|  |
| --- |
| Общество с ограниченной ответственностью «Высокие технологии» |

Инструкция по написанию адаптеров

Москва,

2024 г.

Содержание

[Основные функции адаптера 3](#_Toc165894493)

[Дополнительная функция адаптера 3](#_Toc165894494)

[Приложение А 5](#_Toc165894495)

[Описание API Файлового сервиса 5](#_Toc165894496)

[1. Назначение сервиса 5](#_Toc165894497)

[2. Инфраструктурные зависимости 5](#_Toc165894498)

[3. REST-взаимодействие 5](#_Toc165894499)

[3.1. Загрузка файлов 5](#_Toc165894500)

[3.2. Получение файла 9](#_Toc165894501)

[3.2.1. Получить сам файл 9](#_Toc165894502)

[3.2.2. Получить временную ссылку на файл 9](#_Toc165894503)

[3.2.3. Редирект на файл в файловом хранилище 10](#_Toc165894504)

[3.2.4. Метаданные по файлу 11](#_Toc165894505)

[3.3. Удаление файлов 12](#_Toc165894506)

[3.3.1. Удаление одного файла 12](#_Toc165894507)

[3.3.2. Удаление группы файлов 12](#_Toc165894508)

[3.3.3. Возможные ошибки 12](#_Toc165894509)

[4. Параметры 13](#_Toc165894510)

[Приложение Б 15](#_Toc165894511)

[Описание API Сервиса интеграций 15](#_Toc165894512)

[1. Назначение сервиса 15](#_Toc165894513)

[2. Инфраструктурные зависимости 15](#_Toc165894514)

[3. REST-взаимодействие 15](#_Toc165894515)

[3.1. Входящий запрос 15](#_Toc165894516)

[3.1.1. Добавление входящего запроса в систему 15](#_Toc165894517)

[3.1.2. Получение текущего состояния входящего запроса 18](#_Toc165894518)

[3.1.3. Получение ответа 19](#_Toc165894519)

[3.1.4. Подтверждение о получении ответа отправителем 21](#_Toc165894520)

[3.1.5. Отправка статусов и ответов на ранее полученный запрос 22](#_Toc165894521)

[3.2. Передача исходящего запроса адаптеру 25](#_Toc165894522)

## Основные функции адаптера

Основные функции адаптера включают в себя:

1. Функция на вход: принимать запросы от системы-отправителя и приводить всю информацию запроса к виду, в котором КСМЭВ осуществляет дальнейшую передачу этой информации.

2. Функция на выход: получать данные от Сервиса интеграций и приводить их к виду, в котором система-получатель их принимает, формировать соответствующий запрос и передавать его.

На вход:

1. принять запрос от системы-отправителя;
2. сохранить контент запроса с помощью Файлового сервиса (см.п 3.1 Приложения А) в объектном S3-хранилище и получить objectId;
3. передать POST-запросом в Сервис интеграций метаданные запроса включая objectId (см. п 3.1.1Приложения Б)

На выход:

1. принять запрос с метаданными от Сервиса интеграций: техатрибуты запроса + objectId (см. п 3.2 Приложения Б);
2. GET-запросом в Файловый сервис по objectId получить контент (см. п 3.2 Приложения А);
3. сформировать запрос для системы-получателя и отправить его системе-получателю

## Дополнительная функция адаптера

Дополнительная функция адаптера – возможность подключения к Монитору Запросов.

Если в создаваемом адаптере будет реализована трассировка запросов, то для подключения к Монитору Запросов необходимо организовать отправку в его топик monitor.requests.event.new сообщений c хидером (KafkaHeaders.KEY) "correlationId": “trace-id”, следующего вида:

|  |
| --- |
| { "httpMethod": "GET", "url": "http://localhost:8080/foo?bar=baz", "requestHeaders": { "Content-Type": [ "text/xml" ] }, "requestObjectId": "requestObjectId", " statusCode ": 200, "responseHeaders": { "Content-Type": [ "text/xml" ] }, "responseObjectId": "responseObjectId", "timestamp": null, "applicationId": "applicationId", "uuid": "115aa098-9277-0087-49a8-00cb901fc2f7", "flowId": "flowId", "operation": "REQUEST"} |

| **Parameter** | **Description** |
| --- | --- |
| httpMethod | Метод запроса (GET, POST и т.д.) |
| url | url куда был отправлен запрос или откуда был получен |
| requestHeaders | Заголовки запроса |
| requestObjectId | ObjectId сохраненного запроса в S3 хранилище |
| statusCode | Статус запроса |
| responseHeaders | Заголовки ответа |
| responseObjectId | ObjectId сохраненного запроса в S3 хранилище |
| timestamp | Когда был отправлен или получен запрос от внешней системы |
| applicationId | Имя адаптера |
| uuid | Сгенерированный идентификатор  |
| flowId | Id цепочки запросов |
| operation | Тип операции, для адаптеров всегда "REQUEST" |

## Приложение А

## Описание API Файлового сервиса

## Назначение сервиса

Сервис Файлового Хранилища предоставляет возможность хранения файлов.

