Modulo 05

Projeto #01 Criando uma plataforma de perguntas e respostas com Mysql

* Aula 02 versao Boot

<https://getbootstrap.com/docs/4.5/getting-started/introduction/>

* Aula 03 EJS e criação de projeto

Criamos uma nova pasta chamada guiaPerguntas

Vamos iniciar um novo projeto node nessa pasta pelo cmd

npm init -y

vamos instalar o express e o ejs(engine template):

npm install express --save

npm install ejs --save

* Aula 04 Configurando e exibindo HTML com EJS

No arquivo index.js coloque o código

app.set('view engine', 'ejs')

crie uma pasta views em seu projeto, onde ficará todos os seus arquivos html

como estamos ultilizando o EJS, na estensão iremos usar o .ejs

crie um arquivo index.ejs na pasta views para ser nossa pagina principal

agora no arquivo index.js

iremos setar o caminho ‘/’, como sendo res.render e não res.send, dizendo o nome do arquivo que está na pasta views, nesse caso o arquivo index.ejs

index.js

app.get('/', (req, res) => {

    res.render('index')

})

Ao utilizar o render ele já vai direto na pasta views, e como se estivesse passando

app.get('/', (req, res) => {

    res.render('views/index')

})

Todo arquivo html deverá ficar na pasta views de seu projeto

* Aula 05 Exibindo variáveis no HTML

Com o EJS e possível passar variáveis do javascript para dentro do html

app.get('/', (req, res) => {

    let nome = 'Rodrigo'

    let lang = 'JavaScript'

    res.render('index')

})

Agora para passar esses valores vamos utilizar essa variáveis dentro do res.render()

index.js

app.get('/', (req, res) => {

    let nome = 'Rodrigo'

    let lang = 'JavaScript'

    res.render('index', {

        nome: nome,

        lang: lang,

        empresa: 'Guia do programador',

        inscrito: 8000

    })

})

Passamos para a view index 4 variaveis(nome, lang, empresa e inscrito)

Agora vamos pegar essa variáveis no arquivo index.ejs, usando a tag

<%= %>, passando o nome da variável

index.ejs

<body>

    <%= nome %>

    <%= lang %>

    <%= empresa %>

    <%= inscrito %>

</body>

Outra coisa que pode fazer e misturar essas tag com html

index.ejs

<body>

    <p>Nome: <%= nome %> </p>

    <p>Linguagem favorita: <%= lang %> </p>

    <p>Empresa: <%= lang %> </p>

    <p>Incritos: <%= inscrito %> </p>

</body>

Passagem de parâmetros obrigatorios direto no url

index.js

app.get('/:nome/:lang', (req, res) => {

    let nome = req.params.nome

    let lang = req.params.lang

    res.render('index', {

        nome: nome,

        lang: lang,

    })

})

Os parâmetros obrigatórios passados ficarão salvo em variáveis, que serão chamado no view index

Passagem de parâmetro opcinal

index.js

app.get('/:nome?/:lang?', (req, res) => {

    let nome = req.params.nome

    let lang = req.params.lang

    res.render('index', {

        nome: nome,

        lang: lang,

    })

})

Aqui temos dois parâmetros opacionais nome e lang, que podem ou não ser passado no endereço.

Se no endereço por passado paramentro ficara salvo na variável, se não for passado nada, nada ira ser mostrado

* Aula 06 Estrutura condicionais com EJS

<%= %> -> para chamar variaveis

<% if(){ %> -> para usar expressões

<% } %>

Ou

<% if(){ %> -> para usar expressões

<% } else { %>

<% } %>

index.js

app.get('/:nome/:lang', (req, res) => {

    let exibirMsg = true

    res.render('index', {

        msg: exibirMsg

    })

})

index.ejs

<body>

    <% if(msg == true){ %>

        <h3>Isso é uma msg de ERRO!</h3>

    <% }else { %>

        <h3>Nenhum erro!</h3>

    <% } %>

</body>

Nesse caso como a variável msg esta recebendo exibirMsg que por sua vez e true, será mostrada a msg de erro, de dentro do bloco