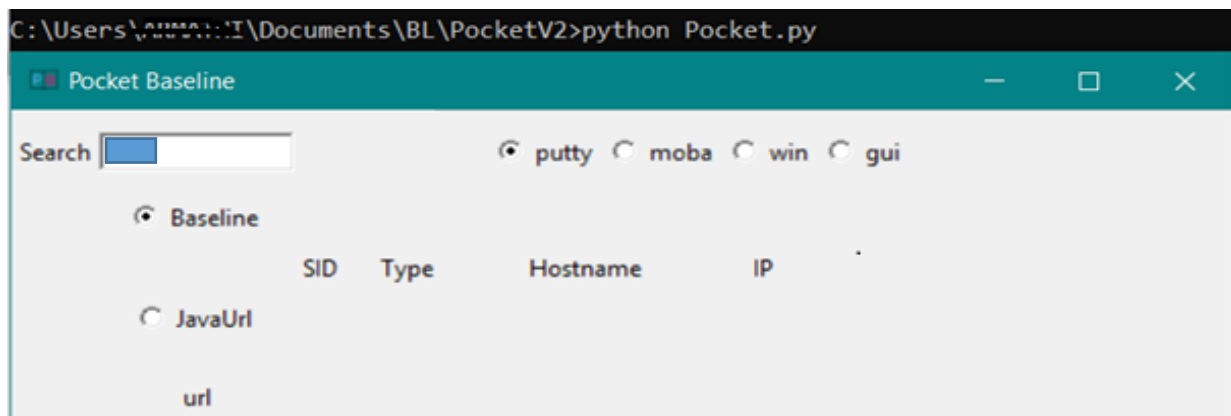
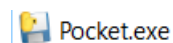


How to Use Pocket Baseline:

Case1) If Python is already installed, just launch for the command line with command `python Pocket.py`



Case2) If python is not installed, we can launch from the executable as below.



Pocket.exe.

In either of the cases make sure below files are available in the same directory where we are launching the program.

baseline.csv	8/1/2020 3:53 AM	Microsoft Excel Co...	1 KB
gui.csv	7/19/2020 1:23 AM	Microsoft Excel Co...	3 KB
interface.ini	8/1/2020 7:34 PM	Configuration setti...	1 KB
javaurl.csv	8/1/2020 4:17 AM	Microsoft Excel Co...	1 KB
logo.ico	7/24/2020 4:39 AM	ICO File	103 KB
Pocket.exe	8/1/2020 8:44 PM	Application	7,876 KB
Pocket.py	8/1/2020 8:35 PM	JetBrains PyCharm ...	8 KB
Pocket.spec	8/1/2020 8:44 PM	SPEC File	1 KB
putty.exe	2/26/2020 7:25 PM	Application	835 KB

This simple program can be used to search the SAP system with SID or with hostname or with IP address. Not only that we can login to the OS level via putty or mobaXterm as well.

For this to work , Make sure we populate the baseline.csv with below format. Column 'Type' is optional.

A	B	C	D
SID	Type	Hostname	IP
ABC	DBCI	hostname1	127.0.0.1
ABC	App1	hostname2	127.0.0.2
ABC	App2	hostname3	127.0.0.3
ABC	App3	hostname3	127.0.0.3
CBA	DBCI	hostname5	127.0.0.5
CBA	App1	hostname6	127.0.0.6
CBA	App2	hostname3	127.0.0.3
CBA	App3	hostname8	127.0.0.8
HAN	HANA	hostname4	127.0.0.4
EX1	DBCI	hostname10	127.0.0.10
EX2	DBCI	hostname11	127.0.0.11

Once this is done, we can search start searching the SIDs as below.

Search with SID:

The screenshot shows the 'Pocket Baseline' application window. At the top, there's a search bar with 'ABC' entered. Below it, there are radio buttons for 'putty', 'moba', 'win', and 'gui', with 'putty' selected. Underneath, there are radio buttons for 'Baseline' and 'JavaUrl', with 'Baseline' selected. The 'Baseline' section displays a table with columns: SID, Type, Hostname, IP, and a 'Launch' button. The table contains four rows of data for SID 'ABC'. Below the table, there's a 'JavaUrl' section with a 'url' label.

SID	Type	Hostname	IP	Launch
ABC	DBCI	hostname1	127.0.0.1	Launch
ABC	App1	hostname2	127.0.0.2	Launch
ABC	App2	hostname3	127.0.0.3	Launch
ABC	App3	hostname3	127.0.0.3	Launch

Search with Hostname:

Pocket Baseline

Search

☒ putty ☐ moba ☐ win ☐ gui

☒ Baseline

SID	Type	Hostname	IP	
ABC	App2	hostname3	127.0.0.3	Launch
ABC	App3	hostname3	127.0.0.3	Launch
CBA	App2	hostname3	127.0.0.3	Launch

☐ JavaUrl

url

Search with IP:

Pocket Baseline

Search

☒ putty ☐ moba ☐ win ☐ gui

☒ Baseline

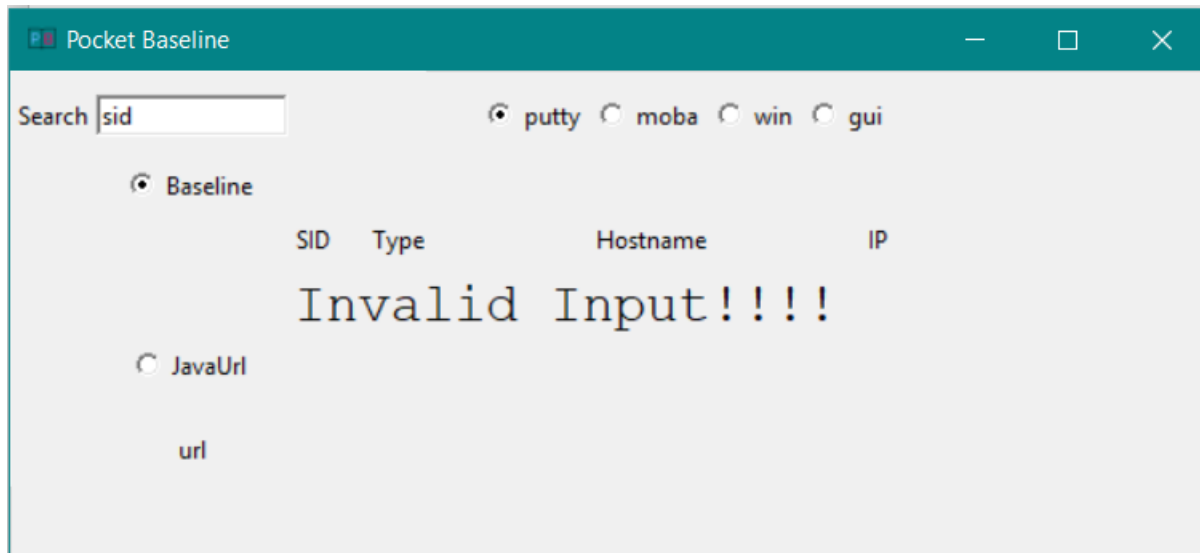
SID	Type	Hostname	IP	
ABC	App1	hostname2	127.0.0.2	Launch

