INTRODUCTION TO SESSIONS IN DJANGO

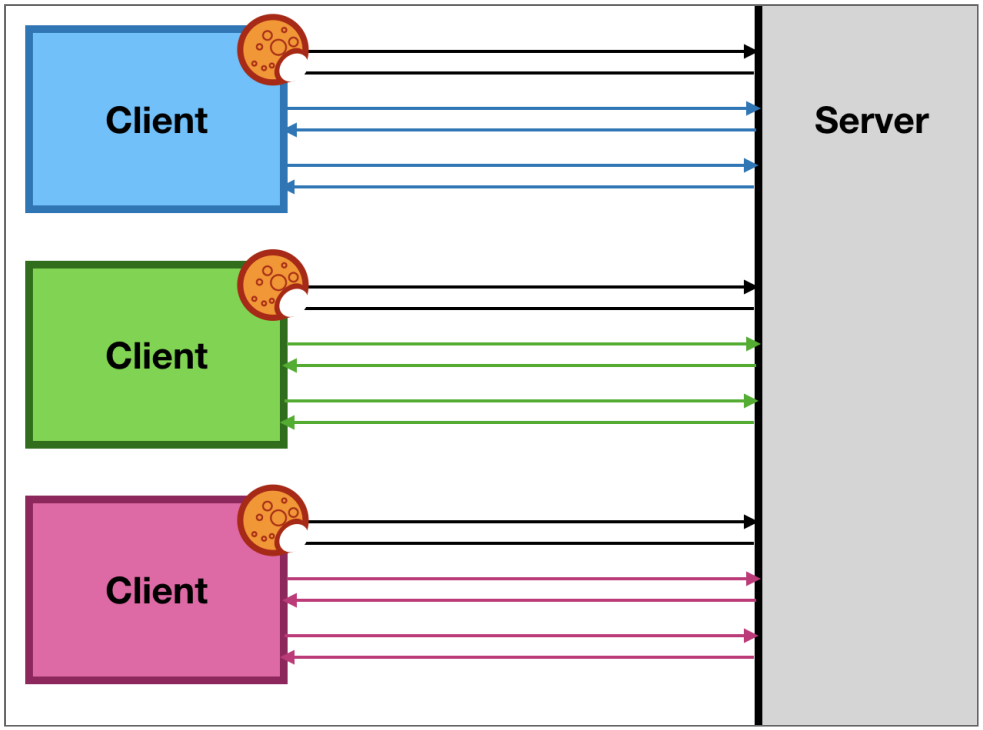
Magdalena Rother
rother.magdalena@gmail.com


