

Projeto escolhido: Cidadão de Olho

Autor: Ítalo Vasconcelos Costa

E-mail: italo.costa99@gmail.com

Arquivos Necessários

Antes de começar a discutir o projeto é de extrema importância que o usuário tenha a versão do PHP 8.1.1, que pode ser obtido em:

<https://www.php.net/downloads>

altere a php.ini localizada dentro da pasta do php e retire o ; de ;extension=pdo_sqlite, feito isso salve o arquivo.

Para a realização dos itens solicitados no projeto foi utilizado o framework Laravel, sendo assim é necessário instala-lo, assim como o composer e inserir as variáveis no sistema. Link do download do composer:

<https://getcomposer.org/download/>

Após a instalação do composer é necessário, caso o Laravel não esteja instalado, digitar no cmd:

Composer global require laravel/installer

Com os arquivos citados instalados, já é possível abrir o projeto que está localizado no github(<https://github.com/ItaloVCosta/ProjetoPHP>).

Seleção dos Links da API da ALMG

Previamente a programação do código, foi selecionado os links necessários da ALMG, são eles:

Deputados do mandato de 2019 (recebendo a relação de nome e ID)

<http://dadosabertos.almg.gov.br/ws/legislaturas/19/deputados/situacao/1?formato=json>

Verbas indenizatórias (usado para obter os valores dos deputados que mais pediram reembolso em 2019, separado por mês)

http://dadosabertos.almg.gov.br/ws/prestacao_contas/verbas_indenizatorias/legislatura_atual/deputados/{id}/2019/{mes}?formato=json

Redes Sociais (se encontram no link por id de deputado):

<http://dadosabertos.almg.gov.br/ws/deputados/{id}?formato=json>

Softwares Utilizados

O projeto foi feito utilizando o editor de código VScode, além disso, para o teste do retorno do método GET da API utilizado o insomnia. Para a melhor visualização do banco de dados, o SQLiteStudio foi empregado.

Elaboração do Código

Inicialmente foi criado na pasta de “WebService” o arquivo ALMG.php, sendo uma classe que faz a requisição para a API da ALMG utilizando o curl e retornado um array. Nessa classe se encontram 3 funções, uma para cada Link descrito em “Seleção dos Links da API da ALMG”.

Com a comunicação com a API disponível foram trabalhados os dados em “index.php”(vendor\index.php) solicitados de forma a aparece a resposta desejada, por exemplo, para as redes sociais foi iterado as redes sociais de cada deputado e somado no array “usoDasRedes” e posteriormente organizado de forma decrescente. Já para a verba indenizatória foi feito a função “verbadosDeputados” que soma todos os pedidos de reembolso realizados no mês informado, retornando um vetor com id, nome do deputado e valor total pedido em reais, dos 5 deputados que mais gastaram.

Por último foi enviado os dados para o banco de dados escolhido (SQLite), através das funções criadas no “enviaDado.php” que utiliza o PDO para pode registrar os conteúdos das tabelas.

Para a estruturação da tabela foi criado as migrations “2022_01_17_183123_create_RedeSociais_table.php” e “2022_01_17_183258_create_VerbaIndenizatorias_table.php”.

Na elaboração da API foram criados duas rotas que executam o método get, também foi criado o controller de cada solicitação (RedesSociais e VerbasIndenizatoria) que lista os dados de cada tabela da API.

Antes de Rodar o Código

Antes de digitar o comando php index.php no cmd é necessário inserir alguns comandos para o perfeito funcionamento do código, são eles:

- “*php artisan migrate*” para utilizar os arquivos localizados em migrations e criar a tabela no banco de dados “database.sqlite”, localizado em (database\database.sqlite). Detalhe importante, o banco de dados tem que estar vazio!
- “*php artisan serve*” para iniciar o servidor o ser possível fazer a solicitação GET através do insomnia.

Rodando o Programa

Dado a restrição de tempo e o limitado conhecimento, ao rodar o programa o usuário terá que esperar um tempo considerável para rodar o programa, já que não foi utilizado `async` e `await` para poder otimizar o processo de leitura de dados do API, sendo assim foram colocados `delays` entre as requisições, dessa forma tornando o programa lento, no PC que estou utilizando o código demorou cerca de 1060 segundos para rodar.

```
[Done] exited with code=0 in 1060.365 seconds
```

Após rodar o programa, os resultados já podem ser observados no banco de dados diretamente:

	id	idDeputado	Mes	NomeDeputado	Valor
1	1	15245	Janeiro	Agostinho Patrus	0
2	2	7752	Janeiro	Alencar da Silveira Jr.	0
3	3	19150	Janeiro	Ana Paula Siqueira	0
4	4	12195	Janeiro	André Quintão	0
5	5	26143	Janeiro	Andréia de Jesus	0
6	6	10434	Fevereiro	Leandro Genaro	30938.8
7	7	22516	Fevereiro	Fábio Avelar de Oliveira	29032.06
8	8	1832	Fevereiro	Sávio Souza Cruz	29024.53
9	9	22515	Fevereiro	Thiago Cota	28713.06
10	10	4458	Fevereiro	Marquinho Lemos	27935.41
11	11	22515	Março	Thiago Cota	45426.87
12	12	10434	Março	Leandro Genaro	34760.1
13	13	10002	Março	Carlos Henrique	32671.38
14	14	26137	Março	Professor Irineu	30632.59
15	15	9652	Março	Arlen Santiago	30193.92
16	16	16019	Abril	Léo Portela	43186.33
17	17	10434	Abril	Leandro Genaro	34444.51
18	18	22581	Abril	Doutor Jean Freire	27783.13
19	19	9652	Abril	Arlen Santiago	27145.85
20	20	13425	Abril	Betinho Pinto Coelho	26925.77
21	21	26062	Maio	Charles Santos	39766.14
22	22	17872	Maio	Professor Wendel Mesquita	37061.43
23	23	10434	Maio	Leandro Genaro	35207.86
24	24	12195	Maio	André Quintão	34747.34
25	25	4458	Maio	Marquinho Lemos	34662.81

....

Observa-se que em janeiro não foram registrado reembolsos, já que é início de mandato jan/2019, sendo assim o programa ordenou pelo nome. Para mais meses abra o banco de dados.

Também temos a resposta das redes sociais:

	id	RedeSocial	Usuarios
1	1	Instagram	75
2	2	Facebook	73
3	3	Twitter	44
4	4	Youtube	24
5	5	WhatsApp	4
6	6	SoundCloud	3
7	7	Flickr	3
8	8	LinkedIn	1

Para testar o funcionamento da API, utilizou-se o método GET:

- <http://127.0.0.1:8000/api/RedesSociais> para obter o ranking de redes sociais

```
Preview ▾ Header 00 Cook
1 ▾ [
2 ▾ {
3   "id": 1,
4   "RedeSocial": "Instagram",
5   "Usuarios": "75"
6 },
7 ▾ {
8   "id": 2,
9   "RedeSocial": "Facebook",
10  "Usuarios": "73"
11 },
12 ▾ {
13   "id": 3,
14   "RedeSocial": "Twitter",
15   "Usuarios": "44"
16 },
17 ▾ {
18   "id": 4,
19   "RedeSocial": "Youtube",
20   "Usuarios": "24"
21 },
22 ▾ {
23   "id": 5,
24   "RedeSocial": "WhatsApp",
25   "Usuarios": "4"
26 },
27 ▾ {
28   "id": 6,
29   "RedeSocial": "SoundCloud",
30   "Usuarios": "3"
31 },
32 ▾ {
33   "id": 7,
34   "RedeSocial": "Flickr",
35   "Usuarios": "3"
36 },
37 ▾ {
38   "id": 8,
39   "RedeSocial": "LinkedIn",
40   "Usuarios": "1"
41 }
42 ]
```

- <http://127.0.0.1:8000/api/VerbalIndenizatoria> para obter os top 5 deputados que mais pediram reembolso de verbas indenizatórias por mês em 2019.

```

{
  "id": 1,
  "idDeputado": "15245",
  "Mes": "Janeiro",
  "NomeDeputado": "Agostinho Patrus",
  "Valor": "0"
},
{
  "id": 2,
  "idDeputado": "7752",
  "Mes": "Janeiro",
  "NomeDeputado": "Alencar da Silveira Jr.",
  "Valor": "0"
},
{
  "id": 3,
  "idDeputado": "19150",
  "Mes": "Janeiro",
  "NomeDeputado": "Ana Paula Siqueira",
  "Valor": "0"
},
{
  "id": 4,
  "idDeputado": "12195",
  "Mes": "Janeiro",
  "NomeDeputado": "André Quintão",
  "Valor": "0"
},
{
  "id": 5,
  "idDeputado": "26143",
  "Mes": "Janeiro",
  "NomeDeputado": "Andréia de Jesus",
  "Valor": "0"
},
{
  "id": 6,
  "idDeputado": "10434",
  "Mes": "Fevereiro",
  "NomeDeputado": "Leandro Genaro",
  "Valor": "30938.8"
},
{
  "id": 7,
  "idDeputado": "22516",
  "Mes": "Fevereiro",
  "NomeDeputado": "Fábio Avelar de Oliveira",
  "Valor": "29032.06"
},
{

```

Atenção ao valor do local Host pode variar.