Информация по эксплуатации ПО

«Цифровое казначейство»

Оглавление

[Инструкция по использованию ПО 3](#_Toc195089877)

[Локальное развертывание 6](#_Toc195089878)

[Описание контейнеров: 7](#_Toc195089879)

[Данные для тестирования: 8](#_Toc195089880)

[Руководство пользователя API 8](#_Toc195089881)

[Содержание 8](#_Toc195089882)

[1. Авторизация 8](#_Toc195089883)

[1.1 Получение токена авторизации 8](#_Toc195089884)

[1.2 Использование токена 8](#_Toc195089885)

[2. Жизненный цикл заявок и соответствующие API эндпоинты 9](#_Toc195089886)

[2.1 Создание заявки 9](#_Toc195089887)

[2.2 Прямая публикация заявки 10](#_Toc195089888)

[2.3 Отправка на модерацию 10](#_Toc195089889)

[2.4 Работа с активной заявкой 10](#_Toc195089890)

[2.5 Работа с откликами 10](#_Toc195089891)

[2.6 Просмотр истории 12](#_Toc195089892)

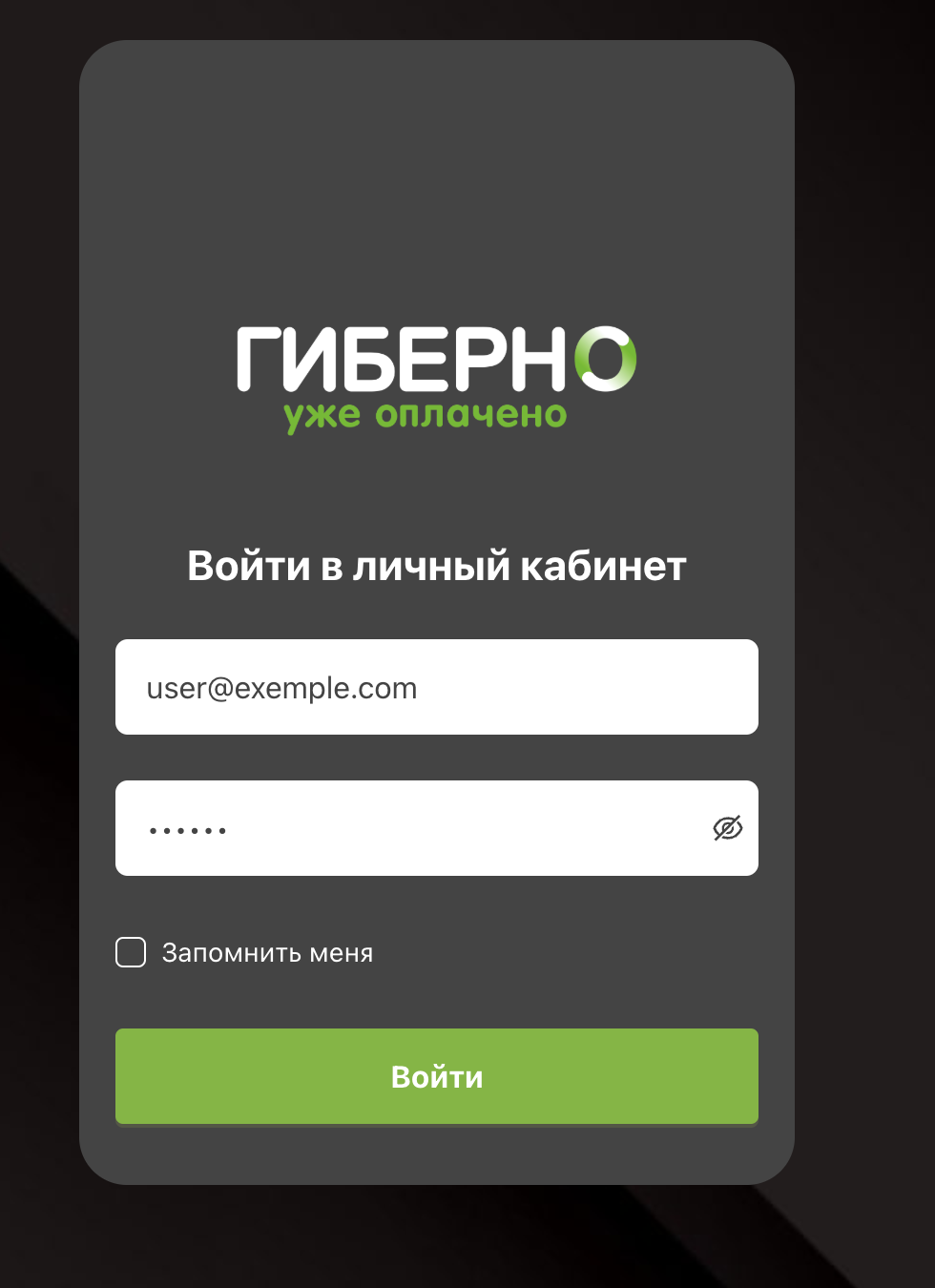
[3. Поиск и фильтрация 12](#_Toc195089893)

