

## Trabalhando com o EJS view engine

Vamos criar o seguinte projeto usando EJS, Express e BootStrap

EJS - Recode Pro 2020   Home   Sobre

EJS view engine!

Bebidas

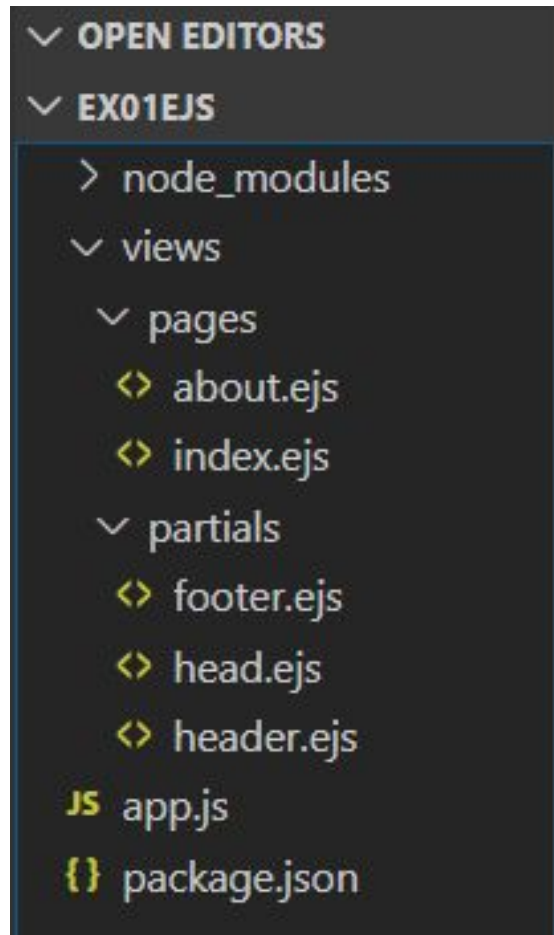
Cerveja	3
Refrigerante	5
Suco	10

© Copyright 2020

## Trabalhando com o EJS view engine

Vamos criar o seguinte projeto usando módulos, EJS, Express e BootStrap

### Estrutura de Arquivos



Para instalar as dependências digite:

`npm install express --save`

`npm install ejs --save`

## Trabalhando com o EJS view engine

### Vamos criar o seguinte projeto usando EJS, Express e BootStrap

Com tudo instalado, vamos configurar a aplicação para utilizar o EJS e definir nossas rotas para as duas páginas que precisaremos, index (*full width*) e about (*sidebar*). Faremos tudo isso em nosso arquivo *app.js*.

Aqui definimos que a aplicação utilizará a porta 8080 e que nossa *view engine* será o *EJS*. É importante lembrar que o *res.render()* irá procurar nossas páginas dentro do diretório *views*.

```
//define uma instância para o express
var express = require('express');
var app = express();

//define o EJS como nossa view engine
app.set('view engine', 'ejs');
//Usar res.render para carregar arquivos de view ejs
//index page
app.get('/', function(req, res){

  var bebidas =
  [{ nome: 'Cerveja', total: 3 },
  { nome: 'Refrigerante', total: 5 },
  { nome: 'Suco', total: 10 }];

  res.render('pages/index', {
    bebidas: bebidas
  });
});
//about page
app.get('/about', function(req, res){
  res.render('pages/about');
});

app.listen(8080);
console.log('8080 é a porta mágica');
```

## Trabalhando com o EJS view engine

Vamos criar o seguinte projeto usando EJS, Express e BootStrap

Partials: head.ejs, header.ejs, e footer.ejs

Como na maioria das aplicações que construímos, muito código é reutilizado. Vamos definir três *partials* que utilizaremos em todo nosso site: *head.ejs*, *header.ejs*, e *footer.ejs*.

*head.ejs*

```
<!-- views/partials/header.ejs -->
<nav class="navbar navbar-default" role="navigation">
  <div class="container-fluid">

    <div class="navbar-header">
      <a class="navbar-brand" href="#">
        EJS - Recode Pro 2020
      </a>
    </div>

    <ul class="nav navbar-nav">
      <li><a href="/">Home</a></li>
      <li><a href="/about">Sobre</a></li>
    </ul>

  </div>
</nav>
```

## Trabalhando com o EJS view engine

Vamos criar o seguinte projeto usando EJS, Express e BootStrap

Partials: head.ejs, header.ejs, e footer.ejs

Como na maioria das aplicações que construímos, muito código é reutilizado. Vamos definir três *partials* que utilizaremos em todo nosso site: *head.ejs*, *header.ejs*, e *footer.ejs*.

