
itunes Documentation

Release 1.0.1

Jonathan Nappi

July 02, 2016

1	itunes.search module	3
2	itunes.base module	7
3	Indices and tables	9
	Python Module Index	11

Release v1.0.1.

The pyitunes module provides a python interface for searching the iTunes Store.

Contents:

itunes.search module

This module contains utilities for performing a variety of different searches against the iTunes store API

`itunes.search.search_track(query, limit=100, offset=0, order=None, store='US')`

Search for a track resource

Parameters

- **query** – The query to perform against the iTunes store API
- **limit** – The max number of responses to return. Default: 100
- **offset** – The offset into a collection of resources. Default: 0
- **order** – The key to order a collection of resources by. Possible values are 'rank' or 'popular.' Default: None
- **store** – The iTunes store localization to search against. Default: *COUNTRY*

`itunes.search.search_album(query, limit=100, offset=0, order=None, store='US')`

Search for an album resource

Parameters

- **query** – The query to perform against the iTunes store API
- **limit** – The max number of responses to return. Default: 100
- **offset** – The offset into a collection of resources. Default: 0
- **order** – The key to order a collection of resources by. Possible values are 'rank' or 'popular.' Default: None
- **store** – The iTunes store localization to search against. Default: *COUNTRY*

`itunes.search.search_artist(query, limit=100, offset=0, order=None, store='US')`

Search for an artist resource

Parameters

- **query** – The query to perform against the iTunes store API
- **limit** – The max number of responses to return. Default: 100
- **offset** – The offset into a collection of resources. Default: 0
- **order** – The key to order a collection of resources by. Possible values are 'rank' or 'popular.' Default: None
- **store** – The iTunes store localization to search against. Default: *COUNTRY*

`itunes.search.search_app(query, limit=100, offset=0, order=None, store='US')`

Search for an app resource

Parameters

- **query** – The query to perform against the iTunes store API
- **limit** – The max number of responses to return. Default: 100
- **offset** – The offset into a collection of resources. Default: 0
- **order** – The key to order a collection of resources by. Possible values are 'rank' or 'popular.' Default: None
- **store** – The iTunes store localization to search against. Default: *COUNTRY*

`itunes.search.search_episode(query, limit=100, offset=0, order=None, store='US')`

Search for a TV Episode resource

Parameters

- **query** – The query to perform against the iTunes store API
- **limit** – The max number of responses to return. Default: 100
- **offset** – The offset into a collection of resources. Default: 0
- **order** – The key to order a collection of resources by. Possible values are 'rank' or 'popular.' Default: None
- **store** – The iTunes store localization to search against. Default: *COUNTRY*

`itunes.search.search_season(query, limit=100, offset=0, order=None, store='US')`

Search for a TV Season resource

Parameters

- **query** – The query to perform against the iTunes store API
- **limit** – The max number of responses to return. Default: 100
- **offset** – The offset into a collection of resources. Default: 0
- **order** – The key to order a collection of resources by. Possible values are 'rank' or 'popular.' Default: None
- **store** – The iTunes store localization to search against. Default: *COUNTRY*

`itunes.search.search(query, media='all', limit=100, offset=0, order=None, store='US')`

Search for any type of resource

Parameters

- **query** – The query to perform against the iTunes store API
- **limit** – The max number of responses to return. Default: 100
- **offset** – The offset into a collection of resources. Default: 0
- **order** – The key to order a collection of resources by. Possible values are 'rank' or 'popular.' Default: None
- **store** – The iTunes store localization to search against. Default: *COUNTRY*

`itunes.search.search_movie(query, limit=100, offset=0, order=None, store='US')`

Search for a Movie resource

Parameters

- **query** – The query to perform against the iTunes store API
- **limit** – The max number of responses to return. Default: 100
- **offset** – The offset into a collection of resources. Default: 0
- **order** – The key to order a collection of resources by. Possible values are ‘rank’ or ‘popular.’ Default: None
- **store** – The iTunes store localization to search against. Default: *COUNTRY*

class `itunes.search.Search`(*query*, *country*='US', *media*='all', *entity*=None, *attribute*=None, *offset*=0, *limit*=50, *order*=None, *lang*='en_us', *version*='2', *explicit*='Yes')

Bases: `itunes.base.BaseObject`

Search iTunes Store for a variety of different resource types

resource = 'search'

1.1 Example Usage

The `itunes.search` module is made up of a handful of different utility functions for performing a variety of different searches against the iTunes Store API.

1.1.1 Artist

Searching for an artist is as simple as passing an artist name to the `search_artist` function

```
>>> import itunes
>>> results = itunes.search_artist('Frank Sinatra')
>>> frank = results[0]
>>> frank
... '<Artist>: Frank Sinatra'
```

1.1.2 Albums

To find an artist's albums, you can either use the `search_album` or access all of an artists albums via an `Artist` instance. Continuing from our above example we could do something similar to:

```
>>> for album in frank.get_albums():
...     print(album)
... <Collection>: Nothing But the Best (Remastered)
... <Collection>: Ultimate Sinatra
... <Collection>: Christmas With the Rat Pack
... <Collection>: Christmas
... <Collection>: A Jolly Christmas from Frank Sinatra (50th Anniversary)
... <Collection>: Sinatra At the Sands
... <Collection>: Classic Sinatra: His Great Performances 1953-1960
... ...
```

