

```

#!/usr/bin/env python

import socket
import subprocess
import time

import uno
from com.sun.star.beans import PropertyValue

class LIBO(object):
    HOST = 'localhost'
    PORT = '8100'
    ARG = f'socket,host={HOST},port={PORT};urp;StarOffice.ComponentContext'
    CMD = ['soffice',
           '-env:SingleAppInstance=false',
           '-env:UserInstallation=file:///tmp/LIBO_Process8100',
           '--headless', '--norestore', '--nologo', f'--accept={ARG}']

    def __init__(self):
        self._app = None
        self._start_office()
        self._init_values()

    def _init_values(self):
        self._ctx = None
        self._sm = None
        self._desktop = None
        if self.is_running:
            ctx = uno.getComponentContext()
            service = 'com.sun.star.bridge.UnoUrlResolver'
            resolver = ctx.ServiceManager.createInstanceWithContext(service, ctx)
            self._ctx = resolver.resolve('uno:{ }'.format(self.ARG))
            self._sm = self._ctx.ServiceManager
            self._desktop = self._create_instance('com.sun.star.frame.Desktop')
        return

    def _create_instance(self, name, with_context=True):
        if with_context:
            instance = self._sm.createInstanceWithContext(name, self._ctx)
        else:
            instance = self._sm.createInstance(name)
        return instance

    @property
    def is_running(self):
        try:

```

```

        s = socket.create_connection((self.HOST, self.PORT), 5.0)
        s.close()
        return True
    except ConnectionRefusedError:
        return False

def _start_office(self):
    if self.is_running:
        return

    for i in range(3):
        self.app = subprocess.Popen(self.CMD,
            stdout=subprocess.PIPE, stderr=subprocess.PIPE)
        time.sleep(5)
        if self.is_running:
            break
    return

def _set_properties(self, values):
    ps = tuple(
        [PropertyValue(Name=n, Value=v) for n, v in values.items()]
    )
    return ps

def open(self, path, options):
    options = self._set_properties(options)
    path = self._path_url(path)
    try:
        doc = self._desktop.loadComponentFromURL(path, '_blank', 0, options)
        return doc
    except:
        return None

def _path_url(self, path):
    if path.startswith('file://'):
        return path
    return uno.systemPathToFileUrl(path)

def close(self):
    if self.is_running:
        if not self._desktop is None:
            self._desktop.terminate()
        if not self._app is None:
            self._app.terminate()
    return

```

```
def main():
    PATH = '/home/elmau/Projects/test/python/test.xlsx'

    app = LIBO()
    if app.is_running:
        options = {'Hidden': True}
        doc = app.open(PATH, options)

        sheet = doc.Sheets[0]
        sheet['A1'].String = 'Work fine...'

        doc.calculateAll()
        doc.store()
        doc.close(True)

    app.close()
    print('Ok...')
    return

if __name__ == '__main__':
    main()
```