

MyPySpace Tutorial

MyPySpace allows you to crawl the MySpace artist network, generating RDF and creating local copies of media.

Download

Download the current source of the MyPySpace package on <http://mypyspace.svn.sourceforge.net/viewvc/mypyspace/>

Installation

- You need to have Python 2.4 or greater (<http://python.org/download/>)
- Get Easy Install
<http://peak.telecommunity.com/DevCenter/EasyInstall>

- Use Easy Install to install rdflib:

```
$ easy_install -U 'rdflib<3a'
```

- Use Easy Install to install mutagen:

```
$ easy_install mutagen
```

- Get mopy (part of motools) from <http://sourceforge.net/projects/motools>
- Build mopy by running genpy.py:

```
$ cd motools/mopy
$ python genpy.py
```

- Create symlink to mopy inside of MyPySpace:

```
$ cd ../../mypyspace/musicGrabber/trunk
$ ln -s ../../../../motools/mopy/mopy mopy
```

Collect media from a single artist

The simplest task to accomplish with MyPySpace is to collect media from the MySpace artist network.

To start with, we will collect media from a single artist and stop:

```
$ python musicGrabber.py -v -d -e 0 -a 5875721 -p
path/to/music/directory/
```

This will download the audio files from the MySpace artist page found at <http://www.myspace.com/index.cfm?fuseaction=user.viewProfile&friendID=5875721> and then exit.

Collect media from an artist and her top friends

Using a similar command we can collect the media for an artist and the artists associated “top friends”:

```
$ python musicGrabber.py -v -d -e 1 -a 5875721 -p
path/to/music/directory/
```

This will gather all media associated with the target artist and her top friends. The argument given with `-e` can be increased to recursively crawl more top friends links, up to 6.

Publishing RDF descriptions of artist pages

The second function accessible via `musicGrabber.py` is the ability to translate MySpace artist pages into RDF using the Music Ontology² and FOAF³:

```
$ python musicGrabber.py -v -d -a 5875721 -f path/to/RDF/directory/
```

or if you also want to collect media files:

```
$ python musicGrabber.py -v -d -a 5875721 -f path/to/RDF/directory/ -p
path/to/music/directory/
```

It is important to note that the program will create two RDF files for each artist: one describing the artist's friend relationship, the other describing the artist's media.

This will use the default collection behaviour of the specified artist and her “top friends”. If you desire further crawling in one command add `-e n`, where `n` is the number of “top friends” levels.

Further functionalities

There are two other ways of defining the target artist:

- `-u [--url] http://myspace.com/targetArtist`
- `-m [--random] n`
 where `0` to `n` represent the range of possible IDs that can be randomly picked.
 As UIDs are randomly selected they are verified to be artists. If the UID is not an artist, another will be randomly selected.

¹“Top friends” are socially linked artists that have been selected by the current artist as being more significant.

² <http://musicontology.com/>

³ <http://www.foaf-project.org/>