

ApiMarcos

Referencia y funcionalidad

La ApiMarcos está diseñada en arquitectura REST con el fin de mantener estándares y buenas prácticas para su fácil implementación e integración con sistemas que están planteados en arquitectura de microservicios, o simplemente fácil integración con soluciones de FrontEnd. Esta api está diseñada para almacenar y consultar imágenes e información a través de un servicio web basado en el protocolo HTTP.

La Api es una solución específica para una aplicación de marcos, sin embargo es fácilmente adaptable a cualquier otro sistema, también, motivar la mayor cantidad de componentes de integración para ser fácilmente replicada en el número de aplicaciones que requieran el uso de imágenes.

Guías y tutoriales

- Autenticación

Para la autenticación de la Api utiliza Json Web Tokens (JWT) por lo cual se necesitará proveer un token válido para las peticiones restringidas, entre las cuales se encuentran carga de imágenes, consulta de usuarios o creación de eventos.

Ingreso

POST /api/marcos/v1/api/token/

Payload: username, password

Response: access, refresh

Example

POST

▼

/api/marcos/v1/api/token/

Params

Authorization

Headers (8)

Body ●

Pre-request Script

Tests

Settings

● none

● form-data

● x-www-form-urlencoded

● raw

● binary

● GraphQL

JSON ▼

1

{

2

- - - "username" : "username" ,

3

- - - "password" : "user password"

4

}

Respuesta:

```
"refresh": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2t1b190eXB1IjoicmVmcVzaCIsImV4cCI6MTY2NDM5MzU2NiwiYWFOIjoxNjY0MzA3MTY2LCJqdGkiOiJkY2EwZTkzMzQzOTc0MzUxOGY1ODFhZmIyM2ViNDM4NiIsInVzZXJfaWQ1OjF9.M4FSRvAQbxAB49yqQC3Eu18_Sy2UTDC-EYzVBuddzY",
"access": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2t1b190eXB1IjoiyWNjZXNzIiwiaXhwIjoxNjY0MzA3NDY2LCJpYXQ1OjE2NjQzMDCxNjY5Imp0aSI6IjBkYTFlODI2NjNkYTQ5MTk4MDAxN2U0ND8iMjFhYzQ4IiwidXNlc19pZCI6MX0.za6NUSiDtisUUcoHD0em3G40D8DHjckjCe4XhuVpyw0"
```

Refrescar Token

POST /api/marcos/v1/api/token/refresh/

Payload: refresh

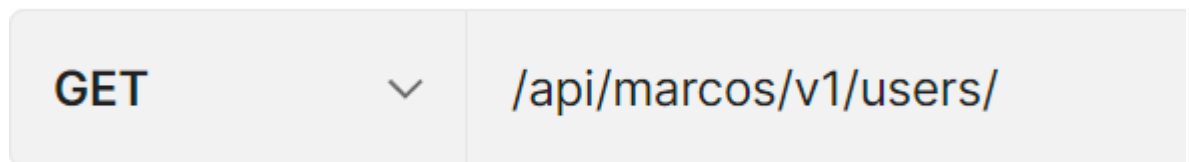
Response: access

$$\vdots$$

GET /api/marcos/v1/users/
GET /api/marcos/v1/users/?page=3

Payload: n/a
Response: count, next, previous, results
Variables: page

Example :



Respuesta:

```
1  {  
2    "count": 1,  
3    "next": null,  
4    "previous": null,  
5    "results": [  
6      {  
7        "url": "http://192.168.128.128:8000/api/marcos/v1/users/1/",  
8        "username": "admin",  
9        "email": "",  
10       "groups": []  
11     }  
12   ]  
13 }
```

Consultar usuario específico

GET /api/marcos/v1/users/<id>/
GET /api/marcos/v1/users/1/

Payload: n/a

Response: url, username, email, groups

Example

:

GET



/api/marcos/v1/users/1

Respuesta:

```
1  {
2    "url": "http://192.168.128.128:8000/api/marcos/v1/users/1/",
3    "username": "admin",
4    "email": "",
5    "groups": []
6  }
```

- Todos los usuarios deben tener asociado un perfil asociado que en este caso se llamará persona.

Consultar personas

Para consultar personas deberá proveer un token válido.

GET /api/marcos/v1/personas/

GET /api/marcos/v1/personas/?page=3

Payload: n/a

Response: count, next, previous, results

Variables: page

MarcosApp / Personas / ListPersonas

GET /api/marcos/v1/personas/

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Headers 6 hidden

KEY	VALUE
<input checked="" type="checkbox"/> Authorization	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2t1b190eXBlljoiYWYWN

Respuesta:

```

1  {
2      "count": 1,
3      "next": null,
4      "previous": null,
5      "results": [
6          {
7              "id": 1,
8              "nombre": "Juan",
9              "identificacion": 10213123123,
10             "email": "bmcastellanos@juan.com",
11             "telefono": 321312312,
12             "user": 1
13         }
14     ]
15 }

```

- Marcos

Esta sección es la razón principal de la creación de la API, esta permite consultar las imágenes sin la necesidad de contar con autenticación. Teniendo en cuenta la seguridad de la App, para el cargue y modificación de imágenes, si es requerida la autenticación.

