

УТВЕРЖДЕН  
RU.82564873.КАРТОТЕКА 32-ЛЮ

**Программное обеспечение для обработки анкетных данных «Картотека»**

**Руководство системного программиста**

**RU.82564873.КАРТОТЕКА 32**

**Листов 48**

## **АННОТАЦИЯ**

Документ содержит руководство системного программиста программы для электронных вычислительных машин «Программное обеспечение для обработки анкетных данных «Картотека»» (далее – Программа), разработанного Обществом с ограниченной ответственностью «М-Софт Медикал Имаджинг».

«Картотека» предназначена для повышения качества и автоматизации операций проверки анкетных данных.

Основная часть документа содержит информацию по настройке Программы, включая описание действий по настройке Программы на условия конкретного применения, а также описание способов проверки, позволяющих дать общее заключение о работоспособности Программы.

Документ предназначен для персонала, ответственного за эксплуатацию Программы и содержит информацию, достаточную для подготовки Программы к использованию по функциональному назначению.

Документ разработан в соответствии с требованиями межгосударственных стандартов, входящих в комплекс стандартов, устанавливающих правила разработки, оформления и обращения программ и программной документации «Единая система программной документации».

Документ разработан согласно ГОСТ 19.503-79, структура и оформление документа соответствуют ГОСТ 19.105-78.

## СОДЕРЖАНИЕ

<b>1. Общие сведения о Программе</b> .....	<b>5</b>
1.1. Наименование .....	5
1.2. Назначение Программы .....	5
1.3. Функции Программы.....	5
1.4. Сведения о технических и программных средствах, обеспечивающих выполнение Программы.....	6
1.4.1. Сведения о технических средствах, обеспечивающих выполнение Программы .....	6
1.4.2. Сведения о программных средствах, обеспечивающих выполнение данной Программы .....	6
<b>2. Структура Программы</b> .....	<b>7</b>
2.1. Подкомпонент API клиентского приложения .....	7
2.2. Подкомпонент поиска ошибок и дублей.....	7
2.3. Подкомпонент хранения данных .....	8
2.4. Подкомпонент выполнения фоновых задач.....	8
2.5. Подкомпонент формирования истории изменений.....	8
2.6. Подкомпонент формирования файла для экспорта.....	8
2.7. Подкомпонент взаимодействия с API клиентского приложения .....	8
2.8. Подкомпонент импорта файлов .....	8
2.9. Подкомпонент работы с черновиками.....	8
2.10. Подкомпонент работы с Реестром анкет.....	9
2.11. Подкомпонент просмотра истории изменений.....	9
2.12. Подкомпонент управления настройками .....	10
<b>3. Настройка Программы</b> .....	<b>11</b>
3.1. Порядок настройки Программы:.....	11
3.2. Описание действий по настройке Программы .....	11
3.2.1. Подготовка окружения.....	11
3.2.2. Настройка Программы на условия применения .....	11
3.2.3. Запуск сервиса .....	12
3.3. Выбор функций Программы.....	13
<b>4. Проверка Программы</b> .....	<b>14</b>
<b>5. Дополнительные возможности</b> .....	<b>15</b>

<b>Перечень принятых терминов и сокращений .....</b>	<b>16</b>
<b>Приложение А. Описание API.....</b>	<b>17</b>
A.1 GET/questionnaires/{id}/lock .....	18
A.2 POST/questionnaires/batch_check .....	20
A.3 GET/questionnaires/{id}/unlock .....	22
A.4 POST/questionnaires/search.....	24
A.5 GET/questionnaires/draft.....	26
A.6 GET/questionnaires/published.....	27
A.7 PATCH/questionnaires/batch_update .....	29
A.8 PATCH/questionnaires/batch_destroy .....	32
A.9 GET/questionnaires/{id} .....	33
A.10 PUT/questionnaires/{id} .....	35
A.11 GET/questionnaires/autocomplete.....	39
A.12 GET/questionnaire_histories .....	41

## **1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ**

### **1.1. Наименование**

Полное наименование: программа для электронных вычислительных машин «Программное обеспечение для обработки анкетных данных «Картотека»».

Сокращенное наименование: «Картотека», Программа.

### **1.2. Назначение Программы**

Программа предназначена для повышения качества и автоматизации операций проверки анкетных данных и решает следующие задачи:

- 1) Импорт анкетных данных;
- 2) Экспорт анкетных данных;
- 3) Выявление и устранение ошибок в анкетных данных;
- 4) Выявление и устранение дублей в анкетных данных;
- 5) Ведение картотеки данных, прошедших проверку.

### **1.3. Функции Программы**

Для решения задач по повышению качества анкетных данных и автоматизации ручных операций проверки списков наблюдателей (черновиков) реализованы следующие функциональные возможности:

- 1) Гибкая настройка параметров импорта черновиков;
- 2) Обработка черновиков, включая поиск и фильтрацию;
- 3) Автоматическое выявление ошибок в черновиках;
- 4) Автоматический поиск дублирующей информации в черновиках;
- 5) Корректировка черновиков;
- 6) Экспорт черновиков;
- 7) Просмотр истории изменений черновиков.

## **1.4. Сведения о технических и программных средствах, обеспечивающих выполнение Программы**

### **1.4.1. Сведения о технических средствах, обеспечивающих выполнение Программы**

Для обеспечения использования Программы по функциональному назначению требуются технические средства в составе и с характеристиками, приведенными ниже.

Характеристики вычислительных ресурсов, предназначенных для функционирования серверной части Программы: количество серверов – 1. Основные технические характеристики: процессор — не менее 4 CPU, объем оперативной памяти (ОЗУ) — не менее 8 Гбайт, размер дискового пространства — не менее 50 Гбайт.

Клиентская часть технических средств должна быть представлена в виде устройств типа «ноутбук» или «настольный компьютер» с периферийным оборудованием ввода и вывода информации.

Характеристики клиентских рабочих станций: частота центрального процессора — 2.20 ГГц, ОЗУ — 4 Гбайта, размер дискового пространства — 50 Гбайт, диагональ экрана: 15".

### **1.4.2. Сведения о программных средствах, обеспечивающих выполнение данной Программы**

Установка Программы осуществляется с использованием технологии контейнеризации на базе программного обеспечения Docker.

Программные средства, используемые серверной частью Программы:

- Операционная система — Ubuntu;
- Веб-сервер — Nginx;
- СУБД — Postgres.

## **2. СТРУКТУРА ПРОГРАММЫ**

Программа реализована в виде двух компонентов:

- Компонент «Серверное приложение»;
- Компонент «Клиентское приложение».

Компоненты состоят из следующих подкомпонентов:

- Подкомпонент API (см. п. 2.1);
- Подкомпонент поиска ошибок и дублей (см. п. 2.2);
- Подкомпонент хранения данных (см. п. 2.3);
- Подкомпонент выполнения фоновых задач (см. п. 2.4);
- Подкомпонент формирования истории изменений (см. п. 2.5);
- Подкомпонент формирования файла для экспорта (см. п. 2.6);
- Подкомпонент взаимодействия с API клиентского приложения (см. п. 2.7);
- Подкомпонент импорта файлов (см. п. 2.8);
- Подкомпонент работы с черновиками (см. п. 2.9);
- Подкомпонент работы с Реестром анкет (см. п. 2.10);
- Подкомпонент просмотра истории изменений (см. п. 2.11);
- Подкомпонент управления настройками (см. п. 2.12).

### **2.1. Подкомпонент API клиентского приложения**

Описание API приводится в приложении к настоящему документу (Приложение А. Описание API).

### **2.2. Подкомпонент поиска ошибок и дублей**

Подкомпонент поиска ошибок и дублей реализует следующие функции:

- автоматическое выявление ошибок в черновиках анкет;
- автоматический поиск дублирующейся информации в черновиках анкет.

### **2.3. Подкомпонент хранения данных**

Подкомпонент хранения данных реализует функцию хранения данных черновиков, истории изменений черновиков, справочников.

### **2.4. Подкомпонент выполнения фоновых задач**

Подкомпонент выполнения фоновых задач реализует функцию выполнения длительных асинхронных операций.

### **2.5. Подкомпонент формирования истории изменений**

Подкомпонент формирования истории изменений реализует функцию формирования истории изменения атрибутов записей, а также информации об операторах, выполнивших изменения.

### **2.6. Подкомпонент формирования файла для экспорта**

Подкомпонент формирования файла для экспорта реализует функцию формирования файла для экспорта на основе табличных данных, с заданными параметрами форматирования.

### **2.7. Подкомпонент взаимодействия с API клиентского приложения**

Подкомпонент взаимодействия с API клиентского приложения реализует функцию обеспечения взаимодействия клиентского веб-приложения и API клиентского приложения.

### **2.8. Подкомпонент импорта файлов**

Подкомпонент импорта файлов реализует следующие функции:

- чтение и разбор файла в табличном формате;
- нахождение наименований колонок в файле и сопоставление их атрибутам анкеты;
- установка фиксированного значения любого из выбранных атрибутов для всех импортируемых данных.

### **2.9. Подкомпонент работы с черновиками**

Подкомпонент работы с черновиками реализует следующие функции:



- работа со списком анкет в черновике, включая поиск и фильтрацию;
- редактирование анкет в черновике;
- отображение найденных ошибок и дублей;
- выполнение проверок по требованию;
- редактирование выбранной анкеты;
- удаление анкеты;
- массовые действия с анкетами;
- выгрузка анкетных данных в файл табличного формата;
- публикация анкет из черновика в реестр анкет.

## **2.10. Подкомпонент работы с Реестром анкет**

Подкомпонент работы с Реестром анкет реализует следующие функции:

- работа со списком анкет в реестре анкет, включая поиск и фильтрацию;
- редактирование анкет в реестре анкет;
- отображение найденных ошибок и дублей;
- выполнение проверок по требованию;
- редактирование выбранной анкеты;
- удаление анкеты;
- массовые действия с анкетами;
- выгрузка анкетных данных в файл табличного формата;
- совместная работа со списком анкет нескольких операторов.

## **2.11. Подкомпонент просмотра истории изменений**

Подкомпонент просмотра истории изменений реализует следующие функции:

- отображение истории изменений одной анкеты;

- отображение истории изменений всех анкет с возможностью поиска и фильтрации.

## **2.12. Подкомпонент управления настройками**

Подкомпонент управления настройками реализует функцию настройки перечня выполняемых проверок анкет на наличие ошибок и дублей.

### **3. НАСТРОЙКА ПРОГРАММЫ**

Ниже приведен порядок настройки и описание действий по настройке Программы с использованием инструментов контейнеризации компании Docker.

#### **3.1. Порядок настройки Программы:**

- 1) Подготовка окружения;
- 2) Настройка Программы на условия применения;
- 3) Проверка Программы.

#### **3.2. Описание действий по настройке Программы**

##### **3.2.1. Подготовка окружения**

Программа распространяется в виде docker образа. Для запуска необходимо установить на сервер операционную систему Ubuntu 18.04 или выше, а также Docker.

Установка Docker осуществляется в соответствии с инструкциями, размещенными по адресу:

<https://docs.docker.com/engine/install/ubuntu/>

Установка Docker-compose осуществляется в соответствии с инструкциями, размещенными по адресу:

<https://docs.docker.com/compose/install/>

##### **3.2.2. Настройка Программы на условия применения**

Распакуйте архив в каталог на сервере (например, в домашний каталог ~):

- \$ unzip kartoteka.zip -d ~

Структура полученного каталога:

- docker-compose.yml – compose файл для запуска сервиса;
- config.json – конфигурационный файл сервиса;
- kartoteka.conf – конфигурационный файл веб сервера;

- ssl/ – каталог для ssl сертификатов.

В файлах `docker-compose.yml` и `kartoteka.conf` замените все вхождения `example.com` на доменное имя, на котором будет работать сервис.

Для работы веб сервера по `https` протоколу поместите в папку `ssl/` ssl сертификат с именем `certificate.pem` и приватный ключ `privkey.pem`.

В файле `config.json` измените следующие поля:

- `VUE_APP_API_URL` – адрес бекенд API картотеки;
- `VUE_APP_KEYCLOAK_REMOTE_HOST` – адрес identity провайдера `keycloak`;
- `VUE_APP_KEYCLOAK_REALM` – название `keycloak realm`;
- `VUE_APP_KEYCLOAK_CLIENT_ID` – id приложения (клиента) `keycloak`;
- `VUE_APP_KEYCLOAK_REDIRECT_URL` – адрес картотеки.

Загрузите `docker`-образы из архива на хост командами:

- `$ docker load < kartoteka.tar.gz;`
- `$ docker load < kartoteka_backend.tar.gz;`
- `$ rm *.tar.gz.`

### 3.2.3. Запуск сервиса

Для запуска сервиса выполните следующую команду:

- `$ docker-compose up -d`

Ответ:

Starting nginx ... done

Starting kartoteka ... done

Starting postgres ... done

Starting postgres-keycloak ... done

Starting backend ... done

### **3.3. Выбор функций Программы**

Выбор функций Программы или расширение их номенклатуры возможны путем внесения изменений в исходный код.

Отключение невостребованных функций выполняется путем преобразования строк исходного текста в языковые конструкции, используемые для включения в текст Программы пояснений, не влияющих на ее выполнение.

#### **4. ПРОВЕРКА ПРОГРАММЫ**

Для проверки Программы необходимо выполнить следующие действия:

- запустить загрузочный модуль web-браузера;
- в адресной строке браузера необходимо ввести http-адрес Программы и нажать на клавиатуре клавишу <Enter> для загрузки страницы авторизации;
- ввести логин и пароль доступа к Программе.

## **5. ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ**

Дополнительные возможности не предусмотрены.

## ПЕРЕЧЕНЬ ПРИНЯТЫХ ТЕРМИНОВ И СОКРАЩЕНИЙ

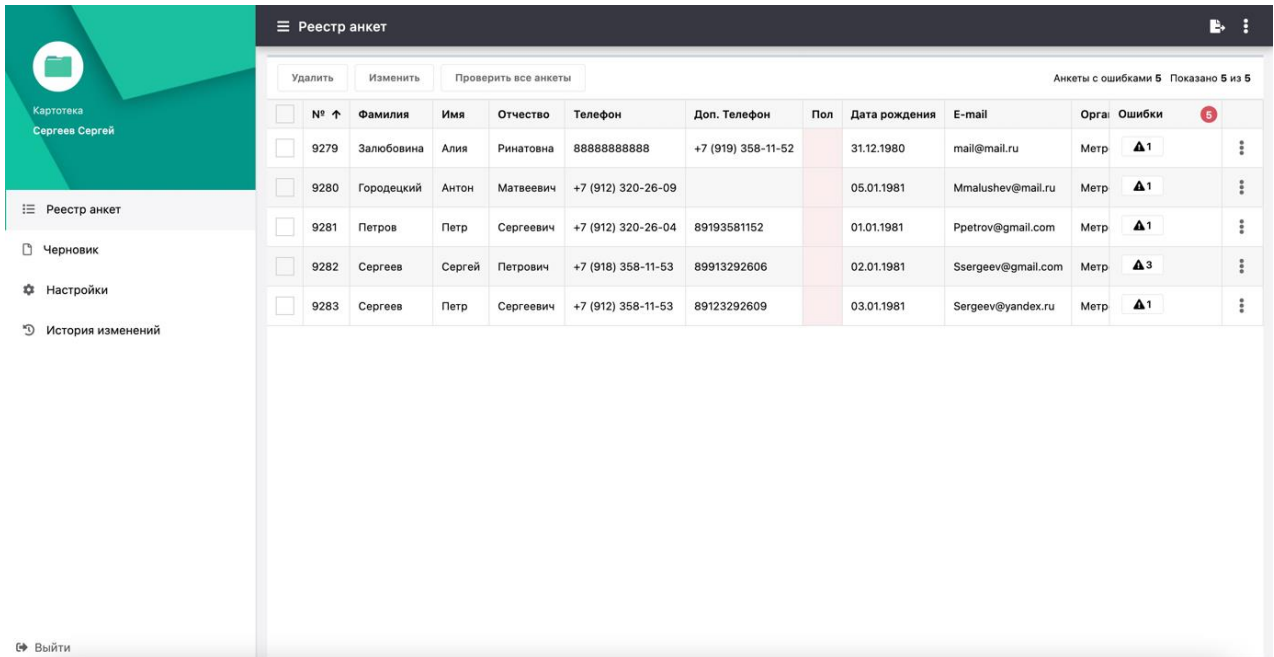
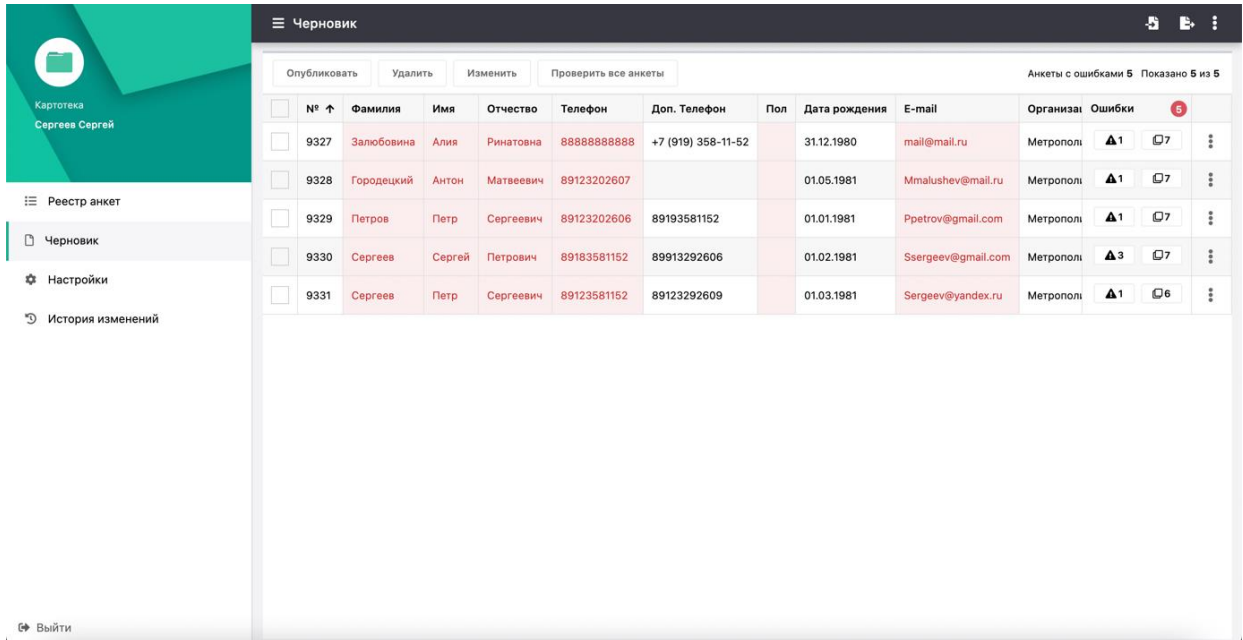
Таблица 1 — Перечень терминов

Термин	Определение
API	Application Programming Interface — программный интерфейс
ГОСТ	Государственный стандарт
Компонент	Часть программы, выделенная по определенному признаку или совокупности признаков и рассматриваемая как единое целое
Программа	Программное обеспечение для обработки анкетных данных «Картотека».
Сервер	Программный компонент вычислительной системы, выполняющий сервисные (обслуживающие) функции по запросу клиента, предоставляя ему доступ к определенным ресурсам или услугам
Сервис	Программный компонент, обслуживающий запросы других программных компонентов с использованием прикладных сетевых протоколов, например, SOAP
Функция	Совокупность действий системы, направленная на достижение определенной цели
Экспорт	Передача данных в определенном формате во внешнюю среду для использования вне программы



## Приложение А. Описание API

Данное API используется на экране работы с черновиком и реестром анкет.



## A.1 GET/questionnaires/{id}/lock

Захват анкеты оператором для редактирования

### Parameters

Name	Description
------	-------------

AuthorizationKeycloak access token with Bearer prefix	
---	--

**string**

*(header)*

id *	
------	--

**integer**

*(path)*

### Responses

Response content type

application/json

### CodeDescription

200 Questionnaire found and locked

```
{
  "id": 0,
  "bankAccount": "string",
  "bankBik": "string",
  "bankCard": "string",
  "bankCoraccount": "string",
  "bankInn": "string",
  "bankKpp": "string",
  "bankName": "string",
  "birthday": "2022-09-06",
  "districtName": "string",
```

## CodeDescription

```
"email": "string",
"groupName": "string",
"inn": "string",
"lastName": "string",
"middleName": "string",
"name": "string",
"organisation": "string",
"passportCode": "string",
"passportIssuedAt": "2022-09-06",
"passportIssuedBy": "string",
"passportNumber": "string",
"passportSeries": "string",
"phone": "string",
"provider": "string",
"publicationStatus": "string",
"recommendations": "string",
"rejectReward": true,
"secondPhone": "string",
"sex": "string",
"snils": "string",
"source": "string",
"validationErrors": {},
"createdAt": "2022-09-06",
"updatedAt": "2022-09-06",
"updatedBy": {
  "id": 0,
  "shortFullName": "string"
}
```

## CodeDescription

```
}
```

401 Invalid Authorization header

```
{  
  "errors": [  
    "string"  
  ]  
}
```

404 Questionnaire not found

```
{  
  "errors": [  
    "string"  
  ]  
}
```

### A.2 POST/questionnaires/batch\_check

Запуск механизма проверки корректности данных в выбранных анкетах

#### Parameters

Name	Description
------	-------------

params

<b>object</b> <i>(body)</i>	{ "questionnaires_ids": [ [ 100, 101
--------------------------------	--

Name	Description
------	-------------

	]
	]
	}
	Parameter content type
	application/json

AuthorizationKeycloak access token with Bearer prefix  
**string**  
*(header)*

### Responses

Response content type  
application/json

### CodeDescription

200 questionnaires batch checked ok

```
{  
  "currentPage": 0,  
  "pageItems": 0,  
  "totalPages": 0,  
  "totalCount": 0,  
  "errorsCount": 0,  
  "questionnairesCount": 0  
}
```

401 Invalid Authorization header

```
{  
  "errors": [  

```

## CodeDescription

```
"string"  
]  
}
```

### A.3 GET/questionnaires/{id}/unlock

Освобождение анкеты оператором после редактирования

#### Parameters

Name	Description
------	-------------

AuthorizationKeycloak access token with Bearer prefix

**string**

*(header)*

**id \***

**integer**

*(path)*

#### Responses

Response content type

application/json

#### CodeDescription

200 Questionnaire found and unlocked

```
{  
  "id": 0,  
  "bankAccount": "string",  
  "bankBik": "string",  
  "bankCard": "string",
```

## CodeDescription

```
"bankCoraccount": "string",  
"bankInn": "string",  
"bankKpp": "string",  
"bankName": "string",  
"birthday": "2022-09-06",  
"districtName": "string",  
"email": "string",  
"groupName": "string",  
"inn": "string",  
"lastName": "string",  
"middleName": "string",  
"name": "string",  
"organisation": "string",  
"passportCode": "string",  
"passportIssuedAt": "2022-09-06",  
"passportIssuedBy": "string",  
"passportNumber": "string",  
"passportSeries": "string",  
"phone": "string",  
"provider": "string",  
"publicationStatus": "string",  
"recommendations": "string",  
"rejectReward": true,  
"secondPhone": "string",  
"sex": "string",  
"snils": "string",  
"source": "string",  
"validationErrors": { }
```

## CodeDescription

```
"createdAt": "2022-09-06",  
"updatedAt": "2022-09-06",  
"updatedBy": {  
  "id": 0,  
  "shortFullName": "string"  
}  
}
```

401 Invalid Authorization header

```
{  
  "errors": [  
    "string"  
  ]  
}
```

404 Questionnaire not found

```
{  
  "errors": [  
    "string"  
  ]  
}
```

### A.4 POST/questionnaires/search

Поиск по анкетам

#### Parameters



Name	Description
------	-------------

page

**integer**

*(query)*

perPage *Default value : 25*

**integer**

*(query)*

AuthorizationKeycloak access token with Bearer prefix

**string**

*(header)*

### Responses

Response content type

application/json

### CodeDescription

200 Questionnaires found

```
{  
  "currentPage": 0,  
  "pageItems": 0,  
  "totalPages": 0,  
  "totalCount": 0,  
  "errorsCount": 0,  
  "questionnairesCount": 0  
}
```

401 Invalid Authorization header

## CodeDescription

```
{  
  "errors": [  
    "string"  
  ]  
}
```

404 Page number overflow

### A.5 GET/questionnaires/draft

Черновик с анкетами текущего оператора

#### Parameters

Name	Description
------	-------------

page

**integer**

*(query)*

perPage *Default value : 25*

**integer**

*(query)*

AuthorizationKeycloak access token with Bearer prefix

**string**

*(header)*

#### Responses

Response content type

application/json

## CodeDescription

## CodeDescription

200 Questionnaires drafted found

```
{  
  "currentPage": 0,  
  "pageItems": 0,  
  "totalPages": 0,  
  "totalCount": 0,  
  "errorsCount": 0,  
  "questionnairesCount": 0  
}
```

401 Invalid Authorization header

```
{  
  "errors": [  
    "string"  
  ]  
}
```

404 Page number overflow

### A.6 GET/questionnaires/published

Реестр опубликованных анкет

#### Parameters

Name	Description
------	-------------

page

**integer**

(*query*)

<b>Name</b>	<b>Description</b>
-------------	--------------------

perPage	<i>Default value : 25</i>
---------	---------------------------

**integer**

*(query)*

AuthorizationKeycloak access token with Bearer prefix

**string**

*(header)*

### **Responses**

Response content type

application/json

### **CodeDescription**

200 Questionnaires published found

```
{  
  "currentPage": 0,  
  "pageItems": 0,  
  "totalPages": 0,  
  "totalCount": 0,  
  "errorsCount": 0,  
  "questionnairesCount": 0  
}
```

401 Invalid Authorization header

```
{  
  "errors": [  
    "string"  
  ]  
}
```

## CodeDescription

```
}
```

404 Page number overflow

### A.7 PATCH/questionnaires/batch\_update

Массовое обновление полей в выбранных анкетах

#### Parameters

Name	Description
------	-------------

params

<b>object</b> ( <i>body</i> )	<pre>{   "questionnaires_ids": [     [       100     ]   ],   "questionnaire": {     "name": "Пирожков",     "recommendations": "Пирожков рекомендует использовать"   } }</pre>
----------------------------------	---

Parameter content type

application/json

AuthorizationKeycloak access token with Bearer prefix

**string**

(*header*)

#### Responses

Response content type

application/json

## CodeDescription

200 questionnaires batch updated

```
{  
  "questionnaires": [  
    {  
      "id": 0,  
      "bankAccount": "string",  
      "bankBik": "string",  
      "bankCard": "string",  
      "bankCoraccount": "string",  
      "bankInn": "string",  
      "bankKpp": "string",  
      "bankName": "string",  
      "birthday": "2022-09-06",  
      "districtName": "string",  
      "email": "string",  
      "groupName": "string",  
      "inn": "string",  
      "lastName": "string",  
      "middleName": "string",  
      "name": "string",  
      "organisation": "string",  
      "passportCode": "string",  
      "passportIssuedAt": "2022-09-06",  
      "passportIssuedBy": "string",  
      "passportNumber": "string",  
      "passportSeries": "string",
```

## CodeDescription

```
"phone": "string",  
"provider": "string",  
"publicationStatus": "string",  
"recommendations": "string",  
"rejectReward": true,  
"secondPhone": "string",  
"sex": "string",  
"snils": "string",  
"source": "string",  
"validationErrors": {},  
"createdAt": "2022-09-06",  
"updatedAt": "2022-09-06",  
"updatedBy": {  
  "id": 0,  
  "shortFullName": "string"  
}  
}  
]  
}
```

401 Invalid Authorization header

```
{  
  "errors": [  
    "string"  
  ]  
}
```

## A.8 PATCH/questionnaires/batch\_destroy

Удаление выбранных анкет

### Parameters

Name	Description
params	
<b>object</b> <i>(body)</i>	{ "questionnaires_ids": [ [ 100 ] ] }
	Parameter content type
	application/json

AuthorizationKeycloak access token with Bearer prefix

**string**

*(header)*

### Responses

Response content type

application/json

### CodeDescription

200 questionnaires batch deleted

401 Invalid Authorization header

```
{  
  "errors": [  
    
```



## CodeDescription

```
"string"  
]  
}
```

### A.9 GET/questionnaires/{id}

Получение актуальной информации полей анкеты

#### Parameters

Name	Description
------	-------------

AuthorizationKeycloak access token with Bearer prefix

**string**

*(header)*

**id \***

**integer**

*(path)*

#### Responses

Response content type

application/json

#### CodeDescription

200 Questionnaire found

```
{  
  "id": 0,  
  "bankAccount": "string",  
  "bankBik": "string",  
  "bankCard": "string",
```

## CodeDescription

```
"bankCoraccount": "string",  
"bankInn": "string",  
"bankKpp": "string",  
"bankName": "string",  
"birthday": "2022-09-06",  
"districtName": "string",  
"email": "string",  
"groupName": "string",  
"inn": "string",  
"lastName": "string",  
"middleName": "string",  
"name": "string",  
"organisation": "string",  
"passportCode": "string",  
"passportIssuedAt": "2022-09-06",  
"passportIssuedBy": "string",  
"passportNumber": "string",  
"passportSeries": "string",  
"phone": "string",  
"provider": "string",  
"publicationStatus": "string",  
"recommendations": "string",  
"rejectReward": true,  
"secondPhone": "string",  
"sex": "string",  
"snils": "string",  
"source": "string",  
"validationErrors": { }
```

## CodeDescription

```
"createdAt": "2022-09-06",  
"updatedAt": "2022-09-06",  
"updatedBy": {  
  "id": 0,  
  "shortFullName": "string"  
}  
}
```

401 Invalid Authorization header

```
{  
  "errors": [  
    "string"  
  ]  
}
```

404 Questionnaire not found

```
{  
  "errors": [  
    "string"  
  ]  
}
```

### A.10 PUT/questionnaires/{id}

Обновление полей анкеты

#### Parameters

Name	Description
------	-------------

<b>Name</b>	<b>Description</b>
params	
<b>object</b> <i>(body)</i>	{ "bankAccount": "string", "bankBik": "string", "bankCard": "string", "bankCoraccount": "string", "bankInn": "string", "bankKpp": "string", "bankName": "string", "birthday": "2022-09-06", "districtName": "string", "email": "string", "groupName": "string", "inn": "string", "lastName": "string", "middleName": "string", "name": "string", "organisation": "string", "passportCode": "string", "passportIssuedAt": "2022-09-06", "passportIssuedBy": "string", "passportNumber": "string", "passportSeries": "string", "phone": "string", "provider": "string", "publicationStatus": "string", "recommendations": "string", "rejectReward": true,

Name	Description
	<pre>"secondPhone": "string", "sex": "string", "snils": "string", "source": "string" }</pre>
	Parameter content type application/json

AuthorizationKeycloak access token with Bearer prefix

**string**

*(header)*

**id \***

**integer**

*(path)*

### Responses

Response content type

application/json

### CodeDescription

200 Questionnaires found

```
{  
  "id": 0,  
  "bankAccount": "string",  
  "bankBik": "string",  
  "bankCard": "string",  
  "bankCoraccount": "string",  
  "bankInn": "string",
```

## CodeDescription

```
"bankKpp": "string",  
"bankName": "string",  
"birthday": "2022-09-06",  
"districtName": "string",  
"email": "string",  
"groupName": "string",  
"inn": "string",  
"lastName": "string",  
"middleName": "string",  
"name": "string",  
"organisation": "string",  
"passportCode": "string",  
"passportIssuedAt": "2022-09-06",  
"passportIssuedBy": "string",  
"passportNumber": "string",  
"passportSeries": "string",  
"phone": "string",  
"provider": "string",  
"publicationStatus": "string",  
"recommendations": "string",  
"rejectReward": true,  
"secondPhone": "string",  
"sex": "string",  
"snils": "string",  
"source": "string",  
"validationErrors": {},  
"createdAt": "2022-09-06",  
"updatedAt": "2022-09-06",
```

## CodeDescription

```
"updatedBy": {  
  "id": 0,  
  "shortFullName": "string"  
}
```

401 Invalid Authorization header

```
{  
  "errors": [  
    "string"  
  ]  
}
```

404 Questionnaire not found

```
{  
  "errors": [  
    "string"  
  ]  
}
```

### A.11 GET/questionnaires/autocomplete

Предложение вариантов заполнения поля в анкете на основе существующих данных

## Parameters

Name	Description
------	-------------

<b>Name</b>	<b>Description</b>
field ( <i>query</i> )	Show distincted array by field name <i>Example</i> : last_name
q[lastName_cont] ( <i>query</i> )	Ransack find q param for example by last_name <i>Example</i> : Пирожков
page <b>integer</b> ( <i>query</i> )	
perPage <b>integer</b> ( <i>query</i> )	<i>Default value</i> : 25
Authorization <b>string</b> ( <i>header</i> )	Keycloak access token with Bearer prefix

### **Responses**

Response content type  
application/json

### **CodeDescription**

200 Questionnaires found

```
{  
  "currentPage": 0,  
  "pageItems": 0,  
  "totalPages": 0,  
  "totalCount": 0,
```



## CodeDescription

```
“result”: [  
  {  
    “additionalProp1”: “string”,  
    “additionalProp2”: “string”,  
    «additionalProp3»: «string»  
  }  
]  
}
```

401 Invalid Authorization header

```
{  
  “errors”: [  
    “string”  
  ]  
}
```

404 Page number overflow

### A.12 GET/questionnaire\_histories

Получение сведений об истории изменения анкет

#### Parameters

Name	Description
------	-------------

questionnaire_id	Questionnaire ID (если не указать то берем всю)
------------------	---

(*query*)

q[created\_at\_lteq]Ransack find q param for example by created\_at

<b>Name</b>	<b>Description</b>
<i>(query)</i>	
page	
<b>integer</b>	
<i>(query)</i>	
perPage	<i>Default value : 25</i>
<b>integer</b>	
<i>(query)</i>	
Authorization	Keycloak access token with Bearer prefix
<b>string</b>	
<i>(header)</i>	

### **Responses**

Response content type  
application/json

### **CodeDescription**

200 QuestionnaireHistories found

```
{
  "currentPage": 0,
  "pageItems": 0,
  "totalPages": 0,
  "totalCount": 0,
  "questionnaireHistories": [
    {
      "id": 0,
      "itemId": 0,
      "fio": "string",
      "name": "string",
```

## CodeDescription

```
"middle_name": "string",
"last_name": "string",
"event": "string",
"createdAt": "2022-09-06",
"changedBy": {
  "id": 0,
  "shortFullName": "string"
},
"objectChanges": {
  "questionnaire": [
    {
      "id": 0,
      "bankAccount": "string",
      "bankBik": "string",
      "bankCard": "string",
      "bankCoraccount": "string",
      "bankInn": "string",
      "bankKpp": "string",
      "bankName": "string",
      "birthday": "2022-09-06",
      "districtName": "string",
      "email": "string",
      "groupName": "string",
      "inn": "string",
      "lastName": "string",
      "middleName": "string",
      "name": "string",
      "organisation": "string",
      "passportCode": "string",
```

## CodeDescription

```
"passportIssuedAt": "2022-09-06",
"passportIssuedBy": "string",
"passportNumber": "string",
"passportSeries": "string",
"phone": "string",
"provider": "string",
"publicationStatus": "string",
"recommendations": "string",
"rejectReward": true,
"secondPhone": "string",
"sex": "string",
"snils": "string",
"source": "string",
"validationErrors": {},
"createdAt": "2022-09-06",
"updatedAt": "2022-09-06",
"updatedBy": {
  "id": 0,
  "shortFullName": "string"
}
}
]
},
"object": {
  "questionnaire": [
    {
      "id": 0,
      "bankAccount": "string",
      "bankBik": "string",
```

## CodeDescription

```
"bankCard": "string",  
"bankCoraccount": "string",  
"bankInn": "string",  
"bankKpp": "string",  
"bankName": "string",  
"birthday": "2022-09-06",  
"districtName": "string",  
"email": "string",  
"groupName": "string",  
"inn": "string",  
"lastName": "string",  
"middleName": "string",  
"name": "string",  
"organisation": "string",  
"passportCode": "string",  
"passportIssuedAt": "2022-09-06",  
"passportIssuedBy": "string",  
"passportNumber": "string",  
"passportSeries": "string",  
"phone": "string",  
"provider": "string",  
"publicationStatus": "string",  
"recommendations": "string",  
"rejectReward": true,  
"secondPhone": "string",  
"sex": "string",  
"snils": "string",  
"source": "string",  
"validationErrors": { }
```

## CodeDescription

```
"createdAt": "2022-09-06",  
"updatedAt": "2022-09-06",  
"updatedBy": {  
  "id": 0,  
  "shortFullName": "string"  
}  
}  
]  
}  
}  
]  
}
```

