

Обмен на данни между слоевете

Иван Арабаджийски

- Какво значи обмяна на данни?
- Кои са слоевеете?

Клиент-Сървър-База данни

- Как?

Клиентът изпраща заявка към сървъра, а той от своя страна прави необходимите записи в базата данни.



ORM

Object relational mapping (ORM)

- Какво прави? Улеснява ни живота
- За да извлечем информация от база данни трябва да направим заявка. Това означава ли, че трябва да знаем SQL? Далеч не. Благодарение на ORM можем да пишем заявки на език за програмиране по избор.
- Transactions? Roll back?



Видове полезни ORM библиотеки

ORM библиотеките са софтуерни инструменти, които улесняват комуникацията между обектно-ориентираното програмиране и реляционните бази данни. Те позволяват на програмистите да работят с обекти в програмния код, като автоматично превръщат тези обекти в редове в базата данни и обратно, без директно манипулиране на SQL заявки.



Генератори на Заявки (Query Builders)

Query builders са инструменти, които улесняват създаването на SQL заявки чрез използване на програматични интерфейси или методи. Те предлагат удобен начин за конструиране на сложни заявки, като използват обекти или функции за генериране на SQL код. Query builders са полезни, когато искате да създадете динамични заявки, без да се налага да пишете ръчно SQL код. Някои популярните такива включват Knex.js , Sequelize, Objection.js, Mongoos и други за JavaScript



Скриптове за Миграции (Migration Scripts)

Скриптовите за миграции са средство за автоматизирано управление на схемата на базата данни. Те позволяват на разработчиците да дефинират и прилагат промени в структурата на базата данни посредством скриптове, които се изпълняват последователно. Това включва създаване, промяна и изтриване на таблици, индекси, колони и други обекти в базата данни. Те са особено полезни в среди за разработка на софтуер, където е необходимо да се поддържат версии и управление на схемата на базата данни.



Функционалност за попълване на данни (Seeding Feature)

Функционалността за попълване на данни е инструмент, който позволява на разработчиците да автоматизират процеса на добавяне на тестови или изходни данни в базата данни. Това е полезно, когато искате да попълните базата данни с предварително дефинирани данни или със случайно генерирани данни за тестване на приложението. Тя на данни може да се използва за създаване на начални данни за приложението или за тестване на различни сценарии и случаи на употреба.



JS ORM библиотеки

- Knex.js: SQL Query Builder
- Sequelize
- Objection.js
- Bookshelf
- Waterline
- Mongoose



TS ORM библиотеки

- TypeORM
- MikroORM
- Prisma



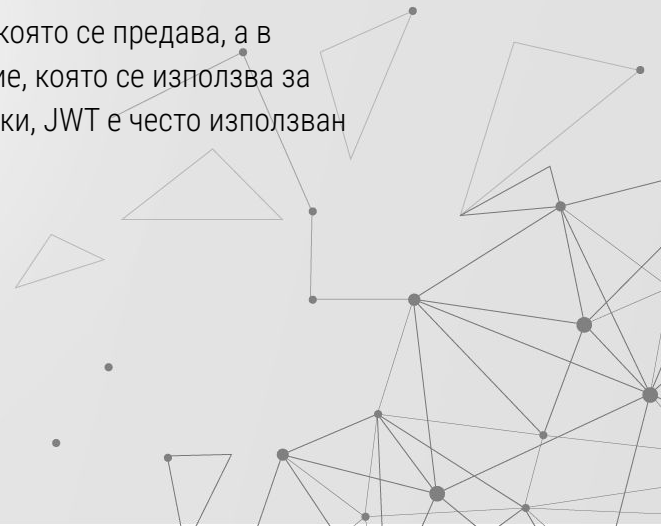
MongoDB и NoSQL бази данни

NoSQL базите данни като MongoDB предлагат гъвкав модел за съхранение на данни, различен от този на релационните бази данни. MongoDB използва документ-ориентиран подход, където данните се съхраняват в JSON-подобни документи, което улеснява обмена на данни с приложенията.



JWT

JWT (JSON Web Token) е отворен стандарт за сигурно обмяне на токени между страни чрез JSON формат. Този токен съдържа информация за потребителя и се използва за потвърждаване на техния идентитет по време на авторизация. Един от ключовите аспекти на JWT е, че той е самодостатъчен, което означава, че данните в него са достатъчни за валидиране на токена без необходимост от допълнителни заявки към сървъра. Този стандарт е широко използван в уеб разработката за създаване на сигурни и мащабируеми приложения. В основата си, JWT се състои от три основни части: заглавие (header), полезно съдържание (payload) и подпис (signature). В заглавието се посочва типът на токена и използваният алгоритъм за криптиране, в полезното съдържание се съдържа информацията, която се предава, а в подписа се включва криптираната версия на заглавието и полезното съдържание, която се използва за потвърждение на автентичността на токена. Благодарение на тези характеристики, JWT е често използван за сесийно управление и за осигуряване на сигурност в уеб приложенията.

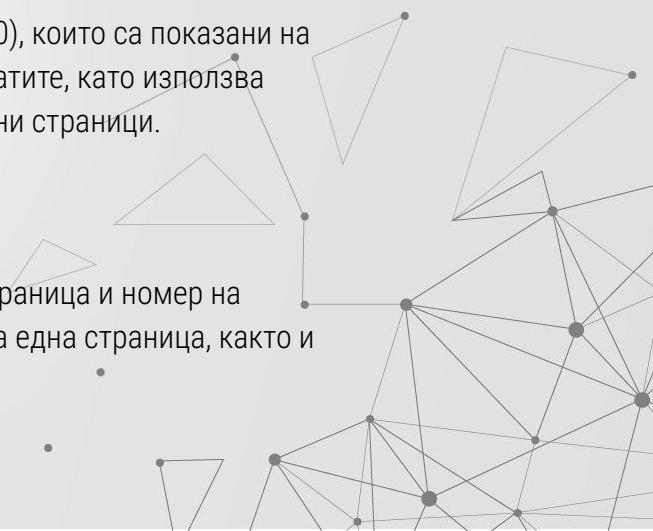


Странициране

Страницирането при заявки е техника, която се използва за ограничаване на резултатите от заявка към база данни или друг източник на данни до определен брой записи на страница. Това е полезно, когато имаме големи обеми от данни, които трябва да бъдат представени на потребителя, но е непрактично или неефективно да се зареждат всичките едновременно.

То работи като заявката връща само определен брой записи (например 10 или 20), които са показани на потребителя на текущата страница. Потребителят може да навигира през резултатите, като използва бутони за предишна и следваща страница или номерирани линкове към конкретни страници.

При страницирането се използва обикновено два параметъра: брой записи на страница и номер на страница. Това позволява на потребителя да променя броя записи, които види на една страница, както и да преминава към различни части от резултатите.



Други важни теми засегнати из примерите

Обработка на заявката за
галерия. Странициране.
Подходящи заявки за
странициране на данните.

Обработка на POST заявка за
редактиране или създаване на
потребителски профил от
учебния проект.

