**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Образовательная программа «Программная инженерия»

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО  Научный руководитель,  Приглашенный преподаватель департамента программной инженерии    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.К. Горденко  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г. |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия»  профессор департамента программной инженерии, канд. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **Тестирующая система c заданиями в виде изображений**  **Программа и методика испытаний**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.10.03-01 51 01-1-ЛУ** | | |
|  |  | |
| Исполнитель:  студент группы БПИ206  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Г. В. Вавилов /  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г. | |
|  | | |
|  | |  |

**Москва 2022**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| УТВЕРЖДЕН  RU.17701729.05.11-01 51 01-1-ЛУ |  | |  | |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **Тестирующая система c заданиями в виде изображений**  **Программа и методика испытаний**  **RU.17701729.05.11-01 51 01-1**  **Листов 25** | | | | |
|  | |  | | |
|  | | |
|  | | | | |
|  | | | |  |

**Москва 2023**

Содержание

[1. Объект испытаний 4](#_Toc1)

[1.1. Наименование программы 4](#_Toc2)

[1.2. Область применения 4](#_Toc3)

[1.3. Обозначение испытуемой программы 4](#_Toc4)

[2. Цель испытаний 5](#_Toc5)

[3. Требования к программе 6](#_Toc6)

[3.1. Требования к функциональным характеристикам 6](#_Toc7)

[3.1.1. Требования к панели преподавателя 6](#_Toc8)

[3.1.2. Требования к серверной части 6](#_Toc9)

[3.2. Требования к надежности. 7](#_Toc10)

[3.3. Требования к информационной и программной совместимости. 7](#_Toc11)

[4. Требования к программной документации 10](#_Toc12)

[4.1. Состав программной документации 10](#_Toc13)

[5. Средства и порядок испытаний 11](#_Toc14)

[5.1. Технические средства 11](#_Toc15)

[5.2. Программные средства 11](#_Toc16)

[5.3. Порядок проведения испытаний 11](#_Toc17)

[6. Методы испытаний 12](#_Toc18)

[6.1. Функциональные требования 12](#_Toc19)

[6.1.1. Требования к панели преподавателя 12](#_Toc20)

[6.1.2. Требования к серверной части 16](#_Toc21)

[6.1.3. Требования к чат-боту 17](#_Toc22)

[6.2. Требованиям к информационной и программной совместимости 18](#_Toc23)

[6.3. Требования к надежности. 22](#_Toc24)

[Лист регистрации изменений 24](#_Toc25)

# Объект испытаний

## Наименование программы

* + 1. **Название темы разработки на русском языке:**

Тестирующая система c заданиями в виде изображений.

* + 1. **Название темы разработки на английском языке:**

Testing System with Tasks in the Form of Images.

## Область применения

Тестирующая система является важным инструментом для автоматизации проведения тестов в образовательных учреждениях. Преподавателям система предоставляет удобный функционал для создания учебных курсов, добавления учеников в курсы, создания тестовых заданий и объединения их в тесты. Также система позволяет проводить тестирование среди учеников курса и собирать статистику по каждому тесту.

Для учеников система предоставляет возможность проходить назначенные тесты и получать оценки своих знаний. Все процессы проходят в режиме онлайн, что облегчает и ускоряет процесс проведения тестов.

Таким образом, система является неотъемлемой частью образовательных учреждений, таких как школы, колледжи и университеты. Она позволяет проводить тестирование и оценивать знания студентов, а также использовать ее для для онлайн-образования.

## Обозначение испытуемой программы

Тестирующая система **blank**

# Цель испытаний

Цель проведения испытаний — проверка соответствия характеристик разработанной программы функциональным требованиям, требованиям к информационной и программной совместимости, требованиям к надежности, изложенных в программном документе «Тестирующая система c заданиями в виде изображений» Техническое задание.

