

Anleitung um den Advantage Database Server mittels Python 2.7 zu verwenden

Laden Sie Python 2.7 von der Seite python.org herunter:

Python 2.7.16

Release Date: March 4, 2019

Python 2.7.16 is a bugfix release in the Python 2.7 series.

Note: Attention macOS users: As of 2.7.16, all current python.org macOS installers ship with builtin copies of OpenSSL and Tcl/Tk 8.6. See the installer README for more information.

Full Changelog

Files

Version	Operating System	Description	MDS Sum	File Size	GPG
Gzipped source tarball	Source release		f1a2ace63106844831d01485466e0	17431748	SIG
XZ compressed source tarball	Source release		30157d85a2c0479c09ea2cbe61f2aaf5	12752104	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	70b0f58eba7b78b174056369b076c085	30252432	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	a3af70c13c654276d66c3c1cb1772dc7	23743901	SIG
Windows debug information files	Windows		f9469e0dbf58b10b4718badc08b1f8	25088166	SIG
Windows debug information files for 64-bit binaries	Windows		4292c4db30c27fe0bbee544967b6452	25899174	SIG
Windows help file	Windows		3bbf29b6712b231d2df9211f7b21e2	6263118	SIG
Windows x86-64 MSI installer	Windows	for AMD64/EM64T/x64	2fe86194bb4027be75b298520271a79	20361216	SIG
Windows x86 MSI installer	Windows		912428345b7e0428544ec4edcd7f0286	19419136	SIG

Achten sie darauf den richtigen Release auszuwählen:

Download the latest version for Windows

Download Python 3.7.4

Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [Mac OS X](#), [Other](#).

Want to help test development versions of Python? [Prereleases](#), [Docker images](#)

Looking for Python 2.7? See below for specific releases

Looking for a specific release?

Python releases by version number:

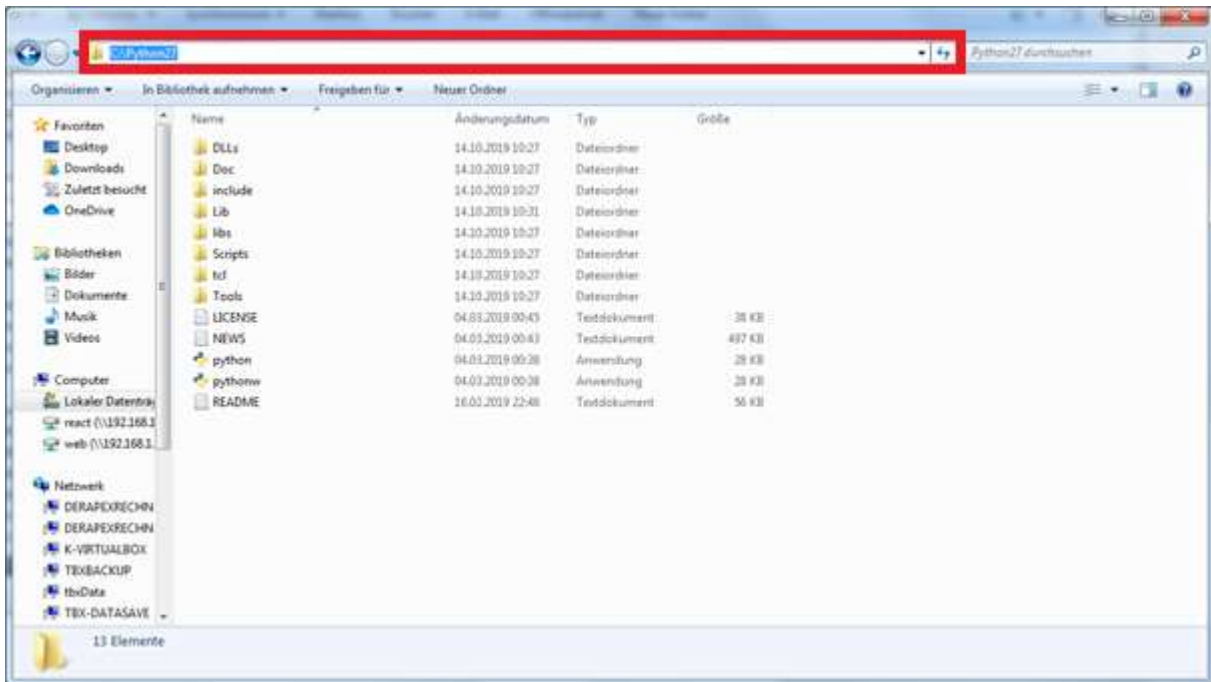
Release version	Release date	Click for more
Python 3.7.4	July 8, 2019	Download Release Notes
Python 3.6.9	July 2, 2019	Download Release Notes
Python 3.7.3	March 25, 2019	Download Release Notes
Python 3.4.10	March 16, 2019	Download Release Notes
Python 3.5.7	March 18, 2019	Download Release Notes
Python 2.7.16	March 4, 2019	Download Release Notes
Python 3.7.2	Dec. 24, 2018	Download Release Notes
Python 3.6.8	Mar. 31, 2018	Download Release Notes

Anschließend installieren Sie Python.

