АО "НТЦ ИТ РОСА"

СЕРВИС ИДЕНТИФИКАЦИИ И УПРАВЛЕНИЯ АККАУНТАМИ "РОСА ID"

Руководство администратора

РСЮК.10903-01 32 01

Листов 94

2026

АННОТАЦИЯ

Данное руководство предназначено для системных администраторов, осуществляющих развёртывание, сопровождение и эксплуатацию программного продукта "Сервис идентификации и управления аккаунтами РОСА ID" РСЮК.10903-01 (далее – РОСА ID, Сервис).

Перед использованием РОСА ID рекомендуется внимательно ознакомиться с настоящим руководством.

Сведения, необходимые для эксплуатации РОСА ID, приведены в документе "РОСА ID. Руководство пользователя" (индекс РСЮК.10903-01 34 01).

При разработке документа использованы ссылки на следующие стандарты:

– ГОСТ Р 2.105-2019 "Единая система конструкторской документации (ЕСКД). Общие требования к текстовым документам";

– ГОСТ 2.601 "Единая система программной документации. Виды программных документов";

– ГОСТ 19.101-77 "Единая система программной документации. Виды программ и программных документов";

– ГОСТ 19.105-78 "Единая система программной документации. Общие требования к программным документам";

– ГОСТ 19.503-79 "Единая система программной документации. Руководство системного программиста".

Настоящий документ подготовлен в соответствии с технологической инструкцией "РОСА. Регламент формирования документации к программным продуктам" (индекс РСЮК.11001-03 90 01).

СОДЕРЖАНИЕ

Содержание....3
1 Назначение Комплекса....6
1.1 Общие сведения....6
1.2 Обеспечение безопасности в POCA ID....6
1.2.1 Шифрование данных....6
1.2.2 Использование протокола HTTPS для обеспечения сетевой безопасности....7
2 Общая идеология API Сервиса....9
3 Управление авторизацией....10
3.1 Авторизация (Auth)....10
3.1.1 Запросить OTP-код....10
3.1.2 Регистрация пользователя....12
3.1.3 Подтверждение email при регистрации....14
3.1.4 Авторизация (логин)....15
3.1.5 Подтверждение входа через OTP....17
3.1.6 Восстановление пароля....19
3.1.7 Обновление токена....20
3.1.8 Проверка пароля....21
3.2 История авторизаций....23
3.2.1 Список авторизаций пользователя....23
3.2.2 Удалить всю историю авторизации пользователя....25
3.2.3 Удалить запись авторизации пользователя....25
4 Пользователь....26
4.1 Профиль пользователя....26
4.2 Профиль указанного пользователя....28
4.3 Обновление профиля....29
4.4 Удаление аккаунта....34
4.5 Получить текущий аватар....35
4.6 Загрузить аватар....36
4.7 Удалить аватар....37
5 Управление сессиями и устройствами....38
5.1 Список сессий текущего пользователя....38
5.2 Удалить все сессии текущего пользователя....40
5.3 Удалить указанную сессию текущего пользователя....41
5.4 Список устройств текущего пользователя....42

5.5 Удалить все устройства пользователя ... 44
5.6 Информация об указанном устройстве ... 44
5.7 Удалить указанное устройство ... 46

6 Документы пользователя ... 47

6.1 Список документов пользователя ... 47
6.2 Создание документа ... 48
6.3 Получить документ ... 50
6.4 Обновить документ ... 51
6.5 Удалить документ ... 53

7 Партнёры пользователя ... 54

7.1 Список партнёров ... 54
7.2 Создать партнёра ... 55
7.3 Получить партнёра ... 56
7.4 Обновить партнёра ... 58
7.5 Удалить партнёра ... 59

8 OAuth-клиенты пользователя ... 60

8.1 Список OAuth-клиентов ... 60
8.2 Создать OAuth-клиента ... 61
8.3 Получить OAuth-клиента ... 63
8.4 Обновить OAuth-клиента ... 64
8.5 Удалить OAuth-клиента ... 66
8.6 Авторизация (OAuth) ... 67

8.6.1 OAuth-авторизация ... 67
8.6.2 Подтверждение OAuth-авторизации ... 68
8.6.3 Получение токена клиента ... 69
8.6.4 Данные пользователя (OIDC) ... 71

9 Вспомогательные методы API ... 72

9.1 Справочники ... 72

9.1.1 Список часовых поясов ... 72
9.1.2 Список регионов ... 74

9.2 Сообщения пользователя ... 75
9.3 Состояние сервиса ... 76
9.4 Функционал для разработчика ... 77

9.4.1 Список отправленных dev-OTP ... 77
9.4.2 Принудительно завершить access-токен ... 78

10 Используемые схемы данных ... 79

10.1 Timezone ... 79

10.2 Region 79
10.3 OTP 80
10.4 User 80
10.5 UserDetail 81
10.6 UserToken 82
10.7 AuthAuthorizeResponse 82
10.8 UserAvatar 83
10.9 Document 83
10.10 Partner 84
10.11 Client 84
10.12 DevOTPMessage 84
10.13 Feedback 85
10.14 OAuthAuthorizeGetResponse 85
10.15 OAuthAuthorizePostResponse 85
10.16 OIDCUserinfo 85
10.17 OAuthTokenResponse 86
10.18 PaginatedTimezones 86
10.19 PaginatedRegions 87
10.20 PaginatedDocuments 87
10.21 PaginatedPartners 87
10.22 PaginatedClients 88
10.23 AuthHistoryEntry 88
10.24 DeviceInfo 89
10.25 PaginatedAuthHistory 89
10.26 PaginatedSessions 89
10.27 PaginatedDevices 90
10.28 PaginatedDevOTPMessages 90
10.29 Error 90

ПРИЛОЖЕНИЕ А. Требования к заголовку User-Agent 91

Валидация со стороны Сервиса 92
Рекомендации разработчикам 92

Приложение Б. Коды ошибок, используемые в API 93
Перечень сокращений 94

1 НАЗНАЧЕНИЕ КОМПЛЕКСА

1.1 Общие сведения

POCA ID — это единый и безопасный способ входа (учетная запись) в приложения в экосистеме программных продуктов POCA.

Клиентом в Сервисе называется подключенный внешний сервис, взаимодействующий с зарегистрированным пользователем. Авторизация пользователя во внешнем сервисе осуществляется в соответствии со стандартом OpenID.

Основные возможности Сервиса:

– регистрация пользователей;
– авторизация по email/телефону и паролю;
– подтверждение операций через OTP;
– управление сессиями и refresh-токенами;
– хранение и управление профилем пользователя;
– работа с документами, партнёрами и клиентами;
– OAuth2/OpenID Connect-авторизация для сторонних сервисов;
– отправка обратной связи;
– dev-endpoint для тестирования.

1.2 Обеспечение безопасности в POCA ID

1.2.1 Шифрование данных

– Использование отраслевых стандартов шифрования (AES256, curl25519).
– Использование надёжных средств генерации и обмена ключами.
– Использование сессионных ключей для предупреждения взлома информации при передаче по открытым сетям:

– запрет на использование исходных ключей для обмена информацией;
– использование алгоритма HKDF высоких порядков для генерации производных ключей;
– смена ключей спустя короткое время (1 час, сутки).

– Унификация применения одних и тех же методов шифрования для защиты сообщений, файлов, изображений, видео, голосовых сообщений.
– Хранение ключей в специальном защищённом хранилище на уровне операционной системы, с невозможностью доступа сторонних приложений к ключам, сгенерированными другими приложениями, даже при доступе к shared

memory. Возможность автоматического стирания этого хранилища в случае утери телефона или несанкционированного доступа к нему, что делает расшифровку переписки полностью невозможной.

– Использование разных схем использования ключей на разных устройствах:

– синхронизация ключей на разных устройствах для оптимизации трафика;

– использование различных ключей на разных устройствах для того, чтобы исключить доступ к другим устройствам при компрометации одного из них.

– Возможность подключения "товарища майора" для получения доступа к защищённой переписке по требованию гос. органов.

– Исключение сервера из обеспечения шифрования, кроме хранения публично доступной информации (публичные ключи, ключи для доступа к группам), таким образом сервер, включая хранение информации в базе данных (БД), приравнивается к открытым каналам обмена информацией. Это делает невозможным получение доступа к переписке администратором базы данных или при взломе сервера БД, даже при получении к нему администраторского доступа.

– Применение шифрования к потоковому голосу и видео, с использованием тех же методов обмена ключами, что и для обычной переписки, метод шифрования — AES256.

1.2.2 Использование протокола HTTPS для обеспечения сетевой безопасности

– Весь обмен данными между клиентом и сервером идёт исключительно по протоколу HTTPS, нигде не используются сокеты или веб-сокеты.

– Проверка серверного сертификата, вплоть до зашивания в клиент самоподписанного сертификата SSL, позволяет устранить атаку MITM (Man-In-the-Middle), когда злоумышленник пытается использовать прокси для вскрытия обмена с сервером.

– Возможно использование широчайшего спектра возможностей, программных и аппаратных, для отражения атак типа DDOS на серверы, включая:

– возможности автоматической блокировки IP-адресов, с которых производится атака;

– включение шейперов трафика с ограничением по числу запросов для конкретных пользователей;

– возможности балансировки нагрузки между разными серверами для достижения оптимальной производительности системы.

– Проверка клиентских сертификатов для получения доступа к защищённой инфраструктуре.

– Проксирование запросов HTTPS через специальные прокси, которые бы занимались проверкой валидности пользователей, их возможностью доступа к серверу в определённое время суток, изоляцией доступа к защищённому контуру компании от внешнего мира.

– Использование аппаратных border manager, отличных от используемого по умолчанию nginx, для решения вопросов безопасности при сохранении высокой нагрузочной способности сервера.

2 ОПИСАНИЕ API

2.1 Общая идеология API-Сервиса

API POCA ID использует только HTTPS-запросы, направляемые на домен, на котором разворачивается платформа Сервиса. В запросе всегда указывается путь к нужному URL в домене относительно "/api". Например, для домена id.rosa.ru нужно использовать – id.rosa.ru/api. Если в описании указан запрос auth/authorize, то его URL необходимо сформировать следующим образом: https://id.rosa.ru/api/auth/authorize.

Все запросы, требующие авторизации пользователя, должны передавать токен доступа в заголовке X-Lamb-Auth-Token. Для OAuth-запросов используется стандартный заголовок "Authorization: Bearer <token>". Предусмотрена следующая логика авторизации:

а) пользователь регистрируется через /auth/register и подтверждает email через OTP-код;

б) выполняет вход по email/паролю через /auth/authorize, получая access_token и refresh_token;

в) при запросах к защищенным ресурсам передает access_token в заголовке X-Lamb-Auth-Token;

г) для OAuth-клиентов используется стандартный поток authorization_code.

Все приложения, выполняющие авторизацию, запрос токенов или управление сеансами через Роса ID должны передавать корректно сформированную информацию в заголовке User-Agent (требования см. Приложение А).

Заголовок User-Agent передаётся в каждом HTTP-запросе к API Роса ID.

Информация заголовка User-Agent используется Сервисом для:

– идентификации типа устройства (ПК, смартфон, сервер);

– корректного отображения списка подключённых устройств в личном кабинете пользователя;

– построения статистики по версиям ОС и приложений;

– обнаружения аномалий и подозрительных сеансов.

Некорректное или отсутствующее значение может привести к невозможности отобразить устройство в разделе «Устройства»/«Сеансы».

