

DOCKER COMPOSE

MERN STACK

DOCUMENTATION POUR CONFIGURER UN STACK MERN AVEC DOCKER COMPOSE

Objectif

Ce guide détaille les étapes nécessaires pour configurer un environnement de développement local utilisant le stack **MERN** (**M**ongoDB, **E**xpress.js, **R**ead.js, **N**ode.js) avec Docker Compose. **Docker Compose** permet de gérer plusieurs conteneurs Docker en même temps, simplifiant ainsi le déploiement et la gestion de l'application.

Prérequis

Avant de commencer, assurez-vous d'avoir les éléments suivants installés sur votre système :

- Docker : [Installation de Docker](#)
- Docker Compose : [Installation de Docker Compose](#)

Création de la structure du projet

On crée un répertoire pour le projet et on l'organise comme suit :

```
mern-docker/
```

```
|-- back/
```

```
| |-- package.json
```

```
| |-- server.js
```

```
| |-- ...
```

```
|-- front/
| |-- package.json
| |-- src/
| | |-- App.js
| | |-- index.js
| | |-- ...
|-- docker-compose.yaml
```

Configuration du Back

- Initialisation du serveur Express :

Créez un fichier server.js : **touch bacck/server.js**

Initialisation : **npm install express mongoos**

Configuration du Front

- Initialisation de l'application React :

npx create-react-app front

cd front

npm install axios

Dockerisation de l'application

1 – Création des Dockerfiles

- Créer un Dockerfile pour le serveur backend.
- Créer un Dockerfile pour le client frontend.

Création du fichier docker-compose.yaml

2 – Définir les services pour le serveur backend, la base de données MongoDB et le client frontend.

Construction des images Docker

docker-compose build

Démarrage des conteneurs

docker-compose up ou **docker-compose up -d**