

УДК 0004.42

*Болотіна В. В., асистент кафедри
Житомирський державний технологічний університет*

ОГЛЯД ПОПУЛЯРНИХ JAVASCRIPT ФРЕЙМВОРКІВ

Ми живемо в епоху використання компонент, і це стосується не тільки front-end розробки. Зростає популярність таких інструментів, як Git, що дозволяють без зайвих зусиль ділитися компонентами, використовувати їх повторно і створювати нові додатки швидше, збираючи їх разом і синхронізуючи з іншими проектами. Майбутнє розробки за компонентно-орієнтованими платформами. Популярність використання фреймворків зростає з кожним днем і фундаментальною причиною використання фреймворків є те, що вони допомагають вирішувати завдання синхронізації, призначеного для користувача інтерфейсу і внутрішнього стану програми.

JS-framework – це інструменти для побудови динамічних веб, мобільних, настільних додатків на мові програмування JavaScript.

До найбільш популярних серед front-end розробників фреймворків відносяться:

- React – це ефективна і гнучка декларативна бібліотека JavaScript для збірки UI від команди Facebook. Вона дозволяє без зусиль створювати інтерактивний інтерфейс користувача. У React використовується компонентний підхід. Бібліотека немає контролерів, в'ю, моделей, шаблонів – все є компонент. Компоненти можна перевикористовувати, успадковувати один від одного, об'єднувати. Компонент – це свого роду будівельна одиниця, з якої збирається інтерфейс. Перевагою є можливість порівняння React'ом віртуального DOM'а з реальним, і виконання мінімальних змін для їх синхронізації. Віртуальний DOM вирішує проблему з обробкою подій в різних браузерях, за рахунок цього React надає сумісну модель подій в будь-якому браузері. React дозволяє використовувати будь-який інструмент при розробці, він добре поєднується з іншими фреймворками. Також дану бібліотеку дуже просто освоїти. Дизайн React поєднується з асинхронними серверними архітектурами для адаптації до майбутніх технологій

Таким чином, React варто використовувати, якщо необхідно створити якісний і в найкоротші терміни швидкий, легкий, зручний однострінковий додаток.

- Angular – фреймворк розроблений компанією Google, що являє собою платформу, яка спрощує збірку веб-додатків. У Angular поєднуються декларативні шаблони, впровадження залежності, двостороннє зв'язування даних і кращі практики вирішення проблем розробки.

Ця платформа дозволяє розробляти додатки для веб, мобільних пристроїв і настільних ПК. Angular.js використовується, якщо необхідна оболонка декларативного стилю для збору даних про поведінку користувача через HTML і компоненти. Також даний фреймворк надає просту систему тестування і URL-навігацію. Перевагами даного фреймворку є прив'язка даних між Vue і Model, що дозволяє проводити автоматичне оновлення з обох сторін при зміні даних. Angular.js легко завантажується. Даний фреймворк суттєво зменшує розмір коду, за рахунок повторного використання компонентів. Розширює синтаксис HTML. Також перевагами є можливість написання додатка без jQuery і без безпосереднього управління DOM'ом. До недоліків можна віднести високий поріг вивчення, за рахунок складної документації. А також неможливість вказувати переходи станів UI, зі зміною моделі.

У Angular.js логіка описується всередині HTML, що призводить до високорівневої абстракції.

- Backbone – простий MVP-фреймворк, який вміщається в один JS файл. Даний фреймворк включає в себе все те, що необхідно для побудови додатків без будь-яких складнощів, наприклад, він підтримує RESTfull JSON інтерфейс. Перевагою є невеликий набір блоків для побудови додатків (моделі, уявлення, колекції). Функціональність фреймворка дозволяє швидко в ньому розібратися.

До недоліків Backbone фреймворку можна віднести те, що він не в змозі вирішити всі проблеми, з якими зіткнеться розробник при розробці програми, тому доведеться поверх всього вигадувати своє рішення. При використанні Backbone буде не вистачати способів створення вторинних властивостей, їх зв'язування, а також способу рендеру уявлень всередині елемента.

За рахунок мінімалістичної концепції Backbone має невеликий поріг входження, але його вузька спеціалізація призводить до того, що розробникам доводиться вигадувати багато своїх рішень поверх наявного фреймворка. Отже, основними перевагами використання JavaScript фреймворків є, те що за їх допомогою можна легко реалізувати SPA (Single Page Application), також при розробці з використанням фреймворків розробник дотримується чіткої структури коду, його помітно стає менше і він більш зрозуміліший, що впливає на швидкість розробки та підтримки при усуненні помилок в коді програми, за рахунок чого полегшується робота над додатком багатьом розробникам одночасно, а також посилюється можливість швидко створити мобільний або настільний додаток.