Цель­

Сделать back-end приложение на Node.js, которое предоставляет REST API по протоколу HTTP 1.1, представляющий функциональность пользовательских комментариев с древовидной структурой.

Требования

* В качестве СУБД использовать MongoDB.
* Оформить проект и его структуру так, как вы делали бы это на большом highload проекте.
* Покрыть тестами >90% кода (1) интеграционными тестами (2).
* Проект должен быть оформлен так, чтобы он начал работать после этого:

npm i  
npm test  
# тут должно быть много тестов и желательно, чтобы они завершились успехом.  
rm -rf node\_modules  
npm i --prod  
npm start

При условии, что MongoDB и Node.js уже установлены

* Версии Node.js и MongoDB должны соответствовать официальным стабильным версиям на момент выполнения тестового задания.
* Для REST API можно использовать любые модули из npm.

*(1) - подсчёт покрытия тестами можно сделать при помощи* [*https://www.npmjs.com/package/istanbul*](https://www.npmjs.com/package/istanbul)*. В подсчёте покрытия не нужно учитывать сами файлы с тестами и внешние модули. Будет плюсом, если покрытие тестами будет значительно больше 90%.*

*(2) - можно почитать на википедии, но в общем смысле - это тест, который делает запрос к REST API имитируя действия пользователя.*

Описание задания

1. Должен быть реализован REST API.
2. Сделать методы по регистрации и авторизации через логин с паролем.
3. Методы для создания и получения списка комментариев.
4. Метод, который рассчитывает максимальный уровень вложенности в дереве комментариев. Уровень комментария в модели хранить нельзя, метод должен рассчитывать его.
5. Метод, который будет возвращать массив пользователей с количеством их комментариев и этот массив должен быть отсортирован по убыванию количества комментариев (пользователь с наибольшим количеством комментариев должен быть всегда сверху). Желательно реализовать одним запросом к БД.

Пожелания

* В файле package.json установить требование к Node.js в соответствии с <https://docs.npmjs.com/files/package.json>, так мы узнаем о требовании к версии Node.js. А требования к версии MongoDB можно будет сообщить отдельно в письме.

Пожелания к использованию модулей:

* <https://www.npmjs.com/package/jsonwebtoken>
* <https://www.npmjs.com/package/gulp>
* <https://www.npmjs.com/package/chai>
* <https://www.npmjs.com/package/passport>
* <https://www.npmjs.com/package/mongoose>

Их использование не обязательно, можно использовать другие, аналогичные модули. Если вам не жалко времени, можно вообще всё самому сделать без использования модулей. Например, для того чтобы писать и запускать тесты не обязательно использовать gulp и chai, достаточно нативного assert и правильно оформленного scripts.test и package.json.

Примечание

Структуру хранения комментариев в БД можно выбрать любую, но нужно будет обосновать своё решение, указать на плюсы и минусы вашего решения.

Алгоритм подсчёта 4 и 5 методов backend-а так же можно реализовать разными способами. При обсуждении вашего решения хотелось бы узнать чем вы руководствовались выбирая алгоритм расчёта.

Желательно ознакомиться с информацией:

http://jstherightway.org/

https://github.com/rwaldron/idiomatic.js