

## Анализ проекта security.py от 01.12.2024, 08:40:16

📅 Дата последнего изменения проекта: 01.12.2024, 08:40:16

### 🗨 Замечаний: 3

Аутентификация и авторизация: 1

Работа с данными: 2

## ■ Аутентификация и авторизация

📄 /tmp/tmp743wwhnb/security.py

- ❗ Проверьте правильность обработки JWT-токенов и убедитесь, что все необходимые данные о пользователе содержатся в токене. Также убедитесь, что токен правильно расшифровывается с использованием пакета pyjwt.

```
12.     return {
13.         'user_id': identity.id,
14.         'username': identity.username,
15.         'roles': [role.name for role in identity.roles]
16.     }
17.
```

## ■ Работа с данными

📄 /tmp/tmp743wwhnb/security.py

- ❗ Использование ORM-запроса `User.query.get(user_id)` может быть диалекто-зависимым. Рассмотрите возможность использования более универсального подхода или пометьте этот участок кода как `# TODO: dialect dependent`.

```
28. def user_loader(user_id):
29.     # the user will be now available in current_user
30.     return User.query.get(user_id)
31.
32.
```

📄 /tmp/tmp743wwhnb/security.py

- ❗ Функция `validate_file_upload` не относится к слою `adapters` и не связана с обработкой JWT-токенов. Перенесите эту функцию в соответствующий слой приложения, например, в `core`.

```
70. def validate_file_upload(filename):
71.     return '.' in filename and \
```

