

# 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	/grouplist	Get a list of all groups	▼
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	/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			

## 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/sort/{name}	Updates a playlist status: adds a song	▼
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	▼

### Code Description

200 The result of the query is sent back

Example Value | Model

```
{
  "status": 200,
  "results": [
    "acoustic",
    "afrobeats",
    "alt-rock",
    "alternative"
  ]
}
```

GET /types Gets a list of types

Parameters

Try it out

No parameters

Responses

Response content type

application/json

### Code Description

200 The known types are sent back

Example Value | Model

```
[
  "album",
  "artist",
  "episode",
  "show",
  "track"
]
```

## User Everything related to users

POST	/register	Creates a new user	▼
Parameters			
Try it out			

Name

Description

obj

object

(body)

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

Code

Description

200

The user has been inseted into the database

400

The user was already registered, or at least his email or username were, or some other invalid data was inserted

Example Value

Model

```
{
  "code": -1,
  "reason": "You are missing some fields..."
}
```

500

The database refused the insertion, more details are provided in the response

Example Value

Model

```
{
  "code": 6,
  "reason": "Generic error: explanation"
}
```

POST

/login

Logs in

Parameters

Try it out

Name

Description

obj

object

(body)

Example Value

Model

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

Parameter content type

application/json

Responses

Response content type

application/json

Code

Description

200

The user has correctly logged in

Example Value

Model

```
{
  "code": 4,
  "reason": "Logged successfully!"
}
```

400

The email was not an email

Example Value

Model

```
{
  "code": 2,
  "reason": "This isn't really an email, is it?"
}
```

401

A user with such credentials does not exist

Example Value

Model

```
{
  "code": 3,
  "reason": "This user does not exist or its password is not the one you inserte d."
}
```

GET

/logout

Performs logout

Parameters

Try it out

No parameters

Responses

Response content type

application/json

Code

Description

200

The logout was successful

Example Value

Model

```
{
  "success": true
}
```

400

The logout failed - probably the user was not logged in

Example Value

Model

```
{
  "success": false
}
```

GET

/checkLogin

Checks if the user is logged in

Parameters

Try it out

No parameters

Responses

Response content type

application/json

Code

Description

200

The user is logged in

Example Value

Model

```
{
  "name": "Name",
  "surname": "Surname",
  "userName": "User Name",
  "email": "email@email.email",
  "birthDate": "YYYY-MM-DD",
  "favoriteGenres": [],
  "favorites": {
    "album": [],
    "artist": [],
    "audiobook": [],
    "episode": [],
    "show": [],
    "track": []
  },
}
```

Code

Description

401

The user is not logged in

Example Value

Model

```
{
  "reason": "Invalid login"
}
```

PUT

/user

Updates a user

Parameters

Try it out

Name

Description

obj

object

(body)

Example Value

Model

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

Parameter content type

application/json

Responses

Response content type

application/json

Code

Description

200

The user successfully updates his data

Example Value

Model

```
{
  "code": 0,
  "explanation": "you will now be logged out. Please re-login with your new crede
```

Code

Description

ntials"

}

400

A user with such an email is already present. The email needs to be unique.

Example Value | Model

{

"code": 5,

"reason": "Already present user: please choose a different username or email"

}

401

The user is not logged in

Example Value | Model

{

"reason": "Invalid login"

}

500

The database refused the operation. More details are available in the response

Example Value | Model

{

"code": 6,

"reason": "Generic error: Explanation"

}

DELETE

/user

Deletes a user

Parameters

Try it out

No parameters

Responses

Response content type

application/json

Code

Description

200

The user was successfully deleted

Example Value | Model

{

"reason": "ok"

}

400

There was an error. More details in the response

Example Value | Model

Code

Description

{

"reason": "Generic error: Explanation"

}

401

The user is not logged in

Example Value | Model

{

"reason": "Invalid login"

}

Data

Can be used to interact with data

POST

/search

Performs a search on Spotify

Parameters

Try it out

Name

Description

obj

object

(body)

User data.

Example Value | Model

{

"string": "the string to search for",

"type": [

"album",

"track"

],

"limit": 18,

"offset": 0

}

Parameter content type

application/json

Responses

Response content type

application/json

Code

Description

200

The result is sent back

GET

/requireInfo/{kind}/{id}

Gets infos about a specific item

Parameters

Try it out

Name

Description

kind

string

(path)

kind

id

string

(path)

id

Responses

Response content type

application/json

Code

Description

200

The result is sent back

Favorites

Can be used to interact with Favorites

POST

/addOrRemoveFavorite

Adds or removes favorites

Parameters

Try it out

Name

Description

obj

object

(body)

User data.

Example Value | Model

{

"category": "album",

"id": "1kCHru7uhxBUdzkm4gzRQc",

"name": "Hamilton (Original Broadway Cast Recording)"

}

Parameter content type

application/json

Responses

Response content type

application/json

Code

Description

200

A boolean value is sent back to indicate if the element is starred or not

Example Value | Model

{

"favorite": true

}

Responses

Response content type

application/json

Code

Description

200

The query was executed correctly

Example Value | Model

{

"removed": true

}

401

The user was not logged in

Example Value | Model

{

"reason": "Invalid login"

}

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

Code

Description

200

A boolean value is sent back to indicate if the element is starred or not

Example Value | Model

{

"favorite": true

}

Code	Description
	<pre>}</pre>
401	The user was not logged in
	Example Value   Model
	<pre>{   "reason": "Invalid login" }</pre>

## Playlists

Can be used to interact with playlists

GET	/playlist/info/{name}	Gets infos about a playlist
Parameters		
Try it out		
Name		
Description		
name <span>★ required</span>		
string		name
(path)		
Responses		
Response content type		
application/json		
Code		
Description		
200	The response contains the requested informations	
	Example Value   Model	
	<pre>{   "name": "myList",   "description": "this is a playlist about old finnish songs",   "tags": [     "finnish",     "old",     "42"   ],   "visibility": true,   "owner": "userName",   "doIownIt": false,   "following": true }</pre>	
400	Something failed. Refer to the response for more details	
	Example Value   Model	

Code

Description

```
{
  "reason": "Generic error: Explanation"
}
```

401

The user is not logged in

Example Value | Model

```
{
  "reason": "Invalid login"
}
```

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

Code

Description

200

The response contains the requested informations

Example Value | Model

```
[
  {
    "name": "myList",
    "description": "this is a playlist about old finnish songs",
    "tags": [
      "finnish",
      "old",
      "42"
    ],
    "visibility": true,
    "owner": "userName"
  }
]
```

400

Something failed. Refer to the response for more details

Example Value | Model

```
{
  "reason": "Generic error: Explanation"
}
```

Code

Description

401

The user is not logged in

Example Value

Model

```
{
  "reason": "Invalid login"
}
```

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

Code

Description

200

The response contains the requested informations

Example Value

Model

```
[
  {
    "name": "myList",
    "description": "this is a playlist about old finnish songs",
    "tags": [
      "finnish",
      "old",
      "42"
    ],
    "visibility": true,
    "owner": "userName"
  }
]
```

400

Something failed. Refer to the response for more details

Example Value

Model

```
{
  "reason": "Generic error: Explanation"
}
```

Code

Description

401

The user is not logged in

Example Value | Model

```
{  "reason": "Invalid login"}
```

POST

/playlist

Creates a group

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

Code

Description

200

A playlist was created successfully

Example Value | Model

```
{  "reason": "inserted correctly"}
```

400

Something failed. Refer to the response for more details

Example Value | Model

```
{  "reason": "Probably you haven't specified the right params"}
```

Code

Description

401

The user is not logged in

Example Value | Model

```
{
  "reason": "Invalid login"
}
```

PUT

/playlist/sort/{name} Updates a playlist status: adds a song

Parameters

Try it out

Name	Description
name <span>★ required</span>	
string (path)	name
obj object (body)	Playlist data. Example Value   Model
	<pre>{   "order": [     "7m9XR7FquXLP1FwdACnS9",     "6oF8ueLnShI14P8p17sXW6",     "6p7JxatI3dpcGmN00ok2jYr",     "31XyAQ0ekAvY5LodpWmUs",     "27MB8qHAYAZi1Iwg25js1Y",     "7EqEBP0ohgk7NnKv8GFwo",     "1CzeuSrm71wHP9qsJg7p3F",     "3dP0pLbg90FVwssDjp9aT0",     "54Sc7m2Q1RM03Tpk4sFaA",     "2y8Wvng96w50HbMd8601F",     "3mJvcY9yKp80i2m18brXt",     "60G1S805gTrH5nAQ0E0PV3",     "2G91ekfCh83S01t2yfffBz",     "71X7bPD1jHrmEGYCe7kQ8",     "0NJWm3hLwIZSy5s0TG38q",     "4cxv1udVmQxryrnX1n9FqL",     "6dr7ekfh1bquvsVY8D7gyk",     "4TIV7EcFroSLwzXRYogLv6"   ] }</pre>
Parameter content type	application/json

