

API Calculadora com Swagger UI



Vagner Bom Jesus · Follow

2 min read · Apr 24, 2021



Share

Swagger permite que você descreva a estrutura de suas APIs para que as máquinas possam lê-las. O **Swagger** faz isso solicitando que sua API retorne um **YAML** ou **JSON** que contém uma descrição detalhada de toda a sua API.



O **Swagger UI** é um projeto de código aberto para renderizar visualmente a documentação de uma API definida com a especificação OpenAPI (**Swagger**).

A interface **Swagger** permite que você visualize e interaja com os recursos da API, o que simplifica a implementação de back-end e o uso do lado do cliente.

Vamos começar...

1º Passo Instalar o **Nodejs** ([aqui](#)).

2º Passo Criar o diretório onde vamos desenvolver o projeto, usando a linha de comandos:

a) Para iniciar o projeto com o seguinte comando:

```
npm init -y
```

b) A instalação do **Express.js** com o seguinte comando:

```
npm install express
```

c) Em seguida instalamos o **swagger-ui** com o seguinte comando:

```
npm install swagger-jsdoc@6.1.0 swagger-ui-express -s
```

3º Passo Depois de instalar as dependências, vamos criar o arquivo **index.js** e colocar o seguinte código

```
1  const express = require("express");
2  const app = express();
3
4  const swaggerJsDoc = require('swagger-jsdoc');
5  const swaggerUI = require('swagger-ui-express');
6
7  const swaggerOptions = {
8    swaggerDefinition: {
9      info: {
10        title: "Calculadora",
11        description: "API Calculadora com swagger Yam1",
12        version: "1.0.0"
13      },
14      host: "localhost:3333",
15      basePath: "/",
16      schemes: [
17        "http"
18      ],
19    },
20    apis: ['swagger.js'],
21  };
22
23  const swagger = swaggerJsDoc(swaggerOptions);
24  app.use('/api-calculadora', swaggerUI.serve, swaggerUI.setup(swagger));
25
26
27  //Rotas para acessar as operações
28  require('./routes/index')(app);
29
30  app.listen(3333, () => {
31    console.log("server listening on port 3333");
32  });
```

index.js hosted with ❤ by GitHub

[view raw](#)

4º Passo Criar Routes com onde ira ter as nossas operações com o seguinte comandos:

a) Criar o diretório, e criar o ficheiro **index.js**

```
mkdir routes
```

```
cd routes
```

```
touch index.js
```

b) routes/index.js , colocar o seguinte código:

```
1  module.exports = function(app){
2
3
4    app.post('/sum/:sum1/:sum2', (req, res) => {
5
6      var result = parseInt(req.params.sum1) + parseInt(req.params.sum2);
7      res.send(result.toString())
8    })
9
10   //SOMA
11   app.post("/soma/:num1/:num2", (req, res) => {
12     var result = parseInt(req.params.num1) + parseInt(req.params.num2)
13     res.send(result.toString())
14   });
15
16   //SUBTRAÇÃO
17   app.post("/sub/:num1/:num2", (req, res) => {
18     const result = parseInt(req.params.num1) - parseInt(req.params.num2)
19     res.send(result.toString())
20   });
21
22   // MULTIPLICAÇÃO
23   app.post("/mult/:num1/:num2", (req, res) => {
24     const result = parseInt(req.params.num1) * parseInt(req.params.num2)
25     res.send(result.toString())
26   });
27
28   // DIVISÃO
29   app.post("/div/:num1/:num2", (req, res) => {
30     const result = parseInt(req.params.num1) / parseInt(req.params.num2)
31     res.send(result.toString())
```

Open in app ↗

Sign up

Sign in

Medium



Search



4º Passo Criar o ficheiro swagger.js e coloca o seguinte código

```
1
2  /**
3   * FILE SWAGGER
4   * Author: Vagner Bom Jesus
5   */
6  /**
7   * @swagger
8   * /soma/{num1}/{num2}:
9   *   post:
10    *     tags:
11    *       - SOMA
12    *     description: "Escolha dois números."
13    *     parameters:
14    *       - in: path
15    *         name: num1
16    *         default: "2"
17    *         description: "Primeiro número. Valor default <code> 2</code>."
18    *         schema:
19    *           type: number
20    *         required: true
21    *       - in: path
22    *         name: num2
23    *         default: "2"
24    *         description: "Segundo número. Valor default <code> 2</code>."
25    *         schema:
26    *           type: number
27    *         required: true
28    *     responses:
29    *       200:
30    *         name: result
31    *         description: Success
32    */
33
34
35  /**
36   * @swagger
37   * /sub/{num1}/{num2}:
38   *   post:
39   *     tags:
40   *       - SUBTRAÇÃO
41   *     description: "Escolha dois números."
42   *     parameters:
43   *       - in: path
44   *         name: num1
45   *         default: "2"
46   *         description: "Primeiro número. Valor default <code> 2</code>."
47   *         schema:
48   *           type: number
```

