

Задание: "Система мониторинга и управления доступом для банковских операций"

Цель: Создать систему, которая будет обрабатывать банковские транзакции, добавляя к ним сквозную функциональность: логирование, проверку прав доступа, контроль лимитов и кэширование.

Вы создаете ядро банковского приложения. У вас есть базовая функция `process_transaction(account_id: int, amount: float, transaction_type: str) -> dict`, которая выполняет транзакцию (например, "withdraw" или "deposit") и возвращает словарь с результатом.

Ваша задача — не меняя код самой функции, обернуть ее с помощью декораторов, чтобы добавить следующую функциональность:

1. Логирование (`@log_transaction`): Записывает в файл или выводит в консоль детали каждой транзакции: время, тип операции, сумму, ID счета и результат.
2. Контроль доступа (`@require_role(role)`): Проверяет, имеет ли пользователь (его роль) право на выполнение операции. Например, роль "client" не может выполнять "withdraw" на сумму более 50_000.
3. Лимит операций (`@limit(rate, period)`): Ограничивает количество вызовов функции `process_transaction` для одного `account_id`. Не более `rate` вызовов за указанный `period` в секундах.
4. Кэширование баланса (`@cache_balance(ttl)`): Кэширует последний известный баланс для счета на определенное время (TTL). Если в течение TTL запрашивается баланс того же счета, он возвращается из кэша, без вызова основной функции.