**Практическая работа 5**

**1 Табличная спецификация API**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Resource** | **Operation** | **Method** | **Path** | **Auth** | **Description** |
| collections | Create collection (идемпотентно) | POST | /v1/collections | Bearer | Создать новый сбор документов (если уже создавался с тем же Idempotency-Key — вернуть прежний ответ). |
| collections | Get collection by id | GET | /v1/collections/{collection\_id} | Bearer | Получить сбор по id. |
| collections | List collections (фильтры) | GET | /v1/collections | Bearer | Получение сбора по фильтрам: id, company\_id, contact\_id, counterparty\_id, type, period\_start, period\_end, status, page, limit, sort. |
| collections | Cancel collection | POST | /v1/collections/{collection\_id}/cancel | Bearer | Отмена (soft) сбора — перевод в статус CANCELLED. |
| collection\_documents | List requested documents for client | GET | /v1/collections/{collection\_id}/documents | Bearer | Список документов, которые нужно загрузить по сбору с метаданными |
| documents | Upload file for document | POST | /v1/documents/{document\_id}/files | Bearer | Загружает файл (multipart/form-data). Создаёт ресурс File и возвращает doc\_storage\_id. |
| files | Download file | GET | /v1/files/{file\_id}/download | Bearer | Возвращает перенаправление/Signed URL для скачивания. |
| files | Delete file | DELETE | /v1/files/{file\_id} | Bearer | Удаление файла (soft), проверка прав. |
| documents | Submit uploaded files (submit button) | POST | /v1/collections/{collection\_id}/submit | Bearer | Отправка всех загруженных файлов на верификацию. Идемпотентна в пределах одного collection\_id (если уже отправлено — вернуть статус). |

**1.1 POST /v1/collections — создание сбора документов (идемпотентно)**

Headers

|  |  |  |  |
| --- | --- | --- | --- |
| **Header** | **Req** | **Description** | **Example** |
| Authorization | R | Bearer JWT токен | Bearer eyJhbGci... |
| Idempotency-Key | R | UUID для идемпотентности | 0d9aef2e-3c1b-4d7b-9f77-6c1f1a3a0b0e |

Request body

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Req** | **Rules / Enum** | **Example** |
| company\_id | uuid | R | FK → Company | "c\_123" |
| counterparty\_id | uuid | O | FK → Counterparty | "cp\_555" |
| contact\_id | uuid | O | FK → Contact | "u\_321" |
| type | string | R | Enum: INVOICE\_COLLECTION, CONTRACT\_COLLECTION | "INVOICE\_COLLECTION" |
| documents | array[object] | R | minItems ≥ 1 | – |
| documents[].type | string | R | Enum: INVOICE, ACT, CONTRACT | "INVOICE" |
| documents[].period\_start | string | O | RFC3339/UTC | "2025-01-01T00:00:00Z" |
| documents[].period\_end | string | O | RFC3339/UTC | "2025-01-31T23:59:59Z" |
| documents[].required | boolean | R | true/false | true |
| pushing | string | O | Enum: EMAIL, SMS, PUSH | "EMAIL" |
| expired\_date | string | O | RFC3339/UTC | "2025-10-01T00:00:00Z" |

Пример запроса

|  |
| --- |
| {  "company\_id": "company\_123",  "counterparty\_id": "cp\_789",  "contact\_id": "contact\_456",  "type": "INVOICE\_COLLECTION",  "documents": [  {"type":"INVOICE","period\_start":"2025-01-01T00:00:00Z","period\_end":"2025-01-31T23:59:59Z","required":true}  ],  "pushing":"EMAIL",  "expired\_date":"2025-10-01T00:00:00Z"  } |

Response 201 Created

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Req** | **Rules / Enum** | **Example** |
| id | uuid | R | PK | "5d0e99c1-f0a9-4e2a-9cc7-ef7c7a7df92a" |
| company\_id | uuid | R | FK → Company | "company\_123" |
| type | string | R | Enum | "INVOICE\_COLLECTION" |
| status | string | R | Enum: NEW, IN\_PROGRESS, CANCELLED | "NEW" |
| created\_at | string | R | RFC3339/UTC | "2025-09-01T12:00:00Z" |

