



ESPECIFICACIÓN SERVICIO REST PARA LA ENTIDAD IMAGEN

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:	6
No satisfactorio:	7
edit	8
Satisfactorio:	8
No satisfactorio:	9
count	9

Control de versiones

Versión	Fecha	Autor	Revisado por	Cambios realizados
0.1	13/04/2017	Nicolás Rubiano		Versión inicial del documento.

Objetivo

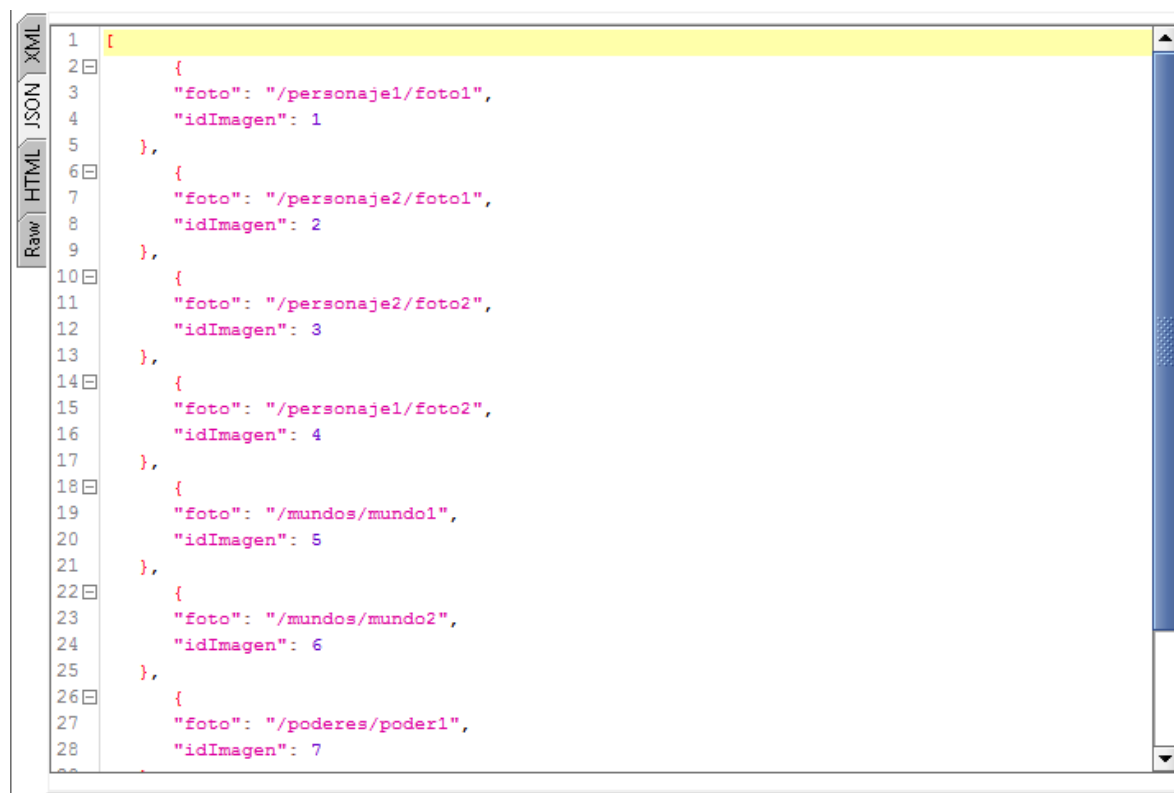
El objetivo del presente documento, es brindar herramientas de conocimiento los desarrolladores que desean consumir los servicios REST que proporcionará el área de desarrollo de Backend, específicamente sobre la entidad Imagen.

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/imagen>

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 screenshot of a code editor with a tab labeled 'Raw' selected. The editor displays a JSON array of objects, each representing an image. The objects are listed as follows:

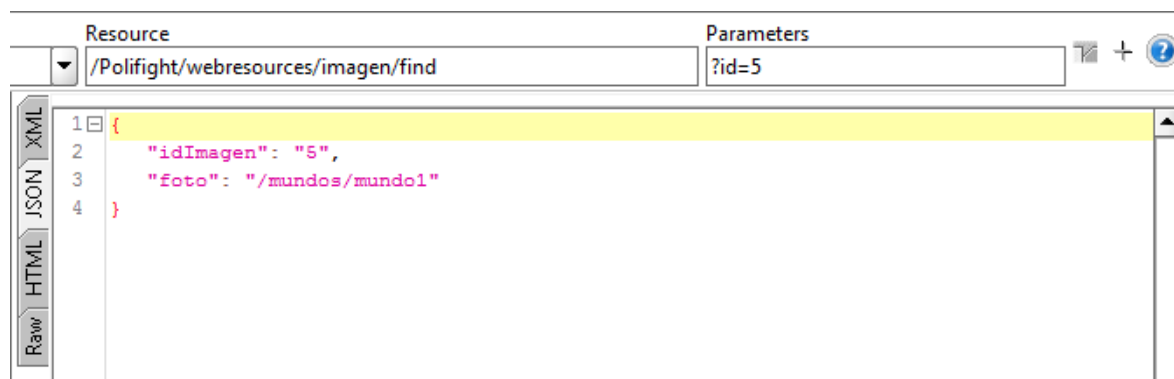
```
1  [
2    {
3      "foto": "/personaje1/foto1",
4      "idImagen": 1
5    },
6    {
7      "foto": "/personaje2/foto1",
8      "idImagen": 2
9    },
10   {
11     "foto": "/personaje2/foto2",
12     "idImagen": 3
13   },
14   {
15     "foto": "/personaje1/foto2",
16     "idImagen": 4
17   },
18   {
19     "foto": "/mundos/mundo1",
20     "idImagen": 5
21   },
22   {
23     "foto": "/mundos/mundo2",
24     "idImagen": 6
25   },
26   {
27     "foto": "/poderes/poder1",
28     "idImagen": 7
29   },
30 ]
```

Métodos

find

EndPoint: /Polifight/webresources/imagen/find

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



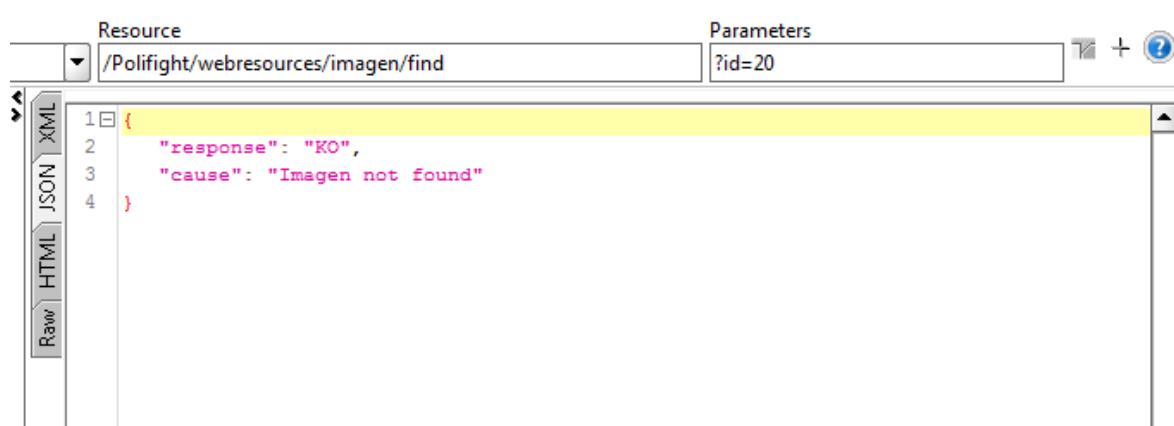
Satisfactorio:

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 de la imagen no encontrado en la DB



create

EndPoint: /Polifight/webresources/imagen/create

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



The image shows a screenshot of a REST client interface. At the top, there is a 'Media Type' dropdown menu set to 'application/json'. To the right of the dropdown is a checkbox labeled 'Post QueryString' which is unchecked. Below the header, the request body is displayed in a text area. The body contains a JSON object:

```
{  "foto": "/mundos/mundo2.png"}

```

 The text is color-coded: the opening curly brace is red, the key "foto" is in pink, and the value is in black. The closing curly brace is red. The entire JSON object is highlighted with a yellow background.

Es importante recalcar que el/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:
id_imagen	foto		
4	/personaje1/foto2		
5	/mundos/mundo1		
6	/mundos/mundo2		
7	/poderes/poder1		
8	/poderes/poder2		
9	/mundos/mundo2.png		
*	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/imagen/edit

Método que servirá para modificar una entidad Imagen. Se debe consumir por POST enviando un formato JSON con el parámetro "id". Éste es obligatorio, y seguido del/los atributo(s) que se deseen modificar.

Para efectos de ejemplo, modificaremos el nombre del datos insertado anteriormente en el ejemplo del método [create](#).

Media Type: ☐ Post QueryString

```
{
  "idImagen": "9",
  "foto": "/mundos/newMundo2.png"
}
```

Result Grid		Filter Rows:	Edit:	Exp
id_imagen	foto			
4	/personaje1/foto2			
5	/mundos/mundo1			
6	/mundos/mundo2			
7	/poderes/poder1			
8	/poderes/poder2			
9	/mundos/newMundo2.png			
* NULL	NULL			

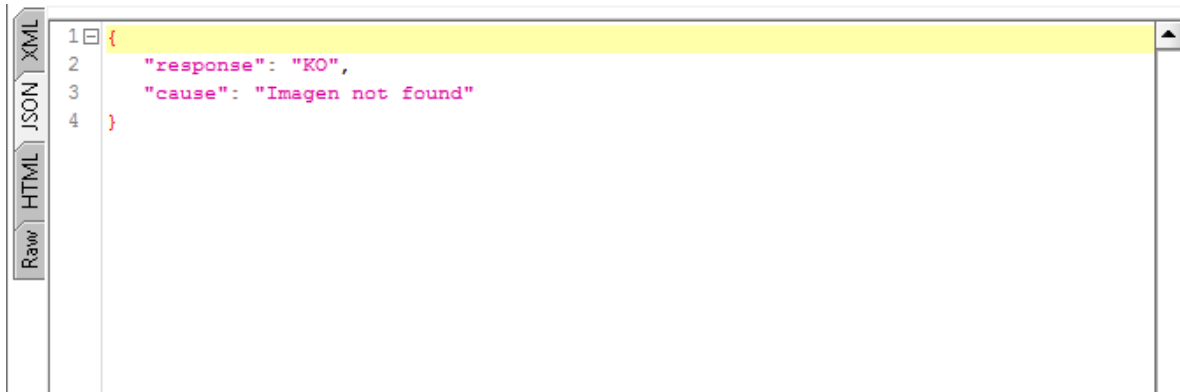
Satisfactorio:

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

Raw	HTML	JSON	XML
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



The screenshot shows a REST client interface with a sidebar on the left containing 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": "Imagen not found"  
4 }
```

count

EndPoint: /Polifight/webresources/imagen/count

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

