Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

КАФЕДРА ИНФОРМАТИКИ

Отчёт по первой части проекта

По теме “Техническое задание на систему.”

Выполнили:

студенты гр. 853504

Карпыза А.А., Грек Р.Д., Шевченя И.В.

Проверил:

Чащин С.В.

Оглавление

[Введение](#_heading=h.6g72wrcc0plx) **3**

[Основания для разработки](#_heading=h.qxrbi9wcebtx) **4**

[Назначение разработки](#_heading=h.7cra8ts3pjsl) **5**

[Функциональные требования](#_heading=h.ibkomyr0rq4o) **7**

[ER-диаграмма](#_heading=h.poicbehh92m3) **13**

# Введение

В настоящее время сервисы автоматизации управления составом сотрудников компании набирают всё большее и большее распространение среди компаний самых разных сфер деятельности. Это позволяет улучшить коммуникацию между сотрудниками, и повысить продуктивность их совместной работы. Примерами сервисов, использующих подобный функционал, можно считать такие сервисы, как GitHub, облачные сервисы наподобие AWS, GCP и Azure, а также приватные сервисы различных крупных компаний, доступ к которым разрешён исключительно их сотрудникам.

Базовым элементом подобных сервисов является функционал распределения привилегий и функциональных возможностей управления среди пользователей сервиса. Это своеобразный «корень», без которого сервис не может существовать.

Исходя из популярности, функциональности и высокой возможности масштабирования в самые разные среды, было принято решение разработать проект, реализующий функционал управления пользователями, группами, а также управления их привилегиями.

# Основания для разработки

Основанием для разработки является получение задания по написанию проекта по предмету МДиСУБД.

# Назначение разработки

Назначением разработки системы является реализация функционала управления пользователями, группами, а также их привилегиями с высокой степенью дальнейшей применимости в различного рода проектах, в большинстве своём – системах управления и автоматизации работы сотрудников компании, и их масштабируемости.

В приложении будет реализована возможность создания пользователей различных типов, а также управления их привилегиями.

Привилегии будут распределяться при помощи добавления пользователям соответствующих ролей, где роль – это сущность в базе данных, являющаяся агрегатором привилегий различного типа, также содержащимся в базе данных, что позволяет в дальнейшем расширять функционал приложения в различных направлениях, не меняя уже существующую структуру базы данных, а просто добавляя новые типы привилегий пользователей в соответствующую таблицу. Таким образом, роли являются полностью конфигурируемыми объектами, что позволяет администратору предоставлять пользователям только необходимые им привилегии.

Для экономии занимаемого пространства таблицей «роль» предполагается создать таблицу «группа», к которой может принадлежать сразу несколько пользователей, между которыми впоследствии будут распределены привилегии, предоставляемые связанной с группой ролью. В ином случае пришлось бы для каждого отдельного пользователя создавать собственную роль, что неизбежно увеличивало бы число занимаемого пространства в базе данных.

Для разделения пользователей по признаку их принадлежности к какой-либо отдельной структурной единице, в проект будут добавлены такие таблицы, как «департамент» и «юнит», логически связанный с департаментом.

Для определения позиции пользователя в структурной иерархии пользователей будет добавлена таблица «квалификация», каждой из которых будет соответствовать определенный уровень приоритета, представляемый беззнаковым целым числом. Аккаунт корневого пользователя с максимальными привилегиями будет принадлежать к должности с максимальным уровнем приоритета.

Дополнительной функциональностью сущности «Квалификация» является то, что пользователи с привилегиями администратора не смогут воздействовать на пользователей с Квалификацией, обладающей приоритетом, выше либо равной приоритету должности самого пользователя.

**Архитектурная составляющая проекта**

Проект будет представлять собой API - интерфейс, реализуемый при помощи следующих программных средств и технологий:

1. Язык программирования, используемый для разработки - Python версии 3.8;
2. База данных - реляционная база данных PostgreSQL;
3. Операционная система - Linux;
4. Flask - технология языка Python, позволяющая организовать передачу данных посредством HTTP/HTTPS

# Функциональные требования

**Цикл обработки информации в системе. Порядок интегрирования системы в сторонние проекты:**

При выполнении какого-либо действия пользователем будет происходить запрос его привилегий из базы данных при помощи специальной функции, в результате чего выполняемое действие либо будет выполнено, либо отброшено с оповещением о недостаточном наборе привилегий для выполнения данного действия.

Функционал, связанный с пользователями, группами, а также связанными с ними ролями будет реализован в проекте для того, чтобы предоставить необходимую базу для дальнейшего расширения проекта при взятии разрабатываемой системы в качестве основы для какого-либо другого проекта. В ситуациях, не связанных с менеджментом пользователей, групп и ролей, то есть в ситуациях, не покрытых стандартным функционалом системы, при осуществлении запроса к базе данных будет возвращаться список назначенных заданному пользователю привилегий, который должен быть обработан непосредственно программным кодом разработчика продукта.

Расширение базы данных исходной системы в соответствии с функциональными требованиями конкретного проекта, разрабатываемого на основе предоставляемой системы, будет осуществляться путём добавления новых таблиц в исходную базу данных, а также добавлением соответствующих типов привилегий в таблицу “Policy”. В данном случае интерфейс работы с таблицей “Policy” возлагается на плечи разработчика, при этом система осуществляет непосредственное управление назначением этих привилегий посредством ролей пользователям и группам с возможностью выдачи их для дальнейшей обработки.  
 Стоит также отметить, что хранение всех данных в базе данных системы не является обязательным, так как, опять же, система выдаёт лишь список привилегий, которые не привязаны к конкретной таблице, или базе данных. Это позволяет стороннему разработчику использовать для хранения данных приложения и системы разные базы данных, при этом база данных приложения может быть любого типа на усмотрение разработчика.

**Перечень выполняемых функций:**

1. Создание/изменение/удаление пользователя;
2. Авторизация пользователя;
3. Создание/изменение/удаление группы;
4. Добавление пользователя в группу;
5. Добавление/изменение/удаление должности;
6. Добавление должности пользователю;
7. Создание/изменение/удаление департамента;
8. Добавление пользователя в департамент;
9. Создание/изменение/удаление юнита;
10. Добавление пользователя в юнит;
11. Создание/изменение/удаление роли;
12. Назначение/удаление роли пользователю;
13. Назначение/удаление роли группе;
14. Журналирование изменения данных в таблицах;

**Информационные сущности:**

1. Пользователь (User);
2. Группа (Group);
3. Квалификация (Position);
4. Департамент (Department);
5. Юнит (Unit);
6. Роль (Role);
7. Привилегия (Policy)

**Связи между сущностями:**

1. Пользователь и Группа связаны отношением Many-To-Many, так как пользователь может состоять одновременно в нескольких группах привилегий (например, группы для управления разными департаментами), и в одной группе может состоять сразу несколько пользователей.
2. Пользователь и Роль связаны связаны отношением Many-To-Many, так как пользователь может иметь несколько ролей, и одна роль может быть назначена нескольким пользователям.
3. Пользователь и Квалификация связаны отношением One-To-Many, так как пользователь может иметь только одну квалификацию, и одна квалификация может быть у нескольких пользователей.
4. Пользователь и Департамент связаны отношением One-To-Many, так как один пользователь может состоять одновременно в одном департаменте, в то же время в одном департаменте могут состоять сразу несколько пользователей.
5. Пользователь и Юнит связаны отношением One-To-Many, так как один пользователь может состоять одновременно в одном юните, в то же время в одном юните могут состоять сразу несколько пользователей.
6. Группа и Роль связаны отношением отношением Many-To-Many, так как к группе может быть присоединено несколько ролей, и одна роль может быть назначена нескольким группам.
7. Департамент и Юнит связаны отношением One-To-Many, так как к в одном департаменте могут содержаться сразу несколько юнитов, в то же время один юнит может быть связан только с одним Департаментом. Данная связь является идентифицирующей.
8. Роль и Привилегия связаны отношением Many-To-Many, так как к роли может быть привязано несколько привилегий, и одна и та же привилегия может быть привязана к нескольким ролям.
9. Пользователь и его непосредственный начальник - рекурсивная связь, т.е. родитель и потомок являются одной и той же сущностью.

