

DOCUMENTACION

API RESTFUL

CONFIGURACIÓN INICIAL –CREDENCIALES BASE DE DATOS

En la ruta `models/connection.php` dirigirse al método `infoDatabase()`

Allí se configura la conexión a la base de datos, agregando el nombre de la base de datos, el usuario y la contraseña

```
/*=====
Información de la base de datos
=====*/

4 references | 0 overrides
static public function infoDatabase(): array{
    ...
    $infoDB = array(
        "database" => "database",
        "user" => "root",
        "pass" => ""
    );

    return $infoDB;
}
```

CONFIGURACIÓN INICIAL – APIKEY

En la ruta `models/connection.php` dirigirse al método `apikey()`

Allí agregas la llave segura de la API (Una contraseña creada por el administrador).

Esta llave solo la conocerá el administrador de la API, y es la clave que se usará como Autorización en las cabeceras de cada solicitud HTTP.

```
/*=====
APIKEY
=====*/

1 reference | 0 overrides
static public function apikey(): string{
    ...
    return "asdafasf87a98r74fg65j798ertyu4t3";
}
```

CONFIGURACIÓN INICIAL – ACCESO PÚBLICO

En la ruta `models/connection.php` dirigirse al método `publicAccess()`

Allí agregas dentro de la matriz las tablas (separadas por coma) que serán de dominio público, es decir, la información a la que podrán acceder de forma libre los usuarios de la API.

```
/*=====
Acceso público
=====*/

1 reference | 0 overrides
static public function publicAccess(): array{
    $tables = [""];

    return $tables;
}
```

CONFIGURACIÓN INICIAL – Base de datos

Las tablas deben estar escritas en plural y el nombre de las columnas deben terminar con el nombre de la tabla en singular (_sufijo)*.

Tabla: **categories**

Columnas:

- **id_category**
- **name_category**
- **date_created_category**
- **date_updated_category**

*Recomendación: Usar idioma inglés para acortar palabras y unir palabras con guion de pis

Estructura de tabla		Vista de relaciones							
#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1 id_category 	int(11)			No	Ninguna		AUTO_INCREMENT	 Cambiar  Eliminar  Más
<input type="checkbox"/>	2 name_category	text	utf8_general_ci		Sí	NULL			 Cambiar  Eliminar  Más
<input type="checkbox"/>	3 date_created_category	date			Sí	NULL			 Cambiar  Eliminar  Más
<input type="checkbox"/>	4 date_updated_category	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	 Cambiar  Eliminar  Más

CONFIGURACIÓN INICIAL – Base de datos

La primera columna debe ser el ID (auto incrementable) y las dos últimas columnas deben ser la fecha de creación y la fecha de actualización.

Tabla: **categories**

Columnas:

- **id_category**

... Se agregan las columnas necesarias ...

- **name_category**
- **date_created_category**
- **date_updated_category**

Estructura de tabla		Vista de relaciones							
#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1 id_category 	int(11)			No	Ninguna		AUTO_INCREMENT	 Cambiar  Eliminar  Más
<input type="checkbox"/>	2 name_category	text	utf8_general_ci		Sí	NULL			 Cambiar  Eliminar  Más
<input type="checkbox"/>	3 date_created_category	date			Sí	NULL			 Cambiar  Eliminar  Más
<input type="checkbox"/>	4 date_updated_category	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	 Cambiar  Eliminar  Más

