



DOCUMENTACIÓN DE LA API GRAPHQL (ENVOLTORIO) MENDELEY

INDICE

INTRODUCCIÓN	3
AUTENTICACIÓN	3
GUARDAR TOKEN DE ACCESO	3
PRUEBE LAS SOLUCIONES	3
ESPECIFICACIONES GENERALES	3
CAMPOS.....	4
ARGUMENTOS Y VARIABLES.....	4
DOCUMENTOS	4
CAMPOS PRINCIPALES	4
CAMPOS SECUNDARIOS	5
ARGUMENTOS	5
OBTENER DOCUMENTO(S)	6
CREAR UN DOCUMENTO	6
EDITAR UN DOCUMENTO	6
ELIMINAR UN DOCUMENTO	7
CARPETAS	7
CAMPOS.....	7
ARGUMENTOS	7
OBTENER CARPETA(S)	7
CREAR UNA CARPETA	8
EDITAR UNA CARPETA	8
ELIMINAR UNA CARPETA	8
AGREGAR DOCUMENTO A UNA CARPETA	8
ELIMINAR DOCUMENTO DE UNA CARPETA	8
GRUPOS	9
CAMPOS.....	9
ARGUMENTOS	9
OBTENER GRUPO(S).....	9
CREAR UN GRUPO	10
EDITAR UN GRUPO.....	10
ELIMINAR UN GRUPO	10
ARCHIVOS	10
CAMPOS.....	10
ARGUMENTOS	11
OBTENER ARCHIVO(S)	11
ELIMINAR UN ARCHIVO	11
MI PERFIL	11
CAMPOS.....	11
OBTENER MI PERFIL	12

INTRODUCCIÓN

El presente manual de documentación, muestra al envoltorio GraphQL de Mendeley en donde se detalla el uso de cada una de las operaciones de gestión de recursos de Mendeley con ejemplos sencillos. Adicionalmente, los IDEs GraphQL Playground <https://graphql-mendeley.herokuapp.com/playground> y GraphiQL <https://graphql-mendeley.herokuapp.com/graphiql> generan la documentación automáticamente la cual es completa y muy intuitiva.

AUTENTICACIÓN

El proceso de autenticación del envoltorio se lo realiza mediante la página de inicio de sesión de Mendeley. Desde la página de inicio del envoltorio <https://graphql-mendeley.herokuapp.com/> dirigir en PROBAR API, en la siguiente ventana confirme su identidad proporcionando sus credenciales.

Guardar token de acceso

Después de proporcionar sus credenciales será redirigido a una página en donde le pedirá que guarde el token generado en una cookie para poder usarlo en las peticiones.

Pruebe las soluciones

En la misma página de generación de la cookie, se presenta tres opciones para realizar solicitudes a la API. GraphQL Playground y GraphiQL son IDEs en donde puede probar sus peticiones adjuntando la estructura de Query o Mutación. GraphQL Mendeley Prueba de concepto es una implementación en React.js y Apollo Client de la API; una pequeña aplicación en donde se demuestra el concepto de la API.



ESPECIFICACIONES GENERALES

Todos los recursos tienen el argumento **limit** que hace referencia a la cantidad de registros que se espera obtener, este campo va de 1 a 500; este argumento en el envoltorio no se define como una paginación, sin embargo, si desea realizarla, se espera que dicha acción se la haga en el lado del cliente.

Para obtener un recurso en específico se requiere adicionalmente el argumento **id**

Todas las mutaciones tienen como campo de salida ***statusText*** que informa si la mutación de realizo con éxito o hubo algún error.

Campos

Los campos son los atributos de cada recurso los cuales pueden ser de tipos de datos como cadenas, enteros, booleanos, objetos, etc.

```
{
  documents{
    title
    type
    authors{
      first_name
    }
  }
}
```

Argumentos y variables.

Los argumentos son datos de entrada que recibe una mutación o una consulta generalmente para hacer filtrado de información. Las variables son un dato de entrada de una operación de consulta o mutación los cuales reemplazan valores estáticos que se asignan a las operaciones.

```
{
  documents(limit:"10") {
    title
  }
}
```

Argumento "limit"

```
query docs($limit: String){
  documents(limit:$limit) {
    title
  }
}

// Query variables
{
  "limit": "10"
}
```

Variable limit

Los siguientes ejemplos son estructuras de consultas y mutaciones que puede usar en los IDEs anteriormente especificados.

