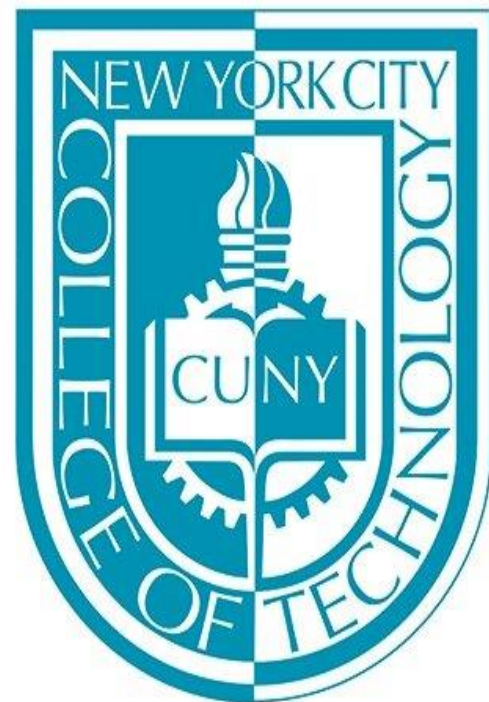


December 7, 2016



FINAL PROJECT: MUSIC SERVER

Presented by: Elizabeth Ferreira & Matthew Visconti

EMT 2390L OPERATING SYSTEMS LAB

PROF: HAMILTON

TABLE OF CONTENT

INTRODUCTION.....	3
Raspberry Pi 3	4
Mopidy.....	5
Mopidy- Spotify	6
Moped	6
Steps to Install the Music Server.....	7
Troubleshooting.....	9

[Click to follow link](#)

INTRODUCTION

The main goal of this project is to have a music server that will allow us to connect to our favorite music on a more customized way. We will be using the Raspberry Pi 3 as a music server. This project could be used as a music server in our living room or bedroom and just have it connected to speakers (it could be done plugged in directly or Bluetooth) and have a touchscreen and it is an embedded system that will specifically be designed to be our music stream and music player. To make this happen we will be installing the following packages on the raspberry: Mopidy, Mopidy-Spotify and Moped.

MUSIC SERVER

Raspberry Pi 3



The Raspberry Pi is a small wallet sized computer that plugs into virtually any monitor/TV that has an HDMI port. All that is needed to run this little computer is a display, mouse, keyboard, 5v micro-USB, pre-installed version on Raspbian on a microSD card and internet connection (optional). The Raspberry Pi is mainly used in electronics projects, and as a basic desktop PC. We are using this Raspberry Pi as a music streamer and player that will be using Mopidy, Mopidy-Spotify (backend extension) and Moped.

Mopidy

```
pi@raspberrypi:~ $ mopidy
INFO     Starting Mopidy 2.0.1
INFO     Loading config from builtin defaults
INFO     Loading config from /home/pi/.config/mopidy/mopidy.conf
INFO     Loading config from command line options
INFO     Enabled extensions: spotify, mpd, http, moped, stream, m3u, softwaremixer, file, musicbox_webclient
, local
INFO     Disabled extensions: none
INFO     Starting Mopidy mixer: SoftwareMixer
INFO     Starting Mopidy audio
INFO     Starting Mopidy backends: StreamBackend, M3UBackend, FileBackend, LocalBackend, SpotifyBackend
INFO     Audio output set to "autoaudiosink"
INFO     Loaded 7 local tracks using json
INFO     Starting Mopidy core
INFO     Logged in to Spotify in offline mode
INFO     Logged in to Spotify in online mode
INFO     Starting Mopidy frontends: MpdFrontend, HttpFrontend
INFO     MPD server running at [::ffff:127.0.0.1]:6600
INFO     HTTP server running at [::ffff:127.0.0.1]:6680
INFO     Starting GLib mainloop
```