**Обработка исключений:**

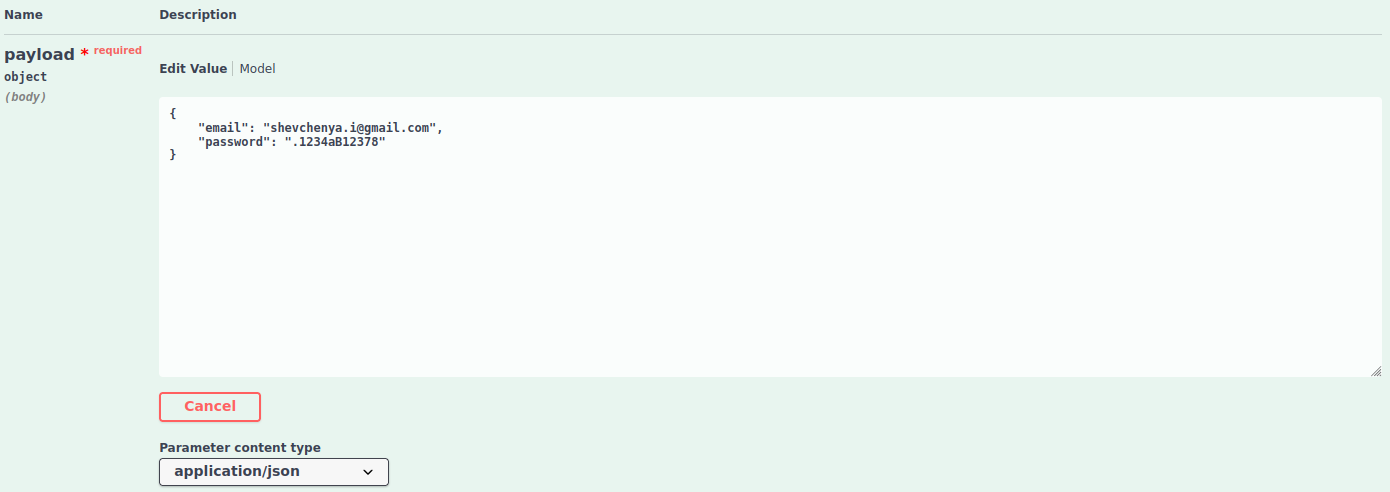
1. Обработка попытки использования пользователем не разрешенного ему функционала;
2. Обработка попытки добавления пользователю функционала администратора. Данная возможность должна быть добавлена только корневому пользователю (сервисному аккаунту администратора), при необходимости повторно ввести пароль для подтверждения действия;
3. Обработка попытки воздействия пользователя на другого пользователя, обладающего квалификацию с более высоким, либо равным уровнем приоритета.
4. Обработка попытки пользователя с привилегиями изменения должности пользователей, изменения должности на квалификация своего уровня привилегий и выше.
5. Обработка попытки авторизации пользователя с использованием неверного сочетания электронной почты и пароля.
6. Обработка попытки пользователя изменения собственных привилегий.

**Тестирование на соответствие функциональным требованиям**

1. *Авторизация пользователей*

Шаги воспроизведения

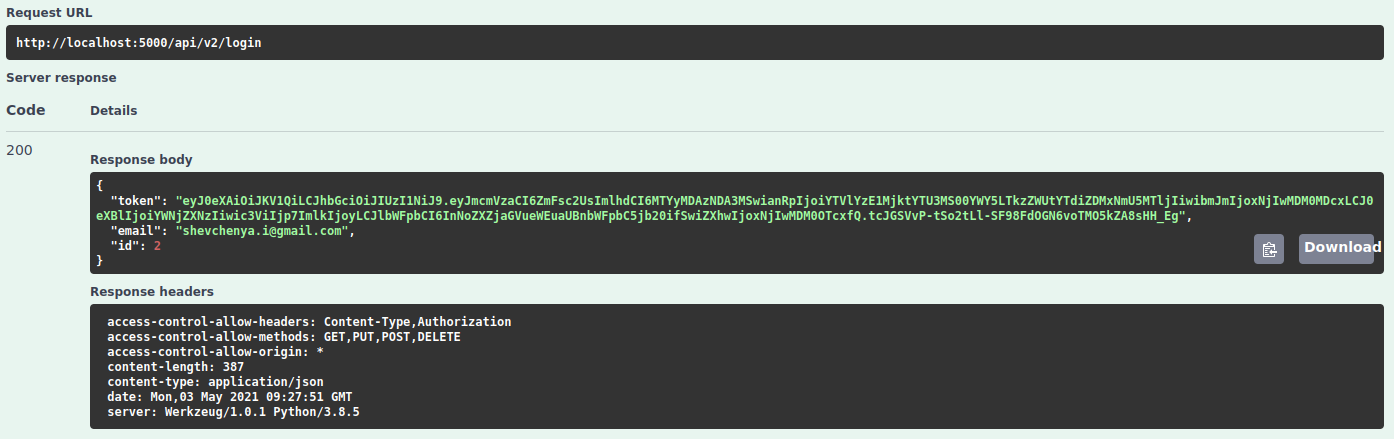
1. Осуществить запрос к API на авторизацию в аккаунт существующего в системе пользователя с соответствующими параметрами запроса.



Ожидаемый результат

Данные в формате JSON, содержащие JWT валидационный токен, идентификатор пользователя и продублированный email.

