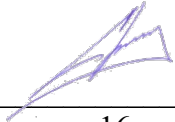


**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Образовательная программа бакалавриата «Программная инженерия»

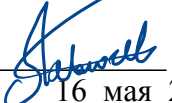
**СОГЛАСОВАНО**

Преподаватель департамента  
программной инженерии ФКН,  
кандидат компьютерных наук

  
\_\_\_\_ С.А.Виденнин  
\_\_16\_\_ мая 2025 г.

**УТВЕРЖДАЮ**

Академический руководитель  
образовательной программы  
«Программная инженерия»,  
старший преподаватель департамента  
программной инженерии

  
\_\_\_\_ Н.А. Павлов  
\_\_16\_\_ мая 2025 г.

**ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА  
УПРАВЛЕНИЯ ИНЦИДЕНТАМИ В ОБЛАСТИ ОБЕСПЕЧЕНИЯ  
БЕЗОПАСНОСТИ (СЕРВЕРНАЯ ЧАСТЬ: УПРАВЛЕНИЕ  
УЧАСТНИКАМИ СИСТЕМЫ, УВЕДОМЛЕНИЯМИ И  
ДЕЖУРСТВАМИ)**

Руководство оператора

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.05.05-01 34 01-1-ЛУ

Исполнитель  
студент группы БПИ214  
\_\_\_\_ / Е.К.Фортов/  
\_\_\_\_ 16 мая 2025 г.



Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

УТВЕРЖДЕН  
RU.17701729.05.05-01 34 01-1-ЛУ

**ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА  
УПРАВЛЕНИЯ ИНЦИДЕНТАМИ В ОБЛАСТИ ОБЕСПЕЧЕНИЯ  
БЕЗОПАСНОСТИ (СЕРВЕРНАЯ ЧАСТЬ: УПРАВЛЕНИЕ УЧАСТНИКАМИ  
СИСТЕМЫ, УВЕДОМЛЕНИЯМИ И ДЕЖУРСТВАМИ)**

**Руководство оператора**

**RU.17701729.05.05-01 34 01-1-ЛУ**

**Листов 20**

Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## Оглавление

1. НАЗНАЧЕНИЕ.....	3
1.1. Функциональное назначение.....	3
1.2. Эксплуатационное назначение.....	3
2. УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ.....	4
2.1. Минимальный состав аппаратных средств.....	4
2.2. Минимальный состав программных средств.....	4
2.3. Требования к персоналу (пользователю).....	4
3. ВЫПОЛНЕНИЕ ПРОГРАММЫ.....	5
3.1. Установка и запуск приложения.....	5
3.2. Работа с API.....	5
3.3. Мониторинг.....	17
4. СООБЩЕНИЯ ОПЕРАТОРУ.....	18
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ.....	19
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ.....	20

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 34				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## 1. НАЗНАЧЕНИЕ

### 1.1. Функциональное назначение

Функциональным назначением программы является автоматизация обработки тревожных событий, управлении бизнес-процессами, связанных с инцидентами, и координации действий всех участников процесса реагирования. Система сочетает элементы трекера задач (по аналогии с Jira) и платформы для интеграции данных от датчиков, обеспечивая реализацию заранее настроенных протоколов реагирования без задержек.

Серверная часть системы предназначена для реализации бизнес-логики системы. Рассматриваемая часть серверной части включает в себя сервисы по хранению и управлению участниками системы, дежурствами и уведомлениями, включая отправку последних. Эти сервисы предоставляют свое API для взаимодействия с ними.

### 1.2. Эксплуатационное назначение

Эксплуатационное назначение системы заключается в обеспечении работы комплекса программных и аппаратных средств для управления и реагирования на события, связанные с безопасностью на объекте. Как система безопасности, ее основная функция — получение сигналов от различных датчиков, управление событиями и инцидентами, координация участников инцидентных процессов, а также автоматизация обработки бизнес-процессов, связанных с реагированием на события.