Пример ответа

|  |
| --- |
| {  "id":"5d0e99c1-f0a9-4e2a-9cc7-ef7c7a7df92a",  "company\_id":"company\_123",  "type":"INVOICE\_COLLECTION",  "status":"NEW",  "created\_at":"2025-09-01T12:00:00Z"  } |

**1.2 GET /v1/collections — получение списка сборов**

Headers

|  |  |  |  |
| --- | --- | --- | --- |
| **Header** | **Req** | **Description** | **Example** |
| Authorization | R | Bearer JWT токен | Bearer eyJ... |

Query params

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Param** | **Type** | **Req** | **Rules / Enum** | **Example** |
| id | uuid | O | фильтр по id | "5d0e..." |
| company\_id | uuid | O | FK | "company\_123" |
| counterparty\_id | uuid | O | FK | "cp\_456" |
| type | string | O | Enum | "INVOICE\_COLLECTION" |
| status | string | O | Enum | "NEW" |
| period\_start | string | O | RFC3339/UTC | "2025-01-01T00:00:00Z" |
| period\_end | string | O | RFC3339/UTC | "2025-01-31T23:59:59Z" |
| page | int | O | ≥1 | 1 |
| limit | int | O | ≤100 | 20 |
| sort | string | O | `created\_at:asc | desc` |

Response 200 OK

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Req** | **Rules / Enum** | **Example** |
| items | array[object] | R | список сборов | – |
| items[].id | uuid | R | PK | "5d0e..." |
| items[].company\_id | uuid | R | FK | "company\_123" |
| items[].type | string | R | Enum | "INVOICE\_COLLECTION" |
| items[].status | string | R | Enum | "NEW" |
| items[].created\_at | string | R | RFC3339/UTC | "2025-09-01T12:00:00Z" |
| page | int | R | ≥1 | 1 |
| limit | int | R | ≤100 | 20 |
| total | int | R | ≥0 | 42 |

Пример ответа

|  |
| --- |
| {  "items":[  {"id":"5d0e99c1-f0a9-4e2a-9cc7-ef7c7a7df92a","company\_id":"company\_123","type":"INVOICE\_COLLECTION","status":"NEW","created\_at":"2025-09-01T12:00:00Z"}  ],  "page":1,  "limit":20,  "total":1  } |

**1.3 GET /v1/collections/{id} — получение сбора по ID**

Headers

|  |  |  |  |
| --- | --- | --- | --- |
| **Header** | **Req** | **Description** | **Example** |
| Authorization | R | Bearer JWT токен | Bearer eyJ... |

Response 200 OK

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Req** | **Rules / Enum** | **Example** |
| id | uuid | R | PK | "5d0..." |
| company\_id | uuid | R | FK | "company\_123" |
| contact\_id | uuid | O | FK | "u\_1" |
| type | string | R | Enum | "INVOICE\_COLLECTION" |
| status | string | R | Enum: NEW, IN\_PROGRESS, UPLOADED, CANCELLED | "IN\_PROGRESS" |
| documents | array[object] | R | список документов | – |
| documents[].id | uuid | R | PK | "doc\_1" |
| documents[].type | string | R | Enum | "INVOICE" |
| documents[].period\_start | string | O | RFC3339/UTC | "2025-01-01T00:00:00Z" |
| documents[].period\_end | string | O | RFC3339/UTC | "2025-01-31T23:59:59Z" |
| documents[].required | boolean | R | true/false | true |
| documents[].status | string | R | Enum: NEW, UPLOADED, VERIFIED, REJECTED | "UPLOADED" |

Пример ответа

|  |
| --- |
| {  "id":"5d0e99c1-f0a9-4e2a-9cc7-ef7c7a7df92a",  "company\_id":"company\_123",  "type":"INVOICE\_COLLECTION",  "status":"IN\_PROGRESS",  "documents":[  {"id":"doc\_1","type":"INVOICE","period\_start":"2025-01-01T00:00:00Z","period\_end":"2025-01-31T23:59:59Z","required":true,"status":"UPLOADED"}  ]  } |

**1.4 POST /v1/collections/{id}/cancel — отмена сбора**

Headers

|  |  |  |  |
| --- | --- | --- | --- |
| **Header** | **Req** | **Description** | **Example** |
| Authorization | R | Bearer JWT токен | Bearer eyJ... |
| If-Match | O | ETag для конкурентного контроля | "\"v1\_23a5\"" |

Response 200 OK

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Req** | **Rules / Enum** | **Example** |
| id | uuid | R | PK | "5d0..." |
| status | string | R | Enum: CANCELLED | "CANCELLED" |
| updated\_at | string | R | RFC3339/UTC | "2025-09-02T14:00:00Z" |

**1.5 GET /v1/collections/{id}/documents — список документов клиента**

Headers

|  |  |  |  |
| --- | --- | --- | --- |
| **Header** | **Req** | **Description** | **Example** |
| Authorization | R | Bearer JWT токен клиента | Bearer eyJ... |

Response 200 OK

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Req** | **Rules / Enum** | **Example** |
| collection\_id | uuid | R | FK | "col\_123" |
| items | array[object] | R | список документов | – |
| items[].id | uuid | R | PK | "doc\_1" |
| items[].type | string | R | Enum | "INVOICE" |
| items[].required | boolean | R | true/false | true |
| items[].status | string | R | Enum | "UPLOADED" |
| items[].uploaded\_files | array[object] | O | — | – |
| items[].uploaded\_files[].file\_id | uuid | R | FK | "f\_1" |
| items[].uploaded\_files[].file\_name | string | R | — | "invoice.pdf" |
| items[].uploaded\_files[].size | int | R | bytes | 23122 |

**1.6 POST /v1/documents/{id}/files — загрузка файла**

Headers

|  |  |  |  |
| --- | --- | --- | --- |
| **Header** | **Req** | **Description** | **Example** |
| Authorization | R | Bearer JWT токен | Bearer eyJ... |
| Idempotency-Key | R | UUID для идемпотентности | "8ab3..." |

Form-data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Req** | **Rules / Enum** | **Example** |
| file | binary | R | MIME: pdf, png, jpg ≤50MB | (файл) |
| uploader\_id | uuid | R | FK → Contact | "u\_456" |

Response 201 Created

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Req** | **Rules / Enum** | **Example** |
| file\_id | uuid | R | PK | "f\_1" |
| document\_id | uuid | R | FK | "d\_1" |
| file\_name | string | R | – | "invoice2025.pdf" |
| size | int | R | bytes | 34215 |
| status | string | R | Enum: UPLOADED | "UPLOADED" |
| uploaded\_at | string | R | RFC3339/UTC | "2025-09-10T12:00:00Z" |

**1.7 POST /v1/collections/{id}/submit — отправка файлов (submit)**

Headers

|  |  |  |  |
| --- | --- | --- | --- |
| **Header** | **Req** | **Description** | **Example** |
| Authorization | R | Bearer JWT токен | Bearer eyJ... |
| Idempotency-Key | R | UUID | "submit-abc" |

Response 202 Accepted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Req** | **Rules / Enum** | **Example** |
| collection\_id | uuid | R | FK | "col\_123" |
| verify\_task\_id | uuid | R | FK → VerifyTask | "vt\_456" |
| status | string | R | Enum: SENT\_FOR\_VERIFICATION | "SENT\_FOR\_VERIFICATION" |
| submitted\_at | string | R | RFC3339/UTC | "2025-09-10T12:00:00Z" |

**1.8 GET /v1/files/{file\_id}/download — скачивание файла**

Headers

|  |  |  |  |
| --- | --- | --- | --- |
| **Header** | **Req** | **Description** | **Example** |
| Authorization | R | Bearer JWT токен | Bearer eyJ... |

Response 302 Found

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Type | Req | Rules / Enum | Example |
| location | string | R | Signed URL, TTL ≤60s | "https://storage.example.com/file/123?sig=abc" |

Пример ответа

https://storage.example.com/file/123?sig=abc

**1.9 DELETE /v1/files/{file\_id} — удаление файла**

Headers

|  |  |  |  |
| --- | --- | --- | --- |
| **Header** | **Req** | **Description** | **Example** |
| Authorization | R | Bearer JWT токен | Bearer eyJ... |

Response 204 No Content

**1.10 Норматив по кодам и ошибкам**

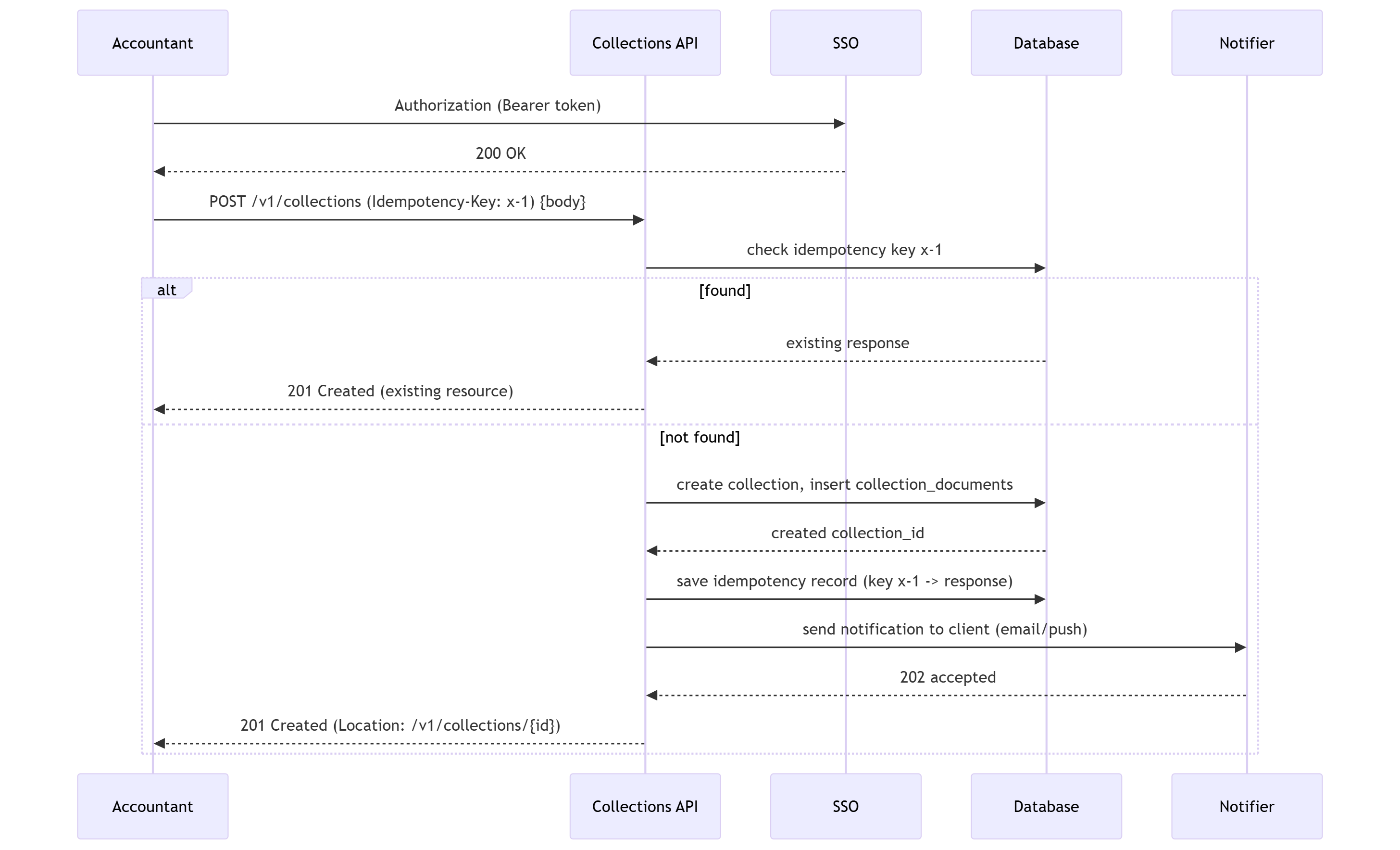
|  |  |  |
| --- | --- | --- |
| **Code** | **When** | **Body** |
| 200 | успешное чтение | объект/массив |
| 201 | создан ресурс | Location + объект |
| 202 | принято в обработку | {status:"accepted"} |
| 204 | удалено / нет тела | – |
| 400 | неверный запрос | {code:"invalid", field:"…", message:"…"} |
| 401 | нет/некорректен токен | {code:"unauthorized"} |
| 403 | нет прав | {code:"forbidden"} |
| 404 | не найдено | {code:"not\_found"} |
| 409 | конфликт версий / идемпотентности | {code:"conflict"} |
| 422 | ошибка доменной валидации | {code:"invalid\_field", field:"…", message:"…"} |
| 429 | превышен лимит | Retry-After header |
| 5xx | сбой сервера | {code:"server\_error", traceId:"..."} |

**Единый формат ошибки:**

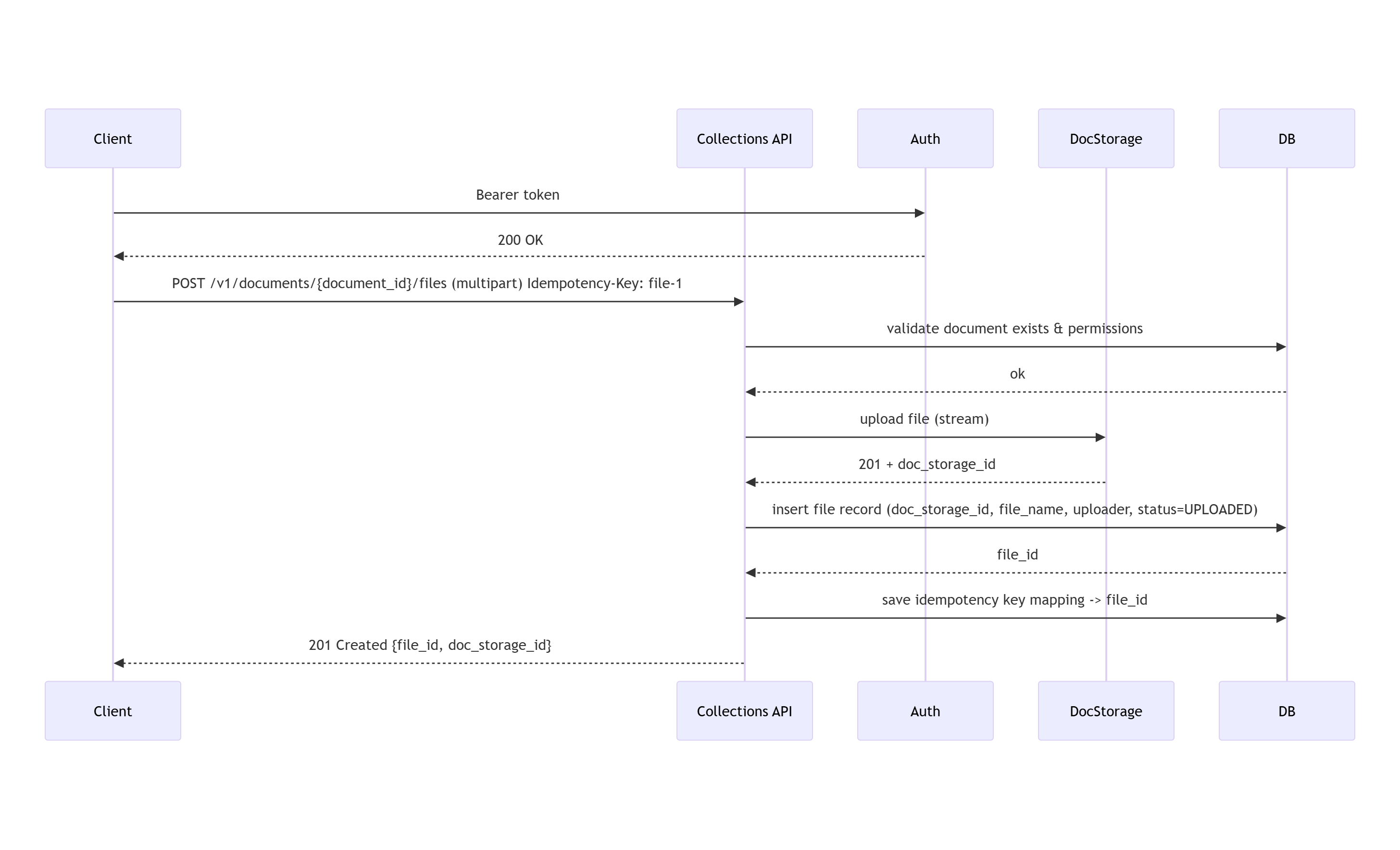
|  |
| --- |
| {  "code": "string",  "message": "string",  "field": "string|null",  "traceId": "uuid"  } |

**2. Диаграммы последовательностей — 3 сценария**

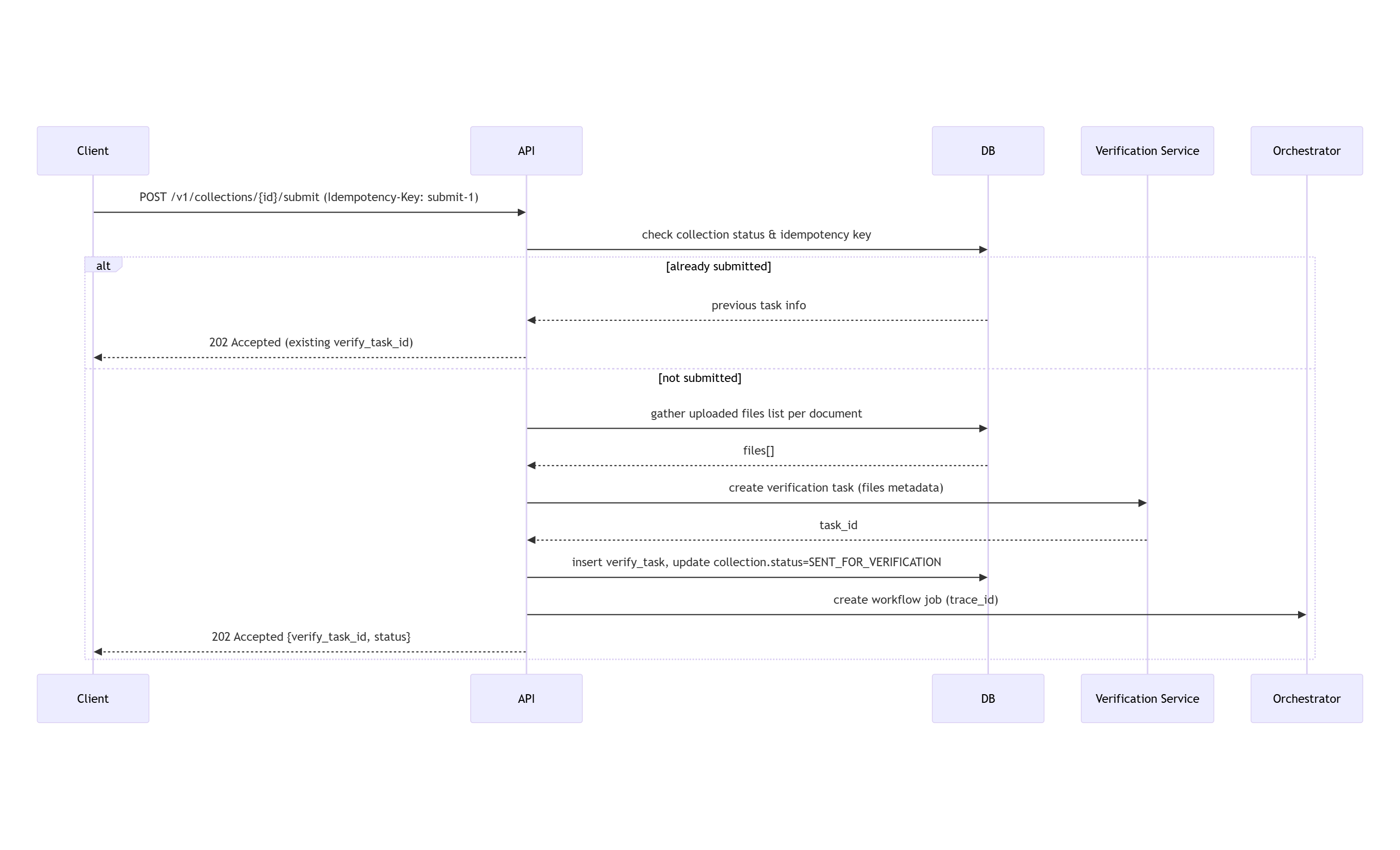
**2.1 Sequence: Create collection**

****

**2.2 Sequence: Upload file for document**

****

**2.3 Sequence: Submit files**

****

**3. Таблица надежности**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Operation** | **Safe?** | **Идемпотентна?** | **Key/Scope** | **Retry when** | **Backoff** | **Client timeout** | **Server timeout** |
| GET /v1/collections | yes | yes | — | 5xx | none/short | 2s | 1.5s |
| POST /v1/collections | no | yes | Idempotency-Key per client | 408/5xx, | exponential (100ms → 200ms → 400ms) | 30s (p95 ≤ 30s NFR) | 20s |
| POST /v1/documents/{id}/files | no | yes | Idempotency-Key per file upload | 5xx | exponential | 30s | 25s |
| POST /v1/collections/{id}/submit | no | yes | Idempotency-Key unique per collection | 5xx | linear/exponential | 10s | 8s |
| POST /v1/collections/{id}/cancel | no | yes (If-Match) | If-Match (ETag) | 5xx | linear (200ms) | 5s | 4s |
| DELETE /v1/files/{id} | no | yes | — | 5xx | quick retries | 5s | 4s |

Если клиент повторил запрос с тем же Idempotency-Key и тело запроса эквивалентно (проверка сигнатуры/хеша), сервер возвращает исходный результат (тот же resource\_id/код). Если тело отличается — вернуть 409 Conflict.

**7. Политика версионирования и совместимость (контракт эволюции)**

Стратегия версий: URL версионирование: /v{major}/... — простая и явная стратегия.

Совместимые (backward-compatible) изменения (minor/patch):

* Добавление новых необязательных полей в response.
* Добавление новых enum-значений, если клиенты игнорируют незнакомые значения.
* Добавление новых endpoints.

Ломающие изменения (require major version):

* Удаление/переименование обязательных полей.
* Изменение типов существующих полей (string→object).
* Изменение семантики статусов (например, status meaning change).
* Изменения формата ошибок.

Deprecation process:

1. Внести новую версию (v2). Поддерживать v1 минимум 90 дней (или SLA согласованное).
2. Пометить Deprecation заголовком в ответах: Deprecation: true, Sunset: <date> и публиковать changelog.
3. Нотификация клиентов (email, webhook, changelog).
4. После Sunset — отключение.

**Матрица совместимости:**

|  |  |  |
| --- | --- | --- |
| **Client \ Server** | **v1** | **v2** |
| v1 | ✅ full | ❌ |
| v2 | ⚠️ только чтение старых данных | ✅ full |

**4 Аутентификация / авторизация / безопасность**

* Межсервисная авторизация: Bearer JWT.
* Клиентская авторизация: Bearer JWT (SSO).
* Rate-limit: 100 req/min. Return 429 + Retry-After.
* ETag / If-Match для конкурентных обновлений.
* Валидация загружаемых файлов: проверка расширения, mime-type, max size (например 50MB), сканирование на вирусы, проверка содержимого ZIP.
* Audit logs: 100% критичных операций (create collection, upload file, submit) — писать trace\_id в ответе.