### *header.ejs*

```
<!-- views/partials/head.ejs -->
<meta charset="UTF-8">
<title>Template com EJS</title>
<!-- Bootstrap CSS -->
<link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
<style>
  body { padding-top:50px; }
</style>
```

## Trabalhando com o EJS view engine

Vamos criar o seguinte projeto usando EJS, Express e BootStrap

Partials: head.ejs, header.ejs, e footer.ejs

Como na maioria das aplicações que construímos, muito código é reutilizado. Vamos definir três *partials* que utilizaremos em todo nosso site: *head.ejs*, *header.ejs*, e *footer.ejs*.

*footer.ejs*

```
<!-- views/partials/footer.ejs -->  
  
<p class="text-center text-muted">© Copyright 2020</p>
```



## Trabalhando com o EJS view engine

Vamos criar o seguinte projeto usando EJS, Express e BootStrap

### Usando partials EJS

Agora que temos nossas *partials* definidas, tudo o que precisamos é carregar os arquivos sempre que necessário. Para incluir uma *partial* basta digitar o seguinte código:

**<% include NOME\_ARQUIVO %>**

*index.ejs*

```
<!DOCTYPE html>
<html lang="en">
<head>
  <% include ../partials/head %>
</head>
<body class="container">
  <header>
    <% include ../partials/header %>
  </header>
  <main>
    <div class="jumbotron">
      <h1>EJS view engine!</h1>
    </div>
    <h2>Bebidas</h2>
    <table class="table table-bordered">
      <tbody>
        <% bebidas.forEach(function(bebida){ %>
          <tr>
            <td><%= bebida.nome %></td>
            <td><%= bebida.total %></td>
          </tr>
        <% }); %>
      </tbody>
    </table>
  </main>
  <footer>
    <% include ../partials/footer %>
  </footer>
</body>
</html>
```

## Trabalhando com o EJS view engine

Vamos criar o seguinte projeto usando EJS, Express e BootStrap

### Usando partials EJS

Agora que temos nossas *partials* definidas, tudo o que precisamos é carregar os arquivos sempre que necessário. Para incluir uma *partial* basta digitar o seguinte código:

<% include NOME\_ARQUIVO %>

*about.ejs*

```
<!DOCTYPE html>
<html lang="en">
<head>
  <% include ../partials/head %>
</head>
<body class="container">
  <header>
    <% include ../partials/header %>
  </header>
  <main>
    <div class="row">
      <div class="col-sm-8">
      </div>
      <div class="col-sm-4">
        <div class="well">
          <h3>Recode Pro 2020 - Professores Flávio Mota / Flavio Sousa</h3>
        </div>
      </div>
    </div>
  </main>
  <footer>
    <% include ../partials/footer %>
  </footer>
</body>
</html>
```



## Trabalhando com o EJS view engine

Vamos criar o seguinte projeto usando EJS, Express e BootStrap

Passando dados para a view

*app.ejs*

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

  var bebidas =
  [{ nome: 'Cerveja', total: 3 },
  { nome: 'Refrigerante', total: 5 },
  { nome: 'Suco', total: 10 }];

  res.render('pages/index', {
    bebidas: bebidas
  });
});
```

*index.ejs*

```
<h2>Bebidas</h2>
<table class="table table-bordered">
  <tbody>
    <% bebidas.forEach(function(bebida){ %>
      <tr>
        <td><%= bebida.nome %></td>
        <td><%= bebida.total %></td>
      </tr>
    <% }); %>
  </tbody>
</table>
```

Essa lista será utilizada na página index

## Desafio 60minutos

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

var server = http.createServer(function(req, res){
  var categoria = req.url;

  if(categoria == '/front-end'){
    res.end("<html><body>Tecnologias Front-End: TypeScript, Angular, React.</body></html>");
  }else if(categoria == '/back-end'){
    res.end("<html><body>Tecnologias Back-End: NodeJS, Python, PHP, MySQL...</body></html>");
  }else if(categoria == '/infraestrutura'){
    res.end("<html><body>Azure Cloud, Linux, MySQL Server...</body></html>");
  }else{
    res.end("<html><body>Programador Full Stack</body></html>");
  }
});

server.listen(3000);
```

### FOLDERS

- projeto\_node\_recode
  - app
    - routes
      - /\* backend.js
      - /\* frontend.js
      - /\* home.js
      - /\* infraestrutura.js
    - views
      - home
        - index.ejs
      - secao
        - backend.ejs
        - frontend.ejs
        - infraestrutura.ejs
    - config
      - /\* server.js
    - node\_modules
      - /\* app.js
      - /\* mod\_teste.js
      - /\* package-lock.json
      - /\* package.json

Organize o projeto antigo como a estrutura  
ao lado, usando módulos, express e ejs