



RuBackup

**Система резервного копирования
и восстановления данных**

REST API

ВЕРСИЯ 2.5.7.0.0, 02.07.2025

Содержание

1. Системные требования	29
1.1. Зависимости <code>rubackup-rest-api</code>	30
1.2. Зависимости <code>rubackup-server</code>	31
1.3. Зависимости <code>rubackup-common-gui</code>	38
1.4. Зависимости <code>rubackup-client</code>	39
1.5. Зависимости <code>rubackup-common</code>	41
2. Установка	43
2.1. Установка на выделенный хост	44
3. Настройка переменных окружения	46
3.1. Настройка на выделенном хосте	47
4. Настройка журналирования	49
4.1. Логгеры	49
4.2. Хэндлеры	49
4.3. Форматеры	49
5. Запуск и остановка сервиса	51
6. Токены и аутентификация	53
6.1. Выпуск токенов	53
6.1.1. Выпуск токенов в браузере	53
6.1.2. Выпуск токенов в командной строке	55
6.2. Обновление токенов	57
6.2.1. Обновление токенов в браузере	57
6.2.2. Обновление токенов в терминале	58
6.3. Отзыв токенов	60
6.4. Аутентификация пользователя	62
6.5. Аутентификация с Microsoft Active Directory	64
6.5.1. Выпуск <code>access_token</code> , <code>refresh_token</code> и <code>csrf_token</code> через браузер	64
6.5.2. Выпуск <code>access_token</code> , <code>refresh_token</code> и <code>csrf_token</code> в командной строке	66
7. Отправка запросов	69
7.1. Отправка запросов в браузере	69
7.2. Отправка запросов в терминале	71
8. Описание методов	72
8.1. Версия 2 (v2)	72
8.1.1. Авторизация и разрешения	72
Получение токенов и прохождение авторизации (POST <code>/auth/login</code>)	72

Обновление токена (POST /auth/refresh)	73
Деактивация токенов (DELETE /auth/logout)	73
Получение списка прав пользователя (GET /auth/permissions)	73
8.1.2. Пользователи	73
Получение списка пользователей (GET /users)	73
Получение информации о пользователе (GET /users/{user_id})	73
Получение списка прав пользователя (GET /users/permissions)	74
Добавление пользователя (POST /users)	74
Изменение пароля пользователя (PATCH /users/change_password)	74
Редактирование пользователей (PATCH /users/{user_id})	75
Удаление списка пользователей (DELETE /users)	75
Администраторы	76
Получение списка администраторов (GET /users/administrators)	76
Получение информации об одном администраторе (GET /users/administrators/{administrator_id})	76
Назначение пользователю роли администратора (POST /users/administrators/{administrator_id})	76
Удаление роли администратора у списка пользователей (DELETE /users/administrators)	76
Аудиторы	76
Получение списка аудиторов (GET /users/auditors)	77
Получение информации об аудиторе (GET /users/auditors/{auditor_id})	77
Назначение пользователю роли аудитора (POST /users/auditors)	77
Удаление роли аудитора у списка пользователей (DELETE /users/auditors)	77
Сопровождающие	77
Получение списка сопровождающих (GET /users/maintainers)	77
Получение информации о сопровождающем (GET /users/maintainers/{maintainer_id})	78
Назначение пользователю роли сопровождающего (POST /users/maintainers)	78
Удаление роли сопровождающего у списка пользователей (DELETE /users/maintainers)	78
Супервайзеры	78
Получение списка супервайзеров (GET /users/supervisors)	78
Получение информации о супервайзере (GET	

/users/supervisors/{supervisor_id})	78
Назначение пользователю роли супервайзера (POST	
/users/supervisors)	79
Удаление списка пользователей роли супервайзера (DELETE	
/users/supervisors)	79
Группы для уведомлений	79
Получение списка групп для уведомлений (GET /users/groups/)	79
Получение информации о группе уведомлений (GET	
/users/groups/{group_id})	79
Добавление группы уведомлений (POST /users/groups/)	80
Изменение группы для уведомлений (PATCH	
/users/groups/{group_id})	80
Удаление списка групп для уведомлений (DELETE /users/groups/)	80
Получение списка уведомлений (GET /users/groups/notifications)	81
Получение информации об уведомлении (GET	
/users/groups/notifications/{notification_id})	81
8.1.3. Сбор сведений	81
Сведения о сервере	81
Получение информации о сервере (GET /servers/hello)	81
Получение информации о состоянии приложения (GET	
/servers/states)	81
Получение информации о серверах и лицензии (GET	
/servers/licenses)	81
Получение информации о сервере и лицензии (GET	
/servers/licenses/{license_id})	81
Типы сущностей	82
Получение списка доступных типов резервного копирования (GET	
/types/backups)	82
Получение списка доступных типов сжатия (GET /types/compressions)	82
Получение списка доступных криптоалгоритмов (GET	
/types/cryptoalgorithms)	82
Получение списка доступных алгоритмов хеш-функций (GET	
/types/deduplication_hash_algorithms)	82
Получение списка доступных типов пулов (GET	
/types/destination_storages)	82
Получение списка типов лицензий (GET /types/licenses)	82
Получение информации о типе лицензии (GET /types/licenses/{id}) ..	82

Получение списка доступных уровней логирования (GET /types/log_levels)	83
Получение списка доступных операционных систем (GET /types/os) ..	83
Получение списка доступных типов ресурсов (GET /types/resources) ..	83
Получение информации о типе ресурса (GET /types/resources/{id}) ..	83
Получение списка доступных типов картриджей (GET /types/tapes) ..	83
Статусы	83
Получение списка статусов уведомлений (GET /statuses/notifications)	83
Получение списка статусов правил (GET /statuses/rules)	83
Получение списка статусов задач (GET /statuses/tasks)	83
Получение списка статусов задач на облачные хранилища (GET /statuses/s3_clouds)	84
Получение списка статусов верификации (GET /statuses/verifications)	84
Метрики (статистика)	84
Статистика уведомлений (GET /metrics/notifications)	84
Статистика объектов репозитория (GET /metrics/repositories)	84
Статистика очереди задач облаков (GET /metrics/tasks/clouds/s3) ..	84
Статистика очереди задач (GET /metrics/tasks)	84
Статистика очереди задач ленточных библиотек (GET /metrics/tasks/tape_libraries)	84
8.1.4. Глобальная конфигурация и сервисный режим	84
Получение параметров глобальной конфигурации (GET /global_configuration)	84
Редактирование параметров глобальной конфигурации (PATCH /global_configuration)	84
Включение и выключение сервисного режима (PATCH /global_configuration/service_mode)	86
8.1.5. Клиенты	86
Получение списка клиентов (GET /clients)	86
Получение клиента (GET /clients/{client_id})	86
Получение дерева клиентов (POST /clients/tree)	87
Редактирование существующего клиента (PATCH /clients/{client_id}) ..	87
Удаление нескольких клиентов (DELETE /clients)	88
Группы клиентов	88
Получение списка групп клиентов (GET /clients/groups)	88

Получение информации о группе клиентов (GET /clients/groups/{group_id})	88
Добавление групп клиентов (POST /clients/groups)	88
Изменение группы клиентов (PATCH /clients/groups/{group_id})	89
Удаление списка групп клиентов (DELETE /clients/groups)	89
Клиентские ресурсы хранения	90
Получение списка доступных директорий клиента (POST /clients/filesystem)	90
Получение списка доступных блочных устройств клиента (POST /clients/filesystem/block_devices)	90
Доступные типы ресурсов	90
Получение списка доступных ресурсов модуля (POST /clients/resources)	90
Получение доступных типов ресурсов для создания копии (GET /clients/{client_id}/resource_types)	91
Ограничения пропускной способности клиента	91
Получение списка ограничений пропускной способности клиента (GET /clients/{client_id}/bandwidth)	91
Получение информации об ограничении пропускной способности клиента (GET /clients/bandwidth/{bandwidth_id})	91
Создание ограничения пропускной способности клиента (POST /clients/{client_id}/bandwidth)	91
Изменение ограничения пропускной способности клиента (PATCH /clients/bandwidth/{bandwidth_id})	91
Удаление списка ограничений пропускной способности клиента (DELETE /clients/bandwidth)	92
Неавторизованные клиенты	92
Получение списка неавторизованных клиентов (GET /clients/unauthorised)	92
Получение информации о неавторизованном клиенте (GET /clients/unauthorised/{client_id})	92
Авторизация клиентов (POST /clients/unauthorised/authorise)	93
Удаление списка неавторизованных клиентов (DELETE /clients/unauthorised)	93
8.1.6. Хранилища	93
Локальные файловые хранилища	93
Получение списка локальных файловых хранилищ (GET	

/storages/local_catalogs)	93
Добавление локального файлового хранилища (POST	
/storages/local_catalogs)	93
Редактирование существующего локального файлового хранилища	
(PATCH /storages/local_catalogs/{catalog_id})	94
Удаление нескольких локальных файловых хранилищ (DELETE	
/storages/local_catalogs)	94
Получение информации о локальном файловом хранилище (GET	
/storages/local_catalogs/{catalog_id})	94
Блочные устройства	95
Получение списка блочных устройств (GET /storages/block_devices) ..	95
Получение информации о блочном устройстве (GET	
/storages/block_devices/{block_device_id})	95
Добавление блочного устройства (POST /storages/block_devices)	95
Редактирование блочного устройства (PATCH	
/storages/block_devices/{block_device_id})	95
Удаление списка блочных устройств (DELETE	
/storages/block_devices)	96
Клиентские хранилища	96
Получение списка клиентских хранилищ (GET	
/storages/client_defined)	96
Получение информации о клиентском хранилище (GET	
/storages/client_defined/{client_defined_id})	96
Добавление клиентского хранилища (POST /storages/client_defined) .	96
Изменение клиентского хранилища (PATCH	
/storages/client_defined/{client_defined_id})	97
Удаление списка клиентских хранилищ (DELETE	
/storages/client_defined)	97
Облачные хранилища	98
Получение списка доступных облаков (GET /storages/clouds/s3)	98
Получение информации о существующем облаке (GET	
/storages/clouds/s3/{s3_id})	98
Получение списка доступных бакетов существующего облака (GET	
/storages/clouds/s3/{s3_id}/buckets)	98
Получение списка доступных бакетов не добавленного в СРК облака	
(POST /storages/clouds/s3/buckets)	98
Добавление нового облака (POST /storages/clouds/s3)	99

Редактирование параметров облака (PATCH	
/storages/clouds/s3/{s3_id})	99
Удаление списка облаков (DELETE /storages/clouds/s3)	99
Картриджи	100
Получение списка доступных картриджей (GET	
/storages/tape_cartridges)	100
Получение информации о ленточном картридже (GET	
/storages/tape_cartridges/{cartridge_id})	100
Добавление ленточного картриджа (POST /storages/tape_cartridges)	100
Редактирование картриджа (PATCH	
/storages/tape_cartridges/{cartridge_id})	100
Удаление списка картриджей (DELETE /storages/tape_cartridges)	101
Заморозка картриджа (PATCH /storages/tape_cartridges/freeze)	101
Разморозка картриджа (PATCH /storages/tape_cartridges/unfreeze)	101
Приостановка картриджа (PATCH /storages/tape_cartridges/suspend)	102
Возобновление картриджа (PATCH	
/storages/tape_cartridges/unsuspend)	102
Ленточные библиотеки	103
Получение списка ленточных библиотек (GET /tape_libraries)	103
Получение дерева ленточных библиотек (POST /tape_libraries/tree)	103
Получение информации о ленточной библиотеке (GET	
/tape_libraries/{library_id})	103
Добавление ленточной библиотеки (POST /tape_libraries)	104
Удаление списка ленточных библиотек (DELETE /tape_libraries)	104
Получение списка доступных для добавления роботов и приводов (GET	
/tape_libraries/candidates)	104
Получение информации о роботе и слотах (POST	
/tape_libraries/candidates/info)	104
Синхронизация данных ленточной библиотеки (POST	
/tape_libraries/{library_id}/sync)	105
Роботы	105
Слоты	106
Ленточные приводы	108
8.1.7. Стратегии	110
Получение списка стратегий (GET /backup_strategies)	110
Получение информации о стратегии (GET	
/backup_strategies/{strategy_id})	111

Создание стратегии (POST /backup_strategies)	111
Изменение параметров стратегии (PATCH /backup_strategies/{strategy_id})	112
Изменение статуса стратегии (PATCH /backup_strategies)	114
Удаление списка стратегий (DELETE /backup_strategies)	114
Получение списка правил стратегий (GET /backup_strategies/rules)	115
Получение информации о правиле стратегии (GET /backup_strategies/rules/{rule_id})	115
Добавление правила в стратегию (POST /backup_strategies/rules)	115
Изменение правила стратегии (PATCH /backup_strategies/rules/{rule_id})	116
Удаление списка правил стратегии (DELETE /backup_strategies/rules)	117
Получение дополнительных параметров правила стратегии (GET /backup_strategies/extension)	118
Получение списка администраторов стратегии (GET /backup_strategies/{strategy_id}/administrators)	118
Назначение администраторов стратегии (POST /backup_strategies/{strategy_id}/administrators)	118
Удаление списка администраторов стратегии (DELETE /backup_strategies/administrators)	118
Правила копирования в пул	119
Получение списка настроенного копирования резервных копий по стратегии в пул (GET /backup_strategies/{strategy_id}/replications)	119
Получение информации о правиле копирования резервных копий по стратегии в пул (GET /backup_strategies/replications/{replication_id})	119
Создание правила копирования резервных копий по стратегии в пул (POST /backup_strategies/{strategy_id}/replications)	119
Редактирование правила копирования резервных копий по стратегии в пул (PATCH /backup_strategies/replications/{replication_id})	119
Удаление списка настроенного копирования резервных копий по стратегиям (DELETE /backup_strategies/replications)	120
8.1.8. Правила глобального расписания и резервное копирование	120
Получение списка правил глобального расписания (GET /global_schedule)	120
Получение дополнительных параметров глобального расписания (GET /global_schedule/extensions)	121

Получение информации о правиле глобального расписания (GET /global_schedule/{rule_id})	121
Добавление правила глобального расписания (POST /global_schedule) ..	121
Изменение правила глобального расписания (PATCH /global_schedule/{rule_id})	123
Изменение статуса правила глобального расписания (PATCH /global_schedule)	125
Удаление списка правил глобального расписания (DELETE /global_schedule)	125
Создание полной резервной копии по правилу глобального расписания (POST /global_schedule/{rule_id}/execution)	125
Репликации в глобальном расписании	126
Получение списка настроенных репликаций правил глобального расписания (GET /global_schedule/{rule_id}/replication)	126
Получение информации о настроенной репликации правила глобального расписания (GET /global_schedule/replication/{replication_id})	126
Создание правила репликации в глобальном расписании (POST /global_schedule/{rule_id}/replication)	126
Изменение параметров репликации правила глобального расписания (PATCH /global_schedule/replication/{replication_id})	126
Удаление списка настроенных репликаций правил глобального расписания (DELETE /global_schedule/replication)	127
Ограничения пропускной способности в глобальном расписании	127
Получение списка ограничений пропускной способности правил глобального расписания (GET /global_schedule/bandwidth)	127
Получение информации об ограничении пропускной способности правила глобального расписания (GET /global_schedule/bandwidth/{bandwidth_id})	128
Ограничение пропускной способности правила глобального расписания (POST /global_schedule/{rule_id}/bandwidth)	128
Изменение ограничения пропускной способности правила глобального расписания (PATCH /global_schedule/bandwidth/{bandwidth_id})	128
Удаление списка ограничений пропускной способности правил глобального расписания (DELETE /global_schedule/bandwidth)	129
Запросы клиентов на добавление и удаление правил	129
Создание правил	129

Удаление правил	130
8.1.9. Срочное резервное копирование	132
Создание срочной полной резервной копии (POST <code>/urgent_backup</code>)	132
8.1.10. Резервные копии	133
Получение списка резервных копий (GET <code>/repositories</code>)	133
Получение информации о резервных копиях (GET <code>/repositories/{repository_id}</code>)	133
Изменение периода хранения резервной копии (PATCH <code>/repositories/{repository_id}</code>)	133
Удаление списка резервных копий из репозитория (DELETE <code>/repositories</code>)	133
Получение списка зависимостей для нескольких цепочек (POST <code>/repositories/deletion_reference_chain</code>)	134
Получение дополнительных параметров восстановления (GET <code>/repositories/extensions</code>)	134
Копирование полной резервной копии (POST <code>/repositories/{repository_id}/copy</code>)	134
Перемещение полной резервной копии (POST <code>/repositories/{repository_id}/move</code>)	134
Восстановление резервной копии (POST <code>/repositories/{repository_id}/restore</code>)	135
Верификация резервной копии (POST <code>/repositories/{repository_id}/verification</code>)	136
Получение списка клиентов, доступных для восстановления резервной копии (GET <code>/repositories/{repository_id}/restore/available_clients</code>)	136
Получение списка ресурсов, включенных в копию для гранулярного восстановления (GET <code>/repositories/{repository_id}/granular_list</code>)	136
8.1.11. Задачи	136
Получение очереди задач (GET <code>/tasks</code>)	136
Получение дополнительных параметров созданной задачи (GET <code>/tasks/extensions</code>)	136
Получение информации о задаче из очереди (GET <code>/tasks/{task_id}</code>)	136
Удаление нескольких задач (DELETE <code>/tasks</code>)	137
(DELETE <code>/task_queue/{status}</code>) Удаление задачи с определенным статусом	137
Перезапуск задачи (POST <code>/tasks/restart</code>)	137
Приостановка выполнения задачи (POST <code>/tasks/{task_id}/pause</code>)	138

Возобновление приостановленной задачи (POST	
/tasks/{task_id}/continue)	138
Убийство задачи (POST /tasks/{task_id}/kill)	138
Получение журнала клиентских операций по задаче (GET	
/tasks/{task_id}/info/clients)	139
Получение журнала серверных операций по задаче (GET	
/tasks/{task_id}/info/servers)	139
Очередь задач облака	139
Получение задач очереди облаков (GET /tasks/clouds/s3)	139
Получение информации о задаче из очереди облаков (GET	
/tasks/clouds/s3/{task_id})	139
Удаление списка задач из очереди облаков (DELETE /tasks/clouds/s3)	139
Удаление ошибочных задач из очереди облаков (DELETE	
/tasks/clouds/s3/errors_deletion)	139
Очередь задач ленточных библиотек	140
Получение списка задач из очереди ленточных библиотек (GET	
/tasks/tape_libraries)	140
Получение информации о задаче из очереди ленточных библиотек	
(GET /tasks/tape_libraries/{task_id})	140
8.1.12. Медиасерверы	140
Получение списка медиасерверов (GET /media_servers)	140
Получение информации о медиасервере (GET	
/media_servers/{server_id})	140
Редактирование медиасервера (PATCH /media_servers/{server_id})	140
Удаление нескольких медиасерверов (DELETE /media_servers)	141
Неавторизованные медиасерверы	141
Получение списка неавторизованных медиасерверов (GET	
/media_servers/unauthorised)	141
Получение информации о неавторизованном медиасервере (GET	
/media_servers/unauthorised/{server_id})	141
Авторизация медиасервера (POST	
/media_servers/unauthorised/authorise)	141
Удаление списка неавторизованных медиасерверов (DELETE	
/media_servers/unauthorised)	142
8.1.13. Пулы	142
Получение списка пулов (GET /pools)	142
Получение информации о пуле (GET /pools/{pool_id})	142

Добавление пула (POST /pools)	142
Редактирование пула (PATCH /pools/{pool_id})	143
Удаление списка пулов (DELETE /pools)	144
Получение списка пулов, доступных для перемещения и копирования (GET /pools/destination_available)	144
Динамические группы пулов	144
Получение списка динамических групп пулов (GET /pools/groups)	144
Получение информации о динамической группе пулов (GET /pools/groups/{group_id})	145
Добавление динамической группы пулов (POST /pools/groups)	145
Изменение динамической группы пулов (PATCH /pools/groups/{group_id})	145
Удаление списка динамических групп пулов (DELETE /pools/groups)	146
Получение списка добавленных в динамическую группу пулов (GET /pools/groups/relationships)	146
Получение информации о добавленном в динамическую группу пуле (GET /pools/groups/relationships/{record_id})	146
Добавление пула в динамическую группу пулов (POST /pools/groups/relationships)	146
Удаление списка добавленных в динамическую группу пулов (DELETE /pools/groups/relationships)	147
Подмена пулов	147
Получение списка подмены пулов (GET /pools/substitutions)	147
Получение информации о подмене пулов (GET /pools/substitution/{substitution_id})	147
Добавление подмены пулов (POST /pools/substitutions)	147
Удаление списка подмены пулов (DELETE /pools/substitutions)	148
8.1.14. Журналы	148
Журнал глобального расписания	148
Получение списка записей в журнале глобального расписания (GET /global_schedule/logs)	148
Получение информации о записи в журнале глобального расписания (GET /global_schedule/logs/{record_id})	148
Журнал операций клиента	149
Получение журнала со списком операций клиентов (GET /clients/logs)	149
Получение одной записи из журнала с операций клиентов (GET	

<code>/clients/logs/{log_id}</code>)	149
Получение общих настроек журналов для всех клиентов (GET	
<code>/clients/logs/common_settings</code>)	149
Изменение общих настроек журналов для всех клиентов (PATCH	
<code>/clients/logs/common_settings</code>)	149
Получение списка всех персональных настроек журналов клиентов	
(GET <code>/clients/logs/personal_settings</code>)	149
Создание персональных настроек журналов клиента (POST	
<code>/clients/logs/personal_settings</code>)	149
Удаление персональных настроек журналов клиента (DELETE	
<code>/clients/logs/personal_settings</code>)	150
Изменение персональных настроек журналов клиентов (PATCH	
<code>/clients/{client_id}/logs/personal_settings</code>)	150
Получение значений по умолчанию настроек журналов клиентов (GET	
<code>/clients/logs/settings/defaults</code>)	151
Журналы медиасерверов	151
Получение журнала со списком операций медиасерверов (GET	
<code>/media_servers/logs</code>)	151
Получение журнала с операцией медиасервера (GET	
<code>/media_servers/logs/{log_id}</code>)	151
Получение значений по умолчанию настроек журналов для	
медиасерверов (GET <code>/media_servers/logs/settings/defaults</code>)	151
Получение списка всех персональных настроек журналов	
медиасерверов (GET <code>/media_servers/logs/personal_settings</code>)	151
Создание персональных настроек журналов медиасервера (POST	
<code>/media_servers/logs/personal_settings</code>)	151
Изменение персональных настроек журналов медиасервера (PATCH	
<code>/media_servers/{server_id}/logs/personal_settings</code>)	152
Удаление персональных настроек журналов медиасервера (DELETE	
<code>/media_servers/logs/personal_settings</code>)	152
Получение общих настроек журналов для всех медиасерверов (GET	
<code>/media_servers/logs/common_settings</code>)	152
Изменение общих настроек журналов для всех медиасерверов (PATCH	
<code>/media_servers/logs/common_settings</code>)	153
Журналы мониторинга сервера	153
Получение списка записей в журнале мониторинга сервера (GET	
<code>/media_servers/system_monitoring</code>)	153

Получение информации о записи в журнале мониторинга сервера (GET /media_servers/system_monitoring/{row_id})	153
Журнал репозитория	153
Получение списка записей в журнале репозитория (GET /repositories/logs)	153
Получение информации о записи в журнале репозитория (GET /repositories/logs/{record_id})	153
Журнал задач	154
Получение списка записей в журнале задач (GET /tasks/logs)	154
Получение информации о записи в журнале задач (GET /tasks/logs/{record_id})	154
Журнал событий информационной безопасности	154
Просмотр журналов событий информационной безопасности (GET /security/audit/logs)	154
Просмотр записи из журнала событий информационной безопасности (GET /security/audit/logs/{record_id})	154
Удаление списка записей из журналов событий информационной безопасности (DELETE /security/audit/logs)	154
8.1.15. Отчеты	155
Получение информации об отчете (GET /users/reports/{report_id})	155
Получение списка отчетов (GET /users/reports)	155
Создание отчета (POST /users/reports)	155
Изменение отчета (PATCH /users/reports/{report_id})	156
Изменение статуса отчета (PATCH /users/reports)	156
Удаление списка отчетов (DELETE /users/reports)	157
8.1.16. Microsoft Active Directory	157
Соединение с контроллером домена MS Active Directory	157
Получение параметров настроенного соединения с MS Active Directory (GET /auth/ldap/configuration)	157
Создание соединения с контроллером домена MS Active Directory (POST /auth/ldap/configuration)	157
Удаление соединения с контроллером домена (DELETE /auth/ldap/configuration)	158
Ассоциация групп MS Active Directory и ролей RuBackup	158
Получение списка ассоциаций групп MS Active Directory и ролей RuBackup (GET /auth/ldap/assosiations)	158
Ассоциация групп MS Active Directory и ролей RuBackup (POST	

/auth/ldap/assosiations)	158
Удаление ассоциации групп MS Active Directory и ролей RuBackup (DELETE /auth/ldap/assosiations)	158
8.2. Версия 1 (v1)	159
8.2.1. Авторизация и разрешения	159
Получение токенов и прохождение авторизации (POST /auth/login)	159
Деактивация имеющихся токенов (DELETE /auth/logout)	160
Обновление access токена после его истечения (POST /auth/refresh)	160
Получение списка прав пользователя (GET /auth/permissions)	160
8.2.2. Пользователи	160
Администраторы	160
Получение списка администраторов (GET /administrators)	160
Удаление списка ролей администратор (DELETE /administrators)	160
Назначение пользователю роли администратора (POST /administrators)	161
Получение информации об одном администраторе (GET /administrators/{id})	161
Удаление роли администратора. (DELETE /administrators/{id})	161
Супервайзеры	161
Удаление списка пользователей роли супервайзера (DELETE /supervisors)	161
Получение списка супервайзеров (GET /supervisors)	162
Назначение пользователю роли супервайзера (POST /supervisors)	162
Удаление пользователю роли супервайзера (DELETE /supervisors/{id})	162
Получение информации о супервайзере (GET /supervisors/{id})	162
Сопровождающие	163
Удаление списка пользователей роли сопровождающего (DELETE /maintainers)	163
Получение списка сопровождающих (GET /maintainers)	163
Назначение пользователю роли сопровождающего (POST /maintainers)	163
Удаление пользователю роли сопровождающего (DELETE /maintainers/{id})	163
Получение информации о сопровождающем (GET /maintainers/{id})	164
Аудиторы	164
Назначение пользователю роли аудитора (POST /auditors)	164

Получение информации об аудиторе (GET /auditors/{id})	164
Удаление аудитора (DELETE /auditors/{id})	164
Получение списка аудиторов (GET /auditors)	164
Удаление списка аудиторов (DELETE /auditors)	165
Пользователи	165
Удаление списка пользователей (DELETE /users)	165
Получение списка пользователей (GET /users)	165
Добавление пользователя (POST /users)	165
Изменение пароля пользователя (PATCH /users/change_password)	166
Редактирование пользователей (PATCH /users/{id})	166
Удаление пользователя (DELETE /users/{id})	167
Получение информации о пользователе (GET /users/{id})	167
Получение списка прав пользователя (GET /users/permissions)	167
8.2.3. Сбор сведений	167
Лицензии	167
Получение списка типов лицензий (GET /license_types)	168
Получение информации о типе лицензии (GET /license_types/{id})	168
Информация о сервере	168
Получение информации о сервере (GET /server_hello)	168
Сведения о состоянии приложения	168
Получение информации о состоянии приложения (GET /server_state)	168
Типы резервного копирования	168
Получение списка доступных типов резервного копирования (GET /backup_types)	168
Типы сжатия	168
Получение списка доступных типов сжатия (GET /compression_type)	168
Алгоритмы защиты данных	169
Получение списка доступных криптоалгоритмов (GET /crypto)	169
Алгоритмы дедупликации	169
Получение списка доступных алгоритмов хеш-функций (GET /deduplication_hash_algorithm)	169
Поиск медиасерверов	169
Поиск клиента медиасервера по HWID сервера (GET /servers_hw_id_tmp)	169
Получение информации о сервере и лицензии (GET /servers_hw_id_tmp/{id})	169
Доступные типы ресурсов	169

Получение списка доступных типов ресурсов (GET <code>/resource_types</code>) ..	169
Получение информации о типе ресурса (GET <code>/resource_types/{id}</code>)...	169
Доступные операционные системы	170
Получение списка доступных операционных систем (GET <code>/os_types</code>) ..	170
Статусы верификации	170
Получение списка доступных статусов верификации (GET <code>/verification_status</code>)	170
Статусы правил	170
Получение списка доступных статусов правил (GET <code>/status_of_rule</code>) ..	170
Ресурсы модулей	170
Получение списка доступных ресурсов модуля (POST <code>/module_resource_list</code>)	170
Метрики (статистика)	170
Получение статистики по уведомлениям (GET <code>/metrics/notifications</code>)	170
Получение статистики по объектам репозитория (GET <code>/metrics/repository</code>)	171
Получение статистики по очереди задач облаков (GET <code>/metrics/s3_cloud_task_queue</code>)	171
Получение статистики по очереди задач (GET <code>/metrics/task_queue</code>) ...	171
Получение статистики по очереди задач ленточных библиотек (GET <code>/metrics/tl_task_queue</code>)	171
8.2.4. Глобальная конфигурация	171
Редактирование параметров глобальной конфигурации (PATCH <code>/global_configuration</code>)	171
Получение параметров глобальной конфигурации (GET <code>/global_configuration</code>)	173
Включение и выключение сервисного режима (PATCH <code>/global_configuration/service_mode</code>)	173
8.2.5. Клиенты	173
Клиенты	173
Получение списка клиентов (GET <code>/clients</code>)	173
Удаление нескольких клиентов (DELETE <code>/clients</code>)	173
Авторизация клиентов (POST <code>/clients/authorize</code>)	174
Получение дерева клиентов (POST <code>/clients/tree</code>)	174
Редактирование существующего клиента (PATCH <code>/clients/{id}</code>)	174
Получение клиента (GET <code>/clients/{id}</code>)	175
Удаление клиента (DELETE <code>/clients/{id}</code>)	175

Получение доступных типов ресурсов для создания копии (GET /clients/{id}/resource_type)	175
Группы клиентов	176
Получение списка групп клиентов (GET /client_groups)	176
Удаление списка групп клиентов (DELETE /client_groups)	176
Добавление групп клиентов (POST /client_groups)	176
Изменение группы клиентов (PATCH /client_groups/{id})	176
Получение информации о группе клиентов (GET /client_groups/{id})	177
Удаление группы клиентов (DELETE /client_groups/{id})	177
Ограничения пропускной способности клиента	177
Получение списка ограничений пропускной способности клиента (GET /client_bandwidth)	177
Удаление списка ограничений пропускной способности клиента (DELETE /client_bandwidth)	178
Создание ограничения пропускной способности клиента (POST /client_bandwidth)	178
Изменение ограничения пропускной способности клиента (PATCH /client_bandwidth/{id})	178
Получение информации об ограничении пропускной способности клиента (GET /client_bandwidth/{id})	179
Удаление ограничения пропускной способности клиента (DELETE /client_bandwidth/{id})	179
Неавторизованные клиенты	179
Удаление списка неавторизованных клиентов (DELETE /unauthorised_clients)	179
Получение списка неавторизованных клиентов (GET /unauthorised_clients)	180
Удаление неавторизованного клиента (DELETE /unauthorised_clients/{id})	180
Получение информации о неавторизованном клиенте (GET /unauthorised_clients/{id})	180
8.2.6. Хранилища	180
Пулы и их группы	180
Доступные типы пулов	180
Пулы	180
Подмена пулов	183
Динамические группы пулов	184

Пулы в динамических группах	186
Локальные файловые хранилища	187
Удаление нескольких локальных файловых хранилищ (DELETE /storage_local_catalogs)	187
Получение списка локальных файловых хранилищ (GET /storage_local_catalogs)	188
Добавление локального файлового хранилища (POST /storage_local_catalogs)	188
Редактирование существующего локального файлового хранилища (PATCH /storage_local_catalogs/{id})	188
Удаление локального файлового хранилища (DELETE /storage_local_catalogs/{id})	188
Получение информации о локальном файловом хранилище (GET /storage_local_catalogs/{id})	189
Блочные устройства	189
Удаление списка блочных устройств (DELETE /storage_block_devices)	189
Получение списка блочных устройств (GET /storage_block_devices) ..	189
Добавление блочного устройства (POST /storage_block_devices)	189
Редактирование блочного устройства (PATCH /storage_block_devices/{id})	190
Удаление блочного устройства (DELETE /storage_block_devices/{id}) ..	190
Получение информации о блочном устройстве (GET /storage_block_devices/{id})	190
Ленточные хранилища	191
Ленточные библиотеки	191
Типы картриджей	193
Картриджи	193
Операции с ленточными приводами	195
Картриджи	197
Роботы	199
Облачные хранилища	200
Облачное хранилище	200
Статус задач облачного хранилища	203
Задачи облачного хранилища	203
Клиентские хранилища	204
Получение списка клиентских хранилищ (GET /client_defined_storages)	204

Добавление клиентского хранилища (POST /client_defined_storages)	204
Удаление списка клиентских хранилищ (DELETE /client_defined_storages)	204
Изменение клиентского хранилища (PATCH /client_defined_storages/{id})	204
Получение информации о клиентском хранилище (GET /client_defined_storages/{id})	205
Удаление клиентского хранилища (DELETE /client_defined_storages/{id})	205
Клиентские блочные устройства	205
Получение списка доступных блочных устройств указанного клиента (POST /list_client_block_devices)	205
Клиентские файловые системы	206
Получение списка доступных директорий указанного клиента (POST /list_client_filesystem)	206
Медиасерверы	206
Удаление нескольких медиасерверов (DELETE /media_servers)	206
Получение списка медиасерверов (GET /media_servers)	206
Авторизация медиасервера (POST /media_servers/authorize)	207
Получение дерева медиасерверов (GET /media_servers/tree)	207
Получение списка медиасерверов, имеющих пул типа ленточная библиотека (GET /media_servers/filtered_by_tape_library_pool)	207
Редактирование медиасервера (PATCH /media_servers/{id})	207
Удаление медиасервера (DELETE /media_servers/{id})	208
Получение информации о медиасервере (GET /media_servers/{id})	208
Неавторизованные медиасерверы	208
Удаление списка неавторизованных медиасерверов (DELETE /unauthorised_media_servers)	208
Получение списка неавторизованных медиасерверов (GET /unauthorised_media_servers)	208
Удаление неавторизованного медиасервера (DELETE /unauthorised_media_servers/{id})	209
Получение информации о неавторизованном медиасервере (GET /unauthorised_media_servers/{id})	209
Репозиторий	209
Получение списка объектов репозитория (GET /repository)	209
Удаление списка резервных копий из репозитория (DELETE	

/repository)	209
Получение списка зависимостей для нескольких цепочек (POST repository/deletion_reference_chain)	209
Получение дополнительных параметров восстановления (GET /repository/extension)	210
Получение информации об объекте репозитория (GET /repository/{id})	210
Удаление резервной копии из репозитория (DELETE /repository/{id})	210
Изменение периода хранения резервной копии (PATCH /repository/{id})	210
Копирование полной резервной копии (POST /repository/{id}/copying)	210
Перемещение полной резервной копии (POST /repository/{id}/moving)	211
Восстановление резервной копии (POST /repository/{id}/restoring)	211
Верификация резервной копии (POST /repository/{id}/verification)	212
Содержимое копии для гранулярного восстановления	212
Получение списка ресурсов, включенных в копию для гранулярного восстановления (GET /repository_record_file_list)	212
8.2.7. Стратегии	212
Стратегии резервного копирования	212
Изменение статуса стратегии (PATCH /backup_strategies)	212
Получение списка стратегий (GET /backup_strategies)	213
Удаление списка стратегий (DELETE /backup_strategies)	213
Создание стратегии (POST /backup_strategies)	213
Изменение параметров стратегии (PATCH /backup_strategies/{id}) ..	215
Получение информации о стратегии (GET /backup_strategies/{id}) ..	217
Удалении одной стратегии (DELETE /backup_strategies/{id})	217
Копирование в пул по стратегиям	217
Получение списка настроенного копирования резервных копий по стратегиям в пул (GET /backup_strategies_replication)	217
Удаление списка настроенного копирования резервных копий по стратегиям (DELETE /backup_strategies_replication)	217
Создание правила копирования резервных копий по стратегиям в пул (POST /backup_strategies_replication)	217
Получение информации о копировании резервных копий по стратегии в пул (GET /backup_strategies_replication/{id})	218

Удаление настроенного копирования резервных копий по стратегиям (DELETE backup_strategies_replication/{id})	218
Изменение параметров правила копирования резервных копий по стратегии в пул (PATCH backup_strategies_replication/{id})	218
Правила стратегий	219
Удаление списка правил стратегии (DELETE /strategy_rules)	219
Получение списка правил стратегий (GET /strategy_rules)	219
Добавление правила в стратегию (POST /strategy_rules)	219
Получение дополнительных параметров правила стратегии (GET /strategy_rules/extension)	220
Изменение правила стратегии (PATCH /strategy_rules/{id})	220
Удаление правила стратегии (DELETE /strategy_rules/{id})	222
Получение информации о правиле стратегии (GET /strategy_rules/{id})	222
Администраторы стратегий	222
Получение списка администраторов для стратегии (GET /strategy_administrators)	222
Получение информации об администраторе стратегии (GET /strategy_administrators/{id})	222
Назначение администраторов для стратегии (POST /strategy_administrators)	222
Удаление списка администраторов стратегии (DELETE /strategy_administrators)	223
Удаление администратора стратегии (DELETE /strategy_administrators/{id})	223
8.2.8. Глобальное расписание	223
Правила глобального расписания	223
Изменение статуса правила глобального расписания (PATCH /global_schedule)	223
Получение списка правил глобального расписания (GET /global_schedule)	224
Удаление списка правил глобального расписания (DELETE /global_schedule)	224
Добавление правила глобального расписания (POST /global_schedule)	224
Получение дополнительных параметров глобального расписания по id правила (GET /global_schedule/extension)	226
Изменение правила глобального расписания (PATCH	

<code>/global_schedule/{id}</code>)	226
Получение информации о правиле глобального расписания (GET <code>/global_schedule/{id}</code>)	228
Удаление правила глобального расписания (DELETE <code>/global_schedule/{id}</code>)	228
Создание полной резервной копии по правилу глобального расписания (POST <code>/global_schedule/{id}/execution</code>)	228
Репликация правил глобального расписания	229
Получение списка настроенных репликаций правил глобального расписания (GET <code>/global_schedule_replication</code>)	229
Настройка репликации правил глобального расписания (POST <code>/global_schedule_replication</code>)	229
Удаление списка настроенных репликаций правил глобального расписания (DELETE <code>/global_schedule_replication</code>)	229
Получение информации о настроенной репликации правила глобального расписания (GET <code>/global_schedule_replication/{id}</code>) ...	230
Удаление настроенной репликации правила глобального расписания (DELETE <code>/global_schedule_replication/{id}</code>)	230
Изменение параметров репликации правил глобального расписания (PATCH <code>/global_schedule_replication/{id}</code>)	230
Запросы на добавление правил глобального расписания	231
Получение списка запросов клиента на добавление правила глобального расписания (GET <code>/new_rule_requests</code>)	231
Одобрение запроса клиента на добавление правила глобального расписания (POST <code>/new_rule_requests/approve</code>)	231
Отклонение запроса клиента на добавление правила глобального расписания (DELETE <code>/new_rule_requests/decline</code>)	231
Получение информации о запросе клиента на добавление правила глобального расписания (GET <code>/new_rule_requests/{id}</code>)	232
Запросы на удаление правил глобального расписания	232
Получение списка запросов клиента на удаление правила глобального расписания (GET <code>/delete_rule_requests</code>)	232
Одобрение запроса клиента на удаление правила глобального расписания (POST <code>/delete_rule_requests/approve</code>)	232
Отклонение запроса клиента на удаление правила глобального расписания (DELETE <code>/delete_rule_requests/decline</code>)	233
Получение информации о запросе клиента на удаление правила	

