

Порівняльний аналіз популярних JavaScript - фреймворків та бібліотек для front-end розробки

Олексій Танянський
Студент кафедри Інформатики
Харківський національний університет
радіоелектроніки
Харків, Україна
oleksii.tanianskyi@nure.ua

Діана Руденко
Доцент кафедри Інформатики
Харківський національний університет
радіоелектроніки
Харків, Україна
diana.rudenko@nure.ua

A comparative analysis of popular JavaScript-frames and libraries for front-end development

Oleksii Tanianskyi
Student of Informatics Department
Kharkiv National University
of Radio Electronics
Kharkiv, Ukraine
oleksii.tanianskyi@nure.ua

Diana Rudenko
Associate Professor of Informatics Department
Kharkiv National University
of Radio Electronics
Kharkiv, Ukraine
diana.rudenko@nure.ua

Анотація—У даній роботі проводиться порівняльний аналіз сучасних JavaScript фреймворків, таких як Angular, React, Vue, Backbone, Ember, Knockout. Розглядаються можливості та характеристики, властиві даним фреймворкам, та проводиться їх порівняльний аналіз.

Abstract—This article describes a comparative analysis of tool of modern JavaScript frameworks like Angular, React and Vue.js, Backbone, Ember, Knockout. There are described all possibilities and characteristics of frameworks.

Ключові слова—фреймворк; бібліотека; інструмент; компонент; реалізація.

Keywords—framing; library; tool; component; implementation.

Для різних людей слова «фреймворк», «бібліотека» і «інструмент» набувають різного значення в залежності від контексту.

Бібліотека - це структурований набір корисного функціоналу. Стандартна бібліотека повинна містити функції для роботи з рядками, датами, DOM-елементами, подіями, cookie, анімацією, запитами, і багатьом іншим. Кожна функція повертає значення до додатку, що зробив запит, яке може в подальшому використовувати їх в залежності від логіки розробника. Бібліотека зазвичай скорочує час розробки приблизно на 20%, дозволяючи програмісту не турбуватися про дрібниці.

Негативні сторони:

- баг в реалізації бібліотеки може викликати складності в його знаходженні та способи їх усунення;
- немає гарантії, що команда розробників оперативно випустить патч;
- патч може змінити API, що спричинить значні зміни у коді програмного продукту.

Фреймворк - це каркас додатка. Він зобов'язує розробника вибудувати архітектуру програми відповідно до деякої логіки. Будь-який фреймворк зазвичай надає функціонал на кшталт подій, сховищ і зв'язування даних. Фреймворк знаходиться на більш високому рівні абстракції в порівнянні з бібліотекою і дозволяє без складнощів вибудувати близько 80% розроблюваного застосунку. Програмний фреймворк — це готовий до використання комплекс програмних рішень, включаючи дизайн, логіку та базову функціональність системи або підсистеми. Відповідно, програмний фреймворк може містити в собі також допоміжні програми, деякі бібліотеки коду, скрипти та загалом все, що полегшує створення та поєднання різних компонентів великого програмного забезпечення чи швидке створення готового і не обов'язково об'ємного програмного продукту. Побудова кінцевого продукту відбувається, зазвичай, на базі єдиного API.



Недоліки:

- останні 20% можуть викликати чималі труднощі через обмеження, що накладаються фреймворком;
- оновлення фреймворку може бути пов'язане зі складнощами, а часом і зовсім неможливо;
- основний код і концепції рідко задовольняють розробників в своєму первозданному вигляді. Вони завжди знайдуть «кращий» спосіб зробити що-небудь.

Інструмент - це допоміжний засіб розробки, але він не є невід'ємною частиною проекту. Інструменти включають в себе системи збирання, компілятори, транспайлери, механізми розгортання та інше. Інструменти покликані полегшувати процес розробки. Наприклад, багато хто воліє Sass чистому CSS, тому що він надає можливість використовувати цикли, функції, локальні змінні і багато іншого. Браузери не розуміють Sass / SCSS синтаксис, так що код переводиться в CSS.

jQuery - найпопулярніша js-бібліотека всіх часів. Вона зробила революцію в програмуванні клієнтської частини веб-додатків, ввівши селектори CSS для доступу до вузлів DOM-дерева, обробники подій, анімації та аjax-запити. Останнім часом jQuery втрачає колишню популярність, але як і раніше залишається життєздатним варіантом для проектів, які потребують невеликого js-функціоналу.

Переваги:

- малий розмір дистрибутиву;
- низький поріг входження, вичерпна документація в інтернеті;
- лаконічний синтаксис;
- легко розширюваний.

Недоліки:

- уповільнює роботу програми;
- може спричинити проблеми сумісності з браузером;
- співтовариство розробників протестує проти його широкого використання.

React - перлина веб-програмування: навіть деякі завзяті послідовники Angular зробили свій вибір на користь React. React зміг те, з чим не впорався Angular – він дозволяє розробляти фронтенд, не занурюючись у нетрі й складнощі фреймворку. React-JavaScript-бібліотека, створена Facebook, і основна її мета – забезпечити комфортну роботу з Shadow DOM. Для цього використовується Virtual DOM, модель, яка при кожній зміні генерує список операцій, що необхідно провести над DOM, для приведення його до нового стану.

Переваги:

- компактність, ефективність, продуктивність і гнучкість;

- легка модель компонентів;
- гарна документація і велика кількість онлайн-ресурсів;
- можливість рендерингу на стороні сервера;
- зростаюча популярність.

Недоліки:

- нові концепції і синтаксис, які доведеться вивчити;
- необхідні системи збирання;
- може вимагати сторонніх інструментів і бібліотек;
- може бути несумісний з кодом та іншими бібліотеками, що модифікують DOM-дерево.

AngularJS - фреймворк від Google, що став дуже популярним останнім часом, ідеальний для розробників, які з великою увагою ставляться до останніх новинок галузі. Angular являє собою набір сучасних функцій для розробки і проектування, що забезпечують високу швидкість розробки. Компанія Google навіть створила спеціальний компонент Angular Material, що дозволяє легко реалізувати справжній Material Design за допомогою Angular.

Переваги:

- популярний фреймворк, підтримуваний кількома великими компаніями;
- універсальне рішення для створення сучасних веб-додатків;
- частина «стандартного» стека MEAN (MongoDB, ExpressJS, AngularJS, NodeJS), доступні безліч статей і туторіали.

Недоліки:

- складніше в освоєнні в порівнянні з деякими альтернативами;
- велика база коду;
- несумісність з Angular 2.x +;
- незважаючи на те, що є проектом Google, самим Google не використовується.

Angular 2+ - написаний на TypeScript front-end фреймворк з відкритим кодом, який розробляється під керівництвом Angular Team у компанії Google, а також спільнотою приватних розробників та корпорацій. Angular — це AngularJS, який переосмислили та який був повністю переписаний тією ж командою розробників.

Переваги:

- універсальне рішення для створення сучасних веб-додатків;
- все ще частина стека MEAN;
- TypeScript надає деякі переваги для людей, знайомих з об'єктно-орієнтованими мовами на зразок C# і Java.



Недоліки:

- складний у вивченні для новачків;
- несумісність з Angular 1.x.

Vue.js - це простий у використанні й водночас дуже потужний фреймворк. Початківці належним чином оцінять документацію. Vue бере все краще від інших фреймворків і об'єднує в один зручний інструмент. Так само, як і React, Vue працює з однофайловими компонентами і односпрямованими зв'язками між ними, але працювати з ним простіше завдяки CLI-інструменту vue-cli. Всередині Vue-компонентів можна використати препроцесори (Jade, Stylus та ін.).

Переваги:

- глядацькі симпатії і зростаюча популярність;
- простий в освоєнні з нуля;
- мало залежностей і хороша продуктивність.

Недоліки:

- молодий проект - високі ризики;
- менше ресурсів, ніж у альтернатив.

Backbone побудований на ідеї того, що вся робота із сервером повинна здійснюватися через API, мінімізуючи кількість коду, який потрібний для досягнення такої ж широкої функціональності, як у складних веб-фреймворків.

Backbone – один з найпопулярніших фреймворків для веб-розробки, в основному завдяки двом причинам: принцип застосування модулів дуже простий для розуміння, а отже, простим є і навчання використанню самого фреймворка.

Переваги:

- компактний, легкий і простий у вивченні;
- не додає додаткової логіки в HTML;
- прекрасна документація;
- використаний в розробці багатьох додатків включаючи Trello, WordPress, LinkedIn і Groupon.

Недоліки:

- низький рівень абстракції в порівнянні з альтернативами (наприклад, AngularJS);
- вимагає додаткових компонентів для реалізації фич начебто зв'язування даних;
- більш сучасні фреймворки відмовилися від моделі MVC.

Ember – сучасний JavaScript-фреймворк для амбітних розробників і проектів. Будучи неймовірно потужним, він дуже простий в освоєнні.

Однією з речей, за які люблять Ember, є шаблонізатор Handlebars, який, за словами розробників, набагато

елегантніший директив Angular. У цілому Ember майже все робить за програміста, тому Вам залишиться тільки створювати свою бізнес-логіку і насолоджуватися процесом.

Переваги:

- єдине рішення для клієнтських додатків;
 - дозволяє розробникам бути продуктивними (використовує jQuery);
 - хороша зворотна сумісність;
 - схвалюється сучасними стандартами веб-розробки.
- Недоліки:
- великий дистрибутив;
 - вважається монолітним в порівнянні з іншими фреймворками, які базуються на модульній моделі;
 - непростий у вивченні.

Knockout – це відкритий фреймворк, який працює з моделлю MVVM. Він розроблений на чистому JavaScript без зовнішніх залежностей. Knockout легкий і зручний у використанні й особливо ефективний у зв'язуванні даних. Чудово підійде для створення односторінкових додатків.

Переваги:

- легкий і не має залежностей;
- прекрасна підтримка браузерів аж до IE6;
- хороша документація.

Недоліки:

- вносить зайву складність в великі проекти;
- розробка сповільнилася;
- популярність сходить нанівець.

Отже, кожен із розглянутих засобів розробки є досить потужним та має свої недоліки і переваги. Але обираючи фреймворк або бібліотеку, потрібно спиратися не тільки на переваги програмного продукту, а виходити з потреб конкретного проекту. В зв'язку з тим, що js- фреймворки схожі за синтаксисом, не виникає проблем при переході з одного фреймворка на інший.

ЛІТЕРАТУРА REFERENCES

- [1] Блог: Матеріал з Вікіпедії — вільної енциклопедії: [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/%D0%91%D0%BB%D0%BE%D0%B3>
- [2] AngularJS official documentation. [Електронний ресурс]. - Режим доступу: <https://angularjs.org/>
- [3] ReactJS official documentation: [Електронний ресурс]. - Режим доступу: <https://reactjs.org/docs/getting-started.html>
- [4] React – A JavaScript library for building user interfaces [Електронний ресурс]. – Режим доступу: <https://reactjs.org/>
- [5] Guillaume Chau . Vue.js 2 Web Development Projects. Birmingham, 2017, - p.382.

