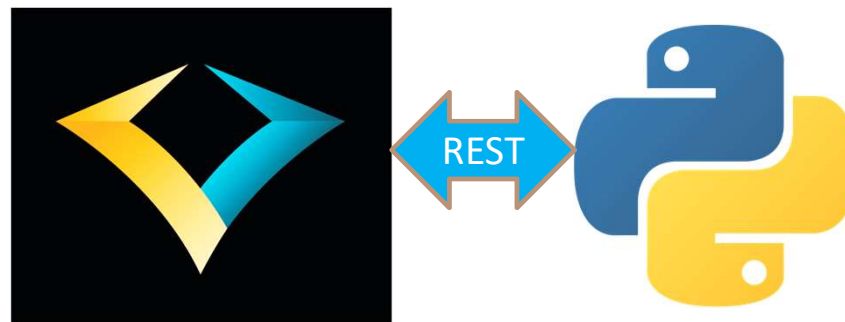


Fusion Remote Control API Using REST (REpresentational State Transfer) © Andor Technology 2019



First get Python - <https://www.python.org/ftp/python/3.7.2/python-3.7.2-amd64.exe>

1. Download installer from Link above
2. Run installer in Express mode
3. Start Windows Command Prompt
4. Run pip install requests as shown
5. Note update in yellow below
6. Run update of pip

```
Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\m.browne>pip install requests
Collecting requests
  Downloading https://files.pythonhosted.org/packages/7d/e3/20f3d364d6c8e5d2353c72a67778eb189176f08e873c9900e10c0287b84b
/requests-2.21.0-py2.py3-none-any.whl (57kB)
    100% |#####| 61kB 707kB/s
Collecting chardet<3.1.0,>=3.0.2 (from requests)
  Downloading https://files.pythonhosted.org/packages/bc/a9/01ffebfb562e4274b6487b4bb1ddec7ca55ec7510b22e4c51f14098443b8
/chardet-3.0.4-py2.py3-none-any.whl (133kB)
    100% |#####| 143kB 1.3MB/s
Collecting urllib3<1.25,>=1.21.1 (from requests)
  Downloading https://files.pythonhosted.org/packages/62/00/ee1d7de624db8ba7090d1226aebefab96a2c71cd5cfa7629d6ad3f61b79e
/urllib3-1.24.1-py2.py3-none-any.whl (118kB)
    100% |#####| 122kB 2.3MB/s
Collecting certifi>=2017.4.17 (from requests)
  Downloading https://files.pythonhosted.org/packages/9f/e0/accfc1b56b57e9750eba272e24c4dddeac86852c2bebd1236674d7887e8a
/certifi-2018.11.29-py2.py3-none-any.whl (154kB)
    100% |#####| 163kB 9.2MB/s
Collecting idna<2.9,>=2.5 (from requests)
  Downloading https://files.pythonhosted.org/packages/14/2c/cd551d81dbe15200be1cf41cd03869a46fe7226e7450af7a6545bfc474c9
/idna-2.8-py2.py3-none-any.whl (58kB)
    100% |#####| 61kB 1.8MB/s
Installing collected packages: chardet, urllib3, certifi, idna, requests
Successfully installed certifi-2018.11.29 chardet-3.0.4 idna-2.8 requests-2.21.0 urllib3-1.24.1
You are using pip version 18.1, however version 19.0.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Users\m.browne>python -m pip install --upgrade pip
Collecting pip
  Downloading https://files.pythonhosted.org/packages/d8/f3/413bab4ff08e1fc4828dfc59996d721917df8e8583ea85385d51125dceff
/pip-19.0.3-py2.py3-none-any.whl (1.4MB)
    100% |#####| 1.4MB 1.2MB/s
Installing collected packages: pip
  Found existing installation: pip 18.1
  Uninstalling pip-18.1:
    Successfully uninstalled pip-18.1
  Successfully installed pip-19.0.3

C:\Users\m.browne>
```

If running Windows 7 then the ACL needs to be updated

ACL – refers to the Access Control List which provides permissions to access files or other resources.

