

# Importación del proyecto Postman

La aplicación **Postman** es una simple pero potente aplicación para poder consumir APIs o recursos a través de la red (para nuestro caso, consumiremos nuestro API REST).

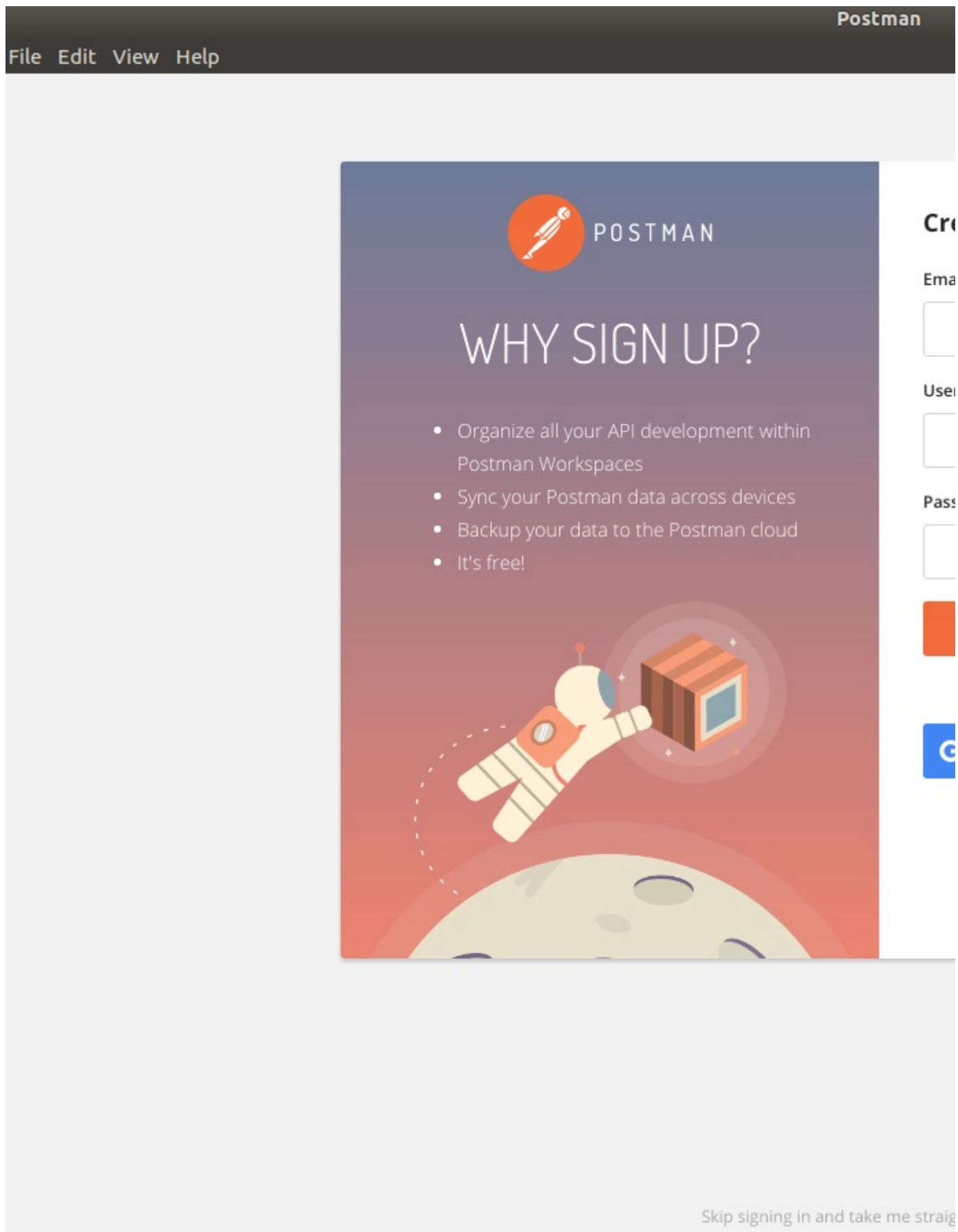
Esta aplicación la podemos emplear como plugin de Google Chrome (actualmente deprecado) o como una aplicación nativa. Vamos a instalar la aplicación nativa, si ya la tuvieses instalada puedes saltar a la siguiente sección.

