



Desenvolvimento Backend

Aula 09 - Utilizando o MySQL



Material Didático do Instituto Metrópole Digital - IMD

Termo de uso

Os materiais didáticos aqui disponibilizados estão licenciados através de Creative Commons **Atribuição-SemDerivações-SemDerivados CC BY-NC-ND**. Você possui a permissão para realizar o download e compartilhar, desde que atribua os créditos do autor. Não poderá alterá-los e nem utiliza-los para fins comerciais.

Atribuição-SemDerivações-SemDerivados

CC BY-NC-ND



<https://creativecommons.org/licenses/by-nc-nd/4.0/>

Apresentação

Nesta aula vamos começar a utilizar o banco de dados *MySQL* para armazenar os dados.

Objetivos

- Conhecer o conceito de variáveis de ambiente
- Conhecer a variável *NODE_ENV*
- Conhecer o uso do *cross-env*
- Conhecer como recorrer ao banco de dados *MySQL*.

Controlando ambientes com *NODE_ENV*

Link do video da aula: <https://youtu.be/QmzKQwxayZk>

Até o momento, estamos utilizando o banco de dados em arquivo, o que facilita muito o desenvolvimento. Porém, para colocar a aplicação no ar, é importante utilizar um banco de dados mais robusto.

Variáveis de Ambientes

São variáveis do sistema operacional usadas na aplicação para controlar o funcionamento dela. Nesse caso, será usada uma variável de ambiente que indica se a aplicação está em modo de desenvolvimento ou produção.

O Node.js já possui uma variável, chamada *NODE_ENV*, que ajuda a dizer em qual ambiente a aplicação está sendo executada. Atualmente, esse valor ainda não está definido, por isso, o *sequelize* utiliza o *development* como configuração padrão de banco de dados.

Para resolver isso, na próxima aula, teremos que mudar o valor da variável para *production* e configurar o *Sequelize* para utilizar o *MySQL*, neste caso.

Uso do *cross-env* para controlar o ambiente

Link do video da aula: <https://youtu.be/1ESlCr62hfQ>

Cada sistema operacional tem sua maneira de definir suas variáveis de ambiente e

isso pode gerar problemas futuros. Para contornar esse problema existe o *cross-env*, uma biblioteca no node que faz a definição independentemente do sistema que está sendo usado.

Utilizando o *cross-env*

Para utilizar a biblioteca, temos que antes fazer sua instalação. Então, execute no terminal o comando:

```
npm i cross-env
```

Agora, no *package.json* altere o *script* de *start* para definir o valor de *NODE_ENV*.

```
"scripts": {  
  "start": "cross-env NODE_ENV=development nodemon --exec node  
src/index.js",  
},
```

Daí em diante, a variável já possui o valor passado e pode ser utilizada.

Utilizando o valor de *NODE_ENV*

Atualmente, a variável armazena o valor de *development*. No entanto, isso será alterado para uso do *MySQL*. Antes, porém, é necessário alterar o arquivo de configuração do banco de dados para o ambiente de produção executar com o banco de dados desejado.

Para isso, é necessário instalar previamente o *MySQL* na máquina e após isso alterar o arquivo de configuração do banco de dados. Então, as configurações *production* do arquivo *config.json* devem ser alteradas para ficar como abaixo:

```
"production": {  
  "username": "root",  
  "password": "changeme",  
  "database": "dbend",  
  "host": "127.0.0.1",  
  "dialect": "mysql"  
}
```

Para utilizar esse campo *"dialect: mysql"*, é necessário fazer sua instalação antes, além da instalação prévia do *MySQL* citada acima. Então, execute no terminal o comando:

```
npm install --save mysql2
```

Configurando a inicialização da aplicação

Até o momento, a inicialização utiliza o valor de `NODE_ENV` como *development*. Por isso, agora será criado um *script* para iniciar a aplicação com um novo valor para essa variável de ambiente. A área de *scripts* deve, então, ficar como o modelo abaixo:

```
"scripts": {  
  "start": "cross-env NODE_ENV=development nodemon --exec node  
src/index.js",  
  "production": "cross-env NODE_ENV=production node src/index.js"  
},
```

Executando em produção com MySQL

Link do video da aula: <https://youtu.be/INSVNYZRZms>

Com o *MySQL* configurado, é necessário fazer apenas pequenos ajustes para finalizar o armazenamento nesse banco de dados.

Criando o banco de dados

Antes de utilizar o banco de dados, é necessário criá-lo no *MySQL*. Para isso, será utilizado o [MySQL Workbench](#), que é o *software* de gestão do sistema *MySQL*. Acompanhe a aula utilizando essa ferramenta.



Realizando as migrações

Para criar as tabelas no banco é necessário rodar as *migrations* no banco de produção. Uma forma de fazer isso é criar dois novos *scripts*: um para migração de desenvolvimento e outro para migração de produção. O arquivo *package.json* deverá ficar de acordo com o código abaixo:

```
"scripts": {  
  "start": "cross-env NODE_ENV=development nodemon --exec node  
src/index.js",  
  "production": "cross-env NODE_ENV=production node src/index.js",  
  "migrate-dev": "cross-env NODE_ENV=development npx sequelize-cli  
db:migrate",  
  "migrate-prod": "cross-env NODE_ENV=production npx sequelize-cli  
db:migrate"  
},
```

Execução

Por fim, rode no terminal o comando de migração:

```
npm run migrate-prod
```

Após isso, para rodar a aplicação em produção execute:

```
npm run production
```

Resumo

Nesta aula, foi visto o conceito de variáveis de ambiente e como usá-las para melhor manutenção do código. Além disso, foi apresentada especificamente a variável `NODE_ENV`, que servirá para informar em qual ambiente o código está executando. Também foi feito o uso de uma biblioteca chamada *cross-env* que contorna problemas de definição de variáveis de ambiente.