Работа с данным сервисом происходит через REST-методы.

## Инфраструктурные зависимости

S3 - файловое хранилище.

## REST-взаимодействие

## Загрузка файлов

Загрузка файла происходит:

1. с использованием Multipart.

Пример запроса:

| *Table 1. /files/{bucketName}/{dirName}* |
| --- |
| **Parameter** | **Description** |
| bucketName | Наименование рабочей корзины |
| dirName | Наименование папки, в которую кладется файл (не обязательный параметр) |

#### HTTP request

POST /files/dev-bucket/ HTTP/1.1

Content-Type: multipart/form-data;charset=UTF-8; boundary=6o2knFse3p53ty9dmcQvWAIx1zInP11uCfbm

Host: localhost:8080

--6o2knFse3p53ty9dmcQvWAIx1zInP11uCfbm

Content-Disposition: form-data; name=file; filename=file.txt

Content-Type: text/plain

Байты вложенного файла

--6o2knFse3p53ty9dmcQvWAIx1zInP11uCfbm--

#### HTTP request

POST /files/dev-bucket/directory-name HTTP/1.1

Content-Type: multipart/form-data;charset=UTF-8; boundary=6o2knFse3p53ty9dmcQvWAIx1zInP11uCfbm

Host: localhost:8080

--6o2knFse3p53ty9dmcQvWAIx1zInP11uCfbm

Content-Disposition: form-data; name=file; filename=file.txt

Content-Type: text/plain

Байты вложенного файла

--6o2knFse3p53ty9dmcQvWAIx1zInP11uCfbm--

1. побайтово.

Пример запроса:

| *Table 2. /files/{bucketName}/{dirName}/{filename}* |
| --- |
| **Parameter** | **Description** |
| bucketName | Наименование рабочей корзины |
| dirName | Наименование папки, в которую кладется файл (не обязательный параметр) |
| filename | Наименование загружаемого файла |

#### HTTP request

POST /files/dev-bucket/file.txt HTTP/1.1

Content-Length: 42

Host: localhost:8080

Байты вложенного файла

#### HTTP request

POST /files/dev-bucket/directory-name/file.txt HTTP/1.1

Content-Length: 42

Host: localhost:8080

Байты вложенного файла

1. В ответ приходит информация о загруженном файле.

Пример ответа:

#### HTTP response

HTTP/1.1 200 OK

Content-Type: application/json

Content-Length: 185

{

 "fileName" : "file.txt",

 "size" : 42,

 "objectId" : "2022/07/directory-name/07ecd2fa-4a06-11ec-81d3-0242ac130003.txt",

 "contentType" : "text/plain",

 "bucket" : "dev-bucket"

}

#### Response fields

| **Path** | **Type** | **Description** |
| --- | --- | --- |
| fileName | String | Наименование файла |
| size | Number | Размер файла в байтах |
| objectId | String | Внутреннее наименование файла |
| contentType | String | Тип данных |
| bucket | String | Наименование корзины, в которую положен файл |

1. Файлы можно загрузить группой с последующей архивацией в ZIP-архив.

Пример запроса:

| *Table 3. /files/{bucketName}/zip/{dirName}/{filename}* |
| --- |
| **Parameter** | **Description** |
| bucketName | Наименование рабочей корзины |
| dirName | Наименование папки, в которую кладется файл (не обязательный параметр) |
| filename | Наименование итогового архива |

#### HTTP request

POST /files/dev-bucket/zip/archive.zip HTTP/1.1

Content-Type: multipart/form-data;charset=UTF-8; boundary=6o2knFse3p53ty9dmcQvWAIx1zInP11uCfbm