## Instalación de Postman

Accedemos al siguiente enlace

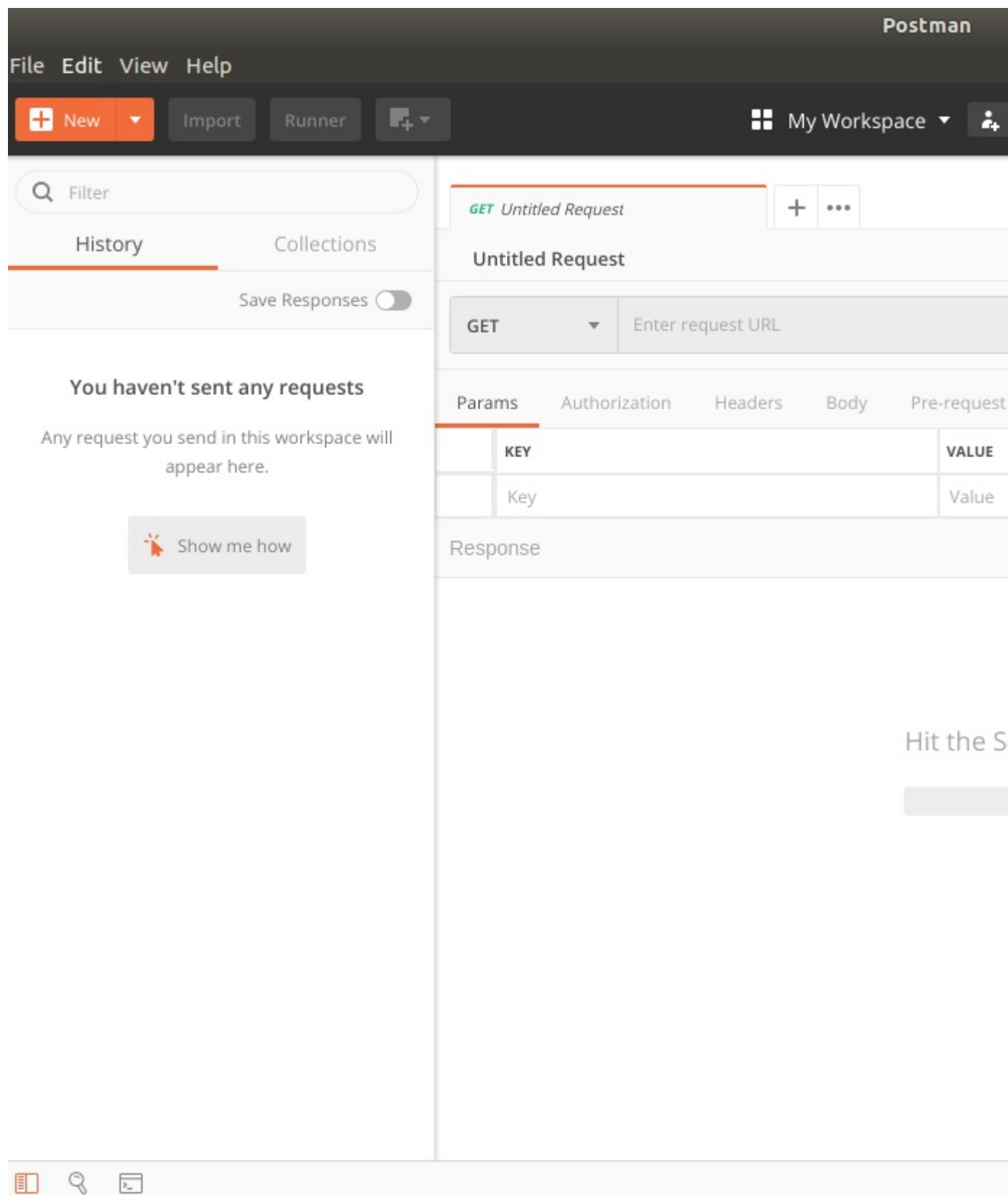
<https://www.getpostman.com/downloads/>

Y tras descargar y abrir la app nos pedirá una cuenta de usuario, podemos crear una nueva o emplear nuestro usuario de Google:



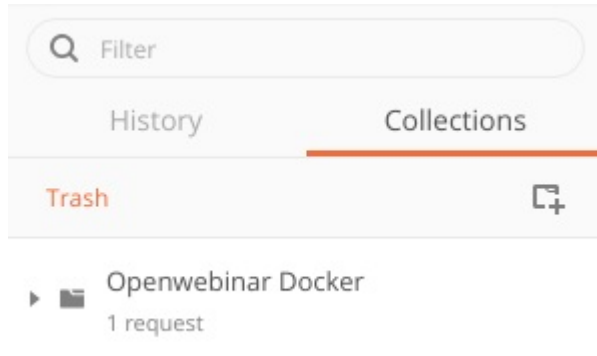
El login lo hacen especialmente para que puedas guardar tus proyectos y poder

acceder a ellos desde cualquier sitio (*cuida con la seguridad...*). Tras registrarnos veremos una vista como la que sigue a continuación:

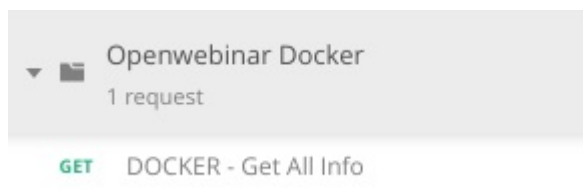


**Importar el proyecto de demo**

En la nueva vista que se nos ha abierto deberemos hacer clic sobre el botón situado en la esquina superior izquierda *Import* y sobre la ventana emergente que aparece arrastrar y soltar (o ir al path determinado) el archivo `Openwebinar Docker.postman_collection`. Automáticamente quedará el proyecto importado y deberemos hacer clic sobre la opción de *Collections* para verlo:



Si hacemos clic sobre *Openwebinar Docker* se nos desplegarán las dos opciones que tenemos de demo para poder consumir la API:



## Lanzar la petición

A continuación deberemos hacer clic sobre *Get All info*, se nos abra la petición que nos permite recuperar todos las entradas que hay en la BBDD, pulsemos sobre *Headers* es donde estarán las cabeceras necesarias para nuestra API:

History

Collections

Trash

Openwebinar Docker

1 request

GET DOCKER - Get All Info

GET DOCKER - Get All Info

GET

http://localhost:7001/app-rest-api/avengers/g

Params

Authorization

Headers (4)

Body

Pre-req

	KEY	VALUE
<input checked="" type="checkbox"/>	Accept	applica
<input checked="" type="checkbox"/>	Accept-Encoding	gzip, de
<input checked="" type="checkbox"/>	Accept-Language	en-US,e
<input checked="" type="checkbox"/>	X-Requested-With	XMLHtt
	Key	Value

Acto seguido pulsaremos el botón *Send* para ejecutar la petición y deberemos obtener:

```
Body Cookies Headers (3) Test Results

Pretty Raw Preview JSON

1 [
2   {
3     "id": 0,
4     "lang": "EN",
5     "name": "Iron Man",
6     "actor": "Robert Downey Jr.",
7     "description": "Iron Man (Anthony Edward \"Tony\" Stark) is a fictional superhero appearing in American comic books published by Marvel Comics, designed by artists Don Heck and Jack Kirby...",
8     "urlimage": "https://d29fhpw069ctt2.cloudfront.net/icon/image/59598/preview.svg"
9   },
10  {
11    "id": 1,
12    "lang": "EN",
13    "name": "Spider-Man",
14    "actor": "Tom Holland",
15    "description": "Spider-Man is a fictional superhero created by writer-editor Stan Lee and writer-artist Steve Ditko, appearing in American comic books published by Marvel Comics.",
16    "urlimage": "https://d29fhpw069ctt2.cloudfront.net/icon/image/59595/preview.svg"
17  },
18  {
19    "id": 2,
20    "lang": "EN",
21    "name": "American Captain",
22    "actor": "Chris Evans",
23    "description": "Captain America (Steve Rogers) is a fictional superhero appearing in American comic books published by Marvel Comics, the character was created by Joe Simon and Jack Kirby (cover dated March 1941) from Timely Comics, a predecessor of Marvel Comics...",
24    "urlimage": "https://d29fhpw069ctt2.cloudfront.net/icon/image/59598/preview.svg"
25  },
26  {
27    "id": 3,
28    "lang": "EN",
29    "name": "Black Widow",
30    "actor": "Scarlett Johansson",
31    "description": "Natalia Alianovna Romanova (alias: Natasha Romanoff), colloquial: Black Widow is a fictional character appearing in American comic books published by Marvel Comics.",
32    "urlimage": "https://d29fhpw069ctt2.cloudfront.net/icon/image/59601/preview.svg"
33  },
34  {
35    "id": 4,
36    "lang": "EN",
37    "name": "Thor",
38    "actor": "Chris Hemsworth",
```

## Resultado final

Si la petición ha ido correctamente la aplicación desplegada en el contenedor de WL que consume la BBDD del contenedor de Oracle DB habrá quedado correctamente configurada. Sino, habrá que revisar como están desplegados los contenedores y como se ha desplegado la aplicación dentro de WL.

Habiendo finalizado esto ya tendríamos un entorno funcional Oracle basado en contenedores. En los siguientes temas vamos a empezar con las primeras prácticas.