



**COLLEGE OF ENGINEERING AND COMPUTER
SCIENCE**

Fall 2020

CPSC 449 Web Backend

Project 3 - Documentation

Prepared By:

<u>CWID</u>	<u>Name</u>
<u>887480747</u>	<u>Venkata Pranathi Immaneni</u>

Table of Contents

1.	<i>Cookie-based sessions</i>	3
2.	<i>Key-value store service</i>	9
3	<i>Server Side Sessions</i>	12
4	<i>Testing</i>	14

1. Cookie-based sessions

STEP 1:

- Initially **USE_SESSION_STORE** is set to **False** in counter.cfg. When this set to False, Flask uses the default session for storing the session variable count in a signed cookie on the server-side.

STEP 2:

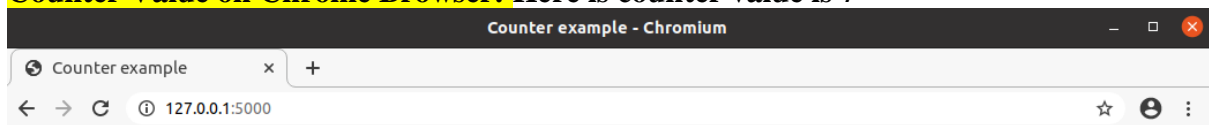
Observation of **count** variable on **Firefox** and **Chrome**:

- Download the project from “<https://github.com/ProfAvery/cpsc449>”
- Navigate to the “counter” folder.
- Open the terminal and set the path to the “counter” folder.
- Then run the following command: `foreman start`
- You will have the app running on localhost on 5000 port.
- Open the browser windows and type - `127.0.0.1:5000`

Let's observe Count values in Chrome and Firefox, by refreshing the browser windows, respectively.

Test - 1:

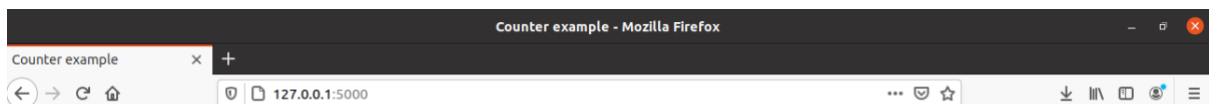
Counter Value on Chrome Browser: Here is counter value is 7



Count: 7

Reset

Counter Value on Firefox Browser: Here is counter value is 10

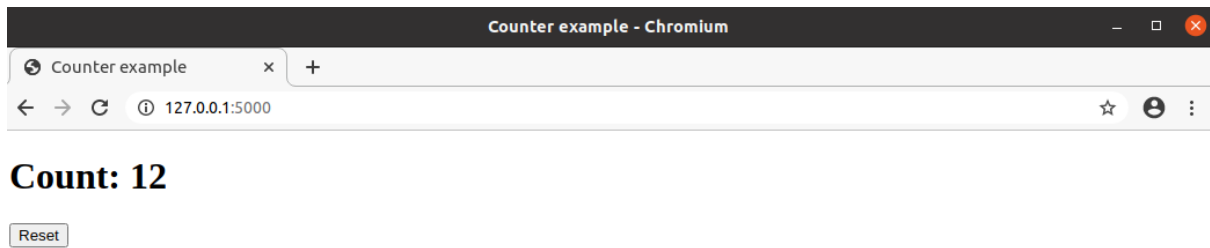


Count: 10

Reset

Test - 2:

Counter Value on Chrome Browser: Here is counter value is 12

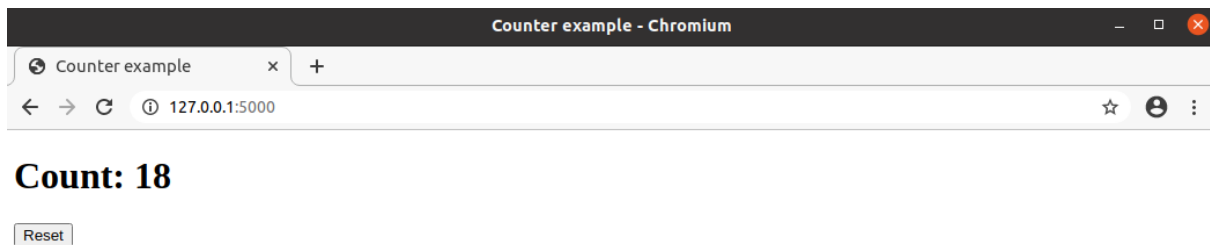


Counter Value on Firefox Browser: Here is counter value is 16



Test - 3:

Counter Value on Chrome Browser: Here is counter value is 18

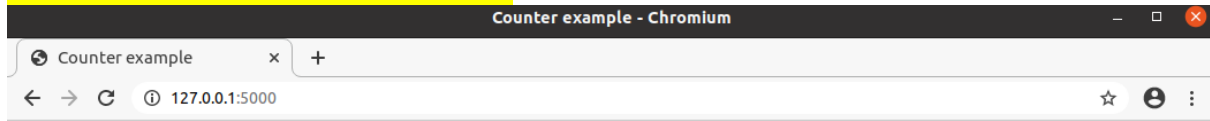


Counter Value on Firefox Browser: Here is counter value is 24



Test - 4:

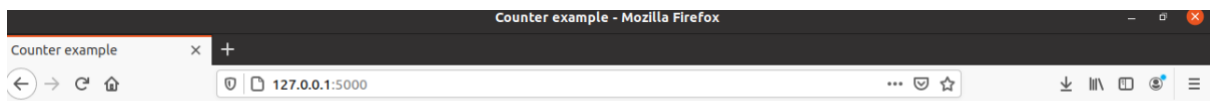
Counter Value on Chrome Browser: Here is counter value is 32



Count: 32

Reset

Counter Value on Firefox Browser: Here is counter value is 28

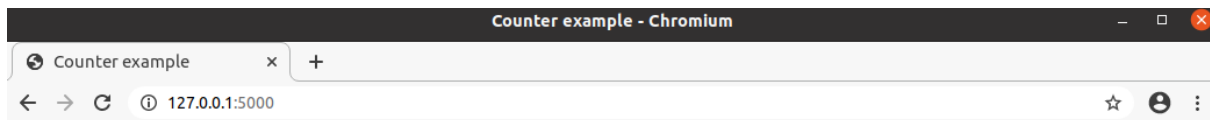


Count: 28

Reset

Test - 5:

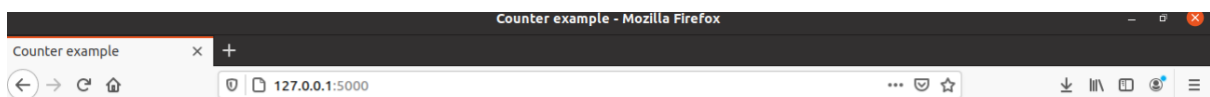
Counter Value on Chrome Browser: Here is counter value is 34



Count: 34

Reset

Counter Value on Firefox Browser: Here is counter value is 30



Count: 30

Reset

OBSERVATION: As we observe from the above screenshots, in each of the tests, each time we refresh the browsers, the count is getting incremented. Chrome and Firefox are maintaining different count values.

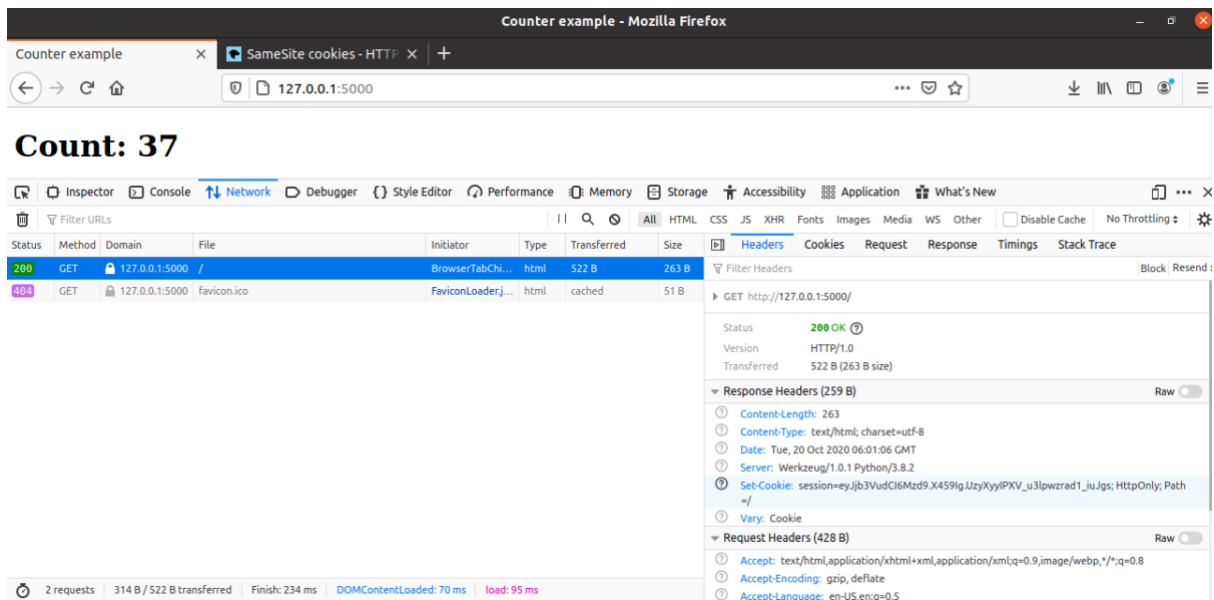
STEP 3:

Observation of **Set-Cookie** Response Header:

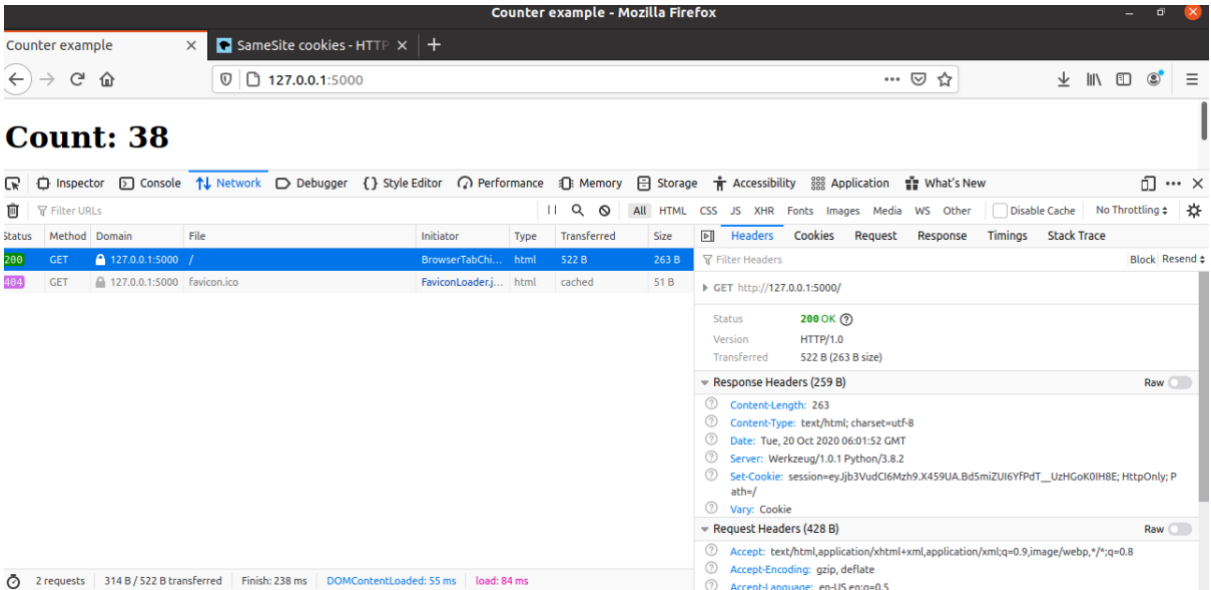
Steps to open developer tools:

- Open Developer tool, either by right-clicking and selecting “Inspect Element” or Fn+F12
- Next, click on the “Network” tab and refresh the browser.
- Then click on the First row showing the GET method. On the right side, we will be able to see the screen as below.