Фактический результат



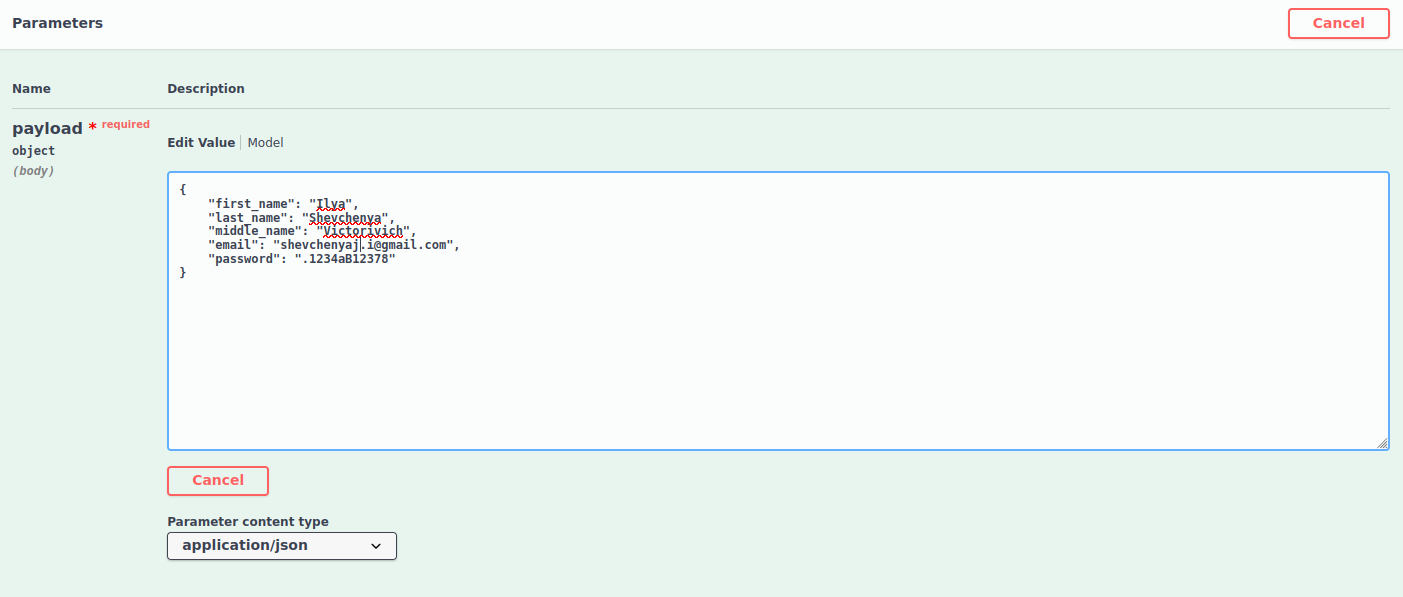
Acceptance status

***Passed***

1. *Создание пользователя*

Шаги воспроизведения

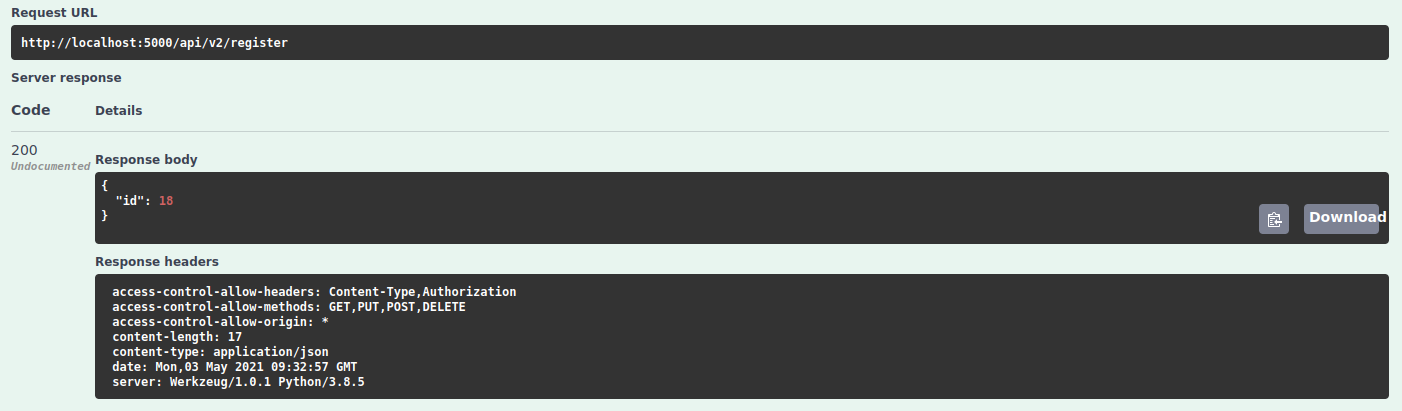
1. Авторизоваться в аккаунт с привилегиями на создание пользователей;
2. Осуществить запрос к API на создание нового пользователя с соответствующими параметрами запроса.

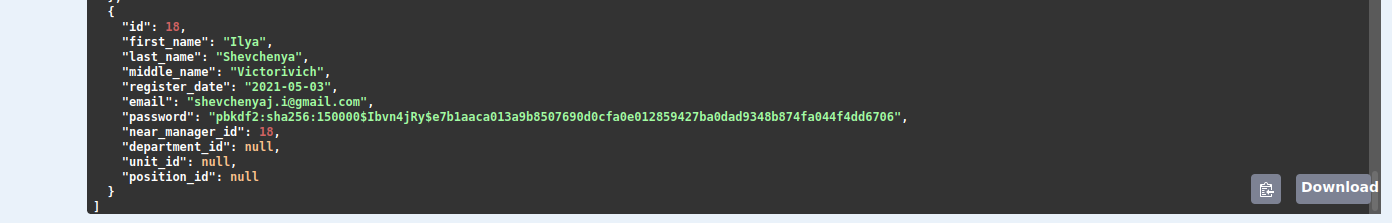


Ожидаемый результат

В системе создан новый пользователь, если удовлетворены все условия создания пользователей. В качестве тела ответа приходит JSON с информацией о пользователе.

Фактический результат





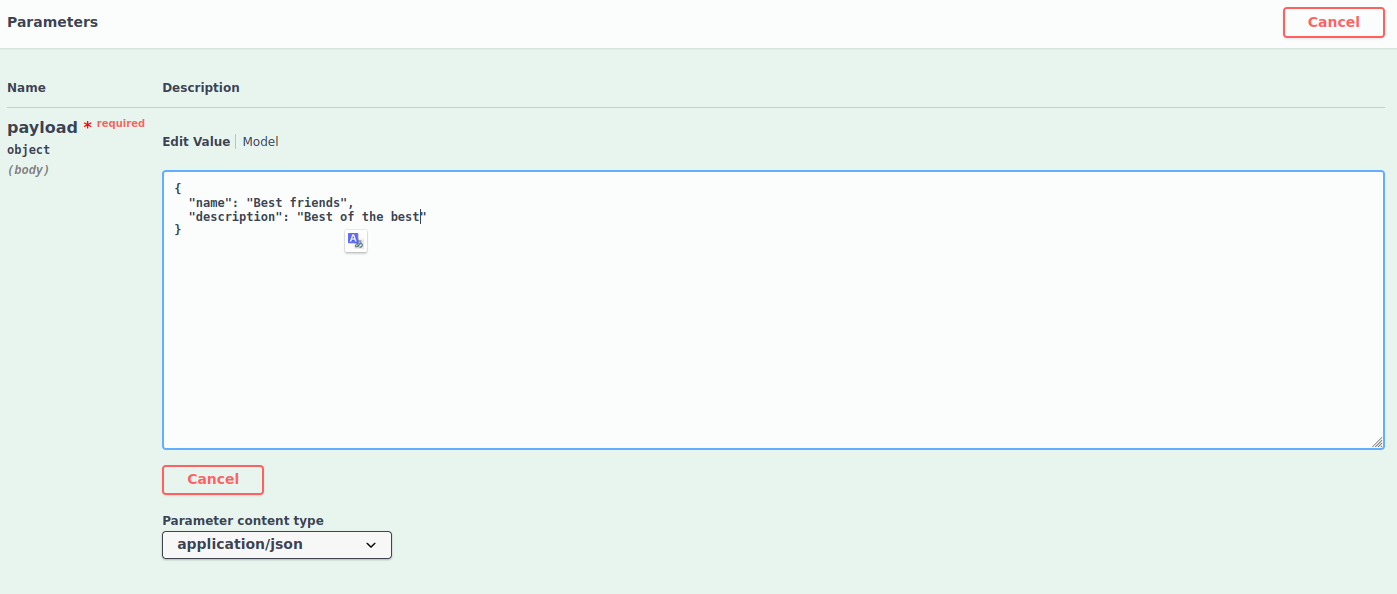
Acceptance status

***Passed***

1. *Создание группы*

Шаги воспроизведения

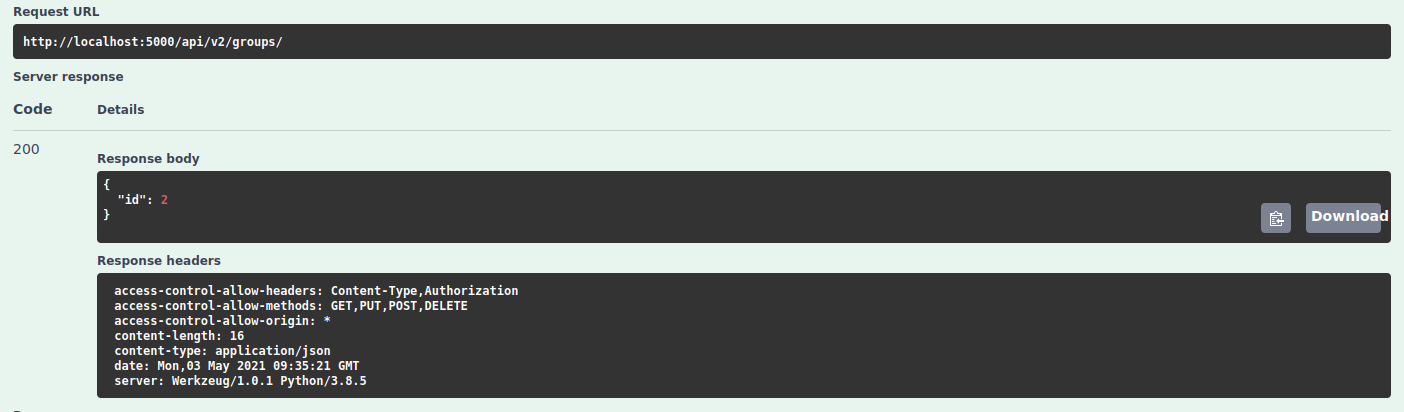
1. Авторизоваться в аккаунт с привилегиями на создание групп;
2. Осуществить запрос к API на создание новой группы с соответствующими параметрами запроса.



Ожидаемый результат

Фактическое создание новой группы, а в качестве тела отмета - JSON с идентификатором групп.