глобального расписания (GET <code>/delete_rule_requests/{id}</code>)	233
Ограничения пропускной способности	233
Получение списка ограничений пропускной способности правила глобального расписания (GET <code>/rule_bandwidth</code>)	233
Удаление списка ограничений пропускной способности правила глобального расписания (DELETE <code>/rule_bandwidth</code>)	233
Ограничение пропускной способности правила глобального расписания (POST <code>/rule_bandwidth</code>)	234
Изменение ограничения пропускной способности правила глобального расписания (PATCH <code>/rule_bandwidth/{id}</code>)	234
Удаление ограничения пропускной способности правила глобального расписания (DELETE <code>/rule_bandwidth/{id}</code>)	235
Получение информации об ограничении пропускной способности правила глобального расписания (GET <code>/rule_bandwidth/{id}</code>)	235
8.2.9. Задачи	235
Задачи	235
Удаление нескольких задач (DELETE <code>/task_queue</code>)	235
Получение очереди задач (GET <code>/task_queue</code>)	236
Создание срочной полной резервной копии (POST <code>/task_queue</code>)	236
Получение дополнительных параметров созданной задачи (GET <code>/task_queue/extension</code>)	237
Удаление задачи (DELETE <code>/task_queue/{id}</code>)	237
Получение информации о задаче из очереди (GET <code>/task_queue/{id}</code>)	237
Удаление задачи с определенным статусом (DELETE <code>/task_queue/{status}</code>)	237
Статусы задач	238
Получение списка возможных статусов задач (GET <code>/task_status</code>)	238
Перезапуск задач	238
Перезапуск задачи (POST <code>/restart_task</code>)	238
Приостановка задач	238
Приостановка выполнения задачи (POST <code>/put_task_on_pause</code>)	238
Возобновление выполнения задач	239
Возобновление приостановленной задачи (POST <code>/continue_task_working</code>)	239
Принудительное завершение задач	239
Убийство задачи (POST <code>/kill_task</code>)	239
Очередь задач ленточных библиотек	239

Получение списка задач из очереди ленточных библиотек (GET /tl_task_queue)	240
Получение информации о задаче из очереди ленточных библиотек (GET /tl_task_queue/{id})	240
8.2.10. Уведомления	240
Статус уведомлений	240
Получение списка статусов уведомлений (GET /notifications_status)	240
Уведомления	240
Получение списка уведомлений (GET /notifications)	240
Получение информации об уведомлении (GET /notifications/{id})	240
Уведомления групп	240
Удаление списка групп для уведомлений (DELETE /user_groups)	240
Получение списка групп для уведомлений (GET /user_groups)	241
Добавление группы уведомлений (POST /user_groups)	241
Изменение группы для уведомлений (PATCH /user_groups/{id})	241
Удаление группы для уведомлений (DELETE /user_groups/{id})	242
Получение информации о группе уведомлений (GET /user_groups/{id})	242
8.2.11. Журналы	242
Журнал операций клиента	242
Получение журнала со списком операций клиентов (GET /clients_log)	242
Получение одной записи из журнала с операций клиентов (GET /clients_log/{id})	242
Журнал глобального расписания	243
Получение списка записей в журнале глобального расписания (GET /global_schedule_log)	243
Получение информации о записи в журнале глобального расписания (GET /global_schedule_log/{id})	243
Журнал операций медиасерверов	243
Получение журнала со списком операций медиасерверов (GET /media_servers_log)	243
Получение журнала с операцией медиасервера (GET /media_servers_log/{id})	243
Журнал репозитория	243
Получение списка записей в журнале репозитория (GET /repository_log)	243
Получение информации о записи в журнале репозитория (GET	

/repository_log/{id})	243
Журнал задач	244
Получение информации о записи в журнале задач (GET	
/task_log/{id})	244
Получение списка записей в журнале задач (GET /task_log)	244
Журнал операций сервера	244
Получение журнала серверных операций по задаче (GET	
/server_task_info)	244
Журнал клиентских операций	244
Получение журнала клиентских операций по задаче (GET	
/client_task_info)	244
Журнал мониторинга сервера	245
Получение списка записей в журнале мониторинга сервера (GET	
/rubackup_server_system_monitoring)	245
Получение информации о записи в журнале мониторинга сервера (GET	
/rubackup_server_system_monitoring/{id})	245
Журнал событий информационной безопасности	245
Удаление списка записей из журналов событий ИБ (DELETE	
/security_audit_log)	245
Просмотр журналов событий ИБ (GET /security_audit_log)	246
Удаление записи из журнала событий ИБ (DELETE	
/security_audit_log/{id})	246
Просмотр записи из журнала событий ИБ (GET	
/security_audit_log/{id})	246
8.2.12. Отчеты	246
Изменение статуса отчета (PATCH /reports)	246
Удаление списка отчетов (DELETE /reports)	247
Получение списка отчетов (GET /reports)	247
Создание отчета (POST /reports)	247
Изменение отчета (PATCH /reports/{id})	248
Получение информации об отчете (DELETE /reports/{id})	248
Удаление отчета (GET /reports/{id})	248
8.2.13. Microsoft Active Directory	249
Настройка LDAP	249
Удаление соединения (DELETE /ldap/configuration)	249
Настройка соединения с MS AD (POST /ldap/configuration)	249
Получение параметров настроенного соединения с MS AD (GET	

/ldap/configuration)	249
Ассоциация групп MS AD и ролей RuBackup	249
Удаление ассоциации групп MS AD и ролей RuBackup (DELETE	
/ldap/assosiations)	250
Получение списка ассоциаций групп MS AD и ролей RuBackup (POST	
/ldap/assosiations)	250
Ассоциация групп MS AD и ролей RuBackup (GET /ldap/assosiations) .	250

Система резервного копирования и восстановления данных RuBackup предоставляет пользователю возможность взаимодействия с активным сервером резервного копирования посредством HTTP-запросов к его ресурсам.

Программный интерфейс RuBackup реализован и документирован с использованием Swagger.

Настоящее руководство описывает базовые шаги установки, настройки и эксплуатации RuBackup REST API. Руководство предназначено для системных администраторов, отвечающих за сопровождение СРК.

Глава 1. Системные требования

Перед установкой RuBackup API убедитесь, что выполнены условия для установки ([Развёртывание](#)):

- скачаны и установлены все пакеты актуальной версии CPK, требуемые для `rubackup-rest-api` в соответствии с диаграммой зависимостей:

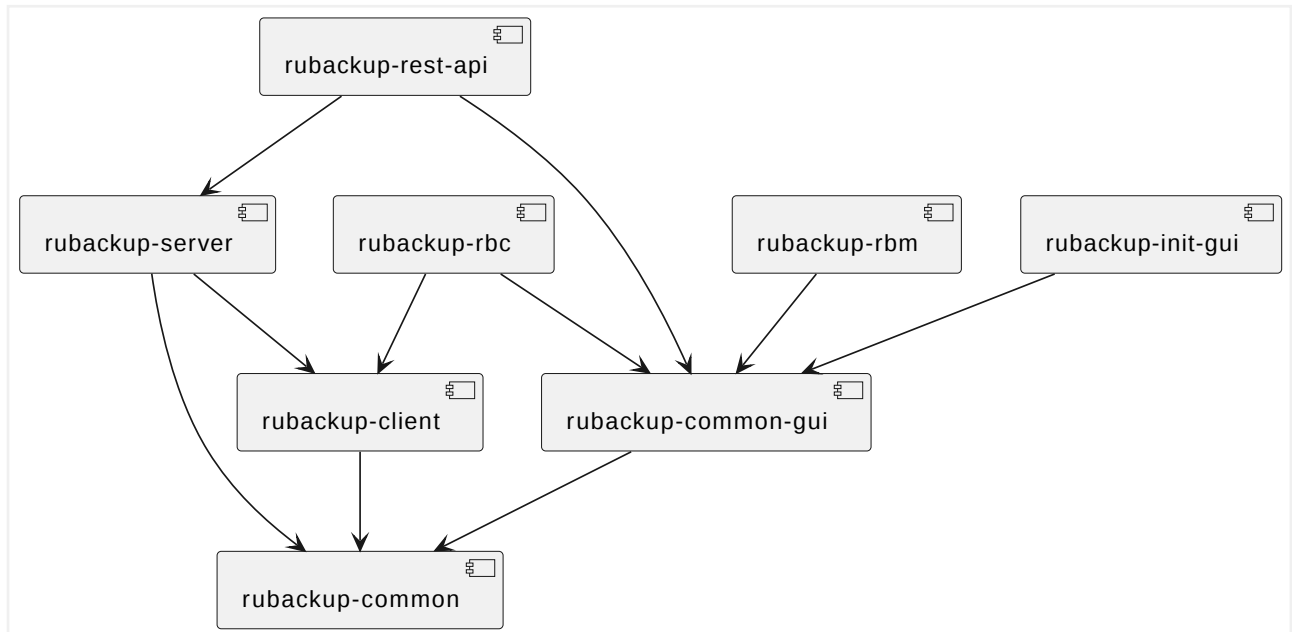


Рисунок 1. Зависимости пакетов RuBackup

- установлены сторонние зависимости всех требуемых пакетов CPK RuBackup для используемой операционной системы ([Раздел 1.1](#), [Раздел 1.2](#), [Раздел 1.3](#), [Раздел 1.4](#), [Раздел 1.5](#));
- проведена настройка основного сервера с помощью интерактивной утилиты `rb_init`.

Подробнее о вариантах установки REST API см. [Глава 2](#).

Проверьте наличие файлов-сертификатов RuBackup, которые используются для установления защищённого соединения.

Расположение сертификатов в файловой системе:

- `/opt/rubackup/keys/server/serverCert.crt`
- `/opt/rubackup/keys/server/serverKey.key`
- `/opt/rubackup/keys/client/clientCert.crt`
- `/opt/rubackup/keys/client/clientKey.key`
- `/opt/rubackup/keys/rootCA/serverRootCACert.crt`

1.1. Зависимости rubackup-rest-api

Таблица 1. Зависимости rubackup-rest-api

Операционная система	Пакеты
Astra 1.6	gnupg2 wget xauth
Astra 1.7	gnupg2 wget xauth
Astra 1.8	gnupg2 wget xauth
CentOS 7	postgresql-libs
Rosa Cobalt 7.3	bash gdbm glibc libffi libgcc libstdc++ libuuid ncurses-libs openssl-libs postgresql-libs readline sqlite zlib
Rosa Cobalt 7.9	bash gdbm glibc libffi libgcc libstdc++ libuuid ncurses-libs openssl-libs postgresql-libs readline sqlite zlib
Ubuntu 22.04	gnupg2 wget xauth
Альт 10	libpq5

1.2. Зависимости rubackup-server

Таблица 2. Зависимости rubackup-server

Операционная система	Пакеты
Astra 1.6	exim4-base exim4-config exim4-daemon-light gnupg2 libcurl3 или libcurl4 libldap-2.4-2 libldap-2.4-2 libldap-common liblockfile-bin liblockfile1 libnghttp2-14 librtmp1 libsasl2-2 libssh2-1 mailutils или bsd-mailx openssl psmisc wget xauth

Операционная система	Пакеты
Astra 1.7	exim4-base exim4-config exim4-daemon-light gnupg2 guile-2.2-libs libcurl3 или libcurl4 libevent-2.1-6 libfribidi0 libgc1c2 libgnutls-dane0 libgsasl7 libkyotocabinet16v5 libldap-2.4-2 libltdl7 liblzo2-2 libmailutils5 libmariadb3 libntlm0 libpugixml1v5 libsasl2-2 libunbound8 mailutils или bsd-mailx mailutils-common mariadb-common mysql-common openssl psmisc wget xauth

Операционная система	Пакеты
Astra 1.8	exim4-base exim4-config exim4-daemon-light gnupg2 gsasl-common guile-3.0-libs ibmariadb3 libcurl3 или libcurl4 libevent-2.1-7 libgc1 libgnutls-dane0 libgnutls30 libgsasl18 libgssglue1 libidn12 libldap-2.5-0 libltdl7 libmailutils9 libncurses6 libncursesw6 libntlm0 libpq5 libpugixml1v5 libsasl2-2 libtinfo6 libunbound8 mailutils или bsd-mailx mailutils-common mariadb-common mysql-common ncurses-base ncurses-bin ncurses-term openssl psmisc wget xauth
CentOS 7	cyrus-sasl mailx openldap pugixml
CentOS 8	cyrus-sasl mailx openldap pugixml

Операционная система	Пакеты
Debian 10	exim4-base exim4-config exim4-daemon-light gnupg2 guile-2.2-libs libcurl3 или libcurl4 libcurl4 libevent-2.1-6 libfribidi0 libgc1c2 libgnutls-dane0 libgsasl7 libkyotocabinet16v5 libldap-2.4-2 libltdl7 liblzo2-2 libmailutils5 libmariadb3 libntlm0 libpugixml1v5 libpython2.7 libsasl2-2 libunbound8 mailutils или bsd-mailx mailutils-common mariadb-common mysql-common openssl psmisc wget xauth

Операционная система	Пакеты
Debian 12	exim4-base exim4-config exim4-daemon-light gnupg2 gssasl-common guile-3.0-libs libcurl3 или libcurl4 libevent-2.1-7 libfribidi0 libgc1 libgnutls-dane0 libgnutls30 libgssasl18 libgssglue1 libidn12 libldap-2.5-0 libltdl7 libmailutils9 libmariadb3 libncurses6 libntlm0 libpq5 libpugixml1v5 libpython3.11 libpython3.11-minimal libpython3.11-stdlib libsasl2-2 libunbound8 mailutils или bsd-mailx mailutils-common mariadb-common mysql-common openssl psmisc python3.11 python3.11-minimal wget xauth
RHEL 9	cyrus-sasl mailx openldap pugixml s-nail
RedOS 7.3	cyrus-sasl mailx openldap pugixml

Операционная система	Пакеты
RedOS 8	cyrus-sasl mailx openldap pugixml
Rosa Chrome 12	cyrus-sasl lib64db5.2 lib64ldap2.4_2 lib64ltdl7 lib64mailutils9 lib64mu_auth9 lib64mu_dbm9 lib64mu_dotmail9 lib64mu_imap9 lib64mu_maildir9 lib64mu_mailer9 lib64mu_mbox9 lib64mu_pop9 lib64mu_sieve9 lib64muaux9 lib64pugixml1 lib64sasl2 mailutils mailutils-locales qt5-qtbase-gui
Rosa Cobalt 7.3	cyrus-sasl mailx openldap
Rosa Cobalt 7.9	cyrus-sasl libicu libxkbcommon-x11 mailx openldap qt5-qtbase-gui

Операционная система	Пакеты
Ubuntu 18.04	gnupg2 guile-2.0-libs libcurl3 или libcurl4 libgc1c2 libgsasl7 libkyotocabinet16v5 libldap-2.4-2 libltdl7 liblz0-2 libmailutils5 libmysqlclient20 libnghttp2-14 libntlm0 libpython2.7 libpython2.7-minimal libpython2.7-stdlib librtmp1 libsasl2-2 mailutils или bsd-mailx mailutils-common mysql-common openssl postfix ssl-cert wget xauth
Ubuntu 20.04	gnupg2 guile-2.2-libs libcurl3 или libcurl4 libgc1c2 libgsasl7 libidn11 libkyotocabinet16v5 libldap-2.4-2 libmailutils6 libmysqlclient21 libntlm0 libpugixml1v5 libsasl2-2 mailutils или bsd-mailx mailutils-common mysql-common openssl postfix ssl-cert wget xauth

Операционная система	Пакеты
Ubuntu 22.04	gnupg2 gssasl-common guile-3.0-libs libcurl3 или libcurl4 libfribidi0 libgc1 libgssasl7 libidn12 libldap-2.5-0 libltdl7 libmailutils8 libmysqlclient21 libntlm0 libpq5 libpugixml1v5 libsasl2-2 mailutils или bsd-mailx mailutils-common mysql-common openssl postfix ssl-cert wget xauth
Альт 10	libldap libsasl2-3 mailutils pugixml xauth

1.3. Зависимости **rubackup-common-gui**

Таблица 3. Зависимости **rubackup-common-gui**

Операционная система	Пакеты
Astra 1.6	gnupg2 wget xauth
Astra 1.7	gnupg2 wget xauth
Astra 1.8	gnupg2 wget xauth
Debian 10	gnupg2 wget xauth

Операционная система	Пакеты
Debian 12	gnupg2 wget xauth
Rosa Chrome 12	qt5-qtbase-gui
Rosa Cobalt 7.9	libicu libxkbcommon-x11 qt5-qtbase-gui
Ubuntu 18.04	gnupg2 wget xauth
Ubuntu 20.04	gnupg2 wget xauth
Ubuntu 22.04	gnupg2 wget xauth
Альт 10	xauth

1.4. Зависимости rubakup-client

Таблица 4. Зависимости rubakup-client

Операционная система	Пакеты
Astra 1.6	gnupg2 openssl parsec-base parsec-cap parsec-mac wget xauth
Astra 1.7	gnupg2 openssl parsec-base parsec-cap parsec-mac wget xauth
Astra 1.8	gnupg2 openssl parsec-base parsec-cap parsec-mac wget xauth
CentOS 7	qt5-qtbase-gui

Операционная система	Пакеты
CentOS 8	qt5-qtbase-gui
Debian 10	gnupg2 openssl wget xauth
Debian 12	gnupg2 openssl wget xauth
RHEL 9	qt5-qtbase-gui
RedOS 7.3	qt5-qtbase-gui
RedOS 8	qt5-qtbase-gui
Rosa Chrome 12	lib64qt5gui5 qt5-qtbase-gui
Rosa Cobalt 7.3	cups-libs fontconfig fontpackages-filesystem glx-utils libICE libSM libX11 libX11-common libXau libXdamage libXext libXfixes libXi libXrender libXxf86vm libicu libpng libxcb libxshmfence mesa-libEGL mesa-libGL mesa-libgbm mesa-libglapi qt5-qtbase qt5-qtbase-common qt5-qtbase-gui qt5-qtbase-gui xcb-util xcb-util-image xcb-util-keysyms xcb-util-renderutil xcb-util-wm

Операционная система	Пакеты
Rosa Cobalt 7.9	libicu libxkbcommon-x11 qt5-qtbase-gui
Ubuntu 18.04	gnupg2 openssl wget xauth
Ubuntu 20.04	gnupg2 openssl wget xauth
Ubuntu 22.04	gnupg2 openssl wget xauth
Альт 10	qt5-qtbase-gui xauth

1.5. Зависимости rubakup-common

Таблица 5. Зависимости rubakup-common

Операционная система	Пакеты
Astra 1.6	gnupg2 wget xauth
Astra 1.7	gnupg2 wget xauth
Astra 1.8	gnupg2 wget xauth
Debian 10	gnupg2 wget xauth
Debian 12	gnupg2 wget xauth
Rosa Chrome 12	qt5-qtbase-gui
Rosa Cobalt 7.9	libicu libxkbcommon-x11 qt5-qtbase-gui
Ubuntu 18.04	gnupg2 wget xauth

Операционная система	Пакеты
Ubuntu 20.04	gnupg2 wget xauth
Ubuntu 22.04	gnupg2 wget xauth
Альт 10	xauth

Глава 2. Установка



Перед использованием `rubackup-rest-api` рекомендуется внести изменения в конфигурационный файл `postgresql.conf` для увеличения количества зарезервированных подключений суперпользователя, например, командой:

```
nano /etc/postgresql/<номер версии>/main/postgresql.conf
```

где `<номер версии>` — это номер версии PostgreSQL.

Необходимо выставить значение:

```
superuser_reserved_connections = 50
```

Дистрибутив RuBackup REST API поставляется в виде deb и rpm-пакетов. Для разных дистрибутивов Linux, по причине их отличий друг от друга, предусмотрены специально подготовленные пакеты RuBackup.

В зависимости от типа используемого пакетного менеджера в вашем дистрибутиве Linux, процедура установки и удаления пакетов может использовать команды `dpkg`, `rpm`, `apt`, `yum` и пр. В настоящем руководстве процедуры установки описаны для пакетного менеджера, который оперирует пакетами deb. Например, команда установки пакета в операционной системе Ubuntu 20.04 выглядит следующим образом:

```
sudo dpkg -i rubackup-rest-api.deb
```

Для установки API в ОС с пакетным менеджером, который оперирует rpm-пакетами, вместо вышеуказанной команды следует выполнить команду:

```
sudo rpm -i rubackup-rest-api.rpm
```

Процедуры удаления пакетов в настоящем руководстве описаны для пакетного менеджера, который оперирует пакетами deb. Например, процедура удаления пакета RuBackup API выглядит следующим образом:

```
sudo apt remove rubackup-rest-api
```

Для удаления RuBackup API в операционной системе с пакетным менеджером,

который оперирует rpm пакетами, вместо вышеуказанной команды следует выполнить:

```
sudo yum remove rubackup-rest-api
```

либо:

```
sudo rpm -e rubackup-rest-api
```

Некоторые операционные системы, например Alt Linux, используют пакетную систему rpm, но вместо `yum` используют `apt`. Перед установкой или удалением пакета RuBackup REST API следует уточнить, какие команды необходимо использовать для вашего дистрибутива Linux.

Для инсталляции RuBackup API установите пакет `rubackup-rest-api` командой:

```
sudo dpkg -i rubackup-rest-api.deb
```

Имя файла пакета может отличаться в зависимости от сборки.

После установки пакета вы можете сразу запустить процесс RuBackup API, если у вас уже определён FQDN для хоста и он явно указан в файле `/etc/hosts`.

2.1. Установка на выделенный хост

Перед инсталляцией RuBackup API убедитесь, что выполнены все действия для установки СРК RuBackup согласно документу «Руководство по установке серверов резервного копирования и Linux-клиентов»:

- скачаны все необходимые пакеты актуальной версии `rubackup-common`, `rubackup-client`, `rubackup-server` и `rubackup-rest-api`;
- пакеты `rubackup-common`, `rubackup-client`, `rubackup-server` установлены. Проводить настройку с помощью утилиты `rb_init` не нужно;
- существует хост с установленным, настроенным и запущенным основным сервером `rubackup`;
- существует хост с базой данных `rubackup`.

Для инсталляции RuBackup API установите пакет `rubackup-rest-api`, например, командой:

```
sudo dpkg -i rubackup-rest-api.deb
```



Имя файла пакета может отличаться в зависимости от сборки.

Глава 3. Настройка переменных окружения

Настройка RuBackup API осуществляется пользователем при помощи изменения переменных окружения.



Для RuBackup API данные из конфигурационного файла RuBackup (/opt/rubackup/etc/config.file) не используются.

Ниже представлен перечень переменных окружения доступных для изменения из файла /opt/rubackup/etc/rubackup_api.env:

Таблица 6. Переменные окружения

Имя переменной	Описание	Возможные значения
APP_HOST	Желаемый IP-адрес или FQDN, который будет использоваться как часть адреса сервера API. Если IP-адрес или FQDN указан некорректно, то при запуске RuBackup API не будут записываться access_token и refresh_token в cookies.	IP/FQDN (localhost)
APP_PORT	Желаемый порт, который будет использоваться как часть адреса сервера API	Порт (5656)
DB_HOST	IP или FQDN сервера PostgreSQL с базой данных RuBackup	IP/FQDN (localhost)
DB_PORT	Порт сервера PostgreSQL с базой данных RuBackup	Порт (5432)
RB_SERVER_HOST	IP или FQDN основного сервера RuBackup	IP/FQDN (localhost)
DEBUG	Режим расширенного логирования	true, false

Указанные переменные могут быть применены локально с помощью export:

```
export APP_HOST=api.rubackup.local
```

Пользователь может зафиксировать значения переменных глобально, описав их в файле /opt/rubackup/etc/rubackup_api.env.

Пример 1. Задание переменных окружения в rubackup_api.env

```
APP_HOST=localhost
APP_PORT=5656
DB_HOST=localhost
DB_PORT=5432
RB_SERVER_HOST=localhost
```

```
DEBUG=False
```

Запуск Swagger и Tuscany будет произведён по адресу, указанному в параметре `APP_HOST`.

Чтобы запуск был произведен по доменному имени, достаточно указать его:

```
APP_HOST=api.rubakup.local
```

Тот же FQDN должен быть указан `/etc/hosts`.

Для запуска через `localhost` можно оставить параметры по умолчанию:

```
APP_HOST=localhost
```

Для запуска через IP-адрес нужно указать IP-адрес хоста:

```
APP_HOST=10.177.33.100
```

3.1. Настройка на выделенном хосте



Для RuBackup API из конфигурационного файла RuBackup (`/opt/rubakup/etc/config.file`) данные не используются.

Ниже представлен перечень переменных окружения доступных для изменения из файла `/opt/rubakup/etc/rubakup_api.env`:

Таблица 7. Переменные окружения

Имя переменной	Описание	Возможные значения
APP_HOST	Желаемый IP-адрес или FQDN, который будет использоваться как часть адреса сервера API. Если IP-адрес или FQDN указан некорректно, то при запуске RuBackup API не будут записываться <code>access_token</code> и <code>refresh_token</code> в cookies.	IP/FQDN (<code>localhost</code>)

Имя переменной	Описание	Возможные значения
APP_PORT	Желаемый порт, который будет использоваться как часть адреса сервера API	Порт (5656)
DB_HOST	IP или FQDN сервера PostgreSQL с базой данных RuBackup	IP/FQDN (localhost)
DB_PORT	Порт сервера PostgreSQL с базой данных RuBackup	Порт (5432)
RB_SERVER_HOST	IP или FQDN основного сервера RuBackup	IP/FQDN (localhost)
DEBUG	Режим расширенного логирования	true , false

Указанные переменные могут быть применены локально с помощью `export`:

```
export APP_HOST=api.rubakup.local
```

У пользователя есть возможность зафиксировать значения описанных переменных, описав их в файле `.bashrc`.

Пример 2. Файл `.bashrc`

```
# RuBackup API Settings
export APP_HOST=192.168.10.11
export APP_PORT=5655
export DB_HOST=localhost
export DB_PORT=5432
```

После этого необходимо перезагрузить переменные окружения:

```
. ~/.bashrc
```

Для запуска Swagger и Tusana на выделенном хосте в файл переменных окружения `/opt/rubakup/etc/rubakup_api.env` нужно установить следующие параметры:

```
APP_HOST= # IP или FQDN хоста, на котором установлен и будет запущен rest-api
APP_PORT=5656
DB_HOST= # IP или FQDN хоста с базой данных
DB_PORT=5432
RB_SERVER_HOST= # IP или FQDN хоста основного сервера rubakup
```

Глава 4. Настройка журналирования

Для обеспечения гибкости процесса журналирования действий сервера предусмотрен специальный конфигурационный файл, расположенный по пути `/opt/rubackup/etc/rubackup_api_logger.conf`. Лог-файлы располагаются в директории по пути `/opt/rubackup/logs/rubackup-api`.

4.1. Логгеры

В конфигурационном файле присутствует четыре типа логгеров:

Таблица 8. Типы логгеров

logger_root	логгер, от которого должны наследоваться все остальные. По умолчанию записывает сообщения уровня Info и выше
logger_rb_api	логирует общие ошибки и информационные сообщения по API. По умолчанию записывает сообщения уровня Info и выше. Ведёт запись в консоль и в файл
logger_rb_access	логирует инциденты, связанные с авторизацией. По умолчанию записывает сообщения уровня Info и выше. Ведёт запись в консоль и в файл
logger_tornado	логирует ошибки web-сервера Tornado. По умолчанию записывает сообщения уровня Info и выше. Ведёт запись в консоль и в файл

4.2. Хэндлеры

Для каждого логгера в конфигурационном файле присутствуют хэндлеры, в которых написано, как и какие ошибки обрабатывать и куда их записывать.

Всего есть три хэндлера:

Таблица 9. Типы логгеров

handler_file	перехватывает сообщения уровня Info, записывает их в <code>/opt/rubackup/log/rubackup_api/access/rubackup-api.log</code> , применяет определённое форматирование. Также после 00:00 создаёт новый файл для записи логов. Количество файлов журнала не превышает 15
handler_console	перехватывает сообщения уровня Info, применяет определённое форматирование и выводит их в консоль
handler_access	перехватывает сообщения уровня Info, применяет определённое форматирование, записывает инциденты, связанные с доступом, в <code>/opt/rubackup/log/rubackup_api/access/rubackup-api.access.log</code> . Также после 00:00 создаёт новый файл для записи логов. Количество файлов журнала не превышает 15

4.3. Форматеры

Описывают формат, в котором нужно записывать и/или отображать сообщения в

файле и/или консоли.

Дополнительную информацию по журналированию можно найти по ссылке <https://docs.python.org/3/library/logging.config.html#configuration-file-format>

Глава 5. Запуск и остановка сервиса

После установки пакета и настройки переменных окружения можно производить запуск RuBackup API.

Сервер RuBackup API представляет собой фоновое приложение (сервис, демон).

Расположение

```
/opt/rubackup/bin/rubackup_api
```

Запуск в терминальном режиме

```
rubackup_api --start
```

Остановка

```
rubackup_api --stop
```

Перезагрузка

```
rubackup_api --restart
```

Для штатной эксплуатации RuBackup API рекомендуется запустить его как сервис. Для этого выполните следующие действия:

Включите сервис RuBackup API:

```
sudo systemctl enable \  
/opt/rubackup/etc/systemd/system/rubackup_api.service
```

Перезагрузите systemctl:

```
sudo systemctl daemon-reload
```

Запустите сервис rubackup_api:

```
sudo systemctl start rubackup_api.service
```

Уточнить статус RuBackup API можно при помощи команды:

```
systemctl status rubackup_api.service
```

```
rubackup_api.service - RuBackup API
Loaded: loaded (/etc/systemd/system/rubackup_api.service; enabled; vendor
preset: enabled)
Active: active (running) since Tue 2024-05-07 22:06:24 MSK; 24min ago
Main PID: 69213 (rubackup_api)
Tasks: 2 (limit: 9430)
Memory: 61.0M
CGroup: /system.slice/rubackup_api.service
   69213 /bin/sh /opt/rubackup/bin/rubackup_api --start
   69214 /opt/rubackup/lib/rubackup_rest_api_lib/rubackup_api.bin --start
мая 07 22:06:24 rb-primary systemd[1]: Started RuBackup API.
мая 07 22:06:25 rb-primary rubackup_api[69214]: RuBackup API Logger
initializing
мая 07 22:06:26 rb-primary rubackup_api[69214]: 2024-05-07 22:06:26,066 -
[WARNING] - 'The rubackup database has not been initialized. Please
authenticate'
мая 07 22:06:26 rb-primary rubackup_api[69214]: 2024-05-07 22:06:26,070 -
[INFO] - 'RuBackup REST API is running on
[https://rubackup.api.local:5656/[https://rubackup.api.local:5656]]'
```



Сообщение 'The rubackup database has not been initialized. Please authenticate' является предупреждением пользователя о необходимости пройти аутентификацию хотя бы один раз для продолжения работы с сервисом. Для прохождения аутентификации воспользуйтесь [методом POST /auth/login](#) напрямую или Tucana.

Глава 6. Токены и аутентификация

6.1. Выпуск токенов

Перед тем как пользователь сможет обратиться к ресурсам сервера RuBackup, он должен пройти аутентификацию и получить токены доступа: `access_token`, `refresh_token` и `csrf_token`.

Сгенерированный `access_token` будет действовать в течение 15 минут с момента получения, `refresh_token` — 24 часа с момента получения, `csrf_token` действует до перезагрузки сервиса `rubackup_api`.

По истечении срока жизни `access_token` его можно перевыпустить с помощью `refresh_token`. Если истек срок жизни `refresh_token`, необходимо перевыпустить новую пару токенов с помощью логина и пароля. После перезапуска `rubackup_api` также необходимо перевыпустить новую пару токенов с помощью логина и пароля.

6.1.1. Выпуск токенов в браузере

1. Перейдите по адресу `https://<app_host>:<app_port>/api/v1/` (Рисунок 2):

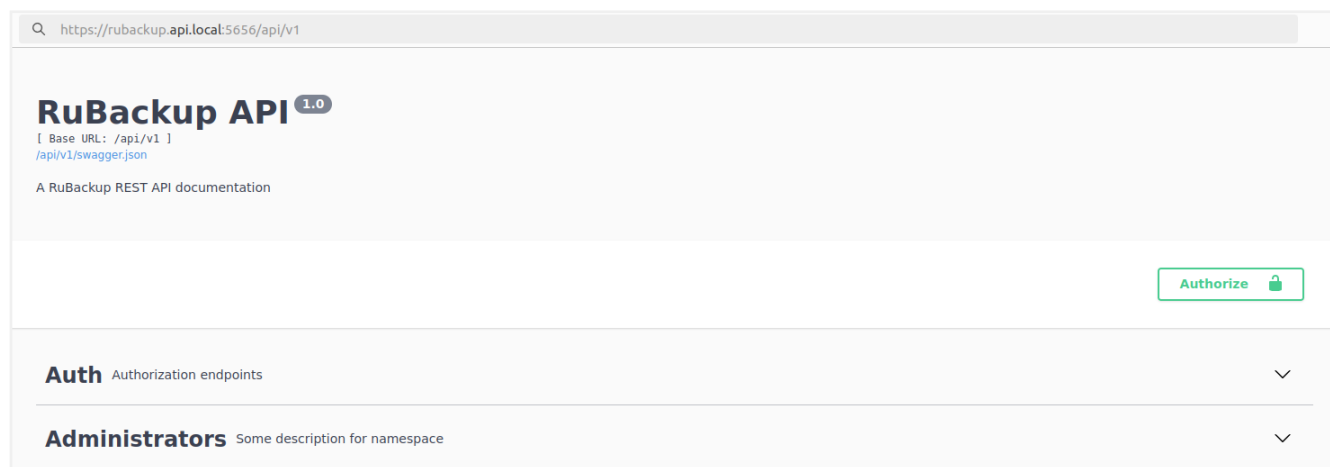


Рисунок 2. Внешний вид Swagger

1. Перейдите на вкладку «Auth» и выберите эндпоинт `/auth/login`.
2. Нажмите кнопку «Try it out», заполните payload актуальными данными и нажмите «Execute» (Рисунок 3):

Auth

Authorization endpoints

POST

/auth/login

⌵ 🔒

Endpoint description

Possible authentication types:

- domain;
- database.

Parameters

Cancel

Name	Description
payload object (body) ✖ required Edit Value Model	<pre>{ "rubakup_database": "rubakup", "rubakup_user": "rubakup", "password": "Testing12345", "auth_type": "database", "domain": "rubakup" }</pre> Cancel Parameter content type application/json

Execute

Clear

Рисунок 3. Заполнение payload авторизационными данными

В результате проделанных операций будут получены `access_token`, `refresh_token` и `csrf_token`, а также сопутствующая информация о пользователе прошедшем авторизацию (Рисунок 4). Также сервис автоматически разместит `access_token` и `refresh_token` в cookie-файлах.

Request URL

https://10.10.10.10:5656/api/v2/auth/login

Server response

Code	Details
200 Undocumented	Response body <pre>{ "data": { "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cGU6IjYwIiwiaXNjaW50IjpmYW93bmJmIjozeDZmOTI0MDIyLkCjc3MlMTQ4ZmYxOTg0YzY3ZW80MDk4LWE3MDItYmViOGE4NGM3ZGV0IiwiaWF0IjoxNzAzOTI0OTIyLkCjYmY1pbHkiOiJpbGxldHJxcGRjcXZemWntIn0.uBKK6zb9JWLfYh9M92N3baAct8Fbbn8fm10Kv3AMs", "csrf_token": "148ff198-c67e-4098-a702-beb8a14c7df4", "last_fail_login": null, "last_success_login": "2024-12-11T16:12:50.649160", "package_version": "2", "refresh_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cGU6IjYwIiwiaXNjaW50IjpmYW93bmJmIjozeDZmOTI0MDIyLkCjc3MlMTQ4ZmYxOTg0YzY3ZW80MDk4LWE3MDItYmViOGE4NGM3ZGV0IiwiaWF0IjoxNzAzOTI0OTIyLkCjYmY1pbHkiOiJpbGxldHJxcGRjcXZemWntIn0.uBKK6zb9JWLfYh9M92N3baAct8Fbbn8fm10Kv3AMs", "role": { "superuser": true }, "rubackup_server_address": "10.10.10.10", "user_name": "ruBackup" }, "errors": {}, "is_error": false, "message": "" }</pre>

Response headers

```
access-control-allow-credentials: true
access-control-allow-origin: https://10.10.10.10:5656
content-length: 1174
content-type: application/json
server: TornadoServer/6.2
vary: Origin
```

Рисунок 4. Ответ сервера с выпущенными токенами



Об авторизации через `auth_type: domain` — в [Раздел 6.5](#).

