# Understanding Sessions in Django

- Ankit Pradhan

#1

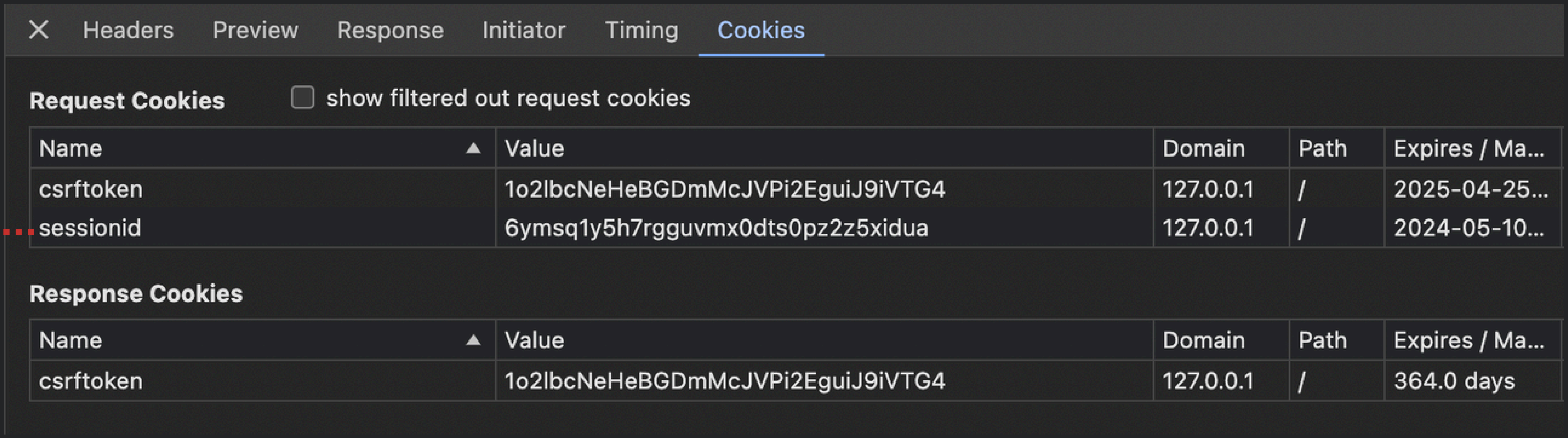
# Why sessions?

Sessions are a crucial part of web development, enabling websites to remember user data and preferences across different pages or visits

Sessions can be used to store temporary data that needs to be preserved between page requests, such as a shopping cart in an e-commerce website.

#2

# Inspecting Api



**Request Cookies** □ show filtered out request cookies

| Name ▲ | Value | Domain | Path | Expires / Ma... |
|---|---|---|---|---|
| csrftoken | 1o2lbcNeHeBGDmMcJVPi2EguiJ9iVTG4 | 127.0.0.1 | / | 2025-04-25... |
| sessionid | 6ymsq1y5h7rgguvmx0dts0pz2z5xidua | 127.0.0.1 | / | 2024-05-10... |

