Grupo 2: Lucas Montelongo, Maximilano Langorta, Mauro Piston, Sebastian Diaz

BLOG

¿Qué es una API?

Una API(Interfaz de Programación de Aplicaciones) es un conjunto de subrutinas, funciones y métodos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

WEB SERVICE

Es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

REST

Es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP, nos permite crear servicios y aplicaciones que pueden ser usadas por cualquier dispositivo o cliente que entienda HTTP, por lo que es más simple y convencional que otras alternativas como por ejemplo SOAP.

Nuestra API

La API está organizada en torno a Api REST para un manejo más sencillo de nuestra interfaz, esta es orientada a recursos, y utiliza códigos de respuesta HTTP para indicar errores. Usamos verbos HTTP, que son entendidos por los clientes HTTP estándar, fue elegido JSON para las respuestas de la API ya que brinda una comunicación ligera y estandarizada. Los métodos utilizados fueron elaborados pensando siempre en la practicidad de uso para cualquier aplicación a desarrollar, y aprovechando los estándares de HTML y Api REST para integrar una interfaz mas intuitiva.

PARAMETROS PARA LISTA DE PUBLICACIONES		
PARAMETRO	**EJEMPLO**	**DESCRIPCION**
_page={int}	*?_page=2*	Determina la

		página a devolver, comenzando siempre en 0, los elementos son agrupados
{Atributo}={value}	*?title=titulo_post&author=persona*	Exige una igualdad en un atributo determinado en su devolucion
_start={int}	*?_start=10*	Determina primer elemento a devolver
_end={int}	*?_end=10*	Determina último elemento a devolver
_limit={int}	*?_limit=40*	Establese cantidad de elementos a devolver(tamaño de pagina)
_sort={int,(asc/desc)}	*?_sort=id,asc*	Determina el orden de los elementos
{Atributo}_gte={int}	*?id_gte=10*	Determina valor mínimo
{Atributo}_lte={int}	*?id_lte=20*	Determina valor máximo
{Atributo}_like={string}	*?title_like=Hola%20Mundo* *?title_like=__ola%20___*	Pensado para hacer búsquedas en un campo especifico
?q={string}	*?q=internet*	Pensado para hacer búsquedas en todos los campos
View={array de atributo}	*?View=Title,Author*	Determinará los campos que desea obtener

Listar publicaciones	
Descripción:	Método utilizado para solicitar una lista de publicaciones, utilizando o no algún filtro, por defecto sin aclaración

	alguna se darán los primeros 10 registros encontrados.
Verbo: GET	Ruta de acceso : /posts/
Request	**Response**

Ver en tabla de Filtros	Content-Type application/json
	{
	result:
	[
	{
	title:”Titulo 1”,
	author:”Autor 1”
	},{
	title: ”Titulo 1”,
	author: ”Autor 1”
	}
	],
	page:0,
	totl:12
	}

Posibles códigos de estado	200,201,204,400 ,405,406,409,415,500

Ver publicación	
Descripción:	Método utilizado para solicitar una publicación, el único parámetro aceptado es “*View*” (ver en tabla de filtros) para establecer los datos deseados, de igual manera sin parámetros devolverá “*id*”, “*title*”, “*category*”, “*tag*”, “*full*”, “*files*”, “*state*”, “*autor*”, “*creation*”.

Verbo: GET	Ruta de acceso : /posts/{post_id}
Request	**Response**

	Content-Type application/json
	{
	id: 1,
	title: "Title_post",
	
	category:["categoria1","categoria2",...
	],
	tag:["etuiqueta1","etuiqueta2",...
	],
	introduction: "Vrebe descripcion"
	full:"Cuerpo del articulo ... ",
	files:["Url_archivo
	adjunot1","Url_archivo adjunot2",...],
	state: true,
	author: "Nick",
	creation : 27/04/2018 23:00
	}

Posibles códigos de estado	200,201,204,400 ,405,406,409,415,500

Crear publicación		
Descripción:	Método utilizado para crear publicación mediante post, recibirá un json con los datos requeridos y retornara un código ademas de un id del nuevo post	
Verbo: POST		Ruta de acceso : /posts/create
Request		**Response**
Content-Type application/json		Content-Type application/json
{ id: user_id, title: Title_post, category: name_Categoria, tag: etuiqueta1, introduction: "texto maximo 60 caracteres" full:"Cuerpo del articulo ... ", files:["Url_archivo adjunot1","Url_archivo adjunot1",...], state: true/false }		*{ "id": new_post_id, "status_code": Código de respuesta }*
Ejemplo Request		
{ id: 1, title: "Title_post", category:[1,2,200], tag:["etuiqueta1","etuiqueta2"], introduction: "texto maximo 60 caracteres" full:"Cuerpo del articulo ... ", files:["Url_archivo_adjunot_1","Url_archivo_adjunot_2",...], state: true/false }		
Posibles códigos de estado		200,201,204,400 ,405,406,409,415,500
OVSERVACIONES	id, title, category, full son atributos obligatorios	

Editar publicación		
Descripción:	Método utilizado para editar una publicación, llamada a una url agregado al id del post a editar y pasándole como parámetro un json con el id del usuario autor además de los nuevos datos, utilizando el método put	
Verbo:	PUT	Ruta de acceso : /posts/{post_id}
Request		**Response**
Content-Type application/json		Content-Type application/json
{ id: user_id, … }		*{ "status_code": Código de respuesta }*
Posibles códigos de estado		200,400,401,403,406,409,415,500
Ejemplo Request		
{ id: user_id,, title: "Nuevo titulo" }		
OBSERVACIONES	En el argumento además del id recibirá todos los campos modificados	

Publicar / Despublicar		
Descripción:	Método utilizado Publicar o despublicar de forma rápida, mediante put llamado desde la url /posts/ y el id del post, pasándole el id y el nuevo estado conjunto con el id del usuario lograran cambiar el estado de este.	
Verbo:	PUT	Ruta de acceso : / posts /{post_id}
Request		**Response**
Content-Type application/json		Content-Type application/json
{ id: user_id, estado: true/false }		*{ "status_code": Código de respuesta }*
Ejemplo Request		
{ id: 1, estado: true }		
Posibles códigos de estado		200,400,401,403,406,409,415,500

Marcar publicación cómo favorita		
Descripción:	Método utilizado para agregar una publicación a los favoritos de un usuario. Necesitara el id del post como argumento, llamado en user/ y el id del usuario	
Verbo:	PUT	Ruta de acceso : /user/{user_id}
Request		**Response**

Content-Type application/json	Content-Type application/json
{ id_posts: post_id }	*{ "status_code": Código de respuesta }*
Ejemplo Request	
{ id_posts: 1 }	
Posibles códigos de estado	200,400,401,403,406,409,415,500

Mostrar Perfil de usuario	
Descripción:	Método utilizado para solicitar mediante get los datos de los usuarios, el único parámetro aceptado es "*View*" (ver en tabla de filtros) para establecer los datos deseados, de no utilizarlo devolverá "*avatar*", "*nick*", " *articles*", "*favortios*"
Verbo: GET	Ruta de acceso : /user/user_id
Request	**Response** application/json
/?user_id={int}	*{ avatar:url_avatar, nick: Nick_de_usuarioo,, articles:[id_post, id_post…], favortios:[id_post, id_post…], }*
Ejemplo Response	
{ avatar:"http://muraknockout.com/wp-content/uploads/2015/05/generica-featured-image.jpg", nick:"Nick", articles:[1, 2], favortios:[2,3] }	
Posibles códigos de estado	200,204,400 ,405,406,409,415,500

Actualizar avatar	
Descripción:	Método put requiere un avatar como argumento, y retornara un código de estado.
Verbo: PUT	Ruta de acceso : /user/{user_id}
Request	**Response**
Content-Type application/json	Content-Type application/json
{ avatar: url_new_avatar }	*{ "status_code": Código de respuesta }*
Ejemplo Request	
{ avatar: "http://muraknockout.com/wp-content/uploads/2015/05/generica-	

featured-image.jpg” *}*	
Posibles códigos de estado	200,400,401,403,406,409,415,500

Códigos de estado

Código de respuesta	Descripción
200 OK	Solicitud aceptada; la respuesta contiene el resultado. Este es un código de respuesta general a cualquier solicitud. En las solicitudes GET, el recurso o datos solicitados están en el cuerpo de la respuesta.
201 CREADOS	Las operaciones PUT o POST devuelven este código de respuesta e indica que se ha creado un recurso de forma satisfactoria. El cuerpo de la respuesta podría, por ejemplo, contener información acerca de un nuevo recurso o información de validación (por ejemplo, cuándo se actualiza un activo).
204 SIN CONTENIDO	Indica que se ha aceptado la solicitud, pero no había datos para devolver. Este respuesta se devuelve cuando se ha procesado la solicitud, pero no se ha devuelto ninguna información adicional acerca de los resultados.
400 PETICIÓN INCORRECTA	La solicitud no fue válida. Este código se devuelve cuando el servidor ha intentado procesar la solicitud, pero algún aspecto de la solicitud no es válido; por ejemplo, un recurso formateado de forma incorrecta o un intento de despliegue de un proyecto de sucesos no válido en el tiempo de ejecución de sucesos. La información acerca de la solicitud se proporciona en el cuerpo de la respuesta e incluye un código de error y un mensaje de error.
401 NO AUTORIZADO	El servidor de aplicaciones devuelve este código de respuesta cuando está habilitada la seguridad y faltaba la información de autorización en la solicitud.
403 PROHIBIDO	Indica que el cliente ha intentado acceder a un recurso al que no tiene acceso.
404 NO ENCONTRADO	Indica que el recurso de destino no existe. Esto podría deberse a que el URI no está bien formado o a que se ha suprimido el recurso.
405 MÉTODO NO PERMITIDO	Se produce cuando el recurso de destino no admite el método HTTP solicitado; por ejemplo, el recurso de funciones solo permite operaciones GET.
406 NO ACEPTABLE	El recurso de destino no admite el formato de datos solicitado en la cabecera de Accept o el parámetro accept. Es decir, el cliente ha solicitado la devolución de los datos en un determinado formato, pero el servidor no puede devolver datos en ese formato.
409 CONFLICTO	Indica que se ha detectado un cambio conflictivo durante un intento de modificación de un recurso. El cuerpo de la respuesta contiene más información.
415 TIPO DE MEDIOS NO COMPROMETIDOS	El recurso de destino no admite el formato de datos del cuerpo de la solicitud especificado en la cabecera de Content-Type.
ERROR INTERNO DE SERVIDOR 500	Se ha producido un error interno en el servidor. Esto podría indicar un problema con la solicitud o un problema en el código del lado del servidor.

	Se puede encontrar información acerca del error en el cuerpo de respuesta.