

ПОРІВНЯННЯ АРХІТЕКТУР SPA ТА MPA В РЕАЛІЗАЦІЇ ВЕБ ЗАСТОСУНКІВ

Зі зростання популярності веб технологій і їхнього поширення у найрізноманітніших прикладних областях, архітектура розроблюваних веб застосунків стає все більш складною. Веб пройшов шлях від набору простих статичних сторінок, злінкованих між собою гіперпосиланнями, до складних хмарних застосунків, функціональні можливості яких можуть змагатися з десктопними програмами, а інколи і перевершувати їх. Перевагами таких веб застосунків стає їхня мобільність, доступ з будь якого місця та будь якого пристрою, зберігання даних та виконання обчислень у хмарі.

Для забезпечення сталої роботи складних веб застосунків необхідно використовувати архітектурний підхід, що надасть найліпшу швидкодію та продуктивність для кінцевого користувача. На сьогодні існує дві архітектури веб застосунків: односторінкові застосунки (Single page application або SPA) та багатосторінкові застосунки (Multi Page Application або MPA). Кожен з цих варіантів має свої особливості застосування, на які треба зважати на фазі проектування програмного забезпечення.

Розглянемо різницю між цими архітектурами та випадки, в яких кожна з архітектур дає свої переваги.

SPA дозволяє будувати застосунки, які за своєю поведінкою схожі на десктопні програми. Такі застосунки не вимагають перезавантаження сторінки в ході роботи. SPA дуже швидко реагує на дії користувача і є настільки інтерактивним, що у користувача складається враження роботи з десктопною програмою, він не відчуває затримок, необхідних для завантаження нових сторінок. Типовий SPA застосунок - це веб сторінка, що підвантажує дані та оновлює свій вміст без перезавантаження за допомогою мови JavaScript. Для побудови SPA використовують сучасні JavaScript фреймворки, такі як AngularJS, ReactJS, тощо.

Перевагами SPA є:

- розділення на front-end та back-end розробку;
- висока швидкодія завдяки тому, що більшість ресурсів (HTML, CSS, скрипти) завантажуються лише один раз при ініціалізації застосунку, після чого завантажуються лише дані;
- швидка розробка завдяки сучасним фреймворкам, які спрощують створення архітектури проекту і надають велику кількість готових елементів

для роботи;

- можливість легкого створення мобільного застосунку з тим же кодом на стороні серверу;
- застосунки можуть кешувати дані на локальних носіях та працювати в режимі оффлайн.

До недоліків SPA можна віднести наступне:

- складність SEO-оптимізації;
- довготривале початкове завантаження застосунку;
- потребують ввімкненого JavaScript в браузері користувача;
- менш захищені адже за допомогою міжсайтового скриптингу (XSS), злоумисники можуть впроваджувати додаткові скрипти на стороні користувача.

Багатосторінкові застосунки мають більш класичну архітектуру, в якій в результаті запитів на сервер повністю оновлюється та рендериться весь вміст сторінки, навіть у тому випадку коли нових даних не багато. Таким чином витрачається час на відображення одних і тих самих елементів, що впливає на швидкість та продуктивність. Зазвичай така архітектура використовується при необхідності відображення великої кількості інформації.

Можна виділити наступні основні переваги МРА:

- спрощення SEO оптимізації, адже архітектура МРА дозволяє легко оптимізувати кожен сторінку під пошукові системи;
- легка розробка, необхідний менший стек технологій, тобто розробка такого застосунку може бути дешевшою;
- більш звичний інтерфейс для кінцевого користувача.

В той же час у архітектури є і недоліки:

- довший час і складність розробки;
- зв'язані між собою front-end та back-end, складно розвивати їх паралельно;
- додаткова розробка для отримання можливості завантаження даних без перевантаження сторінки з використанням технології аїах;
- багаторівневий дизайн інтерфейсу користувача.

Отже розглянувши особливості кожної архітектури, можна зробити висновок, що кожна з них підходить для певного типу проєктів. Так SPA краще підходить для проєктів, в яких пошукова оптимізація не є визначальною, наприклад, SaaS застосунки або соціальні мережі. Для створення ж e-commerce застосунків або корпоративних сайтів краще підійде архітектура МРА.