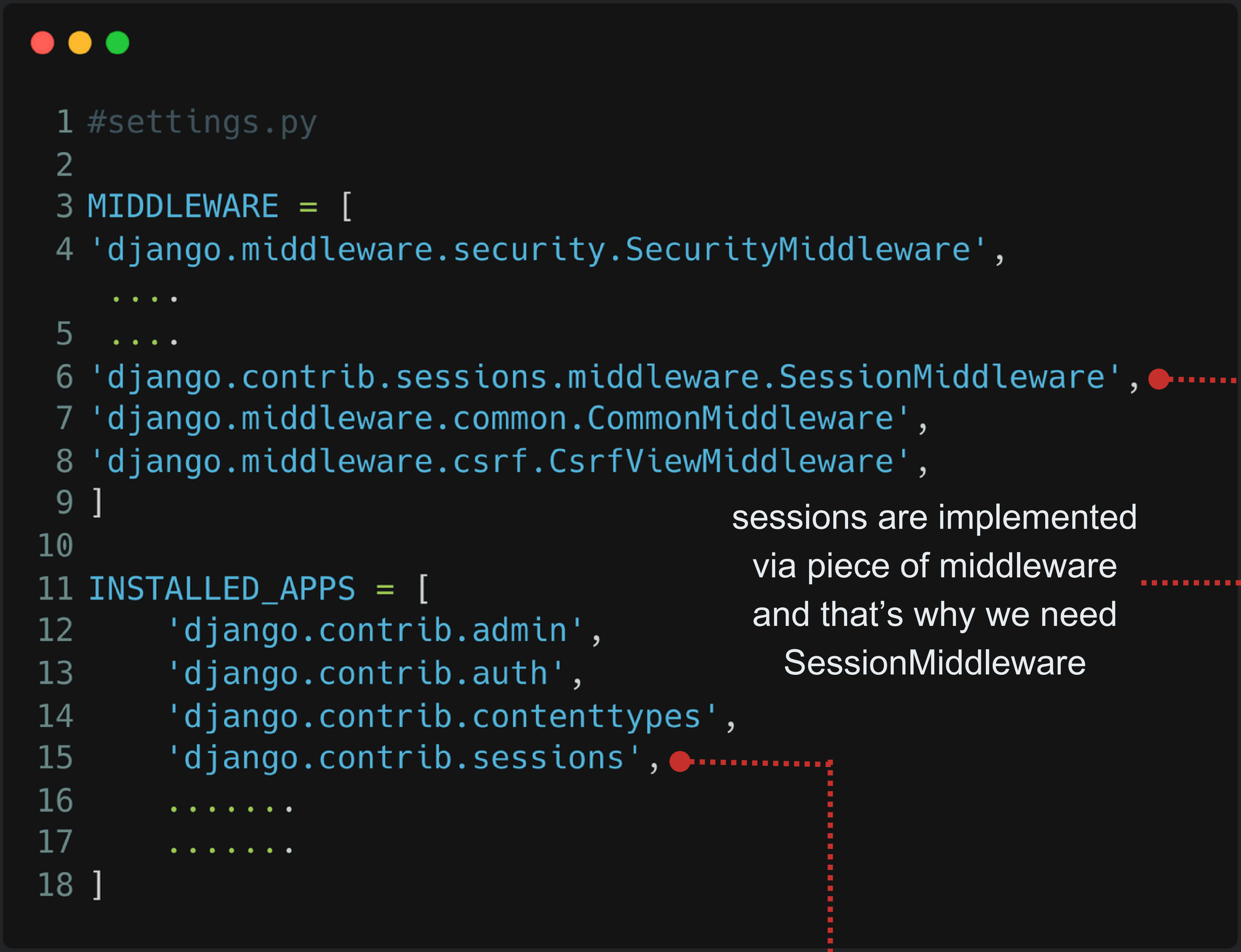
**Response Cookies**

| Name ▲ | Value | Domain | Path | Expires / Ma... |
|---|---|---|---|---|
| csrftoken | 1o2lbcNeHeBGDmMcJVPi2EguiJ9iVTG4 | 127.0.0.1 | / | 364.0 days |

The **session ID** is stored on the user's browser under cookies. This ID is used to retrieve session data stored on the server that can be passed to different pages of a website. e.g. user name, preferences.

#3

# Enabling sessions in django



```
#settings.py

MIDDLEWARE = [
'django.middleware.security.SecurityMiddleware',
  ....
  ....
'django.contrib.sessions.middleware.SessionMiddleware',
'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
]

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    .......
    .......
]
```

sessions are implemented via piece of middleware and that’s why we need SessionMiddleware

