УДК 004.424

В.В. Яцишин канд. техн. наук, доц., В.О.Найда
Тернопільський національний технічний університет імені Івана Пулюя, Україна

SINGLE PAGE APPLICATION ЯК ТЕХНОЛОГІЯ FRONT END РОЗРОБКИ

V.V. Yatcyshyn Ph.D., Assoc. Prof., V.O. Naida

SINGLE PAGE APPLICATION AS A FRONT END TECHNOLOGY DEVELOPMENT

Сучасні технології розробки програмного забезпечення характеризуються великою кількістю інструментальних засобів і методів проектування, які, в залежності від сфери застосування, формують окремі напрями інженерії програмного забезпечення. Зокрема, на сьогодні сформовано напрям – web-інженерії (web-engineering). Даний напрям характеризується високою технологічністю та розвинутими інструментами підтримки і супроводу. Оскільки, більшість програмних продуктів під WEB використовують на концептуальному рівні клієнт-серверну архітектуру, то це призвело до розділення логіки роботи web-орієнтованих систем на дві частини – front end та back end. В програмній інженерії терміни «front end» та «back end» розрізняють за принципом розділення відповідальності між рівнем представлення та рівнем доступу до даних відповідно. Front end відповідає за взаємодію користувача з інтерфейсом web-системи, back end – за обробку подій на стороні сервера. Сучасний WEB дедалі частіше використовує технології Single Page Application (SPA) для front end розробки. SPA – це збірна назва набору технологій, що дозволяють реалізувати WEB-додаток, який виконується WEB-браузером як одна WEB-сторінка. Прикладом технології SPA є реалізований сервіс Gmail від Google. З точки зору користувача, дана технологія забезпечує високу швидкість відгуку на дії в інтерфейсі. При цьому не потрібно повного або навіть часткового перезавантаження WEB-сторінки з сервера – всі візуальні елементи конструюються прямо в браузері за допомогою технології JavaScript, шляхом маніпуляцій з DOM- структурою документа.

Таким чином, WEB-додатки стають дуже схожі на звичайні програми для робочих станцій, що завантажують інформацію з мережі Інтернет. Середовищем виконання для них є не операційна система, а браузер, який в результаті змушений нести на собі все навантаження, пов'язане з виконанням стороннього коду.

Центральне місце SPA-архітектури займає відображення (View) – те, що бачить і з чим взаємодіє користувач. Результатом роботи відображення є звичайний HTML, що відображається браузером. На відміну від «перехідних» «WEB 2.0»-додатків, що активно працюють з DOM-структурою документа, наприклад, за допомогою jQuery або underscore, SPA-додаток використовує DOM тільки для запису змін, але не для читання, тобто не для зберігання даних . Для зберігання даних тепер використовується ще один компонент SPA-архітектури – модель (Model).

Модель представляє собою сукупність даних, функцій для маніпуляції з даними і подіями. Всі дані моделі повністю зберігаються в пам'яті. Для того, щоб дані, що знаходяться в моделі, і дані, які відображаються представленням, зберігали цілісність, відображення підписується на події моделі, відстежуючи таким чином зміни даних в моделі. У свою чергу, модель також реагує на повідомлення відображення і забезпечує нерозривний зв'язок WEB-додатка з сервером, виконуючи запити для отримання або відправки даних (зокрема із застосуванням методології REST).

Відображення – це найважливіша і найбільш складна частина сучасних SPA. Також відображення оновлює отриманий HTML при зміні моделі і навпаки –

повідомляє модель про дії користувача з представленням. При натисненні клавіші мишки чи введенні з клавіатури модель може виконати маніпуляції з даними, а після цього повідомити відображення про зміну даних для того, щоб відображення оновило або згенерувало новий HTML.

Робота класичного WEB-додатка (або WEB-сайту) повністю будується поверх кешування даних: на сервері, на проксі-сервері і на клієнті. Якщо дані і стан додатку оновлюються дуже часто, перевага від використання кешування практично нівелюється.

Односторінкові додатки стають все більш популярнішими за мультисторінкові (класичні) додатки тому є актуальними дослідження структури, поведінки, швидкодії односторінкових веб – додатків для того, щоб підвищувати їх якість та надійність.

Односторінкові веб - додатки слід використовувати тоді, коли необхідно розробити кросплатформний додаток. Як показує практика, вони добре функціонують як на персональних ПК, так і мобільних пристроях. На рисунку 1 зображено структуру односторінкового та мультисорінкового додатка.

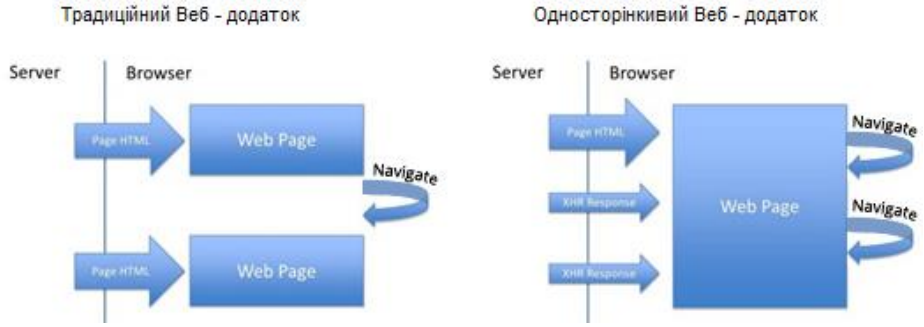

Рисунок 1. Структура веб – додатка

Порівнявши вище наведені структури, можна зробити висновок, що справді односторінкові додатки є високопродуктивними завдяки тому, що не навантажують серверну частину частими запитами як це роблять класичні додатки. Односторінкові додатки завантажують усі дані при зверненні до сторінки. При виконанні певних дій користувачем, SPA завантажує дані по мірі необхідності без повного перезавантаження. Це є однією із основних переваг на яку слід звернути увагу при виборі типу веб – додатку.

З розвитком сучасного WEB, виникає необхідність побудови додатків на основі модульного підходу. Модулі в середині веб – додатку повинні бути малозв'язними для того, щоб можна було в будь – який момент масштабувати програму. Найкращим вирішенням даної проблеми є використання саме односторінкових додатків через те, що вони будуються на основі модулів, які незалежні один від одного і можуть бути повторно використані. Клієнтська та серверна частини в SPA є повністю незалежними, що дає змогу ізолювати їх один від одного. Зміни у серверній частині після заміни ніяким чином не вплине на клієнтську частину.

На даному етапі розвитку WEB, односторінкові додатки швидко витісняють класичні додатки і є великим внеском у розробку масштабних, швидких, динамічних веб – систем. Однак технології реалізації front end частини web-додатків потребують додаткового дослідження в контексті вибору кращих альтернатив для реалізації систем під конкретні вимоги користувачів чи замовників продукту.