6.1.2. Выпуск токенов в командной строке

Чтобы произвести выпуск пары токенов через терминал, необходимо отправить POST-запрос с помощью консольной утилиты `curl` или любым другим удобным способом. В данном примере используется `curl`:

```
curl -k -X POST 'https://api.rubackup.local:5656/api/v1/auth/login' \ ❶ ❷ ❸
-H 'accept: application/json' \ ❹
-H 'Content-Type: application/json' \ ❹
-d '{
  "rubackup_database": "rubackup", \ ❺
  "rubackup_user": "rubackup", \ ❻
  "password": "Testing12345", \ ❼
  "auth_type": "database", \ ❽
  "domain": "rubackup" ❾
}'
```

- ❶ `https://api.rubackup.local` — адрес, где запущен `rubackup_api`
- ❷ `5656` - порт, который будет использоваться как часть адреса сервера API
- ❸ `/api/v1/auth/login` — путь до запроса на авторизацию
- ❹ `-H <argument>` - значения, передаваемые в заголовках
- ❺ `rubackup_database` — имя служебной базы данных
- ❻ `rubackup_user` — имя суперпользователя
- ❼ `password` — пароль суперпользователя
- ❽ `auth_type` — тип аутентификации (база данных или контроллер домена)
- ❾ `domain` — домен для авторизации через контроллер домена

Результат вернётся в формате JSON:

```
{
  "data": {
    "access_token":
      "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJmcmVzaCI6ZmFsc2UsIm1hdCI6MTcxNTE0Dc
      2MCwianRpIjoInThjZjQ5MDAtZjE5Ni00OTZlLThkZjQtMTMhZjY2M3Y2QxIiwidHlwZSI6ImFj
      Y2VzcyIsInN1YiI6InJ1bmFja3VwIiwibmJmIjojNzE1MTU0NzYwLCJjc3JmIjojNGM4YzY0NTctN
      DAwNS00YzZiLTk2MWMtMGM1MDU5Njc5ZTA2IiwiaXNjaXNzE1MTU5NjYwLCJmYW1pbHkiOiJ4cn
      ByeHR3ZHJyZ3FsamN4In0.VK5K6v-0_NxSx42bU5dEMIQAYYzxn-GTxmbjhxXjYSs",
    "csrf_token": "4c8c3457-4005-4c6b-961c-0c5059671e06",
    "csrf_token": "148ff198-c67e-4098-a702-beb8a14c7df4",
    "last_fail_login": "2024-12-11T17:39:43.564638",
    "last_success_login": "2024-12-11T17:40:07.036618",
    "package_version": "<версия пакетов>",
  }
}
```

```

    "refresh_token":
    "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJmcmVzaCI6ZmFsc2UsImhhdCI6MTcxNTE0Dc2MCwianRpIjoiotUzZjAxZWYtMTM4Zi00Y2ZlTg4ODItNGI0NWQ0N2YxM2I2IiwidHlwZSI6InJlZnJlc2giLCJzdWIiOiJydWJhY2t1cCIsIm5iZiI6MTcxNTE0Dc2MCwiY3NyZiI6IjRjOGMzNDU3LTQwMDUtNGM2Yi05NjFjLTBjNTA1OTY3MWUwNiIsImV4cCI6MTcxNTI0NTE2MCwiZmFtaWx5IjoieHJwcnh0d2RycmdxbGpjcCJ9.7TmdI0Cmm4knApNINDoYJuJIYdRlzuuc1hS-1c4Y8Ws",
    "role": [
        "superuser"
    ],
    "rubackup_server_address": "10.177.xx.xxx",
    "user_name": "rubackup"
},
"errors": {},
"is_error": false,
"message": ""
}

```

Если необходимо получить еще и access_token и refresh_token из cookies, то `curl` следует передать опцию `--cookie-jar -`, например:

```

curl -k --cookie-jar - -X POST
'https://api.rubackup.local:5656/api/v1/auth/login' \ ❶
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "rubackup_database": "rubackup", \
  "rubackup_user": "rubackup", \
  "password": "Testing12345", \
  "auth_type": "database", \
  "domain": "rubackup"
}'

```

❶ Добавлена опция `--cookie-jar -`

С этой опцией к выводу добавится следующая информация:

```

#HttpOnly.rubackup.local FALSE / TRUE 0 refresh_token_cookie
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJmcmVzaCI6ZmFsc2UsImhhdCI6MTcxNTE0Dc2MDUzNywianRpIjoiyZM4MTE0N2YtNDI0MS00Y2NjLTlhZjYtNDcwODAxMWFmOTRjIiwidHlwZSI6InJlZnJlc2giLCJzdWIiOiJydWJhY2t1cCIsIm5iZiI6MTcxNTE0Dc2MDUzNywiY3NyZiI6IjRjOGMzNDU3LTQwMDUtNGM2Yi05NjFjLTBjNTA1OTY3MWUwNiIsImV4cCI6MTcxNTI0NjRjZmFtaWx5IjoieXRqemR4dG1xd212bHlyayJ9.g0aFkoob7jAwZ5Bv6FzbOPo3_Q6y-vBpASqXXk19tyw

#HttpOnly.rubackup.local FALSE / TRUE 0 access_token_cookie

```

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJmcmVzaCI6ZmFsc2UsIm1hdCI6MTcxNTE2MDUzNywianRpIjoiodmNDI0N2Q0NjJlMy00ZjFjLTk4NzgtNDY4OTM2YjRlNTJmIiwidHlwZSI6ImFjY2VzcyIsInN1YiI6InJ1YmFja3VwIiwibmJmIjoxNzE1MTYwNTM3LCJjc3JmIjoingM4YzM0NTctNDAwNS00YzZiLTk2MWMtMGM1MDU5Njc0ZTA2IiwiaXNjaXNzE1MTYxNDM3LCJmYW1pbHkiOiJ5dGp6ZWh0bXF3bXZseXJrIn0.nj-UbudgyYqGmopRpK9du9gPaZcn_j0_5yhp0A5DLw8
```

6.2. Обновление токенов

6.2.1. Обновление токенов в браузере

1. Перейдите по адресу `https://<app_host>:<app_port>/api/v1/`.
2. Перейдите на вкладку «Auth» и выберите эндпоинт `/auth/refresh` (Рисунок 5):

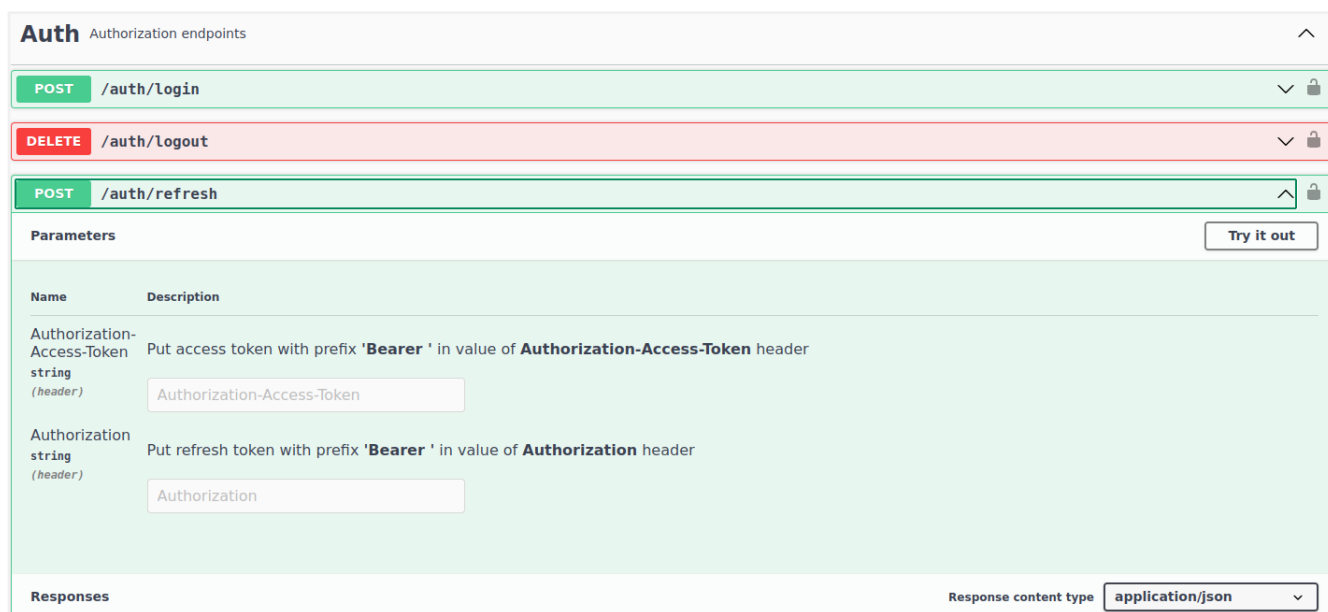


Рисунок 5. Эндпоинт `/auth/refresh`

1. Нажмите кнопку «Try it out». Если авторизация была пройдена в этом же браузере и `access_token` и `refresh_token` все ещё находятся в cookie, то нажмите кнопку «Execute». В ином случае явно укажите токены с префиксом `Bearer` для параметров `Authorization-Access-Token` и `Authorization` (Рисунок 6).

Auth Authorization endpoints

POST	<code>/auth/login</code>	↕ 🔒
DELETE	<code>/auth/logout</code>	↕ 🔒
GET	<code>/auth/permissions</code> Current user's permissions	↕ 🔒
POST	<code>/auth/refresh</code>	↗️ 🔒

Parameters Cancel

Name	Description
Authorization-Access-Token string (header)	Put access token with prefix ' Bearer ' in value of Authorization-Access-Token header
Authorization-string (header)	Put refresh token with prefix ' Bearer ' in value of Authorization header

Execute

Рисунок 6. Заполнение полей авторизации

В результате проделанных операций Вам вернётся перевыпущенный `access_token`, перевыпущенные `refresh_token` и `csrf_token`, а также сопутствующая информация о пользователе прошедшем авторизацию (Рисунок 7). Также сервис автоматически разместит перевыпущенные `access_token` и `refresh_token` в cookie-файлах.

```
Curl
curl -X 'POST' \
  'https://10.10.10.10:5656/api/v1/auth/refresh' \
  -H 'accept: application/json' \
  -H 'X-CSRF-TOKEN: 0497040f-ac65-4859-aa0a-1e9b0959bdef' \
  -d ''

Request URL
https://10.10.10.10:5656/api/v1/auth/refresh

Server response
200
[Documented] Response body
{
  "data": {
    "access token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cGU6IjYwOTQwNTM2ZG0tMTB0ZC00OWhlLWlyNDU0NjE1YThtMTFjZTRjIiwidHlwZSI6ImFjY2VzcycySiNMIiY1IjbnJmYja3VwIiwibmJmIjoiaXozIzhjOWMzY1IjLCJpc3NmIjoiodMOQ5NzA0MGYtYmZNS00ODUSLWFhMGctTmUsYjEASNTLiZGVmIiwizXhwIjoxnZiI2MjQxMjY1Y1IjLCJmY1pbHkiOiIvZmdkd2lxZG5zbWl6bHphIn0.r-z6ofacy3M4QLrWbaXhMt-irZPxdvYaQSLedumJo",
    "csrf token": "0497040f-ac65-4859-aa0a-1e9b0959bdef",
    "package version": "2.0.0",
    "refresh token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cGU6IjYwOTQwNTM2ZG0tMTB0ZC00OWhlLWlyNDU0NjE1YThtMTFjZTRjIiwidHlwZSI6ImFjY2VzcycySiNMIiY1IjbnJmYja3VwIiwibmJmIjoiaXozIzhjOWMzY1IjLCJpc3NmIjoiodMOQ5NzA0MGYtYmZNS00ODUSLWFhMGctTmUsYjEASNTLiZGVmIiwizXhwIjoxnZiI2MjQxMjY1Y1IjLCJmY1pbHkiOiIvZmdkd2lxZG5zbWl6bHphIn0.r-z6ofacy3M4QLrWbaXhMt-irZPxdvYaQSLedumJo",
    "role": "superuser"
  },
  "rubackup_server_address": "10.10.10.10",
  "user_name": "rubackup"
},
"errors": {},
"is_error": false,
"message": ""
}

Response headers
access-control-allow-credentials: true
access-control-allow-origin: https://10.177.33.245:5656
content-length: 1082
content-type: application/json
server: Tornadover/6.2
vary: Origin
```

Рисунок 7. Ответ с перевыпущенными токенами

6.2.2. Обновление токенов в терминале

Чтобы произвести перевыпуск пары токенов через терминал, необходимо отправить POST-запрос с помощью консольной утилиты `curl` или любым другим удобным способом. В данном примере используется `curl`:

```
curl -k -X POST 'https://api.rubackup.local:5656/api/v1/auth/refresh' \ ① ②
③
-H 'accept: application/json' \ ④
-H 'Authorization-Access-Token: Bearer <access_token>' \ ④ ⑤
-H 'Authorization: Bearer <refresh_token>' ④ ⑥
```

- ① `https://api.rubackup.local` — адрес, где запущен `rubackup_api`
- ② `5656` - порт, который будет использоваться как часть адреса сервера API
- ③ `/api/v1/auth/refresh` — путь до запроса на обновление токена
- ④ `-H <argument>` - значения, передаваемые в заголовках
- ⑤ `Authorization-Access-Token` — значение полученного при авторизации `access_token` с префиксом `Bearer`
- ⑥ `Authorization` — значение полученного при авторизации `refresh_token` с префиксом `Bearer`

Результат вернётся в формате JSON:

```
{
  "data": {
    "access_token":
      "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJmcmVzaCI6ZmFsc2UsIm1hdCI6MTcxNTE2MjAwOSwianRpIjoiodDhlZDcwYzUtM2Y4Yi00MDVmlWJlMjMtYzYzNTA3YzA5MmViIiwidHlwZSI6ImFjY2VzcyIsInN1YiI6InJ1YmFja3VwIiwibmJmIjoxNzE1MTYyMDA5LCJjc3JmIjoingM4YzMTc0NTctNDh1Y250cWVja29ma2p0In0.XHppz9B-eYoEcXZAwcf-nbXKnu8rC_kXIMRWIU4gZxc",
    "csrf_token": "4c8c3457-4005-4c6b-961c-0c5059671e06",
    "package_version": "<version>",
    "refresh_token":
      "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJmcmVzaCI6ZmFsc2UsIm1hdCI6MTcxNTE2MjAwOSwianRpIjoioGNlYmY4NGMtOTM4M000YTLhLWE1OGItNmU4ZDM3OTgxMTI0IiwidHlwZSI6InJ1ZnJlc2giLCJzdWIiOiJydWJhY2t1cCI6Im5iZi00MTcxNTE2MjAwOSwiy3NyZiI6ImRjOGMzNDU3LTQwMDUtNGM2Yi05NjFjLTBjNTA1OTY3MWUwNiIsImV4cCI6MTcxNTI0ODM3MCwiZmFtaWx5IjoizXl4dWNudHFLY2tvZmtqdCJ9.H_TPOn-CF70bYUKa5AP4U10MsiPEnzy5foxTZXHi0dE",
    "role": [
      "superuser"
    ],
    "rubackup_server_address": "10.177.xx.xxx",
    "user_name": "rubackup"
  },
  "errors": {},
  "is_error": false,
  "message": ""
}
```

```
}
```

Если необходимо получить еще и `access_token` и `refresh_token` из cookies, то в команде `curl` следует передать опцию `--cookie-jar -`, например:

```
curl -k --cookie-jar - -X POST
'https://api.rubackup.local:5656/api/v1/auth/refresh' \
-H 'accept: application/json' \
-H 'Authorization-Access-Token: Bearer <access_token>' \
-H 'Authorization: Bearer <refresh_token>'
```

С этой опцией к выводу добавится следующая информация:

```
#HttpOnly.rubackup.local FALSE / TRUE 0 refresh_token_cookie
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJmcmVzaCI6ZmFsc2UsIm1hdCI6MTcxNTE2MjAw
OSwianRpIjoIj0GNlYmY4NGMtOTM4MS00YTlhLWE1OGItNmU4ZDM3OTgxMTI0IiwidHlwZSI6InJlZ
nJlc2giLCJzdWIiOiJydWJhY2t1cCIscm5iZiI6MTcxNTE2MjAwOSwiY3NyZiI6IjRjOGMzNDU3LT
QwMDUtNGM2Yi05NjFjLTBjNTA1OTY3MWUwNiIsImV4cCI6MTcxNTI0DM3MCwiZmFtaWx5IjoIjXl
4dWNudHF1Y2tvZmtqdCJ9.H_TP0n-CF70bYUKa5AP4U10MsiPEnzy5foxTZXHidE

#HttpOnly.rubackup.local FALSE / TRUE 0 access_token_cookie
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJmcmVzaCI6ZmFsc2UsIm1hdCI6MTcxNTE2MjAw
OSwianRpIjoIj0DhlZDcwYzUtM2Y4Yi00MDVmLWJlMjMtYzcvNTA3YzA5MmViIiwidHlwZSI6ImFjY
2VzcyIsInN1YiI6InJlYmFja3VwIiwibmJmIjoIjoxNzE1MTYyMDA5LCJjc3JmIjoIj0NGM4YzY0NTctND
AwNS00YzZiLTk2MWMtMGM1MDU5NjcxZTA2IiwidXhwIjoIjoxNzE1MTYyOTA5LCJmYW1pbHkiOiJleXh
1Y250cWVja29ma2p0In0.XHppz9B-eYoEcXZAwcf-nbXKnu8rC_kXIMRWIU4gZXc
```

6.3. Отзыв токенов

Если ваши токены были скомпрометированы, то можно сделать отзыв двумя способами: через терминал с помощью консольной команды `curl`, либо через web-интерфейс.

Для отзыва токенов через терминал отправьте DELETE-запрос с RefreshToken с помощью консольной утилиты `curl` или любым другим удобным для вас способом.

В данном примере используется `curl`:

```
curl -k -X DELETE https://api.rubackup.local:5656/api/v1/logout \
-H 'Content-Type: application/json' \
-H 'Authorization: <refresh_token>'
```

Для отзыва токенов с помощью web-интерфейса необходимо выполнить следующие действия:

Перейдите по адресу `https://<app_host>:<app_port>/api/v1/` (Рисунок 8):

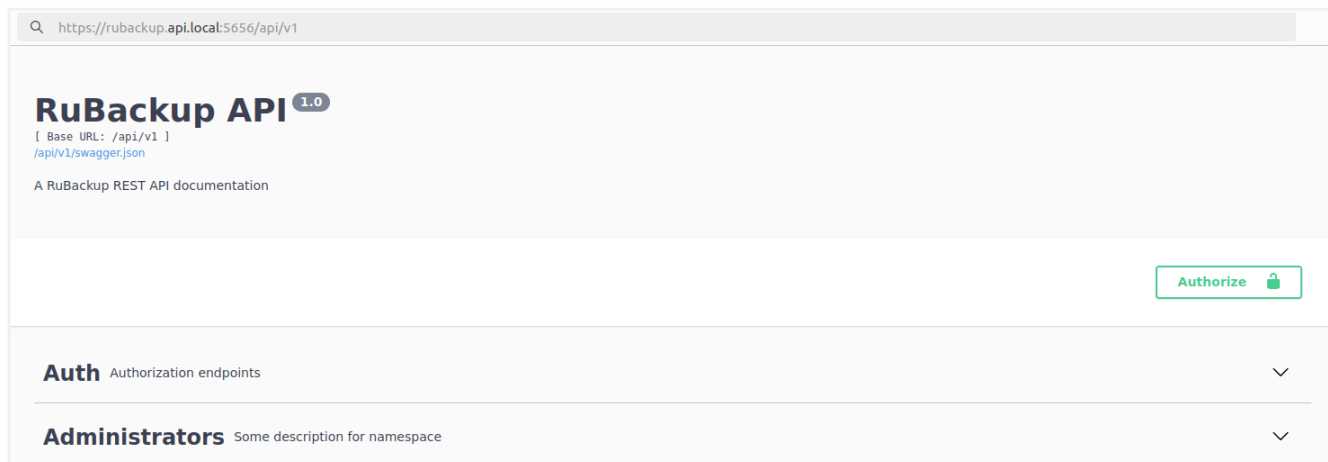


Рисунок 8. Внешний вид Swagger

Перейдите на вкладку «Auth» и выберите эндпоинт `/auth/logout` (Рисунок 9). Предварительно убедитесь, что выполнена авторизация (подробнее об авторизации написано в разделе «Авторизация пользователя в веб-интерфейсе»).

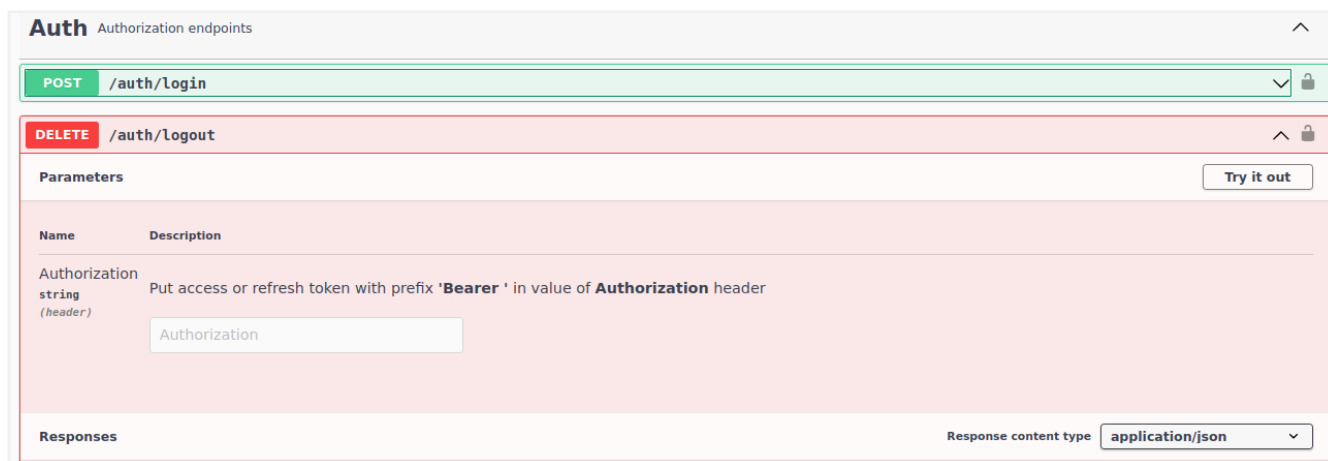


Рисунок 9. Эндпоинт `/auth/logout`

Нажмите кнопку «Try it out». Если авторизация была пройдена в этом же браузере и `access_token` и `refresh_token` все ещё находятся в cookie, то нажмите кнопку «Execute» (Рисунок 10). В ином случае явно укажите отзываемый RefreshToken (с префиксом `Bearer`) для параметра `Authorization`.

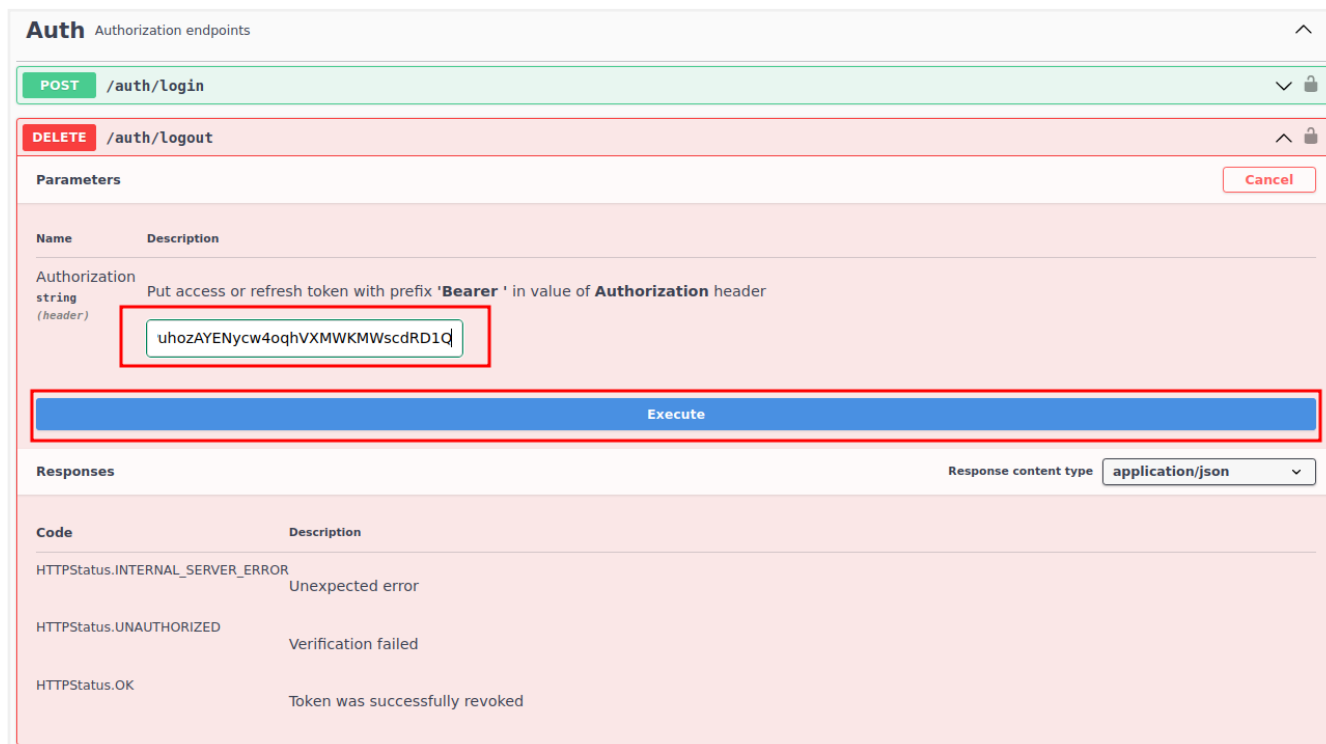


Рисунок 10. Отзываемый токен

В результате проделанных операций вернётся информационное сообщение о том, что токены деактивированы ([Рисунок 11](#)).

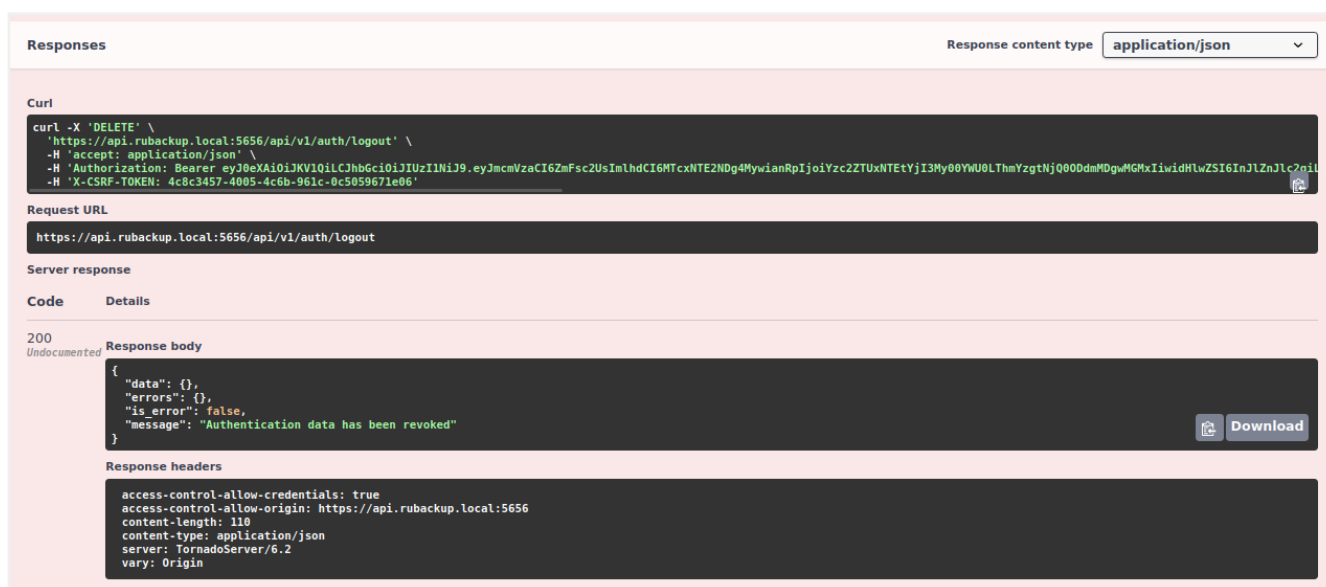


Рисунок 11. Сообщение об успешном отзыве токена

6.4. Аутентификация пользователя

Для начала работы с ресурсами сервера перейдите по адресу `https://<app_host>:<app_port>/api/v1/` в Вашем браузере, выполните получение токенов любым удобным способом и нажмите на кнопку «Authorize» (Рисунок 12):



Рисунок 12. Кнопка авторизации

В открывшейся форме заполните поле «Value». В данном поле необходимо ввести ранее полученный csrf_token. Затем нажмите кнопку «Authorize» ([Рисунок 13](#)):

Рисунок 13. Форма ввода токена авторизации

Закройте форму нажатием на кнопку «Close» ([Рисунок 14](#)). Теперь вы авторизованы и можете работать с ресурсами сервера RuBackup.

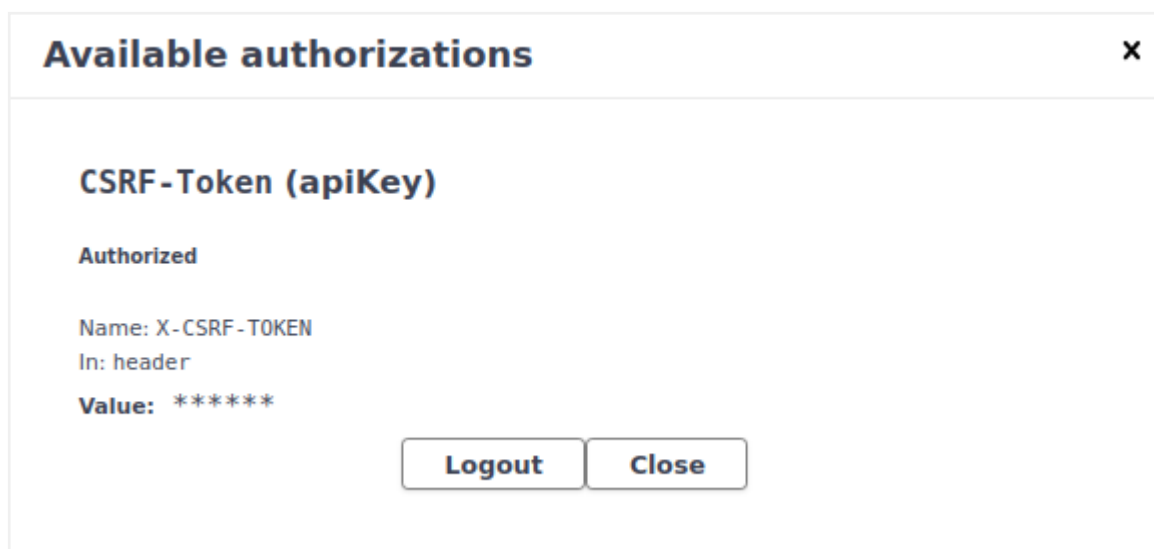


Рисунок 14. Кнопка закрытия формы

6.5. Аутентификация с Microsoft Active Directory

Аутентификация в REST API возможна через контроллер домена Microsoft Active Directory.

CPK поддерживает интеграцию с Microsoft Active Directory версий 2012 R2 или 2016, развернутой на Microsoft Windows Server 2016.



Предварительные условия для аутентификации через контроллер домена

1. Настроен контроллер домена Microsoft Active Directory. В контроллере домена существует хотя бы одна группа, в которой есть хотя бы один пользователь.
2. В CPK настроено соединение с контроллером домена Microsoft Active Directory.
3. В CPK создана ассоциация групп RuBackup и ролей MS Active Directory.

Подробнее о настройке контроллера домена MS Active Directory — в [Интеграция с Microsoft Active Directory](#).

6.5.1. Выпуск `access_token`, `refresh_token` и `csrf_token` через браузер

Перед авторизацией убедитесь, что все [предварительные условия](#) для аутентификации через контроллер домена выполнены.

Для примера будут использованы следующие данные MS AD:

- Группа контроллера домена: `test_group`
- Пользователь контроллера домена: `testuser`
- Роль группы: Супервайзер

- Домен: RUBACKUP
- Пароль пользователя в контроллере: P@ssw0rd

Для выпуска пары токенов:

1. Перейдите по адресу `https://<app_host>:<app_port>/api/v1/` (Рисунок 15):

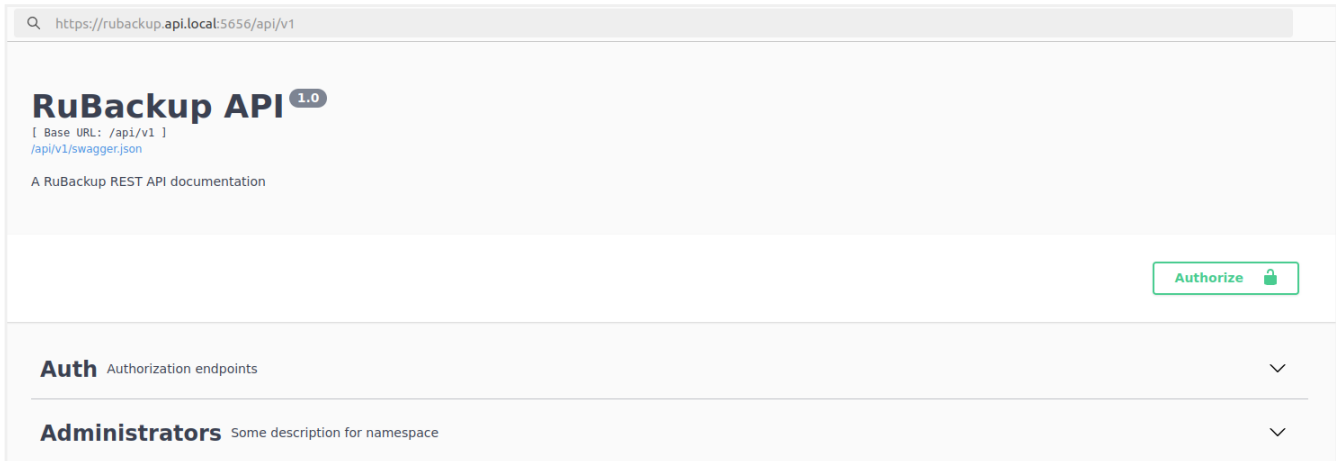


Рисунок 15. Внешний вид Swagger

1. Перейдите на вкладку «Auth» и выберите эндпоинт `/auth/login`.
2. Нажмите кнопку «Try it out», заполните payload актуальными данными и нажмите «Execute» (Рисунок 16):

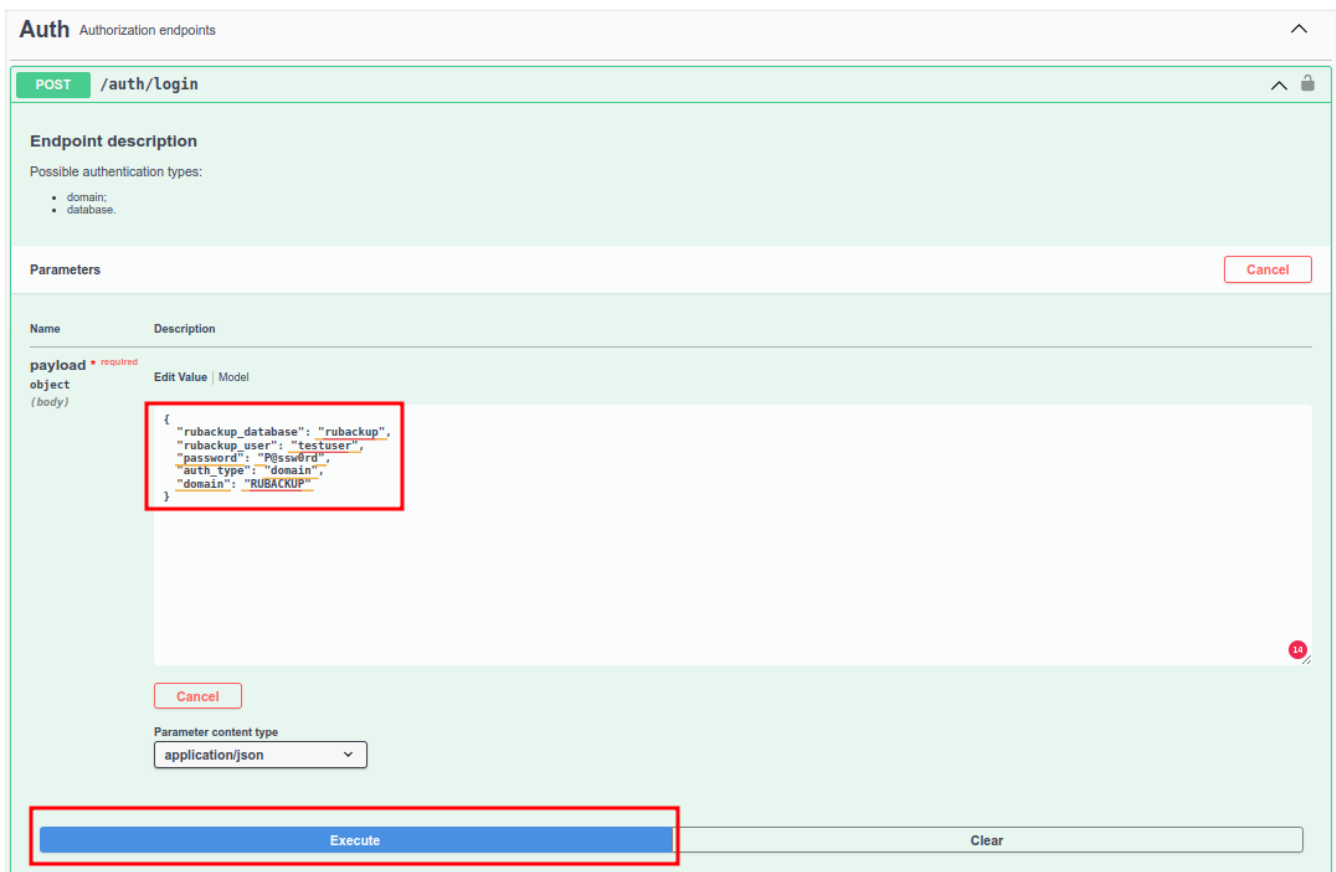


Рисунок 16. Заполнение payload авторизационными данными

В результате получены `access_token`, `refresh_token` и `csrf_token`, а также сопутствующая информация о пользователе, прошедшем авторизацию (Рисунок 17). Сервис автоматически разместит `access_token` и `refresh_token` в cookie-файлах.

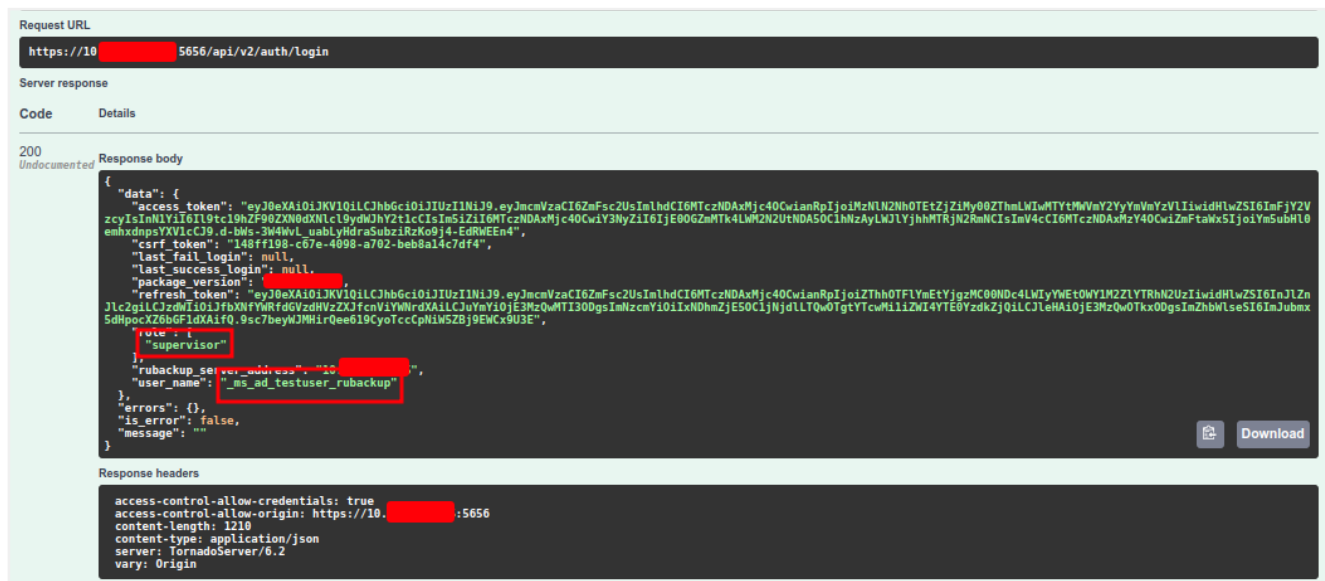


Рисунок 17. Ответ сервера

В параметре `role` будет возвращена роль пользователя, которая соответствует роли группы, в которой находится пользователь. Роль группы определяется через ассоциацию групп и ролей. Подробнее о создании ассоциации — в «Интеграция RuBackup со средствами управления доменом Microsoft Active Directory».

6.5.2. Выпуск `access_token`, `refresh_token` и `csrf_token` в командной строке

Перед авторизацией убедитесь, что все условия из раздела Предварительные условия для аутентификации через контроллер домена выполнены.

Для примера будут использованы следующие данные MS AD:

- Группа контроллера домена: `test_group`
- Пользователь контроллера домена : `testuser`
- Роль группы: Супервайзер
- Домен: `RUBACKUP`
- Пароль пользователя в контроллере: `P@ssw0rd`

Чтобы произвести выпуск пары токенов через терминал, необходимо отправить POST-запрос с помощью консольной утилиты `curl` или любым другим удобным способом. В данном примере используется `curl`:

```
curl -k -X POST 'https://api.rubackup.local:5656/api/v1/auth/login' \ ❶ ❷ ❸
--header 'Content-Type: application/json' \ ❹
--data '{
```

```
"rubbackup_database": "rubbackup", ⑤
"rubbackup_user": "testuser", ⑥
"password": "P@ssw0rd", ⑦
"auth_type": "domain", ⑧
"domain": "RUBACKUP" ⑨
}'
```

- ① `https://api.rubbackup.local` — адрес, где запущен `rubbackup_api`
- ② `5656` - порт, который будет использоваться как часть адреса сервера API
- ③ `/api/v1/auth/login` — путь до запроса на авторизацию
- ④ `--header <argument>` - значения, передаваемые в заголовках
- ⑤ `rubbackup_database` — имя служебной базы данных
- ⑥ `rubbackup_user` — имя суперпользователя
- ⑦ `password` — пароль суперпользователя
- ⑧ `auth_type` — тип аутентификации (база данных или контроллер домена)
- ⑨ `domain` — домен для авторизации через контроллер домена

Результат вернётся в формате JSON:

```
{
  "data": {
    "access_token":
      "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJmcmVzaCI6ZmFsc2UsIm1hdCI6MTczNDExNDMzNCwianRpIjoizWRjYzY0NzItYTdhOC00MmQ5LWlxNmMtMWE4NzQ2ZTI5NWNkIiwidHlwZSI6ImFjY2VzcyIsInN1YiI6Iml9tc19hZF90ZXN0dXNlc19ydWJhY2t1cCI6Im5iZiI6MTczNDExNDMzNCwiY3NyZiI6IjE0OGZmMTk4LW02N2U0tNDU0c01hNzAyLWJlYjhhMTRjN2RmNCIsImV4cCI6MTczNDExNTIzNCwiZmFtaWx5IjoiaH0Z3hkcWFobmxcncWdidCJ9.5YU2yxGZshimv6kqqgvP-t6ynn timerVxDhlGurcXFHg0s",
    "csrf_token": "148ff198-c67e-4098-a702-beb8a14c7df4",
    "last_fail_login": null,
    "last_success_login": "2024-12-12T17:13:08.305226",
    "package_version": "<Версия пакетов>",
    "refresh_token":
      "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJmcmVzaCI6ZmFsc2UsIm1hdCI6MTczNDExNDMzNCwianRpIjoizWRjYzYzZDQ0YzIwIm00MmI0LWlxZjItNDI4MWIwZWZmNGY0IiwidHlwZSI6ImFjY2VzcyIsInN1YiI6Iml9tc19hZF90ZXN0dXNlc19ydWJhY2t1cCI6Im5iZiI6MTczNDExNDMzNCwiY3NyZiI6IjE0OGZmMTk4LW02N2U0tNDU0c01hNzAyLWJlYjhhMTRjN2RmNCIsImV4cCI6MTczNDExNTIzNCwiZmFtaWx5IjoiaH0Z3hkcWFobmxcncWdidCJ9.5YU2yxGZshimv6kqqgvP-t6ynn timerVxDhlGurcXFHg0s",
    "role": [
      "supervisor"
    ],
  },
}
```

```

    "rubackup_server_address": "10.177.xxx.xxx",
    "user_name": "_ms_ad_testuser_rubackup"
  },
  "errors": {},
  "is_error": false,
  "message": ""
}

```

Если необходимо получить еще и `access_token` и `refresh_token` из cookies, то в команде `curl` следует указать опцию `--cookie-jar -`:

```

curl -k -X --cookie-jar - POST
'https://api.rubackup.local:5656/api/v1/auth/login' \ ❶
--header 'Content-Type: application/json' \
--data '{
  "rubackup_database": "rubackup",
  "rubackup_user": "testuser",
  "password": "P@ssw0rd",
  "auth_type": "domain",
  "domain": "RUBACKUP"
}'

```

❶ Добавлена опция `--cookie-jar -`

С этой опцией к выводу добавится следующая информация:

```

#HttpOnly.rubackup.local FALSE / TRUE 0 refresh_token_cookie
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJmcmVzaCI6ZmFsc2UsImhhdCI6MTcxNTE2MDUz
NywianRpIjoIYzM4MTE0N2YtNDI0MS00Y2NjLTlhZjYtNDcwODAxMWFmOTRjIiwidHlwZSI6InJlZ
nJlc2giLCJzdzIiOiJydWJhY2t1cCIsc29iOiJmcmVzaCI6ZmFsc2UsImhhdCI6MTcxNTE2MDUzNywiY3NyZiI6IjRjOGMzNDU3LT
QwMDU0NGM2Yi05NjFjLTBjNTA1OTY3MwUwNiIsImV4cCI6MTcxNTI0NjkzNywiZmFtaWx5IjoieXR
qemR4dG1xd212bHlyayJ9.g0aFkoob7jAwZ5Bv6Fzb0Po3_Q6y-vBpASqXXk19tyw__
#HttpOnly_.rubackup.local FALSE / TRUE 0 access_token_cookie
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJmcmVzaCI6ZmFsc2UsImhhdCI6MTcxNTE2MDUz
NywianRpIjoI0DhmNDI0N2Q0NjJlMy00ZjFjLTk4NzgtNDY4OTM2YjRlNTJmIiwidHlwZSI6ImFjY
2VzcyIsInN1YiI6InJlYmFja3VwIiwibmJmIjojNzE1MTYwNTM3LCJjc3JmIjoIbG4yZm0NTctND
AwNS00YzZiLTk2MWMtMGM1MDU5Njc0ZTA2IiwiaXhwIjojNzE1MTYxNDM3LCJmYW1pbHkiOiJ5dGp
6ZHh0bXF3bXZseXJrIn0.nj-UbudgyYqGmopRpK9du9gPaZcn_j0_5yhp0A5DLw8

```

Глава 7. Отправка запросов

7.1. Отправка запросов в браузере

Для отправки запроса на создание срочной резервной копии в браузере необходимо выполнить следующие действия:

Перейдите по адресу `https://<app_host>:<app_port>/api/v1/` (Рисунок 18):

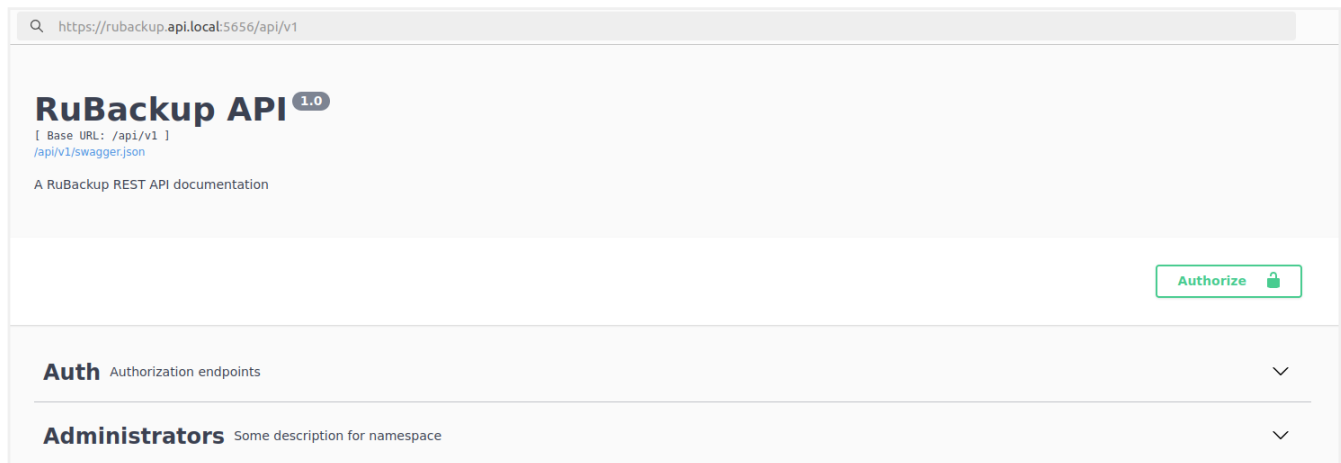


Рисунок 18. Внешний вид интерактивной документации Swagger

Перейдите к разделу Task queue и выберите метод POST (Рисунок 19):

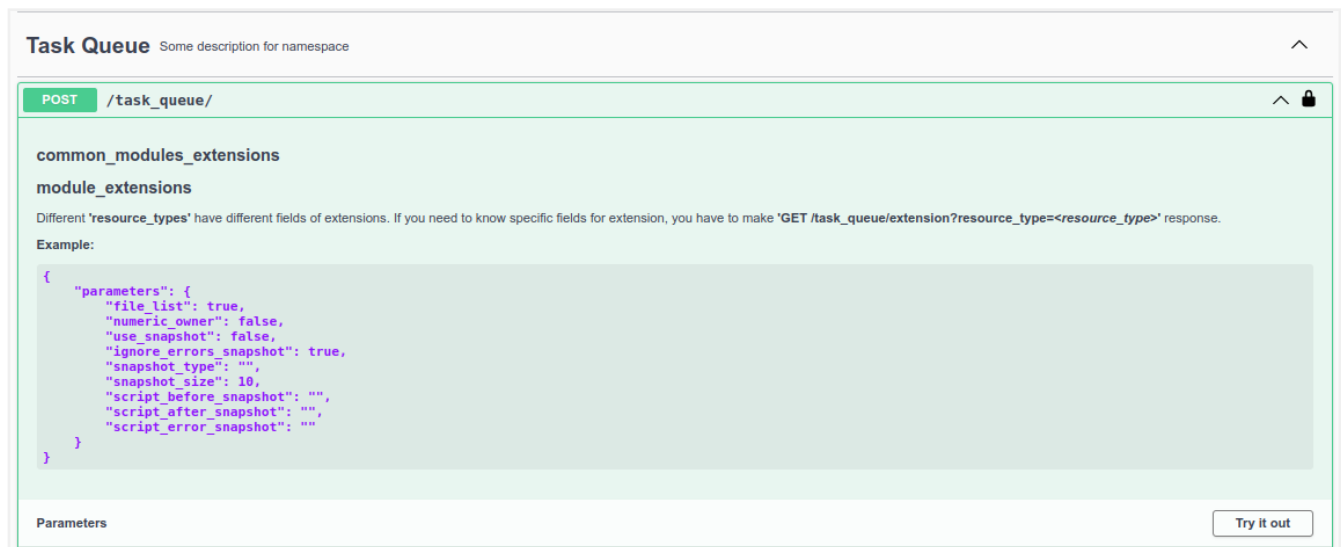


Рисунок 19. Метод `/task_queue`

Нажмите кнопку «Try it out». Заполните обязательные поля тела запроса, например (для создания копии файлового ресурса):

```
{
  "data": {
    "task": {
```

```

"resource_type": "File system",
"resource": "<имя ресурса>",
"client": "<client hostname> (<client HW ID>)",
"backup_type": "full",
"pool": "<имя пула>",
"crypto": "nocrypt",
"priority": 100,
"storage_duration": "1 Years",
"archiving": false
}
}
}

```

Нажмите «Execute» (Рисунок 20):

The screenshot shows the 'Parameters' tab of a REST client. A table lists parameters with columns 'Name' and 'Description'. The 'payload' parameter is marked as 'required' and its value is a JSON object:


```
{
  "data": {
    "task": {
      "resource_type": "File system",
      "resource": "<имя ресурса>",
      "client": "<client hostname> (<client HW ID>)",
      "backup_type": "full",
      "pool": "<имя пула>",
      "crypto": "nocrypt",
      "priority": 100,
      "storage_duration": "1 Years",
      "archiving": false
    }
  }
}
```

 The 'Parameter content type' is set to 'application/json'. At the bottom, the 'Execute' button is highlighted with a red box, and a 'Clear' button is next to it. The 'Responses' section at the bottom shows the 'Response content type' set to 'application/json'.

Рисунок 20. Кнопка **Execute**

В результате должен быть получен статус-код 201 CREATED и ответ с информацией о том, что задача добавлена в очередь (Рисунок 21):

The screenshot shows the 'Code' tab of the REST client. It displays a '201' status code and the 'Response body' as a JSON object:


```
{
  "is_error": false,
  "message": "Resource with name 'Task Queue' was created successfully! TASK WAS ADDED TO QUEUE",
  "data": {},
  "errors": {}
}
```

 Below the response body, the 'Response headers' are listed:


```
access-control-allow-credentials: true
access-control-allow-origin: https://10...:5656
content-length: 162
content-type: application/json
server: TornadoServer/6.2
vary: Origin
```

 A 'Download' button is visible next to the response body. The 'Responses' section at the bottom shows the 'Response content type' set to 'application/json'.

Рисунок 21. Успешный ответ сервера

Для отслеживания статуса выполнения задачи можно использовать [метод GET](#)

/task_queue.

7.2. Отправка запросов в терминале

Для того, чтобы отправить запрос на сервер, используя терминал, необходимо отправить запрос с помощью консольной утилиты `curl` или любым другим удобным способом. В данном примере используется `curl`. В запросе необходимо передать полученные ранее любым удобным способом `csrf-token` и `access-token`.

Для того, чтобы отправить запрос на создание срочной резервной копии файлового ресурса, необходимо выполнить:

```
curl --location 'https://<app_host>:<app_port>/api/v1/task_queue/' \
--header 'Content-Type: application/json' \
--header 'X-CSRF-TOKEN: <полученный при авторизации csrf-token>' \
--header 'Authorization: Bearer <полученный при авторизации access-token>' \
--data '{
  "data": {
    "task": {
      "client": "<client hostname> (<client HW ID>)",
      "resource": "<имя ресурса>",
      "resource_type": "File system",
      "backup_type": "full",
      "pool": "<имя пула>",
      "crypto": "nocrypt",
      "priority": 100,
      "storage_duration": "1 Years",
      "archiving": false
    }
  }
}'
```

После отправки данного запроса должен быть получен ответ с информацией о том, что задача добавлена в очередь:

```
{
  "is_error": false,
  "message": "Resource with name 'Task Queue' was created successfully! TASK WAS ADDED TO QUEUE: <номер в очереди>",
  "data": {},
  "errors": {}
}
```

Глава 8. Описание методов

RuBackup REST API содержит методы для взаимодействия с сервером резервного копирования. Пользователь может обращаться к каждому методу либо через браузер, используя визуальное представление спецификации, либо через консольную утилиту `curl`.

В каждом из методов каждого эндпоинта должен передаваться CSRF токен.

В части запросов `GET` в query-параметрах могут передаваться:

- номер страницы списка (`page`)
- лимит отображения списка (`limit=(5|10|25|50|100|200)`)
- параметр, по которому будет отсортирован список (`sort_by`)
- направление сортировки (`sorting_order=(asc|desc)`)
- фильтр (`filter_data`). Формат, в котором можно указать фильтр (пример, зависит от эндпоинта):
 - `poolname="test"`
 - `poolname~"te"`
 - `lastonline<"2024-08-19T19:20:28.614838+03:00"`
 - `lastonline>"2024-08-19T19:20:28.614838+03:00"`

8.1. Версия 2 (v2)

Пути методов начинаются с `/api/v2`:

- `/auth/login` → `/api/v2/auth/login`
- `/metrics/tasks/clouds/s3` → `/api/v2/metrics/tasks/clouds/s3`

8.1.1. Авторизация и разрешения

Получение токенов и прохождение авторизации (POST `/auth/login`)

Получение токенов и прохождение авторизации. Access токен действителен 15 минут, refresh токен 24 часа, CSRF токен — до перезагрузки сервиса rest-api.

```
{
  "rubakup_user": "rubakup", ①
  "password": "12345", ①
  "auth_type": "database", ①
  "domain": "rubakup" ①
}
```

```
}
```

1 Обязательное поле

200 OK

Успешная авторизация, получены токены.

401 UNAUTHORIZED

Verification failed.

Обновление токена (POST /auth/refresh)

Обновление access токена после его истечения (раз в 15 минут).

200 OK

Успешная деактивация токенов.

401 UNAUTHORIZED

Verification failed.

Деактивация токенов (DELETE /auth/logout)

Деактивация имеющихся токенов (после выполнения запроса access и refresh токены более не действуют).

200 OK

Успешная деактивация токенов.

401 UNAUTHORIZED

Verification failed.

Получение списка прав пользователя (GET /auth/permissions)

200 OK

Успешно получен список прав текущего пользователя.

8.1.2. Пользователи

Получение списка пользователей (GET /users)

200 OK

Получен список пользователей.

Получение информации о пользователе (GET /users/{user_id})

Тела запроса нет. В query-параметрах передается `id` пользователя, информацию о котором необходимо получить.

200 OK

Информация о пользователе получена.

404 NOT_FOUND

Пользователь не найден.

Получение списка прав пользователя (GET /users/permissions)**200 OK**

Информация о правах пользователя получена.

Добавление пользователя (POST /users)

Также в query-параметрах передаются:

- роль пользователя (администратор, супервайзер, сопровождающий, аудитор),
- имя группы клиентов (для администраторов),
- хостнейм медиасервера (для сопровождающих).

```
{
  "group_id": "Nobody", ①
  "fullname": "",
  "address": "",
  "tel": "",
  "office": "",
  "email": "", ①
  "username": "", ①
  "user_password": "" ①
}
```

① Обязательное поле

201 CREATED

Пользователь создан.

Изменение пароля пользователя (PATCH /users/change_password)

```
{
  "new_password": "" ①
}
```

① Обязательное поле

204 NO_CONTENT

Пароль пользователя изменен.

Редактирование пользователей (PATCH /users/{user_id})

Также в query-параметрах передается `id` пользователя, информацию о которой необходимо изменить.

```
{
  "group_id": "Nobody",
  "fullname": "",
  "address": "",
  "tel": "",
  "office": "",
  "email": "",
  "new_password": ""
}
```

200 OK

Параметры пользователя изменены.

400 BAD_REQUEST

Некорректное тело запроса.

404 NOT_FOUND

Пользователь не найден.

Удаление списка пользователей (DELETE /users)

```
{
  "ids": [ ①
    1
  ]
}
```

① Обязательное поле

204 NO_CONTENT

Пользователи удалены.

404 NOT_FOUND

Пользователи не найдены.

Администраторы

Получение списка администраторов (GET `/users/administrators`)

200 OK

Получен список администраторов.

Получение информации об одном администраторе (GET `/users/administrators/{administrator_id}`)

Тела запроса нет. В query-параметрах передается `id` администратора, информацию о котором необходимо получить.

200 OK

Получена информация об администраторе.

Назначение пользователю роли администратора (POST `/users/administrators/{administrator_id}`)

```
{
  "client_group": "Some group", ①
  "administrator": "username" ①
}
```

① Обязательное поле

201 CREATED

Роль администратора добавлена пользователю.

Удаление роли администратора у списка пользователей (DELETE `/users/administrators`)

```
{
  "ids": [ ①
    1
  ]
}
```

① Обязательное поле

204 NO_CONTENT

Роль администратора удалена.

404 NOT_FOUND

Роль администратора не найдена.

Аудиторы

Получение списка аудиторов (GET /users/auditors)**200 OK**

Получен список аудиторов.

Получение информации об аудиторе (GET /users/auditors/{auditor_id})

Тела запроса нет. В query-параметрах передается `id` аудитора, информацию о котором необходимо получить.

200 OK

Получена информация об аудиторе.

Назначение пользователю роли аудитора (POST /users/auditors)

```
{
  "auditor": "username" ❶
}
```

❶ Обязательное поле

201 CREATED

Роль аудитора добавлена пользователю.

Удаление роли аудитора у списка пользователей (DELETE /users/auditors)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

204 NO_CONTENT

Роль аудитора удалена.

404 NOT_FOUND

Роль аудитора не найдена.

Сопровождающие**Получение списка сопровождающих (GET /users/maintainers)****200 OK**

Получен список сопровождающих.

Получение информации о сопровождающем (GET /users/maintainers/{maintainer_id})

Тела запроса нет. В query-параметрах передается `id` сопровождающего, информацию о котором необходимо получить.

200 OK

Получена информация о сопровождающем.

Назначение пользователю роли сопровождающего (POST /users/maintainers)

```
{
  "media_server": "hostname", ①
  "maintainer": "username" ①
}
```

① Обязательное поле

201 CREATED

Роль администратора добавлена пользователю.

Удаление роли сопровождающего у списка пользователей (DELETE /users/maintainers)

```
{
  "ids": [ ①
    1
  ]
}
```

① Обязательное поле

204 NO_CONTENT

Роль сопровождающего удалена.

404 NOT_FOUND

Роль сопровождающего не найдена.

Супервайзеры**Получение списка супервайзеров (GET /users/supervisors)****200 OK**

Получен список супервайзеров.

Получение информации о супервайзере (GET /users/supervisors/{supervisor_id})

Тела запроса нет. В query-параметрах передается `id` супервайзера, которого необходимо удалить.

200 OK

Роль супервайзера удалена.

404 NOT_FOUND

Роль супервайзера не найдена.

Назначение пользователю роли супервайзера (POST /users/supervisors)

```
{
  "supervisor": "Nobody" ❶
}
```

❶ Обязательное поле

201 CREATED

Роль супервайзера добавлена пользователю.

Удаление списка пользователей роли супервайзера (DELETE /users/supervisors)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

204 NO_CONTENT

Роль супервайзера удалена.

404 NOT_FOUND

Роль супервайзера не найдена.

Группы для уведомлений**Получение списка групп для уведомлений (GET /users/groups/)****200 OK**

Список групп пользователей получен.

Получение информации о группе уведомлений (GET /users/groups/{group_id})

Тела запроса нет. В query-параметрах передается `id` группы пользователей, информацию о которой необходимо получить.

200 OK

Получена информация о группе пользователей.

404 NOT_FOUND

Группа пользователей не найдена.

Добавление группы уведомлений (POST /users/groups/)

```
{
  "groupname": "my_group_name", ①
  "description": ""
}
```

① Обязательное поле

201 CREATED

Создана группа пользователей.

Изменение группы для уведомлений (PATCH /users/groups/{group_id})

Также в query-параметрах передается `id` группы пользователей, информацию о которой необходимо изменить.

```
{
  "groupname": "my_group_name",
  "description": ""
}
```

200 OK

Параметры группы пользователей изменены.

400 BAD_REQUEST

Некорректное тело запроса.

404 NOT_FOUND

Группа пользователей не найдена.

Удаление списка групп для уведомлений (DELETE /users/groups/)

```
{
  "ids": [ ①
    1
  ]
}
```

1 Обязательное поле

204 NO_CONTENT

Список групп пользователей удален.

404 NOT_FOUND

Группа пользователей не найдена.

Получение списка уведомлений (GET /users/groups/notifications)

200 OK

Получен список уведомлений.

Получение информации об уведомлении (GET /users/groups/notifications/{notification_id})

Тела запроса нет. В query-параметрах передается `id` уведомления, информацию о котором необходимо получить.

200 OK

Получена информация об уведомлении.

8.1.3. Сбор сведений

Сведения о сервере

Получение информации о сервере (GET /servers/hello)

200 OK

Получена информация о версии RuBackup сервера.

Получение информации о состоянии приложения (GET /servers/states)

Возвращает список неавторизованных клиентов и медиасерверов, просроченных правил стратегий и глобального расписания, клиентских запросов на добавление и удаление правил.

200 OK

Получена информация о текущем состоянии приложения.

Получение информации о серверах и лицензии (GET /servers/licenses)

200 OK

Получена информация о серверах, их HWID и лицензии.

Получение информации о сервере и лицензии (GET /servers/licenses/{license_id})

Тела запроса нет. В query-параметрах передается `id` медиасервера, информацию о котором необходимо получить.

200 OK

Получена информация о сервере и лицензии.

404 NOT_FOUND

Информация о медиасerverе и лицензии не найдена.

Типы сущностей

Получение списка доступных типов резервного копирования (GET /types/backups)

200 OK

Получен список типов резервного копирования.

Получение списка доступных типов сжатия (GET /types/compressions)

200 OK

Получен список типов сжатия.

Получение списка доступных криптоалгоритмов (GET /types/cryptoalgorithms)

200 OK

Получен список доступных криптоалгоритмов.

Получение списка доступных алгоритмов хеш-функций (GET /types/deduplication_hash_algorithms)

200 OK

Получен список алгоритмов хеш-функций.

Получение списка доступных типов пулов (GET /types/destination_storages)

200 OK

Получен список доступных типов пулов.

Получение списка типов лицензий (GET /types/licenses)

200 OK

Получен список типов лицензий.

Получение информации о типе лицензии (GET /types/licenses/{id})

Тела запроса нет. В query-параметрах передается `id` типа лицензии, информацию о которой необходимо получить.

200 OK

Получена информация о типе лицензии.

404 NOT_FOUND

Тип лицензии не найден.

Получение списка доступных уровней логирования (GET /types/log_levels)

200 OK

Получен список всех уровней логирования.

Получение списка доступных операционных систем (GET /types/os)

200 OK

Получен список доступных типов операционных систем.

Получение списка доступных типов ресурсов (GET /types/resources)

200 OK

Получен список типов ресурсов.

Получение информации о типе ресурса (GET /types/resources/{id})

Тела запроса нет. В query-параметрах передается `id` типа ресурса, информацию о котором необходимо получить.

200 OK

Получена информация о типе ресурса.

404 NOT_FOUND

Тип ресурса не найден.

Получение списка доступных типов картриджей (GET /types/tapes)

200 OK

Получен список доступных типов картриджей.

Статусы

Получение списка статусов уведомлений (GET /statuses/notifications)

200 OK

Получен список статусов уведомлений.

Получение списка статусов правил (GET /statuses/rules)

200 OK

Получен список статусов правил.

Получение списка статусов задач (GET /statuses/tasks)

200 OK

Получен список статусов задач.

Получение списка статусов задач на облачные хранилища (GET /statuses/s3_clouds)

200 OK

Получен список статусов очереди задач облаков.

Получение списка статусов верификации (GET /statuses/verifications)

200 OK

Получен список статусов верификации.

Метрики (статистика)

Статистика уведомлений (GET /metrics/notifications)

200 OK

Получена статистика уведомлений.

Статистика объектов репозитория (GET /metrics/repositories)

200 OK

Получена статистика репозитория.

Статистика очереди задач облаков (GET /metrics/tasks/clouds/s3)

200 OK

Получена статистика очереди задач облаков.

Статистика очереди задач (GET /metrics/tasks)

200 OK

Получена статистика очереди задач.

Статистика очереди задач ленточных библиотек (GET /metrics/tasks/tape_libraries)

200 OK

Получена статистика очереди задач ленточных библиотек.

8.1.4. Глобальная конфигурация и сервисный режим

Получение параметров глобальной конфигурации (GET /global_configuration)

200 OK

Параметры глобальной конфигурации получены.

Редактирование параметров глобальной конфигурации (PATCH /global_configuration)

```
{  
  "file_transfer_block_size": 16384,  
}
```

```

"data_spred_into_pool": "sequentially",
"emergency_storage_local_catalog":
"/tmp/rubackup_emergency_storage_local_catalog",
"done_tasks_remove_period": 1440,
"error_tasks_remove_period": 10080,
"broken_tasks_remove_period": 10080,
"killed_tasks_remove_period": 10080,
"obsoleted_backup_notification_period": 1440,
"rule_validity_end_notification_period": 1440,
"strategy_validity_end_notification_period": 1440,
"rule_storage_capacity_reserve": 0,
"strategy_storage_capacity_reserve": 0,
"rule_storage_capacity_notification_period": 1440,
"strategy_storage_capacity_notification_period": 1440,
"service_window_start": "00:00:00",
"service_window_end": "23:59:59",
"pool_storage_capacity_reserve": 2000000000,
"tape_library_mount_point": "/opt/rubackup/mnt",
"suspended_task_restart_period": 1,
"unload_tape_cartridges_whes_media_server_starts": "yes",
"ltfs_umount_timeout": 50,
"max_system_monitoring_records": 3600,
"monitoring_period": 1,
"digital_signature_public_key_obsolescence": 1440,
"rb_key_hash": "",
"verify_archive_after_creation": "no",
"kill_task_of_offline_client": "yes",
"create_new_task_if_client_offline": "create new task",
"auto_delete_archive_from_broken_chain": "no",
"consolidating_tape_library_tasks": "yes",
"clean_tape_drives_period": 30,
"auto_clean_tape_drives": "no",
"dedup_clean_unused_blocks": "yes",
"dedup_clean_period": 30,
"dedup_provide_common_hash_table_for_client": "yes",
"dedup_provide_common_hash_table": "yes",
"dedup_verify_only_meta_data": "yes",
"mandatory_storage_time": 0,
"clients_capacity_limits": false,
"global_schedule_capacity_limits": false,
"backup_strategies_capacity_limits": false,
"immutable_archives": true,
"delete_archives_with_zeroing": true,
"dont_delete_last_gs_rule_archive": true,
"dont_delete_last_strategy_rule_archive": true,

```

```

"wrong_auth_attempts_count": 5,
"wrong_auth_block_period": 30,
"repository_remove_chain": "yes",
"filesystems_clean_period": 30,
"show_previous_login_attempts_after_logging_in": "yes",
"search_backup_in_cluster_group": true,
"bandwidth_limit_advantage": "client",
"bandwidth_client_limit": "minimum",
"bandwidth_rule_limit": "minimum",
"remote_replica_rule_validity_end_notification_period": 1440,
"cartridge_unload_timeout": 60,
"tape_library_warnings_limit": 10
}

```

200 OK

Параметры глобальной конфигурации изменены.

400 BAD_REQUEST

Некорректное тело запроса.

Включение и выключение сервисного режима (PATCH /global_configuration/service_mode)

```

{
  "service_mode": false ❶
}

```

❶ Обязательное поле

200 OK

Статус сервисного режима изменен.

400 BAD_REQUEST

Некорректное тело запроса.

8.1.5. Клиенты**Получение списка клиентов (GET /clients)****200 OK**

Получен список авторизованных клиентов.

Получение клиента (GET /clients/{client_id})

Тела запроса нет. В query-параметрах передается `id` клиента, информацию о котором необходимо получить.

200 OK

Получена информация о клиенте.

Получение дерева клиентов (POST /clients/tree)

```
{
  "data": {
    "clients": [
      {
        "client_group_id": 1, ❶
        "clients_limit": 20,
        "clients_page": 1 ❶
      }
    ],
    "client_groups_page": 1,
    "client_groups_limit": 20,
    "filter_clients_hostname": "rubackup-client"
  }
}
```

❶ Обязательное поле

200 OK

Получено дерево клиентов.

404 NOT_FOUND

Авторизованный клиент не найден.

Редактирование существующего клиента (PATCH /clients/{client_id})

Также в query-параметрах передается `id` клиента, информацию о котором необходимо изменить.

```
{
  "group_id": "No group",
  "lsf_flag": false,
  "ls_pool": "Default",
  "lrl_flag": false,
  "storage_capacity": 10,
  "description": "Client description",
  "client_side_backup": false,
  "client_side_restore": false
}
```

200 OK

Параметры клиента изменены.

400 BAD_REQUEST

Некорректное тело запроса.

404 NOT_FOUND

Авторизованный клиент не найден.

Удаление нескольких клиентов (DELETE /clients)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

200 OK

Список авторизованных клиентов удален.

404 NOT_FOUND

Авторизованный клиент не найден.

Группы клиентов**Получение списка групп клиентов (GET /clients/groups)****200 OK**

Получена информация о группах клиентов.

Получение информации о группе клиентов (GET /clients/groups/{group_id})

Тела запроса нет. В query-параметрах передается `id` группы клиентов, информацию о которой необходимо получить.

200 OK

Получена информация о группе клиентов.

Добавление групп клиентов (POST /clients/groups)

```
{
  "name": "group_name",
  "shared": false,
  "clustered": false,
```

```

"description": "some description",
"launch_attempts_limit": 1
}

```

201 CREATED

Группа клиентов создана.

Изменение группы клиентов (PATCH /clients/groups/{group_id})

Также в query-параметрах передается `id` группы клиентов, информацию о которой необходимо изменить.

```

{
  "name": "group_name",
  "shared": false,
  "clustered": false,
  "description": "some description",
  "launch_attempts_limit": 1
}

```

200 OK

Параметры группы клиентов изменены.

400 BAD_REQUEST

Некорректное тело запроса.

404 NOT_FOUND

Группа клиентов не найдена.

Удаление списка групп клиентов (DELETE /clients/groups)

```

{
  "ids": [ ①
    1
  ]
}

```

① Обязательное поле

200 OK

Список групп клиентов удален.

404 NOT_FOUND

Группа клиентов не найдена.

Клиентские ресурсы хранения

Получение списка доступных директорий клиента (POST `/clients/filesystem`)

```
{
  "path": "/", ❶
  "client_hwid": "525a99154f3505a2" ❶
}
```

❶ Обязательное поле

200 OK

Получен список доступных директорий клиента.

Получение списка доступных блочных устройств клиента (POST `/clients/filesystem/block_devices`)

```
{
  "path": "/dev/", ❶
  "client_hwid": "525a99154f3505a2" ❶
}
```

❶ Обязательное поле

200 OK

Получен список доступных блочных устройств клиента.

Доступные типы ресурсов

Получение списка доступных ресурсов модуля (POST `/clients/resources`)

Например, для PostgreSQL Universal это версия PostgreSQL (например, PostgreSQL 16.3).

```
{
  "client_hwid": "525a99154f3505a2", ❶
  "resource_type": "LVM logical volume", ❶
  "secret_method": 0
}
```

❶ Обязательное поле

200 OK

Получен список доступных ресурсов модуля.

Получение доступных типов ресурсов для создания копии (GET /clients/{client_id}/resource_types)

Тело запроса отсутствует. В query-параметрах передается `id` клиента, информацию о типах ресурсов для копирования необходимо получить.

200 OK

Получен список доступных ресурсов модуля.

Ограничения пропускной способности клиента**Получение списка ограничений пропускной способности клиента (GET /clients/{client_id}/bandwidth)****200 OK**

Получен список ограничений пропускной способности.

Получение информации об ограничении пропускной способности клиента (GET /clients/bandwidth/{bandwidth_id})

Тела запроса нет. В query-параметрах передается `id` ограничения, информацию о котором необходимо получить.

200 OK

Получена информация об ограничении пропускной способности.

Создание ограничения пропускной способности клиента (POST /clients/{client_id}/bandwidth)

В query-параметрах передается `id` клиента, для которого необходимо создать ограничение пропускной способности.

```
{
  "backup_bandwidth": 0,
  "restore_bandwidth": 0,
  "window_start": "13:35:28",
  "window_end": "13:35:28"
}
```

201 CREATED

Ограничение пропускной способности создано.

Изменение ограничения пропускной способности клиента (PATCH /clients/bandwidth/{bandwidth_id})

Также в query-параметрах передается `id` ограничения, информацию о котором необходимо изменить.

```
{
  "backup_bandwidth": 0,
  "restore_bandwidth": 0,
```

```
"window_start": "13:35:28",  
"window_end": "13:35:28"  
}
```

200 OK

Параметры ограничения пропускной способности изменены.

400 BAD_REQUEST

Некорректное тело запроса.

404 NOT_FOUND

Ограничение пропускной способности не найдено.

Удаление списка ограничений пропускной способности клиента (DELETE /clients/bandwidth)

```
{  
  "ids": [ ①  
    1  
  ]  
}
```

① Обязательное поле

200 OK

Список ограничений пропускной способности удален.

404 NOT_FOUND

Ограничение пропускной способности не найдено.

Неавторизованные клиенты

Получение списка неавторизованных клиентов (GET /clients/unauthorised)

200 OK

Получен список неавторизованных клиентов.

Получение информации о неавторизованном клиенте (GET /clients/unauthorised/{client_id})

Тело запроса отсутствует. В query-параметрах передается `id` неавторизованного клиента, информацию о котором необходимо получить.

200 OK

Получена информация о неавторизованном клиенте.

Авторизация клиентов (POST /clients/unauthorised/authorise)

```
{
  "clients_ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

204 NO_CONTENT

Клиенты авторизованы.

Удаление списка неавторизованных клиентов (DELETE /clients/unauthorised)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

204 NO_CONTENT

Список неавторизованных клиентов удален.

404 NOT_FOUND

Неавторизованные клиенты не найдены.

8.1.6. Хранилища**Локальные файловые хранилища****Получение списка локальных файловых хранилищ (GET /storages/local_catalogs)****200 OK**

Получен список локальных файловых хранилищ.

Добавление локального файлового хранилища (POST /storages/local_catalogs)

```
{
  "path": "",
  "pool": "Default",
  "description": ""
}
```

201 CREATED

Локальное файловое хранилище добавлено.

Редактирование существующего локального файлового хранилища (PATCH /storages/local_catalogs/{catalog_id})

Также в query-параметрах передается `id` локального файлового хранилища, который необходимо изменить.

```
{
  "description": ""
}
```

200 OK

Параметры локального файлового хранилища изменены.

400 BAD_REQUEST

Некорректное тело запроса.

404 NOT_FOUND

Локальное файловое хранилище не найдено.

Удаление нескольких локальных файловых хранилищ (DELETE /storages/local_catalogs)

```
{
  "ids": [ ①
    1
  ]
}
```

① Обязательное поле

204 NO_CONTENT

Список локальных файловых хранилищ удален.

404 NOT_FOUND

Локальный файловый хранилище не найдено.

Получение информации о локальном файловом хранилище (GET /storages/local_catalogs/{catalog_id})

Также в query-параметрах передается `id` локального файлового хранилища, информацию о котором необходимо получить.

200 OK

Информация о локальном файловом хранилище получена.

404 NOT_FOUND

Локальное файловое хранилище не найдено.

Блочные устройства

Получение списка блочных устройств (GET /storages/block_devices)

200 OK

Получен список блочных устройств.

Получение информации о блочном устройстве (GET /storages/block_devices/{block_device_id})

Также в query-параметрах передается `id` блочного устройства, информацию о котором необходимо получить.

200 OK

Информация о блочном устройстве получена.

404 NOT_FOUND

Блочное устройство не найдено.

Добавление блочного устройства (POST /storages/block_devices)

```
{
  "device": "", ❶
  "pool": "Default", ❶
  "overwrite_file_system": false, ❶
  "description": ""
}
```

❶ Обязательное поле

201 CREATED

Блочное устройство добавлено.

Редактирование блочного устройства (PATCH /storages/block_devices/{block_device_id})

Также в query-параметрах передается `id` блочного устройства, которое необходимо изменить.

```
{
  "description": "",
  "parallel_workers": 2
}
```

200 OK

Параметры блочного устройства изменены.

400 BAD_REQUEST

Некорректное тело запроса.

404 NOT_FOUND

Блочное устройство не найдено.

Удаление списка блочных устройств (DELETE /storages/block_devices)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

204 NO_CONTENT

Список блочных устройств удален.

404 NOT_FOUND

Блочное устройство не найдено.

Клиентские хранилища**Получение списка клиентских хранилищ (GET /storages/client_defined)****200 OK**

Получен список клиентских хранилищ.

Получение информации о клиентском хранилище (GET /storages/client_defined/{client_defined_id})

Тела запроса нет. В query-параметрах передается `id` клиентского хранилища, информацию о котором необходимо получить.

200 OK

Получена информация об клиентском хранилище.

404 NOT_FOUND

Клиентское хранилище не найдено.

Добавление клиентского хранилища (POST /storages/client_defined)

```
{
```

```

"pool": "Default", ❶
"metadata_fs_pool": "Default2", ❶
"name": "", ❶
"description": ""
}

```

❶ Обязательное поле

201 CREATED

Клиентское хранилище создано.

