



LAMINA STUDIOS, LLC DATA ANALYTICS INTERNSHIP

**H2O Wave: Beer Wall Tutorial Activity
Documentation**

Jewel Anne A. Reyes
BS Computer Science
Polytechnic University of the Philippines

September 1, 2023



Table of Contents

Table of Contents.....	2
Task Overview.....	3
Codes.....	4
Screenshots of Work.....	6

Task Overview

H2O Wave is a software stack for building beautiful, low-latency, realtime, browser-based applications and dashboards entirely in Python without using HTML, Javascript, or CSS.

It excels at capturing data, visualizations, and graphics from multiple sources and broadcasting them live over the web.

H2O Wave gives your Python programs the ability to push content to connected clients as it happens, in realtime. In other words, it lets your program display up-to-date information without asking your users to hit their browser's reload button. You can use H2O Wave for:

- Dashboards and visualizations for live monitoring.
- Live information displays: news, tickers, health, or performance data.
- Apps that require instant notifications, updates, events, or alerts.
- Apps that involve messaging: chat, bots, etc.
- Collaborative apps: whiteboards, sharing, etc.

You can also use H2O Wave when you find yourself reaching for a web application framework - it can handle regular (non-realtime) apps just fine.

In this task, we are required to create a program using H2O Wave and Python to run a page periodically and observe the changes live in the browser.

The program we'll be writing is a verse generator for the runaway mid-20th century smash hit, 99 Bottles of Beer, which looks something like this:

- 99 bottles of beer on the wall, 99 bottles of beer.
- Take one down, pass it around, 98 bottles of beer on the wall...

- 98 bottles of beer on the wall, 98 bottles of beer.
- Take one down, pass it around, 97 bottles of beer on the wall...

The program will be analogous to the "Hello, World!" program, with the addition of a loop. We'll generate a verse every second, and observe the verse change in the browser in realtime. After that, we'll take a stab at making our program a bit more efficient, introducing how expressions work.

Codes

The following are the steps with codes that was used for this task.

Step 1: Open a terminal and start the Wave Server

```
cd wave  
waved.exe
```

Step 2: Set up a virtual environment

```
python -m venv venv  
.\venv\Scripts\activate
```

Step 3: Write the program

```
import time  
from h2o_wave import site, ui  
  
page = site['/beer']  
  
beer_card = page.add('wall', ui.markdown_card(  
    box='1 1 4 2',  
    title='99 Bottles of Beer',  
    content='',  
))  
  
for i in range(99, 0, -1):  
    beer_card.content = f"""  
{i} bottles of beer on the wall, {i} bottles of beer.  
  
Take one down, pass it around, {i - 1} bottles of beer on the wall...  
"""  
    page.save()  
    time.sleep(1)
```

Step 4: Run the program

```
python beer_wall.py
```

Step 5: Write the program efficiently

```
import time
from h2o_wave import site, ui

page = site['/beer']

beer_verse = '''={{before}} bottles of beer on the wall, {{before}} bottles of
beer.

Take one down, pass it around, {{after}} bottles of beer on the wall...
'''

beer_card = page.add('wall', ui.markdown_card(
    box='1 1 4 2',
    title='99 Bottles of Beer',
    content=beer_verse,
    data=dict(before='99', after='98'),
))

for i in range(99, 0, -1):
    beer_card.data.before = str(i)
    beer_card.data.after = str(i - 1)
    page.save()
    time.sleep(1)
```

Step 6: Run the edited program

```
python efficient_beer_wall.py
```

Screenshots of Work

Figure 1. Start the wave server using waved.exe command

```
C:\Users\reyes\wave>waved.exe
2023/09/01 12:04:46 #
2023/09/01 12:04:47 #
2023/09/01 12:04:47 # H2O Wave
2023/09/01 12:04:47 # 0.26.2 20230803101450
2023/09/01 12:04:47 # © 2021 H2O.ai, Inc.
2023/09/01 12:04:47 #
2023/09/01 12:04:47 #
2023/09/01 12:04:47 # Running at http://localhost:10101/
2023/09/01 12:04:47 # {"address":"10101","base-url":"/","t":"listen","web-dir":"C:\\Users\\reyes\\wave\\www"}
2023/09/01 12:04:47 # {"error":"listen tcp :10101: bind: Only one usage of each socket address (protocol/network address/port) is normally permitted"}
2023/09/01 12:04:47 # {"t":"listen_no_tls"}
C:\Users\reyes\wave>2023/09/01 12:05:00 # {"addr":["::1":64126","route":"/","t":"ui_add"]}
```

