



ESPECIFICACIÓN SERVICIO REST PARA LA ENTIDAD PERSONAJE

Pacto de Honor

Facultad de Ingeniería y Ciencias básicas.
Politécnico Grancolombiano.

Contenido

Control de versiones	3
Objetivo	4
URL	4
Métodos	5
find	5
Satisfactorio:	5
No satisfactorio:	5
create	6
Satisfactorio:	7
No satisfactorio:	7
edit	8
Satisfactorio:	8
No satisfactorio:	8
getData	9
count	9

Control de versiones

Versión	Fecha	Autor	Revisado por	Cambios realizados
0.1	11/04/2017	Nicolás Rubiano		Versión inicial del documento.

Objetivo

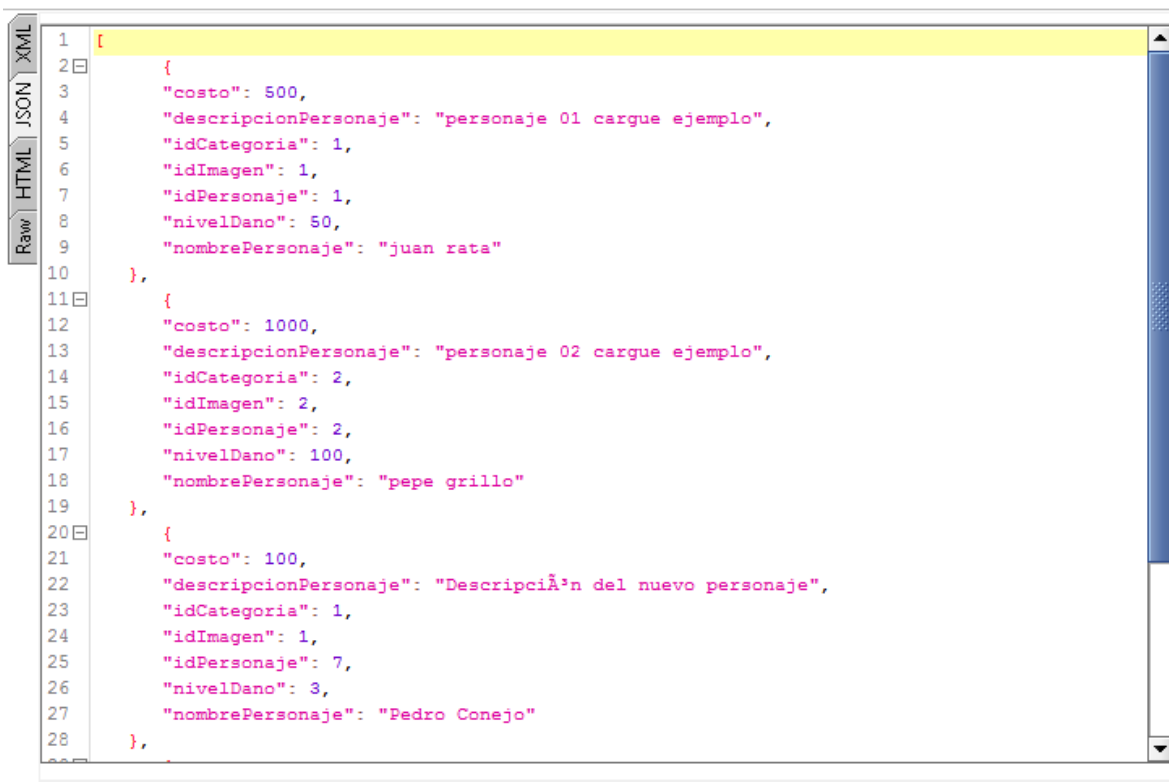
El objetivo del presente documento, es brindar herramientas de conocimiento los desarrolladores que desean consumir los servicios REST que proporcionará el área de desarrollo de Backend, específicamente sobre la entidad Personaje.

Los siguientes, son las especificaciones que se proporcionarán para tener acceso a todos los datos referentes sobre la entidad:

URL

<http://IP:Puerto/Polifight/webresources/personaje>

Al consumir la URL directa del servicio, por método GET, para la entidad Personaje, se listarán todos los datos que se encuentren en la DB:



The image shows a code editor window with a sidebar on the left containing tabs for 'Raw', 'HTML', 'JSON', and 'XML'. The 'JSON' tab is selected. The main editor area displays a JSON array of three objects, each representing a character. The objects are formatted with syntax highlighting and line numbers from 1 to 28. The first object has a cost of 500, description 'personaje 01 cargue ejemplo', category 1, image 1, damage level 50, and name 'juan rata'. The second object has a cost of 1000, description 'personaje 02 cargue ejemplo', category 2, image 2, damage level 100, and name 'pepe grillo'. The third object has a cost of 100, description 'DescripciÃ³n del nuevo personaje', category 1, image 1, damage level 3, and name 'Pedro Conejo'.

