

# Archivo de configuración para generación de test suites

-swagger2postman-

Noviembre 2019

/cloudappi

## Control de Cambios

Versión	Autor	Fecha	Cambios
v1.0	David Marín	07/11/2019	Documento inicial

## Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Motivación.</b>	<b>3</b>
<b>3. Formato.</b>	<b>4</b>
<b>4. Ejemplos.</b>	<b>5</b>
4.1 Ejemplo sencillo	5
4.2 Un entorno real con host y basepath definidos en swagger	5
4.3 Varios entornos sin esquema de seguridad	5
4.4 Varios entornos con colecciones de autorización externas	6
4.5 Entorno productivo que no permite escritura ni borrado de datos:	7

## 1. Introducción

En el presente documento se define un formato de archivo de configuración para generar de forma personalizada las test suites para una API, así como las variables propias del entorno en el que se ejecutarán, a partir de una definición swagger.

Actualmente la herramienta no permite configurar la ejecución, siendo la misma reglada y generando siempre tres colecciones con sus correspondientes entornos. La nomenclatura de nombres es fija, sólo puede alterarse mediante la edición del código fuente.

consideramos que debemos dar la posibilidad de personalizar la ejecución de la misma para que genere las test suites que mejor se adapten a las necesidades del usuario. Para ello se define este formato de archivo de configuración. En el mismo se especificarán los entornos para los que se quieren generar las test suites, el nombre que se le dará a las colecciones, qué tipo de llamadas podrán contener y otras opciones.

## 2. Motivación.

Nos debemos situar en el contexto de una empresa u organización que ha decidido apifizar su negocio dentro de su estrategia de transformación digital. Probablemente habrán optado por una arquitectura de microservicios e implementaran cada API como uno de ellos. Es posible que se enmarquen dentro de un gobierno de APIs. Normalmente las empresas u organizaciones disponen de varios entornos para testear y exponer sus servicios. Lo que no es posible saber a priori es es cómo se llaman dichos entornos, las URLs contra las que probar, los puertos si los hay, etc. En general no se pueden predecir todas las variables que dependen del entorno.

Para poder generar colecciones que se ajusten a las características de cada entorno lo mejor que podemos hacer es darle al usuario la opción de indicarlo directamente.

Por este motivo se define el siguiente formato de archivo de configuración.

## 3. Formato.

El formato de archivo de configuración será JSON, siendo su principal elemento un array de información sobre los distintos entornos. Cada entorno se modela mediante un objeto. Los campos marcados con \* son obligatorios.

- **environments:**
  - tipo: array
  - elementos: tipo environment
- **environment:**
  - tipo: objeto
  - campos:

- **name \***: (string) indica el nombre del entorno para el que se va a generar la test suite. Es meramente informativo, la información que contiene se usará para depuración en consola, indicando mediante logs el entorno para el que se está generando la colección.
- **postman\_connection\_name \***: (string) indica el nombre que se desea dar a la colección postman que implementa la test suite.
- **postman\_environment\_name \***: (string) indica el nombre que se desea dar al entorno postman que contiene las variables que utilizará la test suite.
- **host**: (string) URL o dirección IP del servidor en el que se aloja la API a testear. Si no se indica el campo se
- **port**: (string) puerto en el que escucha el servidor de la API, si lo hay. En caso de que el host sea un DNS no será necesario indicar este campo.
- **basepath**: (string) base path o contexto de la API a probar. Al igual que el puerto en caso de que no haya contexto se indicará una cadena vacía. Si no se indica el campo se tomará el valor especificado en el swagger, que puede ser correcto o no.
- **read\_only**: (boolean) indicará si se deben generar llamadas a endpoints de escritura (POST, PUT, PATCH y DELETE). En entornos productivos no suele ser habitual modificar datos durante las pruebas, salvo que se haya preparado un juego de datos de prueba que sea posible modificar. Si no se indica el campo el valor por defecto será *false*.
- **target\_folder**: (string) indica la ruta local donde se desea que se almacenen la colección y el entorno generados para el entorno concreto. Si no se indica se depositarán en el directorio desde el que se ha ejecutado la herramienta.
- **custom\_authorizations\_file**: (string) indica la ruta donde se encuentra la colección postman con las llamadas a realizar para obtener los tokens de autorización necesarios. Si no se dispone de la misma no se indicará este campo.