Mopidy is an extensible music server written in Python and plays music from a local disk, Spotify, SoundCloud, Google Play Music, and more. Mopidy is a python application that runs in a terminal or in the background on a linux or Mac OS that have network connectivity and audio output. You are also able to edit the playlist from any phone, tablet, or computer sing a range of different MPD and web clients. Mopidy is a simple server that runs in the terminal or in the background on Linux computer or Macs. Anybody can connect to your Mopidy server using their favorite MPD or webclient. Mopidy is also hackable and can be changed within a development environment that will not affect the original Mopidy that is currently installed. The virtualenv will wall off Mopidy and its dependencies from the rest of your system. All development and installation of Python dependencies, versions of Mopidy, and extensions are done inside the virtualenv. This way your regular Mopidy install, which you set up in the first step, is unaffected by your hacking and will always be working.

A virtual environment is a tool to keep the dependencies required by different projects in separate places. Also, is a tool to create isolated python environments that creates folders which contain all the necessary executables to use the packages that python projects would need.

Mopidy- Spotify

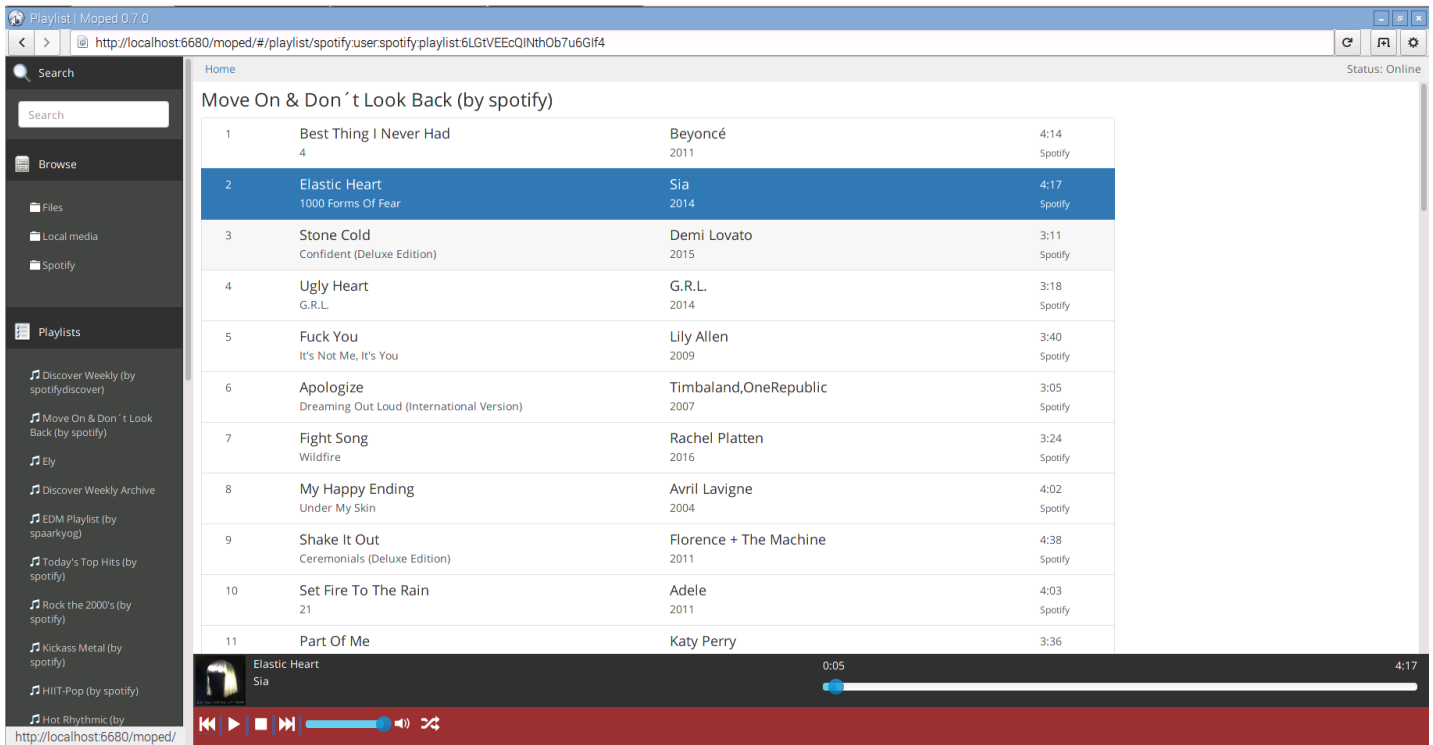
A Mopidy backend extension for playing music from Spotify. A Spotify Premium subscription is needed in order from Mopidy-Spotify to work. A non-Facebook Spotify username will also need to be provided in order for it to work, and if the account was created using Facebook then you need to create a “device password” to be able to use Mopidy-Spotify. Before starting Mopidy and using Mopidy-Spotify we need to add our Spotify Premium username and password to our Mopidy configuration file.