☐ JavaUrl

url

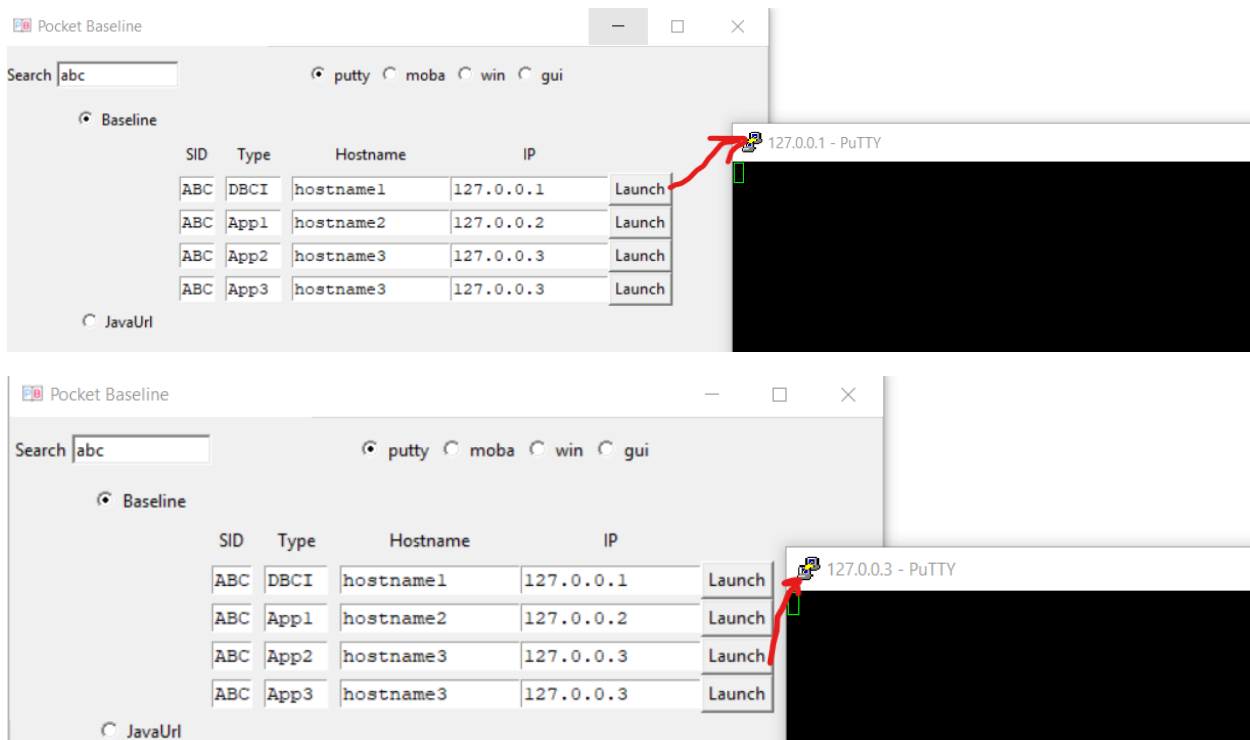
Invalid Input:

Incase if we input something invalid, we will end up with Invalid input!!! text as below.



Launching with Putty:

This tool is not only used to search the sap systems, but can be used to login to the os level via putty.



Launching with MobaXterm:

If MobaXterm is installed in your PC first update the absolute path of MobaXterm.exe in interface.ini as below.

 interface.ini - Notepad

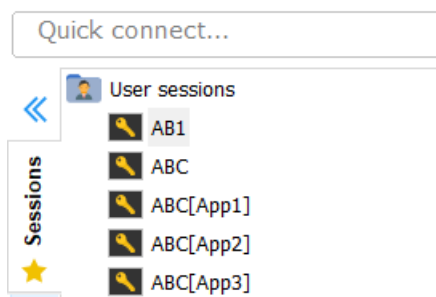
File Edit Format View Help

```
putty = "putty.exe"  
moba = "C:\Users\ARMATHI\Downloads\Software\MobaXterm\Moba.exe"  
gui = "C:\Program Files (x86)\SAP\FrontEnd\SAPgui\sapshcut.exe"  
browser = "C:\Program Files (x86)\Google\Chrome\Application\chrome.exe"
```

There are multiple ways to realize the launch of mobaXterm from this tool. Since it is openssh, we cannot launch unlike putty.