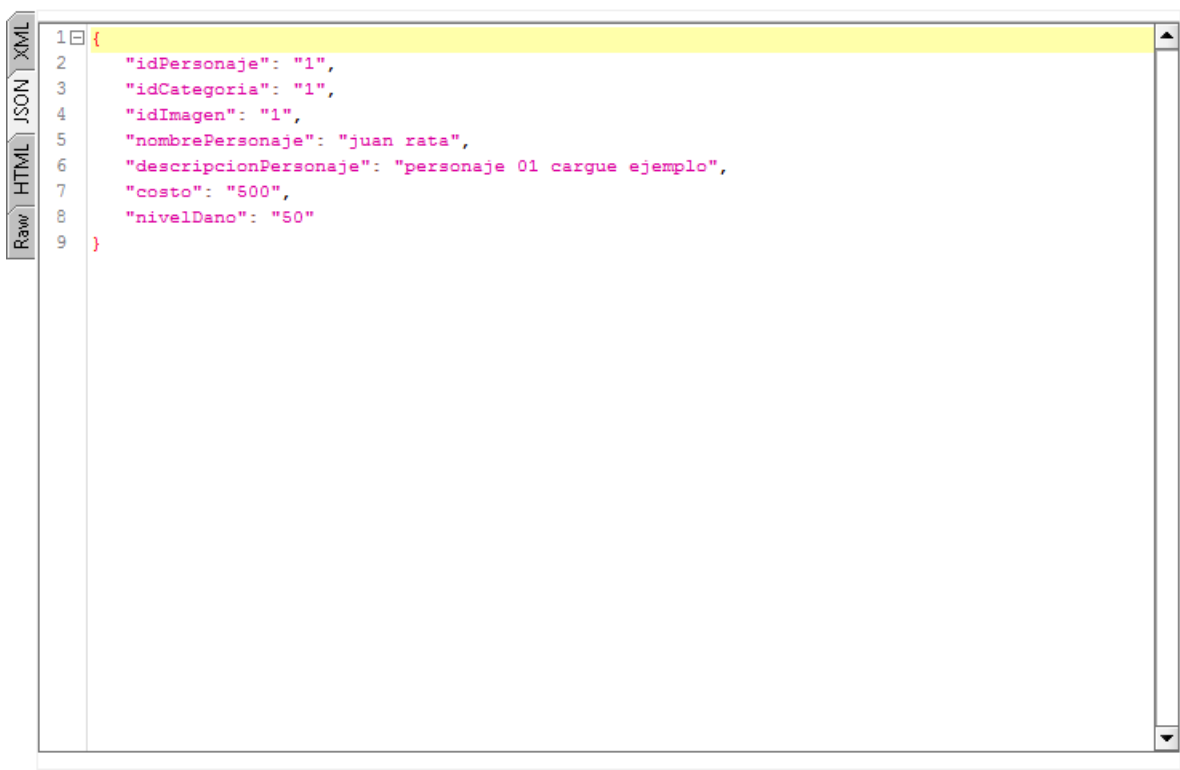
```
1  [
2  {
3      "costo": 500,
4      "descripcionPersonaje": "personaje 01 cargue ejemplo",
5      "idCategoria": 1,
6      "idImagen": 1,
7      "idPersonaje": 1,
8      "nivelDano": 50,
9      "nombrePersonaje": "juan rata"
10 },
11 {
12     "costo": 1000,
13     "descripcionPersonaje": "personaje 02 cargue ejemplo",
14     "idCategoria": 2,
15     "idImagen": 2,
16     "idPersonaje": 2,
17     "nivelDano": 100,
18     "nombrePersonaje": "pepe grillo"
19 },
20 {
21     "costo": 100,
22     "descripcionPersonaje": "DescripciÃ³n del nuevo personaje",
23     "idCategoria": 1,
24     "idImagen": 1,
25     "idPersonaje": 7,
26     "nivelDano": 3,
27     "nombrePersonaje": "Pedro Conejo"
28 },
29 ]
```

Métodos

find

EndPoint: /Polifight/webresources/personaje/find

Este método se deberá consumir por método GET. Se recibirá el id (Parámetro "id") del Personaje y retornará todos los atributos asociados a dicho personaje:



```
1 {  
2   "idPersonaje": "1",  
3   "idCategoria": "1",  
4   "idImagen": "1",  
5   "nombrePersonaje": "juan rata",  
6   "descripcionPersonaje": "personaje 01 cargue ejemplo",  
7   "costo": "500",  
8   "nivelDano": "50"  
9 }
```