In order to provide permissions to the current user then the following commands must be run from an **administrator command prompt**. (right-click Command Prompt in the start menu, pick "Run as administrator").

Run the below two commands to allow port 15120, which is the default Fusion port.

You'll need to update the "DOMAIN\username" part based on whatever user account will be running Fusion.

```
netsh http add urlacl url=http://127.0.0.1:15120/ user=DOMAIN\username
```

```
netsh http add urlacl url=http://localhost:15120/ user=DOMAIN\username
```

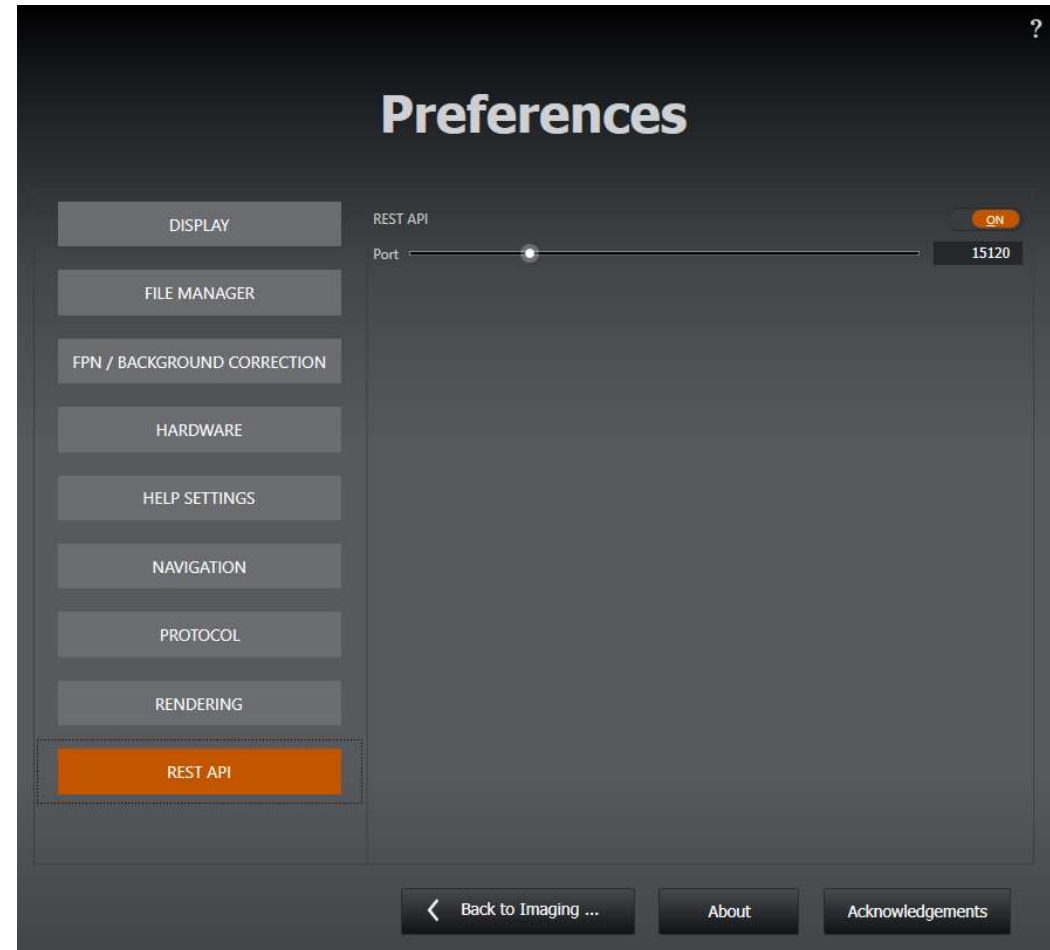
If using just a local user account on the PC, the user account will be "PCNAME\username".

