Aula 4 - Padrões

Tópicos especiais em Sistemas

Prof. Juliana Costa Silva - juliana.silva@up.edu.br

O que veremos hoje

1. Revendo...
2. Introdução
3. Pacotes/ módulos
4. Export
5. Módulo de configuração
6. Requisição POST
    - 6.1 Postman

Revendo...

O que já aprendemos?

- Criamos um projeto node;
- Configuramos a porta do através do arquivo **index.js**;
- Definimos rotas de get da nosso aplicação;
- Configuramos nodemon no ambiente dev;

O projeto da disciplina

- Faremos um sistema de controle financeiro pessoal;
- Este sistema deve ter:
  - Registro de gastos;
  - Login de usuários;
  - Registro de renda (salários - comissões - negócios);
  - Registro de cartões de créditos;
  - Registro de contas bancárias;

Organização

Qual a função do index.js?

- Inicia o servidor;
- Configura porta a escutar;
- Define rotas get de atendimentos e outras;

Falta organização ao projeto?

Controller

Vamos criar a pasta **controller**.

- Esta pasta será responsável por organizar e gerenciar as ações da nossa aplicação.
- Crie, dentro dessa pasta um arquivo chamado **movimento.js**, é nele que iremos controlar a rota app.get;

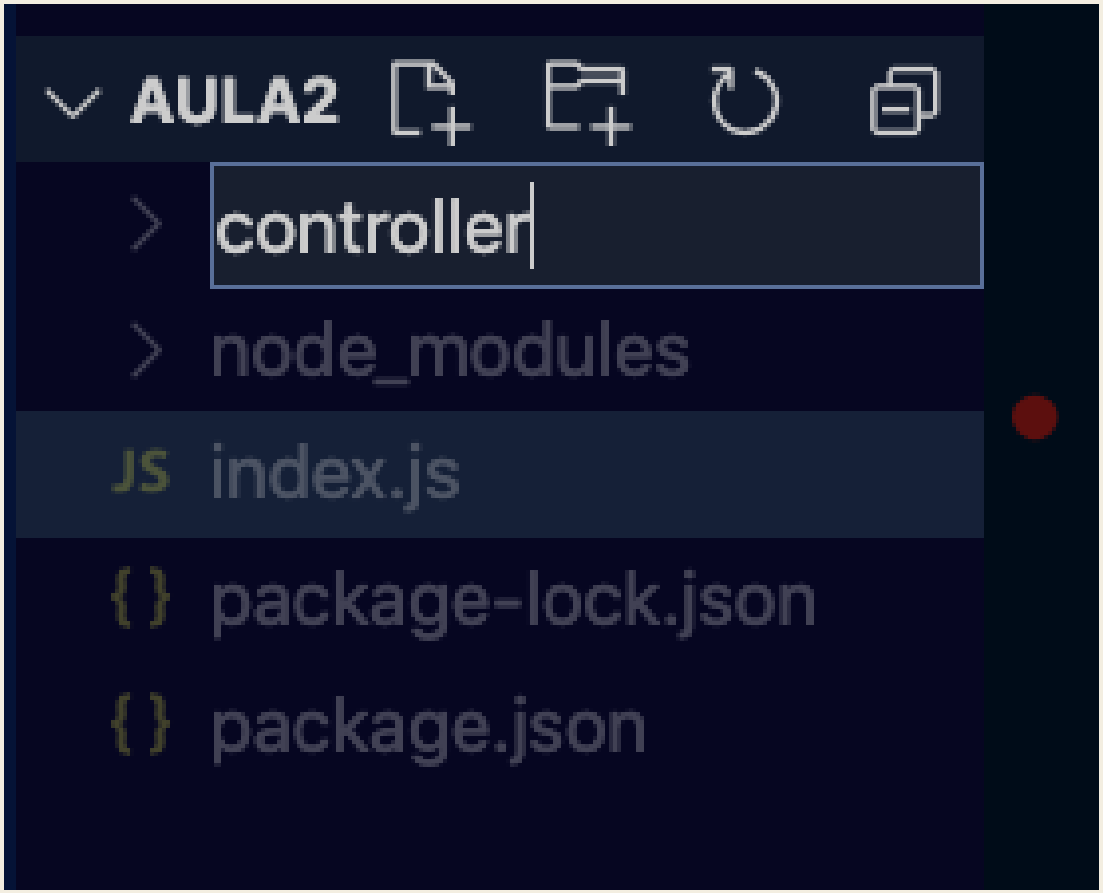
Mova a linha 7 do arquivo **index.js**, conforme apresentado abaixo, para **movimento.js**.

```
7   app.get('/gastos', (req, res) => res.send('Você está em GASTOS com um GET'))
```

