














Zelda BOTW API (Materials i Monstres)

Aquest projecte proporciona una API RESTful per consultar i gestionar informació de **materials** i **monstres** del joc *The Legend of Zelda: Breath of the Wild*, amb dades traduïdes al castellà.

Contingut

- [Zelda BOTW API \(Materials i Monstres\)](#)
 -  [Contingut](#)
 -  [Característiques](#)
 -  [Instal·lació amb Docker](#)
 -  [Requisits](#)
 -  [Instruccions](#)
 -  [Instal·lació en local](#)
 -  [Requisits](#)
 -  [Instruccions](#)
 -  [Gestió del backup i persistència de dades](#)
 -  Si vols **conservar els canvis** a la base de dades
 -  Si vols **restaurar el backup original**
 -  Si vols **exportar i conservar una còpia del teu estat actual**
 -  [Notes](#)

Característiques

- Dades emmagatzemades a MongoDB
 - CRUD complet per materials i monstres
 - Backend amb Node.js + Express
 - Suport per WebSockets (pròximament)
 - Traduït al castellà
-

Instal·lació amb Docker

Requisits

- [Docker](#)
- [Docker Compose](#)

Instruccions

1. Clona el repositori:

```
git clone https://github.com/el-teu-usuari/zelda-botw-api.git
cd zelda-botw-api
```

2. Executa docker compose:

```
docker compose up --build
```

*El modificador --build només cal quan hi hagi canvis en el codi

3. L'API estarà disponible a http://localhost:3001

4. Si tens problemes amb l'execució de docker pot ser que tinguis el port ocupat, assegura't que no tens una altra instància del projecte funcionant amb

```
docker ps
```

5. Atura els contenidors corresponents amb

```
docker stop <nomdelcontenidor>
```

6. Revisa el fitxer .env per evitar conflictes de ports

```
API_EXT_PORT=3001  
MONGO_EXT_PORT=27018
```

7. Quan no estiguis treballant amb l'API recorda aturar-ho tot fent:

```
docker compose stop
```

8. Comprova si hi ha canvis al repositori:

```
git pull origin master
```

9. Si vols restablir la base de dades amb la còpia de seguretat i assegurar-te que el codi està actualitzat:

```
docker compose down --volumes  
docker compose up --build
```

 [Més detalls tècnics](#)

 Instal·lació en local

 Requisits

- [Mongo](#)
- [Node.js v20+](#)

Instruccions

1. Assegura't que el servei de MongoDB està actiu:

```
sudo systemctl start mongod
```

2. Executa la restauració de la base de dades:

```
mongorestore --db zelda_botw ./backup/zelda_botw
```

3. Instal·la les dependències de node:

```
npm install
```

4. Revisa el fitxer .env perquè no hi hagi ports que puguin estar ocupats

```
API_PORT=3001  
MONGO_URL=mongodb://localhost:27017/zelda_botw
```

5. Executa l'aplicació:

```
npm start
```



Gestió del backup i persistència de dades



Si vols **conservar els canvis** a la base de dades

Els canvis es desen automàticament dins del volum **mongo_data**.

Això passa **sempre que no eliminis el volum**:

```
docker compose down      # ✓ Manté les dades  
docker compose stop      # ✓ Manté les dades
```

! No executis **--volumes** si vols conservar les dades.

Si vols **restaurar el backup original**

Aquesta opció esborra les dades actuals i restaura la base de dades des del backup inclòs al projecte:

```
docker compose down --volumes
docker compose up --build
```

El sistema detectarà que no existeix cap restauració i farà el **mongorestore** automàticament.

Si vols **exportar i conservar una còpia del teu estat actual**

1. Fes un **dump** des del contenidor:

```
docker exec -t <nom_contenidor_mongo> mongodump --db=zelda_botw --out=/backup-temp
```

2. Copia'l al teu sistema:

```
docker cp <nom_contenidor_mongo>:/backup-temp ./backup
```

3. La carpeta **./backup/zelda_botw** contindrà ara una còpia actualitzada del teu estat.
-



Notes

- Pots fer servir Swagger (<http://localhost:3001/api-docs>) per explorar i provar els endpoints.
 - Les dades de materials i monstres estan basades en l'API original: <https://botw-compendium.herokuapp.com>
-