Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

КАФЕДРА ИНФОРМАТИКИ

Отчёт по предмету МДиСУБД

По теме “Руководство пользователя к системе”

Выполнили:

студенты гр. 853504

Карпыза А.А., Грек Р.Д., Шевченя И.В.

Проверил:

Чащин С.В.

Оглавление

[**1. Введение**](#_k8kvjul8cbn4) **2**

[1.1. Область применения](#_8yd44ynf4ki9) 2

[1.2. Краткое описание возможностей](#_zi7s6er5otys) 2

[1.3. Уровень подготовки пользователя](#_o1m2vdl3ha54) 2

[**2. Назначение и условия применения**](#_v7t2jir8zxol) **3**

[**3. Подготовка к работе**](#_89gjt0ye6k4r) **3**

[3.1. Состав и содержание дистрибутивного носителя данных](#_9dzo63215d0o) 3

[3.2. Порядок загрузки данных и программ](#_53zlqagdgt9l) 3

[3.3. Порядок проверки работоспособности](#_cbc2523mg0hk) 4

[**4. Описание операций**](#_3rhg6a80ftdz) **4**

[**5. Аварийные ситуации**](#_4c0n4nwthe04) **6**

# 

# 1. Введение

## 1.1. Область применения

Требования настоящего документа применяются при:

* Разработке программного продукта, в котором существует необходимость разделения привилегий доступа между пользователями и группами пользователей;
* Внедрении готового программного продукта в эксплуатацию.

## 1.2. Краткое описание возможностей

Данный программный продукт при своём подключении к разрабатываемому стороннему проекту предоставляет разработчикам API, главной функцией которого является создание пользователей, и предоставление разработчику возможностей гибкого управления их привилегиями.

Система предоставляет разработчикам необходимый минимум для работы с привилегиями пользователей и, благодаря своей высокой гибкости, позволяет пользователю (разработчику) внедрять её в сторонние проекты различной направленности.

Система способна выполнять следующий перечень 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;

## 1.3. Уровень подготовки пользователя

Пользователь (сторонний разработчик) должен иметь опыт работы с API, а также практические навыки во внедрении его в проекты. Вместе с тем, необходимы знания формата данных JSON для беспрепятственного взаимодействия с предоставляемым сервисом. Так как система представляет собой встраиваемый шаблон для управления пользователями написанный на языке python с использованием фреймворка flask, то для расширения функциональности уже имеющегося API могут понадобиться навыки владения указанными технологиями.

# 2. Назначение и условия применения

Данный программный продукт предназначен для внедрения в сторонние системы на ранних этапах их разработки, предоставляя необходимый базовый функционал управления пользователями, их должностями, структурными подразделениями компании, а также привилегиями.

Доступ к упомянутым функциям системы предоставляется посредством осуществления запросов к API-серверу.

Функционал разделения привилегий является основополагающим в данной системе, так, внедряя данную систему в сторонние проекты, разработчик получает возможность управлять привилегиями пользователей, что позволяет гибко ограничивать их доступ к ресурсам.

Программный продукт применим при существовании возможности развертывания данного сервиса и предоставления доступа к нему из разрабатываемого приложения.

Наибольшую эффективность программный продукт показывает при его внедрении в сторонний проект на самых ранних этапах его разработки, так как именно в этом случае разработчик имеет возможность тщательно продумать структуру своего проекта во взаимодействии с внедряемой системой.

# 3. Подготовка к работе

## 3.1. Состав и содержание дистрибутивного носителя данных

Для работы с системой необходим любой дистрибутив операционной системы Linux со следующим программным обеспечением:

1. Язык программирования Python 3.8;
2. База данных PostgreSQL;
3. Пакетный менеджер

Система может быть развёрнута в docker вместе со всеми зависимостями.

## 3.2. Порядок загрузки данных и программ

Перед началом работы с системой на сервере, где она будет установлена, необходимо произвести следующие действия:

1. Произвести загрузку исходного кода проекта;
2. Установить на машину сервис PostgreSQL (прим. - для Debian-based архитектуры - команда “sudo apt install postgresql postgresql-contrlib”);
3. Запустить от имени суперпользователя скрипт config\_postgresql - это произведёт установку и настройку базы данных, в том числе создание всей схемы базы данных вместе с функциями и триггерами;
4. Перейти в директорию backend и выполнить команду pipenv install для создания virtual environment и установки всех зависимостей необходимых для запуска проекта проекта.

## 3.3. Порядок проверки работоспособности

Для проверки работоспособности сервиса необходимо осуществить следующую последовательность действий:

1. Запустить backend - сервис;
2. Осуществить запрос на авторизацию в аккаунт корневого пользователя системы с максимальными привилегиями, воспользовавшись следующими авторизационными данными:

* login - root
* password - root

В результате должен быть возвращен ответ с кодом 200, что означает верную работу сервиса. В ином случае следует убедиться в выполнении всех пунктов из предыдущего подраздела.

# 4. Описание операций

Предоставляемая система выполняет функции, представленные в таблице ниже.

Пояснения к таблице:

1. Столбец “Функция” - краткое описание области задач, выполняемой системой;
2. Столбец “Задача” - функция, выполняемая конкретным API-запросом;
3. Столбец “Запрос, привилегия” - текст запроса и привилегия, необходимая для его осуществления;
4. Столбец “Параметры запроса” - параметры, передаваемые в теле запроса, и их предназначение.

|  |  |  |  |
| --- | --- | --- | --- |
| Функция | Задача | Запрос, привилегия | Параметры запроса |
| Обеспечивает возможность управления пользователями в системе | Создание пользователей | “Create users” | first\_name, last\_name, middle\_name,  email,  password |
| Редактирование пользователей | “Manage users” | first\_name, last\_name, middle\_name,  email,  password,  department\_id,  unit\_id,  position\_id  Для обновления не обязательно указывать все поля в теле запроса |
| Удаление пользователей | “Manager user” | email, в качестве параметра url |
| Обеспечивает возможность управления организационными единицами (Department, Unit) и должностями пользователей (Position) | Создание департамента | “Manage departments” | name, description, head\_id |
| Редактирование департамента | “Manage departments” | department\_id, в качестве параметра url, а тело запроса может содержать все те же поля, что и в запросе на создание |
| Удаление департамента | “Manage departments” | department\_id, в качестве параметра url |
| Создание юнита | “Manage units” | name, description, head\_id, department\_id |
| Редактирование юнита | “Manage units” | unit\_id, в качестве параметра url, а тело запроса может содержать все те же поля, что и в запросе на создание |
| Удаление юнита | “Manage units” | unit\_id, в качестве параметра url |
| Создание должности | “Manage positions” | title, level |
| Назначение должности пользователю | “Manage users” | position\_id, user\_id |
| Редактирование должности | “Manage positions” | position\_id, в качестве параметра url, а тело запроса может содержать все те же поля, что и в запросе на создание |
| Удаление должности | “Manage positions” | position\_id, в качестве параметра url |
| Обеспечивает возможность распределения привилегий пользователей | Создание роли | “Manage roles” | name, description |
| Назначение роли привилегии | “Manage roles” | policy\_id, role\_id |
| Назначение роли пользователю | “Manage users” | role\_id, user\_id |
| Назначение роли группе | “Manage groups” | role\_id, group\_id |
| Удаление роли | “Manage roles” | role\_id, в качестве параметра url |
| Создание групп | “Manage groups” | name, description |
| Редактирование групп | “Manage groups” | group\_id, в качестве параметра url, а тело запроса может содержать все те же поля, что и в запросе на создание |
| Удаление групп | “Manage groups” | group\_id, в качестве параметра url |
| Получение списка привилегий, назначенных пользователю | “Manage users” | user\_id, в качестве параметра url |
| Проверка существования конкретной привилегии у пользователя | “Manage users” | role\_id, policy\_id, в качестве параметра url |

# 5. Аварийные ситуации

|  |  |  |  |
| --- | --- | --- | --- |
| Класс ошибки | Ошибка | Описание ошибки | Способ устранения |
| Внутренняя ошибка встраиваемого сервиса | Сбой аутентификации | Неверно введен адрес электронной почты или пароль. | Ввести верные данные и повторить попытку. |
| Ошибка привилегий | Текущий авторизованный пользователь не имеет доступа к требуемому функционалу. | Предоставить пользователю соответствующую привилегию. |
| Ошибка параметров | При запросе не был передан один или несколько обязательных параметров. | Переписать запрос, указав все необходимые параметры и осуществить его |
| Ошибка формата запроса | Запрос к API-сервису не соответствует заданному (отличен он application/json) | Переформатировать запрос к сервису так, чтобы поле content type было равно application/json и повторить его. |
| Сбой локальной сети | Нет сетевого взаимодействия между сервисом и разрабатываемым приложением | Отсутствует возможность отправки запросов к API - сервису. “Сервер не отвечает” | Перезагрузить рабочую станцию. Проверить доступность сервера по порту 80. После восстановления работы сети повторить попытку подключения |