

ING1 - Développement WEB

Lundi 02 Juin 2025
10h15 - 12h15

Examen de fin de module
Session 1

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 entrainera 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

Barème de notation

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

Précisions

- Cet examen se décompose en deux parties : un backend NodeJS à implémenter qui permettra d'alimenter une application frontend en React. La seconde partie nécessite donc que la première soit réalisée pour fonctionner correctement (en cas de difficultés pour la première partie, vous pouvez copier le jeu de données dans le frontend ce qui vous permettra tout de même de réaliser la seconde partie)
- Vous disposez des supports et codes des TDs 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. Démarrage du backend

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. Démarrage du frontend

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 ! »

Présentation du sujet

Le but de cet examen est de réaliser une application qui permettra à l'utilisateur de pouvoir faire une recherche en précisant un ingrédient afin de récupérer l'ensemble des recettes qui contiennent cet ingrédient au clic sur un bouton « Filtrer ».

Voici quelques cas d'usage de l'application à réaliser :

1. Aucun ingrédient saisi : toutes les recettes sont affichées

Filter par ingrédient :

Saisir un ingrédient pour filtrer les recettes

Liste des recettes :

Poulet rôti :

- poulet
- ail
- huile
- sel
- poivre

Salade de tomates :

- tomate
- salade
- vinaigre
- huile
- sel

Omelette nature :

- oeuf
- sel
- poivre
- beurre

Gratin de courgettes :

- courgette
- crème
- fromage
- sel
- poivre

Spaghetti bolognaise :

- pâtes
- boeuf
- tomate
- oignon
- ail

2. Saisie d'un ingrédient présent dans une seule recette

Filter par ingrédient :

Saisir un ingrédient pour filtrer les recettes

Liste des recettes :

Poulet rôti :

- poulet
- ail
- huile
- sel
- poivre

3. Saisie d'un ingrédient présent dans plusieurs recettes

Filter par ingrédient :

Saisir un ingrédient pour filtrer les recettes

Liste des recettes :

Omelette nature :

- oeuf
- sel
- poivre
- beurre

Quiche lorraine :

- oeuf
- crème
- lardons
- pâte brisée
- fromage

Salade composée :

- salade
- oeuf
- tomate
- fromage
- vinaigrette

Tortilla espagnole :

- oeuf
- pomme de terre
- oignon
- huile
- sel

4. Saisie d'un ingrédient qui n'est présent dans aucune recette

Filtre par ingrédient :

Saisir un ingrédient pour filtrer les recettes

muscade

Liste des recettes :

Aucune recette trouvée.

Partie n° 1 - Backend NodeJS

Dans cette partie, 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 la route suivante dans l'application :

GET	/recipes	Retourne la liste des recettes en fonction d'un filtre (ingrédient)
-----	----------	---

La particularité de cet appel est qu'il devra tenir compte de l'ingrédient saisi dans le filtre de recherche. Il s'agit d'une notion que l'on a abordé en TD mais que vous n'avez pas mis en pratique.

Vous devrez donc utiliser l'attribut « req.query » au niveau du contrôleur afin de récupérer l'ingrédient saisi dans le filtre, qui sera transmis au service. Ce dernier se chargera de retourner les recettes en conséquences.

Vous pouvez tester le retour de votre appel HTTP via ces URLs directement dans un navigateur :

<http://localhost:5000/recipes>

Cet appel retournera la liste de toutes les recettes.

<http://localhost:5000/recipes?ingredient=poulet>

Cet appel retournera la liste des recettes qui contiennent l'ingrédient poulet.

Il y a plusieurs possibilités :

- soit un ingrédient est précisé et il faut retourner uniquement les recettes qui contiennent cet ingrédient
- soit l'ingrédient précisé n'est présent dans aucune recette, alors on retourne un tableau vide
- soit aucun ingrédient n'est précisé, alors on retourne toutes les recettes

Vous trouverez dans les ressources des documentations qui vous permettront d'implémenter ce fonctionnement.

Partie 2 - Frontend React

Dans cette partie, vous allez mettre en place le champ de saisie permettant de renseigner un ingrédient, le bouton « Filtrer » qui permettra de filtrer les recettes selon l'ingrédient saisi, l'appel au backend afin de récupérer les recettes, ainsi que le composant permettant d'afficher une recette.

Travail à réaliser

Créer un composant `<Recipes />` qui constituera notre composant racine (appelé depuis le fichier « page.tsx »). C'est ce composant qui fera appel aux autres composants.

Ce composant contiendra :

- un titre `<h3>` « Filtre par ingrédient »
- un titre `<h4>` « Saisir un ingrédient pour filtrer les recettes »
- l'appel au composant `<Search />`
- l'appel au composant `<Recipe />` pour chaque recette à afficher
- une fonction permettant de faire appel au backend afin de rafraichir la liste des recettes en fonction de l'ingrédient saisi dans le filtre

Au chargement de ce composant, un appel sera également réalisé au backend afin de récupérer l'ensemble des recettes pour les afficher.

Créer un composant `<Search />` qui contiendra les éléments permettant de saisir un ingrédient pour filtrer les recettes.

Ce composant sera appelé par le composant `<Recipes />` et contiendra :

- un champ de saisie de texte (`<input />`)
- un bouton « Filtrer »

Ce composant aura comme propriété (props) la fonction créée dans le composant `<Recipes />` permettant de réaliser l'appel au backend permettant de rafraichir les recettes selon l'ingrédient renseigné dans le filtre.

Créer un composant `<Recipe />` dont le rôle sera d'afficher une recette. Ce composant aura comme propriété (props) une recette et sera appelé par le composant `<Recipes />` pour chaque recette à afficher, uniquement s'il y a des recettes à afficher, sans quoi on affichera « Aucune recette trouvée. ». Il contiendra :

- le titre de la recette en gras
- la liste des ingrédients (utilisation de la balise ``)

Barème

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

Partie n°1 - Backend NodeJS : 6 points

Critère n°1	Mise en place la route GET /recipes	1 point
Critère n°2	Mise en place du contrôleur	1 point
Critère n°3	Récupération du filtre	1 point
Critère n°4	Mise en place du service	1 point
Critère n°5	Prise en compte de l'ingrédient afin de retourner les recettes correspondantes	2 points

Exercice n°2 - Frontend React : 14 points

Critère n°1	Création des éléments du composant <Recipes />	3 points
Critère n°2	Création des éléments du composant <Search />	2 points
Critère n°3	Création des éléments du composant <Recipe />	2 points
Critère n°4	Appel au backend pour récupérer les recettes au chargement du composant	2 points
Critère n°5	Appel au backend pour récupérer les recettes au clic sur le bouton « Filtrer »	4 points
Critère n°6	Affichage du texte « Aucune recette trouvée. »	1 point

Fin de l'énoncé