

# Rest API Guide for YouTubeRank

# Table of Contents

Members .....	1
Sign-up .....	1
Example Request .....	1
Request headers .....	1
Request fields .....	1
Example Response .....	1
Response headers .....	2
Response fields .....	2
Login .....	3
Example Request .....	3
Request headers .....	3
Request fields .....	3
Example Response .....	3
Response headers .....	4
Response fields .....	4
Retrieve member information .....	5
Example Request .....	5
Request headers .....	5
Example Response .....	5
Response headers .....	6
Response fields .....	6
Update member information .....	6
Example Request .....	6
Request headers .....	6
Request fields .....	7
Example Response .....	7
Response headers .....	7
Response fields .....	7
Unsubscribe .....	8
Example Request .....	8
Request headers .....	8
Example Response .....	8
Keywords .....	9
Add my keyword .....	9
Example Request .....	9
Request headers .....	9
Request fields .....	9
Example Response .....	9

Response headers .....	10
Response fields.....	10
Retrieve my keyword .....	11
Example Request .....	11
Request headers.....	11
Example Response.....	11
Response headers .....	12
Response fields.....	12
Retrieve all my keywords.....	13
Example Request .....	13
Request headers.....	13
Example Response.....	13
Response headers .....	14
Response fields.....	14
Delete my keyword .....	15
Example Request .....	15
Request headers.....	15
Example Response.....	15
YouTube .....	17
Search youtube.....	17
Example Request .....	17
Request headers.....	17
Request parameters .....	17
Example Response.....	17
Response fields.....	19

# Members

## Sign-up

### Example Request

```
POST /api/members HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
Content-Length: 78
Host: localhost:8080

{
  "email" : "user@email.com",
  "name" : "user",
  "password" : "password"
}
```

### Request headers

Name	Description
Accept	application/json
Content-Type	application/json

### Request fields

Path	Type	Description
email	String	The email address to sign-up that should be unique
name	String	The user name to sign-up
password	String	The password to sign-up

### Example Response

```
HTTP/1.1 201 Created
Location: /api/members/1
Content-Type: application/json
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 410

{
  "id" : 1,
  "email" : "user@email.com",
  "name" : "user",
  "password" : "{bcrypt}$2a$10$/BYb0/LrRuB74FKmxUbiZeVM94eG0/G0.5k8yqIj0LJFRW4IxaZA2",
  "_links" : {
    "profile" : {
      "href" : "/docs/api-guide.html#resources-members-create"
    },
    "self" : {
      "href" : "http://localhost:8080/api/members"
    },
    "login" : {
      "href" : "http://localhost:8080/api/members/login"
    }
  }
}
```

## Response headers

Name	Description
Content-Type	application/json

## Response fields

Path	Type	Description
id	Number	The intrinsic member number issued on the server side
email	String	The unique email of a valid member in the server
name	String	The name of a valid member in the server
password	String	The password of a valid member in the server and it is bcrypt encoded

Path	Type	Description
<code>_links.self.href</code>	String	link to self
<code>_links.profile.href</code>	String	link to profile
<code>_links.login.href</code>	String	link to login

## Login

### Example Request

```
POST /api/members/login HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
Content-Length: 105
Host: localhost:8080

{
  "email" : "user@email.com",
  "password" : "password",
  "accessToken" : null,
  "tokenType" : null
}
```

### Request headers

Name	Description
Accept	application/json
Content-Type	application/json

### Request fields

Path	Type	Description
<code>email</code>	String	The email to login
<code>password</code>	String	The password to login
<code>accessToken</code>	Null	It should be null and will be produced on the server side if the member is a valid member
<code>tokenType</code>	Null	It should be null and will be produced on the server side

### Example Response

```
HTTP/1.1 201 Created
Location: /oauth/token
Content-Type: application/json
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 473

{
  "accessToken" :
  "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyQGVtYWlsLmNvbSIsIm1hdCI6MTU5MTEyMzEzMTIzMTM3fQ.hd3qerfG4ZqPJbsti2x_kGCUmYIio0YlQ-Tahqu99Rc",
  "tokenType" : "Bearer ",
  "_links" : {
    "profile" : {
      "href" : "/docs/api-guide.html#resources-members-login"
    },
    "self" : {
      "href" : "http://localhost:8080/api/members/login"
    },
    "retrieve-myInfo" : {
      "href" : "http://localhost:8080/api/members/my-info/1"
    }
  }
}
```

## Response headers

Name	Description
Content-Type	application/json
Location	The resource is crated

## Response fields

Path	Type	Description
accessToken	String	The accessToken produced by the server when login is successful
tokenType	String	It should be Bearer type in this application
_links.self.href	String	link to self
_links.profile.href	String	link to profile
_links.retrieve-myInfo.href	String	link to retrieve my information

