Desenvolver aplicações Backend para WEB

Prof. Renan Ponick

Senac

Agenda

17/08/2023 - Postman + Post

18/08/2023 - Exercícios + Review

24/08/2023 - Teste Unitário

25/08/2023 - Exercícios + Testes

31/08/2023 - Review

01/09/2023 - Exercícios + Testes

07/09/2023 - FERIADO

08/09/2023 - FERIADO

14/09/2023 - Boas Práticas + Arq. REST e RESTFull

15/09/2023 - Exercícios + Testes

21/09/2023 - MVC?

22/09/2023 - CRUD

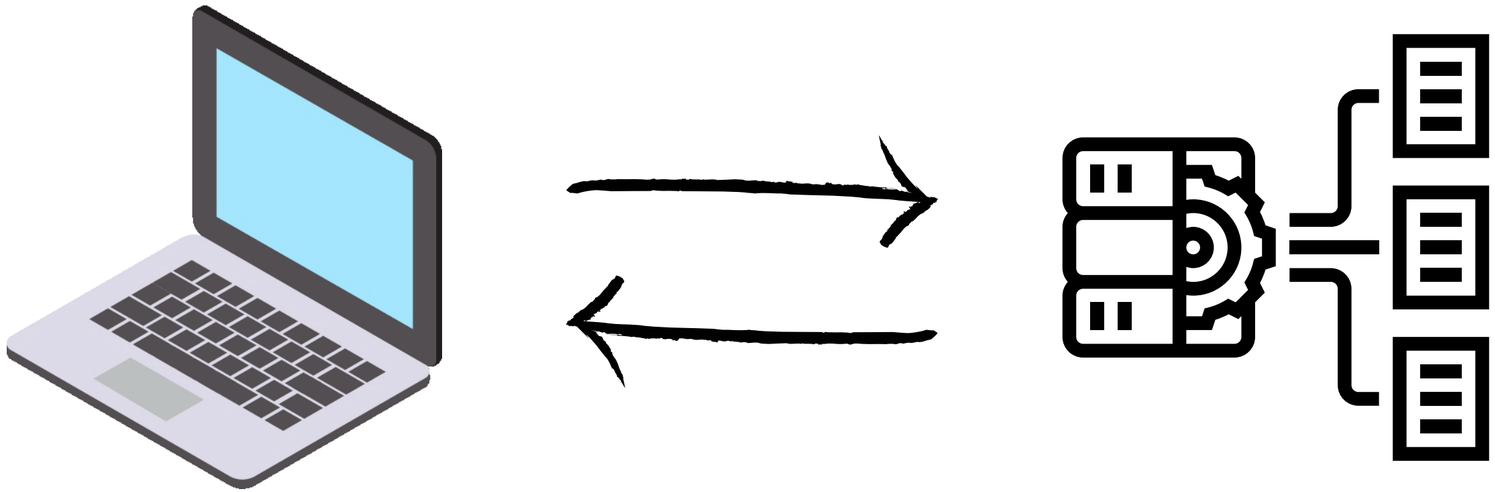
29/08/2023 - MySQL - Async - Await

29/09/2023 - Exercícios + Testes

Cliente e Servidor

Papeis

O front-end é responsável por fornecer a interface do usuário e interagir com o usuário, enquanto o back-end fornece a funcionalidade central, gerencia o banco de dados e realiza a lógica do aplicativo.

