УДК 004.41 DOI: https://doi.org/10.30837/IYF.IIS.2024.810

РОЗРОБКА ФРЕЙМВОРКУ ДЛЯ ГЕНЕРАЦІЇ КОДУ ВЗАЄМОДІЇ МІЖ ВЕБ-ЗАСТОСУНКОМ І РЕЛЯЦІЙНОЮ БАЗОЮ ДАНИХ

Ільїн І. О.
Науковий керівник – ст. викл. каф. СТ Калайда Н. С.
Харківський національний університет радіоелектроніки, каф. СТ
м. Харків, Україна
e-mail: ivan.ilin@nure.ua

The report discusses the stages of designing and creating a framework for generating code for the interaction between a web application and a relational database, using the metaprogramming paradigm on the Node.js and PostgreSQL technology stack. The code base and physical data model are generated at the compilation stage of the project using a subject-oriented programming language with the possibility of further modification.

У докладі розглядаються етапи проектування та створення фреймворку для генерації коду взаємодії між веб застосунком і реляційною базою даних [1], з використанням парадигми метапрограмування на технологічному стеку Node.js та PostgerSQL. Генерація кодової бази та фізичної моделі даних відбувається на етапі компіляції проєкту за допомогою предметно-орієнтованої мови програмування з можливістю подальшої модифікації.

Під час прикладної розробки інформаційних систем (ІС) в розробника виникає потреба в реалізацій фізичної моделі даних згідно з архітектурою проєкту, а також в написанні коду для серверної частини веб-застосунку. А саме для прикладних програмних інтерфейсів (API), для репрезентації вхідних даних data transfer object (DTO) або репрезентації об'єктів внутрішньої моделі даних (Entity), для валідації DTO, для перетворення DTO у Entity та навпаки (Mapper), для операції вводу-виводу між серверною частиною веб-застосунка і реляційною базою даних, для формування документації API [2]. Зазвичай цей код є типовим для багатьох ІС і відрізняється здебільшого лише структурою даних. В деяких випадках перед та/або після перетворення DTO на Entity можуть відбуватися додаткові обчислення або запити до інших систем.

У поточному докладі розглядається лише простий користувацький сценарій, де до API веб-застосунку для створення нового запису у ІС спочатку надходить відповідний запити з DTO. Потім цей об'єкт перетворюється на Entity і зберігається у базі даних. Після завершення створення нового запису у системі застосунок надсилає відповідь про успішне виконання роботи. Треба зазначити, що в даному випадку немає значення яка структура буде в об’єктів або в моделі даних, як їх потрібно перетворювати один в одного та яку кількість API матиме сервіс. Увесь код призначений для цього є шаблонним і може бути масштабованим, а це свідчить про те, що процес його написання можливо автоматизувати.

Для автоматизації процесу генерації коду необхідно визначити лише структуру фізичної моделі даних, структуру даних що надходять до веб-застосунку та правила їх перетворення. Цих даних буде достатньо для генерації API, DTO, Entity, коду відповідного за валідацію DTO, коду відповідного за перетворення DTO на Entity та навпаки, шару відповідного за взаємодію з базою даних Repository, а також генерації документації для API.

Фреймворк надає можливість розробнику описувати за допомогою предметно-орієнтованої мови програмування структуру фізичної моделі даних у вигляді JavaScript об'єкта. За допомогою CLI-команди або під час запуску застосунку фреймворк на основі цього об'єкта згенерує та виконає SQL-скрипти для створення необхідних таблиць, колонок, обмежень, тригерів, послідовностей тощо [3]. Далі буде згенеровано Entity та шар для взаємодії з базою даних. Цей шар буде містити методи для виклику бази даних де попередньо на основі Entity буде сформовано, або сирий SQL-запит до бази даних, або буде сформовано запит за допомогою Knex.js.

Наступним кроком розробник так само у вигляді JavaScript об'єкта описує структуру вхідних даних, на основі яких фреймворк згенерує DTO та API яке його приймає, супутні роутери у разі використання інших фреймворків, наприклад Express.js, а також документацію для API, наприклад за специфікацією OpenAPI [4].

Останнім кроком буде опис правил перетворення DTO на Entity та навпаки. Ці правила також реалізовані у вигляді JavaScript об'єкта, де розробник явно вказує які співставляються атрибути DTO та Entity. На основі цієї інформації генеруються Mapper та шар з бізнес-логікою Service, який з’єднаний з API та Repository

Використовуючи цей підхід до проектування системного коду можна реалізувати фреймворк для генерації коду взаємодії між веб застосунком і реляційною базою даних.

Список використаних джерел:

1. Morozova A., Petrova R. Study of the methods and means of data migration between relative and non-relative databases. Information systems in project and program management. Kharkiv, Ukraine, 2023. URL: https://doi.org/10.30837/mmp.2023.229 (дата звернення: 25.03.2024).

2. Mario Casciaro, Luciano Mammino. Node.js Design Patterns: Design and implement production-grade Node.js applications using proven patterns and techniques, 3rd Edition 3rd ed. Edition. Birmingham: Packt Publishing Ltd., 2020.

3. Joe Reis, Matt Housley. Fundamentals of Data Engineering: Plan and Build Robust Data Systems 1st Edition. Sebastopol. O'Reilly Media, Inc. 2022.

4. Ethan Brown. Web Development with Node and Express: Leveraging the JavaScript Stack 2nd Edition. Sebastopol. O'Reilly Media, Inc. 2019.