

## Lista de exercícios

1- Na aula passada, implementou-se um servidor extremamente simples utilizando-se o módulo 'http' do próprio Node. Antes de prosseguir com servidores, proponho um exercício para familiarização com o esquema de módulos do Node. A tarefa é simples: você deve importar três módulos do próprio node, a saber 'fs', 'crypto' e 'net'. Crie variáveis para cada módulo com o respectivo nome. Antes de prosseguir com o exercício, observe o exemplo em anexo na pasta Exercício 1/exemplo.js.

2- Uma vez importados os módulos, resta aprender a criar módulos. Você pode produzir módulos ao se criar um novo arquivo e exportar um objeto do JavaScript utilizando-se um objeto chamado module. Mas antes de se exportar módulos, relembremos que funções, objetos propriamente ditos, constantes, strings, arrays, números são todos objetos do JavaScript. Normalmente em módulos, exporta-se uma função ou um objeto do JavaScript, como ilustrado no arquivo Exercício 2/exemplo.js. A tarefa é exportar uma função do JavaScript que receba dois números e retorne a multiplicação deles.

3- Agora resta poder utilizar os módulos criados. Importe o módulo criado no exercício 2 e imprima a multiplicação de alguns números, como 14 e 98, 45 e 23, e.g. . Há uma pasta em Exercício 3 com um exemplo, o qual serve de ajuda.

4- Já importamos, criamos, agora vamos baixar novos módulos utilizando o npm. Baixaremos o módulo **nodemon**, que serve para executar continuamente um app do Node. Para isso, utilizaremos o npm install.

Explicando rapidamente o npm: quando se quer instalar um módulo, utiliza-se o **npm install <nome do módulo>**; quando esse módulo deve ser executado diretamente do console/terminal/linha de comando/cmd, adicionamos um -g antes ou depois do nome do módulo, e.g. **npm install <nome do módulo> -g** ou **npm install -g <nome do módulo>**.

Para este exercício, você deve instalar o módulo **nodemon**, o qual deve ser executado de qualquer pasta do sistema e diretamente do cmd. Há um exemplo dentro da pasta Exercício 4/Exemplo que possui os comandos a serem executados para se instalar o módulo **express** e o módulo **forever**, os quais não são a tarefa principal, mas servem de aprendizado para a tarefa principal. A resposta para instalar o **nodemon**, a tarefa principal, está dentro da pasta Exercício 4.

Para testar se funcionou, você deve rodar o comando **nodemon <arquivo a ser iniciado>**. Há um exemplo dentro da pasta Exercício 4/Exemplo que mostra como rodar o **nodemon** no arquivo index.js. Para cessar a execução do **nodemon** é suficiente apertar Ctrl-C.

Em seguida, devemos rodar o **nodemon** em dois arquivos na pasta Exercício 4, a saber: **sem\_errores.js** e **com\_errores.js**. A segunda parte da tarefa consiste em executar cada um, **utilizando-se o nodemon**. O primeiro vai rodar normalmente e sem erros, já o segundo possui um erro de **sintaxe**. A linha com erros está indicada. Peço que **não** cessem a execução do segundo e corrijam a linha com erro **com o nodemon executando**. Observem o que acontece e como é bizurado o **nodemon**.

5- O último exercício foi bem suga, mas o próximo não será. A próxima tarefa é para familiarizá-los com o HTTP. A proposta é que vocês enviem para o cliente uma mensagem de **erro**, afirmando que o que ele procura não está no seu servidor. Para isso, você deve mandar um código de status com numeração 404 e mandar uma mensagem explicitando que o arquivo não pôde ser encontrado. Utilize uma combinação do código do servidor básico da última aula e do exemplo fornecido em Exercício 5/Exemplo. Uma possível resposta encontra-se em Exercício 5/. Para verificar se funcionou corretamente, utilize o Firebug/Terminal do Chrome para observar a resposta do servidor. No caso, clique com o botão direito no Firefox, depois aperte Q e em seguida vou em Network.

Pressiono Reload e observo os pacotes que foram transmitidos para o navegador.

6- Agora vamos a um exercício bem interessante sobre HTTP. Existem algumas maneiras de se enviar dados para um servidor HTTP. Entre elas, estão os **métodos** GET e POST. Por enquanto, apenas enviamos requisições(req) do tipo GET. Faremos, portanto, uma requisição do tipo POST no servidor.

No entanto, não discriminamos o tipo de requisições no servidor que fizemos até agora, e para termos certeza de que o servidor realmente está respondendo às requisições POST, enviaremos páginas diferentes, a depender do tipo de requisição. A tarefa é simples: enviar duas páginas diferentes para o cliente a depender do tipo de requisição que for feita. Caso seja POST, envie uma mensagem de bem vindo. Do contrário, um 404 Not Found.

Há um exemplo na pasta Exercício 6/Exemplo que mostra como discriminar o tipo de requisição. Há também uma página em Exercício 6, a qual deve ser aberta no seu navegador favorito para fazer uma requisição POST para o servidor - o qual deve funcionar **localmente** na porta 8000. Uma resposta possível está em Exercício 6