Moped

A web extension for the Mopidy extensible music server that provides the user with an interface. It's created using a combination of both HTML5 & JavaScript and can be easily accessed on all web browsers when you type in your Mopidy server (<http://localhost:6680/moped>) into the URL. The Mopidy HTTP frontend uses Web Sockets. Most modern browsers support this but not the default Android browser (4.3 and older). To use Moped on Android you have to use a different browser like Firefox or Chrome. Searching radio streams is still experimental.

The next image is the graphical interface of Moped to manage and control mopidy-spotify.



We were able to create a link from the terminal that will take us to Moped by using a bash script

```

GNU nano 2.2.6          File: url.sh
! /bin/bash
node server.js;
xdg-open http://localhost:6680/moped

```

Steps to Install the Music Server

1. Downloaded the stable version of Jessie that works with Mopidy.
2. Install Python 2.7 “sudo apt-get install build-essential python-dev python-pip”
3. Install Gstreamer “sudo apt-get install python-gst-1.0 \
 gir1.2-gstreamer-1.0 gir1.2-gst-plugins-base-1.0 \
 gstreamer1.0-plugins-good gstreamer1.0-plugins-ugly \
 gstreamer1.0-tools”

4. Install the latest release of Mopidy from source “sudo pip install -U mopidy”
5. Install backend extensions for Mopidy “pip search mopidy”
6. Install backend extension Mopidy-Spotify from apt.mopidy.com “sudo apt-get install mopidy-spotify”
7. You may also install Mopidy-Spotify dependencies from Mopidy and install the package from PyPI “pip install Mopidy-Spotify”
8. Edit the Mopidy configuration file “mopidy config” by adding our Spotify username and password.
[spotify]
username = Elizabeth
password = secret
9. Run Mopidy to check if the backend extensions are working and everything is logging.
10. Update the local library on our RPi “mopidy local scan”
11. Refresh the library “curl -d '{"jsonrpc": "2.0", "id": 1, "method": "core.library.refresh"}'
<http://localhost:6680/mopidy/rpc>”
12. Service management with system “sudo systemctl enable mopidy”
13. Install moped from PyPI “sudo pip install Mopidy-Moped”
14. Start moped while Mopidy is running in the background by typing in the HTTP
“<http://localhost:6680/moped>”
15. Make sure that Mopidy sees Moped is running and that web client shows both the local music and Spotify playlists.
16. Create a bash script to be able to run the Mopidy-Spotify through Moped by doing this:
17. “sudo nano url.sh”
#!/bin/bash
Wget -O - -q -t 1 <http://localhost:6680/moped>
18. Run mopidy and run url.sh to link and it's ready to be used.

Troubleshooting

```
pi@raspberrypi:~ $ mopidy
INFO     Starting Mopidy 2.0.1
INFO     Loading config from builtin defaults
INFO     Loading config from /home/pi/.config/mopidy/mopidy.conf
INFO     Loading config from command line options
INFO     Enabled extensions: mpd, http, moped, stream, m3u, softwaremixer, file, musicbox_
webclient, local
INFO     Disabled extensions: spotify
WARNING  Found spotify configuration errors, the extension has been automatically disabled
:
WARNING  spotify/username must be set.
WARNING  spotify/password must be set.
WARNING  Please fix the extension configuration errors or disable the extensions to silenc
e these messages.
INFO     Starting Mopidy mixer: SoftwareMixer
INFO     Starting Mopidy audio
INFO     Starting Mopidy backends: StreamBackend, M3UBackend, FileBackend, LocalBackend
INFO     Audio output set to "autoaudiosink"
INFO     Loaded 7 local tracks using json
INFO     Starting Mopidy core
INFO     Starting Mopidy frontends: MpdFrontend, HttpFrontend
ERROR    Frontend (MpdFrontend) initialization error: MPD server startup failed: [Errno 98
] Address already in use
ERROR    Frontend (HttpFrontend) initialization error: HTTP server startup failed: [Errno
98] Address already in use
INFO     Starting GLib mainloop
```

In this error screen the Spotify configuration were showing as disabled. To fix this error we opened the Mopidy config file “sudo nano mopidy.conf” located in “.config/mopidy/mopidy.conf”

In the Spotify section it asks for username and password. We needed to open the Spotify account to request a username that will be used in mobile devices. We used that unique username and created a new password that will be used for our Mopidy and saved the file and restarted running mopidy.

For Mopidy to work as a client we needed to stop mopidy as a service using “sudo systemctl stop mopidy” this allowed the MPD Frontend to be released and available for the Mopidy as client as well as the HTTP frontend.

Why was this happening? Because Mopidy as a service was taking the place of the MPD and HTTP ports and address and could only be used by one Mopidy.

MPD is an extension that provides full MPD Server implementations to make mopidy available to MPD clients.