Запросы с методом POST или PATCH обычно содержат тело запроса в формате JSON, поэтому в запросе обязательно должно быть указано поле заголовка:

```
>Content-type: application/json
```

Тело запроса формируется как структура данных, содержащая перечисление параметров. Таким образом, если в описании конкретного запроса указаны параметры P1 и P2, то тело запроса должно быть сформировано как JSON следующего вида:

```
{"P1": value, "P2": value}
```

Все даты и время передаются в формате ISO 8601 в часовом поясе UTC. Unix-метки времени (TIME64) также поддерживаются в некоторых ответах, но в основном API использует ISO-строки для читаемости.

Пример: "2026-04-17T12:37:09Z".

Все методы, возвращающие коллекции, поддерживают пагинацию через параметры запроса offset и limit. Ответ с пагинацией содержит поля items, total_count, offset, limit.

Ответ сервера всегда приходит в формате JSON. В случае успеха возвращается HTTP-код 200 (или 204), а в теле — запрошенные данные.

В случае ошибки возвращается соответствующий HTTP-код (400, 401, 403, 404, 429, 500) и JSON-объект с полями error (символьный код ошибки) и message (описание ошибки) (см. Приложение Б).

2.2 Управление авторизацией

2.2.1 Авторизация (Auth)

2.2.1.1 Запросить OTP-код

Запрос OTP-кода (зависит от предаваемого значения intent: регистрация, восстановление, верификация и т. д.).

Запрос:

```
/auth/otp
```

Метод запроса: POST.

Интенты:

Таблица 1 – Интенты auth/otp

Код	Описание	Пояснение
1	подтверждение email при регистрации	требует email

2	двухфакторная авторизация	требует действующий токен
6	верификация email	требует действующий токен
7	верификация телефона	требует действующий токен
9	восстановление пароля по email	требует email
10	восстановление пароля по SMS	требует phone_number

Параметры запроса:

Таблица 2 – Параметры тела запроса auth/otp

Наименование	Тип данных	Пример
Intent	integer	*1*
Email	string	*user@example.com*
phone_number	string	*+79001234567*

Пример тела запроса:

```
{
  "intent": 1,
  "email": "user@example.com"
}
```

Ответ:

Таблица 3 – Ответ auth/otp

Код	Описание	Связи
200	OTP отправлен	нет

Схема данных ответа (Schema): OTP.

Пример ответа:

```
{
  "transport": "email",
  "intent": "email_registration",
  "credential": "user@example.com",
  "expires_at": "2026-04-18T12:37:09Z"
}
```

Возможные сообщения об ошибках:

Таблица 4 – Сообщения об ошибках auth/otp

Код	Значение
400	Неверные параметры запроса
401	Не авторизован
403	Ошибка доступа. Возможные error коды: - user_verified_already (1007) — пользователь уже верифицирован (intent=1); - credential_not_verified (1003) — нет подтверждённого контакта для данного transport.
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.2.1.2 Регистрация пользователя

Запрос на регистрацию нового пользователя.

Запрос:

```
/auth/register
```

Метод запроса: POST.

Параметры запроса:

Таблица 5 – Параметры тела запроса auth/register

Наименование	Тип данных	Пример
Email	string	*user@example.com*
password	string	*Pa$$w0rd*
Profile	object Profile	см. параметры Profile
newsletter_sub	boolean	*false (по умолчанию)*

Таблица 6 – Параметры Profile

Наименование	Тип данных	Пример
first_name	string	*Иван*
last_name	string, nullable: true	*Иванов*
middle_name	string	*Иванович*
Gender	string, nullable: true	*MALE, Enum: [MALE, FEMALE]*

birthdate	integer, nullable: true	*1990-01-15*

Пример тела запроса:

```
{
  "email": "user@example.com",
  "password": "SecurePass123",
  "profile": {
    "first_name": "Иван",
    "last_name": "Иванов",
    "middle_name": "Иванович",
    "gender": "male",
    "birthdate": "1990-01-15"
  },
  "newsletter_sub": false
}
```

Ответ:

Таблица 7 – Ответ auth/register

Код	Описание	Связи
200	Пользователь зарегистрирован. Требуется подтверждение email.	нет

Схема данных ответа (Schema): User.

Пример ответа:

```
{
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "email": "user@example.com",
  "email_verified": false,
  "phone_number": null,
  "phone_number_verified": false,
  "is_active": true,
  "otp_transport": "email",
  "first_name": "Иван",
  "last_name": "Иванов",
  "middle_name": "Иванович",
```

```
  "gender": "male",
  "birthdate": "1990-01-15",
  "timezone_id": null,
  "region_id": null,
  "address_actual": null,
  "email_candidate": null,
  "phone_number_candidate": null
}
```

Возможные сообщения об ошибках:

Таблица 8 – Сообщения об ошибках auth/register

Код	Значение
400	Неверные параметры запроса
401	Не авторизован
403	Ошибка доступа. Возможные error коды: - email_taken (1004) — email уже занят; - phone_number_taken (1005) — телефон уже занят.
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.2.1.3 Подтверждение email при регистрации

Запрос на подтверждение регистрации пользователя.

Запрос:

```
/auth/register/confirm
```

Метод запроса: POST.

Параметры запроса:

Таблица 9 – Параметры тела запроса auth/register/confirm

Наименование	Тип данных	Пример
Email	string	*user@example.com*
Code	string	*123456*

Пример тела запроса:

```
{
```

```
  "email": "user@example.com",
  "code": "123456"
}
```

Ответ:

Таблица 10 – Ответ auth/register/confirm

Код	Описание	Связи
204	Email подтверждён успешно	нет

Схема данных ответа (Schema): User.

Возможные сообщения об ошибках:

Таблица 11 – Сообщения об ошибках auth/register/confirm

Код	Значение
400	Неверные параметры запроса
401	Не авторизован
403	Ошибка доступа. Возможные error коды: - email_taken (1004) — email уже занят; - phone_number_taken (1005) — телефон уже занят.
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.2.1.4 Авторизация (логин)

Запрос на авторизацию пользователя по email/телефону и паролю.

Запрос:

```
/auth/authorize
```

Метод запроса: POST.

Параметры запроса:

Таблица 12 – Параметры тела запроса auth/authorize

Наименование	Тип данных	Пример
Email	string	*user@example.com*
phone_number	string	*+79001234567*
password	string	*SecurePass123*

Пример тела запроса:

```
{
  "email": "user@example.com",
  "password": "SecurePass123"
}
```

Ответ:

Таблица 13 – Ответ auth/authorize

Код	Описание	Связи
200	Возвращает созданную сессию авторизации с токенами и метаданными устройства/клиента.	нет

Схема данных ответа (Schema): AuthAuthorizeResponse.

Пример ответа:

```
{
  "time_created": 1742774400000,
  "time_updated": 1742774400000,
  "access_token":
"d0e88af65fbd63ca1e1d45b0ddda3265e64c50cba3041f0d5a09686f74448de
6",
  "refresh_token":
"40c06bf69b0e8ce5afff490260973954e4ad2b2417caaffdf23473e75d8d323
f",
  "user_id": "53ecec94-ae83-4d40-a39b-970a95018110",
  "expires_at": "2026-04-18T12:37:09Z",
  "status": "PENDING",
  "client": null,
  "device": {
    "device_platform": "mac",
    "device_os": "Mac OS X",
    "device_version": "10.15.7",
    "app_device": "Chrome",
    "app_build": "146.0.0",
    "device_locale": "ru",
```

```
        "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X
10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/146.0.0.0
Safari/537.36"
      }
    }
```

Возможные сообщения об ошибках:

Таблица 14 – Сообщения об ошибках auth/authorize

Код	Значение
400	Неверные параметры запроса
401	Не авторизован
403	Ошибка доступа. Возможные error коды: - user_deactivated (1001) — аккаунт деактивирован; - user_not_verified (1002) — email не подтверждён (регистрация не завершена); - credential_not_verified (1003) — email или телефон не подтверждён.
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.2.1.5 Подтверждение входа через OTP

Запрос на подтверждение авторизации пользователя по OTP-коду.

Запрос:

```
/auth/authorize/confirm
```

Метод запроса: POST.

Параметры запроса:

Таблица 15 – Параметры тела запроса auth/authorize/confirm

Наименование	Тип данных	Пример
code	string	*123456*

Пример тела запроса:

```
{
  "code": "123456"
}
```

Ответ:

Таблица 16 – Ответ auth/authorize/confirm

Код	Описание	Связи
200	Вход подтверждён. Возвращает подтверждённую сессию авторизации с токенами и метаданными устройства/клиента.	нет

Схема данных ответа (Schema): AuthAuthorizeResponse.

Пример ответа:


```
    {
      "time_created": 1742774400000,
      "time_updated": 1742774400000,
      "access_token":
"d0e88af65fbd63ca1e1d45b0ddda3265e64c50cba3041f0d5a09686f74448de
6",
      "refresh_token":
"40c06bf69b0e8ce5afff490260973954e4ad2b2417caaffdf23473e75d8d323
f",
      "user_id": "53ecec94-ae83-4d40-a39b-970a95018110",
      "expires_at": "2026-04-18T12:37:09Z",
      "status": "ACTIVE",
      "client": null,
      "device": {
        "device_platform": "mac",
        "device_os": "Mac OS X",
        "device_version": "10.15.7",
        "app_device": "Chrome",
        "app_build": "146.0.0",
        "device_locale": "ru",
        "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X
10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/146.0.0.0
Safari/537.36"
      }
    }
```

Возможные сообщения об ошибках:

Таблица 17 – Сообщения об ошибках auth/authorize/confirm

Код	Значение
400	Неверные параметры запроса
401	Не авторизован
403	Ошибка доступа. Возможные error коды: - otp_invalid (1021) — неверный OTP-код; - otp_expired (1022) — OTP-код истёк; - otp_attempts_exceeded (1023) — превышен лимит попыток OTP; - auth_forbidden — токен уже подтверждён.
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.2.1.6 Восстановление пароля

Запрос на восстановление доступа (смена пароля по OTP коду).

Запрос:

```
/auth/restore
```

Метод запроса: POST.

Параметры запроса:

Таблица 18 – Параметры тела запроса /auth/restore

Наименование	Тип данных	Пример
email	string	*user@example.com*
phone_number	string	*+79001234567*
code	string	*123456*
password	string	*Pa$$w0rd*

Пример тела запроса:

```
{
  "email": "user@example.com",
  "code": "123456",
  "password": "NewSecurePass123"
```

```
}
```

Ответ:

Таблица 19 – Ответ /auth/restore

Код	Описание	Связи
204	Пароль успешно изменён.	нет

Схема данных ответа (Schema): Нет.

Возможные сообщения об ошибках:

Таблица 20 – Сообщения об ошибках /auth/restore

Код	Значение
400	Неверные параметры запроса. Включает случай, когда OTPInvalidError перехватывается и возвращается как 400: invalid_param_value — неверный код OTP или неверный credential.
403	Ошибка доступа. Возможные error коды: - otp_expired (1022) — OTP-код истёк - otp_attempts_exceeded (1023) — превышен лимит попыток OTP.
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.2.1.7 Обновление токена

Запрос на обновление сессии по ранее полученному токену.

Запрос:

```
/auth/refresh
```

Метод запроса: POST.

Параметры запроса:

Таблица 21 – Параметры тела запроса /auth/refresh

Наименование	Тип данных	Пример
refresh_token	string	*dGhpcyBpcyBhIHJlZnJlc2ggdG9rZW4...*

Пример тела запроса:

```
{
```

```
  "refresh_token": "dGhpcyBpcyBhIHJlZnJlc2ggdG9rZW4..."
}
```

Ответ:

Таблица 22 – Ответ /auth/refresh

Код	Описание	Связи
200	Новый токен выдан.	нет

Схема данных ответа (Schema): UserToken.

