

# Оптимизация времени отклика эндпоинта /users/getReview

Nikita

23 ноября 2025 г.

## 1 Проблема

Исходный код эндпоинта **GET /users/getReview** в **UserService** выполнял запрос к базе данных без оптимизации загрузки связанных данных (ревьюеров PR). Это приводило к выполнению дополнительных SQL-запросов для каждого найденного PR, что увеличивало время отклика, особенно под нагрузкой.

## 2 Решение

Было применено предзагрузка связанных данных с использованием метода **selectinload** из SQLAlchemy. Это позволяет загрузить связанные объекты **reviewers** для всех найденных PR в рамках одного или нескольких дополнительных запросов, избегая N+1 проблеме.

Изменения в `app/services/user_service.py`:

```
# До:
prs_orm = self.db.query(PullRequestORM).filter(
    PullRequestORM.reviewers.any(UserORM.user_id == user_id)
).all()

# После:
from sqlalchemy.orm import selectinload

prs_orm = self.db.query(PullRequestORM).options(
    selectinload(PullRequestORM.reviewers)
).filter(
    PullRequestORM.reviewers.any(UserORM.user_id == user_id)
).all()
```

## 3 Результаты нагрузочного тестирования

Тестирование проводилось с помощью инструмента **hey** с одинаковыми параметрами до и после оптимизации:

- Количество запросов: 1000
- Количество конкурентных соединений: 10

- Таймаут: 10 секунд
- URL: [http://193.47.60.230:8080/users/getReview?user\\_id=load\\_user\\_2](http://193.47.60.230:8080/users/getReview?user_id=load_user_2)

Метрика	До оптимизации	После оптимизации
Среднее время отклика	307.6 мс	290.0 мс
RPS	32.14	34.24

Таблица 1: Сравнение метрик до и после оптимизации

## 4 Анализ

- Среднее время отклика снизилось с 307.6 мс до 290.0 мс.
- RPS увеличилась с 32.14 до 34.24.