

# Songwich API - v0.4

## Summary

Status Codes.....	2
Users API.....	3
POST /v0.4/users.....	3
GET /v0.4/users.....	4
PUT /v0.4/users/<userId>.....	5
PUT /v0.4/users/<userId>/markAsVerified.....	6
Scrobbles API.....	8
POST /v0.4/scrobbles.....	8
GET /v0.4/scrobbles/<userId>.....	9
PUT /v0.4/scrobbles/<scrobbleId>.....	11
DELETE /v0.4/scrobbles/<scrobbleId>.....	12
Stations API.....	13
POST /v0.4/stations.....	13
GET /v0.4/stations.....	15
PUT /v0.4/stations/<stationId>.....	18
PUT /v0.4/stations/<stationId>/markAsVerified.....	19
PUT /v0.4/stations/<stationId>/addScrobblers.....	19
PUT /v0.4/stations/<stationId>/removeScrobblers.....	20
POST /v0.4/nextSong.....	21
Song Feedback API.....	23
POST /v0.4/songFeedback.....	23
GET /v0.4/starredSongs/<userId>.....	24
GET /v0.4/isSongStarred/<userId>.....	24
DELETE /v0.4/songFeedback/<idForFeedback>/<feedbackType>.....	26
Subscriptions API.....	27
POST /v0.4/subscriptions.....	27
GET /v0.4/subscriptions/<userId>.....	28
PUT /v0.4/subscriptions/<subscriptionId>/end.....	28

# Status Codes

Status codes sent in the JSON response on API calls:

Code	Semantics
-1	Unknown error
0	Success
1	Bad request
2	Invalid X-Songwich.devAuthToken
3	Invalid X-Songwich.userAuthToken
4	Unauthorized
5	Invalid parameter

# Users API

API methods related to end-users.

**POST** /v0.4/users

Creates a user and relates it to the app given by devAuthToken.

Parameter	Required?	Values	Location
X-Songwich.devAuthToken	yes	3bde6fba-1ae5-4d7f-8000-f2aba160b71a	header
userEmail	yes	john.smith@example.com	body
name	no	John Smith	body
imageUrl	no	http://image.com/image.jpg	body
shortBio	no	I'm from Brazil	body

Example of user creation:

```
curl
--request POST http://api.songwich.com/v0.4/users
--header 'Content-Type: application/json'
--header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
--data '
{
  "name": "John Smith",
  "userEmail": "john.smith@example.com"
},
'
```

JSON response:

```
{
  "status": "0",
  "message": "Success",
  "user": {
    "userEmail": "john.smith@example.com",
    "name": "John Smith",
    "userId": "525c10fa45690266809924ed",
    "verified": "false",
    "userAuthToken": "02bbf9f8-a65d-407b-92fc-d02c1138fcd8"
  }
}
```

## GET /v0.4/users

Gets information on end-users.

Parameter	Required?	Values	Location
X-Songwich.devAuthToken	yes	3bde6fba-1ae5-4d7f-8000-f2aba160b71a	header
userId	no	525c379145691ceff24838d1	URL param
userEmail	no	john.smith@example.com	URL param

Get all users:

```
curl
--request GET http://api.songwich.com/v0.4/users
--header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
```

JSON response:

```
{
  "status": "0",
  "message": "Success",
  "user": [
    {
      "userEmail": "john.smith@example.com",
      "name": "John Smith",
      "userId": "525c379145691ceff24838d1",
      "verified": "false"
    },
    {
      "userEmail": "david.jones@example.com",
      "name": "David Jones",
      "userId": "525c3ecb45691ceff24838d2",
      "verified": "true"
    }
  ]
}
```

Get detailed information of a specific user:

```
curl
--request GET http://api.songwich.com/v0.4/users?
userId=5259e16d92e667731a1bd6d7
--header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
```

JSON response:

```
{
  "status": "0",
  "message": "Success",
  "user": {
    "userEmail": "gabrielcs@gmail.com",
```

```

"name": "Gabriel Cypriano",
"userId": "5259e16d92e667731a1bd6d8",
"verified": "false",
"scrobblers": [
  {
    "stationId": "5259e26b92e667731a1bd6e0",
    "stationName": "Gabriel Cypriano FM",
    "scrobblers": [
      "5259e16d92e667731a1bd6d8"
    ],
    "imageUrl": "http://example.com/image.jpg",
    "active": "false"
  },
  {
    "stationId": "5259e2c892e667731a1bd6e6",
    "stationName": "Songwich FM",
    "scrobblers": [
      "5259e16d92e667731a1bd6d8",
      "5259e18092e667731a1bd6d9"
    ],
    "imageUrl": "http://example.com/image.jpg",
    "active": "true"
  }
],
"activeStationSubscriptions": [
  {
    "id": "52a8c3ed4d474dee367c4104",
    "station": {
      "stationId": "52a8c3c94d474dee367c4103",
      "stationName": "David Jones FM",
      "imageUrl": "http://images.com/image.jpg"
    }
  },
  {
    "id": "52a8c1a04d474dee367c4101",
    "station": {
      "stationId": "52a8c16d4d474dee367c4100",
      "stationName": "John Smith FM",
      "imageUrl": "http://images.com/image.jpg"
    }
  }
]
}

```

**PUT** **/v0.4/users/<userId>**

Updates a user's data.

Parameter	Required?	Values	Location
X-Songwich.devAuthToken	yes	3bde6fba-1ae5-4d7f-8000-f2aba160b71a	header

X-Songwich.userAuthToken	yes	3dadf599-5307-4f2c-a5a7-cecb4eb47979	header
userId	yes	525cb2fc4569da1eb4df8a91	URL
userEmail	no	john.smith@example.com	body
name	no	John Smith	body
imageUrl	no	http://image.com/image.jpg	body
shortBio	no	I'm from Brazil	body

Example of user update:

```
curl
--request PUT http://api.songwich.com/v0.4/users/525cb2fc4569da1eb4df8a91
--header 'Content-Type: application/json'
--header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
--header 'X-Songwich.userAuthToken: 3dadf599-5307-4f2c-a5a7-cecb4eb47979'
--data '
{
  "userEmail": "john.smith@better-example.com"
}
```

JSON response:

```
{
  "status": "0",
  "message": "Success",
  "user": {
    "userEmail": "john.smith@better-example.com",
    "name": "John Smith",
    "imageUrl": "http://image.com/image1.jpg",
    "shortBio": "I'm the crazy guy",
    "userId": "525cb72f4569da1eb4df8aa6",
    "verified": "true"
  }
}
```

**PUT** /v0.4/users/<userId>/markAsVerified

Marks a user as verified. The user's individual station(s) will be marked as verified as well.

Parameter	Required?	Values	Location
X-Songwich.devAuthToken	yes	3bde6fba-1ae5-4d7f-8000-f2aba160b71a	header
X-Songwich.userAuthToken	yes	3dadf599-5307-4f2c-a5a7-cecb4eb47979	header
userId	yes	525cb2fc4569da1eb4df8a91	URL

Example of marking a user as verified:

```
curl
--request PUT
http://api.songwich.com/v0.4/users/52af21f74d47446b756d5128/markAsVerified
--header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
--header 'X-Songwich.userAuthToken: 3dadf599-5307-4f2c-a5a7-cecb4eb47979'
```

JSON response:

```
{
  "status": "0",
  "message": "Success",
  "user": {
    "userEmail": "david.jones@example.com",
    "name": "David Jones",
    "imageUrl": "http://image.com/image2.jpg",
    "shortBio": "I'm not a crazy guy",
    "userId": "52af21f74d47446b756d5128",
    "verified": "true"
  }
}
```

# Scrobbles API

API methods related to scrobbling.

**POST** /v0.4/scrobbles

Scrobbles a track listen by a given user, who is identified by the `userAuthToken`.

Parameter	Required?	Values	Location
X-Songwich.devAuthToken	yes	3bde6fba-1ae5-4d7f-8000-f2aba160b71a	header
X-Songwich.userAuthToken	yes	0b115241-0ff4-4a08-8ecc-d9745beff24a	header
trackTitle	yes	Linoleum	body
artistsNames	yes	Arctic Monkeys	body
albumTitle	no	Random Access Memories	body
chosenByUser	no	true, false	body
player	no	Youtube, Deezer, Spotify	body
timestamp	no	1381777870827	body

Scrobble example with 1 artist and `timestamp` set to the current time:

```
curl
--request POST http://api.songwich.com/v0.4/scrobbles
--header 'Content-Type: application/json'
--header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
--header 'X-Songwich.userAuthToken: f0f514d5-9eb2-41b1-81dd-db39ad1ed19c'
--data '
{
  "trackTitle": "Supersoaker",
  "artistsNames": "Kings of Leon",
  "albumTitle": "Mechanical Bull",
  "chosenByUser": "true",
  "player": "Youtube"
}
```

JSON response:

```
{
  "status": "0",
  "message": "Success",
  "scrobble": {
    "scrobbleId": "525eca254569e2fce31d3d05",
    "userId": "525ec9d34569e2fce31d3d04",
```



```

    "trackTitle": "Supersoaker",
    "artistsNames": [
        "Kings of Leon"
    ],
    "chosenByUser": "true",
    "player": "Youtube",
    "timestamp": "1381943845547"
}
}

```

Scrobble example with multiple artists and timestamp in the past:

```

curl
--request POST http://api.songwich.com/v0.4/scrobbles
--header 'Content-Type: application/json'
--header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
--header 'X-Songwich.userAuthToken: f0f514d5-9eb2-41b1-81dd-db39ad1ed19c'
--data '
{
    "trackTitle": "Get Lucky",
    "artistsNames": ["Daft Punk", "Pharrell Williams"],
    "albumTitle": "Random Access Memories",
    "chosenByUser": "false",
    "player": "Deezer",
    "timestamp": "1381777870838"
}
'

```

JSON Response:

```

{
    "status": "0",
    "message": "Success",
    "scrobble": {
        "scrobbleId": "525ecb344569e2fce31d3d06",
        "userId": "525c379145691ceff24838d1",
        "trackTitle": "Get Lucky",
        "artistsNames": [
            "Daft Punk",
            "Pharrell Williams"
        ],
        "albumTitle": "Random Access Memories",
        "chosenByUser": "false",
        "player": "Deezer",
        "timestamp": "1381777870838"
    }
}

```

**GET** /v0.4/scrobbles/<userId>

Gets the scrobbles by a given user.

This API method supports pagination via the `since`, `sinceInclusive`, `until`, and `untilInclusive` parameters. A maximum of 1 of these parameters might be set.

Parameter	Required?	Values	Location
X-Songwich.devAuthToken	yes	3bde6fba-1ae5-4d7f-8000-f2aba160b71a	header
X-Songwich.userAuthToken	yes	0b115241-0ff4-4a08-8ecc-d9745beff24a	header
userId	yes	525bf0124569f1d4a7651092	URL
chosenByUserOnly	no	“true” or “false” (default=false)	URL param
results	no	Integer (default=30, max=100)	URL param
since	no	ObjectId (exclusive)	URL param
sinceInclusive	no	ObjectId (inclusive)	URL param
until	no	ObjectId (exclusive)	URL param
untilInclusive	no	ObjectId (inclusive)	URL param

Gets the last 2 scrobbles by a certain user:

```
curl
--request GET http://api.songwich.com/v0.4/scrobbles/525c379145691ceff24838d1?
results=2
--header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
--header 'X-Songwich.userAuthToken: f0f514d5-9eb2-41b1-81dd-db39ad1ed19c'
```

JSON Response:

```
{
  "status": "0",
  "message": "Success",
  "scrobbles": [
    {
      "scrobbleId": "52cd8d02c543dcbdc6662bbf",
      "trackTitle": "Get Lucky",
      "artistsNames": [
        "Daft Punk",
        "Pharrell Williams"
      ],
      "chosenByUser": "false",
      "player": "Deezer",
      "timestamp": "1381778203410"
    },
    {
      "scrobbleId": "52cd8cf7c543dcbdc6662bbe",
      "trackTitle": "Supersoaker",
      "artistsNames": [
        "Kings of Leon"
      ],
      "chosenByUser": "true",
      "player": "Youtube",
      "timestamp": "1381777870827"
    }
  ]
}
```

```

    ]
    "paging": {
      "newer":
"http://api.songwich.com/v0.4/scrobbles/525c379145691ceff24838d1?
chosenByUserOnly=false&results=2&since=52cd8d02c543dcbdc6662bbf",
      "older":
"http://api.songwich.com/v0.4/scrobbles/525c379145691ceff24838d1?
chosenByUserOnly=false&results=2&until=52cd8cf7c543dcbdc6662bbe"
    }
  }
}

```

**PUT** **/v0.4/scrobbles/<scrobbleId>**

Updates a scrobble's data.

Parameter	Required?	Values	Location
X-Songwich.devAuthToken	yes	3bde6fba-1ae5-4d7f-8000-f2aba160b71a	header
X-Songwich.userAuthToken	yes	3dadf599-5307-4f2c-a5a7-cecb4eb47979	header
scrobbleId	yes	525cb2fc4569da1eb4df8a91	URL
chosenByUser	yes	“true” or “false”	body

Example of scrobble update:

```

curl
--request PUT http://api.songwich.com/v0.4/scrobbles/526923712c048df73a447dac
--header 'Content-Type: application/json'
--header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
--header 'X-Songwich.userAuthToken: 3dadf599-5307-4f2c-a5a7-cecb4eb47979'
--data '
{
  "chosenByUser": "false"
}
'

```

JSON response:

```

{
  "status": "0",
  "message": "Success",
  "scrobble": {
    "scrobbleId": "526923712c048df73a447dac",
    "userId": "5269235d2c048df73a447dab",
    "trackTitle": "Get Lucky",
    "artistsNames": [
      "Daft Punk",
      "Pharrell Williams"
    ],
    "albumTitle": "Random Access Memories",
    "chosenByUser": "false",
  }
}

```

```
    "player": "Deezer",  
    "timestamp": "1382622065284"  
  }  
}
```

**DELETE** /v0.4/scrobbles/<scrobbleId>

Deletes scrobbles.

Parameter	Required?	Values	Location
X-Songwich.devAuthToken	yes	3bde6fba-1ae5-4d7f-8000-f2aba160b71a	header
X-Songwich.userAuthToken	yes	0b115241-0ff4-4a08-8ecc-d9745beff24a	header
scrobbleId	yes	525c379145691ceff24838d1	URL

Deletes a given scrobble:

```
curl
--request DELETE
http://api.songwich.com/v0.4/scrobbles/525c379145691ceff24838d1
--header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
--header 'X-Songwich.userAuthToken: f0f514d5-9eb2-41b1-81dd-db39ad1ed19c'
```

JSON Response:

```
{
  "status": "0",
  "message": "Success"
}
```

# Stations API

API methods related to radio stations.

**POST** /v0.4/stations

Creates either an individual or a group radio station. Group stations are marked with a `groupName` element, both in the request and in the response.

If the scrobbles have enough scrobbles to activate the station than it is automatically activated and both `nowPlaying` and `lookAhead` will be set. Otherwise, it will have a `stationReadiness` element in the result that says how ready the station is from 0.00 to 1.00.

If it is an individual station being created by a verified user than the station will be marked as verified as well. If `subscribeScrobbles=true`, the station's scrobbles will automatically have a subscription.

Parameter	Required?	Values	Location
X-Songwich.devAuthToken	yes	3bde6fba-1ae5-4d7f-8000-f2aba160b71a	header
X-Songwich.userAuthToken	yes	0b115241-0ff4-4a08-8ecc-d9745beff24a	header
subscribeScrobbles	no	“true” or “false”	URL
scrobberIds	yes	525c379145691ceff24838d1	body
stationName	yes	Test Group FM	body
groupName	no	Test Group	body
imageUrl	no	http://example.com/image.jpg	body
description	no	This is a description	body

Creates an individual radio station:

```
curl
--request POST http://api.songwich.com/v0.4/stations
--header 'Content-Type: application/json'
--header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
--header 'X-Songwich.userAuthToken: f0f514d5-9eb2-41b1-81dd-db39ad1ed19c'
--data '
{
  "scrobberIds": "525c379145691ceff24838d1",
  "stationName": "John Smith FM",
  "imageUrl": "http://example.com/image.jpg"
}
```

JSON response (for an inactive station):

```
{
  "status": "0",
  "message": "Success",
  "station": {
    "stationId": "527bb4d2db8834d96d9ddb8b",
    "stationName": "John Smith FM",
    "scrobbleIds": [
      "525c379145691ceff24838d1"
    ],
    "stationReadiness": "0.33",
    "imageUrl": "http://example.com/image.jpg",
    "verified": "false",
    "active": "false"
  }
}
```

Creates a group radio station:

```
curl
--request POST http://api.songwich.com/v0.4/stations
--header 'Content-Type: application/json'
--header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
--header 'X-Songwich.userAuthToken: f0f514d5-9eb2-41b1-81dd-db39ad1ed19c'
--data '
{
  "scrobbleIds": ["526826fe2c043cb13d63e7a5", "526827282c043cb13d63e7a6"],
  "stationName": "Test Group FM",
  "groupName": "Test Group",
  "imageUrl": "http://example.com/image.jpg"
}
'
```

JSON response (for an active station):

```
{
  "status": "0",
  "message": "Success",
  "station": {
    "stationId": "526827f82c043cb13d63e7ad",
    "stationName": "Test Group FM",
    "groupName": "Test Group",
    "scrobbleIds": [
      "526826fe2c043cb13d63e7a5",
      "526827282c043cb13d63e7a6"
    ],
    "nowPlaying": {
      "trackTitle": "Get Lucky",
      "artistName": "[Daft Punk, Pharrell Williams]",
      "idForFeedback": "526827f82c043cb13d63e7ab",
      "recentScrobblers": [
        {
          "name": "John Smith",
          "userId": "526826fe2c043cb13d63e7a5"
        }
      ]
    }
  }
}
```

```

    },
    {
      "name": "David Jones",
      "userId": "526827282c043cb13d63e7a6"
    }
  ]
},
"lookAhead": {
  "trackTitle": "NOFX",
  "artistName": "[Linoleum]",
  "idForFeedback": "526827f82c043cb13d63e7ac",
  "recentScrobbles": [
    {
      "name": "David Jones",
      "userId": "526827282c043cb13d63e7a6"
    }
  ]
},
"imageUrl": "http://example.com/image.jpg",
"verified": "false",
"active": "true"
}
}

```

## GET /v0.4/stations

Gets information on available stations.

When using the `stationID` optional parameter, if the station was previously inactive but it is ready (its active scrobbles have enough scrobbles to activate it) then it is automatically activated and both `nowPlaying` and `lookAhead` will be set. Otherwise, it will have a `stationReadiness` element in the result that says how ready the station is from 0.00 to 1.00. Moreover, `includeScrobbles=true` returns not only scrobbles' ids but also their name and `imageUrl`. `onlyActiveStations=true` returns stations that have been launch and are active only.

When not using the `stationID` optional parameter, an `isActive` element will point out whether the station is active or not. To try and activate an inactive station, please call this method with the `stationID` optional parameter (which will only happen if it is ready, that is `stationReadiness=1.00`). Moreover, `includeScrobbles=true` and `onlyActiveStations=true` have no effect in this case.

Parameter	Required?	Values	Location
X-Songwich.devAuthToken	yes	3bde6fba-1ae5-4d7f-8000-f2aba160b71a	header
stationId	no	525bf0124569f1d4a7651092	URL param
onlyActiveStations	no	“true” or “false”	URL param
includeScrobblesData	no	“true” or “false”	URL param



Get all stations:

```
curl
--request GET http://api.songwich.com/v0.4/stations
--header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
```

JSON Response:

```
{
  "status": "0",
  "message": "Success",
  "stations": [
    {
      "stationId": "527bb6fedb88f7f96cf39729",
      "stationName": "John Smith FM",
      "scrobblers": [
        "527bb692db88f7f96cf39722"
      ],
      "numberSubscribers": 0,
      "imageUrl": "http://example.com/image.jpg",
      "verified": "false",
      "active": "false"
    },
    {
      "stationId": "527bb739db88f7f96cf3972c",
      "stationName": "Test Group FM",
      "groupName": "Test Group",
      "scrobblers": [
        "527bb680db88f7f96cf39721",
        "527bb692db88f7f96cf39722"
      ],
      "numberSubscribers": 2,
      "imageUrl": "http://example.com/image.jpg",
      "verified": "false",
      "active": "true"
    }
  ]
}
```

Gets detailed information on a specific station with includeScrobblers=true:

```
curl
--request GET http://api.songwich.com/v0.4/stations?stationId=
52682b892c043cb13d63e7b7&includeScrobblers=true
--header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
```

JSON Response (of an active station):

```
{
  "status": "0",
  "message": "Success",
  "station": {
    "stationId": "52682b892c043cb13d63e7b7",
    "stationName": "Test Group FM",
    "activeScrobblers": [
```

```

    {
      "name": "David Johnes",
      "imageUrl": "http://image.com/image2.jpg",
      "userId": "52aa59ea4d47aa76ca6eb363",
      "verified": "false"
    },
    {
      "name": "John Smith",
      "imageUrl": "http://image.com/image.jpg",
      "userId": "52aa59bc4d47aa76ca6eb362",
      "verified": "false"
    }
  ],
  "numberSubscribers": 2,
  "nowPlaying": {
    "trackTitle": "Supersoaker",
    "artistsNames": [
      "Kings of Leon"
    ],
    "idForFeedback": "52682c232c043cb13d63e7b9",
    "recentScrobbles": [
      {
        "name": "John Smith",
        "userId": "52682a672c043cb13d63e7af"
      }
    ]
  },
  "lookAhead": {
    "trackTitle": "Get Lucky",
    "artistsNames": [
      "Daft Punk",
      "Pharrell Williams"
    ],
    "idForFeedback": "52682c292c043cb13d63e7ba",
    "recentScrobbles": [
      {
        "name": "John Smith",
        "userId": "52682a672c043cb13d63e7af"
      },
      {
        "name": "David Jones",
        "userId": "52682a7b2c043cb13d63e7b0"
      }
    ]
  },
  "imageUrl": "http://example.com/image.jpg",
  "active": "true"
}

```

Gets detailed information on an inactive station. JSON response:

```

{
  "status": "0",
  "message": "Success",
  "station": {

```

```

    "stationId": "527bb6fedb88f7f96cf39729",
    "stationName": "John Smith FM",
    "scrobblersIds": [
        "527bb692db88f7f96cf39722"
    ],
    "numberSubscribers": 1,
    "stationReadiness": "0.82",
    "imageUrl": "http://example.com/image.jpg",
    "verified": "false",
    "active": "false"
}
}

```

**PUT** **/v0.4/stations/<stationId>**

Updates a station's data.

Parameter	Required?	Values	Location
X-Songwich.devAuthToken	yes	3bde6fba-1ae5-4d7f-8000-f2aba160b71a	header
X-Songwich.userAuthToken	yes	0b115241-0ff4-4a08-8ecc-d9745beff24a	header
stationId	yes	525c4bc045691ceff24838db	URL
stationName	no	Test Group FM	body
imageUrl	no	http://example.com/newimage.jpg	body
description	no	This is a description	body

Updates a station's imageUrl:

```

curl
--request PUT http://api.songwich.com/v0.4/stations/525c4bc045691ceff24838db
--header 'Content-Type: application/json'
--header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
--header 'X-Songwich.userAuthToken: f0f514d5-9eb2-41b1-81dd-db39ad1ed19c'
--data '
{
    "imageUrl": "http://example.com/newimage.jpg"
}
'

```

JSON response:

```

{
    "status": "0",
    "message": "Success",
    "station": {
        "stationId": "525c4bc045691ceff24838db",
        "stationName": "Test Group FM",
        "active": "true"
        "scrobblersIds": [

```

```

        "525c4d2645691ceff24838dc",
        "525c379145691ceff24838d1"
    ],
    "imageUrl": "http://example.com/newimage.jpg"
}

```

## PUT /v0.4/stations/<stationId>/markAsVerified

Marks a station as verified.

Parameter	Required?	Values	Location
X-Songwich.devAuthToken	yes	3bde6fba-1ae5-4d7f-8000-f2aba160b71a	header
X-Songwich.userAuthToken	yes	0b115241-0ff4-4a08-8ecc-d9745beff24a	header
stationId	yes	525c4bc045691ceff24838db	URL

Example request:

```

curl
  --request PUT
  http://api.songwich.com/v0.4/stations/52af5e004d47437b10fad238/markAsVerified
  --header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
  --header 'X-Songwich.userAuthToken: f0f514d5-9eb2-41b1-81dd-db39ad1ed19c'

```

JSON response:

```

{
  "status": "0",
  "message": "Success",
  "station": {
    "stationName": "John Smith FM",
    "scrobblers": [
      "52af5dde4d47437b10fad237"
    ],
    "imageUrl": "http://images.com/image.jpg",
    "verified": "true",
    "stationId": "52af5e004d47437b10fad238",
    "active": "false"
  }
}

```

## PUT /v0.4/stations/<stationId>/addScrobblers

Adds new scrobblers to a given station.

If the station was previously inactive and the new scrobblers make it ready then it is automatically

activated. If `subscribeScrobbles=true`, the station's scrobbles will automatically have a subscription.

Parameter	Required?	Values	Location
X-Songwich.devAuthToken	yes	3bde6fba-1ae5-4d7f-8000-f2aba160b71a	header
X-Songwich.userAuthToken	yes	0b115241-0ff4-4a08-8ecc-d9745beff24a	header
stationId	yes	525c4bc045691ceff24838db	URL
subscribeScrobbles	no	“true” or “false”	URL
scrobbleIds	yes	525c4d2645691ceff24838dc	body

Adds two scrobbles to a station:

```
curl
--request PUT
http://api.songwich.com/v0.4/stations/525c4bc045691ceff24838db/addScrobbles
--header 'Content-Type: application/json'
--header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
--header 'X-Songwich.userAuthToken: f0f514d5-9eb2-41b1-81dd-db39ad1ed19c'
--data '
{
  "scrobbleIds": ["525c3ecb45691ceff24838d2", "525c4d2645691ceff24838dc"]
}
'
```

JSON response:

```
{
  "status": "0",
  "message": "Success",
  "station": {
    "stationId": "525c4bc045691ceff24838db",
    "stationName": "Test Group FM",
    "active": "true",
    "scrobbleIds": [
      "525c3ecb45691ceff24838d2",
      "525c4d2645691ceff24838dc",
      "525c379145691ceff24838d1"
    ],
    "imageUrl": "http://example.com/image.jpg"
  }
}
```

**PUT** `/v0.4/stations/<stationId>/removeScrobbles`

Mark a station's current scrobble(s) as inactive.

If the station was previously active and the removed scrobbles reduce its `stationReadiness` to less

than 1.00 than it is automatically deactivated.

Parameter	Required?	Values	Location
X-Songwich.devAuthToken	yes	3bde6fba-1ae5-4d7f-8000-f2aba160b71a	header
X-Songwich.userAuthToken	yes	0b115241-0ff4-4a08-8ecc-d9745beff24a	header
stationId	yes	525c4bc045691ceff24838db	URL
scrobblersIds	yes	525c3ecb45691ceff24838d2	body

Removes a scrobber from a station:

```
curl
--request PUT
http://api.songwich.com/v0.4/stations/525c4bc045691ceff24838db/removeScrobblers
--header 'Content-Type: application/json'
--header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
--header 'X-Songwich.userAuthToken: f0f514d5-9eb2-41b1-81dd-db39ad1ed19c'
--data '
{
  "scrobblersIds": "525c3ecb45691ceff24838d2"
}
'
```

JSON response:

```
{
  "status": "0",
  "message": "Success",
  "station": {
    "stationId": "525c4bc045691ceff24838db",
    "stationName": "Test Group FM",

    "active": "false",
    "scrobblersIds": [
      "525c4d2645691ceff24838dc",
      "525c379145691ceff24838d1"
    ],
    "imageUrl": "http://example.com/image.jpg"
  }
}
```

**POST** /v0.4/nextSong

Runs the songlist engine for stations. A look ahead song is always given for allowing for caching.

This method returns an error if the station is not active.

Parameter	Required?	Values	Location
-----------	-----------	--------	----------

X-Songwich.devAuthToken	yes	3bde6fba-1ae5-4d7f-8000-f2aba160b71a	header
stationId	yes	525c548d4569d98908dc7fa7	body

Asks what the next song that should be played for a given station:

```
curl
--request POST http://api.songwich.com/v0.4/nextSong
--header 'Content-Type: application/json'
--header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
--data '
{
  "stationId": "52682b892c043cb13d63e7b7"
}
'
```

JSON response:

```
{
  "status": "0",
  "message": "Success",
  "station": {
    "stationId": "52682b892c043cb13d63e7b7",
    "nowPlaying": {
      "trackTitle": "Supersoaker",
      "artistsNames": [
        "Kings of Leon"
      ],
      "idForFeedback": "52682c232c043cb13d63e7b9",
      "recentScrobbles": [
        {
          "name": "John Smith",
          "userId": "52682a672c043cb13d63e7af"
        }
      ]
    },
    "lookAhead": {
      "trackTitle": "Get Lucky",
      "artistsNames": [
        "Daft Punk",
        "Pharrell Williams"
      ],
      "idForFeedback": "52682c292c043cb13d63e7ba",
      "recentScrobbles": [
        {
          "name": "John Smith",
          "userId": "52682a672c043cb13d63e7af"
        },
        {
          "name": "David Jones",
          "userId": "52682a7b2c043cb13d63e7b0"
        }
      ]
    }
  }
}
```





# Song Feedback API

API methods related to song feedback.

**POST** /v0.4/songFeedback

Allows for posting song feedback like “thumbs-up”, “thumbs-down”, and “star”.

Parameter	Required?	Values	Location
X-Songwich.devAuthToken	yes	3bde6fba-1ae5-4d7f-8000-f2aba160b71a	header
X-Songwich.userAuthToken	yes	0b115241-0ff4-4a08-8ecc-d9745beff24a	header
idForFeedback	yes	526a6c852c04b6128cb84757	body
feedbackType	yes	“thumbs-up” or “thumbs-down” or “star” or “skip”	body

Posts a thumbs-up feedback:

```
curl
--request POST http://api.songwich.com/v0.4/songFeedback
--header 'Content-Type: application/json'
--header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
--header 'X-Songwich.userAuthToken: 0b115241-0ff4-4a08-8ecc-d9745beff24a'
--data '
{
  "idForFeedback": "526a6c852c04b6128cb84757",
  "feedbackType": "thumbs-up"
},
'
```

JSON response:

```
{
  "status": "0",
  "message": "Success",
  "songFeedback": {
    "idForFeedback": "526a6c852c04b6128cb84757",
    "feedbackType": "thumbs-up",
    "song": {
      "trackTitle": "Supersoaker",
      "artistsNames": [
        "Kings of Leon"
      ]
    },
    "userId": "5259e16d92e667731a1bd6d8"
  }
}
```

**GET** /v0.4/starredSongs/<userId>

Gets songs that were starred by a given user.

Parameter	Required?	Values	Location
X-Songwich.devAuthToken	yes	3bde6fba-1ae5-4d7f-8000-f2aba160b71a	header
X-Songwich.userAuthToken	yes	0b115241-0ff4-4a08-8ecc-d9745beff24a	header
userId	yes	5259e16d92e667731a1bd6d8	URL
results	no	Integer (default=30, max=100)	URL param
since	no	ObjectId (exclusive)	URL param
sinceInclusive	no	ObjectId (inclusive)	URL param
until	no	ObjectId (exclusive)	URL param
untilInclusive	no	ObjectId (inclusive)	URL param

Gets the last 2 starred songs:

```
curl
--request GET
http://api.songwich.com/v0.4/starredSongs/527d3cd52c04acf8c1d084b0?results=2
--header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
--header 'X-Songwich.userAuthToken: 0b115241-0ff4-4a08-8ecc-d9745beff24a'
```

JSON Response:

```
{
  "status": "0",
  "message": "Success",
  "starredSongsResult": {
    "userId": "527d3cd52c04acf8c1d084b0",
    "starredSongs": [
      {
        "trackTitle": "Supersoaker",
        "artistsNames": [
          "Kings of Leon"
        ],
        "idForFeedback": "527d3d202c04acf8c1d084b7"
      },
      {
        "trackTitle": "Do I Want to Know?",
        "artistsNames": [
          "Arctic Monkeys"
        ],
        "idForFeedback": "527d3d0d2c04acf8c1d084b4"
      }
    ]
  }
}
```

```

    },
    "paging": {
      "newer":
"http://api.songwich.com/v0.4/starredSongs/527d3cd52c04acf8c1d084b0?
results=2&since=527d3d202c04acf8c1d084b7",
      "older":
"http://api.songwich.com/v0.4/starredSongs/527d3cd52c04acf8c1d084b0?
results=2&until=527d3d0d2c04acf8c1d084b4"
    }
  }
}

```

## GET /v0.4/isSongStarred/<userId>

Gets to know if a song has been starred by a given user. So far, this method is case-sensitive.

artistsNames with multiple artists must be comma-separated. It also needs to be in the same order as previously sent by the API.

Parameter	Required?	Values	Location
X-Songwich.devAuthToken	yes	3bde6fba-1ae5-4d7f-8000-f2aba160b71a	header
X-Songwich.userAuthToken	yes	0b115241-0ff4-4a08-8ecc-d9745beff24a	header
userId	yes	5259e16d92e667731a1bd6d8	URL
songTitle	yes	Je Veux	URL param
artistsNames	yes	Daft Punk, Pharrell Williams (separated by comma)	URL param
albumTitle	no	Ensemble	URL param

Gets starred songs:

```

curl
--request GET
http://api.songwich.com/v0.4/isSongStarred/527d54852c04acf8c1d084cf?
songTitle=Get Lucky&artistsNames=Daft Punk, Pharrell Williams
--header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
--header 'X-Songwich.devAuthToken: 0b115241-0ff4-4a08-8ecc-d9745beff24a'

```

Example of a JSON positive response:

```

{
  "status": "0",
  "message": "Success",
  "isSongStarredResult": {
    "userId": "527d54852c04acf8c1d084cf",
    "song": {
      "trackTitle": "Get Lucky",
      "artistsNames": [

```

```

        "Daft Punk",
        "Pharrell Williams"
    ]
},
"isStarred": "true",
"idForFeedback": "527d54d72c04acf8c1d084d9"
}
}

```

Example of a JSON negative response:

```

{
  "status": "0",
  "message": "Success",
  "isSongStarredResult": {
    "userId": "527d48bb2c04acf8c1d084b9",
    "song": {
      "trackTitle": "Do I Want to Know?",
      "artistsNames": [
        "Arctic Monkeys"
      ]
    },
    "isStarred": "false"
  }
}

```

**DELETE** /v0.4/songFeedback/<idForFeedback>/<feedbackType>

Deletes a songFeedback.

Parameter	Required?	Values	Location
X-Songwich.devAuthToken	yes	3bde6fba-1ae5-4d7f-8000-f2aba160b71a	header
X-Songwich.userAuthToken	yes	0b115241-0ff4-4a08-8ecc-d9745beff24a	header
idForFeedback	yes	526d915b2c04dfd970fd392b	URL
feedbackType	yes	“star” or “thumbs-up” or “thumbs-down”	URL

Deletes a songFeedback:

```

curl
  --request DELETE
  http://api.songwich.com/v0.4/songFeedback/526d915b2c04dfd970fd392b/star
  --header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
  --header 'X-Songwich.userAuthToken: f0f514d5-9eb2-41b1-81dd-db39ad1ed19c'

```

JSON Response:

```
{  
  "status": "0",  
  "message": "Success"  
}
```

# Subscriptions API

Basic API methods related to station subscriptions.

**POST** /v0.4/subscriptions

Allows for posting subscriptions.

Parameter	Required?	Values	Location
X-Songwich.devAuthToken	yes	3bde6fba-1ae5-4d7f-8000-f2aba160b71a	header
X-Songwich.userAuthToken	yes	0b115241-0ff4-4a08-8ecc-d9745beff24a	header
userId	yes	52a7d0d14d4718fd20d08f26	body
stationId	yes	52a7d0f14d4718fd20d08f27	body

Posts a station subscription:

```
curl
--request POST http://api.songwich.com/v0.4/subscriptions
--header 'Content-Type: application/json'
--header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
--header 'X-Songwich.userAuthToken: 0b115241-0ff4-4a08-8ecc-d9745beff24a'
--data '
{
  "userId": "52a8c0724d474dee367c40fe",
  "stationId": "52a8c16d4d474dee367c4100"
}
```

JSON response:

```
{
  "status": "0",
  "message": "Success",
  "subscription": {
    "id": "52a8c1c64d474dee367c4102",
    "userId": "52a8c0724d474dee367c40fe",
    "station": {
      "stationId": "52a8c16d4d474dee367c4100",
      "stationName": "John Smith FM",
      "imageUrl": "http://images.com/image.jpg"
    }
  }
}
```

**GET** /v0.4/subscriptions/<userId>

Gets active subscriptions by a given user.

Parameter	Required?	Values	Location
X-Songwich.devAuthToken	yes	3bde6fba-1ae5-4d7f-8000-f2aba160b71a	header
userId	yes	52a7d0d14d4718fd20d08f26	URL

Gets subscriptions:

```
curl
--request GET
http://api.songwich.com/v0.4/subscriptions/52a7d0d14d4718fd20d08f26
--header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
```

JSON Response:

```
{
  "status": "0",
  "message": "Success",
  "subscriptions": [
    {
      "id": "52a8c3ed4d474dee367c4104",
      "station": {
        "stationId": "52a8c3c94d474dee367c4103",
        "stationName": "David Jones FM",
        "imageUrl": "http://images.com/image.jpg"
      }
    },
    {
      "id": "52a8c1a04d474dee367c4101",
      "station": {
        "stationId": "52a8c16d4d474dee367c4100",
        "stationName": "John Smith FM",
        "imageUrl": "http://images.com/image.jpg"
      }
    }
  ]
}
```

**PUT** /v0.4/subscriptions/<subscriptionId>/end

End an active subscription.

Parameter	Required?	Values	Location
X-Songwich.devAuthToken	yes	3bde6fba-1ae5-4d7f-8000-f2aba160b71a	header
X-Songwich.userAuthToken	yes	0b115241-0ff4-4a08-8ecc-d9745beff24a	header

subscriptionId	yes	52a7d1aa4d4718fd20d08f2a	URL
----------------	-----	--------------------------	-----

Ends a subscription:

```
curl
--request PUT
http://api.songwich.com/v0.4/subscriptions/52a7d1aa4d4718fd20d08f2a/end
--header 'X-Songwich.devAuthToken: 3bde6fba-1ae5-4d7f-8000-f2aba160b71a'
--header 'X-Songwich.userAuthToken: f0f514d5-9eb2-41b1-81dd-db39ad1ed19c'
```

JSON response:

```
{
  "status": "0",
  "message": "Success",
  "subscription": {
    "id": "52a7d1aa4d4718fd20d08f2a"
  }
}
```