

АВТОРИЗАЦИЯ В FLASK

Спикер

Немков Максим Юрьевич

СОДЕРЖАНИЕ ТЕМЫ

→ Регистрация и аутентификация пользователей

→ Работа с сессиями и куками

→ Обеспечение безопасности приложения

ТОКЕН

это уникальный код, который генерируется для каждого пользователя при регистрации. Токен может быть сохранен в базе данных или передан пользователю в виде файла cookie. При каждом запросе к приложению токен проверяется на подлинность. Если токен действителен, пользователь считается аутентифицированным

СЕССИЯ

это набор данных, который сохраняется на сервере и связан с определенным пользователем. Сессии используются для хранения информации о текущем состоянии пользователя, такой как логин, роль, предпочтения и т. д. Во Flask сессии можно реализовать с помощью встроенного механизма сеансов

РАБОТА С СЕССИЯМИ И КУКАМИ

В этом примере мы создаем сессию с именем «**username**» и присваиваем ей значение «**John Doe**». Теперь мы можем получить доступ к этой сессии в других функциях приложения

```
from flask import Flask, session

app = Flask(__name__)
app.config['SECRET_KEY'] = 'key'
@app.route("/login/")
def login():
    session["username"] = "John Doe"
    return "Вы успешно вошли в систему!"
```


РАБОТА С СЕССИЯМИ И КУКАМИ

Здесь мы проверяем, существует ли сессия с именем «`username`», и если да, то выводим приветственное сообщение. Если сессии нет, мы просим пользователя войти в систему

```
...
@app.route("/welcome")
def welcome():
    username = session.get("username", None)
    if username is not None:
        return f"Добро пожаловать, {username}!"
    else:
        return "Пожалуйста, войдите в систему!"
```

КУКИ

это небольшие файлы, которые хранятся на компьютере пользователя и содержат информацию о его взаимодействии с сайтом. Куки используются для отслеживания предпочтений пользователя, сохранения состояния сеанса и других целей. Во Flask куки можно создать и прочитать с помощью модуля «flask.cookies»

РАБОТА С СЕССИЯМИ И КУКАМИ

В этом примере мы устанавливаем куки с именем «username» и значением «John Doe». Теперь мы можем получить доступ к этой куке в других функциях приложения

```
...  
  
from flask import Flask, request, make_response  
  
app = Flask(__name__)  
  
@app.route("/set_cookie")  
def set_cookie():  
    response = make_response(render_template("index.html"))  
    response.set_cookie("username", "John Doe")  
    return response
```


РАБОТА С СЕССИЯМИ И КУКАМИ

Здесь мы проверяем, существует ли куки с именем «**username**», и если да, то выводим приветственное сообщение. Если куки нет, мы просим пользователя войти в систему. Аналогичным образом можно наладить работу и других функций вашего приложения

```
...  
@app.route("/welcome")  
def welcome():  
    username = request.cookies.get("username")  
    if username is not None:  
        return f"Добро пожаловать, {username}!"  
    else:  
        return "Пожалуйста, войдите в систему!"
```

Определение понятия

АУТЕНТИФИКАЦИЯ

это процесс проверки подлинности пользователя.
Во Flask можно реализовать аутентификацию
с помощью токенов или других методов

АВТОРИЗАЦИЯ

это определение прав доступа пользователя к определенным ресурсам. Во Flask можно использовать декораторы для ограничения доступа к функциям приложения

SQL-ИНЪЕКЦИЯ

это атака, при которой злоумышленник пытается внедрить вредоносный код в запрос к базе данных. Во Flask можно предотвратить SQL-инъекции, используя параметризованные запросы

ШИФРОВАНИЕ

это преобразование данных в нечитаемый формат. Во Flask можно шифровать данные, передаваемые по сети, с помощью модуля «flask_bcrypt»

HTTPS

это протокол, обеспечивающий защищенное соединение между клиентом и сервером. Во Flask можно настроить сервер для использования HTTPS с помощью модуля «werkzeug»

ПОДВЕДЕМ ИТОГИ

- ✓ Сделали регистрацию и аутентификацию пользователей
- ✓ Узнали про сессии куки
- ✓ Узнали про возможности обеспечения безопасности приложения

