**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Образовательная программа «Программная инженерия»

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО  Руководитель проекта,  доцент департамента больших данных и информационного поиска, к.ф-м.н    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Т. В. Вознесенская  «20» \_\_\_\_\_мая\_\_\_\_\_\_\_ 2022 г. |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия», кандидат технических наук  Автограф В В.В. Шилов  «20» мая 2022 г. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **Интерактивная визуализация данных приемной кампании**  **Программа и методика испытаний**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.10.03-01 51 01-1-ЛУ** | | |
|  |  | |
| Исполнитель  Изображение выглядит как текст  Автоматически созданное описаниестудент группы БПИ203  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Я. Ю. Щербаков/  «20» \_\_\_мая\_\_\_ 2022 г. | |
|  | | |
|  | |  |

**Москва 2022**

|  |  |  |  |
| --- | --- | --- | --- |
| УТВЕРЖДЕН  RU.17701729.10.03-01 51 01-1-ЛУ |  |  | |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **Интерактивная визуализация данных приемной кампании**  **Программа и методика испытаний**  **RU.17701729.10.03-01 51 01-1**  **Листов 22** | | | | |
|  |  | | | |
|  | | | |
|  | | | | |
|  | | |  | |

**Москва 2022**

СОДЕРЖАНИЕ

[АННОТАЦИЯ 3](#_Toc103951955)

[1. ОБЪЕКТ ИСПЫТАНИЙ 4](#_Toc103951956)

[**1.1. Наименование программы 4**](#_Toc103951957)

[**1.2. Область применения 4**](#_Toc103951958)

[2. ЦЕЛЬ ИСПЫТАНИЙ 5](#_Toc103951959)

[3. ТРЕБОВАНИЯ К ПРОГРАММЕ 6](#_Toc103951960)

[**3.1. Требования к составу выполняемых функций 6**](#_Toc103951961)

[**3.2. Требования к надежности 6**](#_Toc103951962)

[**3.3. Требования к интерфейсу 6**](#_Toc103951963)

[4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ 7](#_Toc103951964)

[**4.1. Состав программной документации 7**](#_Toc103951965)

[**4.2. Специальные требования к программной документации 7**](#_Toc103951966)

[5. СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ 8](#_Toc103951967)

[**5.1. Технические средства, используемые во время испытаний 8**](#_Toc103951968)

[**5.2. Программные средства, используемые во время испытаний 8**](#_Toc103951969)

[**5.3. Порядок проведения испытаний 8**](#_Toc103951970)

[6. МЕТОДЫ ИСПЫТАНИЙ 9](#_Toc103951971)

[**6.1. Испытание выполнения требований к программной документации 9**](#_Toc103951972)

[**6.2. Проверка требований к интерфейсу 9**](#_Toc103951973)

[**6.3. Проверка требований к функциональным характеристикам 10**](#_Toc103951974)

[**6.3.1. Добавление абитуриента 10**](#_Toc103951975)

[**6.3.2. Редактирование абитуриента 11**](#_Toc103951976)

[**6.3.3. Удаление абитуриента 12**](#_Toc103951977)

[**6.3.4. Удаление всех абитуриентов 12**](#_Toc103951978)

[**6.3.5. Получение абитуриента по индивидуальному номеру 13**](#_Toc103951979)

[**6.3.6. Получение всех абитуриентов 14**](#_Toc103951980)

[**6.3.7. Получение абитуриентов по году поступления 15**](#_Toc103951981)

[**6.3.8. Получение абитуриентов по различным фильтрам 16**](#_Toc103951982)

[**6.3.9. Получение статистики по регионам 18**](#_Toc103951983)

[**6.3.10. Добавление абитуриентов при помощи загрузки документа 19**](#_Toc103951984)

[ПРИЛОЖЕНИЕ 1 21](#_Toc103951985)

[ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ 22](#_Toc103951986)

**АННОТАЦИЯ**

Программа и методика испытаний – это документ, в котором содержится информация о программном продукте, а также полное описание приемочных испытаний для данного программного продукта.

Настоящая Программа и методика испытаний для «Интерактивная визуализация данных приемной кампании» содержит следующие разделы: «Объект испытаний», «Цель испытаний», «Требования к программе», «Требования к программной документации», «Средства и порядок испытаний».

В разделе «Объект испытаний» указано наименование и область применения программы.

В разделе «Цель испытаний» указана цель проведения испытаний. Раздел «Требования к программе» содержит основные требования к программе, которые подлежат проверке во время испытаний.

Раздел «Требования к программным документам» содержит состав программной документации, которая представляется на испытания.

Раздел «Средства и порядок испытаний» содержит информацию о технических и программных средствах, которые следует использовать во время испытаний, а также порядок этих испытаний.

Раздел «Методы испытаний» содержит информацию об используемых методах испытаний.

1. **ОБЪЕКТ ИСПЫТАНИЙ**

## Наименование программы

Наименование программы на русском языке: «Интерактивная визуализация данных приёмной кампании».

Наименование программы на английском языке**: «**Interactive visualization of admission campaign data».

## Область применения

Программа была задумана преимущественно для помощи сотрудникам приемной компании НИУ ВШЭ, обладающих доступом к персональному компьютеру (в рабочих целях).

1. **ЦЕЛЬ ИСПЫТАНИЙ**

Испытания проводились с целью проверки корректности выполнения функций программы, перечисленных в разделе «Требования к программе».

1. **ТРЕБОВАНИЯ К ПРОГРАММЕ**
   1. **Требования к составу выполняемых функций**

Программа должна соответствовать следующим требованиям к составу выполняемых функций, указанным в документе «Интерактивная визуализация данных приемной кампании. Техническое задание»:

* обеспечить возможность хранения и аккумулирования информации об абитуриентах;
* обеспечить возможность изменения информации об абитуриентах;
* обеспечить возможность удаления абитуриента;
* обеспечить возможность удаления всех абитуриентов;
* обеспечить возможность добавления информации об абитуриенте в формате CSV и при помощи запросов в формате JSON;
* обеспечить возможность фильтрации абитуриентов;
* обеспечить возможность вывода статистики по регионам.
  1. **Требования к надежности**

Web-приложение должно корректно обрабатывать все обращения пользователя к серверу и отображать его результаты. Должны быть предусмотрены возможные некорректные действия пользователей, в случае которых не должно возникать ошибок или исключений.

