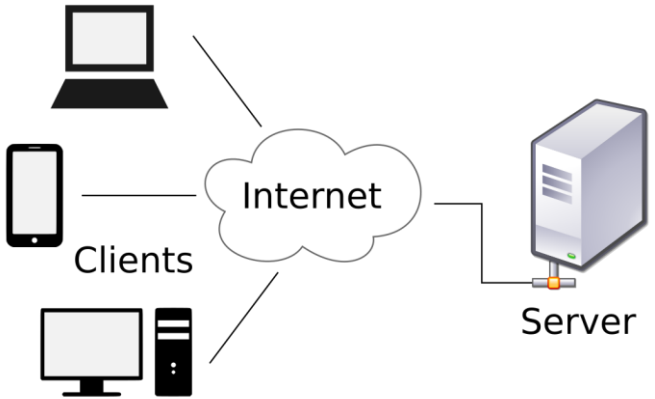
API - Intro

Aplicações Web

Instituto Politécnico de Viana do Castelo

Programming / **web** development is divided into **two** major **domains**:

1. Development on the **client** side (front end);
2. **Server-side** development (back end).

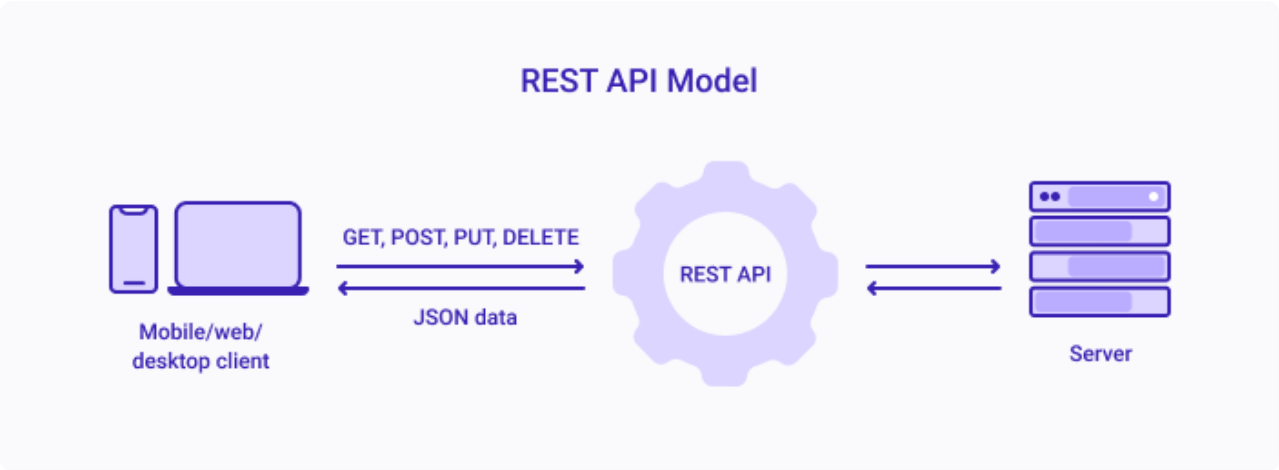


Bootstrap
Vue.js
React
jQuery
write less, do more.

ASP.NET
Core
Java
python
php
node
express

Rest API Model

Instituto Politécnico de Viana do Castelo



Rest API Model

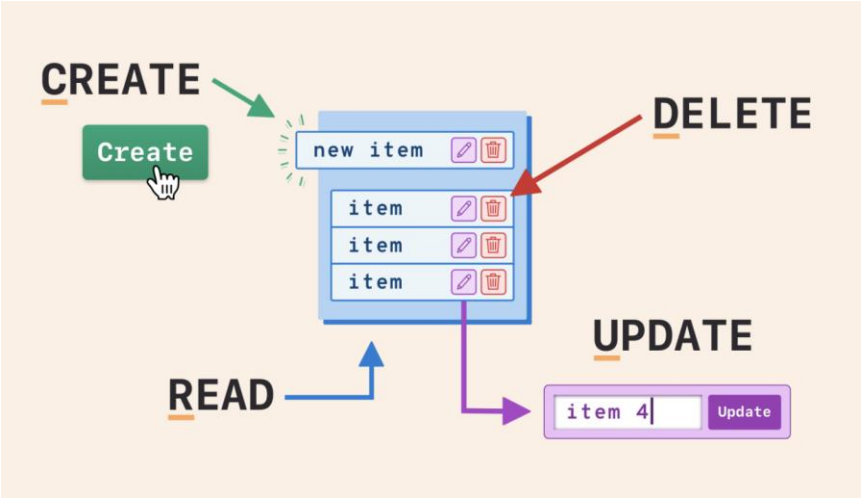
An API, or **application programming interface**, is a set of rules that define how applications or devices can connect to and communicate with each other.

https://www.ibm.com/topics/rest-apis

GET
PUT
POST
DELETE

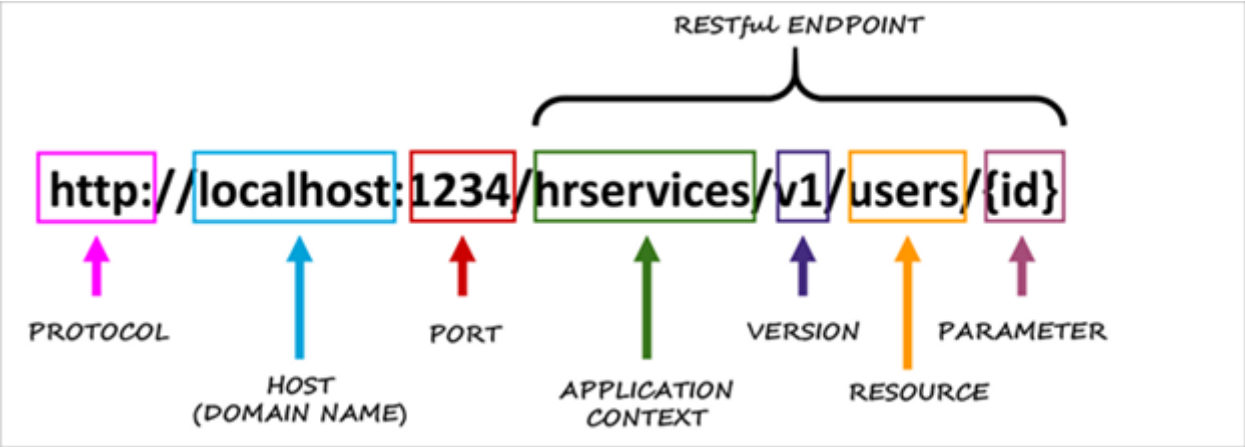
Web development
HTTP response status codes

https://tracy.hashnode.dev/http-response-status-codes



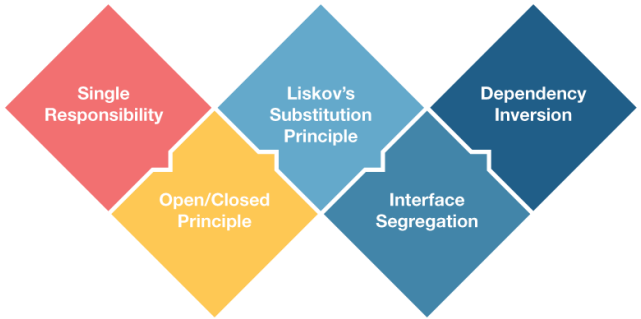
https://tracy.hashnode.dev/http-response-status-codes