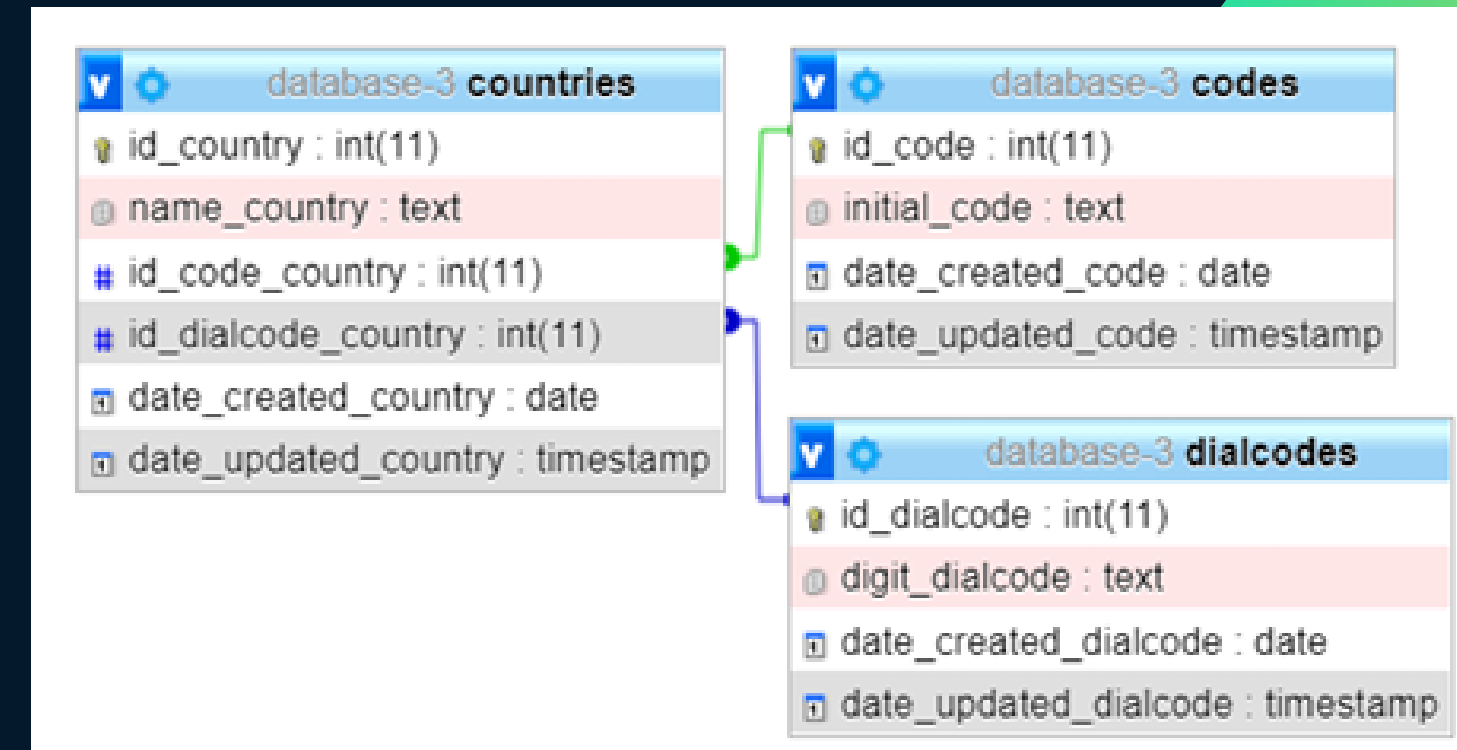
CONFIGURACIÓN INICIAL – Base de datos

Para relacionar tablas, el número del ID de la Tabla Relacionada debe estar en una columna de la Tabla Principal, y dicha columna debe tener el siguiente orden de palabras: id_(Tabla Relacionada en singular)_(Tabla Principal en singular)

Tabla Principal: **countries**

Tabla Relacionada: **codes** | Columna: **id_code_country**

Tabla Relacionada: **dialcodes** | Columna: **id_dialcode_country**



CONFIGURACIÓN INICIAL – Base de datos

Para realizar autenticaciones con la API RESTful, es indispensable que la tabla de autenticación tenga siempre estas 4 columnas

