



ESPECIFICACIÓN SERVICIO REST PARA LA ENTIDAD JUGADOR TIENE PERSONAJE

Pacto de Honor

Facultad de Ingeniería y Ciencias básicas.
Politécnico Grancolombiano.

Contenido

Control de versiones	3
Objetivo	4
Métodos	4
getID	4
Parámetro:	4
Consumo invalido:.....	4
create	5
Satisfactorio:	5
No satisfactorio:	5
count	6
find	6
Satisfactorio:	6
No satisfactorio:	6

Control de versiones

Versión	Fecha	Autor	Revisado por	Cambios realizados
0.1	18/05/2017	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 Jugador tiene personaje.

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

Métodos

getID

EndPoint: /Polifightwebresources/jugadortienepersonaje/getId

Este método retorna una lista con los Id con todos los registros existentes en la entidad. Se debe consumir por método GET y recibe como parámetro el token que fue enviado al realizar el login.

Parámetro:

“token”



Imagen 1: Lista de los idJugadortienepersonaje al consumir el método getId enviando el token

Consumo invalido:

Muestra el mensaje de error en caso de enviar un token inválido



Imagen 2: Consumir método getId enviando token inválido

create

EndPoint: /Polifight/webresources/jugadortienepersonaje/create

Se debe enviar un JSON por el método POST con los atributos de la entidad Jugador tiene personaje y se debe enviar el token para que permita la creación

Parameters

?token=87252C725AA64E2CB104E8A985B47405170417155

Media Type: application/json

```
{
  "idJugador": 1,
  "idPersonaje": 4,
  "idJugadortienepersonaje": 3
}
```

Imagen 3: Estructura JSON para la asignación de un personaje a un jugador

Satisfactorio:

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

Resource

Polifight/webresources/jugadortienepersonaje/create

1 {"response": "OK"}

id_jugador	id_personaje	id_Jugador_tiene_personaje
1	1	1
1	2	2
1	3	3
1	4	4

Imagen 4: Respuesta satisfactoria del servicio create y visualización de los datos creados en la BD

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



Imagen 5: Consumir método create enviando token inválido

count

EndPoint: /Polifight/webresources/jugadortienepersonaje /count

Este método retorna la cantidad de datos en formato HTML que se encuentran en la DB. Se debe consumir por método GET y recibe como parámetro el token de login:

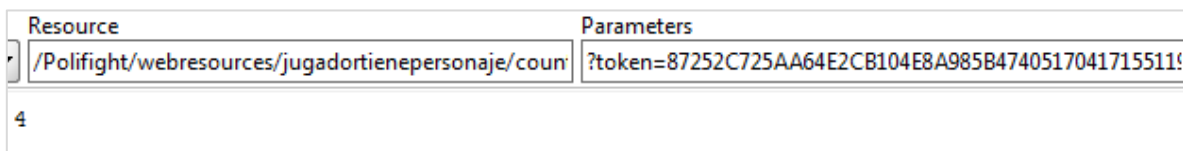


Imagen 6: Consumir método count enviando token válido

find

EndPoint: /Polifight/ webresources/ jugadortienepersonaje /find

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

Satisfactorio:

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



Imagen 7: Consumir método find enviando token válido

No satisfactorio:

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

- "response": KO 'Id not found' cuando el id no existe
- "response": KO 'token invalid' cuando el token es incorrecto
-

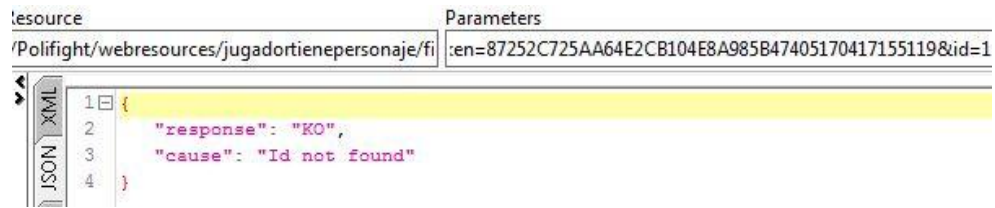


Imagen 8: Consumir método find enviando id inexistente



Imagen 9: Consumir método find enviando token inválido