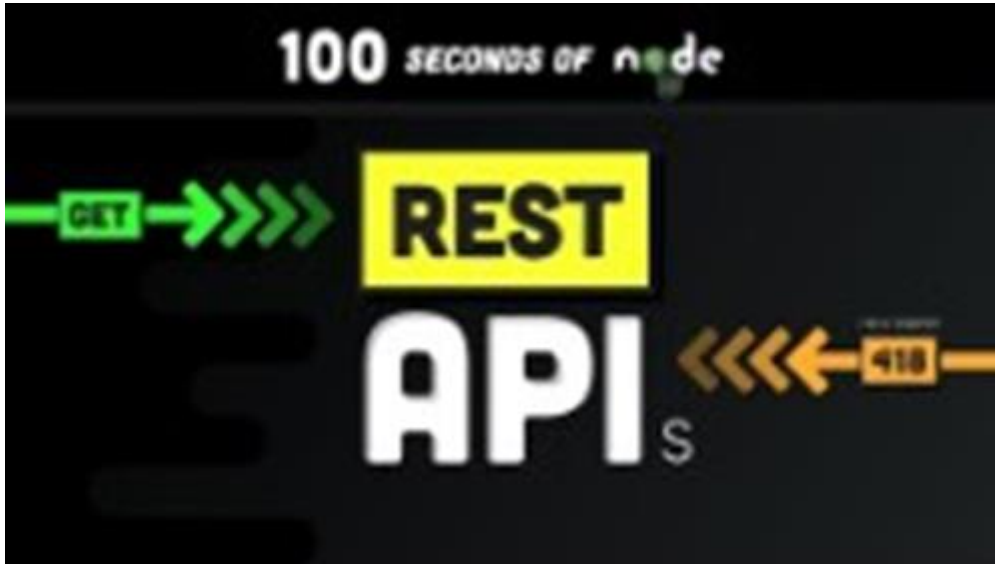
URL Anatomy



https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods

S.O.L.I.D.


Single Responsibility
Liskov's Substitution Principle
Dependency Inversion
Open/Closed Principle
Interface Segregation


100 SECONDS OF node
GET
REST
API
S
418

Instituto Politécnico de Viana do Castelo

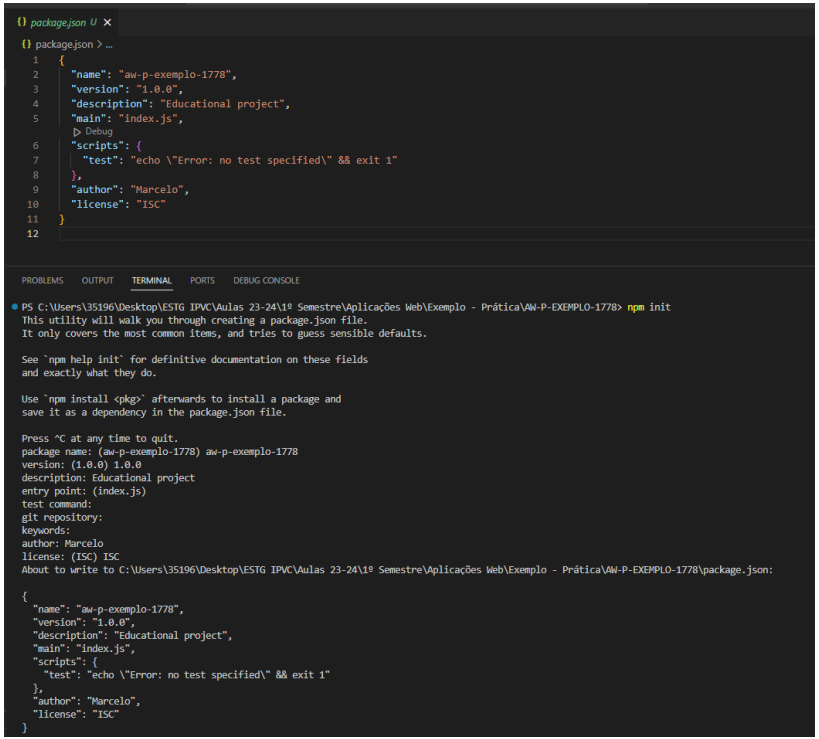
1. Criar conta no GitHub;
2. Criar um repositório – AW-P-EXEMPLO{{número-de-aluno}};
3. Instalar GitHub desktop e mapear o repositório;
4. Instalar node;
    1. https://nodejs.org/en
5. Instalar Postman;
    1. https://www.postman.com/
6. Abrir projeto mapeado com o VSCode;

Instituto Politécnico de Viana do Castelo

1. Abrir terminal e executar npm init;
2. Preencher os dados pedidos;
3. Executar:
    1. npm install express nodemon cors dotenv morgan --save
4. Criar ficheiro .env na raiz do projeto.

    SERVER_HOST=localhost

    SERVER_PORT=4242

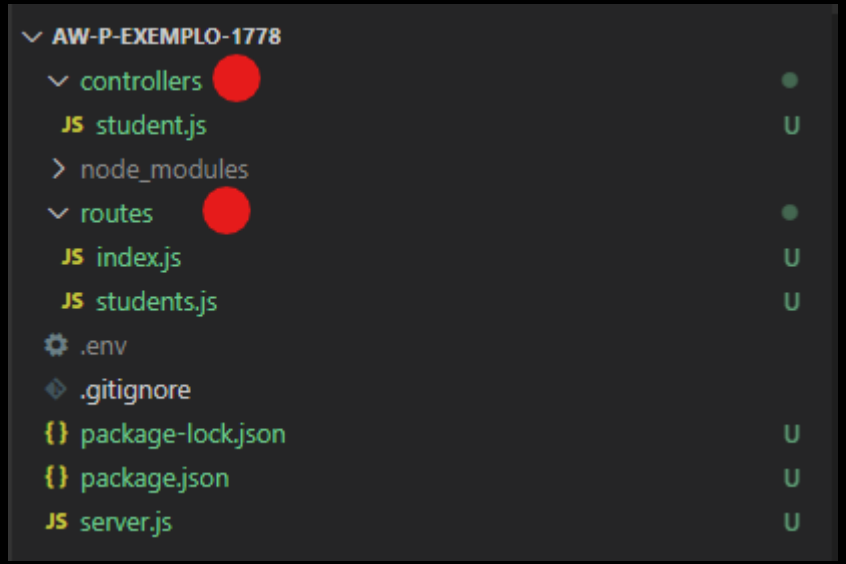


1. Criar ficheiro na raiz do projeto server.js
    1. Alterar no package.json "main": "server.js"
2. Adicionar script no package.json:
    1. "start": "node server.js“
3. Alterar ficheiro server.js:

server.js > ...

```
1   require('dotenv').config();
2
3   const bodyParser = require('body-parser');
4   const cors = require('cors');
5   const express = require('express');
6
7   const router = require('./routes/index.js');
8
9   const app = express();
10  app.use(bodyParser.json());
11  app.use(cors());
12
13  app.use('/api/', router);
14
15  const port = process.env.SERVER_PORT || 8080;
16  app.listen(port, () => {
17      console.log('Express server listening on port', port)
18  });
19
```

1. Criar a estrutura de pastas e ficheiros



1. Routes
    1. Index.js
    2. Students.js

index.js U

routes > index.js > ...

```
1   const router = require('express').Router();
2   const studentRouter = require('./students');
3
4   router.use('/students', studentRouter);
5
6   module.exports = router;
```

students.js U

routes > students.js > ...

```
1   const studentRouter = require('express').Router();
2   const controller = require('../controllers/student');
3
4   studentRouter.get('/', controller.getAll);
5   studentRouter.get('/:id', controller.getById);
6   studentRouter.post('/create', controller.create);
7   studentRouter.put('/update', controller.update);
8   studentRouter.delete('/delete', controller.delete);
9
10  module.exports = studentRouter;
```

Instituto Politécnico de Viana do Castelo

1. Controllers
    1. Student.js

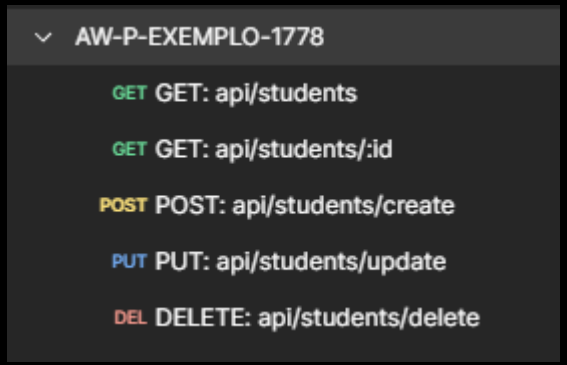
student.js U

controllers > student.js > getAll

```
1   exports.getAll = async (req, res) => {
2       return res.send("ok");
3   }
4
5   exports.getById = async (req, res) => {
6       const id = req.params.id;
7
8       res.send(id);
9   }
10
11  exports.create = async (req, res) => {
12      const {name, city, age } = req.body;
13
14      return res.send({name, city, age });
15  }
16
17  exports.update = async (req, res) => {
18      const {name, city, age } = req.body;
19
20      return res.send({name, city, age });
21  }
22
23  exports.delete = async (req, res) => {
24      const id = req.body.id;
25      return res.send(id);
26  }
```

Instituto Politécnico de Viana do Castelo

1. Executar no terminar npm start:
2. Testar no postman;
3. Push para o Git;



https://github.com/marcelowolfsmartindustries/AW-P-EXEMPLO-1778

o **teu** ● de partida

www.ipvc.pt