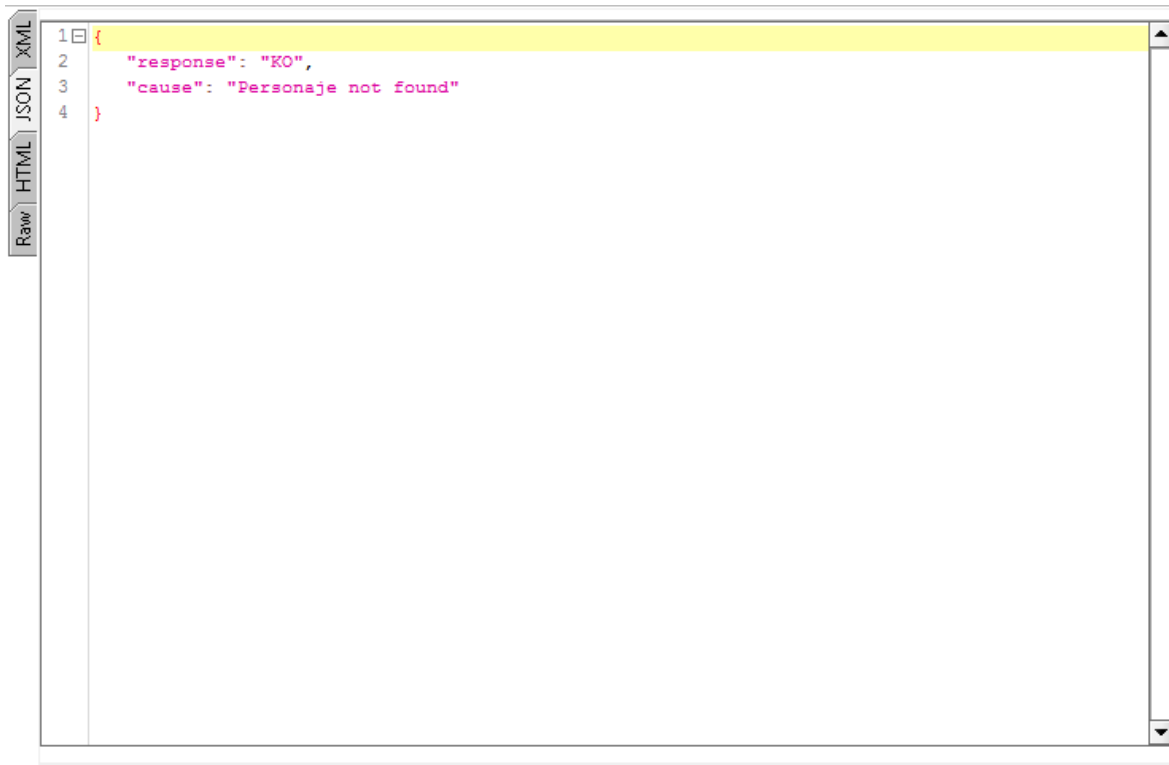
Satisfactorio:

En el caso satisfactorio, se enviarán los atributos de la entidad en formato JSON.

No satisfactorio:

Si el servicio tiene algún problema con la solicitud, se responderán dos campos:

- "response": En caso de algún inconveniente se responderá "KO".
- "cause": Causa del error, para éste método se identifica sólo 1 error, Id del personaje no encontrado en la DB



A screenshot of a web client interface. On the left, there are tabs for 'Raw', 'HTML', 'JSON', and 'XML'. The 'JSON' tab is selected. The main area displays a JSON object with the following structure:

```
1 {
2   "response": "KO",
3   "cause": "Personaje not found"
4 }
```

create

EndPoint: /Polifight/webresources/personaje/create

Se debe enviar un JSON por el método POST con los atributos de la entidad Personaje:



A screenshot of a web client interface. At the top, there is a 'Media Type' dropdown menu set to 'application/json' and a checkbox for 'Post QueryString' which is unchecked. The main area displays a JSON object with the following structure:

```
{
  "nombrePersonaje": "Miguel Vaca",
  "descripcionPersonaje": "Descripción de la vaca",
  "costo": "230",
  "nivelDano": "20",
  "idImagen": "1",
  "idCategoria": "1"
}
```

Es importante recalcar que los nombres de los atributos que se envían deben tener los nombres exactos que muestran en la imagen anterior.

