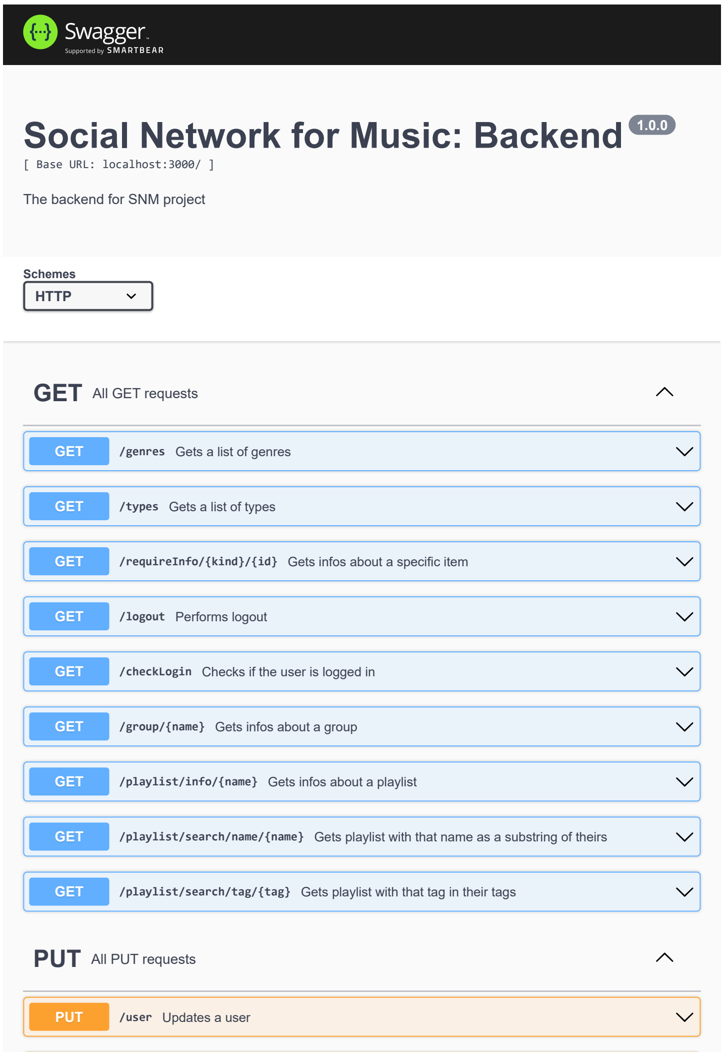


Swagger Supported by SMARTBEAR

Social Network for Music: Backend 1.0.0

[Base URL: localhost:3000/]

The backend for SNM project

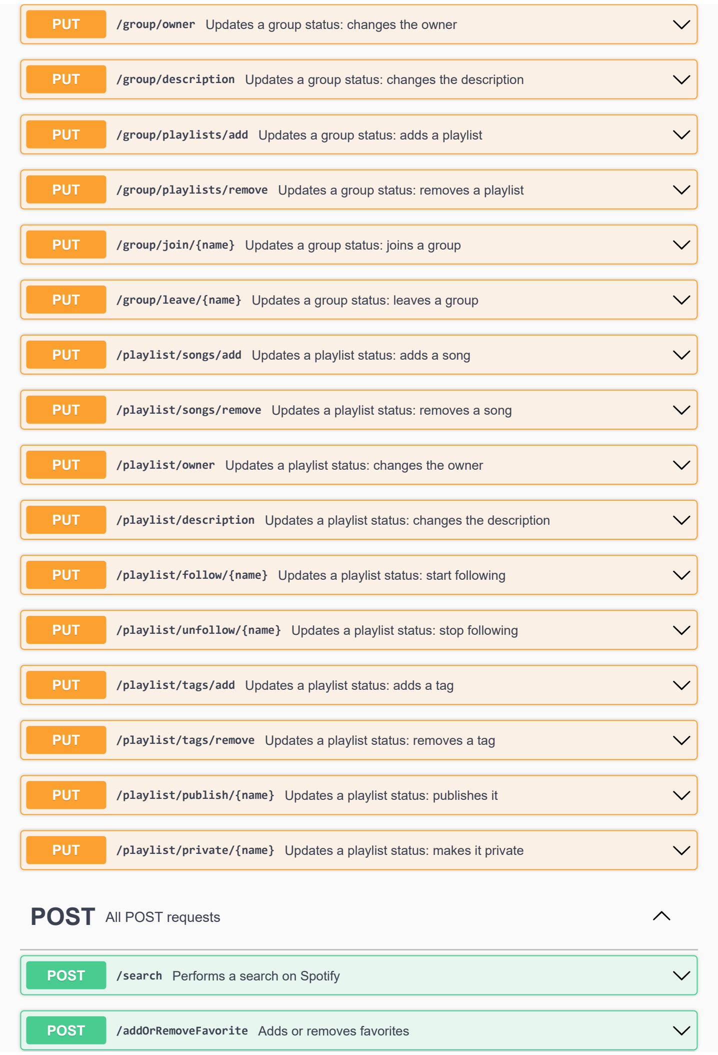
Schemes
HTTP

GET All GET requests

- GET `/genres` Gets a list of genres
- GET `/types` Gets a list of types
- GET `/requireInfo/{kind}/{id}` Gets infos about a specific item
- GET `/logout` Performs logout
- GET `/checkLogin` Checks if the user is logged in
- GET `/group/{name}` Gets infos about a group
- GET `/playlist/info/{name}` Gets infos about a playlist
- GET `/playlist/search/name/{name}` Gets playlist with that name as a substring of theirs
- GET `/playlist/search/tag/{tag}` Gets playlist with that tag in their tags

PUT All PUT requests

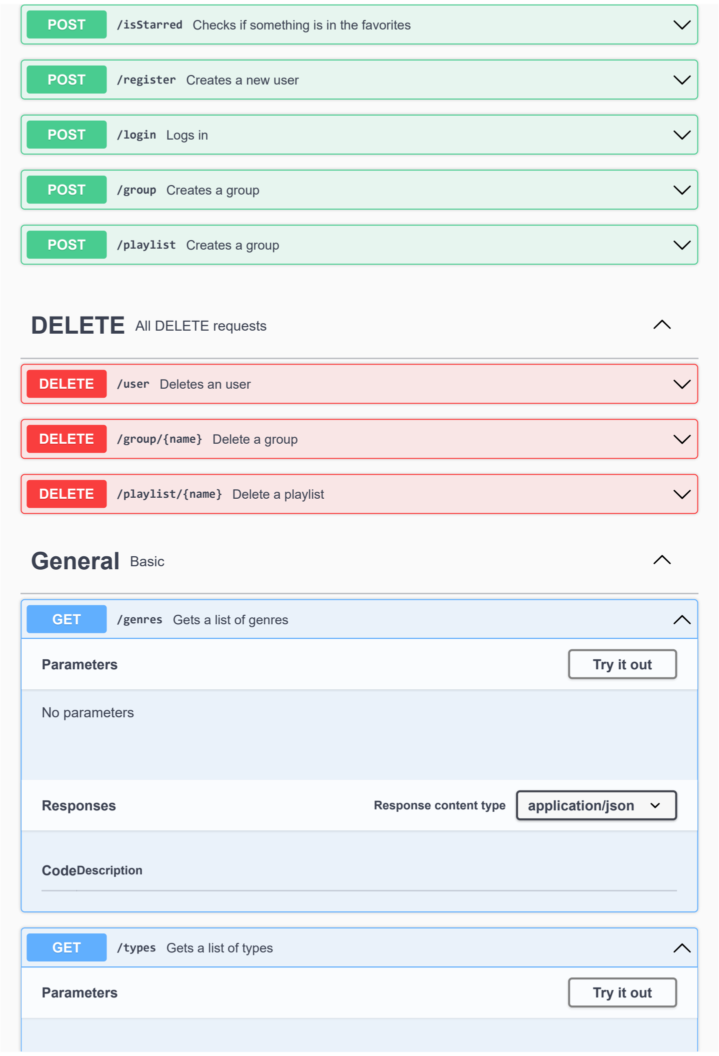
- PUT `/user` Updates a user



- PUT `/group/owner` Updates a group status: changes the owner
- PUT `/group/description` Updates a group status: changes the description
- PUT `/group/playlists/add` Updates a group status: adds a playlist
- PUT `/group/playlists/remove` Updates a group status: removes a playlist
- PUT `/group/join/{name}` Updates a group status: joins a group
- PUT `/group/leave/{name}` Updates a group status: leaves a group
- PUT `/playlist/songs/add` Updates a playlist status: adds a song
- PUT `/playlist/songs/remove` Updates a playlist status: removes a song
- PUT `/playlist/owner` Updates a playlist status: changes the owner
- PUT `/playlist/description` Updates a playlist status: changes the description
- PUT `/playlist/follow/{name}` Updates a playlist status: start following
- PUT `/playlist/unfollow/{name}` Updates a playlist status: stop following
- PUT `/playlist/tags/add` Updates a playlist status: adds a tag
- PUT `/playlist/tags/remove` Updates a playlist status: removes a tag
- PUT `/playlist/publish/{name}` Updates a playlist status: publishes it
- PUT `/playlist/private/{name}` Updates a playlist status: makes it private

POST All POST requests

- POST `/search` Performs a search on Spotify
- POST `/addOrRemoveFavorite` Adds or removes favorites



- POST `/isStarred` Checks if something is in the favorites
- POST `/register` Creates a new user
- POST `/login` Logs in
- POST `/group` Creates a group
- POST `/playlist` Creates a group

DELETE All DELETE requests

- DELETE `/user` Deletes an user
- DELETE `/group/{name}` Delete a group
- DELETE `/playlist/{name}` Delete a playlist

General Basic

GET `/genres` Gets a list of genres

Parameters Try it out

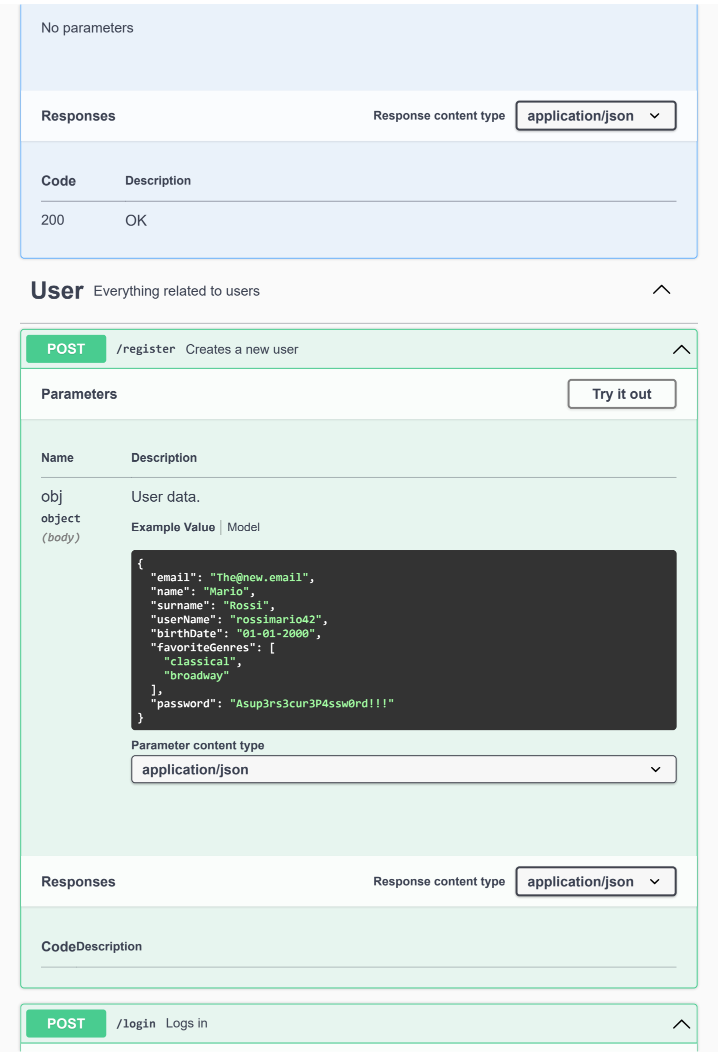
No parameters

Responses Response content type application/json

CodeDescription

GET `/types` Gets a list of types

Parameters Try it out



No parameters

Responses Response content type application/json

Code	Description
200	OK

User Everything related to users

POST `/register` Creates a new user

Parameters Try it out

Name	Description
obj object *(body)*	User data.

Example Value | Model

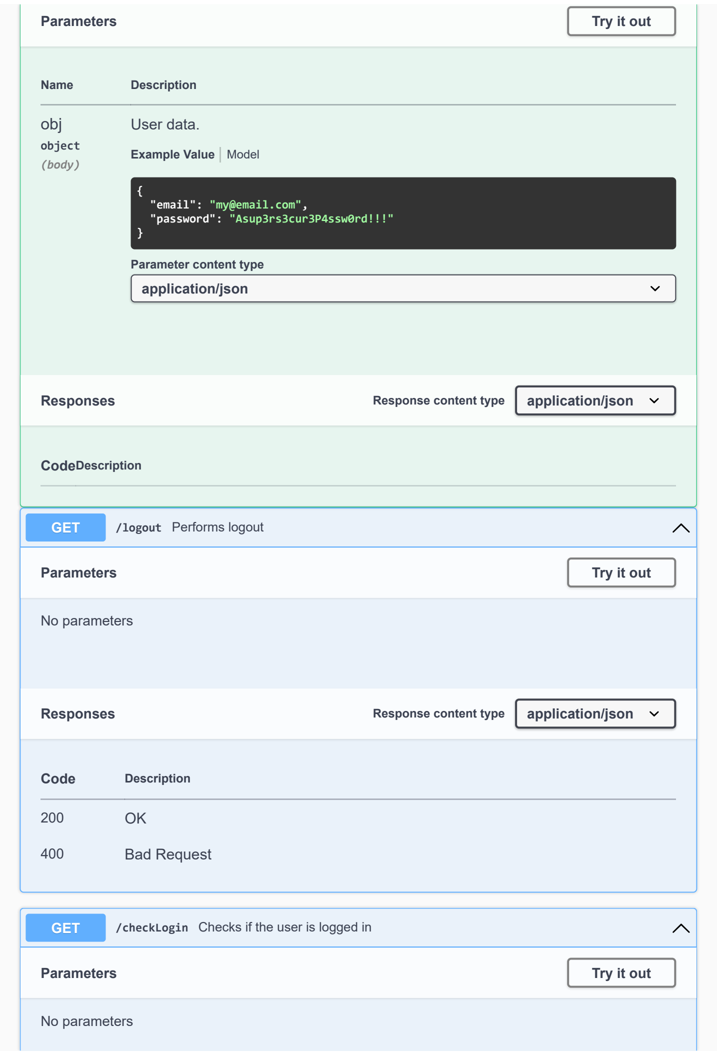
```
{
  "email": "The@new.email",
  "name": "Mario",
  "surname": "Rossi",
  "userName": "rossimario42",
  "birthDate": "01-01-2000",
  "favoriteGenres": [
    "classical",
    "broadway"
  ],
  "password": "Asup3rs3cur3P4ssw0rd!!!"
}
```

Parameter content type
application/json

Responses Response content type application/json

CodeDescription

POST `/login` Logs in



Name	Description
obj object *(body)*	User data.

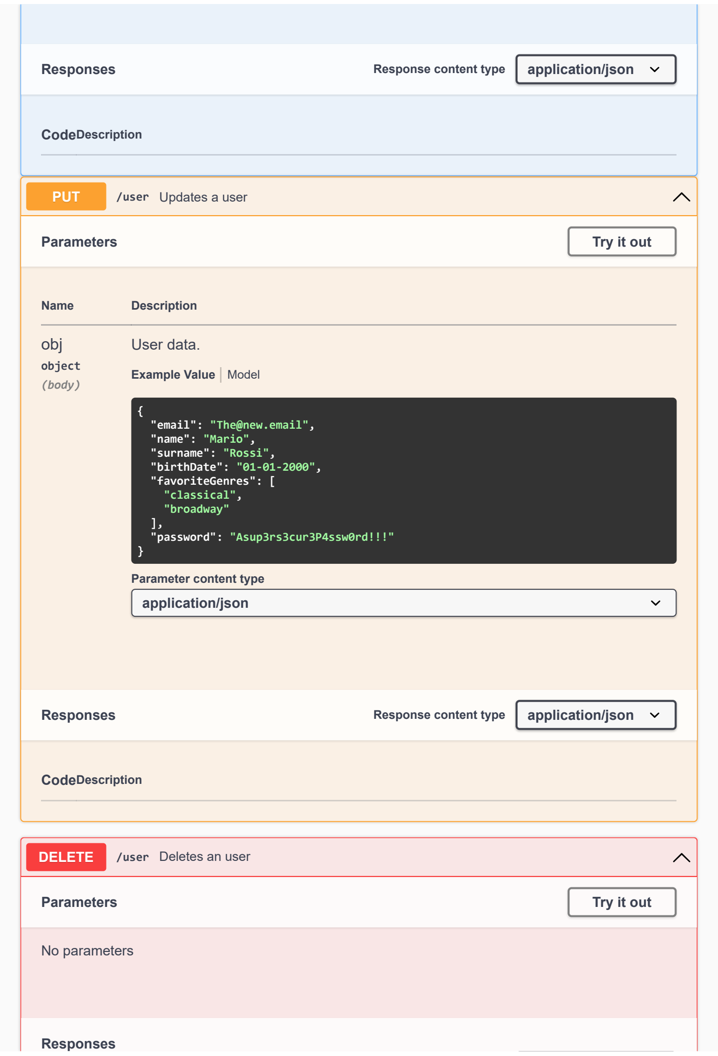
```
{
  "email": "my@email.com",
  "password": "Asup3rs3cur3P4ssw0rd!!!"
}
```

Parameter content type: application/json

No parameters

Code	Description
200	OK
400	Bad Request

No parameters

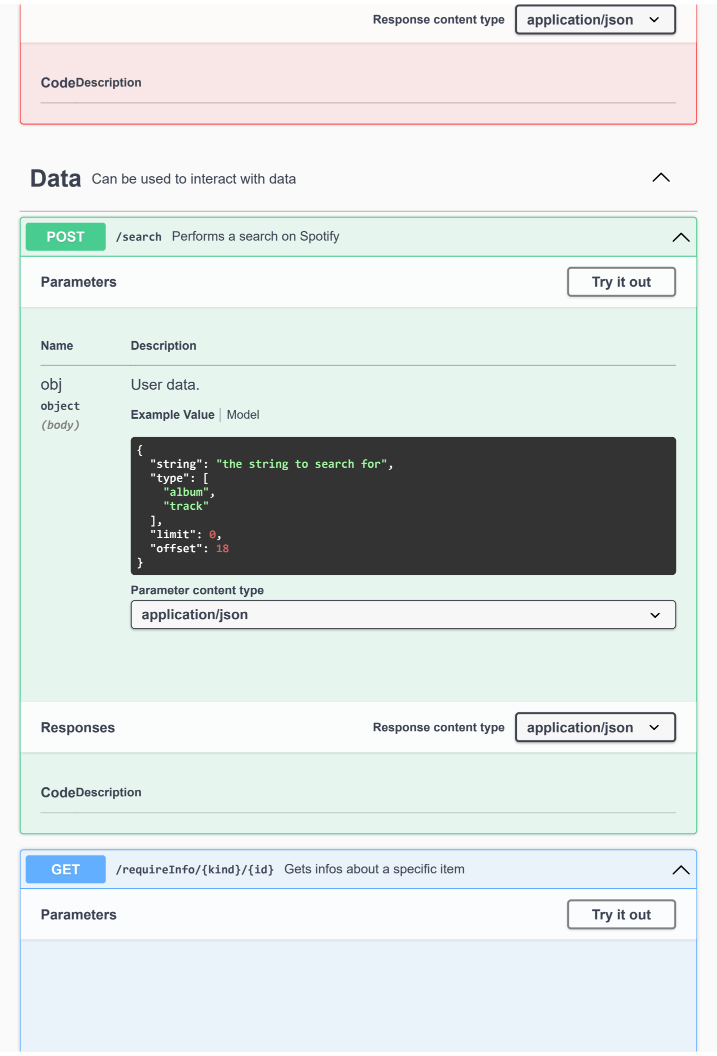


Name	Description
obj object *(body)*	User data.

```
{
  "email": "The@new.email",
  "name": "Mario",
  "surname": "Rossi",
  "birthDate": "01-01-2000",
  "favoriteGenres": [
    "classical",
    "broadway"
  ],
  "password": "Asup3rs3cur3P4ssw0rd!!!"
}
```

Parameter content type: application/json

No parameters



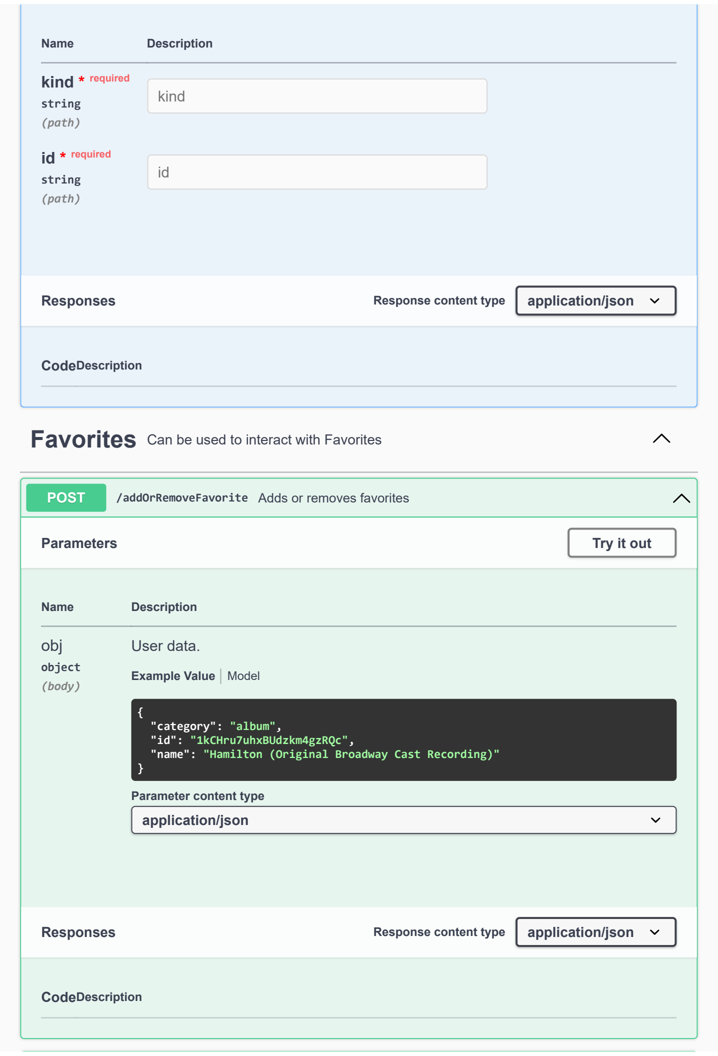
Data

Can be used to interact with data

Name	Description
obj object *(body)*	User data.

```
{
  "string": "the string to search for",
  "type": [
    "album",
    "track"
  ],
  "limit": 0,
  "offset": 18
}
```

Parameter content type: application/json



Name	Description
kind * required string *(path)*	kind
id * required string *(path)*	id

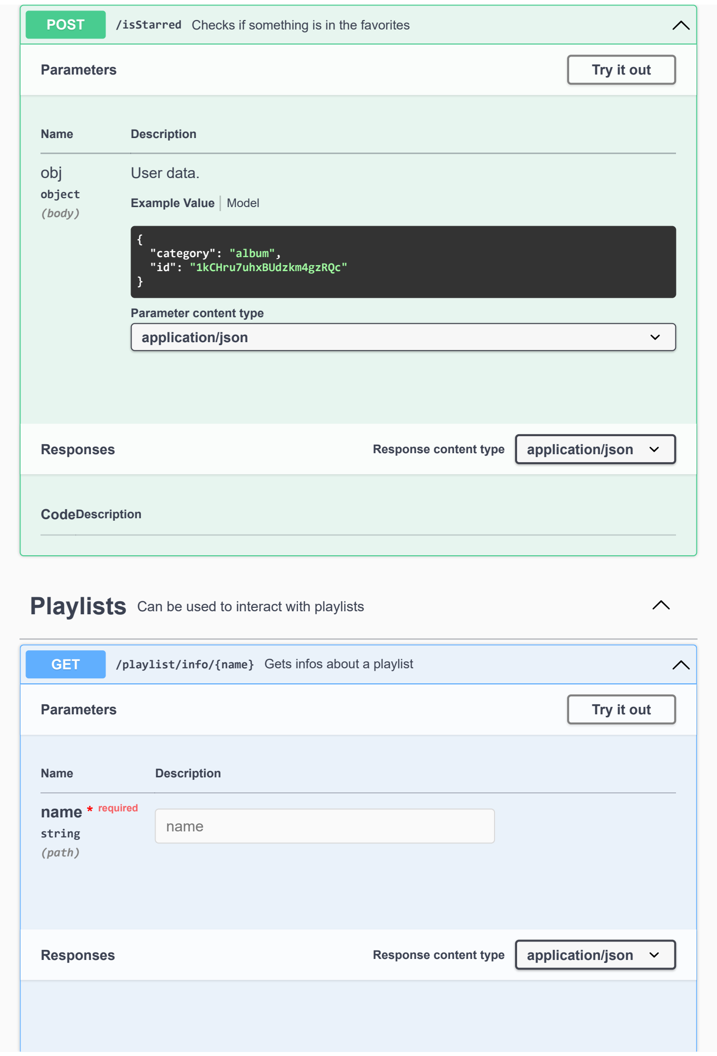
Favorites

Can be used to interact with Favorites

Name	Description
obj object *(body)*	User data.

```
{
  "category": "album",
  "id": "1kCHru7uhxBUdzkm4gzRQc",
  "name": "Hamilton (Original Broadway Cast Recording)"
}
```

Parameter content type: application/json



POST /isStarred Checks if something is in the favorites

Parameters

Try it out

Name	Description
obj object *(body)*	User data.

Example Value | Model

```
{
  "category": "album",
  "id": "1kCHru7uhxBUdzkm4gzRQc"
}
```

Parameter content type

application/json

Responses

Response content type application/json

CodeDescription

Playlists

Can be used to interact with playlists

GET /playlist/info/{name} Gets infos about a playlist

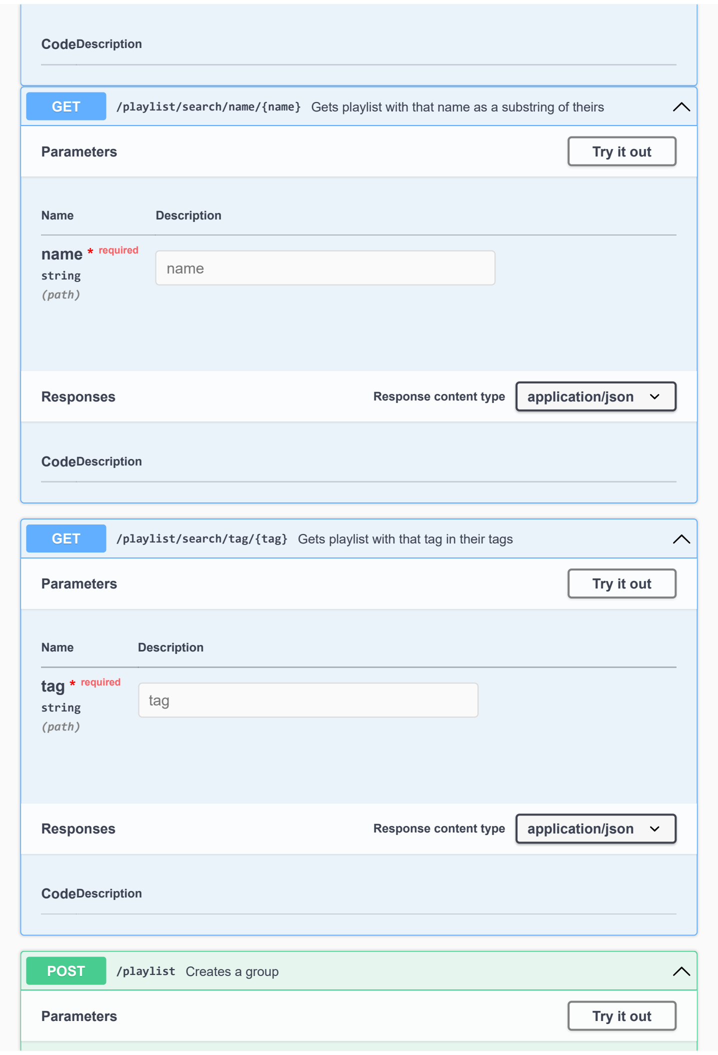
Parameters

Try it out

Name	Description
name * required string *(path)*	name

Responses

Response content type application/json



CodeDescription

GET /playlist/search/name/{name} Gets playlist with that name as a substring of theirs

Parameters

Try it out

Name	Description
name * required string *(path)*	name

Responses

Response content type application/json

CodeDescription

GET /playlist/search/tag/{tag} Gets playlist with that tag in their tags

Parameters

Try it out

Name	Description
tag * required string *(path)*	tag

Responses

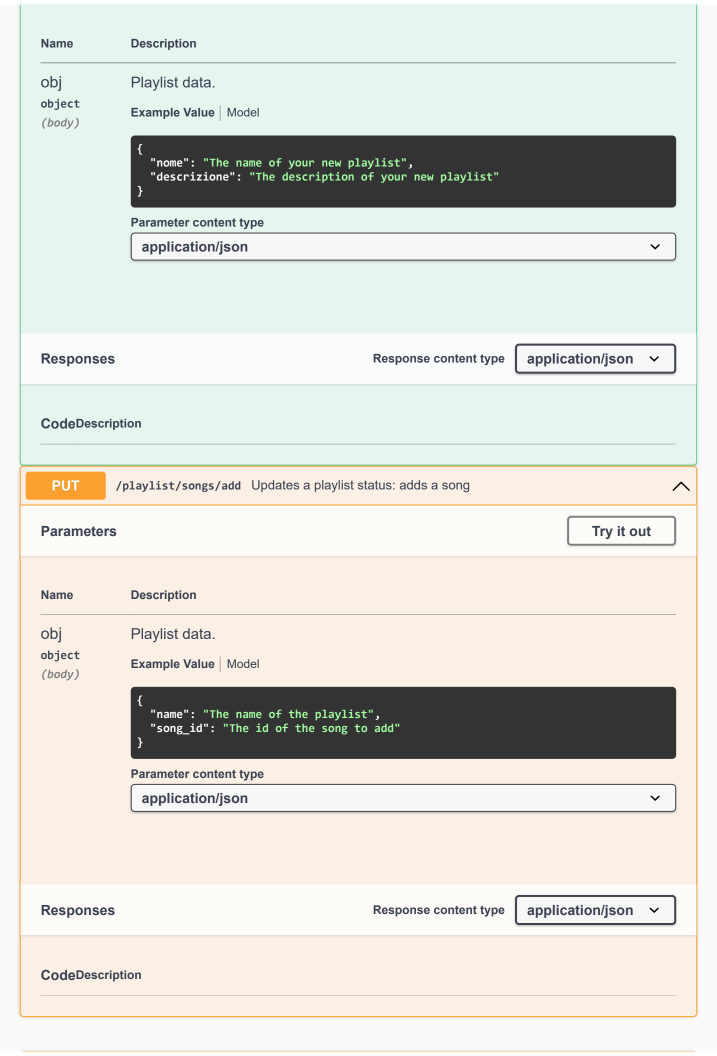
Response content type application/json

CodeDescription

POST /playlist Creates a group

Parameters

Try it out



Name	Description
obj object *(body)*	Playlist data.

Example Value | Model

```
{
  "nome": "The name of your new playlist",
  "descrizione": "The description of your new playlist"
}
```

Parameter content type

application/json

Responses

Response content type application/json

CodeDescription

PUT /playlist/songs/add Updates a playlist status: adds a song

Parameters

Try it out

Name	Description
obj object *(body)*	Playlist data.

Example Value | Model

```
{
  "name": "The name of the playlist",
  "song_id": "The id of the song to add"
}
```

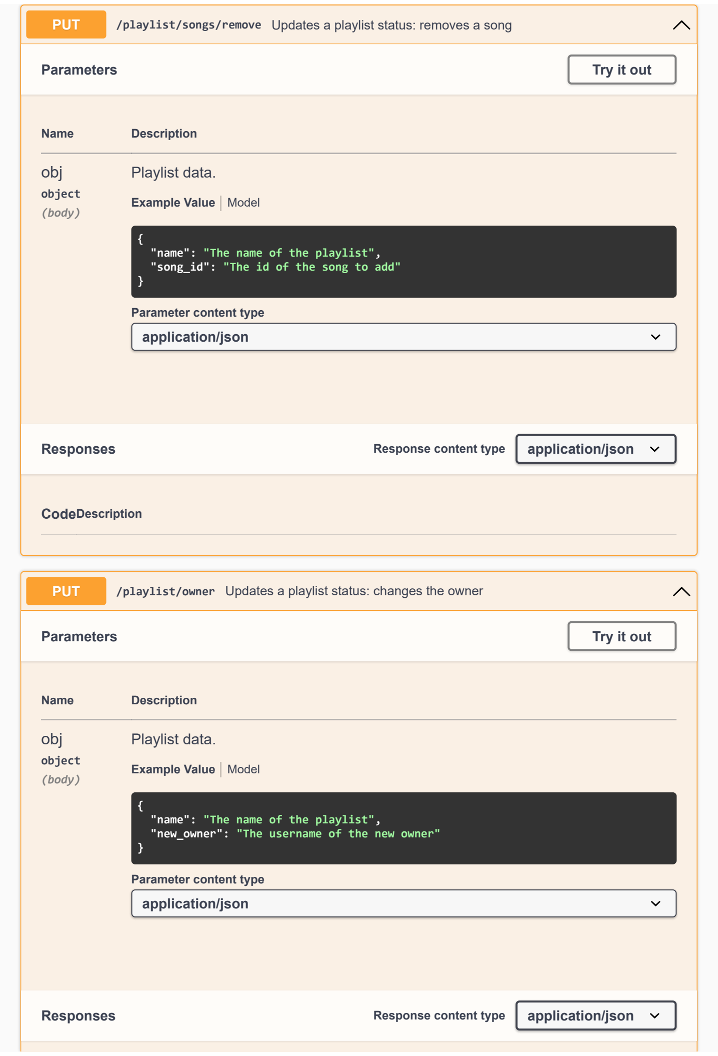
Parameter content type

application/json

Responses

Response content type application/json

CodeDescription



PUT /playlist/songs/remove Updates a playlist status: removes a song

Parameters

Try it out

Name	Description
obj object *(body)*	Playlist data.

Example Value | Model

```
{
  "name": "The name of the playlist",
  "song_id": "The id of the song to add"
}
```

Parameter content type

application/json

Responses

Response content type application/json

CodeDescription

PUT /playlist/owner Updates a playlist status: changes the owner

Parameters

Try it out

Name	Description
obj object *(body)*	Playlist data.