HTTP is an extension that lets you control mopidy through HTTP and web sockets for example from a web client. Its bundled with mopidy and enabled by default.

```
pi@raspberrypi: ~/mopidy-dev/mopidy/mopidy
File Edit Tabs Help

pi@raspberr... x pi@raspberr... x

faul=1.10,ifacecommon=4.2.1,ifacedigitalio=3.1.0,piippi=1.30,pillow=2.6.1,ply=3.4,py=1.4.31,pyasn1=0.1.7,pycparser=2.10,pycurl=7.19.5,pygame=1.9.2a0,pyobject=3.14.0,Pykka=1.2.1,pyope
nSSL=13.1,pyserial=2.6,pyspotify=2.0.5,pytest=3.0.4,pytest-capturelog=0.7,pytest-cov=2.4.0,pytest-xdist=1.5.0,python-apt=0.9.3.12,requests=2.12.1,responses=0.5.1,RPi.GPIO=0.6.2,RTIMU
Lib=7.2.1,sense-hat=2.1.0,singledispatch=3.4.0.3,six=1.10.0,spidev=3.0,stevedore=1.18.0,tornado=2.3,urllib3=1.9.1,virtualenv=15.1.0,virtualenv-clone=0.2.6,virtualenvwrapper=4.7.2
py27-tornado23 runtests: PYTHONHASHSEED=66290029
py27-tornado23 runtests: commands[0] | py.test tests/http
===== test session starts =====
platform linux2 -- Python 2.7.9, pytest-3.0.4, py-1.4.31, pluggy-0.4.0
rootdir: /home/pi/mopidy-dev/mopidy, inifile:
plugins: xdist-1.15.0, cov-2.4.0, capturelog-0.7
collected 20 items

tests/http/test_events.py ..
tests/http/test_handlers.py ..
tests/http/test_server.py .....

===== pytest-warning summary =====
W1 /home/pi/mopidy-dev/mopidy/.tox/py27-tornado23/local/lib/python2.7/site-packages/pytest_capturelog.py:171: 'pytest_runtest_makereport' hook uses deprecated _multicall_ argument
W1 None pytest_funcarg_caplog: declaring fixtures using 'pytest_funcarg_' prefix is deprecated and scheduled to be removed in pytest 4.0. Please remove the prefix and use the @pytest.fixture d
ecorator instead.
W1 None pytest_funcarg_capturelog: declaring fixtures using 'pytest_funcarg_' prefix is deprecated and scheduled to be removed in pytest 4.0. Please remove the prefix and use the @pytest.fixture
decorator instead.
===== 20 passed, 3 pytest-warnings in 0.85 seconds =====
py27-tornado31 inst-nodesps: /home/pi/mopidy-dev/mopidy/.tox/dist/Mopidy-2.0.1.zip
py27-tornado31 installed: apipkg=1.4,backports-abc=0.5,certifi=2016.9.26,cffi=1.1.2,charset=2.3.0,colorama=0.3.2,cookies=2.2.1,coverage=4.2,execnet=1.4.1,funcsigs=1.0.2,gpiozero=1.2.0,h
tml5lib=0.999,tkkeymap=0.1,mcpi=0.1.1,mock=2.0.0,Mopidy=2.0.1,Mopidy-MusicBox-WebClient=2.3.0,Mopidy-Spotify=3.0.0,MySQL-python=1.2.3,ndg-httpclient=0.3.2,numpy=1.0.2,phras=1.10.0,picame
ra=1.10,ifacecommon=4.2.1,ifacedigitalio=3.1.0,piippi=1.30,pillow=2.6.1,ply=3.4,py=1.4.31,pyasn1=0.1.7,pycparser=2.10,pycurl=7.19.5,pygame=1.9.2a0,pyobject=3.14.0,Pykka=1.2.1,pyope
nSSL=13.1,pyserial=2.6,pyspotify=2.0.5,pytest=3.0.4,pytest-capturelog=0.7,pytest-cov=2.4.0,pytest-xdist=1.5.0,python-apt=0.9.3.12,requests=2.12.1,responses=0.5.1,RPi.GPIO=0.6.2,RTIMU
Lib=7.2.1,sense-hat=2.1.0,singledispatch=3.4.0.3,six=1.10.0,spidev=3.0,stevedore=1.18.0,tornado=3.1.1,urllib3=1.9.1,virtualenv=15.1.0,virtualenv-clone=0.2.6,virtualenvwrapper=4.7.2
py27-tornado31 runtests: PYTHONHASHSEED=66290029
py27-tornado31 runtests: commands[0] | py.test tests/http
===== test session starts =====
platform linux2 -- Python 2.7.9, pytest-3.0.4, py-1.4.31, pluggy-0.4.0
rootdir: /home/pi/mopidy-dev/mopidy, inifile:
plugins: xdist-1.15.0, cov-2.4.0, capturelog-0.7
collected 24 items

tests/http/test_events.py ..
tests/http/test_handlers.py .....
tests/http/test_server.py .....

===== pytest-warning summary =====
W1 /home/pi/mopidy-dev/mopidy/.tox/py27-tornado31/local/lib/python2.7/site-packages/pytest_capturelog.py:171: 'pytest_runtest_makereport' hook uses deprecated _multicall_ argument
W1 None pytest_funcarg_caplog: declaring fixtures using 'pytest_funcarg_' prefix is deprecated and scheduled to be removed in pytest 4.0. Please remove the prefix and use the @pytest.fixture d
ecorator instead.
W1 None pytest_funcarg_capturelog: declaring fixtures using 'pytest_funcarg_' prefix is deprecated and scheduled to be removed in pytest 4.0. Please remove the prefix and use the @pytest.fixture
decorator instead.
===== 24 passed, 3 pytest-warnings in 1.01 seconds =====
docs create: /home/pi/mopidy-dev/mopidy/.tox/docs
docs installdeps: -r/home/pi/mopidy-dev/mopidy/docs/requirements.txt
===== 24 passed, 3 pytest-warnings in 1.01 seconds =====
===== 24 passed, 3 pytest-warnings in 1.01 seconds =====
```

This error that we encountered was caused when running the command for testing Mopidy to make sure it was properly working. That command was “Tox” and it was testing the entire program even what we do not need. To fix this problem we needed to specify the exact area that we need to test by running “Tox -e py27”