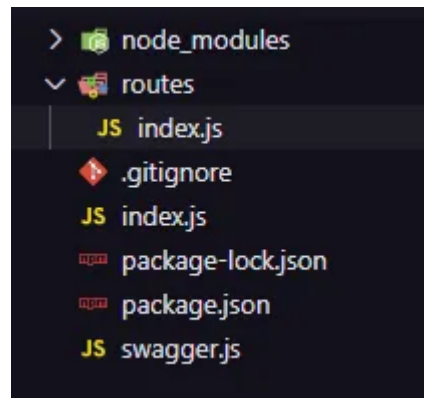
```

49 *         required: true
50 *     - in: path
51 *         name: num2
52 *         default: "2"
53 *         description: "Segundo número. Valor default <code> 2</code>."
54 *         schema:
55 *             type: number
56 *             required: true
57 *     responses:
58 *         200:
59 *             name: result
60 *             description: Success
61 */
62
63
64 /**
65 * @swagger
66 * /mult/{num1}/{num2}:
67 *   post:
68 *     tags:
69 *       - MULTIPLICAÇÃO
70 *     description: "Escolha dois números."
71 *     parameters:
72 *       - in: path
73 *         name: num1
74 *         default: "2"
75 *         description: "Primeiro número. Valor default <code> 2</code>."
76 *         schema:
77 *             type: number
78 *             required: true
79 *       - in: path
80 *         name: num2
81 *         default: "2"
82 *         description: "Segundo número. Valor default <code> 2</code>."
83 *         schema:
84 *             type: number
85 *             required: true
86 *     responses:
87 *         200:
88 *             name: result
89 *             description: Success
90 */
91
92
93 /**
94 * @swagger
95 * /div/{num1}/{num2}:

```

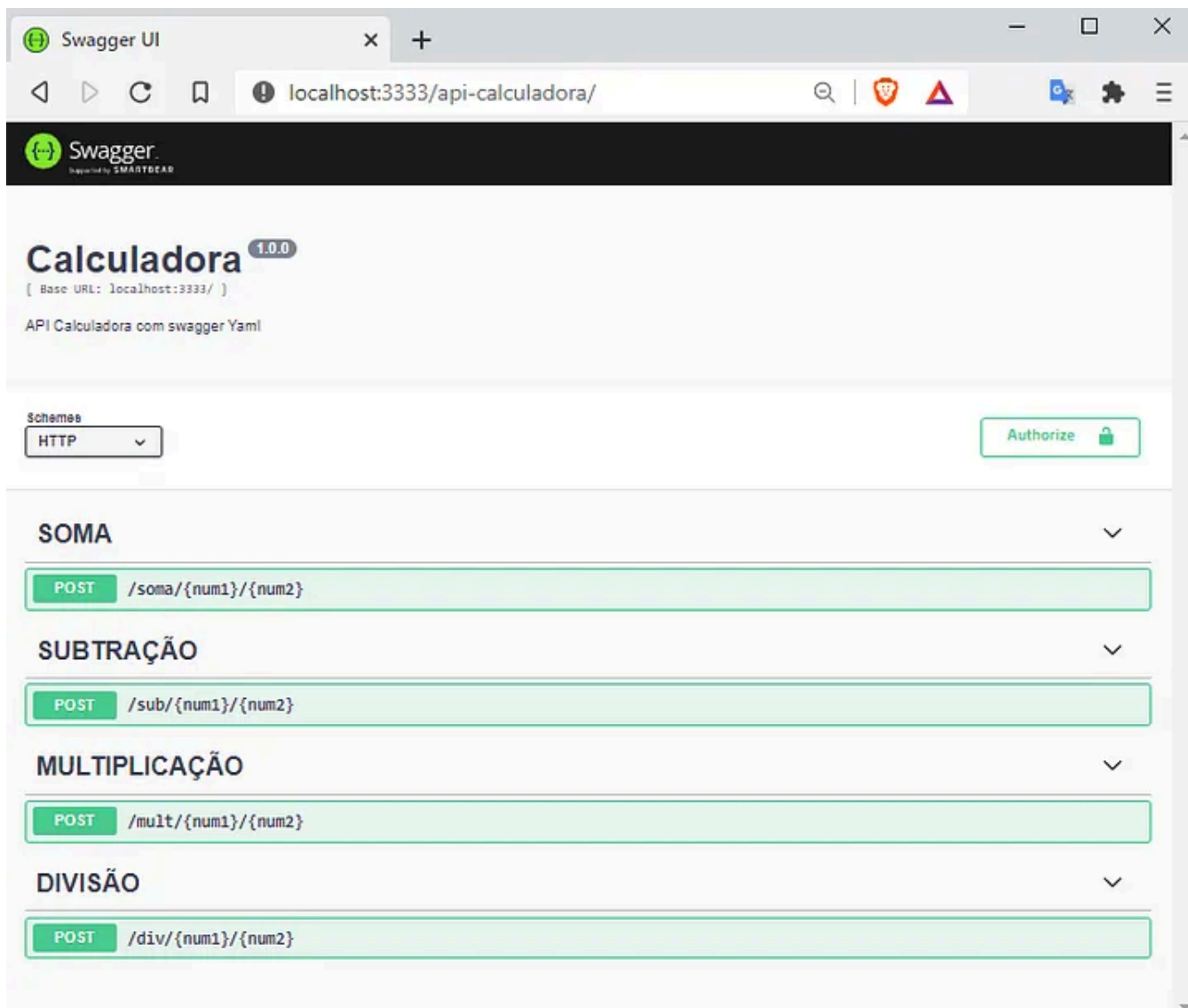
```
96  *   post:
97  *     tags:
98  *       - DIVISÃO
99  *     description: "Escolha dois números."
100 *     parameters:
101 *       - in: path
102 *         name: num1
103 *         default: "2"
104 *         description: "Primeiro número. Valor default <code> 2</code>."
105 *         schema:
106 *           type: number
107 *         required: true
108 *       - in: path
109 *         name: num2
110 *         default: "2"
111 *         description: "Segundo número. Valor default <code> 2</code>."
112 *         schema:
113 *           type: number
114 *         required: true
115 *     responses:
116 *       200:
117 *         name: result
118 *         description: Success
119 */
```

Estrutura do projeto



Depois de realizar toda a programação necessária, como foi apresentada aqui neste artigo, basta executar a aplicação usando o comando **node app.js** e verificar se está tudo a funcionar da mesma forma apresentada abaixo.

URL: <http://localhost:3333/api-calculadora/>



Exemplo da operação Soma: $2 + 2 = 4$

The screenshot shows the Swagger UI for a POST endpoint named `/soma/{num1}/{num2}`. The interface includes a 'Parameters' section with two required path parameters, `num1` and `num2`, both with default values of 2. Below the parameters is an 'Execute' button and a 'Clear' button. The 'Responses' section shows a 200 status code with a response body of 4. The 'Curl' section displays the command: `curl -X 'POST' \ 'https://localhost:3333/soma/2/2' \ -H 'accept: application/json' \ -d ''`. The 'Request URL' is `http://localhost:3333/soma/2/2`. The 'Server responses' section shows a 200 status code with a response body of 4.

Conclusão

Está pronta a nossa API!

API completa Calculadora com **swagger Yaml** ([aqui!](#)).

Da mesma forma o mesmo exercício feito usando ficheiro **Json**, API completa Calculadora com **swagger Json** ([aqui!](#)).

[Swagger](#)[Swagger Ui](#)[Nodejs](#)[Expressjs](#)[API](#)[Follow](#)

