

17/06/2021

# Framework

Agile – Git – Angular

GIMENEZ Tim – JEANTET Joey  
LP CASIR

## Table des matières

Contenu de l'application .....	1
Mon investissement .....	1
Choix techniques .....	2
Retour d'expérience sur le projet.....	2

## Contenu de l'application

TutorProject est une application qui consiste à offrir la possibilité à un utilisateur de lister puis de réserver des véhicules.

Ce projet ayant débuté tardivement pour plusieurs contraintes de rendu s'enchainant les uns par-dessus les autres, toutes les fonctionnalités n'ont malheureusement pu être effectuées. En effet, dans sa dernière version, celui-ci comporte :

- Une page d'accueil ;
- Une liste des véhicules côté utilisateur ;
- Une page de détails d'un véhicule ;
- Une page admin offrant la possibilité de lister les véhicules, d'en ajouter, d'en modifier ou d'en supprimer. (La gestion d'accès à un utilisateur admin uniquement n'a pas pu être réalisée) à l'url « /admin » ;
- Une authentification à l'url « /login ».

## Mon investissement

Parlons maintenant de l'investissement du travail que j'ai fourni pour ce projet. En effet, de part ma disponibilité, j'ai organisé un maximum de temps afin de comprendre au mieux les concepts d'agile tout en liant nos UserStory avec git. Ainsi, les parties sur lesquelles j'ai travaillé sont diverses et la liste est la suivante :

- Lister les véhicules pour la partie utilisateur non-admin ;
- Filtrer les véhicules par couleur ;
- Filtrer les véhicules par marque ;
- Lister les véhicules pour la partie admin ;
- Ajouter un véhicule ;
- Supprimer un véhicule ;
- Et actuellement, je travaille sur la modification d'un véhicule.

Pour chaque élément cité ci-dessus, une branche et une UserStory a été liée (sauf pour les filtres regroupant les couleurs et les marques). A chaque UserStory ont été associés des commits atomiques représentant au mieux diverses fonctions implémentées.

A une fonctionnalité, un test a essayé d'être réalisé, or, je ne sais pour quelle raison, celui-ci n'a pas voulu s'exécuter comme je le voulais. Ayant passé pas mal de temps à essayer de comprendre et de le faire fonctionner, je n'ai pu aller au bout de celui-ci.

Un KANBAN a également été réalisé et géré. Sur celui-ci j'ai essayé de placer au mieux les fonctionnalités attendues réalisables actuellement ainsi que d'y assigner à chacune une personne afin de pouvoir se répartir le travail.

## Choix techniques

Abordons les choix techniques apporté au projet. Celui-ci a nécessité la réalisation de divers composants avec, pour chacun une page d'associé :

- Admin → Administrateur ;
- Home → Accueil ;
- listVehicule → Liste des véhicules ;
- details-vehicule → Détails d'un véhicule ;

Sur ces composants, nous avons utilisé pour notre template, Angular Material qui nous permet de facilement générer du code html simple, propre et efficace en quelques lignes de code.

Afin de gérer la connexion et le traitement avec une base de données, nous avons utilisé une api Swagger. Nous sommes partis pour générer un service Angular par objet existant dans notre api. Ainsi, nous avons réalisé un service Vehicule permettant de récupérer :

- Une liste de véhicule ;
- Un véhicule ;
- D'ajouter un véhicule ;
- De modifier un véhicule ;
- De supprimer un véhicule.

Enfin, Une interface « Vehicule » a été utilisé afin de gérer les objets véhicules. Celle-ci comporte des informations d'un véhicule. Et deux méthodes nommé « » et « » ont été créé permettant de savoir si un véhicule comportait toutes les informations nécessaires à l'ajout ou la modification.

Parlons désormais de git. Pour le gitflow, nous avons adopté une structure comportant des noms de branche commençant par :

- Feature
- Fix

Et des commit indiquant la nature du commit :

- Feat/feature → nouvelle fonctionnalité
- Fix → Correction d'un bug
- Change/Update → Modification d'une fonctionnalité

## Retour d'expérience sur le projet

La découverte d'agile a été pour moi un véritable choc. En effet, n'ayant jamais pratiqué cette méthode, j'ai trouvé quelques difficultés à comprendre et m'adapter au concept. Cette difficulté était présente tout au long du module mais malgré tout, j'ai fini par maîtriser cette pratique. C'est en pratiquant que l'on apprend et l'autonomie a été le plus présente. Tous les tutos sur internet m'ont beaucoup appris, augmentant ainsi mes compétences. Je suis donc satisfait de ce que j'ai pu apprendre pendant le déroulement de ce projet.

Passant par la réalisation d'une StoryMap, de User Story, d'itération, de Milestone, d'issues et de kanban, l'agile est loin de m'avoir fait surprendre.

Pour ce projet, j'ai eu quelques difficultés à comprendre comment positionner les Milestone et d'indiquer combien de temps il fallait leurs attribuer. Même si la plupart du temps, les entreprises et autres développeurs réalisent des sprints d'une semaine, nous n'avons malheureusement pu faire uniquement des sprints de 3 ou 4 jours.

Pour chaque Milestone ont été attribué des issues. Dans notre projet, les commit atomiques réalisés sur une branche sont liés à une issue. Et chaque branche concerne une fonctionnalité. Les commit atomiques ont fait ressentir quelques difficultés. En effet, lorsqu'une grande fonction est réalisée, des petites fonctions permettant d'exécuter une fonctionnalité simple doivent être créés or j'ai trouvé des difficultés à trouver quelles parties séparer de la grande fonction.

Enfin, je trouve dommage d'avoir eu le temps d'attaquer ce projet que tardivement car il n'a malheureusement pu arriver à finalisation. Mais cela ne m'empêche d'être satisfait d'avoir compris les bases de la méthode agile et d'avoir pu découvrir un nouveau framework : Angular.