If you're not sure, you can check the exact account name from a normal (non-admin) command prompt by:

```
echo %USERDOMAIN%\%USERNAME%
```

Switch ON Fusion REST API

1. Start Fusion_REST – version 2.1.0.34 or later
2. Switch to Preferences View
3. Select REST API
4. Enable REST_API slider to “ON”
5. Default Port 15120 should be left at Default
6. Click “Back to Imaging”...



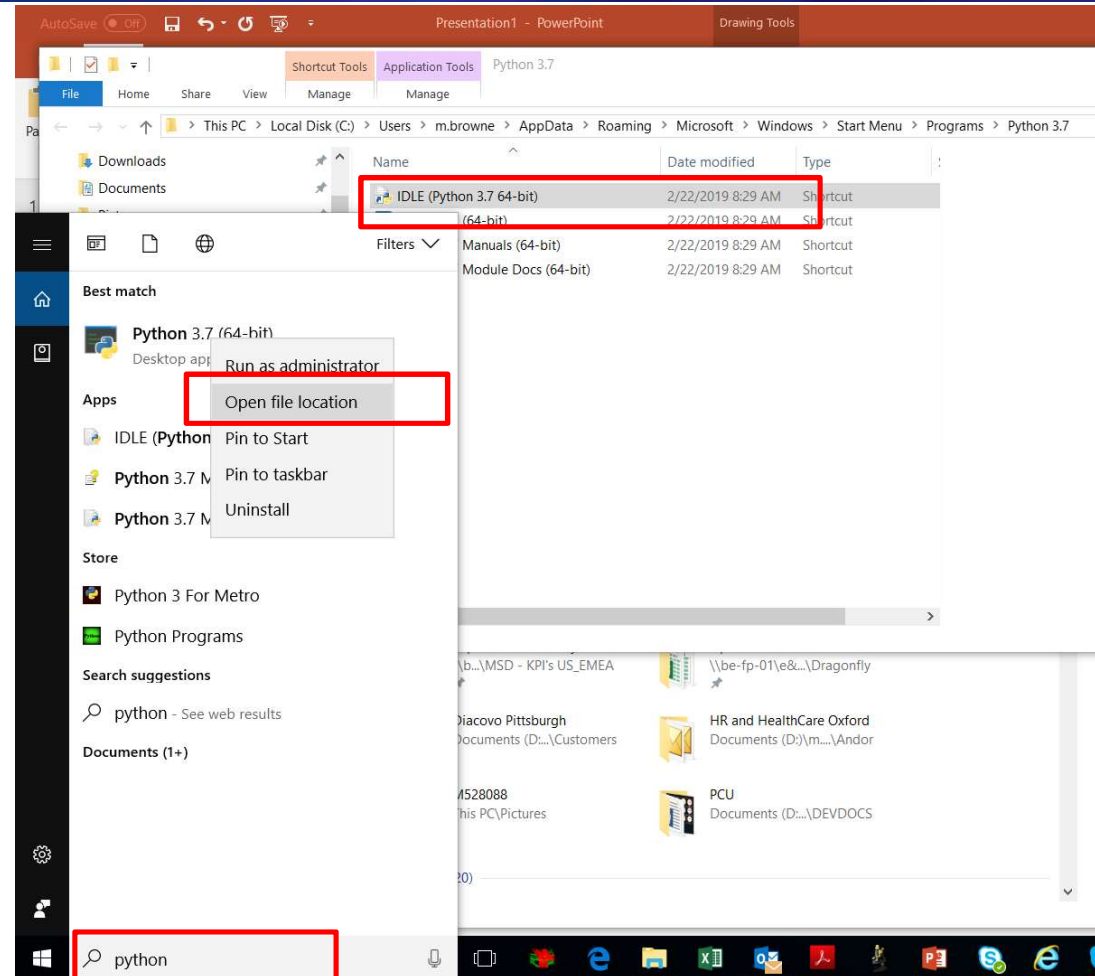
Set up Fusion Protocols for REST control

