

[payment-service] Реализация механизма управления токеном для API Paynet

Цель

Задействовать существующий механизм работы с Bearer-токеном при обращении к защищённым ресурсам (на примере Acquiring-сервиса) в новом сервисе.

Обеспечить:

- автоматическое получение и обновление токена без использования шедулера;
- безопасное хранение токена;
- устойчивость к параллельным обращениям и повторному запросу токена.

Описание функционала

Первичное обращение

- При первом запросе к защищённому ресурсу система проверяет наличие актуального токена в БД.
- Если токен отсутствует или просрочен:
 - Выполняется запрос к эндпоинту получения токена у провайдера;
 - Токен сохраняется в БД;
 - Далее выполняется исходный запрос к целевому ресурсу с новым токеном.

Проверка валидности токена

- При каждом запросе токен передаётся в теле запроса `"token": "{{token}}"`
- Если получен ответ:

```
1 {  
2   "jsonrpc": "2.0",  
3   "id": 123,  
4   "error": {  
5     "code": -9999,  
6     "message": "unauthorized"  
7   }  
8 }
```

- система определяет, что токен недействителен, и инициирует процесс обновления.

Обновление токена

- Выполняется запрос к эндпоинту получения токена `/v2/user`.

request

```
1 {
```

```

2    "jsonrpc": "2.0",
3    "method": "login",
4    "id": 1,
5    "params": {
6        "username": "username_17627_56530",
7        "password": "password_17627_56530",
8        "terminal_id": 4027096
9    }
10 }

```

response

```

1  {
2      "jsonrpc": "2.0",
3      "id": 1,
4      "result": {
5          "agentId": "17627",
6          "terminalId": "4027096",
7          "userId": 56530,
8          "userLogin": "username_17627_56530",
9          "token": "RV5LwDHwB1ZJUBEfDbRNNaA64LynT5VG"
10     }
11 }

```

- Новый токен сохраняется в БД .
- После обновления выполняется повтор исходного запроса.

Механизм блокировки (Lock)

- При параллельных запросах от разных потоков используется шедлок или аналогичный механизм, чтобы предотвратить множественное обновление токена.
- Если один поток уже инициировал обновление токена, другие ожидают завершения операции и используют обновлённый токен из БД.

Хранение данных

- Конфигурационные данные (URL, username, password, terminal_id и др.) хранятся в YAML, аналогично другим интеграциям.

Нефункциональные требования

- Обновление токена выполняется **только по необходимости**.
- Система должна логировать все ошибки получения токена (недоступность API, неверные креды и т.д.).

Критерии приёмки

ID	Имя	Дано	Когда	Тогда
----	-----	------	-------	-------

1	Первичное получение токена	Токен отсутствует в БД	Отправляется запрос к защищённому ресурсу	Система получает токен, сохраняет его и выполняет исходный запрос
2	Обновление истекшего токена	В БД есть токен, но он недействителен	Ресурс возвращает "message": "unauthorized"	Система получает новый токен, сохраняет и повторно выполняет запрос успешно
3	Обработка недоступности токена-эндпоинта	Эндпоинт получения токена недоступен	Отправляется запрос на обновление токена	Система логирует ошибку и возвращает корректное описание сбоя
4	Конкурентное обращение	Несколько потоков одновременно обращаются к ресурсу с истёкшим токеном	Один поток обновляет токен, остальные ждут	Все потоки используют обновлённый токен