- email_sufijo
- password_sufijo
- token_sufijo
- token_exp_sufijo

Estructura de tabla		Vista de relaciones							
#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1 id_user	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
<input type="checkbox"/>	2 name_user	text	utf8_general_ci		Si	NULL			Cambiar Eliminar Más
<input type="checkbox"/>	3 email_user	text	utf8_general_ci		Si	NULL			Cambiar Eliminar Más
<input type="checkbox"/>	4 password_user	text	utf8_general_ci		Si	NULL			Cambiar Eliminar Más
<input type="checkbox"/>	5 token_user	text	utf8_general_ci		Si	NULL			Cambiar Eliminar Más
<input type="checkbox"/>	6 token_exp_user	text	utf8_general_ci		Si	NULL			Cambiar Eliminar Más
<input type="checkbox"/>	7 date_created_user	date			Si	NULL			Cambiar Eliminar Más
<input type="checkbox"/>	8 date_updated_user	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Cambiar Eliminar Más

CONFIGURACIÓN INICIAL – Autorizacion

Todas las solicitudes a la API deben llevar en la cabecera (HEADERS) la propiedad **Authorization** con la respectiva **APIKEY** creada por el administrador.

HEADERS

Authorization

c5LTA6WPbMwHhEabYu77nN9cn4VcMj

Solicitudes de tipo Get

La **selección básica**, consiste simplemente en colocar el nombre de la tabla luego del ENDPOINT.

*Nota: Si el nombre de la tabla está mal escrito o no existe en la base de datos recibirá como respuesta un status 404.

```
1 {  
2   "status": 404,  
3   "results": "Not Found",  
4   "method": "get"  
5 }
```

GET GET - Selección básica

<http://apirest.com/instructors>

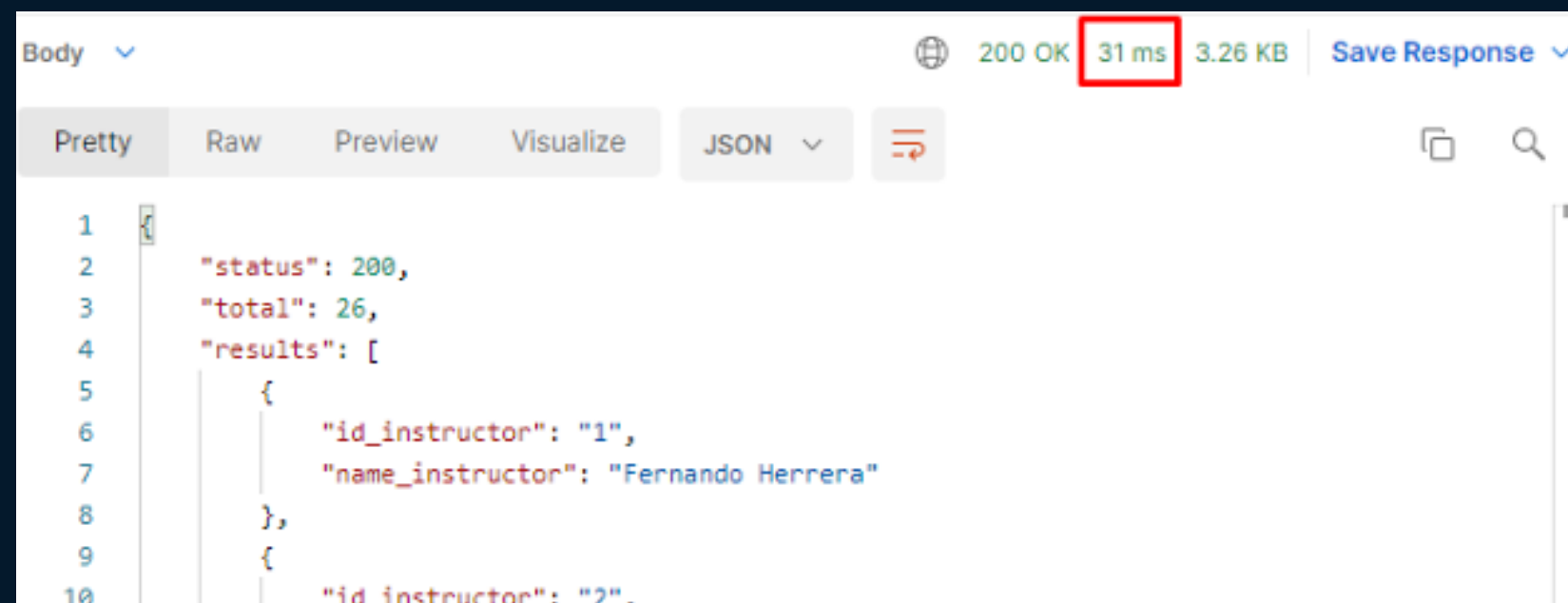
HEADERS

Authorization	c5LTA6WPbMwHhEabYu77nN9cn4VcMj
---------------	--------------------------------

