



ESPECIFICACIÓN SERVICIO REST PARA LA ENTIDAD LOG

Pacto de Honor

Facultad de Ingeniería y Ciencias básicas.
Politécnico Grancolombiano.

Contenido

Control de versiones	3
Objetivo.....	4
Métodos	4
find	4
Satisfactorio:	4
No satisfactorio:	4
create	5
Satisfactorio:	5
No satisfactorio:	5
count	6
listUsers.....	6

Control de versiones

Versión	Fecha	Autor	Revisado por	Cambios realizados
0.1	06/05/2017	Edna Espejo		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, sobre la entidad Log.

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

Métodos

find

EndPoint: /Polifight/ webresources/log/find

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

Satisfactorio:

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

Resource		Parameters
/L/webresources/log/find		?id=1
1	{ 'idLog': '1', 'idJugador': '1' 'tiempoLog': 'Thu Jan 01 16:14:23 COT 1970' } 'fe	

No satisfactorio:

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

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

Resource		Parameters
/L/webresources/log/find		?id=50
1	{	
2	"response": "Error",	
3	"cause": "Log not found"	
4	}	

create

EndPoint: /Polifight/webresources/log/create

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

Media Type: ☐ Post QueryString

```
{
  "idLog": 3,
  "idJugador": 1,
  "tiempoLog": "1970-01-01T16:14:42-05:00",
  "fechaInicio": "2017-05-06T16:14:45-05:00",
  "fechaFinal": "2017-05-06T16:14:51-05:00",
  "idNivel": 2,
  "idMundo": 1,
  "idPersonaje": 1
}
```

Satisfactorio:

El servicio responde Log create en el campo "response" en caso de que la creación se haya realizado con éxito:

Resource: /L/webresources/log/create

1 ☐ XML ☒ JSON ☐ Text

```
{'response': 'Log create'}
```

id_log	id_jugador	tiempo_log	fecha_inicio	fecha_final	id_nivel	id_mundo	id_personaje
1	1	16:14:23	2017-03-02 16:14:25	2017-03-04 16:14:31	NULL	NULL	NULL
2	1	16:14:42	2017-03-03 16:14:45	2017-03-04 16:14:51	NULL	NULL	NULL
3	1	16:14:42	2017-05-06 16:14:45	2017-05-06 16:14:51	2	1	1

No satisfactorio:

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

```
{
  "response": "Error",
  "cause": "Descripción del error"
}
```

count

EndPoint: /Polifight/webresources/log/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:

Resource	
	/L/webresources/log/count
4	

listLogs

EndPoint: /Polifight/webresources/log/listLogs

Se debe realizar el consumo por método GET. Se retorna una lista en formato JSON con todos los datos contenidos en la base de datos:

Resource		Parameters
	/L/webresources/log/listLogs	
1	[{"idLog": "null", "idJugador": "1", "tiempoLog": "Thu Jan 01 16:14:23"}]	