Flask❤️4일

플라스크의 세션은 어디에 저장될까

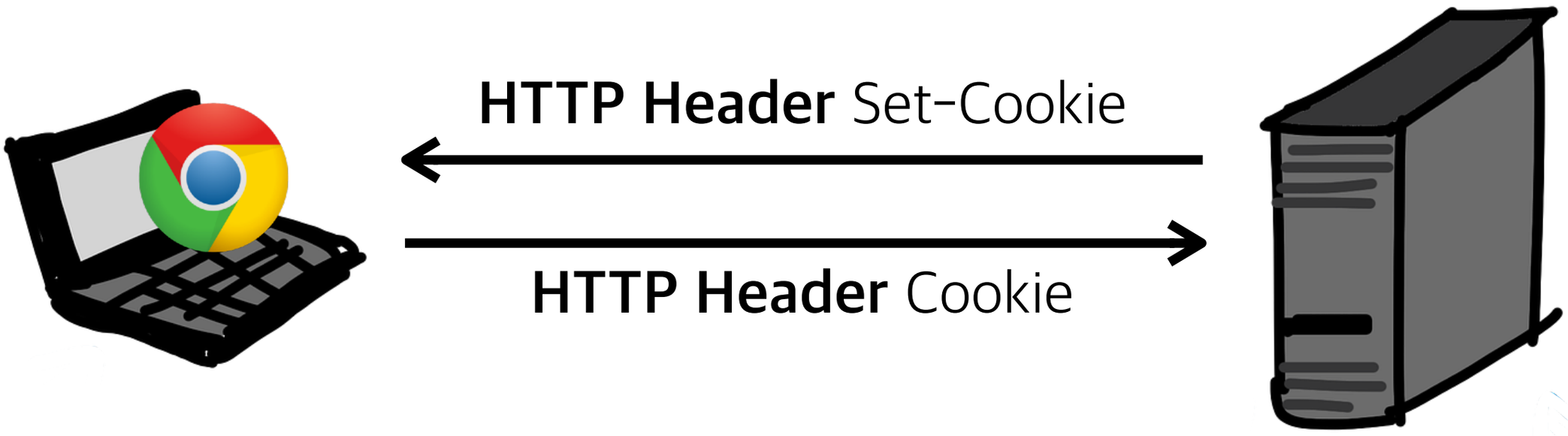
jangjunha<jangjunha113@gmail.com>

Session

로그인 기능을 추가해보자

Cookie

- 웹 사이트에서 **사용자 컴퓨터(브라우저)**에 저장하는 데이터



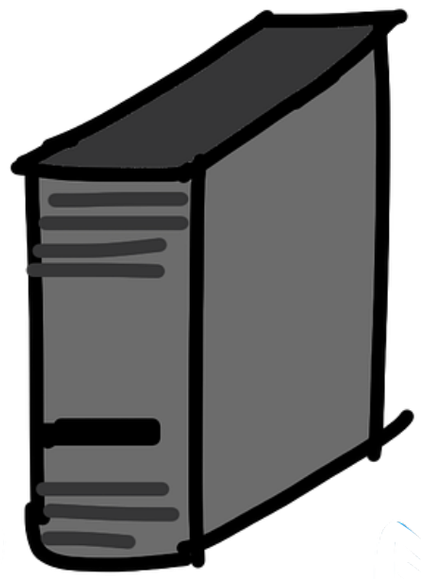
전통적 Session

Session ID	Data
fsdaf239das	10번 유저
d343r2rdq13	15번 유저
d13532reda21	22번 유저

Flask Session

양방향 암호화된 데이터

양방향 암호화된 데이터



dasf3rojiofdajfdklaf

SECRET_KEY

Cookie 읽기 / 쓰기

읽기

```
if request.cookies['popup_skip'] == 'yes':
        popup = None
```

쓰기

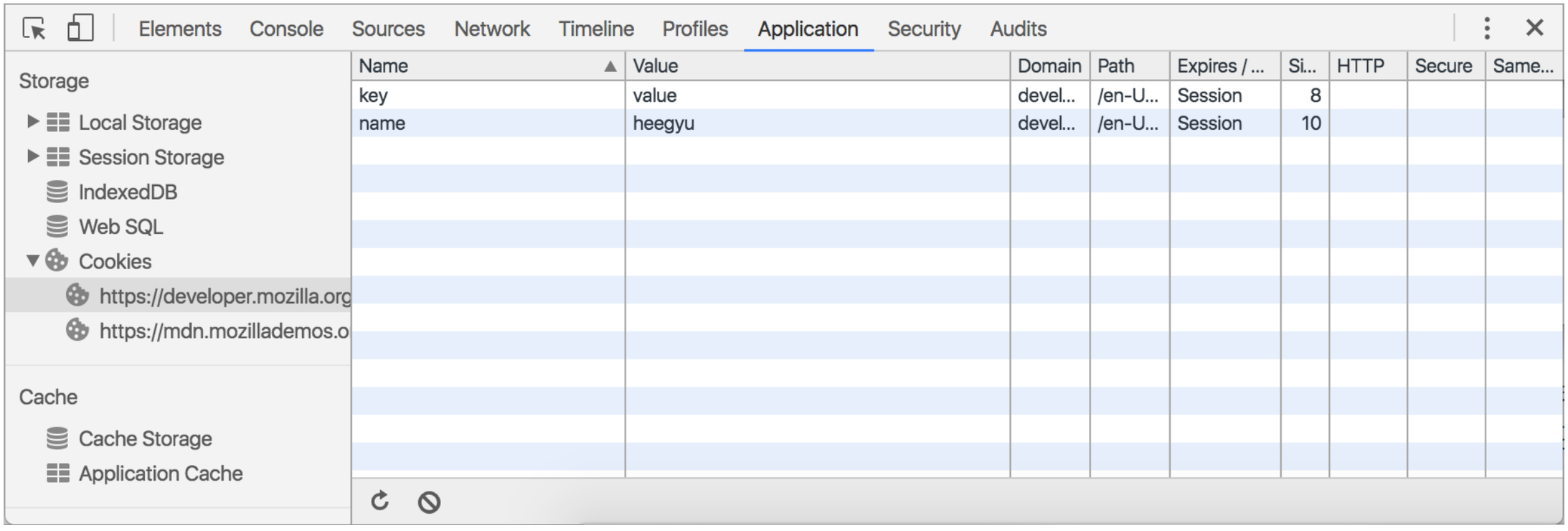
```
response = make_response(render_template('index.html'))
response.set_cookie('popup_skip', 'yes')
return response
```

브라우저에서 쿠키 접근

Javascript

```
document.cookie = "key=value"
document.cookie = "name=heegyu"
alert(document.cookie)
```

Chrome 개발자 도구



세션 읽기 / 쓰기

읽기

```
user_id = session['user_id']
```

쓰기

```
session['user_id'] = user.id
```

실습

로그인/로그아웃 구현

1. 로그인했다면 '로그아웃' 버튼을 보여주고,
   아니라면 '회원가입', '로그인' 버튼을 보여주기

2. 로그인했다면 메인화면에 "…님 환영합니다!" 보여주기

실습

- 회원가입하면 자동으로 로그인되도록 해보자

g

- 전역 객체
- 한 request 내에서만 유효

before_request

- 매 요청 전에 실행됨

```
@app.before_request
def before_request():
    pass
```

teardown_request

- 매 요청 마지막에 실행됨

```
@app.teardown_request
def teardown_request(exception):
    pass
```

실습

1. countdown을 g 객체에 저장해 이용하도록 변경
2. 로그인한 유저 정보를 g 객체에 저장하기

    - 메뉴바에 로그인 사용자 정보 보여주기

DB Relationship

두 테이블 사이에 관계를 설정해보자

Foreign Key

Join

SQLAlchemy Relationship

```
class User(db.Model):
    id = db.Column(db.Integer, primary_key=True)

    messages = db.relationship('Message',
                               backref=db.backref('writer',
                                   lazy='joined'),
                              lazy='dynamic')

class Message(db.Model):
    id = db.Column(db.Integer, primary_key=True)

    writer_user_id = db.Column(db.Integer,
                               db.ForeignKey('user.id'))
```

실습

1. 덧글을 로그인 한 사용자만 달 수 있도록 하고,

   작성자를 로그인 한 사용자를 참조하도록 수정해보자

2. 사용자 프로필 페이지를 만들고,

   해당 사용자가 작성한 글을 모아서 보여주자

   — db.relationship() 사용

뭐하지

- 메일보내기
- Deploy
- 기초시간 때 했던 거