[3.1 Поиск заявок 12](#_Toc195089894)

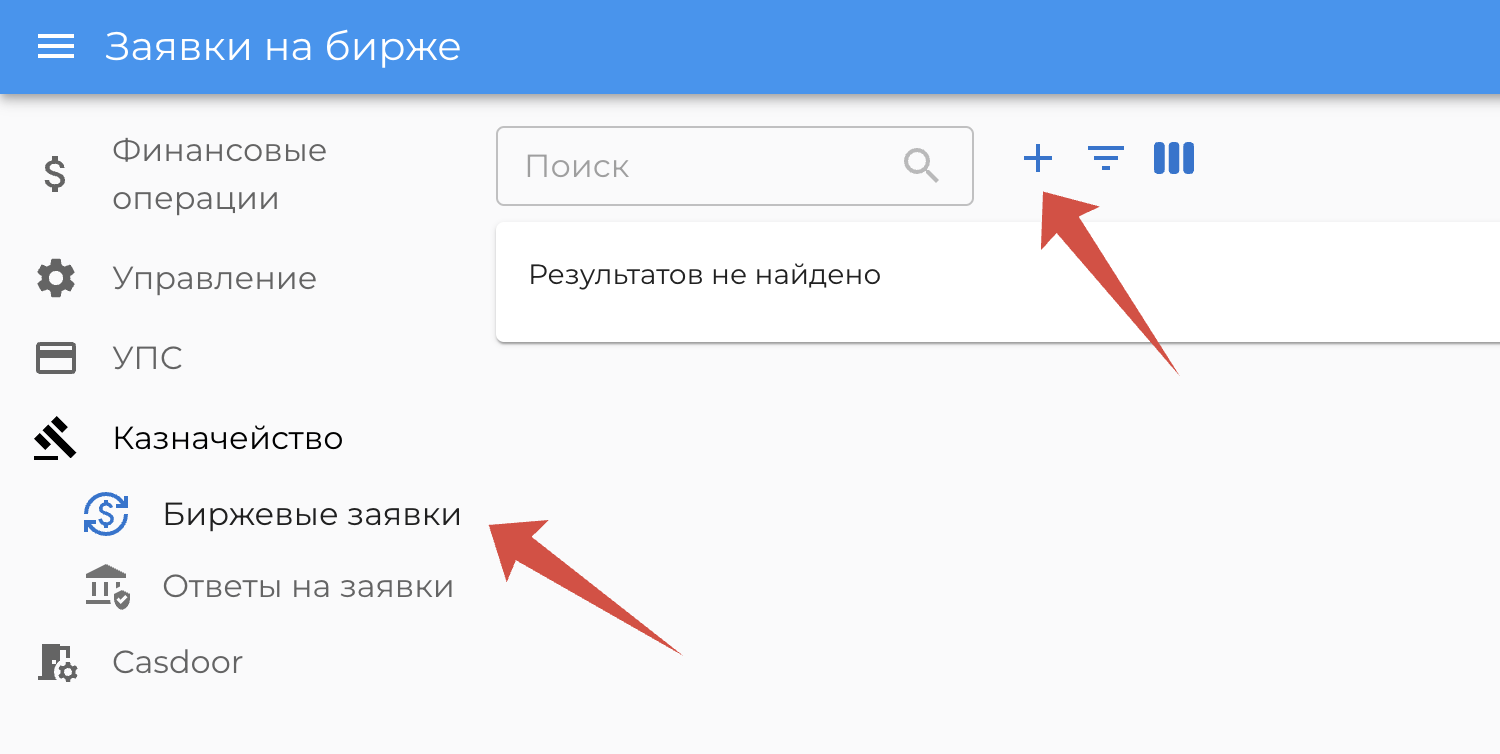
[3.2 Поиск откликов 12](#_Toc195089895)

