

Ejercicio clase 9 GTICS - 20191450

Consumo API REST para listar personajes

URL	Method
http://localhost:8080/ws/personaje/list	GET

1. Argumentos de entrada:

Ninguno

2. Argumentos de salida:

```
[
  {
    "id": 1026,
    "name": "Absalom (Earth-616)",
    "url": "/Absalom_(Earth-616)",
    "identity": "Secret Identity",
    "align": null,
    "eye": "Blue Eyes",
    "hair": "Blond Hair",
    "sex": "Male Characters",
    "gsm": null,
    "alive": "Deceased Characters",
    "appearances": 6,
    "first_appearance": "May-92",
    "year": 1992
  },
  {
    "id": 1027,
    "name": "Carl Creel (Earth-616)",
    "url": "/Carl_Creel_(Earth-616)",
    "identity": "Public Identity",
    "align": "Bad Characters",
    "eye": "Blue Eyes",
    "hair": "Bald",
    "sex": "Male Characters",
    "gsm": null,
    "alive": "Living Characters",
    "appearances": 202,
  }
]
```

```

    "first_appearance": "Apr-64",
    "year": 1964
  },
  {
    "id": 1029,
    "name": "Adam Warlock (Earth-616)",
    "url": "/Adam_Warlock_(Earth-616)",
    "identity": "No Dual Identity",
    "align": "Good Characters",
    "eye": "White Eyes",
    "hair": "Gold Hair",
    "sex": "Male Characters",
    "gsm": null,
    "alive": "Living Characters",
    "appearances": 252,
    "first_appearance": "Sep-67",
    "year": 1967
  },
  {
    "id": 1030,
    "name": "Adam Neramani (Earth-616)",
    "url": "/Adam_Neramani_(Earth-616)",
    "identity": "Secret Identity",
    "align": "Good Characters",
    "eye": "Blue Eyes",
    "hair": "Blond Hair",
    "sex": "Male Characters",
    "gsm": null,
    "alive": "Living Characters",
    "appearances": 21,
    "first_appearance": "Oct-93",
    "year": 1993
  },
  {
    "id": 1032,
    "name": "Eric Cameron (Earth-616)",
    "url": "/Eric_Cameron_(Earth-616)",
    "identity": "Secret Identity",
    "align": "Bad Characters",
    "eye": "Blue Eyes",
    "hair": "Brown Hair",
    "sex": "Male Characters",
    "gsm": null,
    "alive": "Living Characters",
    "appearances": 5,
    "first_appearance": "Mar-80",
    "year": 1980
  },
  ...

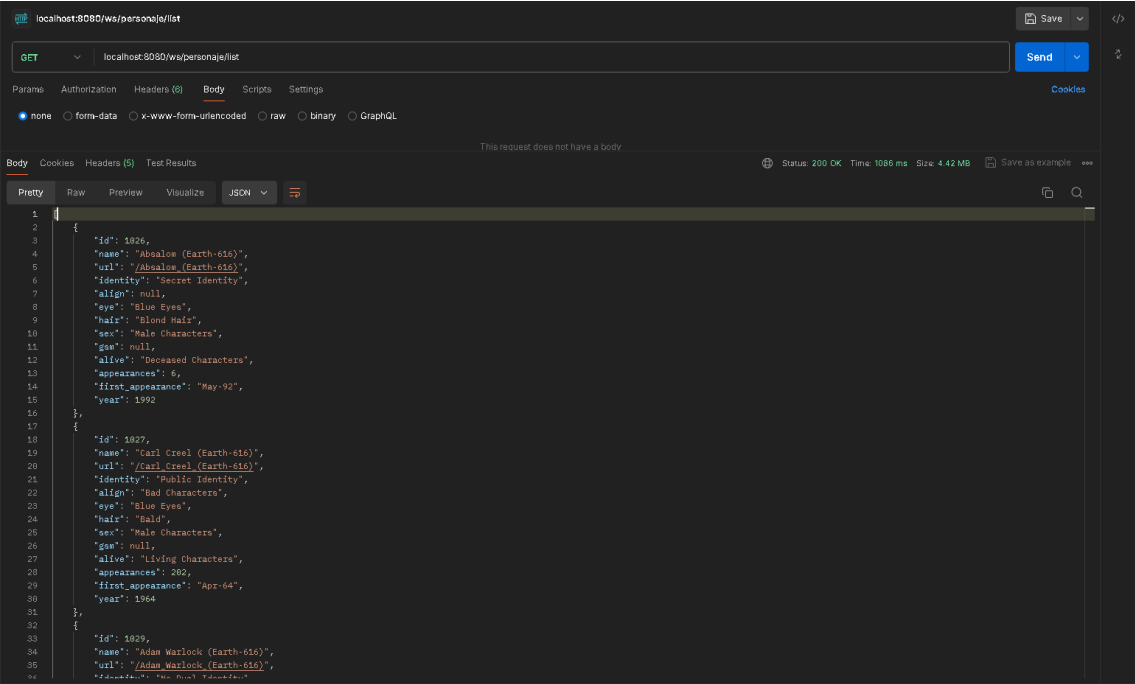
```