# Требования к программе

## **Требования к функциональным характеристикам**

Приложение состоит из серверной части и двух клиентов – панели преподавателя и чат-бота в системе обмена сообщениями *telegram.*

## Требования к панели преподавателя

* + - 1. Пользователь должен иметь возможность зарегистрироваться в системе в качестве преподавателя.
      2. Преподаватель должен иметь возможность просматривать список учебных курсов, в которых он является преподавателем.
      3. Преподаватель должен иметь возможность создать новый учебный курс.
      4. Преподаватель должен иметь возможность добавить других преподавателей в учебный курс.
      5. Преподаватель должен иметь возможность добавить учеников в учебный курс
      6. Преподаватель должен иметь возможность создать тестовое задание.
      7. Преподаватель должен иметь возможность создать тест, состоящий из тестовых заданий.
      8. Преподаватель должен иметь возможность назначить тест в качестве элемента контроля для учебного курса.

## Требования к серверной части

* + - 1. Сервер должен реализовывать REST API[3] для управления ресурсами из клиентской части приложения
      2. Сервер должен иметь возможность генерации изображения с тестовым заданием в формате *png*[4].
      3. Требования к чат-боту
      4. Бот должен требовать от пользователя авторизацию при помощи электронной почты.
      5. Бот должен уведомлять ученика после добавления преподавателем теста в учебный курс, к которому принадлежит ученик.
      6. Ученик должен иметь возможность дать ответ на тестовое задание, вводя его через интерфейс бота.

## Требования к надежности.

Должна быть предусмотрена корректная обработка ошибок, связанных с некорректными действиями пользователя. В случае некорректных входных данных программа не должна завершаться аварийно или уходить в бесконечный цикл. Пользователь должен быть уведомлен о некорректных входных данных.

## Требования к информационной и программной совместимости.

* + 1. **Требования к информационным структурам.**

Требования предъявляются к REST API, который реализуется на серверной части приложения.

Ожидаемый формат ответа сервера в формате JSON на запрос

**GET api/<название ресурса>/<id ресурса>**  
{

"data": {

"id":<id ресурса> ,

"attributes": {

... <атрибуты ресурса>

}

},

"meta": {}

}

Описание аттрибутов ресурсов:

User {

Username: string;  
 createdAt: string;

    updatedAt: string;

email: string;

    assigned\_courses: Course[];

    owned\_courses: Course[];

    answers: Answer[];

}

Test {

    title: string,

    createdAt: string,

    updatedAt: string,

    publishedAt: null | string,

    course: Course;

    solutions: TestSolution[];

    tasks: Task[];

}

Task {

    name: string,

    createdAt: string,

    updatedAt: string,

    publishedAt: null | string,

    course: Course;

    answers: Answer[];

    fields: (RichTextField | CodeField)[];

}

Course {

    name: string,

    createdAt: string,

    updatedAt: string,

    tasks: Task[];

    students: User[];

    owners: User[];

    tests: Test[];

}

* + 1. **Требования к исходным кодам и языкам программирования.**

Исходный код программы должен быть написан на языках программирования HTML, CSS и TypeScript с возможностью компиляции в JavaScript.

* + 1. **Требования к программным средствам, используемым программой.**

Для развертывания системы на машине необходимы следующие программные средства:

* Операционная система Linux
* docker
* docker-compose
* git

# Требования к программной документации

## Состав программной документации

* + 1. «Тестирующая система c заданиями в виде изображений» Техническое задание (ГОСТ 19.201-78)
    2. «Тестирующая система c заданиями в виде изображений» Программа и методика испытаний (ГОСТ 19.301-79) [5]
    3. «Тестирующая система c заданиями в виде изображений» Пояснительная записка (ГОСТ 19.404-79)
    4. «Тестирующая система c заданиями в виде изображений» Текст программы (ГОСТ 19.401-78)
    5. «Тестирующая система c заданиями в виде изображений» Руководство оператора (ГОСТ 19.505-79)

# Средства и порядок испытаний

## Технические средства

Тестирование программы производилось на устройстве со следующими характеристиками:

* + - Процессор: AMD Ryzen 5 5600X 3.70 GHz
    - Видеокарта: Nvidia GeForce RTX 3060.
    - Операционная система: Windows 10
    - Оперативная память: 16 ГБ
    - Доступ в интернет.
    - Клавиатура.
    - Мышь.