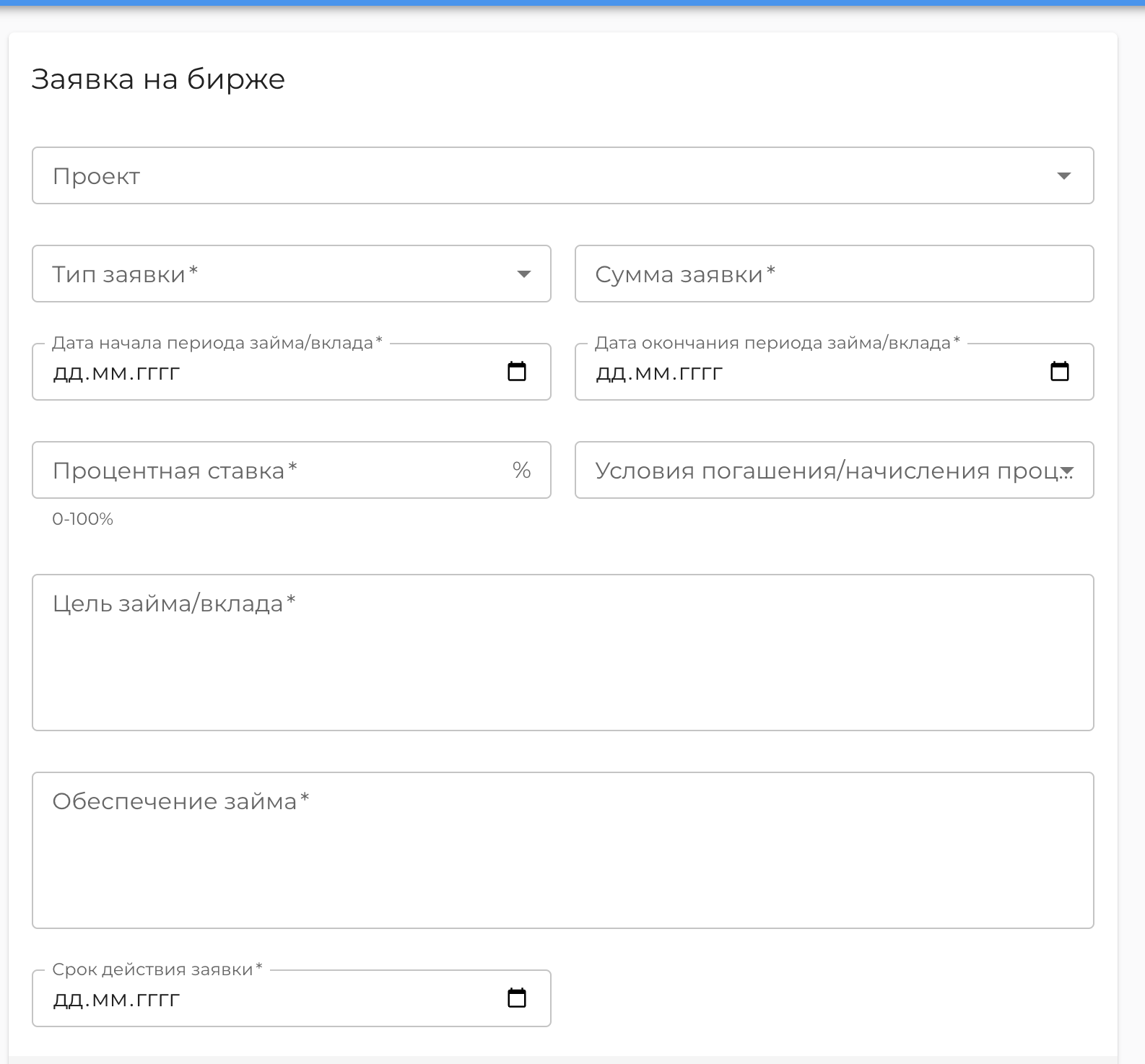
# Инструкция по использованию ПО

1. Перейдите по ссылке в личный кабинет и авторизуйтесь. Login - [user@exemple.com](mailto:user@exemple.com) Password 123456



