

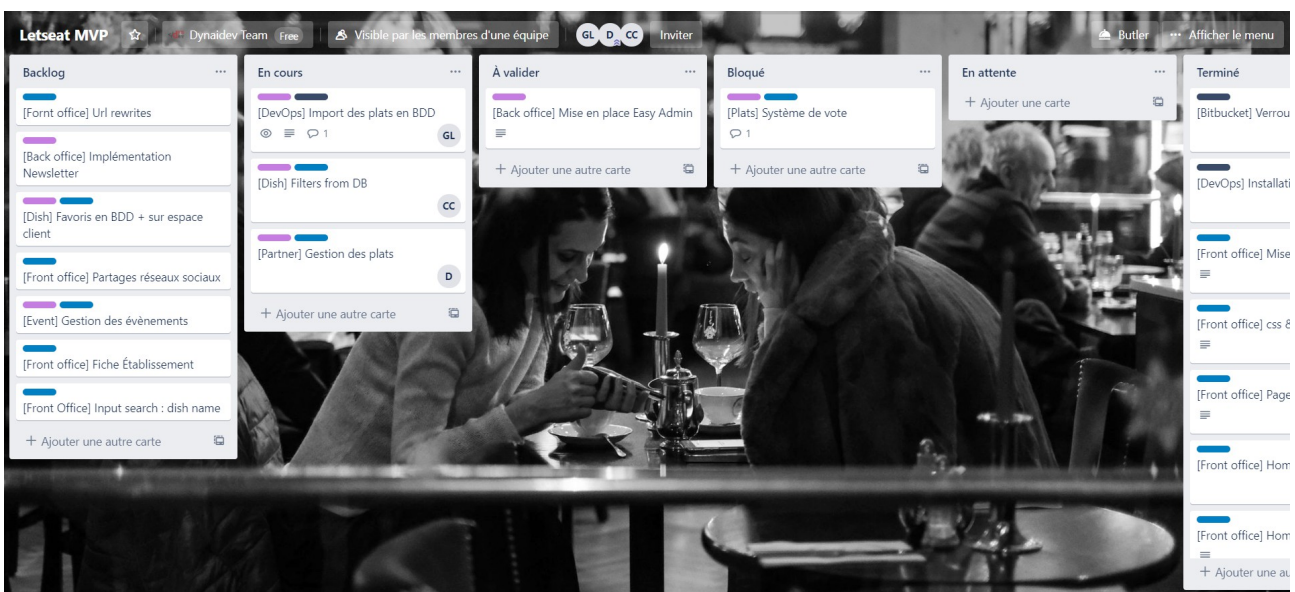
## Présentation:

Le projet est la création d'une solution permettant de visualiser et commander les plats proposés à proximité du client par les restaurateurs de tout type (traditionnels, rapides, food truck ou encore boulangerie).

L'idée étant de donner plus de visibilité aux restaurateurs en répondant aux nouvelles demandes d'immédiateté du marché et de facilité de choix apporté par le visuel.

## Trello:

Trello est un outil de gestion de projet en ligne, lancé en septembre 2011 et inspiré par la méthode Kanban de Toyota. Il repose sur une organisation des projets en planches listant des cartes, chacune représentant des tâches. Les cartes sont assignables à des utilisateurs et sont mobiles d'une planche à l'autre, traduisant leur avancement.



Voici le tableau trello, quand vous souhaitez effectuer une tâche, choisissez la dans "Backlog" et mettez la dans "En cours" ( Nb: penser à vous assigner à la tâche , voir image suivante )



Quand vous aurez terminé votre tâche , passer la dans "à valider" le temps que les autres développeurs valident votre travail.

## **GIT:**

GitHub est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git.

### **Une liste commande Git:**

git clone:

Permet de récupérer un fichier qui se situe sur un dépôt git ( pour nous, il se situe sur bitbucket ).

git pull:

Permet de récupérer les MAJ qui ont été effectuées sur le projet ( cette commande est à effectuer tous les matins et midi à chaque fois que vous changez de branche ) .

git status:

Permet de voir les fichiers que vous avez modifier par rapport au dernier commit sur cette branche.

git add . :

On prépare les fichiers modifier à être envoyé.

git commit -m "message du commit" :

Permet de lier un msg au travail effectué.

git push :

Envoie le contenu du travail ainsi que le msg lié.

git checkout "nom de la branche" :

Permet de changer de branche

git checkout -b "nom de la branche" :

Permet de créer une branche afin d'effectuer son travail (toujours se mettre sur la branche la plus à jour pour créer sa branche).