## Программные средства

Сборка и тестирование системы производилось на подсистеме Windows для Linux, характеристики которой соответствуют п. 3.3.4 «Требования к программным средствам, используемым программой».  
 Микросервисы запускаются в отдельных docker-контейнерах в сети, созданной docker-compose. Файл конфигурации находится в корне проекта.  
 Исходный код компилируется при помощи typescript, после чего запускается средой исполнения Node.js.  
 Для тестирования панели администратора и чат-бота для социальной сети ВКонтакте использовался веб-браузер Firefox Developer Edition версии 101.0b5. Для тестирования чат-бота для сервиса сообщений telegram использовалось десктопное клиентское приложение telegram. Для тестирования конечных точек серверное части использовалась программа insomnia.

## Порядок проведения испытаний

Программа запускается на виртуальной машине, отвечающей требованиям к составу и параметрам технических средств.

Последовательно проверяются:

* Функциональные требования
* Требования к информационной и программной совместимости
* Требования к надежности

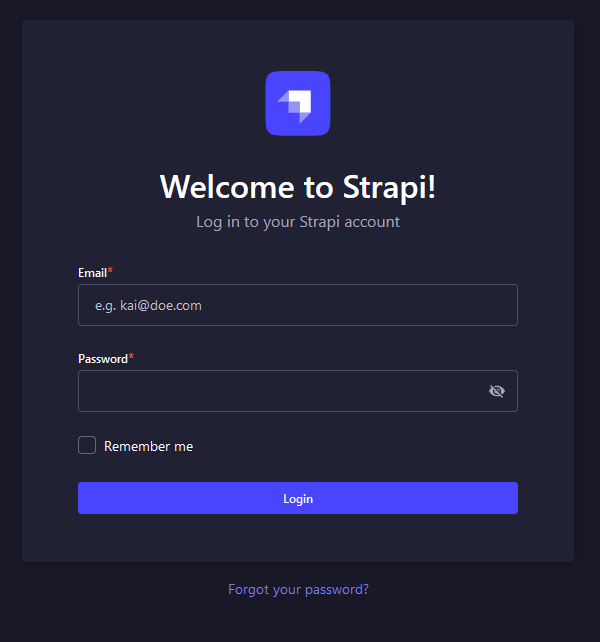
Дополнительно запускаются автоматические модульные тесты для микросервиса **webshot**.