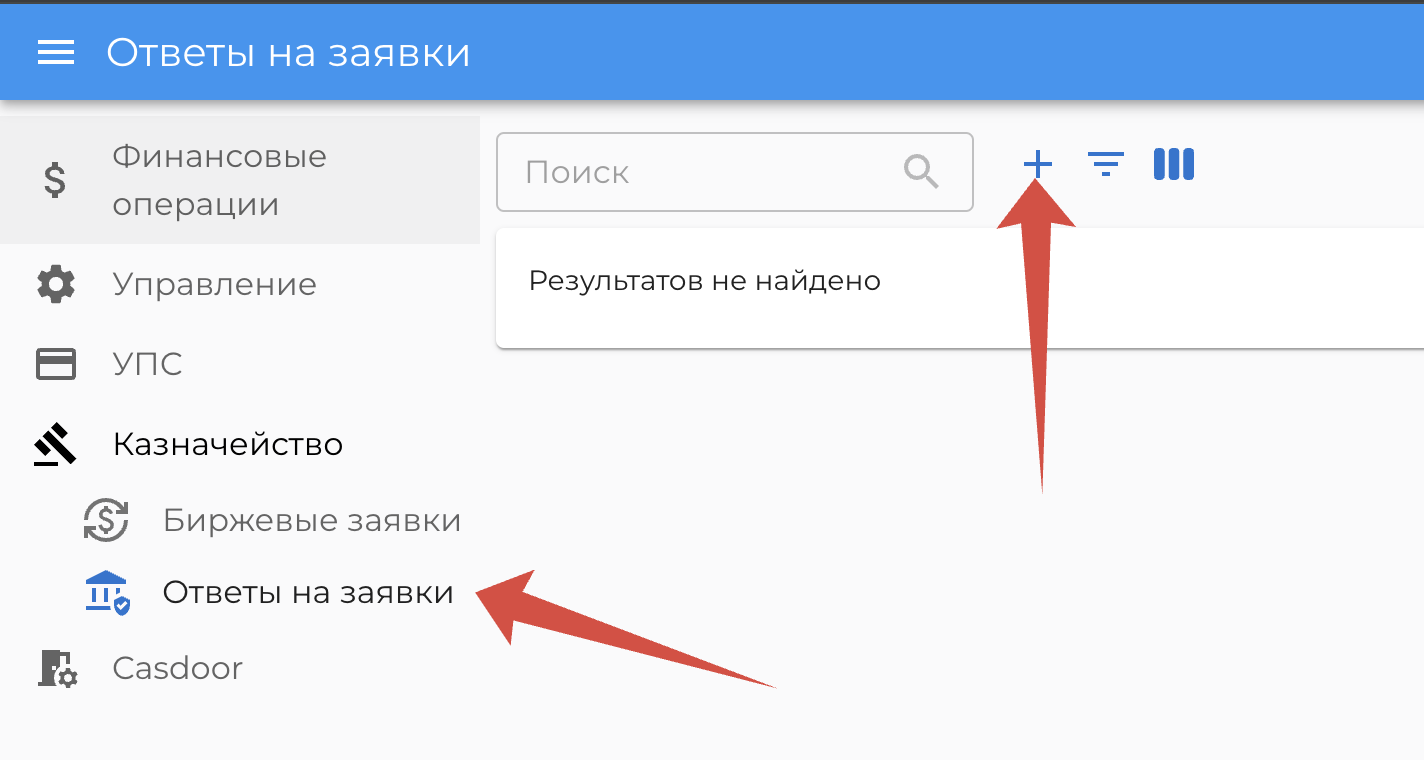
2. Для создания и публикации биржевой заявки перейдите в раздел «Биржевые заявки» и нажмите на кнопку «Создать заявку»



