WSPython

1.1 Liquid UI Python Interop - Download and Installation

Starting the Interop

It is customary to start the Interop using the SESSION.sjs file (different languages will use a different session.sjs file).

Loading the extension

You can load the extension with this WS command

```
load('wspython');
```

After loading, create the main Interop object like this:

```
wsPython = new _Python({
dll:"C:\\Users\\te\\AppData\\Local\\Programs\\Python\\Python311\\pytho
n311.dll"});
```

wsPython is now the main object in which you interact with the Python session.

Loading python scripts/modules

You can load Python scripts with this method:

```
wsPython._load('
R:\
\SYNACTIVE\\SynJSObjects\\wspython\\SCRIPTS\\mobilenetv2_wdbg.py');
```

You can provide the full path name of the Python code to the _load command. Behind the code will take out the folder/directory name and add it to the sys.path system variable of Python. It will remove the .py from the filename, and then perform an import of the basename.

Accessing Python Variables and functions

Python is organized by ‘modules’. Therefore, underneath your main Python object, here called wsPython, there is an object called "modules.”.

(c) 2026 Liquid UI | Synactive | GuiXT <dev@guixt.com> | 2026-11-03 01:12
URL: https://www.liquid-ui.com/knowledge_base/content/182/1444/en/11-liquid-ui-python-interop-_-download-and-installation.html

WSPython

You will be able to access your Python’s functions and variables directly from your WS script.

For example, if you have a Python object declared in Python like this:

```
  python_dict = {"key1":"val1"}
```

And the output of the above command will look like this:

```
[WS] wsPython.modules.testmodule01.python_dict:{"key1":"val1"}
wsPython._load('
R:\
\SYNACTIVE\\SynJSObjects\\wspython\\SCRIPTS\\mobilenetv2_wdbg.py');
```

Accessing Python Variables and functions

For example, if you have a Python script that looks like this:

```
def hello():
print("Hello world from Python")
return "string from python"
// And then invoke this function from WS like this:
println("Python returns:"+wsPython.modules.testmodule01.hello());
// The output will look like this:
Hello world from Python
Python returns:string from python
```

Unique solution ID: #2458
Author: Devi Prasanthi Korada
Last update: 2025-02-27 07:17

(c) 2026 Liquid UI | Synactive | GuiXT <dev@guixt.com> | 2026-11-03 01:12
URL: https://www.liquid-ui.com/knowledge_base/content/182/1444/en/11-liquid-ui-python-interop-_-download-and-installation.html