This will enable database-backed session

#4

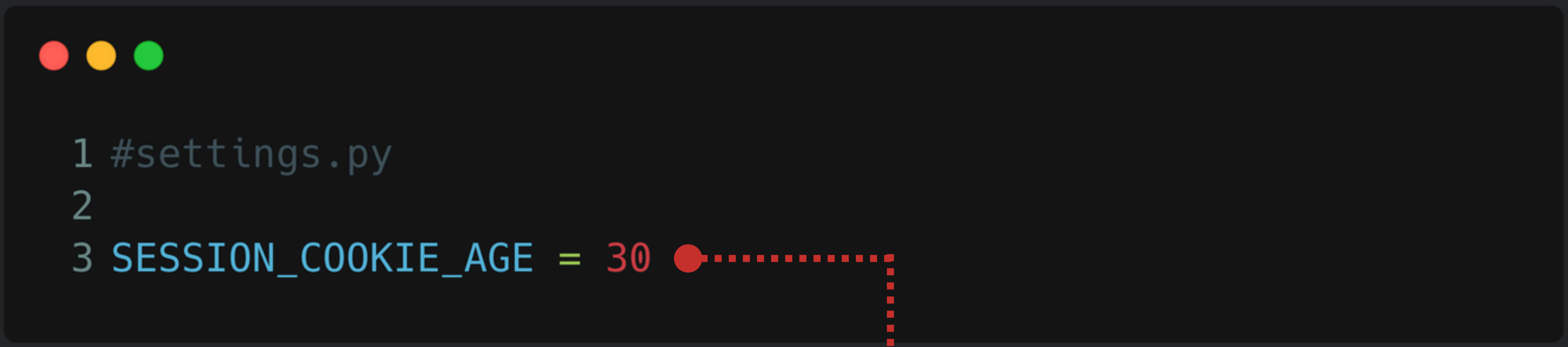
# Creating table for sessions

Once the configuration is done in settings.py file. Run
**manage.py migrate**

This will create a table named as **django_sessions** along with other tables

#5

# How to create session



```
#settings.py

SESSION_COOKIE_AGE = 30
```

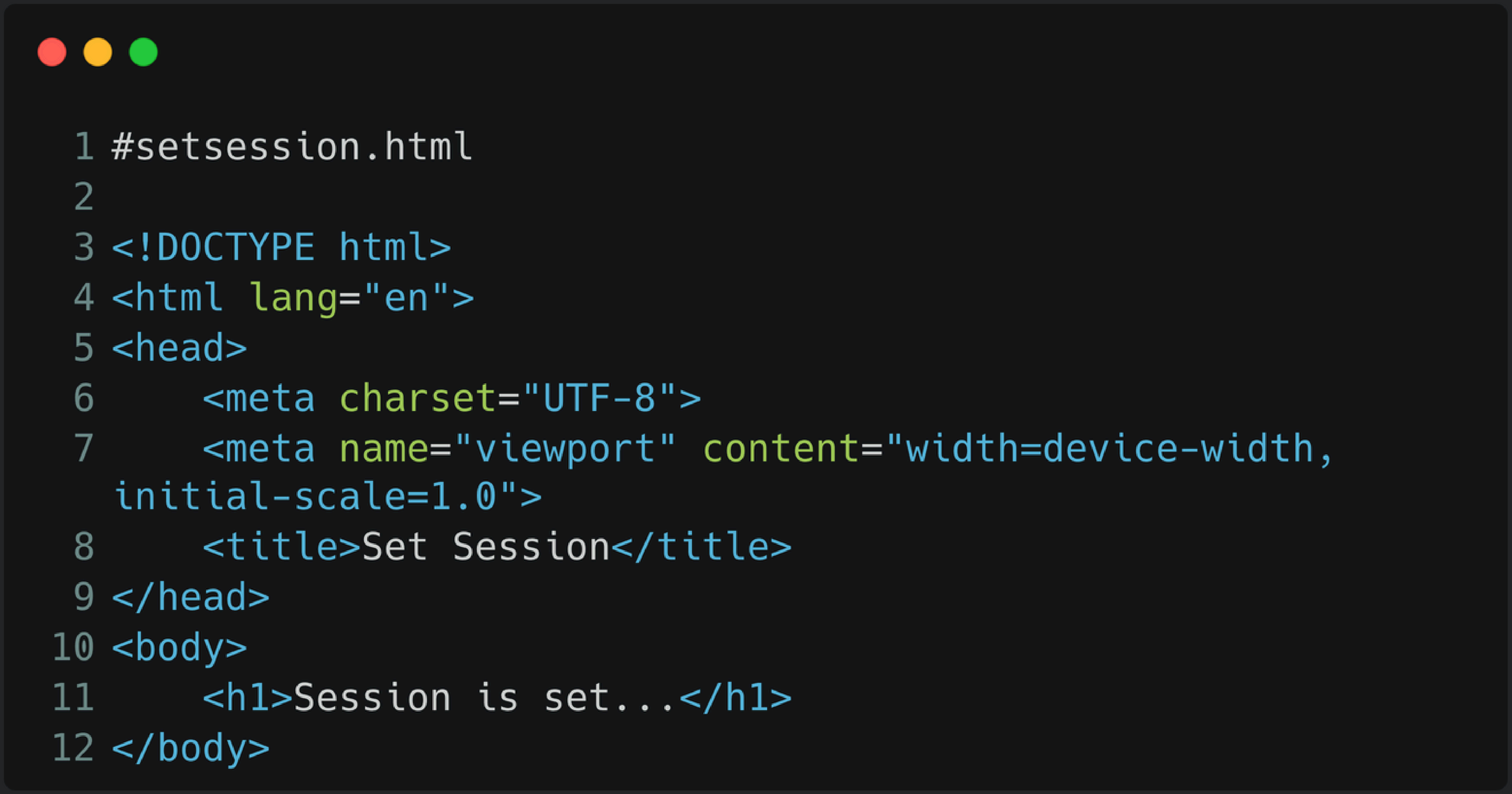
sets the cookie duration. In this case the session cookie will be deleted from the user’s browser after 30 seconds

We will create 2 simple pages under templates folder of our app

- setsession.html
- getsession.html

#6

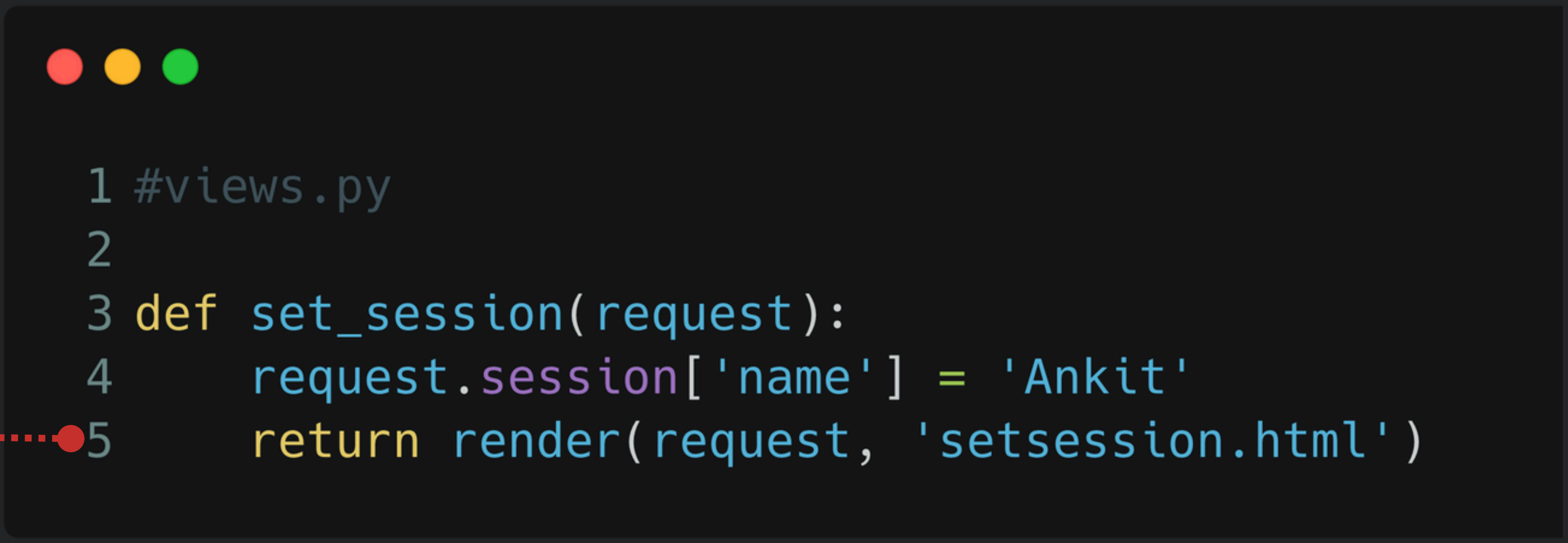
# Creating a simple webpage to set session



```
#setsession.html

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Set Session</title>
</head>
<body>
    <h1>Session is set...</h1>
</body>
```

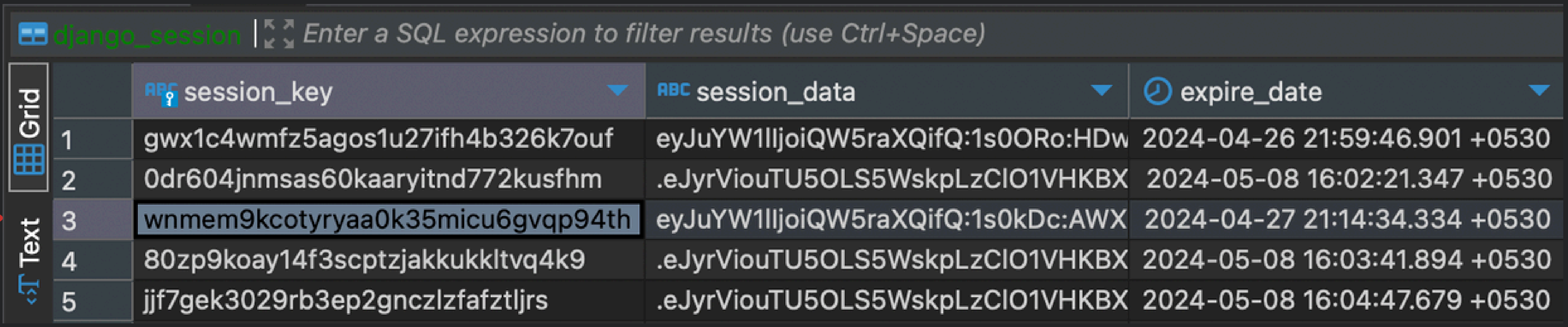
#7

# Creating function to set session in views



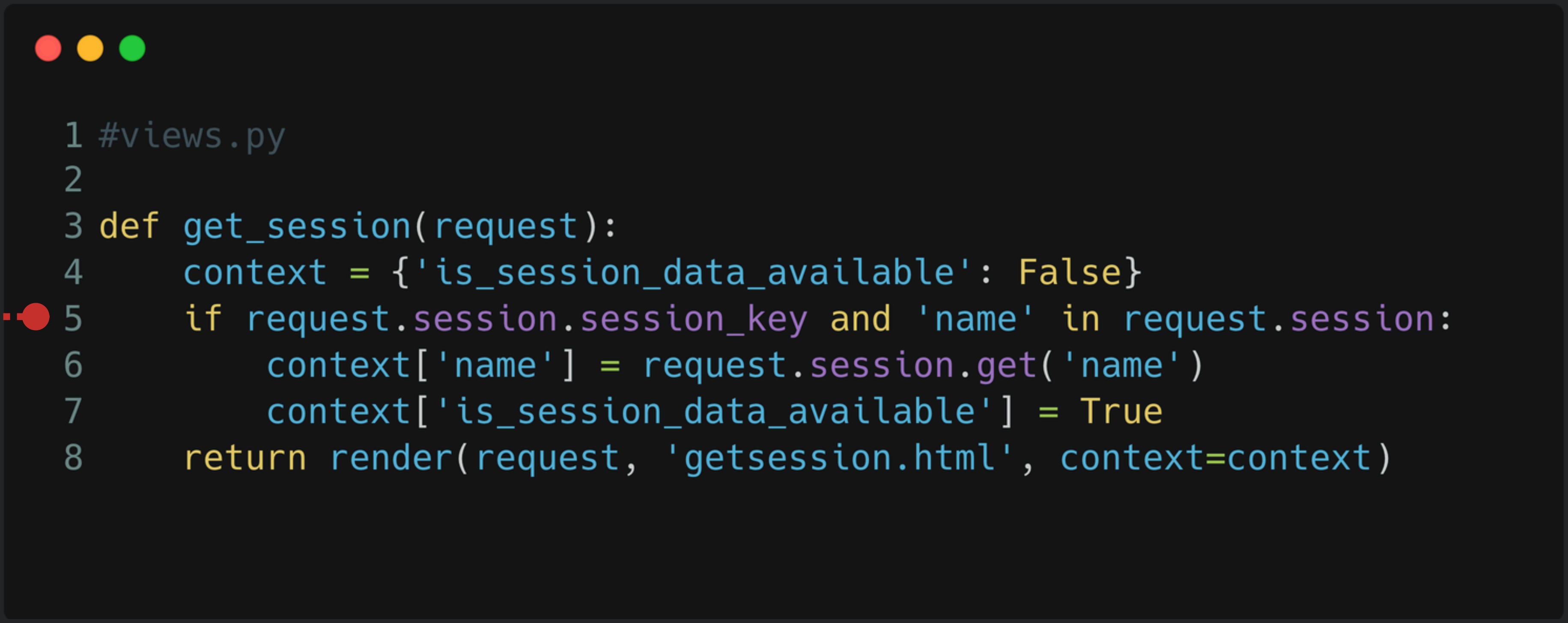
```python
#views.py

def set_session(request):
    request.session['name'] = 'Ankit'
    return render(request, 'setsession.html')
```

once this function completes its execution a session id is created in ***django_session*** table along with the session data i.e. name='Ankit' in a dictionary in base64 encoded. Also the expiry date is stored in the **expire_date** column



