

Python Client Library

1. Classes:

There are currently two classes, MF_API_Client which exposes methods that are callable and MF_socketIO a thin wrapper for the socketIO client that could be potentially be swapped for another connection implementation e.g. native websockets.

- **MF_API_Client**
 - MF_socketIO

2. Methods

These methods are public and can be called by the client

2.1. __init__

Constructor, creates a new instance of a client targeting a single room.

```
def __init__(self, controllerUrl, controllerPort, password, room)
```

Parameter	Required	Type	Use
controllerUrl	Yes	String	The URL of the controller
controllerPort	Yes	Int	The port of the controller's web socket server
password	Yes	String	The password for authorisation to connect to web socket server
room	Yes	String	The initial room to route commands to

2.2. sendScenesAndThemes

Sends a collection of scenes and themes to the controller for playback

```
def sendScenesAndThemes(self, scenes, themes = [])
```

Parameter	Required	Type	Use
Scenes	Yes	Any valid collection	The list of scenes to be played
Themes	No	Any valid collection	The list of themes to be player

2.3. changeRoom

Changes the room targeted by the client

```
def changeRoom(self, newRoom)
```

Parameter	Required	Type	Use
Room	Yes	String	The room to target

3. Example Usage

```
from MF_API_Client import *  
  
#these variables would be pulled from a config in a real client  
room = MF_API_Client('URL', 80, 'password', 'room')  
  
print "Connected to room"  
  
  
print "Sending scenes and themes"  
room.sendScenesAndThemes(['5797309781a29c700e9ddf41'], ['theme'])  
  
  
print "Sending scene with empty themes"  
room.sendScenesAndThemes(['5797309781a29c700e9ddf41'], [])  
  
  
print "Sending Scene with no themes"  
room.sendScenesAndThemes(['5797309781a29c700e9ddf41'])
```