

# MOCKAPI.IO

<b>OBJETIVOS</b>	<b>2</b>
<b>Mockapi.io: descripción</b>	<b>2</b>
Acceso y Registro	3
Crear un proyecto nuevo	4
<b>Creación de un endpoint</b>	<b>5</b>
Probar nuestro API Endpoint	9
Interactuar con nuestro API endpoint	9
<b>Métodos HTTP</b>	<b>10</b>

# OBJETIVOS

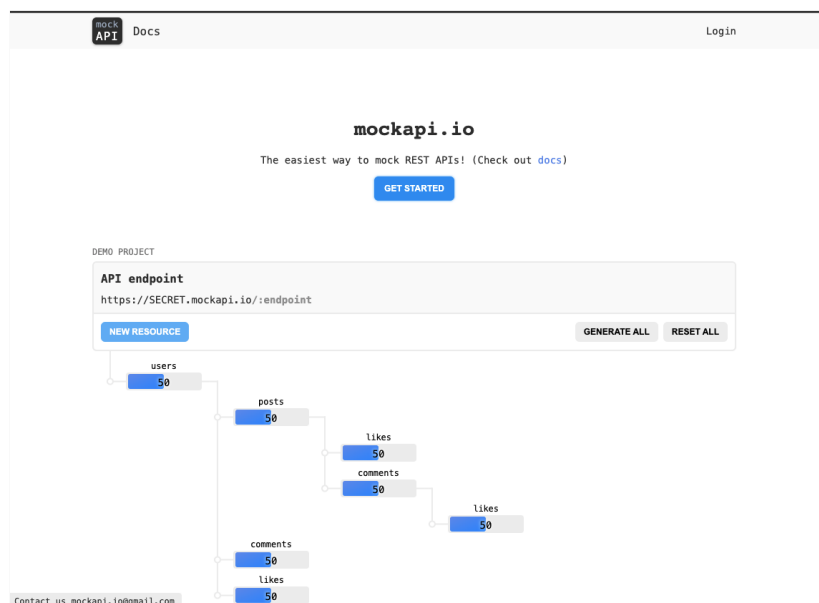
A través de este tutorial, configuraremos un **API ENDPOINT** remoto para realizar pruebas lo más cercanas a la realidad en cuanto a interacción contra un backend, refiere.

Recomendamos seguir este tutorial para poder llevar a cabo de forma efectiva, la creación y configuración del backend que nos proveerá de datos para la práctica.

## Mockapi.io: descripción

Mockapi es una [aplicación del tipo SaaS](#) que nos permite configurar una [API RestFul](#) y definir endpoints para que una aplicación pueda acceder y trabajar con datos remotos.

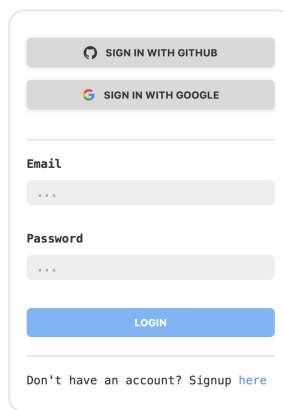
Utilizaremos su capa de uso gratuito para diseñar el backend de nuestra App y, así, poder consumir y guardar datos remotos utilizando las diferentes formas de interacción que provee JavaScript. 💪



## Acceso y Registro

- 1) Ingresa a su URL: <https://mockapi.io/>
- 2) A continuación, pulsa el botón Login para registrarte. Puedes realizar el registro utilizando tu usuario **Github** o tu **Cuenta de Google**.

Sign in



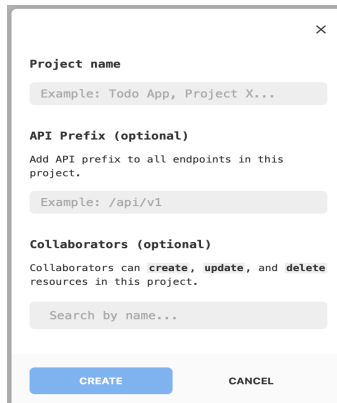
The image shows a mobile-style sign-in interface. At the top, it says "Sign in". Below this are two buttons: "SIGN IN WITH GITHUB" and "SIGN IN WITH GOOGLE". Underneath these are two input fields labeled "Email" and "Password", each with a small eye icon to toggle visibility. Below the password field is a blue "LOGIN" button. At the bottom, there is a link that says "Don't have an account? Signup [here](#)".

- 3) También puedes hacerlo de la forma convencional (*ingresando tus datos, un email personal y una contraseña a elección*).

