



Part 78

-

Auto Updating/Upgrading

Update from the Website (URL)

You may use the `urllib` module to open and read the URL directly in Python. It is a pre-installed module in most Python versions. So you can just import this module and use the following codes to download the data from any URL.

```
import urllib.request

url = '<Your URL Here>'

output = '<Your output filename/location>'
urllib.request.urlretrieve(url, output)

#. Continue your code here
```

For instance, an example script below shows an example of updating German COVID-19 data from Robert Koch-Institut. The data is in `csv` format and available at this [URL](#). In real practice, some `URLs` may change or depreciate over time, which makes the Python script run to an error, so we may include `try/except` to check if the URL is still valid or not. Assume that you want to keep all versions of downloaded files, you may use the `datetime` module to add a timestamp to the output filename too.

```
import urllib.request
from datetime import datetime
print("Start Program ... ")
try:
    print("Start Downloading file ... ")
    timenow = datetime.now()
    #Time ISO to second resolution
    timenow_iso = timenow.strftime('%Y-%m-%dT%H:%M:%S')
    #url
    url = 'https://opendata.arcgis.com/datasets/dd4580c810204019a7b8eb3e0b329dd6_0.csv'
    # output's file name/ location
    output= f'data_RKI_{timenow_iso}.csv'
    urllib.request.urlretrieve(url, output)
    print(f"File {output} - saved!")

except Exception as e:
    print("Downloading file error: " + str(e))

    ## Continue your data analysis here...
```

If you want a list of apps to be checked for updatesn you can use a JSON file. Sample code based on a JSON file is listed below.

Of course you can even more elaborate this sample but I leave that up to you.

```

import urllib.request
import os
import json
from datetime import datetime as dt

Version          = "V2-1.40"
Debug            = True
strScriptBase    = 'VentiWater'

#-----
# current versions installed
#-----
verDict = {
    "I2C.py"       : "1.00",
    "LCD.py"       : "1.00",
    "LED.py"       : "1.00",
    "MCP4725.py"   : "1.00",
    "SR04T.py"     : "1.00",
    "VentiWater.py": Version
}

base_url = '<Your URL Here>'
jsn_file = strScriptBase + '.json'
#-----

#-----
# checks if updates are posted on url
def CheckForUpdates():

    global Debug
    global base_url, verDict, jsn_file

    dtNow      = dt.now()
    timestamp  = dtNow.strftime("%Y-%m-%d %H:%M:%S")
    strMessage = timestamp + f" - Checking for updates ... {base_url}"
    strLogging = strLogging + strMessage + "\n"
    if Debug:
        print(strMessage)

    try:
        # get file with version info
        # first delete previous downloaded
        if os.path.exists(jsn_file):
            os.remove(jsn_file)
        url = base_url + jsn_file
        urllib.request.urlretrieve(url, jsn_file)
        strMessage = timestamp + f" - {jsn_file} saved ..."
        if Debug:
            print(strMessage)

        # read file and convert to dictionary
        jString = ""
        with open(jsn_file) as f:
            for line in f:
                jString = jString + line.strip()
        vDict = json.loads(jString)

        # check if updates are needed
        for key, value in vDict.items():
            if value == verDict[key]:
                strMessage = timestamp + f" - File : {key.ljust(15)} -> Version : {value} -> Up to date"
                if Debug:
                    print(strMessage)
            else:
                strMessage = timestamp + f" - File : {key.ljust(15)} -> Version : {value} <> Version :
{verDict[key]} -> Needs Updating"
                if Debug:
                    print(strMessage)
                url = base_url + key
                urllib.request.urlretrieve(url, key)
                strMessage = timestamp + f" - {key} saved ..."
                if Debug:
                    print(strMessage)

        # remove as no longer needed
        if os.path.exists(jsn_file):
            os.remove(jsn_file)

    except Exception as e:
        strMessage = timestamp + " - Error in CheckForUpdates: " + str(e)
        if Debug:
            print(strMessage)
#-----

CheckForUpdates()

```