

Міністерство освіти і науки України
Запорізький національний технічний університет

кафедра програмних засобів

РЕФЕРАТ

з дисципліни «Якість програмного забезпечення» на тему:

"Тестування навантаження веб-додатків. Fiddler"

Виконала:

студентка групи КНТ-415

В.С. Хохлова

Прийняла:

Г.В. Табунщик

2018

ЗМІСТ

1. АРХІТЕКТУРА ВЕБ-ДОДАТКІВ.....	3
2. ТЕСТУВАННЯ НАВАНТАЖЕННЯ.....	5
3. FIDDLER.....	6
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	9

1. АРХІТЕКТУРА ВЕБ-ДОДАТКІВ

Почнемо з основ і визначимося, що саме розуміється під поняттям «веб-додаток» і які нюанси в реалізації цих програм додають роботу тестувальникам.

Веб-додаток - це клієнт-серверний додаток, в якому клієнтом виступає браузер, а сервером - веб-сервер. Основна частина програми, як правило, знаходиться на стороні веб-сервера, який обробляє отримані запити відповідно до бізнес-логіки продукту і формує відповідь, що відправляється користувачеві. На цьому етапі в роботу включається браузер, саме він перетворює отриману відповідь від сервера в графічний інтерфейс, зрозумілий користувачеві.

Отже, першою і однією з ключових особливостей веб-додатків є їх архітектура (рис. 1.1). Розглянемо це питання, так як воно представляє особливу цінність для тестування.

Веб-додаток представлено наступними складовими («сторонами»):

1. Клієнт.

Як правило, клієнт - це браузер. У класичній ситуації (коли роль клієнта виконує браузер) для того, щоб користувач побачив графічний інтерфейс програми у вікні браузера, останній повинен обробити отриману відповідь веб-сервера, в якій буде міститися інформація, реалізована із застосуванням HTML, CSS, JS (найбільш використовувані технології). Саме ці технології «дають зрозуміти» браузеру, як саме необхідно «зобразити» все, що він отримав у відповіді.

2. Сервер.

Веб-сервер - це сервер, що приймає HTTP-запити від клієнтів і видає їм HTTP-відповіді. Щоб уникнути можливої плутанини, відзначимо, що веб-сервером називають як програмне забезпечення, яке виконує функції веб-сервера, так і безпосередньо комп'ютер, на якому це програмне забезпечення працює. Найбільш поширеними видами ПО веб-серверів є Apache, IIS і

NGINX. На веб-сервері функціонує тестоване додаток, яке може бути реалізовано з застосуванням найрізноманітніших мов програмування: PHP, Python, Ruby, Java, Perl та ін.

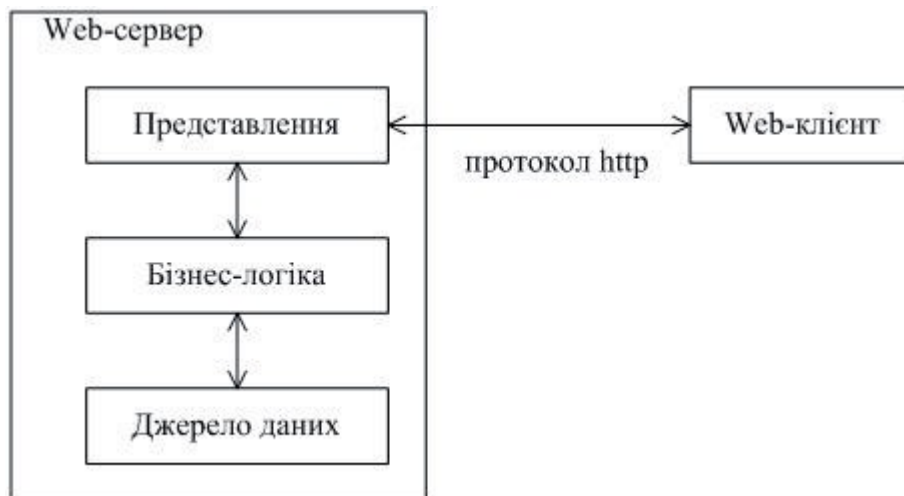


Рисунок 1.1 — Багаторівнева архітектура веб-додатку

3. База даних.

Ми можемо відзначити, що в алгоритмі роботи незримо, але досить активно бере участь ще одна «сторона» - база даних. Фактично, вона не є частиною веб-сервера, але більшість програм просто не можуть виконувати всі покладені на них функції без неї, так як саме в базі даних зберігається вся динамічна інформація додатка.

Бази даних функціонують під управлінням так званих систем управління базами даних (СУБД). Найпопулярнішими СУБД є MySQL, MS SQL Server, PostgreSQL, Oracle (всі — клієнт-серверні). Також існують вбудовані і файл-серверні СУБД. Для загального розвитку зазначу лише одну популярну вбудовану СУБД — SQLite.

Тепер розглянемо архітектуру веб-додатків з точки зору тестування ПО.

2. ТЕСТУВАННЯ НАВАНТАЖЕННЯ

Тестування частини веб-додатки, розміщеної на веб-сервері, можна провести і минаючи графічний (клієнтський) інтерфейс, проте це вимагає від фахівця певного рівня знань і навичок технічного характеру, а також застосування додаткових інструментів. Розглянемо веб-сервер з точки зору навантажувального тестування.

Тестування навантаження імітує роботу з додатком певної кількості користувачів. Цей вид тестування здійснюється за допомогою спеціальних інструментів (наприклад, Fiddler та jMeter), головна мета яких - визначити профілі навантаження і штучно створити для них навантаження, що виявляє граничні можливості програми (або сервера) в умовах роботи з ним тієї чи іншої кількості користувачів.

Отримана інформація піддається ретельному аналізу з подальшим виявленням «вузьких шляхів» і граничних програмних і апаратних можливостей, які в подальшому використовуються для забезпечення стабільності веб-сервера і самого додатка, що працює на ньому.

Наведемо приклад з практики функціонування великого комерційного продукту, який довгий час працював з різноманітними типами договорів. Одного разу в реліз випустили черговий особливо очікуваний тип договору, і на наступний день система повністю перестала працювати, а служба підтримки була завалена величезною кількістю звернень. Всьому виною став прорахунок в тестуванні: команда перевіряла одночасну роботу з десятками тисяч договорів, але ніхто не зміг передбачити, що на практиці мова піде про сотні тисяч, а іноді і про мільйони договорів. Тестування навантаження дозволяє виявити потенційні проблеми такого характеру ще на етапі тестування.

3. FIDDLER

Fiddler - ПО для відстеження всього вашого трафіку та генерації запитів. Використовуючи Fiddler, можна з легкістю відстежувати всі запити від клієнта і відповіді, переглядати їх деталі, а також вносити свої зміни і відправляти модифіковані запити на сервер, оцінюючи поведінку системи в такому випадку.

На практиці, використовуючи при тестуванні Fiddler, ви без зусиль зможете впорядкувати свої запити і відповіді за кодом стану і відібрати, наприклад, все 400-і і 500-і коди з подальшим їх аналізом. Таким чином дуже швидко «відловлюються» дефекти з стилями, скриптами, файлами, функціями програми, що не працюють.

Почавши роботу з ним, ми тим самим розробляємо віртуальний web проху, а значить, кожен запит від вашого локального браузера спрямовуватиметься виключно через його структуру. Всередині програми ви зможете розглядати всі до єдиного HTTP запити, наприклад, картинки, структуру CSS коду, JS скрипти і інші складові елементи класичного веб-ресурсу. Будь-який запит і відповідь має свої персональні параметри, за допомогою яких можна судити про оптимальність або проблемах функціонування розроблюваного веб-сайту.

Наприклад, ви можете детально проаналізувати такі ситуації, як віддача веб-порталом невеликих ресурсів - невеликих картинок або іконок. Логічно, що їх за просто можна об'єднати в один спрайт, щоб виконувати лише один запит, що кардинальним чином знизить ступінь і час завантаження медіаресурсів.

Вас також можуть насторожити деякі формати і розміри веб-ресурсів. Дуже великі зображення найкраще оптимізувати, наприклад, за допомогою конвертації в інший формат. Для текстових даних можна використовувати поширену конвертацію в Gzip-формат.

До слова, саме Fiddler демонструє, включено стиснення для вхідного трафіку чи ні. Даний параметр можна запросто простежити за спеціальним заголовку Content-Encoding: gzip.

Кожен цікавить запит можна проаналізувати в формі спеціальної часової стрічки, по якій максимально чітко видно, який проміжок часу необхідний, щоб вся сторінка повністю завантажилася, а також час, необхідний для оптимального відображення кожного її структурного елементу.

