

Тестовое задание - Meetup API

Разработать CRUD REST Web API для работы с митапами

Функционал Web API

1. Получение списка всех митапов;
2. Получение определённого митапа по его Id;
3. Регистрация нового митапа;
4. Изменение информации о существующем митапе;
5. Удаление митапа.

Информация о митапе

1. Название / тема;
2. Описание;
3. Набор тегов / ключевых слов;
4. Время и место проведения.

Стек

1. Node.js;
2. Express;
3. PostgreSQL 14.

Примечания

1. Качество очень важно. Несмотря на то, что задача достаточно проста, прошу, не оставляйте без внимания качество кода и следование принципам разработки. Очень часто допускаются следующие проблемы:
 - a. Качество кода (в том числе нейминг и форматирования) оставляют желать лучшего;
 - b. Некорректно ведётся работа с Git (указан не тот email пользователя, весь код залит 1м коммитом, закоммичены не те файлы);
 - c. DTO не используются вообще, или используются некорректно
 - d. Нарушается REST принцип, или общие правила использования HTTP методов, статус кодов и, особенно, правила формирования URL
2. Не стоит разрабатывать большое и сложное решение. Начните с простого – напишите весь код в сразу контроллерах, приведите его в порядок. Если после этого у вас останется время и желание для усовершенствования задания,

обратите внимание на возможные направления расширения (описаны далее).

Приступайте к решению архитектурно-организационных вопросов только если вы полностью реализовали все перечисленные расширения.

3. Внимательно читайте описание задания – задание состоит в том, чтобы реализовать REST Web API, а не полноценное Full Stack Web приложение (не нужно добавлять веб-интерфейс).

Дополнительная работа

Если вы реализовали основные требования тестового задания, привели в порядок код, Git-репозиторий и неделя с момента выдачи задания ещё не прошла – можно расширить проект следующими способами (отсортированы в порядке нарастания сложности и затрачиваемого времени на реализацию):

1. Валидация DTO (достаточно будет использовать [joi](#)).
2. Переработать запрос на получение списка митапов так, чтобы с его помощью можно было осуществить поиск по митапам, отфильтровать их, отсортировать. Результат также должен быть разбит на страницы.
3. Добавить документацию к API через Swagger.
4. Система аутентификации (предлагается к использованию библиотека [PassportJS](#)). Систему аутентификации можно разрабатывать поэтапно, постепенно расширяя и усложняя её:
 - a. Пользователь регистрируется, авторизуется (получая JWT Access Token). Используя Access Token можно получить информацию о владельце токена (т.е. получить текущего пользователя по токену) и записаться на митап.
 - b. Добавить персистентный JWT Refresh Token.
 - c. Разделить авторизованных пользователей на 2 типа: обычные пользователи (описаны в пункте 3.а) и организаторы митапов (только они могут регистрировать новые митапы, редактировать/удалять зарегистрированные ими митапы)