]

- Datos:

Campo	Tipo de Dato	Observaciones
id	entero	-
name	cadena	-
url	cadena	-
identity	cadena	-
align	cadena	-
eye	cadena	-
hair	cadena	-
sex	cadena	-
gsm	cadena	-
alive	cadena	-
appearances	entero	-
first_appearance	cadena	-
year	entero	-

Ejemplo:



Consumo API REST para obtener personaje

URL	Method
http://localhost:8080/ws/personaje/{id}	GET

1. Argumentos de entrada:

Campo	Tipo de Dato	Observaciones
id	entero	-

2. Argumentos de salida:

Opción 1: Solicitud correcta

- **Http Status:** HTTP 200 OK
- **Body:**

```

• {
•   "result": "sucess",
•   "personaje": {
•     "id": 1025,
•     "name": "Emil Blonsky (Earth-616)",
•     "url": "/Emil_Blonsky_(Earth-616)",
•     "identity": "Secret Identity",
•     "align": "Bad Characters",
•     "eye": "Blue Eyes",
•     "hair": "Blond Hair",
•     "sex": "Male Characters",
•     "gsm": "",
•     "alive": "Living Characters",
•     "appearances": 115,
•     "first_appearance": "Apr-67",
•     "year": 1967
•   }
• }

```

Opción 2: Error al ingresar un parámetro que no sea entero

- **Http Status:** HTTP 400 Bad Request
- **Body:**

```

• {
•   "date": "2024-05-25T16:12:57.5598748",
•   "error": "ID Personaje NO encontrado."
• }

```

Opción 3: El ID que se busca es menor a 0 o no esta dentro de la lista de ID's existentes

- **Http Status:** HTTP 400 NOT FOUND
- **Body:**

```

• {
•   "date": "2024-05-25T16:12:57.5598748",
•   "error": "ID Personaje NO encontrado."
• }

```

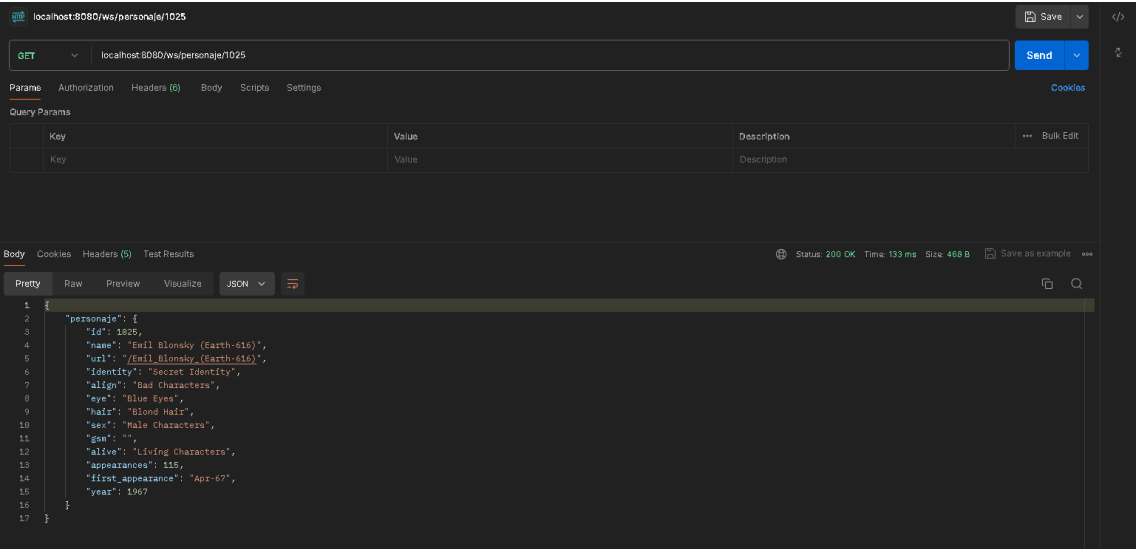
3. Datos argumentos de salida

Campo	Tipo de Dato	Observaciones
-------	--------------	---------------

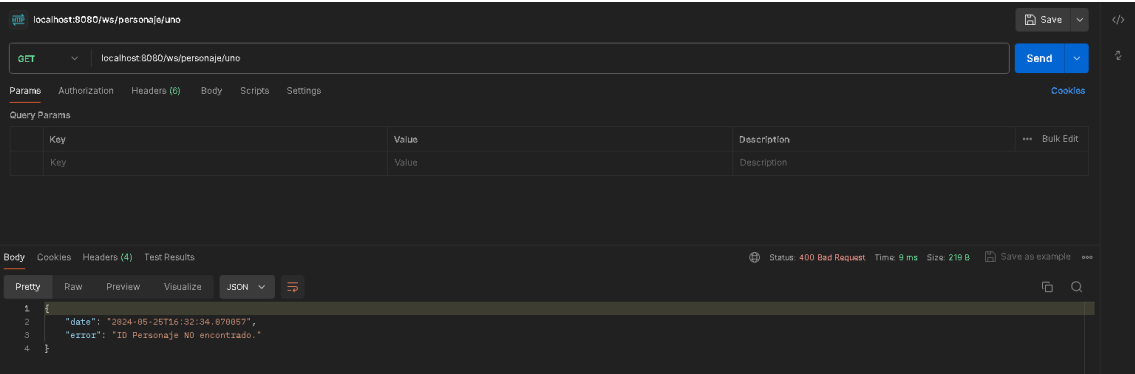
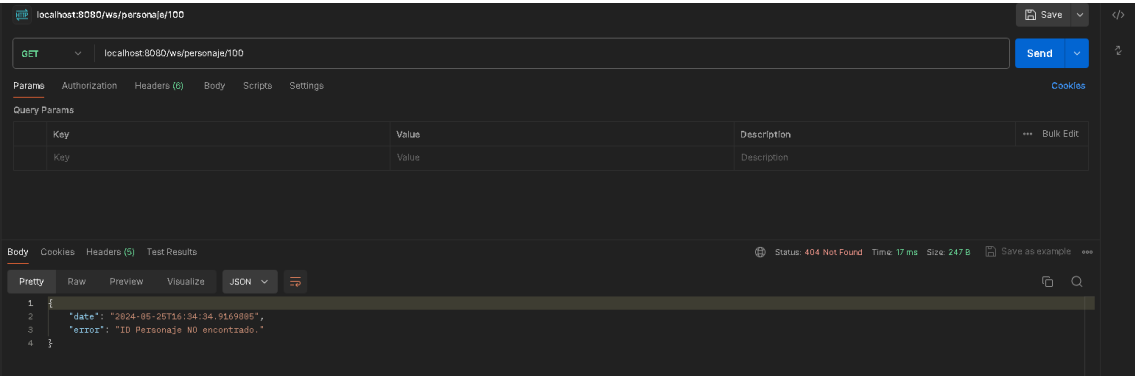
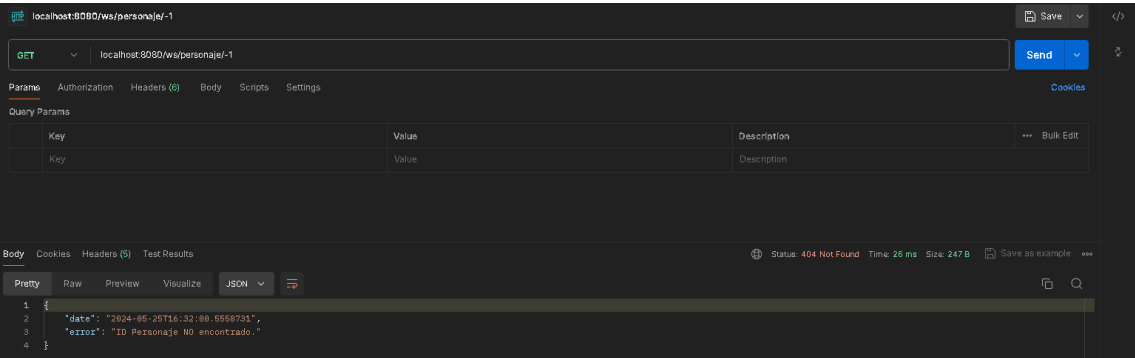
id	entero	-
name	cadena	-
url	cadena	-
identity	cadena	-
align	cadena	-
eye	cadena	-
hair	cadena	-
sex	cadena	-
gsm	cadena	-
alive	cadena	-
appearances	entero	-
first_appearance	cadena	-
year	entero	-

4. Ejemplos

Solicitud correcta:



Solicitud incorrecta:



Consumo API REST para crear o actualizar un personaje

URL	Method
http://localhost:8080/ws/personaje/save	POST
http://localhost:8080/ws/personaje/save	PUT

5. Argumentos de entrada:

Opción 1: Metodo POST

- **Body:**

```
{
  "name": "Prueba Personaje",
  "url": "/Prueba_Personaje",
  "identity": "Secret Identity",
  "align": "Bad Characters",
  "eye": "Blue Eyes",
  "hair": "Blond Hair",
  "sex": "Male Characters",
  "appearances": 115,
  "firstAppearance": "Jun-2019",
  "year": 2019
}
```

Opción 2: Metodo PUT

- **Body:**

```
{
  "id": 1025,
  "name": "Emil Blonsky (Earth-616)",
  "url": "/Emil_Blonsky_(Earth-616)",
  "identity": "Secret Identity",
  "align": "Bad Characters",
  "eye": "Blue Eyes",
  "hair": "Blond Hair",
  "sex": "Male Characters",
  "gsm": "",
  "alive": "Living Characters",
  "appearances": 115,
  "firstAppearance": "Apr-67",
  "year": 1967
}
```

6. Datos de argumentos de entrada:

Campo	Tipo de Dato	Observaciones
id	entero	-
name	cadena	-

url	cadena	-
identity	cadena	-
align	cadena	-
eye	cadena	-
hair	cadena	-
sex	cadena	-
gsm	cadena	-
alive	cadena	-
appearances	entero	-
first_appearance	cadena	-
year	entero	-

7. Argumentos de salida:

Opción 1: Solicitud correcta – Metodo POST

- **Http Status:** HTTP 201 Created
- **Body:**

```

• {
•   "personaje creado": {
•     "id": 755287,
•     "name": "Prueba Personaje",
•     "url": "/Prueba_Personaje",
•     "identity": "Secret Identity",
•     "align": "Bad Characters",
•     "eye": "Blue Eyes",
•     "hair": "Blond Hair",
•     "sex": "Male Characters",
•     "gsm": null,
•     "alive": null,
•     "appearances": 115,
•     "first_appearance": null,
•     "year": 2019
•   }
• }