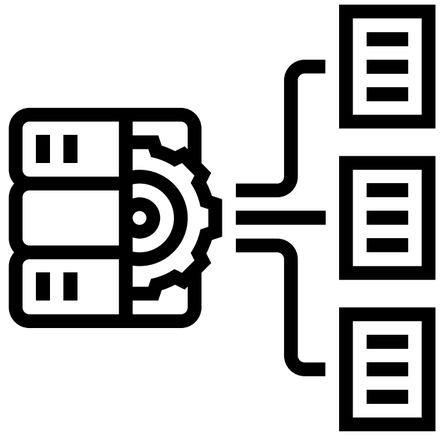


lado do cliente **lado do servidor**

REQUEST

lado do cliente

JSON | | XML

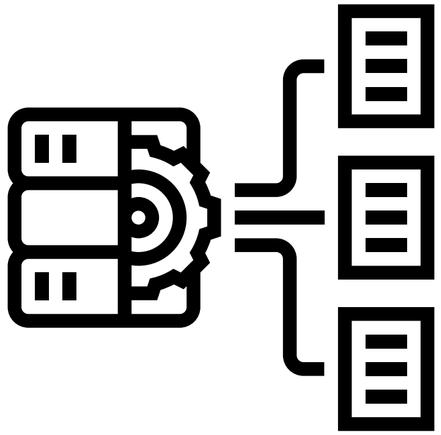


lado do servidor

RESPONSE

API

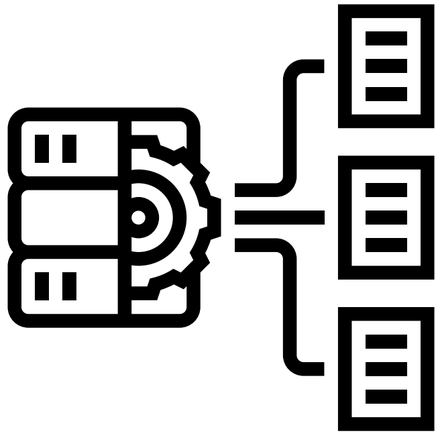
A partir de agora o nosso objetivo é construir APIs (Interfaces de Programação de Aplicativos) backends, que irão receber requisições, processar e responder as mesmas



lado do servidor

API

Ou seja, o nosso programa ficará hospedado em algum lugar, esperando receber essas requisições, mas como as requisições chegam ate ele?

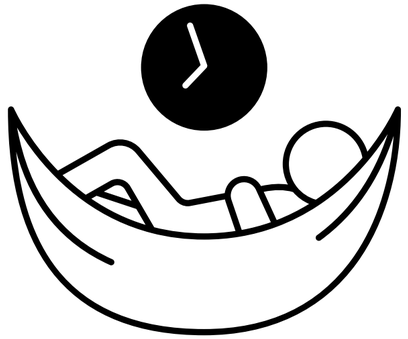


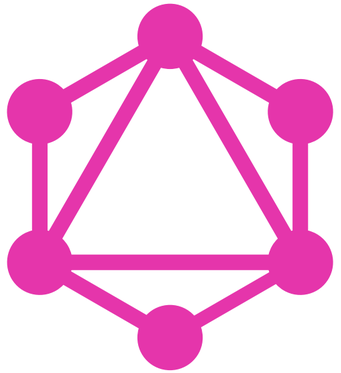
lado do servidor

Comunicação

Comunicação

Para que seja possivel estabelecer uma comunicação limpa entre o Frontend e o Backend, existem padrões de comunicação utilizados pelos desenvolvedores, sendo os mais conhecidos:





REST

Representational State Transfer (Transferência de Estado Representacional) e é um tipo de arquitetura de software.

Arquitetura cliente-servidor: indica uma arquitetura baseada em clientes, servidores e recursos, em que as solicitações são feitas via protocolo HTTP

HTTP

A abreviatura de “Hypertext Transfer Protocol” descreve um protocolo sem estado com o qual os dados podem ser transmitidos em uma rede IP

Rotas HTTP/HTTPS

- Get
  - Utilizada para recuperar uma lista ou um unico recurso.
- Post
  - Utilizada para criar um novo recurso.
- Put
  - Utilizada para atualizar um recurso
- Delete
  - Utilizada para excuir um recurso.

Quer melhorar seu código? Pesquise por padrão REST

E a linguagem?

JavaScript

Com o surginmento do Node.js, JavaScript passou a ser usado tanto no frontend quanto no backend.

Senac

Lib

Lib

Lib é uma abreviação comum para "library" (biblioteca), elas normalmente possuem um conjunto de funções, classes e métodos pré-criados que podem ser usados para realizar tarefas específicas de maneira mais eficiente e conveniente.

Express

É um framework de aplicativo web para Node.js. Ele simplifica muito o processo de criação de aplicativos web e APIs usando Node.js, fornecendo uma estrutura robusta e flexível para lidar com várias tarefas relacionadas ao **servidor** e ao **roteamento**.