DOCUMENTOS

Campos principales

id	string		Identificador UUID del document, establecido por el servidor al crear el recurso, no modificable
title	string	255	Título del document, es requerido

type	string		Tipo de documento: journal, book, book_section, generic, conference_proceedings, working_paper, report, thesis web_page, magazine_article, statute, patent, hearing, case newspaper_article, computer_program, film, bill television_broadcast, encyclopedia_article.
profile_id	string		Id de Perfil del usuario Mendeley que agrego el documento
group_id	string		Id de grupo al cual el document pertenece.
created	string		Fecha creación del documento, establecido por el servidor.
last_modified	string		Fecha última edición del recurso, establecido por el servidor.
abstract	string	10000	Resumen del documento
source	string	255	Punto de publicación del documento
year	integer		Año de publicación o emisión del document.
authors	array		Lista de autores del document.
identifiers	array	500 each	Identificadores del documento tales como: arxiv, doi, isbn, issn, pmid (PubMed), scopus and ssrn.
keywords	array	50 each	Palabras clave del documento establecidas por el autor del mismo.

Campos secundarios

En este link <https://dev.mendeley.com/methods/#additional-document-attributes>, se especifican los detalles de los campos adicionales.

Argumentos

Nota: El argumento *view* es un dato el cual habilita y deshabilita ciertos campos de un documento, si no especifica este argumento solo podrá obtener valores de los campos principales y algunos campos tendrán el valor de null dependiendo del tipo de vista, en el siguiente link se detalla el valor de view y que campos habilita <https://dev.mendeley.com/methods/#document-views>

limit	string	Cantidad de documentos a obtener.
order	string	Orden de clasificación.
sort	string	Campo criterio para ordenar.
view	string	Tipo de vista
group_id	string	Documentos de un grupo específico.
folder_id	string	Documentos de una carpeta específica.

Obtener documento(s)

```
documents(limit:"100"
          view:"all" order:"desc"){
  id
  title
  type
  abstract
  authors{
    first_name
  }
  year
  files{
    file_name
  }
  read
}
```

```
document(id:"document_id"
         view:"all" order:"desc"){
  id
  title
  type
  abstract
  authors{
    first_name
  }
  year
  files{
    file_name
  }
  read
}
```

Crear un documento

```
mutation{
  createDocument(document:{
    title:"DocumentTest"
    type:"journal"
    authors:{
      last_name:"Hernandez"
      first_name:"Jose"
    }
  }){
    statusText
  }
}
```

Editar un documento

```
mutation{
  updateDocument(id:"id_document" document:{
    title:"DocumentTest_update"
    type:"book"
    authors:{
      last_name:"Hernandez"
      first_name:"Carlos"
    }
  }){
    statusText
  }
}
```

Eliminar un documento

```
mutation{
  deleteDocument (id:"id_document" ){
    statusText
  }
}
```

CARPETAS

Campos

id	string	ID de la carpeta.
name	string	Nombre de la carpeta
parent_id	string	ID de la carpeta padre.
group_id	string	ID del grupo que pertenece la carpeta.
created	string	Fecha de creación de la carpeta, establecido por el servidor.
modified	string	Fecha de última edición de la carpeta, establecido por el servidor
documents	Document	Lista de documentos asociados a la carpeta.

Argumentos

limit	string	Cantidad de carpetas a obtener.
group_id	string	Todas las carpetas del grupo especificado.