```

```
• }  
•
```

Opción 2: Solicitud correcta – Metodo PUT

- **Http Status:** HTTP 201 Created
- **Body:**

```
• {  
•   "personaje creado": {  
•     "id": 755287,  
•     "name": "Prueba Personaje",  
•     "url": "/Prueba_Personaje",  
•     "identity": "Secret Identity",  
•     "align": "Bad Characters",  
•     "eye": "Blue Eyes",  
•     "hair": "Blond Hair",  
•     "sex": "Male Characters",  
•     "gsm": null,  
•     "alive": null,  
•     "appearances": 115,  
•     "first_appearance": null,  
•     "year": 2019  
•   }  
• }  
•
```

Opción 3: Error – Metodo POST y PUT

- **Http Status:** HTTP 400 Bad Request
- **Body:**

```
• {  
•   "date": "2024-05-25T17:42:34.253733900",  
•   "error": "Error en validación de Datos."  
• }
```

8. Datos argumentos de salida

Campo	Tipo de Dato	Observaciones
id	entero	-

name	cadena	-
url	cadena	-
identity	cadena	-
align	cadena	-
eye	cadena	-
hair	cadena	-
sex	cadena	-
gsm	cadena	-
alive	cadena	-
appearances	entero	-
first_appearance	cadena	-
year	entero	-

9. Ejemplos

Solicitud correcta:

```
1 {
2   "name": "Prueba Personaje",
3   "url": "/Prueba Personaje",
4   "identity": "Secret Identity",
5   "align": "Bad Characters",
6   "eye": "Blue Eyes",
7   "hair": "Blond Hair",
8   "sex": "Male Characters",
9   "appearances": 115,
10  "firstAppearance": "Jun-2019",
11  "year": 2019
12 }
13
14
15
```

```
1
2  "personaje creado": {
3    "id": 755287,
4    "name": "Prueba Personaje",
5    "url": "/Prueba Personaje",
6    "identity": "Secret Identity",
7    "align": "Bad Characters",
8    "eye": "Blue Eyes",
9    "hair": "Blond Hair",
10   "sex": "Male Characters",
11   "gsm": null,
12   "alive": null,
13   "appearances": 115,
14   "first_appearance": null,
15   "year": 2019
16 }
17
```

localhost:8080/ws/personaje/save

PUT localhost:8080/ws/personaje/save

Body

```
1 {
2   "id": 1826,
3   "name": "Emil Blonsky 2 (Earth-616)",
4   "url": "/Emil_Blonsky_2_(Earth-616)",
5   "identity": "Secret Identity",
6   "align": "Bad Characters",
7   "eye": "Blue Eyes",
8   "hair": "Blond Hair",
9   "sex": "Male Characters",
10  "gsm": "",
11  "alive": "Living Characters",
12  "appearances": 115,
13  "firstAppearance": "Apr-67",
14  "year": 1967
15 }
16
```

Body

Personaje actualizado: {

```
1 {
2   "id": 1826,
3   "name": "Emil Blonsky 2 (Earth-616)",
4   "url": "/Emil_Blonsky_2_(Earth-616)",
5   "identity": "Secret Identity",
6   "align": "Bad Characters",
7   "eye": "Blue Eyes",
8   "hair": "Blond Hair",
9   "sex": "Male Characters",
10  "gsm": "",
11  "alive": "Living Characters",
12  "appearances": 115,
13  "firstAppearance": null,
14  "year": 1967
15 }
16
```

Solicitud incorrecta:

localhost:8080/ws/personaje/save

POST localhost:8080/ws/personaje/save

Body

```
1
2
3
4
```

Body

```
1 {
2   "data": "2024-05-26T17:43:34.263733998",
3   "error": "Error en validación de Datos."
4 }
```

localhost:8080/ws/personaje/save

PUT localhost:8080/ws/personaje/save

Body

```
1
2
3
4
```

Body

```
1 {
2   "data": "2024-05-26T17:43:36.24223998",
3   "error": "Error en validación de Datos."
4 }
```

Consumo API REST para eliminar un personaje

URL	Method
http://localhost:8080/ws/personaje/delete/{id}	DELETE

10. Argumentos de entrada:

Campo	Tipo de Dato	Observaciones
id	entero	-

11. Argumentos de salida:

Opción 1: Solicitud correcta

- **Http Status:** HTTP 200 OK
- **Body:**

Opción 2: ID ingresado no esta en la lista de ID's existentes

- **Http Status:** HTTP 400 NOT FOUND
- **Body:**

```
{
  "date": "2024-05-25T16:12:57.5598748",
  "error": "ID Personaje NO encontrado."
}
```

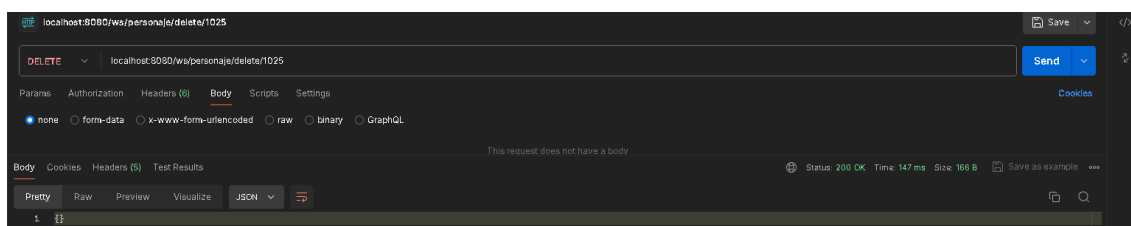
12. Datos argumentos de salida

Campo	Tipo de Dato	Observaciones
id	entero	-
name	cadena	-

url	cadena	-
identity	cadena	-
align	cadena	-
eye	cadena	-
hair	cadena	-
sex	cadena	-
gsm	cadena	-
alive	cadena	-
appearances	entero	-
first_appearance	cadena	-
year	entero	-

13. Ejemplos

Solicitud correcta:



localhost:8080/ws/personaje/list

GET localhost:8080/ws/personaje/list

Params Authorization Headers (6) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have a body

Status: 200 OK Time: 643 ms Size: 4.42 MB Save as example

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "id": 1026,
4     "name": "Absalom (Earth-616)",
5     "url": "/Absalom (Earth-616)",
6     "identity": "Secret Identity",
7     "align": null,
8     "eye": "Blue Eyes",
9     "hair": "Blond Hair",
10    "sex": "Male Characters",
11    "gaw": null,
12    "alive": "Deceased Characters",
13    "appearances": 6,
14    "first_appearance": "May-92",
15    "year": 1992
16  },
17  {
18    "id": 1027,
19    "name": "Carl Creel (Earth-616)",
20    "url": "/Carl_Creel (Earth-616)",
21    "identity": "Public Identity",
22    "align": "Bad Characters",
23    "eye": "Blue Eyes",
24    "hair": "Bald",
25    "sex": "Male Characters",
26    "gaw": null,
27    "alive": "Living Characters",
28    "appearances": 282,
29    "first_appearance": "Apr-64",
30    "year": 1964
31  },
32  {
33    "id": 1029,
34    "name": "Adam Warlock (Earth-616)",
35    "url": "/Adam_Warlock (Earth-616)",
36    "identity": "Public Identity",
37    "align": "Good Characters",
38    "eye": "Blue Eyes",
39    "hair": "Black Hair",
40    "sex": "Male Characters",
41    "gaw": null,
42    "alive": "Living Characters",
43    "appearances": 282,
44    "first_appearance": "Apr-64",
45    "year": 1964
46  }
47 }
```

Solicitud incorrecta:

localhost:8080/ws/personaje/delete/10

DELETE localhost:8080/ws/personaje/delete/10

Params Authorization Headers (6) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have a body

Status: 404 Not Found Time: 889 ms Size: 247 B Save as example

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "date": "2024-05-26T18:00:01.4887154",
3   "error": "ID Personaje NO encontrado."
4 }
```