* 1. **Требования к интерфейсу**

Реализация серверной части командного проекта не предусматривала наличие визуализации, но документация для приложения должна быть создана с помощью Swagger.

1. **ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ**

## Состав программной документации

* Техническое задание (ГОСТ 19.201-78)
* Программа и методика испытаний (ГОСТ 19.301-78)
* Пояснительная записка (ГОСТ 19.404-79)
* Текст программы (ГОСТ 19.401-78)
* Руководство оператора (ГОСТ 19.505-79)

## Специальные требования к программной документации

* Все документы к программе должны быть выполнены в соответствии с ГОСТ 19.106-78 [6] и ГОСТ к этому виду документа.
* Вся документация сдается в электронном виде в формате .pdf или .docx. Все документы перед защитой курсовой работы должны быть загружены в информационно-образовательную среду НИУ ВШЭ Smart LMS (Learning management system).

1. **СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ**

## Технические средства, используемые во время испытаний

Для полноценного тестирования web-приложения необходим компьютер с доступом в интернет, обладающий следующими минимальными характеристиками:

* процессор Intel или AMD с 4 ядрами, 1.7 GHz или мощнее;
* оперативная память 4 GB RAM или более;
* графический процессор с памятью не менее 128 Мб;
* место в памяти ПК: 3 ГБ или более.

## Программные средства, используемые во время испытаний

Для проведения тестирования на компьютере должны быть установлены:

* .NET Framework 5;
* Visual Studio 2019;
* Microsoft SQL Server Management Studio 18;
* браузер.

## Порядок проведения испытаний

Испытания должны проводиться в следующем порядке:

* проверка требований к программной документации;
* конфигурирование базы данных и развертывание web-приложения;
* проверка требований к интерфейсу;
* проверка требований к функциональным характеристикам;

1. **МЕТОДЫ ИСПЫТАНИЙ**

## Испытание выполнения требований к программной документации

Состав программной документации проверяется визуально, проверяется наличие всех подписей и наличие программной документации в системе Smart LMS. Также визуально проверяется соответствие документации требованиям ГОСТ. Все документы удовлетворяют представленным требованиям.

## Проверка требований к интерфейсу

Документация к приложению Swagger корректно отображается в разных браузерах.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1.1. Интерфейс документации Swagger



Рисунок 1.2. Схемы моделей ответов на запросы Swagger.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1.3. Схема EntrantView.

## Проверка требований к функциональным характеристикам

### Добавление абитуриента

Для добавления абитуриента требуется нажать на кнопку «Try it out» в верхнем правом углу конечной точки «/Entrant/UpdateEntrant» и заполнить тело запроса в формате JSON данными об абитуриенте, после чего нажать на кнопку «Execute» для выполнения запроса. В случае успешного выполнения запроса (код 200) пользователь получит ответ с новым уникальным номером абитуриента, по которому он будет храниться в базе данных. В случае неудачного выполнения (код 400) пользователь получит ответ с сообщением об отловленной ошибке.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2.1. Запроса добавления абитуриента.

Изображение выглядит как текст, снимок экрана, монитор, внутренний

Автоматически созданное описание

Рисунок 2.2. Ответ на запрос добавления абитуриента.

### Редактирование абитуриента

Для редактирования абитуриента требуется нажать на кнопку «Try it out» в верхнем правом углу конечной точки «/Entrant/UpdateEntrant» и заполнить тело запроса в формате JSON изменёнными данными об абитуриенте с указанием его индивидуального номера в базе данных, после чего нажать на кнопку «Execute» для выполнения запроса. В случае успешного выполнения запроса (код 200) данные об абитуриенте с указанным индивидуальным номером будут изменены и сохранены в базе данных. В случае неудачного выполнения (код 400) пользователь получит ответ с сообщением об отловленной ошибке.

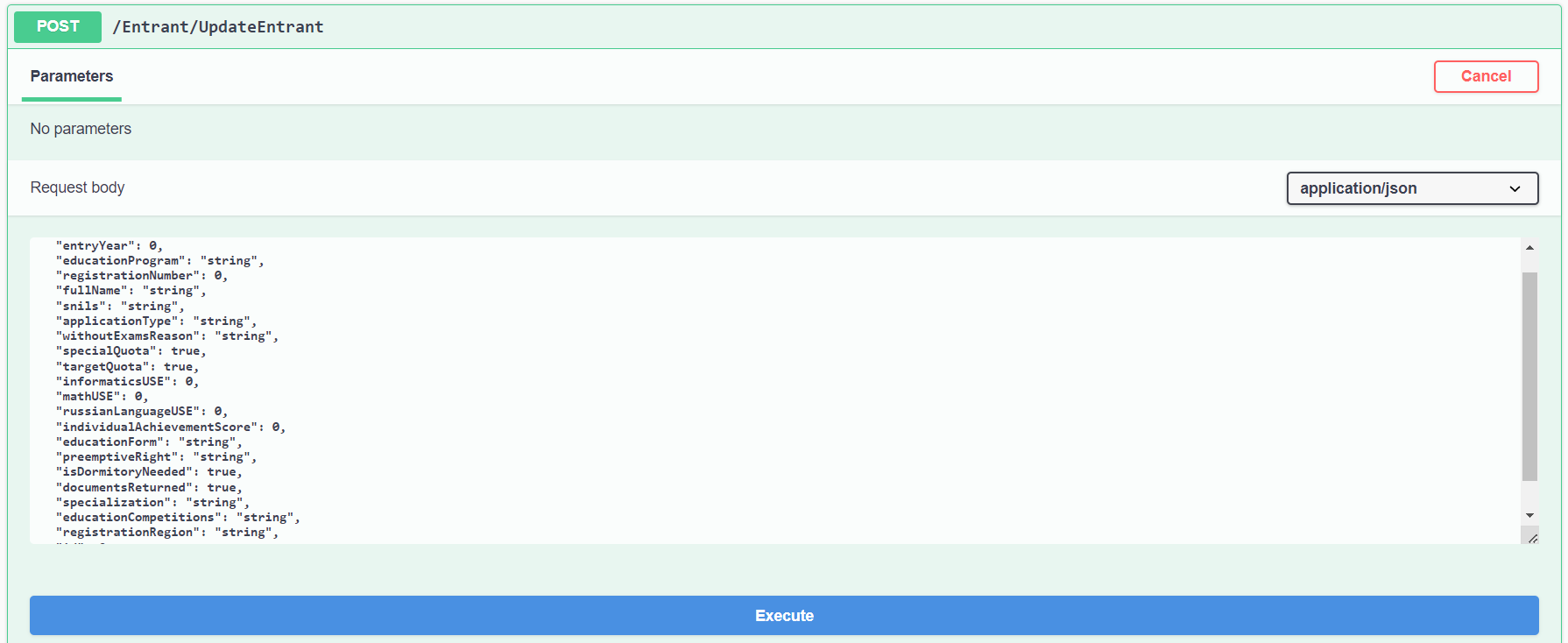


Рисунок 3.1. Запроса редактирования абитуриента.

Изображение выглядит как текст, монитор, внутренний, снимок экрана

Автоматически созданное описание

Рисунок 3.2. Ответ на запрос редактирования абитуриента.

### Удаление абитуриента

Для удаления абитуриента требуется нажать на кнопку «Try it out» в верхнем правом углу конечной точки «/Entrant/RemoveEntrant/{id}» и заполнить параметр запроса индивидуальным номером абитуриента, после чего нажать на кнопку «Execute» для выполнения запроса. В случае успешного выполнения запроса (код 200) данные об абитуриенте с указанным индивидуальным номером будут удалены из базы данных. В случае неудачного выполнения (код 400) пользователь получит ответ с сообщением об отловленной ошибке.

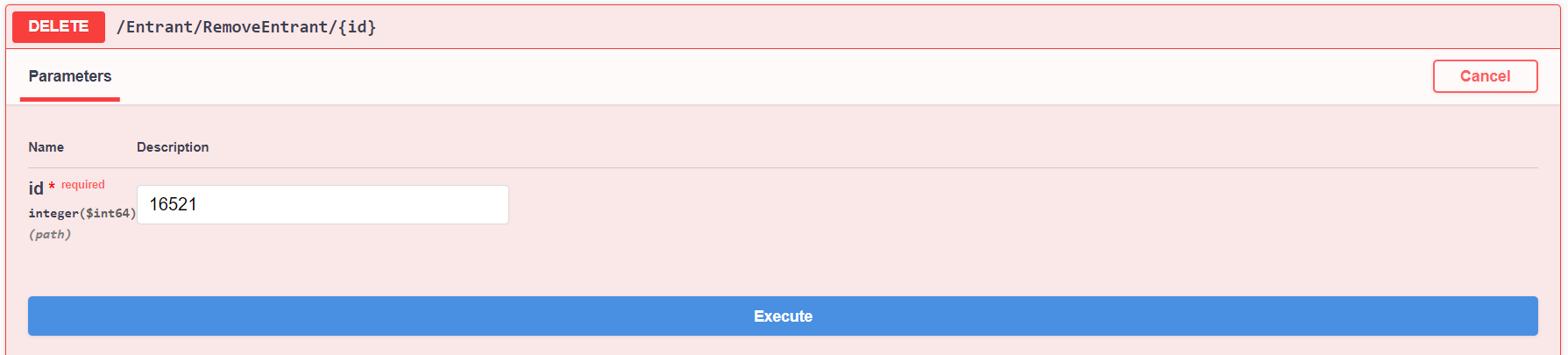


Рисунок 4.1. Запроса удаления абитуриента.

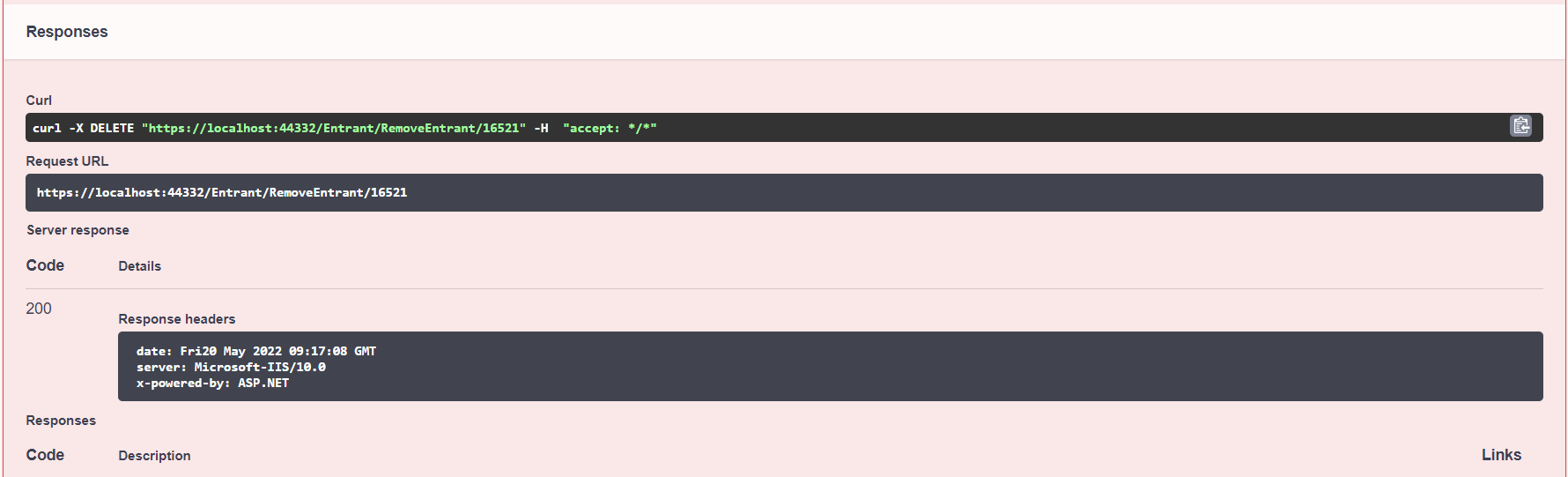


Рисунок 4.2. Ответ на запрос удаления абитуриента.

