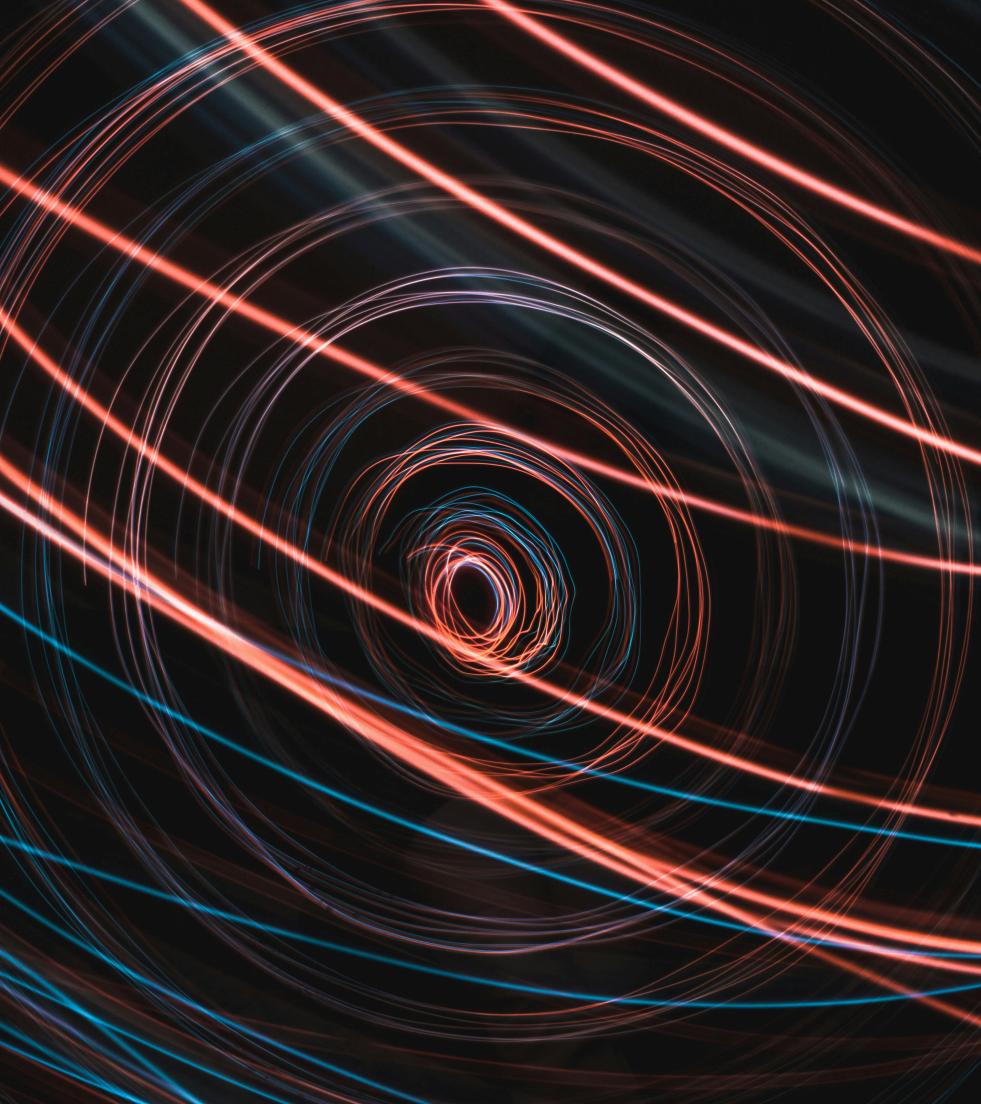


Templates

Programação Web 1



Objetivos de Aprendizagem

- O que são *Template Engines*?
- Instalar o *Template Engine* EJS
- Utilizar funções básicas de *templates* em geral

Agenda

- *Template engines*
- Instalação
- Exemplo de aplicação com EJS

Template Engines

O que são *Template Engines*?

Template engines são especialmente úteis em projetos que requerem a geração de HTML **dinamicamente**. Permitem que os desenvolvedores insiram dados em *templates* (modelos) pré-definidos, resultando em páginas web personalizadas e desenvolvidas rapidamente.

Exemplos

Com Express podem ser utilizados diversos TE.

- Pug 
- Handlebars 
- EJS

EJS

Embedded JavaScript templating

EJS is a simple templating language that lets you generate HTML markup with plain JavaScript.

Instalação

Como usar o EJS?

