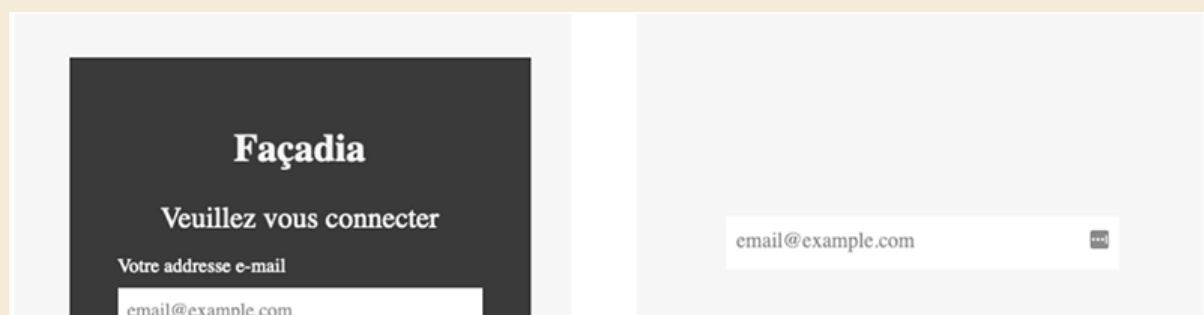
Tous les sites web ont des bugs ! Certains en contiennent plus que d'autres mais le site parfait n'existe pas, et il est de notre devoir, en tant que développeurs, de **corriger ces bugs** : c'est une partie essentielle de notre travail.

Dans ce cours, nous allons nous intéresser aux bugs qui peuvent survenir sur **la partie front-end d'un site** et à comment les résoudre de manière efficace.

Quand vous travaillez sur la partie front-end d'un site, il n'existe pas un seul bug type (ce serait trop simple !) En effet, c'est une interface assez complexe et vous allez réaliser un grand nombre d'opérations différentes où :

- les données peuvent s'afficher et se modifier de façon dynamique,
- le contenu doit s'adapter à la taille de l'écran,
- etc.

De ce fait, certains bugs vont être détectables via des erreurs, dans la console par exemple, alors que d'autres vont être un petit peu plus sournois, comme un problème d'affichage sur un navigateur.





Ci-dessus l'original (sans bug, à gauche) et une version (à droite) avec un bug d'affichage CSS



Ci-dessus une erreur JavaScript présente dans la console

Prenons un moment pour réfléchir à toutes les sources potentielles de bug côté front :

- **Côté HTML** - on peut avoir oublié de fermer une balise ou alors avoir utilisé la mauvaise classe (et donc que le style appliqué ne soit pas le bon).
- **Côté CSS** - il peut s'agir d'éléments qui s'affichent mal (ou pas du tout) sur un certain type d'appareil ; les smartphones par exemple.
- **Côté DOM** - les bugs peuvent être liés à la mauvaise sélection, modification ou suppression d'un nœud. Par exemple, on souhaite supprimer un nœud qui n'existe pas (parce qu'on l'a mal sélectionné).
- **Côté API** - dans le cas d'une API interne (l'API de géolocalisation ou de stockage par exemple), il se peut que vous l'utilisiez mal, autrement dit que vous n'ayez pas bien compris la documentation. Dans le cas d'une API externe (API de météo par exemple), il se peut que vous ne lui envoyiez pas les bonnes informations dans la requête HTTP.

Découvrez le projet fil rouge : le site Façadia

Avant de nous lancer dans le vif du sujet et d'aborder les outils de débogage, nous allons faire un petit détour par notre projet fil rouge : le projet Façadia, un site de façades d'immeubles connectées.

Vous pouvez dès à présent récupérer [le code source du projet ici](#).

J'ai aussi utilisé Figma pour réaliser les maquettes de ce projet.

Vous pouvez ouvrir le [Figma du projet](#) dans un onglet de votre navigateur, nous l'utiliserons durant le cours.

Si vous avez suivi le cours sur le [testing d'application en JavaScript](#), il est possible que ce projet vous dise quelque chose : c'est normal, c'est le même ! Cela dit, si vous comparez le code source du projet de testing et celui-ci, vous verrez que pas mal de choses ont changé côté code. Nous sommes sur **un site statique plus "traditionnel"** :

- Il y a un dossier `pages` qui comprend toutes les pages HTML du projet, en dehors de la page `index.html` qui est située à la racine du projet.
- Le projet n'utilise aucune librairie externe. C'est du JavaScript pur ; on appelle ça aussi du Vanilla JavaScript.
- Chaque page comprend plusieurs balises `script` ; elles vous permettent de récupérer les fichiers JavaScript à utiliser pour chaque page.

Le screencast ci-dessous va vous montrer comment prendre en main le projet et nous allons découvrir ensemble les différents fichiers le constituant.

En résumé

- Les bugs sur une interface front-end peuvent venir de nombreux endroits : HTML, CSS, DOM, APIs.
- Les bugs HTML et CSS sont plus compliqués à détecter : ces derniers ne lèvent aucune erreur dans la console.
- Les bugs relatifs au DOM sont très fréquents. Ils peuvent survenir si un nœud a été mal sélectionné ou supprimé.
- Les bugs d'API peuvent arriver aussi bien sur les API internes (géolocalisation, LocalStorage) qu'externes (récupération de données météo par exemple).

Vous connaissez maintenant les différentes sources de bug, il est temps de s'intéresser aux outils à votre disposition !