

Prépa à l'ENVOL : atelier du 13 octobre après-midi

« L'Intelligence Artificielle en aide au développement »

Après avoir discuté avec les organisateurs, nous souhaitons considérer une présentation réalisée en avril dernier ([vidéo IA4Dev](#)) comme un prérequis et socle commun. En gros, si vous savez ce que sont les modes « ask / edit / agent » et vous voyez un peu près ce qu'est un « MCP », vous êtes bon. Sinon, vous pouvez soit regarder la vidéo, soit faire quelques petites recherches web pour vous mettre à jour.

Les objectifs de l'atelier du 13 octobre seront une mise à jour (plusieurs choses ont évolué depuis cette présentation IA4Dev), un approfondissement sur certains points et une mise en pratique.

Nous ne reviendrons pas sur le fonctionnement vulgarisé des grands modèles de langue. Une présentation rapide peut être regardée ici (je ne suis pas fan de la forme mais ça a le mérite d'être une présentation rapide et efficace) : <https://www.youtube.com/watch?v=1RRHr3dFogQ>

Pour la partie pratique, nous avons besoin que vous installiez des outils et fassiez quelques démarches en amont : **Attention, si vous faites ça la veille, il y a un risque que cela ne marche pas car il y a validation par Github de votre compte « éducation » qui n'est pas immédiate.**

Une question légitime que certains d'entre vous vont peut-être se poser : pourquoi proposer dans cet atelier d'utiliser « Vscodé + Github copilot » alors qu'il existe d'autres outils ?

Je présenterai aussi Continue.dev avec un endpoint de l'ESR mais l'offre « éducation » de github permet d'accéder gratuitement à des modèles de qualité avec une infrastructure qui marche très bien. N'étant pas certain que les autres solutions gratuites puissent donner des réponses satisfaisantes (notamment en termes de temps de réponse avec une quinzaine de demandes en simultané puisque nous travaillerons en binôme), une grande partie de l'atelier sera effectuée avec « Vscodé + Github copilot ».

Nous utiliserons aussi « Spec Kit » pour aborder ce qu'est le développement « guidé par les spécifications ». De même, je pourrais vous fournir des alternatives possibles dans l'atelier.

Ci-dessous, vous trouvez les étapes pour installer « VsCode + GitHub Copilot », puis celles pour « Spec Kit ». À la fin, j'ai mis mon contact mail si jamais vous rencontrez des soucis.

Étapes pour VsCode + Github Copilot à réaliser :

- 1) Création d'un compte copilot (si vous n'en avez pas)

Une partie qui peut ne pas être évidente est la configuration de l'authentification à 2 facteurs (<https://docs.github.com/fr/authentication/securing-your-account-with-two-factor-authentication-2fa/configuring-two-factor-authentication>). De mon côté, j'utilise le gestionnaire de mot de passe KeePassXC pour générer le TOTP. D'autres solutions sont bien entendu possibles comme le mentionne l'url précédente.

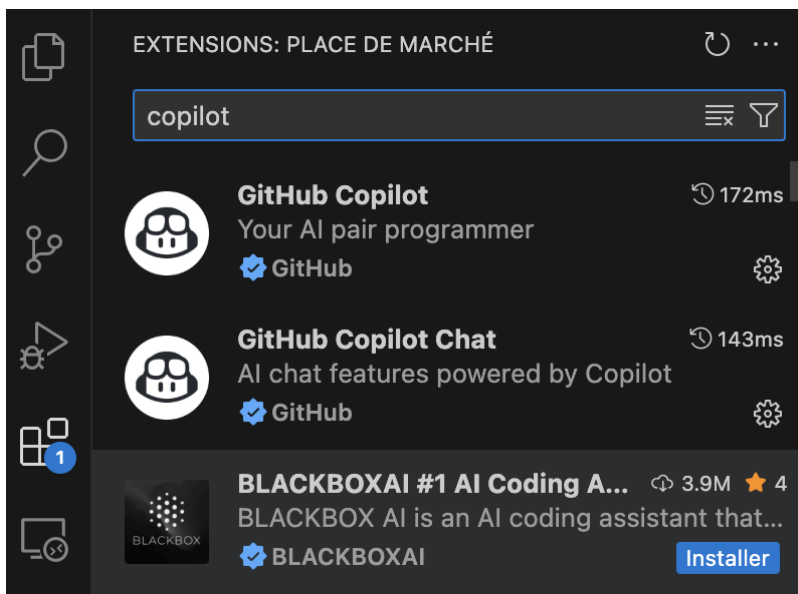
- 2) S'inscrire sur <https://docs.github.com/fr/education/about-github-education/github-education-for-teachers/apply-to-github-education-as-a-teacher>

- 3) Installer Visual Studio Code : <https://code.visualstudio.com/>

Si vous l'avez déjà installé depuis un petit moment, faire une mise à jour : Menu « Help » dans Visual Studio Code et « check for updates »

Si vous n'avez jamais utilisé Vscode, vous pouvez essayer de créer un petit projet « Hello World » et en profiter pour installer une extension liée à votre ou vos langage(s) de prédilection

- 4) Dans Visual Studio Code, cliquer sur l'icône des extensions dans le menu latéral gauche (sur l'image, c'est la petite icône où il y a un 1 dessus mais il n'y a pas toujours de chiffre dessus...), puis rechercher GitHub Copilot dans la barre de recherche.



Cliquer sur l'extension GitHub Copilot, puis sur Installer. Une fois l'installation, l'icône devrait apparaître à droite du menu de recherche.



En cliquant dessus, Visual Studio Code devrait nous demander de vous authentifier à GitHub Copilot par l'intermédiaire du navigateur Web. Identifiez-vous.

- 5) Si vous êtes sur Windows, installer WSL 2 (cela vous servira pour la suite)

Ouvrez PowerShell en mode **Administrateur** en cliquant avec le bouton droit et en sélectionnant « Exécuter en tant qu'administrateur », entrez la commande `wsl --install`, puis redémarrez votre ordinateur.

Vérifier votre installation de WSL en recherchant WSL dans votre barre windows. Normalement, une petite icône à tête de pingouin apparaît

Installer dans Vscode, l'extension pour WSL :

<https://marketplace.visualstudio.com/items?itemName=ms-vscode-remote.remote-wsl>

Étapes pour installer Spec Kit :

Attention, ça ne s'installe pas sous Windows mais si vous avez fait les étapes précédentes, vous devez avoir WSL 2.

En amont, vérifiez que vous avez bien git et uv

Pour uv, la commande pour MacOS, Linux ou WSL2 est « `curl -Lsf https://astral.sh/uv/install.sh | sh` »

Ensuite pour Spec Kit, tout est expliqué ici : <https://github.com/github/spec-kit>

Arrêtez-vous à la première étape :

`uv tool install specify-cli --from git+https://github.com/github/spec-kit.git`

car nous allons voir la suite ensemble 😊

Si vous rencontrez des soucis, n'hésitez pas à m'envoyer un mail : max.beligne@univ-grenoble-alpes.fr