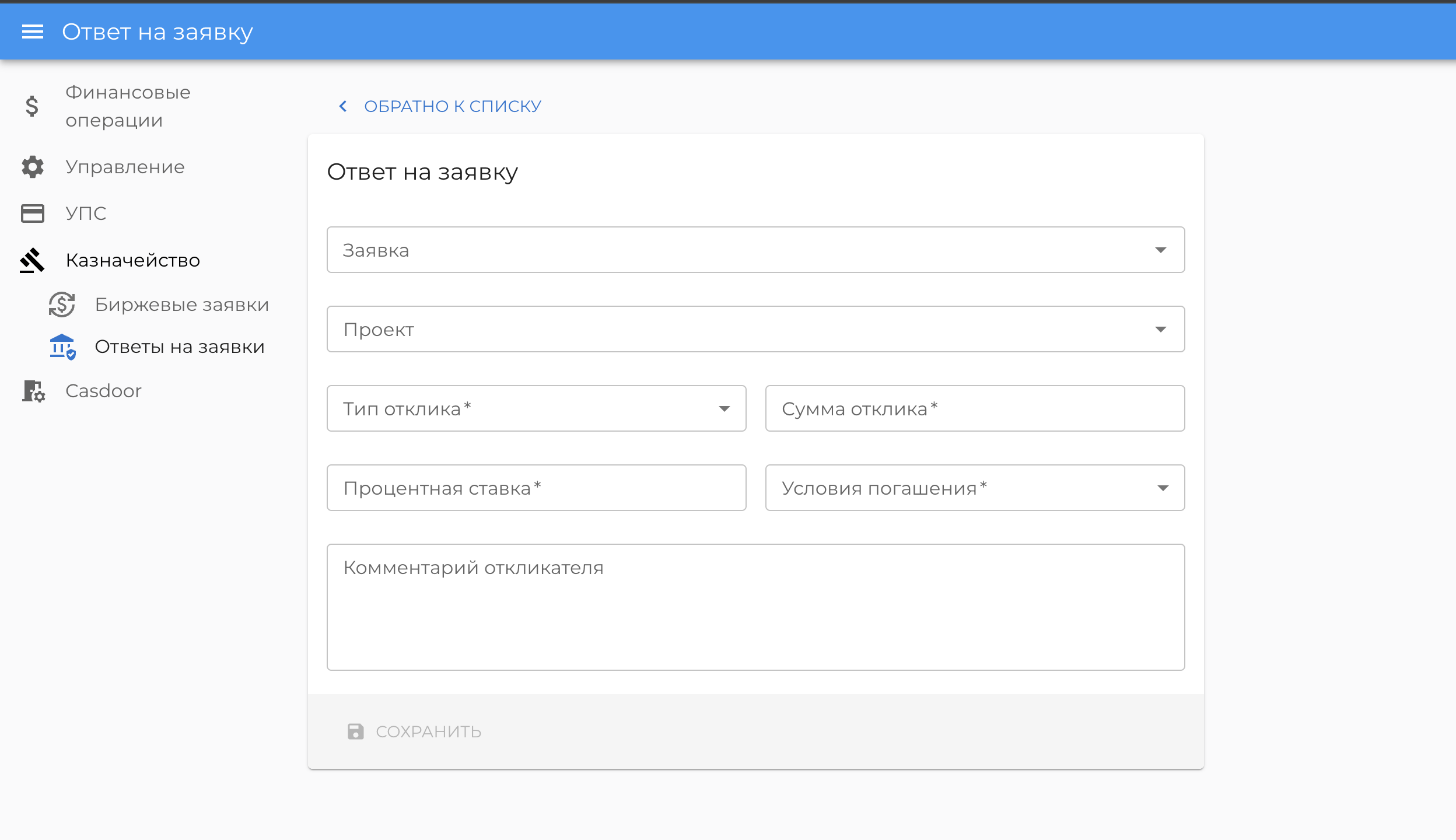
3. Заполните и сохраните данные по заявке

4. Выберите подходящий формат заявки и отправьте её на модерацию.

5. Если вы хотите протестировать функционал займополучателя, то вам необходимо перейти в раздел «Ответы на заявки» и создать ответ.



6. Создайте ответ на биржевую заявку



7. Выберите подходящий формат ответа на заявку и отправьте его на модерацию.

# Локальное развертывание

Ссылка для скачивания - https://cloud.mail.ru/public/68ve/uKUMcgeD4

Исходный код приложения можно развернуть локально для тестирования и разработки.

**Для этого необходимо:**

**Шаг 1: Установка средств контейнеризации**

Установите Docker и docker-compose с официального сайта: <https://www.docker.com>

**Шаг 2: Подготовка проекта**

Разархивируйте скачанный архив и перейдите в созданный каталог.

**Шаг 3: Запуск контейнеров**

Внутри каталога выполните следующую команду:

docker compose -f docker-compose.yml -d --build

Файл `docker-compose.yml` включает шесть контейнеров, необходимых для работы модуля.

## Описание контейнеров:

1. API: Контейнер с бэкендом приложения, содержащий всю бизнес-логику. При сборке загружаются все необходимые зависимости и выполняются миграции базы данных. После запуска стартует сервер Gunicorn, обрабатывающий API-запросы через порт 8000. Контейнер взаимодействует с основной базой данных PostgreSQL и Redis для кэширования.

2. DB: Контейнер с основной базой данных PostgreSQL 15, используемый для хранения всей информации приложения. Данные сохраняются в Docker volume `giberno-patent-db`. Миграции схемы базы данных выполняются автоматически при запуске. Порт доступа к базе данных — 5432.

3. Casdoor: Контейнер с системой авторизации и аутентификации. Обеспечивает управление пользователями, их правами и ролями. При первом запуске выполняется начальная настройка системы с предустановленными данными. Порт доступа — 8000.

4. Casdoor-DB: Контейнер с базой данных PostgreSQL для системы Casdoor. Хранит информацию о пользователях, ролях и правах доступа. Данные сохраняются в Docker volume `giberno-patent-casdoor-db`. Порт доступа к базе данных — 5433.

5. Scheduler: Контейнер для выполнения фоновых периодических задач. Запускается вместе с основным API и обрабатывает задачи по расписанию, такие как проверка статусов заявок, обновление данных и другие автоматические процессы. Использует те же зависимости и конфигурацию, что и основной API.

После успешного завершения сборки всех контейнеров Платежный модуль готов к использованию. Для отправки запросов к API используйте адрес: <http://127.0.0.1:5181/docs>.

## Данные для тестирования:

Для тестирования в базу был занесён проект его uuid = 25442faf-cfc3-4d27-8598-d9b83f5e4dab

Также был занесён пользователь casdoor его данные: username = [user@exemple.com](mailto:user@exemple.com), password = 123456

# Руководство пользователя API

## Содержание

1. [Авторизация](#X174ddf289cf287de28f72ac1f846f295c19e9ea)
2. [Жизненный цикл заявок и соответствующие API эндпоинты](#X27029431bd930c324c976b9e89b89b9f1a3ca57)
3. [Поиск и фильтрация](#X06909d871159b4672016bbef407fcf601ec188a)

## 1. Авторизация

Авторизация в системе осуществляется через Casdoor. Для доступа к API необходимо:

### 1.1 Получение токена авторизации

1. Отправьте POST запрос на адрес:

<http://127.0.0.1:8000/api/login>

В теле запроса укажите:

{  
 "application": "lkz",  
 "organization": "giberno",  
 "username": "user@exemple.com",  
 "password": "123456",  
 "type": "token",  
 "signinMethod": "Password"  
}

1. В ответе вы получите токен авторизации:

{  
 "status": "ok",  
 "msg": "",  
 "data": "your\_access\_token"  
}

### 1.2 Использование токена

1. Полученный токен необходимо использовать в заголовке Authorization при каждом запросе к API:

Authorization: Bearer your\_access\_token

1. Токен действителен в течение определенного времени
2. При истечении срока действия токена необходимо получить новый, повторив процесс авторизации

## 2. Жизненный цикл заявок и соответствующие API эндпоинты

### 2.1 Создание заявки

POST /exchange\_orders

* Создает новую заявку в статусе “Черновик”
* Пользователь заполняет все необходимые данные заявки
* Заявка сохраняется в системе
* Требует прав доступа на создание заявок

#### Входные параметры:

* project\_id (UUID) — идентификатор проекта в Гиберно (обязательный)
* order\_type (int) — тип заявки:
  + 1 - Заёмщик
  + 2 - Вкладчик (обязательный)
* amount (Decimal) — сумма заявки (обязательный)
* period\_start (datetime) — дата начала периода займа/вклада (обязательный)
* period\_end (datetime) — дата окончания периода займа/вклада (обязательный)
* interest\_rate (Decimal) — процентная ставка (обязательный)
* purpose (string) — цель займа/вклада (необязательный)
* collateral (string) — обеспечение займа (необязательный)
* repayment\_terms (int) — условия погашения (обязательный):
  + 1 - Ежемесячно
  + 2 - Ежегодно
  + 3 - В конце срока
* expiration\_date (datetime) — дата истечения срока действия заявки (обязательный)

#### Валидация данных:

* Сумма заявки (amount) должна быть больше 0.01
* Дата окончания периода (period\_end) должна быть больше даты начала (period\_start)
* Для заявок типа “Заёмщик” (order\_type = 1) поле collateral (обеспечение) является обязательным
* Процентная ставка (interest\_rate) должна быть >= 0.00

### 2.2 Прямая публикация заявки

POST /exchange\_orders/{uuid}/publish

* Позволяет опубликовать заявку сразу в статусе “Активна”, минуя модерацию
* Доступно только для пользователей с соответствующими правами
* Заявка сразу становится доступной для просмотра и откликов
* Требует прав доступа на публикацию заявок
* Доступно только из статуса “Черновик”

### 2.3 Отправка на модерацию

POST /exchange\_orders/{uuid}/moderate

* Заявка переходит из статуса “Черновик” в статус “На модерации”
* Модератор получает уведомление о новой заявке
* Модератор может:
* PATCH /exchange\_orders/{uuid}/moderation\_status
  + Одобрить заявку (статус меняется на “Активна”)
  + Отклонить заявку (статус меняется на “Отклонена”)
* Требует прав доступа на отправку на модерацию

### 2.4 Работа с активной заявкой

#### 2.4.1 Обновление статуса заявки

PATCH /exchange\_orders/{uuid}/status

* Позволяет изменить статус заявки
* Доступные переходы статусов:
  + ACTIVE → COMPLETED (4) - заявка исполнена
  + ACTIVE → CANCELLED (6) - заявка отменена
  + MODERATION → REJECTED (5) - заявка отклонена
* Требует прав доступа на изменение статуса заявки

