

Nom et prénom : Mohamed Hichem LAOUAR

Manuelle d'utilisation pour le test technique

- 1) La première étape est de cloner le repo github du test avec la commande

Git clone <https://github.com/hichemCodes/AdCoinSolutionsTest.git>

Exercice 1

Pour le premier exercice il se fait juste d'ouvrir le fichier array.php qui se trouve dans le dossier Exercice 1, pour faire vos tests sur la fonction sum_array il faut remplacer les paramètres de l'appel de cette fonction dans la ligne 36 par vos paramètres

Exercice 2

Pour le deuxième exercice j'ai choisi de créer une API pour gérer des posts d'un blog avec Nods js à l'aide de la bibliothèque Express et une petite Interface React js pour faire les tests sur l'API créée

Pour la base de données j'ai utilisé MySQL, en gros j'ai créé 2 tables : la table post qui contient les posts et la table comment qui contient les commentaires des posts avec une clé étrangère pour référencer le post qui appartient à un commentaire

Voici les endpoints que j'ai créé

GET /posts : qui retourne la liste des posts qui sont stockés dans la bdd

GET /posts/id : qui retourne les informations du post de l'id id

GET /posts/id/comments : qui retourne la liste des commentaires du post de l'id id

POST /post : qui permet de stocker un nouveau post

Comment installer l'application

- 1) Après avoir cloné le projet, il faut se positionner sur le fichier Exercice 2 et de lancer la commande npm install pour installer toutes les dépendances du backend (express,mysql,body parser....)

- 2) importé la base de données **TestADcoin.sql** qui se trouve dans le repo cloné
- 3) Après avoir installé les dépendances et importé la bdd, il faut modifier le fichier `connection.js` qui se trouve dans le dossier `server` pour l'adapter à votre serveur mysql (modifier le user et le password si ils sont adaptés à votre user et password du mysql)
- 4) Maintenant vous pouvez tester le back end en lançant la commande "**node .**" Après avoir se positionner sur le dossier `server`
- 5) Normalement vous aurez un message `connected to the database` (sinon, donc vous avez une erreur du user au mot de passe de votre mysql)
- 6) Maintenant vous pouvez tester l'api en allant sur l'url :
<http://localhost:3001/api/posts>
Vous pouvez aussi testé tout les endpoint créés
- 7) Pour rendre le test plus interactif vous pouvez installer le front en se positionnant sur le dossier `client` et de lancer la commande **npm install et après npm start**
- 8) Maintenant vous pouvez aller sur l'url <http://localhost:3000/blog> pour tester le mini blog que j'ai créé a l'aide de l'api créée précédemment et en utilisant `react js` et `tailwind css` pour le style
- 9) Vous avez maintenant la possibilité de tester toutes les fonctionnalités offertes par l'api :
 - lister les posts
 - Afficher un post précis
 - Ajouter un post

Note : pour l' endpoint `POST /post` (ajout d'un post) j'avais un petit souci avec `axios` dans le front, donc si l'ajout d'un post à travers l'interface ne marche pas vous pouvez utiliser `axios` pour le tester :

POST



http://localhost:3001/api/post

Params

Authorization

Headers (9)

Body ●

Pre-request Script

Tests

Settings

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

JSON ▾

```
1 {  
2   ... "title": "new title",  
3   ... "description": "new description 2"  
4 }
```