

Practica

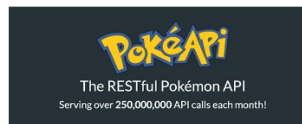
El siguiente menú sera la pantalla que veremos al ingresar a la url de la página que se desarrolló como práctica, tendremos 2 imágenes con la descripción de lo que podrás encontrar al dar clic sobre ellas, en la parte superior izquierda tenemos el escrito “ Práctica-Hermenegildo”, al dar clic sobre este mensaje regresaremos a esta misma pantalla que será el menú principal, en la parte superior derecha tenemos un enlace que nos redireccionará hacia la pagina oficial de PokeAPI, donde podremos encontrar la documentación oficial de esta API.

Práctica - Hermenegildo

[PokeAPI - mas información](#)

Práctica

REST API y Contenedores



Observa un posible uso del consumo de PokeApi, una RESTful API con toda la informacion de los pokemon.



API REST creada
Con informacion de jugadores de futbol.



Al dar clic en la imagen de PokeAPI podremos ver en nuestra pantalla el siguiente formulario, en el encontraremos una imagen haciendo referencia a Pokemon que es precisamente el API que se seleccionó para usar internamente en nuestra práctica.

El único paso que debemos hacer es escribir el nombre del Pokemon que queramos buscar e inmediatamente debemos presionar en botón buscar, unos instantes después veremos la información del Pokemon Seleccionado, en ella encontraremos datos como el número que ocupa, su nombre, su habilidad principal y la clase a la que pertenece, al finalizar veremos una imagen de nuestro pokemon.



Escribe el nombre del pokemon...

Buscar

Numero: 25
Nombre: Pikachu
Habilidad principal: Static
Clase: electric



Si presionamos la segunda imagen con el mensaje RESTAPI podremos ver en nuestra pantalla una serie de datos en formato json el cual es parte de la RESTAPI que hicimos como parte de la prueba, estos datos están relacionados con futbolistas y datos sobre ellos como nombre, edad y país de origen.

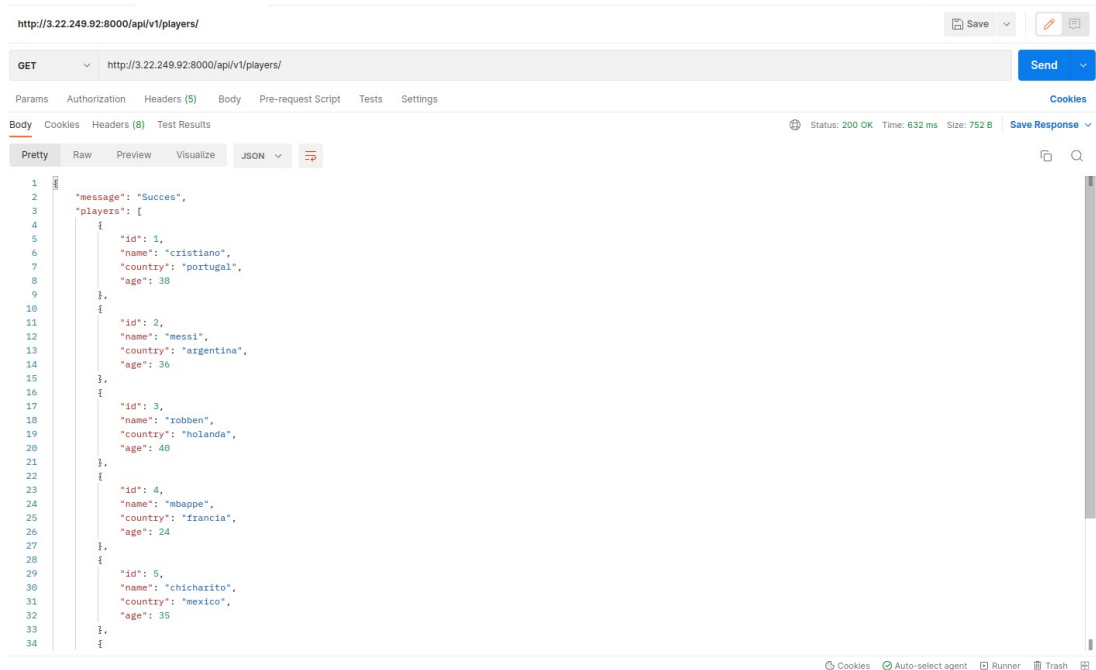
```
1 // 20230205214615
2 // http://3.22.249.92:8000/api/v1/players/
3
4 {
5   "message": "Succes",
6   "players": [
7     {
8       "id": 1,
9       "name": "cristiano",
10      "country": "portugal",
11      "age": 38
12    },
13    {
14      "id": 2,
15      "name": "messi",
16      "country": "argentina",
17      "age": 36
18    },
19    {
20      "id": 3,
21      "name": "robben",
22      "country": "holanda",
23      "age": 40
24    },
25    {
26      "id": 4,
27      "name": "mbappe",
28      "country": "francia",
29      "age": 24
30    },
31    {
32      "id": 5,
33      "name": "chicharito",
34      "country": "mexico",
35      "age": 35
36    },
37  ]
38 }
```