Sie werden neben Python noch die Advantage Client Engine benötigen. Diese finden Sie unter dem Link:

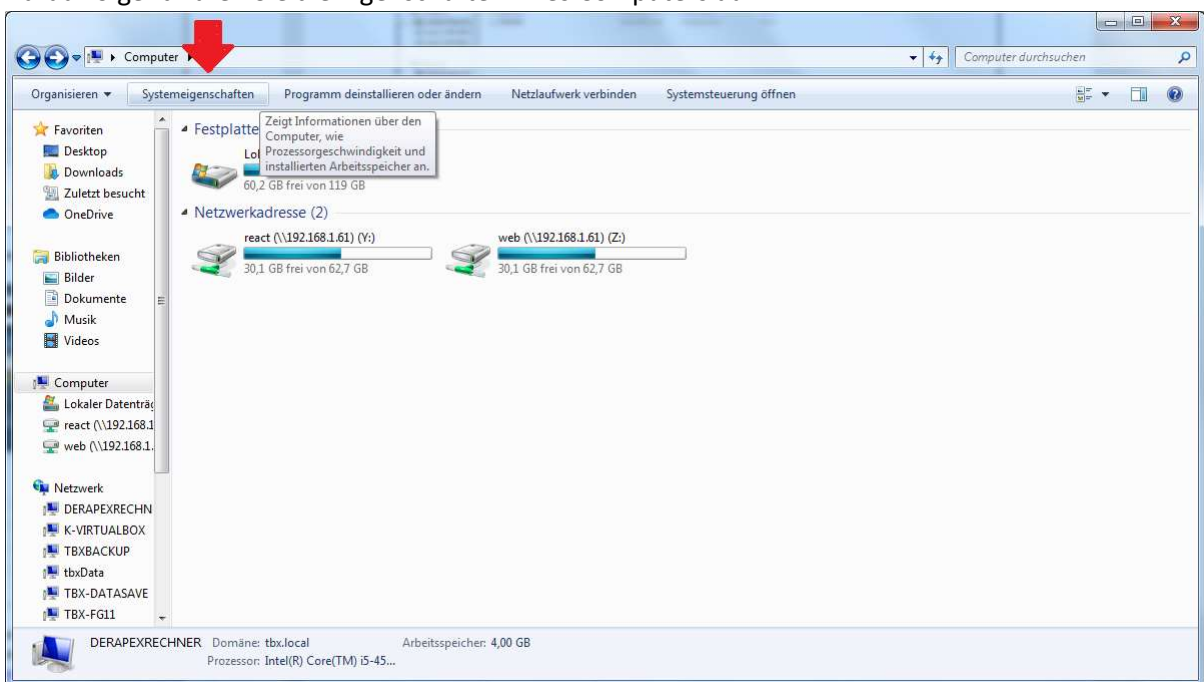
<https://devzone.advantagedatabase.com/dz/content.aspx?Key=20&Release=19&Product=5&Platform=14>

Bitte installieren Sie diese. Sie sollten folgend den Installationspfad von Python ermitteln. Im Bild finden Sie ein Beispiel:

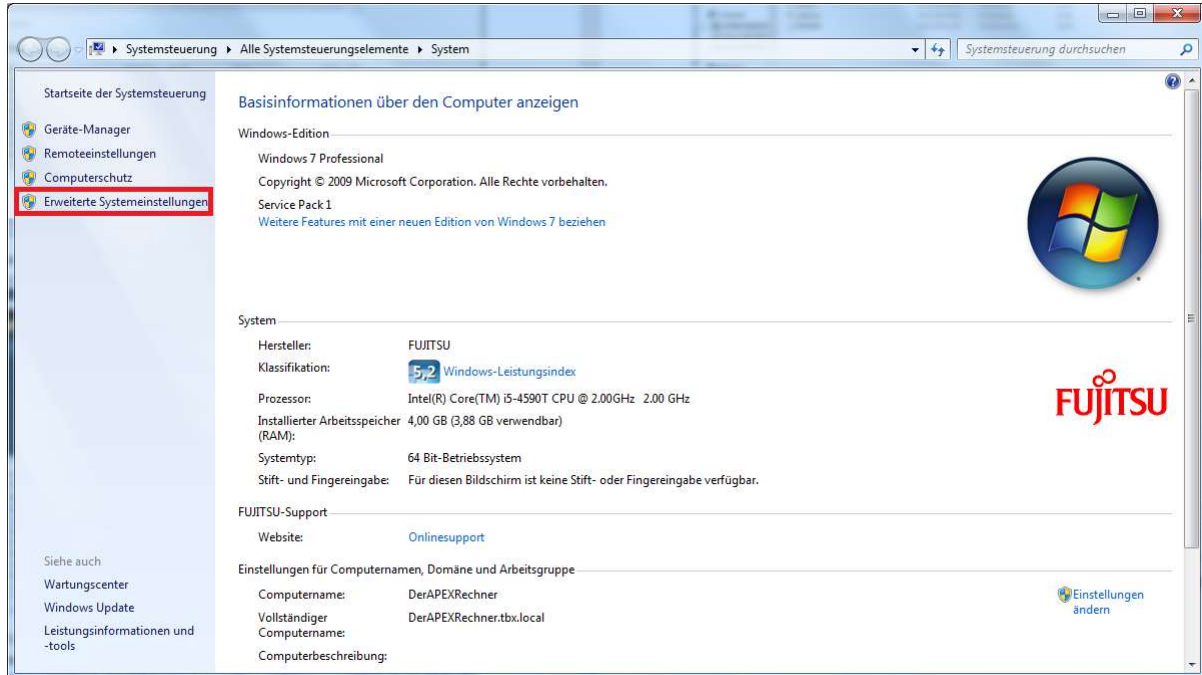


Merken Sie sich den Pfad und ermitteln Sie den Standort der ace32.dll (der Installationspfad der Advantage Client Engine).

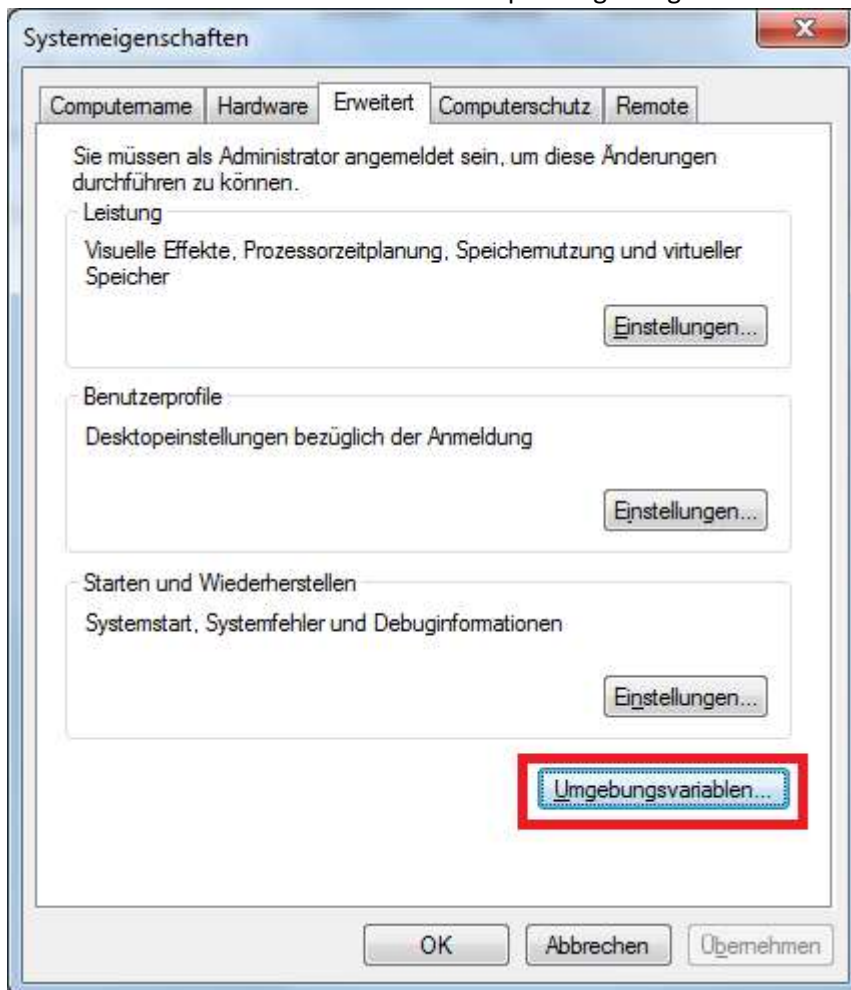
Darauffolgend rufen Sie die Eigenschaften ihres Computers auf:



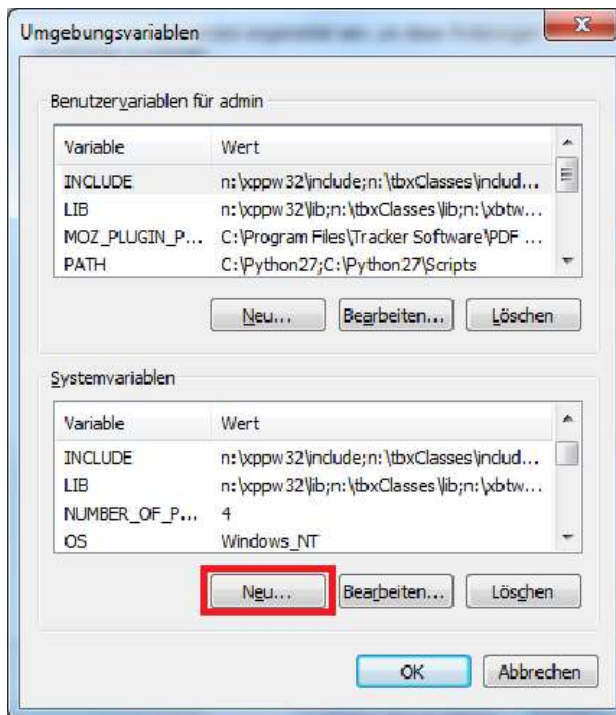
Ein neues Fenster öffnet sich. Dort klicken Sie auf "erweiterte Systemeinstellung".



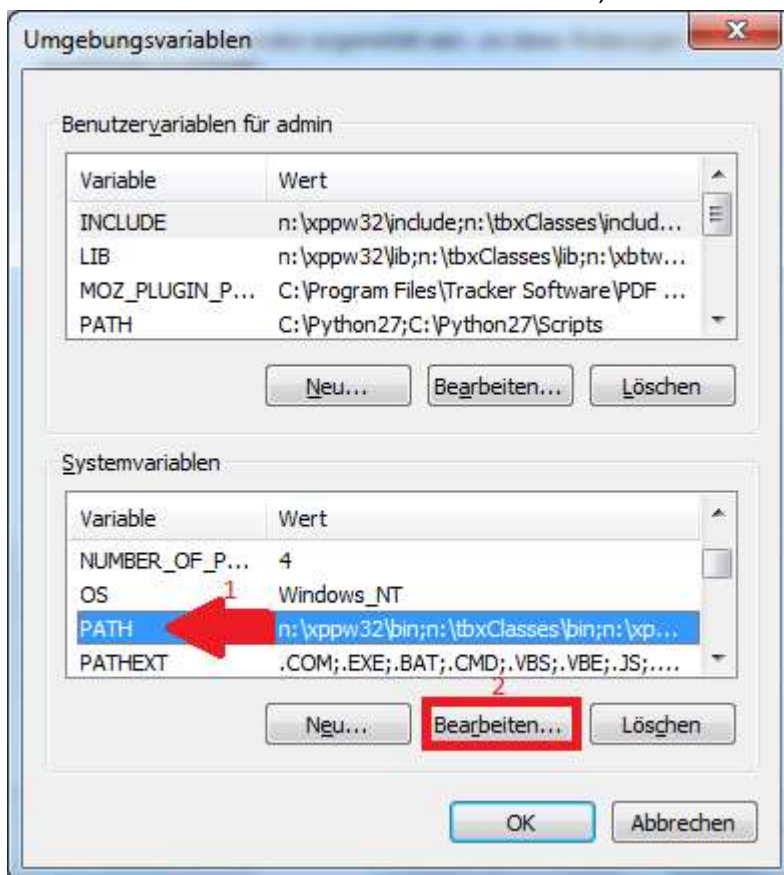
In diesem Fenster klicken Sie auf dem Knopf "Umgebungsvariablen...":



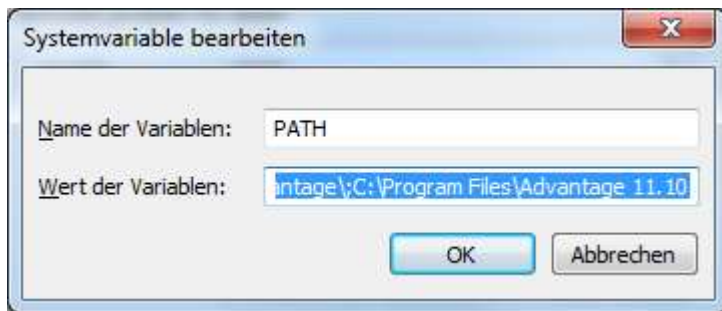
Damit Python funktioniert müssen Sie die PATH Variable anpassen. Wenn Sie die PATH Variable nicht haben, legen Sie diese bitte neu an:



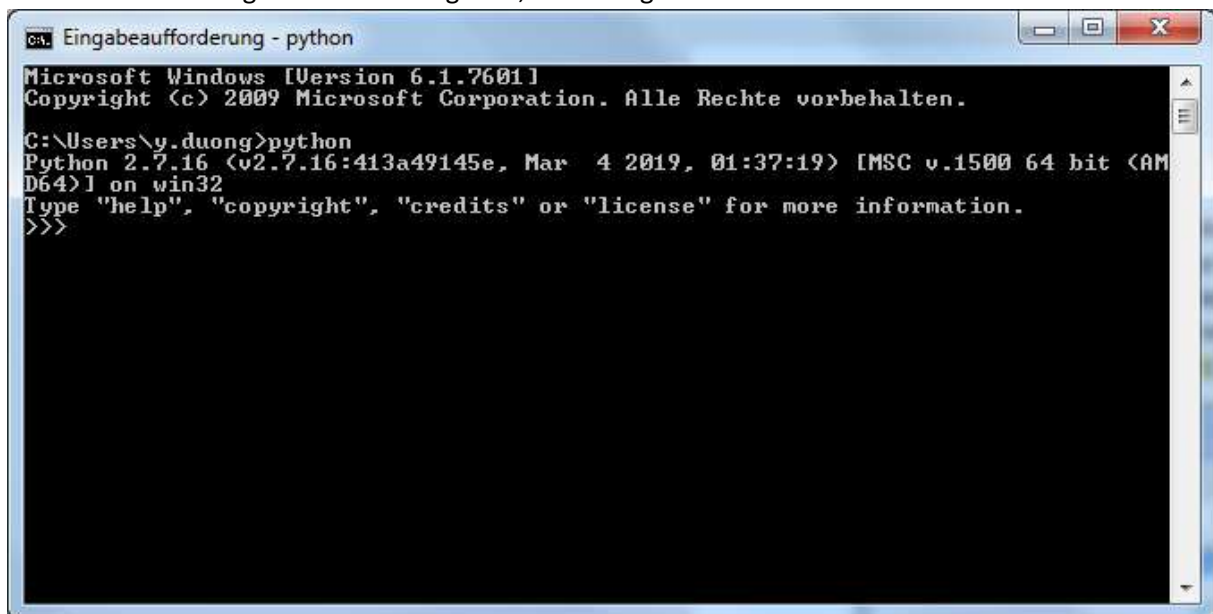
Sollte die Variable "PATH" bereits vorhanden sein, müssen Sie diese nur bearbeiten:



Drücken Sie hierfür auf "Bearbeiten". Jetzt sollte sich ein kleines Fenster mit zwei Textfeldern öffnen: Der Name der Variable und der Wert der Variablen. Geben Sie Folgendes für die PATH Variable an: "C:\Python27;C:\Python27\Scripts;C:\Program Files\Advantage 11.10":



Bestätigen Sie folgend beide Dialoge mit "OK". Daraufhin sollte Python in der Kommandozeile von Windows ordnungsgemäß funktionieren. Sie können dies testen indem Sie "python" in der Kommandozeile eingeben. Falls erfolgreich, sollte Folgendes erscheinen:



Um die Funktionsfähigkeit des Treibers zu testen, erstellen Sie eine Datei: "adspy.py" und kopiere folgendem Code:

```
import adsdb
import json
import sys

reload(sys)
sys.setdefaultencoding('utf-8')
# Adjust the connection path (C:\) and server type as necessary
conn = adsdb.connect(DataSource='\\\\\\YourServerPath ', ServerType='remote',
UserID='ADSSYS')
cur = conn.cursor()
input1 = sys.argv[1]
input2 = sys.argv[2]
print input1
cur.execute(input1)
if input2 != 'update':
    arr = cur.fetchall()
    for d in arr:
        for x in d:
            if type(x) == str:
```

```
x = x.decode('windows-1252').strip()
conn.close()
print (json.dumps(arr,encoding='latin1'))
else:
conn.close()
```

Haben Sie die Python-Dateien angelegt und dementsprechend benannt, können Sie dies mittels Kommandozeile ausprobieren. Vorher sollten Sie die “adsdb.py” Datei runterladen. Sie finden die Datei unter folgendem Link: <http://code.google.com/p/adsdb/downloads/list>. Laden sie die neuste Version der adsdb.py herunter. Extrahieren Sie die .zip-Datei in das gewünschte Verzeichnis, damit der Python-Compiler diese leichter finden kann.