Host: localhost:8080

--6o2knFse3p53ty9dmcQvWAIx1zInP11uCfbm

Content-Disposition: form-data; name=data; filename=file1.txt

Content-Type: text/plain

Байты вложенного файла

--6o2knFse3p53ty9dmcQvWAIx1zInP11uCfbm

Content-Disposition: form-data; name=data; filename=file2.txt

Content-Type: text/plain

Байты вложенного файла

--6o2knFse3p53ty9dmcQvWAIx1zInP11uCfbm--

#### HTTP request

POST /files/dev-bucket/zip/directory-name/archive.zip HTTP/1.1

Content-Type: multipart/form-data;charset=UTF-8; boundary=6o2knFse3p53ty9dmcQvWAIx1zInP11uCfbm

Host: localhost:8080

--6o2knFse3p53ty9dmcQvWAIx1zInP11uCfbm

Content-Disposition: form-data; name=data; filename=file1.txt

Content-Type: text/plain

Байты вложенного файла

--6o2knFse3p53ty9dmcQvWAIx1zInP11uCfbm

Content-Disposition: form-data; name=data; filename=file2.txt

Content-Type: text/plain

Байты вложенного файла

--6o2knFse3p53ty9dmcQvWAIx1zInP11uCfbm--

#### HTTP response

HTTP/1.1 200 OK

Content-Disposition: inline;filename=f.txt

Content-Type: application/json

Content-Length: 194

{

 "fileName" : "archive.zip",

 "size" : 326,

 "objectId" : "2022/07/directory-name/07ecd2fa-4a06-11ec-81d3-0242ac130003.zip",

 "contentType" : "application/zip",

 "bucket" : "dev-bucket"

}

#### Response fields

| **Path** | **Type** | **Description** |
| --- | --- | --- |
| fileName | String | Наименование файла |
| size | Number | Размер файла в байтах |
| objectId | String | Внутреннее наименование файла |
| contentType | String | Тип данных |
| bucket | String | Наименование корзины, в которую положен файл |

## Получение файла

### Получить сам файл

Пример запроса:

###### **Path parameters**

| *Table 4. /files/{bucketName}/{objectId}* |
| --- |
| **Parameter** | **Description** |
| bucketName | Наименование рабочей корзины |
| objectId | Внутреннее наименование файла |

###### **HTTP request**

GET /files/dev-bucket/2022/02/optional-directory-name/file.txt?format=binary HTTP/1.1

Host: localhost:8080

###### **HTTP response**

HTTP/1.1 200 OK

Content-Length: 42

Content-Disposition: attachment; filename="file.txt"

Content-Type: text/plain

Байты вложенного файла

### Получить временную ссылку на файл

Пример запроса:

###### **Path parameters**

| *Table 4. /files/{bucketName}/{objectId}* |
| --- |
| **Parameter** | **Description** |
| bucketName | Наименование рабочей корзины |
| objectId | Внутреннее наименование файла |

###### **HTTP request**

GET /files/dev-bucket/2022/02/optional-directory-name/file.txt?format=url HTTP/1.1

Host: localhost:8080

###### **HTTP response**

HTTP/1.1 200 OK

Content-Type: text/plain;charset=UTF-8

Content-Length: 610

http://localhost:9000/dev-bucket/2022/02/optional-directory-name/file.txt?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=CDK5PDJSN251BGEBRYMA%2F20220210%2Fus-east-1%2Fs3%2Faws4\_request&X-Amz-Date=20220210T111216Z&X-Amz-Expires=604799&X-Amz-Security-Token=eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.eyJhY2Nlc3NLZXkiOiJDREs1UERKU04yNTFCR0VCUllNQSIsImV4cCI6MzYwMDAwMDAwMDAwMCwicG9saWN5IjoiY29uc29sZUFkbWluIn0.ojMEcx9OnATG-RfDYr3zGo5Ob89ubCcDQJ6YxG3mRpOptAiZNr36WyNxRvGAAeXdrhfJ2mtMEb-4JPWTFlEozw&X-Amz-SignedHeaders=host&versionId=null&X-Amz-Signature=2ec38d940c9034f38d25ebe0315f2283a8da1f173902cdc4d31c7de11628a4b6

localhost:9000 - само файловое хранилище S3, в котором находится файл

### Редирект на файл в файловом хранилище

Пример запроса:

###### **Path parameters**

| *Table 4. /files/{bucketName}/{objectId}* |
| --- |
| **Parameter** | **Description** |
| bucketName | Наименование рабочей корзины |
| objectId | Внутреннее наименование файла |

###### **HTTP request**

GET /files/dev-bucket/2022/02/optional-directory-name/file.txt HTTP/1.1

Host: localhost:8080

###### **HTTP response**

HTTP/1.1 301 Moved Permanently

Location: http://localhost:9000/dev-bucket/2022/02/optional-directory-name/file.txt

