

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:



POSTMAN

WHY SIGN UP?

- Organize all your API development within Postman Workspaces
- Sync your Postman data across devices
- Backup your data to the Postman cloud
- It's free!

[Skip signing in and take me straight to Postman](#)

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:

The screenshot shows the Postman application interface. At the top, there is a dark header bar with the word "Postman" on the right. Below it is a navigation bar with "File", "Edit", "View", and "Help" menus, followed by "New", "Import", "Runner", and a plus icon. To the right of the runner button is "My Workspace" with a dropdown arrow, and a user icon.

The main area has a left sidebar with a "History" tab (which is selected, indicated by an orange underline) and a "Collections" tab. A "Save Responses" toggle switch is also present. Below the sidebar, a message says "You haven't sent any requests" and "Any request you send in this workspace will appear here." There is a "Show me how" button with a cursor icon pointing to it.

The main workspace is titled "Untitled Request". It shows a "GET Untitled Request" entry. The "Params" tab is selected, showing a table with one row: "Key" under "KEY" and "Value" under "VALUE". Below the table, the word "Response" is visible.

At the bottom of the screen, there are three small icons: a document, a magnifying glass, and a folder. On the right side of the main workspace, the text "Hit the S" is partially visible.

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:

The screenshot shows the Postman application interface. At the top, there is a search bar labeled 'Filter'. Below it, there are tabs for 'History' and 'Collections', with 'Collections' being the active tab, indicated by a red underline. Underneath these tabs is a 'Trash' section. On the right side of the header, there is a '+' icon for creating new collections. The main content area displays a single collection named 'Openwebinar Docker' with a small folder icon next to it, indicating it contains requests. Below the collection name, it says '1 request'.

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

This screenshot shows the same 'Openwebinar Docker' collection from the previous image, but it is now expanded. The folder icon is followed by a downward arrow, indicating that requests are visible. Below the collection name, it says '1 request'. Underneath the collection, a single request is listed: 'GET DOCKER - Get All Info'.

Lanzar la petición

A continuación deberemos hacer clic sobre *Get All info*, se nos abrirá 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:

The screenshot shows the Postman application interface. On the left, there's a sidebar with a search bar labeled 'Filter', a 'History' tab, a 'Collections' tab which is currently selected and highlighted in orange, a 'Trash' tab, and a plus sign icon. Below the tabs, there's a section for 'Openwebinar Docker' containing one request. On the right, the main area displays a specific request: 'GET DOCKER - Get All Info'. This request has its method set to 'GET' and its URL to 'http://localhost:7001/app-rest-api/avengers/get'. Below the URL, there are tabs for 'Params', 'Authorization', 'Headers (4)', 'Body', and 'Pre-req'. The 'Headers (4)' tab is active and highlighted in orange. A table below shows four headers with their values:

	KEY	VALUE
<input checked="" type="checkbox"/>	Accept	application/json
<input checked="" type="checkbox"/>	Accept-Encoding	gzip, deflate
<input checked="" type="checkbox"/>	Accept-Language	en-US,en;q=0.9
<input checked="" type="checkbox"/>	X-Requested-With	XMLHttpRequest
	Key	Value

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

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. He was created by Stan Lee, Jack Kirby, and Bob Layton. Iron Man's alter ego is Tony Stark, a genius-level industrialist, engineer, inventor, and weapons designer who built the Iron Man suit to protect the world from threats like the terrorist organization HYDRA. The suit has been updated multiple times, with the most recent version featuring arc reactors and repulsor technology.", "urlimage": "https://d29fhpw069ctt2.cloudfront.net/icon/image/59598/preview.svg"
8   },
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. He first appeared in the comic book Amazing Fantasy in 1962. Spider-Man's powers come from a bite he received from a radioactive spider. He uses his agility, strength, and web-slinging abilities to fight crime and protect the innocent.", "urlimage": "https://d29fhpw069ctt2.cloudfront.net/icon/image/59595/preview.svg"
16    },
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. He was created by Stan Lee and Jack Kirby. Captain America's powers come from a shield made of vibranium and a suit of advanced armor. He is known for his leadership, courage, and unwavering sense of justice.", "urlimage": "https://d29fhpw069ctt2.cloudfront.net/icon/image/59598/preview.svg"
25      },
26      {
27        {
28          "id": 3,
29          "lang": "EN",
30          "name": "Black Widow",
31          "actor": "Scarlett Johansson",
32          "description": "Natalia Alianovna Romanova (alias: Natasha Romanoff), colloquial: Black Widow is a fictional superhero appearing in comic books published by Marvel Comics. She was created by Stan Lee and Roy Thomas. She is a master assassin and expert marksman, using her skills to protect the world from various threats.", "urlimage": "https://d29fhpw069ctt2.cloudfront.net/icon/image/59601/preview.svg"
33      },
34      {
35        {
36          "id": 4,
37          "lang": "EN",
38          "name": "Thor",
39          "actor": "Chris Hemsworth",
40          "description": "Thor Odinson is a fictional superhero appearing in comic books published by Marvel Comics. He was created by Stan Lee and Jack Kirby. Thor is the God of Thunder and uses his hammer Mjolnir to fight evil. He is known for his strength, courage, and sense of honor.", "urlimage": "https://d29fhpw069ctt2.cloudfront.net/icon/image/59604/preview.svg"
42      }
43    }
44  ]

```

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.