Este RESTAPI fue creada con instrucciones para los metodos GET, PUT, POST, DELETE, para la demostración de su uso y correcto funcionamiento haremos uso de POSTMAN el cual nos ayudara mostrando la salida que deberiamos tener al consultar esta API.

GET

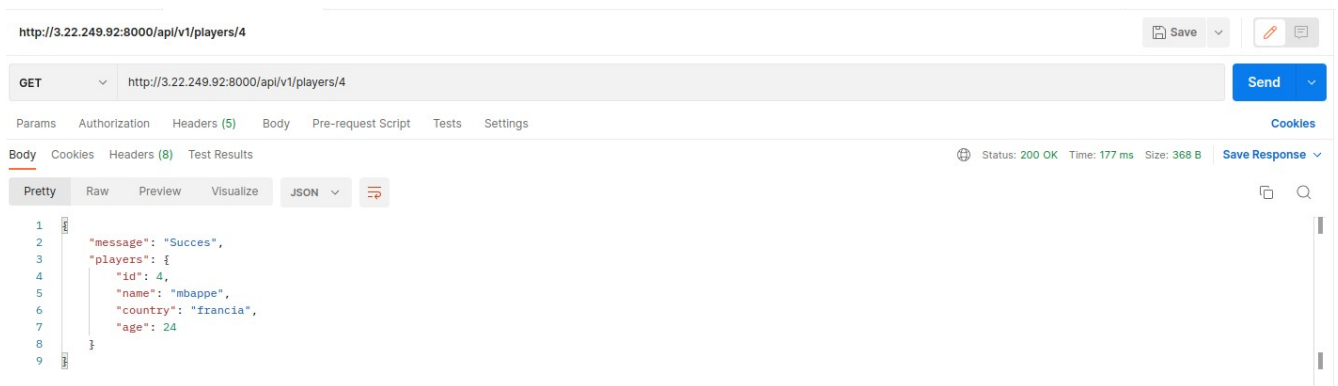
<http://3.22.249.92:8000/api/v1/players/>

La dirección antes mostrada sera nuestro enlace para obtener los datos guardados en nuestra base de datos, con la instrucción “players” al final de la url hacemos uso del metodo GET, el cual nos mostrará absolutamente todos los datos, de la misma manera obtendremos el mensaje “Succes” al tener acceso satisfactorio a la url.



<http://3.22.249.92:8000/api/v1/players/{ID}>

Usando el mismo metodo y agragando al final de la url un numero que correspondera al id de nuestro jugador se nos mostrará unicamente la información correspondiente de ese jugador.



En caso de escribir un ID no existente obtendremos el mensaje
`"message": "No players found."`

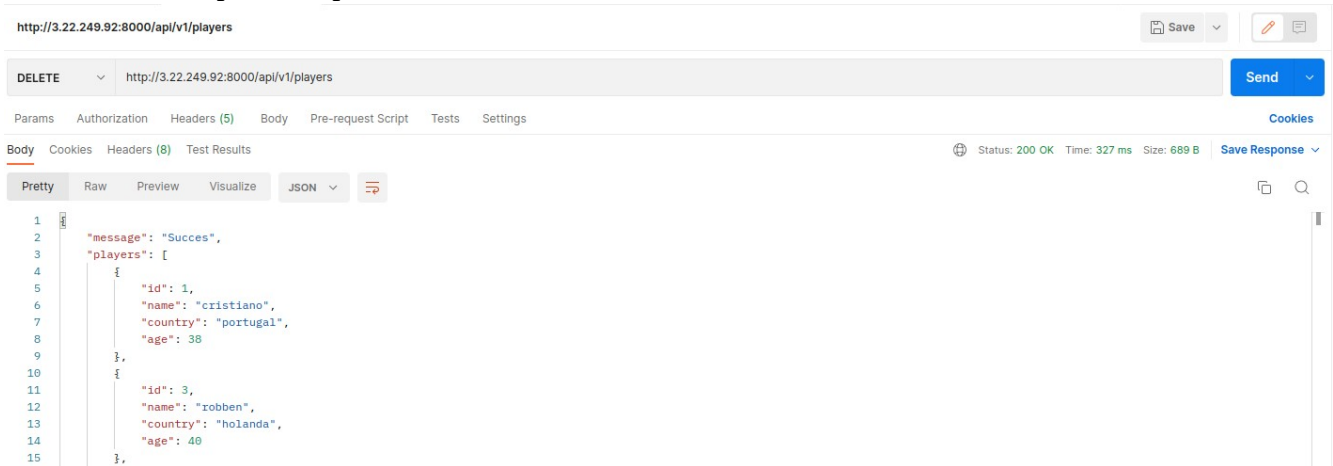
DELETE

<http://3.22.249.92:8000/api/v1/players/{ID}>

Al ingresar en la url un ID pero usando el método DELETE borraremos toda a información correspondiente a ese jugador y obtendremos el siguiente mensaje:

`"message": "Deleted"`

En el ejemplo borramos el jugador correspondiente al ID 2, los datos en cuestión son sobre messi, y al volver a consultar los datos de todos los jugadores podemos notar que los datos sobre messi ya no aparecen, por lo tanto nuestro método DELETE funciona correctamente.



PUT

<http://3.22.249.92:8000/api/v1/players/{ID}>

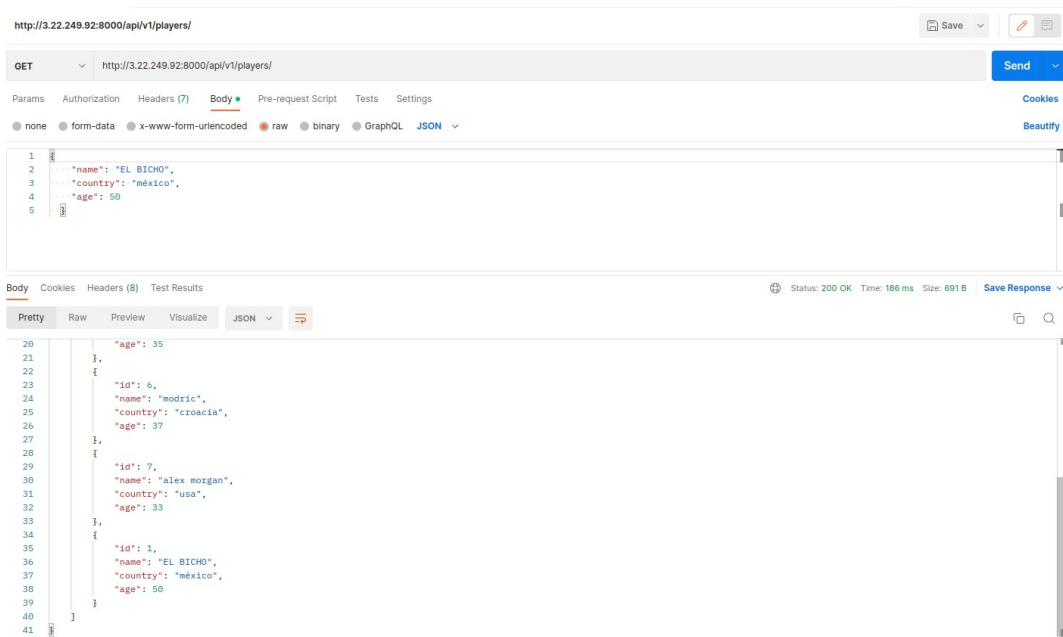
Usaremos el metodo PUT para actualizar la informacion de algun jugador, necesitaremos el id del jugador en cuestión y necesitaremos en el body usar el siguiente formato:

```
{
  "name": dato tipo string
  "country": dato tipo string
  "age": dato tipo num
}
```

Al mandar los datos correctamente recibiremos el siguiente mensaje:

```
{
  "message": "Updated."
}
```

Al acceder nuevamente a los datos de los jugadores podremos observar que el jugador que tenia la información correspondiente a “cristiano”, “portugal”, “38”, se ha actualizado correctamente, por lo tanto sabemos que nuestro método funciona correctamente.



POST

<http://3.22.249.92:8000/api/v1/players/>

Usaremos el m\u00e9todo POST para agregar nuevos datos, usaremos el siguiente formato en el apartado body, el dato id es auto incremental por lo tanto no lo agregamos:

```

{
  "name": dato tipo string
  "country": dato tipo string
  "age": dato tipo num
}

```

Una vez realizada la consulta obtendremos el siguiente mensaje:

```

{
  "message": "Player saved correctly."
}

```

Al consultar nuevamente la informaci\u00f3n de todos los jugadores podemos ver los datos que reci\u00e9n agregamos:

