Данюк В.М., магістрант,
Кравченко С.М., старший викладач,
Житомирський державний технологічний університет

ПРО ФРЕЙМВОРК VUE.JS

Vue – це прогресивний фреймворк для створення користувацьких інтерфейсів. На відміну від фреймворків-монолітів, Vue створено придатним для поступового впровадження. Його ядро в першу чергу вирішує завдання рівня представлення (view), що спрощує інтеграцію з іншими бібліотеками, та існуючими проектами. З іншого боку, Vue повністю підходить і для створення складних односторонніх додатків (SPA, Single-Page Applications), якщо використовувати його спільно з сучасними інструментами та додатковими бібліотеками.

Реактивне програмування – це парадигма програмування, побудована на потоках даних і розповсюдженні змін. Це означає, що у мовах програмування має бути можливість легко виразити статичні чи динамічні потоки даних, а реалізована модель виконання буде автоматично розсилати зміни через потік даних.

Реактивне програмування спочатку пропонувалося як засіб простого створення інтерфейсів користувача, анімацій у системах реального часу, але стало загальною парадигмою програмування.

Бібліотека Vue.js з кожним днем набирає все більше і більше популярності, хоча не всі розуміють чому саме Vue, і в яких саме випадках останню можна і потрібно використовувати.

Технічно, Vue.js визначена як ViewModel шар шаблону MVVM. Вона з'єднує модель і представлення у двосторонньому зв’язуванні даних. Поточні DOM-зміни і відформатований вивід абстрагуються у директивах і фільтрах.

Мета надати переваги швидких зв'язувань даних і складних уявлень компонентів з API, як прості, так і зрозумілі. Бібліотека не є повномасштабним фреймворком, вона є всього лише рівнем уявлення. Можна використовувати її як окремо, для швидкого прототипування, або змішувати і поєднувати з іншими бібліотеками для кастомізації інтерфейсу користувача.

На Vue.js значно вплинули Angular, Knockout, React і Rivets. Незважаючи на схожість, Vue.js може запропонувати цінну альтернативу цим існуючим бібліотекам, в пошуках золотого перетину між простотою і функціональністю.

У чому різниця між Vue.js і React.js?

1. React і Vue.js мають невелику схожість в тому, що обидва підтримують реактивне і компоноване уявлення компонентів. Проте, внутрішня реалізація в корені відрізняється. React заснований на віртуальному DOM - в пам'яті представлено те, як DOM виглядає насправді. Дані в React в значній мірі є незмінними, і DOM-зміни розраховуються на основі визначення відмінностей. У Vue.js, навпаки, дані змінювані за замовчуванням, і змінюються через події. Замість віртуального DOM, Vue.js працює безпосередньо з DOM як шаблон, зберігаючи посилання на існуючі вузли для зв'язування.

2. Підхід з віртуальним DOM забезпечує функціональний спосіб описати уявлення в будь-який момент, що дуже приємно. Тому що не використовується шаблон спостерігача і не перемальовується увесь додаток при кожному оновленні, уявлення щодо визначення гарантує бути синхронним з даними, що також відкриває можливості для ізоморфних JavaScript-додатків.

3. Проблема при використанні React в тому, що логіка і уявлення тісно переплітаються. Для деяких розробників це плюс, але для дизайнерів/розробників гібридів, представляти шаблон наочно набагато простіше в дизайні і CSS. JSX зливаючись з Javascript-логікою ламає ту візуальну модель, яка необхідна для відображення коду в дизайні. Vue.js, на відміну, розплачується своїми легкими директивами, але завжди є візуально представлений шаблон і інкапсульована логіка в директивах і фільтрах.

4. Ще одна проблема React полягає в тому, що всі оновлення делеговані на віртуальний DOM - це трохи складно, коли насправді потрібно контролювати DOM самостійно (хоча теоретично можна, потрібно по суті працювати на противагу бібліотеці, коли це необхідно). Для додатків, яким потрібна складно-синхронізована анімація за часом - це може стати дуже дратівливим обмеженням. На цьому фронті Vue.js є більш гнучким.

Проаналізувавши це все, можна зробити висновок, що Vue дуже легкий у використанні і швидший, ніж інші прогресивні фреймворки. Бібліотека має свої вузькі місця і не тривіальні рішення, але однозначно Vue заслуговує увагу при розробці призначених для користувача інтерфейсів.