Written by **Vagner Bom Jesus**

28 Followers · 28 Following

Licenciatura em Engenharia Informática

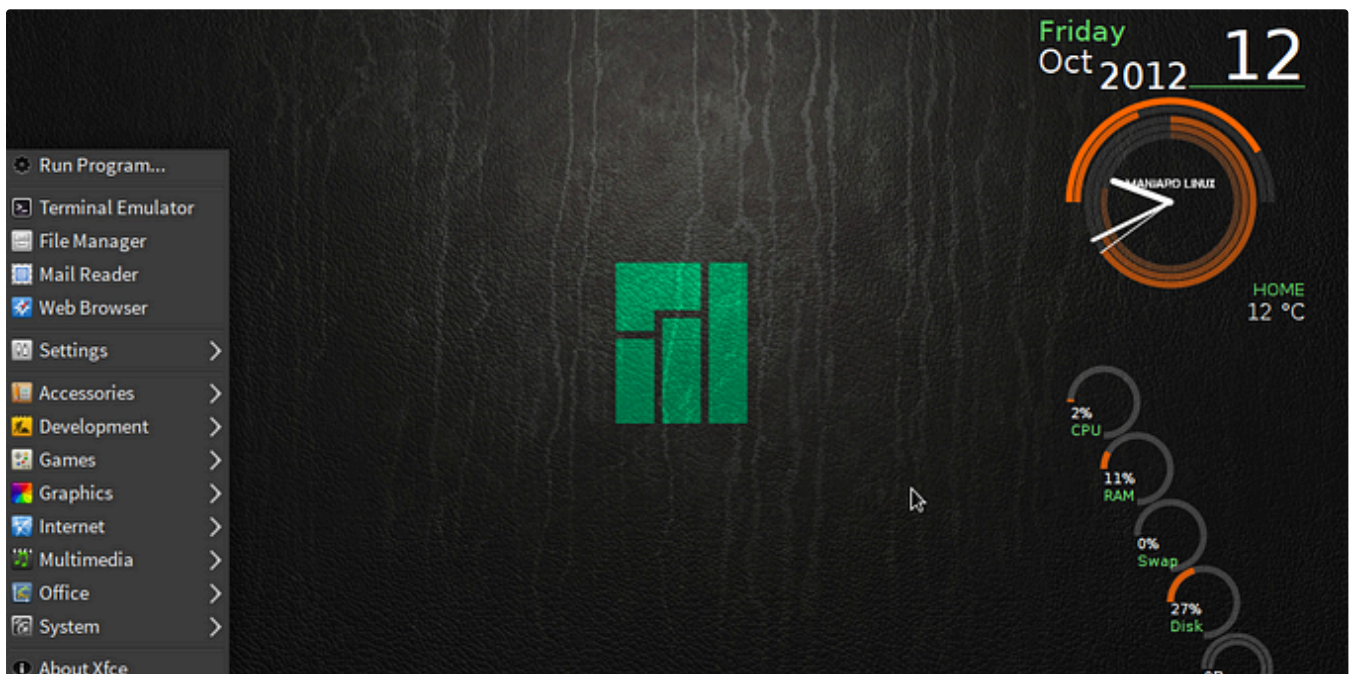
No responses yet



What are your thoughts?

Respond

More from Vagner Bom Jesus



Vagner Bom Jesus


Manjaro: Instalar ambientes de área de trabalho

Existem vários ambientes de desktop e gerenciadores de janelas disponíveis para o Manjaro, cada um com seu próprio estilo, interface e...

May 7, 2024



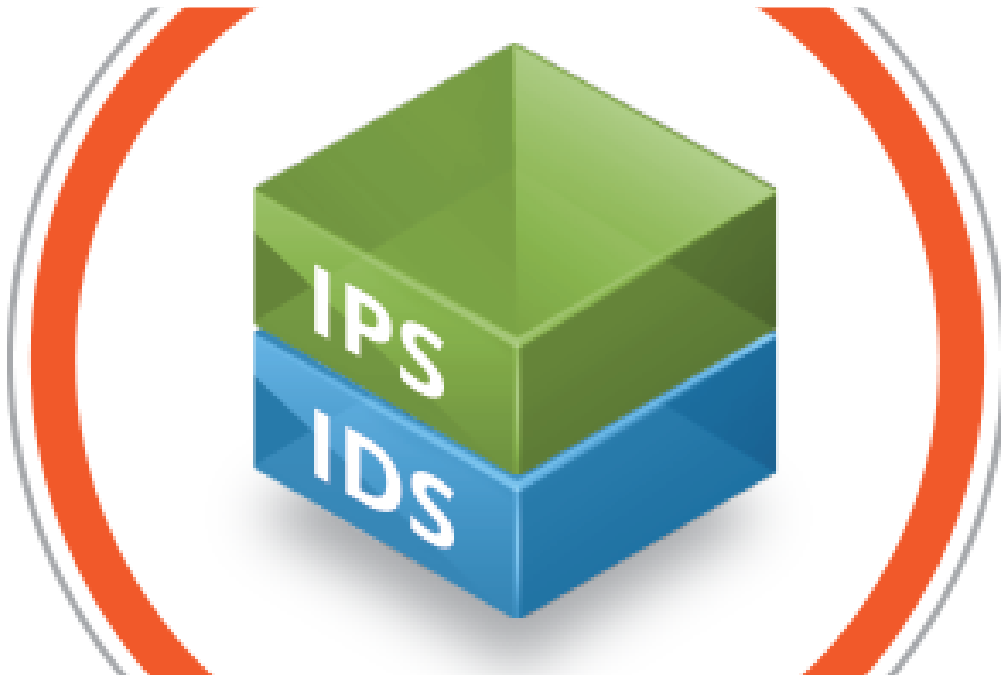



 Vagner Bom Jesus

Instalar e configurar um cluster Apache Spark no Hadoop Ubuntu 18.04 (Multi-Node)

Neste artigo, fornecerei instruções passo a passo sobre como instalar o Apache Spark em um cluster em dois Slave.

Jun 1, 2021  100

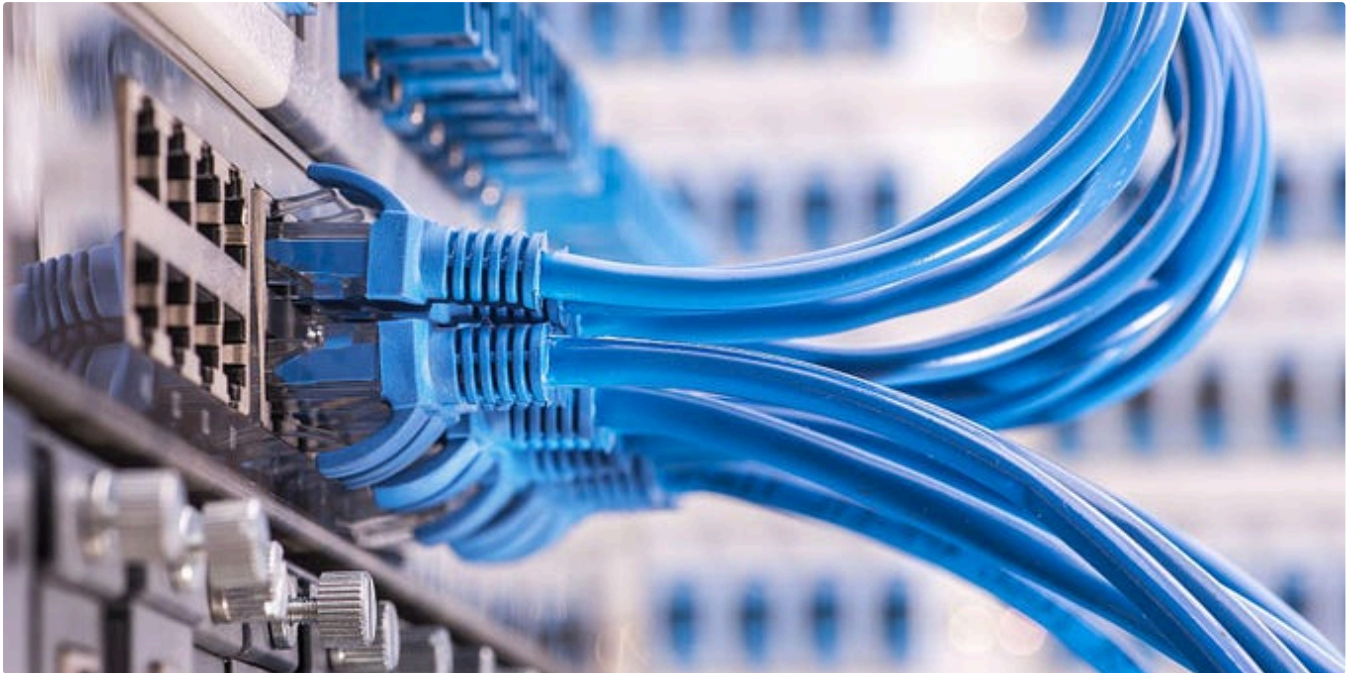


 Vagner Bom Jesus

Lab 7—IPS/IDS

No âmbito do Lab 7—IPS/IDS, partilho a minha experiência na configuração e implementação de um Sistema de Detecção de Intrusão (IDS) e de...

2d ago 🖱 105



Vagner Bom Jesus

Understanding Network Notations: Slash 24 (/24) vs. 255.255.255.0—Any Differences?

