

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Департамент программной инженерии

**СОГЛАСОВАНО**

Профессор департамента  
программной инженерии факультета  
компьютерных наук, к.т.н

**УТВЕРЖДАЮ**

Академический руководитель  
образовательной программы  
«Программная инженерия» профессор  
департамента программной  
инженерии, канд. техн. наук

\_\_\_\_\_ Е. М. Гринкруг  
«\_\_\_\_\_» \_\_\_\_\_ 2020 г.

\_\_\_\_\_ В. В. Шилов  
«\_\_\_\_\_» \_\_\_\_\_ 2020 г.

**СИСТЕМА УПРАВЛЕНИЯ ЗАДАНИЯМИ ПО  
АВТОМАТИЧЕСКОМУ СБОРУ ДАННЫХ ИЗ СЕТИ  
ИНТЕРНЕТ**

**Программа и методика испытаний**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.04.13-01 51 01-1**

|             |              |              |              |              |
|-------------|--------------|--------------|--------------|--------------|
| Инв. № подл | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |
|             |              |              |              |              |

Исполнитель: студент группы БПИ 174  
\_\_\_\_\_ Д. Ю. Редникова  
«\_\_\_\_\_» \_\_\_\_\_ 2020 г.

|             |              |              |              |              |
|-------------|--------------|--------------|--------------|--------------|
| Инв. № подл | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |
|             |              |              |              |              |

**СИСТЕМА УПРАВЛЕНИЯ ЗАДАНИЯМИ ПО  
АВТОМАТИЧЕСКОМУ СБОРУ ДАННЫХ ИЗ СЕТИ  
ИНТЕРНЕТ**

**Программа и методика испытаний**

**RU.17701729.04.13-01 51 01-1**

**Листов 29**

## Содержание

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Объекты испытаний</b>  | <b>4</b>  |
| 1.1      | Наименование программы . . . . .                                | 4         |
| 1.2      | Область применения программы . . . . .                          | 4         |
| 1.2.1    | Функциональное назначение . . . . .                             | 4         |
| 1.2.2    | Эксплуатационное назначение . . . . .                           | 4         |
| 1.2.3    | Область применения . . . . .                                    | 4         |
| <b>2</b> | <b>Цель испытаний</b>   | <b>5</b>  |
| <b>3</b> | <b>Требования к программе</b>                                   | <b>6</b>  |
| 3.1      | Требования к функциональным характеристикам . . . . .           | 6         |
| 3.2      | Требования к интерфейсу . . . . .                               | 8         |
| 3.3      | Требования к надежности . . . . .                               | 8         |
| <b>4</b> | <b>Требования к программной документации</b>                    | <b>9</b>  |
| 4.1      | Состав программной документации . . . . .                       | 9         |
| <b>5</b> | <b>Средства и порядок испытаний</b>                             | <b>10</b> |
| 5.1      | Технические средства, используемые во время испытаний . . . . . | 10        |
| 5.2      | Программные средства, используемые во время испытаний . . . . . | 10        |
| 5.3      | Порядок проведения испытаний . . . . .                          | 10        |
| <b>6</b> | <b>Методы испытаний</b>   | <b>11</b> |
| 6.1      | Проверка требований к функциональным характеристикам . . . . .  | 11        |
| 6.1.1    | Регистрация . . . . .   | 11        |
| 6.1.2    | Авторизация . . . . .   | 12        |
| 6.1.3    | Проекты (6.2) . . . . .   | 13        |
| 6.1.4    | Краулеры (6.2) . . . . .  | 16        |
| 6.1.5    | Запуски (6.2) . . . . .   | 17        |
| 6.1.6    | Периодические задачи (6.2) . . . . .                            | 20        |
| 6.1.7    | Результаты тестирования . . . . .                               | 25        |
| 6.2      | Проверка требований к программной документации . . . . .        | 26        |
|          | <b>Приложение А</b>   | <b>27</b> |
|          | <b>Список источников</b>  | <b>28</b> |
|          | <b>Лист регистрации изменений</b>                               | <b>29</b> |

|                              |              |              |              |              |
|------------------------------|--------------|--------------|--------------|--------------|
| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

# 1 Объекты испытаний

## 1.1 Наименование программы

Система управления заданиями по автоматическому сбору данных из сети Интернет (System for managing tasks of collecting data from the Internet )

## 1.2 Область применения программы

### 1.2.1 Функциональное назначение

Система будет применяться как средство управления проектами по созданию, редактированию и запуску веб краулеров для сбора данных в сети интернет. Продукт позволит следить за запусками в режиме реального времени, а также создавать периодические запуски по расписанию.

### 1.2.2 Эксплуатационное назначение

Программа будет использоваться как инструмент для самостоятельной или совместной работы над проектами для запуска, управления сбора данных с помощью веб краулеров в сети интернет.

Таким образом, программный продукт позволит создавать, запускать образы пауков (см. 6.2) для сбора, управления, логирования и дальнейшего экспорта данных в целях сбора, изучения и мониторинга данных (см. 6.2).

### 1.2.3 Область применения

Технологии web-scraping используются как в науке, так и в бизнесе - многие люди чувствуют потребность в извлечении данных из HTML разметки интернет страниц. Существующие аналоги реализуют базовый функционал(сбор данных), но не предоставляют такие дополнительные возможности как периодический запуск или совместное редактирование. Многие аналоги (прим. scrapy [1]) имеют ограниченный функционал. Главные возможности, которыми продукт обеспечит предполагаемых пользователей:

- Совместное управление запусками краулеров
- Периодический запуск задач
- Сбор логов, ошибок
- Группировка краулеров, а также их запусков в проект
- Бесплатная функциональность

| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
|------------------------------|--------------|--------------|--------------|--------------|
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

## 2 Цель испытаний

Проверка на соответствие требованиям, указанным в документе «Система управления заданиями по автоматическому сбору данных из сети Интернет. Техническое задание».

| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
|------------------------------|--------------|--------------|--------------|--------------|
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

## 3 Требования к программе

### 3.1 Требования к функциональным характеристикам

Программа выполняется в рамках темы курсовой работы в соответствии с учебным планом подготовки бакалавров по направлению 09.03.04 «Программная инженерия» Национального исследовательского университета «Высшая школа экономики», факультет компьютерных наук.

Программа должна удовлетворять следующим требованиям:

#### 1. Авторизация

Чтобы использовать сервис, клиентская программа должна иметь возможность авторизоваться в системе с помощью REST API

- (a) Для регистрации пользователю нужно указать следующие данные
  - i. Почта - уникальна для каждого зарегистрированного пользователя;
  - ii. Имя - длина больше 1 символ;
  - iii. Логин - длина больше 2 символов;
  - iv. Пароль - длина больше 2 символов;
- (b) Для авторизации пользователя в системе должны быть указаны следующие данные
  - i. Почта;
  - ii. Пароль;

#### 2. Проекты

Должны быть реализованы запросы REST API для предоставления клиенту следующей функциональности

- (a) Создание проекта со следующей информацией
  - i. Имя проекта;
  - ii. Описание проекта - опциональное поле;
- (b) Обновление метаданных о проекте (редактирование) могут быть обновлены только участником с минимальным уровнем дотупа **ReadAndWrite**. Следующие данные могут быть обновлены:
  - i. Имя проекта;
  - ii. Описание проекта;
  - iii. Настройки проекта для запуска краулеров;
  - iv. Аргументы для запуска краулеров проекта;
- (c) Обновление **egg** файла проекта (редактирование) – минимальный уровень доступа участника, обновляющий данные о проекте **ReadAndWrite**.
- (d) Удаление данных о проекте. Удалить проект может только владелец **Owner**.
- (e) Просмотр списка проектов (с пагинацией), к которым у пользователя есть как минимум **ReadOnly** доступ.

#### 3. Участники проектов

Должны быть реализованы запросы REST API для предоставления клиенту следующей функциональности

| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
|------------------------------|--------------|--------------|--------------|--------------|
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

- (a) Просмотр информации об участниках проекта;
  - i. Имя, почта, логин участника;
  - ii. Статус участника в проекте (`ReadOnly`, `ReadAndWrite` или `Owner`);
- (b) Обновление статуса участника проекта. Это действие совершать может только владелец проекта;
- (c) Удаление участника из проекта. Данное действие может совершать только владелец проекта;
- (d) Добавление нового участника с указанными правами на редактирование. Данное действие может совершать только владелец проекта;

#### 4. Краулеры

Должны быть реализованы запросы REST API для предоставления клиенту следующей функциональности

- (a) Просмотр списка краулеров проекта;
- (b) Редактирование информации о краулере для последующих запусков. Следующая информация может быть изменена
  - i. Настройки краулера для запуска;
  - ii. Аргументы для запуска;

#### 5. Запуски краулеров

Должны быть реализованы запросы REST API для предоставления клиенту следующей функциональности

- (a) Просмотр списка запусков в определенном статусе (`Pending`, `Running` или `Finished`) с пагинацией, совершенных в проектах, к которым у пользователя есть как минимум `ReadOnly` доступ;
- (b) Редактирование запуска - остановка запуска, перевод его в состояние `Finished`. Операция может быть применена только к запускам в состоянии `Running` или `Pending`;
- (c) Удаление запуска - удаление всех данных о запуске из базы данных. Операция может быть применена только к запускам в состоянии `Finished`;
- (d) Создание запуска со следующей информацией
  - i. Краулер, с которым происходит запуск;
  - ii. Настройки запуска – это могут быть как и предопределенные настройки на `scrapy`<sup>1</sup>, так и собственные настройки;
  - iii. Аргументы запуска – аргументы для запуска краулера, которые передаются через командную строку;
  - iv. Описание запуска;

#### 6. Периодические запуски

Должны быть реализованы запросы REST API для предоставления клиенту следующей функциональности

- (a) Просмотр списка периодических запусков с пагинацией;

---

<sup>1</sup><http://doc.scrapy.org/en/latest/topics/settings.html>

| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
|------------------------------|--------------|--------------|--------------|--------------|
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

- (b) Редактирование следующей информации о периодическом запуске
  - i. Настройки будущих запусков – это могут быть как и предопределенные настройки на **scrapyd**, так и собственные настройки;
  - ii. Аргументы будущих запусков – аргументы для запуска краулера, которые передаются через командную строку;
  - iii. Краулер, с помощью которого будет совершен запуск;
  - iv. cron-expression расписания запуска;
- (c) Удаление периодического запуска;
- (d) Отмена последующих запусков - перевод периодической задачи в состояние **Disabled**;
- (e) Возобновление запусков - перевод периодической задачи в состояние **Enabled**;
- (f) Создание периодического запуска со следующими данными
  - i. Название;
  - ii. Описание – опциональное;
  - iii. Краулер;
  - iv. Приоритетность, влияющая на очередь запусков (**Low**, **Normal** или **High**);
  - v. Статус (**Enabled** или **Disabled**);
  - vi. Настройки будущих запусков – это могут быть как и предопределенные настройки на **scrapyd**, так и собственные настройки;
  - vii. Аргументы будущих запусков – аргументы для запуска краулера, которые передаются через командную строку;

## 3.2 Требования к интерфейсу

Доступ к программе должен быть предоставлен посредством интерфейса консольной программы.

## 3.3 Требования к надежности

- Программа должна осуществлять проверку корректности посылаемого текста в кодировке utf-8 и возвращать ошибку в случае наличия невалидных символов.
- Программа должна работать непрерывно (не должно быть ситуаций, в которой сервер возвращает status code 500).
- В базу данных не должны заноситься некорректные данные.

| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
|------------------------------|--------------|--------------|--------------|--------------|
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |



## 4 Требования к программной документации

### 4.1 Состав программной документации

Состав программной документации должен включать в себя следующие компоненты:

1. Техническое задание «Система управления заданиями по автоматическому сбору данных из сети Интернет » (ГОСТ 19.201-78)
2. Программа и методика испытаний «Система управления заданиями по автоматическому сбору данных из сети Интернет » (ГОСТ 19.301-78)
3. Пояснительная записка «Система управления заданиями по автоматическому сбору данных из сети Интернет » (ГОСТ 19.404-79)
4. Руководство оператора «Система управления заданиями по автоматическому сбору данных из сети Интернет » (ГОСТ 19.505-79)
5. Текст программы «Система управления заданиями по автоматическому сбору данных из сети Интернет » (ГОСТ 19.401-78)

Вся документация должна быть составлена согласно ЕСПД (ГОСТ 19.101-77, 19.104-78, 19.105-78, 19.106-78 и ГОСТ к соответствующим документам (см. выше)) [2]. Все документы сдаются в электронном виде в составе курсовой работы LMS НИУ ВШЭ.

Пояснительная записка «Система управления заданиями по автоматическому сбору данных из сети Интернет » должна быть проверена на плагиат (< 40% заимствований). Документ, подтверждающий проверку Пояснительной записки сдается в печатном виде вместе с подписанным отзывом от научного руководителя.

| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
|------------------------------|--------------|--------------|--------------|--------------|
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

## 5 Средства и порядок испытаний

### 5.1 Технические средства, используемые во время испытаний

1. Компьютер оснащенный процессором Intel Core i5 с тактовой частотой 2,3 ГГц;
2. 16 Гб ОЗУ;
3. Жесткий диск с объемом свободной памяти более чем 50 ГБ;
4. Клавиатура и мышь;
5. Доступ в интернет.

### 5.2 Программные средства, используемые во время испытаний

1. macOS 10.15.2;
2. scrapyd [1];
3. Scala 2.12.6;
4. Play-framework 2.6.13;
5. PostgreSQL 11 [5];

### 5.3 Порядок проведения испытаний

В процессе разработки программы были написаны unit-тесты, а также автотесты, тестирующие API сервера.

Автотесты выполняются на отдельной тестовой базе данных, которая перед началом каждого теста заполняется тестовыми данными.

Для запуска автотестов пользователю надо ввести команду `sbt test` в директории проекта `sw_backend_scala`, перед этим в отдельной директории запустить сервер scrapyd [1] командой `scrapyd`.

| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
|------------------------------|--------------|--------------|--------------|--------------|
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

## 6 Методы испытаний

### 6.1 Проверка требований к функциональным характеристикам

Для тестирования API запросов были написаны автотесты с помощью библиотеки `scralatestplus` [3]. Данная библиотека позволяет подменять зависимости основной программы и использовать `FakeRequest` как упрощенный способ выполнять запросы к контроллерам приложения.

Также для тестирования приложения используется библиотека `play-silhouette-testkit` [4]. Эта библиотека позволяет авторизовываться в приложении без использования cookies.

Для тестирования API запросов были написаны следующие тест-кейсы:

#### 6.1.1 Регистрация

| № | Название  | Предусловие   | Описание  | Ожидаемый результат   |
|---|---|---|---|---|
| 1 | Регистрация пользователя  | Пользователь неавторизован  | Пользователь отправляет запрос POST <code>/api/signup</code> с персональными данными в формате <code>app.forms.SignUpForm</code>  | Status code 200.<br>Cookies в хэдере                                  |
| 2 | Регистрация пользователя с уже зарегистрированным email или логином | Пользователь неавторизован.<br>В БД системы уже существует пользователь с вводимым логином и/или email. | Пользователь отправляет запрос POST <code>/api/signup</code> с персональными данными, которые совпадают с login и/или email другого пользователя в базе данных системы. | Status code 409.<br>Сообщение об ошибке «User Already Exists Message» |

Таблица 1 — Тест кейсы: регистрация

|                              |              |              |              |              |
|------------------------------|--------------|--------------|--------------|--------------|
| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

### 6.1.2 Авторизация

| № | Название   | Предусловие   | Описание  | Ожидаемый результат                  |
|---|--|---|---|--------------------------------------|
| 1 | Авторизация пользователя   | Пользователь неавторизован в системе. Пользователь был ранее зарегистрирован в системе  | Пользователь отправляет запрос на POST /api/signin с персональными данными в формате app.forms.SignInForm   | Status code 200.<br>Cookies в хэдере |
| 2 | Авторизация пользователя с неправильным паролем                  | Пользователь неавторизован в системе. Пользователь был ранее зарегистрирован в системе. | Пользователь отправляет запрос на POST /api/signin с персональными данными (email, пароль) так, что пароль не совпадает с уже введенным при регистрации.        | Status code 403.                     |
| 3 | Авторизация пользователя с несуществующими в системе credentials | Пользователь неавторизован в системе. Пользователь был ранее зарегистрирован в системе. | Пользователь отправляет запрос на POST /api/signin с персональными данными (email, пароль), которые не совпадают ни с одной парой credentials, хранящихся в БД. | Status code 409.                     |

Таблица 2 — Тест кейсы: авторизация

|                              |              |              |              |              |
|------------------------------|--------------|--------------|--------------|--------------|
| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

### 6.1.3 Проекты (6.2)

| № | Название                                | Предусловие   | Описание  | Ожидаемый результат  |
|---|---|---|---|--|
| 1 | Запрос на список проектов без пагинации | Пользователь авторизован в системе. У пользователя имеются 20 проектов. | Пользователь отправляет запрос GET /api/projects/10 для получения первых 10-и из списка проектов, к которому у него есть доступ.  | Status code 200. Json со списком проектов, к которым у пользователя есть как минимум readonly access, в хронологическом порядке их создания. |
| 2 | Запрос на список проектов с пагинацией  | Пользователь авторизован в системе. У пользователя имеются 20 проектов. | Пользователь отправляет запрос GET с query argument limit=10 и id для получения 10-и проектов из списка, которые в хронологическом порядке (по дате создания) располагаются раньше чем указанный проект id. | Status code 200. Json со списком проектов, к которым у пользователя есть как минимум readonly access, id которых >= idProject                |

Таблица 3 — Тест кейсы: просмотр списка проектов

| № | Название         | Предусловие               | Описание  | Ожидаемый результат                             |
|---|------------------|---------------------------|---|---|
| 3 | Создание проекта | Пользователь авторизован. | Пользователь отправляет запрос POST /api/projects с телом запроса в виде app.forms.ProjectCreForm | Status code 200 и json с id созданного проекта. |

Таблица 4 — Тест кейсы: создание проекта

|                              |              |              |              |              |
|------------------------------|--------------|--------------|--------------|--------------|
| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

| № | Название   | Предусловие  | Описание   | Ожидаемый результат   |
|---|--|--|--|---|
| 4 | Редактирование проекта                                   | Пользователь авторизован. У пользователя есть хотя бы 1 созданный проект. Пользователь имеет owner доступ к проекту. | Пользователь отправляет запрос PUT /api/projects/id с указанием id проекта для редактирования, с телом запроса в виде app.forms.ProjectForm. | Status code 200 и json с id созданного проекта. В таблице projects должна обновиться запись с id созданного проекта, где name, description, spiderSettings совпадают с переданными пользователем. |
| 5 | Редактирование проекта: доступ пользователя readonly     | Пользователь авторизован. У пользователя доступ к проекту id - readonly  | Пользователь с отправляет PUT /api/projects/id с корректным телом запроса (см. п. 4)   | Status code 403   |
| 6 | Редактирование проекта: доступ пользователя writeandread | Пользователь авторизован. У пользователя доступ к проекту id - writeandread  | Пользователь с отправляет PUT /api/projects/id с корректным телом запроса (см. п. 4)   | Status code 200. Информация о проекте успешно обновлена   |

Таблица 5 — Тест кейсы: проекты - редактирование метаданных

| № | Название                                   | Предусловие   | Описание  | Ожидаемый результат   |
|---|--|---|---|---|
| 7 | Удаление проекта владельцем                | Пользователь авторизован. У пользователя есть доступ к проекту id с правами owner. (таблица membership) | Пользователь отправляет DELETE /api/projects/:projectId запрос. | Status code 200, в таблице membership нет никаких записей о проекте с id, в таблице project тоже. |
| 8 | Удаление проекта: неверный id              | Пользователь авторизован. Id проекта нет в таблице membership.  | Пользователь отправляет DELETE /api/projects/:projectId запрос. | Status code 403   |
| 9 | Удаление проекта: права на проект НЕ owner | Пользователь авторизован. У пользователя права на проект НЕ owner.                                      | Пользователь отправляет DELETE /api/projects/:projectId запрос. | Status code 403   |

Таблица 6 — Тест кейсы: проект DELETE

|                              |              |              |              |              |
|------------------------------|--------------|--------------|--------------|--------------|
| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

| №  | Название  | Предусловие   | Описание  | Ожидаемый результат   |
|----|---|---|---|---|
| 10 | Deploy архива проекта   | Пользователь авторизован. У пользователя есть права на редактирование проекта (либо owner, либо writeandread) | Пользователь делает запрос по адресу PUT /api/projects/:projectId/ deploy с телом eggFile: <файл с расширением egg>       | Status code 200. В таблице проектов у проекта с id появился bytes в колонке eggFile |
| 11 | Deploy архива проекта: файл неверного формата                   | Пользователь авторизован. У пользователя есть права на редактирование проекта (либо owner, либо writeandread) | Пользователь делает запрос по адресу PUT /api/projects/:projectId/ deploy с телом eggFile: файл не того формата (не egg). | Status code 422   |
| 12 | Deploy архива проекта: права доступа не позволяют менять проект | Пользователь авторизован. У пользователя есть права на редактирование проекта readonly                        | Пользователь делает запрос по адресу PUT /api/projects/:projectId/ deploy с телом eggFile: <файл с расширением egg>       | Status code 403   |

Таблица 7 — Тест кейсы: проект PUT - deploy

|                              |              |              |              |              |
|------------------------------|--------------|--------------|--------------|--------------|
| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

#### 6.1.4 Краулеры (6.2)

| № | Название  | Предусловие  | Описание  | Ожидаемый результат                            |
|---|---|--|---|--|
| 1 | Отображения списка краулеров                        | Пользователь авторизован и имеет доступ к проекту id минимум readonly. | Пользователь отправляет запрос GET /api/projects/:projectId /crawlers с id проекта. | Status code 200, список пауков в проекте из БД |
| 2 | Отображение списка краулеров: нет доступа к проекту | Пользователь авторизован и не имеет доступ к проекту id                | Пользователь отправляет запрос GET /api/projects/:projectId /crawlers с id проекта. | Status code 403                                |

Таблица 8 — Тест кейсы: краулеры GET

| № | Название   | Предусловие   | Описание  | Ожидаемый результат |
|---|--|---|---|---------------------|
| 1 | Изменение настроек паука                                     | Пользователь авторизован. У пользователя есть write или owner доступ к проекту. | Пользователь отправляет запрос PUT /api/project/:projectId /crawlers. | Status code 200     |
| 2 | Изменение настроек паука: нет прав доступа к редактированию  | Пользователь авторизован. У пользователя есть readonly к проекту.               | Пользователь отправляет запрос PUT /api/project/:projectId /crawlers. | Status code 403     |
| 3 | Изменение настроек паука: не совпадает id проекта с id паука | Пользователь авторизован.   | Пользователь отправляет запрос PUT /api/project/:projectId /crawlers. | Status code 403     |

Таблица 9 — Тест кейсы: краулеры PUT

|                              |              |              |              |              |
|------------------------------|--------------|--------------|--------------|--------------|
| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |



### 6.1.5 Запуски (6.2)

| № | Название                 | Предусловие   | Описание  | Ожидаемый результат  |
|---|--------------------------|---|---|--|
| 1 | Список джобов            | Пользователь авторизован. У пользователя есть список из 20 запущенных job-ов в статусе finished, запущенных с пауками из разных проектов. | Пользователь отправляет запрос GET /api/projects/jobs/:limit, где указывает limit=10 и тип джобы finished. После этого отправляет тот же запрос со статусом running | Status code 200. Возвращается массив джобов со статусами finished и длиной 10. Отсортирован по id. |
| 2 | Список джобов: пагинация | Пользователь авторизован. У пользователя есть список из 20 запущенных job-ов в статусе finished, запущенных с пауками из разных проектов. | Пользователь отправляет запрос GET /api/projects/jobs/:limit, где указывает limit=10, excludeFrom=3 и тип джобы finished  | Status code 200. Возвращается массив джобов со статусами finished и длиной 2. Отсортирован по id.  |

Таблица 10 — Тест кейсы: список запусков GET

|                              |              |              |              |              |
|------------------------------|--------------|--------------|--------------|--------------|
| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

| № | Название  | Предусловие  | Описание  | Ожидаемый результат   |
|---|---|--|---|---|
| 3 | Создание запуска  | Пользователь авторизован. Имеет ReadAndWrite или Owner доступ к проекту projectId.                             | Клиент отправляет запрос POST /api/projects/:projectId/jobs | Status 200. Результат - id созданной задачи. При запросе GET /api/jobs status = Running задача с id возвращается. |
| 4 | Создание запуска: ReadOnly доступ                         | Пользователь авторизован. Имеет ReadOnly доступ к проекту projectId.   | Клиент отправляет запрос POST /api/projects/:projectId/jobs | Status 403. Сообщение об ошибке «NoPermission».   |
| 5 | Создание запуска: id краулера не соответствует id проекта | Пользователь авторизован. Имеет ReadAndWrite доступ к проекту projectId. Id паука не соответствует id проекта. | Клиент отправляет запрос POST /api/projects/:projectId/jobs | Status 403. Сообщение об ошибке «Crawler Doesnt Correspond To Project»  |

Таблица 11 — Тест кейсы: создание запуска POST

| № | Название   | Предусловие   | Описание   | Ожидаемый результат   |
|---|--|---|--|---|
| 6 | Отмена джобы                                     | Пользователь запустил джобу. Работа находится в статусе pending или running.      | Клиент отправляет запрос PUT /api/projects/:projectId/jobs/:jobScrapyId/:jobId, где jobScrapyId, jobId - id джобы в статусе running. | Status 200. Возвращается объект JobExecution с обновленными данными: status = finished. |
| 7 | Отмена джобы: id проекта не совпадает с id джобы | Пользователь авторизован в приложении. В проекте с id нет запущенных джоб.        | Клиент отправляет запрос PUT /api/projects/:projectId/jobs/:jobId.   | Status 403. Сообщение об ошибке «No Project Found».                                     |
| 8 | Отмена джобы: у пользователя readonly права      | Пользователь авторизован в приложении. Имеет readonly access к проекту projectId. | Клиент отправляет запрос PUT /api/projects/:projectId/jobs/:jobId.   | Status 403. Сообщение об ошибке «No Permission».  |

Таблица 12 — Тест кейсы: отмена запуска PUT

|                              |              |              |              |              |
|------------------------------|--------------|--------------|--------------|--------------|
| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

| №  | Название  | Предусловие   | Описание   | Ожидаемый результат  |
|----|---|---|--|--|
| 9  | Удаление запуска  | Пользователь авторизован. Имеет ReadAndWrite доступ к проекту projectId. Джоба находится в статусе finished.        | Клиент отправляет запрос DELETE /api/projects/:projectId/jobs/:jobId | Status 200. Результат - id удаленной джобы.                                  |
| 10 | Удаление запуска: джоба в статусе running/pending       | Пользователь авторизован. Имеет ReadAndWrite доступ к проекту projectId. Джоба находится в статусе running/pending. | Клиент отправляет запрос DELETE /api/projects/:projectId/jobs/:jobId | Status 422. Сообщение об ошибке «Couldn't delete job execution».             |
| 11 | Удаление запуска: ReadOnly доступ к проекту             | Пользователь авторизован. Имеет ReadOnly доступ к проекту projectId. Джоба находится в статусе finished.            | Клиент отправляет запрос DELETE /api/projects/:projectId/jobs/:jobId | Status 403. Сообщение об ошибке «No access».                                 |
| 12 | Удаление запуска: id работы не соответствует id проекта | Пользователь авторизован. Имеет ReadAndWrite доступ к проекту projectId. Id джобы не соответствует id проекта.      | Клиент отправляет запрос DELETE /api/projects/:projectId/jobs/:jobId | Status 403. Сообщение об ошибке «Job Execution Doesnt Correspond To Project» |

Таблица 13 — Тест кейсы: удаление запуска DELETE

|                              |              |              |              |              |
|------------------------------|--------------|--------------|--------------|--------------|
| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

### 6.1.6 Периодические задачи (6.2)

| № | Название                                | Предусловие   | Описание   | Ожидаемый результат  |
|---|---|---|--|--|
| 1 | Список периодических задач              | Пользователь авторизован и имеет Owner доступ к проекту с номером projectId.  | Пользователь отправляет запрос GET api/projects/:projectId/periodicJobs/:limit, где limit = 5                                    | Статус-код 200. Json с массивом из 5 периодических задач проекта в порядке убывания id.        |
| 2 | Список периодических задач с пагинацией | Пользователь авторизован и имеет Owner доступ к проекту с номером pId. Всего периодических задач в проекте 10 штук. | Пользователь отправляет запрос GET api/projects/:projectId/periodicJobs/:limit/:exclusiveFrom, где limit = 5, exclusiveFrom = 2. | Статус-код 200. Json с массивом из 2 периодических задач проекта в порядке убывания id (2, 1). |
| 3 | Список периодических задач: доступа нет | Пользователь авторизован и не имеет доступа к проекту с номером pId.  | Пользователь отправляет запрос GET api/projects/:projectId/periodicJobs/:limit с любым параметром limit.                         | Статус-код 403. Сообщение "NoPermission"   |

Таблица 14 — Тест кейсы: список периодических задач GET

|                              |              |              |              |              |
|------------------------------|--------------|--------------|--------------|--------------|
| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

| № | Название   | Предусловие   | Описание  | Ожидаемый результат   |
|---|--|---|---|---|
| 1 | Создание периодического запуска                                | Пользователь авторизован и имеет доступ ReadAndWrite к проекту projectId.   | Пользователь отправляет запрос POST /api/projects/:projectId/periodicJobs с телом запроса PeriodicJobCreateForm.  | Статус-код 200. Возвращается JSON с id созданной периодической задачи.      |
| 2 | Создание периодического запуска: некорректный cron-expression  | Пользователь авторизован и имеет доступ ReadAndWrite к проекту projectId.   | Пользователь отправляет запрос POST /api/projects/:projectId/periodicJobs с телом запроса PeriodicJobCreateForm, но cron-expression указывает в теле запроса некорректный (прим. «text»). | Статус-код 420. Возвращается сообщение с ошибкой «invalid cron expression». |
| 3 | Создание периодического запуска: права ReadOnly                | Пользователь авторизован и имеет доступ ReadOnly к проекту projectId.   | Пользователь отправляет запрос POST /api/projects/:projectId/periodicJobs с телом запроса PeriodicJobCreateForm.  | Статус-код 403. Сообщение об ошибке «NoPermission»                          |
| 4 | Создание периодического запуска: паук не соответствует проекту | Пользователь авторизован и имеет доступ ReadAndWrite к проекту projectId. Id паука в теле запроса не соответствует projectId. | Пользователь отправляет запрос POST /api/projects/:projectId/periodicJobs с телом запроса PeriodicJobCreateForm.  | Статус-код 403. Сообщение об ошибке «No Corresponding Crawler»              |

Таблица 15 — Тест кейсы: создание периодических запусков POST

|                              |              |              |              |              |
|------------------------------|--------------|--------------|--------------|--------------|
| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

| № | Название  | Предусловие   | Описание  | Ожидаемый результат   |
|---|---|---|---|---|
| 1 | Редактирование периодического запуска   | Пользователь авторизован и имеет ReadAndWrite доступ к проекту projectId, в котором есть периодическая задача periodicJobId, а также паук crawlerId.          | Пользователь отправляет запрос PUT /api/projects/:projectId/periodicJobs/:periodicJobId с телом запроса PeriodicJobChangeForm.  | Статус-код 200.   |
| 2 | Редактирование периодического запуска: некорректный cron-expression                     | Пользователь авторизован и имеет ReadAndWrite доступ к проекту projectId, в котором есть периодическая задача periodicJobId, а также паук crawlerId.          | Пользователь отправляет запрос PUT /api/projects/:projectId/periodicJobs/:periodicJobId с телом запроса PeriodicJobChangeForm, но cron-expression указывает в теле запроса некорректный (прим. "text"). | Статус-код 422. Сообщение об ошибке «invalid cron expression» |
| 3 | Редактирование периодического запуска: в проекте нет паука                              | Пользователь авторизован и имеет ReadAndWrite доступ к проекту projectId, в котором есть периодическая задача periodicJobId. В проекте нет паука с crawlerId. | Пользователь отправляет запрос PUT /api/projects/:projectId/periodicJobs/:periodicJobId с телом запроса PeriodicJobChangeForm с некорректным crawlerId в теле.  | Статус-код 403 и сообщение об ошибке.                         |
| 4 | Редактирование периодического запуска: пользователь имеет ReadOnly доступ к проекту     | Пользователь авторизован и имеет ReadOnly доступ к проекту projectId.   | Пользователь отправляет запрос PUT /api/projects/:projectId/periodicJobs/:periodicJobId с телом запроса PeriodicJobChangeForm.  | Статус-код 403 и сообщение об ошибке.                         |
| 5 | Редактирование периодического запуска: в проекте нет редактируемой периодической задачи | Пользователь авторизован и имеет ReadAndWrite доступ к проекту projectId. В проекте нет редактируемой задачи periodicJobId.                                   | Пользователь отправляет запрос PUT /api/projects/:projectId/periodicJobs/:periodicJobId с телом запроса PeriodicJobChangeForm.  | Статус-код 403 и сообщение об ошибке.                         |

Таблица 16 — Тест кейсы: редактирование информации о периодическом запуске PUT

|                              |              |              |              |              |
|------------------------------|--------------|--------------|--------------|--------------|
| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

| № | Название  | Предусловие  | Описание   | Ожидаемый результат |
|---|---|--|--|---------------------|
| 1 | Отмена запусков периодической задачи                      | Пользователь авторизован и имеет ReadAndWrite доступ к проекту projectId, у которого есть периодическая задача periodicJobId в статусе Enabled.  | Пользователь отправляет запрос PUT /api/projects/:projectId/periodicJobs/:periodicJobId/disable. | Статус-код 200.     |
| 2 | Отмена запусков периодической задачи: задача уже отменена | Пользователь авторизован и имеет ReadAndWrite доступ к проекту projectId, у которого есть периодическая задача periodicJobId в статусе Disabled. | Пользователь отправляет запрос PUT /api/projects/:projectId/periodicJobs/:periodicJobId/disable. | Статус-код 422.     |

Таблица 17 — Тест кейсы: отмена запуска периодических задач PUT

| № | Название  | Предусловие  | Описание  | Ожидаемый результат |
|---|---|--|---|---------------------|
| 1 | Возобновление запусков периодической задачи                               | Пользователь авторизован и имеет ReadAndWrite доступ к проекту projectId, у которого есть периодическая задача periodicJobId в статусе Enabled.  | Пользователь отправляет запрос PUT /api/projects/:projectId/periodicJobs/:periodicJobId/enable. | Статус-код 200.     |
| 2 | Возобновление запусков периодической задачи: задача уже в статусе enabled | Пользователь авторизован и имеет ReadAndWrite доступ к проекту projectId, у которого есть периодическая задача periodicJobId в статусе Disabled. | Пользователь отправляет запрос PUT /api/projects/:projectId/periodicJobs/:periodicJobId/enable. | Статус-код 422.     |

Таблица 18 — Тест кейсы: возобновление запусков периодических задач по расписанию PUT

|                              |              |              |              |              |
|------------------------------|--------------|--------------|--------------|--------------|
| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

| № | Название   | Предусловие   | Описание  | Ожидаемый результат                                |
|---|--|---|---|--|
| 1 | Удаление периодической задачи                            | Пользователь авторизован и имеет ReadAndWrite доступ к проекту projectId. У проекта есть периодическая задача periodicJobId в статусе enabled.  | Пользователь отправляет запрос DELETE /api/projects/:projectId/periodicJobs/:periodicJobId. | Статус код 200.                                    |
| 2 | Удаление периодической задачи: задача в статусе disabled | Пользователь авторизован и имеет ReadAndWrite доступ к проекту projectId. У проекта есть периодическая задача periodicJobId в статусе disabled. | Пользователь отправляет запрос DELETE /api/projects/:projectId/periodicJobs/:periodicJobId. | Статус код 200. Возвращается id удаленной задачи.  |
| 3 | Удаление периодической задачи: доступ Readonly           | Пользователь авторизован и имеет ReadOnly доступ к проекту projectId.   | Пользователь отправляет запрос DELETE /api/projects/:projectId/periodicJobs/:periodicJobId. | Статус код 403. Сообщение об ошибке «NoPermission» |

Таблица 19 — Тест кейсы: удаление периодических зада DELETE

|                              |              |              |              |              |
|------------------------------|--------------|--------------|--------------|--------------|
| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |



### 6.1.7 Результаты тестирования

```
[info] EmailValidatorTest:
[info] - EmailString.wrongEmail
[info] - EmailString.validEmail
[info] AuthorizationSpec:
[info] Authorize person
[info] - should signUp: OK
[info] - should signUp: userAlreadyExists
[info] - should signUp: invalid email format
[info] - should signIn: wrong credentials
[info] JobTestCase:
[info] CrawlerSpec:
[info] - GET crawlers: OK
[info] - GET crawlers: no access
[info] - PUT crawlers
[info] - PUT crawlers: ReadOnly access
[info] - PUT crawlers: spider not found
[info] ProjectSpec:
[info] ProjectsController
[info] - should GET list of projects for user
[info] - should GET list of projects: with pagination
[info] - should CREATE project
[info] - should PUT project's metadata
[info] - should PUT project's metadata: Readonly access - no permission
[info] - should PUT project's metadata: ReadAndWrite access
[info] - should DELETE project: Owner access
[info] - should DELETE project: NOT Owner access
[info] - should DELETE project: doesn't exist
[info] - should PUT deploy
[info] - should PUT deploy: wrong format file
[info] - should PUT deploy: no access
[info] MembershipSpec:
[info] MembershipController
[info] - should GET members: ReadAndWrite access
[info] - should GET members: no access to project
[info] - should DELETE member: Owner access
[info] - should DELETE member: ReadAndWrite access
[info] - should PUT member: Owner access
[info] - should PUT member: ReadAndWrite access
[info] ApplicationSpec:
[info] ApplicationController Logout GET
[info] - should should be unauthorized error
[info] - should redirect if user was found
[info] PeriodicJobSpec:
[info] PeriodicJobController
[info] - should GET periodic jobs: basic
[info] - should GET periodic jobs: pagination
[info] - should GET periodic jobs: no access to project
[info] - should POST periodic job: basic
[info] - should POST periodic job: invalid cron-expression
[info] - should POST periodic job: crawler doesn't correspond to project
[info] - should POST periodic job: ReadOnly access
[info] - should PUT periodic job: basic
[info] - should PUT periodic job: crawler does not correspond to project
[info] - should PUT periodic job: wrong cron-expression
[info] - should DELETE periodic job: enabled status
[info] - should DELETE periodic job: disabled status
```

| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
|------------------------------|--------------|--------------|--------------|--------------|
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

```
[info] - should DELETE periodic job: readonly access
[info] - should DELETE periodic job: no existing job with id
[info] - should PUT cancel periodic job: basic
[info] - should PUT cancel periodic job: already disabled
[info] - should PUT cancel periodic job: readonly access
[info] - should PUT cancel periodic job: no existing job with id
[info] - should PUT enable periodic job: basic
[info] - should PUT enable periodic job: already enabled
[info] - should PUT enable periodic job: readonly access
[info] - should PUT enable periodic job: no existing job found with id
[info] SettingsMergerTest:
[info] - SettingsMerger.basic
[info] - SettingsMerger.wrongInput
[info] - SettingsMerger.testPriority
[info] JobExecutionSpec:
[info] JobsController
[info] - should GET jobs
[info] - should GET jobs: pagination
[info] - should POST schedule: ordinary
[info] - should POST schedule: ReadOnly access
[info] - should POST schedule: ProjectId doesn't match CrawlerId
[info] - should PUT cancel
[info] - should PUT cancel: jobId doesn't match to projectId
[info] - should PUT cancel: ReadOnly access
[info] - should DELETE job
[info] - should DELETE job: still running
[info] - should DELETE job: ReadOnly access
[info] - should DELETE job: jobId doesn't match projectId
[info] ScalaTest
[info] Run completed in 1 minute, 33 seconds.
[info] Total number of tests run: 68
[info] Suites: completed 10, aborted 0
[info] Tests: succeeded 68, failed 0, canceled 0, ignored 0, pending 0
[info] All tests passed.
[info] Passed: Total 68, Failed 0, Errors 0, Passed 68
[success] Total time: 107 s, completed 11-Apr-2020 16:44:40
```

Листинг 1 — Результаты тестирования в консоли

## 6.2 Проверка требований к программной документации

Вся документация, представленная в требованиях, готова.

| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
|------------------------------|--------------|--------------|--------------|--------------|
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

## ПРИЛОЖЕНИЕ А

### Используемые понятия и определения

**Web scraping** – это сбор данных с различных интернет-ресурсов. Общий принцип его работы можно объяснить следующим образом: некий автоматизированный код выполняет GET-запросы на целевой сайт и получая ответ, парсит HTML-документ, ищет данные и преобразует их в заданный формат.

**Проект** – сущность для объединения и предоставления доступа к запускам/краулерам/периодическим задачам.

**Веб краулер** – программа, являющаяся составной частью поисковой системы и предназначенная для перебора страниц Интернета с целью занесения информации о них в базу данных поисковика. неотъемлемая часть проекта. Именно с помощью пауков пользователь может “краулить” сайты для сбора необходимой информации.

**Запуск** – единоразовый запуск краулера с настройками и аргументами, указанными для этого запуска.

**Периодический запуск** – запуск с множеством настроек, повторяющийся в определенные периоды времени (запуски по cron-expression).

| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
|------------------------------|--------------|--------------|--------------|--------------|
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

## Список источников

- [1] Github scrapyd/scrapyd [Электронный ресурс] URL: <https://github.com/scrapy/scrapyd> (Дата обращения: 16.04.2020, режим доступа: свободный)
- [2] Единая система программной документации – М.: ИПК, Издательство стандартов, 2000, 125 стр.
- [3] ScalaTest+Play [Электронный ресурс] URL: <http://www.scalatest.org/plus> (Дата обращения: 16.04.2020, режим доступа: свободный)
- [4] Testing - silhouette [Электронный ресурс] URL: <https://www.silhouette.rocks/docs/testing> (Дата обращения: 16.04.2020, режим доступа: свободный)
- [5] Postgresql [Электронный ресурс] URL: <https://www.postgresql.org> (Дата обращения: 16.04.2020, режим доступа: свободный)

|                              |              |              |              |              |
|------------------------------|--------------|--------------|--------------|--------------|
| Изм.                         | Лист         | № докум.     | Подп.        | Дата         |
| RU.17701729.04.13-01 51 01-1 |              |              |              |              |
| Инв. № подл.                 | Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата |

[illegible]