#### 2.4.2 Отмена заявки

POST /exchange\_orders/{uuid}/cancel

* Позволяет отменить активную заявку
* Статус заявки меняется на “Отменена”
* Требует прав доступа на отмену заявки

### 2.5 Работа с откликами

#### 2.5.1 Создание отклика

POST /exchange\_order\_responses

* Создает новый отклик в статусе “Ожидает рассмотрения”
* Привязывается к конкретной активной заявке
* Требует прав доступа на создание откликов

#### Входные параметры:

* exchange\_order\_id (UUID) — идентификатор заявки (обязательный)
* project\_id (UUID) — идентификатор проекта, от которого подан отклик (обязательный)
* type (int) — тип отклика (обязательный):
  + 1 - LOAN (Займ)
  + 2 - DEPOSIT (Вклад)
* amount (Decimal) — сумма отклика (обязательный)
* interest\_rate (Decimal) — процентная ставка (обязательный)
* repayment\_terms (int) — условия погашения (обязательный):
  + 1 - Ежемесячно
  + 2 - Ежегодно
  + 3 - В конце срока
* responder\_comment (string) — комментарий откликателя (необязательный)

#### Проверки:

* Заявка должна существовать в базе данных
* Заявка должна быть в статусе ACTIVE
* Заявка не должна быть просрочена (expiration\_date <= текущее время)
* Сумма отклика не должна превышать сумму заявки
* Сумма отклика должна быть больше 0.01
* Процентная ставка должна быть >= 0.00

#### 2.5.2 Рассмотрение отклика

PATCH /exchange\_order\_responses/{response\_id}/status

* Позволяет изменить статус отклика
* Автор заявки может:
  + Принять отклик (статус меняется на “Принят”)
  + PATCH /exchange\_orders/{uuid}/status
    - Если отклик принят полностью, заявка переходит в статус “Исполнена”
    - Если отклик принят частично:
      1. Создается новая заявка на остаток суммы
      2. Текущая заявка переходит в статус “Исполнена”
  + Отклонить отклик (статус меняется на “Отклонен”)
* Требует прав доступа на изменение статуса отклика

#### Входные параметры:

* response\_id (UUID) — идентификатор отклика (обязательный)
* status (int) — новый статус отклика (обязательный):
  + 2 - ACCEPTED (Принят)
  + 3 - REJECTED (Отклонён)
* order\_comment (string) — комментарий от заявки (необязательный)

#### 2.5.3 Отмена отклика

POST /exchange\_order\_responses/{uuid}/cancel

* Позволяет отозвать отклик
* Статус отклика меняется на “Отозван”
* Требует прав доступа на отмену отклика
* Доступно только из статуса “Ожидает рассмотрения”

### 2.6 Просмотр истории

#### 2.6.1 История заявок

POST /exchange\_orders/history/find

* Просмотр всех изменений заявки
* Требует прав доступа на просмотр истории заявок

#### 2.6.2 История откликов

POST /exchange\_order\_responses/history/find

* Просмотр всех изменений откликов
* Требует прав доступа на просмотр истории откликов

## 3. Поиск и фильтрация

### 3.1 Поиск заявок

POST /exchange\_orders/find

* Поиск заявок с применением фильтров
* Поддерживает пагинацию
* Требует прав доступа на просмотр заявок

GET /exchange\_orders/{id}

* Получение заявки по ID
* Требует прав доступа на просмотр заявок

### 3.2 Поиск откликов

POST /exchange\_order\_responses/find

* Поиск откликов с применением фильтров
* Поддерживает пагинацию
* Требует прав доступа на просмотр откликов

GET /exchange\_order\_responses/{id}

* Получение отклика по ID
* Требует прав доступа на просмотр откликов

|  |
| --- |
| При возникновении любых вопросов или проблем в процессе тестирования обращайтесь к руководителю проекта Роману Егорову:  Телефон: +7 999 831 08 14  Telegram: @gorov |