# Retrieve member information

## Example Request

```
GET /api/members/my-info/1 HTTP/1.1
Authorization:
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyQGVTYWlsLmNvbSIsImIhdCI6MTU5MTEyMzE0MiwiZXhwIjoxNjI3MTIzMTQyYfQ.VOWmFUM_KmyEvDU5BJkmfC2fRuBd_mA5C-Fd9Qcgk40
Accept: application/json
Host: localhost:8080
```

## Request headers

Name	Description
Accept	application/json
Authorization	The client should have valid access token produced on the server side

## Example Response

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 420

{
  "email" : "user@email.com",
  "name" : "user",
  "password" : "{bcrypt}$2a$10$5k07XxyRzWkADtTs3PV30.hFMzhGTx7pY89RFHQ4F1S/aoGZKNaNC",
  "_links" : {
    "profile" : {
      "href" : "/docs/api-guide.html#resources-members-myInfo"
    },
    "self" : {
      "href" : "http://localhost:8080/api/members/my-info/1"
    },
    "update-myInfo" : {
      "href" : "http://localhost:8080/api/members/my-info/1"
    }
  }
}
```



## Response headers

Name	Description
Content-Type	application/json

## Response fields

Path	Type	Description
email	String	The email address to retrieve member information
name	String	The member's name to retrieve member information
password	String	The member's bcrypt encoded password
_links.self.href	String	link to self
_links.profile.href	String	link to profile
_links.update-myInfo.href	String	link to update my information

## Update member information

### Example Request

```
PUT /api/members/my-info/1 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Authorization:
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyaGVtYWlsLmNvbSI6ImIldCI6MTU5MTEyMzEzMCwiZXhwIjoxNjI3MTIzMTMwfQ.HnaLCz2u9qYORBzfcAymZHnTGKHdhkWAFMQ-YnEjeLE
Accept: application/json
Content-Length: 35
Host: localhost:8080

{
  "newPassword" : "newPassword"
}
```

## Request headers

Name	Description
Accept	application/json
Authorization	The client should have valid access token produced on the server side

## Request fields

Path	Type	Description
<code>newPassword</code>	<code>String</code>	The member's new password to update

## Example Response

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 347

{
  "id" : 1,
  "email" : "user@email.com",
  "name" : "user",
  "password" : "{bcrypt}$2a$10$0fCH/Z7ZssC2r13AuXYAKuFokXtPtTxdn079ATDawx7Nch3PcAudy",
  "_links" : {
    "self" : {
      "href" : "http://localhost:8080/api/members/my-info/1"
    },
    "profile" : {
      "href" : "/docs/api-guide.html#resources-members-update-myInfo"
    }
  }
}
```

## Response headers

Name	Description
<code>Content-Type</code>	application/json

## Response fields

Path	Type	Description
<code>id</code>	<code>Number</code>	The intrinsic member number issued on the server side
<code>email</code>	<code>String</code>	The unique email of a valid member in the server

Path	Type	Description
name	String	The updated name of a valid member in the server
password	String	The updated password of a valid member in the server and it is bcrypt encoded
_links.self.href	String	link to self
_links.profile.href	String	link to profile

## Unsubscribe

### Example Request

```
DELETE /api/members/1 HTTP/1.1
Accept: application/json
Authorization:
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyQGVTYWlsLmNvbSIsIm1hdCI6MTU5MTEyMzE0MCwiZXhwIjoxNjI3MTIzMTQwfQ.qeqAExzxOYYfXCW8RVfRRT_qVFMXwWw1MzZwHtfJqn0
Host: localhost:8080
```

### Request headers

Name	Description
Accept	application/json
Authorization	The client should have valid access token produced on the server side

### Example Response

```
HTTP/1.1 200 OK
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
```

# Keywords

## Add my keyword

### Example Request

```
POST /api/keywords HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
Authorization:
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyQGVtYWlsLmNvbSIsImIhdCI6MTU5MTEyMzE4OSwiZXhwIjoxNjI3MTIzMTg5fQ.ZqW5klcJlI00XwJgGv03d6yQxzGJ0Kfq6fpnLj0C8nw
Content-Length: 28
Host: localhost:8080

{
  "keyword" : "queendom"
}
```

### Request headers

Name	Description
Accept	application/json
Authorization	The client should have valid access token produced on the server side

### Request fields

Path	Type	Description
keyword	String	The keyword to search in youtube that the user wants to add my keyword

