

СТАНДАРТ БАНКА РОССИИ

**ОТКРЫТЫЕ БАНКОВСКИЕ ИНТЕРФЕЙСЫ**

Профили API для расширенного режима безопасности

Дата введения: 2023-

Версия 1.3.3

**МОСКВА**

**2023**

# Предисловие

ПРИНЯТ И ВВЕДЕН в действие приказом Банка России от \_\_\_\_\_ 2022 года № \_\_\_\_\_\_\_«О введении в действие новых редакций стандартов Банка России и стандарта Банка России».

Настоящий Стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Банка России.

Содержание

[Предисловие 2](#_Toc86152987)

[1 Введение 2](#_Toc86152988)

[2 Область применения 2](#_Toc86152989)

[3 Термины и определения 3](#_Toc86152990)

[4 Общие положения 4](#_Toc86152991)

[5 Общие требования к параметрам взаимодействия 4](#_Toc86152992)

[5.1 Кодировка символов 4](#_Toc86152993)

[5.2 Формат даты 4](#_Toc86152994)

[5.3 Структура пути URI ресурса 4](#_Toc86152995)

[5.4 Заголовки сообщений (headers) 5](#_Toc86152996)

[5.5 Коды статусов HTTP 8](#_Toc86152997)

[5.6 Идемпотентность 12](#_Toc86152998)

[5.7 Подписание сообщений 12](#_Toc86152999)

[5.8 Фильтрация 12](#_Toc86153000)

[5.9 Нумерация страниц 13](#_Toc86153001)

[5.10 Архивирование 13](#_Toc86153002)

[5.11 Дополнительные данные 13](#_Toc86153003)

[6 Безопасность и контроль доступа 14](#_Toc86153004)

[6.1 Область действия 14](#_Toc86153005)

[6.2 Типы предоставления доступа 14](#_Toc86153006)

[6.3 Авторизация согласия 14](#_Toc86153007)

[6.4 Аутентификации Пользователя на стороне ППИУ 14](#_Toc86153008)

[6.5 Требования к токену идентификации (ID Token) 16](#_Toc86153009)

[6.6 Требования к токену доступа и токену обновления 16](#_Toc86153010)

[7 Основные сущности комплекса Стандартов 16](#_Toc86153011)

[7.1 Справочники и перечисления 16](#_Toc86153012)

[7.2 Общая структура полезной нагрузки 16](#_Toc86153013)

# Введение

Настоящий Стандарт содержит принципы и рекомендации по обмену данными для осуществления взаимодействия через Открытые банковские интерфейсы при осуществлении доступа к данным в режиме чтения и записи.

# Область применения

Стандарт рекомендован к использованию организациями при обмене финансовыми сообщениями, связанными с:

* получением информации о банковском счете;
* переводом денежных средств в валюте Российской Федерации.

Настоящий Стандарт предназначен для:

* участников получения информации о банковском счете (кредитные организации и их клиенты, а также Сторонние поставщики[[1]](#footnote-1)[[2]](#footnote-2));
* участников перевода денежных средств (банки и их клиенты, а также Сторонние поставщики[[3]](#footnote-3));
* разработчиков информационного и программного обеспечения, информационных систем.

Положения настоящего стандарта применяются на добровольной основе, если только в отношении конкретных положений обязательность их применения не установлена нормативными актами Банка России или условиями договоров.

Положения настоящего стандарта применяются совместно со следующими документами:

* СТО БР ФАПИ.СЕК-1.6-2020 Стандарт Банка России "Безопасность финансовых (банковских) операций. Прикладные программные интерфейсы обеспечения безопасности финансовых сервисов на основе протокола OpenID" (далее ФАПИ.СЕК).
* СТО БР ФАПИ.ПАОК-1.0-2021 Стандарт Банка России "Безопасность финансовых (банковских) операций. Обеспечения безопасности финансовых сервисов при инициации OpenID Connect клиентом потока аутентификации по отдельному каналу" (далее ФАПИ.ПАОК).
* СТО БР Открытые банковские интерфейсы. Общие положения.

#  Термины и определения

В Стандарте применяются термины и определения в соответствии со стандартами ФАПИ.СЕК, ФАПИ.ПАОК, «Открытые банковские интерфейсы. Общие положения» а также следующие:

|  |  |
| --- | --- |
| **Наименование** | **Описание** |
| Полезная нагрузка | Часть пакета данных (сообщения) без служебной информации. |
| Простая аутентификация Пользователя (CA) | Аутентификация пользователя с однофакторной односторонней аутентификации. Применение простой аутентификации означает, что отсутствует гарантия того, что к пользователю была применена усиленная аутентификация |
| Усиленная аутентификация Пользователя (SCA) | Аутентификация пользователя с применением многофакторной аутентификации. Применение усиленной аутентификации означает, что пользователь гарантированно аутентифицировался с использованием не менее двух различных факторов аутентификации |

Таблица 1 - Термины и определения

# Общие положения

Информационный обмен между банками и Сторонними поставщиками осуществляется посредством электронных сообщений, формируемых на стороне банков и на стороне Сторонних поставщиков посредством Открытых банковских интерфейсов.