Pour le nom des branches généralement on utilise 3 étapes :

1re - feature ou fix

2e - front , back ou all

3e - nom de la tâche

Exemple: feature/back/newletters ainsi on comprend que la tâche permet de rajouter une fonctionnalité, avec du code majoritairement back pour les newsletters.

git branch -d

Supprime une branche en local.

git merge


En cas de conflit cela permet de réunir le code des 2 branches et de voir quel code il faut garder.


Dès que votre travail est push et est fini, vous devez aller sur bitbucket afin de faire une pull request afin que vos collègues puissent vérifier votre travail (Nb: Pensez à vérifier la branche vers laquelle vous demandez la PR ).


## feature/geo

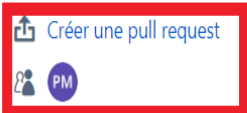
[Consulter ▾](#) [Voir la source](#) [Merge](#) [...](#)

---



 dev [changer la destination](#)

 feature/geo



[Différences](#) [Commits](#) [Pull requests mergées](#)

---

Auteur	Committer	Message	Date	Builds
--------	-----------	---------	------	--------

en cas de conflit effectuer ceci :

- git checkout dev
- git pull
- git checkout ta branche
- git merge dev
- régler les conflits sur phpstorm ( conflit affiché et visible)
- puis push avec en commit "fix/conflits"

## Symfony:

Symfony est un ensemble de composants PHP ainsi qu'un framework MVC libre écrit en PHP. Il fournit des fonctionnalités modulables et adaptables qui permettent de faciliter et d'accélérer le développement d'un site web.

### Liste de commandes utiles sur symfony :

`composer install` :

Permet d'installer toutes les dépendances nécessaires. ( à effectuer à chaque fois que vous changez de branche/git pull )

`symfony server:start` :

Lance un serveur en local afin d'afficher le site (dès que vous fermez votre IDE cela stopera aussi le serveur )

`php bin/console doctrine:database:drop -f` :

Supprime la bdd actuel.

`php bin/console doctrine:database:create` :

Créer une nouvelle bdd

`php bin/console doctrine:migrations:migrate` :

Créer les tables de la bdd

`php bin/console doctrine:fixtures:load` :

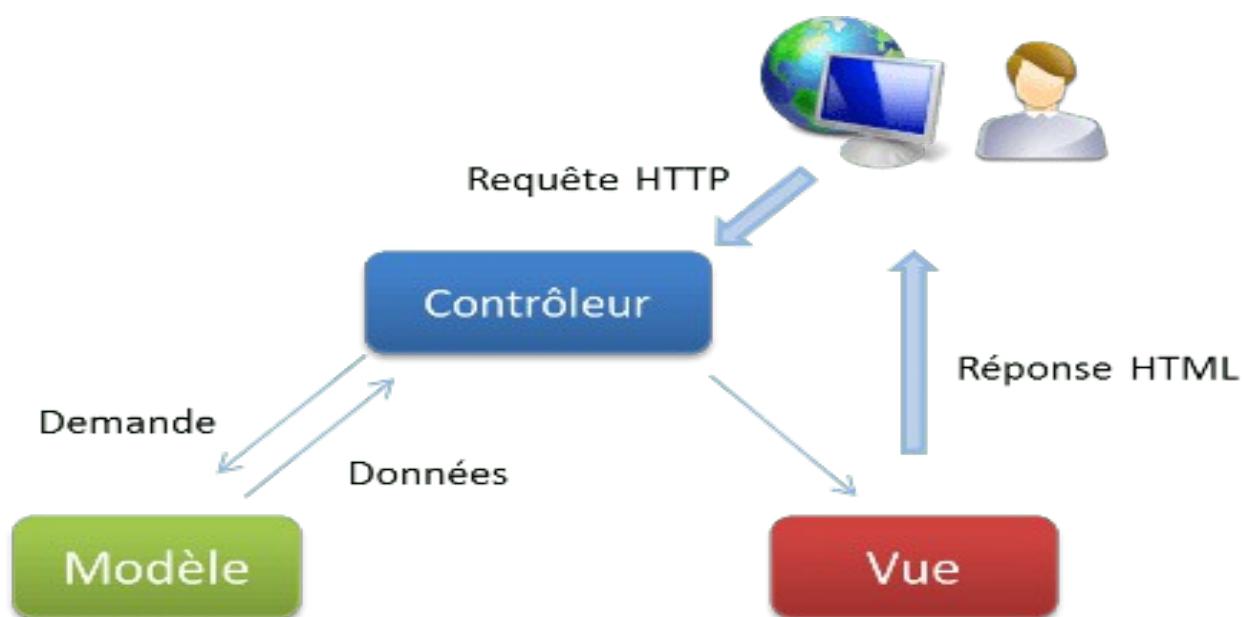
Créer le contenu des tables s'il y en a de prévus

`php bin/console cache:clear` :

Vide le cache

En cas de problème pensez à aller voir la doc "<https://symfony.com/doc/current/index.html>".