# Методы испытаний

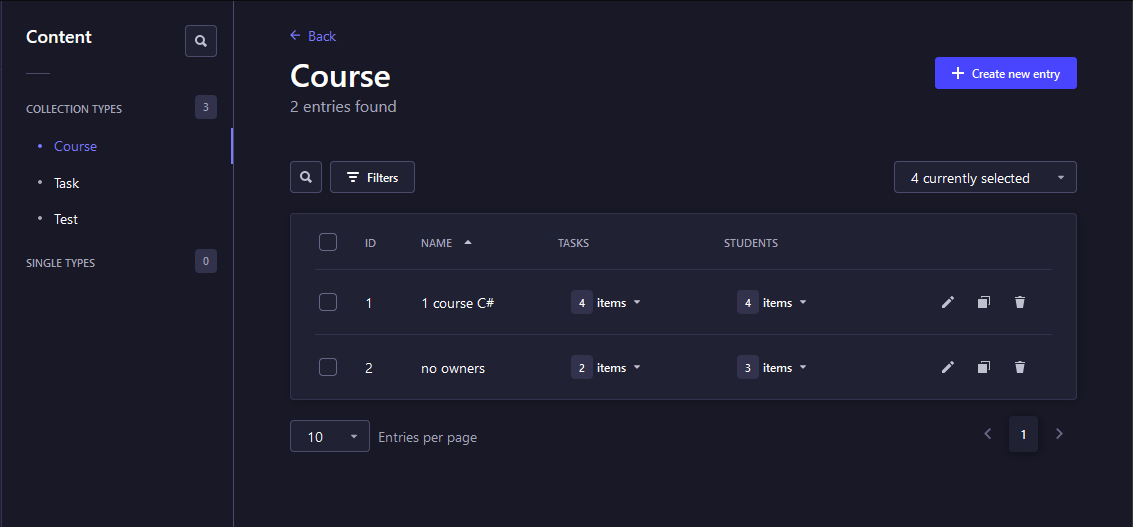
# Функциональные требования

# Требования к панели преподавателя

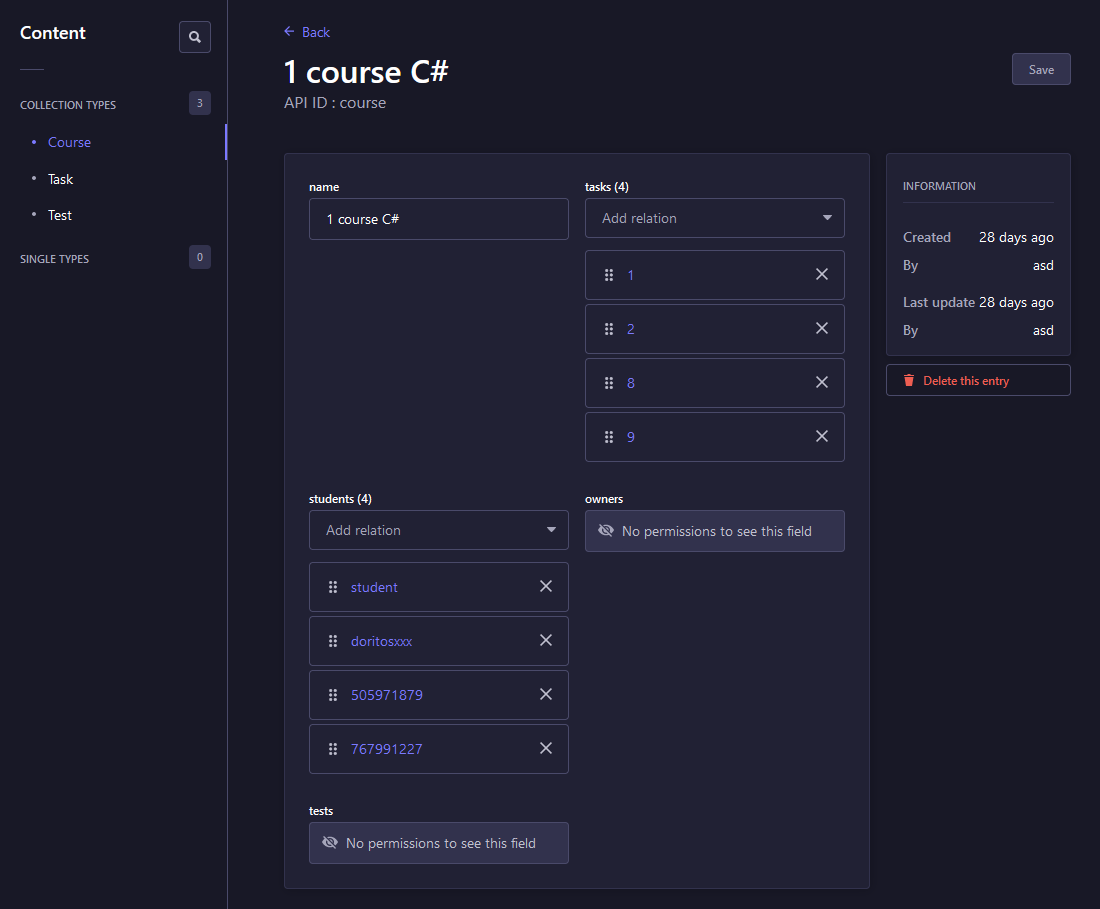
* + - 1. Пользователь должен иметь возможность зарегистрироваться в системе в качестве преподавателя  
          Реализовано в виде страницы входа. Пользователь не может зарегистрироваться самостоятельно, но может быть зарегистрирован администратором в целях безопасности.



* + - 1. Преподаватель должен иметь возможность просматривать список учебных курсов, в которых он является преподавателем. Реализовано



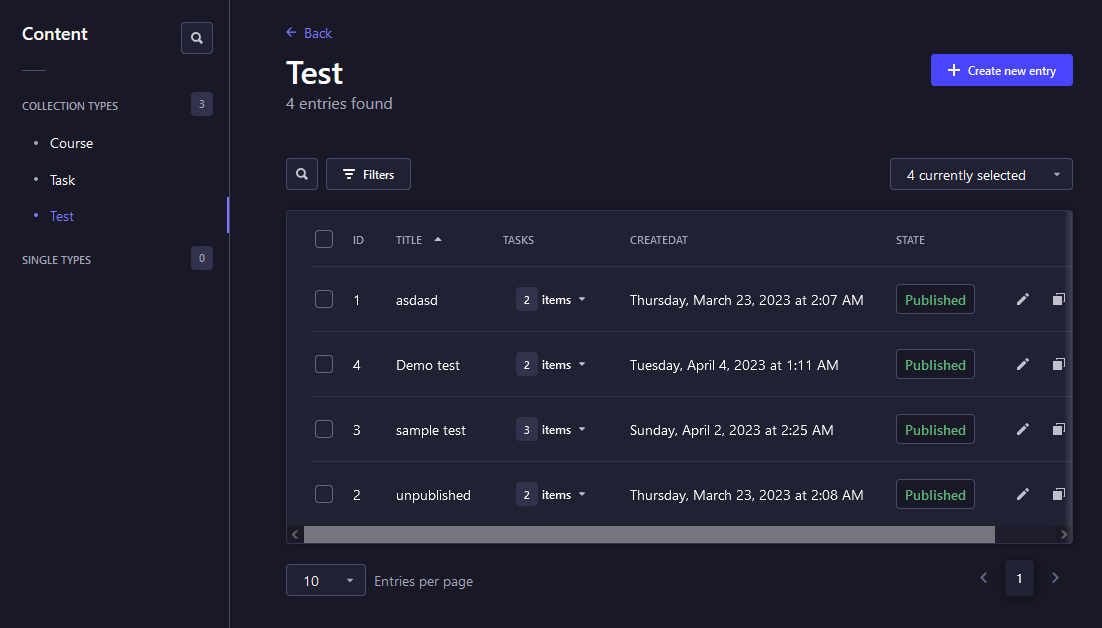
* + - 1. Преподаватель должен иметь возможность создать новый учебный курс. Реализовано. Возможность предоставляется при нажатии на кнопку «Create new entry»
      2. Преподаватель должен иметь возможность добавить других преподавателей в учебный курс.



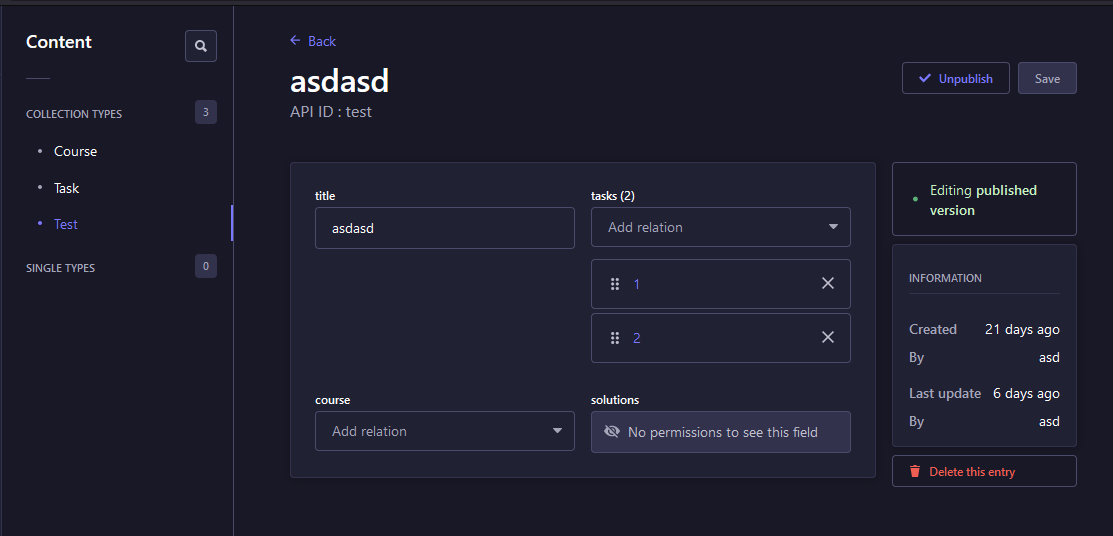
Реализовано. Пользователь может добавить преподавателей в учебный курс на странице редактирования учебного курса.

* + - 1. Преподаватель должен иметь возможность добавить учеников в учебный курс.   
          Пользователь может добавить учеников в учебный курс на странице редактирования учебного курса.
      2. Преподаватель должен иметь возможность создать тестовое задание.

Реализовано. Возможность предоставляется при нажатии на кнопку «Create new entry»



* + - 1. Преподаватель должен иметь возможность создать тест, состоящий из тестовых заданий.



Реализовано. Пользователь может создать тест и добавить в него тестовые задания на странице редактирования теста.

* + - 1. Преподаватель должен иметь возможность назначить тест в качестве элемента контроля для учебного курса.   
          Реализовано. Действие выполняется при нажатии кнопки «Publish»

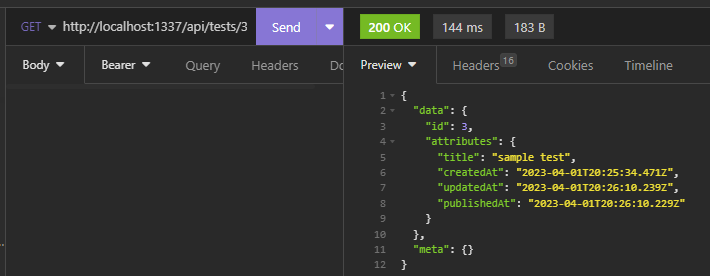
# Требования к серверной части

* + - 1. Сервер должен реализовывать REST API для управления ресурсами из клиентской части приложения.

Реализовано. Сервер предоставляет следующие конечные точки для взаимодействия с ресурсами системы:

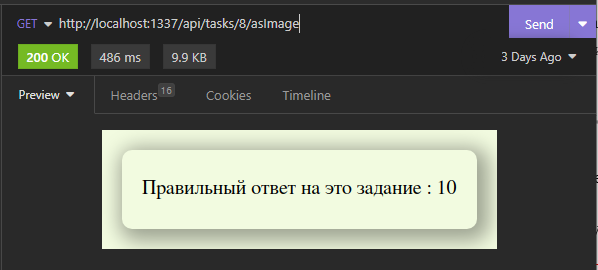
* GET /api/<entity>
* GET /api/<entity>/:id
* DELETE /api/<entity>/:id
* PUT /api/<entity>/:id
* POST /api/<entity>

Для Преподавателя доступны следующие значения <entity> : courses, tasks, tests  
 Пример запроса:



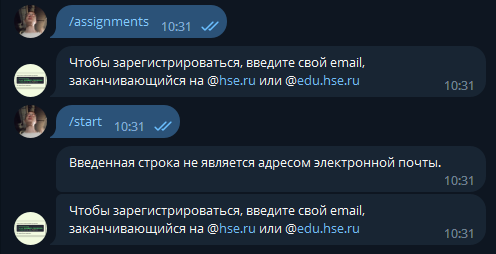
* + - 1. Сервер должен иметь возможность генерации изображения с тестовым заданием в формате *png*.

Реализовано в виде конечной точки GET /api/tasks/:id/asImage



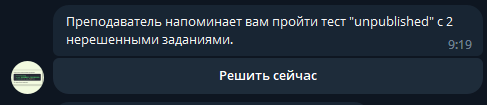
# Требования к чат-боту

* + - 1. Бот должен требовать от пользователя авторизацию при помощи электронной почты.   
          Реализовано



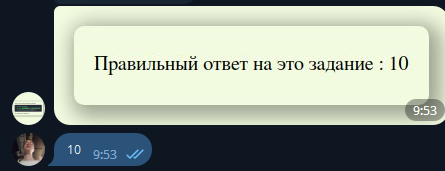
* + - 1. Бот должен уведомлять ученика после добавления преподавателем теста в учебный курс, к которому принадлежит ученик.

Реализовано в виде уведомления при нажатии на кнопку в панели управления тестом.



* + - 1. Ученик должен иметь возможность дать ответ на тестовое задание, вводя его через интерфейс бота.