Approach 1:

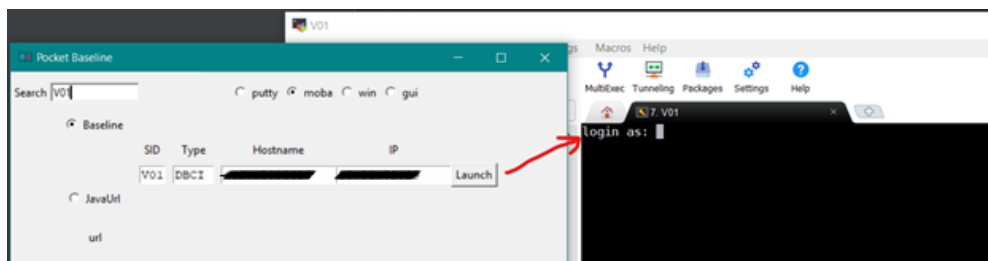
Create bookmark for all the SIDs in the MobaXterm.



If the server is standalone then MobaXterm is started with that particular SID. If the System SID has multiple app servers, maintain the bookmark as SID[Type]. Type name should be same as what is mentioned in type column in baseline.csv.

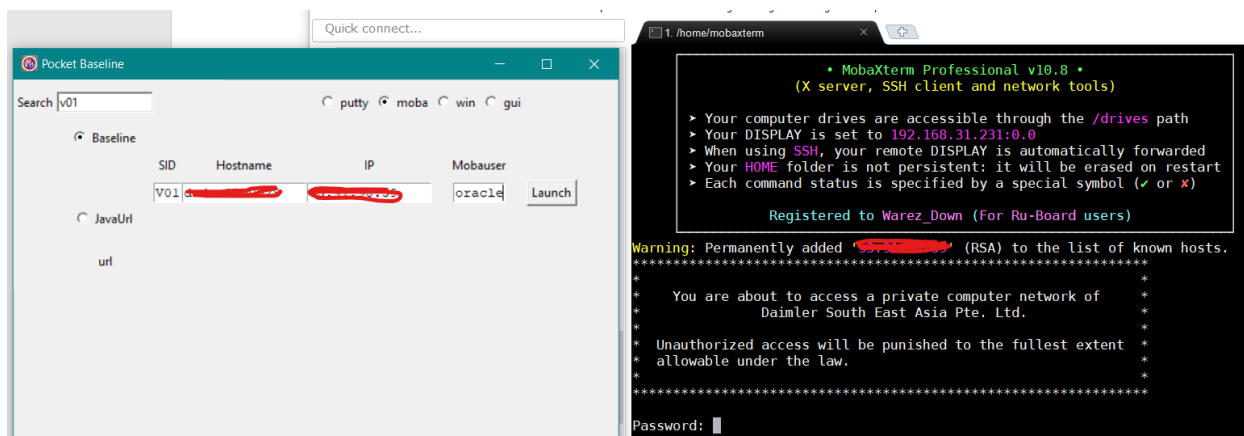
Below is the logic.

```
def mobalaunch(sid, app):  
    if app.upper().startswith("APP"):  
        subprocess.Popen(f'{conf_dict["moba"]} -bookmark {sid}[{app}]')  
    else:  
        subprocess.Popen(f'{conf_dict["moba"]} -bookmark {sid}')
```



Approach 2:

We can give the user name as an input in the textbox mobauser and launch the mobaxterm.



You can launch PocketV3 for mobaXterm launch specifying username in the textbox.

Logic Being.

