

# Intégration continue pour le web

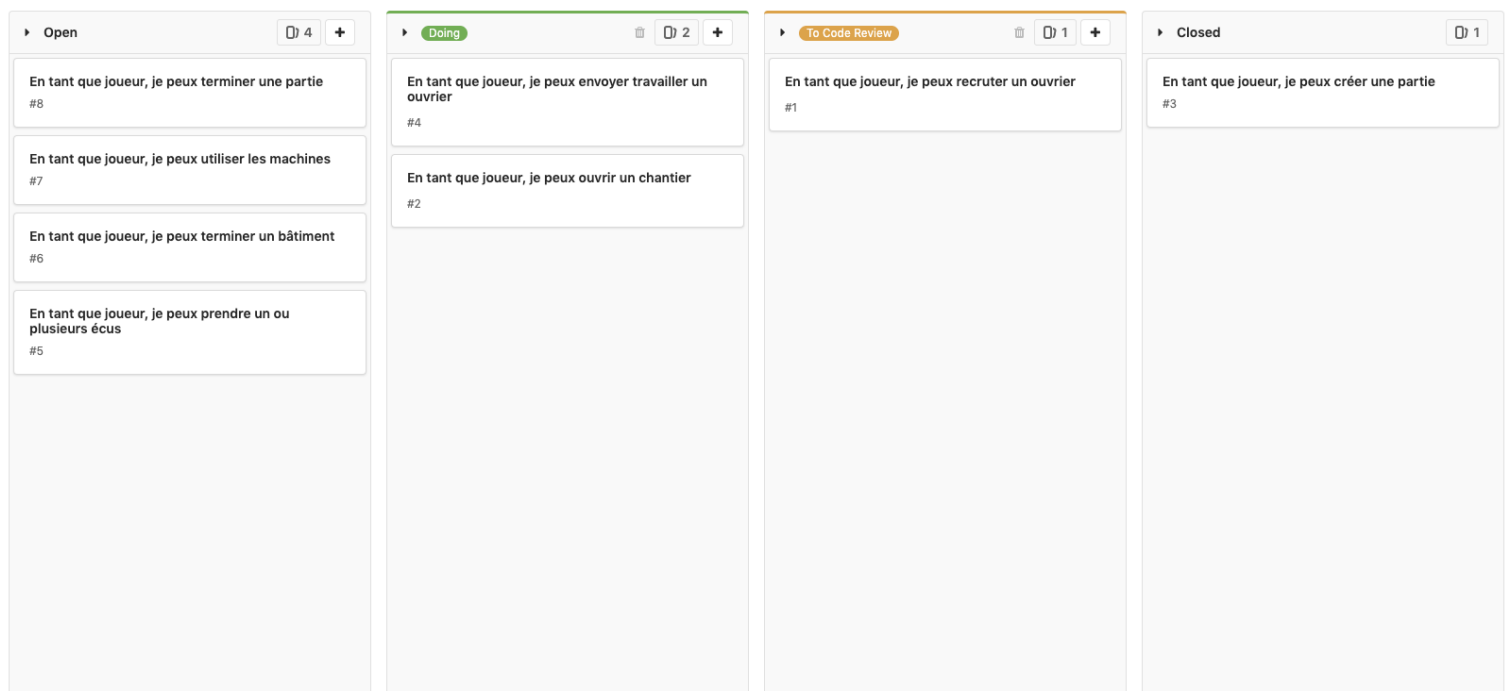
## TP final Les bâtisseurs : Moyen-Âge


### Mise en place du projet

Formez une équipe de trois personnes et choisissez un des projets. Invitez les deux personnes manquantes sur le projet et n'oubliez pas de m'inviter ([@JulienUsson](#)) pour que je puisse vous noter 😊 . Rajoutez un fichier `authors.md` avec votre pseudo suivi de votre nom afin que je sache quel pseudo correspond à quelle personne. Vous êtes prêts à coder 🎉 .

### Gestion de projet

Pour simplifier votre organisation, vous pouvez vous servir de la vue *board* de la section *issues* de Gitlab. En organisant votre *board* comme ci-dessous, vous pourrez savoir qui fait quoi, qu'elle est la prochaine tâche à faire ainsi que les *merge-requests* en attente.



 N'hésitez pas à redécouper les tâches si celles-ci vous paraissent trop longues ou complexes.

⚠ N'oubliez pas de vous assigner aux tâches et de bien mettre à jour le *board* pour profiter au maximum de l'outil.

## Notation

Le tp sera noté selon les critères suivants :

- Utilisation de Git (commits, branches, demande de merge, revue de code, ...)
- Tests (qualité des tests, couverture de code, ...)
- Code (qualité du code, fonctionnalités implémentées, ...)
- Gestion de projet (découpe en ticket, priorisation, partage des tâches, ...)

## Frontend

Une application frontend est disponible sur [battisseurs.usson.me](https://battisseurs.usson.me).

Par défaut, l'application utilise un backend se trouvant à l'adresse `http://localhost:3000`. Si ce n'est pas votre cas, vous pouvez le configurer via l'icône ⚙ en haut à droite.

## Rappels

Une documentation détaillée de l'ensemble des routes attendues est disponible sur <http://localhost:3000/api-docs/>. Il est **important** de suivre cette documentation lors du développement du projet sinon le Front-end ne marchera pas.

Pour tester vos routes, il est **recommandé** d'utiliser le logiciel [Postman](#) [\[aide\]](#).

Les règles du jeu sont disponibles [ici](#) ou [ici](#) en vidéo.

Google est votre ami.

## Tâches

## En tant que joueur, je peux créer une partie

- Création de la route `POST /games` [\[doc\]](#) pour créer une partie.
- Sauvegarder les parties dans un/des fichier(s) dans le répertoire `/storage` afin de conserver les parties entre deux redémarrages.

 [uuidv4\(\)](#) permet de générer un identifiant aléatoire.

 [shuffle\(\)](#) permet de mélanger un tableau.

 [fs.writeFile\(\)](#) permet d'écrire dans un fichier.

 [JSON.stringify\(\)](#) permet de convertir un objet Javascript en string.

 [JSON.parse\(\)](#) permet de convertir une string en objet Javascript.

## En tant que joueur, je peux voir les détails d'une partie

- Création de la route `GET /games/{gameId}` [\[doc\]](#) pour récupérer les informations d'une partie.

## En tant que joueur, je peux lister les parties existante

- Création de la route `GET /games` [\[doc\]](#) pour lister les partie.

## En tant que joueur, je peux ouvrir un chantier

- Création de la route `POST /games/{gameId}/actions` [\[doc\]](#).

## En tant que joueur, je peux recruter un ouvrier

- Amélioration de la route `POST /games/{gameId}/actions` [\[doc\]](#).

## En tant que joueur, je peux envoyer travailler un ouvrier


- Amélioration de la route POST `/games/{gameId}/actions` [\[doc\]](#).

## En tant que joueur, je peux prendre un ou plusieurs écus

- Amélioration de la route POST `/games/{gameId}/actions` [\[doc\]](#).

## En tant que joueur, je peux terminer un bâtiment

## En tant que joueur, je peux utiliser les machines

 Une des possibilité est qu'une fois un bâtiment terminé, on peut créer un nouveau ouvrier correspondant aux caractéristiques de la machine.

## En tant que joueur, je peux terminer une partie