

# BRDGame

## Documentation Docker v2

### Mise en place des fichiers Docker et modification des ports

#### Au préalable :

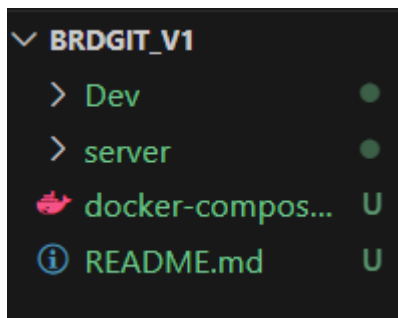
Docker Desktop allumé

#### En ligne de commande dans le dossier de travail :

Pour démarrer le projet : `docker compose up --build -d`

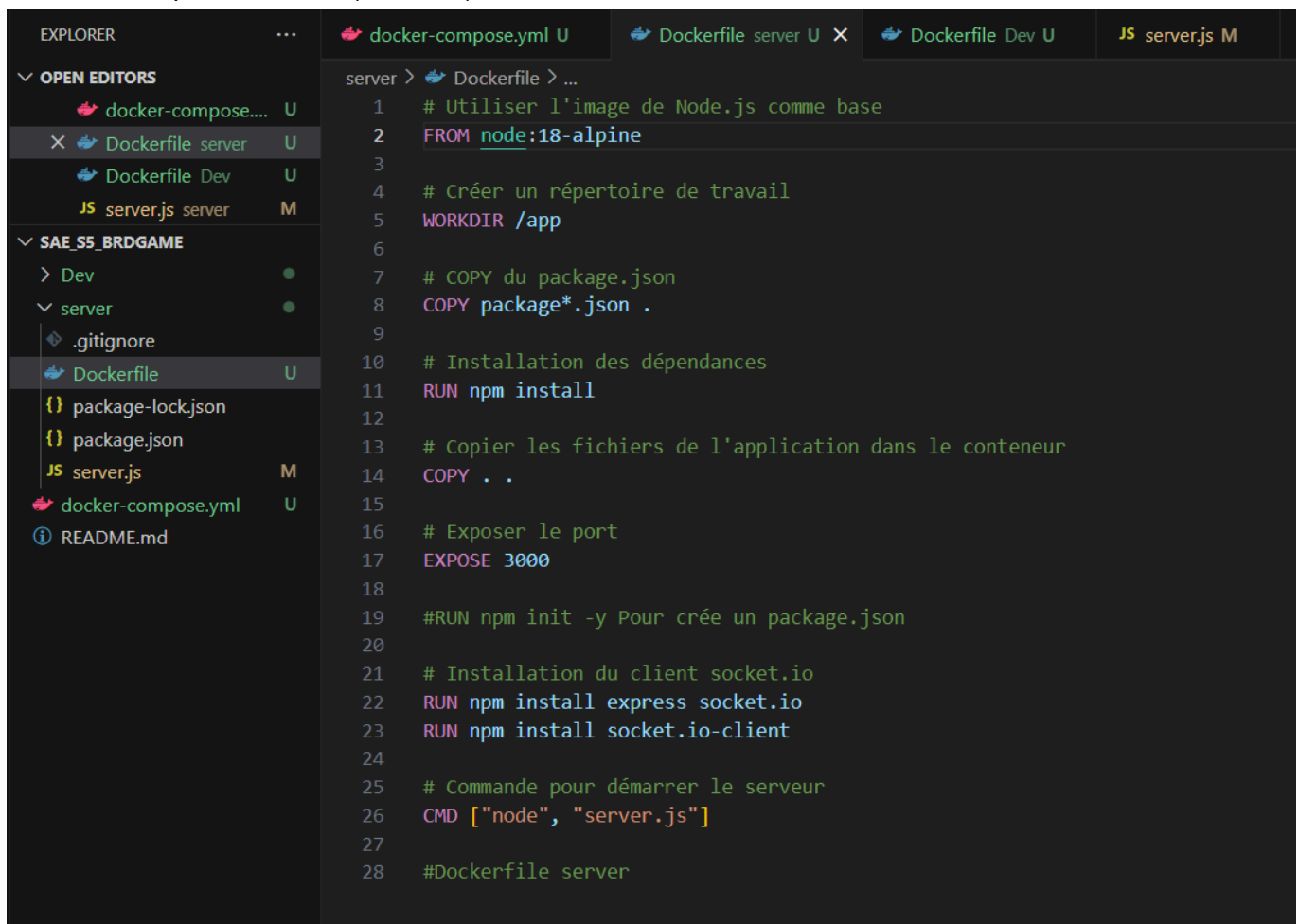
Pour éteindre le projet : `docker compose down`