1. Ensure correct Channels exist
2. Build Protocols
3. Check protocols match script names
4. Here we have “protocol1” and “protocol2”



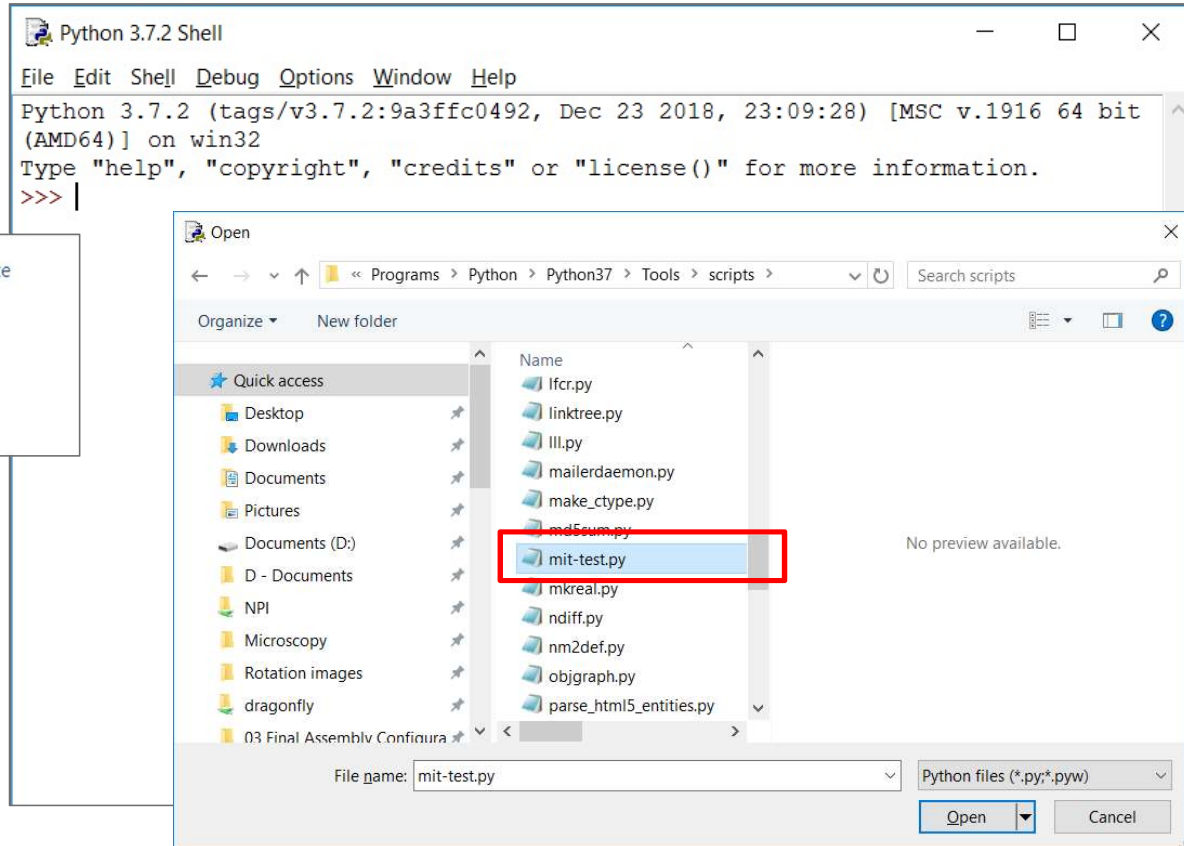
Locate Python shortcuts

1. Type “Python” into Windows search box
2. Windows results opens – Python on top
3. Right mouse Open file location - click
4. Windows File Explorer shows folder
5. Double Click on “IDLE (Python 3.7 64 bit)”
6. Right Mouse “Pin to Taskbar” if required