Серверная часть системы предоставляет API, который будет использоваться клиентской частью веб-приложения. Пользователи будут получать доступ к необходимым им данным из базы данных посредством вызова конечных точек API. Разрабатываемые в рамках данного технического задания микросервисы, а именно сервис профилей, сервис уведомлений и сервис хранения дежурств активных пользователей, будут использоваться другими микросервисами, а также графической частью приложения.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 34				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## 2. УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ

### 2.1. Минимальный состав аппаратных средств

Для бесперебойной работы программного продукта требуется компьютер с:

- Объемом свободной встроенной памяти не меньше 30 ГБ,
- Объёмом оперативной памяти не меньше 1 ГБ.

### 2.2. Минимальный состав программных средств

Для бесперебойной работы программного продукта требуется компьютер с:

- 1) Установленным Docker и Docker-Compose (любая стабильная версия 2024 года)
- 2) Операционной системой Linux со стабильной сборкой, выпущенной не позднее 2015 года
- 3) Системой сборки Maven 3.9.9 с Amazon Corretto JDK 21.0.4.
- 4) Docker-контейнером Postgres версии 13 или выше
- 5) Стабильным интернет-соединением с доступом к центральному Maven-репозиторию

### 2.3. Требования к персоналу (пользователю)

Необходимое количество персонала – 1 человек.

Необходимая квалификация персонала – пользователь.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

### 3. ВЫПОЛНЕНИЕ ПРОГРАММЫ

В данном разделе описан пример работы с программой.

#### 3.1. Установка и запуск приложения

Для того, чтобы локально запустить серверную часть приложения, требуется клонировать репозиторий, находящийся по ссылке <https://github.com/FortovEgor/Diploma> на свое устройство с помощью Git. Из корневой директории репозитория запустить скрипт start.sh (при необходимости предоставив скрипту права на запуск). Данный скрипт соберет сервисы из исходных файлов и запустит их в отдельных докер-контейнерах, дополнительно создав по отдельному контейнеру с базой данных для каждого сервиса. После этого сервисы будут запущены локально на портах 7070 (сервис дежурств), 8080 (сервис уведомлений) и 9090 (сервис профилей).

Для получения доступа к API, находящемуся на удаленном сервере, следует использовать взаимодействие через протокол HTTP, поэтому для получения доступа к API достаточно обращаться к следующим базовым url:

- <http://158.160.133.19:7070> (сервис дежурств)
- <http://158.160.133.19:8080> (сервис уведомлений)
- <http://158.160.133.19:9090> (сервис профилей)

#### 3.2. Работа с API

API, предоставляемое серверной частью, дает доступ к конечным точкам трех сервисов, описанных ранее. Ниже будет приведена полная спецификация RESTful API для всех сервисов с пояснениями.

Группа требований	Формулировка требования	Требования к конечной точке API
-------------------	-------------------------	---------------------------------

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 34				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Сервис управления профилями пользователей	Создать нового пользователя	<b>Относительный url:</b> /users <b>Метод запроса:</b> POST  <b>Параметры:</b>  -  <b>Тело запроса:</b> <ul style="list-style-type: none"><li>• <i>name</i> — имя пользователя</li><li>• <i>email</i> — почта</li><li>• <i>phone</i> — телефон</li><li>• <i>password</i> — пароль</li></ul> <b>Ограничения на входные данные:</b> <ul style="list-style-type: none"><li>• name:<ul style="list-style-type: none"><li>◦ не пусто</li><li>◦ длина: от 2 до 250 символов включительно</li></ul></li><li>• email:<ul style="list-style-type: none"><li>◦ присутствует</li><li>◦ соответствие формату почты</li><li>◦ длина: от 6 до 100 символов включительно</li><li>◦ уникальность</li></ul></li><li>• phone:<ul style="list-style-type: none"><li>◦ присутствует</li><li>◦ соответствует формату телефона</li><li>◦ уникальность</li></ul></li><li>• password:<ul style="list-style-type: none"><li>◦ не пусто</li><li>◦ длина: от 8 до 40 символов</li></ul></li></ul> <b>Возвращаемые данные:</b> <ul style="list-style-type: none"><li>• <b>Status code:</b> 201 при успешном выполнении запроса, 400/409 при некорректных входных данных, 500 — любая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера)</li><li>• <b>Response body:</b> информация в формате JSON:<ul style="list-style-type: none"><li>• id</li><li>• name</li><li>• email</li><li>• phone</li></ul></li></ul>		
	Редактировать	<b>Относительный url:</b> /users		