Solicitudes de tipo Get

Seleccionar un dato específico, consiste en agregar el parámetro **select** a la **url** y colocar el nombre exacto (sin espacios a los lados) de una columna que exista en la tabla. Puede agregar las columnas que desee separándolas por coma.

*Nota: Esto ayuda a optimizar el tiempo de entrega de la información.



```
Body
200 OK 31 ms 3.26 KB Save Response
Pretty Raw Preview Visualize JSON
{
  "status": 200,
  "total": 26,
  "results": [
    {
      "id_instructor": "1",
      "name_instructor": "Fernando Herrera"
    },
    {
      "id_instructor": "2",
```

GET GET - Seleccionar dato específico

`http://apirest.com/instructors?select=id_instructor,name_instructor`

HEADERS

Authorization c5LTA6WPbMwHhEabYu77nN9cn4VcMj

PARAMS

select id_instructor,name_instructor

Solicitudes de tipo Get

Filtrar una selección con un solo valor, consiste en agregar los parámetros `linkTo` y `equalTo` a la `url`. En el primer parámetro escribir el nombre de columna donde se desea encontrar la coincidencia, y en el segundo parámetro colocar el valor exacto de lo que se desea filtrar.

*Nota: Evita filtrar contenido de textos que contengan alguna coma.

GET GET - Filtro de un solo valor

```
http://apiREST.com/courses?select=id_course,title_course,description_course&linkTo=title_course&equalTo=Git y GitHub Completo Desde Cero
```

HEADERS

Authorization	c5LTA6WPbMwHhEabYu77nN9cn4VcMj
---------------	--------------------------------

PARAMS

select	id_course,title_course,description_course
linkTo	title_course
equalTo	Git y GitHub Completo Desde Cero

Solicitudes de tipo Get

Filtrar una selección con varios valores, consiste en agregar más columnas separadas por comas al parámetro `linkTo` y en agregar más valores separados por coma en el parámetro `equalTo`.

*Nota: Agregar un solo valor por cada columna adicional.

GET GET - Filtros de varios valores

```
http://apirest.com/courses?select=id_course,title_course,description_course,id_instructor_course&linkTo=title_course,id_instructor_course&equalTo=Git y GitHub Completo Desde Cero,3
```

HEADERS

Authorization	c5LTA6WPbMwHhEabYu77nN9cn4VcMj
---------------	--------------------------------

PARAMS

select	id_course,title_course,description_course,id_instructor_course
linkTo	title_course,id_instructor_course
equalTo	Git y GitHub Completo Desde Cero,3

Solicitudes de tipo Get

Ordenar datos, consiste en agregar los parámetros **orderBy** y **orderMode** e indicar en **orderBy** la columna que ordenará los datos y usar las palabras reservadas **ASC** o **DESC** en el parámetro **orderMode** para indicar el tipo de orden (Ascendente o Descendente).

*Nota: El orden se puede aplicar a números, alfabeto o fechas.

GET GET - Ordenar datos

```
http://apirest.com/courses?select=title_course&orderBy=title_course&orderMode=ASC
```

HEADERS

Authorization	c5LTA6WPbMwHhEabYu77nN9cn4VcMj
---------------	--------------------------------

PARAMS

select	title_course
orderBy	title_course
orderMode	ASC

Solicitudes de tipo Get

Limitar datos, consiste en agregar los parámetros **startAt** y **endAt** e indicar en **startAt** la posición inicial desde donde se desea limitar los datos (la primera posición es el 0) y en el parámetro **endAt** indicar la cantidad de registros que se desean seleccionar.