Figure 2. Setting the virtual environment

```
Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

C:\Users\reyes\OneDrive\Desktop\h2o wave tutorials>cd C:\Users\reyes\OneDrive\Desktop\h2o wave tutorials\Act4_H2O_Beer Wall

C:\Users\reyes\OneDrive\Desktop\h2o wave tutorials\Act4_H2O_Beer Wall>python -m venv venv

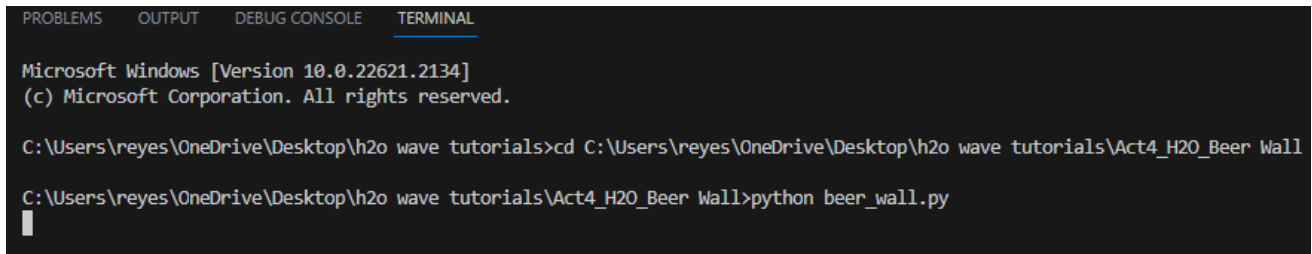
C:\Users\reyes\OneDrive\Desktop\h2o wave tutorials\Act4_H2O_Beer Wall>.\venv\Scripts\activate

(venv) C:\Users\reyes\OneDrive\Desktop\h2o wave tutorials\Act4_H2O_Beer Wall>
```

Figure 3. Python code in Visual Studio Code

```
beer_wall.py X
h2o wave tutorials > Act4_H2O_Beer Wall > beer_wall.py > ...
1 import time
2 from h2o_wave import site, ui
3
4 page = site['/beer']
5
6 beer_card = page.add('wall', ui.markdown_card(
7     box='1 1 4 2',
8     title='99 Bottles of Beer',
9     content='',
10 ))
11
12 for i in range(99, 0, -1):
13     beer_card.content = f"""
14 {i} bottles of beer on the wall, {i} bottles of beer.
15
16 Take one down, pass it around, {i - 1} bottles of beer on the wall...
17 """
18     page.save()
19     time.sleep(1)
```

