# Music App Documentation

## Music App RESTful API Documentation

We added support for the following operations:

### Music:

* Get a specific song
* Get all songs
* Upload/add a new song
* Delete a specific song
* Like/dislike
* favorite

### Playlist:

* Getting a song from a playlist
* Getting a playlist
* Adding a song to a playlist
* Removing a song from a playlist
* Creating a new playlist
* Removing a playlist
* Fork playlist

### User:

* Getting a user
* Adding a user to the system
* Removing a User
* Updating an existing user
* Getting a user’s permissions
* User recommendations
* Groups
* (Friends)?

### /music/all/:offset/:limit

#### Parameters:

:offset

The specified offset starting at 0.

:limit

The limit of songs to send in a single request.

#### Returns:

The list of songs.

### /music/:id

#### Parameters:

:id The identifier of the song to find.

#### Returns:

The song if found otherwise null.

### /music/upload

#### Parameters:

## Stored Procedure Documentation

This is the list of stored procedures and their descriptions.

### get\_songs

#### Parameters:

None

#### Returns:

All songs in the system database.

### get\_song

#### Parameters:

The integer id value of the song.

#### Returns:

The selected song in the system database or false if none found.

### add\_song

#### Parameters:

The name string value of the song.  
The author’s id as an integer value.

#### Returns:

True if successful otherwise false.

### delete\_song

#### Parameters:

The integer id value of the song.

#### Returns:

True if successful otherwise false.

### get\_users

#### Parameters:

None.

#### Returns:

A list of all users if it succeeds otherwise false.

### get\_user

#### Parameters:

The integer id value of the user.

#### Returns:

The user if successful otherwise false.

### add\_user

#### Parameters:

The name of the user.  
The email of the user.  
The password of the user.

#### Returns:

True if successful otherwise false.