

[Core] test technique recrutement (techno libre back ou front)

Développement Back [↗](#)

Stack [↗](#)

- Un projet avec un **langage objet backend** (go, c++, java, python, ruby, php...) , avec une BDD en **Postgres**.
- Tu peux utiliser les librairies supplémentaires de ton choix si tu en sens l'utilité.

Specs [↗](#)

- Un endpoint d'**API REST** en JSON pour s'inscrire
- L'utilisateur envoie un pseudo de **3 lettres exactement** (A à Z, tout en majuscule. Pas de chiffre), ni plus, ni moins pour l'instant.
- Si le pseudo
 - **est** libre, on l'enregistre, et on répond à l'utilisateur avec le pseudo qu'il vient d'enregistrer.
 - **n'est pas** libre
 - on en trouve un libre, avec les mêmes règles que précédemment (3 lettres exactement, A à Z, tout en majuscule, pas de chiffre)
 - on l'enregistre,
 - on renvoie à l'utilisateur ce pseudo en question.
 - RQ: Le pseudo libre que l'app choisit pour l'utilisateur dans ce cas n'a pas besoin de ressembler de près ou de loin au pseudo choisi par l'utilisateur.

Rendu [↗](#)

- Un repo sur **Github**, privé
- Une pull request de la branche intitulée " dev /xxxx" ou " feature /xxxx" vers la branche `main`
 - Nous aidera à plus facilement faire des retours rapides
- Tous les candidats Back passent le même test technique (de junior à senior). La différence se fera sur les skills démontrés (réponse au sujet, architecture, qualité du code, performance de la solution choisie, commits, reflexes techniques, tests/specs, readme, évolutivité ...)
- Ne t'embête pas avec des features qui ne sont pas demandées (authentification, etc.), va à l'essentiel. En revanche n'hésites pas à documenter les étapes qui pourraient être faites par la suite "*si besoin*" ("*Il serait intéressant de ...*")
- Ce que l'on va regarder :
 - La facilité qu'on a à rentrer dans ton projet (naming, architecture, historique git)
 - Ton découpage du problème
 - Les bonnes pratiques que tu auras mises en place
 - Si ça marche, bien sûr
 - Et **surtout**, l'efficacité de ta solution, notamment la manière dont tu vas trouver un pseudo disponible quand celui que demande l'utilisateur est déjà pris
 - La capacité à faire évoluer le code si les demandes évoluent...

Next steps [↗](#)

- On debrief de ta solution, en terme de style, d'architecture, de bonnes pratiques et d'efficacité.
- Si on pense que tu es passé à coté de quelque chose d'important, on te donne des pistes pour que tu puisses éventuellement faire une seconde passe.