## Crear un proyecto nuevo

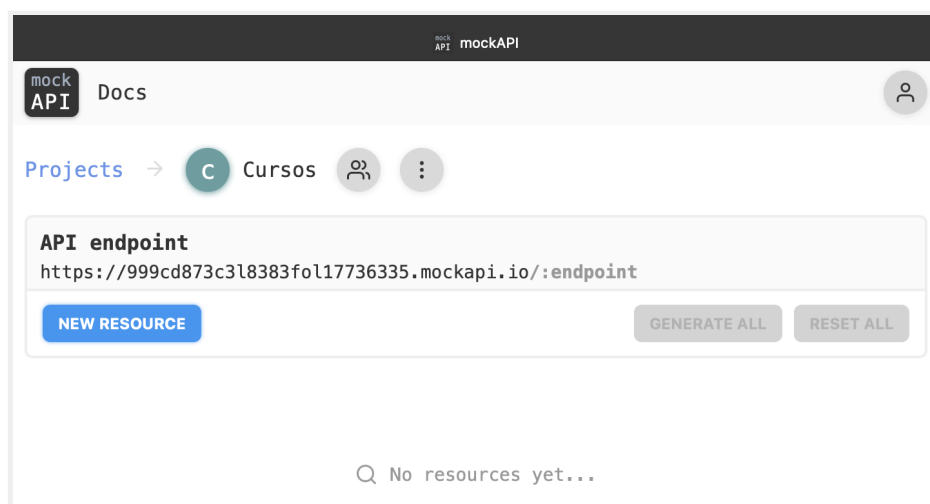
Pulsa el botón ( + ) contiguo al apartado **Projects**, para crear un nuevo proyecto.

Se abrirá a continuación, una ventana emergente, donde deberás completar los datos solicitados. En principio, ingresa **Cursos** como nombre del proyecto, y pulsa el botón **Create**.



A modal window for creating a new project in MockAPI. It contains three sections: 'Project name' with an example 'Todo App, Project X...', 'API Prefix (optional)' with an example '/api/v1' and a note to add the prefix to all endpoints, and 'Collaborators (optional)' with a note that collaborators can create, update, and delete resources. There is a search bar labeled 'Search by name...' and two buttons at the bottom: 'CREATE' and 'CANCEL'.

Obtendrás como resultado, una **URL base** como la que vemos en la siguiente imagen:



Solo resta crear un **endpoint**, para luego consumir su información.

## Creación de un endpoint

Mockapi permite definir qué tipos de datos tendrá nuestro endpoint. Puedes crear cada campo eligiendo su tipo de datos, o generando los mismos a través de una librería como [faker.js](https://fakerjs.dev/).

En esta ocasión, aceleremos el proceso, creando la misma estructura que ves en la siguiente imagen.



The image shows a configuration window for generating mock data. It has two main sections: 'Schema (optional)' and 'Object template (optional)'. The 'Schema' section lists fields: 'id' (Object ID), 'nombre' (String), 'creado' (Date), and 'duracion' (Number), with a plus button to add more. The 'Object template' section explains that a JSON template can be used, referencing Faker.js methods. It shows an example JSON object with fields like 'username', 'knownIps', 'profile' (containing 'firstName' and 'lastName'), and 'staticData' (an array of numbers). Below the example is a text area labeled 'JSON template'.

**Schema (optional)**  
Define Resource schema, it will be used to generate mock data.

id Object ID

nombre String

creado Date

duracion Number

+

**Object template (optional)**  
To define more complex structure for your data use JSON template. You can reference Faker.js methods using `\$.`.

**EXAMPLE**

```
{
  "username": "$internet.userName",
  "knownIps": ["$internet.ip", "$internet.ipv6"],
  "profile": {
    "firstName": "$name.firstName",
    "lastName": "$name.lastName",
    "staticData": [100, 200, 300]
  }
}
```

JSON template

Se generarán automáticamente del lado del backend, los diferentes endpoint, que te permitirán:

- **listar** todos los registros
- **obtener** un registro específico
- **crear** nuevos registros
- **modificar** los existentes
- **eliminar** alguno de los almacenados

Todo, utilizando los métodos GET, POST, PUT y DELETE.

ON GET → /cursos  
\$mockData

ON GET → /cursos/:id  
\$mockData

ON POST → /cursos  
\$mockData

ON PUT → /cursos/:id  
\$mockData

ON DELETE → /cursos/:id  
\$mockData

CREATE CANCEL

Una vez creado el recurso, será visualizado en la pantalla principal de tu perfil.

Projects → C Cursos

**API endpoint**  
https://615ceedec29813001773636d.mockapi.io/:endpoint

NEW RESOURCE

cursos	
5	Data Edit Delete

Posiciona el mouse sobre éste, y pulsa el botón **Data**.

En la ventana emergente, pega el JSON que encontrarás, a continuación:

```
[
  {
    "id": 1,
    "nombre": "JavaScript",
    "creado": "2014-05-15",
    "duracion": 16
  },
  {
    "id": 2,
    "nombre": "React",
    "creado": "2017-08-30",
    "duracion": 18
  },
  {
    "id": 3,
    "nombre": "Angular",
    "creado": "2021-09-01",
    "duracion": 18
  },
  {
    "id": 4,
    "nombre": "Vue",
    "creado": "2021-09-01",
    "duracion": 16
  },
  {
    "id": 5,
    "nombre": "Node JS",
    "creado": "2020-07-14",
    "duracion": 24
  }
]
```

La estructura de los datos debe quedar de manera similar al JSON que vemos en la imagen contigua.