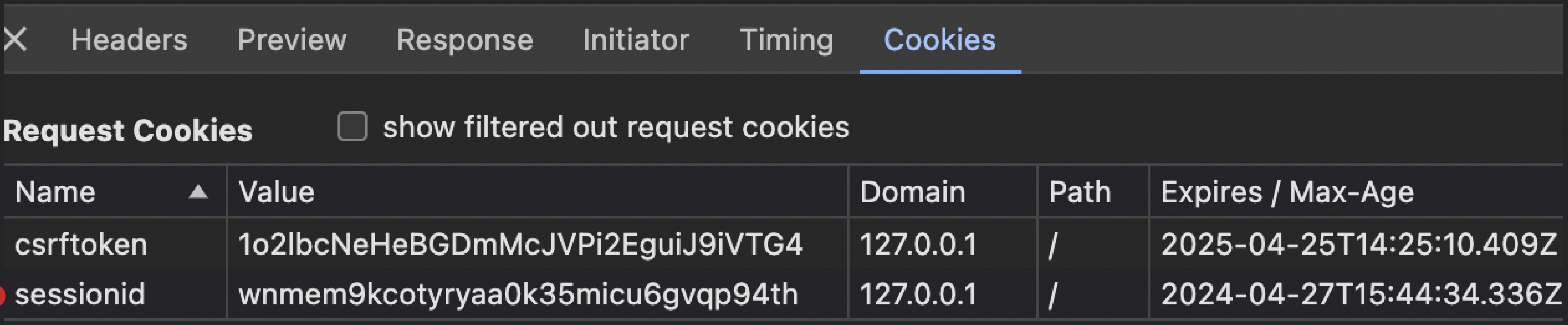
| | session_key | session_data | expire_date |
|---|---|---|---|
| 1 | gwx1c4wmfz5agos1u27ifh4b326k7ouf | eyJuYW1lljoiQW5raXQifQ:1s0ORo:HDw | 2024-04-26 21:59:46.901 +0530 |
| 2 | 0dr604jnmsas60kaaryitnd772kusfhm | .eJyrViouTU5OLS5WskpLzClO1VHKBX | 2024-05-08 16:02:21.347 +0530 |
| 3 | wnmem9kcotyryaa0k35micu6gvqp94th | eyJuYW1lljoiQW5raXQifQ:1s0kDc:AWX | 2024-04-27 21:14:34.334 +0530 |
| 4 | 80zp9koay14f3scptzjakkukkltvq4k9 | .eJyrViouTU5OLS5WskpLzClO1VHKBX | 2024-05-08 16:03:41.894 +0530 |
| 5 | jjf7gek3029rb3ep2gnczlzfafztljrs | .eJyrViouTU5OLS5WskpLzClO1VHKBX | 2024-05-08 16:04:47.679 +0530 |

# Creating function to get session data on our HTML page



```python
#views.py

def get_session(request):
    context = {'is_session_data_available': False}
    if request.session.session_key and 'name' in request.session:
        context['name'] = request.session.get('name')
        context['is_session_data_available'] = True
    return render(request, 'getsession.html', context=context)
```

we are getting sessionid in the request cookies which checks the table ***django_sessions*** for the corresponding session_key and session_data and returns the row.