*Nota: Se puede combinar el orden y los límites, pero siempre el orden será la prioridad para la API.

GET GET - Limitar datos

```
http://apirest.com/courses?select=title_course,description_course&orderBy=title_course&orderMode=DESC&startAt=0&endAt=5
```

HEADERS

Authorization	c5LTA6WPbMwHhEabYu77nN9cn4VcMj
---------------	--------------------------------

PARAMS

select	title_course,description_course
orderBy	title_course
orderMode	DESC
startAt	0
endAt	5

Solicitudes de tipo Get

Relaciones entre tablas sin filtro, consiste en traer información entre varias tablas que estén relacionadas por su ID sin filtrar información.

La estructura de la URL cambia:

- Debemos agregar la palabra **relations** luego del ENDPOINT.
- Agregar el parámetro **rel**: Allí indicar en primer lugar el nombre de la tabla principal y luego separado por comas las tablas relacionadas.
- Luego agregar el parámetro **type**: Allí agregar los sufijos de las tablas puestas en el parámetro **rel**.

*Nota: Al no filtrar datos traemos el total de información de todas las tablas en relación de acuerdo a los datos seleccionados.

GET GET - Relaciones entre tablas sin filtro

```
http://apirest.com/relations?rel=courses,instructors&type=course,instructor&select=name_instructor,title_course,id_course&orderBy=id_course&orderMode=ASC&startAt=0&endAt=6
```

HEADERS

Authorization	c5LTA6WPbMwHhEabYu77nN9cn4VcMj
---------------	--------------------------------

PARAMS

rel	courses,instructors
type	course,instructor
select	name_instructor,title_course,id_course
orderBy	id_course
orderMode	ASC
startAt	0
endAt	6

Solicitudes de tipo Get

Relaciones entre tablas con filtros, consiste en traer información entre varias tablas que estén relacionadas por su ID filtrando información.

Se repite exactamente el mismo procedimiento que se hace en la solicitud de tablas relacionadas agregando los parámetros **linkTo** y **equalTo** para buscar coincidencias en una o varias columnas de cualquiera de las tablas que se estén relacionando.

```
Body 200 OK 10 ms 618 B Save Response
Pretty JSON
1 {
2   "status": 200,
3   "total": 1,
4   "results": [
5     {
6       "name_country": "Colombia",
7       "initial_code": "CO",
8       "digit_dialcode": "+57"
```

GET GET - Relaciones entre tablas con filtro	
http://apirest.com/relations?rel=countries,codes,dialcodes&type=country,code,dialcode&select=name_country,initial_code,digit_dialcode&linkTo=name_country&equalTo=Colombia	
HEADERS	
Authorization	c5LTA6WPbMwHhEabYu77nN9cn4VcMj
PARAMS	
rel	countries,codes,dialcodes
type	country,code,dialcode
select	name_country,initial_code,digit_dialcode
linkTo	name_country
equalTo	Colombia
startAt	0
endAt	6

Solicitudes de tipo Get

Búsqueda sin relaciones y con filtros, consiste en realizar búsquedas en la base de datos en una sola tabla usando palabras claves y filtrando datos exactos. Usamos el parámetro **linkTo** para el nombre de columnas y agregamos el parámetro **search** para la búsqueda de la palabra clave y datos exactos.

*Nota: El primer valor de **linkTo** debe ser siempre la columna en donde se realizará la búsqueda, de igual forma, el primer valor de **search** debe ser la palabra clave. Luego con comas aplicamos los filtros.

```
Body  Cookies  Headers (10)  Test Results
Pretty  Raw  Preview  Visualize  JSON  [icon]
1  {
2    "status": 200,
3    "total": 1,
4    "results": [
5      {
6        "id_product": "3",
7        "name_product": "Sound Intone I65 Earphone White Version",
8        "id_category_product": "3",
9        "date_created_product": "2021-01-19",
10       "date_updated_product": "2021-10-27 06:53:17"
11      }
12    ]
13  }
```

GET GET - Búsqueda sin relaciones y con filtros

http://apirest.com/products?select=*&linkTo=name_product,id_category_product&search=sound,3