Изменение клиентского хранилища (PATCH /storages/client_defined/{client_defined_id})

Также в query-параметрах передается `id` клиентского хранилища, информацию о котором необходимо изменить.

```

{
  "name": "",
  "metadata_fs_pool": "Default2",
  "description": ""
}

```

200 OK

Параметры клиентского хранилища изменены.

400 BAD_REQUEST

Некорректное тело запроса.

404 NOT_FOUND

Клиентское хранилище не найдено.

Удаление списка клиентских хранилищ (DELETE /storages/client_defined)

```

{
  "ids": [ ❶
    1
  ]
}

```

❶ Обязательное поле

204 NO_CONTENT

Список клиентских хранилищ удален.

404 NOT_FOUND

Клиентское хранилище не найдено.

Облачные хранилища

Получение списка доступных облаков (GET /storages/clouds/s3)

200 OK

Получен список облаков.

Получение информации о существующем облаке (GET /storages/clouds/s3/{s3_id})

Тела запроса нет. В query-параметрах передается `id` облака, информацию о котором необходимо получить.

200 OK

Получена информация об облаке.

404 NOT_FOUND

Облако не найдено.

Получение списка доступных бакетов существующего облака (GET /storages/clouds/s3/{s3_id}/buckets)

В query-параметрах передается `id` облака, информацию о доступных бакетах которого необходимо получить.

200 OK

Получена информация о доступных бакетах существующего облака.

Получение списка доступных бакетов не добавленного в СРК облака (POST /storages/clouds/s3/buckets)

```
{
  "access_key_id": "jkepw6ubWUJpS04vtmgn", ①
  "secret_key_access": "R0hj0Gn7MjJ3iTvBBxw9dI1ez9jPd1wDkMuIQ4FU", ①
  "region": "ru-1", ①
  "endpoint_override": "https://192.168.222.197:9000", ①
  "proxy_host": "",
  "proxy_port": "",
  "proxy_username": "",
  "proxy_password": ""
}
```

① Обязательное поле

200 OK

Получен список существующих в облаке бакетов.

Добавление нового облака (POST /storages/clouds/s3)

```
{
  "name": "my_s3_cloud", ①
  "pool": "cloud_pool", ①
  "access_key_id": "", ①
  "secret_key_access": "", ①
  "region": "ru-1",
  "endpoint_override": "https://192.168.222.197:9000", ①
  "proxy_host": "",
  "proxy_port": "",
  "proxy_username": "",
  "proxy_password": "",
  "bucket_name": "rubackup-server-5953bb5c71fc3bf8",
  "allow_auto_bucket_creation": true,
  "transfer_workers_number": 1
}
```

① Обязательное поле

201 CREATED

Облако создано.

Редактирование параметров облака (PATCH /storages/clouds/s3/{s3_id})

В query-параметрах передается `id` облака, информацию о котором необходимо изменить.

```
{
  "allow_auto_bucket_creation": false,
  "bucket_name": ""
}
```

200 OK

Параметры облака изменены.

404 NOT_FOUND

Облако не найдено.

Удаление списка облаков (DELETE /storages/clouds/s3)

```
{
  "ids": [ ①
    1
  ]
}
```

```
]
}
```

① Обязательное поле

204 NO_CONTENT

Удален список облаков.

Картриджи

Получение списка доступных картриджей (GET /storages/tape_cartridges)

200 OK

Получен список доступных картриджей.

Получение информации о ленточном картридже (GET /storages/tape_cartridges/{cartridge_id})

Тела запроса нет. В query-параметрах передается `id` картриджа, информацию о котором необходимо получить.

200 OK

Получена информация о ленточном картридже.

Добавление ленточного картриджа (POST /storages/tape_cartridges)

```
{
  "type": "LTO-9", ①
  "pool": "tape_pool", ①
  "volume_tag": "RB10001L9", ①
  "description": ""
}
```

① Обязательное поле

201 CREATED

Ленточный картридж добавлен.

Редактирование картриджа (PATCH /storages/tape_cartridges/{cartridge_id})

Также в query-параметрах передается `id` ленточного картриджа, информацию о которой необходимо изменить.

```
{
  "volume_tag": "",
  "description": ""
}
```

200 OK

Параметры картриджа изменены.

400 BAD_REQUEST

Некорректное тело запроса.

404 NOT_FOUND

Картридж не найден.

Удаление списка картриджей (DELETE /storages/tape_cartridges)

```
{
  "ids": [
    1
  ]
}
```

204 NO_CONTENT

Список картриджей удален.

404 NOT_FOUND

Картридж не найден.

Заморозка картриджа (PATCH /storages/tape_cartridges/freeze)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

204 NO_CONTENT

Статус картриджа изменен.

207 MULTI_STATUS

Части картриджей изменен статус.

422 UNPROCESSABLE_ENTITY

Нельзя изменить статус картриджа.

Разморозка картриджа (PATCH /storages/tape_cartridges/unfreeze)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

204 NO_CONTENT

Статус картриджа изменен.

207 MULTI_STATUS

Части картриджей изменен статус.

422 UNPROCESSABLE_ENTITY

Нельзя изменить статус картриджа.

Приостановка картриджа (PATCH /storages/tape_cartridges/suspend)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

204 NO_CONTENT

Статус картриджа изменен.

207 MULTI_STATUS

Части картриджей изменен статус.

422 UNPROCESSABLE_ENTITY

Нельзя изменить статус картриджа.

Возобновление картриджа (PATCH /storages/tape_cartridges/unsuspend)

```
{
  "ids": [ ❶
    1
  ]
}
```

1 Обязательное поле**204 NO_CONTENT**

Статус картриджа изменен.

207 MULTI_STATUS

Части картриджей изменен статус.

422 UNPROCESSABLE_ENTITY

Нельзя изменить статус картриджа.

Ленточные библиотеки

Получение списка ленточных библиотек (GET /tape_libraries)

200 OK

Получен список ленточных библиотек.

Получение дерева ленточных библиотек (POST /tape_libraries/tree)

В списке отображаются:

1. Ленточные библиотеки, их роботы, их приводы. Список слотов должен быть пустым всегда.
2. Ленточные пулы, ассоциированные и неассоциированные с библиотекой.

```
{
  "pools": {
    "limit": 50,
    "page": 1
  }
}
```

200 OK

Получено дерево информация о роботе и слотах.

Получение информации о ленточной библиотеке (GET /tape_libraries/{library_id})

200 OK

Информация о ленточной библиотеке получена.

404 NOT_FOUND

Ленточная библиотека не найдена.

Добавление ленточной библиотеки (POST /tape_libraries)

```
{
  "pool": "tape_pool", ❶
  "media_server": "rubackup-media", ❶
  "description": "",
  "medium_changer_candidate_id": 1, ❶
  "tape_drives_candidates_ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

201 CREATED

Ленточная библиотека добавлена.

Удаление списка ленточных библиотек (DELETE /tape_libraries)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

204 NO_CONTENT

Список ленточных библиотек удален.

404 NOT_FOUND

Ленточная библиотека не найдена.

Получение списка доступных для добавления роботов и приводов (GET /tape_libraries/candidates)

Тела запроса нет. В query-параметрах передается имя медиасервера, которому презентована библиотека, а также способ добавления.

200 OK

Получен список доступных для добавления роботов и приводов.

Получение информации о работе и слотах (POST /tape_libraries/candidates/info)

```
{
  "media_server": "rubackup-media-server", ❶
}
```

```
"medium_changer_sg_path": "/dev/sg4"
}
```

1 Обязательное поле

200 OK

Получена информация о роботе и слотах.

Синхронизация данных ленточной библиотеки (POST /tape_libraries/{library_id}/sync)

Тела запроса нет. В query-параметрах передается `id` библиотеки, информацию о которой необходимо синхронизировать.

200 OK

Синхронизация библиотеки прошла успешно.

422 UNPROCESSABLE_ENTITY

Синхронизация завершена с ошибкой, библиотечка содержит незарегистрированный картридж.

Роботы

Получение списка роботов (GET /tape_libraries/{library_id}/medium_changers)

200 OK

Получен список роботов.

Получение информации о роботе (GET /tape_libraries/medium_changers/{changer_id})

Тела запроса нет. В query-параметрах передается `id` робота, информацию о которой необходимо получить.

200 OK

Получена информация о роботе.

404 NOT_FOUND

Робот не найден.

Изменение робота (PATCH /tape_libraries/medium_changers/{changer_id})

Также в query-параметрах передается `id` робота, информацию о котором необходимо изменить.

```
{
  "scsi_path": "[8:0:0:0]",
  "device": "/dev/sg2"
```

```
}
```

200 OK

Параметры робота изменены.

400 BAD_REQUEST

Некорректное тело запроса.

404 NOT_FOUND

Робот не найдена.

Слоты**Получение списка слотов (GET /tape_libraries/{library_id}/slots)**

Тела запроса нет. В query-параметрах передается `id` библиотеки, информацию о слотах которой необходимо получить.

200 OK

Получен список слотов ленточной библиотеки.

Получение информации о слоте ленточной библиотеки (GET /tape_libraries/slots/{slot_id})

Тела запроса нет. В query-параметрах передается `id` слота, информацию о котором необходимо получить.

200 OK

Получена информация о слоте ленточной библиотеки.

Проверка файловой системы на картридже (PATCH /tape_libraries/{library_id}/slots/check_ltf)

Также в query-параметрах передается `id` библиотеки, картридж в слоте которой необходимо проверить.

```
{  
  "volume_tag": "RB0001L9" ①  
}
```

① Обязательное поле

200 OK

Создана задача на проверку наличия ltf на картридже.

400 BAD_REQUEST

Некорректное тело запроса.

Стирание картриджа (PATCH /tape_libraries/{library_id}/slots/erase_cartridge)

Также в query-параметрах передается `id` библиотеки, картридж в слоте которой необходимо стереть.

```
{
  "volume_tag": "RB0001L9", ❶
  "is_long_erase": false
}
```

❶ Обязательное поле

200 OK

Создана задача на стирание картриджа.

400 BAD_REQUEST

Некорректное тело запроса.

Экспорт картриджа (PATCH /tape_libraries/{library_id}/slots/export_cartridge)

Также в query-параметрах передается `id` библиотеки, картридж в слоте которой необходимо экспортировать.

```
{
  "volume_tag": "RB0001L9" ❶
}
```

❶ Обязательное поле

200 OK

Создана задача на экспорт картриджа.

400 BAD_REQUEST

Некорректное тело запроса.

Импорт картриджа (PATCH /tape_libraries/{library_id}/slots/{slot_id}/import_cartridge)

Также в query-параметрах передается `id` библиотеки, картридж в слоте которой необходимо импортировать и `id` слота, в который произойдет импорт.

```
{
  "pool_name": "Tape libraries" ❶
}
```

❶ Обязательное поле

200 OK

Импорт картриджа прошел успешно.

400 BAD_REQUEST

Некорректное тело запроса.

Форматирование картриджа (PATCH /tape_libraries/{library_id}/slots/format_cartridge)

Также в query-параметрах передается `id` библиотеки, картридж в слоте которой необходимо форматировать.

```
{
  "volume_tag": "RB0001L9" ❶
}
```

❶ Обязательное поле

200 OK

Создана задача на форматирование картриджа.

400 BAD_REQUEST

Некорректное тело запроса.

Перемещение картриджа в другой слот (PATCH /tape_libraries/{library_id}/slots/{slot_id}/move_cartridge)

Также в query-параметрах передается `id` библиотеки, картридж в слоте которой необходимо переместить и `id` слота, в который будет произведено перемещение.

```
{
  "volume_tag": "RB0001L9" ❶
}
```

❶ Обязательное поле

200 OK

Создана задача на перемещение картриджа.

400 BAD_REQUEST

Некорректное тело запроса.

Ленточные приводы**Получение списка ленточных приводов (GET /tape_libraries/{library_id}/tape_drives)**

Тела запроса нет. В query-параметрах передается `id` библиотеки, информацию о

приводах которой необходимо получить.

200 OK

Получен список приводов ленточной библиотеки.

Получение информации о ленточном приводе (GET /tape_libraries/tape_drives/{drive_id})

Тела запроса нет. В query-параметрах передается `id` привода, информацию о котором необходимо получить.

200 OK

Получена информация о приводе ленточной библиотеки.

Добавление привода (POST /tape_libraries/{library_id}/tape_drives)

Также в query-параметрах передается `id` библиотеки, в которую необходимо добавить привод.

```
{
  "ids": [ ①
    4,
    6,
    9
  ]
}
```

① Обязательное поле

201 CREATED

Новый привод добавлен к библиотеке.

Редактирование ленточного привода (PATCH /tape_libraries/tape_drives/{drive_id})

Также в query-параметрах передается `id` привода, информацию о котором необходимо изменить.

```
{
  "scsi_path": "[6:0:0:0]",
  "device": "/dev/sg1",
  "sys_transfer_element": 3,
  "nst_path": "/dev/nst0"
}
```

200 OK

Параметры привода изменены.

400 BAD_REQUEST

Некорректное тело запроса.

Удаление списка ленточных приводов (DELETE /tape_libraries/tape_drives)

```
{
  "ids": [ ①
    2,
    6,
    9
  ]
}
```

① Обязательное поле

204 NO_CONTENT

Привод ленточной библиотеки удален.

404 NOT_FOUND

Привод ленточной библиотеки не найден.

Получение списка доступных для добавления ленточных приводов (GET /tape_libraries/tape_drives/candidates)

Тела запроса нет. В query-параметрах передается способ добавления привода (ручной или автоматический), `hostname` медиасервера, на который презентована библиотека и `sg_device_path` робота библиотеки.

200 OK

Информация о доступных для добавления приводах получена.

Очистка ленточного привода (PATCH /tape_libraries/tape_drives/{drive_id}/clean)

Тела запроса нет. В query-параметрах передается `id` привода, который необходимо очистить.

200 OK

Задача на очистку привода создана.

8.1.7. Стратегии

Получение списка стратегий (GET /backup_strategies)**200 OK**

Получен список стратегий.

Получение информации о стратегии (GET /backup_strategies/{strategy_id})

Тела запроса нет. В query-параметрах передается id стратегии, информацию о правилах репликации которой необходимо получить.

200 OK

Получена информация о репликации стратегии.

Создание стратегии (POST /backup_strategies)

```
{
  "name": "my_backup_strategy", ①
  "status": "wait",
  "pool_id": "Default",
  "crypto": "nocrypt",
  "storage_capacity": 50,
  "description": "",
  "validity_start_period": "2025-01-13T16:54:45.541246",
  "validity_end_period": "2026-01-13T16:54:45.541261",
  "verify_flag": true,
  "verify_interval": "1 day",
  "auto_delete_obsoleted_copy_flag": false,
  "inform_when_obsoleted_copy": "Nobody",
  "client_delete_flag": true,
  "full_archive_enabled": false,
  "full_periodic_launch": "1 min",
  "full_storage_duration": "1 day",
  "full_min": 0,
  "full_hour": 0,
  "full_dom": 1,
  "full_mon": 1,
  "full_dow": 1,
  "full_move_copy_flag": false,
  "full_move_copy_while": "1 day",
  "full_move_copy_pool": "Default",
  "inc_archive_enabled": false,
  "inc_periodic_launch": "1 min",
  "inc_storage_duration": "1 day",
  "inc_min": 0,
  "inc_hour": 0,
  "inc_dom": 1,
  "inc_mon": 1,
  "inc_dow": 1,
  "inc_move_copy_flag": false,
  "inc_move_copy_pool": "Default",
```

```

"inc_move_copy_while": "1 day",
"diff_archive_enabled": false,
"diff_periodic_launch": "1 min",
"diff_storage_duration": "1 day",
"diff_min": 0,
"diff_hour": 0,
"diff_dom": 1,
"diff_mon": 1,
"diff_dow": 1,
"diff_move_copy_flag": false,
"diff_move_copy_pool": "Default",
"diff_move_copy_while": "1 day",
"notify_normal": "Nobody",
"notify_normal_cc": "email@http://domain.ru/[domain.ru]",
"notify_exception": "Nobody",
"notify_exception_cc": "email@http://domain.ru/[domain.ru]",
"notify_verify": "Nobody",
"notify_verify_cc": "email@http://domain.ru/[domain.ru]",
"notify_strategy_validity": "Nobody",
"notify_strategy_validity_cc": "email@http://domain.ru/[domain.ru]",
"notify_end_storage_capacity": "Nobody",
"notify_end_storage_capacity_cc": "email@http://domain.ru/[domain.ru]"
}

```

1 Обязательное поле

201 CREATED

Стратегия создана.

Изменение параметров стратегии (PATCH /backup_strategies/{strategy_id})

Также в query-параметрах передается `id` стратегии, информацию о которой необходимо изменить.

```

{
  "name": "my_backup_strategy",
  "status": "wait",
  "pool_id": "Default",
  "crypto": "nocrypt",
  "storage_capacity": 50,
  "description": "",
  "validity_start_period": "2025-01-13T16:54:45.541246",
  "validity_end_period": "2026-01-13T16:54:45.541261",
  "verify_flag": true,
  "verify_interval": "1 day",
}

```

```

"auto_delete_obsoleted_copy_flag": false,
"inform_when_obsoleted_copy": "Nobody",
"client_delete_flag": true,
"full_archive_enabled": false,
"full_periodic_launch": "1 min",
"full_storage_duration": "1 day",
"full_min": 0,
"full_hour": 0,
"full_dom": 1,
"full_mon": 1,
"full_dow": 1,
"full_move_copy_flag": false,
"full_move_copy_while": "1 day",
"full_move_copy_pool": "Default",
"inc_archive_enabled": false,
"inc_periodic_launch": "1 min",
"inc_storage_duration": "1 day",
"inc_min": 0,
"inc_hour": 0,
"inc_dom": 1,
"inc_mon": 1,
"inc_dow": 1,
"inc_move_copy_flag": false,
"inc_move_copy_pool": "Default",
"inc_move_copy_while": "1 day",
"diff_archive_enabled": false,
"diff_periodic_launch": "1 min",
"diff_storage_duration": "1 day",
"diff_min": 0,
"diff_hour": 0,
"diff_dom": 1,
"diff_mon": 1,
"diff_dow": 1,
"diff_move_copy_flag": false,
"diff_move_copy_pool": "Default",
"diff_move_copy_while": "1 day",
"notify_normal": "Nobody",
"notify_normal_cc": "email@http://domain.ru/[domain.ru]",
"notify_exception": "Nobody",
"notify_exception_cc": "email@http://domain.ru/[domain.ru]",
"notify_verify": "Nobody",
"notify_verify_cc": "email@http://domain.ru/[domain.ru]",
"notify_strategy_validity": "Nobody",
"notify_strategy_validity_cc": "email@http://domain.ru/[domain.ru]",
"notify_end_storage_capacity": "Nobody",

```

```
"notify_end_storage_capacity_cc": "email@http://domain.ru/[domain.ru]"
}
```

200 OK

Стратегия изменена.

400 BAD_REQUEST

Некорректное тело запроса.

404 NOT_FOUND

Стратегия не найдена.

Изменение статуса стратегии (PATCH /backup_strategies)

```
{
  "strategies": [
    {
      "strategy_id": 1, ①
      "status": "wait" ①
    }
  ]
}
```

① Обязательное поле

200 OK

Статус стратегии изменен.

400 BAD_REQUEST

Некорректное тело запроса.

404 NOT_FOUND

Стратегия не найдена.

Удаление списка стратегий (DELETE /backup_strategies)

```
{
  "ids": [ ①
    1
  ]
}
```

① Обязательное поле

200 OK

Стратегии удалены.

404 NOT_FOUND

Стратегия не найдена.

Получение списка правил стратегий (GET /backup_strategies/rules)**200 OK**

Получен список правил стратегии.

Получение информации о правиле стратегии (GET /backup_strategies/rules/{rule_id})

Тело запроса отсутствует. В query-параметрах передается `id` правила, информацию о котором необходимо получить.

200 OK

Правило стратегии удалено.

404 NOT_FOUND

Правило стратегии не найдено.

Добавление правила в стратегию (POST /backup_strategies/rules)

```
{
  "strategy_rule": {
    "client_id": "rubackup-client (525a99154f3505a2)", ①
    "strategy_id": "Default", ①
    "resource_type": "File system", ①
    "resource": "/home/", ①
    "normal_execution_script": "",
    "exception_execution_script": "",
    "priority": 100,
    "restore_script": ""
  },
  "module_extensions": {
    "parameters": {
      "file_list": true,
      "numeric_owner": false,
      "use_snapshot": false,
      "ignore_errors_snapshot": true,
      "snapshot_type": "",
      "snapshot_size": 10,
      "script_before_snapshot": "",
      "script_after_snapshot": "",
      "script_error_snapshot": ""
    }
  }
}
```

```

    }
  },
  "common_modules_extensions": {
    "parameters": {
      "worker_parallelism": 8,
      "enable_multithreading": false,
      "enable_flexible_dedup": false,
      "network_parallelism": 8,
      "memory_threshold": 0,
      "deny_memory_exceed": false,
      "restore_corrupted_blocks": "full_analysis",
      "deduplication": true
    }
  }
}

```

1 Обязательное поле

201 CREATED

Правило добавлено в стратегию.

Изменение правила стратегии (PATCH /backup_strategies/rules/{rule_id})

Также в query-параметрах передается `id` правила, информацию о котором необходимо изменить.

```

{
  "strategy_rule": {
    "client_id": "rubackup-client (525a99154f3505a2)",
    "strategy_id": "Default",
    "resource_type": "File system",
    "resource": "/home/",
    "normal_execution_script": "",
    "exception_execution_script": "",
    "priority": 100,
    "restore_script": ""
  },
  "module_extensions": {
    "parameters": {
      "file_list": true,
      "numeric_owner": false,
      "use_snapshot": false,
      "ignore_errors_snapshot": true,
      "snapshot_type": "",
      "snapshot_size": 10,
    }
  }
}

```

```

    "script_before_snapshot": "",
    "script_after_snapshot": "",
    "script_error_snapshot": ""
  }
},
"common_modules_extensions": {
  "parameters": {
    "worker_parallelism": 8,
    "enable_multithreading": false,
    "enable_flexible_dedup": false,
    "network_parallelism": 8,
    "memory_threshold": 0,
    "deny_memory_exceed": false,
    "restore_corrupted_blocks": "full_analysis",
    "deduplication": true
  }
}
}

```

200 OK

Параметры правила стратегии изменены.

400 BAD_REQUEST

Некорректное тело запроса.

404 NOT_FOUND

Правило стратегии не найдено.

Удаление списка правил стратегии (DELETE /backup_strategies/rules)

```

{
  "ids": [ ①
    1
  ]
}

```

① Обязательное поле

204 NO_CONTENT

Список правил стратегии удален.

404 NOT_FOUND

Правило стратегии не найдено.

Получение дополнительных параметров правила стратегии (GET /backup_strategies/extension)

Тело запроса отсутствует. В query-параметрах передается название ресурса (`resource_type`) или название ресурса и `id` правила, информацию о дополнительных параметрах которых необходимо получить.

200 OK

Информация о дополнительных параметрах получена.

Получение списка администраторов стратегии (GET /backup_strategies/{strategy_id}/administrators)

Тела запроса нет. В query-параметрах передается `id` стратегии, информацию об администраторе которой необходимо получить.

200 OK

Получен список администраторов для стратегии.

Назначение администраторов стратегии (POST /backup_strategies/{strategy_id}/administrators)

Также в query-параметрах передается `id` стратегии, которой должен быть назначен администратор.

```
{
  "administrator": "Nobody" ❶
}
```

❶ Обязательное поле

201 CREATED

Администратор для стратегии назначен.

Удаление списка администраторов стратегии (DELETE /backup_strategies/administrators)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

200 OK

Список администраторов стратегии удален.

404 NOT_FOUND

Администратор стратегии не найден.

Правила копирования в пул

Получение списка настроенного копирования резервных копий по стратегии в пул (GET /backup_strategies/{strategy_id}/replications)

200 OK

Получен список репликаций стратегии.

Получение информации о правиле копирования резервных копий по стратегии в пул (GET /backup_strategies/replications/{replication_id})

Тела запроса нет. В query-параметрах передается `id` правила репликации стратегии, информацию о котором необходимо получить.

200 OK

Получена информация о репликации стратегии.

Создание правила копирования резервных копий по стратегии в пул (POST /backup_strategies/{strategy_id}/replications)

В query-параметрах передается `id` стратегии, для которой необходимо добавить правило репликации.

```
{
  "pool": "file_pool", ①
  "capacity": 0,
  "capacity_enabled": false,
  "period": "1 mon",
  "period_enabled": false,
  "work_window_start": "00:00:00",
  "work_window_end": "00:00:00"
}
```

① Обязательное поле

201 CREATED

Правило репликации стратегии создано.

Редактирование правила копирования резервных копий по стратегии в пул (PATCH /backup_strategies/replications/{replication_id})

Также в query-параметрах передается `id` правила репликации стратегии, информацию о котором необходимо изменить.

```
{
```

```

"capacity": 0,
"capacity_enabled": false,
"period": "1 mon",
"period_enabled": false,
"work_window_start": "00:00:00",
"work_window_end": "00:00:00"
}

```

200 OK

Стратегия изменена.

400 BAD_REQUEST

Некорректное тело запроса.

404 NOT_FOUND

Стратегия не найдена.

Удаление списка настроенного копирования резервных копий по стратегиям (DELETE /backup_strategies/replications)

```

{
  "ids": [ ①
    1
  ]
}

```

① Обязательное поле

204 NO_CONTENT

Репликации стратегии удалены.

207 MULTI_STATUS

Часть репликаций удалена.

404 NOT_FOUND

Стратегия не найдена.

8.1.8. Правила глобального расписания и резервное копирование

Получение списка правил глобального расписания (GET /global_schedule)

200 OK

Параметры правила глобального расписания получены.

Получение дополнительных параметров глобального расписания (GET /global_schedule/extensions)

Тело запроса отсутствует. В query-параметрах передается название ресурса (`resource_type`) или название ресурса и `id` правила, информацию о дополнительных параметрах которых необходимо получить.

200 OK

Информация о дополнительных параметрах получена.

Получение информации о правиле глобального расписания (GET /global_schedule/{rule_id})

Тело запроса отсутствует. В query-параметрах передается `id` правила, информацию о котором необходимо получить.

200 OK

Информация о правиле глобального расписания получена.

404 NOT_FOUND

Правило глобального расписания не найдено.

Добавление правила глобального расписания (POST /global_schedule)

```
{
  "global_schedule": {
    "name": "global_schedule_name", ①
    "client_id": "rubackup-client (525a99154f3505a2)", ①
    "resource_type": "File system", ①
    "resource": "/home/", ①
    "backup_type": "full",
    "pool_id": "Default",
    "storage_capacity": 10,
    "client_storage_capacity": 10,
    "priority": 100,
    "crypto": "nocrypt",
    "validity_start_period": "2025-01-13T16:54:45.615738",
    "validity_end_period": "2026-01-13T16:54:45.615752",
    "periodic_launch": "1 min",
    "min": 0,
    "hour": 0,
    "dom": 1,
    "mon": 1,
    "dow": 1,
    "verify_flag": true,
    "verify_interval": "1 day",
```

```

    "storage_duration": "1 day",
    "move_copy_flag": false,
    "move_copy_pool": "Default",
    "move_copy_while": "1 day",
    "auto_delete_obsoleted_copy_flag": false,
    "inform_when_obsoleted_copy": "Nobody",
    "client_delete_flag": true,
    "notify_normal": "Nobody",
    "notify_normal_cc": "email@http://domain.ru/[domain.ru]",
    "notify_exception": "Nobody",
    "notify_exception_cc": "email@http://domain.ru/[domain.ru]",
    "notify_rule_validity": "Nobody",
    "notify_rule_validity_cc": "email@http://domain.ru/[domain.ru]",
    "notify_verify": "Nobody",
    "notify_verify_cc": "email@http://domain.ru/[domain.ru]",
    "status": "wait",
    "notify_normal_script": "",
    "notify_exception_script": "",
    "notify_end_storage_capacity": "Nobody",
    "notify_end_storage_capacity_cc": "email@http://domain.ru/[domain.ru]",
    "restore_script": ""
  },
  "module_extensions": {
    "parameters": {
      "file_list": true,
      "numeric_owner": false,
      "use_snapshot": false,
      "ignore_errors_snapshot": true,
      "snapshot_type": "",
      "snapshot_size": 10,
      "script_before_snapshot": "",
      "script_after_snapshot": "",
      "script_error_snapshot": ""
    }
  },
  "common_modules_extensions": {
    "parameters": {
      "worker_parallelism": 8,
      "enable_multithreading": false,
      "enable_flexible_dedup": false,
      "network_parallelism": 8,
      "memory_threshold": 0,
      "deny_memory_exceed": false,
      "restore_corrupted_blocks": "full_analysis",
      "deduplication": true
    }
  }
}

```

```

    }
  }
}

```

1 Обязательное поле

201 CREATED

Правило глобального расписания создано.

Изменение правила глобального расписания (PATCH /global_schedule/{rule_id})

Также в query-параметрах передается `id` правила, информацию о котором необходимо изменить.

```

{
  "global_schedule": {
    "name": "global_schedule_name",
    "pool_id": "Default",
    "storage_capacity": 10,
    "client_storage_capacity": 10,
    "priority": 100,
    "validity_start_period": "2025-01-13T16:54:45.615738",
    "validity_end_period": "2026-01-13T16:54:45.615752",
    "periodic_launch": "1 min",
    "min": 0,
    "hour": 0,
    "dom": 1,
    "mon": 1,
    "dow": 1,
    "verify_flag": true,
    "verify_interval": "1 day",
    "storage_duration": "1 day",
    "move_copy_flag": false,
    "move_copy_pool": "Default",
    "move_copy_while": "1 day",
    "auto_delete_obsoleted_copy_flag": false,
    "inform_when_obsoleted_copy": "Nobody",
    "client_delete_flag": true,
    "notify_normal": "Nobody",
    "notify_normal_cc": "email@http://domain.ru/[domain.ru]",
    "notify_exception": "Nobody",
    "notify_exception_cc": "email@http://domain.ru/[domain.ru]",
    "notify_rule_validity": "Nobody",
    "notify_rule_validity_cc": "email@http://domain.ru/[domain.ru]",
    "notify_verify": "Nobody",
  }
}

```

```

    "notify_verify_cc": "email@http://domain.ru/[domain.ru]",
    "status": "wait",
    "notify_normal_script": "",
    "notify_exception_script": "",
    "notify_end_storage_capacity": "Nobody",
    "notify_end_storage_capacity_cc": "email@http://domain.ru/[domain.ru]",
    "restore_script": ""
  },
  "module_extensions": {
    "parameters": {
      "file_list": true,
      "numeric_owner": false,
      "use_snapshot": false,
      "ignore_errors_snapshot": true,
      "snapshot_type": "",
      "snapshot_size": 10,
      "script_before_snapshot": "",
      "script_after_snapshot": "",
      "script_error_snapshot": ""
    }
  },
  "common_modules_extensions": {
    "parameters": {
      "worker_parallelism": 8,
      "enable_multithreading": false,
      "enable_flexible_dedup": false,
      "network_parallelism": 8,
      "memory_threshold": 0,
      "deny_memory_exceed": false,
      "restore_corrupted_blocks": "full_analysis",
      "deduplication": true
    }
  }
}

```

200 OK

Параметры правила глобального расписания изменены.

400 BAD_REQUEST

Некорректное тело запроса.

404 NOT_FOUND

Правило глобального расписания не найдено.

Изменение статуса правила глобального расписания (PATCH /global_schedule)

```
{
  "rules": [
    {
      "id": 1, ❶
      "status": "wait" ❶
    }
  ]
}
```

❶ Обязательное поле

200 OK

Статус правила изменен.

400 BAD_REQUEST

Некорректное тело запроса.

404 NOT_FOUND

Правило глобального расписания не найдено.

Удаление списка правил глобального расписания (DELETE /global_schedule)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

200 OK

Правило глобального расписания удалено.

Создание полной резервной копии по правилу глобального расписания (POST /global_schedule/{rule_id}/execution)

Тело запроса отсутствует. В query-параметрах передается `id` правила, которое необходимо выполнить.

200 OK

Правило глобального расписания выполнено.

404 NOT_FOUND

Правило глобального расписания не найдено.

Репликации в глобальном расписании

Получение списка настроенных репликаций правил глобального расписания (GET /global_schedule/{rule_id}/replication)

200 OK

Получен список правил репликации.

Получение информации о настроенной репликации правила глобального расписания (GET /global_schedule/replication/{replication_id})

В query-параметрах передается `id` правила репликации правила глобального расписания, параметры которого необходимо получить.

200 OK

Получена информация правиле репликации.

Создание правила репликации в глобальном расписании (POST /global_schedule/{rule_id}/replication)

Также в query-параметрах передается `id` правила глобального расписания, для которого необходимо создать правило репликации.

```
{
  "pool": "Default", ①
  "capacity": 0,
  "capacity_enabled": false,
  "period": "1 mon",
  "period_enabled": false,
  "work_window_start": "00:00:00",
  "work_window_end": "00:00:00"
}
```

① Обязательное поле

201 CREATED

Правило репликации создано.

Изменение параметров репликации правила глобального расписания (PATCH /global_schedule/replication/{replication_id})

Также в query-параметрах передается `id` правила репликации правила глобального расписания, параметры которого необходимо изменить.

```
{
```

```

"capacity": 0,
"capacity_enabled": false,
"period": "1 mon",
"period_enabled": false,
"work_window_start": "00:00:00",
"work_window_end": "00:00:00"
}

```

200 OK

Правило репликации изменено.

400 BAD_REQUEST

Некорректное тело запроса.

404 NOT_FOUND

Правило репликации не найдено.

Удаление списка настроенных репликаций правил глобального расписания (DELETE /global_schedule/replication)

```

{
  "ids": [ ❶
    1
  ]
}

```

❶ Обязательное поле

204 NO_CONTENT

Правило репликации удалены.

207 MULTI_STATUS

Часть правил репликации удалена.

404 NOT_FOUND

Правило репликации не найдено.

Ограничения пропускной способности в глобальном расписании

Получение списка ограничений пропускной способности правил глобального расписания (GET /global_schedule/bandwidth)

200 OK

Получен список ограничений пропускной способности.

Получение информации об ограничении пропускной способности правила глобального расписания (GET /global_schedule/bandwidth/{bandwidth_id})

В query-параметрах передается `id` ограничения пропускной способности правила глобального расписания, информацию о котором необходимо получить.

200 OK

Получена информация об ограничении пропускной способности.

Ограничение пропускной способности правила глобального расписания (POST /global_schedule/{rule_id}/bandwidth)

Также в query-параметрах передается `id` правила глобального расписания, для которого необходимо создать ограничение.

```
{
  "backup_bandwidth": 0,
  "restore_bandwidth": 0,
  "window_start": "00:00:00",
  "window_end": "00:00:00"
}
```

201 CREATED

Ограничение пропускной способности создано.

Изменение ограничения пропускной способности правила глобального расписания (PATCH /global_schedule/bandwidth/{bandwidth_id})

Также в query-параметрах передается `id` ограничения пропускной способности правила глобального расписания, параметры которого необходимо изменить.

```
{
  "backup_bandwidth": 0,
  "restore_bandwidth": 0,
  "window_start": "00:00:00",
  "window_end": "00:00:00"
}
```

200 OK

Ограничение пропускной способности изменено.

400 BAD_REQUEST

Некорректное тело запроса.

404 NOT_FOUND

Ограничение пропускной способности не найдено.

Удаление списка ограничений пропускной способности правил глобального расписания (DELETE /global_schedule/bandwidth)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

204 NO_CONTENT

Ограничение пропускной способности удалены.

207 MULTI_STATUS

Часть ограничений пропускной способности удалена.

404 NOT_FOUND

Ограничение пропускной способности не найдено.

Запросы клиентов на добавление и удаление правил

Создание правил

Получение списка запросов клиента на добавление правила глобального расписания (GET /global_schedule/requests/creation)

200 OK

Получен список запросов клиента на добавление правила глобального расписания.

Получение информации о запросе клиента на добавление правила глобального расписания (GET /global_schedule/requests/creation/{request_id})

Тела запроса нет. В query-параметрах передается `id` запроса, информацию о котором необходимо получить.

200 OK

Получена информация о запросе клиента на добавление правила глобального расписания.

Одобрение запроса клиента на добавление правила глобального расписания (POST /global_schedule/requests/creation/approve)

```
{
  "ids": [ ❶
    1
  ]
}
```

```
}
```

1 Обязательное поле

204 NO_CONTENT

Запросы клиента на добавление правила глобального расписания одобрены.

207 MULTI_STATUS

Часть из списка запросов клиента на добавление правила глобального расписания одобрена, часть нет.

404 NOT_FOUND

Запросы клиента на добавление правила глобального расписания не найдены.

Отклонение запроса клиента на добавление правила глобального расписания (DELETE /global_schedule/requests/creation/decline)

```
{
  "ids": [ 1
]
}
```

1 Обязательное поле

204 NO_CONTENT

Запросы клиента на добавление правила глобального расписания отклонены.

207 MULTI_STATUS

Часть из списка запросов клиента на добавление правила глобального расписания отклонена, часть нет.

404 NOT_FOUND

Запросы клиента на добавление правила глобального расписания не найдены.

Удаление правил

Получение списка запросов клиента на удаление правила глобального расписания (GET /global_schedule/requests/deletion)

200 OK

Получен запросов клиента на удаление правила глобального расписания.

Получение информации о запросе клиента на удаление правила глобального расписания (GET /global_schedule/requests/deletion/{request_id})

Тела запроса нет. В query-параметрах передается `id` запроса, информацию о

котором необходимо получить.

200 OK

Получена информация о запросе клиента на удаление правила глобального расписания.

Одобрение запроса клиента на удаление правила глобального расписания (POST /global_schedule/requests/deletion/approve)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

204 NO_CONTENT

Запросы клиента на удаление правила глобального расписания одобрены.

207 MULTI_STATUS

Часть из списка запросов клиента на удаление правила глобального расписания одобрена, часть нет.

404 NOT_FOUND

Запросы клиента на удаление правила глобального расписания не найдены.

Отклонение запроса клиента на удаление правила глобального расписания (DELETE /global_schedule/requests/deletion/decline)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

204 NO_CONTENT

Запросы клиента на удаление правила глобального расписания отклонены.

207 MULTI_STATUS

Часть из списка запросов клиента на удаление правила глобального расписания отклонена, часть нет.

404 NOT_FOUND

Запросы клиента на удаление правила глобального расписания не найдены.

8.1.9. Срочное резервное копирование**Создание срочной полной резервной копии (POST /urgent_backup)**

```
{
  "task": {
    "resource_type": "File system", ①
    "resource": "/home/", ①
    "client": "rubcp-client (525a99154f3505a2)", ①
    "backup_type": "full", ①
    "pool": "Default", ①
    "crypto": "nocrypt", ①
    "priority": 100, ①
    "storage_duration": "1 year", ①
    "archiving": false ①
  },
  "module_extensions": {
    "parameters": {
      "file_list": true,
      "numeric_owner": false,
      "use_snapshot": false,
      "ignore_errors_snapshot": true,
      "snapshot_type": "",
      "snapshot_size": 10,
      "script_before_snapshot": "",
      "script_after_snapshot": "",
      "script_error_snapshot": ""
    }
  },
  "common_modules_extensions": {
    "parameters": {
      "worker_parallelism": 8,
      "enable_multithreading": false,
      "enable_flexible_dedup": false,
      "network_parallelism": 8,
      "memory_threshold": 0,
      "deny_memory_exceed": false,
      "restore_corrupted_blocks": "full_analysis",
      "deduplication": true
    }
  }
}
```

```
}
```

1 Обязательное поле

201 CREATED

Создана задача на срочное РК.

8.1.10. Резервные копии

Получение списка резервных копий (GET /repositories)

200 OK

Получен список сделанных резервных копий.

Получение информации о резервных копиях (GET /repositories/{repository_id})

200 OK

Получена информация о резервной копии.

Изменение периода хранения резервной копии (PATCH /repositories/{repository_id})

Также в query-параметрах передается id резервной копии, срок хранения которой необходимо изменить.

