

# Défi : Créez un plan de test



Vous voulez vous entraîner à créer un plan de test pour une application ? Nous vous proposons un challenge : réalisez la mission ci-dessous en 1 heure ! Les méthodes que vous allez appliquer ici vous feront gagner du temps sur votre projet. Suivez nos instructions pas à pas, et découvrez la solution juste en dessous. C'est parti !

## Votre mission

Vous êtes développeur chez Devetmoi.



L'équipe de développeurs vient de terminer la première version du site web pour votre nouveau client : un éditeur de jeux qui propose de calculer le score au jeu du Scrabble.

Voici le brief de ce projet.

“Notre nouveau client veut pouvoir proposer un site web permettant de calculer le nombre de points d’un mot au Scrabble. L’idée est d’avoir quelque chose de très simple :

- une interface permettant à l'utilisateur de saisir un ou plusieurs mot(s) ;
- un bouton de validation pour lancer le calcul des points du mot ou des mots ;
- une fonctionnalité “ajouter” ou “supprimer” des mots à l’aide de deux petits boutons (plus et moins) ;
- des “bonus”, comme le fait de vérifier si un mot est un palindrome, ou bien si un mot contient les 2 lettres “OC” l’une à la suite de l’autre.

Anna, votre lead dev, vous envoie un message pour vous confier une mission sur ce projet.

Anna :

Salut ! Je ne sais pas si tu as vu, mais la première version du site web est prête ! Pourrais-tu réaliser le plan de test pour t’assurer que le site fonctionne correctement ?

Pour rappel, le repo GitHub est [ici](#).

Habituellement, nous utilisons ce [template](#). Je t’ai mis la planification d’un premier test, pour que tu puisses bien visualiser ce que je veux.

L’idée ici est de tester de petites portions de code pour s’assurer du bon fonctionnement de chaque fonctionnalité indépendamment des autres fonctionnalités qui l’entourent. Plus on teste de lignes de code, plus on s’assure du bon fonctionnement du projet. Il est donc nécessaire de tester l’ensemble des fonctions présentes dans le code, dans le but d’avoir un code coverage élevé.

Pour être efficace pense à :

- Identifier les fonctions à tester ;
- Identifier les lignes de code qui correspondent à cette fonction ;
- Identifier le résultat attendu ;

- Déterminer comment atteindre ce résultat ;
- Identifier les problèmes possibles.

Si tu veux en savoir plus sur les plans de tests, je te conseille cet [article](#).

Merci d'avance,

Anna

Moi :

Super, je commence tout de suite.

Avec toutes ces informations, vous vous mettez au travail immédiatement !

## Votre livrable

- Le plan de tests complété

## La solution

- [Le plan de tests corrigé](#)