# **Техническое Задание**

## **Описание**

Разработать систему для управления данными о сотрудниках и департаментах в компании.

## **Функциональные требования**

1. В одном департаменте может работать несколько сотрудников.
2. Департамент может быть без сотрудников.
3. Cотрудник может работать только в одном департаменте.
4. Не может быть сотрудника не привязанного ни к одному департаменту.
5. Хранить следующую информацию о сотрудниках и департаментах. Все поля обязательны:

|  |  |
| --- | --- |
| **Департамент** | |
| **Поле** | **Тип Данных** |
| Идентификатор | Целое число назначаемое системой автоматически. |
| Название | Строка. 0 < кол-во символов <= 50  Значение уникально в рамках всей системы. |

|  |  |
| --- | --- |
| **Сотрудник** | |
| **Поле** | **Тип Данных** |
| Идентификатор | Целое число назначаемое системой автоматически. |
| Имя | Строка. 0 < кол-во символов <= 50 |
| Зарплата | Целое положительное число. |
| Должность | Строка. 0 < кол-во символов <= 50 |

1. Предоставить REST API для осуществленния следующих операций:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Метод** | **Путь** | **Описание** | **Запрос (JSON)** | **Ответ (JSON)** |
| GET | /departments | Получить список всех департаментов | - | 200 OK  [      {          "id": [число],          "name": [строка]      }  ] |
| GET | /departments/{id} | Получить департамент по идентификатору | - | 200 OK  {  "id": [число],      "name": [строка]  } |
| DELETE | /departments/{id} | Удалить департамент по идентификатору. Но нельзя удалить департамент к которому привязан хотя бы один сотрудник. | - | 204 NO CONTENT |
| CREATE | /departments | Создать департамент | {      "name": [строка]  } | 201 CREATED  {      "id": [число],      "name": [строка]  } |
| UPDATE | /departments/{id} | Изменить департамент по идентификатору | {      "name": [строка]  } | 200 OK  {      "id": [число],      "name": [строка]  } |
| GET | /employees | Получить список всех сотрудников | - | 200 OK  [      {          "id": [число],          "name": [строка],          "role": [строка],          "salary": [число],          "department": {              "id": [число],              "name": [строка]          }      }  ] |
| GET | /employees/{id} | Получить сотрудника по идентификатору | - | 200 OK  {  "id": [число],      "name": [строка],      "role": [строка],      "salary": [число],      "department": {      "id": [число],          "name": [строка]      }  } |
| DELETE | /employees/{id} | Удалить сотрудника по идентификатору | - | 204 NO CONTENT |
| CREATE | /employees | Создать сотрудника | {      "name": [строка],      "role": [строка],      "salary": [число],      "departmentId": [число]  } | 201 CREATED  {      "id": [число],      "name": [строка],      "role": [строка],      "salary": [число],      "department": {          "id": [число],          "name": [строка]      }  } |
| UPDATE | /employees/{id} | Изменить сотрудника по идентификатору | {  "id": [число],      "name": [строка],      "role": [строка],      "salary": [число],      "departmentId": [число]  } | 200 OK  {      "id": [число],      "name": [строка],      "role": [строка],      "salary": [число],      "department": {          "id": [число],          "name": [строка]      }  } |

1. Обработка ошибок:

|  |  |
| --- | --- |
| **Условие** | **Поведение** |
| Невалидный формат запроса | Сервер возвращает ответ со статусом 400 (Bad Request) |
| Обращение к несуществующему ресурсу (например указание несуществующего идентификатора сотрудника) | Сервер возвращает ответ со статусом 404 (Bad Request) |
| Попытка удалить департамент к которому привязан хотя бы один сотрудник | Сервер возвращает ответ со статусом 403 (Forbidden) |
| Ошибка на сервере | Сервер возвращает ответ со статусом 500 (Internal Server Error) |

## **Нефункциональные требования**

1. Используемые технологии: Java, Spring Boot, Hibernate, Lombok, PostgresSQL, Maven, Git, Postman.
2. Код проекта доступен на GitHub.
3. Пример схемы базы данных:

