

# Документация на ИС НСК

Набор технических заданий и архитектурных решений для информационной системы НСК и входящих в её состав модулей.

- [Общие сведения](#)
  - [Общие положения](#)
  - [Структура центра обработки данных АО НК](#)
  - [Функционирование НСК-НОК](#)
  - [Функционирование СПК](#)
- [Новая система НСК](#)
  - [Топология размещения и взаимодействия сервисов НСК](#)
  - [Механизмы CI/CD](#)
  - [Аутентификация, Авторизация, Аудит](#)
  - [Проектная кадрово-организационная структура](#)
  - [Технические требования на систему управления задачами](#)
- [Старая система НСК](#)
  - [ЦОК](#)
  - [Мобильный автономный комплекс](#)
  - [Модульный конструктор ПС](#)
  - [Сервис видеозаписи](#)
  - [Алгоритмы синхронизации](#)
- [Образовательная платформа Афина](#)
  - [Интеграция Афины с 1С](#)

# Общие сведения

## Общие положения

### Декомпозиция сервисов

1. Сервис Аутентификации - "Цифровой паспорт НК"
2. Сервер документооборота - Центральное хранилище документов
3. Цифровой СПК
4. Цифровой ЦОК
5. Социальная сеть Я-Эксперт
6. Мониторинг регионов ФЗ: обучение
7. Конструктор проф стандартов и ОС

Все сервисы являются взаимосвязанными, для целей аутентификации используется единый OAuth2 сервис «Цифровой паспорт "Национальные Квалификации"»

### Объектная модель сервисов

□

## Цифровой паспорт АО НК

Цифровой паспорт реализует полную модель OAuth2 аутентификации. Включая выдачу списка ролей, доступных пользователю. OAuth2 токен используется для Baeger механизма аутентификации и по умолчанию выдается на срок 24 часа. Однако может быть запрошен и на больший срок, но не более 30 календарных дней с даты получения токена. Сервер аутентификации формирует Baeger как base64 от json структуры: Профиль пользователя, Привилегии доступа, Подпись сервера аутентификации. При запросе делегирования токена, данные токена дополнительно разбавляются солью и шифруются хешем пинкода пользователя. Для получения делегированного токена пользователь должен правильно ввести пинкод. Примером использования делегированного токена является МАК, где значение ключей не велико и возможна аутентификация эксперта только по пинкоду. В иных случаях требуется дополнительно к делегированному токenu хранить идентификатор пользователя. Пример:

1. Эксперт регистрируется в цифровом паспорте АО НК
2. СПК присваивает эксперту статус Эксперта и назначает права доступа в ЦОК
3. Эксперт проходит аутентификацию на портале ЦОК (в закрытой сети)
4. ЦОК запрашивает обычный и делегированный токен сроком на 30 дней
5. Цифровой паспорт добавляет к токenu соль, шифрует его хешем пинкода пользователя и отправляет на сервер ЦОК.
6. Сервер ЦОК реплицирует полученный делегированный токен на МАК
7. На планшете эксперта подключенному к МАК Эксперт вводит персональный пинкод
8. МАК перебирает хранимые токены экспертов и пытается расшифровать их с помощью хэша пинкода эксперта
9. Если расшифрованный токен проходит проверку подлинности (без соли), токен принимается как токен эксперта. Аутентификация пройдена.
10. Полученный токен используется клиентом для аутентификации как на сервере МАК, так и на сопутствующих сервисах, таких как виртуальный узел сети блокчейн.

## Структура центра обработки данных АО НК

### Актуальная структура ЦОД

□

# Планируемая структура ЦОД

- 

# Функционирование НСК-НОК

Настоящий документ описывает принципы работы НАРК и его взаимодействие с СПК и ЦОК в системе НОК

- 
- 
- 
- 
- 
- 
- 
- 

# Функционирование СПК

Настоящий документ описывает порядок работы СПК и ЦОК с соискателями

- 
- 
-

# Новая система НСК

## Топология размещения и взаимодействия сервисов НСК

□

## Механизмы CI/CD

Базовые механизмы конвейера постоянной интеграции и развертывания программных продуктов АО "Национальные квалификации" состоят из следующих технологических этапов:

## Тестовая эксплуатация

1. Публикация изменений исходного кода
2. Запуск по расписанию (2 часа ночи) или по триггеру процесса сборки ПО
3. Запуск набора автотестов (юнит-тестирование)
4. Упаковка установочного пакета в формате debian
5. Упаковка установочного пакета в формате tarball+setup.sh
6. Передача собранного deb пакета в репозиторий ПО "Тестовая эксплуатация"
7. Установка или обновление собранного пакета в среде тестовой эксплуатации по расписанию на 5 утра по Московскому часовому поясу или по триггеру.

После успешного завершения всех этапов программное обеспечение доступно в среде тестовой эксплуатации. Если хотя бы один этап завершается с ошибкой, владельцу проекта отправляется отчет с уведомлением о возникшей ошибке.

## Опытная и промышленная эксплуатация

1. После успешного тестирования ПО в тестовом контуре тестировщик подписывает Акт тестирования. В Акте фигурирует версия ПО, включающая номер коммита (зафиксированных изменений) и тег версии ПО.
2. После подписания акта тег версии присваивается номеру коммита.
3. Запускается процесс сборки ПО с версией согласно тегу.
4. Формируется журнал изменений из событий между тегами. в журнал попадают только события отмеченные символами +/-/\* в начале каждой строки описания изменений.
5. Запускается набор автотестов (юнит-тестирование)
6. Упаковка установочного пакета в формате debian
7. Упаковка установочного пакета в формате tarball+setup.sh
8. Передача собранного deb пакета в репозиторий ПО "Опытная эксплуатация"
9. Отправка уведомлений кураторам и владельцам среды опытной эксплуатации об обновлении ПО включая список изменений.
10. Обновление ПО в среде опытной эксплуатации при помощи агента Puppet
11. После завершения тестирования ПО ведущий разработчик, технический директор или начальник отдела технической поддержки совместно с двумя кураторами регионов подписывают Акт тестирования. Всего устанавливается 3 подписи разными видами уполномоченных лиц.
12. После подписания акта тестирования ПО передается в репозиторий "Промышленная эксплуатация"
13. Отправка уведомлений кураторам и владельцам среды промышленной эксплуатации об обновлении ПО включая список изменений.

## Документация

Документация должна автоматически собираться на дату сборки и формирования пакетов в формате PDF с использованием системы справки и поддержки АО НК. Файл документации должен автоматически

формировать шапку, подвал, типографику, сноски, версию документа и иные параметры верстки и допечатной подготовки, согласно заданному шаблону.

# Сборка программного обеспечения

Для каждого разрабатываемого продукта АО НК должен формироваться Debian пакет программного обеспечения для целей быстрого точечного развертывания, контроля версионности, зависимостей и целостности распространяемого ПО.

Требования к формируемым пакетам:

1. Имя ПО является базовой частью имени пакета: `<software>`
2. Если пакет содержит компилируемую часть, то все компилируемые файлы должны входить в пакет `<software>-bin`
3. Если пакет содержит набор данных отдельно от программной части - изображения, шрифты, база данных, то все компоненты должны входить в пакет `<software>-data`
4. Если пакет содержит набор типовых настроек, такие настройки включаются в базовый пакет `<software>`, если вариантов настройки пакета несколько, создаются соответствующие конфигурационные пакеты вида: `<software>-config-<kind>`, где *kind* - разновидность предоставляемых настроек.
5. Документация на пакет ПО упаковывается в пакет `<software>-doc`
6. При наличии SDK, такие файлы должны помещаться в пакет `<software>-dev`
7. Если пакет имеет изменяемую структуру БД (миграции), то такие миграции должны быть включены в базовый пакет `<software>`
8. Версия пакета формируется автоматически из последнего тега + порядкового номера коммита относительно тега. ХЭШ коммита попадает в сведения о сборке пакета.
9. Список изменений проекта формируется автоматически из строк фиксации изменений кода в git, при этом учитываются только строки начинающиеся с символов `+/-/*`
10. Правила формирования пакетов описываются DevOps инженером совместно с руководителем проекта или тимлидом.

# Формирование Docker контейнера

Docker файл формирует образ программного обеспечения как результат сборки debian пакета. Docker файл должен предоставлять simple контейнер, из одного или нескольких образов ФС:

- Базовая ОС
- Необходимые зависимости
- Образ с распакованным ПО
- Сценарии при запуске контейнера (преинит)

Итоговый контейнер должен соответствовать следующим принципам:

1. **Монолитность** Отсутствие сборки ПО при установке контейнера
2. **Стабильность** Отсутствие скачивания дополнительных компонент из сети интернет
3. **Повторяемость** Возможность многократно развернуть контейнер в средах отличных от Docker
4. **Атомарность** Иметь отдельные образы контейнеров для нужд БД и рабочей нагрузки
5. **Масштабируемость** Иметь поддержку горизонтального масштабирования (NLB) с общей БД, разделяемой ФС и конфигураций
6. **Управляемость** Контейнер должен содержать базовый набор Unix Shell утилит, на уровне не ниже Busybox 1.7

# Репозиторий образов

Для целей поддержки инфраструктуры Kubernetes для программного обеспечения выделяется общий разделяемый репозиторий корпоративного ПО. Перед передачей ПО в Гостех или иные сторонние площадки оркестрации и управления ПО **должно** пройти все необходимые процедуры тестирования и приемки в среде **Тестовой** и **Опытной** эксплуатации.

# Аутентификация, Авторизация, Аудит

# Общие положения

Основной задачей системы AAA/IAM является управление базой учетных записей пользователей, выдача разрешений и аудит доступа к ресурсам связанным с IAM подсистем.

Основные механизмы аутентификации базируются на принципах протокола Kerberos v5 с расширениями (Claims), применительно к системам веб аутентификации (jwt), включая возможности делегирования, пересылки токена, двусторонней аутентификации, запроса расширений авторизации по сервисному имени принципала (SPN).

