MINISTERUL EDUCAŢIEI ŞI CERCETĂRII

Universitatea Tehnică a Moldovei

Facultatea Calculatoare Informatică şi Microelectronică

Departamentul Ingineria, Software și Automatică

Admis la susţinere

Şef departament: Fiodorov I. dr., conf.univ..

_____ _____

„ __”_____ 2022

ANALIZA ȘI COMPARAREA PERFORMANȚEI MANIPULĂRII DOM CU DIFERITE FRAMEWORK-URI JAVASCRIPT

Proiect de master

Student: _____ Timofeev Maxim

Conducător: _____ conf. univ. Romanenko A.

Coordonator: _____ asistent univ. Cojocaru S.

Chişinău 2022

АННОТАЦИЯ

Веб-сайты 2020 года часто являются многофункциональными и интерактивными приложениями. JavaScript - популярный язык программирования для веб, с множеством фреймворков. Общим знаменателем для интерактивных веб-приложений является необходимость в эффективных методах манипулирования объектной моделью документа для обеспечения надежного пользовательского опыта

В этом исследовании сравниваются чистый JavaScript и JavaScript-фреймворки Angular, React и Vue.js в производительности DOM, методологии манипулирования DOM и размера приложения.

Что касается методологии манипулирования DOM, существует явное различие между чистым JavaScript и выбранными фреймворками. В чистом JavaScript DOM манипулирование осуществляется путем прямого взаимодействия с DOM API. При использовании выбранных фреймворков фактическое взаимодействие с интерфейсом DOM абстрагируется от разработчика и обрабатывается фреймворком. В то время как React и Vue.js реализовали Virtual DOM для оптимизации взаимодействия с DOM, Angular использует инкрементный DOM.

Чистый JavaScript имеет лучшую производительность во всех тестах и наименьший размер приложения. Среди фреймворков React имеет лучшую производительность, Angular показал близкие результаты с React почти во всех тестах, а Vue.js был немного медленнее. Vue.js был немного медленнее в большинстве тестов. Почти во всех тестах приложения работали лучше в Google Chrome.

REZUMAT

Site-urile web 2020 sunt adesea aplicații interactive și bogate în funcții. JavaScript este un limbaj de programare popular pentru web, cu multe cadre de lucru. Numitorul comun al aplicațiilor web interactive este nevoia de metode eficiente de manipulare a modelului de obiect document pentru a oferi o experiență robustă utilizatorului.

Acest studiu compară JavaScript pur și cadrele JavaScript Angular, React și Vue.js în ceea ce privește performanța DOM, metodologia de manipulare a DOM și dimensiunea aplicației.

În ceea ce privește metodologia de manipulare a DOM, există o diferență clară între JavaScript pur și cadrele selectate. În JavaScript pur, manipularea DOM se face prin interacțiunea directă cu API-ul DOM. În cazul cadrelor selectate, interacțiunea efectivă cu interfața DOM este abstractizată de la dezvoltator și este gestionată de cadru. În timp ce React și Vue.js au implementat Virtual DOM pentru a optimiza interacțiunea DOM, Angular utilizează DOM incremental.

Pure JavaScript are cea mai bună performanță în toate testele și cea mai mică dimensiune a aplicației. Dintre framework-uri, React are cea mai bună performanță, Angular s-a comportat aproape de React în aproape toate testele, iar Vue.js a fost puțin mai lent. Vue.js a fost puțin mai lent în majoritatea testelor. Aplicațiile au funcționat mai bine în Google Chrome în aproape toate testele.

ABSTRACT

Websites of 2020 are often feature rich and highly interactive applications. JavaScript is a popular programming language for the web, with many frameworks available. A common denominator for highly interactive web applications is the need for efficient methods of manipulating the Document Object Model to enable a solid user experience.

This study compares Vanilla JavaScript and the JavaScript frameworks Angular, React and Vue.js in regards to DOM performance, DOM manipulation methodology and application size.

In regards to DOM manipulation methodology, there is a distinct difference between Vanilla JavaScript and the selected frameworks. In Vanilla JavaScript DOM manipulation is handled by direct interaction with the DOM interface. When using the selected frameworks the actual interaction with the DOM interface is abstracted away from the developer and handled by the framework. While React and Vue.js both have implemented a Virtual DOM to optimize DOM interactions, Angular has implemented Incremental DOM. Vanilla JavaScript had the best DOM performance in all tests and the smallest application size. Amongst the frameworks React had the best DOM performance, Angular performed close to React in nearly all test, and Vue.js was slightly slower in.

Содержание

ВВЕДЕНИЕ

С момента своего появления в 1995 году JavaScript превратился в один из основных языков программирования для Web. С ростом использования и популярности JavaScript, язык получил распространение в сообществах разработчиков, что привело к появлению обилия библиотек и фреймворков с открытым исходным кодом, которые разрабатывались на протяжении многих лет. Темпы эволюции высоки в текущем состоянии JavaScript, где существующие фреймворки быстро развиваются, и в то же время новые фреймворки появляются каждый год.

В течение многих лет средний размер веб-страницы увеличивался из года в год. Размер увеличивается по мере того, как веб-сайты становятся более функциональными и интерактивными. В то же время, производительность является ключевым фактором для обеспечения удовлетворительного пользовательского опыта, как в отношении быстрого начального времени загрузки страницы, а также своевременного реагирования на действия пользователя с как можно меньшим время ожидания, насколько это возможно. Для веб-сайта метрика времени загрузки страницы является критически важной по нескольким причинам. Оно оказывает непосредственное влияние на пользовательский опыт, а статистика Google статистика показывает, что 53% пользователей, заходящих на сайт с мобильного устройства, покидают страницу, которая загружается дольше трех секунд. Google признал, что скорость сайта является одним из параметров их алгоритмов поискового ранжирования. Значительная частью времени загрузки страницы является время рендеринга страницы, измеряющее количество времени, которое занимает рендеринг страницы в интернет-браузере.

Использование Интернета стремительно растет с начала 1990-х годов, и сейчас более половины населения планеты работает в сети. В то же время влияние Интернета на окружающую среду увеличилось, и сейчас Интернет находится на одном уровне с авиационной промышленностью, когда речь идет о воздействии CO2 на окружающую среду. Ожидается, что доля мирового потребления электроэнергии, которая может быть отнесена к коммуникационным технологиям, достигнет 21% к 2030 году.

С быстрой эволюцией, происходящей в экосистеме JavaScript, front-end фреймворки имеют тенденцию быстро устаревать.

БИБЛИОГРАФИЯ

[1] I. Allen. The brutal lifecycle of javascript frameworks. Disponibil: https://stackoverflow.blog/2018/01/11/brutal-lifecycle-javascript-frameworks/

[2] angular.io. Angular cli. Disponinil: https://cli.angular.io/

[3] chromium.org. Chromedriver - webdriver for chrome. Disponibil: https://chromedriver. chromium.org/downloads

[4] S. Fluin. Version 9 of angular now available - project ivy has arrived. Disponibil: https://blog.angular.io/version-9-of-angular-now-available-project-ivy-has-arrived-23c97b63cfa3

[5] Mozilla Foundation. Mozilla geckodriver. Disponibil: https://github.com/mozilla/geckodriver/releases

[6] Mozilla Foundation. Mdn web docs. Disponibil: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference

[7] A. Goel. 10 best javascript frameworks to use in 2020. Disponibil: https://hackr.io/blog/best-javascript-frameworks

[8] S. Greif. The 12 things you need to consider when evaluating any new javascript library. Disponibil: https://www.freecodecamp.org/newsthe-12-things-you-need-to-consider-when-evaluating-any-new-javascript-library-3

[9] IEEE. Ieee standard glossary of software engineering terminology. IEEE Std 610.12-1990, pages 1–84, 1990.

[10] ECMA International. Standard ecma-262. Disponibil: https://www.ecma-international.org/publications/standards/Ecma-262.html

[11] S. Peyrott. React virtual dom vs incrementaldom vs ember's glimmer. Disponibil: https://auth0.com/blog/face-off-virtual-dom-vs-incremental-dom-vs-glimmer

[12] React. React github page. Disponibil: https://github.com/facebook/react

[13] selenium.dev. Selenium official website. Disponibil: https://www.selenium.dev/