1.1.3 Apps

Apps available from the iTunes store are also available for searching:

```
>>> app = itunes.search_app('angry birds')[0]
>>> app.ratings
... {'avg': {'all': 4.5,
            'current': 4.0},
     'num': {'all': 823399,
            'current': 504}}
```

1.1.4 TV Shows

TV Shows are also searchable:

```
>>> s1 = itunes.search_season('Family Guy Season 1')[0]
>>> s1.release_date
... datetime.datetime(1999, 1, 31, 8, 0)
>>> s1.artwork
... {'60': 'http://is1.mzstatic.com/image/thumb/Music3/v4/60/15/b2/6015b219-e4cb-2a93-0369-c75b5e06e...',
     '600': 'http://is1.mzstatic.com/image/thumb/Music3/v4/60/15/b2/6015b219-e4cb-2a93-0369-c75b5e06e...'}
>>> s1.genre
... 'Comedy'
```

itunes.base module

This module contains base data models for the rest of the iTunes store API

`itunes.base.TS_FORMAT = '%Y-%m-%dT%H:%M:%S'`
iTunes API Timestamp format

`itunes.base.SESSION = <requests.sessions.Session object>`
Globally accessible cache-enabled requests session

exception `itunes.base.ITunesException (message)`
Bases: `exceptions.Exception`
Base iTunes request exception

class `itunes.base.BaseObject (**kwargs)`
Bases: `object`
Base object for representing an iTunes request

get ()
Execute an HTTP GET against the iTunes API and construct an appropriate assortment of *Resource*'s based on the response

resource = None

url
The url pointing at the resource defined by the implementing object

class `itunes.base.Resource (id)`
Bases: `object`
Base class for the various types of Resources returned by the iTunes Store API

get_album ()
Returns the first album associated with this *Resource*

get_albums (*limit=200*)
Returns the albums associated with this *Resource*

get_tracks (*limit=500*)
Returns the tracks associated with this *Resource*

release_date
Accessor for a `datetime.datetime` representation of the release date for this *Resource*

exception `itunes.base.NoResultsFoundException`
Bases: `itunes.base.ITunesException`
iTunes error for when no results are returned from a Lookup

class `itunes.base.Lookup` (*id*, *entity=None*, *limit=50*)

Bases: `itunes.base.BaseObject`

A data model for an individual resource look up against iTunes

resource = 'lookup'

class `itunes.base.Artist` (*id*)

Bases: `itunes.base.Resource`

The Artist `Resource` represents an iTunes artist

class `itunes.base.Album` (*id*)

Bases: `itunes.base.Resource`

The Album `Resource` represents an Album (or collection of single resources) of other resource types

class `itunes.base.Track` (*id*)

Bases: `itunes.base.Resource`

The Track `Resource` represents a single track from the iTunes store

class `itunes.base.Audiobook` (*id*)

Bases: `itunes.base.Album`

The Audiobook `Resource` represents an iTunes Audiobook

class `itunes.base.Software` (**args*, ***kwargs*)

Bases: `itunes.base.Track`

The Software `Resource` represents an iTunes App resource

class `itunes.base.TVEpisode` (*id*)

Bases: `itunes.base.Track`

The TVEpisode `Resource` represents a track type that represents a single TV Episode.

`itunes.base.lookup` (*id*)

Perform an individual `Lookup` on a single resource in the iTunes Store API

Indices and tables

- `genindex`
- `modindex`
- `search`

i

`itunes.base`, 7
`itunes.search`, 3

A

Album (class in `itunes.base`), 8
Artist (class in `itunes.base`), 8
Audiobook (class in `itunes.base`), 8

B

BaseObject (class in `itunes.base`), 7

G

`get()` (`itunes.base.BaseObject` method), 7
`get_album()` (`itunes.base.Resource` method), 7
`get_albums()` (`itunes.base.Resource` method), 7
`get_tracks()` (`itunes.base.Resource` method), 7

I

`itunes.base` (module), 7
`itunes.search` (module), 3
`ITunesException`, 7

L

Lookup (class in `itunes.base`), 7
`lookup()` (in module `itunes.base`), 8

N

`NoResultsFoundException`, 7

R

`release_date` (`itunes.base.Resource` attribute), 7
`Resource` (class in `itunes.base`), 7
`resource` (`itunes.base.BaseObject` attribute), 7
`resource` (`itunes.base.Lookup` attribute), 8
`resource` (`itunes.search.Search` attribute), 5

S

Search (class in `itunes.search`), 5
`search()` (in module `itunes.search`), 4
`search_album()` (in module `itunes.search`), 3
`search_app()` (in module `itunes.search`), 3
`search_artist()` (in module `itunes.search`), 3

`search_episode()` (in module `itunes.search`), 4
`search_movie()` (in module `itunes.search`), 4
`search_season()` (in module `itunes.search`), 4
`search_track()` (in module `itunes.search`), 3
`SESSION` (in module `itunes.base`), 7
`Software` (class in `itunes.base`), 8

T

Track (class in `itunes.base`), 8
`TS_FORMAT` (in module `itunes.base`), 7
`TVEpisode` (class in `itunes.base`), 8

U

`url` (`itunes.base.BaseObject` attribute), 7