

# Checkpoint alternance dev CDA

Les exercices sont indépendants les uns des autres.

Il vaut mieux livrer un exercice incomplet et/ou comportant des bugs plutôt que de ne rien soumettre.

## Partie 2 : backend

Crée un nouveau projet NodeJS Typescript.

Pour un meilleur confort de dev, utilise *ts-node-dev*.

Le but de cet exercice est de créer une API GraphQL avec Apollo Server et TypeGraphQL pour enregistrer et lire des pays avec leur code, leur nom et leur emoji.

Utilise une base de données SQLite et TypeORM en ORM.

Voici les packages dont tu auras besoin et leur version :

```
"dependencies": {  
  "apollo-server": "^3.11.1",  
  "class-validator": "^0.14.0",  
  "reflect-metadata": "^0.1.13",  
  "sqlite3": "^5.1.4",  
  "ts-node-dev": "^2.0.0",  
  "type-graphql": "^1.1.1",  
  "typeorm": "^0.3.12"  
}
```

Crée une mutation qui prend en paramètres :

- un code (FR, BE, AN, ...),
- un nom (France, Belgique, Andorre, ...),
- un emoji (fr, belgium, andorra, ...),
- et qui enregistre cette entrée en BDD.

Crée ensuite 2 queries :

- Une qui renvoie la liste de tous les pays (avec pour attributs, pour chaque pays, le code, le nom, l'emoji),
- Une autre qui prend en paramètre le code du pays et qui renvoie le pays en question.

Bonus : ajoute un code continent à la mutation d'ajout de pays et une query qui permet de récupérer tous les pays d'un continent.