

Proposition technique pour un outil de test du standard d'API de covoiturage

lun. 20 juin 2022

multi

Ce document détaille la proposition fonctionnelle et technique de développement d'un outil de test du standard d'API de covoiturage, pour en faciliter l'implémentation.

1 Objectifs fonctionnels

L'objectif est de développer un outil à l'usage de profils *développeurs* pour les aider à implémenter une API qui respecte le standard sur les données de covoiturage.

L'outil implémentera une API simulée pour chacune des deux parties du standard : ou bien une plateforme de MaaS ou bien un opérateur de covoiturage. Cela permettra d'expérimenter, pour chacun de ces deux acteurs, des scénarios métiers de test, sous forme de séquences d'appels d'API croisés.

Les appels et les réponses des appels d'API seront soumis à une série de tests qui vérifieront en détail la validité des appels et des réponses aux APIs vis-à-vis du standard. Ainsi, les scénarios métier exécutés de manière automatisée pourront servir de tests fonctionnels systématiques.

L'information retournée par ces tests doit faire l'objet d'un soin particulier, pour qu'elle puisse permettre d'identifier rapidement et précisément quel est le non-respect de la spécification, et comment la corriger.

Enfin, même si une liste de scénarios métier sera définie en amont du développement de cet outil, il sera utile de pouvoir faire évoluer ces scénarios et les données de test aisément pour que chacun puisse créer au besoin des tests sur mesure.

2 Proposition d'outil

2.1 Simulation d'un opérateur de covoiturage ou d'une plateforme de MaaS

L'outil proposé est un outil *en ligne de commande* à installer sur la machine de l'utilisateur. Il s'agit d'un serveur local qui simule les APIs de la plateforme de MaaS et de l'opérateur de covoiturage.

L'utilisateur qui dispose d'une API à tester peut ainsi interagir avec une implémentation de référence pour simuler sa contrepartie.

2.2 Deux modes d'interaction avec l'outil : bac à sable ou tests fonctionnels

Cet outil peut tourner de deux manières différentes :

- En mode **bac à sable** : l'utilisateur peut interagir à sa guise avec le serveur, et exécuter les requêtes d'API qu'il souhaite (dans les deux sens). Les requêtes et les réponses sont passées au crible de tests unitaires pour faire un retour sur des écart éventuels avec la spécification du standard covoiturage.
- En mode **tests fonctionnels** : l'utilisateur peut exécuter des scénarios métiers de manière prévisible et d'identifier les déviations des réponses attendues de manière automatique, et obtenir un rapport de validation le cas échéant. Il est attendu que l'utilisateur utilise pour cela des données qui lui sont fournies avec les tests (et créées à cet effet), il lui incombe d'adapter les données aux spécificités de son implémentation.

2.3 Définition des scénarios de manière modulaire

Les scénarios ont vocation à pouvoir être définis et enrichis sans modifier le code, par exemple grâce à un fichier de configuration. Cela permet d'envisager que chacun développe ses propres scénarios à tester, et que la barrière à l'entrée pour enrichir les tests avec de nouveaux scénarios est la plus basse possible.

2.4 Autres détails techniques

L'outil sera développé en golang.

Le serveur local se présentera sous la forme d'un container docker qui exposera les API simulées et les fonctionnalités de test. Aucun outil graphique n'est prévu à ce stade.

L'outil sera soigneusement documenté pour une prise en main rapide et l'interface en ligne de commande conçue pour être d'utilisation intuitive.