

Anderson Nascimento

Vinicius Lessa da Silva

Consumo de API: Pixabay

Atividade desenvolvida pelos alunos Anderson Nascimento e Vinicius Lessa da Silva para a disciplina de Desenvolvimento para Servidores I, administrada pelo Professor Fernando Leonid.

São Roque

2020

Introdução

Atividade proposta para o desenvolvimento de um website que consome a API RESTful do Pixabay, para pesquisar e retornar imagens e vídeos gratuitos, lançados sob a licença do próprio Pixabay. O documento tem como objetivo mostrar de que forma foi feito o desenvolvimento do website.

API

A API retorna objetos codificados em JSON, com a codificação em padrão UTF-8. As chaves hash, podem ser retornadas em qualquer ordem (de forma aleatória) e novas chaves podem ser adicionadas a qualquer momento. Por padrão, pode-se fazer no máximo até 5.000 solicitações por hora, estando estas solicitações associadas à chave da API, lembrando que nunca vinculadas aos endereços IP. As imagens retornadas com a API podem ser utilizadas somente para serem exibidas temporariamente.

Manipulação de Erros

Caso ocorra um erro, é retornado uma resposta com o código de status de erro HTTP de acordo, no corpo da resposta contém uma descrição do problema em formato de texto.

Exemplo

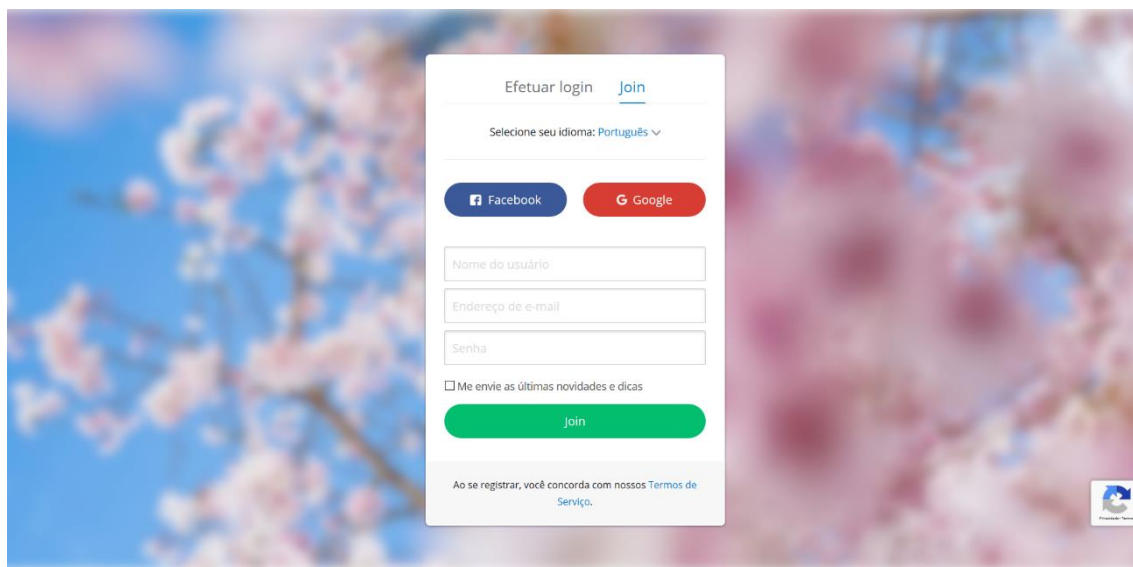
HTTP 429 – Limite de taxa de API excedido.

Como Consumir a API do Pixabay

1º PASSO

Primeiramente é necessário efetuar um cadastro na seguinte URL:

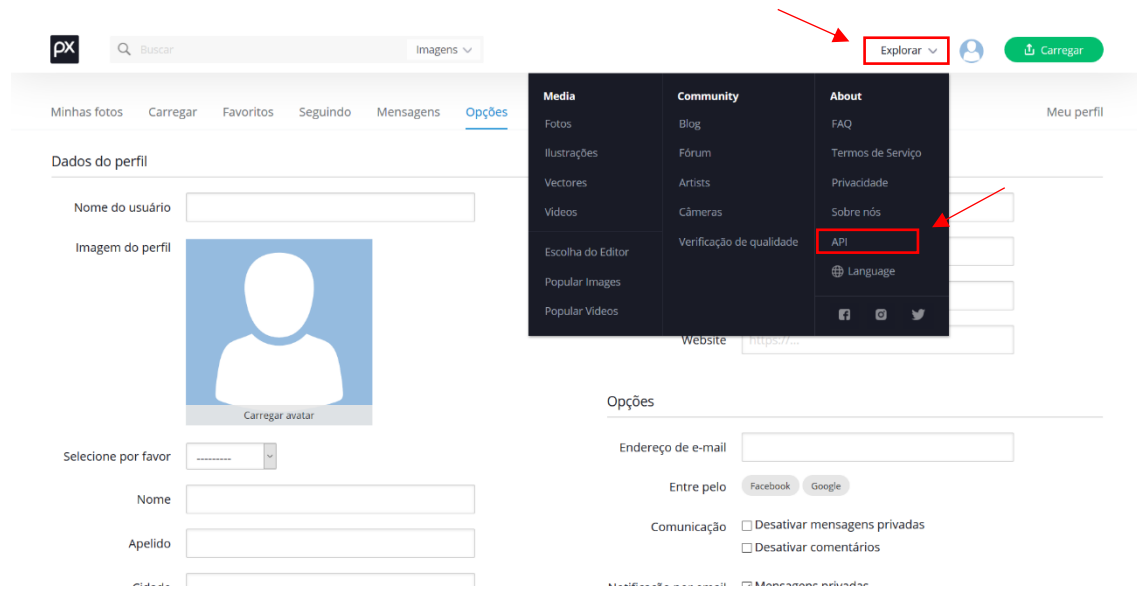
<https://pixabay.com/pt/accounts/register/?source=&next=>



OBS: se o cadastro foi realizado com sucesso você deverá receber um e-mail com o link de confirmação.

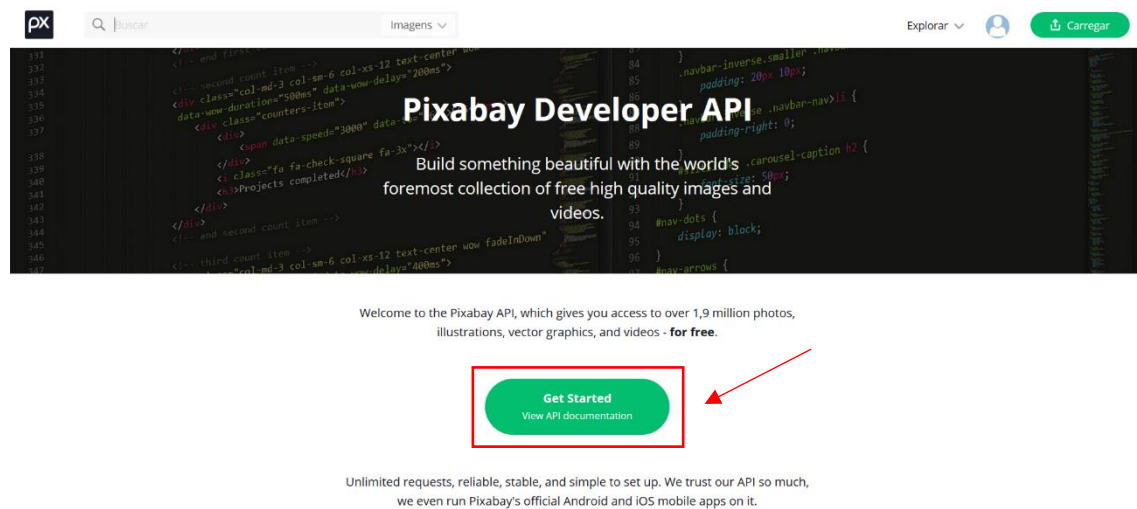
2º PASSO

Feito o cadastro você irá efetuar o seu login, clicar em Explorar e logo em seguida clicar em API.



3º PASSO

Você será redirecionado para a seguinte página:



Onde irá clicar no botão link Get Started (que te redireciona para a documentação da API), que lhe trará diversas informações sobre como fazer o uso da API RESTful da Pixabay.

4º PASSO

Ainda na documentação, na parte de Parameters, você terá acesso ao seu API Key, um conjunto de letras e números diversos, onde deverá implementar no seu código para fazer o uso das requisições da API.

Search Images

<https://pixabay.com/api/>

GET

Parameters

key (required) str Your API key:

5º PASSO

Quando começar a digitar o seu código, você deve fazer o uso da sua API Key, para então poder fazer requisições no seu sistema e ter o retorno do Pixabay. Neste caso armazenamos a nossa API Key dentro da variável \$token.

```
1 <?php
2
3 Function VarResult(){
4
5     $toSeek = "";
6
7     if(isset($_POST["search"]) && isset($_POST["tipo"])){
8
9         // Variáveis
10        $toSeek = $_POST["search"];
11        $token = " ";
12        $type = $_POST["tipo"];
13        $url = "";
14
15        if ($type == "video") {
16
17            $url = "https://pixabay.com/api/?key={$token}&q={$toSeek}&image_type=photo";
18        } elseif (($type == "imagem")){
19
20            $url = "https://pixabay.com/api/videos/?key={$token}&q={$toSeek}";
21        }
22
23        if (!Empty($url)){
24            return $retVar = RequireData($url);
25        }
26    }
27
28    return null;
29
30 }
31 }
```

Neste nosso código, criamos uma função `VarResult()`, onde verifica se as variáveis dos elementos `$_POST["search"]` e `$_POST["tipo"]` do html estão definidas, e que recebe 4 variáveis e verifica duas condições:

```

1 <?php
2
3 Function VarResult(){
4
5     $toSeek = "";
6
7     if(isset($_POST["search"]) && isset($_POST["tipo"])){
8
9         // Variáveis
10        $toSeek    = $_POST["search"];
11        $token      = "XXXXXXXXXXXXXXXXXXXX";
12        $type       = $_POST["tipo"];
13        $url        = "";
14
15        if ($type == "video") {
16
17            $url = "https://pixabay.com/api/?key={$token}&q={$toSeek}&image_type=photo";
18
19        } elseif (($type == "imagem")){
20
21            $url = "https://pixabay.com/api/videos/?key={$token}&q={$toSeek}";
22
23        }
24
25        if (!Empty($url)){
26            return $retVar = RequireData($url);
27        }
28    }
29
30    return null;
31 }

```

-A primeira para verificar se a variável \$type tem valor igual a “vídeo”, caso essa condição seja verdadeira ela retorna dentro da variável \$url o link com as concatenações das variáveis \$token e \$toSeek (que recebe o valor do parâmetro passado pelo usuário no form).

-A segunda condição é para verificar se a variável \$type tem valor igual a “imagem”, caso essa condição se mostrar verdadeira ela retorna dentro da variável \$url o link com as concatenações das variáveis \$token e \$toSeek.

As repostas para as solicitações desses links serão tratadas no seguinte código:

```

32
33 Function RequireData($link){
34
35     $result = json_decode(file_get_contents($link));
36
37     if (!Empty($result)) {
38         return $result;
39     }
40
41     // Caso o Json volte vaizo
42     return null;
43 }
44
45 Function SeekVideo() {
46
47     $result = "";
48
49     if(isset($_POST["nome"])){
50         $nome=$_POST["nome"];
51
52     }
53
54     $result= json_decode(file_get_contents($url));
55     print_r($result);
56 }
57
58 ?>

```

-A primeira função tem como parâmetro a variável \$link, onde joga para dentro da variável \$result o json_decode que decodifica o JSON retornado, e logo após uma verificação para ver se a variável é vazia e retornou algum valor, caso contrário retorna vazio.

- A segunda função declara vazia a variável \$result, verificando se a variável \$_POST["nome"] está definida, caso verdadeiro ela joga para dentro da variável \$nome o valor do input, e logo após joga para dentro da variável \$result o json_decode que decodifica o JSON retornado.