

# Les Echos

## Test Technique pour Développeur Senior/Lead en Nest.js

Ce test technique vise à évaluer vos compétences en développement backend avec Nest.js. Il est conçu pour mesurer non seulement vos capacités techniques mais également votre approche en termes de bonnes pratiques, de sécurité et de qualité du code.

### Contraintes

Développer un backend en Nest.js qui répond aux objectifs suivants :

- **Inscription** : Les données suivantes sont renseignables :
  - pseudonyme, unique
  - mot de passe, sécurisé.
  - name, string, facultatif
  - Adresse postale, structure libre, facultative.
  - Commentaire, string, facultatif
  - Type d'utilisateur : admin (tous les droits) ou utilisateur (droits limités à son propre profil).
- **Authentification** : Implémenter les fonctionnalités de connexion et de déconnexion
- **Gestion des utilisateurs** :
  - Permettre la récupération et la modification des informations de l'utilisateur courant.
  - Autoriser la modification et la suppression des informations des autres utilisateurs par les administrateurs.
- **Liste des utilisateurs** :
  - Uniquement accessible par les utilisateurs admin, avec options de tri et de filtrage sur tous les champs.

## Livrables attendus :

Fournissez un lien vers un répertoire Git contenant :

- Un fichier **README**: instructions d'installation et de désinstallation.
- Un fichier **TODO**: Éléments manquants, bonnes pratiques non appliquées, améliorations d'architecture et de conception envisagées.
- Un fichier **Test-todo**: Stratégie de tests envisagée, sa mise en œuvre et son ordre d'exécution.
- Un fichier **auteur** à la racine contenant l'initiale de votre prénom suivie de votre nom sans espace.
- Le code en Typescript.

## Contraintes :

- Sécurité : Assurez une validation et sécurisation rigoureuses de toutes les entrées utilisateurs.
- Propreté de l'Installation : L'installation locale ne doit laisser aucune trace durable.
- Documentation de l'API : Inclure dans le README comment consulter la documentation API avec les informations minimales pour son usage.
- Choix de l'API : REST ou GraphQL, en respectant les bonnes pratiques.
- Libertés : Vous êtes libre de choisir vos packages (licence non contaminante) et votre méthode de stockage des données (SQL ou NoSQL).
- Code : Le respect des bonnes pratiques de programmation en typescript / js et les recommandations de Nest.js.
- La documentation doit être exprimée en Français.

Ce test est destiné à un profil senior/lead, avec une liberté volontaire dans l'approche.

L'objectif est d'évaluer votre connaissance et application des bonnes pratiques, tout en privilégiant la fiabilité et la sécurité de votre code et de votre API. Mentionnez dans le fichier TODO.md les éléments que vous auriez souhaité implémenter pour démontrer une meilleure maîtrise des patterns et de l'architecture.

Pour toute questions, vous pouvez contacter [mhenry@lesechos.fr](mailto:mhenry@lesechos.fr)