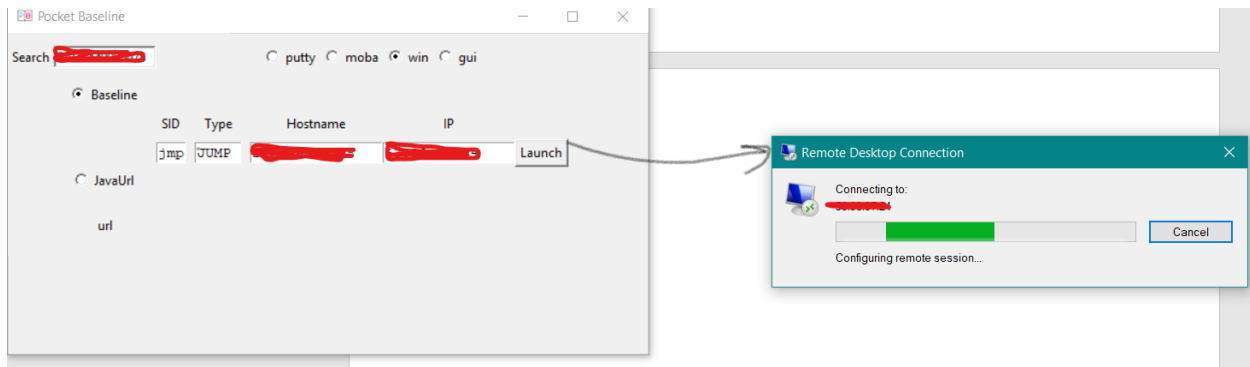
```
elif p.get() == "moba":
    #print(usr_text.get("1.0","end"))
    subprocess.Popen(f'{conf_dict["moba"]} -newtab \\'waitforX; ssh {usr_text.get("1.0","end").strip()}@{ip}\\\'')
```

Launching Windows Servers:

Login for launching windows server is very staright forward.

```
elif p.get() == "win":
    subprocess.Popen(f'mstsc /v {ip.strip()}')
```

It is same as launching mstsc /v <Ipaddress>



Launching SAPGUI:

For SAPGUI to work, maintain the absolute path of sapshcut for parameter gui.

```

interface.ini - Notepad
File Edit Format View Help
putty = "C:\Users\XXXXXXXX\Documents\Python\TKINTER\Baseline\putty.exe"
moba = "C:\Users\XXXXXXXX\Downloads\Software\MobaXterm Professional Edition 10.8 Build 3652 Portable\Moba.exe"
gui = "C:\Program Files (x86)\SAP\FrontEnd\SAPgui\sapshcut.exe"
browser = "C:\Program Files (x86)\Google\Chrome\Application\chrome.exe"
  
```

Prerequisites:

- Make sure SAP systems are maintained in SAPGUI.
- Maintain the entries as below in gui.csv

1	SID	Type	command
2	ABC		-user=Testuser -system=ABC -pw=password -client=0 -maxgui
3	CBD		-user=testuser2 -system=CBD -pw=passwd -client=000 -maxgui
4	DEB		-user=testuser3 -system=DEB -pw=passwd -client=000 -maxgui
5	BBC		-user=testuser4 -system=BBC -pw=passwd -client=200 -maxgui
6	SCC		-system=SCC -user=USERNAME -snc_name=p/secure:CN=sncname.org.net, O=company, C=IN
7	SC2		-system=SC2 -user=USERNAME -snc_name=p/secure:CN=hostname.org.net, O=Company, C=EN
8	RAN		-system=RAN -user=USERNAME -snc_name=p/secure:CN=hostname.org.net, O=Company, C=EN

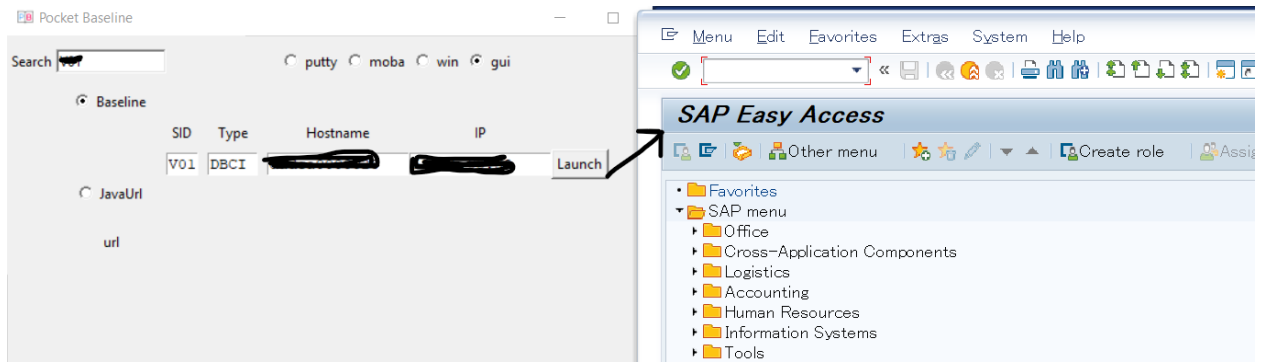
We can use either username password or snc_name along with user.

c) In case Multiple application servers are present in a system[example App1,App2,App3] then maintain one entry as App and one entry for DBCI in column Type as below in gui.csv .

20	MUL	APP	-user=Testuser -pw=password -client=0 -maxgui
21	MUL	DBCI	-user=Testuser -pw=password -system=MUL -client=0 -maxgui

Then in SAPGUI screen maintain the entries, as SID[Type] same as what we maintained in baseline.csv.

Name	System Description	SID	Group/Server	Ins...	Message Server	Router(s)
[App1]	[redacted]	[redacted]	[redacted]	00		
[App2]	[redacted]	[redacted]	[redacted]	01		



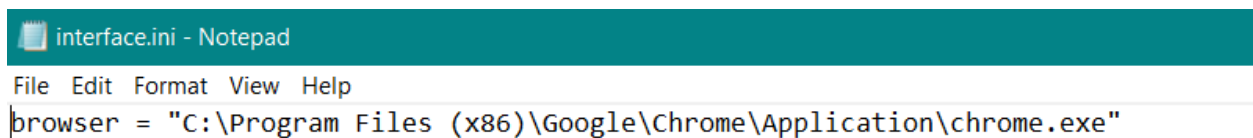
Note: Every time when we switch the radio button between, putty, moba, win or gui, please press enter in Search box.

Launching the Java Url with Web Browser:

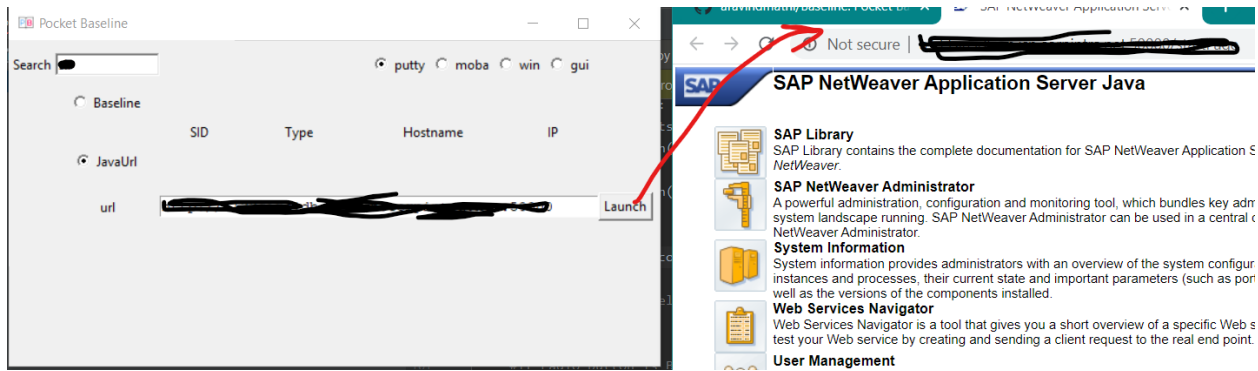
For this to work, we need to populate the file javaurl.csv as below.

	A	B
1	SID	url
2	ABC	http://hostname.example.net:50000
3	BBC	http://hostname2.example.net:50000
4	EXM	http://hostname3.example.net:50000
5	MAR	http://marhost.example.com:50100
6	YAR	https://yarhostname:50101
7	AVI	https://avihostname.example.com:50301

The default browser is maintained as chrome in interface.ini file.



Just in case if we want to use any other browser, maintain the absolute path of that browser exe (example opera.exe or iexplorer.exe).



PS: I am still a beginner in Python. Your valuable suggestions or feedback are much appreciated.

For any ideas on customization of the code, please reach out at aravindfrnd51@gmail.com