HEADERS

Authorization c5LTA6WPbMwHhEabYu77nN9cn4VcMj

PARAMS

select	*
linkTo	name_product,id_category_product
search	sound,3

Solicitudes de tipo Get

Búsqueda con relaciones y con filtros, consiste en realizar búsquedas en la base de datos en entre varias tablas relacionadas usando palabras claves y filtrando datos exactos. Además de los parámetros **linkTo** y **search**, usamos los parámetros de relación **rel** y **type**.

```
Body Cookies Headers (10) Test Results
Pretty Raw Preview Visualize JSON
1 {
2   "status": 200,
3   "total": 1,
4   "results": [
5     {
6       "name_product": "Sound Intone I65 Earphone White Version",
7       "name_category": "Home, Garden and Kitchen"
8     }
9   ]
10 }
```

GET GET - Búsqueda con relaciones y con filtros

```
http://apirest.com/relations?rel=products,categories&type=product,category&select=name_product,
name_category&linkTo=name_product,id_category_product&search=Sound,3
```

HEADERS

Authorization	c5LTA6WPbMwHhEabYu77nN9cn4VcMj
---------------	--------------------------------

PARAMS

rel	products,categories
type	product,category
select	name_product,name_category
linkTo	name_product,id_category_product
search	Sound,3

Solicitudes de tipo Get

Rangos con filtros, consiste en seleccionar un rango de datos entre números o fechas en una sola tabla agregando los parámetros **between1** para el rango inicial y **between2** para el rango final, filtramos la información con el parámetro **filterTo** indicando el nombre de columna y el parámetro **inTo** para indicar el valor del filtro.

```
Body  Cookies  Headers (10)  Test Results
Pretty  Raw  Preview  Visualize  JSON  [icon]
1  {
2    "status": 200,
3    "total": 3,
4    "results": [
5      {
6        "id_product": "1",
7        "name_product": "Marshall Kilburn Portable Wireless",
8        "id_category_product": "1",
9        "date_created_product": "2021-01-19",
10       "date_updated_product": "2021-10-27 06:52:59"
11      },
12      {
13        "id_product": "7",
14        "name_product": "Samsung UHD TV 24inch",
15        "id_category_product": "1",
16        "date_created_product": "2021-02-10"
```

GET GET - Rangos con filtros

```
http://apirest.com/products?linkTo=date_created_product&between1=2021-01-01&between2=2021-04-30&select=*&filterTo=id_category_product&inTo=1
```

HEADERS

Authorization	c5LTA6WPbMwHhEabYu77nN9cn4VcMj
---------------	--------------------------------

PARAMS

linkTo	date_created_product
between1	2021-01-01
between2	2021-04-30
select	*
filterTo	id_category_product
inTo	1

Solicitudes de tipo GET

Rangos entre relaciones con filtros, consiste en seleccionar un rango de datos entre varias tablas relacionadas. Se realiza la solicitud de igual forma a la anterior agregando los parámetros de relación.