Example Value | Model

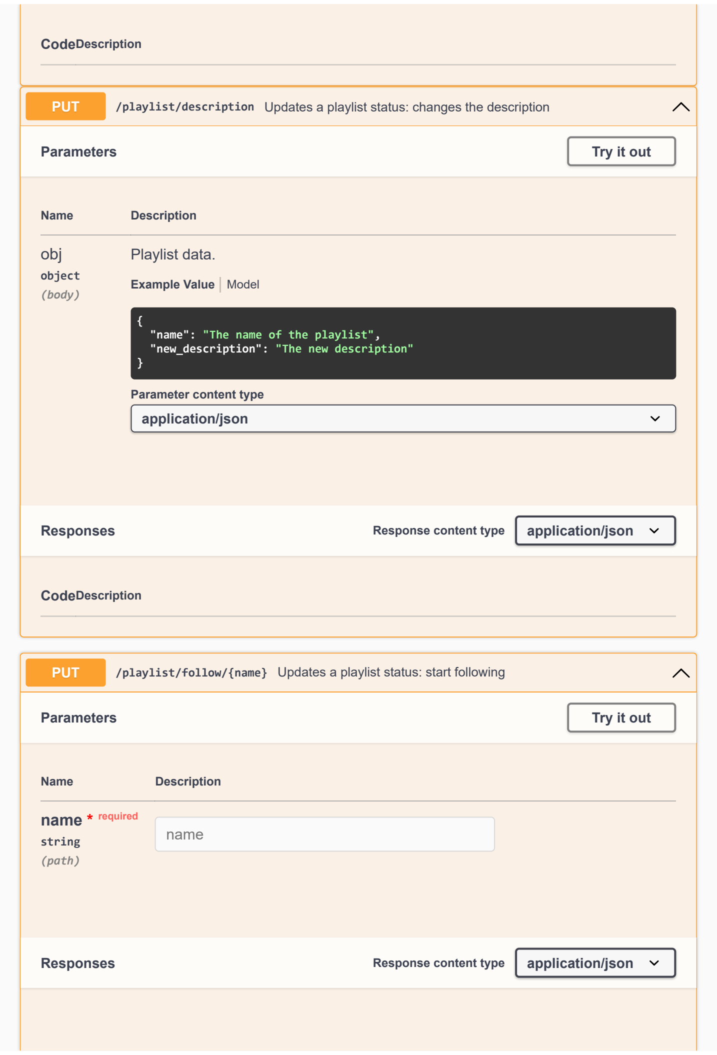
```
{
  "name": "The name of the playlist",
  "new_owner": "The username of the new owner"
}
```

Parameter content type

application/json

Responses

Response content type application/json



CodeDescription

PUT /playlist/description Updates a playlist status: changes the description

Parameters Try it out

Name	Description
obj object *(body)*	Playlist data.

Example Value | Model

```
{
  "name": "The name of the playlist",
  "new_description": "The new description"
}
```

Parameter content type

application/json

Responses Response content type application/json

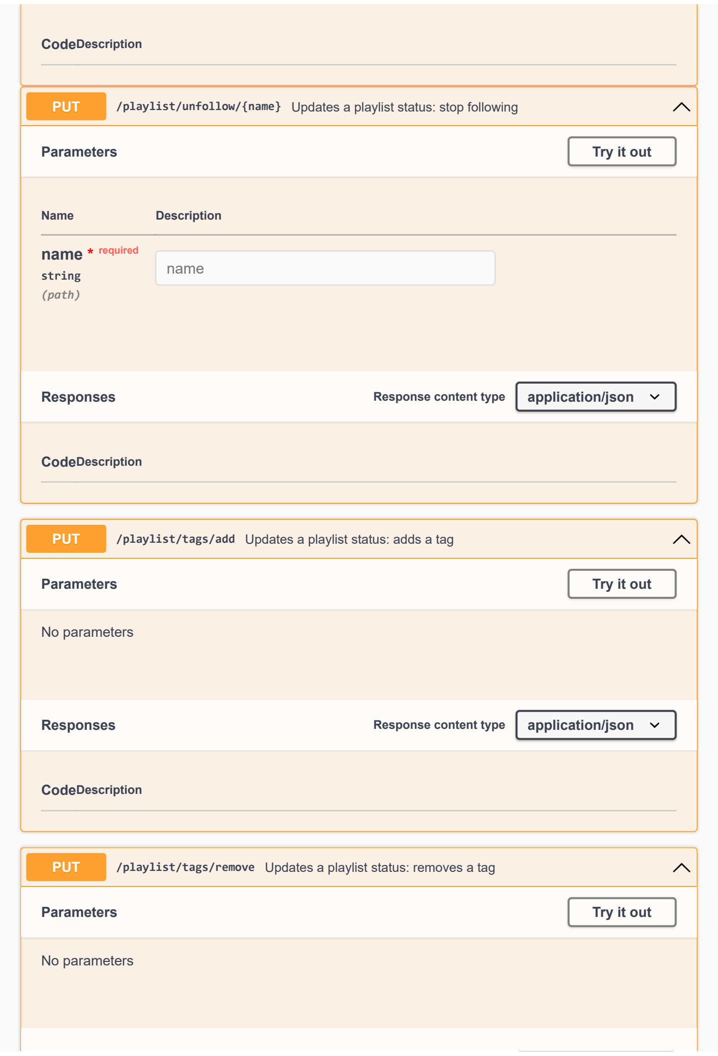
CodeDescription

PUT /playlist/follow/{name} Updates a playlist status: start following

Parameters Try it out

Name	Description
name * required string *(path)*	name

Responses Response content type application/json



CodeDescription

PUT /playlist/unfollow/{name} Updates a playlist status: stop following

Parameters Try it out

Name	Description
name * required string *(path)*	name

Responses Response content type application/json

CodeDescription

PUT /playlist/tags/add Updates a playlist status: adds a tag

Parameters Try it out

No parameters

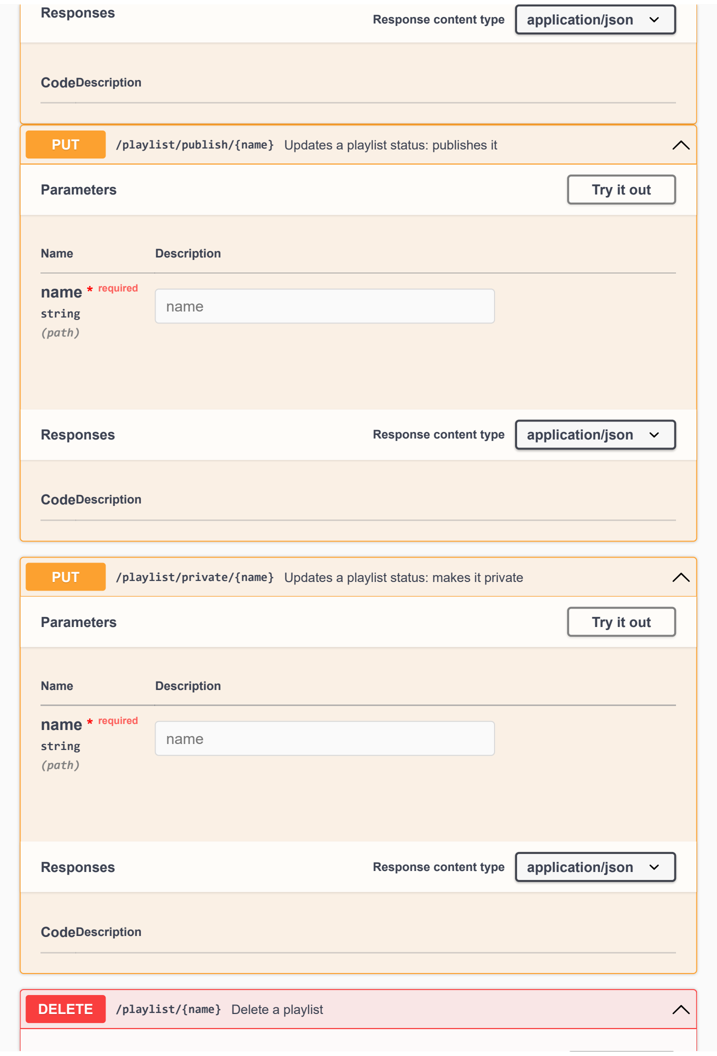
Responses Response content type application/json

CodeDescription

PUT /playlist/tags/remove Updates a playlist status: removes a tag

Parameters Try it out

No parameters



Responses Response content type application/json

CodeDescription

PUT /playlist/publish/{name} Updates a playlist status: publishes it

Parameters Try it out

Name	Description
name * required string *(path)*	name

Responses Response content type application/json

CodeDescription

PUT /playlist/private/{name} Updates a playlist status: makes it private

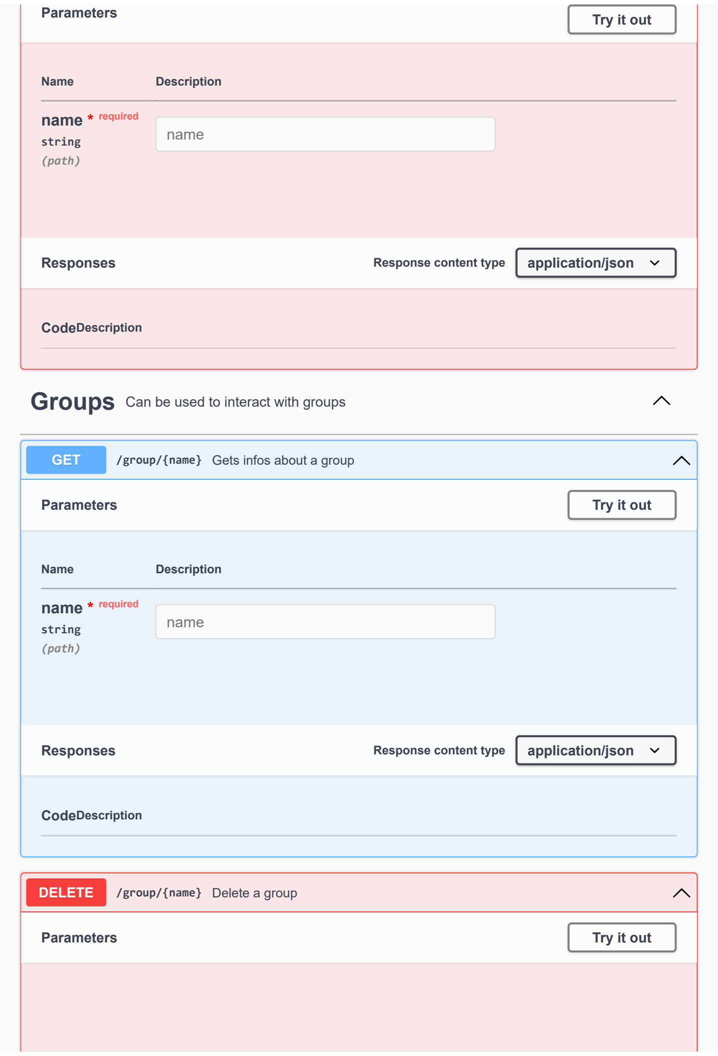
Parameters Try it out

Name	Description
name * required string *(path)*	name

Responses Response content type application/json

CodeDescription

DELETE /playlist/{name} Delete a playlist



Parameters Try it out

Name	Description
name * required string *(path)*	name

Responses Response content type application/json

CodeDescription

Groups Can be used to interact with groups

GET /group/{name} Gets infos about a group

Parameters Try it out

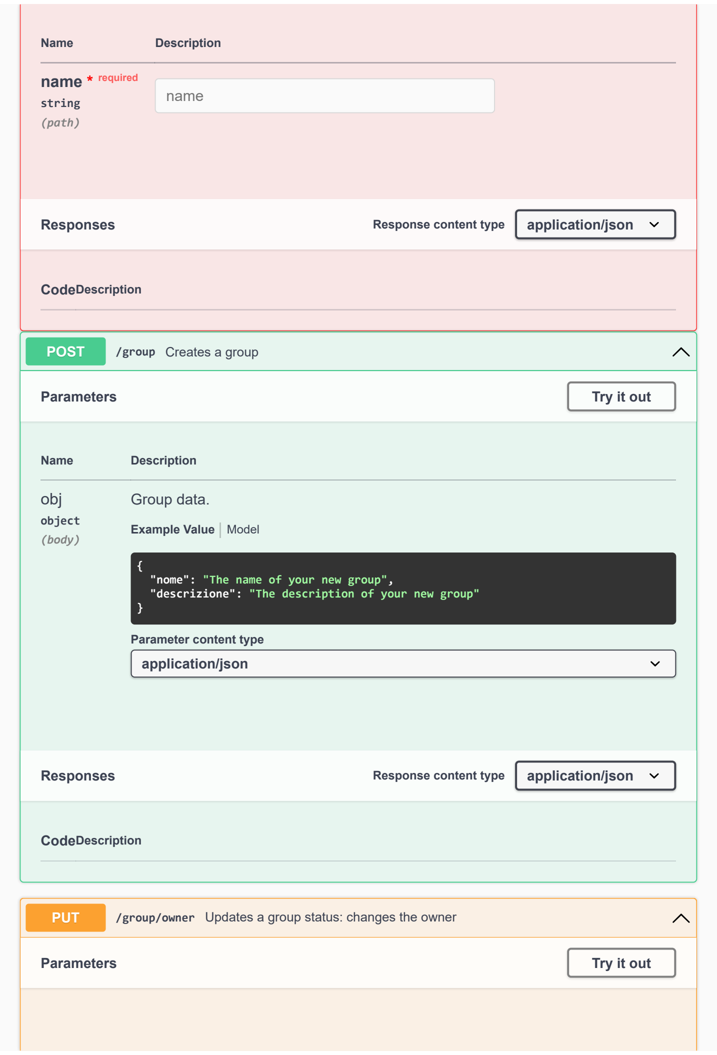
Name	Description
name * required string *(path)*	name

Responses Response content type application/json

CodeDescription

DELETE /group/{name} Delete a group

Parameters Try it out



Name Description

name * required
string
(path)

Responses Response content type application/json

CodeDescription

POST /group Creates a group

Parameters

Name Description

obj
object
(body)

Group data.

Example Value | Model

```
{
  "nome": "The name of your new group",
  "descrizione": "The description of your new group"
}
```

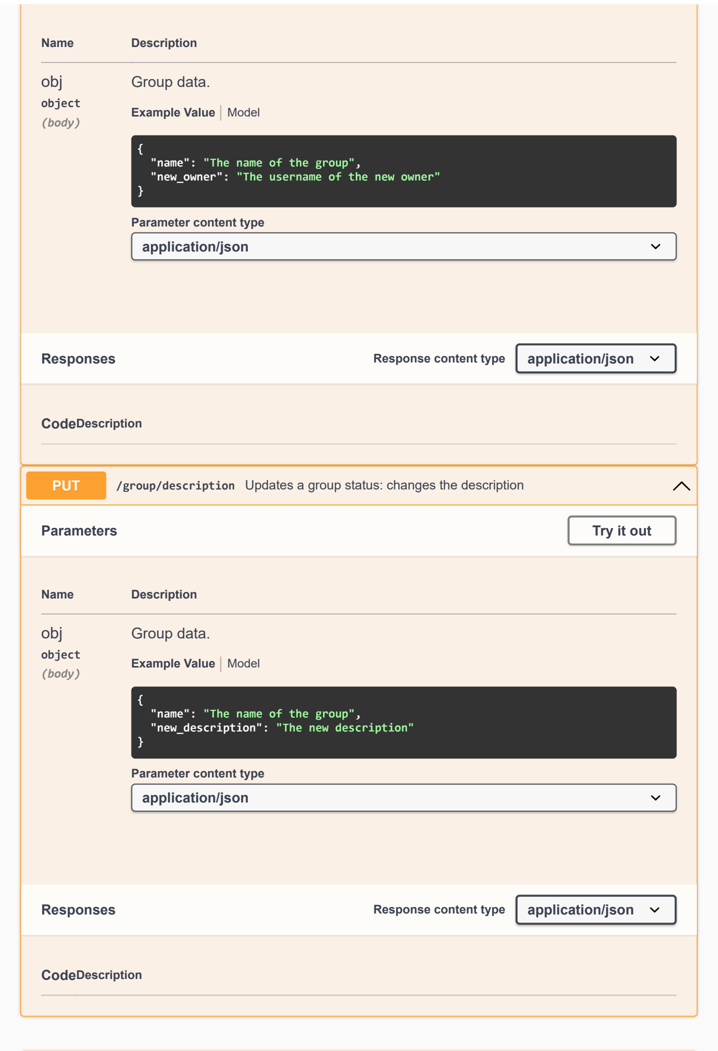
Parameter content type
application/json

Responses Response content type application/json

CodeDescription

PUT /group/owner Updates a group status: changes the owner

Parameters



Name Description

obj
object
(body)

Group data.

Example Value | Model

```
{
  "name": "The name of the group",
  "new_owner": "The username of the new owner"
}
```

Parameter content type
application/json

Responses Response content type application/json

CodeDescription

PUT /group/description Updates a group status: changes the description

Parameters

Name Description

obj
object
(body)

Group data.

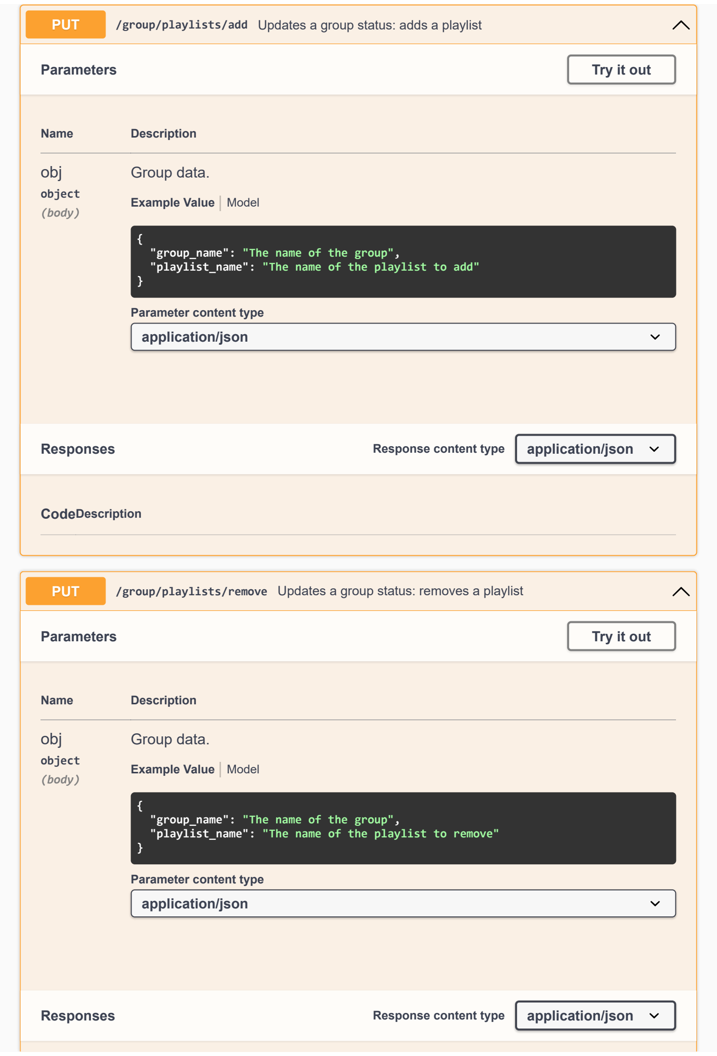
Example Value | Model

```
{
  "name": "The name of the group",
  "new_description": "The new description"
}
```

Parameter content type
application/json

Responses Response content type application/json

CodeDescription



PUT /group/playlists/add Updates a group status: adds a playlist

Parameters

Name Description

obj
object
(body)

Group data.

Example Value | Model

```
{
  "group_name": "The name of the group",
  "playlist_name": "The name of the playlist to add"
}
```

Parameter content type
application/json

Responses Response content type application/json

CodeDescription

PUT /group/playlists/remove Updates a group status: removes a playlist

Parameters

Name Description

obj
object
(body)

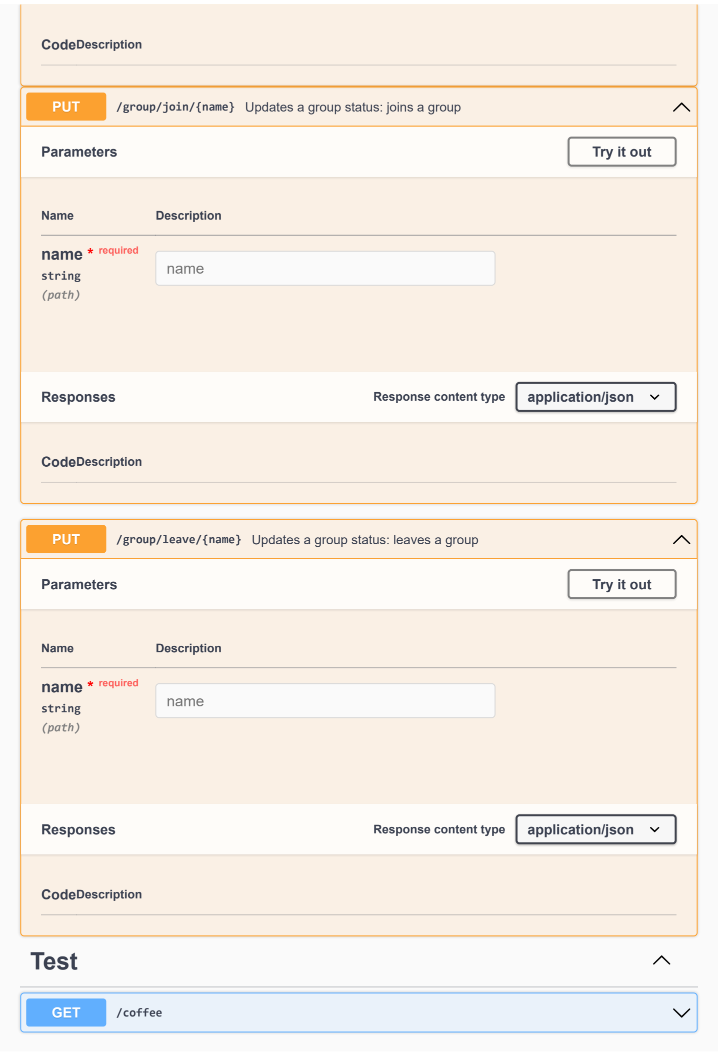
Group data.

Example Value | Model

```
{
  "group_name": "The name of the group",
  "playlist_name": "The name of the playlist to remove"
}
```

Parameter content type
application/json

Responses Response content type application/json



CodeDescription

PUT /group/join/{name} Updates a group status: joins a group

Parameters

Name Description

name * required
string
(path)

Responses Response content type application/json

CodeDescription

PUT /group/leave/{name} Updates a group status: leaves a group

Parameters

Name Description

name * required
string
(path)

Responses Response content type application/json

CodeDescription

Test

GET /coffee