

# **ING1 - Développement WEB**

Mardi 20 Mai 2025  
15h30 - 17h30

Examen blanc TD1  
Contrôle continu

**Jordan SABLON**

**EIL Côte d'Opale, Calais**

## Modalités de l'examen

- L'épreuve est à réaliser sur un ordinateur des salles informatiques de l'école
- L'accès à Internet n'est pas autorisé durant l'épreuve
- Seuls les documents fournis avec le sujet d'examen sont autorisés

## Procédure pour rendre votre évaluation

Dans le dossier backend : supprimer le dossier « node\_modules » s'il existe.

Dans le dossier frontend : supprimer les dossiers « node\_modules » et « .next » s'ils existent.

Compressez votre dossier code dans une archive au format .zip ou .rar nommée « **web\_NOM\_Prenom** » et déposez-le à cet endroit sur votre machine : « V:/donnees/TESTEXAM ».

⚠ Le non respect du format de dépôt ou du nommage de l'archive entraînera une pénalité sur la note de l'examen

## Pré-requis techniques

Afin de pouvoir réaliser cet examen, votre ordinateur doit disposer des logiciels suivants :

- NodeJS
- NPM
- VS Code (ou tout autre environnement de développement de votre choix)
- Un navigateur web

## Barême de notation

Le barême de notation est donné à titre indicatif à la fin du sujet d'examen.

## Précisions

- Vous pouvez commencer au choix par l'exercice 1 ou l'exercice 2
- Vous disposez des supports et codes des TDs et TPs ainsi que de documentations supplémentaires qui pourraient vous être utiles dans le dossier « ressources »
- ⚠ Attention à bien récupérer le contenu qui se trouve dans le dossier « V:/donnees/SUJET » et à bien travailler depuis votre session au risque de ne pas pouvoir sauvegarder votre travail !

# Configuration de l'environnement de développement

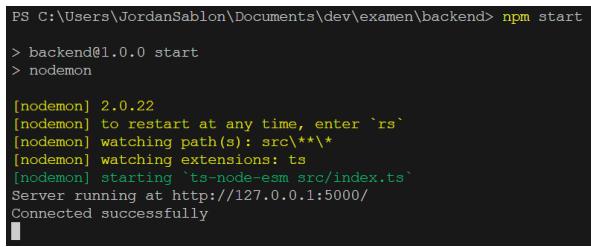
## 1. Récupération du code source

Récupérer le code source fourni avec le sujet d'examen et ouvrir le dossier « code » dans VS Code (ou l'IDE de votre choix)

## 2. Exercice 1 : Node.js

Depuis le dossier « backend », exécutez dans un terminal la commande suivante pour démarrer le serveur : « **npm start** »

La console devrait vous indiquer ceci :



```
PS C:\Users\JordanSablon\Documents\dev\examen\backend> npm start
> backend@1.0.0 start
> nodemon

[nodemon] 2.0.22
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): src\**\*
[nodemon] watching extensions: ts
[nodemon] starting `ts-node-esm src/index.ts`
Server running at http://127.0.0.1:5000/
Connected successfully
```

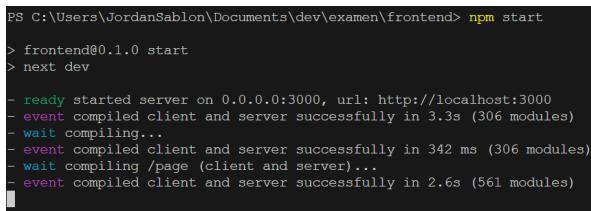
Figure 1: Terminal VS Code indiquant que le serveur est démarré

Et le serveur devrait être accessible via l'adresse (<http://localhost:5000/>) et devrait afficher « Hello ! »

## 3. Exercice 2 : React

Depuis le dossier « frontend », exécutez dans un terminal la commande suivante afin de démarrer le serveur : « **npm start** »

La console devrait vous indiquer ceci :



```
PS C:\Users\JordanSablon\Documents\dev\examen\frontend> npm start
> frontend@0.1.0 start
> next dev

- ready started server on 0.0.0.0:3000, url: http://localhost:3000
- event compiled client and server successfully in 3.3s (306 modules)
- wait  compiling...
- event compiled client and server successfully in 342 ms (306 modules)
- wait  compiling /page (client and server)...
- event compiled client and server successfully in 2.6s (561 modules)
```

Figure 2: Terminal VS Code indiquant que le frontend est démarré

Et le serveur devrait être accessible via l'adresse (<http://localhost:3000/>) et devrait afficher « Hello ! »

## Exercice 1 - Backend NodeJS

Dans cet exercice, vous travaillerez depuis l'application présente dans le dossier « backend »

Vous disposez d'une application NodeJS ainsi que d'un jeu de données mocké.

Pour rappel les données d'un mock sont stockées dans un fichier et sont réinitialisées à chaque redémarrage du serveur. Ils permettent de ne pas avoir recours à l'usage d'une base de données.

On retrouve les différents éléments de l'architecture étudiée en cours, à savoir :

- **une couche dédiée au routage** : permet d'appeler un contrôleur spécifique pour chaque type d'appel HTTP
- **une couche de contrôleur** : permet de récupérer les données de la requête et fait appel à un service dédié
- **une couche de service** : permet de réaliser le traitement souhaité sur la donnée

### Travail à réaliser

Vous devez implémenter les routes suivantes dans l'application :

GET	/users	Retourne la liste des noms de chaque utilisateur triée par ordre alphabétique
GET	/users/:name	Retourne l'utilisateur correspondant au nom passé en paramètre

Vous pouvez tester le retour de vos appels HTTP via ces URLs :

<http://localhost:5000/users>

<http://localhost:5000/users/:name>

Exemple, le premier appel devra retourner le résultat suivant :

`["Bernard", "Caron", "Dubois", "Fontaine", "Girard", "Lefèvre", "Lemoine", "Marchand", "Morel", "Petit"]`

Exemple, l'appel `http://localhost:5000/users/Fontaine` devra retourner le résultat suivant :

`{"id":10, "firstname": "Inès", "name": "Fontaine", "role": "administrateur"}`

## Exercice 2 - Frontend React

Le but de cet exercice est de permettre la saisie de texte et de vérifier si ce texte contient un nombre pair ou impair de caractères. On pourra également transformer le texte saisi en majuscules, minuscules et réinitialiser la saisie.

### Travail à réaliser

Créer un composant `<Saisie />` qui contiendra un champ de saisie de texte (`<input />`). Ce champ de saisie doit être initialisé au chargement du composant avec la valeur « Ceci est le texte par défaut » (attention, on ne demande pas d'initialiser la variable lors de sa déclaration).

Créer un composant `<Pair />` qui prendra en propriété la valeur de la saisie et qui affichera si la phrase contient un nombre pair ou impair de caractères. Cet affichage sera actualisé automatiquement dès que la saisie sera modifiée.

Créer, dans le composant `<Saisie />` un bouton « Effacer la saisie » qui permet de vider le champ de saisie.

Créer, dans le composant `<Saisie />` un bouton « Passer en majuscules » qui permet de transformer la texte saisi en majuscules. Ce bouton sera affiché par défaut au chargement de composant, puis sera par la suite visible uniquement lorsque le texte aura été transformé en minuscules.

Créer, dans le composant `<Saisie />` un bouton « Passer en minuscules » qui permet de transformer la texte saisi en minuscules. Ce bouton ne sera visible que lorsque le texte aura été transformé en majuscules.

Les différents boutons devront être désactivés lorsque le champ de saisie ne contient pas de texte.

### Cas d'usages

Voici quelques captures d'écran de l'application à réaliser :

1. On arrive sur l'application, on obtient ce résultat :

Ceci est le texte par défaut	Effacer la saisie
Passer en majuscules	

La phrase "**Ceci est le texte par défaut**" contient un nombre pair de caractères.

2. Après avoir cliqué sur "Effacer la saisie", on obtient le résultat suivant :

	Effacer la saisie
Passer en majuscules	Aucun texte saisi

2. Après avoir saisi du texte, on obtient le résultat suivant :

Texte d'exemple	Effacer la saisie
Passer en majuscules	

La phrase "**Texte d'exemple**" contient un nombre impair de caractères.

4. Après avoir cliqué sur "Passer en majuscules", on obtient le résultat suivant :

TEXTE D'EXEMPLE	Effacer la saisie
Passer en minuscules	

La phrase "**TEXTE D'EXEMPLE**" contient un nombre impair de caractères.

5. Après avoir cliqué sur "Passer en minuscules", on obtient le résultat suivant :

texte d'exemple	Effacer la saisie
Passer en majuscules	

La phrase "**texte d'exemple**" contient un nombre impair de caractères.

## Barème

Le barème donné ci-dessous est uniquement à titre indicatif et est susceptible d'être adapté lors de la correction de l'examen.

### Exercice n°1 - Backend NodeJS : 8 points

Critère n°1	GET /users	5 points
Critère n°2	GET /users/:name	3 points

### Exercice n°2 - Frontend React : 12 points

Critère n°1	Gestion de la saisie	2 points
Critère n°2	Désactiver le bouton d'ajout	1 point
Critère n°3	Valeur par défaut au chargement du composant	1 point
Critère n°4	Vérifier si le nombre de caractères est pair ou non	1 point
Critère n°5	Affichage du résultat	2 points
Critère n°6	Réinitialisation de la saisie	1 point
Critère n°7	Transformation du texte en majuscules	2 points
Critère n°8	Transformation du texte en minuscules	2 points

---

Fin de l'énoncé