###### **Response headers**

| **Name** | **Description** |
| --- | --- |
| Location | Путь до файла в хранилище |

localhost:9000 - само файловое хранилище S3, в котором находится файл

### Метаданные по файлу

Пример запроса:

###### **Path parameters**

| *Table 4. /files/{bucketName}/{objectId}* |
| --- |
| **Parameter** | **Description** |
| bucketName | Наименование рабочей корзины |
| objectId | Внутреннее наименование файла |

###### **HTTP request**

GET /files/dev-bucket/2022/02/optional-directory-name/file.txt?format=meta HTTP/1.1

Host: localhost:8080

###### **HTTP response**

HTTP/1.1 200 OK

Content-Type: application/json

Content-Length: 185

{

 "name" : "2022/02/optional-directory-name/file.txt",

 "createdTime" : "2022-02-10T14:40:00+05:00",

 "length" : 64,

 "etag" : "3459y234523458372",

 "contentType" : "text/plain"

}

###### **Response fields**

| **Path** | **Type** | **Description** |
| --- | --- | --- |
| name | String | Путь до файла |
| createdTime | String | Дата создания |
| length | Number | Размер файла в байтах |
| contentType | String | Тип данных |
| etag | String |  |

## Удаление файлов

### Удаление одного файла

Пример запроса:

###### **Path parameters**

| *Table 4. /files/{bucketName}/{objectId}* |
| --- |
| **Parameter** | **Description** |
| bucketName | Наименование рабочей корзины |
| objectId | Внутреннее наименование удаляемого файла |

###### **HTTP request**

DELETE /files/dev-bucket/2022/02/optional-directory-name/file.txt HTTP/1.1

Host: localhost:8080

###### **HTTP response**

HTTP/1.1 200 OK

### Удаление группы файлов

В запросе передаётся массив objectId. Не рекомендуется передавать больше 1000 элементов.

Пример запроса:

###### **Path parameters**

| *Table 4. /files/{bucketName}/* |
| --- |
| **Parameter** | **Description** |
| bucketName | Наименование рабочей корзины |

###### **HTTP request**

DELETE /files/dev-bucket/ HTTP/1.1

Content-Type: application/json;charset=UTF-8

Content-Length: 69

Host: localhost:8080

[ "2022/02/file1.txt", "2022/02/optional-dictionary-name/file2.txt" ]

###### **HTTP response**

HTTP/1.1 200 OK

### Возможные ошибки

Неизвестное наименование корзины

Для примера взят запрос загрузки файла используя Multipart

**HTTP request**

POST /files/wrong-bucket-name/dictionary-name HTTP/1.1

Content-Type: multipart/form-data;charset=UTF-8; boundary=6o2knFse3p53ty9dmcQvWAIx1zInP11uCfbm

Host: localhost:8080

--6o2knFse3p53ty9dmcQvWAIx1zInP11uCfbm

Content-Disposition: form-data; name=file; filename=file.txt

Content-Type: text/plain

Байты вложенного файла

--6o2knFse3p53ty9dmcQvWAIx1zInP11uCfbm--

**HTTP response**

HTTP/1.1 404 Not Found

Content-Type: application/json

Content-Length: 98

{

 "errorCode" : "BUCKET\_NOT\_SUPPORTED",

 "message" : "Unknown bucket name: wrong-bucket-name"

}

| *Table 4. Параметры ответа с ошибкой* |
| --- |
| **Path** | **Type** | **Description** |
| errorCode | String | Код ошибки |
| message | String | Подробности ошибки |

## Параметры

| **Наименование** | **Описание** |
| --- | --- |
| s3.buckets | Наименования рабочих корзин в хранилищеУ наименования корзин есть свои правила:- не пустое;- длина от 3 символов до 63;- подряд не должно стоять 2 точки ("..");- имя должно быть совместимо с DNS, т.е. должно содержать английские буквы (a-z), цифры и в середине разрешены точка(.) и тире(-) |
| s3.credentials.access-key | Ключ доступа для работы с файловым хранилищем S3 |
| s3.credentials.secret-key | Секретный ключ для работы с файловым хранилищем S3 |
| s3.enable-bucket-auto-creation | Автоматическое создание недостающих корзин |
| s3.endpoint.url | Ссылка на файловое хранилище S3 |
| s3.endpoint.connect-timeout | Тайм-аут для соединения с файловым хранилищем (в мс). По-умолчанию 10 с. |
| s3.link-expiry-time | Срок действия временной ссылки на файл в хранилище. Минимальный срок - 1 секунда, максимальный - 7 дней. |
| s3.part-size | Размер части загружаемых данных (в Мб). Минимально - 5, максимально - 5120 (5Гб).По-умолчанию 10 Мб. |

## Приложение Б

## Описание API Сервиса интеграций

## Назначение сервиса

Данный сервис предназначен для интеграции настраиваемого взаимодействия различных систем друг с другом. Для технических специалистов доступен интерфейс с настройками, журналом запросов, подробной информацией о конкретном запросе.