Figure 4. Running the program in a terminal



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

C:\Users\reyes\OneDrive\Desktop\h2o wave tutorials>cd C:\Users\reyes\OneDrive\Desktop\h2o wave tutorials\Act4_H2O_Beer Wall
C:\Users\reyes\OneDrive\Desktop\h2o wave tutorials\Act4_H2O_Beer Wall>python beer_wall.py
█
```

Figure 5. Going to localhost:10101/beer and checking the output

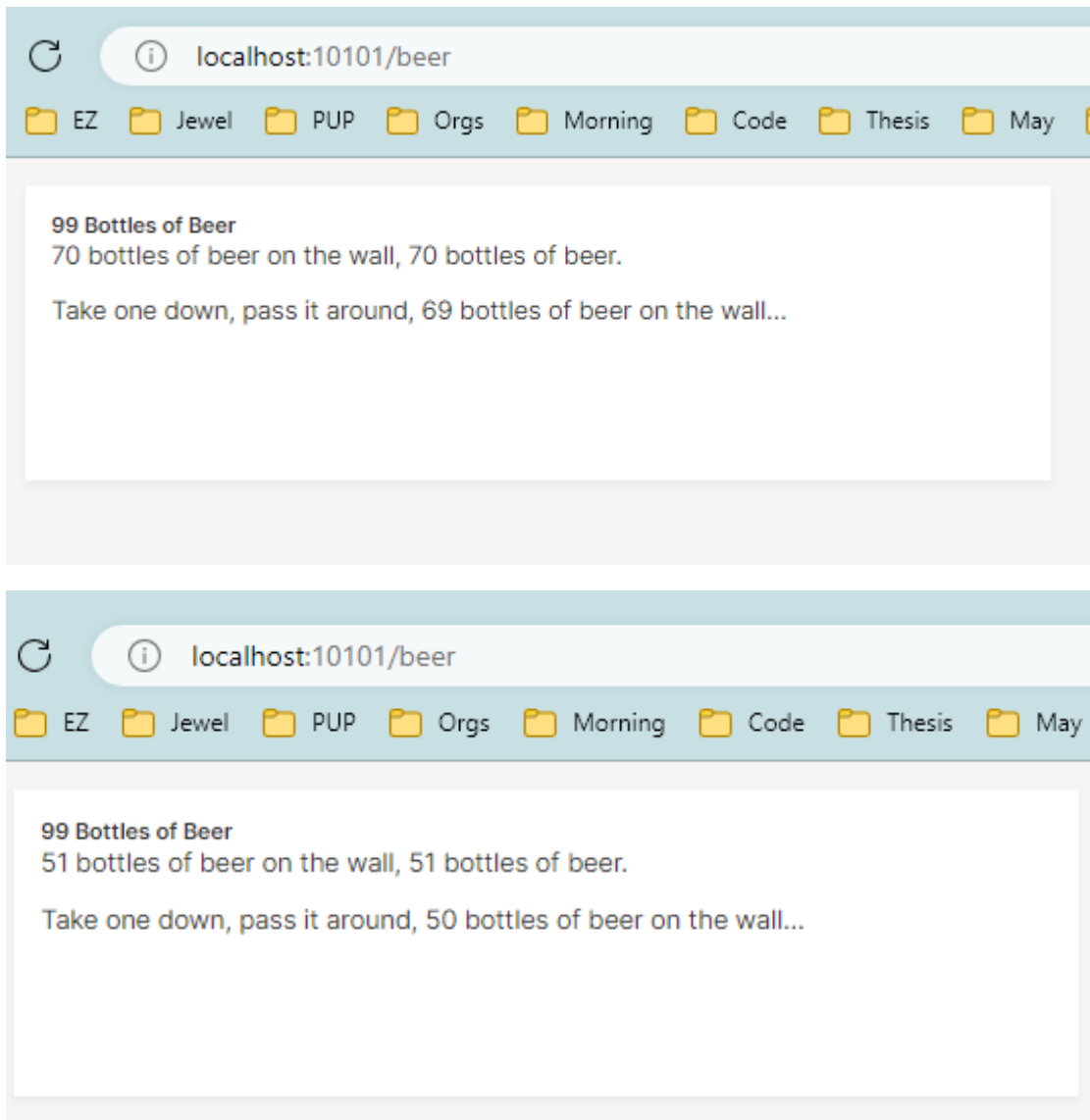


Figure 6. Wave Server Log