Использование механизмов, завязанных на принципала, с использованием технологии электронных подписей совместимых со стандартом JWT позволяют проводить следующие способы аутентификации пользователя:

- Аутентификация по логину-паролю (ключевая пара)
- Сквозная аутентификация по сертификатам электронной подписи
- Аутентификация по одноразовому ключу (SMS, OTP)
- Аутентификация по оффлановому токену (QR-код) без доступа к IAM
- Аутентификация по сохраненному токену (Progressive Web Application, HTML5 Offline Application)
- Прокси-аутентификация со стороннего сервиса (OAuth2/SAML 2.0)

Общий формат записи токена аутентификации:

```
{
  "principal": "User Principal Name",
  "uid": "GUID",
  "displayName": "User Published Name",
  "groups": [...],
  "claims": [...],
  "principalSignature": "jwt digest/RSA|ECDSA 2048bit",
  "signature": "jwt digest/RSA|ECDSA 2048bit"
}
```

Поля *Groups*, *Claims* и *PrincipalSignature* могут отсутствовать. В качестве обязательного идентификатора пользователя выступают поля *Principal* и *UID*

## Архитектура

Архитектура сервиса IAM предполагает наличие модульного механизма работы с поддержкой горизонтального масштабирования. Сервис IAM состоит из нескольких независимых масштабируемых сервисов:

- **AAA** — Сервис аутентификации, авторизации и аудита. Отвечает за выдачу токенов, производных токенов (для каждого SPN) и запись событий доступа.
- **Profile** — Сервис управления профилем пользователя, хранит сведения о профиле пользователя, позволяет хранить ряд дополнительных данных, таких как связанные документы, изображения, дополнительные реквизиты и т.п.
- **Permission** — Сервис управления доступом. Управляет видами групп доступа для каждого сервиса и ролями доступными пользователю.

□

## Общая схема работы OAuth2 сервиса:

Сервер поддерживает как обязательные (базовые) механизмы OAuth2, так и расширенные (опциональные) механизмы аутентификации и делегирования токена. В качестве расширений механизмов аутентификации вводятся дополнительные сущности - *UserPrincipalName*, *OrganizationPrincipalName* и *ServicePrincipalName*, где UPN и OPN относятся к категории субъекта аутентификации, а SPN - к категории конечной точки аутентификации. OAuth2 токен (Bearer) формируется по стандарту JWT и может использоваться в оффлайн операциях в течение всего срока (времени жизни) такого токена.

При прохождении процедуры аутентификации в системе цифрового паспорта пользователь получает базовый цифровой ключ доступа в виде JWT токена, кодированного в Base64 для передачи по техническим каналам связи или в открытом виде для формирования оффлайн QR кода. Допускается использовать формирование QR кода на основе Base64 кодированного JWT токена, для поддержки данного требования рекомендуется использовать следующий алгоритм:

1. Попытка расшифровать JSON
2. Если расшифровать JSON не удастся (ошибка в 1 символе) - производится декодирование Base64 в

- строку
- 3. Осуществляется повторная попытка декодирования JSON
- 4. Проверяется корректность подписи JWT
- 5. Проверяется срок действия JWT

После того как базовый токен аутентификации получен, он может использоваться для доступа к различным сервисам, с минимальным уровнем гарантированных привилегий. В случае если сервис поддерживает механизмы работы оффлайн, он может содержать реплику групп безопасности сервиса для аутентификации пользователя. Для получения полного доступа к ресурсам после процедуры аутентификации производится:

1. Выбор роли пользователя в системе - присвоение UID или OID и расширенных атрибутов (расширенный JWT токен) - утверждений
2. Расширенная аутентификация в сервисе - получение SPN для запрашиваемого ресурса
3. Получение специального токена для запрашиваемого ресурса по его SPN

Общий набор полей всех JWT токенов является одинаковым, но для каждого типа токенов характерны выделенные специальные поля. Состав и форматы полей JWT токенов описаны ниже.

Набор функций сервиса зависит от совпадения набора требований и утверждений сервиса и JWT токена для доступа к тому или иному функционалу приложения. Любой полученный токен имеет свой срок действия, может быть обновлен в пределах окна обновления и содержит валидную (проверяемую) цифровую подпись системы цифрового паспорта АО НК. Любой токен может быть использован оффлайн для аутентификации пользователя, например для каждого сервиса может быть сгенерирован отдельный временный оффлайн токен доступа в виде QR кода. Стандартный период действия токена составляет 2 часа, базового токена - 8 часов. При необходимости может быть запрошен токен расширенного срока действия, но не более 168 часов (7 дней) с момента получения такого токена.

При делегировании токена нижестоящей системе, промежуточный узел может как передать токен в режиме "как есть" (прокси-токен) или получить на его основании новый SPN токен для конечного сервиса. В этом случае для получения нового SPN сервис обращается к системе цифрового паспорта за делегированием токена передавая:

- Токен пользователя (SPN JWT)
- Адрес конечного сервиса

Срок действия делегированного токена сервиса не должен превышать срок использования оригинального токена пользователя.

Возможность продления истекшего JWT токена не допускается, из такого токена могут быть получены основные реквизиты для запроса подтверждения (аутентификации) со стороны пользователя с целью выпуска нового JWT токена с теми же условиями (Владелец, Организация, Сервис).

# Попроектная кадрово-организационная структура

□  
□

## Технические требования на систему управления задачами

## Цели создания системы

Основной целью создания системы является систематизация управленческих и рабочих задач в рамках существующих процессов и реализующихся проектов компании. Сопутствующей целью создания системы - является прослеживаемость выполняемых задач сотрудниками компании для оптимизации трудовых ресурсов.

# Вводимые термины и определения

В рамках настоящей технической спецификации вводятся основные термины и определения:

- **Процесс** — Решаемая глобальная задача имеющая одну или несколько целей, несколько исполнителей, характеризующаяся отсутствием конечного срока и набора подзадач. Подзадачи процесса формируются по мере поступления информации от участников такого процесса или являются следствием выполнения других (связанных) подзадач. Каждая подзадача процесса всегда направлена на достижение конкретной цели и имеет свой приоритет выполнения. Может затрагивать проекты и формировать дополнительные промежуточные цели проектов.
- **Проект** — Любое разрабатываемое решение, такое как: программное обеспечение, аппаратный комплекс, инженерное сооружение, законодательный акт или иной субъект права собственности, являющийся конечным продуктом проекта. Проект всегда завершается созданием артефакта в форме нематериального актива или основного средства. Проект может быть заранее подвергнут процессу декомпозиции на промежуточные цели, задачи, подзадачи, проектные команды. В рамках проекта формируются и закрепляются артефакты, такие как технические спецификации, технические задания, архитектурные планы, дизайн-макеты и иные сопутствующие документы необходимые для целей реализации проекта.
- **Промежуточная цель** — Этап *Процесса* или разработки *Проекта* имеющий собственное описание, точные или оценочные сроки выполнения цели и набор вводных и результирующих документов (артефактов) цели.
- **Задача** — Сформулированное конечное требование для одного или нескольких исполнителей. Задача имеет следующий набор атрибутов:
  - Наименование задачи
  - Описание задачи
  - Вложенные документы
  - Дата постановки задачи
  - Дедлайн задачи
  - Состояние задачи
  - Приоритет задачи
  - Проект/Процесс задачи
  - Промежуточная цель задачи
  - Автор задачи
  - Ответственный по задаче
  - Исполнители задачи
  - Родительская задача

Задача может иметь декомпозицию в виде дерева подзадач, для каждой из которых могут быть назначены свои ответственные и исполнители. Список исполнителей и документов всех подзадач всегда отражается в родительской задаче.

- **Календарь** — Датированное представление задач в разрезе сохраненного фильтра, имеющий ссылку для доступа по протоколу CalDAV для чтения расписания и управления задачами. Цели обозначаются выделенной датой на календаре специальной рамкой и цветом.
- **KanBan Доска** — Постолбцовое представление задач в разрезе сохраненного фильтра. Столбец определяется условием группировки задач, может быть одним из атрибутов задачи (Состояние, Приоритет, Проект/Процесс, Цель, Автор, Ответственный, Дата постановки, Дедлайн), при этом для дат используется округление до недели, месяца или квартала по выбору пользователя. Задачи могут быть отсортированы по дате постановки, дедлайну, состоянию, приоритету, автору или ответственному. Цель выводится как одно из полей задачи.
- **Диаграмма Ганта** — Линейное представление задач в разрезе сохраненного фильтра. Данная диаграмма имеет вид хронологического горизонтального представления календаря, разделенного на Годы, Кварталы, Месяцы, Недели, Дни недели, где задачи представлены в виде плашек, распространяющихся от даты постановки задачи до её дедлайна или назначенной промежуточной цели. Порядок пересекающихся задач, приходящихся на один период времени может быть отсортирован по дате постановки, состоянию, приоритету, автору или ответственному. Цели отображаются в виде сплошных вертикальных плашек, затрагивающих дату завершения промежуточной цели.
- **Список задач** — Представление задач в виде таблицы с указанием наименования задачи, автора, ответственного, исполнителей, цели, приоритета и сроков выполнения задачи. Задачи выстраиваются в рамках иерархии подзадач. Сортировка может осуществляться по любому столбцу с учетом иерархии задач. Задачи могут быть сгруппированы аналогично KanBan доске.
- **Дашборд** — Специальное представление, задач, имеющее набор настроек внешнего вида: Гант, Доска, Календарь, Список; а так же набора фильтров по статусу, сотрудникам, датам, целям, проектам/процессам, приоритету, состоянию. Набор фильтров и внешний вид дашборда может быть сохранен как публичное или приватное представление, а так же иметь публичную ссылку в режиме "Только для чтения". Публичное представление может быть назначено на группу пользователей. При переключении настроек внешнего вида параметры фильтра и группировки сохраняются и применяются к выбранному внешнему виду.
- **Группа пользователей** — Выделенная группа пользователей, имеющих одинаковые настройки интерфейса, перечень доступных проектов и процессов, представлений и набор привилегий. Привилегии назначенные для разных групп пользователей суммируются для каждого конкретного пользователя, в случае их пересечения. Группа пользователей может иметь фильтр автодобавления, при котором все новые пользователи, соответствующие критериям фильтра автоматически будут добавлены в группу (автопополняемая группа).
- **Пользователь** — Сотрудник организации или внешний контрагент прошедший процедуру аутентификации и входящий в одну или несколько групп пользователей. Если пользователь не имеет ни одной назначенной группы, такой пользователь относится к служебной группе "Гости" и может просматривать только публичные представления.



