Stratégie de test

Façadia / Façades connectées

Version du document n°1.1
17/07/2023

Contrôle des documents :

Détails du document :

Titre	*Stratégie de tests Façadia*
Version	*1.1*
Date	*17/07/2023*
Nom du fichier	*Stratégie de test*
Localisation du fichier	*Drive google*
Auteur	*Loïc Calcagno*
Contributeurs	*Thomas Dimnet*

Contrôle des modifications :

Date de la modification	Version	Détails	Auteur
16/07/2023	*1.0*	*Création du fichier et rédaction de la partie présentation des tests.*	*Thomas Dimnet*
17/07/2023	1.1	Finalisation du document avec la rédaction des rôles et responsabilités de chacun	Loïc Calcagno

Référence du document :

Nom du document	Localisation du fichier
Stratégie de test	www.url.com/strategie-test

Contenu

1. Portée et présentation du projet

Le projet Façadia permet de suivre l'état de façade connectées via des capteurs. Ces capteurs donnent des informations sur la localisation géographique, le taux d'humidité ainsi que les conditions météorologiques.

L'objectif du projet est de mieux suivre comment se comportent les bâtiments, par exemple comment ils retiennent la chaleur, dans le temps.

2. Approche des tests

Le projet Façadia est actuellement en phase de croissance. Notre nombre d'utilisateurs grandit très fortement et des clients importants, tels que LaFarge, utilisent notre solution.

De nombreux tests vont être réalisés pour tenir à bien les exigences de ce projet

3. Niveau de tests

Les équipes techniques vont devoir se concentrer sur la réalisation :

- de tests unitaires. En fonction de la complexité du code, il est possible que chaque fonction ne soit pas testée. Il est demandé aux équipes techniques de réaliser des tests unitaires principalement sur les éléments qui sont utilisés à plusieurs endroits de l'application. Nous pensons par exemple aux méthodes de parsing et de transformation de données.
- de tests d'intégration. Avant de se lancer dans la réalisation de tests d'intégration, les équipes techniques devront faire un point avec leur manager (Lead Dev et Produit) concernant les tests d'intégration à réaliser.
- de tests manuels tant durant les phases de développement qu'au moment de la mise en production.

4. Types de tests QA

Ne disposant pas actuellement d'équipe QA dédiée, les tests manuels seront réalisés par le Product Owner du projet (Matthieu Bessol). Il est important que des cas de tests soient co-construits par les équipes techniques et produits.

Les tests QA seront stockés sur Github. Avant chaque mise en production, les cas de tests devront être réalisés pour s'assurer que l'application fonctionne comme convenu.

5. Rôles et responsabilités

Rôles	Responsabilités
Chef de projet	Matthieu Bessol
Lead Dev	Quentin Laurent
QA Engineer	Loïc Calcagno

6. Contraintes de l'environnement

Pour réaliser les deux environnements seront mise à disposition des équipes :

- Un environnement de développement. Cela permettra aux équipes techniques de pouvoir tester le projet à minima lors des phases de développement.
- Un environnement de pré-production (ou stagging). Cet environnement reproduira l'environnement de production. Il permettra à l'équipe "produit" de tester le projet dans des conditions les plus proches de l'environnement de production.

En plus de ces deux environnements, un système d'intégration continue permettra aux développeurs de faire tourner les tests à chaque fois que du code est envoyé sur la solution de versionning choisie.

Pour finir, le projet Façadia dispose :

- d'un front-end propulsé par une solution maison. Cette solution utilise le HTML, le CSS et le JavaScript.
- d'un back-end utilisant NodeJS dans sa version 18

7. Outils de tests

Les outils suivant seront utilisés pour réaliser les tests :

- Jest pour réaliser l'ensemble des tests unitaires. C'est l'outil principal que les équipes utilisent pour réaliser les tests.
- Jest DOM pour réaliser des tests d'intégration. Jest DOM permet de répliquer un DOM pour parcourir des pages et ajouter ou supprimer des nœuds.
- Testing Library est utilisée en complément des Jest DOM et facilite la sélection de nœuds.

8. Livrables de tests

Deux types de livrables vont devoir être réalisés pour le projet :

- Des fichiers de test dédiés au sein du projet. Ils devront comprendre le **.test.**, par exemple ***index.test.js***. Si plusieurs types de tests seront présents au sein d'un même dossier, il est important de préciser lesquels sont des tests unitaires et lesquels sont des tests d'intégrations. Par exemple :*index.integration.test.js* et *index.unit.test.js*
- Des issues Github devront être réalisées pour préciser les cas de tests à réaliser.

9. Mesures des tests

Pour mesurer l'efficacité et le nombre de tests réalisés, plusieurs outils vont devoir être utilisés :

- Un Code Coverage sera généré automatiquement par la CI et sera consulté régulièrement par les porteurs du projet.
- Un outil de monitoring permettra de suivre le nombre d'erreurs qui remontent sur le projet en production.

10. Outils de suivi (reporting)

Pour ce projet :

- Les issues GitHub serviront de backlog pour les bugs à traiter.
- Les issues GitHub serviront de cas de tests.
- GitHub Project sera utilisé pour réaliser le suivi des bugs et tâches.

11. Validations

Les personnes suivantes ont approuvé ce document :

Rôles	Responsabilités
Chef de projet	Matthieu Bessol
Lead Dev	Quentin Laurent
QA Engineer	Loïc Calcagno