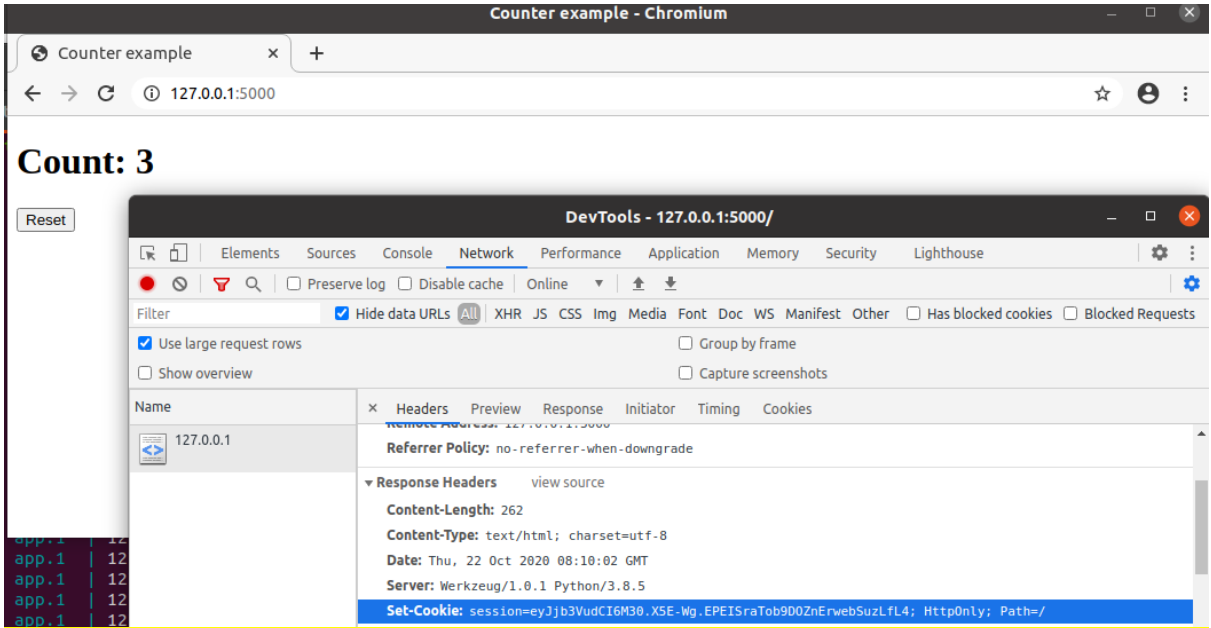
Firefox Browser



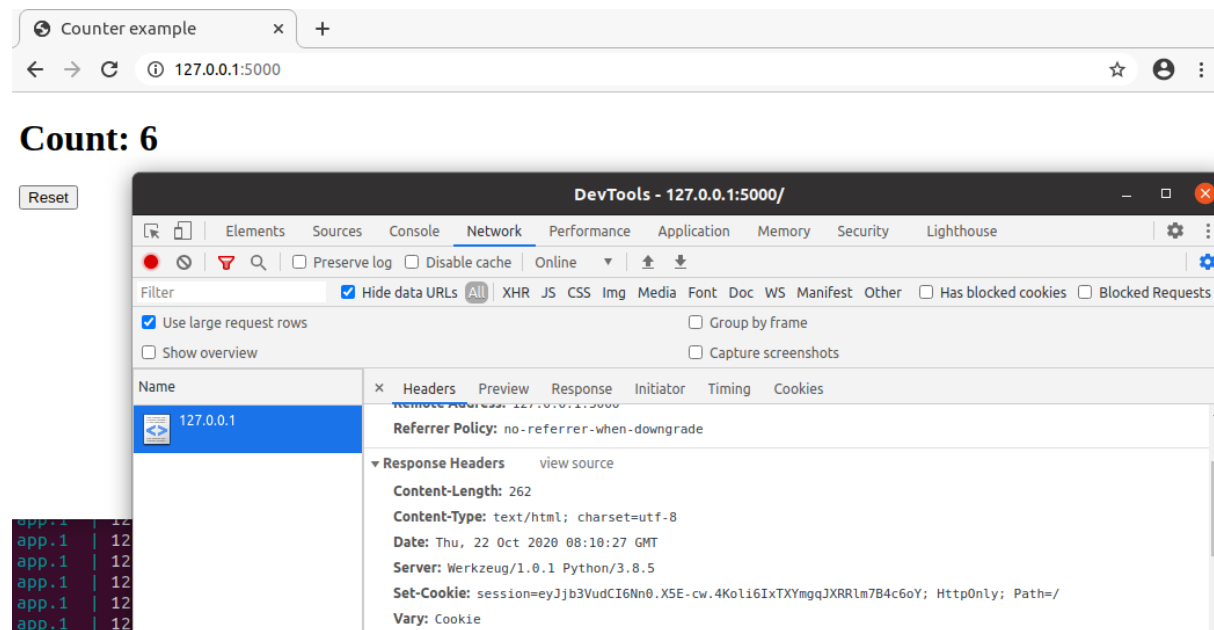
Firefox Browser



Chrome Browser



Chrome Browser



OBSERVATION: As we observe from the above screenshots, each time we refresh the browser, the value of sessionId of Set-Cookie, changes and is **not persisted**.