```
{
  "store_until": "2025-02-04T14:11:19.454204" 1
}
```

1 Обязательное поле

200 OK

Период хранения резервной копии изменен.

400 BAD_REQUEST

Некорректное тело запроса.

404 NOT_FOUND

Резервная копия не найдена.

Удаление списка резервных копий из репозитория (DELETE /repositories)

```
{
  "ids": [ 1
]
```

```
}
```

1 Обязательное поле

200 OK

Список резервных копий удален.

404 NOT_FOUND

Резервная копия не найдена.

Получение списка зависимостей для нескольких цепочек (POST /repositories/deletion_reference_chain)

```
{
  "ids": [ 1
]
}
```

1 Обязательное поле

200 OK

Получен список зависимостей для цепочки резервных копий.

Получение дополнительных параметров восстановления (GET /repositories/extensions)

200 OK

Получен список дополнительных параметров восстановления.

Копирование полной резервной копии (POST /repositories/{repository_id}/copy)

Тела запроса нет. В query-параметрах передается `id` резервной копии и названия пула, в который необходимо провести копирование.

201 CREATED

Задача на копирование резервной копии создана.

404 NOT_FOUND

Резервная копия не найдена.

Перемещение полной резервной копии (POST /repositories/{repository_id}/move)

Тела запроса нет. В query-параметрах передается `id` резервной копии и названия пула, в который необходимо провести перемещение.

201 CREATED

Задача на перемещение резервной копии создана.

404 NOT_FOUND

Резервная копия не найдена.

Восстановление резервной копии (POST /repositories/{repository_id}/restore)

Также в query-параметрах передается `id` резервной копии, которую необходимо восстановить. Для цепочек копий передается `id` последней разностной копии.

```
{
  "restore_task": {
    "target_client": "rubackup-client (525a99154f3505a2)", ❶
    "destination_path": "/home/", ❶
    "data_deployment": false,
    "granular_restore_files": [
      ""
    ]
  },
  "module_extensions": {
    "parameters": {
      "new_name": ""
    }
  },
  "common_modules_extensions": {
    "parameters": {
      "worker_parallelism": 8,
      "memory_threshold": 0
    }
  }
}
```

❶ Обязательное поле

201 CREATED

Задача на восстановление резервной копии создана.

400 BAD_REQUEST

Некорректное тело запроса.

404 NOT_FOUND

Резервная копия не найдена.

Верификация резервной копии (POST /repositories/{repository_id}/verification)

Тела запроса нет. В query-параметрах передается `id` резервной копии, которую необходимо верифицировать.

201 CREATED

Задача на верификацию резервной копии создана.

404 NOT_FOUND

Резервная копия не найдена.

Получение списка клиентов, доступных для восстановления резервной копии (GET /repositories/{repository_id}/restore/available_clients)**200 OK**

Получен список клиентов, доступных для восстановления копии.

Получение списка ресурсов, включенных в копию для гранулярного восстановления (GET /repositories/{repository_id}/granular_list)

Тела запроса нет. В query-параметрах передается `id` резервной копии, список файлов которой необходимо получить.

200 OK

Получен список ресурсов, включенных в копию для гранулярного восстановления.

8.1.11. Задачи

Получение очереди задач (GET /tasks)**200 OK**

Получен список задач из очереди задач.

Получение дополнительных параметров созданной задачи (GET /tasks/extensions)

Тело запроса отсутствует. В query-параметрах передается название ресурса (`resource_type`) или название ресурса и `id` правила, информацию о дополнительных параметрах которых необходимо получить.

200 OK

Информация о дополнительных параметрах получена.

Получение информации о задаче из очереди (GET /tasks/{task_id})

Тело запроса отсутствует. В query-параметрах передается `id` задачи, информацию о которой необходимо получить.

200 OK

Информация о задаче из очереди получена.

404 NOT_FOUND

Задача не найдена в очереди.

Удаление нескольких задач (DELETE /tasks)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

204 NO_CONTENT

Список задач из очереди задач удален.

404 NOT_FOUND

Задача не найдена в очереди.

(DELETE /task_queue/{status}) Удаление задачи с определенным статусом

Возможные статусы: `error`, `killed`, `obsoleted` (устаревшие, в статусе *done*).

Тело запроса отсутствует. В `query`-параметрах передается статус, задачи с которым необходимо удалить.

204 NO_CONTENT

Задачи с выбранным статусом удалены из очереди.

404 NOT_FOUND

Задач с выбранным статусом не найдена в очереди.

Перезапуск задачи (POST /tasks/restart)

```
{
  "data": {
    "tasks_ids": [ ❶
      1
    ]
  }
}
```

1 Обязательное поле

204 NO_CONTENT

Задача перезапущена.

400 BAD_REQUEST

Некорректное тело запроса (задачу с таким статусом нельзя перезапустить).

Приостановка выполнения задачи (POST /tasks/{task_id}/pause)

Тела запроса нет. В query-параметрах передается `id` задачи, которую необходимо поставить на паузу.

204 NO_CONTENT

Задача приостановлена.

400 BAD_REQUEST

Некорректное тело запроса (задачу с таким статусом нельзя приостановить).

Возобновление приостановленной задачи (POST /tasks/{task_id}/continue)

```
{
  "data": {
    "task_id": 1 1
  }
}
```

1 Обязательное поле

204 NO_CONTENT

Задача возобновлена.

400 BAD_REQUEST

Некорректное тело запроса (задачу с таким статусом нельзя возобновить).

Убийство задачи (POST /tasks/{task_id}/kill)

Тела запроса нет. В query-параметрах передается `id` задачи, которую необходимо убить.

204 NO_CONTENT

Задача убита.

400 BAD_REQUEST

Некорректное тело запроса (задачу с таким статусом нельзя убить).

Получение журнала клиентских операций по задаче (GET /tasks/{task_id}/info/clients)

Тела запроса нет. В query-параметрах передается `id` задачи из очереди задач, журнал которой необходимо получить.

200 OK

Получен журнал клиентских операций по задаче.

Получение журнала серверных операций по задаче (GET /tasks/{task_id}/info/servers)

Тела запроса нет. В query-параметрах передается `id` задачи из очереди задач, журнал которой необходимо получить.

200 OK

Получен журнал серверных операций по задаче.

Очередь задач облака**Получение задач очереди облаков (GET /tasks/clouds/s3)****200 OK**

Получен список задач в очереди задач облаков.

Получение информации о задаче из очереди облаков (GET /tasks/clouds/s3/{task_id})**200 OK**

Получена информация о задаче в очереди задач облаков.

404 NOT_FOUND

Задача не найдена.

Удаление списка задач из очереди облаков (DELETE /tasks/clouds/s3)

```
{
  "ids": [ ①
    1
  ]
}
```

① Обязательное поле

204 NO_CONTENT

Удален список задач из очереди задач облаков.

Удаление ошибочных задач из очереди облаков (DELETE /tasks/clouds/s3/errors_deletion)

204 NO_CONTENT

Удален список ошибочных задач из очереди задач облаков.

Очередь задач ленточных библиотек

Получение списка задач из очереди ленточных библиотек (GET /tasks/tape_libraries)

200 OK

Получен список задач из очереди задач ленточных библиотек.

Получение информации о задаче из очереди ленточных библиотек (GET /tasks/tape_libraries/{task_id})

Тело запроса отсутствует. В query-параметрах передается `id` задачи из очереди задач ленточных библиотек, информацию о которой необходимо получить.

200 OK

Получена информация о задаче из очереди задач ленточных библиотек.

404 NOT_FOUND

Задача в очереди задач ленточных библиотек не найдена.

8.1.12. Медиасерверы

Получение списка медиасерверов (GET /media_servers)

200 OK

Получен список медиасерверов.

Получение информации о медиасервере (GET /media_servers/{server_id})

Тела запроса нет. В query-параметрах передается `id` медиасервера, информацию о котором необходимо получить.

200 OK

Получена информация о медиасервере.

Редактирование медиасервера (PATCH /media_servers/{server_id})

Также в query-параметрах передается `id` медиасервера, информацию о котором необходимо изменить.

```
{
  "description": ""
}
```

200 OK

Параметры медиасервера изменены.

400 BAD_REQUEST

Некорректное тело запроса.

404 NOT_FOUND

Медиасервер не найден.

Удаление нескольких медиасерверов (DELETE /media_servers)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

200 OK

Список медиасерверов удален.

404 NOT_FOUND

Медиасервер не найден.

Неавторизованные медиасерверы**Получение списка неавторизованных медиасерверов (GET /media_servers/unauthorised)****200 OK**

Получен список неавторизованных медиасерверов.

Получение информации о неавторизованном медиасервере (GET /media_servers/unauthorised/{server_id})

Тело запроса отсутствует. В query-параметрах передается `id` неавторизованного медиасервера, информацию о котором необходимо получить.

200 OK

Получена информация о неавторизованном медиасервере.

Авторизация медиасервера (POST /media_servers/unauthorised/authorise)

```
{
  "hostname": "my_hostname" ❶
}
```

1 Обязательное поле

200 OK

Медиасервер авторизован.

Удаление списка неавторизованных медиасерверов (DELETE /media_servers/unauthorised)

```
{
  "ids": [ 1
]
}
```

1 Обязательное поле

200 OK

Список неавторизованных медиасерверов удален.

404 NOT_FOUND

Неавторизованные медиасерверы не найдены.

8.1.13. Пулы

Получение списка пулов (GET /pools)

200 OK

Получен список существующих пулов.

Получение информации о пуле (GET /pools/{pool_id})

Тела запроса нет. В query-параметрах передается id пула, информацию о котором необходимо получить.

200 OK

Получена информация о пуле.

Добавление пула (POST /pools)

Набор параметров, которые необходимо передать, зависит от типа пула. В случае, если переданы лишние параметры (которые не подходят создаваемому типу пула), то такие параметры игнорируются.

Параметр `compression_type` не нужно указывать для пулов типа **Block device** и **Client defined**.

Параметры `block_size`, `hash_length`, `algorithm`, `meta_data_pool` указываются только для пулов типа **Block device**.

```
{
  "pool_name": "my_pool_name", ❶
  "pool_type": "Block device", ❶
  "media_server": "rubackup-server", ❶
  "compression_type": "None",
  "description": "",
  "retention_period": "1 year",
  "ignore_free_space_check": false,
  "meta_data_pool": "Default",
  "algorithm": "streebog",
  "block_size": 131072,
  "hash_length": 256
}
```

❶ Обязательное поле

201 CREATED

Пул создан.

Редактирование пула (PATCH /pools/{pool_id})

Также в query-параметрах передается `id` пула, информацию о котором необходимо изменить.

Набор параметров, которые необходимо передать, зависит от типа пула. В случае, если переданы лишние параметры (которые не подходят создаваемому типу пула), то такие параметры игнорируются.

Параметр `compression_type` не нужно указывать для пулов типа **Block device** и **Client defined**.

Параметры `meta_data_pool`, `move_old_metadata` указываются только для пулов типа **Block device**.

```
{
  "pool_name": "my_pool_name",
  "compression_type": "None",
  "description": "",
  "retention_period": "1 year",
  "ignore_free_space_check": false,
  "meta_data_pool": "Default",
  "move_old_metadata": false
}
```

200 OK

Параметры пула изменены.

400 BAD_REQUEST

Некорректное тело запроса.

404 NOT_FOUND

Пул не найден.

Удаление списка пулов (DELETE /pools)

```
{
  "ids": [ ①
    1
  ]
}
```

① Обязательное поле

204 NO_CONTENT

Пулы удалены.

207 MULTI_STATUS

Часть пулов удалена.

404 NOT_FOUND

Пул не найден.

Получение списка пулов, доступных для перемещения и копирования (GET /pools/destination_available)

В query-параметрах передается имя пула, для которого необходимо определить доступные для перемещения или копирования пулы.

Также в query-параметрах передается действие `copy` или `move`.

200 OK

Получен список доступных для перемещения и копирования пулов.

Динамические группы пулов**Получение списка динамических групп пулов (GET /pools/groups)****200 OK**

Получен список групп пулов.

Получение информации о динамической группе пулов (GET /pools/groups/{group_id})

Тела запроса нет. В query-параметрах передается `id` группы пулов, информацию о которой необходимо получить.

200 OK

Получена информация о группе пулов.

Добавление динамической группы пулов (POST /pools/groups)

```
{
  "name": " ", ①
  "max_pool_tasks": 10,
  "max_media_server_tasks": 10,
  "max_cpu_usage": 80,
  "max_in_block_operations": 0,
  "max_out_block_operations": 0,
  "calculation_period": 10,
  "description": " "
}
```

① Обязательное поле

201 CREATED

Группа пулов создана.

Изменение динамической группы пулов (PATCH /pools/groups/{group_id})

Также в query-параметрах передается `id` группы пулов, информацию о которой необходимо изменить.

```
{
  "name": " ",
  "max_pool_tasks": 10,
  "max_media_server_tasks": 10,
  "max_cpu_usage": 80,
  "max_in_block_operations": 0,
  "max_out_block_operations": 0,
  "calculation_period": 10,
  "description": " "
}
```

200 OK

Группа пулов изменена.

400 BAD_REQUEST

Некорректное тело запроса.

404 NOT_FOUND

Группа пулов не найдена.

Удаление списка динамических групп пулов (DELETE `/pools/groups`)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

204 NO_CONTENT

Группы пулов удалены.

207 MULTI_STATUS

Часть групп пулов удалена.

404 NOT_FOUND

Группа пулов не найдена.

Получение списка добавленных в динамическую группу пулов (GET `/pools/groups/relationships`)

200 OK

Получен список пулов динамической группы.

Получение информации о добавленном в динамическую группу пуле (GET `/pools/groups/relationships/{record_id}`)

Тела запроса нет. В query-параметрах передается `id` пула из группы пулов, информацию о котором необходимо получить.

200 OK

Получена информация о пуле динамической группы.

Добавление пула в динамическую группу пулов (POST `/pools/groups/relationships`)

```
{
  "pool": "Default", ❶
  "dynamic_pool_group": 1, ❶
  "enabled": true
}
```

```
}
```

① Обязательное поле

201 CREATED

Пул добавлен в группу.

Удаление списка добавленных в динамическую группу пулов (DELETE /pools/groups/relationships)

```
{
  "ids": [ ①
    1
  ]
}
```

① Обязательное поле

204 NO_CONTENT

Пулы удалены из динамической группы.

207 MULTI_STATUS

Часть пулов удалена из динамической группы.

404 NOT_FOUND

Правило подмены пулов не найдено.

Подмена пулов

Получение списка подмены пулов (GET /pools/substitutions)

200 OK

Получен список настроенных подмен пулов.

Получение информации о подмене пулов (GET /pools/substitution/{substitution_id})

Тела запроса нет. В query-параметрах передается id правила подмены пулов, информацию о котором необходимо получить.

200 OK

Получена информация о правиле подмены пулов.

Добавление подмены пулов (POST /pools/substitutions)

```
{
  "pool": "Default", ①
  "substitution": "Default2" ①
}
```

```
}
```

1 Обязательное поле

201 CREATED

Создано правило подмены пулов.

Удаление списка подмены пулов (DELETE /pools/substitutions)

```
{
  "ids": [ 1
]
}
```

1 Обязательное поле

204 NO_CONTENT

Правила подмены пулов удалены.

207 MULTI_STATUS

Часть правил подмены пулов удалена.

404 NOT_FOUND

Правило подмены пулов не найдено.

8.1.14. Журналы

Журнал глобального расписания

Получение списка записей в журнале глобального расписания (GET /global_schedule/logs)

200 OK

Получен список записей в журнале правил глобального расписания.

Получение информации о записи в журнале глобального расписания (GET /global_schedule/logs/{record_id})

Тела запроса нет. В query-параметрах передается id операции из журнала правил глобального расписания, информацию о которой необходимо получить.

200 OK

Получена запись в журнале правил глобального расписания.

404 NOT_FOUND

Запись в журнале правил глобального расписания не найдена.

Журнал операций клиента

Получение журнала со списком операций клиентов (GET `/clients/logs`)

200 OK

Получен список записей в журнале операций клиентов.

Получение одной записи из журнала с операций клиентов (GET `/clients/logs/{log_id}`)

Тела запроса нет. В query-параметрах передается `id` операции клиента, информацию о которой необходимо получить.

200 OK

Получена запись в журнале операций клиентов.

404 NOT_FOUND

Запись в журнале операций клиентов не найдена.

Получение общих настроек журналов для всех клиентов (GET `/clients/logs/common_settings`)

200 OK

Получены общие настройки журналов клиентов.

Изменение общих настроек журналов для всех клиентов (PATCH `/clients/logs/common_settings`)

```
{
  "log_file_max_size": 1,
  "log_file_max_rotate": 10,
  "log_level": "Debug",
  "apply_to_all_clients": false
}
```

200 OK

Общие настройки журналов клиентов изменены.

400 BAD_REQUEST

Некорректное тело запроса.

Получение списка всех персональных настроек журналов клиентов (GET `/clients/logs/personal_settings`)

200 OK

Получены персональные настройки журналов клиентов.

Создание персональных настроек журналов клиента (POST `/clients/logs/personal_settings`)

```
{
```

```

"log_level": "Debug", ❶
"log_file_max_size": 1, ❶
"log_file_max_rotate": 10, ❶
"client_id": "rubackup-client (525a99154f3505a2)" ❶
}

```

❶ Обязательное поле

201 CREATED

Персональные настройки журналов клиентов созданы.

400 BAD_REQUEST

Некорректное тело запроса.

Удаление персональных настроек журналов клиента (DELETE /clients/logs/personal_settings)

```

{
  "ids": [ ❶
    1
  ]
}

```

❶ Обязательное поле

200 OK

Персональные настройки журналов клиентов удалены.

404 NOT_FOUND

Персональные настройки журналов клиентов не найдены.

Изменение персональных настроек журналов клиентов (PATCH /clients/{client_id}/logs/personal_settings)

Также в query-параметрах передается `id` клиента, персональные настройки которого необходимо изменить.

```

{
  "log_level": "Debug",
  "log_file_max_size": 1,
  "log_file_max_rotate": 10
}

```

200 OK

Персональные настройки журналов клиентов изменены.

400 BAD_REQUEST

Некорректное тело запроса.

Получение значений по умолчанию настроек журналов клиентов (GET `/clients/logs/settings/defaults`)

200 OK

Получены настройки по умолчанию журналов клиентов.

Журналы медиасерверов

Получение журнала со списком операций медиасерверов (GET `/media_servers/logs`)

200 OK

Получен список записей в журнале операций медиасерверов.

Получение журнала с операцией медиасервера (GET `/media_servers/logs/{log_id}`)

Тела запроса нет. В query-параметрах передается `id` записи, информацию о которой необходимо получить.

200 OK

Получена запись из журнала операций медиасерверов.

404 NOT_FOUND

Запись в журнале операций медиасерверов не найдена.

Получение значений по умолчанию настроек журналов для медиасерверов (GET `/media_servers/logs/settings/defaults`)

200 OK

Получены настройки по умолчанию журналов медиасерверов.

Получение списка всех персональных настроек журналов медиасерверов (GET `/media_servers/logs/personal_settings`)

200 OK

Получены персональные настройки журналов медиасерверов.

Создание персональных настроек журналов медиасервера (POST `/media_servers/logs/personal_settings`)

```
{
  "log_level": "Debug", ①
  "log_file_max_size": 1, ①
  "log_file_max_rotate": 10, ①
  "media_server": "rubackup-client (525a99154f3505a2)" ①
}
```

① Обязательное поле

201 CREATED

Персональные настройки журналов медиасерверов созданы.

400 BAD_REQUEST

Некорректное тело запроса.

Изменение персональных настроек журналов медиасервера (PATCH /media_servers/{server_id}/logs/personal_settings)

Также в query-параметрах передается `id` клиента, персональные настройки которого необходимо изменить.

```
{
  "log_level": "Debug",
  "log_file_max_size": 1,
  "log_file_max_rotate": 10
}
```

200 OK

Персональные настройки журналов медиасерверов изменены.

400 BAD_REQUEST

Некорректное тело запроса.

Удаление персональных настроек журналов медиасервера (DELETE /media_servers/logs/personal_settings)

```
{
  "ids": [ ①
    1
  ]
}
```

① Обязательное поле

200 OK

Персональные настройки журналов медиасерверов удалены.

404 NOT_FOUND

Персональные настройки клиента не найдены.

Получение общих настроек журналов для всех медиасерверов (GET /media_servers/logs/common_settings)**200 OK**

Получены общие настройки журналов медиасерверов.

Изменение общих настроек журналов для всех медиасерверов (PATCH `/media_servers/logs/common_settings`)

```
{
  "log_file_max_size": 1,
  "log_file_max_rotate": 10,
  "log_level": "Debug",
  "apply_to_all_media_servers": false
}
```

200 OK

Общие настройки журналов медиасерверов изменены.

400 BAD_REQUEST

Некорректное тело запроса.

Журналы мониторинга сервера

Получение списка записей в журнале мониторинга сервера (GET `/media_servers/system_monitoring`)

200 OK

Получен список записей в журнале мониторинга сервера.

Получение информации о записи в журнале мониторинга сервера (GET `/media_servers/system_monitoring/{row_id}`)

Тела запроса нет. В query-параметрах передается `id` записи в журнале мониторинга сервера, информацию о которой необходимо получить.

200 OK

Получена информация о записи в журнале мониторинга сервера.

404 NOT_FOUND

Запись в журнале мониторинга сервера не найдена.

Журнал репозитория

Получение списка записей в журнале репозитория (GET `/repositories/logs`)

200 OK

Получен список записей в журнале репозитория.

Получение информации о записи в журнале репозитория (GET `/repositories/logs/{record_id}`)

Тела запроса нет. В query-параметрах передается `id` записи в журнале репозитория, информацию о которой необходимо получить.

200 OK

Получена запись в журнале репозитория.

404 NOT_FOUND

Запись в журнале репозитория не найдена.

Журнал задач

Получение списка записей в журнале задач (GET /tasks/logs)

200 OK

Получен список записей в журнале задач.

Получение информации о записи в журнале задач (GET /tasks/logs/{record_id})

Тела запроса нет. В query-параметрах передается `id` записи в журнале задач, информацию о которой необходимо получить.

200 OK

Получена запись в журнале задач.

404 NOT_FOUND

Запись в журнале задач не найдена.

Журнал событий информационной безопасности

Просмотр журналов событий информационной безопасности (GET /security/audit/logs)

200 OK

Список записей в журнале информационной безопасности получен.

Просмотр записи из журнала событий информационной безопасности (GET /security/audit/logs/{record_id})

Тела запроса нет. В query-параметрах передается `id` записи, информацию о которой необходимо получить.

200 OK

Информация о записи из журнала информационной безопасности получена.

Удаление списка записей из журналов событий информационной безопасности (DELETE /security/audit/logs)

```
{
  "ids": [ 1
    1
  ]
}
```

}

1 Обязательное поле

204 NO_CONTENT

Список записей из журналов информационной безопасности удален.

8.1.15. Отчеты

Получение информации об отчете (GET /users/reports/{report_id})

Тела запроса нет. В query-параметрах передается id отчета, информацию о котором необходимо получить.

200 OK

Получена информация об отчете.

404 NOT_FOUND

Отчет не найден.

Получение списка отчетов (GET /users/reports)

200 OK

Получен список отчетов.

Создание отчета (POST /users/reports)

```
{
  "name": "my_report_name", 1
  "owner": "Nobody", 1
  "min": 0,
  "hour": 0,
  "dom": 1,
  "mon": 1,
  "dow": 1,
  "perpetually": true,
  "start_period": "2025-02-04T14:11:19.667438",
  "end_period": "2026-02-04T14:11:19.667453",
  "status": "wait",
  "notify": "Nobody",
  "notify_cc": "email@http://domain.ru[domain.ru]",
  "description": ""
}
```

1 Обязательное поле

201 CREATED

Отчет создан.

Изменение отчета (PATCH /users/reports/{report_id})

Также в query-параметрах передается `id` отчета, информацию о котором необходимо изменить.

```
{
  "min": 0,
  "hour": 0,
  "dom": 1,
  "mon": 1,
  "dow": 1,
  "perpetually": true,
  "start_period": "2025-02-04T14:11:19.667438",
  "end_period": "2026-02-04T14:11:19.667453",
  "status": "wait",
  "notify": "Nobody",
  "notify_cc": "email@http://domain.ru[domain.ru]",
  "description": ""
}
```

200 OK

Параметры отчета изменены.

400 BAD_REQUEST

Некорректное тело запроса.

404 NOT_FOUND

Отчет не найден.

Изменение статуса отчета (PATCH /users/reports)

```
{
  "reports": [
    {
      "id": 1, ①
      "status": "wait" ①
    }
  ]
}
```

① Обязательное поле

200 OK

Статус отчета изменен.

400 BAD_REQUEST

Некорректное тело запроса.

404 NOT_FOUND

Отчет не найден.

Удаление списка отчетов (DELETE /users/reports)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

204 NO_CONTENT

Список отчетов удален.

404 NOT_FOUND

Отчет не найден.

8.1.16. Microsoft Active Directory**Соединение с контроллером домена MS Active Directory****Получение параметров настроенного соединения с MS Active Directory (GET /auth/ldap/configuration)****200 OK**

Получены параметры соединения с контроллером домена.

Создание соединения с контроллером домена MS Active Directory (POST /auth/ldap/configuration)

```
{
  "domain_type": "MS AD", ❶
  "protocol": "ldap", ❶
  "address": "", ❶
  "port": 389, ❶
  "client_cert": "",
  "root_cert": "",
  "domain": "", ❶
  "login": "", ❶
}
```

```

"password": "", ❶
"user_search_base": "", ❶
"group_search_base": "" ❶
}

```

❶ Обязательное поле

201 CREATED

Соединение с контроллером домена создано.

Удаление соединения с контроллером домена (DELETE /auth/ldap/configuration)

204 NO_CONTENT

Соединение с контроллером домена удалено.

Ассоциация групп MS Active Directory и ролей RuBackup

Получение списка ассоциаций групп MS Active Directory и ролей RuBackup (GET /auth/ldap/assosiations)

200 OK

Получен список ассоциаций групп MS Active Directory и ролей RuBackup.

Ассоциация групп MS Active Directory и ролей RuBackup (POST /auth/ldap/assosiations)

```

{
  "associations": [
    {
      "domain_group": "Administrators", ❶
      "rubackup_role": "auditor" ❶
    }
  ]
}

```

❶ Обязательное поле

201 CREATED

Ассоциация группы MS Active Directory и ролей RuBackup создана.

Удаление ассоциации групп MS Active Directory и ролей RuBackup (DELETE /auth/ldap/assosiations)

```

{
  "ids": [ ❶
    1
  ]
}

```

1 Обязательное поле**204 NO_CONTENT**

Ассоциация группы MS Active Directory и ролей RuBackup удалена.

207 MULTI_STATUS

Часть ассоциаций групп MS Active Directory и ролей RuBackup удалена.

404 NOT_FOUND

Ассоциация группы MS Active Directory и ролей RuBackup не найдена.

8.2. Версия 1 (v1)

Пути методов начинаются с `/api/v1`:

- `/auth/login` → `/api/v1/auth/login`
- `/library_slots/check_ltfs` → `/api/v1/library_slots/check_ltfs`

8.2.1. Авторизация и разрешения

Получение токенов и прохождение авторизации (POST `/auth/login`)

`access_token` действителен 15 минут, `refresh_token` 24 часа, `csrf_token` - до перезагрузки сервиса `rest-api`

До 2.3 включительно

```
{
  "rubakup_database": "rubakup", ①
  "rubakup_user": "rubakup", ①
  "password": "12345" ①
}
```

1 Обязательное поле

Начиная с 2.4

```
{
  "rubakup_database": "rubakup",
  "rubakup_user": "rubakup",
  "password": "12345",
  "auth_type": "database", ①
  "domain": "rubakup" ②
}
```

1 `auth_type` — тип аутентификации (база данных или контроллер домена)

2 `domain` — домен для авторизации через контроллер домена
200 OK

Успешная авторизация, получены токены

Деактивация имеющихся токенов (DELETE `/auth/logout`)

После выполнения запроса access и refresh токены более не действуют.

200 OK

Успешная деактивация токенов

401 UNAUTHORIZED

Verification failed

Обновление access токена после его истечения (POST `/auth/refresh`)

Срок действия токена доступа — 15 минут.

200 OK

Успешное обновление токена

401 UNAUTHORIZED

Один или оба токена не указаны

Получение списка прав пользователя (GET `/auth/permissions`)

200 OK

Успешно получен список прав текущего пользователя

8.2.2. Пользователи

Администраторы

Получение списка администраторов (GET `/administrators`)

200 OK

Получен список администраторов

Удаление списка ролей администратор (DELETE `/administrators`)

```
{
  "ids": [ ①
    1
  ]
}
```

① Обязательное поле

200 OK

Роль администратора удалена

404 NOT_FOUND

Роль администратора не найдена

Назначение пользователю роли администратора (POST /administrators)

```
{
  "data": {
    "client_group": "Some group", ❶
    "administrator": "username" ❶
  }
}
```

❶ Обязательное поле

201 CREATED

Роль администратора добавлена пользователю

Получение информации об одном администраторе (GET /administrators/{id})

Тела запроса нет. В query-параметрах передается `id` администратора, информацию о котором необходимо получить

200 OK

Получена информация об администраторе

Удаление роли администратора. (DELETE /administrators/{id})

Тела запроса нет. В query-параметрах передается `id` администратора, которого необходимо удалить

200 OK

Роль администратора удалена

404 NOT_FOUND

Роль администратора не найдена

Супервайзеры**Удаление списка пользователей роли супервайзера (DELETE /supervisors)**

```
{
  "ids": [ ❶
    1
  ]
}
```

```
]
}
```

① Обязательное поле

200 OK

Роль супервайзера удалена

404 NOT_FOUND

Роль супервайзера не найдена

Получение списка супервайзеров (GET /supervisors)

200 OK

Получен список супервайзеров

Назначение пользователю роли супервайзера (POST /supervisors)

```
{
  "data": {
    "supervisor": "Nobody" ①
  }
}
```

① Обязательное поле

201 CREATED

Роль супервайзера добавлена пользователю

Удаление пользователю роли супервайзера (DELETE /supervisors/{id})

Тела запроса нет. В query-параметрах передается `id` супервайзера, информацию о котором необходимо получить

200 OK

Получена информация об супервайзере

Получение информации о супервайзере (GET /supervisors/{id})

Тела запроса нет. В query-параметрах передается `id` супервайзера, которого необходимо удалить

200 OK

Роль супервайзера удалена

404 NOT_FOUND

Роль супервайзера не найдена

Сопровождающие

Удаление списка пользователей роли сопровождающего (DELETE /maintainers)

```
{
  "ids": [ ❶
  ]
}
```

❶ Обязательное поле

200 OK

Роль сопровождающего удалена

404 NOT_FOUND

Роль сопровождающего не найдена

Получение списка сопровождающих (GET /maintainers)

200 OK

Получен список сопровождающих

Назначение пользователю роли сопровождающего (POST /maintainers)

```
{
  "data": {
    "media_server": "hostname", ❶
    "maintainer": "username" ❶
  }
}
```

❶ Обязательное поле

201 CREATED

Роль администратора добавлена пользователю

Удаление пользователю роли сопровождающего (DELETE /maintainers/{id})

Тела запроса нет. В query-параметрах передается `id` сопровождающего, информацию о котором необходимо удалить

200 OK

Роль сопровождающего удалена

404 NOT_FOUND

Роль сопровождающего не найдена

Получение информации о сопровождающем (GET /maintainers/{id})

Тела запроса нет. В query-параметрах передается `id` сопровождающего, информацию о котором необходимо получить

200 OK

Получена информация о сопровождающем

Аудиторы

Назначение пользователю роли аудитора (POST /auditors)

```
{
  "data": {
    "auditor": "username" 1
  }
}
```

1 Обязательное поле

201 CREATED

роль аудитора добавлена пользователю

Получение информации об аудиторе (GET /auditors/{id})

200 OK

получена информация об аудиторе

Удаление аудитора (DELETE /auditors/{id})

Тела запроса нет. В query-параметрах передается `id` аудитора, которого необходимо удалить

200 OK

роль аудитора удалена

404 NOT_FOUND

роль аудитора не найдена

Получение списка аудиторов (GET /auditors)

200 OK

получен список аудиторов

Удаление списка аудиторов (DELETE /auditors)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

200 OK

роль аудитора удалена

404 NOT_FOUND

роль аудитора не найдена

Пользователи**Удаление списка пользователей (DELETE /users)**

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

200 OK

Список пользователей удален

404 NOT_FOUND

Пользователь не найден

Получение списка пользователей (GET /users)

200 OK

Получен список пользователей

Добавление пользователя (POST /users)

Также в query-параметрах передаются:

- роль пользователя (администратор, супервайзер, сопровождающий)
- имя группы клиентов (для администраторов)
- хостнейм медиасервера (для сопровождающих)

```
{
  "data": {
    "group_id": "Nobody", ①
    "fullname": "",
    "address": "",
    "tel": "",
    "office": "",
    "email": "", ①
    "username": "", ①
    "user_password": "" ①
  }
}
```

① Обязательное поле

201 CREATED

Пользователь создан

Изменение пароля пользователя (PATCH /users/change_password)

```
{
  "new_password": "" ①
}
```

① Обязательное поле

200 OK

Пароль пользователя изменен

404 NOT_FOUND

Пользователь не найден

Редактирование пользователей (PATCH /users/{id})

Также в query-параметрах передается `id` пользователя, информацию о которой необходимо изменить

```
{
  "data": {
    "group_id": "Nobody",
    "fullname": "",
    "address": "",
    "tel": "",
    "office": "",

```

```

    "email": "",
    "new_password": ""
  }
}

```

200 OK

Параметры пользователя изменены

400 BAD_REQUEST

Некорректное тело запроса

404 NOT_FOUND

Пользователь не найден

Удаление пользователя (DELETE /users/{id})

Тела запроса нет. В query-параметрах передается `id` пользователя, которого необходимо удалить

200 OK

Пользователь удален

404 NOT_FOUND

Пользователь не найден

Получение информации о пользователе (GET /users/{id})

Тела запроса нет. В query-параметрах передается `id` пользователя, информацию о котором необходимо получить

200 OK

Информация о пользователе получена

404 NOT_FOUND

Пользователь не найден

Получение списка прав пользователя (GET /users/permissions)**200 OK**

Получен список прав пользователей СРК с разными ролями

8.2.3. Сбор сведений**Лицензии**

Получение списка типов лицензий (GET /license_types)**200 OK**

Получен список типов лицензий

Получение информации о типе лицензии (GET /license_types/{id})

Тела запроса нет. В query-параметрах передается `id` типа лицензии, информацию о которой необходимо получить

200 OK

Получена информация о типе лицензии

404 NOT_FOUND

Тип лицензии не найден

Информация о сервере**Получение информации о сервере (GET /server_hello)****200 OK**

Получена информация о версии rubackup сервера

Сведения о состоянии приложения**Получение информации о состоянии приложения (GET /server_state)**

Возвращает список неавторизованных клиентов и медиасерверов, просроченных правил стратегий и глобального расписания, клиентских запросов на добавление и удаление правил

200 OK

Получена информация о текущем состоянии приложения

Типы резервного копирования**Получение списка доступных типов резервного копирования (GET /backup_types)**

Тела запроса нет.

200 OK

Получен список типов резервного копирования

Типы сжатия**Получение списка доступных типов сжатия (GET /compression_type)****200 OK**

Получен список типов сжатия

Алгоритмы защиты данных

Получение списка доступных криптоалгоритмов (GET /crypto)

200 OK

Получен список доступных криптоалгоритмов

Алгоритмы дедупликации

Получение списка доступных алгоритмов хеш-функций (GET /deduplication_hash_algorithm)

OK 200 - получен список алгоритмов хеш-функций

Поиск медиасерверов

Поиск клиента медиасервера по HWID сервера (GET /servers_hw_id_tmp)

200 OK

Получен список медиасерверов с информацией о них, их HWID и лицензии

Получение информации о сервере и лицензии (GET /servers_hw_id_tmp/{id})

Тела запроса нет. В query-параметрах передается id медиасервера, информацию о котором необходимо получить

200 OK

Получена информация о сервере и лицензии

404 NOT_FOUND

Информация о медиасервере и лицензии не найдена

Доступные типы ресурсов

Получение списка доступных типов ресурсов (GET /resource_types)

200 OK

Получен список типов ресурсов

Получение информации о типе ресурса (GET /resource_types/{id})

Тела запроса нет. В query-параметрах передается id типа ресурса, информацию о котором необходимо получить

200 OK

Получена информация о типе ресурса

404 NOT_FOUND

Тип ресурса не найден

Доступные операционные системы

Получение списка доступных операционных систем (GET `/os_types`)

200 OK

Получен список доступных типов операционных систем

Статусы верификации

Получение списка доступных статусов верификации (GET `/verification_status`)

200 OK

Получен список доступных статусов верификации

Статусы правил

Получение списка доступных статусов правил (GET `/status_of_rule`)

200 OK

Получен список существующих статусов правил

Ресурсы модулей

Получение списка доступных ресурсов модуля (POST `/module_resource_list`)

Например, для PostgreSQL Universal это версия PostgreSQL (например, PostgreSQL 16.3)

```
{
  "data": {
    "client_hwid": "525a99154f3505a2", ①
    "resource_type": "LVM logical volume", ①
    "secret_method": 0
  }
}
```

① Обязательное поле

200 OK

Получен список доступных ресурсов модуля

Метрики (статистика)

Получение статистики по уведомлениям (GET `/metrics/notifications`)

200 OK

Получена статистика уведомлений

Получение статистики по объектам репозитория (GET `/metrics/repository`)

200 OK

Получена статистика репозитория

Получение статистики по очереди задач облаков (GET `/metrics/s3_cloud_task_queue`)

200 OK

Получена статистика очереди задач облаков

Получение статистики по очереди задач (GET `/metrics/task_queue`)

200 OK

Получена статистика очереди задач

Получение статистики по очереди задач ленточных библиотек (GET `/metrics/tl_task_queue`)

200 OK

Получена статистика очереди задач ленточных библиотек

8.2.4. Глобальная конфигурация

Редактирование параметров глобальной конфигурации (PATCH `/global_configuration`)

```
{
  "data": {
    "file_transfer_block_size": 16384,
    "data_spred_into_pool": "sequentially",
    "emergency_storage_local_catalog":
"/tmp/rubackup_emergency_storage_local_catalog",
    "done_tasks_remove_period": 1440,
    "error_tasks_remove_period": 10080,
    "broken_tasks_remove_period": 10080,
    "killed_tasks_remove_period": 10080,
    "obsoleted_backup_notification_period": 1440,
    "rule_validity_end_notification_period": 1440,
    "strategy_validity_end_notification_period": 1440,
    "rule_storage_capacity_reserve": 0,
    "strategy_storage_capacity_reserve": 0,
    "rule_storage_capacity_notification_period": 1440,
    "strategy_storage_capacity_notification_period": 1440,
    "service_window_start": "00:00:00",
    "service_window_end": "23:59:59",
    "pool_storage_capacity_reserve": 2000000000,
    "tape_library_mount_point": "/opt/rubackup/mnt",
    "suspended_task_restart_period": 1,
    "unload_tape_cartridges_whes_media_server_starts": "yes",
```

```

"ltfs_umount_timeout": 50,
"max_system_monitoring_records": 3600,
"monitoring_period": 1,
"digital_signature_public_key_obsolescence": 1440,
"rb_key_hash": "",
"verify_archive_after_creation": "no",
"kill_task_of_offline_client": "yes",
"create_new_task_if_client_offline": "create new task",
"auto_delete_archive_from_broken_chain": "no",
"consolidating_tape_library_tasks": "yes",
"clean_tape_drives_period": 30,
"auto_clean_tape_drives": "no",
"dedup_clean_unused_blocks": "yes",
"dedup_clean_period": 30,
"dedup_provide_common_hash_table_for_client": "yes",
"dedup_provide_common_hash_table": "yes",
"dedup_verify_only_meta_data": "yes",
"mandatory_storage_time": 0,
"clients_capacity_limits": false,
"global_schedule_capacity_limits": false,
"backup_strategies_capacity_limits": false,
"immutable_archives": true,
"delete_archives_with_zeroing": true,
"dont_delete_last_gs_rule_archive": true,
"dont_delete_last_strategy_rule_archive": true,
"wrong_auth_attempts_count": 5,
"wrong_auth_block_period": 30,
"repository_remove_chain": "yes",
"filesystems_clean_period": 30,
"show_previous_login_attempts_after_logging_in": "yes",
"search_backup_in_cluster_group": true,
"bandwidth_limit_advantage": "client",
"bandwidth_client_limit": "minimum",
"bandwidth_rule_limit": "minimum"
}
}

```

200 OK

Параметры глобальной конфигурации изменены

400 BAD_REQUEST

Некорректное тело запроса

Получение параметров глобальной конфигурации (GET /global_configuration)**200 OK**

Параметры глобальной конфигурации получены

Включение и выключение сервисного режима (PATCH /global_configuration/service_mode)

```
{  
  "service_mode": false ①  
}
```

① Обязательное поле**200 OK**

Статус сервисного режима изменен

400 BAD_REQUEST

Некорректное тело запроса

8.2.5. Клиенты**Клиенты****Получение списка клиентов (GET /clients)****200 OK**

Получен список авторизованных клиентов

Удаление нескольких клиентов (DELETE /clients)

```
{  
  "ids": [ ①  
    1  
  ]  
}
```

① Обязательное поле**200 OK**

Список авторизованных клиентов удален

404 NOT_FOUND

Авторизованный клиент не найден

Авторизация клиентов (POST /clients/authorize)

```
{
  "data": {
    "clients_ids": [ ❶
      1
    ]
  }
}
```

❶ Обязательное поле

201 CREATED

Клиент авторизован

Получение дерева клиентов (POST /clients/tree)

```
{
  "data": {
    "clients": [
      {
        "client_group_id": 1, ❶
        "clients_limit": 20,
        "clients_page": 1 ❶
      }
    ],
    "client_groups_page": 1,
    "client_groups_limit": 20,
    "filter_clients_hostname": "rubackup-client"
  }
}
```

❶ Обязательное поле

200 OK

Получено дерево клиентов

404 NOT_FOUND

Авторизованный клиент не найден

Редактирование существующего клиента (PATCH /clients/{id})

Также в query-параметрах передается `id` клиента, информацию о котором необходимо изменить

```
{
  "data": {
    "group_id": "No group",
    "lsf_flag": false,
    "ls_pool": "Default",
    "lrl_flag": false,
    "storage_capacity": 10,
    "description": "Client description",
    "centralized_restore": false,
    "client_side_backup": false,
    "client_side_restore": false
  }
}
```

200 OK

Параметры клиента изменены

400 BAD_REQUEST

Некорректное тело запроса

404 NOT_FOUND

Авторизованный клиент не найден

Получение клиента (GET /clients/{id})

Тела запроса нет. В query-параметрах передается **id** клиента, информацию о котором необходимо получить

200 OK

Получена информация о клиенте

Удаление клиента (DELETE /clients/{id})

Тела запроса нет. В query-параметрах передается **id** клиента, информацию о котором необходимо получить

200 OK

Клиент удален (помещен в неавторизованные)

404 NOT_FOUND

Авторизованный клиент не найден

Получение доступных типов ресурсов для создания копии (GET /clients/{id}/resource_type)

Тела запроса нет. В query-параметрах передается **id** клиента, информацию о типах ресурсов которого необходимо получить

200 OK

Получена информация о типах ресурса для клиента

Группы клиентов

Получение списка групп клиентов (GET /client_groups)

200 OK

Получена информация о группах клиентов

Удаление списка групп клиентов (DELETE /client_groups)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

200 OK

Список групп клиентов удален

404 NOT_FOUND

Группа клиентов не найдена

Добавление групп клиентов (POST /client_groups)

```
{
  "data": {
    "name": "group_name", ❶
    "shared": false,
    "clustered": false,
    "description": "some description",
    "launch_attempts_limit": 1
  }
}
```

❶ Обязательное поле

201 CREATED

Группа клиентов создана

Изменение группы клиентов (PATCH /client_groups/{id})

Также в query-параметрах передается id группы клиентов, информацию о кото-

рой необходимо изменить

```
{
  "data": {
    "name": "group_name",
    "shared": false,
    "clustered": false,
    "description": "some description",
    "launch_attempts_limit": 1
  }
}
```

200 OK

Параметры группы клиентов изменены

400 BAD_REQUEST

Некорректное тело запроса

404 NOT_FOUND

Группа клиентов не найдена

Получение информации о группе клиентов (GET /client_groups/{id})

Тела запроса нет. В query-параметрах передается `id` группы клиентов, информацию о которой необходимо получить

200 OK

Получена информация о группе клиентов

Удаление группы клиентов (DELETE /client_groups/{id})

Тела запроса нет. В query-параметрах передается `id` группы клиентов, которую необходимо удалить

200 OK

Группа клиентов удалена

404 NOT_FOUND

Группа клиентов не найдена

Ограничения пропускной способности клиента

Получение списка ограничений пропускной способности клиента (GET /client_bandwidth)

200 OK

Получен список ограничений пропускной способности

Удаление списка ограничений пропускной способности клиента (DELETE /client_bandwidth)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

200 OK

Список ограничений пропускной способности удален

404 NOT_FOUND

Ограничение пропускной способности не найдено

Создание ограничения пропускной способности клиента (POST /client_bandwidth)

```
{
  "data": {
    "client": "rubackup-client (525a99154f3505a2)", ❶
    "backup_bandwidth": 0,
    "restore_bandwidth": 0,
    "window_start": "14:33:31",
    "window_end": "14:33:31"
  }
}
```

❶ Обязательное поле

201 CREATED

Ограничение пропускной способности создано

Изменение ограничения пропускной способности клиента (PATCH /client_bandwidth/{id})

Также в query-параметрах передается `id` ограничения, информацию о котором необходимо изменить

```
{
  "data": {
    "client": "rubackup-client (525a99154f3505a2)",
    "backup_bandwidth": 0,
    "restore_bandwidth": 0,
    "window_start": "14:33:31",
    "window_end": "14:33:31",
  }
}
```

```

    "id": 1,
    "client_hwid": "525a99154f3505a2"
  }
}

```

200 OK

Параметры ограничения пропускной способности изменены

400 BAD_REQUEST

Некорректное тело запроса

404 NOT_FOUND

Ограничение пропускной способности не найдено

Получение информации об ограничении пропускной способности клиента (GET /client_bandwidth/{id})

Тела запроса нет. В query-параметрах передается `id` ограничения, информацию о котором необходимо получить

200 OK

Получена информация об ограничении пропускной способности

Удаление ограничения пропускной способности клиента (DELETE /client_bandwidth/{id})

Тела запроса нет. В query-параметрах передается `id` ограничения, которое необходимо удалить

200 OK

Ограничение пропускной способности удалено

404 NOT_FOUND

Ограничение пропускной способности не найдено

Неавторизованные клиенты

Удаление списка неавторизованных клиентов (DELETE /unauthorised_clients)

```

{
  "ids": [ ❶
    1
  ]
}

```

❶ Обязательное поле

200 OK

Список неавторизованных клиентов удален

404 NOT_FOUND

Неавторизованные клиенты не найдены

Получение списка неавторизованных клиентов (GET /unauthorised_clients)

200 OK

Получен список неавторизованных клиентов

Удаление неавторизованного клиента (DELETE /unauthorised_clients/{id})

Тело запроса отсутствует. В query-параметрах передается `id` неавторизованного клиента, которого необходимо удалить

200 OK

Неавторизованный клиент удален

404 NOT_FOUND

Неавторизованный клиент не найден

Получение информации о неавторизованном клиенте (GET /unauthorised_clients/{id})

Тело запроса отсутствует. В query-параметрах передается `id` неавторизованного клиента, информацию о котором необходимо получить

200 OK

Получена информация о неавторизованном клиенте

8.2.6. Хранилища

Пулы и их группы

Доступные типы пулов

Получение списка доступных типов пулов (GET /destination_storage_types)

200 OK

Получен список доступных типов пулов

Пулы

Удаление списка пулов (DELETE /pool_list)

```
{
  "ids": [ 1 ]
}
```

```
]
}
```

① Обязательное поле

200 OK

Список пулов удален

404 NOT_FOUND

Пул не найден

Получение списка пулов (GET /pool_list)

200 OK

Получен список существующих пулов

Добавление пула (POST /pool_list)

Набор параметров, которые необходимо передать, зависит от типа пула. В случае, если переданы лишние параметры (которые не подходят создаваемому типу пула), то такие параметры игнорируются.

Параметр `compression_type` не нужно указывать для пулов типа Block device и Client defined.

Параметры `block_size`, `hash_length`, `algorithm` указываются только для пулов типа Block device.

```
{
  "data": {
    "pool_name": "my_pool_name", ①
    "pool_type": "Block device", ①
    "media_server": "rubackup-server", ①
    "compression_type": "None",
    "description": "",
    "retention_period": "1 year",
    "ignore_free_space_check": false,
    "algorithm": "streebog",
    "block_size": 131072,
    "hash_length": 256,
    "meta_data_pool": "Default"
  }
}
```

① Обязательное поле

201 CREATED

Пул создан

Получение списка доступных для перемещения и копирования пулов (GET /pool_list/available_for_copy_or_move)

200 OK

Получен список доступных для перемещения и копирования пулов

Редактирование пула (PATCH /pool_list/{id})

Также в query-параметрах передается `id` пула, информацию о котором необходимо изменить

```
{
  "data": {
    "pool_name": "my_pool_name",
    "compression_type": "None",
    "description": "",
    "retention_period": "1 year",
    "ignore_free_space_check": false
  }
}
```

200 OK

Параметры пула изменены

400 BAD_REQUEST

Некорректное тело запроса

404 NOT_FOUND

Пул не найден

Удаление пула (DELETE /pool_list/{id})

Тела запроса нет. В query-параметрах передается `id` пула, который необходимо удалить

200 OK

Пул удален

404 NOT_FOUND

Пул не найден

Получение информации о пуле (GET /pool_list/{id})

Тела запроса нет. В query-параметрах передается `id` пула, информацию о котором необходимо получить

200 OK

Получена информация о пуле

Подмена пулов**Удаление списка подмены пулов (DELETE /pool_substitution)**

```
{
  "ids": [ ①
    1
  ]
}
```

① Обязательное поле

200 OK

Список настроенных подмен пулов удален

404 NOT_FOUND

Подмена пулов не найдена

Получение списка подмены пулов (GET /pool_substitution)**200 OK**

Получен список настроенных подмен пулов

Добавление подмены пулов (POST /pool_substitution)

```
{
  "data": {
    "pool": "Default", ①
    "substitution": "Default2" ①
  }
}
```

① Обязательное поле

201 CREATED

Создано правило подмены пулов

Удаление подмены пулов (DELETE /pool_substitution/{id})

Тела запроса нет. В query-параметрах передается `id` правила подмены пулов, которое необходимо удалить

200 OK

Правило подмены пулов удалено

404 NOT_FOUND

Подмена пулов не найдена

Получение информации о подмене пулов (GET /pool_substitution/{id})

Тела запроса нет. В query-параметрах передается `id` правила подмены пулов, информацию о котором необходимо получить

200 OK

Получена информация о правиле подмены пулов

Динамические группы пулов**Получение списка динамических групп пулов (GET /dynamic_pool_groups)****200 OK**

Получен список групп пулов

Удаление списка динамических групп пулов (DELETE /dynamic_pool_groups)

```
{
  "ids": [ ①
    1
  ]
}
```

① Обязательное поле

200 OK

Группы пулов удалены

404 NOT_FOUND

Группа пулов не найдена

Добавление динамической группы пулов (POST /dynamic_pool_groups)

```
{
  "data": {
    "name": "", ①
  }
}
```

```

    "max_pool_tasks": 10,
    "max_media_server_tasks": 10,
    "max_cpu_usage": 80,
    "max_in_block_operations": 0,
    "max_out_block_operations": 0,
    "calculation_period": 10,
    "description": ""
  }
}

```

1 Обязательное поле

201 CREATED

Группа пулов создана

Изменение динамической группы пулов (PATCH `/dynamic_pool_groups/{id}`)

Также в query-параметрах передается `id` группы пулов, информацию о которой необходимо изменить

```

{
  "data": {
    "name": "",
    "max_pool_tasks": 10,
    "max_media_server_tasks": 10,
    "max_cpu_usage": 80,
    "max_in_block_operations": 0,
    "max_out_block_operations": 0,
    "calculation_period": 10,
    "description": ""
  }
}

```

200 OK

Группа пулов изменена

400 BAD_REQUEST

Некорректное тело запроса

404 NOT_FOUND

Группа пулов не найдена

Получение информации о динамической группе пулов (GET `/dynamic_pool_groups/{id}`)

Тела запроса нет. В query-параметрах передается `id` группы пулов, информацию о

которой необходимо получить

200 OK

Получена информация о группе пулов

Удаление динамической группы пулов (DELETE /dynamic_pool_groups/{id})

Тела запроса нет. В query-параметрах передается `id` группы пулов, информацию о которой необходимо удалить

200 OK

Стратегия удалена

404 NOT_FOUND

Группа пулов не найдена

Пулы в динамических группах

Получение списка добавленных в динамическую группу пулов (GET /dynamic_pool_groups_records)

200 OK

Получен список пулов динамической группы

Удаление списка добавленных в динамическую группу пулов (DELETE /dynamic_pool_groups_records)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

200 OK

Список пулов динамической группы удален

404 NOT_FOUND

Пулов динамической группы не найдено

Добавление пула в динамическую группу пулов (POST /dynamic_pool_groups_records)

```
{
  "data": {
    "dynamic_pool_group": 1, ❶
    "pool": "Default", ❶
    "enabled": true
  }
}
```

```
}
}
```

1 Обязательное поле

201 CREATED

Пул добавлен в группу

Получение информации о добавленном в динамическую группу пуле (GET /dynamic_pool_groups_records/{id})

Тела запроса нет. В query-параметрах передается `id` пула из группы пулов, информацию о котором необходимо получить

200 OK

Получена информация о пуле динамической группы

Удаление добавленного в динамическую группу пула (DELETE /dynamic_pool_groups_records/{id})

Тела запроса нет. В query-параметрах передается `id` пула, который необходимо удалить из группы пулов

200 OK

Пул удален из динамической группы

404 NOT_FOUND

Пул в динамической группе не найден

Локальные файловые хранилища

Удаление нескольких локальных файловых хранилищ (DELETE /storage_local_catalogs)

```
{
  "ids": [ 1
  ]
}
```

1 Обязательное поле

200 OK

Список локальных файловых хранилищ удален

404 NOT_FOUND

Локальный файловый хранилище не найдено

Получение списка локальных файловых хранилищ (GET /storage_local_catalogs)**200 OK**

Получен список локальных файловых хранилищ

Добавление локального файлового хранилища (POST /storage_local_catalogs)

```
{
  "data": {
    "path": "", ①
    "pool": "Default",
    "description": ""
  }
}
```

① Обязательное поле

201 CREATED

Локальное файловое хранилище добавлено

Редактирование существующего локального файлового хранилища (PATCH /storage_local_catalogs/{id})

Также в query-параметрах передается `id` локального файлового хранилища, который необходимо изменить

```
{
  "data": {
    "description": ""
  }
}
```

200 OK

Параметры локального файлового хранилища изменены

400 BAD_REQUEST

Некорректное тело запроса

404 NOT_FOUND

Локальное файловое хранилище не найдено

Удаление локального файлового хранилища (DELETE /storage_local_catalogs/{id})

Также в query-параметрах передается `id` локального файлового хранилища, которое необходимо удалить

200 OK

Блочное устройство удалено

404 NOT_FOUND

Локальное файловое хранилище не найдено

Получение информации о локальном файловом хранилище (GET /storage_local_catalogs/{id})

Также в query-параметрах передается `id` локального файлового хранилища, информацию о котором необходимо получить

200 OK

Информация о локальном файловом хранилище получена

404 NOT_FOUND

Локальное файловое хранилище не найдено

Блочные устройства

Удаление списка блочных устройств (DELETE /storage_block_devices)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

200 OK

Список блочных устройств удален

404 NOT_FOUND

Блочное устройство не найдено

Получение списка блочных устройств (GET /storage_block_devices)

200 OK

Получен список блочных устройств

Добавление блочного устройства (POST /storage_block_devices)

```
{
  "data": {
    "device": "", ❶
    "pool": "Default", ❶
  }
}
```

```
"overwrite_file_system": false, ❶
"description": ""
}
}
```

❶ Обязательное поле

201 CREATED

Блочное устройство добавлено

Редактирование блочного устройства (PATCH /storage_block_devices/{id})

Также в query-параметрах передается `id` блочного устройства, которое необходимо изменить

```
{
  "data": {
    "description": "",
    "parallel_workers": 2
  }
}
```

200 OK

Параметры блочного устройства изменены

400 BAD_REQUEST

Некорректное тело запроса

404 NOT_FOUND

Блочное устройство не найдено

Удаление блочного устройства (DELETE /storage_block_devices/{id})

Также в query-параметрах передается `id` блочного устройства, которое необходимо удалить

200 OK

Блочное устройство удалено

404 NOT_FOUND

Блочное устройство не найдено

Получение информации о блочном устройстве (GET /storage_block_devices/{id})

Также в query-параметрах передается `id` блочного устройства, информацию о котором необходимо получить+

200 OK

Информация о блочном устройстве получена

404 NOT_FOUND

Блочное устройство не найдено

Ленточные хранилища**Ленточные библиотеки****Удаление списка ленточных библиотек (DELETE /tape_libraries)**

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

200 OK

Список ленточных библиотек удален

404 NOT_FOUND

Ленточная библиотека не найдена

Получение списка ленточных библиотек (GET /tape_libraries)**200 OK**

Получен список ленточных библиотек

Добавление ленточной библиотеки (POST /tape_libraries)

```
{
  "data": {
    "pool": "tape_pool", ❶
    "media_server": "rubackup-media", ❶
    "description": "",
    "medium_changer_candidate_id": 1, ❶
    "tape_drives_candidates_ids": [ ❶
      1
    ]
  }
}
```

1 Обязательное поле

201 CREATED

Ленточная библиотека добавлена

Удаление ленточной библиотеки (DELETE /tape_libraries/{id})

Тела запроса нет. В query-параметрах передается `id` ленточной библиотеки, информацию о которой необходимо получить

200 OK

Ленточная библиотека удалена

404 NOT_FOUND

Ленточная библиотека не найдена

Получение информации о ленточной библиотеке (GET /tape_libraries/{id})

200 OK

Информация о ленточной библиотеке получена

404 NOT_FOUND

Ленточная библиотека не найдена

Получение списка доступных для добавления роботов и приводов (GET /tape_libraries/candidates)

Тела запроса нет. В query-параметрах передается имя медиасервера, которому презентована библиотека, а также способ добавления

200 OK

Получен список доступных для добавления роботов и приводов

Получение информации о роботе и слотах (POST /tape_libraries/candidates/info)

```
{
  "data": {
    "media_server": "rubackup-media-server", ①
    "medium_changer_sg_path": "/dev/sg4"
  }
}
```

200 OK

Получена информация о роботе и слотах

Получение дерева ленточных библиотек (POST /tape_libraries/tree)

В списке отображаются: 1. Ленточные библиотеки, их роботы, их приводы. Список слотов должен быть пустым всегда. 2. Ленточные пулы, ассоциированные и неассоциированные с библиотекой

```
{
  "data": {
    "pools": {
      "limit": 50,
      "page": 1
    }
  }
}
```

200 OK

Получено дерево информация о роботе и слотах

Синхронизация данных ленточной библиотеки (POST /tape_libraries/{id}/sync)

Тела запроса нет. В query-параметрах передается `id` библиотеки, информацию о которой необходимо синхронизировать

200 OK

Синхронизация библиотеки прошла успешно

422 Unprocessable Entity - синхронизация завершена с ошибкой, библиотечка содержит незарегистрированный картридж

Типы картриджей**Получение списка доступных типов картриджей (GET /tape_types)****200 OK**

Получен список доступных типов картриджей

Картриджи**Удаление списка картриджей (DELETE /tape_cartridges)**

```
{
  "ids": [
    1
  ]
}
```

200 OK

Список картриджей удален

404 NOT_FOUND

Картридж не найден

Получение списка доступных картриджей (GET /tape_cartridges)**200 OK**

Получен список доступных картриджей

Добавление ленточного картриджа (POST /tape_cartridges)

```
{
  "data": {
    "type": "LT0-9", ①
    "pool": "tape_pool", ①
    "volume_tag": "RB10001L9", ①
    "description": ""
  }
}
```

① Обязательное поле

201 CREATED

Ленточный картридж добавлен

Редактирование картриджа (PATCH /tape_cartridges/{id})

Также в query-параметрах передается `id` ленточного картриджа, информацию о которой необходимо изменить

```
{
  "data": {
    "volume_tag": "",
    "description": ""
  }
}
```

200 OK

Параметры картриджа изменены

400 BAD_REQUEST

Некорректное тело запроса

404 NOT_FOUND

Картридж не найден

Получение информации о ленточном картридже (GET /tape_cartridges/{id})

Тела запроса нет. В query-параметрах передается `id` картриджа, информацию о котором необходимо получить

200 OK

Получена информация о ленточном картридже

Операции с ленточными приводами

Получение списка ленточных приводов (GET /library_tape_drives)

200 OK

Получен список приводов ленточной библиотеки

Получение информации о ленточном приводе (GET /library_tape_drives/{id})

Тела запроса нет. В query-параметрах передается `id` привода, информацию о которой необходимо получить

200 OK

Получена информация о приводе ленточной библиотеки

Добавление привода (POST /library_tape_drives)

```
{
  "data": {
    "ids": [ ①
      4,
      6,
      9
    ],
    "tape_library_id": 5 ①
  }
}
```

① Обязательное поле

201 CREATED

Новый привод добавлен к библиотеке

Удаление списка ленточных приводов (DELETE /library_tape_drives)

```
{
```

```

"ids": [ ①
  2,
  6,
  9
]
}

```

① Обязательное поле

200 OK

Привод ленточной библиотеки удален

Получение списка ленточных приводов, доступных для добавления (GET /library_tape_drives/candidates)

Тела запроса нет. В query-параметрах передается способ добавления привода (ручной или автоматический), hostname медиасервера, на который презентована библиотека и sg_device_path робота библиотеки

200 OK

Информация о доступных для добавления приводах получена

Редактирование ленточного привода (PATCH /library_tape_drives/{id})

Также в query-параметрах передается id привода, информацию о котором необходимо изменить

```

{
  "data": {
    "scsi_path": "[6:0:0:0]",
    "device": "/dev/sg1",
    "sys_transfer_element": 3
  }
}

```

200 OK

Параметры привода изменены

400 BAD_REQUEST

Некорректное тело запроса

Очистка ленточного привода (PATCH /library_tape_drives/{id}/clean)

Тела запроса нет. В query-параметрах передается id привода, который необходимо очистить

200 OK

Задача на очистку привода создана

Картриджи**Получение списка слотов (GET /library_slots)****200 OK**

Получен список слотов ленточной библиотеки

Получение информации о слоте (GET /library_slots/{id})

Тела запроса нет. В query-параметрах передается `id` слота, информацию о которой необходимо получить

200 OK

Получена информация о слоте ленточной библиотеки

Проверка ltfs на картридже (PATCH /library_slots/check_ltfs)

```
{
  "data": [
    {
      "library_id": 5, ❶
      "volume_tag": "ACS512L9" ❶
    }
  ]
}
```

❶ Обязательное поле

200 OK

Создана задача на проверку наличия ltfs на картридже

400 BAD_REQUEST

Некорректное тело запроса

Стирание картриджа (PATCH /library_slots/erase_cartridge)

```
{
  "data": [
    {
      "library_id": 5, ❶
      "is_long_erase": false,
      "volume_tag": "ACS512L9" ❶
    }
  ]
}
```

```
]
}
```

① Обязательное поле

200 OK

Создана задача на стирание картриджа

400 BAD_REQUEST

Некорректное тело запроса

Экспорт картриджа (PATCH /library_slots/export_cartridge)

```
{
  "data": {
    "library_id": 5, ①
    "volume_tag": "ACS512L9" ①
  }
}
```

① Обязательное поле

200 OK

Создана задача на экспорт картриджа

400 BAD_REQUEST

Некорректное тело запроса

Импорт картриджа (PATCH /library_slots/import_cartridge)

```
{
  "data": {
    "library_id": 5, ①
    "pool_name": "Tape libraries" ①
  }
}
```

① Обязательное поле

200 OK

Импорт картриджа прошел успешно

400 BAD_REQUEST

Некорректное тело запроса

Форматирование картриджа (PATCH /library_slots/format_cartridge)

```
{
  "data": [
    {
      "library_id": 5, ❶
      "volume_tag": "ACS512L9" ❶
    }
  ]
}
```

❶ Обязательное поле

200 OK

Создана задача на форматирование картриджа

400 BAD_REQUEST

Некорректное тело запроса

Перемещение картриджа в другой слот (PATCH /library_slots/move_cartridge)

```
{
  "data": {
    "library_id": 5, ❶
    "volume_tag": "ACS512L9", ❶
    "slot_id": 6 ❶
  }
}
```

❶ Обязательное поле

200 OK

Создана задача на перемещение картриджа

400 BAD_REQUEST

Некорректное тело запроса

Роботы**Получение списка роботов (GET /medium_changers)****200 OK**

Получен список роботов

Получение информации о роботе (GET /medium_changers/{id})

Тела запроса нет. В query-параметрах передается `id` робота, информацию о которой необходимо получить

200 OK

Получена информация о роботе

404 NOT_FOUND

Робот не найден

Изменение робота (PATCH /medium_changers/{id})

Также в query-параметрах передается `id` робота, информацию о котором необходимо изменить

```
{
  "data": {
    "scsi_path": "[8:0:0:0]",
    "device": "/dev/sg2"
  }
}
```

200 OK

Параметры робота изменены

400 BAD_REQUEST

Некорректное тело запроса

404 NOT_FOUND

Робот не найдена

Облачные хранилища**Облачное хранилище****Удаление списка облаков (DELETE /s3_clouds)****204 NO_CONTENT**

удален список облаков

Получение списка доступных облаков (GET /s3_clouds)**200 OK**

получен список облаков

Добавление нового облака (POST /s3_clouds)

```
{
  "data": {
    "name": "my_s3_cloud", ①
    "pool": "cloud_pool", ①
    "access_key_id": "", ①
    "secret_key_access": "", ①
    "region": "ru-1",
    "endpoint_override": "https://192.168.222.197:9000",
    "proxy_host": "",
    "proxy_port": "",
    "proxy_username": "",
    "proxy_password": "",
    "bucket_name": "rubackup-server-5953bb5c71fc3bf8",
    "allow_auto_bucket_creation": true,
    "transfer_workers_number": 1
  }
}
```

① Обязательное поле

201 CREATED

облако создано

Получение списка доступных бакетов не добавленного в СРК облака (POST /s3_clouds/buckets)

```
{
  "data": {
    "access_key_id": "jkepw6ubWUJpS04vtmgm", ①
    "secret_key_access": "R0hj0Gn7MjJ3iTvBBxw9dI1ez9jPd1wDkMuIQ4FU", ①
    "region": "ru-1", ①
    "endpoint_override": "https://192.168.222.197:9000", ①
    "proxy_host": "",
    "proxy_port": "",
    "proxy_username": "",
    "proxy_password": ""
  }
}
```

① Обязательное поле

201 CREATED

облако создано

Удаление облака DELETE /s3_clouds/{id}

Тела запроса нет. В query-параметрах передается `id` облака, которое необходимо удалить

204 NO_CONTENT

удалено облако

Получение информации о существующем облаке (GET /s3_clouds/{id})

Тела запроса нет. В query-параметрах передается `id` облака, информацию о котором необходимо получить

200 OK

получена информация об облаке

404 NOT_FOUND

облако не найдено

Редактирование параметров облака (PATCH /s3_clouds/{id})

В query-параметрах передается `id` облака, информацию о котором необходимо изменить

```
{
  "data": {
    "allow_auto_bucket_creation": false,
    "bucket_name": ""
  }
}
```

200 OK

параметры облака изменены

404 NOT_FOUND

облако не найдено

Получение списка доступных бакетов существующего облака (GET /s3_clouds/{id}/buckets)

В query-параметрах передается `id` облака, информацию о доступных бакетах которого необходимо получить

200 OK

получена информация о доступных бакетах существующего облака

Статус задач облачного хранилища**Получение списка доступных статусов задач на облачные хранилища (GET /s3_clouds_task_status)****200 OK**

получен список статусов очереди облаков

Задачи облачного хранилища**Получение задач очереди облаков (GET /s3_clouds_task_queue)****200 OK**

Получен список задач в очереди задач облаков

Получение информации о задаче из очереди облаков (GET /s3_clouds_task_queue/{id})**200 OK**

Получена информация о задаче в очереди задач облаков

404 NOT_FOUND

Задача не найдена

Удаление списка задач из очереди облаков (DELETE /s3_clouds_task_queue)

```
{
  "ids": [ ①
    1
  ]
}
```

① Обязательное поле**204 NO_CONTENT**

Удален список задач из очереди задач облаков

Удаление ошибочных задач из очереди облаков (DELETE /s3_clouds_task_queue/errors_deletion)**204 NO_CONTENT**

Удален список ошибочных задач из очереди задач облаков

Удаление задачи из очереди облаков (DELETE /s3_clouds_task_queue/{id})**204 NO_CONTENT**

Удалена задача из очереди задач облаков

404 NOT_FOUND

Задача не найдена

Клиентские хранилища

Получение списка клиентских хранилищ (GET /client_defined_storages)

200 OK

Получен список клиентских хранилищ

Добавление клиентского хранилища (POST /client_defined_storages)

```
{
  "data": {
    "pool": "Default", ①
    "metadata_fs_pool": "Default2", ①
    "name": "", ①
    "description": ""
  }
}
```

① Обязательное поле

201 CREATED

Клиентское хранилище создано

Удаление списка клиентских хранилищ (DELETE /client_defined_storages)

```
{
  "ids": [ ①
    1
  ]
}
```

① Обязательное поле

200 OK

Список клиентских хранилищ удален

404 NOT_FOUND

Клиентское хранилище не найдено

Изменение клиентского хранилища (PATCH /client_defined_storages/{id})

Также в query-параметрах передается `id` клиентского хранилища, информацию о котором необходимо изменить

```
{
  "data": {
```

```

    "name": "",
    "metadata_fs_pool": "Default2",
    "description": ""
  }
}

```

200 OK

Параметры клиентского хранилища изменены

400 BAD_REQUEST

Некорректное тело запроса

404 NOT_FOUND

Клиентское хранилище не найдено

Получение информации о клиентском хранилище (GET /client_defined_storages/{id})

Тела запроса нет. В query-параметрах передается `id` клиентского хранилища, информацию о котором необходимо получить

200 OK

Получена информация об клиентском хранилище

404 NOT_FOUND

Клиентское хранилище не найдено

Удаление клиентского хранилища (DELETE /client_defined_storages/{id})

Тела запроса нет. В query-параметрах передается `id` клиентского хранилища, информацию о котором необходимо удалить

200 OK

Клиентское хранилище удалено

404 NOT_FOUND

Клиентское хранилище не найдено

Клиентские блочные устройства**Получение списка доступных блочных устройств указанного клиента (POST /list_client_block_devices)**

```

{
  "data": {
    "client_hwid": "525a99154f3505a2", ①
    "path": "/dev/" ①
  }
}

```

```
}
```

① Обязательное поле

200 OK

Получен список доступных блочных устройств клиента

Клиентские файловые системы

Получение списка доступных директорий указанного клиента (POST `/list_client_filesystem`)

```
{
  "data": {
    "client_hwid": "525a99154f3505a2", ①
    "path": "/" ①
  }
}
```

① Обязательное поле

200 OK

Получен список доступных директорий клиента

Медиасерверы

Удаление нескольких медиасерверов (DELETE `/media_servers`)

```
{
  "ids": [ ①
    1
  ]
}
```

① Обязательное поле

200 OK

Список медиасерверов удален

404 NOT_FOUND

Медиасервер не найден

Получение списка медиасерверов (GET `/media_servers`)

200 OK

Получен список медиасерверов

Авторизация медиасервера (POST /media_servers/authorize)

```
{
  "data": {
    "hostname": "my_hostname" ❶
  }
}
```

❶ Обязательное поле

201 CREATED

Медиасервер авторизован

Получение дерева медиасерверов (GET /media_servers/tree)

В списке отображаются медиасерверы, с пулами, медиасерверы, у которых нет пула. Также в дереве отображаются все пулы медиасерверов и ассоциированные с ними хранилища (директории, блочные устройства, ленточные картриджи) и пулы без ассоциированных устройств

200 OK

Получено дерево медиасерверов

Получение списка медиасерверов, имеющих пул типа ленточная библиотека (GET /media_servers/filtered_by_tape_library_pool)**200 OK**

Получено дерево медиасерверов

Редактирование медиасервера (PATCH /media_servers/{id})

Также в query-параметрах передается `id` медиасервера, информацию о котором необходимо изменить

```
{
  "data": {
    "description": "",
    "local_service_mode": "no"
  }
}
```

200 OK

Параметры медиасервера изменены

400 BAD_REQUEST

Некорректное тело запроса

404 NOT_FOUND

Медиасервер не найден

Удаление медиасервера (DELETE /media_servers/{id})

Тела запроса нет. В query-параметрах передается `id` медиасервера, который необходимо удалить

200 OK

Медиасервер удален

404 NOT_FOUND

Медиасервер не найден

Получение информации о медиасервере (GET /media_servers/{id})

Тела запроса нет. В query-параметрах передается `id` медиасервера, информацию о котором необходимо получить

200 OK

Получена информация о медиасервере

Неавторизованные медиасерверы**Удаление списка неавторизованных медиасерверов (DELETE /unauthorised_media_servers)**

```
{
  "ids": [ ①
    1
  ]
}
```

① Обязательное поле

200 OK

Список неавторизованных медиасерверов удален

404 NOT_FOUND

Неавторизованные медиасерверы не найдены

Получение списка неавторизованных медиасерверов (GET /unauthorised_media_servers)**200 OK**

Получен список неавторизованных медиасерверов

Удаление неавторизованного медиасервера (DELETE /unauthorised_media_servers/{id})

Тело запроса отсутствует. В query-параметрах передается `id` неавторизованного медиасервера, которого необходимо удалить

200 OK

Неавторизованный медиасервер удален

404 NOT_FOUND

Неавторизованный медиасервер не найден

Получение информации о неавторизованном медиасервере (GET /unauthorised_media_servers/{id})

Тело запроса отсутствует. В query-параметрах передается `id` неавторизованного медиасервера, информацию о котором необходимо получить

200 OK

Получена информация о неавторизованном медиасервере

Репозиторий**Получение списка объектов репозитория (GET /repository)****200 OK**

Получен список сделанных резервных копий

Удаление списка резервных копий из репозитория (DELETE /repository)

```
{
  "ids": [ ①
    1
  ]
}
```

① Обязательное поле

200 OK

Список резервных копий удален

404 NOT_FOUND

Резервная копия не найдена

Получение списка зависимостей для нескольких цепочек (POST repository/deletion_reference_chain)

```
{
  "ids": [ ①
    1
  ]
}
```

```
]
}
```

1 Обязательное поле

200 OK

Получен список зависимостей для цепочки резервных копий

Получение дополнительных параметров восстановления (GET `/repository/extension`)

200 OK

Получен список дополнительных параметров восстановления

Получение информации об объекте репозитория (GET `/repository/{id}`)

200 OK

Получена информация о резервной копии

Удаление резервной копии из репозитория (DELETE `/repository/{id}`)

200 OK

Резервная копия удалена

404 NOT_FOUND

Резервная копия не найдена

Изменение периода хранения резервной копии (PATCH `/repository/{id}`)

```
{
  "data": {
    "store_until": "2024-09-07T22:06:09.801038"
  }
}
```

200 OK

Период хранения резервной копии изменен

400 BAD_REQUEST

Некорректное тело запроса

404 NOT_FOUND

Резервная копия не найдена

Копирование полной резервной копии (POST `/repository/{id}/copying`)

Тела запроса нет. В query-параметрах передается `id` резервной копии и названия

пула, в который необходимо провести копирование

201 CREATED

Задача на копирование резервной копии создана

404 NOT_FOUND

Резервная копия не найдена

Перемещение полной резервной копии (POST /repository/{id}/moving)

Тела запроса нет. В query-параметрах передается `id` резервной копии и названия пула, в который необходимо провести перемещение

201 CREATED

Задача на перемещение резервной копии создана

404 NOT_FOUND

Резервная копия не найдена

Восстановление резервной копии (POST /repository/{id}/restoring)

Также в query-параметрах передается `id` резервной копии, которую необходимо восстановить. Для цепочек копий передается `id` последней разностной копии

```
{
  "data": {
    "restore_task": {
      "target_client": "rsafin-primary (55e05458152af100)", ❶
      "destination_path": "/for_restore/", ❶
      "data_deployment": false,
      "granular_restore_files": [
        ""
      ]
    },
    "module_extensions": {
      "parameters": {
        "new_name": ""
      }
    },
    "common_modules_extensions": {
      "parameters": {
        "worker_parallelism": 8,
        "memory_threshold": 0
      }
    }
  }
}
```

```
}
```

1 Обязательное поле

201 CREATED

Задача на восстановление резервной копии создана

400 BAD_REQUEST

Некорректное тело запроса

404 NOT_FOUND

Резервная копия не найдена

Верификация резервной копии (POST /repository/{id}/verification)

Тела запроса нет. В query-параметрах передается `id` резервной копии, которую необходимо верифицировать

201 CREATED

Задача на верификацию резервной копии создана

404 NOT_FOUND

Резервная копия не найдена

Содержимое копии для гранулярного восстановления

Получение списка ресурсов, включенных в копию для гранулярного восстановления (GET /repository_record_file_list)

Тела запроса нет. В query-параметрах передается `id` резервной копии, список файлов которой необходимо получить

200 OK

Получен список ресурсов, включенных в копию для гранулярного восстановления

8.2.7. Стратегии

Стратегии резервного копирования

Изменение статуса стратегии (PATCH /backup_strategies)

```
{
  "data": [
    {
      "strategy_id": 1, ①
      "status": "wait" ①
    }
  ]
}
```

```

    }
  ]
}
```

1 Обязательное поле

200 OK

Статус стратегии изменен

400 BAD_REQUEST

Некорректное тело запроса

404 NOT_FOUND

Стратегия не найдена

Получение списка стратегий (GET /backup_strategies)

200 OK

Получен список стратегий

Удаление списка стратегий (DELETE /backup_strategies)

```

{
  "ids": [ 1
    1
  ]
}
```

1 Обязательное поле

200 OK

Стратегии удалены

404 NOT_FOUND

Стратегия не найдена

Создание стратегии (POST /backup_strategies)

```

{
  "data": {
    "name": "my_backup_strategy", 1
    "status": "wait",
    "pool_id": "Default",
    "crypto": "nocrypt",
    "storage_capacity": 50,
```

```
"description": "some text about backup strategy",
"validity_start_period": "2024-08-13T14:33:31.359954",
"validity_end_period": "2025-08-13T14:33:31.360035",
"verify_flag": true,
"verify_interval": "1 day",
"auto_delete_obsoleted_copy_flag": false,
"inform_when_obsoleted_copy": "Nobody",
"client_delete_flag": true,
"full_archive_enabled": false,
"full_periodic_launch": "1 min",
"full_storage_duration": "1 day",
"full_min": 0,
"full_hour": 0,
"full_dom": 1,
"full_mon": 1,
"full_dow": 1,
"full_move_copy_flag": false,
"full_move_copy_while": "1 day",
"full_move_copy_pool": "Default",
"inc_archive_enabled": false,
"inc_periodic_launch": "1 min",
"inc_storage_duration": "1 day",
"inc_min": 0,
"inc_hour": 0,
"inc_dom": 1,
"inc_mon": 1,
"inc_dow": 1,
"inc_move_copy_flag": false,
"inc_move_copy_pool": "Default",
"inc_move_copy_while": "1 day",
"diff_archive_enabled": false,
"diff_periodic_launch": "1 min",
"diff_storage_duration": "1 day",
"diff_min": 0,
"diff_hour": 0,
"diff_dom": 1,
"diff_mon": 1,
"diff_dow": 1,
"diff_move_copy_flag": false,
"diff_move_copy_pool": "Default",
"diff_move_copy_while": "1 day",
"notify_normal": "Nobody",
"notify_normal_cc": "email@http://domain.ru[domain.ru]",
"notify_exception": "Nobody",
"notify_exception_cc": "email@http://domain.ru[domain.ru]",
```

```

"notify_verify": "Nobody",
"notify_verify_cc": "email@http://domain.ru[domain.ru]",
"notify_strategy_validity": "Nobody",
"notify_strategy_validity_cc": "email@http://domain.ru[domain.ru]",
"notify_end_storage_capacity": "Nobody",
"notify_end_storage_capacity_cc": "email@http://domain.ru[domain.ru]"
}
}

```

1 Обязательное поле

201 CREATED

Стратегия создана

Изменение параметров стратегии (PATCH /backup_strategies/{id})

Также в query-параметрах передается `id` стратегии, информацию о которой необходимо изменить.

```

{
  "data": {
    "name": "my_backup_strategy",
    "status": "wait",
    "pool_id": "Default",
    "crypto": "nocrypt",
    "storage_capacity": 50,
    "description": "some text about backup strategy",
    "validity_start_period": "2024-08-13T14:33:31.359954",
    "validity_end_period": "2025-08-13T14:33:31.360035",
    "verify_flag": true,
    "verify_interval": "1 day",
    "auto_delete_obsoleted_copy_flag": false,
    "inform_when_obsoleted_copy": "Nobody",
    "client_delete_flag": true,
    "full_archive_enabled": false,
    "full_periodic_launch": "1 min",
    "full_storage_duration": "1 day",
    "full_min": 0,
    "full_hour": 0,
    "full_dom": 1,
    "full_mon": 1,
    "full_dow": 1,
    "full_move_copy_flag": false,
    "full_move_copy_while": "1 day",
    "full_move_copy_pool": "Default",

```

```
"inc_archive_enabled": false,
"inc_periodic_launch": "1 min",
"inc_storage_duration": "1 day",
"inc_min": 0,
"inc_hour": 0,
"inc_dom": 1,
"inc_mon": 1,
"inc_dow": 1,
"inc_move_copy_flag": false,
"inc_move_copy_pool": "Default",
"inc_move_copy_while": "1 day",
"diff_archive_enabled": false,
"diff_periodic_launch": "1 min",
"diff_storage_duration": "1 day",
"diff_min": 0,
"diff_hour": 0,
"diff_dom": 1,
"diff_mon": 1,
"diff_dow": 1,
"diff_move_copy_flag": false,
"diff_move_copy_pool": "Default",
"diff_move_copy_while": "1 day",
"notify_normal": "Nobody",
"notify_normal_cc": "email@http://domain.ru[domain.ru]",
"notify_exception": "Nobody",
"notify_exception_cc": "email@http://domain.ru[domain.ru]",
"notify_verify": "Nobody",
"notify_verify_cc": "email@http://domain.ru[domain.ru]",
"notify_strategy_validity": "Nobody",
"notify_strategy_validity_cc": "email@http://domain.ru[domain.ru]",
"notify_end_storage_capacity": "Nobody",
"notify_end_storage_capacity_cc": "email@http://domain.ru[domain.ru]"
}
}
```

200 OK

Стратегия изменена

400 BAD_REQUEST

Некорректное тело запроса

404 NOT_FOUND

Стратегия не найдена

Получение информации о стратегии (GET /backup_strategies/{id})

Тела запроса нет. В query-параметрах передается `id` стратегии, информацию о которой необходимо получить

200 OK

Получена информация о стратегии

Удалении одной стратегии (DELETE /backup_strategies/{id})

Тела запроса нет. В query-параметрах передается `id` стратегии, которую необходимо удалить

200 OK

Стратегия удалена

404 NOT_FOUND

Стратегия не найдена

Копирование в пул по стратегиям**Получение списка настроенного копирования резервных копий по стратегиям в пул (GET /backup_strategies_replication)****200 OK**

Получен список правил копирования

Удаление списка настроенного копирования резервных копий по стратегиям (DELETE /backup_strategies_replication)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

200 OK

Правило копирования удалено

404 NOT_FOUND

Правило копирования не найдено

Создание правила копирования резервных копий по стратегиям в пул (POST /backup_strategies_replication)

```
{
```

```

{
  "data": {
    "client": "rubackup-client(525a99154f3505a2)", ❶
    "backup_bandwidth": 0,
    "restore_bandwidth": 0,
    "window_start": "18:10:20",
    "window_end": "18:10:20"
  }
}

```

❶ Обязательное поле

201 CREATED

Правило копирования создано

Получение информации о копировании резервных копий по стратегии в пул (GET /backup_strategies_replication/{id})

Тела запроса нет. В query-параметрах передается `id` правила копирования, которое необходимо удалить

200 OK

Получена информация о правиле копирования

Удаление настроенного копирования резервных копий по стратегиям (DELETE backup_strategies_replication/{id})

Тела запроса нет. В query-параметрах передается `id` правила копирования, информацию о котором необходимо получить

200 OK

Правило копирования удалено

404 NOT_FOUND

Правило копирования не найдено

Изменение параметров правила копирования резервных копий по стратегии в пул (PATCH backup_strategies_replication/{id})

Также в query-параметрах передается `id` правила копирования, информацию о котором необходимо изменить

```

{
  "data": {
    "backup_bandwidth": 0,
    "restore_bandwidth": 0,
    "window_start": "18:10:20",
    "window_end": "18:10:20"
  }
}

```

```
}
}
```

200 OK

Правило копирования удалено

404 NOT_FOUND

Правило копирования не найдено

Правила стратегий

Удаление списка правил стратегии (DELETE /strategy_rules)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле**200 OK**

Список правил стратегии удален

404 NOT_FOUND

Правило стратегии не найдено

Получение списка правил стратегий (GET /strategy_rules)**200 OK**

Получен список правил стратегии

Добавление правила в стратегию (POST /strategy_rules)

```
{
  "data": {
    "strategy_rule": {
      "client_id": "rubackup-client (525a99154f3505a2)",
      "strategy_id": "Default",
      "resource_type": "File system", ❶
      "resource": "/home/", ❶
      "normal_execution_script": "",
      "exception_execution_script": "",
      "priority": 100,
      "restore_script": ""
    }
  }
}
```

```

    },
    "module_extensions": {
      "parameters": {
        "file_list": true,
        "numeric_owner": false,
        "use_snapshot": false,
        "ignore_errors_snapshot": true,
        "snapshot_type": "",
        "snapshot_size": 10,
        "script_before_snapshot": "",
        "script_after_snapshot": "",
        "script_error_snapshot": ""
      }
    },
    "common_modules_extensions": {
      "parameters": {
        "worker_parallelism": 8,
        "enable_multithreading": false,
        "enable_flexible_dedup": false,
        "network_parallelism": 8,
        "memory_threshold": 0,
        "deny_memory_exceed": false
      }
    }
  }
}

```

1 Обязательное поле

201 CREATED

Правило добавлено в стратегию

Получение дополнительных параметров правила стратегии (GET `/strategy_rules/extension`)

Тело запроса отсутствует. В query-параметрах передается название ресурса (`resource_type`) или название ресурса и `id` правила, информацию о дополнительных параметрах которых необходимо получить

200 OK

Информация о дополнительных параметрах получена

Изменение правила стратегии (PATCH `/strategy_rules/{id}`)

Также в query-параметрах передается `id` правила, информацию о котором необходимо изменить

```

{
  "data": {
    "strategy_rule": {
      "client_id": "rubakup-client (525a99154f3505a2)",
      "strategy_id": "Default",
      "resource_type": "File system",
      "resource": "/home/",
      "normal_execution_script": "",
      "exception_execution_script": "",
      "priority": 100,
      "restore_script": ""
    },
    "module_extensions": {
      "parameters": {
        "file_list": true,
        "numeric_owner": false,
        "use_snapshot": false,
        "ignore_errors_snapshot": true,
        "snapshot_type": "",
        "snapshot_size": 10,
        "script_before_snapshot": "",
        "script_after_snapshot": "",
        "script_error_snapshot": ""
      }
    },
    "common_modules_extensions": {
      "parameters": {
        "worker_parallelism": 8,
        "enable_multithreading": false,
        "enable_flexible_dedup": false,
        "network_parallelism": 8,
        "memory_threshold": 0,
        "deny_memory_exceed": false
      }
    }
  }
}

```

200 OK

Параметры правила стратегии изменены

400 BAD_REQUEST

Некорректное тело запроса

404 NOT_FOUND

Правило стратегии не найдено

Удаление правила стратегии (DELETE /strategy_rules/{id})

Тело запроса отсутствует. В query-параметрах передается `id` правила, которое необходимо удалить

200 OK

Информация о правиле стратегии получена

404 NOT_FOUND

Правило стратегии не найдено

Получение информации о правиле стратегии (GET /strategy_rules/{id})

Тело запроса отсутствует. В query-параметрах передается `id` правила, информацию о котором необходимо получить

200 OK

Правило стратегии удалено

404 NOT_FOUND

Правило стратегии не найдено

Администраторы стратегий**Получение списка администраторов для стратегии (GET /strategy_administrators)****200 OK**

Получен список администраторов для стратегии

Получение информации об администраторе стратегии (GET /strategy_administrators/{id})

Тела запроса нет. В query-параметрах передается `id` администратора стратегии, информацию о котором необходимо получить

200 OK

Получена информация об администраторе для стратегии

404 NOT_FOUND

Информация об администраторе для стратегии не найдена

Назначение администраторов для стратегии (POST /strategy_administrators)

```
{  
  "data": {
```

```

    "strategy": "Default", ❶
    "administrator": "Nobody" ❶
  }
}

```

❶ Обязательное поле

201 CREATED

Администратор для стратегии назначен

Удаление списка администраторов стратегии (DELETE /strategy_administrators)

```

{
  "ids": [ ❶
    1
  ]
}

```

❶ Обязательное поле

200 OK

Список администраторов стратегии удален

404 NOT_FOUND

Администратор стратегии не найден

Удаление администратора стратегии (DELETE /strategy_administrators/{id})

Тела запроса нет. В query-параметрах передается `id` администратора стратегии, которого необходимо удалить

200 OK

Администратор стратегии удален

404 NOT_FOUND

Администратор стратегии не найден

8.2.8. Глобальное расписание

Правила глобального расписания

Изменение статуса правила глобального расписания (PATCH /global_schedule)

```

{
  "data": [
    {

```

```

    "id": 1, ❶
    "status": "wait" ❶
  }
]
}

```

❶ Обязательное поле

200 OK

Статус правила изменен

400 BAD_REQUEST

Некорректное тело запроса

404 NOT_FOUND

Правило глобального расписания не найдено

Получение списка правил глобального расписания (GET /global_schedule)

200 OK

Параметры правила глобального расписания получены

Удаление списка правил глобального расписания (DELETE /global_schedule)

```

{
  "ids": [ ❶
    1
  ]
}

```

❶ Обязательное поле

200 OK

Правило глобального расписания удалено

Добавление правила глобального расписания (POST /global_schedule)

```

{
  "data": {
    "global_schedule": {
      "name": "global_schedule_name", ❶
      "client_id": "rubackup-client (525a99154f3505a2)", ❶
      "resource_type": "File system", ❶
      "resource": "/home/", ❶
      "backup_type": "full",
    }
  }
}

```

```

"pool_id": "Default",
"storage_capacity": 10,
"priority": 100,
"crypto": "nocrypt",
"validity_start_period": "2024-09-02T15:40:01.208623",
"validity_end_period": "2025-09-02T15:40:01.208636",
"periodic_launch": "1 min",
"min": 0,
"hour": 0,
"dom": 1,
"mon": 1,
"dow": 1,
"verify_flag": true,
"verify_interval": "1 day",
"storage_duration": "1 day",
"move_copy_flag": false,
"move_copy_pool": "Default",
"move_copy_while": "1 day",
"auto_delete_obsoleted_copy_flag": false,
"inform_when_obsoleted_copy": "Nobody",
"client_delete_flag": true,
"notify_normal": "Nobody",
"notify_normal_cc": "email@http://domain.ru[domain.ru]",
"notify_exception": "Nobody",
"notify_exception_cc": "email@http://domain.ru[domain.ru]",
"notify_rule_validity": "Nobody",
"notify_rule_validity_cc": "email@http://domain.ru[domain.ru]",
"notify_verify": "Nobody",
"notify_verify_cc": "email@http://domain.ru[domain.ru]",
"status": "wait",
"notify_normal_script": "",
"notify_exception_script": "",
"notify_end_storage_capacity": "Nobody",
"notify_end_storage_capacity_cc": "email@http://domain.ru[domain.ru]",
"restore_script": ""
},
"module_extensions": {
  "parameters": {
    "file_list": true,
    "numeric_owner": false,
    "use_snapshot": false,
    "ignore_errors_snapshot": true,
    "snapshot_type": "",
    "snapshot_size": 10,
    "script_before_snapshot": "",

```

```

    "script_after_snapshot": "",
    "script_error_snapshot": ""
  },
  "common_modules_extensions": {
    "parameters": {
      "worker_parallelism": 8,
      "enable_multithreading": false,
      "enable_flexible_dedup": false,
      "network_parallelism": 8,
      "memory_threshold": 0,
      "deny_memory_exceed": false
    }
  }
}

```

1 Обязательное поле

201 CREATED

Правило глобального расписания создано

Получение дополнительных параметров глобального расписания по id правила (GET /global_schedule/extension)

Тело запроса отсутствует. В query-параметрах передается название ресурса (resource_type) или название ресурса и id правила, информацию о дополнительных параметрах которых необходимо получить

200 OK

Информация о дополнительных параметрах получена

Изменение правила глобального расписания (PATCH /global_schedule/{id})

Также в query-параметрах передается id правила, информацию о котором необходимо изменить

```

{
  "data": {
    "global_schedule": {
      "name": "global_schedule_name",
      "pool_id": "Default",
      "storage_capacity": 10,
      "priority": 100,
      "validity_start_period": "2024-09-02T15:40:01.208623",
      "validity_end_period": "2025-09-02T15:40:01.208636",
    }
  }
}

```

```

    "periodic_launch": "1 min",
    "min": 0,
    "hour": 0,
    "dom": 1,
    "mon": 1,
    "dow": 1,
    "verify_flag": true,
    "verify_interval": "1 day",
    "storage_duration": "1 day",
    "move_copy_flag": false,
    "move_copy_pool": "Default",
    "move_copy_while": "1 day",
    "auto_delete_obsoleted_copy_flag": false,
    "inform_when_obsoleted_copy": "Nobody",
    "client_delete_flag": true,
    "notify_normal": "Nobody",
    "notify_normal_cc": "email@http://domain.ru[domain.ru]",
    "notify_exception": "Nobody",
    "notify_exception_cc": "email@http://domain.ru[domain.ru]",
    "notify_rule_validity": "Nobody",
    "notify_rule_validity_cc": "email@http://domain.ru[domain.ru]",
    "notify_verify": "Nobody",
    "notify_verify_cc": "email@http://domain.ru[domain.ru]",
    "status": "wait",
    "notify_normal_script": "",
    "notify_exception_script": "",
    "notify_end_storage_capacity": "Nobody",
    "notify_end_storage_capacity_cc": "email@http://domain.ru[domain.ru]",
    "restore_script": ""
  },
  "module_extensions": {
    "parameters": {
      "file_list": true,
      "numeric_owner": false,
      "use_snapshot": false,
      "ignore_errors_snapshot": true,
      "snapshot_type": "",
      "snapshot_size": 10,
      "script_before_snapshot": "",
      "script_after_snapshot": "",
      "script_error_snapshot": ""
    }
  },
  "common_modules_extensions": {
    "parameters": {

```

```
    "worker_parallelism": 8,  
    "enable_multithreading": false,  
    "enable_flexible_dedup": false,  
    "network_parallelism": 8,  
    "memory_threshold": 0,  
    "deny_memory_exceed": false  
  }  
}  
}
```

200 OK

Параметры правила глобального расписания изменены

400 BAD_REQUEST

Некорректное тело запроса

404 NOT_FOUND

Правило глобального расписания не найдено

Получение информации о правиле глобального расписания (GET /global_schedule/{id})

Тело запроса отсутствует. В query-параметрах передается `id` правила, информацию о котором необходимо получить

200 OK

Информация о правиле глобального расписания получена

404 NOT_FOUND

Правило глобального расписания не найдено

Удаление правила глобального расписания (DELETE /global_schedule/{id})

Тело запроса отсутствует. В query-параметрах передается `id` правила, которое необходимо удалить

200 OK

Правило глобального расписания удалено

404 NOT_FOUND

Правило глобального расписания не найдено

Создание полной резервной копии по правилу глобального расписания (POST /global_schedule/{id}/execution)

Тело запроса отсутствует. В query-параметрах передается `id` правила, которое необходимо выполнить ("принудительно")

200 OK

Правило глобального расписания выполнено

404 NOT_FOUND

Правило глобального расписания не найдено

Репликация правил глобального расписания

Получение списка настроенных репликаций правил глобального расписания (GET /global_schedule_replication)

200 OK

Получен список правил копирования

Настройка репликации правил глобального расписания (POST /global_schedule_replication)

```
{
  "data": {
    "rule": 1, ①
    "pool": "Default", ①
    "capacity": 0,
    "capacity_enabled": false,
    "period": "1mon",
    "period_enabled": false,
    "work_window_start": "00:00:00",
    "work_window_end": "00:00:00"
  }
}
```

① Обязательное поле

201 CREATED

правило копирования создано

Удаление списка настроенных репликаций правил глобального расписания (DELETE /global_schedule_replication)

```
{
  "ids": [ ①
    1
  ]
}
```

① Обязательное поле

200 OK

правило копирования удалено

404 NOT_FOUND

правило копирования не найдена

Получение информации о настроенной репликации правила глобального расписания (GET /global_schedule_replication/{id})

Тела запроса нет. В query-параметрах передается `id` правила копирования, которое необходимо удалить

200 OK

получена информация о правиле копирования

Удаление настроенной репликации правила глобального расписания (DELETE /global_schedule_replication/{id})

Тела запроса нет. В query-параметрах передается `id` правила копирования, информацию о котором необходимо получить

200 OK

правило копирования удалено

404 NOT_FOUND

правило копирования не найдено

Изменение параметров репликации правил глобального расписания (PATCH /global_schedule_replication/{id})

Также в query-параметрах передается `id` правила копирования, информацию о котором необходимо изменить

```
{
  "data": {
    "capacity": 0,
    "capacity_enabled": false,
    "period": "1mon",
    "period_enabled": false,
    "work_window_start": "00:00:00",
    "work_window_end": "00:00:00"
  }
}
```

200 OK

правило копирования удалено

404 NOT_FOUND

правило копирования не найдена

Запросы на добавление правил глобального расписания

Получение списка запросов клиента на добавление правила глобального расписания (GET /new_rule_requests)

200 OK

Получен список запросов клиента на добавление правила глобального расписания

Одобрение запроса клиента на добавление правила глобального расписания (POST /new_rule_requests/approve)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

200 NO_CONTENT

Запросы клиента на добавление правила глобального расписания одобрены

207 MULTI_STATUS

Часть из списка запросов клиента на добавление правила глобального расписания одобрена, часть нет

404 NOT_FOUND

Запросы клиента на добавление правила глобального расписания не найдены

Отклонение запроса клиента на добавление правила глобального расписания (DELETE /new_rule_requests/decline)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

200 NO_CONTENT

Запросы клиента на добавление правила глобального расписания отклонены

207 MULTI_STATUS

Часть из списка запросов клиента на добавление правила глобального расписания отклонена, часть нет

404 NOT_FOUND

Запросы клиента на добавление правила глобального расписания не найдены

Получение информации о запросе клиента на добавление правила глобального расписания (GET /new_rule_requests/{id})

Тела запроса нет. В query-параметрах передается `id` запроса, информацию о котором необходимо получить

200 OK

Получена информация о запросе клиента на добавление правила глобального расписания

Запросы на удаление правил глобального расписания

Получение списка запросов клиента на удаление правила глобального расписания (GET /delete_rule_requests)

200 OK

Получен запросов клиента на удаление правила глобального расписания

Одобрение запроса клиента на удаление правила глобального расписания (POST /delete_rule_requests/approve)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

200 NO_CONTENT

Запросы клиента на удаление правила глобального расписания одобрены

207 MULTI_STATUS

Часть из списка запросов клиента на удаление правила глобального расписания одобрена, часть нет

404 NOT_FOUND

Запросы клиента на удаление правила глобального расписания не найдены

Отклонение запроса клиента на удаление правила глобального расписания (DELETE /delete_rule_requests/decline)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

204 NO_CONTENT

Запросы клиента на удаление правила глобального расписания отклонены

207 MULTI_STATUS

Часть из списка запросов клиента на удаление правила глобального расписания отклонена, часть нет

404 NOT_FOUND

Запросы клиента на удаление правила глобального расписания не найдены

Получение информации о запросе клиента на удаление правила глобального расписания (GET /delete_rule_requests/{id})

Тела запроса нет. В query-параметрах передается `id` запроса, информацию о котором необходимо получить

200 OK

Получена информация о запросе клиента на удаление правила глобального расписания

Ограничения пропускной способности**Получение списка ограничений пропускной способности правила глобального расписания (GET /rule_bandwidth)****200 OK**

получен список ограничений пропускной способности

Удаление списка ограничений пропускной способности правила глобального расписания (DELETE /rule_bandwidth)

```
{
  "ids": [ ❶
    1
  ]
}
```

1 Обязательное поле**200 OK**

список ограничений пропускной способности удален

404 NOT_FOUND

ограничение пропускной способности не найдено

Ограничение пропускной способности правила глобального расписания (POST /rule_bandwidth)

```
{
  "data": {
    "rule": 1, 1
    "backup_bandwidth": 0,
    "restore_bandwidth": 0,
    "window_start": "00:00:00",
    "window_end": "00:00:00"
  }
}
```

1 Обязательное поле**201 CREATED**

ограничение пропускной способности создано

Изменение ограничения пропускной способности правила глобального расписания (PATCH /rule_bandwidth/{id})

Также в query-параметрах передается `id` ограничения, информацию о котором необходимо изменить

```
{
  "data": {
    "rule": 1,
    "backup_bandwidth": 0,
    "restore_bandwidth": 0,
    "window_start": "00:00:00",
    "window_end": "00:00:00",
    "id": 1
  }
}
```

200 OK

параметры ограничения пропускной способности изменены

400 BAD_REQUEST

некорректное тело запроса

404 NOT_FOUND

ограничение пропускной способности не найдено

Удаление ограничения пропускной способности правила глобального расписания (DELETE /rule_bandwidth/{id})

Тела запроса нет. В query-параметрах передается `id` ограничения, которое необходимо удалить

200 OK

ограничение пропускной способности удалено

404 NOT_FOUND

ограничение пропускной способности не найдено

Получение информации об ограничении пропускной способности правила глобального расписания (GET /rule_bandwidth/{id})

Тела запроса нет. В query-параметрах передается `id` ограничения, информацию о котором необходимо получить

200 OK

получена информация об ограничении пропускной способности

404 NOT_FOUND

ограничение пропускной способности не найдено

8.2.9. Задачи

Задачи

Удаление нескольких задач (DELETE /task_queue)

```
{
  "ids": [ ①
    1
  ]
}
```

① Обязательное поле

200 OK

Список задач из очереди задач удален

404 NOT_FOUND

Задача не найдена в очереди

Получение очереди задач (GET /task_queue)

200 OK

Получен список задач из очереди задач

Создание срочной полной резервной копии (POST /task_queue)

```
{
  "data": {
    "task": {
      "resource_type": "File system", ①
      "resource": "/home/", ①
      "client": "rubcp-client (525a99154f3505a2)", ①
      "backup_type": "full", ①
      "pool": "Default", ①
      "crypto": "nocrypt", ①
      "priority": 100, ①
      "storage_duration": "1 Years", ①
      "archiving": false ①
    },
    "module_extensions": {
      "parameters": {
        "file_list": true,
        "numeric_owner": false,
        "use_snapshot": false,
        "ignore_errors_snapshot": true,
        "snapshot_type": "",
        "snapshot_size": 10,
        "script_before_snapshot": "",
        "script_after_snapshot": "",
        "script_error_snapshot": ""
      }
    },
    "common_modules_extensions": {
      "parameters": {
        "worker_parallelism": 8,
        "enable_multithreading": false,
        "enable_flexible_dedup": false,
        "network_parallelism": 8,
        "memory_threshold": 0,
        "deny_memory_exceed": false
      }
    }
  }
}
```

```

    }
  }
}
```

1 Обязательное поле

201 CREATED

Создана задача на срочное РК

Получение дополнительных параметров созданной задачи (GET /task_queue/extension)

Тело запроса отсутствует. В query-параметрах передается название ресурса (resource_type) или название ресурса и id правила, информацию о дополнительных параметрах которых необходимо получить

200 OK

Информация о дополнительных параметрах получена

Удаление задачи (DELETE /task_queue/{id})

Тело запроса отсутствует. В query-параметрах передается id задачи, которую необходимо удалить

200 OK

Задача удалена из очереди

404 NOT_FOUND

Задача не найдена в очереди

Получение информации о задаче из очереди (GET /task_queue/{id})

Тело запроса отсутствует. В query-параметрах передается id задачи, информацию о которой необходимо получить

200 OK

Информация о задаче из очереди получена

404 NOT_FOUND

Задача не найдена в очереди

Удаление задачи с определенным статусом (DELETE /task_queue/{status})

Тело запроса отсутствует. В query-параметрах передается статус, задачи с которым необходимо удалить

Возможные статусы: error, killed, obsoleted (obsoleted - устаревшие, т.к. в статусе done).

200 OK

Задачи с выбранным статусом удалены из очереди

404 NOT_FOUND

Задач с выбранным статусом не найдена в очереди

Статусы задач

Получение списка возможных статусов задач (GET /task_status)

200 OK

Получен список возможных статусов задач

Перезапуск задач

Перезапуск задачи (POST /restart_task)

```
{
  "data": {
    "tasks_ids": [ ❶
      1
    ]
  }
}
```

❶ Обязательное поле

201 CREATED

Задача перезапущена

400 BAD_REQUEST

Некорректное тело запроса (задачу с таким статусом нельзя перезапустить)

Приостановка задач

Приостановка выполнения задачи (POST /put_task_on_pause)

```
{
  "data": {
    "task_id": 1 ❶
  }
}
```

❶ Обязательное поле

200 OK

Задача приостановлена

400 BAD_REQUEST

Некорректное тело запроса (задачу с таким статусом нельзя приостановить)

Возобновление выполнения задач**Возобновление приостановленной задачи (POST /continue_task_working)**

```
{
  "data": {
    "task_id": 1 ①
  }
}
```

① Обязательное поле

200 OK

Задача возобновлена

400 BAD_REQUEST

Некорректное тело запроса (задачу с таким статусом нельзя возобновить)

Принудительное завершение задач**Убийство задачи (POST /kill_task)**

```
{
  "data": {
    "task_id": 1 ①
  }
}
```

① Обязательное поле

200 OK

Задача убита

400 BAD_REQUEST

Некорректное тело запроса (задачу с таким статусом нельзя убить)

Очередь задач ленточных библиотек

Получение списка задач из очереди ленточных библиотек (GET /tl_task_queue)

200 OK

Получен список задач из очереди задач ленточных библиотек

Получение информации о задаче из очереди ленточных библиотек (GET /tl_task_queue/{id})

Тело запроса отсутствует. В query-параметрах передается id задачи из очереди задач ленточных библиотек, информацию о которой необходимо получить

200 OK

Получена информация о задаче из очереди задач ленточных библиотек

404 NOT_FOUND

Задача в очереди задач ленточных библиотек не найдена

8.2.10. Уведомления

Статус уведомлений

Получение списка статусов уведомлений (GET /notifications_status)

200 OK

Получен список существующих статусов для уведомлений

Уведомления

Получение списка уведомлений (GET /notifications)

200 OK

Получен список уведомлений

Получение информации об уведомлении (GET /notifications/{id})

Тела запроса нет. В query-параметрах передается id уведомления, информацию о котором необходимо получить

200 OK

Получена информация об уведомлении

Уведомления групп

Удаление списка групп для уведомлений (DELETE /user_groups)

```
{
  "ids": [ 1 ]
}
```

```
}

```

1 Обязательное поле

200 OK

Список групп пользователей удален

404 NOT_FOUND

Группа пользователей не найдена

Получение списка групп для уведомлений (GET /user_groups)

200 OK

Список групп пользователей получен

Добавление группы уведомлений (POST /user_groups)

```
{
  "data": {
    "groupname": "my_group_name", 1
    "description": ""
  }
}
```

1 Обязательное поле

201 CREATED

Создана группа пользователей

Изменение группы для уведомлений (PATCH /user_groups/{id})

Также в query-параметрах передается `id` группы пользователей, информацию о которой необходимо изменить

```
{
  "data": {
    "groupname": "my_group_name",
    "description": ""
  }
}
```

200 OK

Параметры группы пользователей изменены

400 BAD_REQUEST

Некорректное тело запроса

404 NOT_FOUND

Группа пользователей не найдена

Удаление группы для уведомлений (DELETE /user_groups/{id})

Тела запроса нет. В query-параметрах передается `id` группы пользователей, которую необходимо удалить

200 OK

Группа пользователей удалена

404 NOT_FOUND

Группа пользователей не найдена

Получение информации о группе уведомлений (GET /user_groups/{id})

Тела запроса нет. В query-параметрах передается `id` группы пользователей, информацию о которой необходимо получить

200 OK

Получена информация о группе пользователей

404 NOT_FOUND

Группа пользователей не найдена

8.2.11. Журналы

Журнал операций клиента

Получение журнала со списком операций клиентов (GET /clients_log)**200 OK**

Получен список записей в журнале операций клиентов

Получение одной записи из журнала с операций клиентов (GET /clients_log/{id})

Тела запроса нет. В query-параметрах передается `id` операции клиента, информацию о которой необходимо получить

200 OK

Получена запись в журнале операций клиентов

404 NOT_FOUND

Запись в журнале операций клиентов не найдена

Журнал глобального расписания

Получение списка записей в журнале глобального расписания (GET /global_schedule_log)

200 OK

Получен список записей в журнале правил глобального расписания

Получение информации о записи в журнале глобального расписания (GET /global_schedule_log/{id})

Тела запроса нет. В query-параметрах передается id операции из журнала правил глобального расписания, информацию о которой необходимо получить

200 OK

Получена запись в журнале правил глобального расписания

404 NOT_FOUND

Запись в журнале правил глобального расписания не найдена

Журнал операций медиасерверов

Получение журнала со списком операций медиасерверов (GET /media_servers_log)

200 OK

Получен список записей в журнале операций медиасерверов

Получение журнала с операцией медиасервера (GET /media_servers_log/{id})

Тела запроса нет. В query-параметрах передается id операции медиасервера, информацию о которой необходимо получить

200 OK

Получена запись в журнале операций медиасерверов

404 NOT_FOUND

Запись в журнале операций медиасерверов не найдена

Журнал репозитория

Получение списка записей в журнале репозитория (GET /repository_log)

200 OK

Получен список записей в журнале репозитория

Получение информации о записи в журнале репозитория (GET /repository_log/{id})

Тела запроса нет. В query-параметрах передается id записи в журнале репозитория, информацию о которой необходимо получить

200 OK

Получена запись в журнале репозитория

404 NOT_FOUND

Запись в журнале репозитория не найдена

Журнал задач

Получение информации о записи в журнале задач (GET /task_log/{id})

Тела запроса нет. В query-параметрах передается **id** записи в журнале задач, информацию о которой необходимо получить

200 OK

Получена запись в журнале задач

404 NOT_FOUND

Запись в журнале задач не найдена

Получение списка записей в журнале задач (GET /task_log)

200 OK

Получен список записей в журнале задач

Журнал операций сервера

Получение журнала серверных операций по задаче (GET /server_task_info)

Аналогично действиям:

1. Перейти на вкладку очереди задач
2. Выбрать задачу в списке
3. Нажать "Журналы"
4. Выбрать "Серверные журналы"

Тела запроса нет. В query-параметрах передается **id** задачи из очереди задач, журнал которой необходимо получить

200 OK

Получен журнал серверных операций по задаче

Журнал клиентских операций

Получение журнала клиентских операций по задаче (GET /client_task_info)

Тела запроса нет. В query-параметрах передается **id** задачи из очереди задач, журнал которой необходимо получить

Аналогично действиям:

1. Перейти на вкладку очереди задач
2. Выбрать задачу в списке
3. Нажать "Журналы"
4. Выбрать "Клиентские журналы"

200 OK

Получен журнал клиентских операций по задаче

Журнал мониторинга сервера

Получение списка записей в журнале мониторинга сервера (GET /rubackup_server_system_monitoring)

200 OK

Получен список записей в журнале мониторинга сервера

Получение информации о записи в журнале мониторинга сервера (GET /rubackup_server_system_monitoring/{id})

Тела запроса нет. В query-параметрах передается `id` записи в журнале мониторинга сервера, информацию о которой необходимо получить

200 OK

Получена информация о записи в журнале мониторинга сервера

404 NOT_FOUND

Запись в журнале мониторинга сервера не найдена

Журнал событий информационной безопасности

Удаление списка записей из журналов событий ИБ (DELETE /security_audit_log)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

204 NO_CONTENT

Список записей из журналов ИБ удален

Просмотр журналов событий ИБ (GET /security_audit_log)**200 OK**

Список записей в журнале ИБ получен

Удаление записи из журнала событий ИБ (DELETE /security_audit_log/{id})

Тела запроса нет. В query-параметрах передается `id` записи, которую необходимо удалить

204 NO_CONTENT

Запись из журнала ИБ удалена

Просмотр записи из журнала событий ИБ (GET /security_audit_log/{id})

Тела запроса нет. В query-параметрах передается `id` записи, информацию о которой необходимо получить

200 OK

Информация о записи из журнала ИБ получена

8.2.12. Отчеты

Изменение статуса отчета (PATCH /reports)

```
{
  "data": [
    {
      "id": 1, ①
      "status": "wait" ①
    }
  ]
}
```

① Обязательное поле

200 OK

Статус отчета изменен

400 BAD_REQUEST

Некорректное тело запроса

404 NOT_FOUND

Отчет не найден

Удаление списка отчетов (DELETE /reports)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

200 OK

Список отчетов удален

404 NOT_FOUND

Отчет не найден

Получение списка отчетов (GET /reports)**200 OK**

Получен список отчетов

Создание отчета (POST /reports)

```
{
  "data": {
    "name": "my_report_name", ❶
    "owner": "Nobody", ❶
    "min": 0,
    "hour": 0,
    "dom": 1,
    "mon": 1,
    "dow": 1,
    "perpetually": true,
    "start_period": "2024-09-07T22:06:09.809052",
    "end_period": "2025-09-07T22:06:09.809067",
    "status": "wait",
    "notify": "Nobody",
    "notify_cc": "email@http://domain.ru[domain.ru]",
    "description": ""
  }
}
```

❶ Обязательное поле

201 CREATED

Отчет создан

Изменение отчета (PATCH /reports/{id})

Также в query-параметрах передается `id` отчета, информацию о котором необходимо изменить

```
{
  "data": {
    "perpetually": true,
    "start_period": "2024-09-07T22:06:09.809052",
    "end_period": "2025-09-07T22:06:09.809067",
    "status": "wait",
    "notify": "Nobody",
    "notify_cc": "email@http://domain.ru[domain.ru]",
    "description": ""
  }
}
```

200 OK

Параметры отчета изменены

400 BAD_REQUEST

Некорректное тело запроса

404 NOT_FOUND

Отчет не найден

Получение информации об отчете (DELETE /reports/{id})

Тела запроса нет. В query-параметрах передается `id` отчета, который необходимо удалить

200 OK

Отчет удален

404 NOT_FOUND

Отчет не найден

Удаление отчета (GET /reports/{id})

Тела запроса нет. В query-параметрах передается `id` отчета, информацию о котором необходимо получить

200 OK

Получена информация об отчете

404 NOT_FOUND

Отчет не найден

8.2.13. Microsoft Active Directory

Настройка LDAP

Удаление соединения (DELETE /ldap/configuration)**200 OK**

соединение с контроллером домена удалено

Настройка соединения с MS AD (POST /ldap/configuration)

```
{
  "data": {
    "domain_type": "MSAD", ①
    "protocol": "ldap", ①
    "address": "", ①
    "port": 389, ①
    "client_cert": "",
    "root_cert": "",
    "domain": "", ①
    "login": "", ①
    "password": "", ①
    "user_search_base": "", ①
    "group_search_base": "" ①
  }
}
```

① Обязательное поле**201 CREATED**

соединение с контроллером домена создано

Получение параметров настроенного соединения с MS AD (GET /ldap/configuration)**200 OK**

получены параметры соединения с контроллером домена

Ассоциация групп MS AD и ролей RuBackup

Удаление ассоциации групп MS AD и ролей RuBackup (DELETE /ldap/assosiations)

```
{
  "ids": [ ❶
    1
  ]
}
```

❶ Обязательное поле

200 OK

Ассоциация группы MS AD и ролей RuBackup удалена

404 NOT_FOUND

Ассоциация группы MS AD и ролей RuBackup не найдена

Получение списка ассоциаций групп MS AD и ролей RuBackup (POST /ldap/assosiations)

```
{
  "data": [
    {
      "domain_group": "Administrators", ❶
      "rubackup_role": "auditor" ❶
    }
  ]
}
```

❶ Обязательное поле

201 CREATED

Ассоциация группы MS AD и ролей RuBackup создана

Ассоциация групп MS AD и ролей RuBackup (GET /ldap/assosiations)**200 OK**

Получен список ассоциаций групп MS AD и ролей RuBackup