

My Booking Services

_

Méthodologie de Projet

-

SPE-CLO5



I. Gestion du temps, cycles de travail (Sprint)

A. Sprints d'une semaine et réunions hebdomadaires

Dès le début du projet, nous avons suivi un rythme de travail constitué de cycles d'une semaine (Sprint). Le début de tout Sprint est marqué d'une réunion entre tous les membres. Ces réunions sont définies par un Ordre Du Jour précis, et se voient attribuer une limite de temps.

Leur structure est toujours la même, à savoir :

- Présentation par chacun des actions entreprises lors du cycle de travail précédent
- Remontée au Groupe des problèmes rencontrés
- Recherche de résolution des problèmes éventuels, si le temps imparti le permet
- Mise en perspective de l'avancement du travail vis-à-vis du CDC (Sujet ETNA)
- Identification des priorités et des sujets à traiter lors du prochain Sprint
- Répartition des tâches entre les membres du Groupe
- Mise à jour du Board Kanban.

B. Quels outils?

- Hangout pour les réunions
- Google Docs pour les Compte-Rendus de Réunion si applicable
- Dashboard Kanban GitLab pour le suivi des tâches et leur répartition

II. Gestion des Ressources et méthodologie DevOps

A. Mobilisation des Ressources

Si la pondération des effectifs sur chaque tâche est dictée par des priorités, la répartition des Ressources sur les différentes tâches est faite en fonction des compétences, mais surtout des appétences.

Par exemple, si certains membres souhaitent monter en compétence sur le volet Ansible du projet, nous mettons en place un roulement sur la réalisation des tâches, ou une division de celles-ci (un Rôle Ansible par membre intéressé...).

Pour toute tâche liée à l'infrastructure Système, un tandem Développeur/Ops est créé.



La résolution des problèmes majeurs mobilise l'ensemble du Groupe, qu'ils soient liés au Développement ou à la partie Système/Opérations.

B. Quels outils?

Dashboard Kanban GitLab pour le suivi des tâches et leur répartition.

III. Méthodes de Développement

A. Méthode, documentation et templating

Nous avons mis en place du Développement en TDD (Test-Driven Development), où les spécifications des Services sont écrites avant l'implémentation.

Grâce à la mise en place rapide de documentations Swagger autogénérées, nous avons pu facilement normaliser les interfaces des Services et uniformiser leur développement.

Un template de projet NestJs nous a par ailleurs permis de faciliter l'initialisation des Services, et de les uniformiser.

B. Quels outils?

- Swagger pour la documentation
- Repository GitLab de template

IV. Intégration / Déploiement continus

A. Travail en CI / CD

Nous avons mis en place des pipelines d'intégration continue des sources du projet, ce afin de garantir la qualité de la base de code.

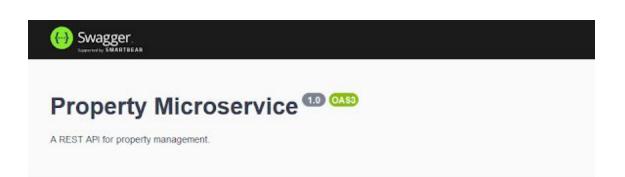
Le déploiement continu de la solution applicative et des services associés (reverse proxy...) est également intégré à nos pipelines.

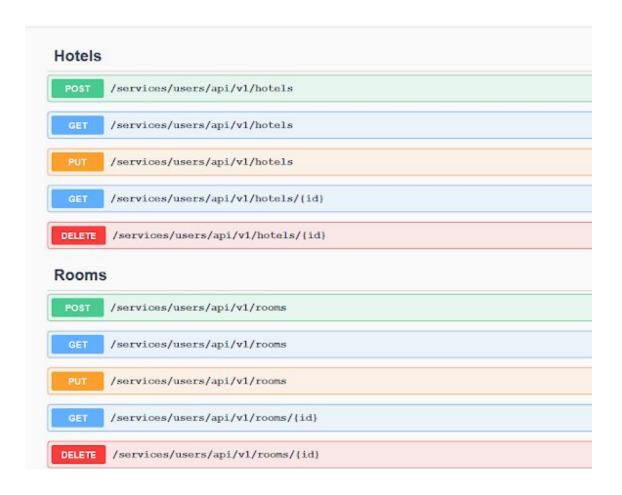
B. Quels outils?

- Pipelines GitLab CI/CD, Runner + Docker executor.

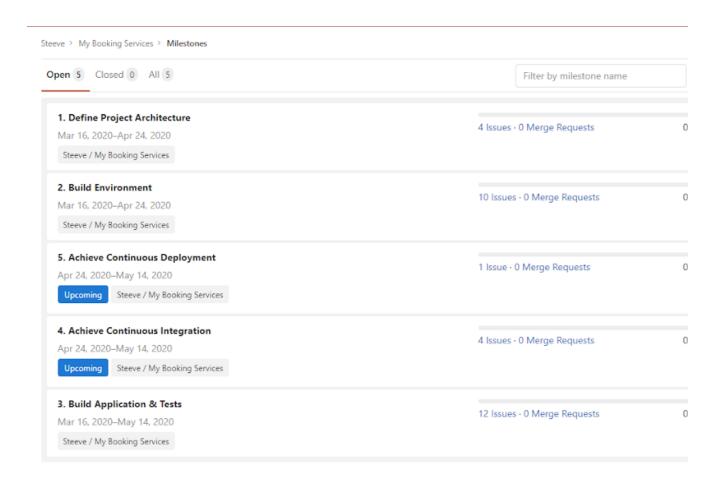


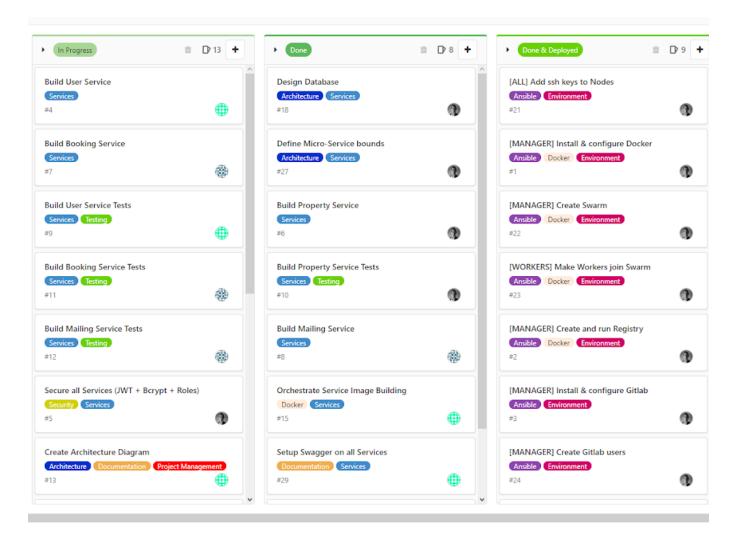
V. Annexe











≝etna