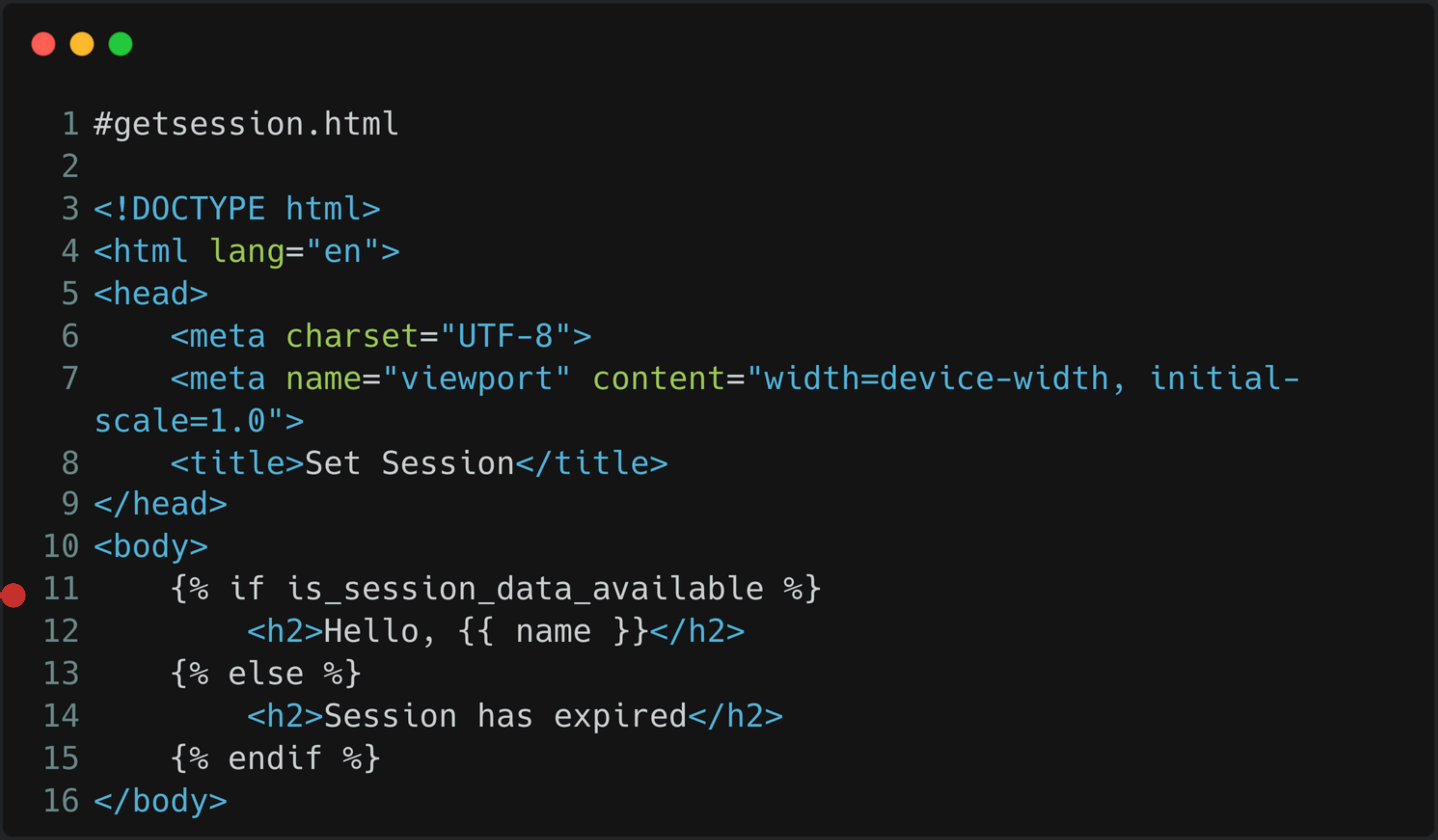
**Request Cookies** ☐ show filtered out request cookies

| Name ▲ | Value | Domain | Path | Expires / Max-Age |
|---|---|---|---|---|
| csrftoken | 1o2lbcNeHeBGDmMcJVPi2EguiJ9iVTG4 | 127.0.0.1 | / | 2025-04-25T14:25:10.409Z |
| sessionid | wnmem9kcotyryaa0k35micu6gvqp94th | 127.0.0.1 | / | 2024-04-27T15:44:34.336Z |

The session id is stored in the user’s browser under cookies

#9

# Creating a webpage to print session’s data

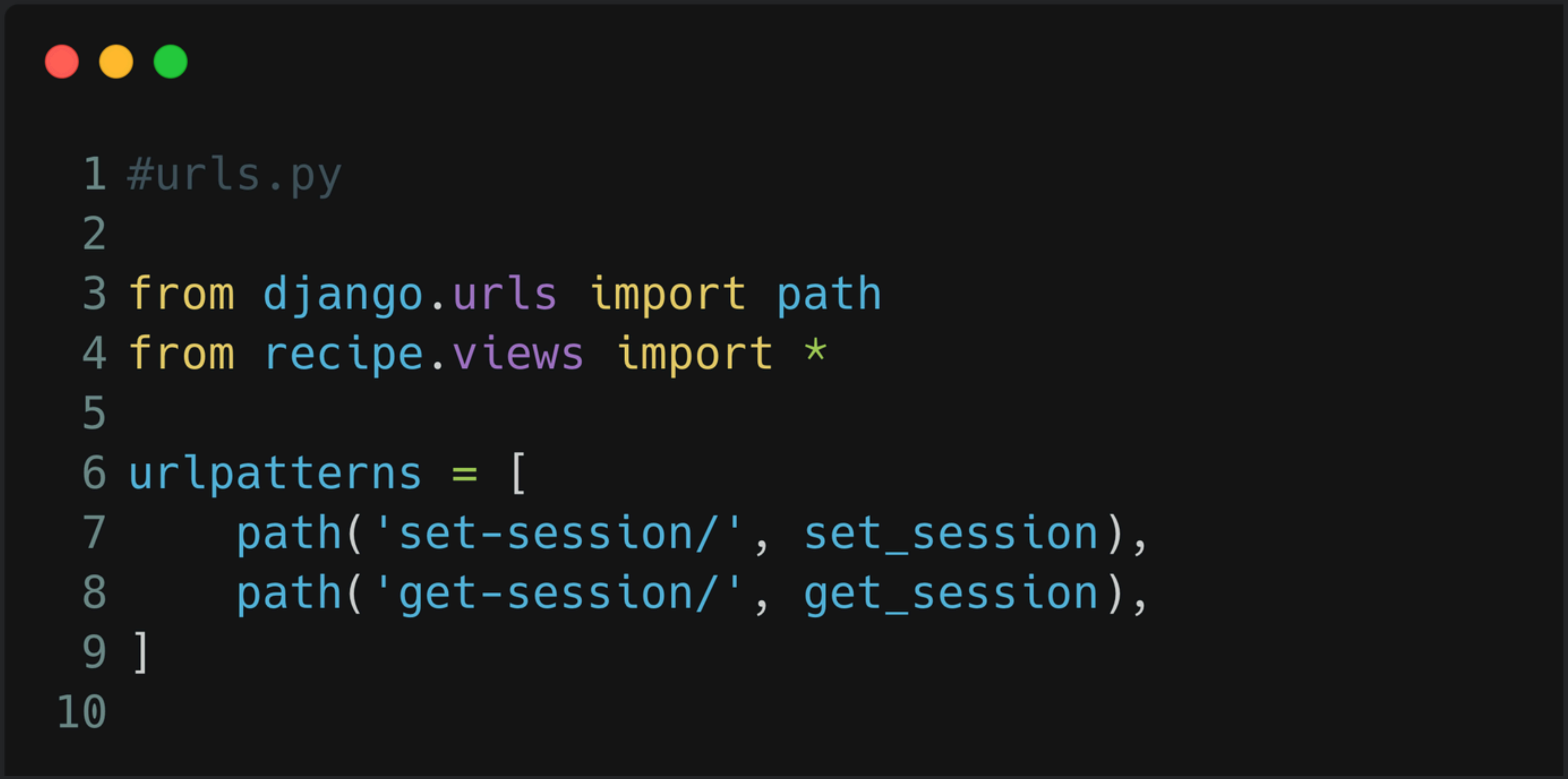


```
#getsession.html

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Set Session</title>
</head>
<body>
    {% if is_session_data_available %}
        <h2>Hello, {{ name }}</h2>
    {% else %}
        <h2>Session has expired</h2>
    {% endif %}
</body>
```