Фактический результат





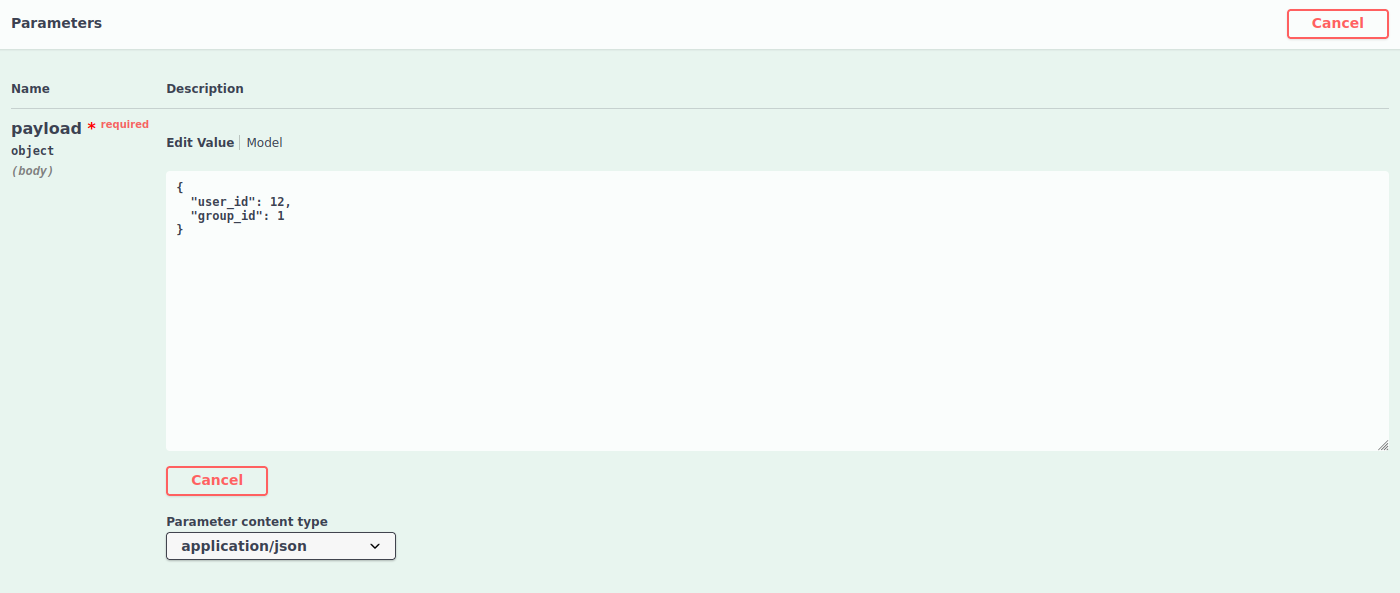
Acceptance status

***Passed***

1. *Добавление пользователя в группу*

Шаги воспроизведения

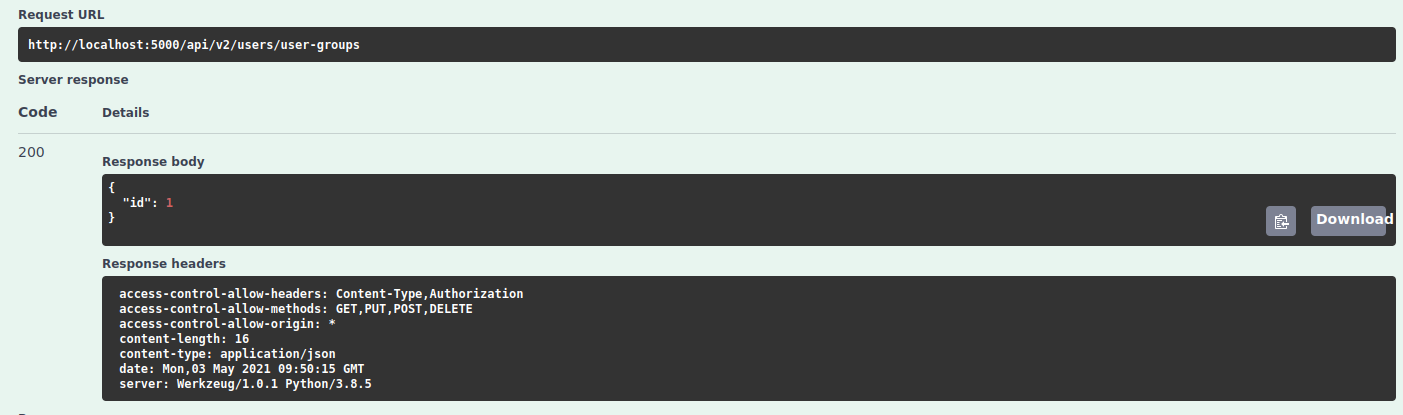
1. Авторизоваться в аккаунт с привилегиями на добавление пользователя в группу;
2. Осуществить запрос к API на добавление пользователя в группу с соответствующими параметрами запроса.



Ожидаемый результат

Ожидается получение ответа с соответствующим статус кодом и телом с идентификатором записи, что свидетельствует о создании записи в таблице базы данных, иначе выдается сообщение о возникшей ошибке.

Фактический результат



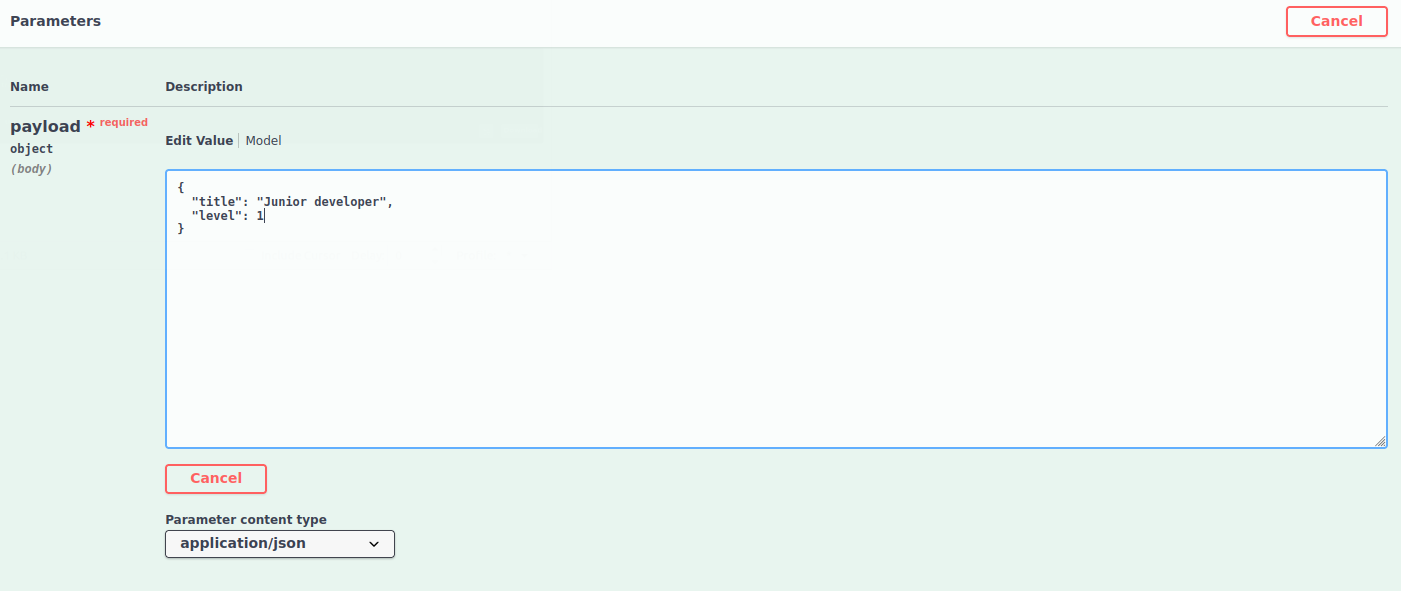
Acceptance status

***Passed***

1. *Создание должности*

Шаги воспроизведения

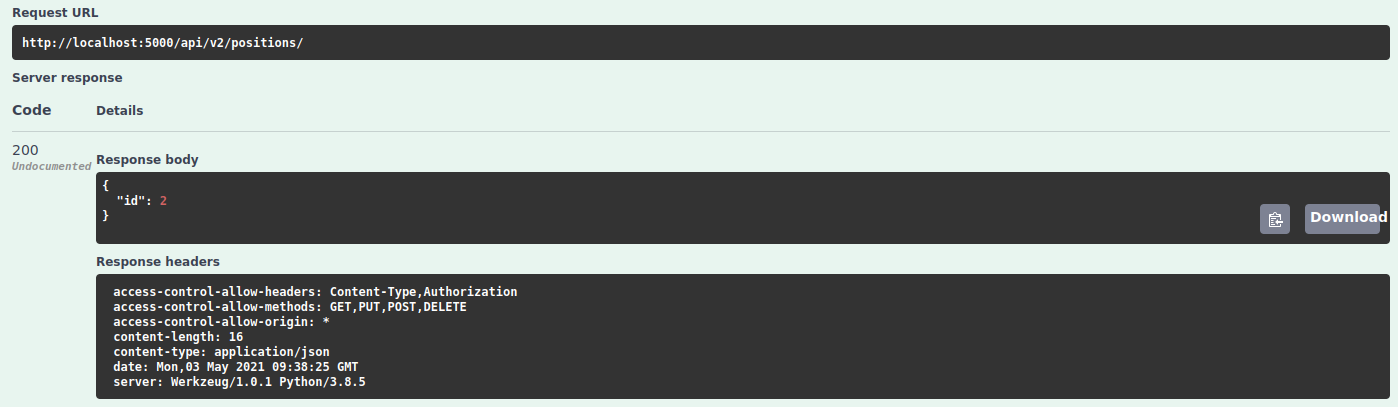
1. Авторизоваться в аккаунт с привилегиями на создание должностей;
2. Осуществить запрос к API на создание новой должности с соответствующими параметрами запроса.