### Удаление всех абитуриентов

Для удаления всех абитуриента требуется нажать на кнопку «Try it out» в верхнем правом углу конечной точки «/Entrant/RemoveAllEntrants», после чего нажать на кнопку «Execute» для выполнения запроса. В случае успешного выполнения запроса (код 200) данные всех абитуриентов будут удалены из базы данных. В случае неудачного выполнения (код 400) пользователь получит ответ с сообщением об отловленной ошибке.

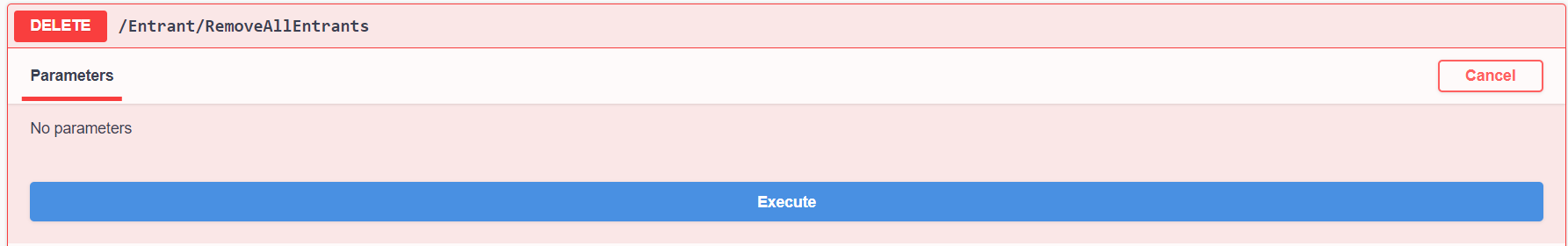


Рисунок 5.1. Запроса удаления всех абитуриентов.

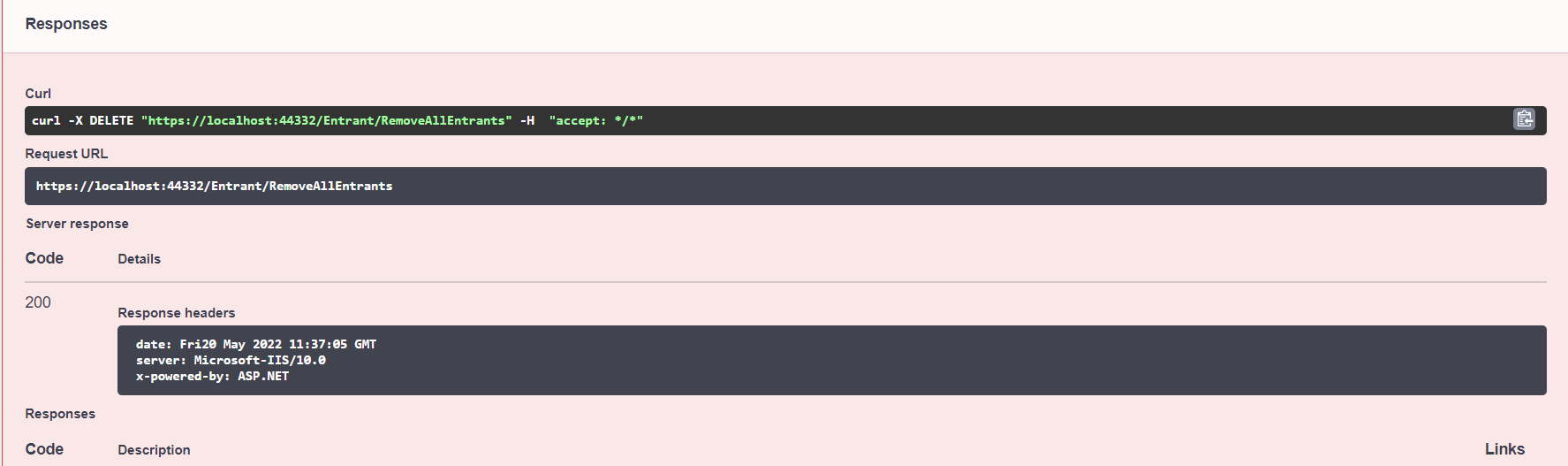


Рисунок 5.2. Ответ на запрос удаления всех абитуриентов.

### Получение абитуриента по индивидуальному номеру

Для получения данных абитуриента требуется нажать на кнопку «Try it out» в верхнем правом углу конечной точки «/Entrant/GetByEntrantId/{id}» и заполнить параметр запроса индивидуальным номером абитуриента, после чего нажать на кнопку «Execute» для выполнения запроса. В случае успешного выполнения запроса (код 200) пользователь получит ответ с полной информацией об абитуриенте в формате JSON. В случае неудачного выполнения (код 400) пользователь получит ответ с сообщением об отловленной ошибке.

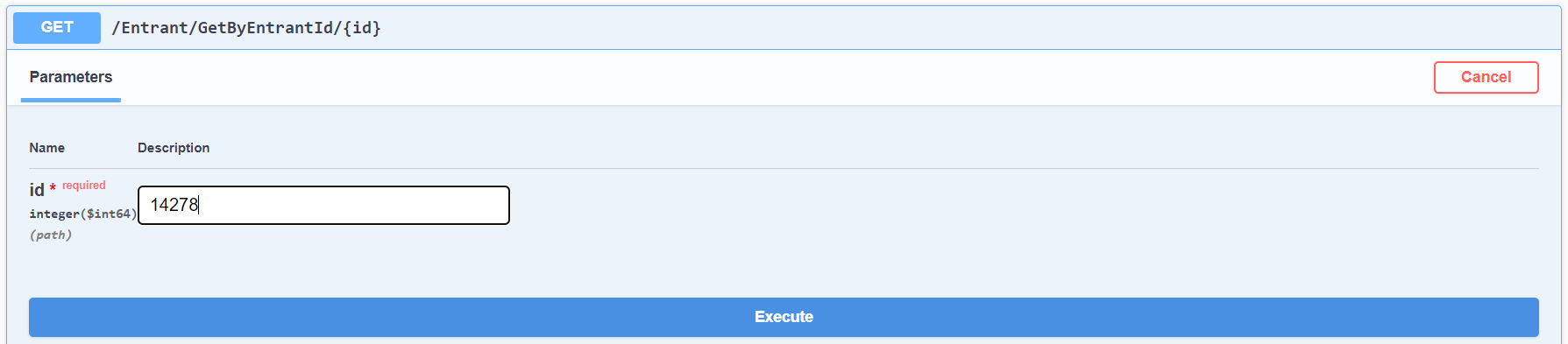


Рисунок 6.1. Запрос получения абитуриента по ID.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 6.2. Ответ на запрос получения абитуриента по ID.

### Получение всех абитуриентов

Для получения данных обо всех абитуриентах требуется нажать на кнопку «Try it out» в верхнем правом углу конечной точки «/Entrant/GetAllEntrants» и заполнить параметры запроса для механизма построчного отображения данных, после чего нажать на кнопку «Execute» для выполнения запроса. В случае успешного выполнения запроса (код 200) пользователь получит ответ с полной информацией обо всех абитуриентах в указанном количестве по указанному номеру страницы в формате JSON. В случае неудачного выполнения (код 400) пользователь получит ответ с сообщением об отловленной ошибке.

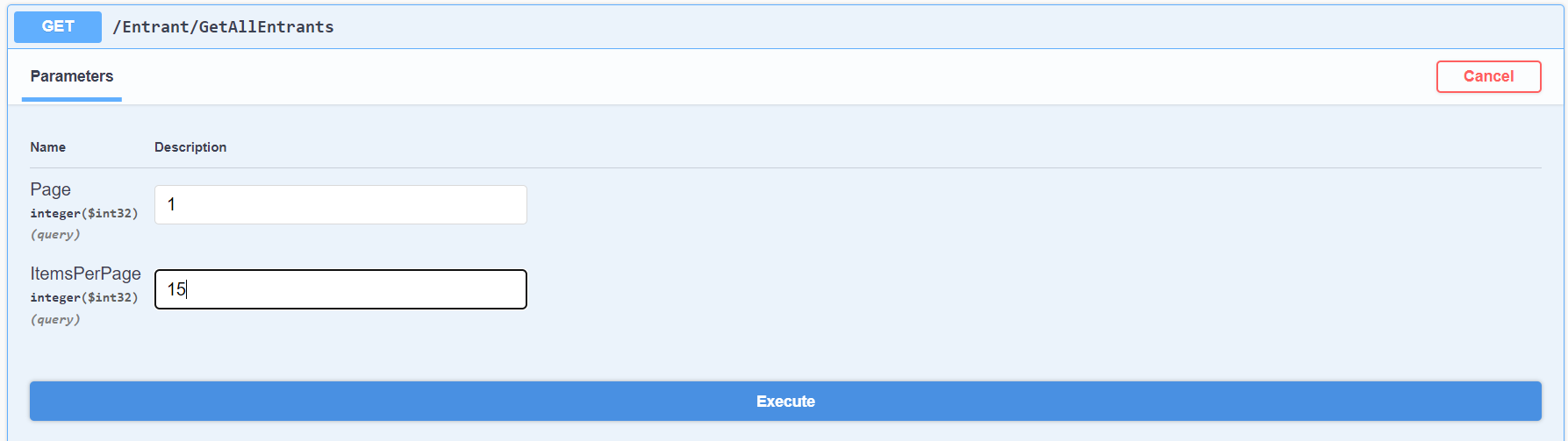


Рисунок 7.1. Запрос получения всех абитуриентов.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 7.2. Ответ на запрос получения всех абитуриентов.

### Получение абитуриентов по году поступления

Для получения данных абитуриентов по году поступления требуется нажать на кнопку «Try it out» в верхнем правом углу конечной точки «/Entrant/GetEntrantsByEntryYear» и заполнить параметры запроса для механизма построчного отображения данных и указать год поступления, после чего нажать на кнопку «Execute» для выполнения запроса. В случае успешного выполнения запроса (код 200) пользователь получит ответ с полной информацией обо всех абитуриентах в указанном количестве по указанному номеру страницы и указанному году поступления в формате JSON. В случае неудачного выполнения (код 400) пользователь получит ответ с сообщением об отловленной ошибке.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 8.1. Запрос получения абитуриентов по году поступления.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 8.1. Ответ на запрос получения абитуриентов по году поступления.

### Получение абитуриентов по различным фильтрам

Для получения данных абитуриентов по различным фильтрам требуется нажать на кнопку «Try it out» в верхнем правом углу конечной точки «/Entrant/FilterEntrants» и заполнить желаемые параметры фильтрации (все они являются опциональными), заполнить параметры запроса для механизма построчного отображения данных, после чего нажать на кнопку «Execute» для выполнения запроса. В случае успешного выполнения запроса (код 200) пользователь получит ответ с полной информацией обо всех отфильтрованных абитуриентах в указанном количестве по указанному номеру страницы и указанному году поступления в формате JSON. В случае неудачного выполнения (код 400) пользователь получит ответ с сообщением об отловленной ошибке.

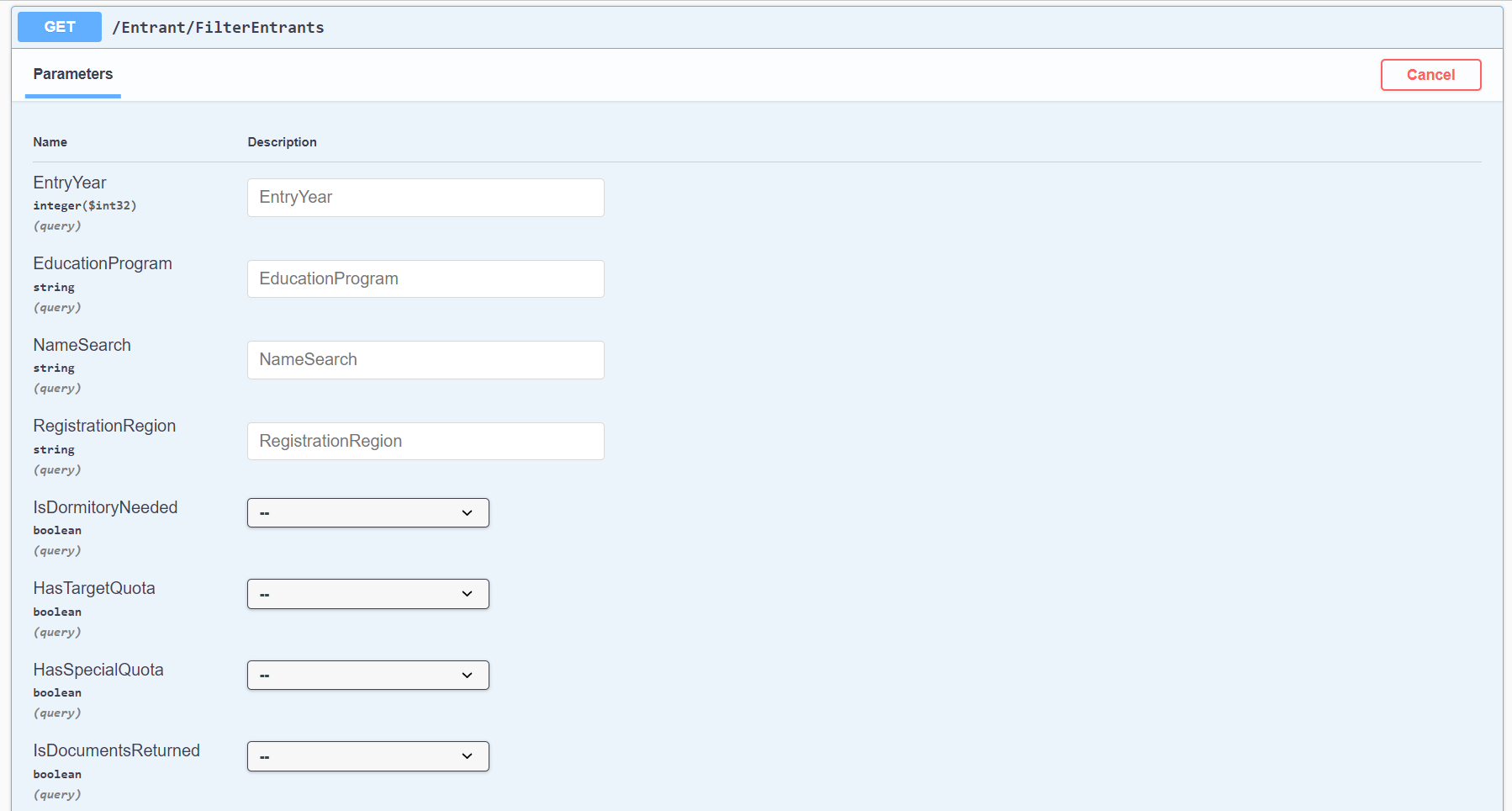


Рисунок 9.1. Примеры фильтров.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 9.2. Примеры фильтров (продолжение).

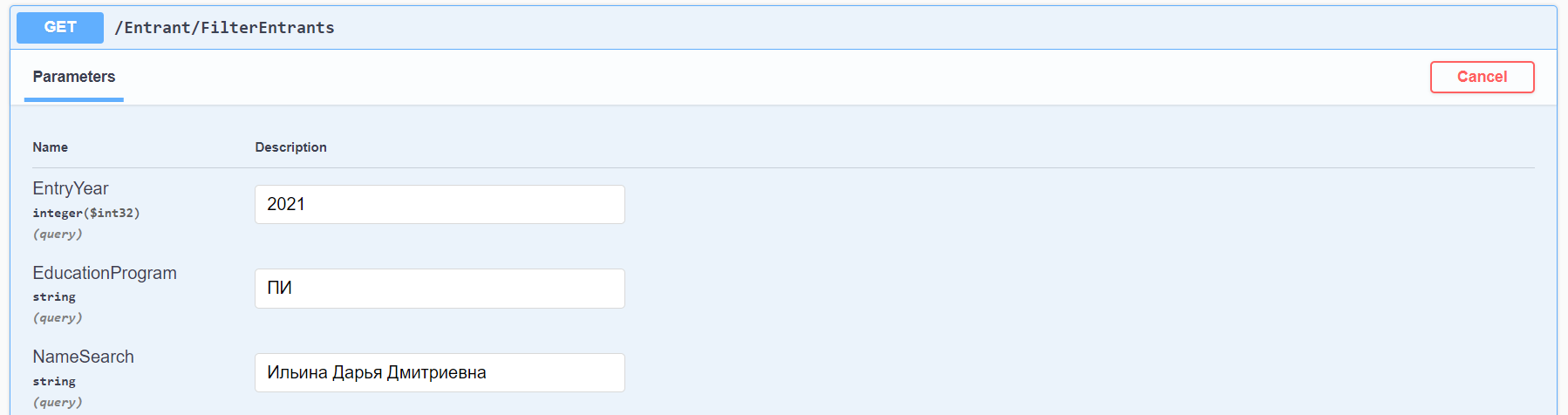


Рисунок 9.3. Запрос получения абитуриентов по фильтрам.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 9.4. Ответ на запрос получения абитуриентов по фильтрам.

### Получение статистики по регионам

Для получения статистики по регионам требуется нажать на кнопку «Try it out» в верхнем правом углу конечной точки «/Entrant/GetStaticticByRegions», заполнить параметры годом поступления с названием образовательной программы и заполнить параметры запроса для механизма построчного отображения данных, после чего нажать на кнопку «Execute» для выполнения запроса. В случае успешного выполнения запроса (код 200) пользователь получит ответ со статистикой по регионам в указанном количестве по указанному номеру страницы и указанному году поступления в формате JSON. В случае неудачного выполнения (код 400) пользователь получит ответ с сообщением об отловленной ошибке.

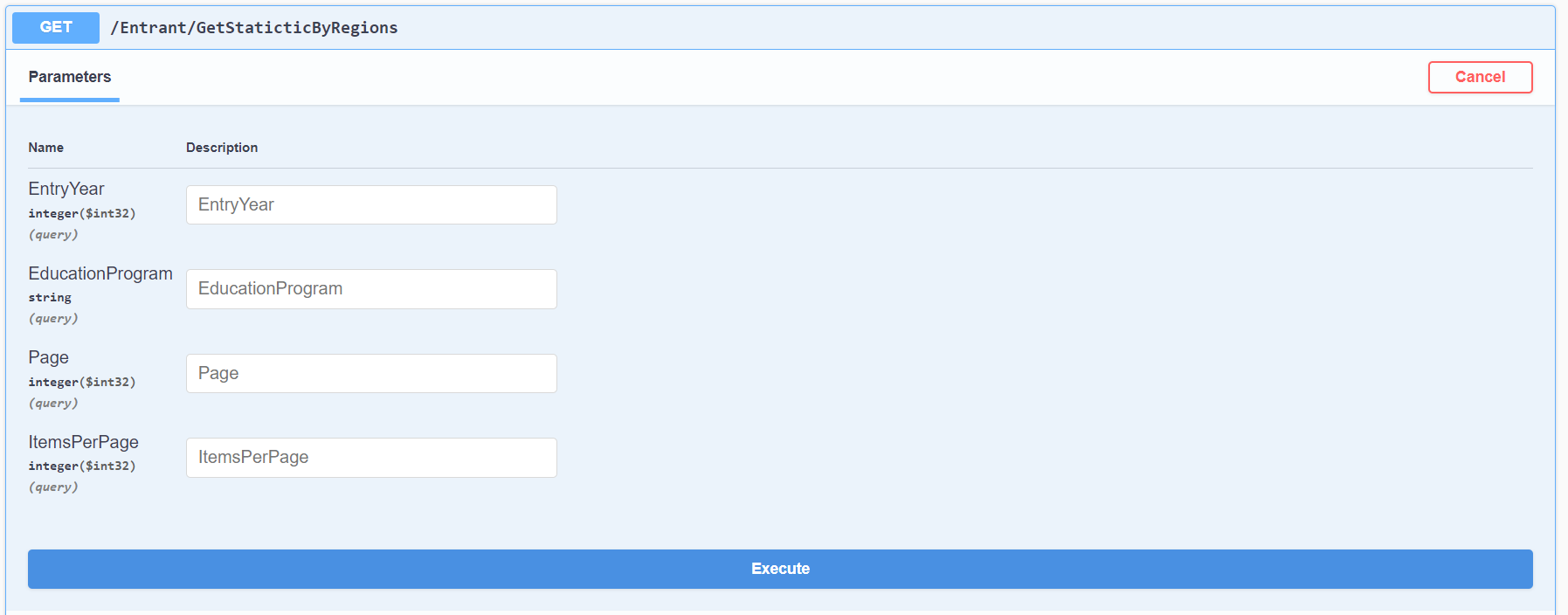


Рисунок 10.1. Модель получения статистики по регионам.

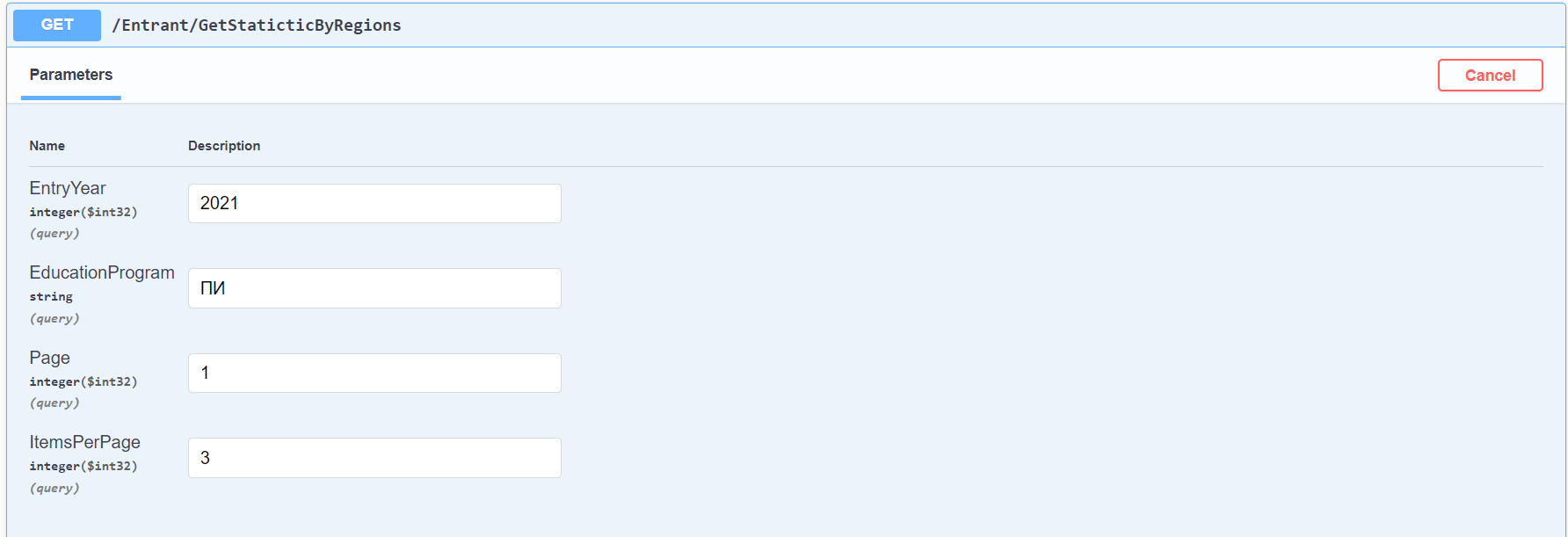


Рисунок 10.2. Запрос получения статистики по регионам.

Изображение выглядит как текст, снимок экрана, монитор, экран

Автоматически созданное описание

Рисунок 10.3. Ответ на запрос получения статистики по регионам.

### Добавление абитуриентов при помощи загрузки документа

Для добавления абитуриентов при помощи загрузки документа требуется нажать на кнопку «Try it out» в верхнем правом углу конечной точки «/Entrant/UploadDocument» и ввести параметры года поступления с наименованием образовательной программы, после чего необходимо нажать на кнопку «Выберите файл» для загрузки необходимого файла со специальными данными об абитуриентах в формате CSV и «Execute» для выполнения запроса. В случае успешного выполнения запроса (код 200) данные всех уникальных абитуриентов будут загружены в базу данных. В случае неудачного выполнения (код 400) пользователь получит ответ с сообщением об отловленной ошибке.

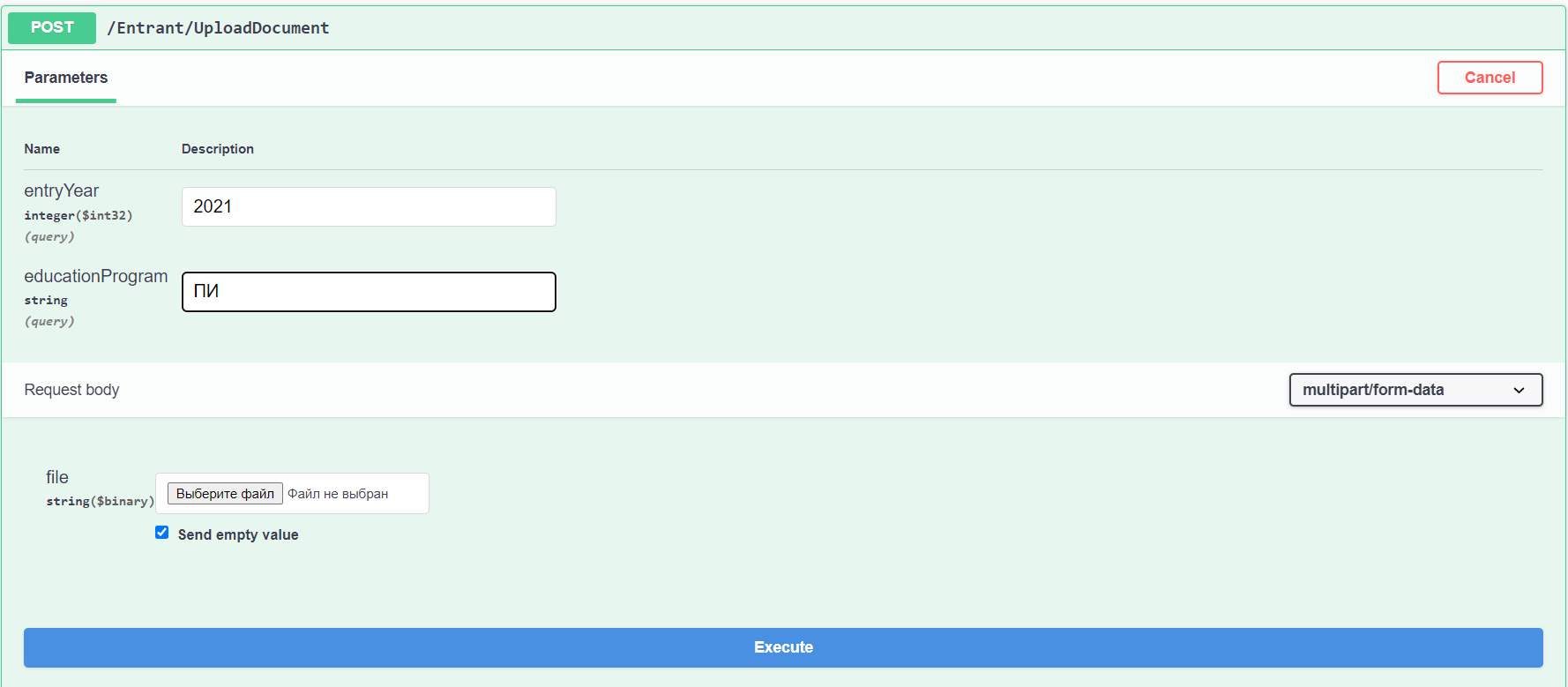


Рисунок 11.1. Запрос загрузки документа.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 11.2. Ответ на запрос загрузки документа.

**ПРИЛОЖЕНИЕ 1**

**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

1. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. ГОСТ 15150-69 Машины, приборы и другие технические изделия. Исполнения для различных климатических районов. Категории, условия эксплуатации, хранения и транспортирования в части воздействия климатических факторов внешней среды. – М.: Изд-во стандартов, 1997.
11. ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

# 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 | Аннулированх |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |