1. **Introdução**

Como foi visto no exercício anterior, é possível que desenvolvedores criem serviços capazes de responder a requisições de um outro desenvolvedor. Esses serviços são construídos pensando no desacoplamento da aplicação, permitindo que elas se tornem independentes de sua interface.

A palavra-chave “desacoplamento” é muito importante nesse contexto. Por exemplo: imagine que você tem uma empresa de transporte. Essa empresa possui mais de 5 mil funcionários e dentro dessa empresa você desempenha o transporte de encomendas (cartas, produtos etc.) e também de passageiros. Considerando esse cenário, imagine que um cliente chega ao balcão de vendas e resolve enviar uma encomenda para um familiar. Esse cliente terá de fornecer todos os dados necessários para que a encomenda seja enviada. Após 3 meses, o mesmo cliente retorna a sua empresa e agora deseja viajar até a cidade do seu familiar. Sabemos que o sistema que trata a entrega de cargas é diferente do sistema que gerencia as viagens. A pergunta é:

Esse cliente terá de realizar novamente seu cadastro?

Pense um pouco... é realmente necessário que a base de dados que armazena os clientes do sistema transporte de carga seja diferente da base de dados do sistema de viagens?

1. **API’s**

No caso anterior, poderíamos unificar as bases de dados e todo sistema que precisasse de dados do cliente poderia consumir uma API que manipula os clientes. Essa é uma prática muito comum em grandes empresas que buscam integrar todos seus sistemas e evitar ao máximo o retrabalho.

1. **API’s públicas**

Agora que já entendemos o que é uma API, podemos falar um pouco mais do que são API’s públicas. Algumas empresas de diversas áreas permitem que usuários façam solicitações (*Requests)* de forma gratuita e sem precisar nem mesmo de autenticação (em alguns casos).

As API’s públicas são ferramentas interessantes para desenvolver algumas ferramentas que sem esse tipo de serviço seria impossível. Por exemplo, a *Google* permite que você utilize gratuitamente o *Google Maps* em sua aplicação, bem como os Correios (nacional) permite que você consulte o CEP da sua rua gratuitamente.

1. **Como utilizar?**

Em primeiro lugar precisamos encontrar uma API que seja passível de ser consumida publicamente. Para auxiliar nessa tarefa deixo a vocês uma pequena lista de sugestões:

- [Clima] - <https://api.hgbrasil.com/weather?woeid=452041>

- [CEP] - [https://viacep.com.br/ws/[digite o seu cep]/json/](https://viacep.com.br/ws/%5bdigite%20o%20seu%20cep%5d/json/)

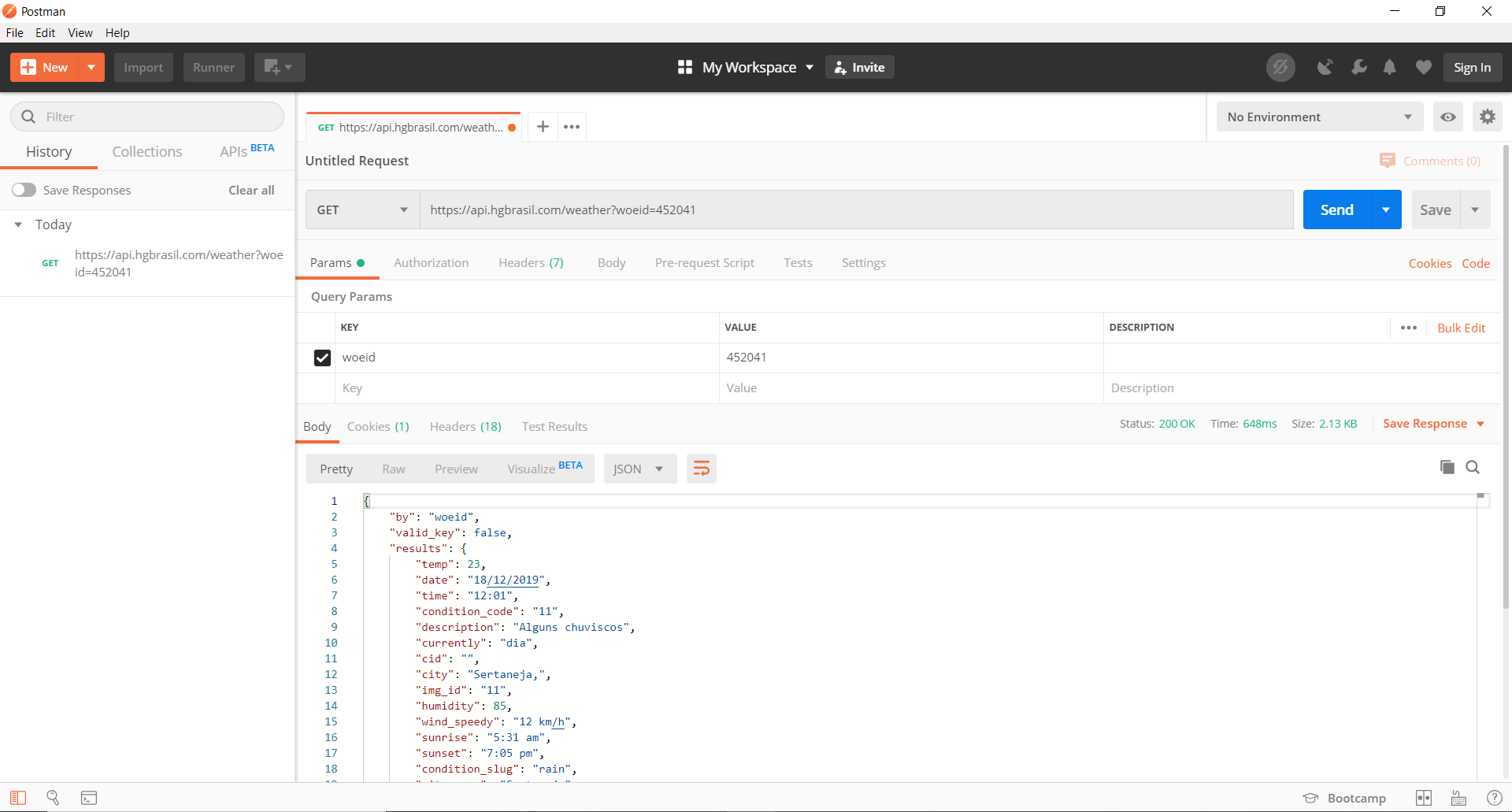
Você conhece outras? Dê sugestões!

Após sabermos qual API queremos consumir, podemos escolher uma das 2 formas mostradas a seguir:

**4.1- Usando um *software***

Podemos utilizar um *software* para consumir API’s, e essa é a forma mais simples e muito utilizada para testar o acesso a essa API. Por exemplo, você pode usar o seu próprio navegador para fazer requisições do tipo *GET* e manipular os parâmetros na URL.

Para você que precisa de algo mais elaborado, bem como usar requisições do tipo *POST, PUT, DELETE* etc. podemos utilizar *softwares* como o SOAP UI ou então o POSTMAN. O SOAP UI é um excelente *software*, porém não é gratuito, enquanto o POSTMAN é um excelente software e é totalmente gratuito para uso.



**4.2- Via código**

Logicamente podemos utilizar o PHP ou JavaScript para fazer requisições a API’s públicas. Essa prática é muito comum e nos nossos exemplos que serão apresentados a seguir iremos mostrar como fazê-lo.

* **Exemplo 01 - Consumindo API usando o PHP**

<!DOCTYPE html>

<html>

<head>

<title>Exemplo 01 - Consumindo via php</title>

</head>

<body>

<?php

$hg = file\_get\_contents("https://api.hgbrasil.com/weather?woeid=452041");

echo $hg;

?>

</body>

</html>

Veja como é simples fazer uma requisição usando o PHP. O documento HTML foi escrito apenas com uma *tag* PHP e usando a função nativa do PHP *file\_get\_contents*(). Essa função retorna o conteúdo da requisição e é possível manipular esse objeto da forma que você quiser. Lembrando que esse tipo de *request* é síncrona.

* **Exemplo 02 – Consumindo API usando JavaScript**

<!DOCTYPE html>

<html>

<head>

<title>Exemplo 02 - consumindo via Javascript</title>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

</head>

<body>

<div id="resultado"></div>

<script type="text/javascript">

$.get( "https://viacep.com.br/ws/13560530/json/", function( data ) {

console.log(data);

$("#resultado").append(JSON.stringify(data));

});

</script>

</body>

</html>

No código acima podemos ver que a requisição do tipo *GET* é feita a API apenas informando o CEP. O resultado é transformado em *String* e inserido no HTML quando o resultado do *Request* fica ponto.