Este apartado se asocia además con eventos y usuarios, lo que permite una mejor adaptabilidad para seguir un registro y análisis de los datos.

La consulta permite usar paginación, orden y tamaño de las páginas.

```
GET /api/marcos/v1/marcos/  
GET /api/marcos/v1/marcos/?page=3  
GET /api/marcos/v1/marcos/?ordering=evento  
GET /api/marcos/v1/marcos/?page_size=4  
GET /api/marcos/v1/marcos/?ordering=evento&page_size=2&page=8
```

Payload: n/a
Response: count, next, previous, results
Variables: page, ordering, page_size

Example :

GET	▼	/api/marcos/v1/marcos/?ordering=evento&page_size=2
-----	---	--

Respuesta:

```

1  {
2    "count": 9,
3    "next": "http://192.168.128.128:8000/api/marcos/v1/marcos/?ordering=evento&page=2&page_size=2",
4    "previous": null,
5    "results": [
6      {
7        "id": 2,
8        "persona": {
9          "id": 1,
10         "nombre": "Juan"
11       },
12       "nombre": "Marco1",
13       "fecha": "2022-09-22",
14       "evento": {
15         "id": 1,
16         "nombre": "Despedida"
17       },
18       "imagen": "iVBORw0KGgoAAAANSUgAAAYAAAAMgCAYAAADbcAZoAAAAGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbG9z

```

....

```

33   }
34   ]
35 }

```

[gJUubIqsWsHc0leKA3GKlgP3vdgIehX3gWd3Bx74Zh6hXLC340cWJEf4zzsxikHXCnLWCUM+ijqNxffg4wiciagnl07DCshsEAoFAIBAIaakpkIqE4BSeAg5SEcy6BQJkPIVnIJg3Br2sjxbCgKrCrUe4ncWqDvrr55+7Gc49jQEC06QCAQCAQCISC1B4JUqBcgIvyU3U8CgGUj1cn4tN+AU2SP0yMQS1g1Y9oFbHsa4IiEAQEAgEAoEQEIFA4AvEBYIQAQCGWCBowYpWEjHYY8HY/f+B7Bc4E86L0yipEIGNcxP+N0AM+PKzFxdSa7eXHpX+06YVpjsG1VHcvYx6PY01ptN9aHlww+wCgUAQEAgECwfiAREIag1RQQQCgUAQEAgBEQgEQkAEaFAIBAIhIAIBI0HSYIUCAQCgUAQAIAREIGg+igAgEAoFAIBACIhAIIfIP4QAQCGUAQEAgBEQgEAoFAIBAIhIAJB40FSsAQCG](http://192.168.128.128:8000/api/marcos/v1/marcos/2?ordering=evento&page=2&page_size=2)

Consultar marco específico

También se puede consultar un marco en específico contando con el id registrado en la base de datos para cada marco. Esta consulta se puede hacer en la siguiente URL.

GET /api/marcos/v1/marcos/<id>

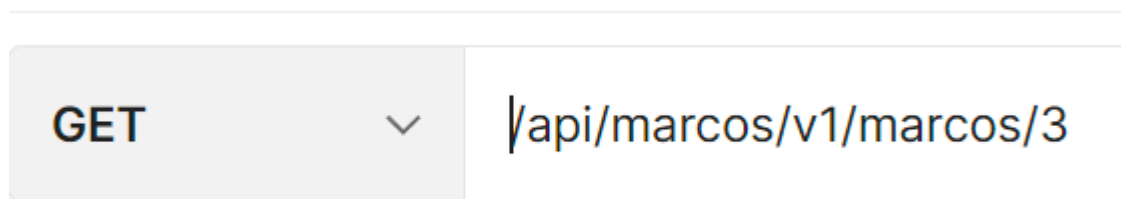
GET /api/marcos/v1/marcos/85

Payload: n/a

Response: id, persona, nombre, fecha, evento, imagen

Variables: n/a

Example :