### Example Response

HTTP/1.1 301 Moved Permanently  
Location: /api/youtube?keyword=queendom  
Content-Type: application/json  
X-Content-Type-Options: nosniff  
X-XSS-Protection: 1; mode=block  
Cache-Control: no-cache, no-store, max-age=0, must-revalidate  
Pragma: no-cache  
Expires: 0  
Content-Length: 454

```
{
  "id" : 1,
  "email" : "user@email.com",
  "keyword" : "queendom",
  "_links" : {
    "self" : {
      "href" : "http://localhost:8080/api/keywords"
    },
    "profile" : {
      "href" : "/docs/api-guide.html#resources-keywords-add-my-keyword"
    },
    "delete-my-keyword" : {
      "href" : "http://localhost:8080/api/keywords/queendom"
    },
    "retrieve-my-keyword" : {
      "href" : "http://localhost:8080/api/keywords/queendom"
    }
  }
}
```

## Response headers

Name	Description
Location	The url for redirect to search on YouTube

## Response fields

Path	Type	Description
id	Number	The intrinsic myKeyword id to retrieve
email	String	The email address of an user who added the keyword
keyword	String	The keyword added to loginUser's my-keyword to retrieve
_links.self.href	String	link to self

Path	Type	Description
<code>_links.profile.href</code>	String	link to profile
<code>_links.delete-my-keyword.href</code>	String	link to delete MyKeyword
<code>_links.retrieve-my-keyword.href</code>	String	link to retrieve MyKeyword

## Retrieve my keyword

### Example Request

```
GET /api/keywords/queendom HTTP/1.1
Accept: application/json
Authorization:
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyQGVtYWlsLmNvbSIsImIhdCI6MTU5MTEyMzE4NiwiZXhwIjoxNjI3MTIzMTg2fQ.fd7DAHBGczxEgVEcUAOIYJJtRm9C7a_U_PE-PFkSaOM
Host: localhost:8080
```

### Request headers

Name	Description
<code>Accept</code>	application/json
<code>Authorization</code>	The client should have valid access token produced on the server side

### Example Response

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 370

{
  "id" : 1,
  "email" : "user@email.com",
  "keyword" : "queendom",
  "_links" : {
    "self" : {
      "href" : "http://localhost:8080/api/keywords/queendom"
    },
    "profile" : {
      "href" : "/docs/api-guide.html#resources-keywords-retrieve-my-keyword"
    },
    "delete-my-keyword" : {
      "href" : "http://localhost:8080/api/keywords/queendom"
    }
  }
}
```

## Response headers

Name	Description
Content-Type	application/json

## Response fields

Path	Type	Description
id	Number	The intrinsic myKeyword id to retrieve
email	String	The email address of an user who added the keyword
keyword	String	The keyword added to loginUser's my-keyword to retrieve
_links.self.href	String	link to self
_links.profile.href	String	link to profile
_links.delete-my-keyword.href	String	link to delete MyKeyword

# Retrieve all my keywords

## Example Request

```
GET /api/keywords HTTP/1.1
Accept: application/json
Authorization:
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyQGVtYWlsLmNvbSIsIm1hdCI6MTU5MTEyMzE4MywiZXhwIjoxNjI3MTIzMTgzfQ.n1snHmOKlUZ1A1pdCNpq9UYw5iDCL-7RFbfkV1ZmBBY
Host: localhost:8080
```

## Request headers

Name	Description
Accept	application/json
Authorization	The client should have valid access token produced on the server side

## Example Response



```
HTTP/1.1 200 OK
Content-Type: application/json
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 458

{
  "myKeywords" : [ {
    "id" : 1,
    "email" : "user@email.com",
    "keyword" : "queendom"
  }, {
    "id" : 2,
    "email" : "user@email.com",
    "keyword" : "(g)idle"
  }, {
    "id" : 3,
    "email" : "user@email.com",
    "keyword" : "GOT7"
  } ],
  "_links" : {
    "self" : {
      "href" : "http://localhost:8080/api/keywords"
    },
    "profile" : {
      "href" : "/docs/api-guide.html#resources-keywords-retrieve-all-my-keywords"
    }
  }
}
```

## Response headers

Name	Description
Content-Type	application/json

## Response fields

Path	Type	Description
myKeywords[].id	Number	The intrinsic myKeyword id to retrieve
myKeywords[].email	String	The email address of an user who added the keyword

Path	Type	Description
<code>myKeywords[].keyword</code>	<code>String</code>	The keyword added to loginUser's my-keyword to retrieve
<code>_links.self.href</code>	<code>String</code>	link to self
<code>_links.profile.href</code>	<code>String</code>	link to profile

## Delete my keyword

### Example Request