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": 200,
3   "total": 3,
4   "results": [
5     {
6       "name_product": "Marshall Kilburn Portable Wireless",
7       "name_category": "Consumer Electric",
8       "date_created_product": "2021-01-19"
9     },
10    {
11      "name_product": "Samsung UHD TV 24inch",
12      "name_category": "Consumer Electric",
13      "date_created_product": "2021-02-19"
14    },
15    {
16      "name_product": "Hazen Military Classical Backpack"
```

GET GET - Rangos entre relaciones con filtros

http://apirest.com/rerelations?rel=products,categories&type=product,category&linkTo=date_created_p
roduct&between1=2021-01-01&between2=2021-04-30&select=name_product,name_category,date_c
reated_product&filterTo=id_category_product&inTo=1

HEADERS

Authorization c5LTA6WPbMwHhEabYu77nN9cn4VcMj

PARAMS

rel	products,categories
type	product,category
linkTo	date_created_product
between1	2021-01-01
between2	2021-04-30
select	name_product,name_category,date_created_product
filterTo	id_category_product
inTo	1

Solicitudes de tipo POST

Registro usuarios, consiste en registrar usuarios en la base de datos para luego generar el token de seguridad con su respectiva fecha de vencimiento al momento de realizar el login.

Debemos agregar luego del ENDPOINT la tabla donde se realizará el registro de usuario, acompañado de los parámetros **register** con valor **true** y el parámetro **suffix** con el nombre de la tabla en singular. Los datos mínimos del formulario deben ser el email y el password.

*Nota: Si la tabla donde se realiza el registro tiene como nombre **users**, no es necesario enviar el parámetro **suffix**.

POST Registro de usuario

`http://apirest.com/instructors?register=true&suffix=instructor`

HEADERS

Authorization `c5LTA6WPbMwHhEabYu77nN9cn4VcMj`

PARAMS

register `true`

suffix `instructor`

BODY urlencoded

email_instructor `instructor@correo.com`

password_instructor `123456`

Solicitudes de tipo POST

Login de usuarios, es la acción de solicitar el token de seguridad con su respectiva fecha de vencimiento usando el email y contraseña que usó en el registro.

Debemos agregar luego del ENDPOINT la tabla donde se hizo el registro de usuario, acompañado de los parámetros **login** con valor true y el parámetro **suffix** con el nombre de la tabla en singular.

POST Login de usuario

`http://apiREST.com/instructors?login=true&suffix=instructor`

HEADERS

Authorization c5LTA6WPbMwHhEabYu77nN9cn4VcMj

PARAMS

login true

suffix instructor

BODY urlencoded

email_instructor instructor@correo.com

password_instructor 123456

Solicitudes de tipo POST

Crear registros, es la acción de crear información en una tabla de la base de datos.

Debemos agregar luego del ENDPOINT la tabla donde se desea crear el nuevo registro, acompañado de los parámetros **token** con el valor del token entregado luego del login y los parámetros **table** y **suffix** con el nombre de la tabla en plural y singular de donde viene el usuario registrado.

*Nota: Los nombres de los campos del formulario deben ser exactos a los nombres de las columnas de la tabla donde se realizará el registro.

POST POST - Crear registros

```
http://apirest.com/courses?token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOiJlE2MzU0NTcwNzAsImV4cCI6MTYzNTU0MzQ3MCwiZGF0YSI6eyJpZCI6IjMwIiwiaWZwIjEhaWwiOiJpbnN0cnVjdG9yQGNvcnJlby5jb20iOiJ0LnVav6yWEBJR0r9VmfeWuBvQBw7jp8-a7Vchhm9Vqmc&table=instructors&suffix=instructor&except
```

HEADERS

Authorization	c5LTA6WPbMwHhEabYu77nN9cn4VcMj
---------------	--------------------------------

PARAMS

token	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOiJlE2MzU0NTcwNzAsImV4cCI6MTYzNTU0MzQ3MCwiZGF0YSI6eyJpZCI6IjMwIiwiaWZwIjEhaWwiOiJpbnN0cnVjdG9yQGNvcnJlby5jb20iOiJ0LnVav6yWEBJR0r9VmfeWuBvQBw7jp8-a7Vchhm9Vqmc
table	Instructors
suffix	instructor
except	

BODY urlencoded

title_course	Aprende PHP de cero a avanzado
--------------	--------------------------------

Solicitudes de tipo PUT

SOLICITUDES DE TIPO PUT

Editar registros, es la acción de cambiar información en una tabla de la base de datos.

Debemos agregar luego del ENDPOINT la tabla donde se desea editar el registro, acompañado de los parámetros token con el valor del token entregado luego del login.

Además agregar los siguientes parámetros:

- El parámetro **id** para indicar el id del registro a editar.
- El parámetro **nameId** para indicar el nombre de la columna donde se buscará el id.
- Los parámetros **table** y **suffix** con el nombre de la tabla en plural y singular de donde viene el usuario registrado.

*Nota: Los nombres de los campos del formulario deben ser exactos a los nombres de las columnas de la tabla donde se editará el registro.

PUT PUT - Editar registros

