

Document the API for a new dating service for people who only care about how their dates sound, not how they look. The company is called SoundDate, and here is the information you'll need to document the request for a user to upload a sound file to their own profile and to get information on sound files for another user's profile.

## Uploading a Sound File to the current User's Profile

**Here's the information you get from the developer team:**

1. The server address is `https://api.sounddate.com`
2. The resource is `/profile/sound`
3. The method is POST
4. There are three headers:
  - a. Bearer has the access token. Required.
  - b. Content-Type has the sound file format, which can be either `audio/mpeg` for mp3 files or `audio/x-wav` for wav files. Default is `audio/mpeg`.
  - c. Accept has the response format, which can be `application/xml` or `application/json`. Default is JSON.
5. The POST body is the sound file
6. The sound file must be 5 minutes or shorter.
7. The response has only two elements:
  - a. id: An integer, which is the ID of the new sound file
  - b. length: A float, which is the length of the sound file, in seconds
8. Don't put in a Status and Errors table. We'll have one for the entire documentation set.

## Retrieving Sound File Information for a User

**Here's the information you get from the developer team:**

1. The server address is `https://api.sounddate.com`
2. The resource is `/user/{user id}/profile/sound`.
3. The method is GET
4. There are two headers:
  - a. Bearer has the access token. Required.
  - b. Accept has the response format, which can be `application/xml` or `application/json`. Default is JSON.
5. There is one query parameter called `sortOrder`. This determines the order that the sound files are returned.

It has four possible values:

  - a. `mostRecent`, sorts most recent to earliest
  - b. `earliest`, sorts earliest to most recent
  - c. `shortest`, sorts shortest to longest
  - d. `longest`, sorts longest to shortest
6. The `sortOrder` parameter is optional. The default is `mostRecent`.

7. Here's a sample response:

```
{
  "soundFiles": [
    {
      "id": 23456,
      "url": "https://api.sounddate.com/profile/sound/23456.mp3",
      "length": 11.2
    },
    {
      "id": 24559,
      "url": "https://api.sounddate.com/profile/sound/24559.mp3",
      "length": 19.8
    }
  ]
}
```

## Status Codes and Errors

Add a status codes and errors table at the very end. Here are the status codes to document.

1. 200, success for both POST and GET
2. 401, access token is no good
3. 413, sound file that was POSTed was too long

# Upload a sound file

---

Uploads a sound file for a user's profile

## URL

POST <https://api.sounddate.com/profile/sound>

## HEADERS

Header Name	Description	Required	Values
<b>Bearer</b>	<b>Access token</b>	<b>Required</b>	<b>See Authorization section</b>
<b>Content-Type</b>	<b>The format of the sound file to upload</b>	<b>Optional</b>	<b>audio/mpeg or audio/x-wav. Default is audio/mpeg.</b>
<b>Accept</b>	<b>The format of the returned data</b>	<b>Optional</b>	<b>Application/xml or application/json. Default is application/json</b>

## POST or PUT Body

The sound file.

**Note:** The sound file must be 5 minutes or shorter.

## Sample Request

POST <https://api.sounddate.com/profile/sound>

Bearer: {access token}

Content-Type: audio/mpeg

Accept: application/json

{sound file}

## Response

Element	Description	Type	Notes
<b>id</b>	<b>The ID of the new sound file</b>	<b>Integer</b>	
<b>length</b>	<b>The length of the sound file</b>	<b>Float</b>	<b>Length is in seconds</b>

### Sample Response

```
{
  "Id" : 3543,
  "Length" : 19.8
}
```

## Retrieve a list of sound files

---

Retrieves a list of profile sound file URLs and their lengths for the specified user

### URL

GET <https://api.sounddate.com/user/{userid}/profile/sound>

Where {user id} is the ID of the user whose profile contains the sound files.

### Query Parameters

Parameter	Description	Type	Required	Notes
sortOrder	The order to return the sound file information	string	Optional	Valid values: mostRecent, earliest, shortest, longest. Default is mostRecent.

Note:

- **mostRecent** returns the most recent sound files to the earliest.
- **earliest** returns the earliest sound files to the most recent.
- **shortest** returns the shortest sound files to longest.
- **longest** returns the longest sound files to the shortest.

### Headers

Header Name	Description	Required	Notes
Bearer	Access token	Required	See Authorization section
Accept	The format of the returned data	Optional	application/xml or application/json. Default is application/json

## Sample Request

GET <https://api.sounddate.com/user/345354/profile/sound?sortOrder=shortest>

Bearer: {access token}

Accept: application/json

## Response

Element	Description	Type	Notes
<b>soundFiles</b>	<b>List of sound file information</b>	<b>Array</b>	
id	The ID of the sound file	integer	
url	The URL of the sound file	string	
length	The length of the sound file	float	The length in seconds.

## Sample Response

```
{
  "soundFiles": [
    {
      "id": 23456,
      "url": "https://api.sounddate.com/profile/sound/23456.mp3",
      "length": 11.2
    },
    {
      "id": 24559,
      "url": "https://api.sounddate.com/profile/sound/24559.mp3",
      "length": 19.8
    }
  ]
}
```

# Status Codes and Errors

---

The following table lists the returned HTTP status codes

Code	Description	Notes
200	OK	Sound file was successfully added.
401	Unauthorized	Access token is invalid
413	Request Entity Too Large	Uploaded sound file is longer than 5 minutes