Satisfactorio:

El servicio responde OK en el campo “response” en caso de que la creación se haya realizado con éxito:



Result Grid							
Filter Rows:		Edit:		Export/Import:		Wrap Cell Content:	
	id_personaje	id_categoria	id_imagen	nombre_personaje	descripcion_personaje	costo	nivel_dano
	7	1	1	Pedro Conejo	Descripción del nuevo personaje	100	3
	8	1	1	Pedro Pez	Descripción del nuevo personaje	100	3
	9	1	1	Miguel Vaca	Descripción de la vaca	230	20
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

No satisfactorio:

Si el servicio no puede procesar la solicitud se responderá un KO en el campo “response”. En este caso se incluirá el campo “cause” donde se indicará la causa del error

```
{
  response: KO
  cause: Time Out
}
```

edit

EndPoint: /Polifight/webresources/personaje/edit

Método que servirá para modificar una entidad Jugador. Se debe consumir por POST enviando un formato JSON con el parámetro "id", éste es obligatorio, y seguido de los atributos que se deseen modificar.

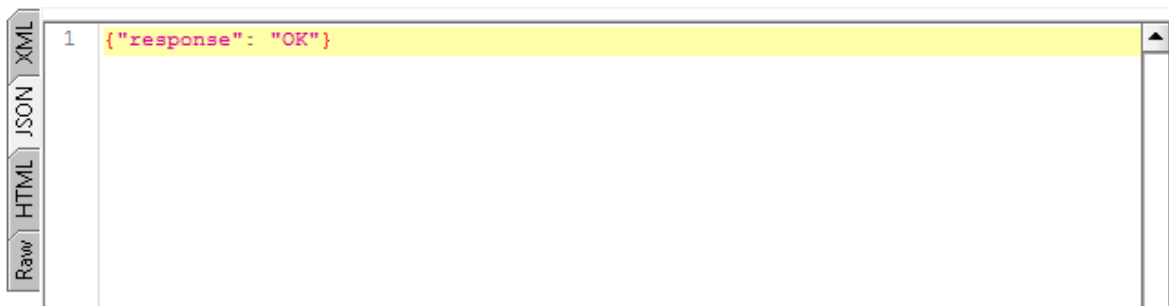
Para efectos de ejemplo, modificaremos el nombre del dato insertado anteriormente en el ejemplo del método [create](#), por "Lola Vaca" enviando el siguiente JSON:



```
{
  "idPersonaje": "9",
  "nombrePersonaje": "Lola Conejo"
}
```

Satisfactorio:

Si la modificación al atributo fue satisfactorio, se responderá con el campo "response": "OK":



```
1 {"response": "OK"}
```

No satisfactorio:

Si el servicio no puede procesar la solicitud se responderá un KO en el campo "response". En este caso se incluirá el campo "cause" donde se indicará la causa del error

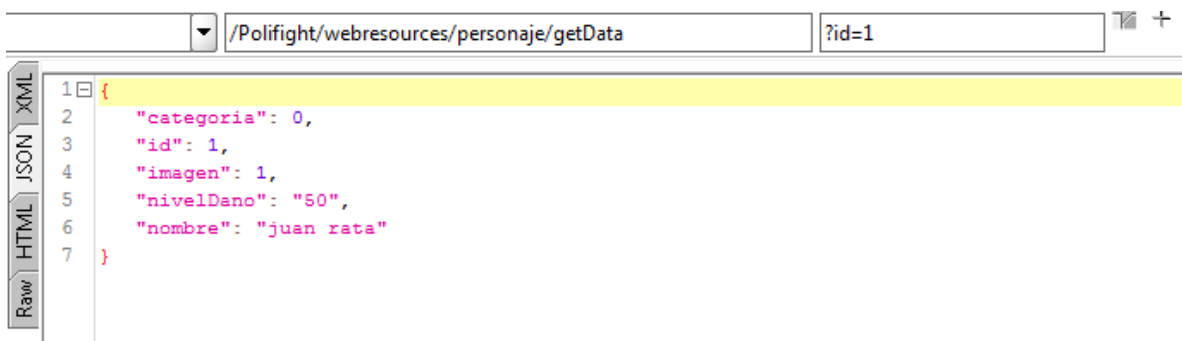


```
1 {
2   "response": "KO",
3   "cause": "Personaje not found"
4 }
```

getData

URL: /Polifight/webresources/personaje/getData

Este método es solicitado con una estructura específica por el equipo de FrontEnd el día Miércoles 5 de Abril. Se consume por método GET y debe enviarse el parámetro “id” y retorna la siguiente estructura:



```
1 {
2   "categoria": 0,
3   "id": 1,
4   "imagen": 1,
5   "nivelDano": "50",
6   "nombre": "juan rata"
7 }
```

count

EndPoint: /Polifight/webresources/personaje/count

Este método retorna la cantidad de datos que se encuentran en la DB. Se debe consumir por método GET directamente sin parámetros:



```
5
```