Respuesta:

```
1  {
2    "id": 3,
3    "persona": {
4      "id": 1,
5      "nombre": "Juan"
6    },
7    "nombre": "Marco1",
8    "fecha": "2022-09-22",
9    "evento": {
10     "id": 1,
11     "nombre": "Despedida"
12   },
13   "imagen": "iVBORw0KGgoAAAANSUHEugAAyAAAMGCAyAAADbcAZoAAAGXRFWHRtb2Z0d2FyZQBBZG9iZSB3bWFnZVJ1YnR5c11PAABwZJJREFUeNrsvQuCw3Wd9/+d3J05dzotnZZ2WqAXv44U2BLKwKuiNlLqut5W6QqjuM+zKxX8cd3F1x/
    Poy5FnRfEC7q6FFhw1wP1CuI0JEK1F6wV6h7Uxbpsy8c83kfvy/vuckMyf35X30cpKcJJ/36/
    V7J2N3J7k709Fv5FW9NiYFEggAAAAAQPExIBFByAmLNgEAAAAAA1kID4AAACBAAAAACgUsSwCQCAAAAEAAAAAABAAAEAAAAAAdU1GMTAAABAgAAAAABQCZD/
    AQAECAAAAAABABxUD4FQAQIAAAAAAFQPhVwBAAAAAAAVayEXuEAAQIAAAAAAUDEQggUABAgAAAAAQEWA9wMACBAAAAAAAgQACBAAAAAAADqD4RfAQABAgAAAAABQMeABAQACBAAAAAAAgQACBAAAAAAAgPoDPUAAgAABAAAAAKgY8TAAAAECAAAAADx
    QAECAAAAAABABgAAAEIAAAAAAAAECAAAAAAAAgAABAAAAAAASqIAFAAQIAAAAAAEFDQ44IABAgAAAAAAAgQAAAAAAAIIAAQAAAAAAAECAAAAAAAACBAAAAAAACBAAAAAAABAgAAAAAAAgQIAAAAAAAAIIEAAAAACAGgI3mmCAY
    FHEwAAgHnxRk0Bd/QEvXSymy8x93zZanfQ02bNo/M6510ezYkNBUC1wRQuABAgAABQT/xm8AA9f0g1evz4PLx/
```

Crear Marco:

Para crear un dato se debe contar con un token vigente ya que es una petición protegida con el fin de asegurar la integridad de la información. La información debe ser enviada como FormData.

POST /api/marcos/v1/marcos/

Payload: nombre, evento, imagen

Response: id, persona, nombre, fecha, evento, imagen

Variables: n/a

Example :

POST

▼

/api/marcos/v1/marcos/

	KEY	VALUE
<input checked="" type="checkbox"/>	nombre	Marco1
<input checked="" type="checkbox"/>	evento	1
<input checked="" type="checkbox"/>	imagen	Imagen_ugc (1).jpg ×

Respuesta:

```
1 {
2   "id": 11,
3   "persona": 1,
4   "nombre": "Marco1",
5   "fecha": "2022-10-03",
6   "evento": 1,
7   "imagen": "http://192.168.128.128:8000/media/marcos/Imagen_ugc_1.jpg"
8 }
```

- Registros

Esta sección de registros permite consignar la información referente al uso de cada uno de los marcos, con el fin de ser utilizados para análisis de datos.

Es posible crear registros sin proveer ningún tipo de autenticación, sin embargo, para consultar, modificar o eliminar registros se deberá proveer autenticación y tener la autorización correspondiente.

La consulta permite usar paginación, orden y tamaño de las páginas.

GET /api/marcos/v1/registros/

GET /api/marcos/v1/registros/?page=3

GET /api/marcos/v1/registros/?ordering=evento

GET /api/marcos/v1/registros/?page_size=4

GET /api/marcos/v1/registros/?ordering=evento&page_size=2&page=8

Payload: n/a

Response: count, next, previous, results

Variables: page, ordering, page_size

Example :

GET	▼	/api/marcos/v1/registros/
-----	---	---------------------------

Respuesta:

```

1  {
2      "count": 2,
3      "next": null,
4      "previous": null,
5      "results": [
6          {
7              "id": 85,
8              "marco": {
9                  "id": 18,
10                 "nombre": "Marco 6",
11                 "fecha": "2022-10-04",
12                 "evento": {
13                     "id": 1,
14                     "nombre": "Evento 1"
15                 }
16             },
17             "fecha": "2022-10-05T10:56:12.150361-05:00"
18         },
19         {
20             "id": 84,
21             "marco": {

```

.....

```
22     "id": 14,  
23     "nombre": "Marco 2",  
24     "fecha": "2022-10-04",  
25     "evento": {  
26         "id": 1,  
27         "nombre": "Evento 1"  
28     }  
29 },  
30     "fecha": "2022-10-04T20:41:00.656393-05:00"  
31 }  
32 ]  
33 }
```

Crear Registro

POST /api/marcos/v1/marcos/

Payload: marco

Response: id, fecha, marc, fecha, evento, imagen

Variables: n/a

Example :

POST	▼	/api/marcos/v1/registros/
------	---	---------------------------

Respuesta:

```
{
```

```
  "id": 86,
```

```
  "fecha": "2022-10-05T11:10:29.364432-05:00",
```

```
  "marco": 21
```

```
}
```