

Технологии:

Spring Boot
Spring JPA Repository
H2 Database

**Spring Boot:** Используйте Spring Boot для создания RESTful API вашего приложения для управления проектами. Spring Boot предоставляет простой и эффективный способ разработки Java-приложений, беря на себя задачи по настройке и инфраструктуре.

**JPA Repository:** Воспользуйтесь JPA (Java Persistence API) и интерфейсом JPA Repository для обработки сохранения данных. Определите сущности вашего проекта (например, Проект, Задача, Пользователь) в виде классов Java и аннотируйте их с использованием JPA-аннотаций, таких как @Entity, @Table и @Column. Создайте интерфейс JPA Repository для каждой сущности, чтобы обеспечить базовые операции CRUD.

**H2 Database:** H2 - это встроенная реляционная база данных, которую легко интегрировать с Spring Boot. Настройте ваше приложение на использование H2 в качестве базы данных. H2 является легким, быстрым и отлично подходит для маломасштабных приложений или для разработки.

Теперь давайте рассмотрим основные функции вашего приложения для управления проектами:

**Управление проектами:** Пользователи должны иметь возможность создавать, обновлять и удалять проекты. Каждый проект может иметь название, описание, дату начала, дату окончания и связанные задачи.

**Управление задачами:** Пользователи могут создавать, обновлять и удалять задачи в рамках проекта. Каждая задача может иметь название, описание, срок выполнения и статус (например, "in progress", "done").

**Назначение задач:** Позвольте пользователям назначать задачи себе или другим участникам команды. Пользователи должны иметь возможность просматривать назначенные им задачи и отмечать их выполненными.

**Поиск и фильтрация:** Реализуйте функции поиска и фильтрации, чтобы пользователи могли находить конкретные проекты или задачи на основе критериев, таких как название, статус или диапазон дат.

Точки доступа для управления пользователями:

HTTP Request Method	Path	Описание
POST	/api/register	Регистрация нового пользователя
GET	/api/user/profile	Получение информации о профиле пользователя
PUT	/api/user/profile	Обновление информации о профиле пользователя
PUT	/api/user/password	Обновление пароля пользователя
DELETE	/api/user/profile	Удаление учетной записи пользователя

Точки доступа для управления проектами:

HTTP Request Method	Path	Описание
<b>GET</b>	/api/projects	Получение списка всех проектов.
<b>POST</b>	/api/projects	Создание нового проекта
<b>GET</b>	/api/projects/{projectId}	Получение информации о конкретном проекте
<b>PUT</b>	/api/projects/{projectId}	Обновление информации о конкретном проекте
<b>DELETE</b>	/api/projects/{projectId}	Удаление конкретного проекта

Точки доступа для управления задачами

HTTP Request Method	Path	Описание
<b>GET</b>	/api/projects/{projectId}/tasks	Получение списка задач для конкретного проекта
<b>POST</b>	/api/projects/{projectId}/tasks	Создание новой задачи в рамках проекта
<b>GET</b>	/api/projects/{projectId}/tasks/{taskId}	Получение информации о конкретной задаче в рамках проекта
<b>PUT</b>	/api/projects/{projectId}/tasks/{taskId}	Обновление информации о конкретной задаче в рамках проекта
<b>DELETE</b>	/api/projects/{projectId}/tasks/{taskId}	Удаление конкретной задачи в рамках проекта

Точки доступа для управления назначениями:

HTTP Request Method	Path	Описание
<b>GET</b>	/api/tasks/assigned/{userId}	Получение списка задач, определенного пользователя
<b>PUT</b>	/api/tasks/{taskId}/assign/{userId}	Назначение задачи пользователю.
<b>PUT</b>	/api/tasks/{taskId}/complete	Пометка задачи как выполненной