2. Key-value store service

STEP 4:

The file kv.py implements Key-value store. It exposes following 4 methods:

POST / - sets a new value for a key

GET /<key> - retrieves a value of a key

DELETE /<key> - removes a key

GET /?prefix=PREFIX - returns a list of all keys matching PREFIX

- Open a terminal and start this service by typing the following command:

```
export FLASK_APP=kv
Flask run
```

Below screenshots demonstrate the use of all the above Methods:

```
student@tuffix-vm: ~/Desktop/Project3/cpsc449-master/counter$ http localhost:5000 foo=bar
HTTP/1.0 200 OK
Content-Length: 19
Content-Type: application/json
Date: Tue, 20 Oct 2020 06:41:22 GMT
Server: Werkzeug/1.0.1 Python/3.8.2

{"foo": "bar"}

student@tuffix-vm: ~/Desktop/Project3/cpsc449-master/counter$ http localhost:5000/foo
HTTP/1.0 200 OK
Content-Length: 19
Content-Type: application/json
Date: Tue, 20 Oct 2020 06:42:25 GMT
Server: Werkzeug/1.0.1 Python/3.8.2

{"foo": "bar"}
```

```
student@tufflix-vm: ~/Desktop/Project3/cpsc449-master/counter
}

student@tufflix-vm:~/Desktop/Project3/cpsc449-master/counter$ http localhost:5000 pranathi=dinesh
HTTP/1.0 200 OK
Content-Length: 27
Content-Type: application/json
Date: Tue, 20 Oct 2020 06:43:29 GMT
Server: Werkzeug/1.0.1 Python/3.8.2

{
  "pranathi": "dinesh"
}

student@tufflix-vm:~/Desktop/Project3/cpsc449-master/counter$ http localhost:5000 praveena=nandan
HTTP/1.0 200 OK
Content-Length: 27
Content-Type: application/json
Date: Tue, 20 Oct 2020 06:43:41 GMT
Server: Werkzeug/1.0.1 Python/3.8.2

{
  "praveena": "nandan"
}

student@tufflix-vm:~/Desktop/Project3/cpsc449-master/counter$ http localhost:5000 prathina=krishnamohan
HTTP/1.0 200 OK
Content-Length: 33
Content-Type: application/json
```

POST Methods: Inserting 3 pairs of Keys and values, to demonstrate GET Prefix method for the next steps

```
student@tufflix-vm:~/Desktop/Project3/cpsc449-master/counter$ http localhost:5000?prefix=p
HTTP/1.0 200 OK
Content-Length: 69
Content-Type: application/json
Date: Tue, 20 Oct 2020 06:45:10 GMT
Server: Werkzeug/1.0.1 Python/3.8.2

{
  "keys": [
    "pranathi",
    "praveena",
    "prathina"
  ]
}

student@tufflix-vm:~/Desktop/Project3/cpsc449-master/counter$ http DELETE localhost:5000/pranathi
HTTP/1.0 200 OK
Content-Length: 27
Content-Type: application/json
Date: Tue, 20 Oct 2020 06:46:39 GMT
Server: Werkzeug/1.0.1 Python/3.8.2

{
  "pranathi": "dinesh"
}
```

GET Prefix method: Fetching all the keys with prefix p

DELETE method: Deleting the key-value pair Pranathi

```
student@tufflix-vm:~/Desktop/Project3/cpsc449-master/counter$ http localhost:5000?prefix=p
HTTP/1.0 200 OK
Content-Length: 52
Content-Type: application/json
Date: Tue, 20 Oct 2020 06:46:43 GMT
Server: Werkzeug/1.0.1 Python/3.8.2

{
  "keys": [
    "praveena",
    "prathina"
  ]
}
```

GET method: Once the key Pranathi is delete, remaining keys with prefix p are fetched using GET method

STEP 5:

Demonstration of Requests, to call the methods of kv.py from a python file. Here we create Dump.py and call all of the 4 methods of the above kv service.

Populating the kv store with the below keys and values:

```
student@tuffix-vm: ~/Desktop/Proje... x student@tuffix-vm: ~/Desktop/Proje... x student@tuffix-vm: ~/Desktop/Proje... x student@tuffix-vm: ~/Desktop/Proje... x
student@tuffix-vm: ~/Desktop/Project3/cpsc449-master/counter$ http localhost:5000 twitter=https://twitter.com/ProfAvery
HTTP/1.0 200 OK
Content-Length: 49
Content-Type: application/json
Date: Tue, 20 Oct 2020 08:30:21 GMT
Server: Werkzeug/1.0.1 Python/3.8.2

{"twitter": "https://twitter.com/ProfAvery"}

student@tuffix-vm: ~/Desktop/Project3/cpsc449-master/counter$ http localhost:5000 github=https://github.com/ProfAvery
HTTP/1.0 200 OK
Content-Length: 47
Content-Type: application/json
Date: Tue, 20 Oct 2020 08:30:44 GMT
Server: Werkzeug/1.0.1 Python/3.8.2

{"github": "https://github.com/ProfAvery"}

student@tuffix-vm: ~/Desktop/Project3/cpsc449-master/counter$ http localhost:5000 cpssc449=https://sites.google.com/view/cpsc449
HTTP/1.0 200 OK
Content-Length: 57
Content-Type: application/json
Date: Tue, 20 Oct 2020 08:31:23 GMT
Server: Werkzeug/1.0.1 Python/3.8.2

{"cpssc449": "https://sites.google.com/view/cpsc449"}
```

After implementing dump.py and executing below command:

\$./dump.py <http://localhost:5000>

Yields the following output:

```
student@tuffix-vm: ~/Desktop/Project3/cpsc449-master/counter$ ./dummy.py http://localhost:5000
{
  "twitter": "https://twitter.com/ProfAvery"
}

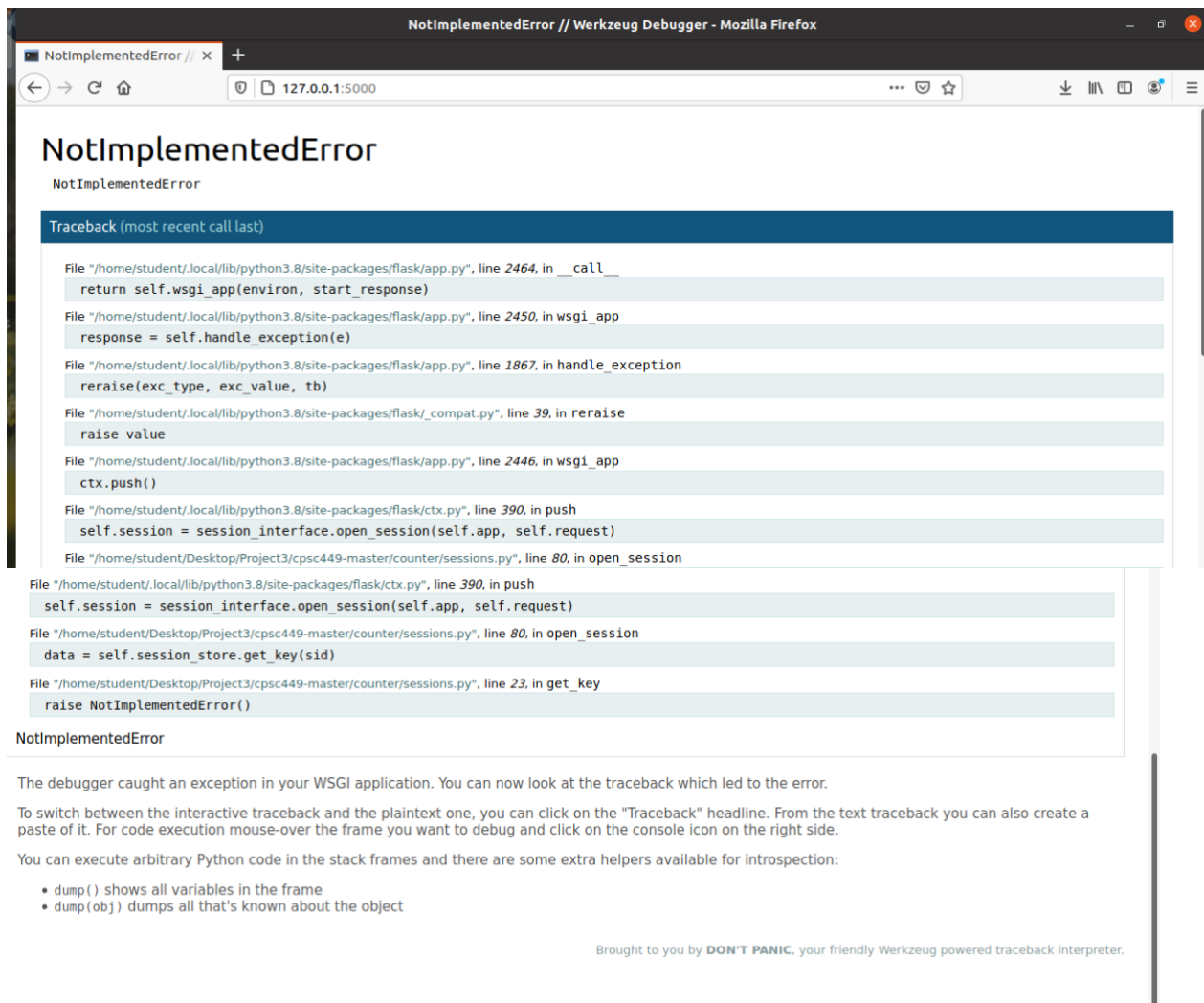
{
  "github": "https://github.com/ProfAvery"
}

{
  "cpssc449": "https://sites.google.com/view/cpsc449"
}
```

3 Server Side Sessions

STEP 6:

- Modify counter.cfg by setting USE_SESSION_STORE= True and uncomment DBM='kv.db' line.
- Open the terminal and start the key-value store service using Foreman start.
- Note that, app.py will be running on 5000 port of localhost and kv.py will be running on 5100 of localhost.
- We have not implemented the methods of **KeyValueSessionStore**. So, below failure message will be displayed.



NotImplementedError

NotImplementedError

Traceback (most recent call last)

```

File "/home/student/.local/lib/python3.8/site-packages/flask/app.py", line 2464, in __call__
    return self.wsgi_app(environ, start_response)
File "/home/student/.local/lib/python3.8/site-packages/flask/app.py", line 2450, in wsgi_app
    response = self.handle_exception(e)
File "/home/student/.local/lib/python3.8/site-packages/flask/app.py", line 1867, in handle_exception
    reraise(exc_type, exc_value, tb)
File "/home/student/.local/lib/python3.8/site-packages/flask/_compat.py", line 39, in reraise
    raise value
File "/home/student/.local/lib/python3.8/site-packages/flask/app.py", line 2446, in wsgi_app
    ctx.push()
File "/home/student/.local/lib/python3.8/site-packages/flask/ctx.py", line 390, in push
    self.session = session_interface.open_session(self.app, self.request)
File "/home/student/Desktop/Project3/cpsc449-master/counter/sessions.py", line 80, in open_session
    data = self.session_store.get_key(sid)
File "/home/student/Desktop/Project3/cpsc449-master/counter/sessions.py", line 23, in get_key
    raise NotImplementedError()

```

NotImplementedError

The debugger caught an exception in your WSGI application. You can now look at the traceback which led to the error.

To switch between the interactive traceback and the plaintext one, you can click on the "Traceback" headline. From the text traceback you can also create a paste of it. For code execution mouse-over the frame you want to debug and click on the console icon on the right side.

You can execute arbitrary Python code in the stack frames and there are some extra helpers available for introspection:

- `dump()` shows all variables in the frame
- `dump(obj)` dumps all that's known about the object

Brought to you by **DON'T PANIC**, your friendly Werkzeug powered traceback interpreter.

STEP 7:

Now, let's override the get, set and delete methods of KeyValueSessionStore. As we have demonstrated in Dump.py file, we need to call the methods of kv service from KeyValueSessionStore to insert session id as Key and count values as Value.