Emplacement du `docker-compose.yml` : Racine du projet



Dans le dossier `server` :

Le Dockerfile pour le `server` (backend)



Dans la configuration du server :

Configurer les ports pour les ouvrir au frontend

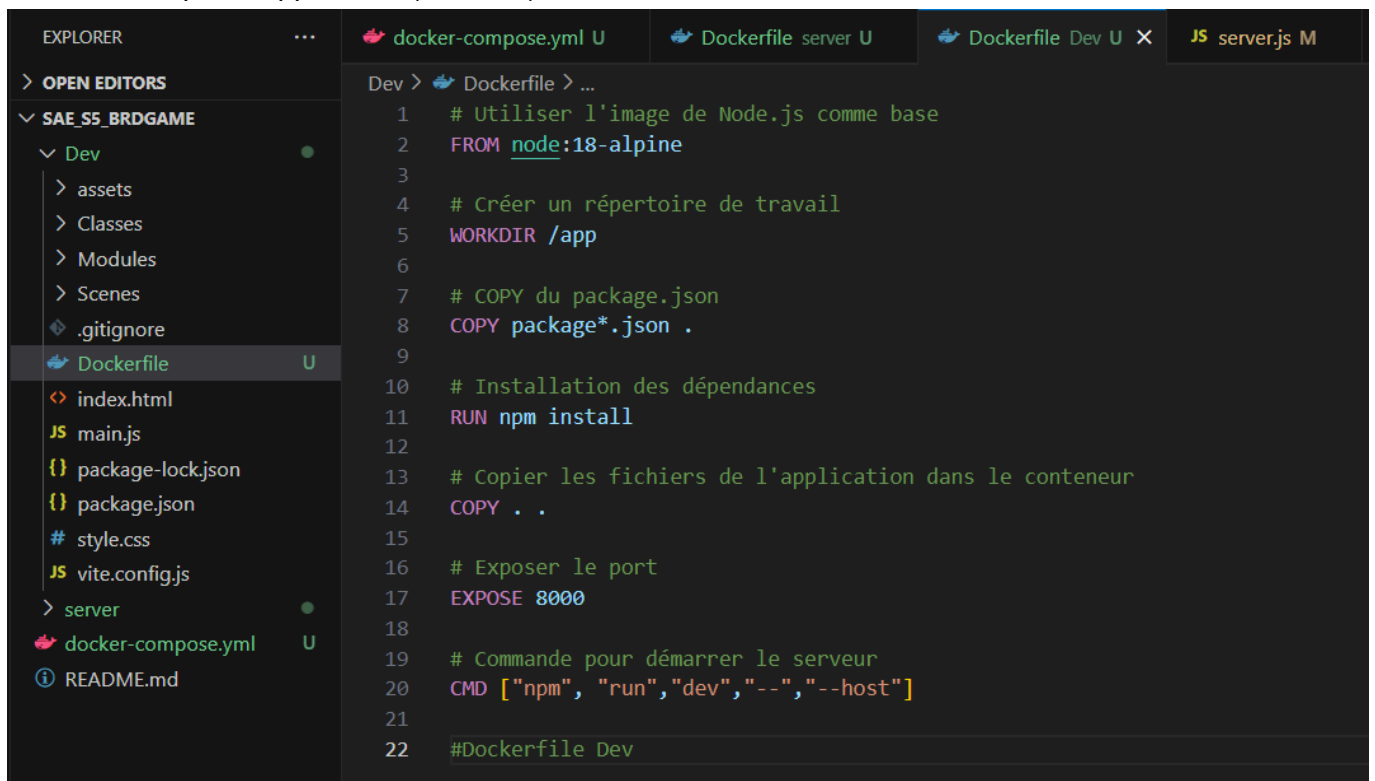


The screenshot shows the VS Code interface with the Explorer on the left and the Editor on the right. The Explorer shows a project structure with a 'server' folder containing 'server.js'. The Editor displays the 'server.js' file with the following code:

```
server > JS server.js > ioServer.on('connection') callback
26 const ioServer = new socketIO.Server(server, {
31   }
32 });
33
34 const PORT = process.env.PORT || 3000;
35 app.listen(PORT, () => {
36   console.log(`Server is running on port ${PORT}.`);
37 });
38
39 createInitialFarmables()
40
41 ioServer.on('connection', (socket) => {
42
43   players.push({ x: 0, y: 0, id: socket.id, inGame: false, hp: 0});
44
45   console.log(`A player connected: ${socket.id}`);
46 }
```