### Présentation MVC



## PHPStorm:

PhpStorm est un éditeur pour PHP , HTML, CSS et JavaScript, édité par JetBrains.  
Le prendre 1 à 2 jours avant le début du stage pour profiter des 30 jours d'essais.

Afin de travailler sur le projet il faut :

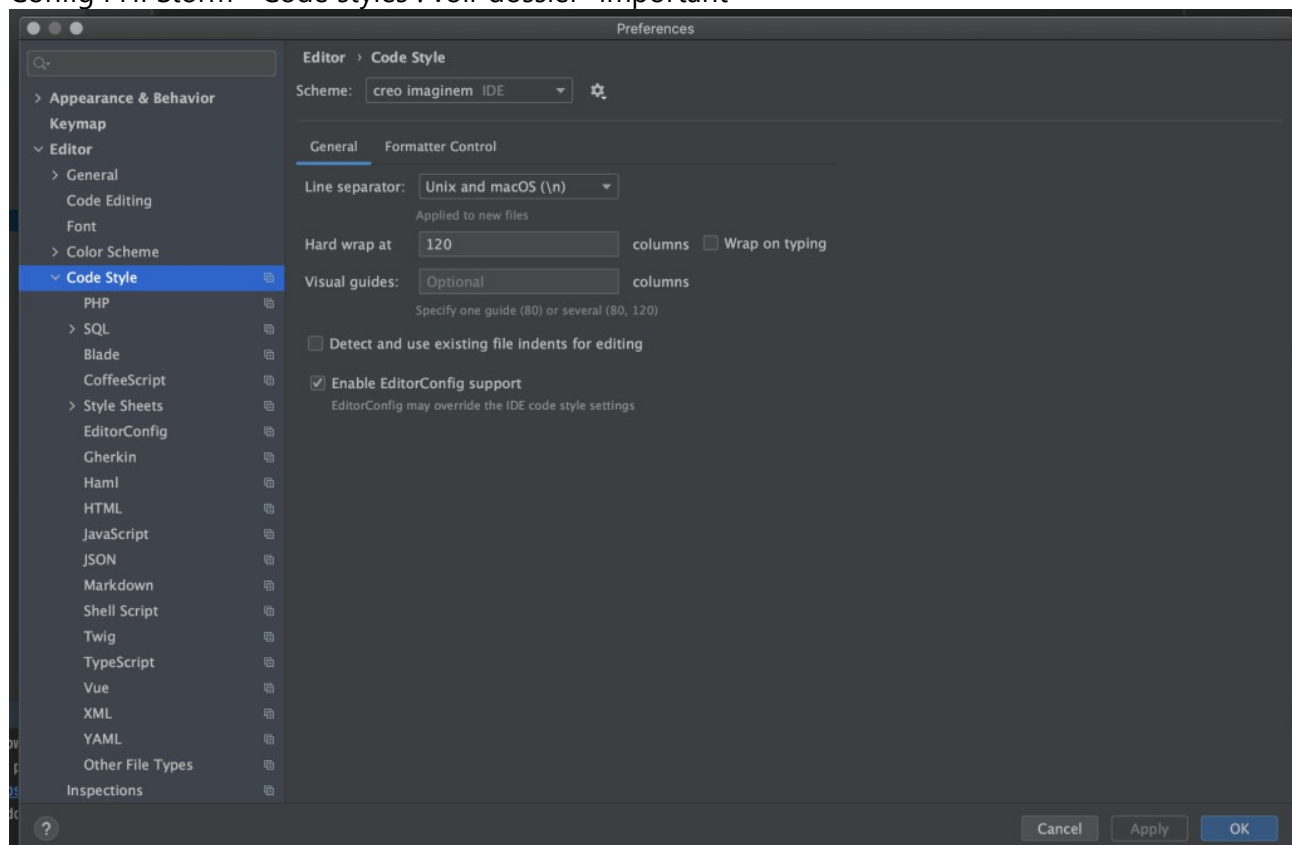
-SASS

<https://www.jetbrains.com/help/phpstorm/transpiling-sass-less-and-scss-to-css.html>

-Babel-minify :

<https://blog.jetbrains.com/webstorm/2015/05/ecmascript-6-in-webstorm-transpiling/>

Config PhpStorm - Code styles : voir dossier "important"



**Comment débiter avec le projet:**

- ouvrir son PHPStorm
- ce placé dans le bon dossier (exemple : C:\wamp64\www )
- ouvrir un terminal
  - git clone "LienTrouvableSurBitbucket"
  - composer install
- Faire un .env.local
- Retour sur le terminal
  - php bin/console doctrine:database:create
  - php bin/console doctrine:migrations:migrate
  - php bin/console doctrine:fixtures:load
  - php bin/console cache:clear
  - symfony server:start