Below is the snapshot of the code snippet of implementation of **KeyValueSessionStore**:

```
class KeyValueSessionStore(SessionStore):

    def __init__(self, url, logger=None):
        super().__init__(logger)

    def set_key(self, key, value):

        #Creating a dictionary to pass key and value as JSON to the post method
        dict = {key: value}

        #Setting the headers to send in the request
        setHeaders = {'Content-type': 'application/json', 'Accept': 'text/plain'}

        #POST Request to kv.py. Key is the session Id and value will be the counter
        return requests.post('http://localhost:5100/', json=dict, headers=setHeaders)

    def delete_key(self, key):
        #DELETE request to kv.py to delete the specified key
        return requests.delete("http://localhost:5100/"+key)

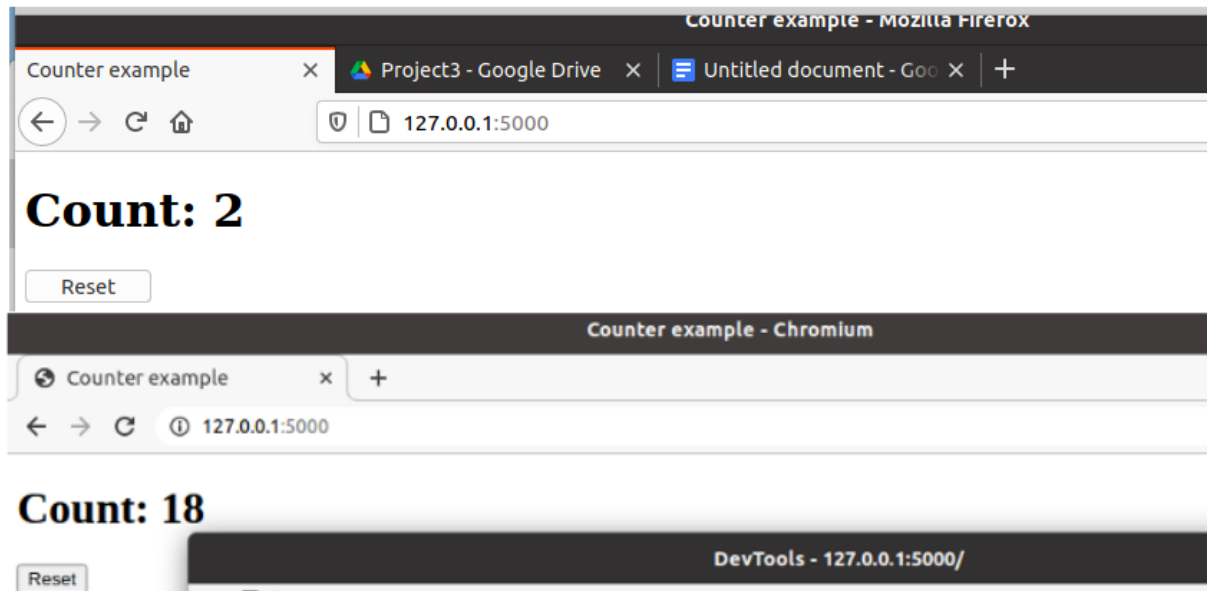
    def get_key(self, key):

        #GET request to kv.py by passing the key.
        res = requests.get("http://localhost:5100/"+key)
        #Retrieving the counter of the specified key
        return res.json().get(key)
```

4 Testing

STEP 8:

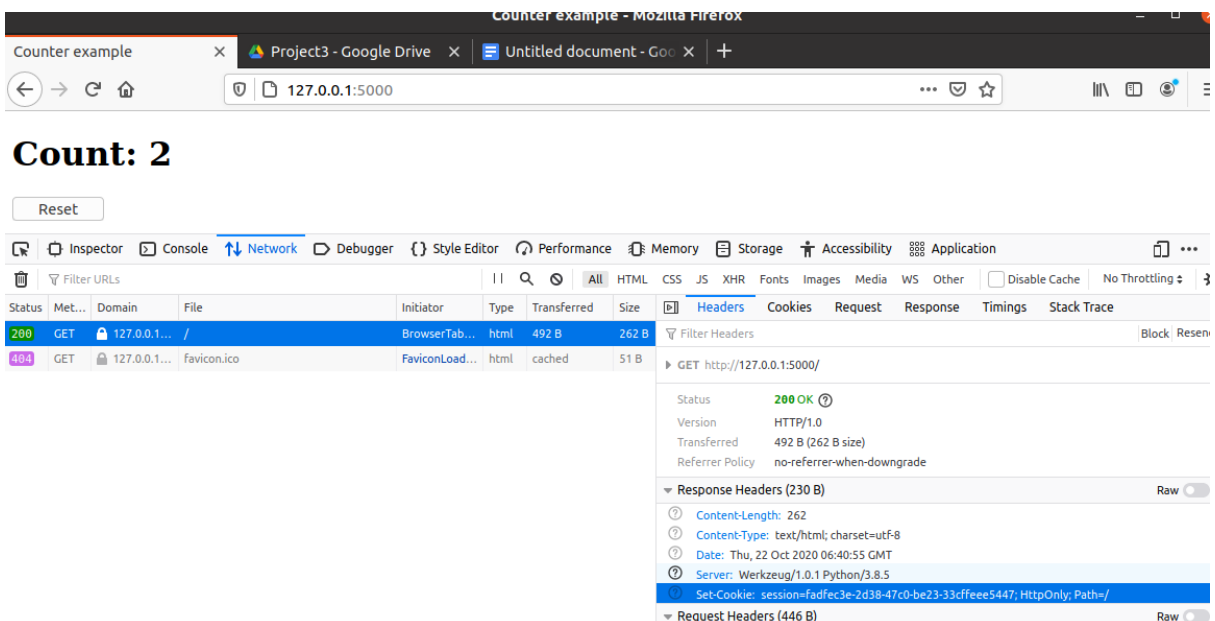
Now, Let's check the session behavior in both the browsers from step 9. As in step 2, counter value increments on each refresh. Each browser maintains separate counter values.



STEP 9:

Let's observe the value of session of SetCookie by refreshing the browser.