Dans le dossier Dev :

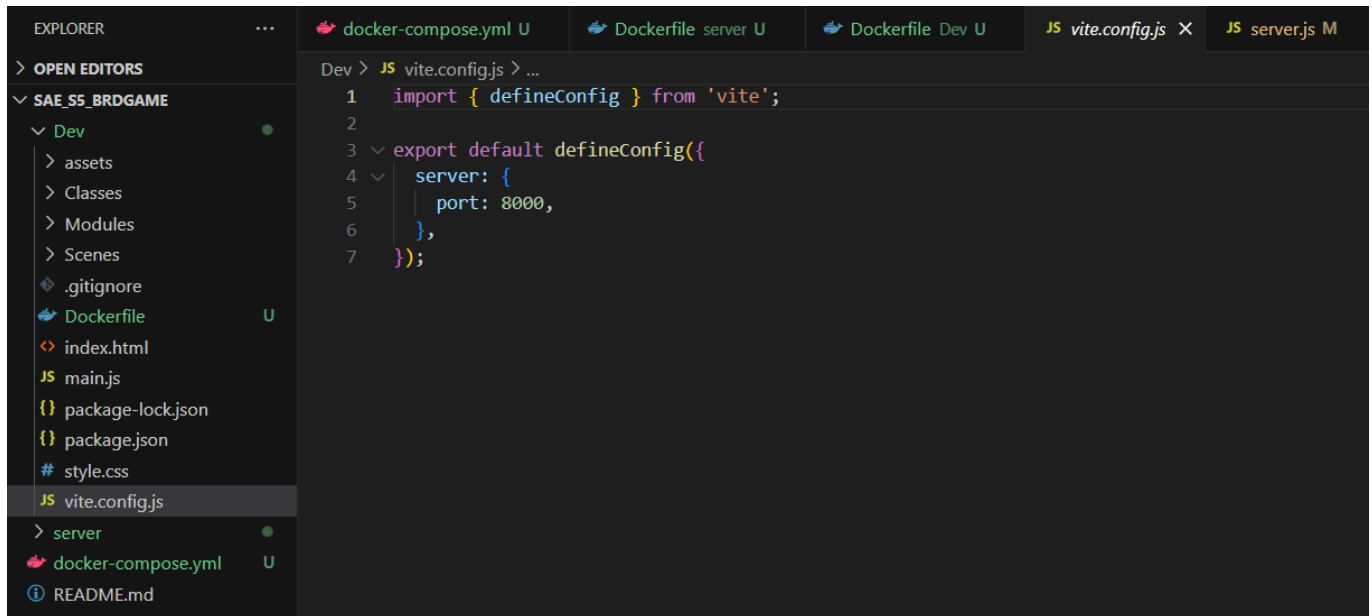
Le Dockerfile pour l'application (frontend)



The screenshot shows the VS Code interface with the Explorer on the left and the Editor on the right. The Explorer shows a project structure with a 'Dev' folder containing 'Dockerfile'. The Editor displays the 'Dockerfile' file with the following code:

```
Dev > Dockerfile > ...
1 # Utiliser l'image de Node.js comme base
2 FROM node:18-alpine
3
4 # Créer un répertoire de travail
5 WORKDIR /app
6
7 # COPY du package.json
8 COPY package*.json .
9
10 # Installation des dépendances
11 RUN npm install
12
13 # Copier les fichiers de l'application dans le conteneur
14 COPY . .
15
16 # Exposer le port
17 EXPOSE 8000
18
19 # Commande pour démarrer le serveur
20 CMD ["npm", "run", "dev", "--", "--host"]
21
22 #Dockerfile Dev
```

La configuration du port de Vite :



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left and the Editor view on the right. The Explorer sidebar shows the project structure for 'SAE\_S5\_BRDGAME', including a 'Dev' folder with subfolders 'assets', 'Classes', 'Modules', and 'Scenes', and files like '.gitignore', 'Dockerfile', 'index.html', 'main.js', 'package-lock.json', 'package.json', 'style.css', 'vite.config.js', 'server', 'docker-compose.yml', and 'README.md'. The Editor view shows the 'vite.config.js' file with the following code:

```
1 import { defineConfig } from 'vite';
2
3 export default defineConfig({
4   server: {
5     port: 8000,
6   },
7 });
```

BRDGame - Documentation Docker v2

Projet de SAE du semestre 5 du BUT Informatique à l'IUT d'Amiens de (Liam Target, Justin Largilliere, Lorenzo Vatin, Lucas Gonthier, Malcolm Walter, Victor Da Silva Lourenco Marques)