Um den “adsdb.py” zu installieren führen Sie folgende Schritte aus:

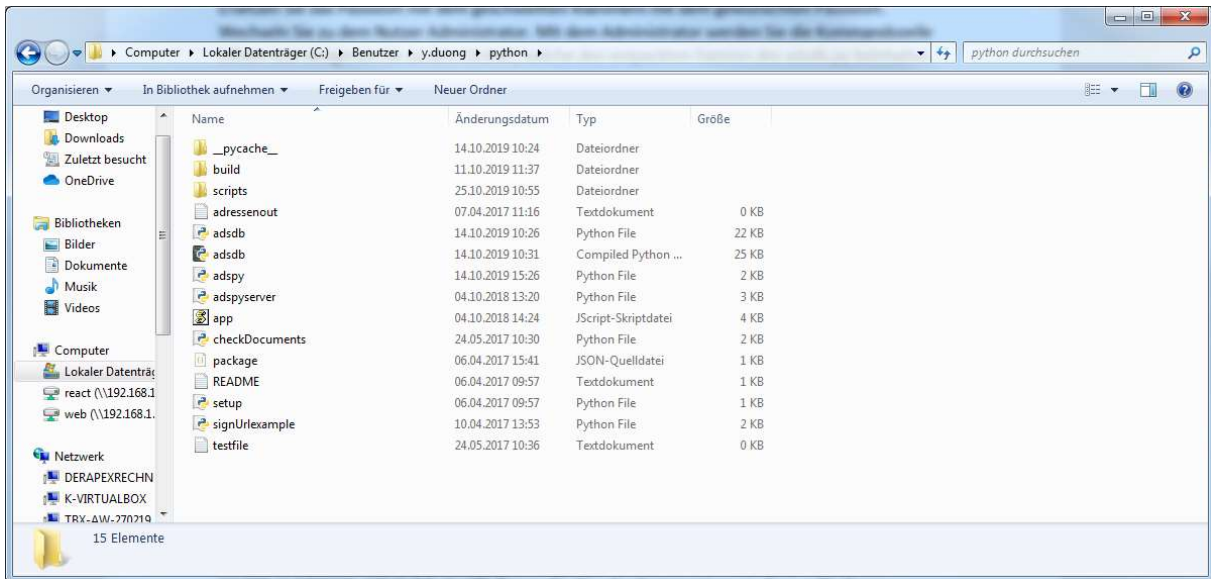
Führen Sie die Kommandozeile als Administrator aus. Falls die nächsten Schritte nicht funktionieren nutzen Sie folgenden workaround:

Workaround

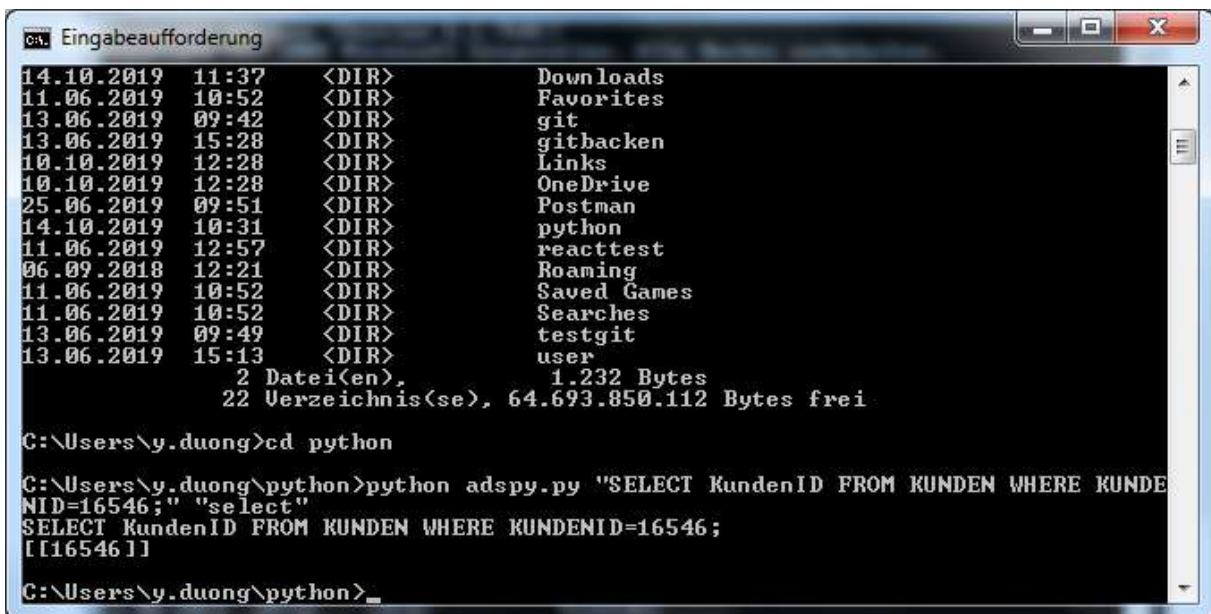
Sie sollten die Kommandozeile als Administrator ausführen und den User Administrator freischalten. Dies können Sie mit dem Befehl: “net user Administrator /active:yes”. Darauffolgend setzen Sie das Passwort für den Administrator. Geben Sie dafür “**net user administrator {Passwort}**” ein. Ersetzen Sie das Passwort mit dem geschweiften Klammern mit dem gewünschten Passwort. Wechseln Sie zu dem Nutzer Administrator. Mit dem Administrator werden Sie die Kommandozeile öffnen und navigieren Sie zu dem Ordner, welche den entpackten Dateien des adsdb.py beinhaltet.

