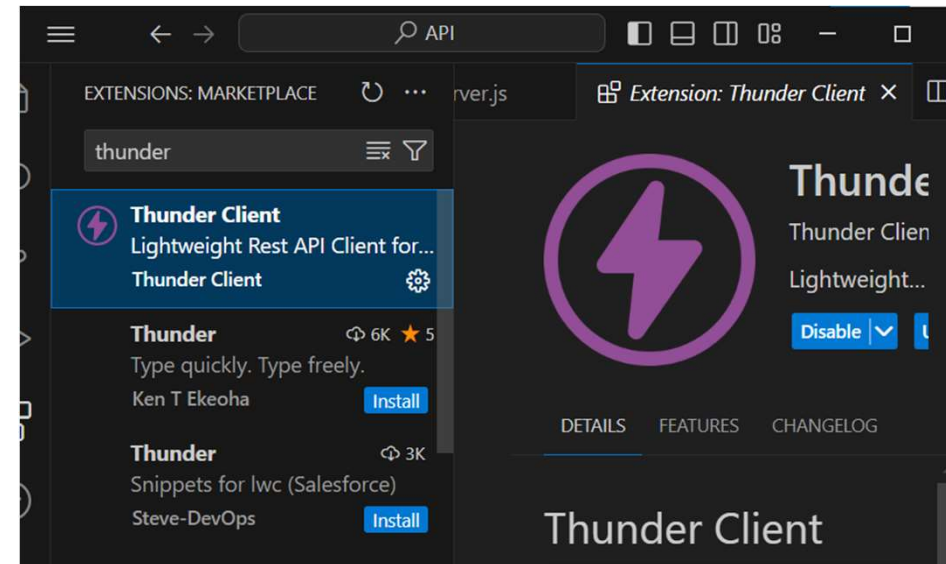


API : Application Programming Interface

- Entender conceitos de Back End e Front End;
- Criar Api;
- Conectar com um banco de dados;
- Query Params / Route Params / Body Params
- Métodos HTTP
- Status HTTP



- `npm init -y`
- `npm i express`
- `node --watch server.js`
- `npm install --save-dev nodemon`
- `npx nodemon server.js`

Dependências

```
JS server.js X TC New Request
JS server.js > ...
1  import express from 'express'
2
3  const app = express()
4
5  app.get('/usuarios', (req, res) => {
6    res.send('ok, deu certo!')
7  })
8
9  app.listen(3000)
10
11
12  /* -ROTAS:
13     1 Tipo de rota / Método HTTP
14     2 Endereço
15
16  */
17
```

```
{
  {
    id: '456',
    nome: 'Rodolfo',
    idade: 33,
    email: 'contato@rodolfo.com',
  },
  {
    id: '789',
    nome: 'Rodolfo',
    idade: 33,
    email: 'contato@rodolfo.com',
  }
}
```

get - users

post - users

delete - users

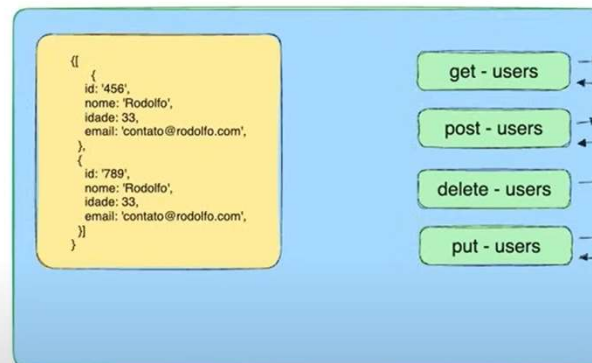
put - users

http://localhost:3000/users

HTTP Métodos

Get → Listar
Post → Criar
Put → Editar vários
Patch → Editar UM
Delete → Deletar

Back End (Servidor)



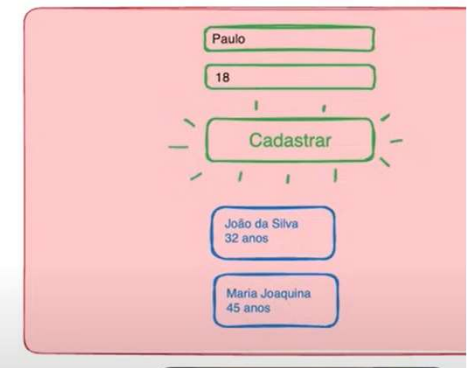
```
{  
  id: '456',  
  nome: 'Rodolfo',  
  idade: 33,  
  email: 'contato@rodolfo.com',  
},  
{  
  id: '789',  
  nome: 'Rodolfo',  
  idade: 33,  
  email: 'contato@rodolfo.com',  
}
```

