

Тестовое задание

Для внутренних нужд отдела разработки требуется подготовить решение, которое позволяет :

- хранить информацию о состоянии счета абонента
- производить операции с счетом абонента
- по запросу выдавать текущие параметры счета абонента

Счет абонента состоит из следующих сущностей:

- уникальный номер абонента (uuid v4)
- ФИО абонента (string)
- текущий баланс на счете (int)
- холды на счете (int)
- статус счета (закрыт/открыт, bool)

Спецификация сущностей:

- уникальный номер — номер счета, уникальный в пределах множества абонентов (uuid v4)
- ФИО абонента — фамилия, имя и отчество абонента, написанные кириллицей или латиницей
- текущий баланс на счете — текущие денежные средства на счете абонента (рубли и копейки)
- холды на счете — зарезервированные к выполнению операции на счете (рубли и копейки)
- статус счета — определяет возможность проведения операций по счету (закрыт — нельзя, открыт — можно)

Технические требования к разработке:

- Осуществить 3 метода api:
 - /api/ping (работоспособность сервиса)
 - /api/add (пополнение баланса)
 - /api/substract (уменьшение баланса)
 - /api/status (остаток по балансу, открыт счет или закрыт)
- При выполнении операции substract проверять возможна ли данная операция по условию (возможна, если размер холда + сумма вычета меньше баланса на счете абонента):

```
result = balance — hold — substraction
if result < 0:
    isPossible = False
```

```
else:  
    isPossible = True
```

- При уменьшении баланса производить сложение поля hold с суммой вычета.
- Каждые 10 минут производить вычет суммы холда из баланса абонента с последующим очищением холда.
- Обмен производится в виде json запросов
- Общий шаблон json сообщения:

```
{  
    "status" = <http_status>,  
    "result": <bool:operation_status>,  
    "addition": {},  
    "description": {}  
}
```

- Описание полей:
 - status — http статус запроса
 - result — статус проведения текущей операции
 - addition — поля для описания текущей операции (uuid, ФИО, сумма, статус и т.п.)
 - description — дополнительные описания к текущей операции (прочие текстовые поля, если необходимо)
- Для реализации использовать python3

Технические требования к эксплуатации:

- Все сервисы разворачиваются, используя docker-compose.
- Сборка образов и разворачивание контейнеров осуществляется через bash-скрипт.
- При падении контейнеров приложения производить автоматический перезапуск контейнера 3 раза.
- Для публикации канала обмена с сервисом использовать nginx как reverse-проxy (80 порт).

Тестовые данные для БД

Uuid	ФИО	Баланс	Холд	Статус
26c940a1-7228-4ea2-a3bc-e6460b172040	Петров Иван Сергеевич	1700	300	открыт
7badc8f8-65bc-449a-8cde-855234ac63e1	Kazitsky Jason	200	200	открыт
5597cc3d-c948-48a0-b711-393edf20d9c0	Пархоменко Антон Александрович	10	300	откры
867f0924-a917-4711-939b-90b179a96392	Петечкин Петр Измаилович	1 000 000	1	закрыт