

Showing Progress in Big TV Screen

Display Excel Automation into Web Browser

Using watchfiles and websocket

Quick Fix and Dirty

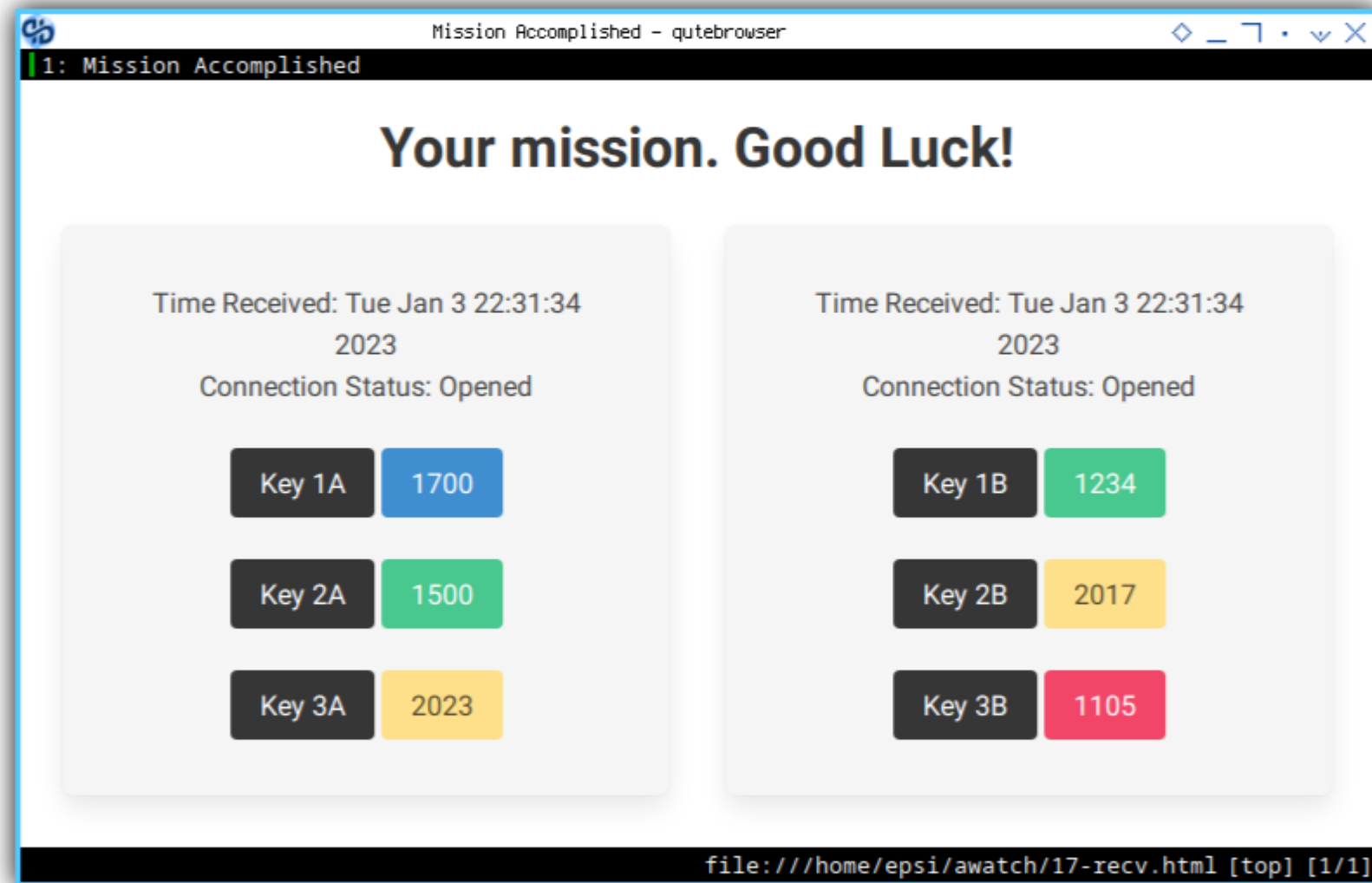
Jakarta, Indonesia