Resposta

servidor.com/users

Requisição

Front End



HTTP: Hypertext Transfer Protocol

- Formas de envio:

REQUEST

Route Params (GET / PUT / DELETE)

Buscar, deletar ou editar algo Específico

```
get  servidor.com/usuarios/22
put  servidor.com/usuarios/22
delete servidor.com/usuarios/22
```

Body Params (POST e PUT)

```
{
  "nome": "Rodolfo", "id": 22
}
```

Query Params (GET)
Consultas

```
servidor.com/usuarios?estado=bahia&cidade=salvador
servidor.com/series?tipo=comedia&streaming=netflix
```

HTTP: Hypertext Transfer Protocol

- No contexto do protocolo HTTP, enviar informações para um servidor pode ser feito de diferentes maneiras, dependendo do método HTTP utilizado (como GET, POST, PUT, etc.) e de como essas informações são estruturadas. Os três principais lugares onde as informações podem ser enviadas são: route (rota), query e body. Cada um tem seu uso apropriado dependendo do tipo de requisição e da necessidade de dados.

HTTP: Hypertext Transfer Protocol

- 1. **Rota (Route)**

A rota faz parte do URL e normalmente identifica o recurso que está sendo acessado. As informações passadas na rota são geralmente parâmetros que indicam um recurso específico. Isso é comum em APIs RESTful.

GET /users/123

HTTP: Hypertext Transfer Protocol

- Aqui, `/users/123` está acessando o recurso usuário com ID 123.

O número "123" é um parâmetro da rota que identifica o usuário específico.

Uso típico (Route):

- Identificação de recursos específicos.
- Passagem de identificadores, como IDs.
- Exemplo de endpoint RESTful: `GET /products/45`, onde "45" pode ser o ID de um produto.

HTTP: Hypertext Transfer Protocol

- **2. Query String**

- A query string é uma parte da URL que começa após um ponto de interrogação (?) e permite enviar parâmetros de forma chave-valor. Esses parâmetros geralmente servem para filtrar, ordenar ou paginar os dados.

GET /products?category=electronics&sort=price

HTTP: Hypertext Transfer Protocol

- Aqui, `category=electronics` e `sort=price` são parâmetros de consulta (query parameters). Eles são usados para obter produtos da categoria "eletrônicos" e ordená-los por preço.

Uso típico (Query String) :

- Filtros e paginação.
- Busca e manipulação de resultados.
- Pode ser usada com métodos como GET e DELETE.

HTTP: Hypertext Transfer Protocol

- **3. Corpo da Requisição (Body)**

O body é a parte da requisição onde dados são enviados diretamente ao servidor. Isso é comum em métodos que modificam ou criam novos recursos, como POST, PUT e PATCH. O conteúdo pode ser em formatos como JSON, XML, form-urlencoded, entre outros.

Exemplo (usando JSON):

```
POST /users
```

```
Content-Type: application/json
```

```
{  
  "name": "João",  
  "email": "joao@example.com"  
}
```

HTTP: Hypertext Transfer Protocol

- Neste exemplo, o corpo da requisição contém os dados do novo usuário (nome e e-mail) que estão sendo enviados ao servidor para criação de um novo registro.

Uso típico:

- Criação e modificação de recursos.
- Envio de dados complexos, como formulários ou objetos JSON.

HTTP: Hypertext Transfer Protocol

- **Resumo**
- **Rota (Route):** Usada para identificar recursos específicos via URL (ex: /users/123).
- **Query String:** Usada para enviar parâmetros de consulta ou filtros via URL (ex: ?category=electronics).
- **Body:** Usado para enviar dados complexos no corpo da requisição, geralmente com métodos POST ou PUT (ex: dados de formulário em JSON).

Cada um desses elementos tem um papel importante dependendo do tipo de operação que você deseja realizar na API ou no servidor.