E na Prática??

Inicializando Projeto

Terminal VSCode

Execute **npm init** e preencha as informações do seu projeto, para deixar tudo default basta dar enter e digitar yes no final, por exemplo:

```
renan@ng0502:~/Documents/senac/JP/javascript$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (javascript)
version: (1.0.0)
description:
entry point: (exercicio1.js)
test command:
git repository: (https://github.com/renanponick/javascript.git)
keywords:
author:
license: (ISC)
About to write to /home/renan/Documents/senac/JP/javascript/package.json:

{
  "name": "javascript",
  "version": "1.0.0",
  "description": "0. Faça um programa (FUP) que escreva na tela \"Hello World!\"",
  "main": "exercicio1.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "repository": {
    "type": "git",
    "url": "git+https://github.com/renanponick/javascript.git"
  },
  "author": "",
  "license": "ISC",
  "bugs": {
    "url": "https://github.com/renanponick/javascript/issues"
  },
  "homepage": "https://github.com/renanponick/javascript#readme"
}


Is this OK? (yes) yes
```

Configurações

package.json U

package.json > ...

```1 {
2   "name": "javascript",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
    Debug
6   "scripts": {
7     "test": "echo \"Error: no test specified\" && exit 1"
8   },
9   "repository": {
10    "type": "git",
11    "url": "git+https://github.com/renanponick/javascript.git"
12  },
13  "author": "",
14  "license": "ISC",
15  "bugs": {
16    "url": "https://github.com/renanponick/javascript/issues"
17  },
18  "homepage": "https://github.com/renanponick/javascript#readme"
19 }
20
```

Express

```
renan@ng0502:~/Documents/senac/JP/javascript$ npm i express

added 58 packages, and audited 59 packages in 6s

8 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Código GET

```
1   import express from "express"
2
3   const app = express();
4
5   app.get("/", (req, res) => {
6       res.json({ message: "Hello World!" });
7   });
8
9   app.listen(3000, () => {
10      console.log("Servidor rodando na porta 3000");
11  });
12
```

Atenção

```
1   import express from "express"
2
3   const app = express();
4
5   app.get("/", (req, res) => {
6       res.json({ message: "Hello World!" });
7   });
8
9   app.listen(3000, () => {
10      console.log("Servidor rodando na porta 3000");
11  });
12
```

Método, rota, e porta

Post e Status

```
// Cria uma instância do aplicativo Express.
const app = express();

// Middleware para permitir que o Express interprete o corpo da requisição como JSON
app.use(express.json());
```

```
app.post("/api/exercicio", (req, res) => {
    console.log(req.body)
    const num1 = req.body.num1;
    const num2 = req.body.num2;
    const result = num1 + num2;

    res.status(201).json({ message: `Resultado: ${result}` });
});
```

Código pronto, e agora como executa?

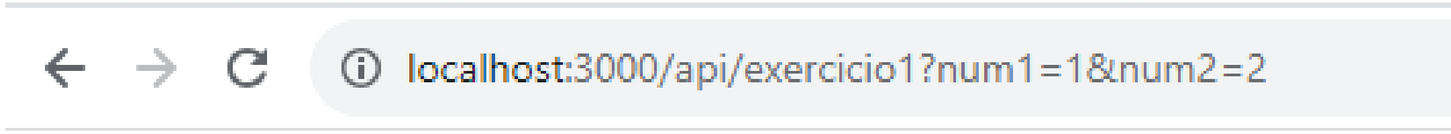
Inicializa o servidor

```
renan@ng0502:~/Documents/senac/JP/javascript$ npm start

> javascript@1.0.0 start
> nodemon index.js

[nodemon] 3.0.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
Servidor rodando na porta 3000
```

Só isso? Mas meu código não executou...

Realizando a requisição

Requisições do tipo GET podem ser realizadas diretamente na URL do seu navegador!



Ou via linha de comando, com o CURL

```
renan@ng0502:~/Documents/senac/JP/javascript$ curl localhost:3000/api/pessoa/123?nome=joao
{"message":"Você passou um queryParam: joao e um routParam: 123"}renan@ng0502:~/Documents/senac
$
```

Realizando a requisição