# Общие требования к параметрам взаимодействия

## Кодировка символов

Запросы и ответы API используют кодировку UTF-8. Это кодировка символов по умолчанию для JSON (RFC 7158 - [раздел 8.1](https://tools.ietf.org/html/rfc7158#section-8.1)).

Однако ППИУ может не принимать некоторые символы UTF-8, такие как символы emoji (например, идеограммы и «смайлики» могут не быть приемлемой ссылкой на платеж). Если ППИУ отклоняет сообщение с символом UTF-8, которое не может быть обработано, то ППИУ отвечает кодом состояния HTTP 400 (неверный запрос).

## Формат даты

ППИУ принимает в запросах все действующие форматы даты стандарта ISO 8601, включая его разрешенные вариации.

Все даты в полезных нагрузках JSON представлены в формате dateTime стандарта ISO 8601. Все поля dateTime в ответах включают часовой пояс. Например:

|  |
| --- |
| 2019-07-08T11:23:03+00:00 |

Все даты в параметрах query имеют формат dateTime стандарта ISO 8601 и не включают часовой пояс. Например:

|  |
| --- |
| 2019-07-01T09:23:012019-02-07 |

Все даты в заголовках HTTP представлены как полные даты [RFC 7231](https://tools.ietf.org/html/rfc7231#section-7.1.1.1). Пример:

|  |
| --- |
| Mon, 26 Aug 2019 14:23:51 GMT |

Все даты в параметрах claims JWT имеют формат number JSON, представляющий количество секунд с 1970-01-01T0:0:0Z, измеренное в GMT до текущей даты (dateTime). Пример:

|  |
| --- |
| //Mon, 26 Aug 2019 14:23:51 GMT1987646700 |

## Структура пути URI ресурса

Путь URI соответствует следующей структуре:

* [participant-path-prefix]/open-banking/[version]/[resource-group]/[resource]/[resource-id]/[sub-resource]

Структура URI пути состоит из следующих элементов:

* [participant-path-prefix]
Необязательный префикс ППИУ.
* open-banking
Постоянное значение "open-banking".
* [version]
Версия API, выраженная в виде /v[major-version].[minor-version]/.
* [resource-group]

Идентификатор группы конечных точек, в соответствии с ролью участника.

* [resource]/[resource-id]
Наименование ресурса и его идентификатор.
* [sub-resource]
Наименование подресурса (ресурса 2-го уровня).

ППИУ использует один и тот же participant-path-prefix и host name для всех своих ресурсов.

Примеры:

|  |
| --- |
| https://bank.ru/oapi-channel/open-banking/v1.2/pisp/paymentshttps://bank.ru/oapi-channel/open-banking/v1.2/pisp/payments/1234https://bank.ru/oapi-channel/open-banking/v1.3/aisp/v1.1/account-consentshttps://bank.ru/oapi-channel/open-banking/v1.3/aisp/accountshttps://bank.ru/oapi-channel/open-banking/v1.3/aisp/accounts/1234https://bank.ru/oapi-channel/open-banking/v1.3/aisp/accounts/1234/transactions |

## Заголовки сообщений (headers)

### Заголовки запросов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Параметр header** | **Комментарий** | **POST**  | **GET**  | **DELETE**  | **PUT**  |
| x-fapi-auth-date | Время последнего входа Пользователя в приложение Стороннего поставщика. Значение предоставляется в виде HTTP-date, как в разделе [7.1.1.1](https://tools.ietf.org/html/rfc7231#section-7.1.1.1) [RFC 7231]. Например, x-fapi-auth-date: Mon, 26 Aug 2019 12:23:11 GMT | Необязательно | Необязательно | Необязательно | Не используется |
| x-fapi-customer-ip-address | IP-адрес Пользователя, если Пользователь в данный момент подключен к Стороннему Поставщику (залогинен в приложении Стороннего Поставщика). | Необязательно | Необязательно | Необязательно | Не используется |
| x-fapi-interaction-id | [RFC 4122](https://tools.ietf.org/html/rfc4122) UID, используемый в качестве идентификатора корреляции.Если необходимо, то ППИУ передает обратно значение идентификатора корреляции в заголовке ответа x-fapi-interaction-id. | Обязательно | Обязательно | Обязательно | Обязательно |
| Authorization | Стандартный заголовок HTTP. Позволяет предоставлять учетные данные серверу авторизации и/или серверу ресурсов в зависимости от типа запрашиваемого ресурса. Для OAuth 2.0 / OIDC включает в себя Basic или Bearer схемы аутентификации. | Обязательно | Обязательно | Обязательно | Обязательно |
| Content-Type | Стандартный заголовок HTTP. Представляет формат полезной нагрузки в запросе.Устанавливается значение application/json, за исключением конечных точек, которые поддерживают Content-Type, отличный от application/json (например, POST /file-payment-consents/{сonsentId}/file).Устанавливается значение application/jose+jwe для зашифрованных запросов.Сторонний поставщик может предоставлять дополнительную информацию.Если установлено другое значение, то ППИУ присылает ответ: 415 Unsupported Media Type. | Обязательно | Не используется | Не используется | Обязательно |
| Accept | Стандартный HTTP заголовок, определяющий тип контента, который требуется от сервера.Если Сторонний поставщик ожидает незашифрованный ответ, то он указывает явно, что только ответ в формате JSON принимается (передавая значение application/json) в качестве заголовка контента для всех конечных точек, которые отвечают в формате JSON.Если Сторонний поставщик ожидает зашифрованный ответ, то он указывает явно, что принимается только ответ JWT (передавая значение application/jose+jwe) в качестве заголовка контента для всех конечных точек, которые отвечают JSON.Для конечных точек, которые не отвечают в формате JSON (например, GET ../statements/{statementId}/file), ППИУ указывает доступные параметры на своем портале для разработчиков.Сторонний поставщик может предоставлять дополнительную информацию.Если установлено недопустимое значение, то ППИУ отвечает: 406 (Not Acceptable).Если значение не указано, по умолчанию используется application/json. | Необязательно | Необязательно | Не используется | Необязательно |
| x-idempotency-key | Не стандартный HTTP заголовок. Уникальный идентификатор запроса для поддержки идемпотентности. Обязательно для запросов POST к конечным точкам идемпотентного ресурса.Для других запросов не указывается. | Необязательно | Не используется | Не используется | Не используется |
| x-jws-signature | Указывает, что тело запроса подписано.В документации на ресурсы отдельно определяется, когда это поле в заголовке указывается. | Условно (зависит от API) | Не используется ) | Не используется | Обязательно |
| x-customer-user-agent | В заголовке указывается тип устройства (user-agent), который использует Пользователь.Сторонний поставщик может заполнить это поле значением типа устройства (user-agent), указанным Пользователем.Если Пользователь использует мобильное приложение Стороннего Поставщика, Сторонний поставщик проверяет, что строка типа устройства (user-agent) отличается от строки типа устройства (user-agent) на основе браузера. | Необязательно | Необязательно | Необязательно | Необязательно |

Таблица 2 Заголовки запросов

Наличие или отсутствие Пользователя определяется через заголовок x-fapi-customer-ip-address. Если указан IP-адрес Пользователя, то предполагается, что Пользователь присутствует во время взаимодействия.

Последствием этого является, что ППИУ полагаются на информацию, предоставленную СППУ.

### Заголовки ответов

|  |  |  |
| --- | --- | --- |
| **Параметр header** | **Комментарий** | **Обязательность** |
| Content-Type | Стандартный параметр заголовка HTTP. Представляет формат полезной нагрузки, возвращаемой в ответе.ППИУ возвращает значение Content-Type, равное application/json, в качестве заголовка для всех незашифрованных конечных точек.ППИУ возвращает значение Content-Type, равное application/jwe, для всех зашифрованных конечных точек | Обязательно |
| Retry-After | Параметр заголовка, указывающий время (в секундах), в течение которого Сторонний поставщик ожидает перед повторением операции.ППИУ следует включать этот заголовок вместе с ответами с кодом состояния HTTP 429 (Too Many Requests) | Необязательно |
| x-fapi-interaction-id | [RFC 4122](https://tools.ietf.org/html/rfc4122) UID, используемый в качестве идентификатора корреляции.ППИУ заполняет параметр заголовка ответа x-fapi-interaction-id значением полученным в соответствующем параметре заголовка запроса или значением UID [RFC 4122](https://tools.ietf.org/html/rfc4122), если значение не было предоставлено в запросе для отслеживания взаимодействия | Обязательно |
| x-jws-signature | Указывает, что тело ответа подписано.В документации на ресурсы отдельно определяется, когда указывается это поле в заголовке | Условно (зависит от API) |

Таблица 3 - Заголовки ответов

## Коды статусов HTTP

Ниже приведены коды ответов HTTP для различных методов HTTP, для всех конечных точек API. Параметры ответа и значения кодов должны соответствовать требованиям разделов 5.4.2.8.и 5.4.2.9 ФАПИ.СЕК.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ситуация** | **Статус HTTP**  | **Комментарий** | **POST** | **GET** | **DELETE** | **PUT** |
| Запрос успешно выполнен | 200 OK |  | Нет | Да | Нет | Да |
| Операция создания выполнена успешно. | 201 Created | Результатом операции является создание нового ресурса. | Да | Нет | Нет | Нет |
| Операция удаления успешно завершена. | 204 No Content |  | Нет | Нет | Да | Нет |
| Запрос имеет не верный формат, отсутствующее или несовместимое тело JSON, параметры URL или поля заголовка | 400 Bad Request | Запрошенная операция не будет выполнена. | Да | Да | Да | Да |
| Заголовок авторизации отсутствует или неверный токен. | 401 Unauthorized | Операции было отказано в доступе.Повторная аутентификация Пользователя может привести к созданию соответствующего токена, который может быть использован. | Да | Да | Да | Да |
| Токен имеет неверную область действия или была нарушена политика безопасности. | 403 Forbidden | Операции было отказано в доступе.Повторная аутентификация Пользователя может привести к созданию соответствующего токена, который может быть использован. | Да | Да | Да | Да |
| Сторонний поставщик пытается получить ресурс, который указан в спецификации, но не реализован на стороне ППИУ (например, ППИУ решил не реализовывать конечную точку API статуса для внутренних запланированных платежей).Сторонний поставщик пытается получить ресурс, который не определен | 404 (Not Found) |  | Да | Да | Да | Да |
| Сторонний поставщик попытался получить доступ к ресурсу с помощью метода, который не поддерживается. | 405 Method Not Allowed |  | Да | Да | Да | да |
| Запрос содержал параметр заголовка Accept, отличный от разрешенных media types, и набор символов, отличный от UTF-8. | 406 Not Acceptable |  | Да | Да | Да | Да |
| Операция была отклонена, поскольку полезная нагрузка находится в формате, не поддерживаемом этим методом на целевом ресурсе. | 415 Unsupported Media Type |  | Да | Нет | Нет | Да |
| Операция была отклонена, так как слишком много запросов было сделано в течение определенного периода времени. | 429 Too Many Requests | ППИУ ограничивают запросы, если они сделаны сверх их политики добросовестного использования.ППИУ документируют свои политики добросовестного использования на своих порталах для разработчиков.ППИУ отвечают этим статусом, если количество запросов в единицу времени было превышено.ППИУ следует включать заголовок Retry-After в ответ, указывающий, как долго Сторонний поставщик ожидает перед повторением операции. | Да | Да | Да | Да |
| Что-то пошло не так на стороне ППИУ. | 500 Internal Server Error | Операция не удалась. | Да | Да | Да | Да |
| Сервис не реализован на стороне ППУ | 501 Not Implemented | ППИУ не поддерживает возможностей, необходимых для обработки запроса. Типичный ответ для случаев, когда ППИУ не понимает указанный в запросе метод. Если же метод серверу известен, но он не применим к данному ресурсу, то нужно вернуть ответ 405. | Нет | Да | Нет | Нет |
| Устаревшая версия сервиса. | 503 Service Unavailable | Если API устарел и больше не поддерживается ППИУ, его путь URI все еще может быть активным и принимать запросы. В этом контексте рекомендуется вернуть 503 Service Unavailable, чтобы TPP знал, что версия API находится в офлайн режиме. | Да | Да | Да | Да |

Таблица 4 - Коды статусов HTTP

ППИУ возвращают другие стандартные коды состояния HTTP (например, от шлюзов и других периферийных устройств), как описано в [RFC 7231 - Раздел 6](https://tools.ietf.org/html/rfc7231#section-6).

ППИУ отвечают ошибкой в потоке OAuth / OIDC с обязательным выравниванием кодов ошибок, указанными в [разделе 3.1.2.6 OpenID Connect Core Specification](http://OpenID).

ППИУ отвечают на все некорректные запросы общей структурой ошибок Открытых банковских интерфейсов.

### 400 (Bad Request) или 404 (Not Found)

Если Сторонний поставщик пытается запросить URL ресурса с идентификатором ресурса, который не существует, то ППИУ отвечает 400 (неверный запрос), а не 404 (не найдено).

Например, если Сторонний поставщик пытается выполнить запрос GET/payment/22289, где 22289 не является действительным paymentId, ППИУ отвечает 400.

Если Сторонний поставщик пытается получить доступ к URL-адресу ресурса, который не определен этими спецификациями (например, GET /bulk), то ППИУ отвечает 404 (Not Found).

Если ППИУ не реализовал конечную точку API, то он отвечает 404 (не найдено) для запросов к этому URL.

Таблица ниже иллюстрирует некоторые примеры предсказуемого поведения:

|  |  |  |
| --- | --- | --- |
| **Ситуация** | **Запрос** | **Ответ** |
| Сторонний поставщик пытается получить платеж с несуществующим идентификатором paymentId | GET /payments/1001 | 400 (Bad Request) |
| Сторонний поставщик пытается получить ресурс, который указан в спецификации, но не реализован на стороне ППИУ.Например, ППИУ решил не реализовывать конечную точку API для получения транзакций по счету | GET /accounts/{accountId}/transactions | 404 (Not Found) |
| Сторонний поставщик пытается получить ресурс, который не определен | GET /bulk | 404 (Not Found) |

Таблица 5 - Возможные ситуации для ответов 400 (Bad Request) и 404 (Not Found)

### 403 (Forbidden)

Когда Сторонний поставщик пытается получить доступ к ресурсу, к которому у него нет разрешения, ППИУ возвращает 403 (Forbidden) с необязательным телом ответа об ошибке.

Ситуация возможна в следующих случаях:

* Сторонний поставщик использует токен доступа (access token), который не имеет соответствующей области действия (scope) для доступа к запрошенному ресурсу.
* Сторонний поставщик попытался получить доступ к ресурсу с идентификатором, к которому у него нет доступа. Например, попытка получить доступ к GET /payments/1001, где платежный ресурс с идентификатором 1001 принадлежит другому Стороннему Поставщику.
* Сторонний поставщик пытается получить доступ к ресурсу транзакции, а у Стороннего Поставщика нет согласия на авторизацию с правами доступа к запрашиваемому ресурсу.
* Сторонний поставщик пытается получить доступ к ресурсу транзакции, а у Стороннего Поставщика нет согласия на авторизацию для accountId. Например, попытка получить доступ к GET /accounts/2001 или GET /accounts/2001/transactions, когда Пользователь не выбрал accountId 2001 для авторизации.
* Сторонний поставщик пытается получить доступ к ресурсу, а ППИУ решает повторно аутентифицировать Пользователя. ППИУ отвечает соответствующим кодом ошибки, чтобы указать, что требуется повторная аутентификация.

### 401 (Unauthorized)

Когда Сторонний поставщик использует токен доступа с истекшим сроком, ППИУ возвращает 401 (Unauthorized) без какого-либо сообщения об ошибке.

Ситуация возникает, если ППИУ завершил срок действия токена доступа по любой из следующих причин:

* Истек срок действия согласия.
* Подозрительное использование токена доступа или подозрение в мошенничестве.
* Плановая реализация усиленной аутентификации.

Эта ошибка потенциально может быть исправлена, если Пользователь повторно пройдет аутентификацию или аутентифицируется с использованием усиленной аутентификации.

### 429 (Too Many Requests)

Если Сторонний поставщик пытается получить доступ к ресурсу слишком часто, то ППИУ может вернуть 429 (Too Many Requests). Это нефункциональное требование, и отдельные ППИУ определяют метрику запросов в единицу времени.

Ситуация возникает, когда:

* Сторонний поставщик решает реализовать функцию “Статус платежа в реальном времени” для своих пользователей и делает это некорректно, опрашивая конечную точку методом GET.
* Сторонний поставщик решает использовать конечную точку для разового единичного платежа, как если бы она была конечной точкой пакетной оплаты, и отправляет большое количество запросов на оплату в очень короткий промежуток времени, так что это превышает политику использования ППИУ.

## Идемпотентность

Ключ идемпотентности используется для защиты от создания дубликатов ресурсов при использовании метода POST для конечных точек API.

Если для конечной точки API требуется ключ идемпотентности:

* Параметр заголовка x-idempotency-key содержит не более 40 символов. Если длина поля превышает 40 символов, то ППИУ отклоняет запрос с кодом состояния 400 (Bad Request).
* Сторонний поставщик не меняет тело запроса при использовании одинакового ключа x-idempotency-key. Если Сторонний поставщик изменяет тело запроса, то ППИУ не меняет конечный ресурс. ППИУ рассматривает это как мошенническое действие.
* ППИУ обрабатывает запрос как идемпотентный, если он получил запрос с существующим параметром x-idempotency-key от того же Стороннего Поставщика в течение 24 часов.
* ППИУ не создает новый ресурс для запроса POST, если он определен как идемпотентный запрос.
* ППИУ отвечает на запрос текущим статусом ресурса (или статусом, максимально близким к текущему, который можно получить в данный момент времени на существующем онлайн канале) и кодом статуса HTTP 201 (Created).
* Сторонний поставщик не использует ключ идемпотентности при опросе состояния ресурсов.
* ППИУ использует подпись сообщения вместе с ключом идемпотентности, чтобы гарантировать, что тело запроса не изменилось.

Если ключ идемпотентности не требуется для конечной точки API, но содержится в запросе, то ППИУ игнорирует ключ идемпотентности.

## Подписание сообщений

Для подписания сообщений в среде Открытых банковских интерфейсов используется цифровая подпись в формате JSON, как определено в разделе 5.7.1 ФАПИ.СЕК. Обязательность подписания полезной нагрузки сообщения документируется в спецификациях API для прикладных стандартов.

## Фильтрация

ППИУ обеспечивает ограниченную поддержку фильтрации для операций GET, которые возвращают множественные записи.

Параметры фильтра всегда разные для конкретного поля (полей) ресурса и следуют правилам/форматам, определенным в справочниках для ресурса.

Для параметров фильтра типа DateTime значения соответствуют стандарту ISO 8601. Если значение поля типа DateTime содержит часовой пояс, то ППИУ игнорирует эту информацию.

Предполагается, что значения фильтра относятся к тому же часовому поясу, что и часовой пояс, в котором поддерживается ресурс.

## Нумерация страниц

ППИУ предоставляет постраничный ответ для операций GET, которые возвращают множественные записи.

В такой ситуации ППИУ:

* Если существует следующая страница записей ресурсов, то ППИУ предоставляет ссылку на следующую страницу ресурсов в поле Links.next ответа. Отсутствие следующей ссылки будет означать, что текущая страница является последней страницей результатов.
* Если предыдущая страница записей ресурсов существует, то ППИУ предоставляет ссылку на предыдущую страницу ресурсов в поле Links.prev ответа. Отсутствие предыдущей ссылки указывает на то, что текущая страница является первой страницей результатов.

Для разбитых на страницы ответов ППИУ гарантирует, что количество записей на странице находится в разумных пределах, минимум 25 записей (кроме последней страницы, где больше нет записей) и максимум 1000 записей.

Дополнительно ППИУ предоставляет:

* Ссылку на первую страницу результатов в поле Links.first.
* Ссылку на последнюю страницу результатов в поле Links.last.
* Общее количество страниц в поле Meta.totalPages.

ППИУ включает “self” ссылку на ресурс в поле Links.self, как описано в разделе “Ссылки”.

Этот стандарт не определяет, каким образом параметры перелистывания страниц передаются ППИУ, каждый ППИУ может использовать свои собственные механизмы для разбивки ответа.

Если исходный запрос от СППУ включал параметры фильтра, то в ответе возвращаются только те результаты (разбитые на страницы), которые соответствуют фильтру.

## Архивирование

Архивация ресурсов будет определяться для ППИУ на основе их внутренних требований.

В дополнение стоит заметить, что:

* ППИУ удаляют просроченные идентификаторы согласия (consentId) только через 24 часа после создания.
* ППИУ могут архивировать просроченные идентификаторы согласия.

## Дополнительные данные

Ряд ресурсов в спецификации включает в себя раздел для дополнительных данных (SupplementaryData). Данный раздел позволит ППИУ принимать или передавать данные, которые не предусмотрены основной структурой ресурса.

Раздел дополнительных данных определяется как пустой объект JSON в данной спецификации.

Если ППИУ использует структуру с дополнительными данными (SupplementaryData), то он выкладывает детальное описание структуры у себя на портале для разработчиков.

ППИУ не используют структуру SupplementaryData, если добавляемый туда элемент уже существует в текущей версии документа Открытых банковских интерфейсов.

# Безопасность и контроль доступа

## Область действия

Для получения доступа к ресурсам среды Открытых банковских интерфейсов вызов к каждой конечной точке API должен производиться с токеном доступа в авторотационном заголовке HTTP.

Требуемая область действия токена доступа (параметр scope, подробное применение представлено в ФАПИ.СЕК) и тип предоставления доступа должны быть определены в каждой спецификации API.

## Типы предоставления доступа

### Тип доступа по учетным данным OIDC клиента (client credentials)

Тип доступа client credenials - используется для идентификации Стороннего поставщика в контексте, где отсутствует конечный Пользователь и где не требуется его согласие.

Поддержка данного типа доступ осуществляется при аутентификации приложения клиента Стороннего поставщика на сервере авторизации ППИУ с использованием механизма аутентификации private\_key\_jwt (в соответствии с разделом 5.5.3 ФАПИ.СЕК). При этом Сторонний поставщик использует тип разрешения на доступ client\_credentials в сочетании с аssertions и jwt\_bearer.

### Тип доступа с кодом авторизации (authorization code)

Тип доступа authorization code используется в рамках сценария Гибридный поток (в соответствии с разделом 5.4.3 ФАПИ.СЕК и ограничениями, определенными в главе 7). Запрос аутентификации должен выполняться с параметром типа запрашиваемого ответа response\_type равного значению “code id\_token”, запрос к конечной точке токена должен выполняться с типом доступа grant\_type равного значению authorization\_code

### Поток аутентификации по отдельному каналу

В случае, когда Пользователь использует для аутентификации на стороне ППИУ устройство, отличное от того, где он взаимодействовал со Сторонним поставщиком, то такая ситуация в среде Открытых банковских интерфейсах регламентируется ФАПИ.ПАОК с применением ограничений, определенных в главе 7.

## Авторизация согласия

Процесс определения Пользователем разрешений для Стороннего поставщика называется авторизацией согласия.

Сторонний поставщик передаёт намерение Пользователя получить сервис и запрашивает у ППИУ создание согласия на предоставление данного сервиса с помощью доступа client credentials. В зависимости от предоставляемого сервиса, намерение может содержать дополнительную информацию о контексте взаимодействия с Пользователем в разделе Data или Risk полезной нагрузки, что позволяет ППИУ оценивать риски, связанные с намерением Пользователя и принимать решение об необходимости проводить усиленную аутентификация Пользователя. ППИУ отвечает сообщением, которое содержит идентификатор ресурса согласия (consentId). Пользователь проходит аутентификацию на стороне ППИУ и авторизует согласие, что приводит к выдаче Стороннему поставщику от ППИУ токена доступа, связанного с авторизованным согласием.

## Аутентификации Пользователя на стороне ППИУ

Для инициирования аутентификации Пользователя на стороне ППИУ Сторонний поставщик использует запрос аутентификации с типом доступа authorization code (5.4.2.2 ФАПИ.СЕК) или в потоке аутентификации по отдельному каналу (6.3.1 ФАПИ.ПАОК) с передачей объекта запроса. Объект запроса может включать в себя список запрашиваемых свойств (параметр claims), содержащий следующие включения комплексного элемента id\_token:

* **openbanking\_intent\_id** - идентификатор ресурса согласия, в привязке к которому запрашивается авторизация;
* **acr\_values** - запрошенные значения класса контекста аутентификации (заявленное свойство acr токена идентификации). Разделенная пробелами строка, определяющая идентификаторы классов контекста аутентификации, отображаемые в порядке предпочтения, которые сервер авторизации запрашивает для обработки запроса аутентификации:
	+ “urn:rubanking:sca” - идентификатор, определяющий контекст аутентификации, указывающий на необходимость применение усиленной аутентификации Пользователя;
	+ “urn:rubanking:ca” - идентификатор, определяющий контекст аутентификации, указывающий на применение необходимость простой аутентификации Пользователя.

Полный перечень применяемых идентификаторов может определяются на этапе разработке сервера авторизации ППИУ. Используемые значения идентификаторов определяются участниками взаимодействия, использующими данное заявленное свойство и должны однозначно определять методы и факторы аутентификации [ГОСТ Р 58833–2020] (определено в 6.3.1 ФАПИ.ПАОК).

|  |
| --- |
| { "response\_type": "code id\_token", "state": "98d6691382344e7fb03c853739d0a988", "scope": "openid accounts offline\_access", "nonce": "642c0152a40a46bbb82bfda4e0799990", "exp": "1618760589", "max\_age": "86400", "**claims**": { "userinfo": { "openbanking\_intent\_id": { "value": "c7cb828b-5224-470b-ae34-8c956761abf3", "essential": true } }, "**id\_token**": { "openbanking\_intent\_id": { "value": "c7cb828b-5224-470b-ae34-8c956761abf3", "essential": true }, "**acr**": { "values": [ "**urn:rubanking:sca**", "urn:rubanking:ca" ], "essential": true } } }, "aud": "https://sb-as.test.openbankingrussia.ru/sandbox/as/aft", "iss": "4ba3b98a4c6b4731a08bcb91229d1250", "client\_id": "4ba3b98a4c6b4731a08bcb91229d1250", "redirect\_uri": "https://localhost.ru/cb"} |

Таблица 6 – Пример объекта запроса с включением списка запрашиваемых свойств (подробное об объекте запроса см. 5.4.2.2 ФАПИ.СЕК)

На основание объекта запроса ППИУ определяет приемлемый способ аутентификации Пользователя, в том числе необходимость прохождение Пользователем усиленной аутентификации. По завершении аутентификации Пользователя ППИУ извещает Стороннего поставщика об методе, которым был аутентифицирован пользователь в токене идентификации (подробнее в разделе 5.4.2.16 ФАПИ.СЕК и 6.3.1 ФАПИ.ПАОК).

## Требования к токену идентификации (ID Token)

Токен идентификации должен быть сформирован проверен в соответствии с разделом 5.4.2.17 ФАПИ.СЕК. При наличии параметра acr\_values в запросе аутентификации полученный токен идентификации должен содержать значение заявленного свойства acr, определяющее контекст аутентификации, примененный к Пользователю.

## Требования к токену доступа и токену обновления

ППИУ выдает токены доступа и токены обновления в соответствии с требованиями, определенными в разделах 5.4.2.18 и 5.4.2.19 ФАПИ.СЕК. В случае предоставление Пользователем долгосрочного согласия на использование сервиса Стороннего поставщика, Сторонний поставщик использует тип доступа client credentials с токеном обновления.

# Основные сущности комплекса Стандартов

## Справочники и перечисления

Спецификации Открытых банковских интерфейсов содержат поля со справочными данными.

Справочники бывают двух видов:

* фиксированные;
* гибкие.

В случае фиксированных справочников все возможные значения будут задаваться статично в Стандарте Открытых банковских интерфейсов.

В случае гибких справочников начальные возможные значения будут задаваться в Стандарте Открытых банковских интерфейсов, но каждый ППИУ может динамично управлять ими, расширяя своими кастомизированными значениями. Кастомизированные значения, которые используют ППИУ, могут в будущем включаться в список начальных возможных значений для гибких справочников.

И фиксированные, и гибкие справочники находятся в разделе “Справочники и перечисления”.

## Общая структура полезной нагрузки

В этом разделе приводится обзор структуры верхнего уровня для полезных нагрузок Открытых банковских интерфейсов.

Данные, которые содержатся в разделе “Data”, документируются для каждой отдельно взятой конечной точки API.

### Структура запроса

Структура верхнего уровня для запросов Открытых банковских интерфейсов имеет следующий вид:

|  |
| --- |
| { "Data": { ... }, "Risk": { ... }} |

Таблица 7 - Верхнеуровневая структура сообщений для запросов

#### Data

Раздел “Data” содержит данные запроса для конкретного запроса API.

Структура этого элемента отличается для каждой конечной точки API.

#### Risk

Раздел “Risk” содержит индикаторы риска для конкретного запроса API, предоставленного Сторонним поставщиком. Требования к наличию раздела в сообщениях определяется для каждого прикладного стандарта.

Индикаторы риска, содержащиеся в этом элементе, могут отличаться для каждой конечной точки API.



Рисунок 1 - UML -диаграмма для структуры запроса, содержащего поле риск

### Структура ответа

В соответствии с принципом API RESTful полный ресурс воспроизводится как часть ответа.

В ответ включаются следующие дополнительные разделы высокого уровня:

* Links
* Meta

Структура верхнего уровня для ответов Открытых банковских интерфейсов имеет следующий вид:

|  |
| --- |
| {  "Data": {    ...  },  "Risk": {    ...  },  "Links": {    ...  },  "Meta": {    ...  }} |

Таблица 8 - Структура сообщений для ответов



Рисунок 2 - UML -диаграмма для структуры успешного ответа

### Структура ответа с ошибками

Структура для ответов с ошибками Открытых банковских интерфейсов имеет следующий вид:

|  |
| --- |
| {  "code": "...",  "id": "...",  "message": "...",  "Errors": [    {      "errorCode": "...",      "message": "...",      "path": "...",      "url": "..."    }  ]} |

Таблица 9 - Структура сообщений для ответов с ошибками



Рисунок 3 - UML -диаграмма для структуры ответов с ошибками

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Тип | Описание | Кратность |
| code | Max40Text | Высокоуровневый текстовый код ошибки, необходимый для классификации | 1..1 |
| id | Max40Text | Уникальный идентификатор ошибки, для целей аудита, в случае неизвестных / не классифицированных ошибок. | 0..1 |
| message | Max500Text | Краткое сообщение об ошибке. Например, «что-то не так с предоставленными параметрами запроса». | 1..1 |
| Errors | OBRUError | Список кодов ошибок | 1..n |

Таблица 10 - Состав данных объекта OBRUErrorResponse

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Тип | Описание | Кратность |
| errorCode | OBRUErrorResponseErrorCode | Низкоуровневое текстовое описание ошибки в виде кода. Например, RU.SBRF.Field.Missing | 1..1 |
| message | Max500Text | Описание ошибки. Например, 'Обязательное поле не указано'. | 1..1 |
| path | Max500Text | Путь к элементу с ошибкой в JSON объекте. Рекомендуемое, но не обязательное поле. | 0..1 |
| url | URI | URL для помощи в устранении проблемы, Также через URL можно предоставлять дополнительную информацию. | 0..1 |

Таблица 11 - Состав данных объекта OBRUError

### Необязательные поля

В объектах, где значение для необязательного поля не указано, поле исключается из полезной нагрузки JSON.

В объектах, где поле массива определено как имеющее значения кратности 0..n, поле массива включается в полезную нагрузку с пустым массивом. Примеры передачи необязательных полей:

|  |
| --- |
| { "name": "", //Неправильно. Поле "name" нужно исключить из полезной нагрузки. "age": 0, // Неправильно. Значение "0" не используется для указания неопределенного значения. "creditorAccount": {}, // Неправильно. Поле "creditorAccount" нужно исключить. "Balance": [] // Правильно. Таким образом передается пустой массив.} |

### Ссылки

Раздел “Links” является обязательным и всегда будет содержать абсолютные URI для связанных ресурсов.

Ссылка “self” является обязательной.

Пример передачи одинарной ссылки "self":

|  |
| --- |
| "Links": {  "self": "https://api.bank.ru/open-banking/v3.1/payments/58923"} |

При передаче большого количества данных раздел “Links” может также содержать элементы “first”, “prev”, “next” и “last”.

Пример передачи всех элементов раздела Links:

|  |
| --- |
| "Links": {  "self": "http://rocks.ru/articles?page[number]=3&page[size]=25",  "first": "http://rocks.ru/articles?page[number]=1&page[size]=25",  "prev": "http://rocks.ru/articles?page[number]=2&page[size]=25",  "next": "http://rocks.ru/articles?page[number]=4&page[size]=25",  "last": "http://rocks.ru/articles?page[number]=6&page[size]=25"} |

### Метаданные

Раздел “Meta” обязателен, но может быть пустым. Необязательный элемент — “totalPages”, указывает на количество передаваемых страниц. Если передается более одной страницы, то элемент “totalPages” обязательно присутствует.

Пример передачи раздела Meta:

|  |
| --- |
| "Meta": {  "totalPages": 6} |

### Пример использования

Пример структуры полезной нагрузки для многостраничного ответа ППИУ:

|  |
| --- |
| HTTP/1.1 200 OKx-fapi-interaction-id: 11bac543-d5de-3446-b687-880a5018434dContent-Type: application/json { "Data": { ... }, "Links": { "self": "https://bank.ru/open-banking/v1.2/aisp/accounts/98765/transactions/", "last": "https://bank.ru/open-banking/v1.2/aisp/accounts/98765/transactions?pg=6",  "first": "https://bank.ru/open-banking/v1.2/aisp/accounts/98765/transactions/",  "next": "https://bank.ru/open-banking/v1.2/aisp/accounts/98765/transactions?pg=2" }, "Meta": { "totalPages": 6, "firstAvailableDateTime": "2019-05-03T00:00:00+00:00", "lastAvailableDateTime": "2019-12-03T00:00:00+00:00" }} |

1. Определение понятия вводится стандартом «Открытые банковские интерфейсы. Общие положения.» [↑](#footnote-ref-1)
2. Получают доступ к информации с согласия владельца счета. [↑](#footnote-ref-2)
3. Получают доступ к счету с согласия владельца счета. [↑](#footnote-ref-3)