

А. В. Шаховский, Е. А. Ружицкая
(ГГУ им. Ф. Скорины, Гомель)

РАЗРАБОТКА СЕРВЕРА ДЛЯ ИНТЕРНЕТ-МАГАЗИНА ЭЛЕКТРОНИКИ

Любое построение сервера лучше начать с таких вопросов: «Как правильно построить архитектуру?», «Где хранить данные?», «Что нужно пользователю?».

Примером хорошей архитектуры приложения является архитектура, которая напрямую не зависит от фреймворка, на котором она написана. Любая минимизация зависимостей улучшает её расширяемость. Следование принципам SOLID, GRASP, а также паттернам проектирования приложения приводит к построению долгоживущего приложения, которое легко как расширить, так и исправить.

Сервер, который используется для интернет-магазина электроники был построен на фреймворке NestJs, который следует принципам модульности, что позволяет легко разрабатывать бизнес-логику

Материалы XXIV Республиканской научной конференции студентов и аспирантов «Новые математические методы и компьютерные технологии в проектировании, производстве и научных исследованиях», Гомель, 22–24 марта 2021 г.

для отдельных сущностей, а также быстро и безболезненно находить и исправлять ошибки. Он является достаточно универсальным инструментом, т.к. модульность позволяет ему использовать различного рода существующие библиотеки, разработанные на JavaScript. Поэтому можно легко воспользоваться GraphQL системой, использовать WebSocket протокол или же быть разбит на микросервисы для enterprise-приложения.

Для NestJs существует модуль `@nestjs/swagger`, который позволяет документировать API. Это полезный инструмент, которым могут воспользоваться не только разработчики сервера, но и фронтенд разработчики, чтобы знать, какие API им доступны и тестировщики, которые напрямую через Swagger UI могут протестировать существующий код.

Для хранения данных используется нереляционная база данных MongoDB, которая является отличным и быстрым инструментом. Отсутствие прямых зависимостей и схем таблиц данных добавляют гибкости в хранении данных, а также уменьшают занимаемое пространство и ускоряют процесс получения данных.

При построении приложения были выявлены следующие необходимые сущности: «Пользователь», который расширяется до сущности «Администратор», «Товар», «Заказ».