```

one down, pass it around, 14 bottles of beer on the wall...\n"]}]
2023/09/01 16:16:58 * /beer {"d":{"k":"wall content","v":"\n14 bottles of beer on the wall, 14 bottles of beer.\n\nTake
one down, pass it around, 13 bottles of beer on the wall...\n"]}]
2023/09/01 16:16:59 * /beer {"d":{"k":"wall content","v":"\n13 bottles of beer on the wall, 13 bottles of beer.\n\nTake
one down, pass it around, 12 bottles of beer on the wall...\n"]}]
2023/09/01 16:17:00 * /beer {"d":{"k":"wall content","v":"\n12 bottles of beer on the wall, 12 bottles of beer.\n\nTake
one down, pass it around, 11 bottles of beer on the wall...\n"]}]
2023/09/01 16:17:01 * /beer {"d":{"k":"wall content","v":"\n11 bottles of beer on the wall, 11 bottles of beer.\n\nTake
one down, pass it around, 10 bottles of beer on the wall...\n"]}]
2023/09/01 16:17:02 * /beer {"d":{"k":"wall content","v":"\n10 bottles of beer on the wall, 10 bottles of beer.\n\nTake
one down, pass it around, 9 bottles of beer on the wall...\n"]}]
2023/09/01 16:17:03 * /beer {"d":{"k":"wall content","v":"\n9 bottles of beer on the wall, 9 bottles of beer.\n\nTake o
ne down, pass it around, 8 bottles of beer on the wall...\n"]}]
2023/09/01 16:17:04 * /beer {"d":{"k":"wall content","v":"\n8 bottles of beer on the wall, 8 bottles of beer.\n\nTake o
ne down, pass it around, 7 bottles of beer on the wall...\n"]}]
2023/09/01 16:17:05 * /beer {"d":{"k":"wall content","v":"\n7 bottles of beer on the wall, 7 bottles of beer.\n\nTake o
ne down, pass it around, 6 bottles of beer on the wall...\n"]}]
2023/09/01 16:17:06 * /beer {"d":{"k":"wall content","v":"\n6 bottles of beer on the wall, 6 bottles of beer.\n\nTake o
ne down, pass it around, 5 bottles of beer on the wall...\n"]}]
2023/09/01 16:17:07 * /beer {"d":{"k":"wall content","v":"\n5 bottles of beer on the wall, 5 bottles of beer.\n\nTake o
ne down, pass it around, 4 bottles of beer on the wall...\n"]}]
2023/09/01 16:17:08 * /beer {"d":{"k":"wall content","v":"\n4 bottles of beer on the wall, 4 bottles of beer.\n\nTake o
ne down, pass it around, 3 bottles of beer on the wall...\n"]}]
2023/09/01 16:17:09 * /beer {"d":{"k":"wall content","v":"\n3 bottles of beer on the wall, 3 bottles of beer.\n\nTake o
ne down, pass it around, 2 bottles of beer on the wall...\n"]}]
2023/09/01 16:17:10 * /beer {"d":{"k":"wall content","v":"\n2 bottles of beer on the wall, 2 bottles of beer.\n\nTake o
ne down, pass it around, 1 bottles of beer on the wall...\n"]}]
2023/09/01 16:17:11 * /beer {"d":{"k":"wall content","v":"\n1 bottles of beer on the wall, 1 bottles of beer.\n\nTake o
ne down, pass it around, 0 bottles of beer on the wall...\n"]}]

```

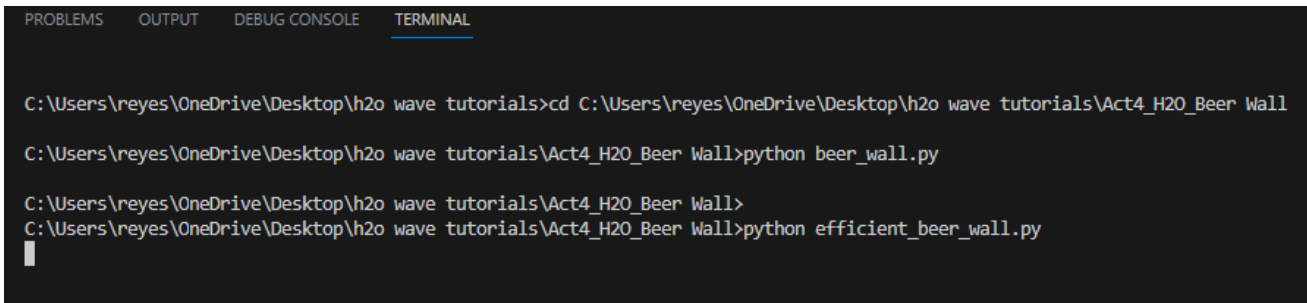
Figure 7. More efficient Python Code in Visual Code Studio

```

beer_wall.py  efficient_beer_wall.py X
h2o wave tutorials > Act4_H2O_Beer Wall > efficient_beer_wall.py > ...
4  page = site['/beer']
5
6  beer_verse = '''= {{before}} bottles of beer on the wall, {{before}} bottles of beer.
7
8  Take one down, pass it around, {{after}} bottles of beer on the wall...
9  ...
10
11  beer_card = page.add('wall', ui.markdown_card(
12      box='1 1 4 2',
13      title='99 Bottles of Beer',
14      content=beer_verse,
15      data=dict(before='99', after='98'),
16  ))
17
18  for i in range(99, 0, -1):
19      beer_card.data.before = str(i)
20      beer_card.data.after = str(i - 1)
21      page.save()
22      time.sleep(1)

```


Figure 8. Running the program in a terminal



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

C:\Users\reyes\OneDrive\Desktop\h2o wave tutorials>cd C:\Users\reyes\OneDrive\Desktop\h2o wave tutorials\Act4_H2O_Beer Wall

C:\Users\reyes\OneDrive\Desktop\h2o wave tutorials\Act4_H2O_Beer Wall>python beer_wall.py

C:\Users\reyes\OneDrive\Desktop\h2o wave tutorials\Act4_H2O_Beer Wall>
C:\Users\reyes\OneDrive\Desktop\h2o wave tutorials\Act4_H2O_Beer Wall>python efficient_beer_wall.py
```