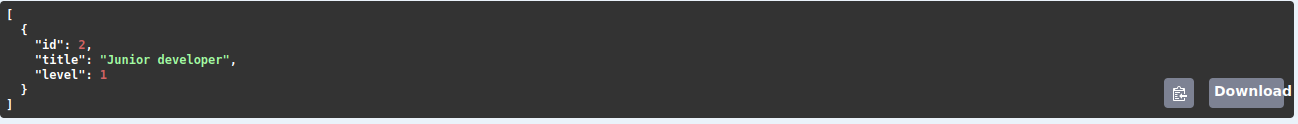


Ожидаемый результат

Фактическое создание новой должности, а в качестве тела ответа - JSON с идентификатором должности и статус кодом ответа.

Фактический результат





Acceptance status

***Passed***

1. *Добавление должности пользователю*

Шаги воспроизведения

1. Авторизоваться в аккаунт с привилегиями на назначение должности пользователю;
2. Осуществить запрос к API на назначение должности пользователю с соответствующими параметрами запроса.

Ожидаемый результат

Ожидается получение ответа с соответствующим статус кодом и телом в виде JSON с идентификатором записи, что свидетельствует о создании записи в таблице базы данных, иначе выдается сообщение о возникшей ошибке.

Фактический результат

Удачное создание записи в таблице базы данных, тем самым добавление пользователю выбранной должности

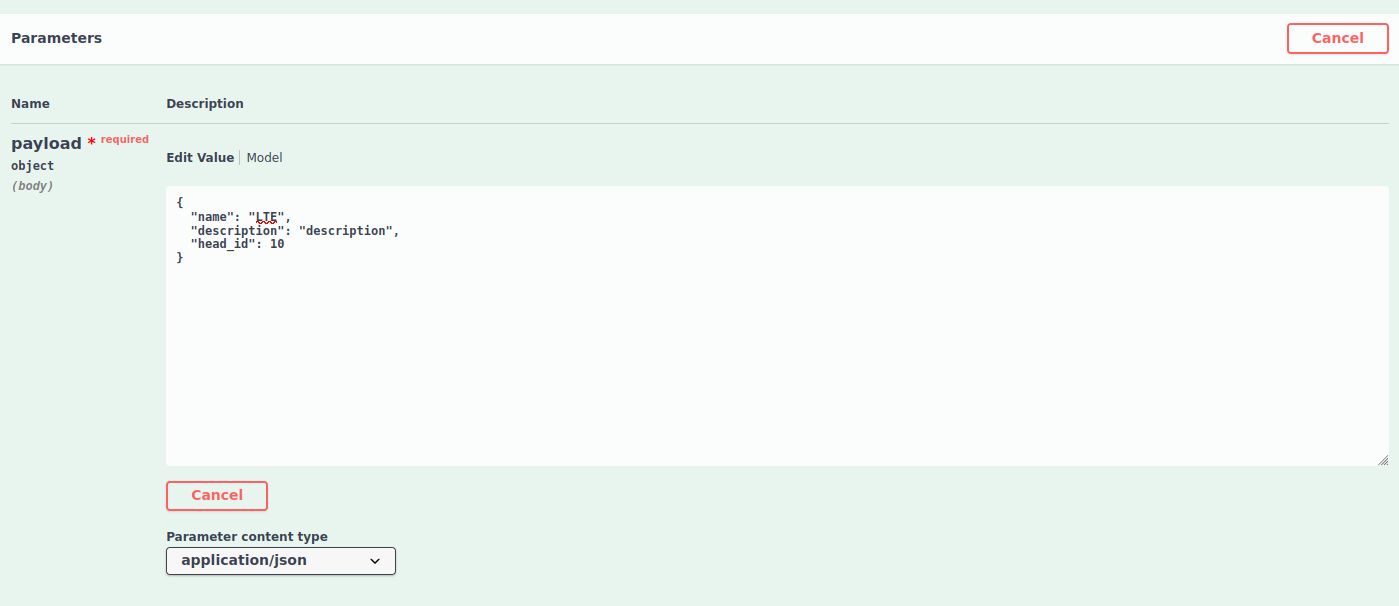
Acceptance status

***Passed***

1. *Создание департамента*

Шаги воспроизведения

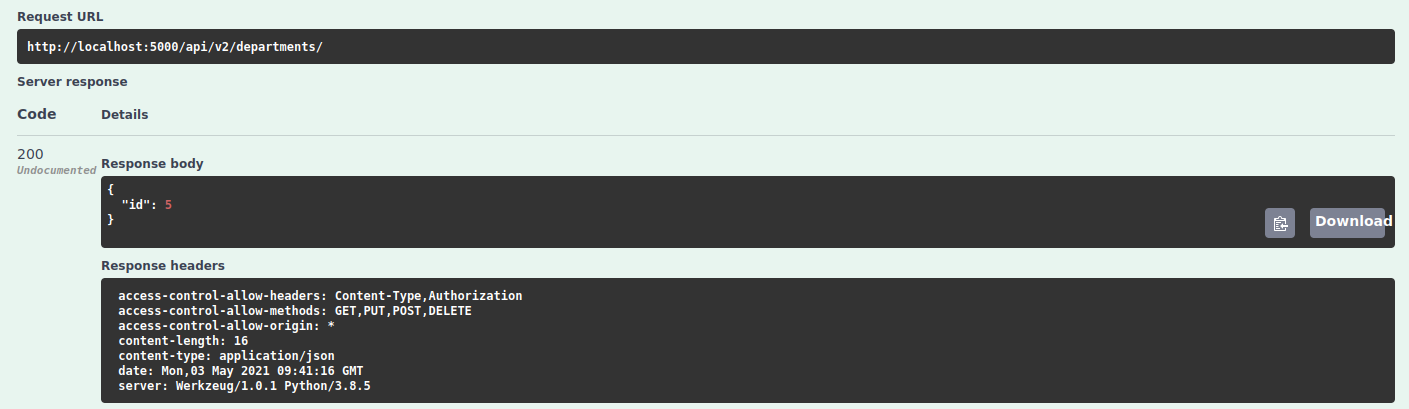
1. Авторизоваться в аккаунт с привилегиями на создание департамента;
2. Осуществить запрос к API на создание департамента с соответствующими параметрами запроса.



Ожидаемый результат

Фактическое создание нового департамента, а в качестве тела ответа - JSON с идентификатором департаментом и статус кодом ответа.

Фактический результат





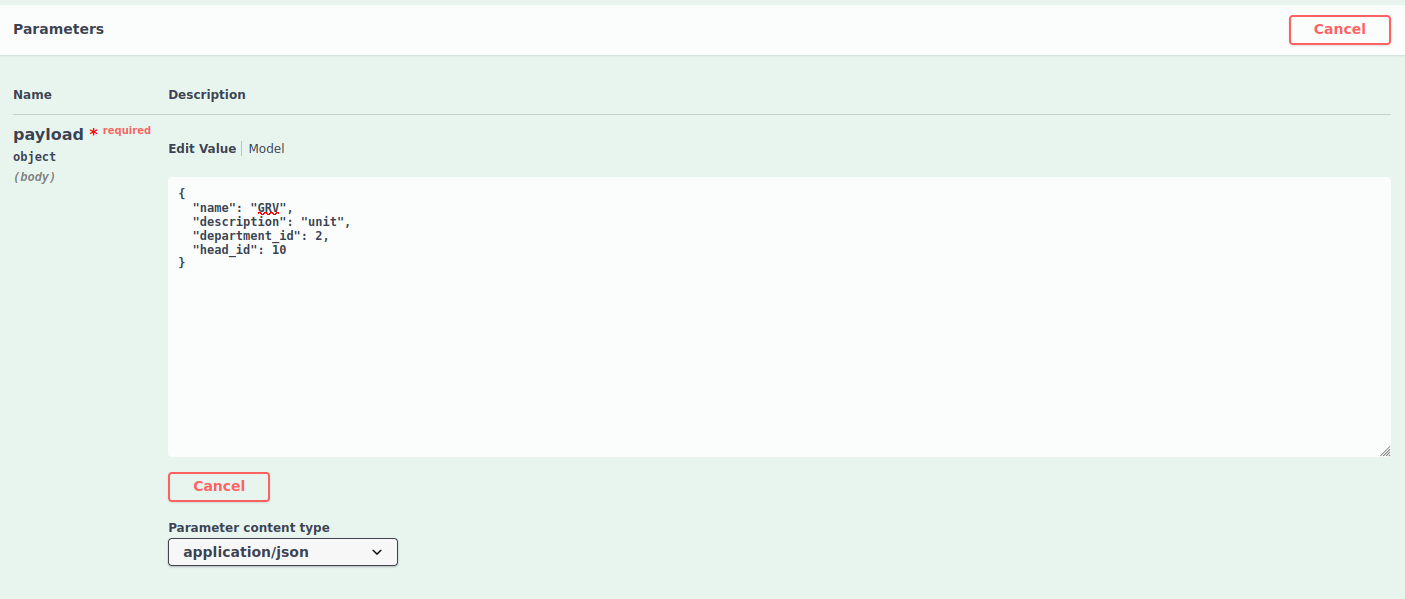
Acceptance status

***Passed***

1. *Создание юнита*

Шаги воспроизведения

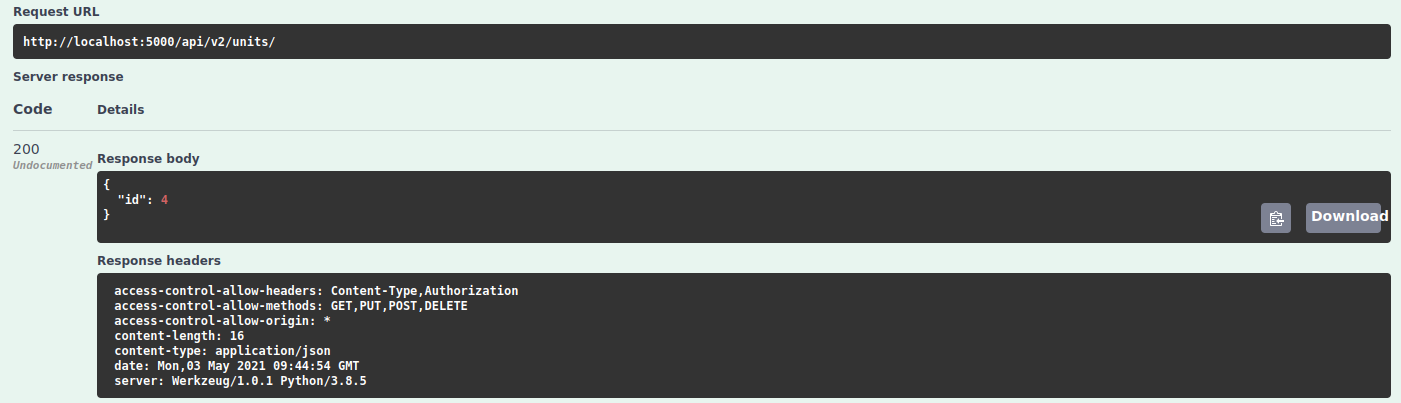
1. Авторизоваться в аккаунт с привилегиями на создание юнита;
2. Осуществить запрос к API на создание юнита с соответствующими параметрами запроса.



Ожидаемый результат

Фактическое создание нового юнита в департаменте, а в качестве тела ответа - JSON с идентификатором юнита и статус кодом ответа.

Фактический результат





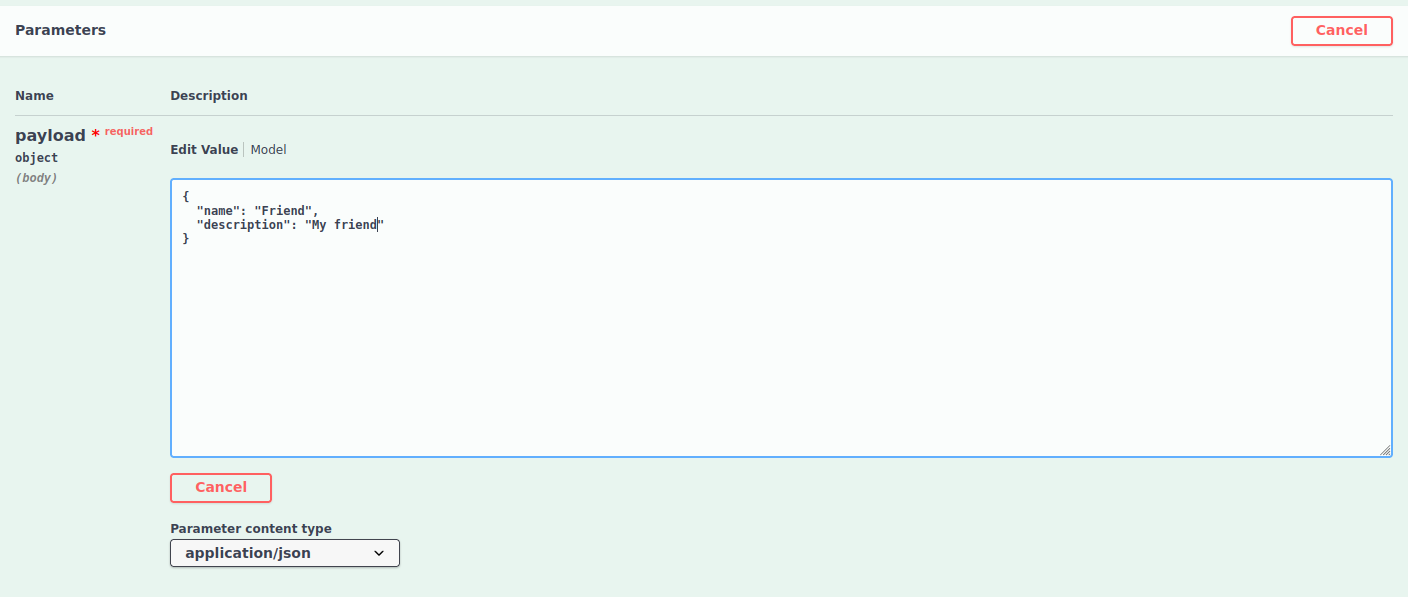
Acceptance status

***Passed***

1. *Создание роли*

Шаги воспроизведения

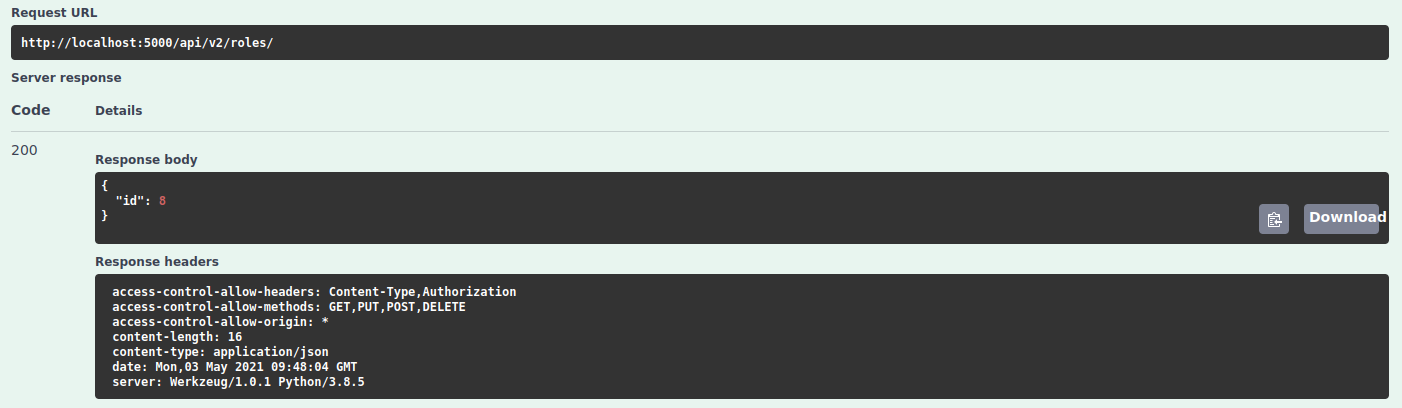
1. Авторизоваться в аккаунт с привилегиями на создание роли;
2. Осуществить запрос к API на создание роли с соответствующими параметрами запроса.



Ожидаемый результат

Фактическое создание новой роли, а в качестве тела ответа - JSON с идентификатором роли и статус кодом ответа.

Фактический результат





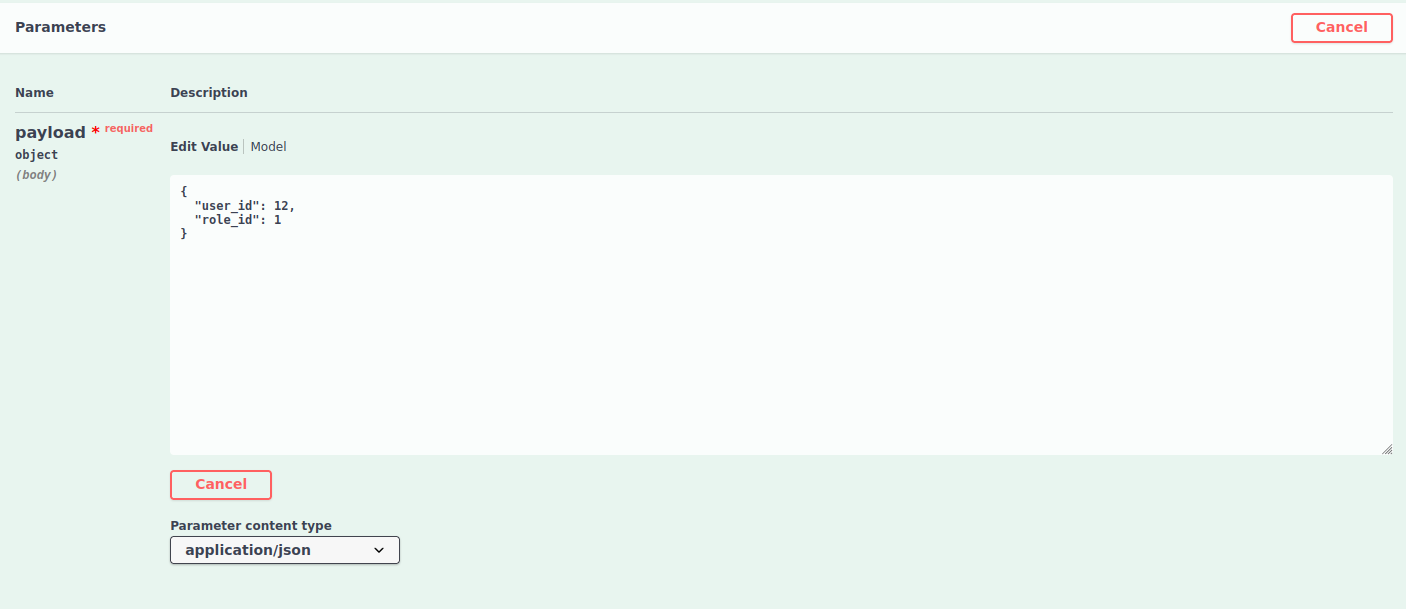
Acceptance status

***Passed***

1. *Назначение роли пользователю*

Шаги воспроизведения

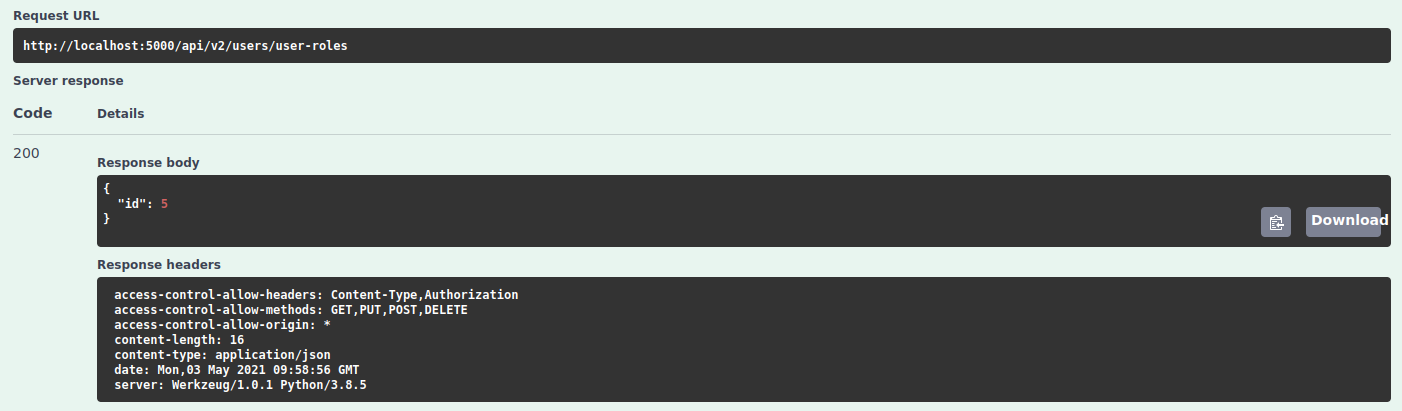
1. Авторизоваться в аккаунт с привилегиями на назначение роли пользователю;
2. Осуществить запрос к API на назначение роли пользователю с соответствующими параметрами запроса.



Ожидаемый результат

Ожидается получение ответа с соответствующим статус кодом и телом в виде JSON с идентификатором записи, что свидетельствует о создании записи в таблице базы данных, иначе выдается сообщение о возникшей ошибке.

Фактический результат



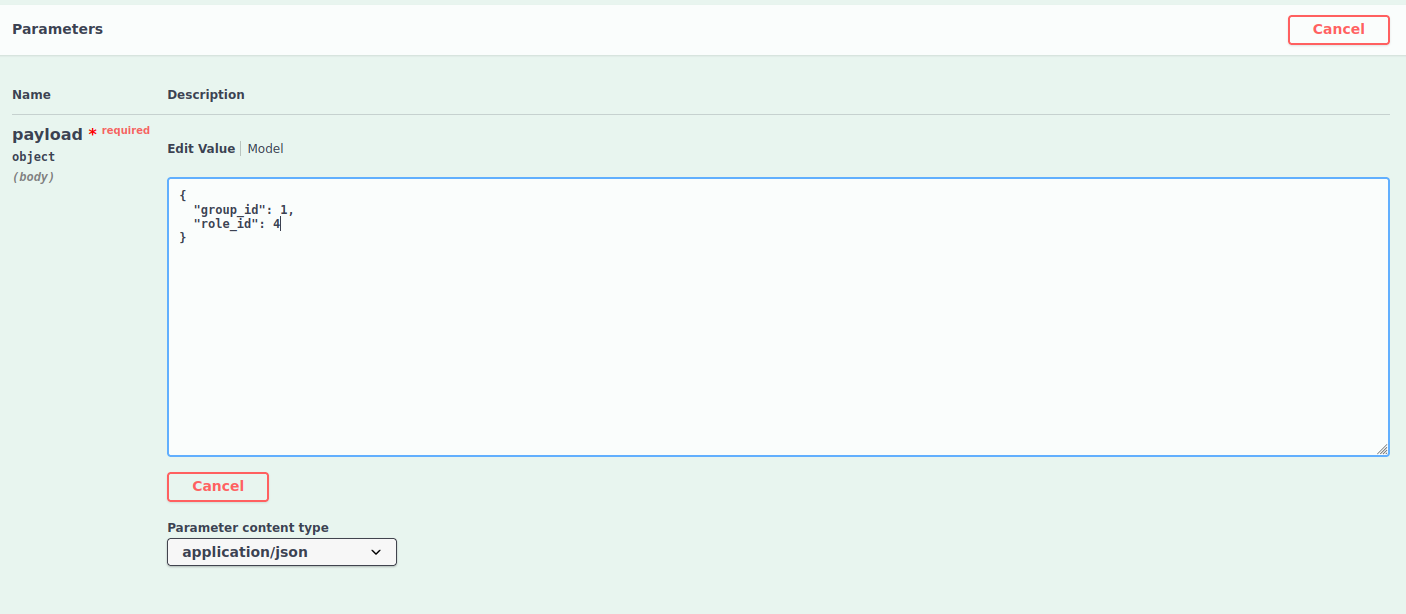
Acceptance status

***Passed***

1. *Назначение роли группе*

Шаги воспроизведения

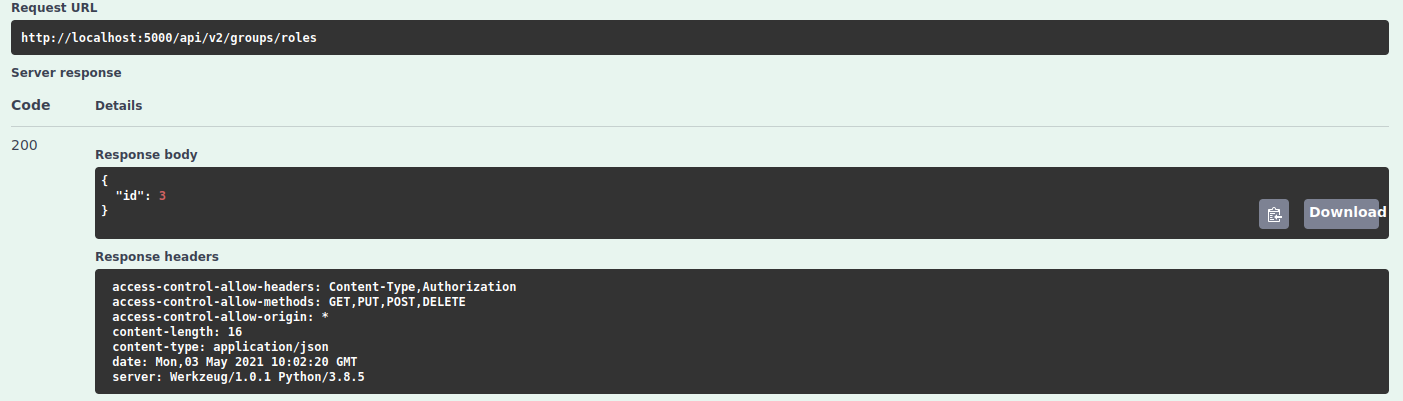
1. Авторизоваться в аккаунт с привилегиями на назначение роли группе;
2. Осуществить запрос к API на назначение роли группе с соответствующими параметрами запроса.



Ожидаемый результат

Ожидается получение ответа с соответствующим статус кодом и телом в виде JSON с идентификатором записи, что свидетельствует о создании записи в таблице базы данных, иначе выдается сообщение о возникшей ошибке.

Фактический результат



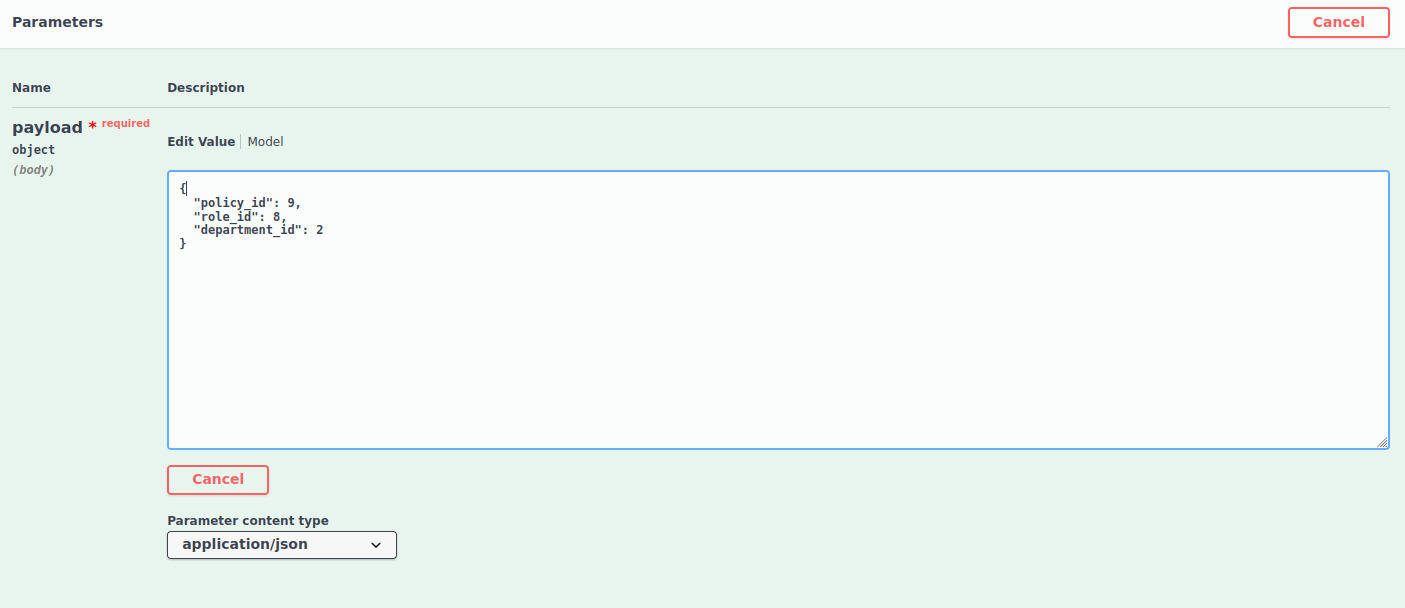
Acceptance status

***Passed***

1. *Добавление привилегии к роли*

Шаги воспроизведения

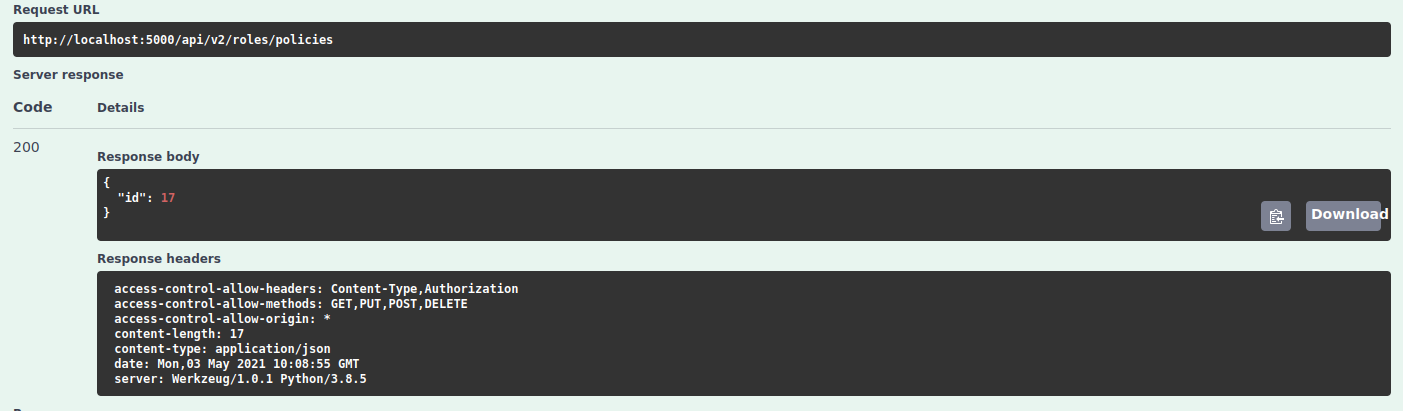
1. Авторизоваться в аккаунт с привилегиями на добавление привилегии роли;
2. Осуществить запрос к API на добавление привилегии роли с соответствующими параметрами запроса.



Ожидаемый результат

Ожидается получение ответа с соответствующим статус кодом и телом в виде JSON с идентификатором записи, что свидетельствует о создании записи в таблице базы данных, иначе выдается сообщение о возникшей ошибке.

Фактический результат



Acceptance status

***Passed***

# ER-диаграмма