Пример ответа:

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "refresh_token": "bmV3UmVmcmVzaFRva2VuSGVyZQ...",
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "expires_at": "2026-04-18T12:37:09Z",
  "otp_transport": "email",
  "is_confirmed": true,
  "device_info": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)"
}
```

Возможные сообщения об ошибках:

Таблица 23 – Сообщения об ошибках /auth/refresh

Код	Значение
400	Неверные параметры запроса
401	Не авторизован
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.2.1.8 Проверка пароля

Запрос на проверку пароля. Необходимо использовать для чувствительных операций.

Запрос:

```
/auth/password_check
```

Метод запроса: POST.

Параметры запроса:

Таблица 24 – Параметры тела запроса /auth/password_check

Наименование	Тип данных	Пример
password	string	*SecurePass123*

Пример тела запроса:

```
{
  "password": "SecurePass123"
}
```

Ответ:

Таблица 25 – Ответ /auth/password_check

Код	Описание	Связи
204	Пароль верный	нет

Схема данных ответа (Schema): Нет.

Возможные сообщения об ошибках:

Таблица 26 – Сообщения об ошибках /auth/password_check

Код	Значение
400	Неверные параметры запроса. Включает случай когда OTPInvalidError перехватывается и возвращается как 400: invalid_param_value — неверный код OTP или неверный credential.
401	Не авторизован
403	Доступ запрещён
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.2.2 История авторизаций

2.2.2.1 Список авторизаций пользователя

Запрос на получение истории авторизаций и выходов из сессий текущего пользователя.

Метод запроса: GET.

Запрос:

```
/history
```

Параметры запроса (Parameters):

Таблица 27 – Параметры history список авторизаций

Наименование	Тип данных	Пример
page	integer	1
page_size	integer	20

Ответ:

Таблица 28 – Ответ history список авторизаций

Код	Описание	Связи
200	История авторизаций	нет

Схема данных ответа (Schema): PaginatedAuthHistory.

Пример ответа:


```
{
  "offset": 0,
  "limit": 100,
  "total_count": 2,
  "items": [
    {
      "auth_history_id": "7fb8d9c0-bdcb-4141-9931-33fae97acb7d",
      "time_created": 1742860800000,
      "source": "DIRECT",
      "action": "LOGIN",
      "user_token_id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
```

```
          "client_user_token_id": null,
          "device": {
            "device_platform": "mac",
            "device_os": "Mac OS X",
            "device_version": "10.15.7",
            "app_device": "Chrome",
            "app_build": "146.0.0",
            "device_locale": "ru",
            "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS
X 10_15_7) AppleWebKit/537.36"
          },
          "client": null
        },
        {
          "auth_history_id":          "0c03d5a7-59e3-43a9-a2c2-
307d3e536c3f",
          "time_created": 1742860800000,
          "source": "OAUTH",
          "action": "LOGIN",
          "user_token_id": null,
          "client_user_token_id":      "b2c3d4e5-f6a7-8901-bcde-
f12345678901",
          "device": null,
          "client": {
            "client_id": "P000000080",
            "client_name": "POCA Контроль"
          }
        }
      ]
    }
```


Возможные сообщения об ошибках:

Таблица 29 – Ошибки history список авторизаций

Код	Значение
401	Не авторизован
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.2.2.2 Удалить всю историю авторизации пользователя

Описание запроса: удаляет всю историю авторизации пользователя.

Метод запроса: DELETE.

Запрос:

```
/history
```

Параметры запроса (Parameters): нет.

Ответ:

Таблица 30 – Ответ history удалить сессии

Код	Описание	Связи
204	История удалена	нет

Схема данных ответа (Schema): нет.

Возможные сообщения об ошибках:

Таблица 31 – Ошибки history удалить сессии

Код	Значение
401	Не авторизован
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.2.2.3 Удалить запись авторизации пользователя

Удаляет одну запись истории авторизаций текущего пользователя по идентификатору.

Метод запроса: DELETE.

Запрос:

```
/history/{auth_history_id}
```

Параметры запроса (Parameters):

Таблица 32 – Параметры history удалить запись

Наименование	Тип данных	Пример
auth_history_id	string ($uuid)	*7fb8d9c0-bdcb-4141-9931-33fae97acb7d*

Ответ:

Таблица 33 – Ответ history удалить запись

Код	Описание	Связи
204	Запись удалена	Нет

Схема данных ответа (Schema): нет.

Возможные сообщения об ошибках:

Таблица 34 – Ошибки history удалить запись

Код	Значение
401	Не авторизован
404	Объект не найден
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.3 Пользователь

2.3.1 Профиль пользователя

Запрос на получение профиля авторизованного пользователя. Эквивалентно выполнению запроса /users/{user_id} с user_id="me".

Метод запроса: GET.

Запрос:

```
/users/me
```

Параметры запроса (Parameters): нет.

Ответ:

Таблица 35 – Ответ users/me профиль пользователя

Код	Описание	Связи
200	Профиль пользователя	нет

Схема данных ответа (Schema): UserDetail.

Пример ответа:


```
{
  "time_created": 1742688000000,
  "time_updated": 1742788000000,
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "email": "user@example.com",
  "email_verified": true,
  "phone_number": "+79001234567",
  "phone_number_verified": false,
  "is_active": true,
  "otp_transport": "email",
  "first_name": "Иван",
  "last_name": null,
  "middle_name": null,
  "gender": "MALE",
  "birthdate": "1990-01-15",
  "timezone_id": 1,
  "region_id": 77,
  "address_actual": null,
  "email_candidate": null,
  "phone_number_candidate": null,
  "avatar": null,
  "avatar_large": null
}
```


Возможные сообщения об ошибках:

Таблица 36 – Ошибки users/me профиль пользователя

Код	Значение
401	Не авторизован
404	Объект не найден

429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.3.2 Профиль указанного пользователя

Запрос на получение профиля пользователя по UUID пользователя.

Метод запроса: GET.

Запрос:

```
/users/{user_id}
```

Параметры запроса (Parameters):

Таблица 37 – Параметры запроса /users/{user_id}

Наименование	Тип данных	Пример
user_id	String	*550e8400-e29b-41d4-a716-446655440000*

Ответ:

Таблица 38 – Ответ по запросу /users/{user_id}

Код	Описание	Связи
200	Профиль пользователя	нет

Схема данных ответа (Schema): UserDetail.

Пример ответа:

```
{
  "time_created": 1742688000000,
  "time_updated": 1742688000000,
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "email": "user@example.com",
  "email_verified": true,
  "phone_number": "+79001234567",
  "phone_number_verified": false,
  "is_active": true,
  "otp_transport": "email",
  "first_name": "Иван",
```

```
    "last_name": null,
    "middle_name": null,
    "gender": "MALE",
    "birthdate": "1990-01-15",
    "timezone_id": 1,
    "region_id": 77,
    "address_actual": null,
    "email_candidate": null,
    "phone_number_candidate": null,
    "avatar": null,
    "avatar_large": null
}
```


Возможные сообщения об ошибках:

Таблица 39 – Ошибки запроса /users/{user_id}

Код	Значение
401	Не авторизован
404	Объект не найден
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.3.3 Обновление профиля

Запрос на частичное обновление профиля пользователя по UUID пользователя

Универсальный метод для профиля. Поведение зависит от переданных полей:

1) обновить базовые поля — без code: first_name, last_name, middle_name, gender, birthdate, address_actual, timezone_id, region_id;
2) установить кандидата на email — без code: email_candidate — сохраняется как кандидат, требует дальнейшего подтверждения;
3) установить кандидата на телефон — без code: phone_number_candidate — сохраняется как кандидат;
4) сменить email — требует code (OTP intent=5, SmsChangeEmail): email (= значение email_candidate) + code. Требует подтверждённый телефон;

5) сменить телефон — требует code (OTP intent=4, EmailChangePhone): phone_number (= значение phone_number_candidate) + code. Требует подтверждённый email;

6) сменить пароль — требует code (OTP intent=3, PasswordChange): password (новый) + old_password + code;

7) верифицировать email — требует code (OTP intent=6, EmailVerify): email_verified: true + code;

8) верифицировать телефон — требует code (OTP intent=7, SmsVerify): phone_number_verified: true + code;

9) сменить OTP-транспорт — требует code (OTP intent=8, TransportChange): otp_transport + code. Требует подтверждённый соответствующий контакт.

Метод запроса: PATCH.

Запрос:

```
/users/{user_id}
```

Параметры запроса (Parameters):

Таблица 40 – Параметры запроса users обновление профиля

Наименование	Тип данных	Пример
user_id	string	*550e8400-e29b-41d4-a716-446655440000*

Схема данных запроса (Schema):

Таблица 41 – Параметры тела запроса users обновление профиля

Наименование	Тип данных	Пример
first_name	string	*Иван*
last_name	string, nullable: true	*Иванов*
middle_name	string	*Иванович*
gender	string, nullable: true Enum: [MALE, FEMALE]	*MALE*
birthdate	integer, nullable: true	*1990-01-15*
address_actual	string, nullable: true	*null*

timezone_id	integer, nullable: true	*1*
region_id	integer, nullable: true	*77*
email	string	*user@example.com*
email_verified	boolean	*true*
phone_number	string, nullable: true	*+79001234567*
phone_number_verified	boolean	*false*
email_candidate	string, nullable: true	*null*
phone_number_candidate	string, nullable: true	*null*
password	string	*NewSecurePass456*
old_password	string	*OldSecurePass123*
otp_transport	string	*sms*
code	string	*123456*

Пример тела запроса:

1) обновить базовые поля:

```
{
  "first_name": "Иван",
  "last_name": "Иванов",
  "gender": "MALE",
  "birthdate": "1990-01-15",
  "timezone_id": 1,
  "region_id": 77
}
```

2) установить кандидата на email:

```
{
  "email_candidate": "newemail@example.com"
}
```

3) установить кандидата на телефон:

```
{
  "phone_number_candidate": "+79009876543"
```

```
}
```

4) сменить email (OTP intent=5):

```
{
  "email": "newemail@example.com",
  "code": "123456"
}
```

5) сменить телефон (OTP intent=4):

```
{
  "phone_number": "+79009876543",
  "code": "123456"
}
```

6) сменить пароль (OTP intent=3):

```
{
  "old_password": "OldSecurePass123",
  "password": "NewSecurePass456",
  "code": "123456"
}
```

7) верифицировать email (OTP intent=6):

```
{
  "email_verified": true,
  "code": "123456"
}
```

8) верифицировать телефон (OTP intent=7):

```
{
  "phone_number_verified": true,
  "code": "123456"
}
```

9) сменить OTP-транспорт (OTP intent=8):

```
{
  "otp_transport": "sms",
  "code": "123456"
}
```

Ответ:

Таблица 42 – Ответ users обновление профиля

Код	Описание	Связи
200	Обновлённый профиль	нет

Схема данных ответа (Schema): User.

Пример ответа:


```
{
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "email": "user@example.com",
  "email_verified": true,
  "phone_number": "+79001234567",
  "phone_number_verified": false,
  "is_active": true,
  "otp_transport": "email",
  "first_name": "Иван",
  "last_name": "Иванов",
  "middle_name": "Иванович",
  "gender": "MALE",
  "birthdate": "1990-01-15",
  "timezone_id": 1,
  "region_id": 77,
  "address_actual": null,
  "email_candidate": null,
  "phone_number_candidate": null
}
```


Возможные сообщения об ошибках:

Таблица 43 – Ошибки users обновление профиля

Код	Значение
401	Не авторизован
403	Ошибка доступа. Возможные error коды: - email_taken (1004) — email уже занят;

	- phone_number_taken (1005) — телефон уже занят; - credential_not_verified (1003) — контакт не подтверждён (нужен для данной операции); - otp_invalid (1021) — неверный OTP-код; - otp_expired (1022) — OTP-код истёк; - otp_attempts_exceeded (1023) — превышен лимит попыток OTP; - auth_forbidden — email/телефон должен быть подтверждён для данной операции.
404	Объект не найден
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.3.4 Удаление аккаунта

Запрос на удаление аккаунта пользователя по UUID пользователя (с проверкой пароля и OTP).

Метод запроса: DELETE.

Запрос:

```
/users/{user_id}
```

Параметры запроса (Parameters):

Таблица 44 – Параметры запроса users удаление аккаунта

Наименование	Тип данных	Пример
user_id	string	*550e8400-e29b-41d4-a716-446655440000*

Схема данных запроса (Schema):

Таблица 45 – Параметры тела запроса users удаление аккаунта

Наименование	Тип данных	Пример
password	string	*NewSecurePass456*
code	string	*123456*

Ответ:

Таблица 46 – Ответ users удаление аккаунта

Код	Описание	Связи
200	Пользователь удалён	Нет

Схема данных ответа (Schema): Нет.

Возможные сообщения об ошибках:

Таблица 47 – Ошибки users удаление аккаунта

Код	Значение
401	Не авторизован
403	Ошибка доступа. Возможные error коды: - auth_forbidden — неверный пароль; - otp_invalid (1021) — неверный OTP-код; - otp_expired (1022) — OTP-код истёк; - otp_attempts_exceeded (1023) — превышен лимит попыток OTP.
404	Объект не найден
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.3.5 Получить текущий аватар

GET /api/users/{user_id}/avatar — получить текущий аватар.

Запрос на получение аватара пользователя по UUID пользователя.

Метод запроса: DELETE.

Запрос:

```
/users/{user_id}/avatar
```

Параметры запроса (Parameters):

Таблица 48 – Параметры запроса users получение аватара

Наименование	Тип данных	Пример
user_id	string	*550e8400-e29b-41d4-a716-446655440000*

Схема данных запроса (Schema): Нет.

Ответ:

Таблица 49 – Ответ users получение аватара

Код	Описание	Связи
200	Аватар пользователя	Нет

Схема данных ответа (Schema): UserAvatar.

Пример ответа:

```
{
  "image_id": 42,
  "user_id": "550e8400-e29b-41d4-a716-446655440000"
}
```

Возможные сообщения об ошибках:

Таблица 50 – Ошибки users получение аватара

Код	Значение
401	Не авторизован
404	Объект не найден
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.3.6 Загрузить аватар

Запрос на загрузку/обновление аватара пользователя по UUID пользователя.

Запрос:

```
/users/{user_id}/avatar
```

Метод запроса: POST.

Параметры запроса:

Таблица 51 – Параметры запроса /users/{user_id}/avatar

Наименование	Тип данных	Пример
user_id	string	*550e8400-e29b-41d4-a716-446655440000*

Таблица 52 – Параметры тела запроса /users/{user_id}/avatar

Наименование	Тип данных	Пример
file	string($binary)	

Ответ:

Таблица 53 – Ответ /users/{user_id}/avatar

Код	Описание	Связи
200	Аватар загружен	Нет

Схема данных ответа (Schema): UserAvatar.

Пример ответа:

```
{
  "image_id": 43,
  "user_id": "550e8400-e29b-41d4-a716-446655440000"
}
```

Возможные сообщения об ошибках:

Таблица 54 – Сообщения об ошибках /users/{user_id}/avatar

Код	Значение
400	Неверные параметры запроса.
401	Не авторизован
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.3.7 Удалить аватар

Запрос на удаление аватара пользователя по UUID пользователя (с проверкой пароля и OTP).

Метод запроса: DELETE.

Запрос:

```
/users/{user_id}/avatar
```

Параметры запроса (Parameters):

Таблица 55 – Параметры запроса users удаление аватара

Наименование	Тип данных	Пример
user_id	string	*550e8400-e29b-41d4-a716-446655440000*

Схема данных запроса (Schema): Нет.

Ответ:

Таблица 56 – Ответ users удаление аватара

Код	Описание	Связи
204	Аватар удалён	Нет

Схема данных ответа (Schema): Нет.

Возможные сообщения об ошибках:

Таблица 57 – Ошибки users удаление аватара

Код	Значение
401	Не авторизован
403	Ошибка доступа. Возможные error коды: - auth_forbidden — неверный пароль; - otp_invalid (1021) — неверный OTP-код; - otp_expired (1022) — OTP-код истёк; - otp_attempts_exceeded (1023) — превышен лимит попыток OTP.
404	Объект не найден
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.4 Управление сессиями и устройствами

2.4.1 Список сессий текущего пользователя

Описание запроса: возвращает список активных сессий (токенов) авторизованного пользователя с информацией об устройстве.

Метод запроса: GET.

Запрос:

```
/sessions
```

Параметры запроса (Parameters): нет.

Ответ:

Таблица 58 – Ответ sessions список сессий

Код	Описание	Связи
200	Список активных сессий пользователя	нет

Схема данных ответа (Schema): PaginatedSessions.

Пример ответа:


```
    {
      "offset": 0,
      "limit": 100,
      "total_count": 2,
      "items": [
        {
          "user_token_id":             "a1b2c3d4-e5f6-7890-abcd-
ef1234567890",
          "is_current": false,
          "time_created": 1742688000000,
          "time_updated": 1742788000000,
          "user_id": "53ecec94-ae83-4d40-a39b-970a95018110",
          "expires_at": "2026-04-18T12:37:09Z",
          "status": "ACTIVE",
          "client": null,
          "device": {
            "device_platform": "mac",
            "device_os": "Mac OS X",
            "device_version": "10.15.7",
            "app_device": "Chrome",
            "app_build": "146.0.0",
            "device_locale": "ru",
            "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS
X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/146.0.0.0
Safari/537.36"
          }
```

```
        },
        {
          "user_token_id":            "b2c3d4e5-f6a7-8901-bcde-
f12345678901",
          "is_current": true,
          "time_created": 1742688000000,
          "time_updated": 1742788000000,
          "user_id": "53ecec94-ae83-4d40-a39b-970a95018110",
          "expires_at": "2026-04-18T12:37:09Z",
          "status": "ACTIVE",
          "device": null,
          "client": {
            "client_id": "P000000080",
            "client_name": "РОСА Контроль",
            "scope": [
              "email",
              "name",
              "openid"
            ]
          }
        }
      ]
    }
```


Возможные сообщения об ошибках:

Таблица 59 – Ошибки sessions список сессий

Код	Значение
401	Не авторизован
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.4.2 Удалить все сессии текущего пользователя

Описание запроса: удаляет все активные сессии (токены) авторизованного пользователя. Выполнение запроса равнозначно выходу со всех устройств.

Метод запроса: DELETE.

Запрос:

```
/sessions
```

Параметры запроса (Parameters): нет.

Ответ:

Таблица 60 – Ответ sessions удалить сессии

Код	Описание	Связи
204	Все сессии удалены	нет

Схема данных ответа (Schema): нет.

Возможные сообщения об ошибках:

Таблица 61 – Ошибки sessions удалить сессии

Код	Значение
401	Не авторизован
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.4.3 Удалить указанную сессию текущего пользователя

Удаляет указанную сессию (токен) текущего пользователя. Пользователь может завершить только свои сессии.

Метод запроса: DELETE.

Запрос:

```
/sessions/{user_token_id}
```

Параметры запроса (Parameters):

Таблица 62 – Параметры sessions удалить сессию

Наименование	Тип данных	Описание
user_token_id	string ($uuid)	*Обязательный*

Ответ:

Таблица 63 – Ответ sessions удалить сессию

Код	Описание	Связи
204	Сессия удалена	Нет

Схема данных ответа (Schema): нет.

Возможные сообщения об ошибках:

Таблица 64 – Ошибки sessions удалить сессию

Код	Значение
401	Не авторизован
404	Объект не найден
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.4.4 Список устройств текущего пользователя

Запрос на получение списка всех активных устройств, с которых получен доступ к аккаунту текущего пользователя.

Метод запроса: GET.

Запрос:

```
/devices
```

Параметры запроса (Parameters): нет.

Ответ:

Таблица 65 – Ответ devices список устройств

Код	Описание	Связи
200	Список активных устройств	нет

Схема данных ответа (Schema): PaginatedDevices.

Пример ответа:

```
{
  "items": [
    {
      "user_token_id":"a1b2c3d4-e5f6-7890-abcd-
ef1234567890",
```

```
          "is_current": false,
          "time_created": 1742688000000,
          "time_updated": 1742788000000,
          "user_id": "550e8400-e29b-41d4-a716-446655440000",
          "expires_at": "2026-04-04T17:04:45Z",
          "status": "ACTIVE",
          "client": null,
          "device": {
            "device_platform": "mac",
            "device_os": "Mac OS X",
            "device_version": "10.15.7",
            "app_device": "Chrome",
            "app_build": "146.0.0",
            "device_locale": "ru",
            "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS
X 10_15_7) AppleWebKit/537.36"
          }
        }
      ],
      "offset": 0,
      "limit": 100,
      "total_count": 2
    }
```


Возможные сообщения об ошибках:

Таблица 66 – Ошибки devices список устройств

Код	Значение
401	Не авторизован
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.4.5 Удалить все устройства пользователя

Запрос на завершение всех активных сессий (токенов) на всех устройствах авторизованного пользователя. Выполнение запроса равнозначно выходу со всех устройств.

Метод запроса: DELETE.

Запрос:

```
/devices
```

Параметры запроса (Parameters): нет.

Ответ:

Таблица 67 – Ответ devices удалить все устройства

Код	Описание	Связи
204	Все сессии завершены	нет

Схема данных ответа (Schema): нет.

Возможные сообщения об ошибках:

Таблица 68 – Ошибки devices удалить все устройства

Код	Значение
401	Не авторизован
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.4.6 Информация об указанном устройстве

Запрос на получение информации об указанном устройстве авторизованного пользователя.

Метод запроса: GET.

Запрос:

```
/devices/{user_token_id}
```

Параметры запроса (Parameters):

Таблица 69 – Параметры devices информация об устройстве

Наименование	Тип данных	Описание
user_token_id	string ($uuid)	*UUID устройства, обязательный*

Ответ:

Таблица 70 – Ответ devices информация об устройстве

Код	Описание	Связи
200	Информация об устройстве	Нет

Схема данных ответа (Schema): UserToken.

Пример ответа:


```
{
  "user_token_id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "is_current": false,
  "time_created": 1742688000000,
  "time_updated": 1742788000000,
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "expires_at": "2026-04-04T17:04:45Z",
  "status": "ACTIVE",
  "client": null,
  "device": {
    "device_platform": "mac",
    "device_os": "Mac OS X",
    "device_version": "10.15.7",
    "app_device": "Chrome",
    "app_build": "146.0.0",
    "device_locale": "ru",
    "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X
10_15_7) AppleWebKit/537.36"
  }
}
```


Возможные сообщения об ошибках:

Таблица 71 – Ошибки devices информация об устройстве

Код	Значение
401	Не авторизован
404	Объект не найден
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.4.7 Удалить указанное устройство

Завершает сессии (токены) текущего пользователя на указанном устройстве.

Метод запроса: DELETE.

Запрос:

```
/devices/{user_token_id}
```

Параметры запроса (Parameters):

Таблица 72 – Параметры devices удалить указанное устройство

Наименование	Тип данных	Описание
user_token_id	string ($uuid)	*UUID устройства, обязательный*

Ответ:

Таблица 73 – Ответ devices удалить указанное устройство

Код	Описание	Связи
204	Сессии завершены	Нет

Схема данных ответа (Schema): нет.

Возможные сообщения об ошибках:

Таблица 74 – Ошибки devices удалить указанное устройство

Код	Значение
401	Не авторизован
404	Объект не найден
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.5 Документы пользователя

2.5.1 Список документов пользователя

Запрос на получение списка документов пользователя.

Метод запроса: GET.

Запрос:

```
/documents
```

Параметры запроса (Parameters):

Таблица 75 – Параметры запроса documents список документов

Наименование	Тип данных	Пример
page	integer	*1*
page_size	integer	*20*

Ответ:

Таблица 76 – Ответ documents список документов

Код	Описание	Связи
200	Список документов пользователя	нет

Схема данных ответа (Schema): PaginatedDocuments.

Пример ответа:


```
{
  "offset": 0,
  "limit": 100,
  "total_count": 90,
  "items": [
    {
      "document_id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
      "user_id": "550e8400-e29b-41d4-a716-446655440000",
      "document_type": "passport",
      "document_number": "4500123456",
      "data": {
        "series": "45 00",
        "issued_by": "ОВД района Арбат",
```

```
            "issue_date": "2010-05-20"
          }
        }
      ]
    }
```

Возможные сообщения об ошибках:

Таблица 77 – Ошибки documents список документов

Код	Значение
401	Не авторизован
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.5.2 Создание документа

Запрос на создание/замену документа(ов).

Метод запроса: POST.

Запрос:

```
/documents
```

Параметры запроса (Parameters):

Таблица 78 – Параметры documents создать документ(ы)

Наименование	Тип данных	Пример
oneOf	object Document	см. схему Document

Пример запроса на один документ:

```
{
  "document_type": "passport",
  "document_number": "4500123456",
  "data": {
    "series": "45 00",
    "issued_by": "ОВД района Арбат",
    "issue_date": "2010-05-20"
  }
```

```
}
```

Пример запроса на нескольких документов:

```
{
  "documents": [
    {
      "document_type": "passport",
      "document_number": "4500123456",
      "data": {}
    },
    {
      "document_type": "snils",
      "document_number": "123-456-789 00",
      "data": {}
    }
  ]
}
```

Ответ:

Таблица 79 – Ответ documents создание документов

Код	Описание	Связи
200	Документ создан/обновлён	нет

Схема данных ответа (Schema): Document.

Пример ответа:

```
{
  "document_id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "document_type": "passport",
  "document_number": "4500123456",
  "data": {
    "series": "45 00",
    "issued_by": "ОВД района Арбат",
    "issue_date": "2010-05-20"
  }
```

```
}
```

Возможные сообщения об ошибках:

Таблица 80 – Ошибки documents создание документов

Код	Значение
401	Не авторизован
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.5.3 Получить документ

Запрос на получение документа.

Метод запроса: GET.

Запрос:

```
/documents/{document_id}
```

Параметры запроса (Parameters):

Таблица 81 – Параметры запроса documents получить документ

Наименование	Тип данных	Пример
document_id	string ($uuid)	*a1b2c3d4-e5f6-7890-abcd-ef1234567890*

Ответ:

Таблица 82 – Ответ по запросу documents получить документ

Код	Описание	Связи
200	Документ	нет

Схема данных ответа (Schema): Document.

Пример ответа:

```
{
  "document_id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "document_type": "passport",
  "document_number": "4500123456",
  "data": {
```

```
        "series": "45 00",
        "issued_by": "ОВД района Арбат",
        "issue_date": "2010-05-20"
      }
    }
```

Возможные сообщения об ошибках:

Таблица 83 – Ошибки запроса documents получить документ

Код	Значение
401	Не авторизован
404	Объект не найден
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.5.4 Обновить документ

Запрос на частичное обновление документа.

Метод запроса: PATCH.

Запрос:

```
/documents/{document_id}
```

Параметры запроса (Parameters):

Таблица 84 – Параметры запроса documents обновить документ

Наименование	Тип данных	Пример
document_id	string ($uuid)	*a1b2c3d4-e5f6-7890-abcd-ef1234567890*

Таблица 85 – Параметры тела запроса documents обновить документ

Наименование	Тип данных	Пример
document_number	string	4500654321
data	array	*{ "series": "45 00",* *"issued_by": "ОВД района Пресненский",* *"issue_date": "2015-03-10"}*

Пример тела запроса:

```
{
  "document_number": "4500654321",
  "data": {
    "series": "45 00",
    "issued_by": "ОВД района Пресненский",
    "issue_date": "2015-03-10"
  }
}
```

Ответ:

Таблица 86 – Ответ по запросу documents обновить документ

Код	Описание	Связи
200	Обновлённый документ	нет

Схема данных ответа (Schema): Document.

Пример ответа:

```
{
  "document_id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "document_type": "passport",
  "document_number": "4500123456",
  "data": {
    "series": "45 00",
    "issued_by": "ОВД района Арбат",
    "issue_date": "2010-05-20"
  }
}
```

Возможные сообщения об ошибках:

Таблица 87 – Ошибки запроса documents обновить документ

Код	Значение
400	Неверные параметры запроса
401	Не авторизован

404	Объект не найден
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.5.5 Удалить документ

Запрос на удаление документа.

Метод запроса: DELETE.

Запрос:

```
/documents/{document_id}
```

Параметры запроса (Parameters):

Таблица 88 – Параметры запроса documents удалить документ

Наименование	Тип данных	Пример
document_id	string ($uuid)	*a1b2c3d4-e5f6-7890-abcd-ef1234567890*

Ответ:

Таблица 89 – Ответ по запросу documents удалить документ

Код	Описание	Связи
204	Документ удалён	нет

Схема данных ответа (Schema): Нет.

Возможные сообщения об ошибках:

Таблица 90 – Ошибки запроса documents удалить документ

Код	Значение
400	Неверные параметры запроса
401	Не авторизован
404	Объект не найден
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.6 Партнёры пользователя

2.6.1 Список партнёров

Запрос на получение списка партнёров пользователя.

Метод запроса: GET.

Запрос:

```
/partners
```

Параметры запроса (Parameters):

Таблица 91 – Параметры запроса partners список партнёров

Наименование	Тип данных	Пример
page	integer	*1*
page_size	integer	*20*

Ответ:

Таблица 92 – Ответ partners список партнёров

Код	Описание	Связи
200	Список партнёров пользователя	нет

Схема данных ответа (Schema): PaginatedPartners.

Пример ответа:

```
    {
      "offset": 0,
      "limit": 100,
      "total_count": 90,
      "items": [
        {
          "partner_id": "b2c3d4e5-f6a7-8901-bcde-f12345678901",
          "owner_user_id":             "550e8400-e29b-41d4-a716-
446655440000",
          "partner_type": "individual",
          "name": "ООО Ромашка",
          "data": {
            "inn": "7700000000"
```

```
        }
      }
    ]
  }
```

Возможные сообщения об ошибках:

Таблица 93 – Ошибки partners список партнёров

Код	Значение
401	Не авторизован
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.6.2 Создать партнёра

Запрос на создание партнёра.

Метод запроса: POST.

Запрос:

```
/partners
```

Параметры запроса (Parameters):

Таблица 94 – Параметры запроса partners создать партнёра

Наименование	Тип данных	Пример
partner_type	string	*ORGANIZATION*
name	string	*ООО Ромашка*
data	array	*OrderedMap { "inn": "7700000000", "ogrn": "1027700000000" }*

Пример запроса:

```
{
  "partner_type": "ORGANIZATION",
  "name": "ООО Ромашка",
  "data": {
    "inn": "7700000000",
    "ogrn": "1027700000000"
  }
```

```
}
```

Ответ:

Таблица 95 – Ответ по запросу partners создать партнёра

Код	Описание	Связи
200	Партнёр создан	нет

Схема данных ответа (Schema): Partner.

Пример ответа:

```
{
  "partner_id": "b2c3d4e5-f6a7-8901-bcde-f12345678901",
  "owner_user_id": "550e8400-e29b-41d4-a716-446655440000",
  "partner_type": "individual",
  "name": "ООО Ромашка",
  "data": {
    "inn": "7700000000",
    "ogrn": "1027700000000"
  }
}
```

Возможные сообщения об ошибках:

Таблица 96 – Ошибки по запросу partners создать партнёра

Код	Значение
400	Неверные параметры запроса
401	Не авторизован
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.6.3 Получить партнёра

Запрос на получение партнёра.

Метод запроса: GET.

Запрос:

```
/partners/{partner_id}
```

Параметры запроса (Parameters):

Таблица 97 – Параметры запроса partners получить партнёра

Наименование	Тип данных	Пример
partner_id	string ($uuid)	*b2c3d4e5-f6a7-8901-bcde-f12345678901*

Ответ:

Таблица 98 – Ответ по запросу partners получить партнёра

Код	Описание	Связи
200	Партнёр	нет

Схема данных ответа (Schema): Partner.

Пример ответа:

```
{
  "partner_id": "b2c3d4e5-f6a7-8901-bcde-f12345678901",
  "owner_user_id": "550e8400-e29b-41d4-a716-446655440000",
  "partner_type": "individual",
  "name": "ООО Ромашка",
  "data": {
    "inn": "7700000000",
    "ogrn": "1027700000000"
  }
}
```

Возможные сообщения об ошибках:

Таблица 99 – Ошибки запроса partners получить партнёра

Код	Значение
401	Не авторизован
404	Объект не найден
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.6.4 Обновить партнёра

Запрос на обновление партнёра.

Метод запроса: PATCH.

Запрос:

```
/partners/{partner_id}
```

Параметры запроса (Parameters):

Таблица 100 – Параметры запроса partners обновить партнёра

Наименование	Тип данных	Пример
partner_id	string ($uuid)	*b2c3d4e5-f6a7-8901-bcde-f12345678901*

Таблица 101 – Параметры тела запроса partners обновить партнёра

Наименование	Тип данных	Пример
name	string	ООО Ромашка обновлённая
data	array	*{"inn": "7700000000"}*

Пример тела запроса:

```
{
  "name": "ООО Ромашка обновлённая",
  "data": {
    "inn": "7700000000"
  }
}
```

Ответ:

Таблица 102 – Ответ по запросу partners обновить партнёра

Код	Описание	Связи
200	Обновлённый партнёр	нет

Схема данных ответа (Schema): Partner.

Пример ответа:

```
{
  "partner_id": "b2c3d4e5-f6a7-8901-bcde-f12345678901",
  "owner_user_id": "550e8400-e29b-41d4-a716-446655440000",
```

```
    "partner_type": "individual",
    "name": "ООО Ромашка обновлённая",
    "data": {
      "inn": "7700000000"
    }
  }
```

Возможные сообщения об ошибках:

Таблица 103 – Ошибки запроса partners обновить партнёра

Код	Значение
400	Неверные параметры запроса
401	Не авторизован
404	Объект не найден
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.6.5 Удалить партнёра

Запрос на удаление партнёра.

Метод запроса: DELETE.

Запрос:

```
/partners/{partner_id}
```

Параметры запроса (Parameters):

Таблица 104 – Параметры запроса partners удалить партнёра

Наименование	Тип данных	Пример
partner_id	string ($uuid)	*b2c3d4e5-f6a7-8901-bcde-f12345678901*

Ответ:

Таблица 105 – Ответ по запросу partners удалить партнёра

Код	Описание	Связи
204	Партнёр удалён	нет

Схема данных ответа (Schema): Нет.

Возможные сообщения об ошибках:

Таблица 106 – Ошибки запроса partners удалить партнёра

Код	Значение
400	Неверные параметры запроса
401	Не авторизован
404	Объект не найден
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.7 OAuth-клиенты пользователя

2.7.1 Список OAuth-клиентов

Запрос на получение списка OAuth-клиентов пользователя.

Метод запроса: GET.

Запрос:

```
/clients
```

Параметры запроса (Parameters):

Таблица 107 – Параметры запроса clients список OAuth-клиентов

Наименование	Тип данных	Пример
page	integer	*1*
page_size	integer	*20*

Ответ:

Таблица 108 – Ответ clients список OAuth-клиентов

Код	Описание	Связи
200	Список OAuth-клиентов пользователя	нет

Схема данных ответа (Schema): PaginatedClients.

Пример ответа:

```
{
  "offset": 0,
```

```
      "limit": 100,
      "total_count": 90,
      "items": [
        {
          "client_id": 1,
          "user_id": "550e8400-e29b-41d4-a716-446655440000",
          "name": "My App",
          "platforms": [
            "WEB"
          ],
          "redirect_uri":
"https://myapp.example.com/oauth/callback"
        }
      ]
    }
```


Возможные сообщения об ошибках:

Таблица 109 – Ошибки clients список OAuth-клиентов

Код	Значение
401	Не авторизован
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.7.2 Создать OAuth-клиента

Запрос на создание OAuth-клиента.

Метод запроса: POST.

Запрос:

```
/clients
```

Параметры запроса (Parameters):

Таблица 110 – Параметры тела запроса clients создать OAuth-клиента

Наименование	Тип данных	Пример
name	string	*My App*

redirect_uri	string	*https://myapp.example.com/oauth/callback:*
platforms	array	*OrderedMap { "type": "array", "items": OrderedMap { "type": "string", "enum": List ["WEB", "MOBILE"] } }*
scope	array	*OrderedMap { "type": "array", "items": OrderedMap { "type": "string" } }*

Схема данных запроса (Schema): Нет.

Пример тела запроса:

```
{
  "name": "My App",
  "redirect_uri":
"https://myapp.example.com/oauth/callback",
  "platforms": [
    "WEB",
    "MOBILE"
  ],
  "scope": [
    "openid",
    "email",
    "profile"
  ]
}
```

Ответ:

Таблица 111 – Ответ по запросу clients создать OAuth-клиента

Код	Описание	Связи
200	OAuth-клиент создан	нет

Схема данных ответа (Schema): Client.

Пример ответа:

```
{
  "client_id": 1,
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "name": "My App",
  "platforms": [
    "web",
```

```
      "ios"
    ],
    "redirect_uri": "https://myapp.example.com/oauth/callback"
  }
```

Возможные сообщения об ошибках:

Таблица 112 – Ошибки по запросу clients создать OAuth-клиента

Код	Значение
400	Неверные параметры запроса
401	Не авторизован
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.7.3 Получить OAuth-клиента

Запрос на получение OAuth-клиента.

Метод запроса: GET.

Запрос:

```
/clients/{client_id}
```

Параметры запроса (Parameters):

Таблица 113 – Параметры запроса clients получить OAuth-клиента

Наименование	Тип данных	Пример
client_id	string ($uuid)	*b2c3d4e5-f6a7-8901-bcde-f12345678901*

Ответ:

Таблица 114 – Ответ по запросу clients получить OAuth-клиента

Код	Описание	Связи
200	OAuth-клиент	нет

Схема данных ответа (Schema): Client.

Пример ответа:

```
{
  "client_id": 1,
```

```
    "user_id": "550e8400-e29b-41d4-a716-446655440000",
    "name": "My App",
    "platforms": [
      "web",
      "ios"
    ],
    "redirect_uri": "https://myapp.example.com/oauth/callback"
  }
```

Возможные сообщения об ошибках:

Таблица 115 – Ошибки запроса clients получить OAuth-клиента

Код	Значение
401	Не авторизован
404	Объект не найден
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.7.4 Обновить OAuth-клиента

PATCH /api/clients/{client_id} — обновить OAuth-клиента.

Запрос на обновление OAuth-клиента.

Метод запроса: PATCH.

Запрос:

```
/clients/{client_id}
```

Параметры запроса (Parameters):

Таблица 116 – Параметры тела запроса clients обновить OAuth-клиента

Наименование	Тип данных	Пример
name	string	*My App*
redirect_uri	string	*https://myapp.example.com/oauth/callback:*
platforms	array	*OrderedMap { "type": "array", "items": OrderedMap { "type": "string", "enum": List ["WEB", "MOBILE"] } }*
scope	array	*OrderedMap { "type": "array", "items": OrderedMap { "type": "string" } }*

Схема данных запроса (Schema): Нет.

Пример тела запроса:

```
{
  "name": "My App v2",
  "platforms": [
    "WEB",
    "MOBILE"
  ]
}
```

Ответ:

Таблица 117 – Ответ по запросу clients обновить OAuth-клиента

Код	Описание	Связи
200	Обновлённый OAuth-клиент	нет

Схема данных ответа (Schema): Client.

Пример ответа:

```
{
  "client_id": 1,
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "name": "My App v2",
  "platforms": [
    "web",
    "ios",
    "android"
  ],
  "redirect_uri": "https://myapp.example.com/oauth/callback"
}
```

Возможные сообщения об ошибках:

Таблица 118 – Ошибки по запросу clients обновить OAuth-клиента

Код	Значение
400	Неверные параметры запроса
401	Не авторизован

404	Объект не найден
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.7.5 Удалить OAuth-клиента

Запрос на удаление OAuth-клиента.

Метод запроса: DELETE.

Запрос:

```
/clients/{client_id}
```

Параметры запроса (Parameters):

Таблица 119 – Параметры запроса clients удалить OAuth-клиента

Наименование	Тип данных	Пример
client_id	string ($uuid)	*b2c3d4e5-f6a7-8901-bcde-f12345678901*

Ответ:

Таблица 120 – Ответ по запросу clients удалить OAuth-клиента

Код	Описание	Связи
204	OAuth-клиент удалён	нет

Схема данных ответа (Schema): Нет.

Возможные сообщения об ошибках:

Таблица 121 – Ошибки запроса clients удалить OAuth-клиента

Код	Значение
401	Не авторизован
404	Объект не найден
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.7.6 Авторизация (OAuth)

2.7.6.1 OAuth-авторизация

Запрос на информацию о клиенте для отображения экрана согласия.

Запрос:

```
/oauth/authorize
```

Метод запроса: GET.

Параметры запроса:

Таблица 122 – Параметры запроса oauth/authorize

Наименование	Тип данных	Пример
scope	string	*openid email profile*
response_type	string	*code*
client_id	string	*1*
redirect_uri	string	*https://myapp.example.com/oauth/callback*

Ответ:

Таблица 123 – Ответ на запрос oauth /authorize

Код	Описание	Связи
200	Информация о клиенте для экрана согласия.	нет

Схема данных ответа (Schema): OAuthAuthorizeGetResponse.

Пример ответа:

```
{
  "client": {
    "time_created": 1742688000000,
    "time_updated": 1742788000000,
    "client_id": "P000000070",
    "user_id": "8e7537fc-7a37-4b80-b8b5-3178dde00e3b",
    "name": "РОСА Диск",
    "redirect_uri": "http://localhost"
  }
}
```

Возможные сообщения об ошибках:

Таблица 124 – Сообщения об ошибках auth/authorize

Код	Значение
400	Неверные параметры запроса
401	Не авторизован
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.7.6.2 Подтверждение OAuth-авторизации

Запрос на подтверждение согласия пользователя. Подтверждение или отклонение запроса авторизации пользователем.

Запрос:

```
/oauth/authorize
```

Метод запроса: POST.

Параметры запроса:

Таблица 125 – Параметры тела запроса oauth /authorize

Наименование	Тип данных	Пример
scope	string	*openid email profile*
response_type	string	*code*
client_id	string	*1*
redirect_uri	string	*https://myapp.example.com/oauth/callback*
accept	boolean	*true*

Пример тела запроса:

```
{
  "scope": "openid email profile",
  "response_type": "code",
  "client_id": "1",
  "redirect_uri":
"https://myapp.example.com/oauth/callback",
  "accept": true
}
```

Ответ:

Таблица 126 – Ответ на запрос oauth /authorize

Код	Описание	Связи
200	URL для редиректа с кодом авторизации	нет

Схема данных ответа (Schema): OAuthAuthorizePostResponse.

Пример ответа:

```
{
  "redirect_url":
"https://myapp.example.com/oauth/callback?code=abc123xyz"
}
```

Возможные сообщения об ошибках:

Таблица 127 – Сообщения об ошибках oauth /authorize

Код	Значение
400	Неверные параметры запроса
401	Не авторизован
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.7.6.3 Получение токена клиента

Запрос на получение токена клиента (authorization_code или refresh_token).

Запрос:

```
/oauth/client/token
```

Метод запроса: POST.

Параметры запроса:

Таблица 128 – Параметры тела запроса oauth/client/token

Наименование	Тип данных	Пример
grant_type	string [authorization_code, refresh_token]	authorization_code

client_id	string	70
client_secret	string	*edce4425460bbbb0a3b7c1f529e3c5993839c75490a54cb6b1a45eb0c2fd8e7f*
code	string	*rJPqibAWGtUDYJdkOfbskr*
redirect_uri	string	*https://myapp.example.com/oauth/callback*
refresh_token	string	*true*

Пример тела запроса при обновлении токена (refresh_token):

```
{
  "grant_type": "refresh_token",
  "client_id": "70",
  "client_secret":
"edce4425460bbbb0a3b7c1f529e3c5993839c75490a54cb6b1a45eb0c2fd8e7
f",
  "refresh_token": "dGhpcyBpcyBhIHJlZnJlc2ggdG9rZW4..."
}
```

Пример тела запроса при обмене кода на токен (authorization_code):

```
{
  "grant_type": "authorization_code",
  "client_id": "70",
  "client_secret":
"edce4425460bbbb0a3b7c1f529e3c5993839c75490a54cb6b1a45eb0c2fd8e7
f",
  "code": "rJPqibAWGtUDYJdkOfbskr",
  "redirect_uri": "http://localhost"
}
```

Ответ:

Таблица 129 – Ответ на запрос oauth/client/token

Код	Описание	Связи
200	OAuth токен	нет

Схема данных ответа (Schema): OAuthTokenResponse.

Пример ответа:

```
{
```

```
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "token_type": "Bearer",
  "refresh_token": "dGhpcyBpcyBhIHJlZnJlc2ggdG9rZW4...",
  "expires_in": 3600,
  "scope": "openid email profile",
  "id_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

Возможные сообщения об ошибках:

Таблица 130 – Сообщения об ошибках oauth/client/token

Код	Значение
400	Неверные параметры запроса
401	Не авторизован
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.7.6.4 Данные пользователя (OIDC)

Запрос на информацию о пользователе (OIDC) по токену клиента. Требует размещения OAuth access_token из запроса /api/oauth/client/token в заголовке "Authorization: Bearer <token>".

Запрос:

```
/oauth/client/userinfo
```

Метод запроса: GET.

Параметры запроса: Нет.

Ответ:

Таблица 131 – Ответ на запрос oauth/client/userinfo

Код	Описание	Связи
200	OIDC userinfo.	нет

Схема данных ответа (Schema): OIDCUserinfo.

Пример ответа:

```
{
  "sub": "550e8400-e29b-41d4-a716-446655440000",
```

```
    "email": "user@example.com",
    "email_verified": true,
    "phone_number": "+79001234567",
    "phone_number_verified": false,
    "given_name": "Иван",
    "family_name": "Иванов",
    "birthdate": "1990-01-15",
    "avatar_large": null
}
```

Возможные сообщения об ошибках:

Таблица 132 – Сообщения об ошибках auth/authorize

Код	Значение
401	Не авторизован
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.8 Вспомогательные методы API

2.8.1 Справочники

2.8.1.1 Список часовых поясов

Возвращает список часовых поясов.

Метод запроса: GET.

Запрос:

```
/handbooks/timezones
```

Параметры запроса (Parameters):

Таблица 133 – Параметры timezones

Наименование	Тип данных	Описание
Offset	integer	*Значение по умолчанию: 0*
Limit	integer	*Значение по умолчанию: 100*

Ответ:

Таблица 134 – Ответ timezones

Код	Описание	Связи
200	Список часовых поясов	нет

Схема данных ответа (Schema): PaginatedTimezones.

