



OTUS


ОНЛАЙН-ОБРАЗОВАНИЕ

Онлайн-образование



Меня хорошо видно && слышно?

Ставьте ☐+, если все хорошо
Напишите в чат, если есть проблемы



Защита проекта

Тема: Сервис хранения прогресса студента



Тимаков Николай Викторович

Web-разработчик

Едем-в-гости.ру

Цели проекта

Сервис должен обеспечивать хранение данных о выполнении студентом учебных заданий

Имеется спектр возможных для обучения навыков

Для каждого задания задаётся распределение спектра навыков с долями (например, задание «диалог с преподавателем на английском языке» тренирует навык говорения на 70% и навык аудирования на 30%)

Задание оценивается в баллах в диапазоне от 1 до 10, баллы распределяются по навыкам пропорционально

Задания группируются в занятия, занятия группируются в модули, а модули — в курсы

Цели проекта

Сервис также должен решать следующие задачи:

- Агрегация баллов за задания по занятиям
- Агрегация баллов по навыкам
- Агрегация баллов по времени
- Агрегация баллов по модулям

Цели проекта

- CRUD-операции с сущностями сервиса должны быть реализованы с использованием API Platform
- Сохранение баллов должно выполняться в асинхронном режиме
- Получение агрегаций по студенту должно быть реализовано с использованием GraphQL
- Агрегированные данные должны кешироваться

Что планировалось

1. Сделать разрабатываемый сервис bundle-ом.

2. Получение агрегаций с использованием GraphQL.

3. Сохранение данных в асинхронном режиме с использованием RabbitMQ.

4. Кеширование данных с использованием Redis.

Используемые технологии

1. API-Platform

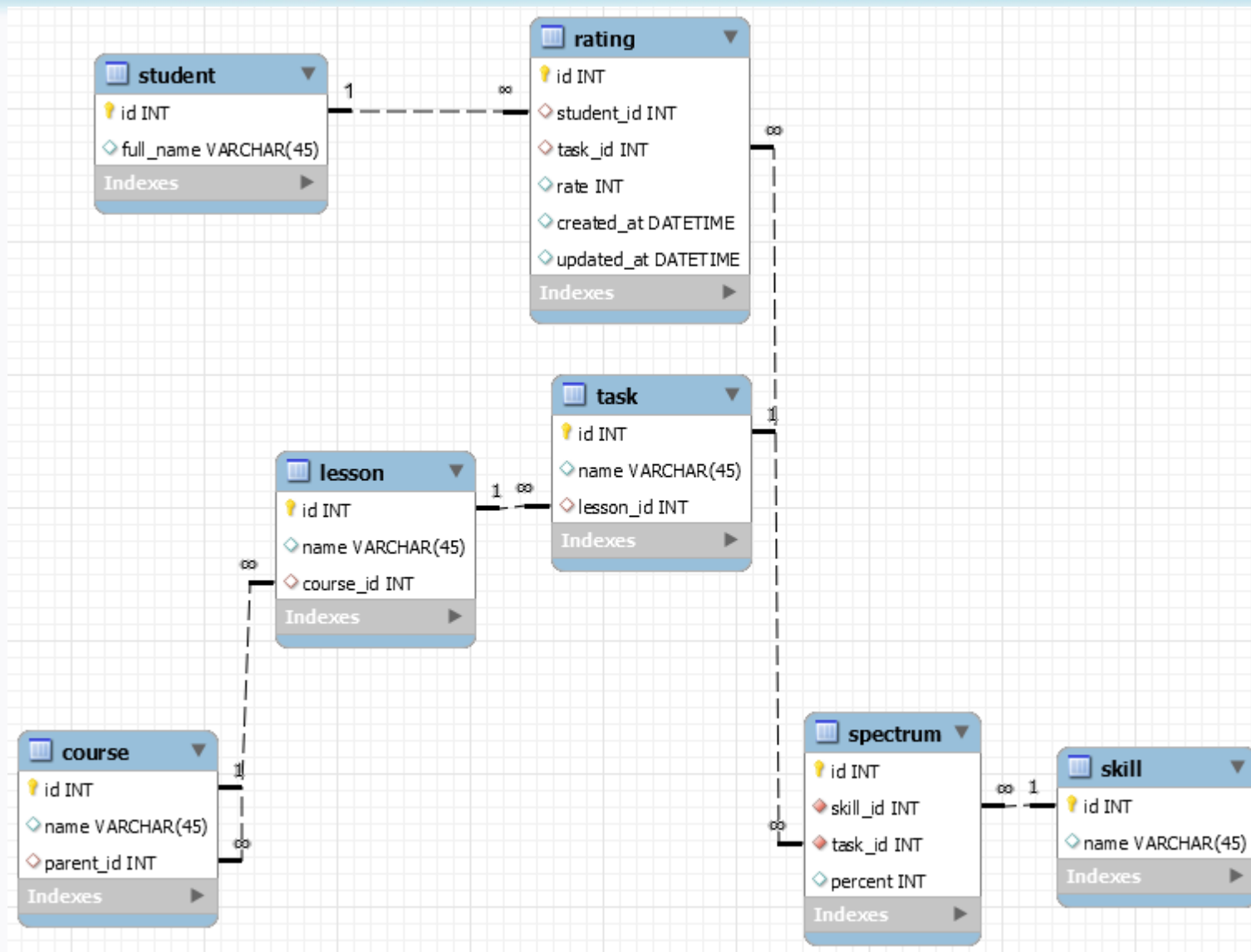
2. GraphQL

3. Redis

4. RabbitMQ



Схема БД



Что получилось

Получение агрегаций с использованием GraphQL

Добавлены 4 одностипных DTO для агрегаций со следующими структурами:

- Агрегация баллов по модулям: id-студента и массив modules. Массив modules содержит следующие ключи: moduleId, moduleName и sum (сумма баллов по соответствующему модулю).
- Агрегация баллов по занятиям: id-студента и массив lessons. Массив lessons содержит следующие ключи: lessonId, lessonName и sum (сумма баллов по соответствующему занятию).
- Агрегация баллов по навыкам: id-студента и массив skills. Массив skills содержит следующие ключи: skillId, skillName и sum (сумма баллов по соответствующему навыку).
- Агрегация баллов по времени: id-студента и массив datetime. Массив datetime содержит следующие ключи: createdAt и sum (сумма баллов по соответствующему времени).

Что получилось

В сущность студента добавлены 4 необязательных свойства (по умолчанию равны null) с типами 4-х указанных выше DTO.

В resolver-е сущности студента анализируется GraphQL-запрос и при необходимости вычисляется необходимая агрегация/и.

За тем, создается экземпляр/ы DTO и присваиваются соответствующим свойствам сущности студента.

Таким образом, по каждому студенту можно получить любую или все агрегации.

Что получилось

Сохранение данных в асинхронном режиме

В контроллер сохранения баллов за задания добавлено входящее DTO, следующей структуры: `Id`, `rate`, `taskId` и `studentId`.

В `transformer`-е происходит валидация данного DTO:

- проверяется значение `rate`, которое должно быть в диапазоне от 1 до 10
- `taskId` — проверяется, что задание с таким `id` существует
- `studentId` — также проверяется, что студент с таким `id` существует.

Если ошибок валидации нет, то в `RabbitMQ` отправляется сообщение. После обработки сообщения из `RabbitMQ` данные сохраняются в БД.

Что получилось

Кеширование агрегированных данных

Кеширование агрегированных данных разработано с использованием Redis.

Агрегации помечаются тегами следующим образом:

- по курсам - тегом «courses» и добавляется параметр studentId
- по занятиям - тегом «lessons» и добавляется параметр studentId
- по навыкам - тегом «skills» и добавляется параметр studentId
- по времени - тегом «courses_studentId», т. е. по данной агрегации столько тегов, сколько студентов.

Что получилось

Инвалидация кеша происходит в subscriber-е события onFlush:

- если изменяется/добавляется/удаляется элемент из таблиц skill или spectrum (навыки и спектр навыков), то инвалидируется тег «skills»
- если изменяется/добавляется/удаляется элемент из таблиц course или lesson (курсы, модули и занятия), то инвалидируются теги «courses» и «lessons»
- если изменяется/добавляется/удаляется элемент из таблицы task (задания), то инвалидируются теги «skills», «courses» и «lessons»
- если изменяется/добавляется/удаляется элемент из таблицы rating (оценки за задания), то инвалидируются теги «skills», «courses», «lessons» и «datetime_studentId».

Вывод и планы по развитию

1. Добавить RBAC-модель, настроить доступ к контроллерам и работу с токенами.
2. Покрытие юнит и интеграционными тестами.
3. Добавить инструменты логирования и мониторинга.





Спасибо за внимание!

Тимаков Николай Викторович

Web-разработчик

Едем-в-гости.ру

Тел: +7 (966) 006-85-85