



API REST CODEIGNITER

Por Lucas Parra

¿QUÉ ES UNA API?

Una interfaz de programación de aplicaciones, abreviada como API (en inglés: Application Programming Interface), es un conjunto, funciones, procedimientos o métodos, que ofrece ciertas características para ser utilizado por otro software para acceder a los servicios que la API defina, los más comunes son accesos a registros en bases de datos.

Son, la forma predilecta que tienen los servicios web para transmitir información.



-Paso 1: "Instalación"



Esquema Básico API

1) Clonar repositorio de Api Rest

Dirigirse a <https://github.com/LucasParra/apiRestMT.git> y clonar el proyecto, cuando se termine la clonación tendrá una carpeta llamada application la cual tendrá todos los archivos necesarios para la instalación del Api.

2) Copiar carpeta application a nuestro proyecto y combinar

Debe hacer click secundario en la carpeta application que se generó con el clon del repositorio apiRestMT de Lucas Parra y combine las carpetas, reemplaze en caso de ser necesario.

-Paso 2: "Configuración"

-Configurar archivo Rest:

Ir a application/config/rest.php y abrir el archivo, una vez abierto hay varias personalizaciones que le puede hacer a la api, por defecto trae un bloqueo de ip restringiendo solo a los servidores de AEDUC, para desactivar esto y hacer pruebas, la variable asignada en la línea 230 debe setearse en false.

Debe estar así:

```
$config['rest_ip_whitelist_enabled'] = FALSE;
```

El archivo de configuración tiene toda la información dentro de si para poder personalizar la api pero viene preconfigurado para que funcione con los softwares de AEDUC®.

-Verificación de archivos:

Deben existir los archivos “models/token_model.php”, “libraries/REST_Controller.php”, “controllers/api/ColegioApi.php”, “config/rest.php”, “language/espanol/rest_controller_lang.php” y “config/routes.php” dentro de la carpeta application.

"El software es como la entropía: difícil de atrapar, no pesa, y cumple la Segunda Ley de la Termodinámica, es decir, tiende a incrementarse" – Norman Augustine

LOGICA DE LA API

Como nuestra api trabaja dentro de CodeIgniter® tiene que cumplir con MVC, para mantener este patron de arquitectura de software tendremos dos modificaciones, las consultas de la base de datos se harán como siempre se han hecho en los clásicos models de CodeIgniter®, mientras que las vistas quedarán a cargo del controlador “REST_Controller” que las manejará con la funcion response() y los controladores tendrán una nueva estructura que se basará en importar el model “model_token” y en heredar de “REST_Controller”.

➤ Controlador

```
require(APPPATH . 'libraries/REST_Controller.php');

class Example_api extends REST_Controller
{

}
```

<https://code.tutsplus.com/es/tutorials/working-with-restful-services-in-codeigniter--net-8814>

Con este ejemplo estamos heredando todas las funciones de REST_Controller y así las tenemos disponibles en la clase Example_api, pudiendo así manejar las respuestas (vistas) con la funcion response

EJEMPLO DE BÚSQUEDA DE DATOS CON CONTROLADOR DE EJEMPLO

Dentro de los archivos descargados existe uno que se llama “ColegioApi” que contiene ejemplos de funciones que se pueden hacer con la api, podemos ver un ejemplo de obtención de colegios en este método.

```
public function getColegios_get()  
  
    {  
  
        $this->token_model->eliminarTokensVencidos();  
  
        $token = $this->token_model->verificarToken($this->get('token'));  
  
        if($token)  
  
        {  
  
            $this->response($this->colegio_model->get_all());  
  
        }  
  
    }
```

Línea de ejecución:

- 1) Se eliminan los tokens vencidos (Llamada obligatoria en TODOS los métodos llamables desde afuera de la api)
- 2) Se verifica la variable que contiene el token que se recibe por get y que se llama “token” (Llamada obligatoria en TODOS los métodos llamables desde afuera de la api)
- 3) Se hace un if que comprueba que la verificación del token fue exitosa
- 4) Si la comprobación hecha en el punto tres es positiva, se responde con los colegios que se obtienen gracias al colegio_model.

IMPORTANTE: REQUERIMIENTOS DE USO (OBLIGATORIOS)!

Para mantener esta api funcional y, lo más importante, segura hay ciertos métodos que SIEMPRE deben estar, estos métodos son:

- getToken_get().
- verificarToken_get().

También debe estar el model “token_model” importado para que los métodos anteriores funcionen y se debe ejecutar el siguiente script:

```
-- Table: token

-- DROP TABLE token;

CREATE TABLE token
(
  id_token serial NOT NULL,
  hora_inicio timestamp without time zone NOT NULL,
  hora_final timestamp without time zone NOT NULL,
  token character varying NOT NULL
)

WITH (
  OIDS=FALSE
);

ALTER TABLE token
  OWNER TO aeduc;
```

Dato Rápido:

- 1) Para nombrar a los métodos de los controladores api se debe explicitar el método http a usar con la forma “nombre_métodoHttp” (Siguiendo el ejemplo anterior se puede ver como el método getColegios_get() está usando el método Http GET, para mas información ver:
<https://developer.mozilla.org/es/docs/Web/HTTP/Methods>