🚀 Быстрый старт: Миграция на httpOnly cookies

Шаг 1: Установка зависимостей

Убедитесь, что у вас установлены необходимые пакеты:

```
pip install fastapi python-jose[cryptography] passlib[bcrypt] python-multipart
```

Добавьте в `requirements.txt`:

```
fastapi
python-jose[cryptography]
passlib[bcrypt]
python-multipart
uvicorn[standard]
```

Шаг 2: Настройка переменных окружения

Создайте или обновите файл `.env`:

```
# Секретный ключ для JWT (ОБЯЗАТЕЛЬНО ИЗМЕНИТЕ!)
SECRET_KEY=your-super-secret-key-minimum-32-characters-long

# Окружение (development или production)
ENVIRONMENT=production

# Остальные ваши переменные...
```

Сгенерируйте безопасный SECRET_KEY:

```
python -c "import secrets; print(secrets.token_urlsafe(32))"
```

Шаг 3: Интеграция кода

Вариант A: Использование готового примера

1. Скопируйте `auth_example.py` в ваш проект
2. Обновите функции работы с БД в `auth_example.py`:
   - `get_user_by_email()`
   - `authenticate_user()`
3. Добавьте в `main.py`:

```python
from fastapi import FastAPI
from fastapi.middleware.cors import CORSMiddleware
import auth_example

app = FastAPI()

# CORS настройки
app.add_middleware(
    CORSMiddleware,
    allow_origins=[
        "http://localhost:3000",
        "https://upak.space",
        "https://www.upak.space"
    ],
    allow_credentials=True,  # ОБЯЗАТЕЛЬНО!
    allow_methods=["*"],
    allow_headers=["*"],
)

# Подключение роутеров
app.include_router(auth_example.router)
app.include_router(auth_example.protected_router)
```

Вариант Б: Ручная интеграция

Следуйте инструкциям в `BACKEND_MIGRATION_GUIDE.md`

Шаг 4: Обновление существующих эндпоинтов

Замените все использования `OAuth2PasswordBearer` на `Cookie`:

Было:

```python
from fastapi.security import OAuth2PasswordBearer
oauth2_scheme = OAuth2PasswordBearer(tokenUrl="/v2/auth/token")

@app.get("/v2/me")
async def get_me(token: str = Depends(oauth2_scheme)):
    # ...
```

Стало:

```python
from auth_example import get_current_user

@app.get("/v2/me")
async def get_me(current_user = Depends(get_current_user)):
    # ...
```

Шаг 5: Тестирование

Локальное тестирование

1. Запустите сервер:

```
uvicorn main:app --reload --host 0.0.0.0 --port 8000
```

1. Тест входа:

```
curl -X POST "http://localhost:8000/v2/auth/token" \
  -H "Content-Type: application/x-www-form-urlencoded" \
  -d "username=test@upak.space&password=StrongPass123" \
  -c cookies.txt -v
```

Проверьте в выводе наличие `Set-Cookie: access_token=...`

1. Тест защищенного эндпоинта:

```
curl -X GET "http://localhost:8000/v2/me" \
  -b cookies.txt
```

1. Тест выхода:

```
curl -X POST "http://localhost:8000/v2/auth/logout" \
  -b cookies.txt -c cookies.txt -v
```

Тестирование с фронтендом

1. Запустите фронтенд (после мерджа PR):

```
cd Upak-frontend-NEW-v3
npm run dev
```

1. Откройте http://localhost:3000/login
2. Войдите с тестовыми данными
3. Проверьте, что перенаправляет на /dashboard
4. Проверьте работу всех страниц
5. Проверьте выход из системы

Шаг 6: Развертывание на сервере

На сервере 51.250.110.59:

1. Подключитесь к серверу:

```
ssh user@51.250.110.59
```

1. Перейдите в директорию проекта:

```
cd /path/to/upak-backend
```

1. Обновите код:

```
git pull origin main
```

1. Обновите зависимости:

```
pip install -r requirements.txt
```

1. Обновите .env файл:

```
nano .env
# Добавьте SECRET_KEY и ENVIRONMENT=production
```

1. Перезапустите сервер:

Если используете systemd:

```
sudo systemctl restart upak-backend
sudo systemctl status upak-backend
```

Если используете Docker:

```
docker-compose down
docker-compose up -d --build
docker-compose logs -f
```

Если используете screen/tmux:

```
# Найдите процесс
ps aux | grep uvicorn
kill <PID>

# Запустите заново
screen -S upak-backend
uvicorn main:app --host 0.0.0.0 --port 8000
# Ctrl+A, D для отсоединения
```

1. Проверьте работу:

```
curl -X GET "https://api.upak.space/v2/me" -v
# Должен вернуть 401 Unauthorized
```

Шаг 7: Проверка HTTPS

⚠️ **КРИТИЧЕСКИ ВАЖНО:** httpOnly cookies с `secure=True` работают только по HTTPS!

Проверьте, что ваш сервер доступен по HTTPS:

```
curl -I https://api.upak.space
```

Если HTTPS не настроен, используйте Let’s Encrypt:

```
sudo apt install certbot python3-certbot-nginx
sudo certbot --nginx -d api.upak.space
```

Шаг 8: Мониторинг

Проверьте логи на наличие ошибок:

```
# Systemd
sudo journalctl -u upak-backend -f

# Docker
docker-compose logs -f

# Файловые логи
tail -f /var/log/upak-backend.log
```

Чеклист развертывания

- [ ] Установлены все зависимости
- [ ] Настроен SECRET_KEY в .env
- [ ] ENVIRONMENT=production в .env
- [ ] CORS настроен с allow_credentials=True
- [ ] Обновлены все защищенные эндпоинты
- [ ] Сервер работает по HTTPS
- [ ] Тестирование входа работает
- [ ] Тестирование /v2/me работает
- [ ] Тестирование выхода работает
- [ ] Фронтенд успешно подключается
- [ ] Логи не показывают ошибок

Откат изменений (если что-то пошло не так)

1. Откатите код:

```
git revert HEAD
git push
```

1. Перезапустите сервер
2. Откатите фронтенд PR

Получение помощи

Если возникли проблемы:

1. Проверьте логи сервера
2. Проверьте консоль браузера (F12)
3. Проверьте Network tab в DevTools
4. Убедитесь, что CORS настроен правильно

5. Убедитесь, что HTTPS работает

Полезные команды для отладки

```
# Проверка cookies в curl
curl -X POST "https://api.upak.space/v2/auth/token" \
  -H "Content-Type: application/x-www-form-urlencoded" \
  -d "username=test@upak.space&password=StrongPass123" \
  -c cookies.txt -v

# Просмотр cookies
cat cookies.txt

# Тест с cookies
curl -X GET "https://api.upak.space/v2/me" \
  -b cookies.txt -v

# Проверка CORS
curl -X OPTIONS "https://api.upak.space/v2/me" \
  -H "Origin: https://upak.space" \
  -H "Access-Control-Request-Method: GET" \
  -v
```

Контакты

При возникновении вопросов обращайтесь к документации:
- `BACKEND_MIGRATION_GUIDE.md` - полное руководство
- `auth_example.py` - готовый код для интеграции