- **Отчет** — Формируемый по расписанию срез задач, согласно заданному фильтру задач, содержит ссылку на интерактивный *дашборд* с применением указанного фильтра, в котором можно изменять внешний вид задач: Список, Календарь, Гант, Доска. Отчет может быть направлен по любому каналу уведомлений.
- **Уведомления** — Системное сообщение отправляемое по одному или нескольким каналам связи: Мгновенное сообщение (Telegram), Электронная почта, Push уведомление и т.п. Уведомления вызываются по триггеру события. Типы получаемых уведомлений настраиваются для каждого пользователя индивидуально. Уведомления по событиям могут аккумулироваться для пользователя, и отправляться одним сообщением, если для данного типа уведомлений указан период группировки сообщений и время отправки уведомления.
- **Журнал** — Системный журнал событий, включающий в себя действия пользователя, от момента аутентификации, до момента получения уведомления. Все операции изменения задачи пользователем попадают в журнал и могут быть отслежены как в отдельной задаче, так и формировать *уведомление* пользователя или группы по условию. По каждой задаче ведется контроль наличия изменений, относительно последней даты доступа пользователя к задаче и отображается пользователю путем цветового выделения задачи более ярким цветом или рамкой.
- **Чат** — Внутренняя система обмена мгновенными сообщениями. Чат является самостоятельным представлением, но может быть открыт в качестве виджета или отображаться в рамках задачи, к которой он относится. У каждой задачи может быть свой чат, который становится доступен всем участникам задачи (Автор, Ответственный, Исполнители) при отправке первого сообщения. Чаты группируются по проекту и задаче к которой они относятся. Для выполненных (завершенных) задач чат скрывается из общего списка чатов, список чатов включает в себя только активные/действующие задачи пользователя, связанного с такой задачей. Количество новых сообщений отображается в виде цифры в круглом блоке поверх соответствующей задачи, в заголовке страницы и в окне чата для каждого чата соответственно.

# Технические требования к реализации системы

Система управления задачами должна обладать набором следующих обязательных характеристик:

- Аутентификация
  - Доменная
  - OAuth2
- Реактивность (Интерфейс зависит от поступаемых данных и отрисовывается на стороне клиента)
- Интерактивность (Все изменения задач в системе отражаются в виде событий и доставляются активным сессиям пользователей)
- Адаптивность (Возможность взаимодействия с основными функциями интерфейса как на мобильных устройствах, так и на ПК)
- Масштабируемость (Возможность запуска множества экземпляров Backend и Frontend серверов с общим центральным хранилищем, например с использованием брокера RabbitMQ Stream)
- Поддерживаемость (Наличие сборочного сценария и документации на систему разработки и сборки проекта)
- Переносимость (Возможность развертывания в виде OnPermissee или облачного решения на стороне конечного заказчика)

# Старая система НСК

## ЦОК

Данная модель описывает REST-API интерфейс сервиса центра оценки квалификаций

### Экзамены

#### Сохранение нового ресурса/видеозаписи в хранилище

Строка запроса **POST**: /api/v1/exam/store

##### Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор экзамена
user_id	Числовой	Идентификатор пользователя
startedAt	Дата и время в формате 'Y-m-d H:i:s'	Дата-время начала экзамена
endedAt	Дата и время в формате 'Y-m-d H:i:s'	Дата-время окончания экзамена

##### Варианты ответа:

###### Ошибка

```
{
  "error": "Текст ошибки",
  "success": false
}
```

###### Успешно

```
{
  "success": true
}
```

#### Получение данных ресурса из хранилища

Строка запроса **POST**: /api/v1/exam/{id}/show

##### Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор экзамена (часть запроса)

##### Варианты ответа:

###### Ошибка

```
{
  "error": "Текст ошибки",
  "success": false
}
```

###### Успешно

```
{
  "exam_id": "Идентификатор экзамена",
  "status": "Состояние экзамена",
}
```

```
{
  "stage_name": "Расшифровка состояния сдачи экзамена"
}
```

## Обновление сведений о ресурсе в хранилище

Строка запроса **POST**: /api/v1/exam/{id}/update

### Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор экзамена <i>(часть запроса)</i>
user_id	Числовой	Идентификатор пользователя
startedAt	Дата и время в формате 'Y-m-d H:i:s'	Дата-время начала экзамена
endedAt	Дата и время в формате 'Y-m-d H:i:s'	Дата-время окончания экзамена

### Варианты ответа:

#### Ошибка

```
{
  "error": "Текст ошибки",
  "success": false
}
```

#### Успешно

```
{
  "success": true
}
```

## Удаление ресурса из хранилища

Строка запроса **POST**: /api/v1/exam/{id}/destroy

### Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор экзамена <i>(часть запроса)</i>

### Варианты ответа:

#### Ошибка

```
{
  "error": "Текст ошибки",
  "success": false
}
```

#### Успешно

```
{
  "success": true
}
```

## Эксперты

### Сохранение нового эксперта в хранилище

Строка запроса **POST**: /api/v1/experts/store

### Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор эксперта

## Варианты ответа:

### Ошибка

```
{
  "error": "Текст ошибки",
  "success": false
}
```

### Успешно

```
{
  "success": true
}
```

## Получение данных эксперта по его идентификатору

Строка запроса **POST**: /api/v1/experts/{id}/show

### Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор эксперта (часть запроса)

## Варианты ответа:

### Ошибка

```
{
  "error": "Текст ошибки",
  "success": false
}
```

### Успешно

```
{
  "id": "Идентификатор эксперта",
  "code": "Код дл авторизации эксперта",
  "is_on": "Вход разрешен",
  "user": [
    {
      "id": "Иднетификатор пользователя",
      "full_name": "ФИО пользователя"
    }
  ],
  "units": [
    {
      "id": "Идентификатор ЦОК для проведения экзамена",
      "name": "Наименование ЦОК",
      "is_operator": "Является оператором квалификации",
      "is_main": "Является основным для квалификации",
    }
  ],
  "skills": [
    {
      "id": "Идентификатор квалификации",
      "name": "Наименование квалификации",
      "code": "Код квалификации",
      "is_tech": "Является технической",
      "is_prof": "Является профессиональной"
    }
  ]
}
```

## Обновление данных эксперта по его идентификатору

Строка запроса **POST**: /api/v1/experts/{id}/update

### Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор эксперта <i>(часть запроса)</i>

### Варианты ответа:

#### Ошибка

<pre>{   "error": "Текст ошибки",   "success": false }</pre>
--

#### Успешно

<pre>{   "success": true }</pre>
----------------------------------

## Удаление эксперта по его идентификатору

Строка запроса **POST**: /api/v1/experts/{id}/destroy

### Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор эксперта <i>(часть запроса)</i>

### Варианты ответа:

#### Ошибка

<pre>{   "error": "Текст ошибки",   "success": false }</pre>
--

#### Успешно

<pre>{   "success": true }</pre>
----------------------------------

## Временные интервалы

### Сохранение нового интервала в хранилище

Строка запроса **POST**: /api/v1/intervals/store

### Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор интервала

### Варианты ответа:

#### Ошибка

<pre>{   "error": "Текст ошибки",   "success": false }</pre>
--

#### Успешно

```
{
  "success": true
}
```

## Получение данных интервала по его идентификатору

Строка запроса **GET**: /api/v1/intervals/{id}/show

### Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор интервала (часть запроса)

### Варианты ответа:

#### Ошибка

```
{
  "error": "Текст ошибки",
  "success": false
}
```

#### Успешно

```
{
  "data": [
    {
      "id": "Идентификатор интервала",
      "start": "Дата-время начала экзамена 'строка Y-m-d H:i:s'",
      "end": "Дата-время окончания экзамена [строка Y-m-d H:i:s]",
      "tz": "Часовой пояс",
      "room_id": "Код учебного класса/ЭП",
      "skill_id": "Код квалификации",
      "skill_name": "Наименование квалификации",
      "stage_id": "Код стадии сдачи экзамена",
      "stage_name": "Расшифровка состояния сдачи экзамена",
      "stage_time": "Время сохранения состояния",
      "mac": "Идентификатор МАк или null",
      "experts": [
        {
          "id": "Идентификатор эксперта",
          "code": "Код входа эксперта",
          "user_id": "Идентификатор пользователя",
          "surname": "Фамилия эксперта",
          "name": "Имя эксперта",
          "patronymic": "Отчество эксперта"
        }
      ],
      "exams": [
        {
          "id": "Идентификатор экзамена",
          "user_id": "Идентификатор пользователя [соискателя] для которого проводится экзамен",
          "surname": "Фамилия эксперта",
          "name": "Имя эксперта",
          "patronymic": "Отчество эксперта",
        }
      ]
    }
  ]
}
```

## Обновление данных интервала по его идентификатору

Строка запроса **POST**: /api/v1/intervals/{id}/update

### Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор интервала ( <i>часть запроса</i> )

Варианты ответа:

Ошибка

<pre>{   "error": "Текст ошибки",   "success": false }</pre>
--

Успешно

<pre>{   "success": true }</pre>
----------------------------------

Удаление интервала по его идентификатору

Строка запроса **POST**: /api/v1/intervals/{id}/destroy

Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор интервала ( <i>часть запроса</i> )

Варианты ответа:

Ошибка

<pre>{   "error": "Текст ошибки",   "success": false }</pre>
--

Успешно

<pre>{   "success": true }</pre>
----------------------------------

Вопросы

Сохранение нового вопроса в хранилище

Строка запроса **POST**: /api/v1/questions/store

Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор вопроса

Варианты ответа:

Ошибка

<pre>{   "error": "Текст ошибки",   "success": false }</pre>
--

Успешно

<pre>{</pre>
--------------

```
{
  "success": true
}
```

## Получение вопроса по его идентификатору

Строка запроса **GET**: /api/v1/questions/{id}/show

### Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор вопроса ( <i>часть запроса</i> )

### Варианты ответа:

#### Ошибка

```
{
  "error": "Текст ошибки",
  "success": false
}
```

#### Успешно

```
{
  "id": "Идентификатор блока вопросов",
  "name": "Наименование блока вопросов",
  "skill_id": "Идентификатор квалификации",
  "questions": [
    {
      "id": "Идентификатор вопроса",
      "text": "Текст вопроса",
      "img": "Ссылка на изображение к вопросу",
      "question_type_id": "Идентификатор типа вопроса",
      "question_type": "Расшифровка типа вопроса",
      "one": "Признак единственного ответа",
      "variants": [
        {
          "id": "Идентификатор варианта ответа",
          "text": "Текст варианта ответа",
          "img": "Ссылка на изображение к варианту ответа",
          "conf": "Подтверждение выбора варианта вопроса?",
          "ans": "Ответ к варианту вопроса или признак правильного ответа?"
        }
      ]
    }
  ]
}
```

## Обновление вопроса по его идентификатору

Строка запроса **POST**: /api/v1/questions/{id}/update

### Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор вопроса ( <i>часть запроса</i> )

### Варианты ответа:

#### Ошибка

```
{
  "error": "Текст ошибки",
  "success": false
}
```

#### Успешно



```
{
  "success": true
}
```

## Удаление вопроса по его идентификатору

Строка запроса **POST**: /api/v1/questions/{id}/destroy

### Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор вопроса ( <i>часть запроса</i> )

### Варианты ответа:

#### Ошибка

```
{
  "error": "Текст ошибки",
  "success": false
}
```

#### Успешно

```
{
  "success": true
}
```

## Квалификации

### Сохранение новой квалификации в хранилище

### Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор квалификации

### Варианты ответа:

#### Ошибка

```
{
  "error": "Текст ошибки",
  "success": false
}
```

#### Успешно

```
{
  "success": true
}
```

### Получение данных квалификации по её идентификатору

Строка запроса **GET**: /api/v1/skills/{id}/show

### Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор квалификации ( <i>часть запроса</i> )

### Варианты ответа:

#### Ошибка

```
{
  "error": "Текст ошибки",
  "success": false
}
```

#### Успешно

```
{
  "id": "Идентификатор квалификации",
  "name": "Наименование квалификации",
  "code": "Код квалификации",
  "created_at": "Дата-время создания квалификации",
  "updated_at": "Дата-время обновления квалификации",
  "acts": [
    {
      "id": "Идентификатор документа",
      "name": "Наименование документа",
      "file": "Ссылка на файл документа",
      "info": "Сведения о документе"
    }
  ]
}
```

## Обновление данных квалификации по её идентификатору

Строка запроса **POST**: /api/v1/skills/{id}/update

### Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор квалификации ( <i>часть запроса</i> )

### Варианты ответа:

#### Ошибка

```
{
  "error": "Текст ошибки",
  "success": false
}
```

#### Успешно

```
{
  "success": true
}
```

## Удаление квалификации по её идентификатору

Строка запроса **POST**: /api/v1/skills/{id}/destroy

### Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор квалификации ( <i>часть запроса</i> )

### Варианты ответа:

#### Ошибка

```
{
  "error": "Текст ошибки",
  "success": false
}
```

Успешно

```
{
  "success": true
}
```

## Состояния

### Сохранение нового состояния в хранилище

Строка запроса **POST**: /api/v1/states/store

#### Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор состояния

#### Варианты ответа:

Ошибка

```
{
  "error": "Текст ошибки",
  "success": false
}
```

Успешно

```
{
  "success": true
}
```

### Получение состояния по его идентификатору

Строка запроса **GET**: /api/v1/states/{id}/show

[size:18](#)Параметры запроса:

#### Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор состояния*

#### Варианты ответа:

Ошибка

```
{
  "error": "Текст ошибки",
  "success": false
}
```

Успешно

```
{
  "success": true
}
```

### Обновление состояния по его идентификатору

Строка запроса **POST**: /api/v1/states/{id}/update

#### Параметры запроса:

Параметр	Значение	Описание
----------	----------	----------

Параметр	Значение	Описание
id	Числовой	Идентификатор состояния ( <i>часть запроса</i> )

### Варианты ответа:

#### Ошибка

<pre>{   "error": "Текст ошибки",   "success": false }</pre>
--

#### Успешно

<pre>{   "success": true }</pre>
----------------------------------

### Удаление состояния по его идентификатору

Строка запроса **POST**: /api/v1/states/{id}/destroy

### Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор состояния ( <i>часть запроса</i> )

### Варианты ответа:

#### Ошибка

<pre>{   "error": "Текст ошибки",   "success": false }</pre>
--

#### Успешно

<pre>{   "success": true }</pre>
----------------------------------

## Модули

### Сохранение нового ЦОК в хранилище

Строка запроса **POST**: /api/v1/units/store

### Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор ЦОК

### Варианты ответа:

#### Ошибка

<pre>{   "error": "Текст ошибки",   "success": false }</pre>
--

#### Успешно

<pre>{</pre>
--------------

```
{
  "success": true
}
```

## Получение данных ЦОК по его идентификатору

Строка запроса **POST**: /api/v1/units/{id}/show

### Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор ЦОК (часть запроса)

### Варианты ответа:

#### Ошибка

```
{
  "error": "Текст ошибки",
  "success": false
}
```

#### Успешно

```
{
  "id": "Идентификатор ЦОК",
  "code": "Код ЦОК",
  "name": "Краткое наименование ЮЛ ЦОК",
  "full_name": "Полное наименование ЮЛ ЦОК",
  "inn": "ИНН ЦОК",
  "ogrn": "ОГРН ЦОК",
  "legal_address": "Юридический адрес",
  "kpp": "Код причины постановки на учет",
  "region_id": "Код региона ИКАО",
  "region": "Наименование региона",
  "places": [
    {
      "id": "Идентификатор площадки",
      "name": "Наименование площадки",
      "timezone": "Часовой пояс",
      "city": "Город",
      "address": "Адрес площадки",
      "rooms": [
        {
          "id": "Идентификатор помещения",
          "name": "Наименование помещения",
        }
      ]
    }
  ]
}
```

## Обновление данных ЦОК по его идентификатору

Строка запроса **POST**: /api/v1/units/{id}/update

### Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор ЦОК (часть запроса)

### Варианты ответа:

#### Ошибка

```
{
  "error": "Текст ошибки",
  "success": false
}
```

Успешно

```
{
  "success": true
}
```

## Удаление ЦОК по его идентификатору

Строка запроса **POST**: /api/v1/units/{id}/destroy

Параметры запроса:

Параметр	Значение	Описание
id	Числовой	Идентификатор ЦОК (часть запроса)

Варианты ответа:

Ошибка

```
{
  "error": "Текст ошибки",
  "success": false
}
```

Успешно

```
{
  "success": true
}
```

# Мобильный автономный комплекс

Данная модель описывает REST-API интерфейс сервиса мобильного автономного комплекса

## Авторизация

Аутентификация в сервисе МАК и получение **токена** сессии

Строка запроса **POST**: /token

Параметры запроса:

Параметр	Значение	Описание
grant_type	Список (enum)	Тип авторизации, по умолчанию <i>password</i>
username	Текстовый	Логин пользователя
password	Текстовый	Пароль пользователя
scope	Текстовый	Область видимости
client_id	Числовой	Идентификатор клиента
client_secret	Текстовый	Ключ доступа клиента

Варианты ответа:

Ошибка 401

```
{
  "detail": "UNAUTHORIZED"
}
```

Ошибка 422

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    }
  ]
}
```

Успешно

```
{
  "access_token": "Токен аутентификации",
  "token_type": "bearer"
}
```

# Пользователь

## Получение информации о пользователе

Строка запроса **GET**: /api/v1/person/info

### Параметры запроса:

Не имеет параметров

### Варианты ответа:

Успешно

```
{
  "name": "Имя пользователя"
}
```

# Экзамен

## Получает список запланированных экзаменов

Строка запроса **GET**: /api/v1/exam/events

### Параметры запроса:

Параметр	Значение	Описание
dtfrom	Дата и время	Дата-время начала периода, по умолчанию - текущее время/
dtto	Дата и время	Дата-время окончания периода, по умолчанию - текущее время

### Варианты ответа:

Ошибка 422

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    }
  ]
}
```

Успешно

```
[
  {
    "id": "Идентификатор экзамена",
    "start": "Дата-время начала экзамена",
    "end": "Дата-время окончания экзамена",
    "title": "Название экзамена",
    "content": "Содержание/Описание экзамена",
    "class": "Вид принимаемого экзамена"
  }
]
```

Получает сведения о запланированном экзамене по его идентификатору

Строка запроса **GET**: /api/v1/exam/{exam\_id}

Параметры запроса:

Параметр	Значение	Описание
exam_id	Числовой	Идентификатор экзамена (часть запроса)

Варианты ответа:

Ошибка 422

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    }
  ]
}
```

Успешно

```
[
  {
    "id": "Идентификатор экзамена",
    "begin_utc": "Дата-время начала экзамена по гринвичу",
    "end_utc": "Дата-время окончания экзамена по гринвичу",
    "s_type": "Тип экзамена",
    "experts": [
      {
        "id": "Идентификатор эксперта",
        "name": "ФИО эксперта"
      }
    ],
    "student": {
      "id": "Идентификатор соискателя",
      "name": "ФИО соискателя"
    },
    "exam": "Экзамен"
  },
  "actions": [
    {
      "id": "Идентификатор действия эксперта",
      "a_type": "Тип действия",
      "created_utc": "Дата возникновения действия по гринвичу",
      "expert_id": "Идентификатор эксперта",
      "data": "Данные действия эксперта"
    }
  ]
}
```



# Получение списка действий экспертов запланированного экзамена по его идентификатору

Строка запроса **GET**: /api/v1/exam/{exam\_id}/action

## Параметры запроса:

Параметр	Значение	Описание
exam_id	Числовой	Идентификатор экзамена <i>(часть запроса)</i>

## Варианты ответа:

### Ошибка 422

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    }
  ]
}
```

### Успешно

```
[
  {
    "id": "Идентификатор действия эксперта",
    "a_type": "Тип действия",
    "created_utc": "Дата возникновения действия по гринвичу",
    "expert_id": "Идентификатор эксперта",
    "data": "Данные действия эксперта",
  }
]
```

# Добавление нового действия эксперта

Строка запроса **POST**: /api/v1/exam/{exam\_id}/practice/action

## Параметры запроса:

Параметр	Значение	Описание
exam_id	Числовой	Идентификатор экзамена <i>(часть запроса)</i>
a_type		Тип действия
data		Данные действия эксперта

## Варианты ответа:

### Ошибка 422

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    }
  ]
}
```

- Успешно //Нет данных ответа// }

# Получение состояния принимаемого экзамена

Строка запроса **GET**: /api/v1/exam/{exam\_id}/practice/viewstate

## Параметры запроса:

Параметр	Значение	Описание
exam_id	Числовой	Идентификатор экзамена (часть запроса)

Варианты ответа:

**Ошибка 422**

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    }
  ]
}
```

**Успешно**

```
{
  "state": "Состояние принимаемого экзамена",
  "task": "Задание",
  "task_lock": [ "Заблокированное задание" ],
  "wait_experts": [ "Идентификатор эксперта" ]
}
```

# Администрирование

## Получение статистики по проведенным экзаменам

Строка запроса **GET**: /api/v1/admin/stat

Параметры запроса:

Не имеет параметров

Варианты ответа:

**Успешно** Строковый результат

## Получает сведения об экзамене? квалификации? по идентификатору

Строка запроса **GET**: /api/v1/admin/test/{skill\_id}

Параметры запроса:

Параметр	Значение	Описание
skill_id	Числовой	Идентификатор экзамена? квалификации? (часть запроса)

Варианты ответа:

**Ошибка 422**

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    }
  ]
}
```

**Успешно** Строковый результат

# Завершение работы МАК

Строка запроса **POST**: /api/v1/admin/poweroff

Параметры запроса:

Не имеет параметров

Варианты ответа:

**Успешно** Строковый результат

## Синхронизация

Получение состояние синхронизации экзаменов

Строка запроса **GET**: /api/v1/sync/exams

Параметры запроса:

Не имеет параметров

Варианты ответа:

**Успешно** Строковый результат

# Модульный конструктор ПС

Модуль предназначен для разработки и определения:

- Стандартов квалификаций
- Профессиональных экзаменов
  - Теоретическая часть экзаменов
  - Практическая часть экзаменов
- Иные документы связанные с образованием и квалификациями

Модуль используется системами обработки и аналитики, проектом "Центр оценки квалификаций", порталом "Качество образования" и в рамках социальной сети "Я - эксперт".

## Ссылка для получения данных

Ссылка для выгрузки учебных оценочных средств в формате JSON:<https://ps.ao-nk.ru/os/edu/{id}> id - id оценочных средств

Ссылка на тестовое оценочное средство: <https://ps.ao-nk.ru/os/edu/69>

Информация об оценочном средстве (os)

Параметр	Значение	Описание
id	Числовое	id оценочного средства в системе
quest_count	Числовое	Количество блоков вопросов
total_ball	Числовое	Суммарное количество баллов в тесте
min_ball	Числовое	Пороговое количество баллов, от которого считается, что тест пройден успешно
time	Числовое	Время прохождения теста (в минутах)
created_at	Дата и время в формате 'Y-m-d H:i:s'	Дата создания
updated_at	Дата и время в формате 'Y-m-d H:i:s'	Дата изменения

Пример:

```
{
  "os": {
    "id": 69,
```

```
{
  "quest_count": 5,
  "total_ball": 100,
  "min_ball": 60,
  "time": 30,
  "created_at": "2022-02-03 16:31:16",
  "updated_at": "2022-02-04 12:29:25"
}
```

Информация о квалификации (skill)

Параметр	Значение	Описание
id	Числовое	id квалификации в системе
name	Текстовое	Название квалификации
level	Числовое	Уровень квалификации
num	Текстовое	Регистрационный номер квалификации

Пример:

```
{
  "skill": {
    "id": 57,
    "name": "Демо квалификация",
    "level": 6,
    "num": "12.00798.02"
  }
}
```

Информация о профессиональном стандарте (standart)

Параметр	Значение	Описание
id	Числовое	id профессионального стандарта в системе
name	Текстовое	Название профессионального стандарта
num	Текстовое	Регистрационный номер профессионального стандарта

Пример:

```
{
  "standart": {
    "id": 239,
    "name": "Демо профессиональный стандарт",
    "num": "1362"
  }
}
```

Информация о темах (трудовых функций) (themes)

Параметр	Значение	Описание
id	Числовое	id темы в системе (трудовой функции)
name	Текстовое	название темы (трудовой функции)

Пример:

```
{
  "themes": [
    {
      "id": 3675,
      "name": "Название ТФ"
    }
  ]
}
```

## Проверяемые знания (zn)

Параметр	Значение	Описание
id	Числовое	id знания в системе
name	Текстовое	Название знания
bloc_count		Количество вопросов для проверки этого знания в тесте
total_ball		Общее количество баллов, которые можно получить, если правильно ответить на все вопросы по этому знанию
created_at	Дата и время в формате 'Y-m-d H:i:s'	Дата создания
updated_at	Дата и время в формате 'Y-m-d H:i:s'	Дата изменения
blocs		Блоки вопросов (в каждом блоке минимум 3 варианта вопросов, при формировании теста из каждого блока берется 1 вопрос)

Пример:

```
{
  "zn": [
    {
      "id": 40346,
      "name": "Демо знание 1",
      "bloc_count": 2,
      "total_ball": 35,
      "created_at": "2022-02-03 16:31:16",
      "updated_at": "2022-02-03 17:51:44",
      "blocs": []
    }
  ]
}
```

## Проверяемые умения (um)

Параметр	Значение	Описание
id	Числовое	id умения в системе
name	Текстовое	Название умения
bloc_count		Количество вопросов для проверки этого умения в тесте
total_ball		Общее количество баллов, которые можно получить, если правильно ответить на все вопросы по этому умению
created_at	Дата и время в формате 'Y-m-d H:i:s'	Дата создания
updated_at	Дата и время в формате 'Y-m-d H:i:s'	Дата изменения
blocs		Блоки вопросов (в каждом блоке минимум 3 варианта вопросов, при формировании теста из каждого блока берется 1 вопрос)

Пример:

```
{
  "um": [
    {
      "id": 30117,
      "name": "Демо умение 1",
      "bloc": 1,
      "ball": 36,
      "created_at": "2022-02-03 16:31:16",
      "updated_at": "2022-02-04 11:38:29",
      "blocs": []
    }
  ]
}
```

## Блоки вопросов (blocs)

Параметр	Значение	Описание
id	Числовое	id умения в системе
created_at	Дата и время в формате 'Y-m-d H:i:s'	Дата создания
updated_at	Дата и время в формате 'Y-m-d H:i:s'	Дата изменения
quests	Массив	Вопросы (в каждом блоке минимум 3 варианта вопросов, при формировании теста из каждого блока берется 1 вопрос)

Пример:

```
{
  "blocs": [
    {
      "id": 413,
      "created_at": "2022-02-03 17:09:00",
      "updated_at": "2022-02-03 17:20:35",
      "quests": []
    }
  ]
}
```

## Вопрос с выбором ответа

Соискателю на выбор предоставляются варианты ответа, из которых он должен выбрать один или несколько. Если правильный ответ один, то в вопросе можно будет выбрать только один вариант ответа

### Вопросы(quests):

Параметр	Значение	Описание
id	Числовое	id вопроса в системе
text	Текстовое	Текст вопроса
img	Текстовое или null	Если вопрос с изображением, то поле будет содержать ссылку на изображение, иначе null
type_id	Числовое	id типа вопроса
type_name	Текстовое	Название типа вопроса
note	Текстовое	Пояснение к вопросу
created_at	Дата и время в формате 'Y-m-d H:i:s'	Дата создания
updated_at	Дата и время в формате 'Y-m-d H:i:s'	Дата изменения
variants		Варианты ответов

### Варианты ответов (variants)

Параметр	Значение	Описание
id	Числовое	id варианта в системе
text	Текстовое	Текст варианта ответа
img	Текстовоеили null	Если ответ с изображением, то поле будет содержать ссылку на изображение, иначе null
conf	Числовое	дополнительный параметр (в вопросах данного типа не используется)
ans	Числовое	Метка ответа (1 - правильный ответ, 0 - неправильный)
created_at	Дата и время в формате 'Y-m-d H:i:s'	Дата создания
updated_at	Дата и время в формате 'Y-m-d H:i:s'	Дата изменения
sort	Текстовое	Параметр для сортировки

Пример:

```
{
  "quests": [
    {
      "id": 1333,
```

```
"text": "Флаг какого государства представлен на изображении",
"img": "https://ps.ao-nk.ru/upload/quest/1333ou.png",
"type_id": 1,
"type_name": "с выбором ответа",
"note": "Государственный флаг Российской Федерации представляет собой прямоугольное полотнище из трёх равновеликих горизонтальных
полос: верхней — белого, средней — синего и нижней — красного цвета. Отношение длины флага к его ширине 2:3.",
"created_at": "2022-02-03 17:09:53",
"updated_at": "2022-02-03 17:11:43",
"variants": [
  {
    "id": 4838,
    "text": "Франция",
    "img": null,
    "conf": 0,
    "ans": 0,
    "sort": "a",
    "created_at": "2022-02-03 17:10:58",
    "updated_at": "2022-02-03 17:10:58"
  },
  {
    "id": 4839,
    "text": "Россия",
    "img": null,
    "conf": 0,
    "ans": 1,
    "sort": "b",
    "created_at": "2022-02-03 17:11:05",
    "updated_at": "2022-02-03 17:11:05"
  },
  {
    "id": 4840,
    "text": "Словакия",
    "img": null,
    "conf": 0,
    "ans": 0,
    "sort": "c",
    "created_at": "2022-02-03 17:11:34",
    "updated_at": "2022-02-03 17:11:34"
  }
]
}
```

## Вопрос на установление последовательности

Соискателю необходимо выстроить в правильном порядке все варианты ответа

### Вопросы(quests):

Параметр	Значение	Описание
id	Числовое	id вопроса в системе
text	Текстовое	Текст вопроса
img	Текстовое или null	Если вопрос с изображением, то поле будет содержать ссылку на изображение, иначе null
type_id	Числовое	id типа вопроса
type_name	Текстовое	Название типа вопроса
note	Текстовое	Пояснение к вопросу
created_at	Дата и время в формате 'Y-m-d H:i:s'	Дата создания
updated_at	Дата и время в формате 'Y-m-d H:i:s'	Дата изменения
variants		Варианты ответов

### Варианты ответов (variants)

Параметр	Значение	Описание
id	Числовое	id варианта в системе
text	Текстовое	Текст варианта ответа
img	Текстовоеили null	Если ответ с изображением, то поле будет содержать ссылку на изображение, иначе null
conf	Числовое	Дополнительный параметр (в вопросах данного типа не используется)
ans	Числовое	Правильный порядковый номер ответа
created_at	Дата и время в формате 'Y-m-d H:i:s'	Дата создания
updated_at	Дата и время в формате 'Y-m-d H:i:s'	Дата изменения
sort	Текстовое	Параметр для сортировки

Пример:

```
{
  "quests": [
    {
      "id": 2879,
      "text": "Установите последовательность стран в зависимости от их площади (от большего к меньшему) ",
      "img": null,
      "type_id": 3,
      "type_name": "на установление последовательности",
      "note": "Согласно данным",
      "created_at": "2022-02-03 16:53:54",
      "updated_at": "2022-02-03 16:54:30",
      "variants": [
        {
          "id": 10781,
          "text": "Канада",
          "img": null,
          "conf": 0,
          "ans": 2,
          "sort": "a",
          "created_at": "2022-02-03 16:54:11",
          "updated_at": "2022-02-03 16:54:11"
        },
        {
          "id": 10780,
          "text": "Россия",
          "img": null,
          "conf": 0,
          "ans": 1,
          "sort": "b",
          "created_at": "2022-02-03 16:54:05",
          "updated_at": "2022-02-03 16:54:11"
        },
        {
          "id": 10782,
          "text": "Франция",
          "img": null,
          "conf": 0,
          "ans": 3,
          "sort": "c",
          "created_at": "2022-02-03 16:54:17",
          "updated_at": "2022-02-03 16:54:17"
        },
        {
          "id": 10784,
          "text": "Исландия",
          "img": null,
          "conf": 0,
          "ans": 4,
          "sort": "d",
          "created_at": "2022-02-03 16:54:30",
          "updated_at": "2022-02-03 16:54:30"
        }
      ]
    }
  ]
}
```



```
    }
  ]
}
]
}
```

## Вопрос на установление соответствия

Соискателю необходимо сопоставить все варианты из левого столбца с вариантами из правого. Каждый вариант из левого столбца обязательно соответствует одному варианту из правого.

### Вопросы(quests):

Параметр	Значение	Описание
id	Числовое	id вопроса в системе
text	Текстовое	Текст вопроса
img	Текстовое или null	Если вопрос с изображением, то поле будет содержать ссылку на изображение, иначе null
type_id	Числовое	id типа вопроса
type_name	Текстовое	Название типа вопроса
note	Текстовое	Пояснение к вопросу
created_at	Дата и время в формате 'Y-m-d H:i:s'	Дата создания
updated_at	Дата и время в формате 'Y-m-d H:i:s'	Дата изменения
variants		Варианты ответов

### Варианты ответов (variants)

Параметр	Значение	Описание
id	Числовое	id варианта в системе
text	Текстовое	Текст варианта ответа
img	Текстовое или null	Если ответ с изображением, то поле будет содержать ссылку на изображение, иначе null
conf	Числовое	Если параметр равен 0 - то это элемент левого столбца, если равен 1 - то правого
ans	Числовое	Для элемента левого столбца параметр равен id элемента из правого столбца, у элемента правого столбца этот параметр равен 0
created_at	Дата и время в формате 'Y-m-d H:i:s'	Дата создания
updated_at	Дата и время в формате 'Y-m-d H:i:s'	Дата изменения
sort	Текстовое	Параметр для сортировки

Пример:

```
{
  "quests": [
    {
      "id": 2883,
      "text": "Установите соответствие между моделями процессоров и производителями",
      "img": null,
      "type_id": 2,
      "type_name": "на установление соответствия",
      "note": "Согласно данным производителей",
      "created_at": "2022-02-03 16:59:31",
      "updated_at": "2022-02-03 17:51:39",
      "variants": [
        {
          "id": 10801,
          "text": "Intel",
          "img": null,
          "conf": 1,
          "ans": 0,
          "sort": "1",

```

```
"created_at": "2022-02-03 16:59:39",
"updated_at": "2022-02-03 16:59:39"
},
{
  "id": 10802,
  "text": "AMD",
  "img": null,
  "conf": 1,
  "ans": 0,
  "sort": "2",
  "created_at": "2022-02-03 16:59:44",
  "updated_at": "2022-02-03 16:59:44"
},
{
  "id": 10803,
  "text": "Apple",
  "img": null,
  "conf": 1,
  "ans": 0,
  "sort": "3",
  "created_at": "2022-02-03 16:59:52",
  "updated_at": "2022-02-03 16:59:52"
},
{
  "id": 10804,
  "text": "Ryzen 9",
  "img": null,
  "conf": 0,
  "ans": 10802,
  "sort": "a",
  "created_at": "2022-02-03 17:00:20",
  "updated_at": "2022-02-03 17:00:20"
},
{
  "id": 10805,
  "text": "Core i9",
  "img": null,
  "conf": 0,
  "ans": 10801,
  "sort": "b",
  "created_at": "2022-02-03 17:00:48",
  "updated_at": "2022-02-03 17:00:48"
},
{
  "id": 10806,
  "text": "M1",
  "img": null,
  "conf": 0,
  "ans": 10803,
  "sort": "c",
  "created_at": "2022-02-03 17:00:57",
  "updated_at": "2022-02-03 17:00:57"
}
]
}
]
```

## Вопрос с открытым ответом

Соискателю необходимо самостоятельно ввести ответ в специальное поле или несколько полей

### Вопросы(quests):

Параметр	Значение	Описание
id	Числовое	id вопроса в системе

Параметр	Значение	Описание
text	Текстовое	Текст вопроса. В фигурных скобках заключаются правильные ответы (варианты правильных ответов разделяются вертикальной чертой " ")
img	Текстовое или null	Если вопрос с изображением, то поле будет содержать ссылку на изображение, иначе null
type_id	Числовое	id типа вопроса
type_name	Текстовое	Название типа вопроса
note	Текстовое	Пояснение к вопросу
created_at	Дата и время в формате 'Y-m-d H:i:s'	Дата создания
updated_at	Дата и время в формате 'Y-m-d H:i:s'	Дата изменения
variants		Варианты ответов (не используется в вопросах такого типа) варианты ответов содержатся в тексте вопроса

*Пример:*

```
{
  "quests": [
    {
      "id": 1337,
      "text": "Столицей Российской Федерации является город {Москва|Moscow}, а столицей Польши является город {Варшава|Warsaw}",
      "img": null,
      "type_id": 4,
      "type_name": "с открытым ответом",
      "note": "В соответствии с Конституцией Российской Федерации столицей Российской Федерации является город Москва.",
      "created_at": "2022-02-03 17:24:04",
      "updated_at": "2022-02-04 12:29:25",
      "variants": []
    }
  ]
}
```

# Сервис видеозаписи

Данная модель описывает REST-API интерфейс сервиса видеозаписи

## Ресурсы (входящие потоки)

### Получение списка ресурсов

Строка запроса **GET**: /api/v1/source

### Параметры запроса:

*Параметры отсутствуют*

### Варианты ответа:

**Успешно**

```
[
  {
    "id": "Идентификатор ресурса",
    "created_utc": "Дата-время создания ресурса",
    "status": "Состояние доступности ресурса",
    "uri_rtsp": "Адрес RTSP потока",
    "uri_http_picture": {
      "uri": "Адрес получения изображения",
      "auth": "basic или digest",
      "user": "Пользователь",
      "password": "Пароль"
    }
  }
]
```

```
{
  "description": "Описание ресурса",
  "ffmpeg_cmd": "Шаблон команды транскодирования потока"
}
```

## Создание ресурса

**POST:** /api/v1/source

### Параметры запроса:

Параметр	Значение	Описание
model	Текстовый	Модель ресурса <i>Source</i> или <i>PictureSource</i>
description	Текстовый	Описание ресурса
uri_rtsp	Строка URI	Адрес RTSP потока
uri_http_picture	Структура	Описание скриншота камеры
uri	Строка URI	Адрес получения изображения
auth	Перечисление	<i>basic</i> или <i>digest</i>
user	Текстовый	Пользователь
password	Текстовый	Пароль
ffmpeg_cmd	Текстовый	Шаблон команды транскодирования потока

### Варианты ответа:

#### Ошибка 422

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    }
  ]
}
```

#### Успешно

```
{
  "id": "Идентификатор ресурса",
  "created_utc": "Дата-время создания ресурса",
  "status": "Состояние доступности ресурса",
  "uri_rtsp": "Адрес RTSP потока",
  "uri_http_picture": {
    "uri": "Адрес получения изображения",
    "auth": "**basic* или *digest*",
    "user": "Пользователь",
    "password": "Пароль"
  },
  "description": "Описание ресурса",
  "ffmpeg_cmd": "Шаблон команды транскодирования потока"
}
```

## Получение ресурса

**GET:** /api/v1/source/{source\_id}/

### Параметры запроса:

Параметр	Значение	Описание
source_id	Числовой	Идентификатор ресурса ( <i>часть запроса</i> )

### Варианты ответа:

Ошибка 422

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    }
  ]
}
```

Успешно

```
{
  "id": "Идентификатор ресурса",
  "created_utc": "Дата-время создания ресурса",
  "status": "Состояние доступности ресурса",
  "uri_rtsp": "Адрес RTSP потока",
  "uri_http_picture": {
    "uri": "Адрес получения изображения",
    "auth": "**basic* или *digest*",
    "user": "Пользователь",
    "password": "Пароль"
  },
  "description": "Описание ресурса",
  "ffmpeg_cmd": "Шаблон команды транскодирования потока"
}
```

Изменение ресурса

PUT: /api/v1/source/{source\_id}/

Параметры запроса:

Параметр	Значение	Описание
source_id	Числовой	Идентификатор ресурса (часть запроса)
model	Текстовый	Модель ресурса
description	Текстовый	Описание ресурса
uri_rtsp	Текстовый	Адрес RTSP потока
uri_http_picture:	Структура URI	
uri	Текстовый	Адрес получения изображения
auth	Текстовый	basic или digest
user	Текстовый	Пользователь
password	Текстовый	Пароль
ffmpeg_cmd	Текстовый	Шаблон команды транскодирования потока

Варианты ответа:

Ошибка 422

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    }
  ]
}
```

Успешно

```
{
```

```
{
  "id": "Идентификатор ресурса",
  "created_utc": "Дата-время создания ресурса",
  "status": "Состояние доступности ресурса",
  "uri_rtsp": "Адрес RTSP потока",
  "uri_http_picture": {
    "uri": "Адрес получения изображения",
    "auth": "**basic* или *digest*",
    "user": "Пользователь",
    "password": "Пароль"
  },
  "description": "Описание ресурса",
  "ffmpeg_cmd": "Шаблон команды транскодирования потока"
}
```

## Удаление ресурса

**DELETE:** /api/v1/source/{source\_id}/

### Параметры запроса:

Параметр	Значение	Описание
source_id	Числовой	Идентификатор ресурса <i>(часть запроса)</i>

### Варианты ответа:

#### Ошибка 422

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    }
  ]
}
```

#### Успешно

```
{
  "id": "Идентификатор ресурса",
  "created_utc": "Дата-время создания ресурса",
  "status": "Состояние доступности ресурса",
  "uri_rtsp": "Адрес RTSP потока",
  "uri_http_picture": {
    "uri": "Адрес получения изображения",
    "auth": "**basic* или *digest*",
    "user": "Пользователь",
    "password": "Пароль"
  },
  "description": "Описание ресурса",
  "ffmpeg_cmd": "Шаблон команды транскодирования потока"
}
```

## Получение скриншота потока

Строка запроса **GET:** /api/v1/source/{source\_id}/picture

### Параметры запроса:

Параметр	Значение	Описание
source_id	Числовой	Идентификатор ресурса <i>(часть запроса)</i>

### Варианты ответа:

#### Ошибка 422

```
{
```

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    }
  ]
}
```

Успешно *Данные изображения*

## Получение скриншота потока

Строка запроса **GET**: /api/v1/source/{source\_id}/video

### Параметры запроса:

Параметр	Значение	Описание
source_id	Числовой	Идентификатор ресурса ( <i>часть запроса</i> )
delay	Числовой	Задержка между кадрами в секундах, по умолчанию 1 секунда

### Варианты ответа:

#### Ошибка 422

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    }
  ]
}
```

Успешно *Данные видеопотока mJPEG*

## Управление записью

### Получение состояния записи ресурсов по метке

Строка запроса **GET**: /api/v1/record

### Параметры запроса:

Параметр	Значение	Описание
tag_name	Текстовый	Идентификатор метки ресурсов

### Варианты ответа:

#### Ошибка 422

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    }
  ]
}
```

Успешно

```
[
```

```
{
  "id": "Идентификатор записи",
  "created_utc": "Дата-время создания записи",
  "status": "Состояние записи",
  "begin_utc": "Дата-время начала записи",
  "end_utc": "Дата-время окончания записи",
  "source": [
    {
      "id": "Идентификатор ресурса",
      "created_utc": "Дата-время создания ресурса",
      "status": "Состояние доступности ресурса",
      "uri_rtsp": "Адрес RTSP потока",
      "uri_http_picture": {
        "uri": "Адрес получения изображения",
        "auth": "**basic* или *digest*",
        "user": "Пользователь",
        "password": "Пароль"
      },
      "description": "Описание ресурса",
      "ffmpeg_cmd": "Шаблон команды транскодирования потока"
    }
  ],
  "hash_md5": "Контрольная сумма записи MD5",
  "location": "Путь хранения записи",
  "sync_location": "Путь для синхронизации видеозаписи с другим видеосервером"
}
```

## Остановка записи группы ресурсов по идентификатору метки

**PUT:** /api/v1/record

### Параметры запроса:

Параметр	Значение	Описание
model	Текстовый	Модель метки записи ресурса для изменений <i>Source</i> или <i>PictureSource</i>
tag_name	Текстовый	Идентификатор метки ресурсов

### Варианты ответа:

**Ошибка 422**

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    }
  ]
}
```

**Успешно** Нет данных

## Запуск записи группы ресурсов с меткой

**POST:** /api/v1/record

### Параметры запроса:

Параметр	Значение	Описание
model	Текстовый	Модель метки записи ресурса для изменений <i>Source</i> или <i>PictureSource</i>
tag_name	Текстовый	Идентификатор метки ресурсов
group_name	Текстовый	Идентификатор группы ресурсов



Варианты ответа:

Ошибка 422

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    }
  ]
}
```

Успешно Нет данных

Получение состояния записи ресурсов с меткой

Строка запроса **GET**: /api/v1/record/runing

Параметры запроса:

Параметр	Значение	Описание
tag_name	Текстовый	Идентификатор метки ресурсов

Варианты ответа:

Ошибка 422

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    }
  ]
}
```

Успешно Значение true или false

Получение записи ресурса

Строка запроса **GET**: /api/v1/record/{record\_id}/

Параметры запроса:

Параметр	Значение	Описание
record_id	Текстовый	Идентификатор записи ресурса (часть запроса)

Варианты ответа:

Ошибка 422

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    }
  ]
}
```

Успешно

```
{
```

```
{
  "id": "Идентификатор записи",
  "created_utc": "Дата-время создания записи",
  "status": "Состояние записи",
  "begin_utc": "Дата-время начала записи",
  "end_utc": "Дата-время окончания записи",
  "source": {
    "id": "Идентификатор ресурса",
    "created_utc": "Дата-время создания ресурса",
    "status": "Состояние доступности ресурса",
    "uri_rtsp": "Адрес RTSP потока",
    "uri_http_picture": {
      "uri": "Адрес получения изображения",
      "auth": "*basic* или *digest*",
      "user": "Пользователь",
      "password": "Пароль"
    },
    "description": "Описание ресурса",
    "ffmpeg_cmd": "Шаблон команды транскодирования потока"
  },
  "hash_md5": "Контрольная сумма записи MD5",
  "location": "Путь хранения записи",
  "sync_location": "Путь для синхронизации видеозаписи с другим видеосервером"
}
```

## Получение части файла записи

Строка запроса **GET**: /api/v1/record/{record\_id}/video/

### Параметры запроса:

Параметр	Значение	Описание
record_id	Текстовый	Идентификатор записи ресурса ( <i>часть запроса</i> )
range	Численный-Численный	Диапазон байт для загрузки, параметр заголовка запроса

### Варианты ответа:

#### Ошибка 422

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    }
  ]
}
```

**Успешно** *Двоичные данные*

## Удаление файла видеозаписи

**DELETE**: /api/v1/record/{record\_id}/file/

### Параметры запроса:

Параметр	Значение	Описание
record_id	Текстовый	Идентификатор записи ресурса ( <i>часть запроса</i> )
location	Текстовый	Путь к файлу видеозаписи

### Варианты ответа:

#### Ошибка 422

```
{
```

```
"detail": [
  {
    "loc": ["Положение ошибки"],
    "msg": "Сообщение ошибки",
    "type": "Тип ошибки"
  }
]
```

**Успешно**

```
{
  "id": "Идентификатор записи",
  "created_utc": "Дата-время создания записи",
  "status": "Состояние записи",
  "begin_utc": "Дата-время начала записи",
  "end_utc": "Дата-время окончания записи",
  "source": {
    "id": "Идентификатор ресурса",
    "created_utc": "Дата-время создания ресурса",
    "status": "Состояние доступности ресурса",
    "uri_rtsp": "Адрес RTSP потока",
    "uri_http_picture": {
      "uri": "Адрес получения изображения",
      "auth": "*basic* или *digest*",
      "user": "Пользователь",
      "password": "Пароль"
    },
    "description": "Описание ресурса",
    "ffmpeg_cmd": "Шаблон команды транскодирования потока"
  },
  "hash_md5": "Контрольная сумма записи MD5",
  "location": "Путь хранения записи",
  "sync_location": "Пусть для синхронизации видеозаписи с другим видеосервером"
}
```

# Управление группами камер

## Получение списка групп камер

Строка запроса **GET**: /api/v1/group

### Параметры запроса:

*Параметры отсутствуют*

### Варианты ответа:

#### Ошибка 422

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    }
  ]
}
```