Responses

Response content type application/json

Code

Description

200

The order of the songs was changed successfully

Example Value | Model

```
{
  "reason": "everything is fine"
}
```

400

Something failed. Refer to the response for more details

Example Value | Model

```
{
  "reason": "Generic error: Explanation"
}
```

401

The user is not logged in

Example Value | Model

```
{
  "reason": "Invalid login"
}
```

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
	<pre>{   "name": "The name of the playlist",   "song_id": "The id of the song to add" }</pre>
Parameter content type	application/json

Responses

Response content type application/json

Code

Description

200

The song was added successfully

Example Value | Model

```
{
  "reason": "done"
}
```

400

Something failed. Refer to the response for more details

Example Value | Model

```
{
  "reason": "Generic error: Explanation"
}
```

401

The user is not logged in

Example Value | Model

```
{
  "reason": "Invalid login"
}
```

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
	<pre>{   "name": "The name of the playlist",   "song_id": "The id of the song to add" }</pre>
Parameter content type	application/json

Responses

Response content type application/json

Code

Description

200

The song was removed successfully

Example Value | Model

```
{
  "reason": "done"
}
```

400

Something failed. Refer to the response for more details

Example Value | Model

```
{
  "reason": "Generic error: Explanation"
}
```

401

The user is not logged in

Example Value | Model

```
{
  "reason": "Invalid login"
}
```

PUT

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

Parameters

Try it out

Name	Description
obj object (body)	Playlist data. Example Value   Model
	<pre>{   "name": "The name of the playlist",   "new_owner": "The username of the new owner" }</pre>
Parameter content type	application/json

Responses

Response content type application/json

Code	Description
200	The owner changed, as requested Example Value   Model <pre>{   "reason": "ok" }</pre>
400	Something failed. Refer to the response for more details Example Value   Model <pre>{   "reason": "Generic error: Explanation" }</pre>
401	The user is not logged in Example Value   Model <pre>{   "reason": "Invalid login" }</pre>

PUT

/playlist/description

Updates a playlist status: changes the description

⌵

Parameters

Try it out

Name	Description
obj object (body)	Playlist data. Example Value   Model <pre>{   "name": "The name of the playlist",   "new_description": "The new description" }</pre> <div>Parameter content type application/json</div>

Responses

Response content type

application/json

Code	Description
200	The playlist changed, as requested Example Value   Model <pre>{   "reason": "ok" }</pre>
400	Something failed. Refer to the response for more details Example Value   Model <pre>{   "reason": "Generic error: Explanation" }</pre>
401	The user is not logged in Example Value   Model <pre>{   "reason": "Invalid login" }</pre>

PUT

/playlist/follow/{name}

Updates a playlist status: start following

⌵

Parameters

Try it out

Name	Description
name * required string (path)	<div>name</div>

Responses

Response content type

application/json

Code	Description
200	The user is now following the playlist Example Value   Model <pre>{   "reason": "ok" }</pre>

Code	Description
400	Something failed. Refer to the response for more details Example Value   Model <pre>{   "reason": "Generic error: Explanation" }</pre>
401	The user is not logged in Example Value   Model <pre>{   "reason": "Invalid login" }</pre>

PUT

/playlist/unfollow/{name}

Updates a playlist status: stop following

⌵

Parameters

Try it out

Name	Description
name * required string (path)	<div>name</div>

Responses

Response content type

application/json

Code	Description
200	The user has ceased following the playlist Example Value   Model <pre>{   "reason": "ok" }</pre>
400	Something failed. Refer to the response for more details Example Value   Model <pre>{   "reason": "Generic error: Explanation" }</pre>

Code	Description
401	The user is not logged in Example Value   Model <pre>{   "reason": "Invalid login" }</pre>

PUT

/playlist/tags/add

Updates a playlist status: adds a tag

⌵

Parameters

Try it out

Name	Description
obj object (body)	Playlist data. Example Value   Model <pre>{   "name": "The name of the playlist",   "tag": "The new tag" }</pre> <div>Parameter content type application/json</div>

Responses

Response content type

application/json

Code	Description
200	A tag was added to the playlist successfully
400	Something failed. Refer to the response for more details Example Value   Model <pre>{   "reason": "Generic error: Explanation" }</pre>
401	The user is not logged in Example Value   Model <pre>{   "reason": "Invalid login" }</pre>

Code

Description

PUT

/playlist/tags/remove

Updates a playlist status: removes a tag

Parameters

Try it out

Name

Description

obj  
object  
(body)

Playlist data.  
Example Value | Model

```
{  
  "name": "The name of the playlist",  
  "tag": "The old tag"  
}
```

Parameter content type  
application/json

Responses

Response content type

application/json

Code

Description

200

A tag was removed from the playlist successfully  
Example Value | Model

```
{  
  "reason": "ok"  
}
```

400

Something failed. Refer to the response for more details  
Example Value | Model

```
{  
  "reason": "Generic error: Explanation"  
}
```

Code

Description

401

The user is not logged in  
Example Value | Model

```
{  
  "reason": "Invalid login"  
}
```

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

Code

Description

200

The playlist is now public  
Example Value | Model

```
{  
  "reason": "ok"  
}
```

400

Something failed. Refer to the response for more details  
Example Value | Model

```
{  
  "reason": "Generic error: Explanation"  
}
```

Code

Description

401

The user is not logged in  
Example Value | Model

```
{  
  "reason": "Invalid login"  
}
```

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

Code

Description

200

The playlist is now private  
Example Value | Model

```
{  
  "reason": "ok"  
}
```

400

Something failed. Refer to the response for more details  
Example Value | Model

```
{  
  "reason": "Generic error: Explanation"  
}
```

Code

Description

401

The user is not logged in  
Example Value | Model

```
{  
  "reason": "Invalid login"  
}
```

DELETE

/playlist/{name}

Delete a playlist

Parameters

Try it out

Name

Description

name \* required  
string  
(path)

name

Responses

Response content type

application/json

Code

Description

200

The playlist was successfully deleted  
Example Value | Model

```
{  
  "reason": "done correctly"  
}
```

400

Something failed. Refer to the response for more details  
Example Value | Model

```
{  
  "reason": "Generic error: Explanation"  
}
```

Code	Description
401	The user is not logged in
Example Value   Model	
<pre>{   "reason": "Invalid login" }</pre>	

## Groups

Can be used to interact with groups



GET

/group/{name}

Gets infos about a group

Try it out

Parameters

Name	Description
<b>name</b> <span>★ required</span>	<div>name</div>
string (path)	

Responses

Response content type 

application/json

Code	Description
200	<div>The response contains the data requested</div> <div>Example Value   Model</div> <div><pre>{  "name": "myGroup",  "decription": "This is a group for ...",  "playlistsShared": [    "classicalMusic",    "musicaClassica"  ],  "owner": "userName",  "users": [    "userName",    "user42",    "lambda",    "SophosIoun"  ],  "doIownIt": false,  "following": true  }</pre></div>
400	<div>Invalid data was providen</div>

Code	Description
	Example Value   Model
	<pre>{   "reason": "bad data" }</pre>
401	The user was not logged in
	Example Value   Model
	<pre>{   "reason": "Invalid login" }</pre>

DELETE

/group/{name} Delete a group

Try it out

Parameters

Name	Description
name <span>★ required</span>	<div>name</div>
string	
(path)	

Responses

Response content type 

application/json

Code	Description
200	<div>The group was deleted</div> <div>Example Value   Model</div> <div><pre>{  "reason": "ok"}</pre></div>
400	<div>Something failed. Refer to the response for more details</div> <div>Example Value   Model</div> <div><pre>{  "reason": "Generic error: Explanation"}</pre></div>
401	<div>The user is not logged in</div>

Code

Description

Example Value

Model

```
{
  "reason": "Invalid login"
}
```

GET

/groupList

Get a list of all groups

↗

Parameters

Try it out

No parameters

Responses

Response content type

application/json

Code

Description

200

The response contains the requested data

Example Value

Model

```
[
  {
    "name": "myGroup",
    "decription": "This is a group for ...",
    "playlistsShared": [
      "classicalMusic",
      "musicaClassica"
    ],
    "owner": "userName",
    "users": [
      "userName",
      "user42",
      "Lambda",
      "SophosIoun"
    ]
  }
]
```

400

Something failed. Refer to the response for more details

Example Value

Model

```
{
  "reason": "Generic error: Explanation"
}
```

401

The user is not logged in

Example Value

Model

```
{
  "reason": "Invalid login"
}
```

Code

Description

POST

/group

Creates a group

Try it out

Parameters

Name	Description
obj object (body)	Group data. <div>Example Value   Model</div> <pre>{   "nome": "The name of your new group",   "descrizione": "The description of your new group" }</pre> <div>Parameter content type</div> <div>application/json</div>

Responses

Response content type

application/json

Code	Description
200	A new group was created <div>Example Value   Model</div> <pre>{   "reason": "inserted correctly" }</pre>
400	Something failed. Refer to the response for more details <div>Example Value   Model</div> <pre>{   "reason": "Probably you haven't specified the right params" }</pre>
401	The user is not logged in <div>Example Value   Model</div> <pre>{   "reason": "Invalid login" }</pre>



