

DOCUMENTO PUESTA EN MARCHA

Documento de puesta en marcha para la prueba técnica de borja sahuquillo Ferrer.

Iniciar el proyecto

Despliegue en macOS / Linux

Necesitamos Docker, node versión 12+ y docker-compose.

1. Navegamos a /api-employee-and-department y hacemos **npm i** de los node_modules.
2. Y luego en la carpeta del proyecto, tenemos que lanzar los comandos:
 - ***docker network create employee-and-department-network***
 - ***docker-compose up -d***

cosas a tener en cuenta:

- Para levantar cada parte del proyecto por separado, por si tuviéramos algún problema, entramos en sus carpetas:
 - **api-employee-and-department**
 - Y usamos el comando: *docker-compose up -d*
 - **db-employee-and-department**
 - Y usamos el comando: *docker-compose up -d*
 - **webapp-employee-and-department**
 - Y usamos el comando: *docker-compose up -d*

Nota: El web app con nginx, al refrescar con una ruta específica da un problema con el nginx, no he podido solucionarlo, está el problema localizado en la configuración del nginx. *Se puede levantar, con un npm run start para comprobar que el reload funciona correctamente.*

Inicio de sesión front

Para poder logearnos, se crea automáticamente un usuario con las credenciales:

```
{  
email: admin@gmail.com,  
password: admin1234  
}
```

Si queremos crear uno, nos metemos en el swagger y realizamos un post con email y password. (<http://localhost:3000/explorer/#/User>).

Propuesta de backend

Proyecto realizado con Loopback 3, un framework de node.js

Tiene un sistema de autenticación básico que lo proporciona Loopback 3 y un mongodb como base de datos no relacional.

Por otro lado, he añadido las relaciones vía backend.

Como valor añadido, he añadido las siguientes características:

- campos de auditoría a los dos modelos de datos
- Un borrado lógico para evitar perder datos
- Población de datos si se detecta que no hay ningún dato vía un boot-script de loopback 3, en el fichero fixtures.js

Credenciales

Base de datos:

```
{  
  host: localhost,  
  port: 37017, ← (internamente en el api usa el puerto 27017),  
  database: employee-and-department,  
  user: db-user,  
  password: 12341234  
}
```

Webapp:

```
{  
  email: admin@gmail.com,  
  password: admin1234  
}
```