checks the key ***is_session_data_available*** that we have passed in the context from the get_session function in views.py file

#10

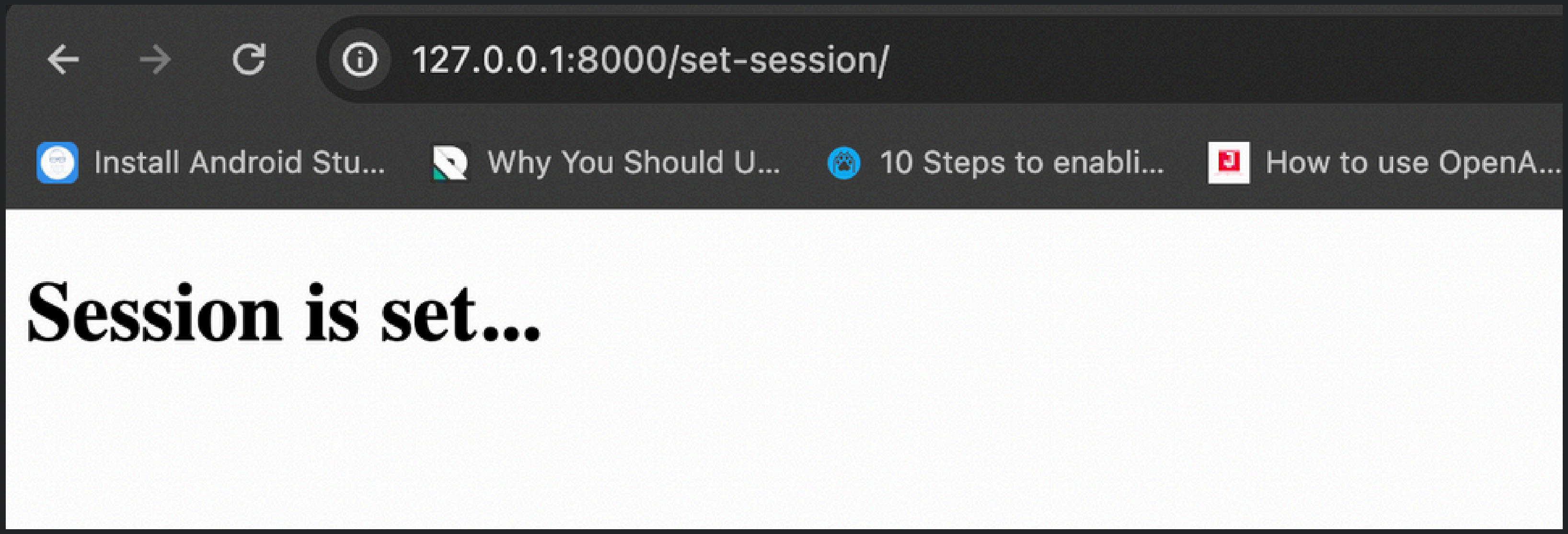
# Creating endpoints



```python
#urls.py

from django.urls import path
from recipe.views import *

urlpatterns = [
    path('set-session/', set_session),
    path('get-session/', get_session),
]

```

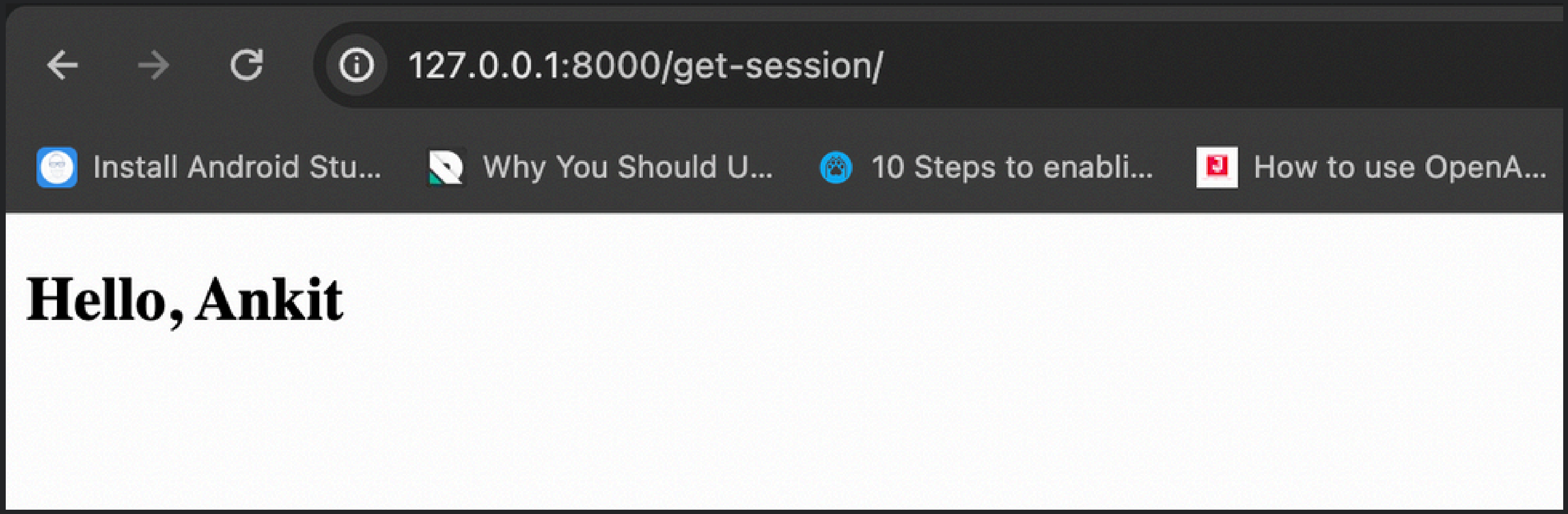
#11

# Show time



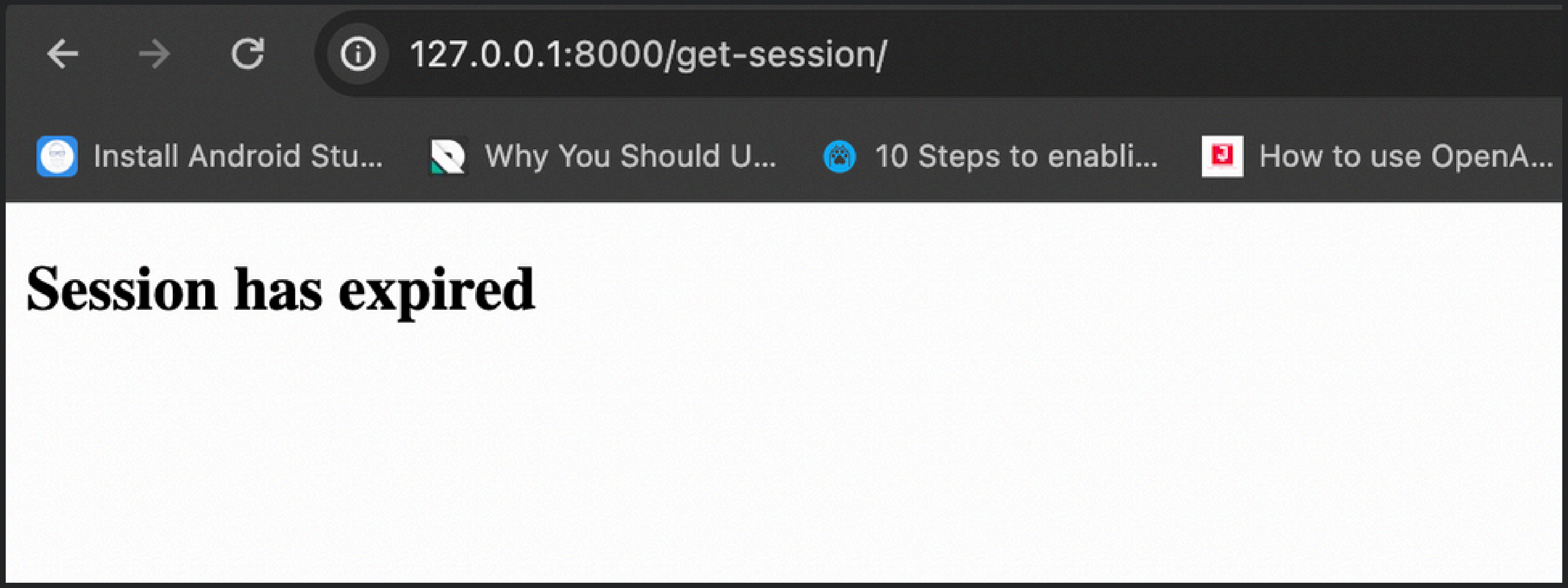
Session is set...

session data is set. In this case I am creating a key *'name'* with its value *'Ankit'*



Hello, Ankit

session cookies get expired after 30 seconds



Session has expired

#12

# That’s it...

Let’s learn python’s django framework together

Ankit Pradhan