## 4. Ejemplos.

### 4.1 Ejemplo sencillo

El caso más sencillo es que sólo queramos generar una test suite para un entorno de desarrollo local en el que no haya esquema de seguridad, se deseen generar todas las peticiones para todos los tipos de operación y queramos que se deposite el resultado en el directorio actual:

test\_suite\_config.json:

```
{  
  environments:[  
    {
```

```
        "name" : "DEV",
        "postman_collection_name" : "test_suite_dev",
        "postman_environment_name" : "dev_environment",
        "host" : "https://localhost",
        "port" : "8080",
        "basepath" : "/some/api/v1",
    }
]
```

## 4.2 Un entorno real con host y basepath definidos en swagger

Si tenemos correctamente definidos el `host` y el `basepath` en el documento swagger no necesitamos especificarlos. El archivo es aún más sencillo que el anterior:

test\_suite\_config.json:

```
{
  environments:[
    {
      "name" : "PRE",
      "postman_collection_name" : "test_suite_pre",
      "postman_environment_name" : "pre_environment",
    }
  ]
}
```

## 4.3 Varios entornos sin esquema de seguridad

En este caso generaremos las colecciones de pruebas para dos entornos, nombrándolas de forma distinta y guardándolas en distintas ubicaciones. No se aporta el puerto dado que el `host` de ambos entornos es un nombre de dominio:

test\_suite\_config.json:

```
{
  environments:[
    {
      "name" : "PRE",
      "postman_collection_name" : "My_API_test_suite_PRE",
      "postman_environment_name" : "My_API_PRE_environment",
      "host" : "https://some.server.pre.org",
      "basepath" : "/my/api/v1",
      "target_folder" : "../apis/pre/myAPI/test_suites"
    }
    {
      "name" : "STAGE",
      "postman_collection_name" : "My_API_test_suite_STAGE",

```

```
        "postman_environment_name" :  
        "My_API_STAGE_environment",  
        "host" : "https://some.server.stage.org",  
        "basepath" : "/apis/my/api/v1",  
        "target_folder" : "../apis/stage/myAPI/test_suites"  
    }  
]  
}
```

## 4.4 Varios entornos con colecciones de autorización externas

Tenemos el mismo caso que antes, pero en esta ocasión tenemos predefinidas las colecciones con las llamadas necesarias para obtener las autorizaciones requeridas por el esquema de seguridad de la API:

test\_suite\_config.json:

```
{  
  environments:[  
    {  
      "name" : "PRE",  
      "postman_collection_name" : "My_API_test_suite_PRE",  
      "postman_environment_name" : "My_API_PRE_environment",  
      "host" : "https://some.server.pre.org",  
      "basepath" : "/my/api/v1",  
      "target_folder" : "../apis/pre/myAPI/test_suites",  
      "custom_authorizations_file" : "../apis/pre/myAPI/auth"  
    }  
    {  
      "name" : "STAGE",  
      "postman_collection_name" : "My_API_test_suite_STAGE",  
      "postman_environment_name" :  
      "My_API_STAGE_environment",  
      "host" : "https://some.server.stage.org",  
      "basepath" : "/apis/my/api/v1",  
      "target_folder" : "../apis/stage/myAPI/test_suites",  
      "custom_authorizations_file" :  
      "../apis/stage/myAPI/auth"  
    }  
  ]  
}
```

4.5 Entorno productivo que no permite escritura ni borrado de datos:

En este caso se debe usar el campo `read_only` con valor `true` para evitar que se generen llamadas que puedan modificar o eliminar datos en el mismo.

test\_suite\_config.json:

```
{
  environments:[
    {
      "name" : "PRE",
      "postman_collection_name" : "My_API_test_suite_PROD",
      "postman_environment_name" : "My_API_PROD_environment",
      "host" : "https://some.server.org",
      "read_only" :true,
      "basepath" : "/my/api/v1",
      "target_folder" : "../apis/prod/myAPI/test_suites",
      "custom_authorizations_file" :
"../apis/prod/myAPI/auth"
    }
  ]
}
```