The mask, associated with an IP address, can be displayed in a number of ways. Today we're going to talk a little bit about how we can...

May 8, 2024



See all from Vagner Bom Jesus

Recommended from Medium



Saurav Sanyal

Creating Robust APIs with Response Templates, Custom Exceptions, and Controller Advice

Mastering API Exception Handling: Turning Errors into Seamless User Experiences



Dec 5, 2024



Sandhuya Sharma

Swagger Documentation

Introduction

★ Sep 13, 2024



Lists



Coding & Development

11 stories · 971 saves



Company Offsite Reading List

8 stories · 175 saves



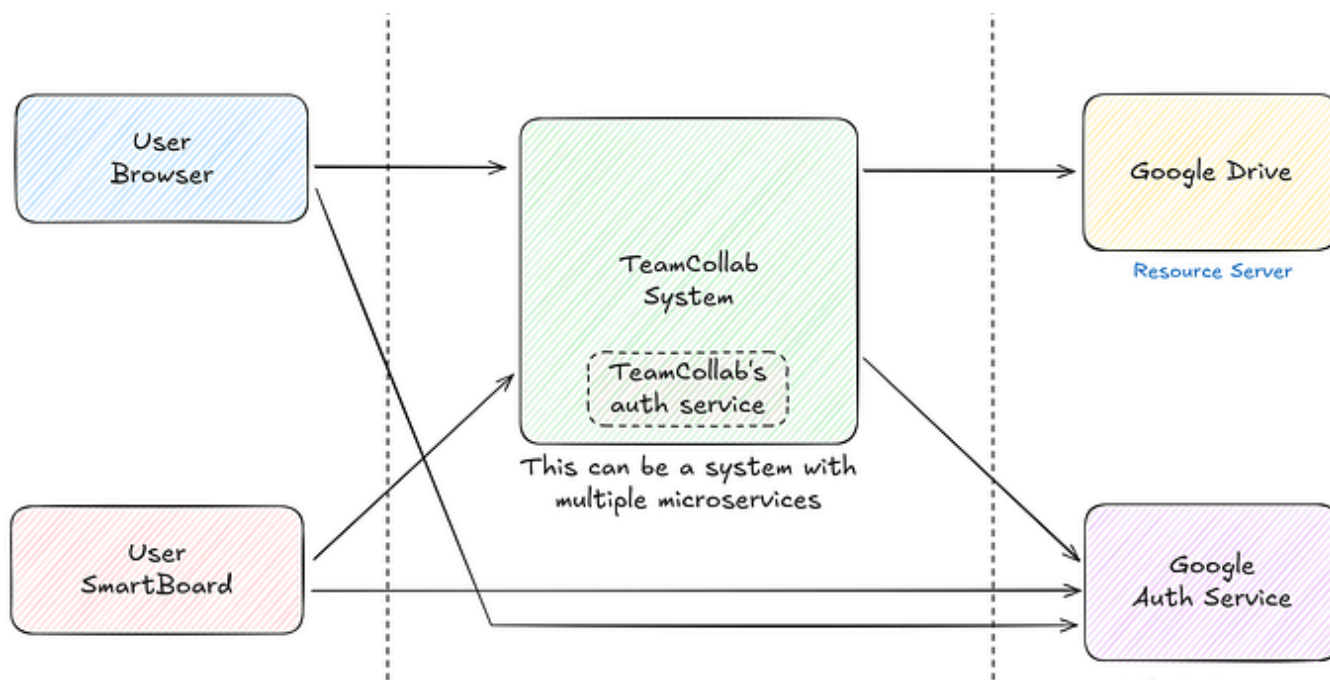
Stories to Help You Grow as a Software Developer

19 stories · 1557 saves



data science and AI

40 stories · 314 saves



🔗 In CodeX by Isuru Cumarathunga

Choosing the Right OAuth 2.0 Grant Type. A Scenario-Based Guide

Apply the Right Grant Type to Your Use Case

★ Oct 13, 2024 🖱 2





Vikrant Dheer

Mastering API-First Development with OpenAPI and Swagger

Non-members can read story from here.



Nov 24, 2024



149



1



Nick Lydon

Speeding up entity framework migrations in integration tests

Important addendum

Oct 22, 2024



4





Harendra

How I Am Using a Lifetime 100% Free Server

Get a server with 24 GB RAM + 4 CPU + 200 GB Storage + Always Free



Oct 26, 2024



8.5K



131



See more recommendations