**Успешно**

```
[
  {
    "id": "Идентификатор группы",
    "created_utc": "Дата-время создания группы",
    "name": "Имя группы"
  }
]
```

```
}
]
```

## Создание группы ресурсов

**POST:** /api/v1/group

### Параметры запроса:

Параметр	Значение	Описание
model	Текстовый	Модель метки записи ресурса для изменений <i>Source</i> или <i>PictureSource</i>
name	Текстовый	Наименование группы

### Варианты ответа:

#### Ошибка 422

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    }
  ]
}
```

#### Успешно

```
[
  {
    "id": "Идентификатор группы",
    "created_utc": "Дата-время создания группы",
    "name": "Имя группы"
  }
]
```

## Получение сведений о группе ресурсов

Строка запроса **GET:** /api/v1/group/{group\_id}/

### Параметры запроса:

Параметр	Значение	Описание
group_id	Текстовый	Идентификатор группы ресурсов ( <i>часть запроса</i> )

### Варианты ответа:

#### Ошибка 422

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    }
  ]
}
```

#### Успешно

```
{
  "id": "Идентификатор группы ресурсов",
  "created_utc": "Дата-время создания группы",
  "name": "Наименование группы",
}
```

```
"sources": [
  {
    "id": "Идентификатор ресурса",
    "created_utc": "Дата-время создания ресурса",
    "uri_rtsp": "Адрес RTSP потока",
    "uri_http_picture": {
      "uri": "Адрес получения изображения",
      "auth": "*basic* или *digest*",
      "user": "Пользователь",
      "password": "Пароль"
    },
    "description": "Описание ресурса",
    "ffmpeg_cmd": "Шаблон команды транскодирования потока"
  },
  ...
]
```

## Удаление группы ресурсов

**DELETE:** /api/v1/group/{group\_id}/

### Параметры запроса:

Параметр	Значение	Описание
group_id	Текстовый	Идентификатор группы ресурсов ( <i>часть запроса</i> )

### Варианты ответа:

**Ошибка 422**

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    },
    ...
  ]
}
```

**Успешно**

```
{
  "id": "Идентификатор группы ресурсов",
  "created_utc": "Дата-время создания группы",
  "name": "Наименование группы"
}
```

## Добавление ресурса в группу ресурсов

**POST:** /api/v1/group/{group\_id}/source/{source\_id}/

### Параметры запроса:

Параметр	Значение	Описание
group_id	Текстовый	Идентификатор группы ресурсов ( <i>часть запроса</i> )
source_id	Текстовый	Идентификатор ресурса ( <i>часть запроса</i> )

### Варианты ответа:

**Ошибка 422**

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      ...
    },
    ...
  ]
}
```

```
{
  "msg": "Сообщение ошибки",
  "type": "Тип ошибки"
}
```

Успешно Значение *true* или *false*

## Удаление ресурса из группы ресурсов

**DELETE:** /api/v1/group/{group\_id}/source/{source\_id}/

Параметры запроса:

Параметр	Значение	Описание
group_id	Текстовый	Идентификатор группы ресурсов ( <i>часть запроса</i> )
source_id	Текстовый	Идентификатор ресурса ( <i>часть запроса</i> )

Варианты ответа:

**Ошибка 422**

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    }
  ]
}
```

Успешно Значение *true* или *false*/\*

## Управление метками записей

### Получение метки записи ресурсов

Строка запроса **GET:** /api/v1/tag

Параметры запроса:

Параметр	Значение	Описание
name	Текстовый	Наименование метки

Варианты ответа:

**Ошибка 422**

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    }
  ]
}
```

Успешно

```
{
  "id": "Идентификатор тега",
  "created_utc": "Дата-время создания тега",
  "name": "Имя тега",
}
```

```
"records": [
  {
    "id": "Идентификатор записи",
    "created_utc": "Дата-время создания записи",
    "status": "Состояние записи",
    "begin_utc": "Дата-время начала записи",
    "end_utc": "Дата-время окончания записи",
    "source": {
      "id": "Идентификатор ресурса",
      "created_utc": "Дата-время создания ресурса",
      "uri_rtsp": "Адрес RTSP потока",
      "uri_http_picture": {
        "uri": "Адрес получения изображения",
        "auth": "**basic* или *digest*",
        "user": "Пользователь",
        "password": "Пароль"
      },
      "description": "Описание ресурса",
      "ffmpeg_cmd": "Шаблон команды транскодирования потока"
    },
    "hash_md5": "Контрольная сумма записи MD5",
    "location": "Путь хранения записи",
    "sync_location": "Пусть для синхронизации видеозаписи с другим видеосервером"
  }
]
```

Получить начатые метки

Строка запроса **GET**: /api/v1/tag/begin

Параметры запроса:

Параметры отсутствуют

Варианты ответа:

Успешно

["Наименование метки"]

Синхронизация данных

Начать синхронизацию видео

**POST**: /api/v1/sync

Параметры запроса:

Параметр	Значение	Описание
tag_name	Текстовый	Наименование метки
location	Текстовый	Расположение файла

Варианты ответа:

Ошибка 422

```
{
  "detail": [
    {
      "loc": ["Положение ошибки"],
      "msg": "Сообщение ошибки",
      "type": "Тип ошибки"
    }
  ]
}
```

**Успешно** Строковые данные

## Различные проверки

### Проверка изображения

Строка запроса **GET**: /test/img

#### Параметры запроса:

*Не имеет параметров*

#### Варианты ответа:

**Успешно** *Не имеет ответа*

### Проверка видео

Строка запроса **GET**: /test/video

#### Параметры запроса:

*Не имеет параметров*

#### Варианты ответа:

**Успешно** *Не имеет ответа*

## Алгоритмы синхронизации

□

□

□

□



# Образовательная платформа Афина

## Интеграция Афины с 1С

### Термины и их синонимы:

Справочник — Catalog

Регистр сведений — InformationRegister

Документ - Document

### Алгоритм создания сотрудника по ГПХ

#### 1. Общие запросы

- Отправляем get запрос в справочник ВидыДокументовФизическихЛиц, фильтр по PredefinedDataName == ПаспортРФ
  - Получаем ref\_key
- Отправляем get запрос в справочник Организации, фильтр по ИНН == ИННАккредАгенства
  - Получаем ref\_key
- Отправляем get запрос в справочник Валюты, фильтр Code == 643
  - Получаем ref\_key

#### 2. Создание физлица

- Отправляем get запрос в справочник ФизическиеЛица, фильтр по ИНН или СНИЛС
  - Если есть физическое лицо, получаем его ref\_key
  - Иначе создаем физическое лицо (post в ФизическиеЛица)

#### 3. Создание паспортных данных

- Отправляем get запрос в регистр сведений ДокументыФизическихЛиц, фильтр по Физлицо\_Key и ВидДокумента\_Key = (ref\_key Паспорт, шаг 2)
  - Если есть паспорт, получаем его ref\_key
  - Иначе создаем паспорт (post в ДокументыФизическихЛиц), поля Period = ДатаВыдачи, формат дат ISO

#### 4. Создание расчетного счета физлица

- Отправляем get запрос в справочник БанковскиеСчета, фильтр Owner = ref\_key физического лица и НомерСчета
  - Если есть счет физического лица, получаем его ref\_key
  - Иначе отправляем get запрос в справочник Банки, фильтр по Code = БИК
    - Если есть Банк — получаем его ref\_key
    - Иначе отправляем get запрос в справочник КлассификаторБанков, фильтр по Code = БИК
      - Создаем Банк (post в Банки) с данными полей из классификатора банков.
  - Создаем БанковскийСчет (post запрос в Банковские счета) полями Owner — ref\_key физического лица, Банк\_key — ref\_key Банка, ВалютаДенежныхСредств\_Key — ref\_key Валюты.

#### 5. Создание сотрудника

- Отправляем get запрос в справочник Сотрудники, фильтр по ФизическоеЛицо\_Key равный ref\_key физического лица
  - Если есть сотрудник, запоминаем его ref\_key
  - Если сотрудника нет
    - Создаем сотрудника (post в Сотрудники)
    - Создаем ассоциацию с физлицом (post запрос в регистр сведений ОсновныеСотрудникиФизическихЛиц)
    - Создаем правила выплаты (post запрос в регистр сведений МестаВыплатыЗарплатыСотрудников, значение поля "Вид" = "БанковскийСчет")
    - Создаем кадровые данные (post запрос в регистр сведений ТекущиеКадровыеДанныеСотрудников, значение поля "ГоловнаяОрганизация\_Key" = Организация.ref\_key, "ДатаПриема" = ISO(сегодня))

#### 2. Создаём договор подряда

- Отправляем get запрос в документ ДоговорРаботыУслуги, фильтр по полю Сотрудник\_Key равный ref\_key сотрудника
  - Если договор существует, запоминаем ref\_key договора
  - Если договор не существует, создаем договор (post запрос в Документы ДоговорРаботыУслуги)
    - Создаем сведения о заключении договора (post запрос в регистр сведений ФизическиеЛицаДокументов, значение поля "ДокументФизическогоЛица\_Type" = "StandardODATA.Document\_ДоговорРаботыУслуги")
    - Создаем сведения о договоре как основании для произведения начислений (post запрос в регистр сведений СоставДокументовЗарплатаКадры, значение поля "ДокументФизическогоЛица\_Type" = "StandardODATA.Document\_ДоговорРаботыУслуги")
  -