SetCookie session Value on Mozilla Browser– when count=2: fadfec3e-2d38-47c0-be23-33cffee5447



SetCookie session Value on Mozilla Browser – when count=6: fadfec3e-2d38-47c0-be23-33cffee5447

Counter example - Mozilla Firefox

Counter example x Project3 - Google Drive x Untitled document - Google Docs x +

127.0.0.1:5000

Count: 6

Reset

Inspector Console Network Debugger Style Editor Performance Memory Storage Accessibility Application

Filter URLs

Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	GET	127.0.0.1...	/	BrowserTab...	html	492 B	262 B
404	GET	127.0.0.1...	favicon.ico	FaviconLoad...	html	cached	51 B

Filter Headers

GET http://127.0.0.1:5000/

Status: 200 OK

Version: HTTP/1.0

Transferred: 492 B (262 B size)

Referrer Policy: no-referrer-when-downgrade

Response Headers (230 B)

Content-Length: 262

Content-Type: text/html; charset=utf-8

Date: Thu, 22 Oct 2020 06:43:34 GMT

Server: Werkzeug/1.0.1 Python/3.8.5

Set-Cookie: session=fadfec3e-2d38-47c0-be23-33cffee5447; HttpOnly; Path=/

SetCookie session Value on Mozilla Browser – when count=11 fadfec3e-2d38-47c0-be23-33cffee5447

Counter example - Mozilla Firefox

Counter example x Project3 - Google Drive x Untitled document - Google Docs x +

127.0.0.1:5000

Count: 11

Reset

Inspector Console Network Debugger Style Editor Performance Memory Storage Accessibility Application

Filter URLs

Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	GET	127.0.0.1...	/	BrowserTab...	html	493 B	263 B
404	GET	127.0.0.1...	favicon.ico	FaviconLoad...	html	cached	51 B

Filter Headers

GET http://127.0.0.1:5000/

Status: 200 OK

Version: HTTP/1.0

Transferred: 493 B (263 B size)

Referrer Policy: no-referrer-when-downgrade

Response Headers (230 B)

Content-Length: 263

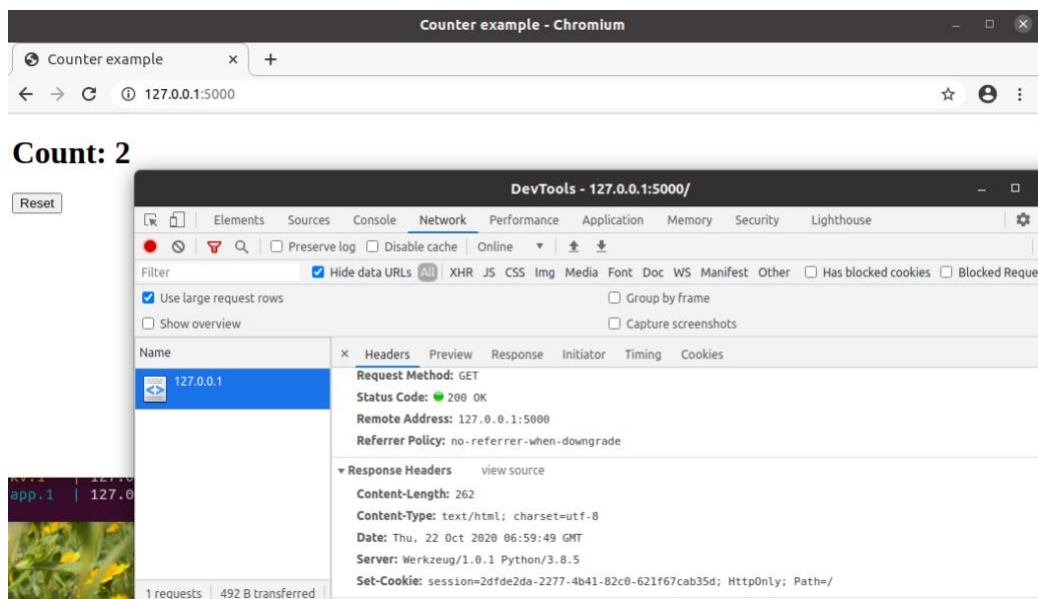
Content-Type: text/html; charset=utf-8

Date: Thu, 22 Oct 2020 06:44:22 GMT

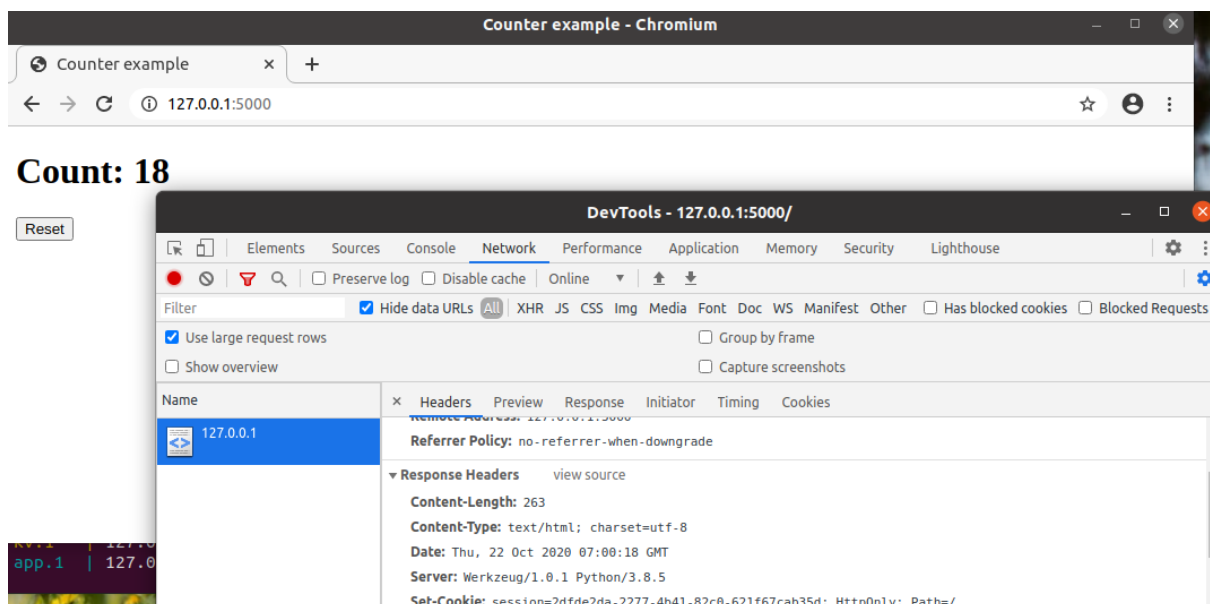
Server: Werkzeug/1.0.1 Python/3.8.5

Set-Cookie: session=fadfec3e-2d38-47c0-be23-33cffee5447; HttpOnly; Path=/

SetCookie session Value on Chrome Browser – when count=2: 2dfde2da-2277-4b41-82c0-621f67cab35d



SetCookie session Value on Chrome Browser – when count=18: 2dfde2da-2277-4b41-82c0-621f67cab35d



OBSERVATION: As we observe from the above screenshots, in each of the tests, each time we refresh the **same** browser, the count is getting incremented and session value of all the SetCookie key remains the same for the **same** browser. For two different browsers two different sessions are maintained. Therefore, the session ids are persisted in the shared session store.

STEP 10:

Now, clicking reset button, clears the session id and new session id gets created once the browser is refreshed again.

On Reset -Firefox browser: session id of SetCookie get cleared

Counter example - Mozilla Firefox

Counter example x Project3 - Google Drive x Untitled document - Google Docs x +

127.0.0.1:5000

Count: 1

Reset

Inspector Console Network Debugger Style Editor Performance Memory Storage Accessibility Application

Filter URLs

Status	Method	Domain	File	Initiator	Type	Transferred	Size	Headers	Cookies	Request	Response	Timings
302	POST	127.0.0.1...	reset	document	html	533 B	262 B	Transferred	533 B (262 B size)			
200	GET	127.0.0.1...	/	document	html	492 B	262 B	Referrer Policy	no-referrer-when-downgrade			
404	GET	127.0.0.1...	favicon.ico	FaviconLoad...	html	cached	51 B					

Response Headers (271 B)

- Content-Length: 209
- Content-Type: text/html; charset=utf-8
- Date: Thu, 22 Oct 2020 07:12:55 GMT
- Location: http://127.0.0.1:5000/
- Server: Werkzeug/1.0.1 Python/3.8.5
- Set-Cookie: session=; Expires=Thu, 01-Jan-1970 00:00:00 GMT; Max-Age=0; Path=/

Request Headers (525 B)

- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
- Accept-Encoding: gzip, deflate
- Accept-Language: en-US,en;q=0.5

On Refresh -Firefox browser:

New session id gets created: bdc7a90a-1244-4426-a73c-ca189db4940d

Counter example - Mozilla Firefox

Counter example x Project3 - Google Drive x Untitled document - Google Docs x +

127.0.0.1:5000

Count: 2

Reset

Inspector Console Network Debugger Style Editor Performance Memory Storage Accessibility Application

Filter URLs

Status	Method	Domain	File	Initiator	Type	Transferred	Size	Headers	Cookies	Request	Response	Timings
200	GET	127.0.0.1...	/	BrowserTab...	html	492 B	262 B	Transferred	492 B (262 B size)			
404	GET	127.0.0.1...	favicon.ico	FaviconLoad...	html	cached	51 B					

GET http://127.0.0.1:5000/

Status: 200 OK

Version: HTTP/1.0

Transferred: 492 B (262 B size)

Referrer Policy: no-referrer-when-downgrade

Response Headers (230 B)

- Content-Length: 262
- Content-Type: text/html; charset=utf-8
- Date: Thu, 22 Oct 2020 07:13:12 GMT
- Server: Werkzeug/1.0.1 Python/3.8.5
- Set-Cookie: session=bdc7a90a-1244-4426-a73c-ca189db4940d; HttpOnly; Path=/

On Reset -Chrome browser: session id of SetCookie get cleared

Counter example - Chromium

Counter example x +

← → ↻ ⓘ 127.0.0.1:5000 ☆ ⚙

Count: 1

Reset

DevTools - 127.0.0.1:5000/

Elements Sources Console Network Performance Application Memory Security Lighthouse

Filter ☒ Hide data URLs ☐ Preserve log ☐ Disable cache Online ☐ Group by frame ☐ Has blocked cookies ☐ Blocked Requests

☒ Use large request rows ☐ Show overview ☐ Capture screenshots

Name	Headers	Preview	Response	Initiator	Timing	Cookies
reset	Referrer Policy: no-referrer-when-downgrade					
127.0.0.1	<div>Response Headers</div> <div>Content-Length: 209</div> <div>Content-Type: text/html; charset=utf-8</div> <div>Date: Thu, 22 Oct 2020 07:11:57 GMT</div> <div>Location: http://127.0.0.1:5000/</div> <div>Server: Werkzeug/1.0.1 Python/3.8.5</div> <div>Set-Cookie: session=; Expires=Thu, 01-Jan-1970 00:00:00 GMT; Max-Age=0; Path=/</div> <div>Request Headers</div> <div>Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,a</div>					

2 requests 763 B transferred

On Refresh -Chrome browser: New session id gets created: e69abe26-a71c-48fa-b16b-18f2388ca111

Counter example - Chromium

Counter example x +

← → ↻ ⓘ 127.0.0.1:5000 ☆ ⚙

Count: 2

Reset

DevTools - 127.0.0.1:5000/

Elements Sources Console Network Performance Application Memory Security Lighthouse

Filter ☒ Hide data URLs ☐ Preserve log ☐ Disable cache Online ☐ Group by frame ☐ Has blocked cookies ☐ Blocked Requests

☒ Use large request rows ☐ Show overview ☐ Capture screenshots

Name	Headers	Preview	Response	Initiator	Timing	Cookies
127.0.0.1	<div>Request Method: GET</div> <div>Status Code: 200 OK</div> <div>Remote Address: 127.0.0.1:5000</div> <div>Referrer Policy: no-referrer-when-downgrade</div> <div>Response Headers</div> <div>Content-Length: 262</div> <div>Content-Type: text/html; charset=utf-8</div> <div>Date: Thu, 22 Oct 2020 07:12:32 GMT</div> <div>Server: Werkzeug/1.0.1 Python/3.8.5</div> <div>Set-Cookie: session=e69abe26-a71c-48fa-b16b-18f2388ca111; HttpOnly; Path=/</div> <div>Request Headers</div> <div>view source</div>					

1 requests 492 B transferred

STEP 11:

Now lets get the active sessions from dump.py by executing the GET method of kv.py.
Below is the list of two active sessions in Firefox and Chrome browsers:

```
student@tufflx-vm:~/Desktop/Project3/cpsc449-master/counter$ ./dump.py
{
  "1676a34d-089c-406c-8048-181067e4afd8": {
    "count": 3
  }
}

{
  "653255d4-40d5-4ca4-a093-25662ea2dc75": {
    "count": 1
  }
}
```