Code

Description

PUT

/group/owner

Updates a group status: changes the owner

Parameters

Try it out

Name	Description
obj	Group data.
object (body)	<div>Example Value   Model</div> <div><pre>{   "name": "The name of the group",   "new_owner": "The username of the new owner" }</pre></div> <div>Parameter content type</div> <div>application/json</div>

Responses

Response content type

application/json

Code	Description
200	<div>The owner was changed, according to the request</div> <div>Example Value   Model</div> <div><pre>{   "reason": "ok" }</pre></div>
400	<div>Something failed. Refer to the response for more details</div> <div>Example Value   Model</div> <div><pre>{   "reason": "you do not own this group, or some other data you inserted is not valid. Stop trying to hack me, please" }</pre></div>

Code

Description

401

The user is not logged in

Example Value | Model

```
{
  "reason": "Invalid login"
}
```

PUT

/group/description

Updates a group status: changes the description

Parameters

Try it out

Name	Description
obj	Group data.
object (body)	<div>Example Value   Model</div> <div><pre>{   "name": "The name of the group",   "new_description": "The new description" }</pre></div> <div>Parameter content type</div> <div>application/json</div>

Responses

Response content type

application/json

Code	Description
200	<div>The description was changed, according to the request</div> <div>Example Value   Model</div> <div><pre>{   "reason": "ok" }</pre></div>
400	<div>Something failed. Refer to the response for more details</div> <div>Example Value   Model</div> <div><pre>{   "reason": "you do not own this group, or some other data you inserted is not valid. Stop trying to hack me, please" }</pre></div>

Code

Description

401

The user is not logged in

Example Value | Model

```
{
  "reason": "Invalid login"
}
```

PUT

/group/playlists/add

Updates a group status: adds a playlist

Parameters

Try it out

Name	Description
obj	Group data.
object (body)	<div>Example Value   Model</div> <div><pre>{   "group_name": "The name of the group",   "playlist_name": "The name of the playlist to add" }</pre></div> <div>Parameter content type</div> <div>application/json</div>

Responses

Response content type

application/json

Code	Description
200	<div>The playlist was added, according to the request</div> <div>Example Value   Model</div> <div><pre>{   "reason": "done successfully" }</pre></div>
400	<div>Something failed. Refer to the response for more details</div> <div>Example Value   Model</div> <div><pre>{   "reason": "invalid data or wrong permissions" }</pre></div>

Code

Description

401

The user is not logged in

Example Value | Model

```
{
  "reason": "Invalid login"
}
```

PUT

/group/playlists/remove

Updates a group status: removes a playlist

Parameters

Try it out

Name	Description
obj	Group data.
object (body)	<div>Example Value   Model</div> <div><pre>{   "group_name": "The name of the group",   "playlist_name": "The name of the playlist to remove" }</pre></div> <div>Parameter content type</div> <div>application/json</div>

Responses

Response content type

application/json

Code	Description
200	<div>The playlist was removed, according to the request</div> <div>Example Value   Model</div> <div><pre>{   "reason": "done successfully" }</pre></div>
400	<div>Something failed. Refer to the response for more details</div> <div>Example Value   Model</div> <div><pre>{   "reason": "invalid data or wrong permissions" }</pre></div>

Code

Description

401

The user is not logged in

Example Value | Model

```
{
  "reason": "Invalid login"
}
```

PUT

/group/join/{name}

Updates a group status: joins a group

Parameters

Try it out

Name

Description

name ★ required

string  
(path)

name

Responses

Response content type

application/json

Code

Description

200

The user joined the group, according to the request

Example Value | Model

```
{
  "reason": "ok"
}
```

400

Something failed. Refer to the response for more details

Example Value | Model

```
{
  "reason": "bad data or already in this group"
}
```

Code

Description

401

The user is not logged in

Example Value | Model

```
{
  "reason": "Invalid login"
}
```

PUT

/group/leave/{name}

Updates a group status: leaves a group

Parameters

Try it out

Name

Description

name ★ required

string  
(path)

name

Responses

Response content type

application/json

Code

Description

200

The user left the group, according to the request

Example Value | Model

```
{
  "reason": "ok"
}
```

400

Something failed. Refer to the response for more details

Example Value | Model

```
{
  "reason": "bad data or not in this group"
}
```

Code

Description

401

The user is not logged in

Example Value | Model

```
{
  "reason": "Invalid login"
}
```

Test

GET

/coffee

Parameters

Try it out

No parameters

Responses

Response content type

application/json

Code

Description

418

I'm a teapot

Example Value | Model

```
{
  "answer": "I'm not a teapot, but I cannot brew coffee..."
}
```