- Para aplicar o *template engine* EJS há dois caminhos
 1. Com Express Generator
 2. Sem o Express Generator

Instalação

EJS com Express Generator

- `npm install express express-generator` (Rever Aula 7)
- Para utilizar o EJS com Express, bastam duas linhas de configuração no `app.js`:

```
1 // define o caminho a partir de /views
2 // válido apenas com o Express Generator
3 app.set('views', path.join(__dirname, 'views'));
4 // define o EJS como template engine
5 app.set('view engine', 'ejs');
```

Instalação

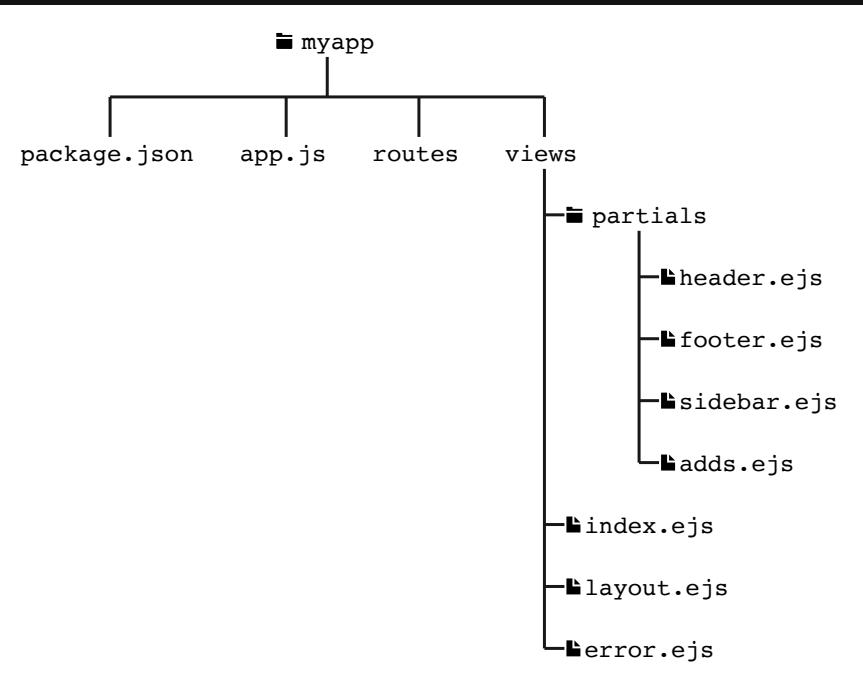
Sem Express Generator

- Os *templates* devem estar em `/views`
- Note que nas linhas 5 e 9 os caminhos são especificados a partir de `/views`

```
1 app.set('view engine', 'ejs');
2 // res.render carrega o template ejs especificado o uso da extensão é opcional
3 app.get('/', function(req, res) {
4     res.render('index');
5 });
6 // about
7 app.get('/about', function(req, res) {
8     res.render('about');
9 });
10 app.listen(8080);
11 console.log('8080 is the magic port');
```

Organização dos arquivos

- Todos os *templates* devem estar organizados em `/views`
- É interessante organizar em subdiretórios para melhor organização, como na figura



/partials

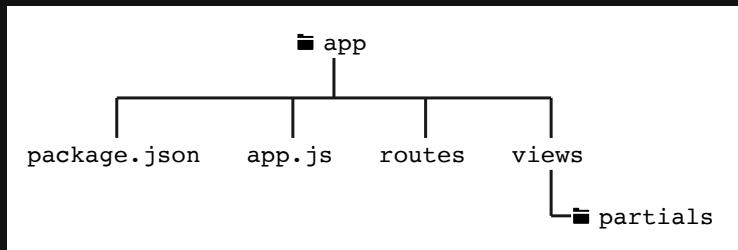
- A palavra chave quando se usa *templates* é **REUTILIZAÇÃO**
- O diretório `/partials` (sugestão de nome) deve conter tudo aquilo que for reutilizável:
 - Cabeçalhos (*headers*)
 - Rodapés (*footers*)
 - Barras de menu (*menu*)
 - *Sidebars*
 - Anúncios (*adbars*)

Exemplo de aplicação com EJS

Exemplo de aplicação

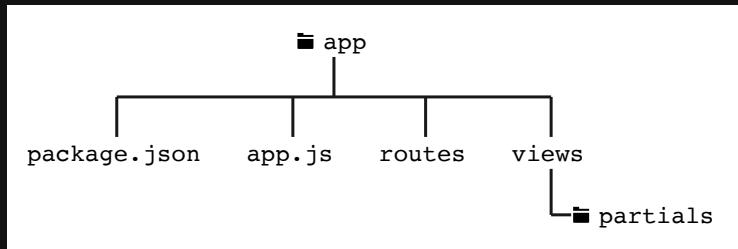
1. Crie uma aplicação com Express Generator ou crie uma aplicação com a estrutura de arquivos e diretórios abaixo.

