

ИССЛЕДОВАНИЕ ПРИМЕНЕНИЯ РАЗЛИЧНЫХ ФРЕЙМВОРКОВ В РАЗРАБОТКЕ СЕРВЕРНОЙ ЧАСТИ ВЕБ-ПРИЛОЖЕНИЯ НА ЯЗЫКЕ JAVASCRIPT

Шевелев Александр Михайлович

магистрант, МГТУ «СТАНКИН», РФ, г. Москва

Прежде всего, перед тем как рассмотреть существующие фреймворки, стоит проговорить о программной платформе Node.js. Node.js является средой выполнения кода на JavaScript, на которой можно писать как десктопные приложения так и проекты по созданию кода для микроконтроллеров. Но большую популярность Node.js получил как среда для создания серверных приложений на языке JavaScript [3, с.57].

К некоторым из наиболее популярных фреймворков относят:

1. Express – популярный фреймворк, который обладает простым порогом входа и предоставляет простой интерфейс для создания API. Данная возможность достигается за счет гибкой и простой системы маршрутизации, которая позволяет управлять обработкой HTTP-запросов. Она основана на методах HTTP-запросов (GET, POST, PUT, DELETE и т.д.), а также на шаблонах маршрутов;
2. Одной из ключевых особенностей в фреймворке Fastify является его скорость работы и оптимизация для работы с большими объемами данных, что позволяет достичь большей производительности по сравнению с другими фреймворками. Поддержка интеграции с OpenAPI позволяет автоматически создавать документацию API;
3. Koa.js – это современный веб-фреймворк для Node.js, который создан на базе библиотеки Express, но использует более новые функции языка JavaScript и предлагает улучшенную обработку ошибок;
4. Одними из ключевых особенностей фреймворка NestJS являются модульная структура и язык программирования TypeScript под капотом. Приложение может быть разбито на независимые модули, каждый из которых может быть разработан и протестирован независимо, что упрощает поддержку веб-приложения. NestJS позволяет использовать все достоинства TypeScript (типизация, декораторы, классы), что помогает отлавливать ошибки на этапе компиляции [2, с.73]. Так же NestJS имеет широкую поддержку сообщества разработчиков и множество модулей, доступных из коробки.

Для того чтобы оценить степень распространенности того или иного фреймворка посмотрим какие крупные компании используют его в своей работе.

В своей работе фреймворк Express использовали такие компании как PayPal, MySpace, Trello, IBM, LinkedIn для создания своих веб-приложений и API-серверов [1, с. 24].

Поисковый движок DuckDuckGo, компания NodeSource, специализирующаяся на разработке инструментов и решений для Node.js, компания Siemens используют фреймворк Fastify для обработки запросов и создания своих микросервисов в различных областях программного обеспечения.

Фреймворк NestJS используют данные компании:

- Компания Crossover Health использует NestJS для создания своих API-серверов, которые поддерживают их систему управления медицинскими данными;
- Компания Finastra использует NestJS для создания своих микросервисов и API-серверов для обработки различных финансовых операций;

- Компания Vendasta использует NestJS для создания своих API-серверов, которые поддерживают их систему управления маркетинговыми данными;
- Компания Jibrel Network использует NestJS для создания своей децентрализованной финансовой платформы, которая позволяет пользователям легко и безопасно обмениваться различными валютами;
- Компания Infraspeak использует NestJS для создания своих API-серверов, которые поддерживают их систему управления оборудованием и сервисных работ.

Фреймворк KoaJS использовался в компаниях:

- Alibaba - китайский интернет-магазин Alibaba использует KoaJS в своих проектах, в том числе в своей облачной платформе Alibaba Cloud;
- Mozilla - компания Mozilla использовала KoaJS в своем проекте, в Firefox Send, сервисе обмена файлами.
- Bitbucket - сервис Bitbucket использует KoaJS в своих API-серверах для управления репозиториями и совместной работы над проектами.

Разберем недостатки каждого из фреймворков отдельно.

К недостаткам фреймворка Express относят:

1. Нехватка стандартизации, что может привести к проблемам с совместимостью кода, когда необходимо использовать сторонние библиотеки;
2. Сложность разработки, связанная с тем, что фреймворк имеет меньше встроенных функций, что может потребовать написание дополнительного кода или использования дополнительных сторонних библиотек;
3. Наличие дополнительного кода для обеспечения безопасности приложения;
4. Низкая производительность при обработке большого количества запросов.

К недостаткам фреймворка Fastify относят:

1. Ограниченная экосистема плагинов, связанная с тем, что данный фреймворк является относительно новым;
2. Меньшее сообщество разработчиков и ресурсов для обучения по сравнению с другими фреймворками;
3. Сложный порог входа по сравнению с другими фреймворками;
4. Ограничение функциональности при работе с протоколами, отличными от HTTP (WebSockets);
5. Большой размер приложений, созданных на данном фреймворке, может замедлить скорость загрузки приложения при низкой скорости интернет-соединения.

К недостаткам фреймворка NestJS относят:

1. Сложный порог входа по сравнению с другими фреймворками;
2. Дополнительные ресурсы на обновление пакетов в проекте, так как фреймворк относительно новый и очень часто обновляется;
3. Сложности с масштабированием приложения, написанном на данном фреймворке.

К недостаткам фреймворка KoaJS относят:

1. Уверенное знание разработчиком асинхронного программирования;
2. Сложность отладки из-за асинхронной природы фреймворка;
3. Минимализм (отсутствие валидации данных, обработки ошибок из коробки) что подразумевает более долгую настройку для разработки крупных проектов.

Агрегируем полученную информацию в одну структурирующую таблицу.

Таблица 1.

Структурирующая таблица

	Популярность (Кол-во звезд на гитхабе)	Загрузки в неделю (NPM)	Преимущества	Недостатки
Express	60800	17000000	Легкий порог входа; Большое комьюнити.	Недостаток
Fastify	27200	1030000	Очень высокая производительность; Поддержка валидации данных.	Сложны
NestJs	56300	508000	Возможность создания микросервисной архитектуры; Мощный механизм внедрения зависимостей.	Сложны Сложности с м
KoaJs	34000	870000	Легкий порог входа; Удобный механизм обработки ошибок.	Сложност асинхронного Долгая настро плагино

Таким образом, разработчикам необходимо выбирать подходящий инструмент в зависимости от специфики проекта. В статье были рассмотрены четыре популярных фреймворка: Express, Fastify, NestJS и KoaJS.

В итоговой таблице были приведены преимущества и недостатки каждого фреймворка. В целом, все эти фреймворки предоставляют мощные инструменты для разработки серверной части веб-приложений на языке JavaScript. Выбор конкретного фреймворка должен быть сделан в соответствии с требованиями проекта, опытом разработчиков и возможностями каждого инструмента. Однако стоит следовать одной небольшой рекомендации: Express и KoaJS являются отличным выбором для небольших и средних проектов, Fastify и NestJS же хорошо подходят для крупных и сложных проектов.

Список литературы:

1. Вагин, Д. В. Современные технологии разработки веб-приложений : учебное пособие / Д. В. Вагин, Р. В. Петров. — Новосибирск : НГТУ, 2019. — 52 с.
2. Розенталс, Н. Изучаем Typescript 3 / Н. Розенталс ; перевод с английского Д. А. Беликова. — Москва : ДМК Пресс, 2019. — 608 с.
3. Сухов, К. К. Node.js. Путеводитель по технологии : учебник / К. К. Сухов. — Москва : ДМК Пресс, 2019. — 416 с.