

CURSO VUE JS + API PUBLICA

¿QUE ES UNA API?

Una API es una interfaz de programación de aplicaciones (del inglés API: Application Programming Interface). Es un conjunto de rutinas que provee acceso a funciones de un determinado software.

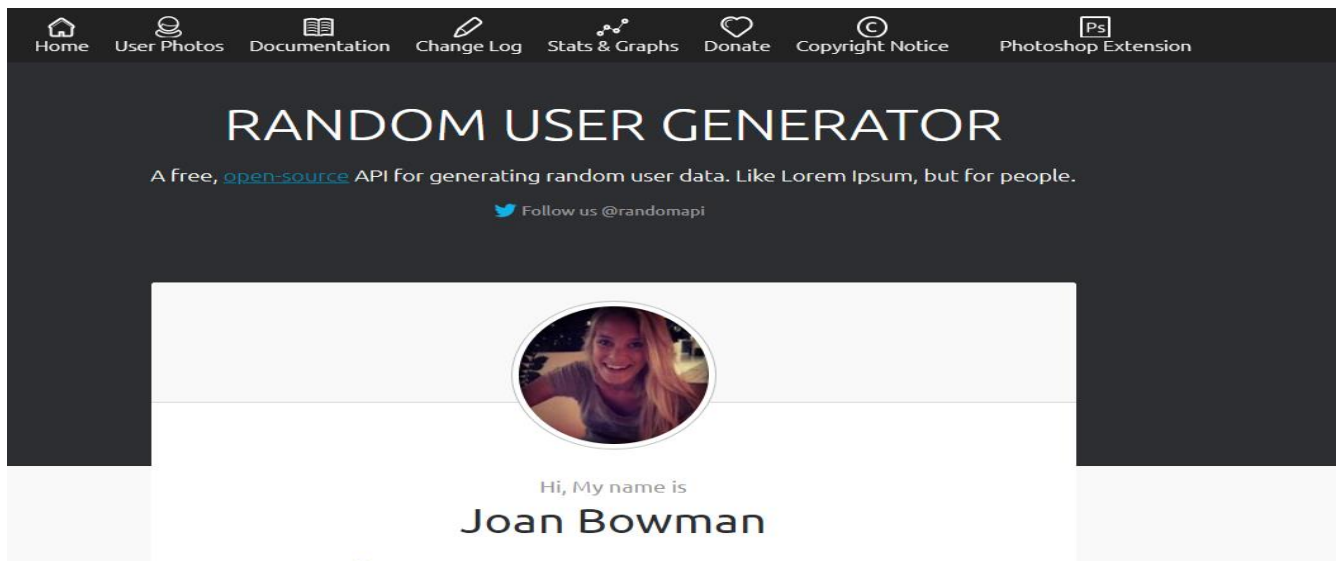
¿Como consumir una API con Vue JS?

Primero necesitan saber que UNA APLICACION **VUE** ES NETAMENTE VISUAL, NO ES CAPAZ DE GUARDAR DATOS EN LA BASE DE DATOS, PARA ESO NECESITAMOS UN SISTEMA COMPLETO COMO **LARAVEL**.

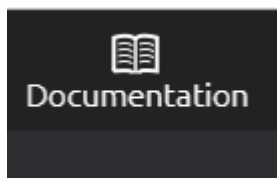
Aquí aprenderás a consumir una API publica para probar tu aplicación **Vue**. Para esto utilizaremos la herramienta:

RAMDON USER GENERATOR: <https://randomuser.me/>

Esta herramienta permite conectarnos a ella y traer datos en formato JSON



1) Hacemos click en DOCUMENTACION:



Click en RESULTS

Introduction	Seeds
How to use	Formats
Results	Versioning
Error Handling	Nationalities
Requesting multiple users	Pagination
Specifying a gender	Including/Excluding fields
Passwords	Miscellaneous

2) Necesitamos una API que nos devuelva datos como estos

```
{
  "results": [
    {
      "gender": "male",
      "name": {
        "title": "mr",
        "first": "rolf",
        "last": "hegdal"
      },
      "location": {
        "street": "l'jan terrasse 346",
        "city": "vear",
        "state": "rogaland",
        "postcode": "3095",
        "coordinates": {
          "latitude": "54.8646",
          "longitude": "-97.3136"
        },
        "timezone": {
          "offset": "-10:00",
          "description": "Hawaii"
        }
      },
      "email": "rolf.hegdal@example.com",
      "login": {
        "uuid": "c4168eac-84b8-46ea-b735-c9da9bfb97fd",
        "username": "bluefrog786",
        "password": "ingrid",
        "salt": "GtRFz4NE",
        "md5": "5c581c5748fc8c35bd7f16eac9efbb55",
        "sha1": "c3feb8887abed9ec1561b9aa2c9f58de21d1d3d9",
        "sha256": "684c478a98b43f1ef1703b35b8bbf61b27dbc93d52acd515e141e97e04447712"
      },
      "dob": {
        "date": "1975-11-12T06:34:44Z",
        "age": 42
      }
    },
    ...
  ]
}
```

Nota IMPORTANTE

Existen otros formatos para traer datos, pero el más utilizado es el formato JSON, ya que es el más fácil de comprender.

3) Al bajar encontraran un enlace

Requesting Multiple Users

Random User Generator allows you to fetch up to 5,000 generated users in one request using the **results** parameter.

```
https://randomuser.me/api/?results=5000
```

<https://randomuser.me/api/?results=5000>

Este enlace dice, que nos devolverá 5000 resultados, pero para probarlo nosotros solo usaremos 10 resultados.

Cambiamos el numero 5000 por 10

```
https://randomuser.me/api/?results=10
```

<https://randomuser.me/api/?results=10>

Pueden elegir la cantidad de datos que deseen

Configurando nuestra aplicación.

1) Iniciaremos creando nuestro archivo HTML

```
<!DOCTYPE html>
<html Lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

Agregaremos el CDN de vue js

```
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
```

****Vue por si solo no puede trabajar con Ajax, necesita de un complemento que si permita esta funcionalidad. Es por eso que agregamos este Script****

```
<script src="https://cdn.jsdelivr.net/npm/vue-resource@1.5.1/vue-resource.min.js"></script>
```

Este se puede encontrar en la página CNDJS escribiendo VUE RESOURCES

Configuramos nuestro componente Vue:

```
<script>
```

```
new Vue({
```

```
el: '#app',
```

```
data: {
```

```
},
```

```
});
```

```
</script>
```

Nuestra aplicación debe verse así:

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <meta http-equiv="X-UA-Compatible" content="ie=edge">
7     <title>Document</title>
8   </head>
9   <body>
10    <!--Referencia del componente VUE-->
11    <div id="app">
12      <!--Todo Lo que este aqui se hara con Vue Si esta por fuera, no funcionara.-->
13    </div>
14
15    <!--CDN DE VUE-->
16    <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
17    <!--CDN DEL RECURSO UTILIZADO PARA USAR AJAX-->
18    <script src="https://cdn.jsdelivr.net/npm/vue-resource@1.5.1/vue-resource.min.js"></script>
19    <!--Componente VUE-->
20    <script>
21      new Vue({
22        el: '#app',
23        data() {
24          return {
25
26          }
27        },
28      });
29    </script>
30  </body>
31 </html>
```

Para el estilo de la página agregamos el CDN DE BOOTSTRAP

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQV3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
```

>Ya podemos comenzar<

Lo primero que debemos hacer es, conocer la ruta con la que vamos a trabajar, en este caso usaremos esta:

<https://randomuser.me/api/?results=10>

Esta ruta la guardaremos en una variable

```
<script>
var url="https://randomuser.me/api/?results=10"

new Vue({
  el: '#app',

  data:{
  },
});
</script>
```

Creamos una variable dentro del Data, en este caso se llamará “**lista**” y sera un array

```
data:{
  lista:[]
},
```

Debajo del data, necesitaremos un método, este método se va llamar **GetUsuarios**

Nuestro script debe verse así

```
<script>
var url="https://randomuser.me/api/?results=10"

new Vue({
  el: '#app',

  data:{
    lista:[]
  },
  methods:{
    getUsuarios:function(){

    }
  }
});
</script>
```

A partir de aquí crearemos la conexión al servidor, trayendo los datos de la URL, almacenándolos en la variable “lista” y luego mostrarlos en formato HTML.

Nuestro método debe verse así:

```
getUsuarios:function(){
  // "$http" Esto lo podemos hacer gracias a que importamos el recurso AJAX que esta arriba
  this.$http.get(url).then(function(response){
    this.lista = response
  });
}
```

```
-----  
  
this.$http.get(url).then(function(response){  
  
this.lista = response  
  
});
```

- Explicación breve y detallada*

`this.$http.get(url)` = Traemos los datos con el método Get y lo conectamos con la variable URL

```
-----  
  
.then(function(response){  
  
this.lista = response
```

`});` = `.then` (Si todo sale bien) creamos una función con el parámetro (Respuesta) esta función nos guardará todos los datos si todo sale bien. `this.lista = response` aquí llenamos la variable lista con los datos.

Configuramos un poco el diseño

>Así debe verse por ahora



Nuestra página debe verse así

Nuestra lista está vacía, ya que todo lo tenemos guardado en el método, pero aún no le hemos dicho a Vue que ejecute el método (`getUsuarios`), vamos a hacerlo.

```
created:function(){  
  
this.getUsuarios();  
  
},
```

Aquí le decimos a Vue, que ejecute nuestro método en su etapa de creación. Debe verse así:

```
'''  
k/script>
```

Nota importante

Los componentes Vue, tienen varias etapas, en este caso utilizamos la de creación, para entender esto revisen el siguiente link:

<https://vuejs.org/v2/guide/instance.html#Lifecycle-Diagram>

Ahora podrán ver su página y verán todos los datos de la Url.

Pero, si se fijan verán datos que no son necesarios, por ejemplo:

},

Estos son los datos que nosotros necesitamos:

```
"body": {
  "results": [
    {
      "gender": "male",
      "name": {
        "title": "monsieur",
        "first": "yvan",
        "last": "garcia"
      },
      "location": {
        "street": "4629 avenue debrousse",
        "city": "châtel-sur-montsalvens",
        "state": "luzern",
        "postcode": 9743,
        "coordinates": {
          "latitude": "-50.4729",
          "longitude": "-18.2385"
        },
        "timezone": {
          "offset": "+7:00",
          "description": "Bangkok, Hanoi, Jakarta"
        }
      }
    }
  ],
}
```

Estos datos se almacenan en el array “results”, ahora vamos a modificar lo que mostramos en la página.

Vamos a nuestro método (getUsuarios) y añadimos “.data” debe verse así.

```
getUsuarios:function(){
  // "$http" Esto lo podemos hacer gracias a que importamos el recurso AJAX que esta arriba
  this.$http.get(url).then(function(response){
    this.lista = response.data
  });
}
```

Vayan a la página y verán que ahora solo muestra los datos que nos interesan, sin embargo, el array que necesitamos, se llama “results”

```

1
"lista": {
  "results": [
    {
      "gender": "male",
      "name": {
        "title": "mr",
        "first": "hunter",
        "last": "patel"
      },
      "location": {
        "street": "6181 george st",
        "city": "wellington",
        "state": "nunavut",
        "postcode": "Z8C 7Y6",
        "coordinates": {
          "latitude": "-60.3373",
          "longitude": "-111.5378"
        },
        "timezone": {
          "offset": "+4:00",
          "description": "Abu Dhabi, Muscat, Baku, Tbilisi"
        }
      }
    }
  ]
}

```

Igualmente, para ingresar de lleno a estos datos, en nuestro método añadimos “results”, así:

```

getUsuarios:function(){
  // "$http" Esto lo podemos hacer gracias a que importamos el recurso AJAX que esta arriba
  this.$http.get(url).then(function(response){
    //Data ignora el estatus de la respuesta
    this.lista = response.data.results
  });
  //.<nombre del array> ingresa de lleno a los datos necesarios para listar
}

```

Ahora verán en su página, que todos los datos están en nuestro array “lista”


```

    {
"lista": [
  {
    "gender": "male",
    "name": {
      "title": "mr",
      "first": "william",
      "last": "white"
    },
    "location": {
      "street": "9315 disputed rd",
      "city": "red rock",
      "state": "nunavut",
      "postcode": "E6D 1Q0",
      "coordinates": {
        "latitude": "-58.3182",
        "longitude": "-149.8190"
      },
      "timezone": {
        "offset": "+8:00",
        "description": "Beijing, Perth, Singapore, Hong Kong"
      }
    },
    .....
```

LISTAMOS LOS DATOS EN HTML

Diseñamos nuestro html

Usamos v-for para recorrer el array.

```

<div class="col-sm-4" >
  <h5>Lista de usuarios VUE + AJAX</h5>
  <ul class="list-group">
    <!--Recorremos nuestro array con un FOR-->
    <li v-for="item in lista" class="list-group-item">
      <!--listamos los datos json que traemos-->
    </li>
  </ul>
</div>
```

En el ahora utilizamos {{item.name}} para listar los nombres de los usuarios.

```

<ul class="list-group">
  <!--Recorremos nuestro array con un FOR-->
  <li v-for="item in lista" class="list-group-item">
    <!--Listamos los datos json que traemos-->
    {{item.name.first}} - {{item.name.last}}
  </li>
</ul>

```

Item.name: Hace referencia al atributo: NAME del usuario

Item.name.first: hace referencia al primer nombre del usuario

Item.name.last: hace referencia al último nombre del usuario

```

"name": {
  "title": "mr",
  "first": "soan",
  "last": "riviere"
},

```

Nota importante

Esto es un simple ejemplo, ustedes podrían llamar cualquier atributo del usuario.

¿Como conocemos el atributo? Ejemplo:

```

"gender": "male",
"__name__": {
  "title": "monsieur",
  "first": "cyrill",
  "last": "rodriguez"
},
"__location__": {
  "street": "3816 rue pasteur",
  "city": "aegerten",
  "state": "zug",
  "postcode": 6053,
  "coordinates": {
    "latitude": "-48.3170",
    "longitude": "-150.8550"
  },
  "timezone": {
    "offset": "-6:00",
    "description": "Central Time (US & Canada), Mexico City"
  }
}

```

Felicitaciones, has utilizado una API publica :)

Resumen

Para consumir una api lo único que necesitamos es una aplicación en internet, que nos devuelva en formato Json una serie de datos.

=====

Bibliografia

>https://es.wikipedia.org/wiki/Web_API

><https://rimorsoft.com/>

AUTOR:

Angel Junior Torres Archbold