

## STAGE 2ème année de Célia Baylet :

### PIXINE

**1ère semaine du 13 au 17 janvier 2025**

13/01/2025

-Aujourd'hui nous avons commencé par la préparation de l'environnement. Ils travaillent sous un environnement Linux.

Ils utilisent aussi slack comme réseau d'entreprise, similaire à discord pour communiquer entre eux.

Les développeurs sont répartis sur plusieurs projets. Moi je vais travailler surtout avec Julien et Maximilien qui travaillent sur la plateforme CNFCE. C'est un organisme de formation professionnelle pour les entreprises. Nous utilisons visual studio code et le framework symfony pour le site de la CNFCE. Il est fait avec un word press.

-J'ai commencé par lire la charte informatique ainsi que le MOOC numéro 1 de l'ANSSI, qui est composé de 5 unités avec des cours et une évaluation (<https://secnumacademie.gouv.fr/>). Dans la charte, il est demandé à tout utilisateur de l'entreprise de faire le premier module du MOOC 'Panorama de la SSI'.

-Maximilien a passé beaucoup de temps à me faire la présentation du projet. Il m'a expliqué qu'ils travaillaient sous forme de tickets.

-Ils utilisent bitbucket et Jira. Jira est une solution pour gérer les tickets par exemple ( A faire – en cours – pull requests...). Bitbucket, c'est une solution git pour faire des pull requests.

Ils utilisent aussi sublime merge qui est une solution pour faire des commits mais sans ligne de commande, c'est visuel. Ce logiciel gère la résolution des conflits (si plusieurs développeurs avaient travaillé sur le même ticket il ne faut pas que le travail qui arrive en 2eme écrase l'autre, il faut les fusionner) et la fusion des branches (si plusieurs commits ont été fait il est possible de les regrouper).

Lors d'un projet, on fait d'abord un commit avec sublime merge et ensuite une pull request sur bitbucket une fois que le ticket est fini en attendant les commentaires des collègues.

14/01/2025

-Aujourd'hui j'ai avancé le MOOC de l'ANSSI.

-Un ticket : un mail à modifier avec des {variable} donc modifier la vue et le contrôleur en conséquence.

-Maximilien m'a expliqué la gestion des tickets et comment je devais procéder. D'abord mon code est stocké localement ensuite je l'ai poussé vers le cloud avec un 'push' pour que les collègues puissent éventuellement faire des corrections avant que ce ne soit poussé sur le staging l'étape juste avant le développement final.

-Avancer le MOOC de l'ANSSI.

-Comme tous les mardis, une réunion entre développeurs permet de faire le point, j'y ai donc assisté.

15/01/2025

-Comme hier mise en place des mails, en cas de désactivation d'un bouton l'utilisateur reçoit alors un mail. J'ai donc modifié le contrôleur et ajouté la vue.

-J'ai fini le premier MOOC de l'ANSSI.

-Julien m'a aidé sur un point précis du code. Même si les collègues sont en télétravail on peut s'appeler en partage d'écran et faire des points ensemble.

16/01/2025

-A la fin de mon premier ticket, j'avais fait une pull request pour que mes collègues puissent vérifier les modifications et éventuellement faire des commentaires. Julien m'avait demandé une modification donc j'ai repris mon ticket ensuite j'ai fait le 2e ticket.

-Le ticket que j'ai commencé, consistait à rajouter un bloc-notes nommé 'note interne'.

-Cet après-midi j'ai fait des recherches sur la migration en Symfony. J'ai aussi regardé comment le code était structuré pour le prochain ticket.

17/01/2025

-Avec l'aide de Maximilien j'ai installé SQLTools qui sert à voir les éléments de la base de données. J'ai aussi installé postgres pour faire les migrations PHP que j'ai fait en ligne de commande.

-On travaille sur du docker et Symfony on utilise la version 5.0

-J'ai fait un peu de visuel de CSS en modifiant des 'card', 'card-body' pour le ticket bloc-notes. J'ai ensuite fait la préparation pour le nouveau ticket, on doit rajouter un onglet entre 2 autres onglets. Il y a par exemple 'à traiter', 'planifiées' et 'fermer' et je dois rajouter 'à reporter' donc j'ai observé le code pour voir comment il était construit.

Vocabulaire de cette semaine :

WYSIWYG :



The screenshot shows a dark-themed interface for a text editor. At the top, it says "Champ d'éditeur de texte EasyAdmin". Below that is a link to "Modifier cette page". A description states: "Ce champ crée un éditeur WYSIWYG basé sur JavaScript basé sur l'éditeur Trix pour gérer les textes longs." It then says: "Dans [les pages de formulaire \(édition et nouvelle\)](#), cela ressemble à ceci :". Below this is a preview of the text editor field, titled "Text Editor Field". The preview shows a rich text editor with a toolbar containing icons for bold, italic, link, unlink, text color, background color, bulleted list, numbered list, indent, and outdent. The text area contains a paragraph of Lorem Ipsum: "Lorem ipsum **dolor sit amet**, consectetur adipisicing elit, sed do eiusmod tempor incididunt *ut labore et dolore* magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur."

staged files : ce sont les fichiers qui vont être commit

par contre les fichiers untracked ne seront pas dans le commit

commande make env-start-worker : pour démarrer l'environnement

commande make env-stop-worker : pour arrêter l'environnement

La commande make env-start-worker démarre un processus de worker (souvent pour traiter des tâches en file d'attente) dans un environnement configuré, en utilisant des outils comme Docker, ou des scripts spécifiques. Elle est définie dans le **Makefile** du projet.

Ici on utilise bien docker. A chaque modification de la vue il est important de relancer cette commande.

make phpcsfixer && make phpstan

Cette commande était effectuée avant de commit car :

- **make phpcsfixer** : Exécute PHP CS Fixer pour corriger automatiquement les problèmes de style de code selon des règles définies.

- **make phpstan** : Lance PHPStan pour analyser statiquement le code PHP et détecter des bugs ou des problèmes potentiels.

Subscriber : Un fichier subscriber est généralement utilisé pour définir un composant ou une classe qui s'abonne à des événements, des notifications ou des messages. Il permet d'écouter et de réagir à des changements ou des actions spécifiques.

Event : Un fichier event sert généralement à définir des événements, c'est-à-dire des actions ou des changements spécifiques qui peuvent être émis et écoutés dans un système. Il décrit souvent la structure ou le type des événements.

git rebase -i HEAD~x (remplacer x par le nombre de commit à fusionner)

puis ensuite c'est sous le format :

pick 7418529637852 nom

pick 6845746586487 nom du deuxième commit

Donc on supprime le deuxième pick pour garder seulement le nom voulu.

localhost:5000 : permet de vérifier si nous avons bien reçu un mail quand on travaille sur les mails

dd("stop") cette commande permet de vérifier que notre code passe bien par la fonction

'Ctr + p' sur visual pour trouver plus vite un fichier

sublime merge : 'stageAll' puis faire un 'push -force' (en cliquant à côté de la flèche vers le haut) pour faire un commit.