Тестовое задание:

Требования:  
Желательно использовать версии: Ruby 2.2+, RoR 4+, PostgreSQL 9.2+.  
Результат лучше всего опубликовать на github.  
  
Задание на знания Ruby on Rails:  
У нас имеется некий блог со следующими сущностями:  
  
1. Юзер. Имеет только логин.  
2. Пост, принадлежит юзеру. Имеет заголовок, содержание, айпи автора (сохраняется отдельно для каждого поста).  
3. Оценка, принадлежит посту. Принимает значение от 1 до 5.  
  
Задача: создать JSON API на RoR со следующими экшенами:  
  
1. Создать пост. Принимает заголовок и содержание поста (не могут быть пустыми), а также логин и айпи автора. Если автора с таким логином еще нет, необходимо его создать. Возвращает либо атрибуты поста со статусом 200, либо ошибки валидации со статусом 422.  
2. Поставить оценку посту. Принимает айди поста и значение, возвращает новый средний рейтинг поста. Важно: экшен должен корректно отрабатывать при любом количестве конкурентных запросов на оценку одного и того же поста.  
3. Получить топ N постов по среднему рейтингу. Просто массив объектов с заголовками и содержанием.  
4. Получить список айпи, с которых постило несколько разных авторов. Массив объектов с полями: айпи и массив логинов авторов.  
  
Базу данных используем PostgreSQL. Для девелопмента написать скрипт в db/seeds.rb, который генерирует тестовые данные. Часть постов должна получить оценки. Скрипт должен использовать тот же код, что и контроллеры, можно вообще дергать непосредственно сервер курлом или еще чем-нибудь.  
  
Постов в базе должно быть хотя бы 200к, авторов лучше сделать в районе 100 штук, айпишников использовать штук 50 разных. Экшены должны на стандартном железе работать достаточно быстро как для указанного объема данных (быстрее 100 мс), так и для намного большего, то есть нужен хороший запас в плане оптимизации запросов. Для этого можно использовать денормализацию данных и любые другие средства БД. Можно использовать любые нужные гемы, обязательно наличие спеков, хорошо покрывающих разные кейсы. В коде желательно не использовать рельсовых антипаттернов типа колбеков и валидаций в моделях, сервис-классы наше все. Также желательно не использовать генераторов и вообще обойтись без лишних мусорных файлов в репозитории.  
  
Задание на знание SQL:  
дана таблица users вида - id, group\_id  
  
create temp table users(id bigserial, group\_id bigint);  
insert into users(group\_id) values (1), (1), (1), (2), (1), (3);  
    1    В этой таблице, упорядоченой по ID необходимо:  
    2    выделить непрерывные группы по group\_id с учетом указанного порядка записей (их 4)  
    3    подсчитать количество записей в каждой группе  
    4    вычислить минимальный ID записи в группе