Пример ответа:


```
{
  "offset": 0,
  "limit": 100,
  "total_count": 38,
  "items": [
    {
      "timezone_id": 1,
      "title": "(UTC-12:00) Линия перемены дат"
    },
    {
      "timezone_id": 18,
      "title": "(UTC+03:00) Багдад, Минск, Москва"
    },
    {
      "timezone_id": 38,
      "title": "(UTC+14:00) О-в Киритимати"
    }
  ]
}
```


Возможные сообщения об ошибках:

Таблица 135 – Возможные сообщения об ошибках

Код	Значение
400	Неверные параметры запроса
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.8.1.2 Список регионов

Возвращает список регионов.

Метод запроса: GET.

Запрос:

```
/handbooks/regions
```

Параметры запроса (Parameters):

Таблица 136 – Параметры regions

Наименование	Тип данных	Описание
Offset	integer	*Значение по умолчанию: 0*
Limit	integer	*Значение по умолчанию: 100*

Ответ:

Таблица 137 – Ответ regions

Код	Описание	Связи
200	Список регионов	нет

Схема данных ответа (Schema): PaginatedRegions.

Пример ответа:


```
{
  "offset": 0,
  "limit": 100,
  "total_count": 90,
  "items": [
    {
      "region_id": 1,
      "title": "Республика Адыгея (Адыгея)"
    },
    {
      "region_id": 77,
      "title": "г. Москва"
    },
    {
      "region_id": 78,
```

```
        "title": "г. Санкт-Петербург"
      },
      {
        "region_id": 90,
        "title": "Иное"
      }
    ]
  }
```

Возможные сообщения об ошибках:

Таблица 138 – Сообщения об ошибках regions

Код	Значение
400	Неверные параметры запроса
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.8.2 Сообщения пользователя

Запрос на отправку пользовательского отзыва/сообщения (Feedback).

Запрос:

```
/feedback
```

Метод запроса: POST.

Параметры запроса:

Таблица 139 – Параметры тела запроса /feedback

Наименование	Тип данных	Пример
feedback	object Feedback	см. схему Feedback

Пример тела запроса:

```
{
  "name": "Иван Иванов",
  "email": "user@example.com",
  "message": "Не могу войти в аккаунт"
}
```

Ответ:

Таблица 140 – Ответ /feedback

Код	Описание	Связи
204	Обращение отправлено	нет

Схема данных ответа (Schema): Нет.

Возможные сообщения об ошибках:

Таблица 141 – Сообщения об ошибках /feedback

Код	Значение
400	Неверные параметры запроса.
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.8.3 Состояние сервиса

GET /api/ping — health-check (возвращает "pong").

Запрос на информацию о состоянии Сервиса. При корректной работе возвращает "pong".

Запрос:

```
/ping
```

Метод запроса: GET.

Параметры запроса: Нет.

Ответ:

Таблица 142 – Ответ на запрос /ping

Код	Описание	Связи
200	Сервер работает.	нет

Схема данных ответа (Schema): Нет..

Пример ответа на **Запрос:**

```
"pong"
```

Возможные сообщения об ошибках:

Таблица 143 – Сообщения об ошибках auth/authorize

Код	Значение
500	Внутренняя ошибка сервера

2.8.4 Функционал для разработчика

Приведенные в этом разделе эндпоинты используются при разработке и тестировании (Dev-эндпоинты). Они доступны только в режиме отладки при DEBUG=true.

2.8.4.1 Список отправленных dev-OTP

Запрос на получение списка отправленных dev OTP.

Запрос:

```
/dev/otp_messages
```

Метод запроса: GET.

Параметры запроса:

Таблица 144 – Параметры запроса /dev/otp_messages

Наименование	Тип данных	Пример
page	integer	*1*
page_size	integer	20
sent_to	string	user@example.com
transport	string	email

Ответ:

Таблица 145 – Ответ /feedback

Код	Описание	Связи
200	Список OTP-сообщений	нет

Схема данных ответа (Schema): PaginatedDevOTPMessages.

Пример ответа:

```
{
  "items": [
    {
      "message_id": 1,
```

```
        "sent_to": "user@example.com",
        "sent_at": 1742688000000,
        "transport": "email",
        "body": "Ваш код подтверждения: 123456"
      }
    ],
    "total": 1,
    "page": 1,
    "page_size": 20
}
```

Возможные сообщения об ошибках:

Таблица 146 – Сообщения об ошибках /dev/otp_messages

Код	Значение
403	Доступ запрещён
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.8.4.2 Принудительно завершить access-токен

Запрос на принудительное завершение access-токена.

Запрос:

```
/dev/token_expire
```

Метод запроса: POST.

Параметры запроса:

Таблица 147 – Параметры тела запроса /dev/token_expire

Наименование	Тип данных	Пример
access_token	string nullable: true	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...

Пример тела запроса:

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

Ответ:

Таблица 148 – Ответ /dev/token_expire

Код	Описание	Связи
200	Токен истёк, возвращает access_token	нет

Схема данных ответа (Schema): Нет.

Пример ответа:

```
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
```

Возможные сообщения об ошибках:

Таблица 149 – Сообщения об ошибках /dev/token_expire

Код	Значение
400	Неверные параметры запроса.
403	Доступ запрещён
429	Превышен лимит запросов
500	Внутренняя ошибка сервера

2.9 Используемые схемы данных

2.9.1 Timezone

Таблица 150 – Параметры схемы Timezone

Наименование	Тип данных	Пример
timezone_id	integer	*1*
title	string	*(UTC+03:00) Багдад, Минск, Москва*

2.9.2 Region

Таблица 151 – Параметры схемы Region

Наименование	Тип данных	Пример
region_id	integer	77
title	string	*г. Москва*

2.9.3 OTP

Таблица 152 – Параметры схемы OTP

Наименование	Тип данных	Пример
transport	string	email
intent	string	*email_registration*
credential	string	*user@example.com*
expires_at	string($date-time)	*2026-04-03T17:04:45Z*

2.9.4 User

Таблица 153 – Параметры схемы User

Наименование	Тип данных	Пример
user_id	string ($uuid), nullable: true	550e8400-e29b-41d4-a716-446655440000
email	string	*user@example.com*
email_verified	boolean	*true*
phone_number	string, nullable: true	*+79001234567*
phone_number_verified	boolean	*false*
is_active	boolean	*true*
otp_transport	string	*email*
first_name	string	*Иван*
last_name	string, nullable: true	*Иванов*
middle_name	string	*Иванович*
gender	string, nullable: true Enum: [MALE, FEMALE]	*MALE*
birthdate	integer, nullable: true	*1990-01-15*
timezone_id	integer, nullable: true	*1*
region_id	integer,	*77*

	nullable: true	
address_actual	string, nullable: true	*null*
email_candidate	string, nullable: true	*null*
phone_number_candidate	string, nullable: true	*null*

2.9.5 UserDetail

Таблица 154 – Параметры схемы UserDetail

Наименование	Тип данных	Пример
user_id	string ($uuid), nullable: true	550e8400-e29b-41d4-a716-446655440000
email	string	*user@example.com*
email_verified	boolean	*true*
phone_number	string, nullable: true	*+79001234567*
phone_number_verified	boolean	*false*
is_active	boolean	*true*
otp_transport	string	*email*
first_name	string	*Иван*
last_name	string, nullable: true	*Иванов*
middle_name	string	*Иванович*
gender	string, nullable: true Enum: [MALE, FEMALE]	*MALE*
birthdate	integer, nullable: true	*1990-01-15*
timezone_id	integer, nullable: true	*1*
region_id	integer, nullable: true	*77*
address_actual	string,	*null*

Наименование	Тип данных	Пример
	nullable: true	
email_candidate	string, nullable: true	*null*
phone_number_candidate	string, nullable: true	*null*
time_created	integer	*1742688000000* *Unix timestamp в миллисекундах*
time_updated	integer	*1742688000000* *Unix timestamp в миллисекундах*
avatar	object UserAvatar	см. схему UserAvatar
avatar_large	object UserAvatar	см. схему UserAvatar

2.9.6 UserToken

Таблица 155 – Параметры схемы UserToken

Наименование	Тип данных	Пример
access_token	string	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
refresh_token	string	*dGhpcyBpcyBhIHJlZnJlc2ggdG9rZW4...*
user_id	string (uuid)	*550e8400-e29b-41d4-a716-446655440000*
expires_at	string($date-time)	*2026-04-18T12:37:09Z*
otp_transport	string	*email*
is_confirmed	boolean	*true*
device_info	string	*Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)*

2.9.7 AuthAuthorizeResponse

Таблица 156 – Параметры схемы AuthAuthorizeResponse

Наименование	Тип данных	Пример
time_created	integer	*1742774400000* *Unix timestamp в миллисекундах*
time_updated	integer	*1742774400000* *Unix timestamp в миллисекундах*

access_token	string	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
refresh_token	string	*dGhpcyBpcyBhIHJlZnJlc2ggdG9rZW4...*
user_id	string (uuid)	*550e8400-e29b-41d4-a716-446655440000*
expires_at	string($date-time)	*2026-04-18T12:37:09Z*
status	string	*PENDING*
client	array (3, string)	*{"client_id": "P000000080", "client_name": "РОСА Контроль", "scope":"List ["openid", "email", "name"]"}*
device	object DeviceInfo nullable: true	*см. схему* DeviceInfo

2.9.8 UserAvatar

Таблица 157 – Параметры схемы UserAvatar

Наименование	Тип данных	Пример
image_id	integer, nullable: true	42
user_id	string (uuid)	*550e8400-e29b-41d4-a716-446655440000*

2.9.9 Document

Таблица 158 – Параметры схемы Document

Наименование	Тип данных	Пример
document_id	string ($uuid), nullable: true	b2c3d4e5-f6a7-8901-bcde-f1234567890
user_id	string ($uuid), nullable: true	550e8400-e29b-41d4-a716-446655440000
document_type	string Enum: [PASSPORT, INN, SNILS]	*PASSPORT*
document_number	string	*4500123456*
data	object nullable: true	*OrderedMap { "series": "45 00", "issued_by": "ОВД района Арбат", "issue_date": "2010-05-20" }*

2.9.10 Partner

Таблица 159 – Параметры схемы Partner

Наименование	Тип данных	Пример
partner_id	string ($uuid), nullable: true	b2c3d4e5-f6a7-8901-bcde-f1234567890
owner_user_id	string ($uuid), nullable: true	550e8400-e29b-41d4-a716-446655440000
partner_type	string Enum: [PERSON, SELF_EMPLOYED, SOLE_PROPRIETOR, ORGANIZATION]	*ORGANIZATION*
name	string	*ООО Ромашка*
data	object nullable: true	*OrderedMap { "inn": "7700000000", "ogrn": "1027700000000" }*

2.9.11 Client

Таблица 160 – Параметры схемы Client

Наименование	Тип данных	Пример
client_id	integer nullable: true	1
user_id	string	*550e8400-e29b-41d4-a716-446655440000*
name	string	*My App*
platforms	string Enum: [WEB, MOBILE]	*WEB*
redirect_uri	string	*https://myapp.example.com/oauth/callback*

2.9.12 DevOTPMessage

Таблица 161 – Параметры схемы DevOTPMessage

Наименование	Тип данных	Пример
message_id	integer nullable: true	1

sent_to	string	*user@example.com*
sent_at	integer	*1742731200000* *Unix timestamp в миллисекундах*
transport	string	*email*
body	string	*Ваш код подтверждения: 123456*

2.9.13 Feedback

Таблица 162 – Параметры схемы Feedback

Наименование	Тип данных	Пример
name	string,	Иван Иванов
email	string	user@example.com
message	string	Не могу войти в аккаунт

2.9.14 OAuthAuthorizeGetResponse

Таблица 163 – Параметры схемы OAuthAuthorizeGetResponse

Наименование	Тип данных	Пример
client	object Client	см. схему Client

2.9.15 OAuthAuthorizePostResponse

Таблица 164 – Параметры схемы OAuthAuthorizePostResponse

Наименование	Тип данных	Пример
redirect_url	string,	https://myapp.example.com/oauth/callback?code=abc123xyz

2.9.16 OIDCUserinfo

Таблица 165 – Параметры схемы OIDCUserinfo

Наименование	Тип данных	Пример
sub	string ($uuid), nullable: true	550e8400-e29b-41d4-a716-446655440000
email	string	*user@example.com*
email_verified	boolean	*true*

phone_number	string, nullable: true	*+79001234567*
phone_number_verified	boolean	*false*
is_active	boolean	*true*
otp_transport	string	*email*
first_name	string	*Иван*
last_name	string, nullable: true	*Иванов*
middle_name	string	*Иванович*
gender	string, nullable: true Enum: [MALE, FEMALE]	*MALE*
birthdate	integer, nullable: true	*1990-01-15*
avatar_large	object UserAvatar	см. схему UserAvatar

2.9.17 OAuthTokenResponse

Таблица 166 – Параметры схемы OAuthTokenResponse

Наименование	Тип данных	Пример
access_token	string	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
token_type	string	*Bearer*
refresh_token	string	*dGhpcyBpcyBhIHJlZnJlc2ggdG9rZW4...*
expires_in	integer	*3600*
scope	string	*openid email profile*
device_info	string nullable: true	*eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9...*

2.9.18 PaginatedTimezones

Таблица 167 – Параметры схемы PaginatedTimezones

Наименование	Тип данных	Пример
items	object Timezone	см. схему Timezone
offset	integer	*0*

limit	integer	*100*
total_count	integer	*38*

2.9.19 PaginatedRegions

Таблица 168 – Параметры схемы PaginatedRegions

Наименование	Тип данных	Пример
items	object Region	см. схему Region
offset	integer	*0*
limit	integer	*100*
total_count	integer	*38*

2.9.20 PaginatedDocuments

Таблица 169 – Параметры схемы PaginatedDocuments

Наименование	Тип данных	Пример
items	object Document	см. схему Document
total	integer	2
page	integer	*1*
page_size	integer	*20*

2.9.21 PaginatedPartners

Таблица 170 – Параметры схемы PaginatedPartners

Наименование	Тип данных	Пример
items	object Partner	см. схему Partner
total	integer	1
page	integer	*1*
page_size	integer	*20*

2.9.22 PaginatedClients

Таблица 171 – Параметры схемы PaginatedClients

Наименование	Тип данных	Пример
items	object AuthHistoryEntry	см. схему AuthHistoryEntry
total	integer	*2*
page	integer	*1*
page_size	integer	*20*

2.9.23 AuthHistoryEntry

Таблица 172 – Параметры схемы AuthHistoryEntry

Наименование	Тип данных	Пример
auth_history_id	string ($uuid)	7fb8d9c0-bdcb-4141-9931-33fae97acb7d
time_created	integer	*1742774400000* *Unix timestamp в миллисекундах*
source	string Enum: [DIRECT, OAUTH]	*DIRECT*
user_token_id	string Enum: [LOGIN, LOGOUT]	*LOGIN*
user_token_id	String nullable: true	d1f6a5078bfda5ebc7c71bdba91e8001e8 3e17e9d2d5985b0f140c8791c3763
device	object DeviceInfo nullable: true	*см. схему* DeviceInfo
client	array (2, string)	*{"client_id": "P000000080",* *"client_name": "РОСА Контроль"}*

2.9.24 DeviceInfo

Таблица 173 – Параметры схемы DeviceInfo

Наименование	Тип данных	Пример
device_platform	string	mac
device_os	string	*Mac OS X*
device_version	string	*10.15.7*
app_device	string	*Chrome*
app_build	string	*146.0.0*
device_locale	string	*ru*
user_agent	string	*Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/146.0.0.0 Safari/537.36*

2.9.25 PaginatedAuthHistory

Таблица 174 – Параметры схемы PaginatedAuthHistory

Наименование	Тип данных	Пример
items	object AuthHistoryEntry	см. схему AuthHistoryEntry
total	integer	*2*
page	integer	*1*
page_size	integer	*20*

2.9.26 PaginatedSessions

Таблица 175 – Параметры схемы PaginatedSessions

Наименование	Тип данных	Пример
items	object UserToken	см. схему UserToken
total	integer	*2*
page	integer	*1*
page_size	integer	*20*

2.9.27 PaginatedDevices

Таблица 176 – Параметры схемы PaginatedDevices

Наименование	Тип данных	Пример
items	object UserToken	см. схему UserToken
total	integer	*2*
page	integer	*1*
page_size	integer	*20*

2.9.28 PaginatedDevOTPMessages

Таблица 177 – Параметры схемы PaginatedDevOTPMessages

Наименование	Тип данных	Пример
items	object DevOTPMessage	см. схему DevOTPMessage
total	integer	3
page	integer	*1*
page_size	integer	*20*

2.9.29 Error

Таблица 178 – Параметры схемы Error

Наименование	Тип данных	Пример
error	string	not_exist
message	string	Requested object does not exist

ПРИЛОЖЕНИЕ А. ТРЕБОВАНИЯ К ЗАГОЛОВКУ USER-AGENT

Информации заголовка User-Agent формируется приложением в виде строки, состоящей из компонентов, описывающих параметры устройства, операционной системы и приложения (Таблица 179). Строка имеет следующую структуру:

(%OS% %VERSION_OS%; %DEVICE% %DEVICE_VERSION%) %APPLICATION%/%APPLICATION_VERSION%.

Таблица 179 – Компоненты строки заголовка User-Agent

Компонент	Описание	Формат	Пример
%OS%	Название операционной системы	RosaOS или RosaMobile	RosaMobile
%VERSION_OS%;	Версия ОС в формате X.Y.Z.W	(\d+)\.(\d+)(?:\.(\d+)(?:\.(\d+))?)?	3.0.0.0
%DEVICE%	Модель устройства (по данным системы)	Строка, буквенно-цифровая	RPhone 1.0, Macintosh
%DEVICE_VERSION%	Номер сборки прошивки устройства (только для RosaMobile)	Build/(\d+)\.(\d+)(?:\.(\d+)(?:\.(\d+))?)? Допускается сокращённый формат	Build/10.20.30.40 Build/10.20
%APPLICATION%	Имя клиентского приложения	RosaMessenger или RosaId	RosaMessenger
%APPLICATION_VERSION%	Версия приложения	(\d+)(?:\.(\d+)(?:\-(\d+)(?:\.(\s+))?)?)?	0.6.18

При формировании строки необходимо учитывать следующий синтаксис:

– вся информация об ОС и устройстве заключается в круглые скобки;

– внутри скобок после ОС и её версии ставится точка с запятой с пробелом;

– модель устройства и номер сборки разделяются одним пробелом;

– между закрывающей скобкой и именем клиентского приложения — один пробел;

– имя приложения и его версия разделяются косой чертой "/".

Компонент %DEVICE_VERSION% используется только для устройств под управлением RosaMobile (смартфоны R-Phone). В этом случае шаблон строки принимает такой вид:

(RosaMobile %VERSION_OS%; RPhone Build/%BUILD_NUMBER%) %APPLICATION%/%APPLICATION_VERSION%.

Например:

(RosaMobile 3.0.0.0; RPhone Build/10.20.30.40) RosaMessenger/0.6.18

Для RosaOS %DEVICE_VERSION% опускается. Соответственно, шаблон будет таким:

(RosaOS %VERSION_OS%; %DEVICE%) %APPLICATION%/%APPLICATION_VERSION%.

Например:

(RosaOS 10.9.4.35; Macintosh) RosaId/1.2.0.0

Валидация со стороны Сервиса

Роса ID проверяет информацию заголовка User-Agent по следующим правилам:

– наличие обязательных компонентов (ОС, версия ОС, устройство, приложение, версия приложения);

– соответствие %OS% значению RosaOS или RosaMobile (регистр важен);

– наличие точки с запятой с пробелом между %VERSION_OS% и %DEVICE%;

– для RosaMobile — наличие Build/X.Y.Z.W после модели устройства;

– для RosaOS — отсутствие Build после модели устройства;

– косая черта между именем приложения и версией приложения.

Если информация заголовка не соответствует формату — устройство отображается в интерфейсе как «Неизвестное устройство» с пометкой «Некорректный User-Agent».

Рекомендации разработчикам

Во избежание ошибок формирования строки заголовка рекомендуется придерживаться следующих правил:

– не применяйте в заголовке статическую строку (например, curl/7.68.0);

– не добавляйте лишних пробелов и символов перевода строки;

– при обновлении версии приложения обязательно меняйте %APPLICATION_VERSION%;

– для RosaMobile всегда передавайте актуальный номер сборки прошивки в формате Build/X.Y.Z.W;

– для RosaOS не добавляйте Build — передавайте только модель устройства;

– перед отправкой запроса с новым заголовком проверьте работу сервиса на одном из корректных примеров.

ПРИЛОЖЕНИЕ Б. КОДЫ ОШИБОК, ИСПОЛЬЗУЕМЫЕ В API

Код ошибки	Описание
invalid_param_value	Неверное значение параметра запроса
auth_credentials_invalid	Неверный логин или пароль
auth_forbidden	Доступ запрещён (недостаточно прав)
user_deactivated	Учётная запись деактивирована
user_not_verified	Email/телефон не подтверждён
email_taken	Email уже занят
phone_number_taken	Телефон уже занят
credential_not_verified	Контакт не подтверждён для данной операции
otp_invalid	Неверный OTP-код
otp_expired	OTP-код истёк
otp_attempts_exceeded	Превышен лимит попыток ввода OTP
user_verified_already	Пользователь уже верифицирован
not_exist	Запрошенный объект не найден
throttling	Превышен лимит запросов
unknown	Неизвестная внутренняя ошибка сервера

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

Сокращение	Определение
БД	База данных
ВМ	Виртуальная машина
ИТ	Информационные технологии
ОС	Операционная система
OTP	Одноразовый пароль (One-Time Password), отправляемый по email или SMS
ПО	Программное обеспечение
ЦС	Центр сертификации
API	Application programming interface – программный интерфейс приложения
CSV	Comma separated values – текстовый формат, предназначенный для представления табличных данных
DHCP	Dynamic host configuration protocol – протокол динамической настройки сетевой конфигурации узла
DNS	Domain name system – система доменных имен
HTTP	Hypertext transfer protocol – протокол передачи гипертекста
HTTPS	Hypertext transfer protocol secure – защищенная версия протокола передачи гипертекста
IP	Internet protocol – протокол межсетевого взаимодействия
SSH	Secure shell – защищенная оболочка
JID	Идентификатор пользователя (для совместимости с OMMG, но в ROSA ID используется user_id)
JWT	JSON Web Token — строковый токен доступа
SSL	Secure sockets layer – уровень защищенных сокетов
TCP	Transmission control protocol – протокол управления передачей данных
TIME64	Время в формате Unix time 64 (число миллисекунд с 01.01.1970), в примерах используется ISO 8601
TIMESTAMP	Дата и время в формате ISO 8601 (UTC), например, 2026-04-03T17:04:45Z
UUID	Универсальный уникальный идентификатор, 36 символов, строка (например, 550e8400-e29b-41d4-a716-446655440000)
UserToken	Сессия пользователя, содержит информацию об устройстве и времени жизни