×

Edit/replace data for **cursos** resource. Data must be an array and a valid JSON.

```
[
  {
    "id": 1,
    "nombre": "JavaScript",
    "creado": "2014-05-15",
    "duracion": 16
  },
  {
    "id": 2,
    "nombre": "React",
    "creado": "2017-08-30",
    "duracion": 18
  },
  {
    "id": 3,
    "nombre": "Angular",
    "creado": "2021-09-01",
    "duracion": 18
  },
  {
    "id": 4,
    "nombre": "Vue",
    "creado": "2021-09-01",
    "duracion": 16
  },
  {
    "id": 5,
    "nombre": "Node JS",
    "creado": "2020-07-14",
    "duracion": 24
  }
]
```

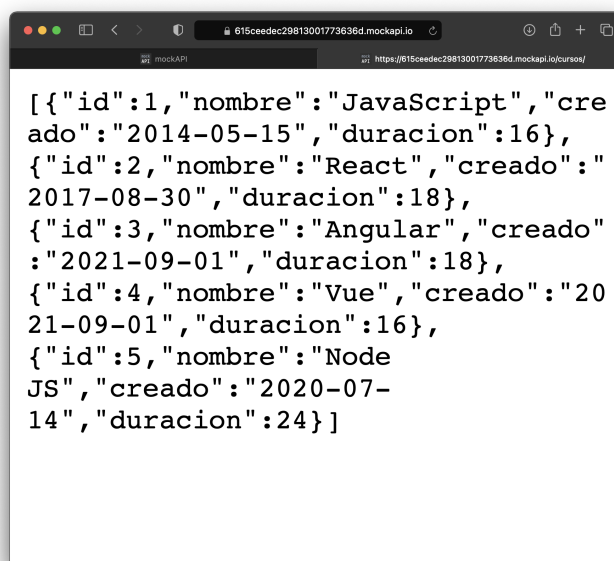
UPDATE CANCEL

Solo resta pulsar el botón **Update**, para que los mismos se generen, y ya puedas trabajar con ellos.



## Probar nuestro API Endpoint

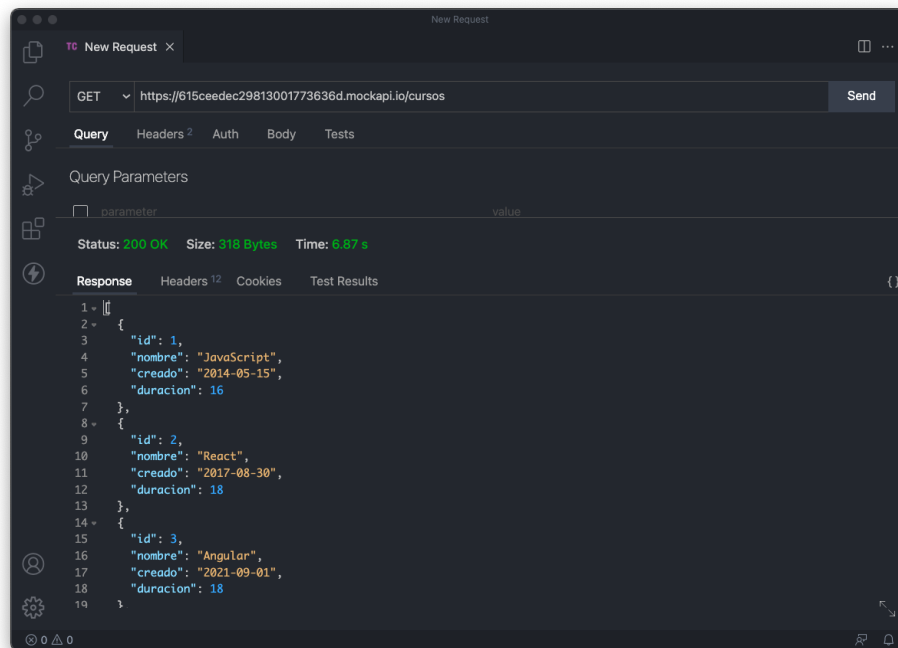
- Copia la URL generada en nuestro perfil, agregándole /cursos al final de la misma
- Pégala en una nueva ventana del navegador web
- Deberás obtener como resultado un JSON similar al de la imagen



Si es así, ya sabemos que nuestro backend... ¡funciona! 🤖

## Interactuar con nuestro API endpoint

Si eres usuario de [Postman](#) o App similar, puedes usarla para probar los endpoint. Si no eres usuario, te recomendamos simplificar su uso, instalando la extensión de VS Code: [Thunder Client](#). Se integra al IDE y se opera igual que Postman.



## Métodos HTTP

Aprovechamos la temática de hoy para refrescar los diferentes métodos, o verbos, HTTP:

Método	Descripción
<b>GET</b>	Solicita un recurso específico. Como respuesta, puede volver un conjunto de datos o un único dato.
<b>POST</b>	Se utiliza para enviar un recurso específico, modificando un estado o agregando datos en el servidor.
<b>PUT</b>	Modifica un recurso remoto específico.
<b>DELETE</b>	Elimina un recurso remoto.

Existen otros, aunque los mencionados son los más utilizados para operar los endpoint generados mediante **mockapi.io**.