## Инфраструктурные зависимости

Сервис использует PostgreSQL для хранения информации о запросах и их настройках.

## REST-взаимодействие

## Входящий запрос

### Добавление входящего запроса в систему

POST запрос на добавление запроса в систему

Пример запроса:

###### **HTTP request**

POST /requests HTTP/1.1

Content-Type: application/json;charset=UTF-8

Content-Length: 394

Host: localhost:8080

{

 "request" : {

 "integrationId" : "ngs",

 "externalId" : "SMEV3\_MVD\_134324",

 "senderId" : "poltava-4",

 "contentFileId" : "poltava-bucket/42",

 "attributes" : null,

 "techAttributes" : null,

 "attachmentsFileIds" : [ "poltava-bucket/poltava-attachment" ],

 "responseUrl" : "http://localhost:8040/ngs/responses",

 "statusUrl" : "http://localhost:8040/status"

 }

}

###### **Request fields**

| **Path** | **Type** | **Description** |
| --- | --- | --- |
| request | Object | Запрос для отправки во внешнюю систему |
| request.integrationId | String | Внешний идентификатор интеграции |
| request.externalId | String | Идентификатор запроса в системе отправителя |
| request.senderId | String | Идентификатор системы отправителя |
| request.statusUrl | String | Callback (webhook) URL для возврата статусов |
| request.responseUrl | String | Callback (webhook) URL для возврата результата |
| request.attributes | Object | Дополнительные параметры запроса |
| request.techAttributes | Object | Дополнительные технические параметры отправителя |
| request.contentFileId | String | Контент запроса в файловом сервисе |
| request.attachmentsFileIds | Array | Список вложений запроса |

###### **HTTP response**

HTTP/1.1 200 OK

X-Content-Type-Options: nosniff

X-XSS-Protection: 0

Cache-Control: no-cache, no-store, max-age=0, must-revalidate

Pragma: no-cache

Expires: 0

Запрос с неизвестным отправителем

###### **HTTP request**

POST /requests HTTP/1.1

Content-Type: application/json;charset=UTF-8

Content-Length: 389

Host: localhost:8080

{

 "request" : {

 "integrationId" : "ngs",

 "externalId" : "SMEV3\_MVD\_134324",

 "senderId" : "RPGU",

 "contentFileId" : "poltava-bucket/42",

 "attributes" : null,

 "techAttributes" : null,

 "attachmentsFileIds" : [ "poltava-bucket/poltava-attachment" ],

 "responseUrl" : "http://localhost:8040/ngs/responses",

 "statusUrl" : "http://localhost:8040/status"

 }

}

###### **HTTP response**

HTTP/1.1 422 Unprocessable Entity

Content-Type: application/json

X-Content-Type-Options: nosniff

X-XSS-Protection: 0

Cache-Control: no-cache, no-store, max-age=0, must-revalidate

Pragma: no-cache

Expires: 0

Content-Length: 82

{

 "errorCode" : "PARTICIPANT\_NOT\_FOUND",

 "message" : "participant.not.found"

}

###### **Response fields**

| **Path** | **Type** | **Description** |
| --- | --- | --- |
| errorCode | String | Код ошибки |
| message | String | Подробности ошибки |

Запрос с неизвестным IntegrationId

###### **HTTP request**

POST /requests HTTP/1.1

Content-Type: application/json;charset=UTF-8

Content-Length: 405

Host: localhost:8080

{

 "request" : {

 "integrationId" : "getTardisCount",

 "externalId" : "SMEV3\_MVD\_134324",

 "senderId" : "poltava-4",

 "contentFileId" : "poltava-bucket/42",

 "attributes" : null,

 "techAttributes" : null,

 "attachmentsFileIds" : [ "poltava-bucket/poltava-attachment" ],

 "responseUrl" : "http://localhost:8040/ngs/responses",

 "statusUrl" : "http://localhost:8040/status"

 }

}

###### **HTTP response**

HTTP/1.1 500 Internal Server Error

Content-Type: application/json

X-Content-Type-Options: nosniff

X-XSS-Protection: 0

Cache-Control: no-cache, no-store, max-age=0, must-revalidate

Pragma: no-cache

Expires: 0

Content-Length: 104

{

 "errorCode" : "BAD\_INTEGRATION\_ID",

 "message" : "Integration with id [getTardisCount] not found"

}

###### **Response fields**

| **Path** | **Type** | **Description** |
| --- | --- | --- |
| errorCode | String | Код ошибки |
| message | String | Подробности ошибки |

### Получение текущего состояния входящего запроса

GET запрос для получения текущего состояния ранее отправленного запроса

Пример получения текущего состояния:

###### **HTTP request**

GET /requests/state/poltava-4/SMEV3\_MVD\_134324 HTTP/1.1

Host: localhost:8080

###### **Path parameters**

| *Table 1. /requests/state/{senderId}/{externalId}* |
| --- |
| **Parameter** | **Description** |
| externalId | Номер запроса во внешней системе |
| senderId | Идентификатор внешней системы |

###### **HTTP response**

HTTP/1.1 200 OK

Content-Type: application/json

X-Content-Type-Options: nosniff

X-XSS-Protection: 0

Cache-Control: no-cache, no-store, max-age=0, must-revalidate

Pragma: no-cache

Expires: 0

Content-Length: 188

{

 "id" : 11,

 "status" : "SENT",

 "created" : "2021-08-03T08:56:00",

 "updated" : "2021-08-03T10:56:00",

 "received" : null,

 "sent" : "2021-08-03T10:56:00",

 "attributes" : null

}

###### **Response fields**

| **Path** | **Type** | **Description** |
| --- | --- | --- |
| id | Number | Идентификатор запроса в сервисе интеграций |
| status | String | Текущий статус запроса |
| created | String | Дата и время создания запроса |
| updated | String | Дата и время последнего изменения запроса |
| received | String | Дата и время получения ответа на запроса |
| sent | String | Дата и время отправки запроса |
| attributes | Object | Дополнительные атрибуты запроса |

Получение данных несуществующего запроса

###### **HTTP request**

GET /requests/state/poltava-4/SMEV3\_MVD\_134324 HTTP/1.1

Host: localhost:8080

###### **HTTP response**

HTTP/1.1 404 Not Found

Content-Type: application/json

X-Content-Type-Options: nosniff

X-XSS-Protection: 0

Cache-Control: no-cache, no-store, max-age=0, must-revalidate

Pragma: no-cache

Expires: 0

Content-Length: 74

{

 "errorCode" : "REQUEST\_NOT\_FOUND",

 "message" : "Request not found"

}

###### **Response fields**

| **Path** | **Type** | **Description** |
| --- | --- | --- |
| errorCode | String | Код ошибки |
| message | String | Подробности ошибки |

### Получение ответа

GET запрос для получения ответа на ранее отправленный запрос

Пример запроса:

###### **HTTP request**

GET /responses HTTP/1.1

Content-Type: application/json;charset=UTF-8

Content-Length: 118

Host: localhost:8080

{

 "senderId" : "poltava-4",

 "techAttributes" : {

 "externalUUID" : "db0486d0-3c08-11e5-95e2-d4c9eff07b78"

 }

}

###### **Request fields**

| **Path** | **Type** | **Description** |
| --- | --- | --- |
| senderId | String | Идентификатор системы отправителя |
| techAttributes | Object | Технические параметры запроса, используются для идентификации |

###### **HTTP response**

HTTP/1.1 200 OK

Content-Type: application/json

X-Content-Type-Options: nosniff

X-XSS-Protection: 0

Cache-Control: no-cache, no-store, max-age=0, must-revalidate

Pragma: no-cache

Expires: 0

Content-Length: 63

{

 "contentFileId" : "poltava-bucket/42",

 "requestId" : 15

}

###### **Response fields**

| **Path** | **Type** | **Description** |
| --- | --- | --- |
| contentFileId | String | Контент тела ответа в файловом сервисе |
| requestId | Number | Идентификатор запроса в сервисе интеграций |

Возможные ошибки:

Ответ не готов

###### **HTTP response**

HTTP/1.1 204 No Content

Content-Type: application/json

X-Content-Type-Options: nosniff

X-XSS-Protection: 0

Cache-Control: no-cache, no-store, max-age=0, must-revalidate

Pragma: no-cache

Expires: 0

Content-Length: 82

{

 "errorCode" : "RESPONSE\_IS\_NOT\_READY",

 "message" : "response.is.not.ready"

}

Запрос не найден

###### **HTTP response**

HTTP/1.1 404 Not Found

Content-Type: application/json

X-Content-Type-Options: nosniff

X-XSS-Protection: 0

Cache-Control: no-cache, no-store, max-age=0, must-revalidate

Pragma: no-cache

Expires: 0

Content-Length: 74

{

 "errorCode" : "REQUEST\_NOT\_FOUND",

 "message" : "Request not found"

}

### Подтверждение о получении ответа отправителем

POST Запрос с подтверждением о получении ответа отправителем

Пример запроса:

###### **HTTP request**

POST /responses/ack HTTP/1.1

Content-Type: application/json;charset=UTF-8

Content-Length: 22

Host: localhost:8080

{

 "requestId" : 15

}

###### **Request fields**

| **Path** | **Type** | **Description** |
| --- | --- | --- |
| requestId | Number | Идентификатор запроса в сервисе интеграций |

###### **HTTP response**

HTTP/1.1 200 OK

X-Content-Type-Options: nosniff

X-XSS-Protection: 0

Cache-Control: no-cache, no-store, max-age=0, must-revalidate

Pragma: no-cache

Expires: 0

Возможные ошибки:

Запрос не находится в состоянии ожидания подтверждения

###### **HTTP response**

HTTP/1.1 409 Conflict

Content-Type: application/json

X-Content-Type-Options: nosniff

X-XSS-Protection: 0

Cache-Control: no-cache, no-store, max-age=0, must-revalidate

Pragma: no-cache

Expires: 0

Content-Length: 144

{

 "errorCode" : "WRONG\_REQUEST\_STATE",

 "message" : "Request doesn't have an ack pending status, request status now is [RESPONSE\_RECEIVED]"

}

Запрос не найден

###### **HTTP response**

HTTP/1.1 404 Not Found

Content-Type: application/json

X-Content-Type-Options: nosniff

X-XSS-Protection: 0

Cache-Control: no-cache, no-store, max-age=0, must-revalidate

Pragma: no-cache

Expires: 0

Content-Length: 74

{

 "errorCode" : "REQUEST\_NOT\_FOUND",

 "message" : "Request not found"

}

### Отправка статусов и ответов на ранее полученный запрос

При получении запросов от сервиса интеграции, для идентификации для дальнейшего взаимодействия по теме данного запроса, отправитель должен в ответ на запрос отправить:

* свой идентификатор системы, помещенный в параметр systemId
* и json-объект, однозначно идентифицирующий данный запрос, помещенный в параметр techAttributes.

###### **Отправка статусов**

POST запрос для отправки статуса на ранее полученный запрос

Пример отправки статуса для систем, отправляющих systemId и tech\_attributes для сравнения с сохранёнными ранее

**HTTP request**

POST /statuses HTTP/1.1

Content-Type: application/json;charset=UTF-8

Application-Name: ngs-adapter-api

b3: 80f198ee56343ba864fe8b2a57d3eff7-e457b5a2e4d86bd1-1-05e3ac9a4f6e3b90

Content-Length: 188

Host: localhost:8080

{

 "statusContentFileId" : "external-integration-service-bucket/IDDQD",

 "businessAttributes" : null,

 "techAttributes" : {

 "asuloApplicationId" : "42"

 },

 "systemId" : "ASULO"

}

**Request fields**

| **Path** | **Type** | **Description** |
| --- | --- | --- |
| statusContentFileId | String | Контент статуса в файловом сервисе |
| businessAttributes | Object | Дополнительные параметры статуса |
| techAttributes | Object | Дополнительные технические параметры ответа, используются для идентификации запроса |
| systemId | String | Идентификатор системы, используется для идентификации запроса |

**HTTP response**

HTTP/1.1 200 OK

X-Content-Type-Options: nosniff

X-XSS-Protection: 0

Cache-Control: no-cache, no-store, max-age=0, must-revalidate

Pragma: no-cache

Expires: 0

Пример ошибки, возникающей при отправке статуса для систем, отправляющих systemId и tech\_attributes для сравнения с сохранёнными ранее

**HTTP request**

POST /statuses HTTP/1.1

Content-Type: application/json;charset=UTF-8

Content-Length: 191

Host: localhost:8080

{

 "statusContentFileId" : "external-integration-service-bucket/IDDQD",

 "businessAttributes" : null,

 "techAttributes" : {

 "asuloApplicationId" : "42"

 },

 "systemId" : "systemId"

}

**HTTP response**

HTTP/1.1 404 Not Found

Content-Type: application/json

X-Content-Type-Options: nosniff

X-XSS-Protection: 0

Cache-Control: no-cache, no-store, max-age=0, must-revalidate

Pragma: no-cache

Expires: 0

Content-Length: 101

{

 "errorCode" : "REQUEST\_NOT\_FOUND",

 "message" : "Recipient with systemId [systemId] not found"

}

###### **Отправка ответов**

POST запрос для отправки ответа на ранее полученный запрос

Пример отправки ответа для систем, отправляющих systemId и tech\_attributes для сравнения с сохранёнными ранее

**HTTP request**

POST /responses HTTP/1.1

Content-Type: application/json;charset=UTF-8

Application-Name: ngs-adapter-api

b3: 80f198ee56343ba864fe8b2a57d3eff7-e457b5a2e4d86bd1-1-05e3ac9a4f6e3b90

Content-Length: 253

Host: localhost:8080

{

 "bodyFileId" : "external-integration-service-bucket/IDDQD",

 "attachmentsFileIds" : [ "external-integration-service-bucket/IDDQD" ],

 "businessAttributes" : null,

 "techAttributes" : {

 "asuloApplicationId" : "42"

 },

 "systemId" : "ASULO"

}

**Request fields**

| **Path** | **Type** | **Description** |
| --- | --- | --- |
| bodyFileId | String | Контент тела ответа в файловом сервисе |
| attachmentsFileIds | Array | Список вложений ответа |
| businessAttributes | Object | Дополнительные параметры ответа |
| techAttributes | Object | Дополнительные технические параметры ответа, используются для идентификации запроса |
| systemId | String | Идентификатор системы, используется для идентификации запроса |

**HTTP response**

HTTP/1.1 200 OK

X-Content-Type-Options: nosniff

X-XSS-Protection: 0

Cache-Control: no-cache, no-store, max-age=0, must-revalidate

Pragma: no-cache

Expires: 0

Пример ошибки, возникающей при отправке ответа для систем, отправляющих systemId и tech\_attributes для сравнения с сохранёнными ранее

**HTTP request**

POST /responses HTTP/1.1

Content-Type: application/json;charset=UTF-8

Content-Length: 256

Host: localhost:8080

{

 "bodyFileId" : "external-integration-service-bucket/IDDQD",

 "attachmentsFileIds" : [ "external-integration-service-bucket/IDDQD" ],

 "businessAttributes" : null,

 "techAttributes" : {

 "asuloApplicationId" : "42"

 },

 "systemId" : "systemId"

}

**HTTP response**

HTTP/1.1 404 Not Found

Content-Type: application/json

X-Content-Type-Options: nosniff

X-XSS-Protection: 0

Cache-Control: no-cache, no-store, max-age=0, must-revalidate

Pragma: no-cache

Expires: 0

Content-Length: 101

{

 "errorCode" : "REQUEST\_NOT\_FOUND",

 "message" : "Recipient with systemId [systemId] not found"

}

## Передача исходящего запроса адаптеру

POST запрос для передачи исходящего запроса в адаптер

Пример запроса:

#### HTTP request

POST /requests HTTP/1.1

Content-Type: application/json;charset=UTF-8

b3: 80f198ee56343ba864fe8b2a57d3eff7-e457b5a2e4d86bd1-1-05e3ac9a4f6e3b90

Content-Length: 99

Host: localhost:8080

{

 "body" : {

 "bucket" : "poltava-bucket",

 "objectId" : "some/strange/object\_id.xml"

 }

}

#### Request fields

| **Path** | **Type** | **Description** |
| --- | --- | --- |
| id | Number | Идентификатор запроса в сервисе интеграции. Не используется |
| body | Object | Контент запроса в файловом сервисе |
| body.bucket | String | Корзина в файловом сервисе |
| body.objectId | String | Идентификатор контента в файловом сервисе |

#### HTTP response

HTTP/1.1 200 OK

Content-Type: application/json

Content-Length: 171

{

 "techResponseFileId" : "external-integration/out\_tech\_responses/07ecd2fa-4a06-11ec-81d3-0242ac130003.xml",

 "techAttributes" : {

 "asuloApplicationId" : "42"

 }

}

#### Response fields

| **Path** | **Type** | **Description** |
| --- | --- | --- |
| techAttributes | Object | Аттрибуты технического ответа на запрос |