camping.info
Karte
Favoriten
Deutsch
Entdecke 23054 Campingplätze
in Europa

OVERVIEW

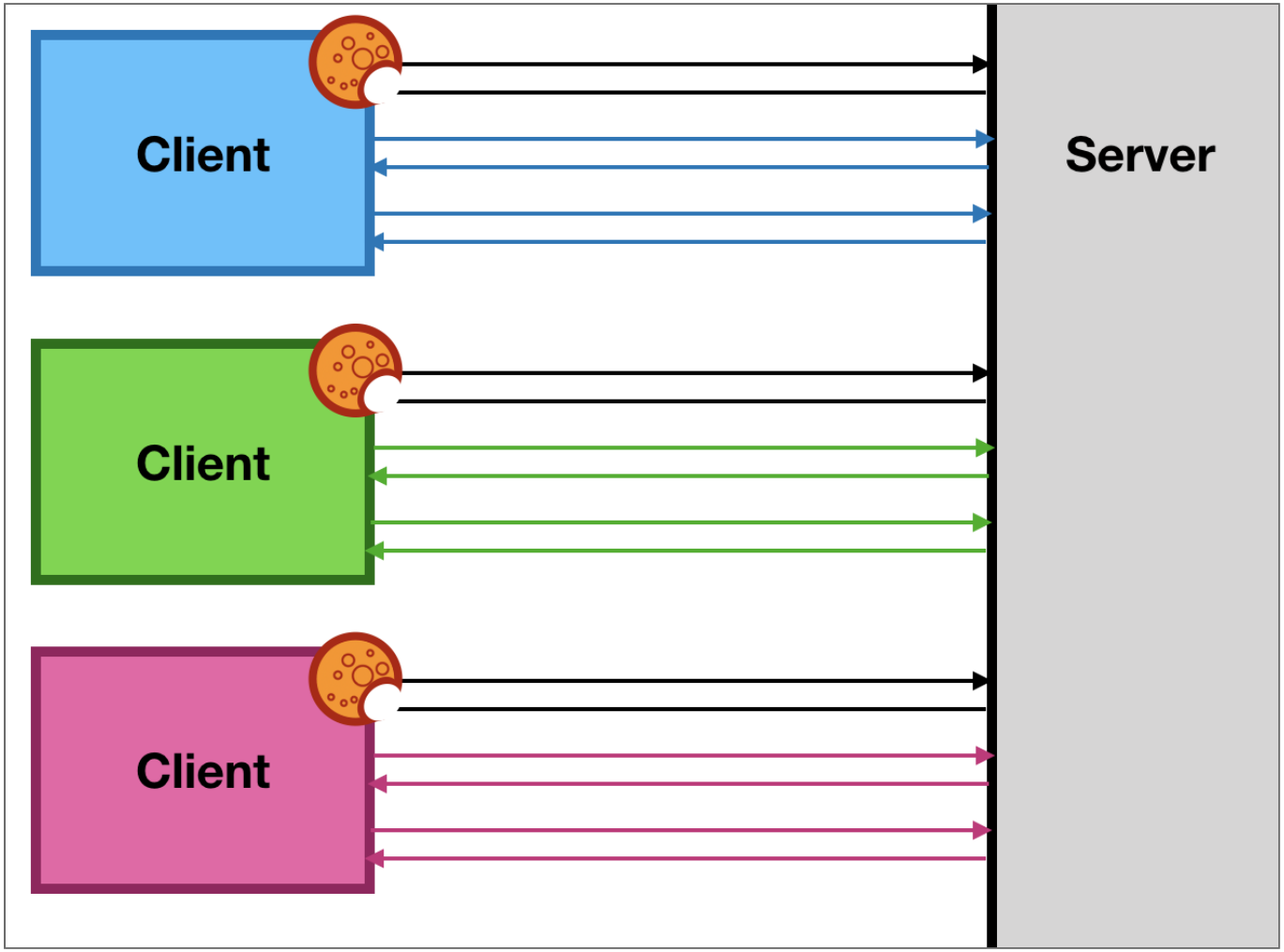
- What are sessions
- Step by step usage
- Examples

WHAT ARE SESSIONS?



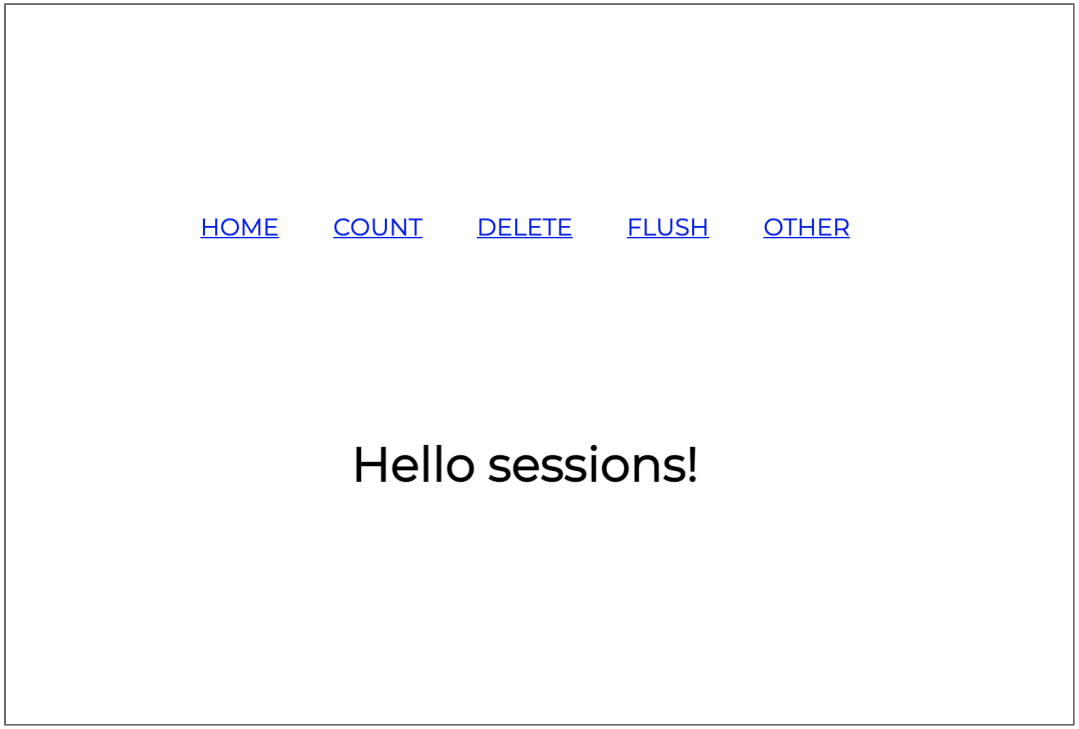
WHAT ARE SESSIONS?

Sessions allow to store arbitrary data per browser



SESSIONS STEP BY STEP

github.com/lenarother/sessions-example



GitHub: lenarother/sessions-example

```
sessions-example
├── README.rst
├── resources
│   └── requirements.txt
├── src
│   ├── manage.py
│   ├── sessionsexample
│   │   ├── __init__.py
│   │   ├── asgi.py
│   │   ├── counter
│   │   │   ├── __init__.py
│   │   │   ├── apps.py
│   │   │   ├── urls.py
│   │   │   └── views.py
│   │   ├── settings.py
│   │   ├── urls.py
│   │   └── wsgi.py
│   └── testing
└── templates
```

settings.py

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

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
```

views.py

```
class CountView(TemplateView):
    template_name = 'counter/count.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)

        counter = self.request.session.get('counter', 0)
        counter += 1
        self.request.session['counter'] = counter
        context['counter'] = counter

        return context
```

views.py

```
class DeleteSessionKeyView(TemplateView):
    template_name = 'counter/delete-key.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)

        counter = self.request.session.pop('counter', 0)
        context['counter'] = counter

        return context
```

views.py

```
class FlushSessionView(TemplateView):
    template_name = 'counter/flush-session.html'

    def get(self, request):
        self.request.session.flush()
        return super().get(request)
```

views.py

```
class OtherExamplesView(TemplateView):
    template_name = 'counter/other-examples.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)

        context['session_age'] =
self.request.session.get_session_cookie_age()
        context['session_expiry_date'] =
self.request.session.get_expiry_date()

        return context
```

Available settings

```
SESSION_CACHE_ALIAS
SESSION_COOKIE_AGE
SESSION_COOKIE_DOMAIN
SESSION_COOKIE_HTTPONLY
SESSION_COOKIE_NAME
SESSION_COOKIE_PATH
SESSION_COOKIE_SAMESITE
SESSION_COOKIE_SECURE
SESSION_ENGINE
SESSION_EXPIRE_AT_BROWSER_CLOSE
SESSION_FILE_PATH
SESSION_SAVE_EVERY_REQUEST
SESSION_SERIALIZER
```

test_views.py

```python
@pytest.mark.django_db
class TestCountView:

    def test_get(self, client):
        url = reverse('counter')

        response = client.get(url)
        assert response.status_code == 200
        assert response.context['counter'] == 1

        client.get(url)
        client.get(url)
        client.get(url)

        response = client.get(url)
        assert response.context['counter'] == 5

        client.cookies.pop('sessionid')
        response = client.get(url)
        assert response.context['counter'] == 1
```

MORE EXAMPLES

- Remember if user has voted
- Save url parameter
- Customising the user experience

Remembe last activity - Dominion Game Generator

```
class DrawFromSessionMixin:

    def set_last_draw_in_session(self, draw_data, game_id):
        self.request.session['dominion-last-draw'] = draw_data
        self.request.session['dominion-last-draw-id'] = game_id
        self.request.session['dominion-last-draw-dt'] =
str(timezone.now())

    def get_last_draw_from_session(self, game_id=None):
        last_draw = self.request.session.get('dominion-last-
draw', None)
        last_draw_dt = self.request.session.get('dominion-last-
draw-dt', None)
        last_draw_id = self.request.session.get('dominion-last-
draw-id', None)
        if not (last_draw and last_draw_dt and last_draw_id):
            return None
        if timezone.now() - parse(last_draw_dt) >
datetime.timedelta(hours=3):
            return None
        if game_id and game_id != last_draw_id:
            return None

        return last_draw
```

```
20
21      def get_game(self, game_data=None):
22          if game_data:
23              draw_service =
    get_draw_service_from_form_data(game_data)
24              game = draw_service.draw_game()
25              self.set_last_draw_in_session(game_data, game.pk)
26              return game
27          return None
```

SUMMARY

- Sessions allow to store arbitrary data per browser
- docs.djangoproject.com/en/3.1/topics/http/sessions/
- github.com/lenarother/sessions-example