- `npm init -y`
- `npm install express express-generator`



Exemplo de aplicação

- Execute o EG
 - `npx express-generator --view=ejs app`, caso o diretório `app` **não tenha sido criado**
 - **OU** `npx express-generator --view=ejs`, caso o comando esteja sendo executado dentro do diretório (**já criado**) da aplicação



Exemplo de aplicação

2. Caso opte pelo EG a estrutura base já será criada.
3. Em `/views` crie o diretório `/partials`
4. Em `/partials` crie os arquivos `footer.ejs` e `header.ejs` (a seguir)

header.ejs

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Random Title</title>
6      <!-- CSS link (loading Semantic UI from a CDN) -->
7      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/seman
8      <style>      body { padding:50px; }      </style>
9  </head>
10 <body class="container">
11 <header>
12     <div class="ui inverted menu">
13         <a class="item" href="/">Home</a>
14         <a class="item" href="/magic">Magic</a>
```

footer.ejs

```
1 <footer>
2   <div class="ui inverted footer">
3     <div class="ui container">
4       Topcoder 2021. All Rights Reserved
5     </div>
6   </div>
7 </footer>
```

Exemplo de aplicação

Como usar as partials?

5. Para inserir o `header.ejs` na `view` (Linha 1) em `index.ejs`
6. Para inserir o `footer.ejs` na `view` (Linha 8)

```
1 <%- include('./partials/header') %>
2   <main>
3     <div class="ui segment">
4       <h1>Index Route</h1>
5       <p>This is the INDEX route</p>
6     </div>
7   </main>
8   <%- include('./partials/footer') %>
9
10 </body>
11 </html>
```

magic.ejs

7. Em `/views` crie o arquivo `magic.ejs`

```
1 <%- include('./partials/header') %>
2   <main>
3     <div class="ui segment">
4       <h1>Magic Route</h1>
5       <p>This is a magic route</p>
6     </div>
7   </main>
8   <%- include('./partials/footer') %>
9
10 </html>
```

Exemplo de aplicação

8. Em `/routes` criar o arquivo `magic.js`

```
1 var express = require('express')
2 var router = express.Router()
3
4 router.get('/', function(req, res, next) {
5   res.render('magic')
6 })
7
8 module.exports = router;
```

Exemplo de aplicação

9. Incluir a rota `/magic` em `app.js` nas linhas indicadas

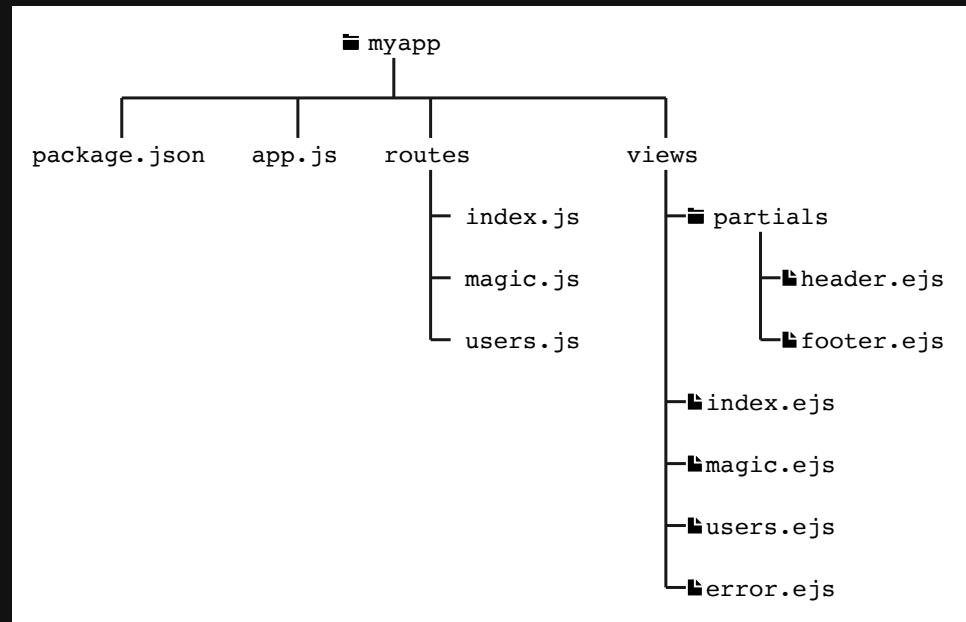
```
9 var magicRouter = require('./routes/magic')
```

```
25 app.use('/magic', magicRouter)
```

10. Teste todas as rotas

Exemplo de aplicação

Estrutura final



Perguntas

Exercício

1

Refazer o exercício #3 da Aula 7 (Atividade Router) adicionando um *template* para cada uma das rotas. Utilize um cabeçalho e um rodapé compartilhado em um diretório `/partials`

Referências

- Embedded JavaScript templating
- Using EJS Template Engine With Express.js

José Roberto Bezerra

✉️ jbroberto@ifce.edu.br

⌚ [jbroberto76](#)

⌚ [pw1-repo](#)



Powered by  Sliddev

Cover image by José Campos from 