```
http://apiREST.com/courses?id=67&nameId=id_course&token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOiJlY2MzU0NTcwNzAsImV4cCI6MTYzNTU0MzQ3MCwiZGF0YSI6eyJpZCI6IjMwIiwiaWZwIjoihaWwiOiJpbmN0cnVjdG9yQG9yQGNvcnJlby5jb20ifX0.LnVav6yWEBJR0r9VmfeWuBvQBw7jp8-a7Vchhm9Vqmc&table=instructors&suffix=instructor
```

HEADERS

Authorization	c5LTA6WPbMwHhEabYu77nN9cn4VcMj
---------------	--------------------------------

PARAMS

id	67
nameId	id_course
token	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOiJlY2MzU0NTcwNzAsImV4cCI6MTYzNTU0MzQ3MCwiZGF0YSI6eyJpZCI6IjMwIiwiaWZwIjoihaWwiOiJpbmN0cnVjdG9yQGNvcnJlby5jb20ifX0.LnVav6yWEBJR0r9VmfeWuBvQBw7jp8-a7Vchhm9Vqmc
table	instructors
suffix	instructor

BODY urlencoded

title_course	Aprende PHP de cero a avanzado
description_course	Aprende todas las bases de este maravillos lenguaje

Solicitudes de tipo DELETE

Eliminar registros, es la acción de eliminar información en una tabla de la base de datos.

Debemos agregar luego del ENDPOINT la tabla donde se desea eliminar el registro, acompañado de los parámetros **token** con el valor del token entregado luego del login.

Además agregar los siguientes parámetros:

- El parámetro **id** para indicar el id del registro a eliminar.
- El parámetro **nameId** para indicar el nombre de la columna donde se buscará el id.
- Los parámetros **table** y **suffix** con el nombre de la tabla en plural y singular de donde viene el usuario registrado.

DEL DELETE - Eliminar registros

```
http://apiREST.com/courses?id=66&nameId=id_course&token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOiJlY2MzU0NTcwNzAsImV4cCI6MTYzNTU0MzQ3MCwiZGF0YSI6eyJpZCI6IjMwIiwiaWZwIjhaWwiOiJpbnN0cnVjdG9yQGNvcnJlby5jb20ifX0.LnVav6yWEBJR0r9VmfeWuBvQBw7jp8-a7Vchhm9Vqmc&table=instructors&suffix=instructor
```

HEADERS

Authorization	c5LTA6WPbMwHhEabYu77nN9cn4VcMj
---------------	--------------------------------

PARAMS

id	66
nameId	id_course
token	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOiJlY2MzU0NTcwNzAsImV4cCI6MTYzNTU0MzQ3MCwiZGF0YSI6eyJpZCI6IjMwIiwiaWZwIjhaWwiOiJpbnN0cnVjdG9yQGNvcnJlby5jb20ifX0.LnVav6yWEBJR0r9VmfeWuBvQBw7jp8-a7Vchhm9Vqmc
table	instructors
suffix	instructor

Crear editar un registro con excepción

CREAR O EDITAR UN REGISTRO CON EXCEPCIÓN

Es la acción de crear o editar un registro sin la necesidad de usar el token de seguridad.

El parámetro **token** debe ir con el valor **no**, y agregamos el parámetro **except** para indicar cualquier nombre de columna de la tabla en que se desea hacer cambios.

*Nota: Esto se utiliza cuando necesitamos hacer cambios en la base de datos sin necesidad de haber generado un "login" de por medio.

POST Crear o editar registros con excepción

http://apirest.com/courses?token=no&table=instructors&suffix=instructor&except=name_instructor

HEADERS

Authorization c5LTA6WPbMwHhEabYu77nN9cn4VcMj

PARAMS

token no
table instructors
suffix instructor
except name_instructor

BODY urlencoded

title_course Aprende PHP de cero a avanzado
description_course Aprende todas las bases de este maravillos lenguaje
id_instructor_course 12
image_course https://ludemycdn.com/course/480x270/672600_1def_7.jpg