

node  
express



Manipulando dados com Node,  
Express e Mysql

Vamos criar nosso primeiro projeto com acesso ao Mysql: crie o código abaixo  
App.js

```
var express = require('express');  
var app = express();
```

```
app.get('/frontend', function(req,res){
```

```
  var mysql = require('mysql');
```

```
  var connection = mysql.createConnection({  
    host: "localhost",  
    user: "root",  
    password: "sua senha",  
    database: "recode"  
  });
```

Credenciais para  
acessar o banco  
de dados MySQL

```
  connection.query("SELECT * FROM conteudo WHERE modulo = 'front-end';", function(error, result){  
    res.send(result);  
  });
```

Comando SQL que vai  
devolver todos os campos  
da tabela conteúdo

```
});  
app.listen(3000, function(){  
  console.log("Servidor rodando com Express");  
});
```

Instale as seguintes  
dependências:

npm install express --save  
npm install mysql --save

No seu Mysql crie um banco de dados chamado RECODE e a tabela abaixo

Nome da tabela: **conteudo**

nome	modulo
HTML	front-end
CSS	front-end
Javascript	front-end
UI/UX	front-end
Bootstrap	front-end
Orientação a objetos	front-end
TypeScript	front-end
Angular	front-end
React	front-end
Virtualização com MS Azure	infraestrutura
Linux	infraestrutura
Apache e MySQL Server	infraestrutura
MySQL Client	back-end
NodeJS	back-end
Python	back-end
PHP	back-end

Após criar seu banco de dados e inserir os dados na tabela conteúdo volte ao seu projeto e digite **node app**, se tudo deu certo você terá o resultado abaixo no seu navegador:

<http://localhost:3000>

```
[{"id":1,"nome":"HTML","modulo":"front-end"},  
{"id":2,"nome":"CSS","modulo":"front-end"},  
{"id":3,"nome":"Javascript","modulo":"front-end"},  
{"id":4,"nome":"UI/UX","modulo":"front-end"},  
{"id":5,"nome":"Bootstrap","modulo":"front-end"},  
{"id":6,"nome":"Orientação a objetos","modulo":"front-end"},  
{"id":7,"nome":"TypeScript","modulo":"front-end"},  
{"id":8,"nome":"Angular","modulo":"front-end"},  
{"id":9,"nome":"React","modulo":"front-end"}]
```

## Renderizando os dados em formato HTML

```
var express = require('express');
var app = express();
app.set('view engine', 'ejs');

app.get('/frontend', function (req, res) {

  var mysql = require('mysql');

  var connection = mysql.createConnection({
    host: "localhost",
    user: "root",
    password: "120202",
    database: "recode"
  });

  connection.query("SELECT * FROM conteudo WHERE modulo = 'front-end';", function (error, result) {
    res.render('./frontend', { dados: result });
  });

});

app.listen(3000, function () {
  console.log("Servidor rodando com Express");
});
```

Para usar o template **ejs** precisamos indicar o módulo como view engine.

Será usado o método `render()` para renderizar o arquivo `frontend.ejs` que recebe o objeto `dados` com o retorno da consulta Sql através do parâmetro `result`

## Resultado do código anterior no navegador

### Tecnologias Front-End

HTML

CSS

Javascript

UI/UX

Bootstrap

Orientação a objetos

TypeScript

Angular

React

**Digite em seu navegador**

**<http://localhost:3000/forntend>**





## Exercício

Use o projeto modelo fornecido pelo professor **Flavio Mota** e faça as seguintes alterações.

Criar um novo arquivo de módulo na pasta **config**, chamado de **dbConnection.js**. Em seguida, o trecho de código de conexão deve ser movido para este módulo, e o módulo criado deve ser importado para os arquivos de rota.

Para conectar o NodeJS ao banco de dados, utilize como exemplo o arquivo de módulo da rota para o front-end (**./app/routes/frontend.js**).

### Resultado no navegador

## Tecnologias Front-End

HTML

CSS

Javascript

UI/UX

Bootstrap

Orientação a objetos

TypeScript

Angular

React