```
DELETE /api/keywords/queendom HTTP/1.1
Authorization:
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyQGVTYWlsLmNvbSIsImIhdCI6MTU5MTEyMzE3NywiZXhwIjoxNjI3MTIzMTc3fQ.FD2cPymVoYK8Recdswxz06rD4pK3NqCq6w-4yOM0KRQ
Accept: application/json
Host: localhost:8080
```

### Request headers

Name	Description
<code>Accept</code>	application/json
<code>Authorization</code>	The client should have valid access token produced on the server side

### Example Response

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 257

{
  "id" : null,
  "email" : null,
  "keyword" : null,
  "_links" : {
    "self" : {
      "href" : "http://localhost:8080/api/keywords/queendom"
    },
    "profile" : {
      "href" : "/docs/api-guide.html#resources-keywords-delete-my-keyword"
    }
  }
}
```

# YouTube

## Search youtube

### Example Request

```
GET /api/youtube?keyword=Queendom HTTP/1.1
Authorization:
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyQGVtYWlsLmNvbSIsImIhdCI6MTU5MTEyMzExNiwiZXhwIjoxNjI3MTIzMTE2fQ.G6lq490SaLYzzdKBeHHMIxfXZSLApUPmUzRkkQBInxA
Accept: application/json
Host: localhost:8080
```

### Request headers

Name	Description
Accept	application/json

### Request parameters

Parameter	Description
keyword	A keyword to search on YouTube that a user typed

### Example Response

```

HTTP/1.1 200 OK
Content-Type: application/json
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 2608

[ {
  "videoId" : "0g_X5gTxF0",
  "url" : "https://www.youtube.com/watch?v=0g_X5gTxF0",
  "title" : "[ENG sub] [4]  Fire - ( ) @2  :  4",
  "thumbnailUrl" : "https://i.ytimg.com/vi/0g_X5gTxF0/hqdefault.jpg",
  "description" : "[4]  Fire - ( ) @2  :  K-POP   Mnet
' (Queendom)' -----...",
  "viewCount" : 21804641
}, {
  "videoId" : "JXC7rtAV0zM",
  "url" : "https://www.youtube.com/watch?v=JXC7rtAV0zM",
  "title" : "[ENG sub] [3]  (Egotistic) - AOA @2  :  3",
  "thumbnailUrl" : "https://i.ytimg.com/vi/JXC7rtAV0zM/hqdefault.jpg",
  "description" : "[3]  (Egotistic) - AOA @2  K-POP   Mnet
' (Queendom)' -----...",
  "viewCount" : 19234333
}, {
  "videoId" : "2KtFPjSp3og",
  "url" : "https://www.youtube.com/watch?v=2KtFPjSp3og",
  "title" : "[ENG sub] [ ]  LION - ( ) @ FINAL  :  10",
  "thumbnailUrl" : "https://i.ytimg.com/vi/2KtFPjSp3og/hqdefault.jpg",
  "description" : "[ ]  LION - ( ) @ FINAL  K-POP   Mnet
' (Queendom)' -----...",
  "viewCount" : 17694213
}, {
  "videoId" : "Knz0j_G7lKg",
  "url" : "https://www.youtube.com/watch?v=Knz0j_G7lKg",
  "title" : "[ENG sub] [2]  LATATA - ( ) @1  :  2",
  "thumbnailUrl" : "https://i.ytimg.com/vi/Knz0j_G7lKg/hqdefault.jpg",
  "description" : "[2]  LATATA - ( ) @1  K-POP   Mnet
' (Queendom)' -----...",
  "viewCount" : 15764132
}, {
  "videoId" : "Qx0r7l7utUo",
  "url" : "https://www.youtube.com/watch?v=Qx0r7l7utUo",
  "title" : "[ENG sub] [3]  ( ) feat. -  @2  :  3",
  "thumbnailUrl" : "https://i.ytimg.com/vi/Qx0r7l7utUo/hqdefault.jpg",
  "description" : "[3]  ( ) feat. -  @2  K-POP   Mnet
' (Queendom)' -----...",
  "viewCount" : 12764130
} ]

```

## Response fields

Path	Type	Description
<code>[].videoId</code>	String	An intrinsic id of video issued by YouTube
<code>[].title</code>	String	A title of the YouTube video written by the uploader
<code>[].url</code>	String	The URL to play the video on YouTube
<code>[].description</code>	String	The description for the video written by the uploader
<code>[].thumbnailUrl</code>	String	The URL to access the thumbnail image of the video in high quality
<code>[].viewCount</code>	Number	The view counts of the video at the time of request