Obtener carpeta(s)

```
{
  folders(limit:"100"){
    id
    name
    created
    group_id
    parent_id
    modified
    documents{
      title
    }
  }
}
```

```
{
  folder(id:"id_folder"){
    id
    name
    created
    group_id
    parent_id
    modified
    documents{
      title
    }
  }
}
```

Crear una carpeta

```
mutation{
  createFolder(folder:{
    name: "FolderTest"
    group_id: "id_group"
    parent_id: "id_parent"
  }){
    statusText
  }
}
```

Editar una carpeta

```
mutation{
  updateFolder(id:"id_folder" folder:{
    name: "FolderTest_update"
    group_id: "id_group"
    parent_id: "id_parent"
  }){
    statusText
  }
}
```

Eliminar una carpeta

```
mutation{
  deleteFolder(id:"id_folder")
  statusText
}
}
```

Agregar documento a una carpeta

```
mutation{
  addDocToFolder(idF:"id_folder" idD:"id_document"){
    statusText
  }
}
```

Eliminar documento de una carpeta

```
mutation{
  removeDocToFolder(id:"id_folder" document_id:"id_document"){
    statusText
  }
}
```


GRUPOS

Campos

id	string	Identificador del grupo, establecido por el servidor.
link	string	URL del grupo.
owning_profile_id	string	Id del dueño del grupo.
access_level	string	Tipo de acceso del grupo como public, private o invite_only.
created	string	Fecha de creación del grupo, establecido por el servidor.
name	string	Nombre del grupo.
description	string	Descripción del grupo.
photo	object	Foto del grupo que contiene una URL del recurso
webpage	string	Sitio web del grupo fuera de Mendeley
disciplines	array	Lista de las disciplinas del cual es parte el grupo.
tags	array	Lista de etiquetas del grup.
role	string	Rol del usuario autenticado: owner, admin, normal, follower
member_count	Int	Total de miembros del grupo
member_limit	Int	Máximo de miembros del grupo
url_slug	string	
used_space	Int	
folders	Folder	Carpetas que pertenecen al grupo.

Argumentos

limit	string	Cantidad de grupos a obtener.
--------------	--------	--------------------------------------

Obtener grupo(s)

```
groups (limit:"100"){
  name
  id
  created
  access_level
  description
  member_count
  role
  url_slug
  folders{
    name
  }
  used_space
}
```

```
group(id:"id_group"){
  name
  id
  created
  access_level
  description
  member_count
  role
  url_slug
  folders{
    name
  }
  used_space
}
```

Crear un grupo

```
mutation{
  createGroup(group:{
    name:"GroupTest"
    access_level:"public"
    description:"Public group test"
    webpage:"http:mygroup.com"
  }){
    statusText
  }
}
```

Editar un grupo

```
mutation{
  updateGroup(id:"id_group" group:{
    name:"GroupTest_update"
    access_level:"public"
    description:"Public group test_v2"
    webpage:"http:mygroup.com"
  }){
    statusText
  }
}
```

Eliminar un grupo

```
mutation{
  deleteGroup(id:"id_group"){
    statusText
  }
}
```

ARCHIVOS

Campos

id	string	Identificador único del archivo.
file_name	string	Nombre del archivo, generado de los metadatos del doc enlazado
mime_type	string	Especifica la extensión del archivo
filehash	string	SHA1 hash del archivo. Usado para verificar la integridad del archivo.
document_id	string	Id del documento al cual pertenece el archivo
size	Int	Tamaño en bytes del archivo.

catalog_id	string	
created	string	Fecha de creación del archivo

Argumentos

limit	string	Cantidad de archivos a obtener.
document_id	string	Todos los archivos del documento especificado

Obtener archivo(s)

La obtención de un archivo no obtiene la información del archivo si no que desencadena la descarga del propio archivo

```
{
  files(limit:"50" document_id:"id_document"){
    id
    file_name
    created
    filehash
    size
    document_id
  }
}
```

```
{
  file(id:"id_file"){
    statusText
    urlFile
  }
}
```

Eliminar un archivo

```
mutation{
  deleteFile(id:"id_file"){
    statusText
  }
}
```

MI PERFIL

Campos

id	string	Identificador único del archivo.
file_name	string	Nombre del archivo, generado de los metadatos del doc enlazado
mime_type	string	Especifica la extensión del archivo
filehash	string	SHA1 hash del archivo. Usado para verificar la integridad del archivo.
document_id	string	Id del documento al cual pertenece el archivo
size	Int	Tamaño en bytes del archivo.
catalog_id	string	
created	string	Fecha de creación del archivo

Obtener mi perfil

```
{
  profileMe{
    id
    first_name
    last_name
    display_name
    created
    email
    link
    photos{
      url
    }
  }
}
```