Fonte: O autor

Nova estrutura do Projeto

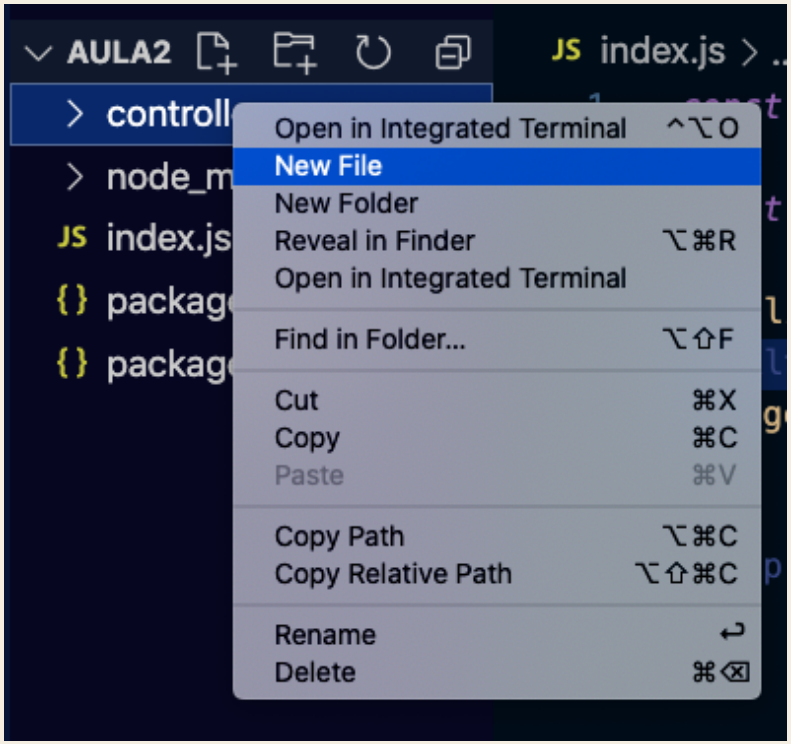
Crie a pasta controller:



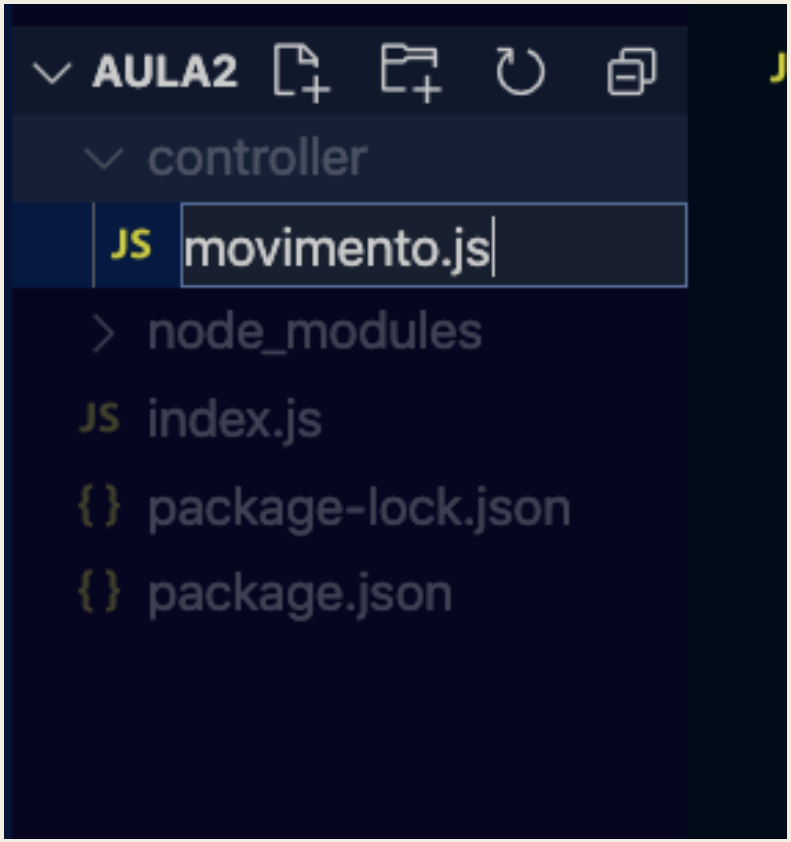
Fonte: O autor

Criando ”Moviemnto.js”

Dentro da pasta controller, crie o arquivo movimento.js:



Passo 1: clique com botão direito sobre a pasta controller e selecione Ńew File”. **Fonte:** O autor



Passo 2: Na caixa de diálogo digite o nome e a extensão do arquivo. **Fonte:** O autor

Funciona?

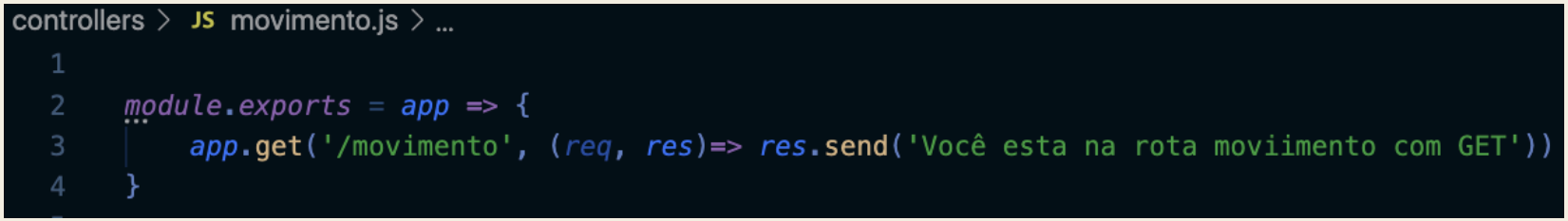
Importações necessárias

- Se executar novamente o projeto, na rota movimento nada será exibido;
- Isto é, **movimentos.js** deixou de ser reconhecido;
- O que ocorreu porque não exportamos este módulo (quando criamos arquivos separados, criamos módulos).

Export

Exportando módulo

- Essa função exporta o comando get configurado;
- Recebe por parâmetro app.



Função module.exports no arquivo movimento.js. **Fonte:** O autor

Consign

Para nos auxiliar nas exportações

- Vamos instalar (via terminal) o consign, que vai agrupar todas as rotas e associa-las ao app.
- Use o comando npm install consign no terminal;

Controllers e app

- Precisamos informar a aplicação que os arquivos de ćontrole”do nosso app, ficarão reunidos na pasta **controllers**
- Para isso utilizaremos um objeto consign;
- Associaremos a ele todo o conteúdo da pasta **controllers** então nossa aplicação verá todas as rotas criadas por lá.

Este código deve ser inserido no arquivo **index.js**, antes da utilização de app. **Fonte:** O autor

Organizando [2]

Nosso arquivo **index.js** ainda tem muitas funções....

Será possível melhorar isso?

index.js

```
1   const express = require('express')
2   const app = express()
3   const consign = require('consign')
4
5   consign()
6       .include('controllers')
7       .into(app)
8
9   app.listen(3000,() => console.log('servidor rodando na porta 3000'))
10
```

Código no arquivo **index.js**. **Fonte:** O autor

Organizando [3]

- Crie uma pasta **config**, vamos separar todas as configurações do nosso app em um módulo;
- nessa pasta crie o arquivo **customExpress.js**.
- Recorte as linhas 1 a 7 do arquivo **index.js** e cole em **customExpress.js**.
- Essas linhas se referem a configuração da nossa aplicação;

Código no arquivo **index.js**. **Fonte:** O autor

customExpress.js

Para garantir que o objeto app exista no arquivo **index.js**, criaremos um module.export e exportaremos a função, que retorna a variável app;

Código no arquivo **customExpress.js**. **Fonte:** O autor

Novo index.js

Agora, no index.js importaremos customExpress, ao invés de express.

customExpress.js | index.js

index.js > ...

```
1   const custemExpress = require('./config/customExpress')
2
3   const app = custemExpress()
4
5   app.listen(3000, () => console.log('Servidor rodando na porta 3000'))
```

Código no arquivo **index.js**. **Fonte:** O autor

movimentos.js

Em **movimentos.js**, como fazer para receber dados do usuário?
Configuraremos o POST de nossa rota!

```
6    app.post('/movimento', (req, res)=> res.send('Você esta na rota moviimento com POST'))
```

Código no arquivo **movimentos.js**. **Fonte:** O autor

Ao executar a aplicação notamos que nada muda....

Postman

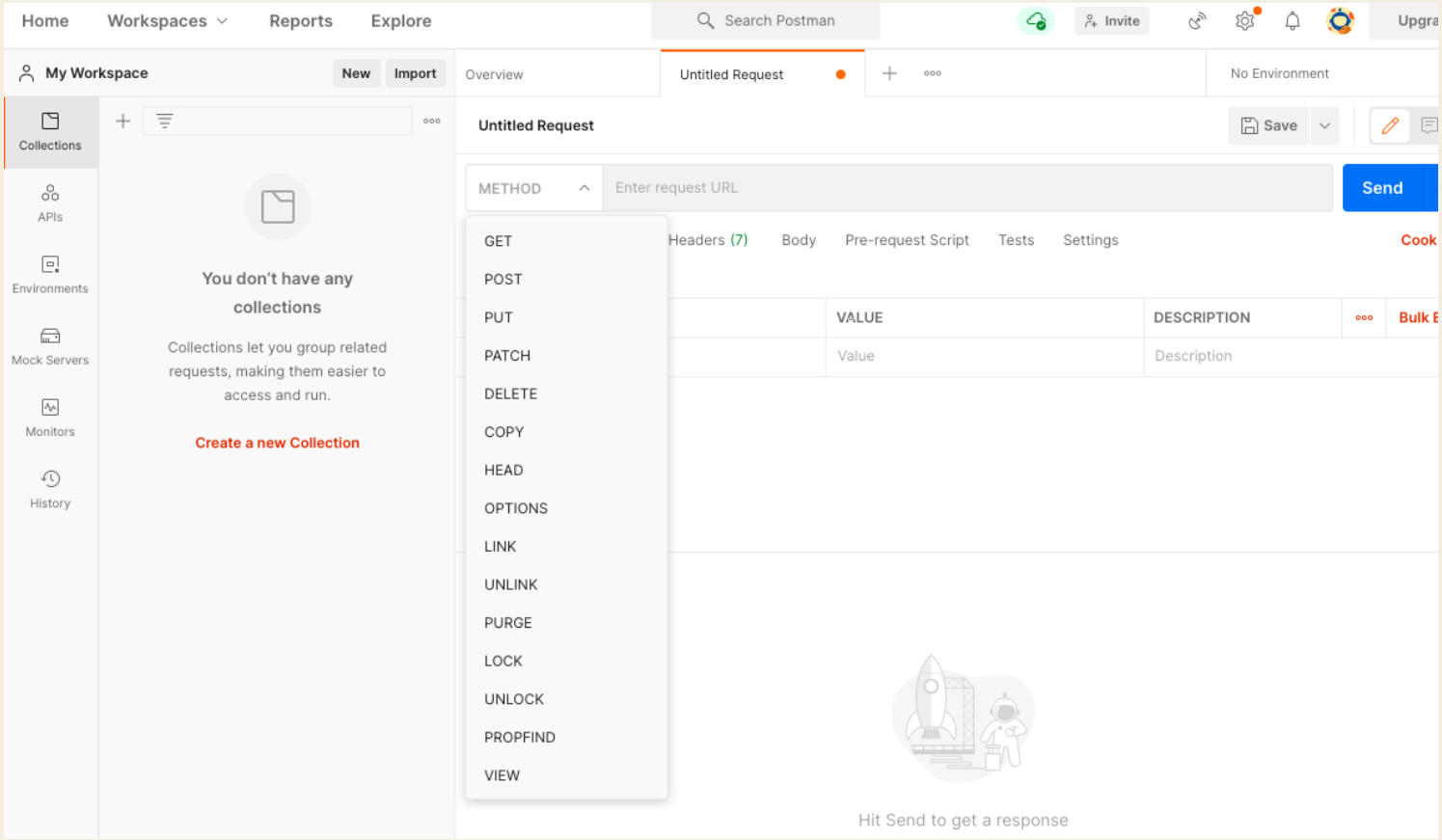
O Postman é uma ferramenta que dá suporte à documentação das requisições feitas pela API. Ele possui ambiente para a documentação, execução de testes de APIs e requisições em geral.

Logo Postman. **Fonte:** POSTMAN, 2020

Acesse https://www.postman.com/ e, faça o download da ferramenta.

Postman - GUI

Aqui poderemos testar vários tipos de requisições feitas ao servidor.



Visual GUI Postman. **Fonte:** POSTMAN, 2020