January 2023



Goal

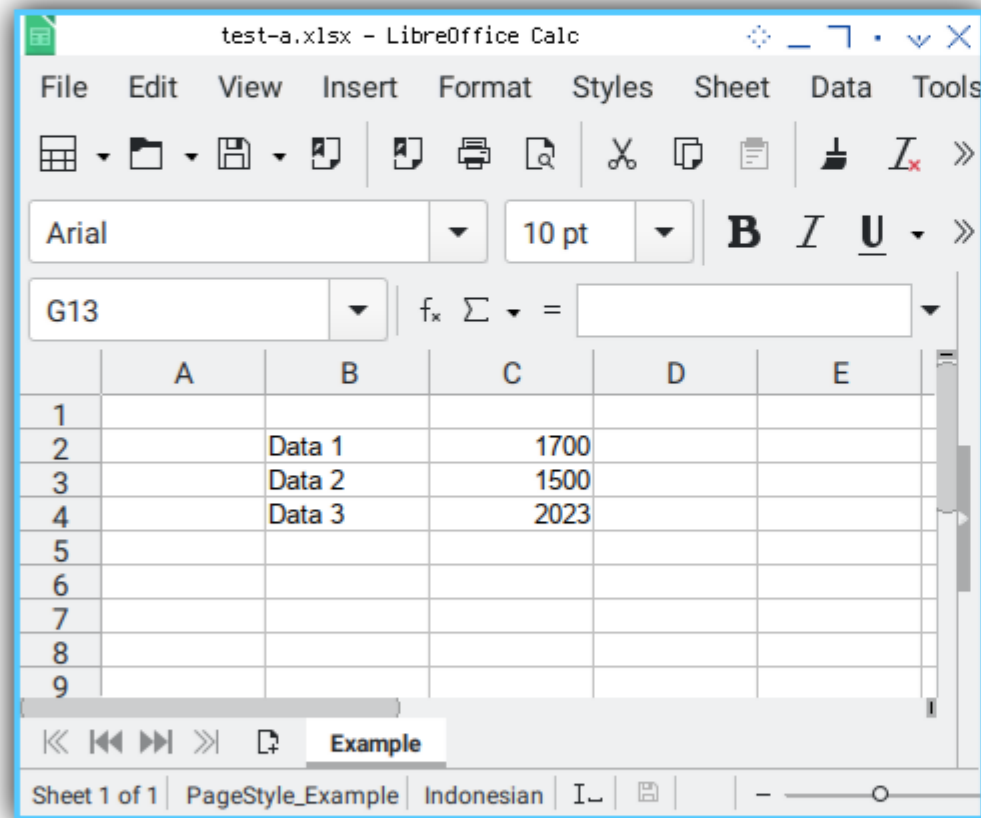
Display progress in big TV Screen.



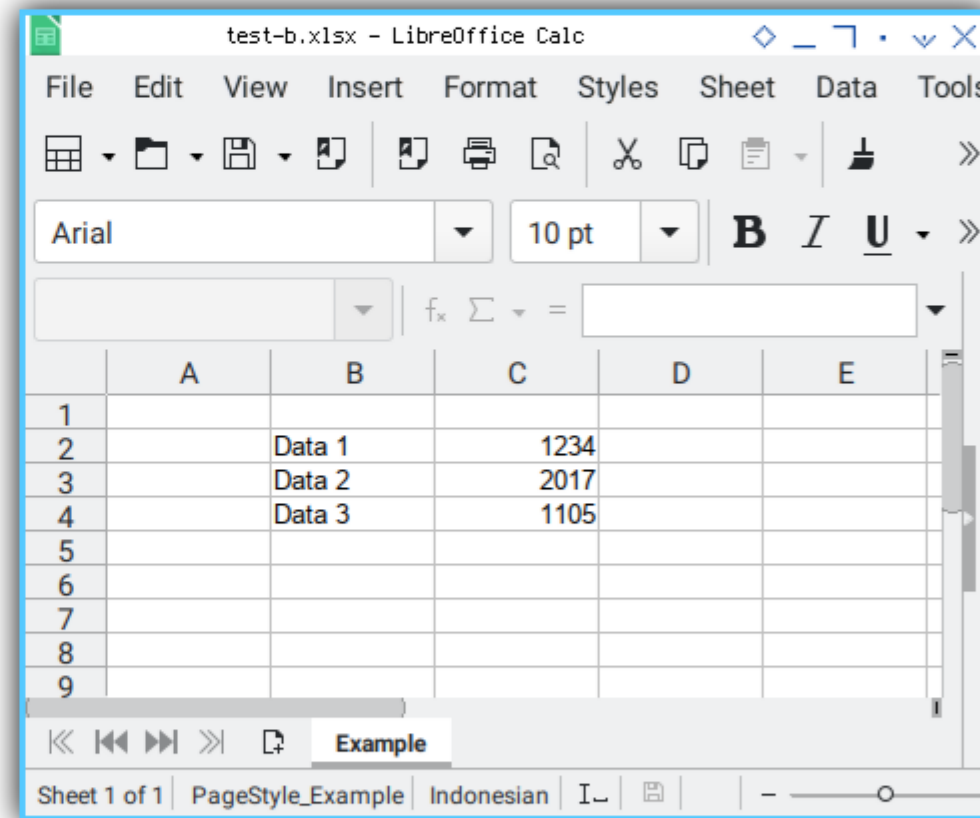
Such as production, sales, or frontoffice.

Basic Idea

Data update, for any **save event** in excel.



	A	B	C	D	E
1					
2		Data 1	1700		
3		Data 2	1500		
4		Data 3	2023		
5					
6					
7					
8					
9					



	A	B	C	D	E
1					
2		Data 1	1234		
3		Data 2	2017		
4		Data 3	1105		
5					
6					
7					
8					
9					

From excel source, **from different PC.**

How does it works?

Using python script
asynchronously.

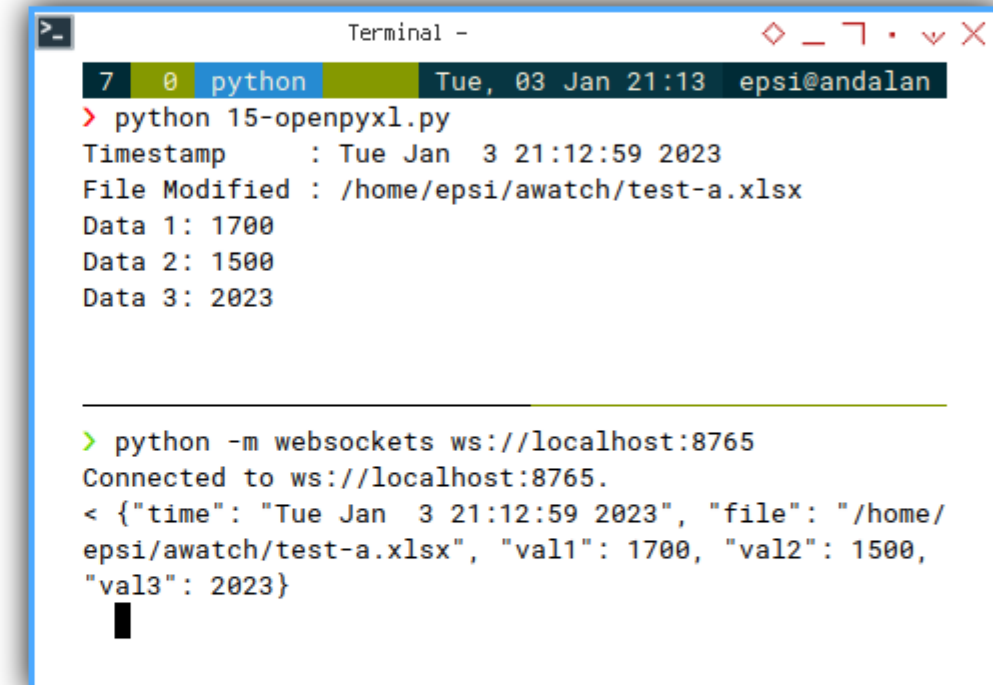
Monitor excel file event
using **watchfiles**.

Read particular **cell** in excel
using **openpyxl**, then bundle as data.

Send data using **websocket**
as **sender**, using specific port.

Open webpage in web browser
in large TV display.

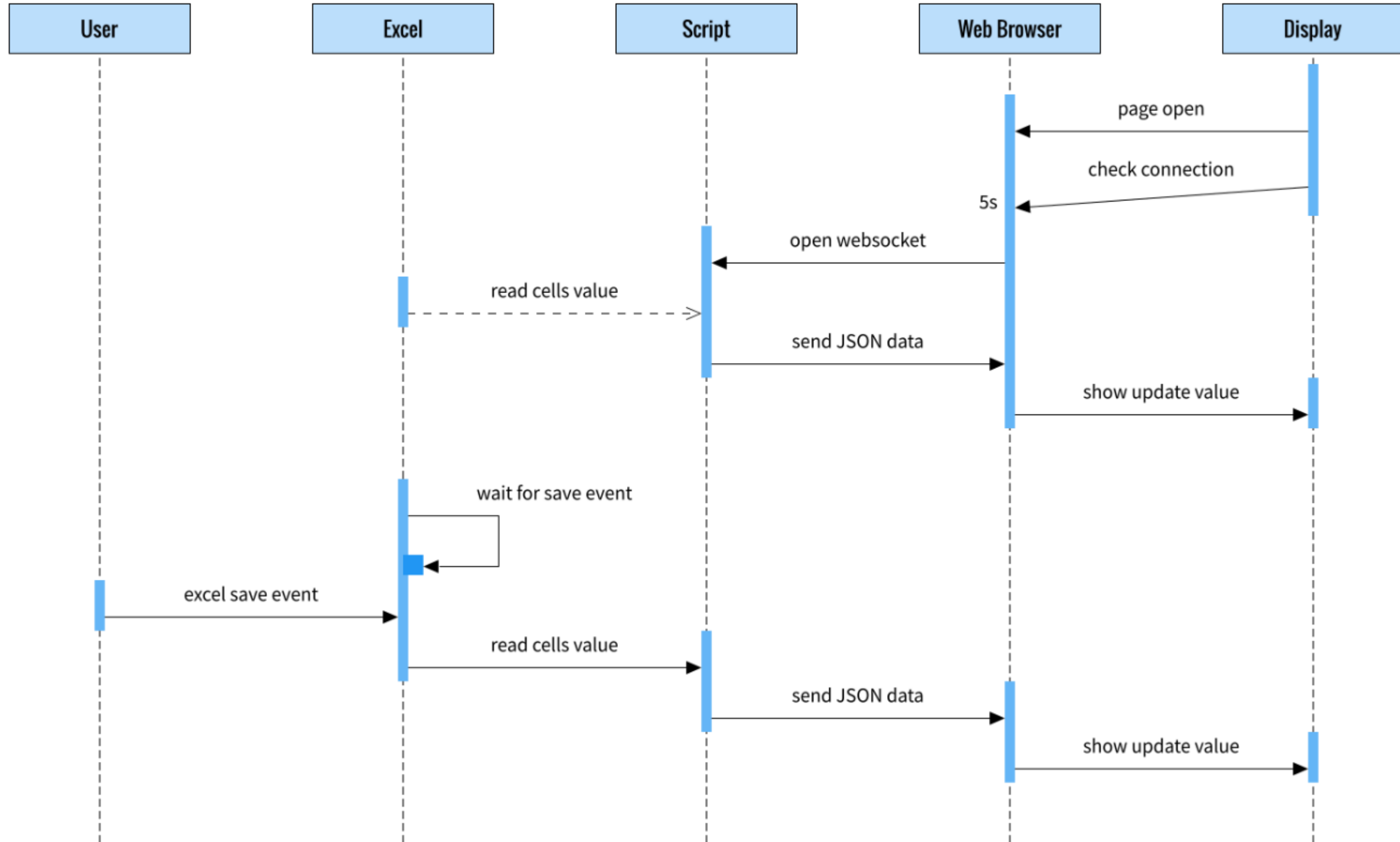
Webpage as **listener**, listen specific ip and port,
using **websocket**, then display in predefined html element.



```
Terminal -
7 0 python Tue, 03 Jan 21:13 epsi@andalan
> python 15-openpyxl.py
Timestamp      : Tue Jan  3 21:12:59 2023
File Modified  : /home/epsi/awatch/test-a.xlsx
Data 1: 1700
Data 2: 1500
Data 3: 2023

> python -m websockets ws://localhost:8765
Connected to ws://localhost:8765.
< {"time": "Tue Jan  3 21:12:59 2023", "file": "/home/epsi/awatch/test-a.xlsx", "val1": 1700, "val2": 1500, "val3": 2023}
```

Really? How does It works?



Folder Monitoring

Library: **watchfiles**.

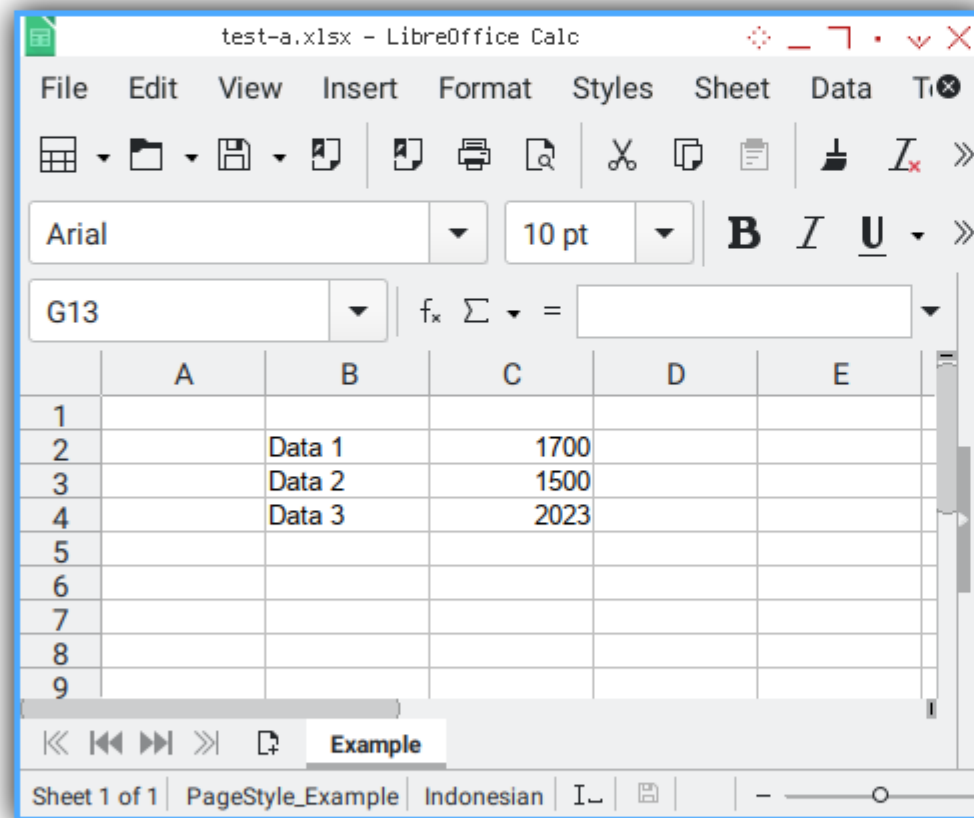
```
Terminal -
5 0 python Tue, 03 Jan 20:53 epsi@andalan
> python 13-websocket.py
Change.added
/home/epsi/awatch/.~lock.test-a.xlsx#
Change.deleted
/home/epsi/awatch/lu22487r35rc.tmp
Change.added
/home/epsi/awatch/test-a.xlsx
Change.added
/home/epsi/awatch/lu22487r35rc.tmp
Change.deleted
/home/epsi/awatch/.~lock.test-a.xlsx#

> python -m websockets ws://localhost:8765
Connected to ws://localhost:8765.
< /home/epsi/awatch/.~lock.test-a.xlsx#
< /home/epsi/awatch/lu22487r35rc.tmp
< /home/epsi/awatch/test-a.xlsx
< /home/epsi/awatch/lu22487r35rc.tmp
< /home/epsi/awatch/.~lock.test-a.xlsx#
>
```

XI2WebExample	
-	filepath
-	filename
-	site
-	port
+	__init__
-	handler()
+	main()

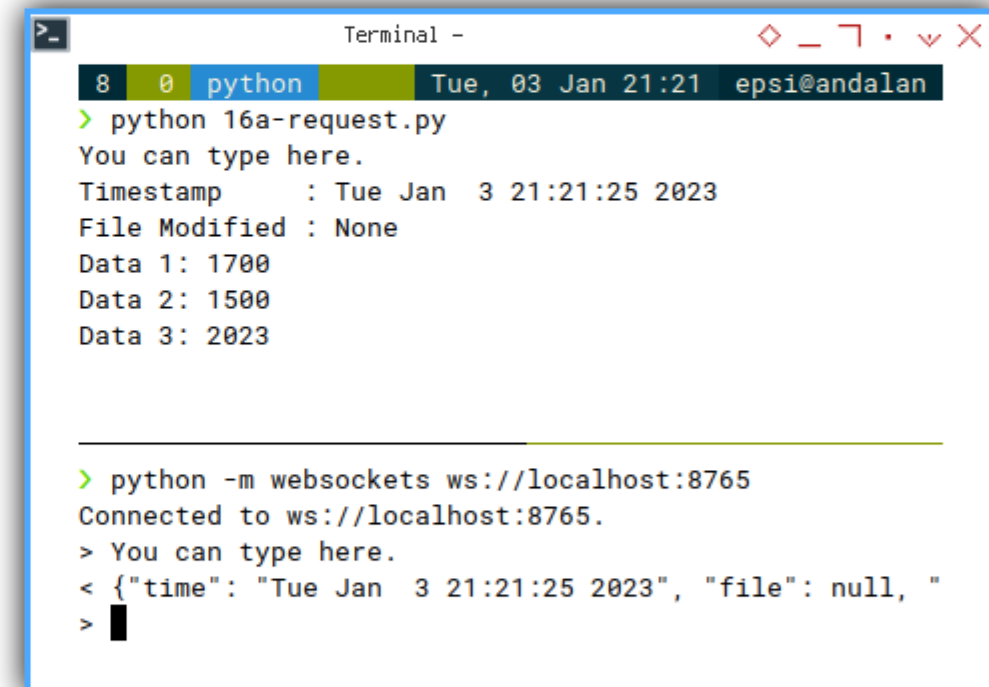
Reading Excel

Library: `openpyxl`



The screenshot shows the LibreOffice Calc application window titled "test-a.xlsx - LibreOffice Calc". The spreadsheet has columns A through E and rows 1 through 9. Column B contains labels "Data 1", "Data 2", and "Data 3" in rows 2, 3, and 4 respectively. Column C contains values "1700", "1500", and "2023" in rows 2, 3, and 4 respectively. The status bar at the bottom indicates "Sheet 1 of 1", "PageStyle_Example", "Indonesian", and "I-".

	A	B	C	D	E
1					
2		Data 1	1700		
3		Data 2	1500		
4		Data 3	2023		
5					
6					
7					
8					
9					



The screenshot shows a terminal window titled "Terminal -". The prompt is "epsi@andalan". The user has entered the command `python 16a-request.py`, which outputs "You can type here.", "Timestamp : Tue Jan 3 21:21:25 2023", "File Modified : None", "Data 1: 1700", "Data 2: 1500", and "Data 3: 2023". The user has then entered the command `python -m websockets ws://localhost:8765`, which outputs "Connected to ws://localhost:8765.", "> You can type here.", and "< {\"time\": \"Tue Jan 3 21:21:25 2023\", \"file\": null, \"".

```
8 0 python Tue, 03 Jan 21:21 epsi@andalan
> python 16a-request.py
You can type here.
Timestamp : Tue Jan 3 21:21:25 2023
File Modified : None
Data 1: 1700
Data 2: 1500
Data 3: 2023

> python -m websockets ws://localhost:8765
Connected to ws://localhost:8765.
> You can type here.
< {"time": "Tue Jan 3 21:21:25 2023", "file": null, "
```


Sending Data as Messages

Library: **websocket**, on specific port

The screenshot shows the Chrome DevTools Network tab with the 'WS' (WebSocket) filter selected. A single message is visible, sent from the client to the server. The message details are as follows:

Stat...	Method	Domain	File	Initiator	Type	Transferred	Size
101	GET	localhost:8765	/	16a-script.js:13 (web...	plain	276 B	0 B

Below the table, the 'Response' tab is selected, showing the received data:

```
↑ Hello There. 17:55:45.621
↓ {"time": "Wed Jan 4 17:55:45 2023", "file": null, "val1": 1700, "val2": 1500, "v... 17:55:45.633
```

The JSON view at the bottom shows the parsed data:

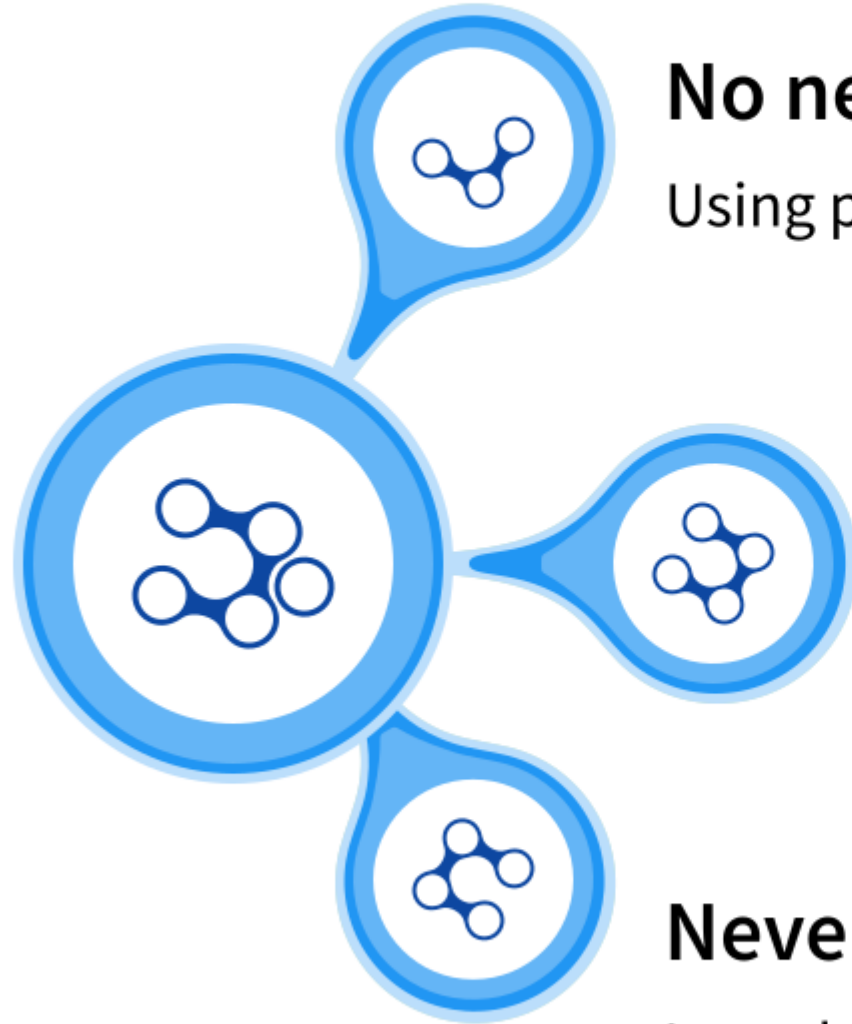
```
{
  "time": "Wed Jan 4 17:55:45 2023",
  "file": null,
  "val1": 1700,
  "val2": 1500,
  "val3": 2023
}
```

Summary: 2 messages, 104 B total, 12 B sent, 92 B received, 11 ms.

XI2WebExample	
-	filepath
-	filename
-	site
-	port
+	__init__
-	pack_data()
-	dump_data()
-	send_data()
-	monitor_localfile()
-	monitor_webclient()
-	handler()
+	main()

For Example: **ws://localhost:8765**

Websocket Notes



No need any web server

Using python script as websocket handler is enough

One webpage can handle many excel files.

Simple to manage

Never been tested in any large TV display

It works in my computer