Anschließend führen Sie die Datei; “setup.py”, aus. Dies tun Sie indem Sie “python setup.py install” in die Kommandozeile eingeben. Versichern Sie sich, dass Sie sich im Installationsordner befinden. Nach der erfolgreichen Installation schließen Sie die Kommandozeile oder im Falle des Workarounds wechseln Sie auf dem vorherigen Nutzer. Sollten Sie den Administrator nicht mehr benötigen, sperren Sie den Nutzer. Dafür müssen Sie die Kommandozeile als Administrator öffnen und geben “**net user administrator /active:no**”, ein. Darauffolgend können sie noch testen, ob “adsdb.py” erfolgreich installiert ist. Navigieren Sie dazu in das Installationsverzeichnis und öffnen Sie den Ordner “scripts”. In diesem Ordner sollte eine Datei mit dem Namen: “test.py”, befinden. Führen Sie diese aus in dem Sie Folgendes in die Kommandozeile eingeben: “python test.py”. Sollte “adsdb.py” erfolgreich installiert sein gibt das Programm “adsdb successfully installed” aus. Sollte das Programm einen Fehler ausgeben, der den Pfad betrifft, dann sollten Sie diesen in der Datei: “test.py”, anpassen. Ändern Sie den Pfad zum tatsächlichen Aufenthaltsort der “adsdb“-Datei. Testen Sie erneut mithilfe der Kommandozeile, ob das Skript nun ordnungsgemäß ausgeführt wird.

Öffnen Sie die Kommandozeile und navigieren Sie in den Ordner in dem sich die “adsdb“-Datei befindet. Kopieren Sie die vorher angelegte Datei “adspy.py”, in diesen Ordner. Der Ordner sollte danach ungefähr so aussehen:



Um die Datei abschließend zu testen verwenden Sie folgende Eingabe: "python adspy.py "SELECT KundenID FROM KUNDEN WHERE KUNDENID=16546;" "select"". Passen Sie hier das Statement so an, dass es für ihre Datenbank funktioniert.



Sie können diese Abfrage ebenfalls in der Form eines Servers testen. Dafür kopieren Sie den folgenden Code:

adspyserver.py:

```

import time
import BaseHTTPServer
import adsdb
import json
import sys
import urlparse
import urllib

```

```

reload(sys)
sys.setdefaultencoding('utf-8')
HOST_NAME = 'localhost' # !!!REMEMBER TO CHANGE THIS!!!
PORT_NUMBER = 9000 # Maybe set this to 9000.
conn = adsdb.connect(DataSource='\\\\\\YourServerPath', ServerType='remote',
UserID='ADSSYS')
cur = conn.cursor()

class MyHandler(BaseHTTPServer.BaseHTTPRequestHandler):
    def do_HEAD(s):
        s.send_response(200)
        s.send_header("Content-type", "text/plain")
        s.end_headers()
    def do_GET(s):
        querystring =
urlparse.parse_qs(urlparse.urlparse(s.path).query).get('query', None)
        typestring =
urlparse.parse_qs(urlparse.urlparse(s.path).query).get('type', None)
        if querystring != None:
            aquerystring =
urllib.unquote_plus(urllib.unquote_plus(querystring[0])).decode('utf8')
            atypestring =
urllib.unquote_plus(urllib.unquote_plus(typestring[0])).decode('utf8')
            #print aquerystring
            #if atypestring == 'select':
                cur.execute(aquerystring)
            #else:
                # print aquerystring
                if atypestring != 'update' and atypestring != 'insert':
                    arr = cur.fetchall()
                    for d in arr:
                        for x in d:
                            if type(x) == str:
                                x = x.decode('windows-1252').strip()
                                #print x
                            databasestring = json.dumps(arr,encoding='latin1')
                            s.send_response(200)
                            s.send_header('Content-type', 'text/html')
                            s.end_headers()
                            s.wfile.write(databasestring)
            else:
                s.send_response(200)
                s.send_header('Content-type', 'text/html')
                s.end_headers()
                s.wfile.write('Success!')
        def log_message(self, format, *args):
            return

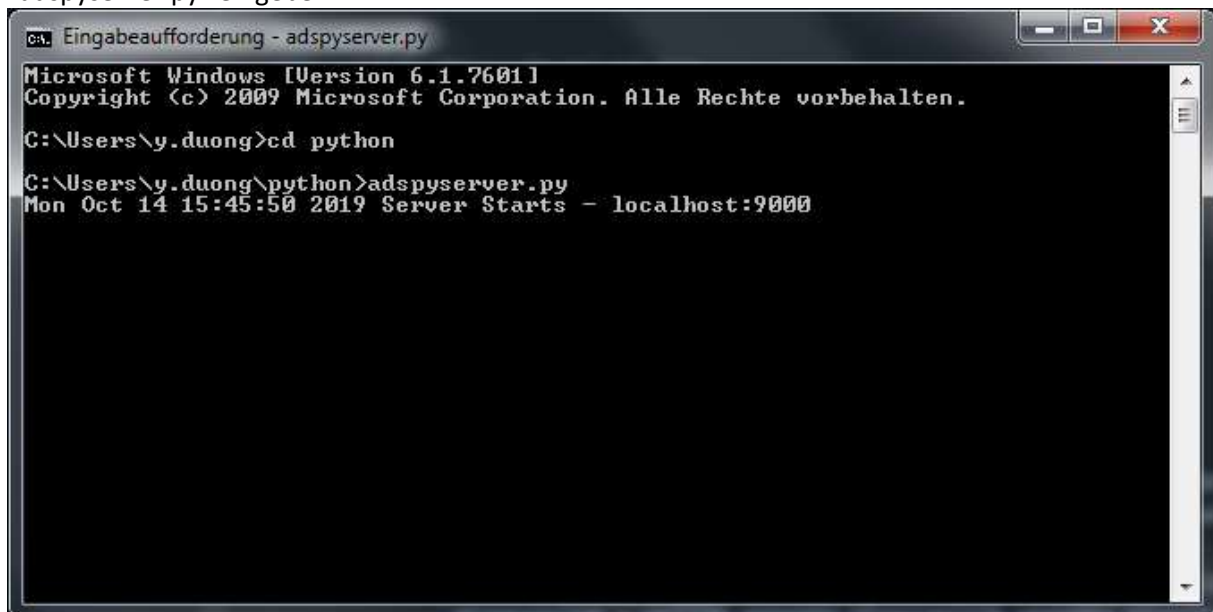
if __name__ == '__main__':
    server_class = BaseHTTPServer.HTTPServer

```



```
httpd = server_class((HOST_NAME, PORT_NUMBER), MyHandler)
print time.asctime(), "Server Starts - %s:%s" % (HOST_NAME, PORT_NUMBER)
try:
    httpd.serve_forever()
except KeyboardInterrupt:
    pass
httpd.server_close()
print time.asctime(), "Server Stops - %s:%s" % (HOST_NAME, PORT_NUMBER)
```

Sollten Sie eine andere Portnummer für den "localhost" bevorzugen, können Sie im Code eine andere Zahl für die Portnummer eingeben. Berücksichtigen Sie dabei, dass die Portnummer nicht für ein anderes Programm vergeben ist. Erstellen sie die Datei für den Server bestmöglich im gleichen Ordner, wie die anderen .py-Dateien. Um den "localhost"-Server zu starten, sollten Sie sich in dem gleichen Verzeichnis wie bei dem Beispiel "adspy.py" befinden. Startet Sie den Server indem Sie "adspyserver.py" eingeben.



```
ca. Eingabeaufforderung - adspyserver.py
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.
C:\Users\y.duong>cd python
C:\Users\y.duong\python>adspyserver.py
Mon Oct 14 15:45:50 2019 Server Starts - localhost:9000
```

Es wird angezeigt, dass der Server gestartet ist und auf den Port 9000 lauscht. Um den Server zu testen geben sie folgenden Link in ihren Browser ein:
"http://localhost:9000/?query=SELECT%20*%20FROM%20BANKEN&type=select". Ändern Sie, falls Sie eine anderen Portnummer angegeben haben die Portnummer. Passen Sie auf, dass die Query-Variable im o.g. Link URL-Kodiert ist.