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 34				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

	пользователя	<b>Метод запроса:</b> PUT  <b>Параметры:</b>  -  <b>Тело запроса:</b> <ul style="list-style-type: none"> <li>• <i>id</i> – идентификатор пользователя</li> <li>• <i>name</i> – новое имя</li> <li>• <i>email</i> — новая почта</li> <li>• <i>phone</i> — новый телефон</li> <li>• <i>password</i> — новый пароль пользователя</li> <li>• <i>organization</i> — новая компания, где работает пользователь</li> <li>• <i>avatar</i> — url-адрес новой аватарки пользователя</li> <li>• <i>about</i> — новое описание пользователя</li> </ul> <p>При этом каждое поле, кроме <i>id</i>, может отсутствовать — в таком случае данное поле сохранит свое старое значение.</p> <b>Ограничения на входные данные:</b> <ul style="list-style-type: none"> <li>• <i>id</i>:               <ul style="list-style-type: none"> <li>◦ присутствует</li> <li>◦ <math>\geq 0</math></li> </ul> </li> <li>• <i>email</i>:               <ul style="list-style-type: none"> <li>◦ уникальность</li> </ul> </li> <li>• <i>телефон</i>:               <ul style="list-style-type: none"> <li>◦ уникальность</li> </ul> </li> </ul> <b>Возвращаемые данные:</b> <ul style="list-style-type: none"> <li>• <b>Status code:</b> 200 при успешном выполнении запроса, 404, если пользователь с текущим <i>id</i> не найден, 400/409 при некорректных входных данных, 500 — любая другая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера)</li> <li>• <b>Response body:</b> информация в формате JSON:               <ul style="list-style-type: none"> <li>• <i>id</i></li> </ul> </li> </ul>		
	Получить данные о пользователе	<b>Относительный url:</b> /users/info <b>Метод запроса:</b> GET  <b>Параметры:</b>  -		
Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 34				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата



		<p><b>Переменные пути:</b></p> <p><i>id</i> — идентификатор пользователя</p> <p><b>Тело запроса:</b></p> <p>Нет</p> <p><b>Ограничения на входные данные:</b></p> <ul style="list-style-type: none"> <li>id: <ul style="list-style-type: none"> <li>&gt; 0</li> </ul> </li> </ul> <p><b>Возвращаемые данные:</b></p> <ul style="list-style-type: none"> <li><b>Status code:</b> 200 при найденном пользователе, 400 при некорректных входных данных, 404 при отсутствии пользователя, 500 — любая другая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера)</li> <li><b>Response body:</b> информация о пользователе в формате JSON: <ul style="list-style-type: none"> <li>id</li> <li>name</li> <li>email</li> <li>phone</li> <li>organization</li> <li>avatar</li> <li>about</li> </ul> </li> </ul>
	Найти пользователей по имени	<p><b>Относительный url:</b> /users</p> <p><b>Метод запроса:</b> GET</p> <p><b>Параметры:</b></p> <ul style="list-style-type: none"> <li><i>name</i> — имя пользователя</li> </ul> <p><b>Тело запроса:</b></p> <p>Нет</p> <p><b>Возвращаемые данные:</b></p> <ul style="list-style-type: none"> <li><b>Status code:</b> 200 при успешном запросе, 404 — если ни один пользователь не был найден, 500 — любая другая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера)</li> <li><b>Response body:</b> информация в формате JSON: <ul style="list-style-type: none"> <li>массив <i>ids</i> пользователей с таким</li> </ul> </li> </ul>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 34				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