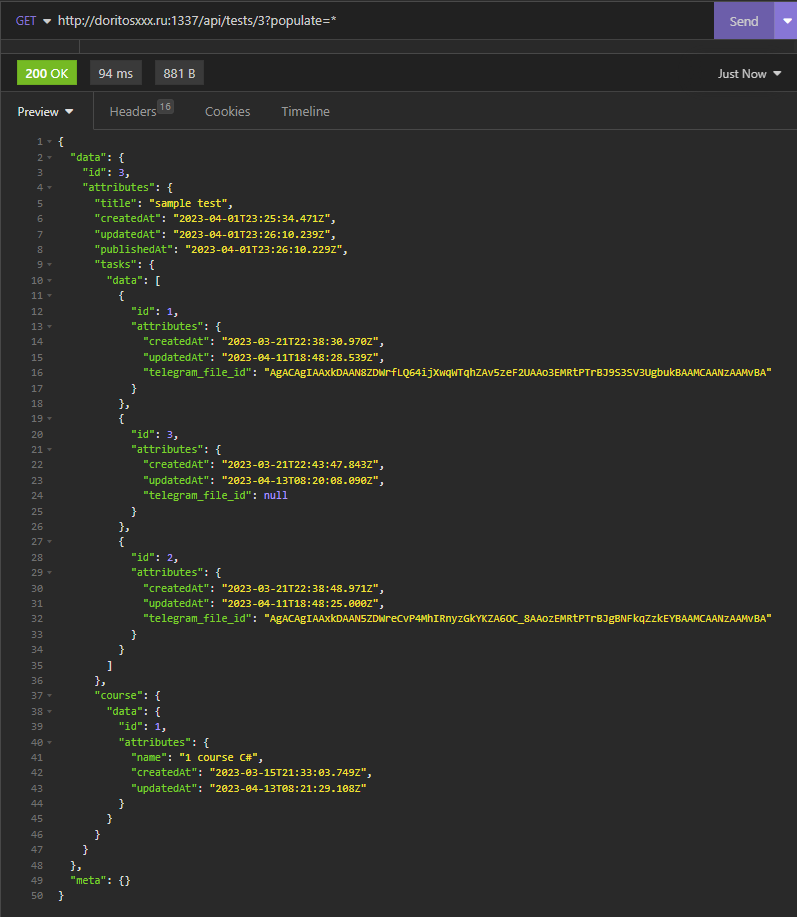
Реализовано.



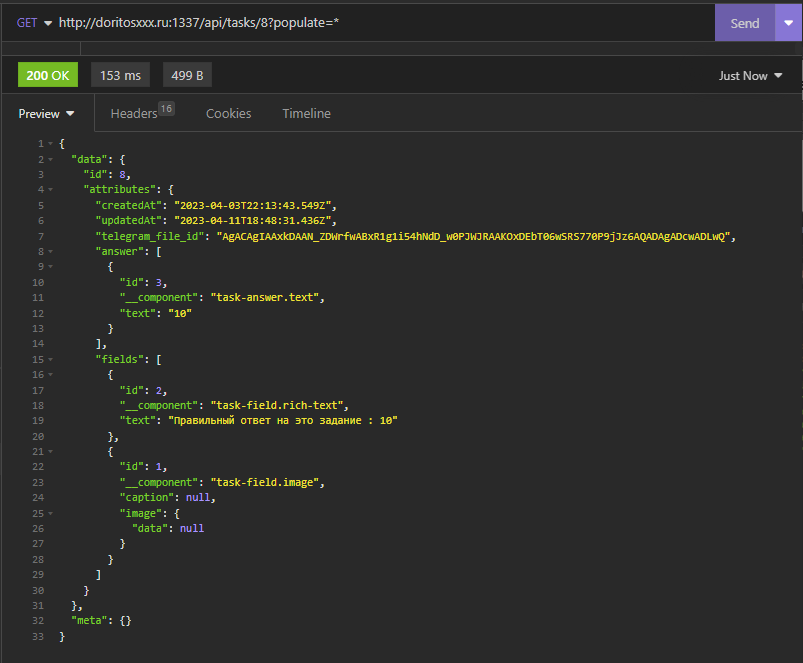
# Требованиям к информационной и программной совместимости

Последовательно проверяются требования к информационной и программной совместимости.

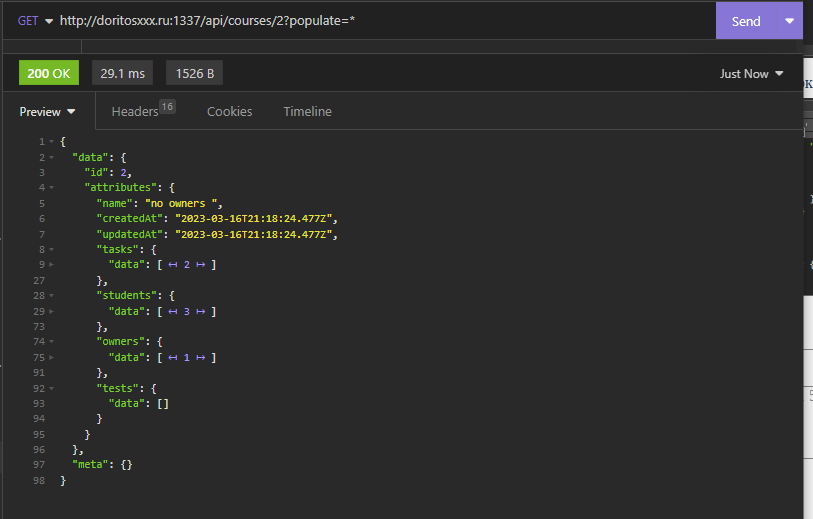
* + 1. **Требования к информационным структурам.**
       1. Ресурс Test



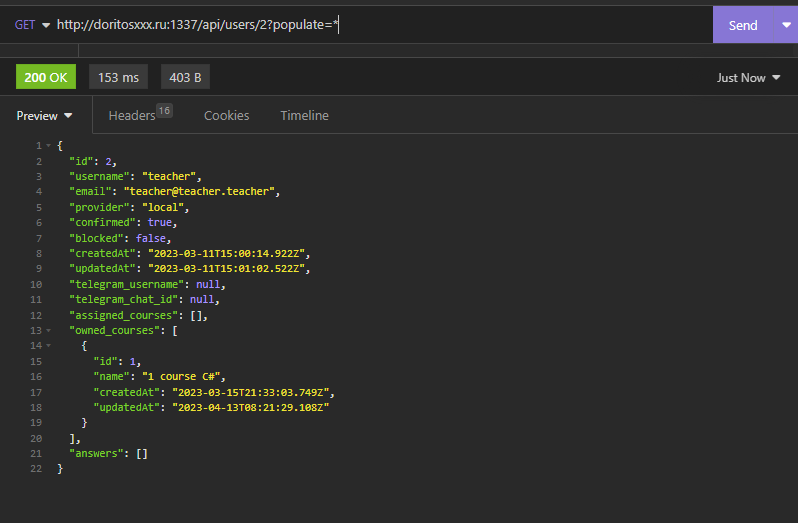
* + - 1. Ресурс Task



* + - 1. Ресурс Course



* + - 1. Ресурс User



* + 1. **Требования к исходным кодам и языкам программирования.**

Исходный код программы написан на языках программирования HTML, CSS и TypeScript. Настроена сборка и транспиляция исходного кода на TypeScript в JavaScript с целью дальнейшего запуска программы в браузере, либо при помощи среды исполнения Node.js на сервере.

# Требования к надежности.

* + 1. При непредвиденной остановке микросервиса он должен быть автоматически перезапущен в течение минуты.
    2. В случае возникновения ошибок, система должна обеспечивать уведомление администраторов о причинах и мерах по исправлению.
    3. При нагрузках, система должна продолжать работать без существенных падений производительности и отказов.
    4. Система должна иметь механизмы резервного копирования данных и восстановления после сбоев.
    5. REST API должен иметь механизмы защиты от несанкционированного доступа.
    6. Чат-боты для Вконтакте и Telegram должны обеспечивать гарантированную и надежную передачу данных между сервером и пользователями

# Лист регистрации изменений

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц в докум.) | № документа | Входящий № сопроводительного докум. и дата | Подп. | Дата |
| Изм. | Измененных | Замененных | Новых | Аннулированных |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |