

Name: Harshada Prabhakar Bhangale  
CWID: 887414969  
Email id: harshada.bhangale@csu.fullerton.edu  
Project Name:: Project 3- Server-Side Sessions

## PROJECT IMPLEMENTATION

### Cookie-based sessions

#### POINT 1

When `USE_SESSION_STORE` in `counter.cfg` is set to `False`, Flask uses the default [session](#) implementation, storing the session variable count in a signed cookie on the client side.

#### POINT 2

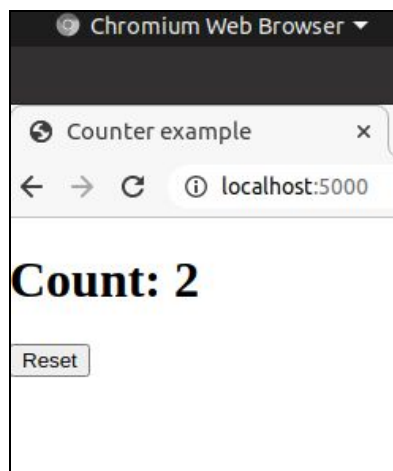
When the app is started and tries to load the page in two different browsers (e.g. Firefox and Chrome). Refreshing the page a few times in each browser, noticed that they maintain separate counts. Screenshots of a few different count values.

1. Run following commands in command prompt (cmd) when we are in counter project:

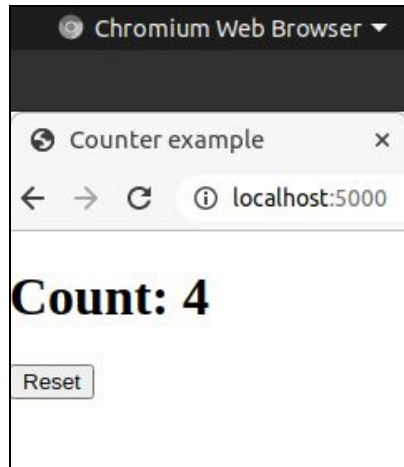
```
$ foreman start
```

Screenshots of Google chrome and Mozilla Firefox maintaining different counts.

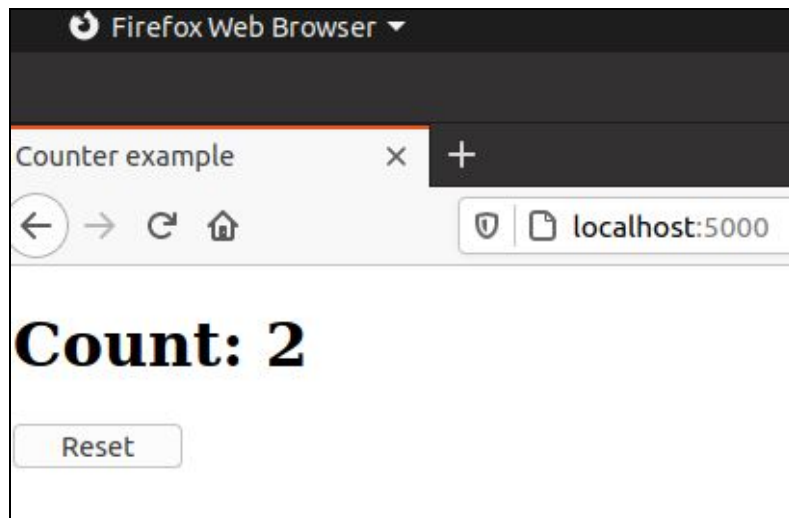
1. Chrome screenshot 1:



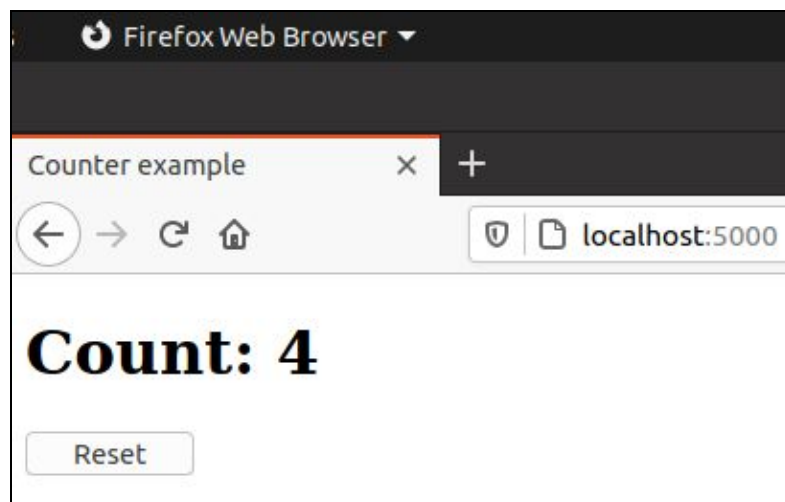
2. Chrome screenshot 2:



3. FireFox screenshot 1:



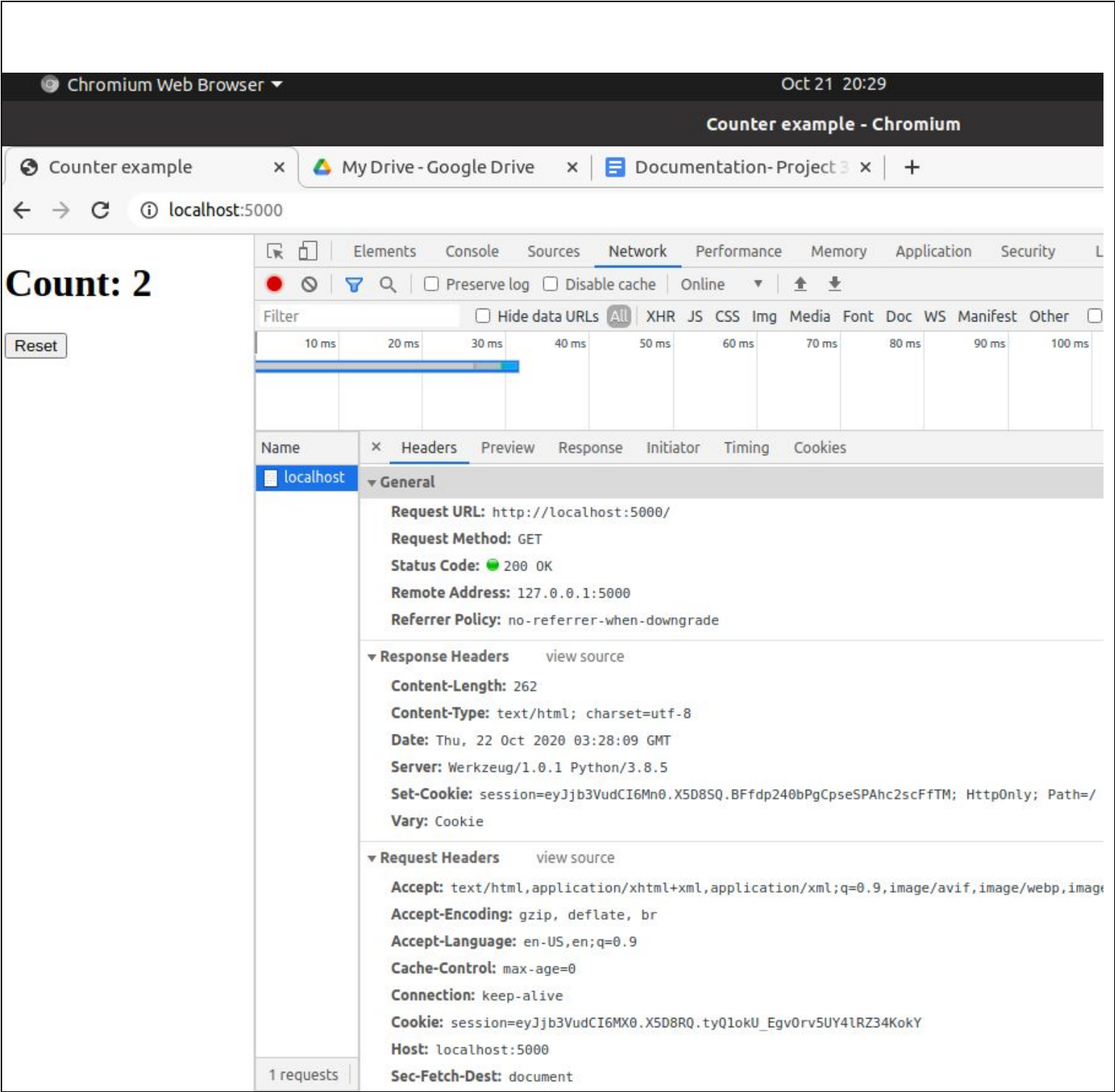
4. FireFox screenshot 2



POINT 3

When browser's [developer tools](#) to examine the Set-Cookie: response header, and noticed that it changes each time the page is refreshed. Below are a few screenshots showing the change.

1. Chrome screenshot 1:



## 2. Chrome screenshot 2 :

The screenshot shows a Chromium Web Browser window titled "Counter example - Chromium". The address bar displays "localhost:5000". The page content shows "Count: 4" and a "Reset" button. The Network tab is open, showing a single request to "localhost".

**Count: 4**

**Reset**

**Network Tab:**

- Name:** localhost
- General:**
  - Request URL: http://localhost:5000/
  - Request Method: GET
  - Status Code: 200 OK
  - Remote Address: 127.0.0.1:5000
  - Referrer Policy: no-referrer-when-downgrade
- Response Headers:**
  - Content-Length: 262
  - Content-Type: text/html; charset=utf-8
  - Date: Thu, 22 Oct 2020 03:30:16 GMT
  - Server: Werkzeug/1.0.1 Python/3.8.5
  - Set-Cookie: session=eyJjb3VudCI6NH0.X5D8yA.1dSVAo7UqhUK3xIoJSKUYrD00CU; HttpOnly; Path=/
  - Vary: Cookie
- Request Headers:**
  - Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/
  - Accept-Encoding: gzip, deflate, br
  - Accept-Language: en-US,en;q=0.9
  - Cache-Control: max-age=0
  - Connection: keep-alive
  - Cookie: session=eyJjb3VudCI6M30.X5D8xg.kmkWJp8ALuVsEtogZVkzEeClAdw
  - Host: localhost:5000
  - Sec-Fetch-Dest: document

1 requests

## Key-value store service

The file `kv.py` implements a simple [key-value store](#) service accessible via HTTP. The service exposes the following 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

### POINT 4

Experimenting with these methods in `kv.py` for the `set_key()`, `get_key()`, `delete_key()`, and `match()` methods include example API calls using HTTPie.

Run below commands:

```
$ FLASK_APP=kv flask run
$ export DB_URL=localhost:5000
```

1. Experiment with POST method for `set_key()`

```
$ http localhost:5000 foo=bar
```

```
student@tuffix-vm:~/Desktop/cpsc449-master/counter$ http localhost:5000 foo=bar
HTTP/1.0 200 OK
Content-Length: 19
Content-Type: application/json
Date: Thu, 22 Oct 2020 03:35:10 GMT
Server: Werkzeug/1.0.1 Python/3.8.5

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

- Experiment with GET method for get\_key()

```
$ http localhost:5000/foo
```

```
student@tuffix-vm:~/Desktop/cpsc449-master/counter$ http localhost:5000/foo
HTTP/1.0 200 OK
Content-Length: 19
Content-Type: application/json
Date: Thu, 22 Oct 2020 03:37:07 GMT
Server: Werkzeug/1.0.1 Python/3.8.5

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

- Experiment with GET method matching prefix for match()

```
$ http localhost:5000?prefix=f
```

```
student@tuffix-vm:~/Desktop/cpsc449-master/counter$ http localhost:5000?prefix=f
HTTP/1.0 200 OK
Content-Length: 30
Content-Type: application/json
Date: Thu, 22 Oct 2020 03:38:56 GMT
Server: Werkzeug/1.0.1 Python/3.8.5

{
  "keys": [
    "foo"
  ]
}
```

- Experiment with DELETE method for delete\_key()

```
$ http DELETE localhost:5000/foo
```

```
student@tuffix-vm:~/Desktop/cpsc449-master/counter$ http DELETE localhost:5000/foo
HTTP/1.0 200 OK
Content-Length: 19
Content-Type: application/json
Date: Thu, 22 Oct 2020 03:40:03 GMT
Server: Werkzeug/1.0.1 Python/3.8.5

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



## POINT 5

Using Requests to write a Python script `dump.py` to connect to the `kv.py` service and display all of its keys and values.

```
$ http localhost:5000 twitter=https://twitter.com/ProfAvery
$ http localhost:5000 github=https://github.com/ProfAvery
$ http localhost:5000 cpsec449=https://sites.google.com/view/cpsec449
```

Screenshot for the command run below:

```
ron@ron-Lenovo-Yoga-2-Pro:~/Documents/Project 3/counter$ http localhost:5000 twitter=https://twitter.com/ProfAvery
HTTP/1.0 200 OK
Content-Length: 49
Content-Type: application/json
Date: Wed, 09 Dec 2020 04:32:53 GMT
Server: Werkzeug/0.16.1 Python/3.8.5

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

ron@ron-Lenovo-Yoga-2-Pro:~/Documents/Project 3/counter$ http localhost:5000 github=https://github.com/ProfAvery
HTTP/1.0 200 OK
Content-Length: 47
Content-Type: application/json
Date: Wed, 09 Dec 2020 04:33:00 GMT
Server: Werkzeug/0.16.1 Python/3.8.5

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

ron@ron-Lenovo-Yoga-2-Pro:~/Documents/Project 3/counter$ http localhost:5000 cpsec449=https://sites.google.com/view/cpsec449
HTTP/1.0 200 OK
Content-Length: 57
Content-Type: application/json
Date: Wed, 09 Dec 2020 04:33:06 GMT
Server: Werkzeug/0.16.1 Python/3.8.5

{
  "cpsec449": "https://sites.google.com/view/cpsec449"
}
```

So when we run the below command:

```
$ python3 ./dump.py http://localhost:5000
```

```
ron@ron-Lenovo-Yoga-2-Pro:~/Documents/Project 3/counter$ python3 ./dump.py http://localhost:5000
{
  "cpssc449": "https://sites.google.com/view/cpsc449"
}

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

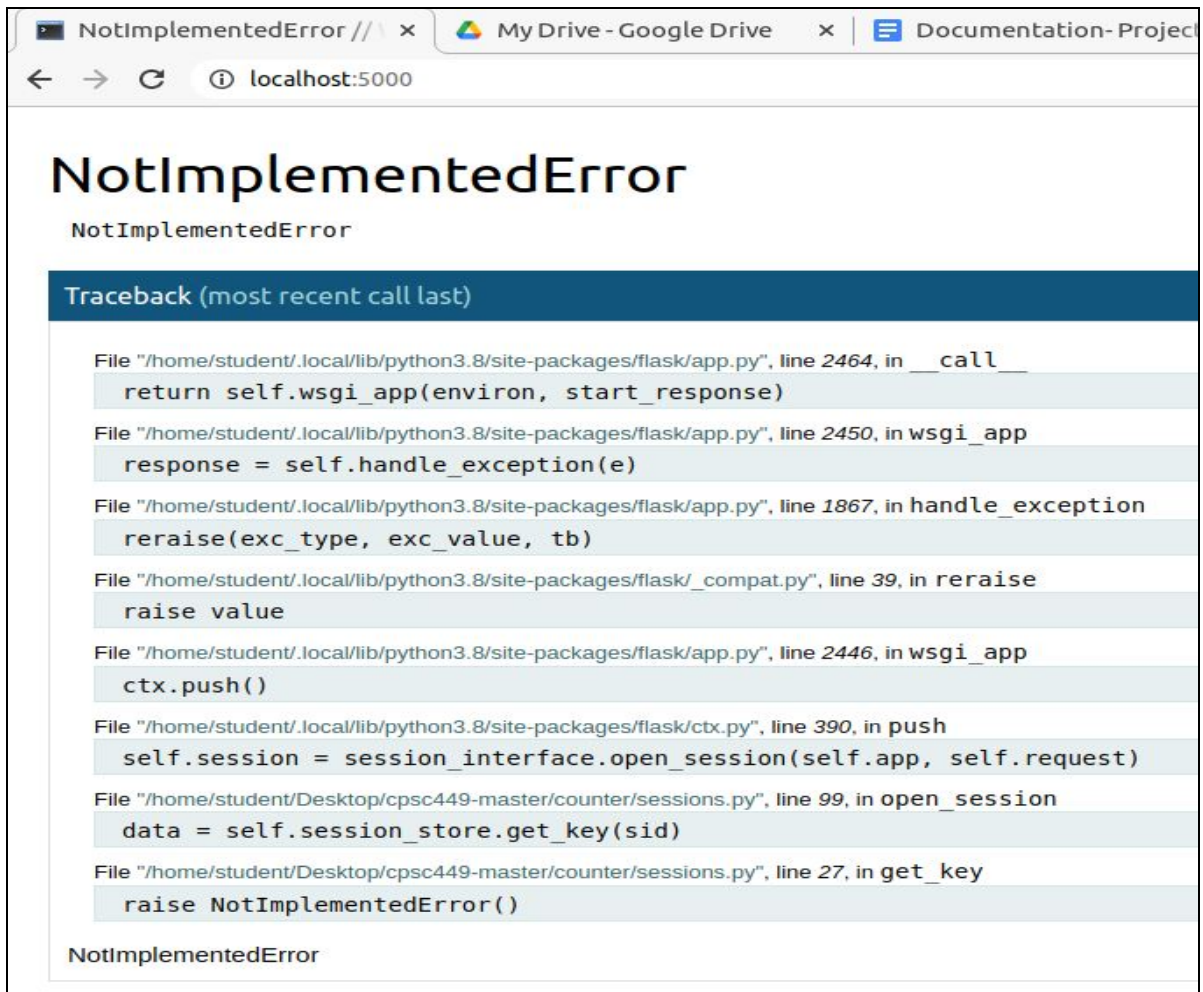
{
  "twitter": "https://twitter.com/ProfAvery"
}
```

## Server-side sessions

### POINT 6

Modify `counter.cfg` to set `USE_SESSION_STORE` to `True`, then start the counter app and the key-value store service with `foreman start`. Load the counter page, and notice that the methods of `KeyValueSessionStore` have not yet been implemented. Take a screenshot showing the failure.

- Screenshot showing failure(when `USE_SESSION_STORE= TRUE` and `KeyValueSessionStore` is not been implemented )





## POINT 7

Now override the `set_key()`, `get_key()`, and `delete_key()` methods in `KeyValueSessionStore` to use the Requests library to store keys and values in the `kv.py` service.

- Screenshot for methods in `KeyValueSessionStore` in `sessions.py`
- In `sessions.py` a variable is set to take url from `KV_URL` in `counter.cfg`
  - `kv_url = app.config['KV_URL']`

```
class KeyValueSessionStore(SessionStore):

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

# sets the key value
    def set_key(self, key, value):
        dict = {key: value}
        res = requests.post(kv_url, json=dict)
        return res.json()

# gets the key
    def get_key(self, key):
        res = requests.get(kv_url + "/" + key)
        return res.json()[key]

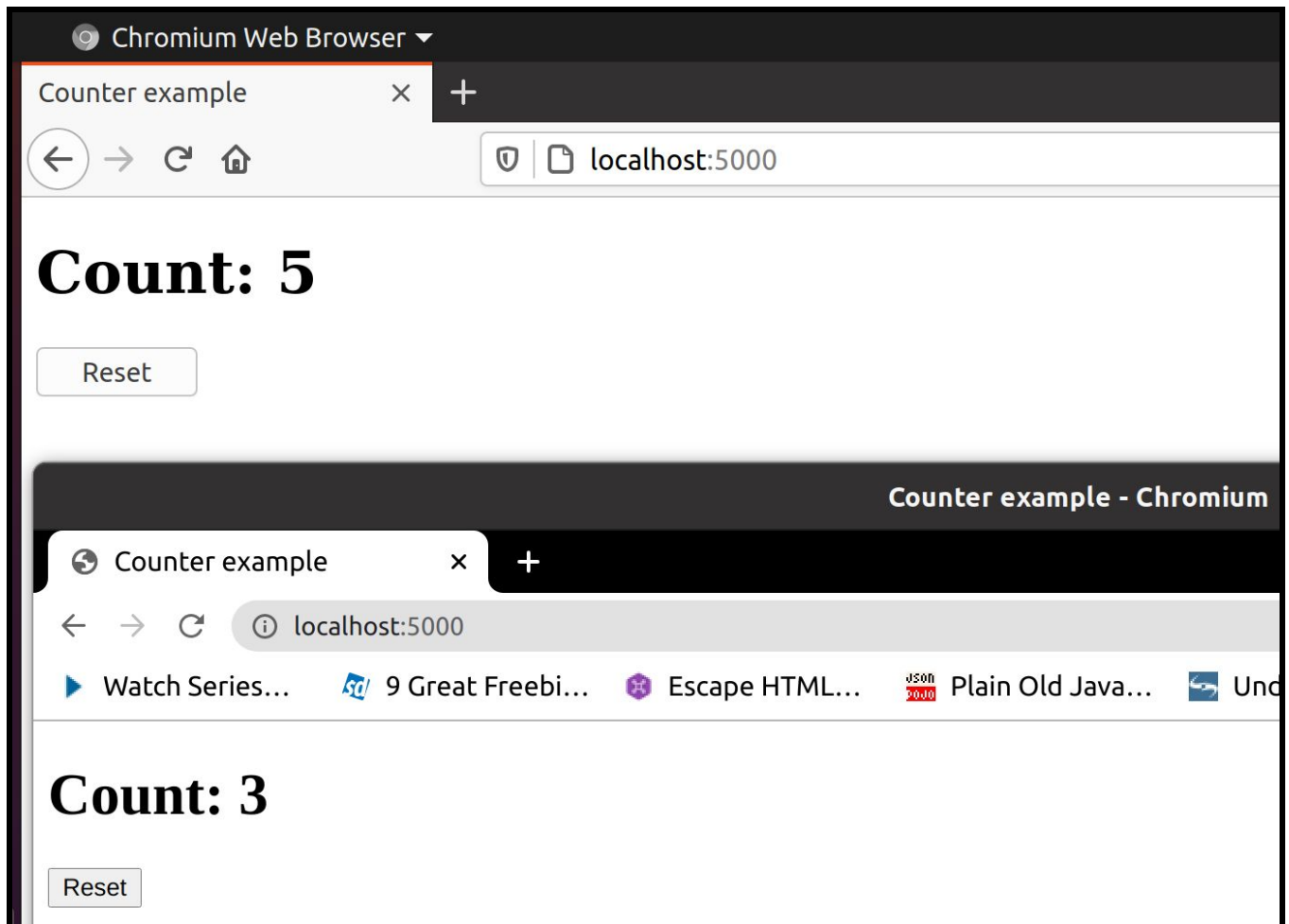
# deletes a key
    def delete_key(self, key):
        res = requests.delete(kv_url + "/" + key)
        return res.json()
```

## TESTING

### POINT 8

Now after implementation of `KeyValueSessionStore` we can repeat step (2) from *Cookie-based Flask sessions* and see the correct session behavior in both browsers.

Screenshot for both browser behaving correctly with different counter



## POINT 9

Now Repeat step (3) from *Cookie-based Flask sessions* and notice that the Set-Cookie: response header only contains a session id, and that the session id does not change when the page is refreshed and the counter is incremented. Take screenshots showing the same session key with different counter values.

- Screenshot (when USE\_SESSION\_STORE= TRUE and KeyValueTypeSessionStore is been implemented)
  - showing the same session key with different counter values.(when page is refreshed)

## On Firefox

When count=6 :

Set-Cookie: session=6cea3f31-60be-422c-9ae0-a281d9cdd0d6

The screenshot shows a web browser window with the address bar at `localhost:5000`. The page content displays **Count: 6** and a `Reset` button. The browser's developer tools are open, showing the **Network** tab. The list of requests includes a `GET` to `localhost:5000/` (status 200 OK) and a `GET` to `localhost:5000/favicon.ico` (status 500). The selected request's response headers are visible, showing `Set-Cookie: session=6cea3f31-60be-422c-9ae0-a281d9cdd0d6; HttpOnly; Path=/'`.

## On Firefox

When count=12 :

Set-Cookie: session=6cea3f31-60be-422c-9ae0-a281d9cdd0d6

Counter example x +

localhost:5000

# Count: 12

Reset

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application

Filter URLs

Sta...	Me...	Domain	File	Initiator	Type	Transferred	Si
200	GET	localho...	/	document	html	494 B	26
500	GET	localho...	favicon.ico	FaviconLoa...	html	cached	24

Headers Cookies Request Response Timings

Filter Headers

GET http://localhost:5000/

Status: 200 OK

Version: HTTP/1.0

Transferred: 494 B (263 B size)

Response Headers (231 B)

- Content-Length: 263
- Content-Type: text/html; charset=utf-8
- Date: Wed, 09 Dec 2020 06:12:08 GMT
- Server: Werkzeug/0.16.1 Python/3.8.5
- Set-Cookie: session=6cea3f31-60be-422c-9ae0-a281d9cdd0d6; HttpOnly; Path=

On Google Chrome When count=5 :

Set-Cookie: session=4c4a9080-3934-4360-b913-3919abae1e1

Counter example - Chromium

Counter example x +

localhost:5000

Watch Series... 9 Great Freebi... Escape HTML... Plain Old Java... Understanding... Online Daily L... IT Commu

# Count: 5

Reset

Elements Console Sources Network Performance Memory Application Security Lighthouse

Filter

Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other Has blocked c

Name	Headers	Preview	Response	Initiator	Timing	Cookies
localhost	<p>General</p> <p>Request URL: http://localhost:5000/</p> <p>Request Method: GET</p> <p>Status Code: 200 OK</p> <p>Remote Address: 127.0.0.1:5000</p> <p>Referrer Policy: no-referrer-when-downgrade</p> <p>Response Headers</p> <p>Content-Length: 262</p> <p>Content-Type: text/html; charset=utf-8</p> <p>Date: Wed, 09 Dec 2020 06:17:43 GMT</p> <p>Server: Werkzeug/0.16.1 Python/3.8.5</p> <p>Set-Cookie: session=4c4a9080-3934-4360-b913-3919abae1e1; HttpOnly; Path=</p> <p>X-DNS-Prefetch-Control: off</p>					

**On Google Chrome When count=11 :**

**Set-Cookie: session=4c4a9080-3934-4360-b913-3919abae1e1**

The screenshot shows a web browser window titled "Counter example - Chromium". The address bar shows "localhost:5000". The page content displays "Count: 11" and a "Reset" button. The browser's developer tools are open to the Network tab, showing a list of requests. The selected request is a GET request to "http://localhost:5000/". The response headers are visible, showing "Set-Cookie: session=4c4a9080-3934-4360-b913-3919abae1e1; HttpOnly; Path=/".

## POINT 10

After clicking the *Reset* button, the POST request clears the session cookie and new session id is generated once refreshed.

**On Firefox**

**When Reset button is clicked(session id gets cleared)**

**Set-Cookie: session=**



Firefox Web Browser Dec 8 10:32 PM

Counter example × +

localhost:5000

# Count: 1

Reset

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application

Filter URLs

Sta...	Me...	Domain	File	Initiator	Type	Transferred	Si
302	POST	localho...	reset	document	html	534 B	26
200	GET	localho...	/	document	html	493 B	26
500	GET	localho...	favicon.ico	FaviconLoa...	html	cached	24

Headers Cookies Request Response Timings

Filter Headers

POST http://localhost:5000/reset

Status: 302 FOUND  
Version: HTTP/1.0  
Transferred: 534 B (262 B size)  
Referrer Policy: no-referrer-when-downgrade

Response Headers (272 B)

- Content-Length: 209
- Content-Type: text/html; charset=utf-8
- Date: Wed, 09 Dec 2020 06:31:57 GMT
- Location: http://localhost:5000/
- Server: Werkzeug/0.16.1 Python/3.8.5
- Set-Cookie: session=; Expires=Thu, 01-Jan-1970 00:00:00 GMT; Max-Age=0; Path=

**On Firefox** New session id is created after refreshing the browser

**Set-Cookie: session= 9dff32c5-ff3f-4b28-85d4-3ec72c28f443**

Firefox Web Browser Dec 8 10:33 PM

Counter example × +

localhost:5000

# Count: 2

Reset

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application

Filter URLs

Sta...	Me...	Domain	File	Initiator	Type	Transferred	Si
200	GET	localho...	/	BrowserTa...	html	493 B	26
500	GET	localho...	favicon.ico	FaviconLoa...	html	cached	24

Headers Cookies Request Response Timings Stack Trace

Filter Headers

GET http://localhost:5000/

Status: 200 OK  
Version: HTTP/1.0  
Transferred: 493 B (262 B size)  
Referrer Policy: no-referrer-when-downgrade

Response Headers (231 B)

- Content-Length: 262
- Content-Type: text/html; charset=utf-8
- Date: Wed, 09 Dec 2020 06:32:40 GMT
- Server: Werkzeug/0.16.1 Python/3.8.5
- Set-Cookie: session=9dff32c5-ff3f-4b28-85d4-3ec72c28f443; HttpOnly; Path=

## On Google Chrome

When Reset button is clicked(session id gets cleared)

Set-Cookie: session=

Chromium Web Browser Dec 8 10:38 PM

Counter example - Chromium

Counter example x +

localhost:5000

Watch Series... 9 Great Freebi... Escape HTML... Plain Old Java... Understandin... Online Daily L... IT Com

Count: 1

Reset

Elements Console Sources Network Performance Memory Application Security Lighthouse

Filter Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other Has blocke

50 ms 100 ms 150 ms 200 ms 250 ms 300 ms 350 ms 400 ms 450 ms 500 ms

Name

- reset
- localhost
- data:image/png;base...
- data:image/svg+xml;...
- data:image/svg+xml;...
- data:image/svg+xml;...
- data:image/svg+xml;...
- extn-utils.html
- extn-utils.js

Headers Preview Response Initiator Timing Cookies

General

- Request URL: http://localhost:5000/reset
- Request Method: POST
- Status Code: 302 FOUND
- Remote Address: 127.0.0.1:5000
- Referrer Policy: no-referrer-when-downgrade

Response Headers view source

- Content-Length: 209
- Content-Type: text/html; charset=utf-8
- Date: Wed, 09 Dec 2020 06:37:34 GMT
- Location: http://localhost:5000/
- Server: Werkzeug/0.16.1 Python/3.8.5
- Set-Cookie: session=; Expires=Thu, 01-Jan-1970 00:00:00 GMT; Max-Age=0; Path=/
- X-DNS-Prefetch-Control: off

## On Google Chrome

New session id is created after refreshing the browser

Set-Cookie: session= 5c8e37df-0200-432b-b362-30c130994257

Chromium Web Browser Dec 8 10:39 PM

Counter example - Chromium

Counter example x +

localhost:5000

Watch Series... 9 Great Freebi... Escape HTML... Plain Old Java... Understandin... Online Daily L... IT

Count: 2

Reset

Elements Console Sources Network Performance Memory Application Security Light

Filter Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other Has

50 ms 100 ms 150 ms 200 ms 250 ms 300 ms 350 ms 400 ms 450 ms

Name x Headers Preview Response Initiator Timing Cookies

localhost

data:image/png;base...

data:image/svg+xml;...

data:image/svg+xml;...

data:image/svg+xml;...

data:image/svg+xml;...

extn-utils.html

extn-utils.js

General

Request URL: http://localhost:5000/

Request Method: GET

Status Code: 200 OK

Remote Address: 127.0.0.1:5000

Referrer Policy: no-referrer-when-downgrade

Response Headers view source

Content-Length: 262

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

Date: Wed, 09 Dec 2020 06:38:54 GMT

Server: Werkzeug/0.16.1 Python/3.8.5

Set-Cookie: session=5c8e37df-0200-432b-b362-30c130994257; HttpOnly; Path=

X-DNS-Prefetch-Control: off

## POINT 11

Use your dump.py script to list the active sessions in the key-value store. dump.py script to list the active sessions in the key-value store for 2 different browsers:

Run the below command:

```
$ FLASK_APP=kv flask run
```

```
$ python3 ./dump.py http://localhost:5000
```

```
ron@ron-Lenovo-Yoga-2-Pro:~/Documents/Project 3/counter$ python3 ./dump.py http://localhost:5000
{
  "9dff32c5-ff3f-4b28-85d4-3ec72c28f443": {
    "count": 4
  }
}
{
  "5c8e37df-0200-432b-b362-30c130994257": {
    "count": 8
  }
}
```