		именем
	По списку id вернуть id пользователей, которых НЕ существует	<p><b>Относительный url:</b> /users/notexisting</p> <p><b>Метод запроса:</b> GET</p> <p><b>Параметры:</b></p> <p>- <i>ids</i> – id пользователей</p> <p><b>Переменные пути:</b></p> <p>-</p> <p><b>Возвращаемые данные:</b></p> <ul style="list-style-type: none"> <li>• <b>Status code:</b> 200 при успешном запросе, 404 — если ни один пользователь не был найден, 500 — любая другая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера)</li> <li>• <b>Response body:</b> информация в формате JSON: <ul style="list-style-type: none"> <li>• id пользователей, которых не существует (из тех id, которые поступили в запрос)</li> </ul> </li> </ul>
	Получить id пользователя по email и паролю	<p><b>Относительный url:</b> /users/userId</p> <p><b>Метод запроса:</b> GET</p> <p><b>Параметры:</b></p> <p>email — имя пользователя password - пароль</p> <p><b>Тело запроса:</b></p> <p>Нет</p> <p><b>Возвращаемые данные:</b></p> <ul style="list-style-type: none"> <li>• <b>Status code:</b> 200 при найденном пользователе, 404 при отсутствии пользователя с заданными параметрами, 500 — любая другая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера)</li> <li>• <b>Response body:</b> информация в формате JSON:</li> </ul>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 34				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

		<ul style="list-style-type: none"> <li>• id пользователя</li> </ul>
	Получить частичную информацию о пользователях	<p><b>Относительный url:</b> /users/info <b>Метод запроса:</b> GET</p> <p><b>Параметры:</b></p> <p>- ids: массив id пользователей</p> <p><b>Тело запроса:</b></p> <p>-</p> <p><b>Возвращаемые данные:</b></p> <ul style="list-style-type: none"> <li>• <b>Status code:</b> 200 при успешном запросе, 404 — если ни один пользователь не был найден, 500 — любая другая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера)</li> <li>• <b>Response body:</b> <ul style="list-style-type: none"> <li>• массив: <ul style="list-style-type: none"> <li>• id</li> <li>• name</li> <li>• avatar</li> </ul> </li> </ul> </li> </ul>
	Получить информацию обо всех существующих пользователях	<p><b>Относительный url:</b> /users/all <b>Метод запроса:</b> GET</p> <p><b>Параметры:</b></p> <p>-</p> <p><b>Тело запроса:</b></p> <p>-</p> <p><b>Возвращаемые данные:</b></p> <ul style="list-style-type: none"> <li>• <b>Status code:</b> 200 при успешном запросе, 404 — если ни один пользователь не был найден, 500 — любая другая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера)</li> <li>• <b>Response body:</b> <ul style="list-style-type: none"> <li>• массив: <ul style="list-style-type: none"> <li>• id</li> <li>• name</li> <li>• email</li> <li>• phone</li> <li>• organization</li> <li>• avatar</li> <li>• about</li> </ul> </li> </ul> </li> </ul>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 34				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Сервис отправки уведомлений	Создать и зарегистрировать уведомление	<b>Относительный url:</b> /notifications <b>Метод запроса:</b> POST  <b>Параметры:</b> -  <b>Тело запроса:</b> <ul style="list-style-type: none"> <li>• <i>type</i> – тип уведомления (эл.письмо / смс)</li> <li>• <i>content</i> - содержание уведомления</li> <li>• <i>time_to_show</i> – когда уведомление должно показываться</li> <li>• <i>interval_to_repeat</i> — интервал повтора уведомления</li> <li>• <i>user_id</i> — на кого назначено уведомление</li> <li>• <i>immediately</i> — флаг мгновенной доставки уведомления</li> </ul> <b>Ограничения на входные данные:</b> <ul style="list-style-type: none"> <li>• <i>type</i>:               <ul style="list-style-type: none"> <li>◦ присутствует</li> <li>◦ допустимые значения: «sms» и «email»</li> </ul> </li> <li>• <i>content</i>:               <ul style="list-style-type: none"> <li>◦ присутствует</li> </ul> </li> <li>• <i>userId</i>:               <ul style="list-style-type: none"> <li>◦ присутствует</li> </ul> </li> <li>• <i>time_to_show</i> (если присутствует):               <ul style="list-style-type: none"> <li>◦ формат: "yyyy-MM-dd'T'HH:mm:ss"</li> </ul> </li> <li>• недопустимо, чтобы <i>immediately</i> = false И <i>time_to_show</i> = null</li> <li>• <i>interval_to_repeat</i> (если присутствует):               <ul style="list-style-type: none"> <li>◦ формат: PnYnMnDTnHnMnS</li> </ul> </li> </ul> <b>Возвращаемые данные:</b> <ul style="list-style-type: none"> <li>• <b>Status code:</b> 201 при успешном выполнении запроса, 400/409 для некорректных входных данных, 500 — любая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера)</li> <li>• <b>Response body:</b> информация в формате JSON:               <ul style="list-style-type: none"> <li>• id созданного уведомления</li> <li>• все, что в теле запроса (см. выше)</li> </ul> </li> </ul>		
	Редактировать уведомление	<b>Относительный url:</b> /notifications <b>Метод запроса:</b> PUT  <b>Параметры:</b>		
Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 34				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