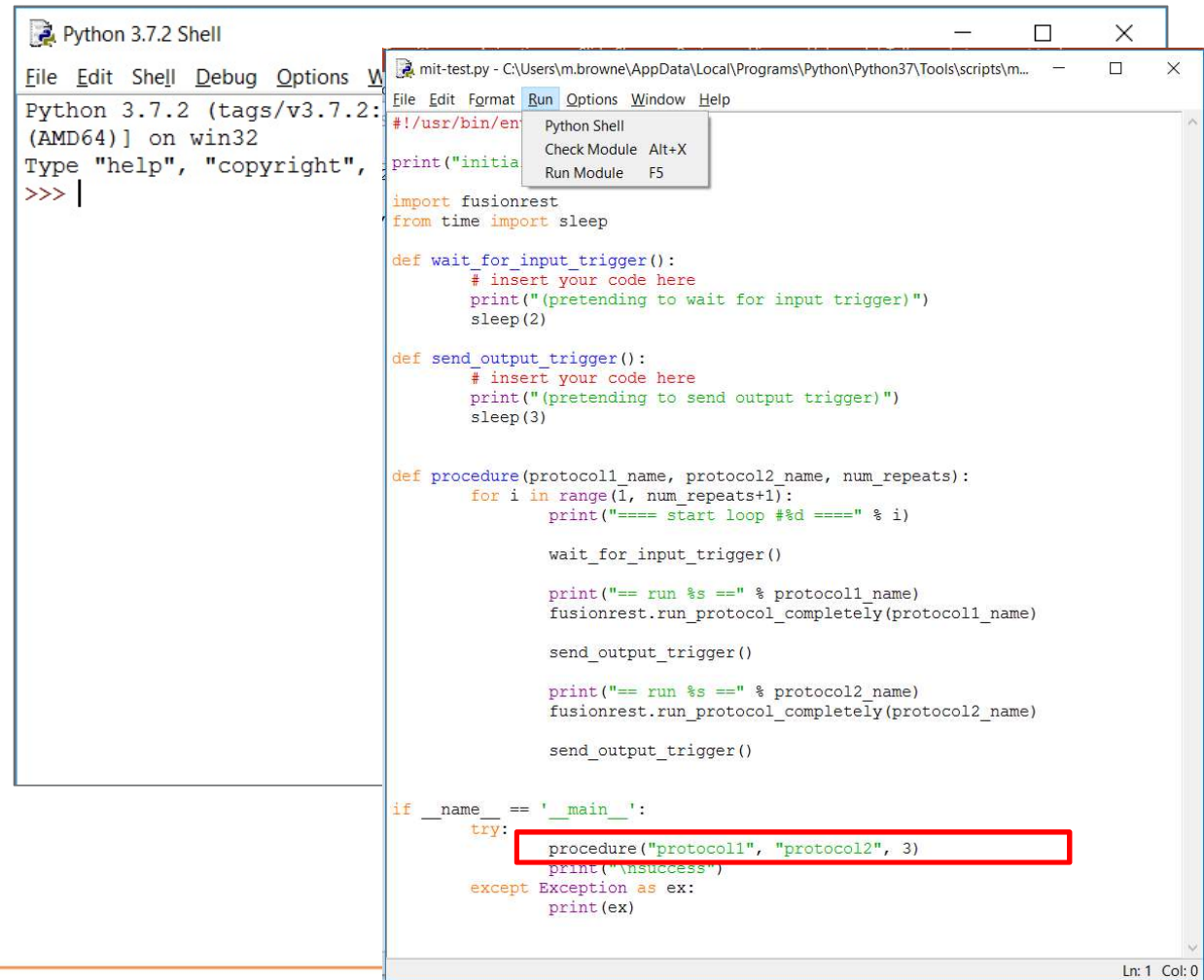
Load Fusion Python Scripts

1. IDLE editor will Open as shown
2. Locate Fusion REST Docs and Examples
3. File Open script file "mit-test.py"



Edit Fusion Python Scripts

1. mit-test.py opens in its own window
2. Edit protocol names as required
3. Change dummy processes to real triggers etc