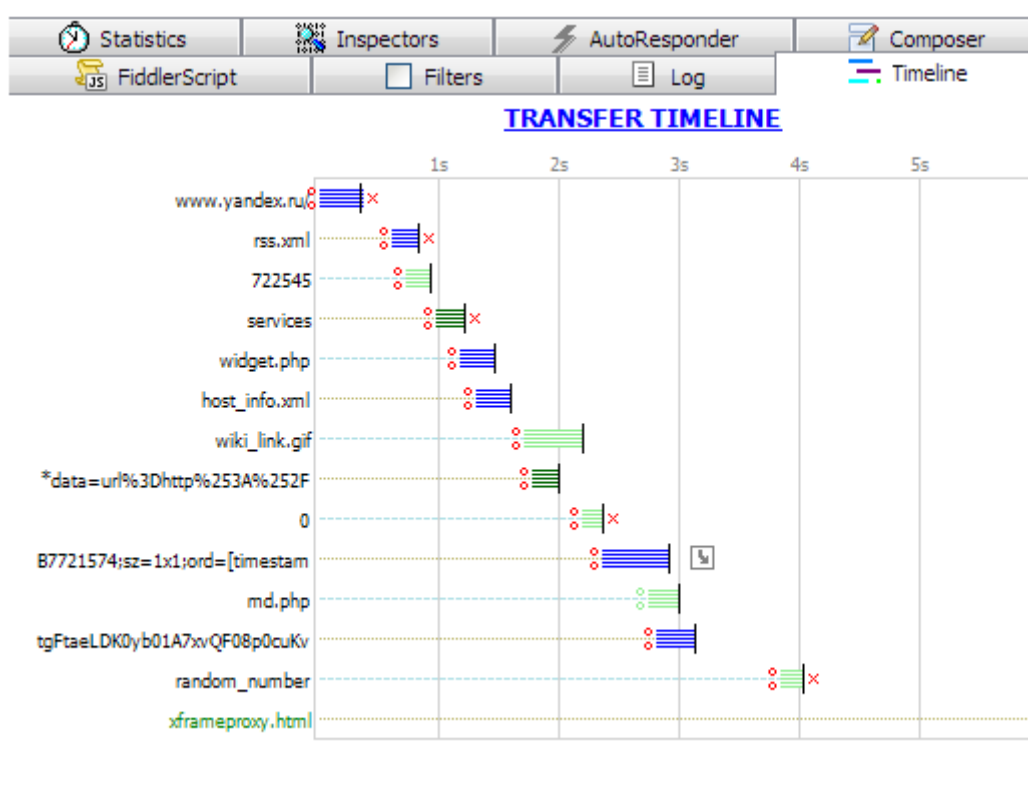


Рисунок 3.1 — Часова стрічка

У додатку є достатня кількість зручних і корисних функцій, за допомогою яких, наприклад, можна виставити параметри повільного з'єднання. Так тестувальники можуть побачити весь процес завантаження кожного окремого елемента сайту за умови, що у клієнтів буде дуже повільний інтернет.

До речі, дуже корисно, що можна наочно переглянути, які елементи завантажуються першими, а які - в останню чергу. Тобто можна оптимізувати порядок вивантаження всіх веб-елементів.

Також можна відзначити наявність спеціального редактора для HTTP запитів. У користувача з'явиться реальна можливість підміняти будь-який обраний HTTP заголовок за допомогою спеціального скриптового редактора.

Також можна використовувати Firebug, а не Fiddler, але з точки зору оптимізації параметрів саме Fiddler можна вважати найбільш багатофункціональним і оптимізованим додатком.

Також можна відзначити, що він абсолютно безкоштовний, постійно розвивається, а значить, дозволяє оперувати виключно свіжими і працездатними версіями.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Особенности тестирования веб-приложений [Электронный ресурс]. – Режим доступа: <http://quality-lab.ru/key-principles-of-web-testing/>;
- 2) Отладка с помощью fiddler и charles [Электронный ресурс]. – Режим доступа: <https://testmatick.com/ru/otladka-s-pomoshhyu-fiddler-i-charles/>;
- 3) Использование Fiddler для эмуляции различных сетевых условий в автотестах[Электронный ресурс]. – Режим доступа: <https://www.slideshare.net/VLDCORP/fiddler-69866720>;
- 4) Методика проектування web-додатків з використанням платформи Java EE [Електронний ресурс]. – Режим доступу: <http://masters.donntu.org/2013/fknt/riabinin/library/article4.htm>.