

Project

- **Durée** : 1 semaine
- **Type de projet** : seul / pair programming
- **Nom du repo** : `adapi-[nom_github]`

Modalités de rendu

L'ensemble de l'exercice devra être pushé sur la branche main du projet du git.

Objectif du projet

Ce projet a pour objectif de réutiliser la base de données **adatabase**, hébergée dans Docker, afin d'accéder au contenu des tables créées auparavant. Il s'agira ensuite de développer une **API REST** avec Express permettant d'effectuer des requêtes **CRUD** (Create, Read, Update, Delete) sur ces différentes tables. (Si besoin, vous pouvez recréer les tables proposées dans **adatabase**.)

Stack technique (obligatoire)

Élément	Technologie	Lien
environnement	Node.js	https://nodejs.org/
framework backend	Express	https://expressjs.com/
Base de données	PostgreSQL	https://www.postgresql.org/

Initialisation du projet

Pour pouvoir utiliser express avec postgres en local, on va créer une instance de base de données grâce à docker.

Pour ce faire, tu peux suivre le guide suivant : **installer-express.md** que vous pouvez retrouver dans le dossier `guides`.

Pré-requis techniques

Pour mener à bien ce projet, il est nécessaire de maîtriser :

- les bases du **terminal**
- les bases du **JavaScript**
- les bases du **SQL**

Fonctionnalités à développer

- ☐ Créer une route **GET** pour les **ressources**
- ☐ Créer une route **GET** pour une **ressource**
- ☐ Créer une route **POST** pour les **ressources**
- ☐ Créer une route **PUT** pour les **ressources**

- ☐ Créer une route **DELETE** pour les **ressources**
- ☐ Créer les même routes pour les **thèmes / skills / resources_skills**

⚠ **Réalisez toutes les fonctionnalités obligatoires avant de passer aux bonus.**

🚫 Ce que vous ne devez pas faire

développer une interface front-end (html/css)

💎 Bonus possibles

- mettre en place swagger pour documenter l'api :
 - <https://swagger.io/tools/swagger-ui/>
 - <https://www.npmjs.com/package/swagger-ui-express>
- △ pensez à bien mettre en place le swagger.json dans le projet à la racine via ce lien :
 - <https://editor.swagger.io/>
 - vous pouvez sinon copier/coller ce code de base :

```
{
  "openapi": "3.0.0",
  "info": {
    "title": "My API",
    "version": "1.0.0"
  },
  "paths": {
    "/resources": {
      "get": {
        "summary": "List all resources",
        "responses": {
          "200": {
            "description": "OK"
          }
        }
      }
    }
  }
}
```

si vous avez besoin d'aide n'hésitez pas à demander à un.e responsable de formation

- Ajouter une table **projects** pour définir les compétences nécessaires à la réalisation d'un projet :
 - le projet aura besoin d'un **titre** et d'une **description**.
 - elle devra faire le lien avec les tables **skills** et **themes**.
 - Chaque projet pourra avoir plusieurs compétences et plusieurs thèmes associés.
- ☐ Créer les routes associées à la table **projets**

- ☐ Ajouter une route `GET /skills/:id/resources` pour récupérer toutes les ressources associées à une compétence donnée.
- ☐ Ajouter une route `GET /themes/:id/resources` pour récupérer toutes les ressources associées à un thème donné.



Modalités d'évaluation

TODO: à copier en fonction de si c'est un projet de groupe ou individuel



Compétences à acquérir

- ☐ Je sais créer une API REST avec Express
- ☐ Je sais me connecter à une base de données PostgreSQL
- ☐ Je sais gérer les requêtes SQL avec pg
- ☐ Je sais utiliser dotenv pour gérer les variables d'environnement
- ☐ Je sais utiliser nodemon pour le développement
- ☐ Je sais structurer un projet Node.js avec Express
- ☐ Je sais créer des routes CRUD (Create, Read, Update, Delete) pour une API REST
- ☐ Je sais gérer les erreurs de connexion à la base de données
- ☐ Je sais utiliser les modules ES6 dans un projet Node.js



Soutenance

X aucune soutenance n'est prévue pour ce projet