The image shows two overlapping windows. The background window is a 'Python 3.7.2 Shell' with a menu bar (File, Edit, Shell, Debug, Options, Window) and a command prompt showing 'Python 3.7.2 (tags/v3.7.2: (AMD64)] on win32' and 'Type "help", "copyright", ...'. The foreground window is an editor for 'mit-test.py' with a menu bar (File, Edit, Format, Run, Options, Window, Help). The code in the editor is as follows:

```
#!/usr/bin/env python
import fusionrest
from time import sleep

def wait_for_input_trigger():
    # insert your code here
    print("(pretending to wait for input trigger)")
    sleep(2)

def send_output_trigger():
    # insert your code here
    print("(pretending to send output trigger)")
    sleep(3)

def procedure(protocol1_name, protocol2_name, num_repeats):
    for i in range(1, num_repeats+1):
        print("==== start loop #%d =====" % i)

        wait_for_input_trigger()

        print("== run %s == " % protocol1_name)
        fusionrest.run_protocol_completely(protocol1_name)

        send_output_trigger()

        print("== run %s == " % protocol2_name)
        fusionrest.run_protocol_completely(protocol2_name)

        send_output_trigger()

if __name__ == '__main__':
    try:
        procedure("protocol1", "protocol2", 3)
    except Exception as ex:
        print(ex)
```

A red rectangle highlights the line `procedure("protocol1", "protocol2", 3)` in the `try:` block. The status bar at the bottom right of the editor window shows 'Ln: 1 Col: 0'.

Run Fusion Python Scripts

1. Select “Run” from the menu
2. Run Module mit-test.py
3. Observe Fusion executes scripts, IDLE reports output



```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\m.browne\AppData\Local\Programs\Python\Python37\Tools\scripts\mit-test.py
initialising
==== start loop #1 ====
(pretending to wait for input trigger)
== run protocol1 ==
(pretending to send output trigger)
== run protocol2 ==
(pretending to send output trigger)
==== start loop #2 ====
(pretending to wait for input trigger)
== run protocol1 ==
(pretending to send output trigger)
== run protocol2 ==
(pretending to send output trigger)
==== start loop #3 ====
(pretending to wait for input trigger)
== run protocol1 ==
(pretending to send output trigger)
== run protocol2 ==
(pretending to send output trigger)

success
>>> |
```

Ln: 26 Col: 4

End of Preliminary setup

1. Now you can create your own protocol sequences
2. Control external hardware through Python or other
3. Note that fusionrest.py includes only the definitions
4. This means they do not execute but define REST methods
5. For more help contact the Andor team
6. email: TBD
7. The Fusion team will be engaged as needed.

```
fusionrest.py - C:\Users\m.browne\AppData\Local\Programs\Python\Python37\Tools\scripts\fusionrest.py (3.7.2)
File Edit Format Run Options Window Help

import requests
import json
import time

host = "localhost"
port = 15120

class ApiError(Exception):
    """
    Indicates an error while calling the Fusion REST API.
    """
    def __init__(self, endpoint, code, reason):
        """
        Creates a new 'ApiError' instance.
        """
        self._endpoint = endpoint
        self._code = code
        self._reason = reason

    def __repr__(self):
        return "<ApiError at {}: {} {}>".format(self._endpoint, self._code, self._reason)

    def __str__(self):
        return self.__repr__()

    def endpoint(self):
        """
        Gives the name of the API endpoint for which the error happened.
        """
        return self._endpoint

    def code(self):
        """
        Gives the HTTP response code for the error, as returned by the API.
        Also see '.reason()' for a more readable description of the problem.
        """
        return self._code

    def reason(self):
        """
        Gives the reason for the error, as returned by the API. (a string)
        """
        return self._reason

def __make_address(endpoint):
    return "http://{}:{}".format(host, port, endpoint)
```

Ln: 1 Col: 0