		<p>-</p> <p><b>Тело запроса:</b></p> <ul style="list-style-type: none"> <li>• <i>type</i> – тип уведомления (эл.письмо / смс)</li> <li>• <i>content</i> - содержание уведомления</li> <li>• <i>time_to_show</i> – когда уведомление должно показываться</li> <li>• <i>interval_to_repeat</i> — интервал повтора уведомления</li> <li>• <i>user_id</i> — на кого назначено уведомление</li> <li>• <i>immediately</i> — нужно ли доставить уведомление мгновенно</li> </ul> <p><b>Ограничения на входные данные:</b></p> <ul style="list-style-type: none"> <li>• <i>type</i>: <ul style="list-style-type: none"> <li>◦ допустимые значения: «sms» и «email» (если присутствует)</li> </ul> </li> <li>• <i>time_to_show</i> (если присутствует): <ul style="list-style-type: none"> <li>◦ формат: "yyyy-MM-dd'T'HH:mm:ss"</li> </ul> </li> <li>• недопустимо, чтобы <i>immediately</i> = false И <i>time_to_show</i> = null</li> </ul> <p>При этом каждое поле, кроме <i>id</i>, может отсутствовать — в таком случае данное поле сохранит свое старое значение.</p> <p><b>Возвращаемые данные:</b></p> <ul style="list-style-type: none"> <li>• <b>Status code:</b> 200 при успешном выполнении запроса, 400/409 для некорректных входных данных, 500 — любая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера)</li> <li>• <b>Response body:</b> информация в формате JSON: все поля обновленного уведомления</li> </ul>
	Удалить уведомление	<p><b>Относительный url:</b> /notifications</p> <p><b>Метод запроса:</b> DELETE</p> <p><b>Переменная пути:</b></p> <ul style="list-style-type: none"> <li>• <i>notificationId</i> – id уведомления</li> </ul> <p><b>Тело запроса:</b></p> <p>Нет</p> <p><b>Возвращаемые данные:</b></p> <ul style="list-style-type: none"> <li>• <b>Status code:</b> 204 при успешном</li> </ul>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 34				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

		<p>выполнении запроса, 404 — если уведомление не найдено, 500 — любая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера)</p> <ul style="list-style-type: none"> <li>• <b>Response body:</b> пустое</li> </ul>
	Удалить уведомления	<p><b>Относительный url:</b> /notifications <b>Метод запроса:</b> DELETE</p> <p><b>Параметры:</b></p> <p>-</p> <p><b>Тело запроса:</b></p> <p>notificationIds — id уведомлений</p> <p><b>Возвращаемые данные:</b></p> <ul style="list-style-type: none"> <li>• <b>Status code:</b> 204 при успешном выполнении запроса, 404 — если ни одно из уведомлений не найдено, 500 — любая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера)</li> <li>• <b>Response body:</b> пустое</li> </ul>
	Получить информацию по уведомлениям по их id	<p><b>Относительный url:</b> /notifications <b>Метод запроса:</b> GET</p> <p><b>Переменная пути:</b></p> <p>notificationIds — id уведомлений</p> <p><b>Тело запроса:</b></p> <p>-</p> <p><b>Возвращаемые данные:</b></p> <ul style="list-style-type: none"> <li>• <b>Status code:</b> 200 при успешном запросе, 404 — если ни одно уведомление не было найдено, 500 — любая другая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера)</li> <li>• <b>Response body:</b> массив с полной информацией об уведомлениях</li> </ul>