Figure 9. Going to localhost:10101/beer and checking the output

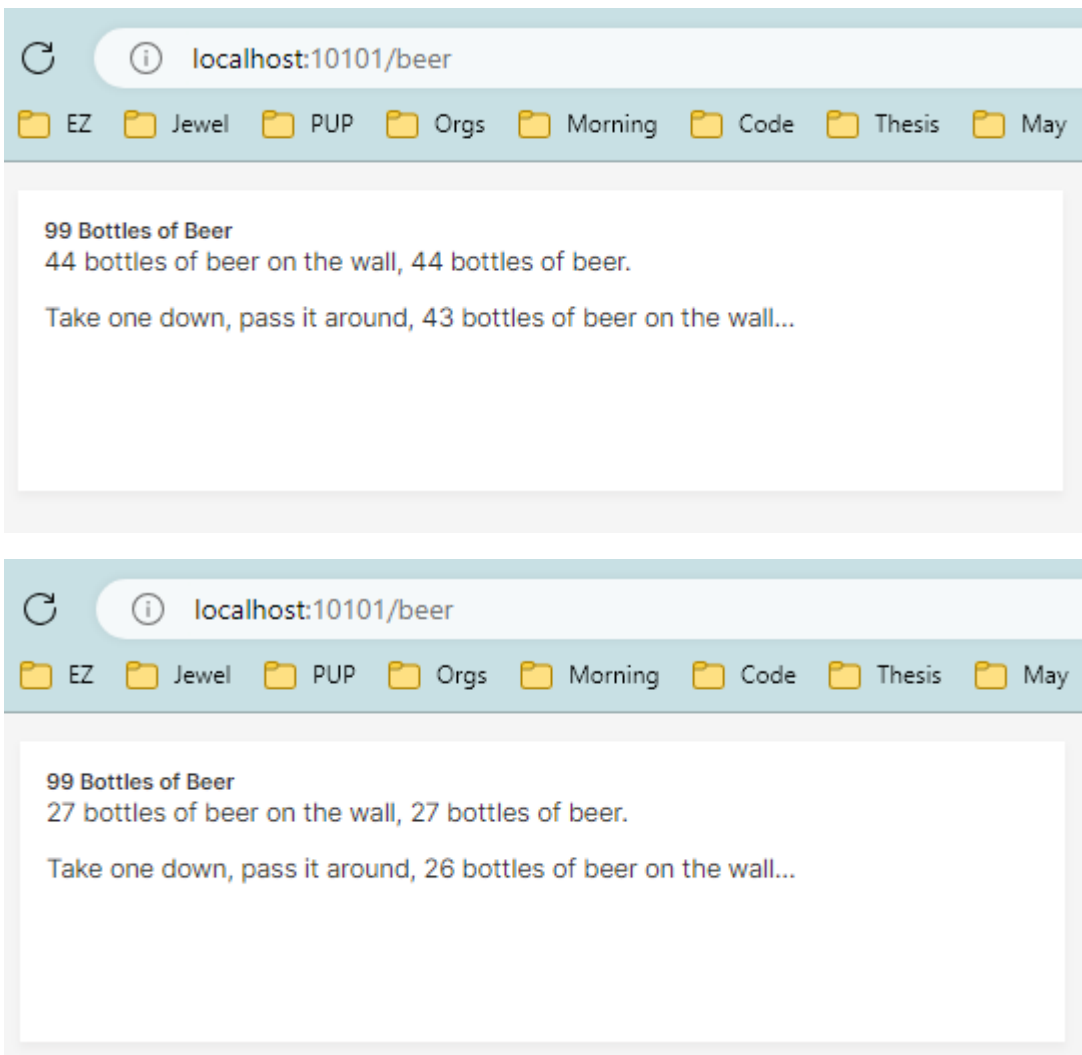


Figure 10. Wave Server Log

```
Command Prompt - waved.exe X + v
2023/09/01 16:28:54 * /beer {"d":[{"k":"wall data before","v":"29"}, {"k":"wall data after","v":"28"}]}
2023/09/01 16:28:55 * /beer {"d":[{"k":"wall data before","v":"28"}, {"k":"wall data after","v":"27"}]}
2023/09/01 16:28:56 * /beer {"d":[{"k":"wall data before","v":"27"}, {"k":"wall data after","v":"26"}]}
2023/09/01 16:28:57 * /beer {"d":[{"k":"wall data before","v":"26"}, {"k":"wall data after","v":"25"}]}
2023/09/01 16:28:58 * /beer {"d":[{"k":"wall data before","v":"25"}, {"k":"wall data after","v":"24"}]}
2023/09/01 16:28:59 * /beer {"d":[{"k":"wall data before","v":"24"}, {"k":"wall data after","v":"23"}]}
2023/09/01 16:29:00 * /beer {"d":[{"k":"wall data before","v":"23"}, {"k":"wall data after","v":"22"}]}
2023/09/01 16:29:01 * /beer {"d":[{"k":"wall data before","v":"22"}, {"k":"wall data after","v":"21"}]}
2023/09/01 16:29:02 * /beer {"d":[{"k":"wall data before","v":"21"}, {"k":"wall data after","v":"20"}]}
2023/09/01 16:29:03 * /beer {"d":[{"k":"wall data before","v":"20"}, {"k":"wall data after","v":"19"}]}
2023/09/01 16:29:04 * /beer {"d":[{"k":"wall data before","v":"19"}, {"k":"wall data after","v":"18"}]}
2023/09/01 16:29:05 * /beer {"d":[{"k":"wall data before","v":"18"}, {"k":"wall data after","v":"17"}]}
2023/09/01 16:29:06 * /beer {"d":[{"k":"wall data before","v":"17"}, {"k":"wall data after","v":"16"}]}
2023/09/01 16:29:07 * /beer {"d":[{"k":"wall data before","v":"16"}, {"k":"wall data after","v":"15"}]}
2023/09/01 16:29:08 * /beer {"d":[{"k":"wall data before","v":"15"}, {"k":"wall data after","v":"14"}]}
2023/09/01 16:29:09 * /beer {"d":[{"k":"wall data before","v":"14"}, {"k":"wall data after","v":"13"}]}
2023/09/01 16:29:10 * /beer {"d":[{"k":"wall data before","v":"13"}, {"k":"wall data after","v":"12"}]}
2023/09/01 16:29:11 * /beer {"d":[{"k":"wall data before","v":"12"}, {"k":"wall data after","v":"11"}]}
2023/09/01 16:29:12 * /beer {"d":[{"k":"wall data before","v":"11"}, {"k":"wall data after","v":"10"}]}
2023/09/01 16:29:13 * /beer {"d":[{"k":"wall data before","v":"10"}, {"k":"wall data after","v":"9"}]}
2023/09/01 16:29:14 * /beer {"d":[{"k":"wall data before","v":"9"}, {"k":"wall data after","v":"8"}]}
2023/09/01 16:29:15 * /beer {"d":[{"k":"wall data before","v":"8"}, {"k":"wall data after","v":"7"}]}
2023/09/01 16:29:16 * /beer {"d":[{"k":"wall data before","v":"7"}, {"k":"wall data after","v":"6"}]}
2023/09/01 16:29:17 * /beer {"d":[{"k":"wall data before","v":"6"}, {"k":"wall data after","v":"5"}]}
2023/09/01 16:29:18 * /beer {"d":[{"k":"wall data before","v":"5"}, {"k":"wall data after","v":"4"}]}
2023/09/01 16:29:19 * /beer {"d":[{"k":"wall data before","v":"4"}, {"k":"wall data after","v":"3"}]}
2023/09/01 16:29:20 * /beer {"d":[{"k":"wall data before","v":"3"}, {"k":"wall data after","v":"2"}]}
2023/09/01 16:29:21 * /beer {"d":[{"k":"wall data before","v":"2"}, {"k":"wall data after","v":"1"}]}
2023/09/01 16:29:22 * /beer {"d":[{"k":"wall data before","v":"1"}, {"k":"wall data after","v":"0"}]}
```