Сервис дежурств	Создать дежурство для пользователя	<p><b>Относительный url:</b> /duties <b>Метод запроса:</b> POST</p> <p><b>Параметры:</b></p> <p>-</p>
-----------------	------------------------------------	---

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 34				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

		<p><b>Тело запроса:</b></p> <ul style="list-style-type: none"> <li>• <i>start_time</i> — начало дежурства</li> <li>• <i>name</i> — название дежурства</li> <li>• <i>interval</i> — продолжительность дежурства (1-ой итерации дежурства)</li> <li>• <i>ids</i> — (упорядоченный) массив id дежурных (дежурят по очереди, по завершении дежурства последнего в списке наступает дежурство 1-ого)</li> </ul> <p><b>Ограничения на входные данные:</b></p> <ul style="list-style-type: none"> <li>• <i>start_time</i>: <ul style="list-style-type: none"> <li>◦ должно присутствовать</li> <li>◦ формат: уууу-ММ-дд'T'HH:mm:ss</li> </ul> </li> <li>• <i>name</i>: <ul style="list-style-type: none"> <li>◦ не пусто</li> </ul> </li> <li>• <i>interval</i>: <ul style="list-style-type: none"> <li>◦ должен присутствовать</li> <li>◦ формат: PnYnMnDTnHnMnS</li> </ul> </li> </ul> <p><b>Возвращаемые данные:</b></p> <ul style="list-style-type: none"> <li>• <b>Status code:</b> 201 при успешном запросе, 400/409 при некорректных входных данных, 500 — любая другая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера)</li> <li>• <b>Response body:</b> <ul style="list-style-type: none"> <li>• id дежурства</li> <li>• name</li> <li>• start_time</li> <li>• interval</li> <li>• ids</li> </ul> </li> </ul>
	Получить следующее дежурство пользователя	<p><b>Относительный url:</b> /duties/user</p> <p><b>Метод запроса:</b> GET</p> <p><b>Переменная пути:</b></p> <p><i>userId</i> — идентификатор пользователя</p> <p><b>Тело запроса:</b></p> <p>Нет</p> <p><b>Возвращаемые данные:</b></p> <ul style="list-style-type: none"> <li>• <b>Status code:</b> 200 при успешном выполнении запроса, 404, если пользователь не найден, 500 — любая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера)</li> </ul>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 34				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

		<ul style="list-style-type: none"> <li>• <b>Response body:</b> информация о дежурстве пользователя в формате JSON: <ul style="list-style-type: none"> <li>• дата его следующего дежурства</li> <li>• продолжительность дежурства</li> <li>• периодичность (через сколько он снова становится дежурным)</li> </ul> </li> </ul>
	Редактировать дежурство	<p><b>Относительный url:</b> /duties <b>Метод запроса:</b> PUT</p> <p><b>Параметры:</b></p> <p>-</p> <p><b>Тело запроса:</b></p> <ul style="list-style-type: none"> <li>• <i>id</i> — идентификатор дежурства</li> <li>• <i>name</i> — название дежурства</li> <li>• <i>start_time</i> — новое начало дежурства</li> <li>• <i>interval</i> – новый конец дежурства</li> <li>• <i>ids</i> – новый массив id дежурных</li> </ul> <p>При этом каждое поле, кроме id, может отсутствовать — в таком случае данное поле сохранит свое старое значение.</p> <p><b>Ограничения на входные данные:</b></p> <ul style="list-style-type: none"> <li>• id: <ul style="list-style-type: none"> <li>◦ должен присутствовать</li> </ul> </li> <li>• start_time (если присутствует): <ul style="list-style-type: none"> <li>◦ формат: уууу-ММ-дд'T'HH:mm:ss</li> </ul> </li> <li>• interval (если присутствует): <ul style="list-style-type: none"> <li>◦ формат: PnYnMnDTnHnMnS</li> </ul> </li> </ul> <p><b>Возвращаемые данные:</b></p> <ul style="list-style-type: none"> <li>• <b>Status code:</b> 200 при успешном выполнении запроса, 404, если дежурство не нашлось, 500 — любая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера)</li> <li>• <b>Response body:</b> информация в формате JSON: все поля обновленного уведомления</li> </ul>
	Получить информацию о дежурстве	<p><b>Относительный url:</b> /duties <b>Метод запроса:</b> GET</p> <p><b>Переменная пути:</b></p>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 34				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата



		<p><i>id</i> — идентификатор пользователя</p> <p><b>Тело запроса:</b></p> <p>Нет</p> <p><b>Возвращаемые данные:</b></p> <ul style="list-style-type: none"> <li>• <b>Status code:</b> 200 при успешном выполнении запроса, 404 — дежурство не найдено, 500 — любая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера)</li> <li>• <b>Response body:</b> информация о дежурстве в формате JSON: <ul style="list-style-type: none"> <li>• id</li> <li>• name</li> <li>• start_time</li> <li>• interval</li> <li>• ids</li> <li>• currentlyUserDutyId</li> </ul> </li> </ul>
	Получить информацию обо всех дежурствах	<p><b>Относительный url:</b> /duties</p> <p><b>Метод запроса:</b> GET</p> <p><b>Переменная пути:</b></p> <p>-</p> <p><b>Тело запроса:</b></p> <p>Нет</p> <p><b>Возвращаемые данные:</b></p> <ul style="list-style-type: none"> <li>• <b>Status code:</b> 200 при успешном выполнении запроса, 500 — любая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера)</li> <li>• <b>Response body:</b> информация о дежурствах в формате JSON: <ul style="list-style-type: none"> <li>• id</li> <li>• name</li> <li>• start_time</li> <li>• interval</li> <li>• ids</li> </ul> </li> </ul>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 34				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

		• currentlyUserDutyId
--	--	-----------------------

### 3.3. Мониторинг

Для мониторинга работы системы предусмотрены:

- healthchecks (флаги «здоровья» сервиса) в docker-контейнерах — проверка того, что сервисы и их базы данных запущены и функционируют
- логирование (файлы-логи сохраняются в директорию logs с поддиректориями для каждого из сервисов) — отчет о работе сервисов (например, в случае сообщения пользователями 500-ой ошибке можно посмотреть в лог — в нем будет более точное сообщение о причине, нежели фраза «Internal Server Error»)
- утилита htop — утилита для мониторинга загрузки ресурсов компьютера — может понадобится для анализа повышенной утилизации ЦП или нехватки ОЗУ во время работы.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 34				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

#### 4. СООБЩЕНИЯ ОПЕРАТОРУ

API может возвращать пользователю сообщения и ошибки со следующими кодами:

Код	Текст	Причина
200	Success	Запрос выполнен успешно
201	Created	Запрос выполнен успешно, запись создана (в базе данных)
204	No content	Запрос выполнен успешно, ничего не возвращено (например, запрос на удаление)
400	Bad Request	В запросе содержится информация, которой нет в базе данных
404	Not Found	По отправленному запросу не нашлось требуемых данных или запрашиваемая конечная точка API не существует
409	Conflict	Запрос на добавление нового ресурса на сервер не был обработан, так как этот ресурс уже существует или нарушены ограничения на данные
500	Internal Server Error	Во время выполнения запроса произошла ошибка, которую невозможно корректно обработать (например, недоступность базы данных, другого сервиса или необработанное исключение в алгоритме)

Каждая ошибка содержит описание, не раскрывающее детали реализации сервисов.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 34				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

### СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

- 1) Статья про REST API в целом [Электронный ресурс]. Режим доступа: <https://habr.com/ru/articles/483202/>, свободный. (дата обращения: 9.03.2024)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 34				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

[illegible]