

**ВИБІР СУЧАСНИХ ВЕБ-ОРІЄНТОВАНИХ ТЕХНОЛОГІЙ ПРОГРАМУВАННЯ ПІД ЧАС РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА JAVA-ПЛАТФОРМІ**

Розробники програмного забезпечення на мові Java потребують ефективних веб-орієнтованих технологій. Краще рішення цієї проблеми полягає у використанні веб-каркасів (фреймворків). У світі існує більше сотні Java веб-фреймворків і вибір оптимального рішення вже не є тривіальним завданням. У даній статті аналізується поточний рейтинг Java веб-каркасів в світі і в пострадянському просторі. Ми розглянули структурні і функціональні особливості найбільш популярних веб-фреймворків, їх переваги та недоліки.

Ключові слова: мова програмування java, веб-фреймворк, веб-каркас, архітектура програмного забезпечення, технології програмування, програмний модуль.

A.A. BURLAKOV, O.O. BALYSHYN  
Khmelnitskyi National University

**CHOICE OF MODERN WEB-BASED PROGRAMMING TECHNOLOGIES DURING SOFTWARE DEVELOPMENT ON JAVA-PLATFORM**

The software developers need of effective java web-oriented programming technologies. The best solution to this problem is to use a web frameworks. In the world there are more than a hundred Java web frameworks and selection of the optimal solution is not trivial task. This article analyzes the current rating of java web frameworks in the world and in the post-Soviet space. We have considered structural and functional features of the most popular web frameworks and their advantages and disadvantages.

Keywords: programming language java, web framework, software architecture, programming techniques, software module, model-view-controller design pattern.

Мова програмування java останні шість років залишається найпопулярнішою мовою для створення ефективного та гнучкого програмного забезпечення (ПЗ) [1]. Ще з моменту появи на ринку розробки ПЗ, мова java містила розвинуті базові засоби для роботи в мережі. Але широке використання веб, зростання вимог замовників до функціональності та термінів розробки програмних продуктів призвело до того, що тільки базовими технологіями задовольнитись вже стало неможливо. Почали з'являтися технології, які дозволяють вирішувати певні задачі програмування, що найбільш часто зустрічаються в розробників ПЗ, в прозорий для них спосіб. Ці технології реалізовані на перевірених часом програмних рішеннях – шаблонах проектування і носять назву каркаси (фреймворки). В залежності від затребуваності області програмування, для вирішення її стандартних задач на ринку можуть існувати одиниці, десятки і навіть сотні каркасів. Таким чином, вибір фреймворка вже не вважається тривіальною задачею і потребує ґрунтовного аналізу існуючих рішень. Для полегшення веб-розробки набули популярності каркаси веб-додатків (Web application framework, WAF) – це фреймворки, призначені для створення динамічних веб-сайтів, мережеских додатків або сервісів. Вони спрощують розробку і позбавляють програміста від необхідності написання рутинного коду. Значна їх кількість має вбудовані засоби, що спрощують доступ до баз даних, розробку інтерфейсу, і також зменшують дублювання коду. Саме вибору каркасу веб-додатків і присвячена дана робота.

Спочатку необхідно відмітити, що більша частина каркасів веб-додатків реалізують шаблон проектування Model-View-Controller (MVC). Шаблон MVC (рис. 1), описує простий спосіб побудови структури програми, метою якого є відділення візуального відображення для користувача (view) від даних системи (model) і механізму керування процесом спілкування цих категорій (controller). В результаті, додаток легше масштабується, тестується та супроводжується.

Всі веб-фреймворки можна поділити за рівнями абстракції на три типи:

- основані на запитах: в цьому випадку фреймворк безпосередньо обробляє вхідні запити та зберігає стани за рахунок серверних сесій. Прикладом є такі Java веб-фреймворки, як Spring MVC, Struts, Grails;
- основані на компонентах: в цьому випадку фреймворк абстрагує обробку запитів всередині стандартних компонентів і самостійно слідкує за станами, наприклад: JavaServer Faces;
- Rich Internet Application-фреймворки (RIA): слугують для розробки повноцінних java-додатків, які виконуються в браузері, наприклад JavaFX.

На даний час є досить велика кількість (більше сотні) різних Java веб-фреймворків і це велике розмаїття на любий смак лишає багатьох розробників необхідності знати їх архітектурну реалізацію. Здається, що фреймворк - це якась велика та складна реалізація, яку нереально відтворити. Насправді, як вже згадувалося вище, це просто набір стандартних патернів (шаблонів). Коли ви вивчаєте фреймворки, Вам потрібно не просто вивчати суто фреймворки, а разом з принципами його реалізації, принципами його роботи та шаблонами проектування, що в ньому використовуються. Який же веб-фреймворк на Java варто вибрати, щоб зробити процес розробки простішим та заощадити час? Є багато можливих варіантів, що надають свої переваги, і це заставляє задуматися над рішенням вибору. Посилаючись на різні закордонні

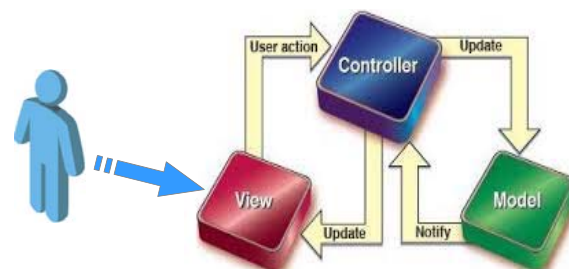


Рис. 1. Схематичне зображення принципу роботи архітектурного шаблона проектування MVC

джерела, зокрема на авторитетну компанію-розробника ПЗ ZeroTurnaroun [2], що веде статистичні дослідження, які стосуються Java-розробки, під назвою "Developer Productivity Report", можна отримати рейтинг популярності Java веб-фреймворків серед спільноти Java-розробників за кордоном (рис. 2).

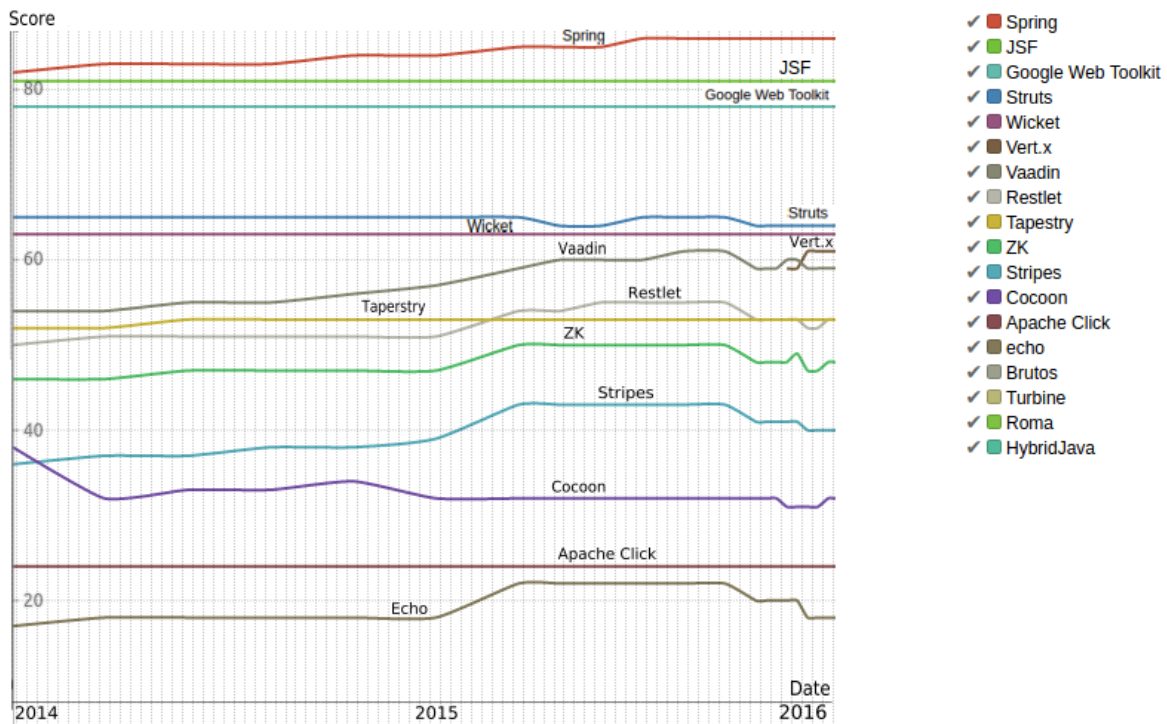


Рис. 2. Графік рейтингу Java веб-фреймворків на протязі 2014–2016 рр.

В якості критеріїв оцінювання при побудові рейтингу затребуваності веб-фреймворків на java слугували наступні: швидке прототипування додатків; складність архітектури; простота використання; повнота документації і спільноти розробників; структура екосистеми фреймворка; масштабованість; технічне обслуговування (в тому числі оновлення версій); зовнішній вигляд (UX).

Отже, результати даного рейтингу є наступними: Spring MVC (30%), JavaServer Faces (20%), Apache Struts (17%), GoogleWebToolkit (10%), Play Framework (8%), SparkFramework (7%), Grails (4%), Vaadin (4%).

Для аналізу вподобань веб-каркасів на java на ринку США розробниками ПЗ, нами було проаналізовано значну кількість IT-блогів та форумів [3–13]. Серед веб-орієнтованих проєктів, що розробляються на Java на даний момент, найбільш популярним є фреймворк Spring MVC. Його популярність обумовлена доступністю значної кількості модулів, що надає повний каркас Spring Framework, гнучкістю, легкістю інтеграції з іншими технологіями, наприклад із Google Web Toolkit (GWT) і Java Server Faces (JSF). Така популярність обумовлена також тим, що вся екосистема Spring Framework стала в свій час альтернативним спрощенням платформи Java2EE і спричинила різкий перехід до даного фреймворку та його популярність серед Java розробників. Серед фундаментальних фреймворків, які користуються попитом серед Java розробників на ринку США є каркас Apache Struts, хоча його частка невелика порівняно із Spring. Також слід зазначити, що для розробки масштабних бізнес рішень активно використовується GWT, але із інтеграцією із Spring. GWT найчастіше використовується у зв'язі із Spring MVC для розробки UI при розробці корпоративних систем. Також є популярною стандартна специфікація Java EE (Java Enterprise Edition), представлена стеком технологій: EJB, ServletAPI, JSP, JSF. Наведемо коротке портфоліо найбільш популярних веб-фреймворків.

**Spring MVC.** Spring MVC – найпопулярніший фреймворк для веб-проєктів на java на даний момент. Він володіє розвинутою екосистемою, тобто сам Spring MVC є лише одним із численних підмодулів каркасу Spring Framework і якщо бракує якоїсь додаткової функціональності при розробці веб-орієнтованих додатків, Ви легко зможете знайти підмодуль, який забезпечить необхідну підтримку. Spring Framework є просто величезним фреймворком завдяки тому, що розробники цього фреймворка охопили практично всі аспекти програмування промислових Java-додатків. Відповідно, і складових частин у Spring Framework чимало: IoC (Inversion of Control) контейнер, AOP-фреймворк, Data Access фреймворк, Transaction management, MVC-фреймворк, Remote Access фреймворк, Batch processing, Фреймворк аутентифікації і авторизації, Remote Management, Messaging-фреймворк, Testing-фреймворк. Величезна спільнота розробників завжди буде готова прийти на допомогу, якщо у Вас виникнуть питання. За допомогою Spring MVC можна без особливих зусиль розробляти великі і масштабовані веб-додатки. Проте, є у нього і кілька мінусів: даний фреймворк досить складний для вивчення, тому, незважаючи на хорошу документацію, початківці, швидше за все, зазнають труднощів в його освоєнні. Крім того, Spring MVC не надає зручних інструментів для створення користувальницького інтерфейсу (GUI). Spring – це, можна сказати, архітектурний фреймворк, тому що він придуманий не стільки для виконання якоїсь прикладної задачі (як, скажімо, Struts, який потрібен, щоб писати web-додатки), а для забезпечення кращої масштабованості, архітектури проєкту, можливості більш простого тестування і більш простої інтеграції з іншими фреймворками. Завдяки цьому писати великі програми стає

простіше – розробники просто уникають ряду проблем, пов'язаних зі створенням промислових додатків, замість того, щоб їх вирішувати. За допомогою Spring MVC є можливість “без особливих зусиль” розробляти великі і масштабовані веб-проекти, адже він має досить велику офіційну документацію та найбільше ресурсів в пострадянському інформаційному просторі для вивчення, порівняно із іншими фреймворками. Для створення візуальних компонентів і реалізації певної логіки можна використовувати відомі всім веб-розробникам бібліотеки - наприклад JQuery, Twitter Bootstrap тощо. Spring не має власних технологій формування представлення, що робить його універсальним та гнучким. До недоліків можна віднести те, що доведеться витратити більше часу на верстку.

Java Server Faces (JSF). JSF представляє собою компонентно-орієнтований фреймворк, який є стандартом платформи Java EE і офіційно підтримується корпорацією Oracle. Компоненти представляють собою готові графічні елементи (кнопки, перемикачі, меню, списки, таблиці тощо), компонує які на динамічних веб-сторінках розробник формує рівень представлення в архітектурі MVC. Моделлю ж виступають звичайні java-класи, поля яких інкапсулюють інформацію предметної області (щоб фреймворк JSF розумів їх як елемент архітектури, можна або анотувати ці класи, або надати їх опис в конфігураційному xml-файлі). Саме каркас JSF виступає в якості **контролера** – з'єднує представлення з моделлю і його діяльність є абсолютно прозора для програміста. Фреймворк легкий у використанні та вивченні завдяки наявності відмінної документації (надає сама Oracle, але слід відмітити, що ця документація практично вся англійською). Якщо стандартних візуальних компонентів недостатньо для реалізації дизайнерської ідеї, розробник може звернутись за додатковими до відомих провайдерів компонентів RichFaces, PrimeFaces, IceFaces. Розробнику доступна потужна екосистема Java EE, що забезпечує широкі можливості на всі випадки життя. Можна представити процес створення користувацького інтерфейсу в JSF в якості аналогу розробки настільних додатків з використанням графічної бібліотеки swing. Недоліками каркасу, на нашу думку, є наступні: розробник жорстко прив'язаний до графічних компонентів JSF при побудові представлення, на відміну від Spring MVC, який співпрацює з будь-якою технологією побудови представлення; обмежені можливості в використанні технології Ajax – зручно користуватись цим механізмом тільки в рамках, обумовлених фреймворком.

Spark Framework. Spark Framework є простим, легким та швидким веб-фреймворком для побудови мікросервісів. Набір Spark фреймворку, як описано на офіційному сайті – це не конкуренція з подібними веб-фреймворками інших мов програмування, а забезпечити чисту Java-альтернативу для розробників, які хочуть (або зобов'язані) швидко розробляти свої веб-додатки на Java. Spark побудований навколо філософії лямбда у Java 8, який робить типове застосування Spark набагато менш “багатослівним”, ніж більшість додатків, написаних на інших веб-фреймворках Java. Даний фреймворк ідеальний для розробки REST-сервісів, хоча має можливості для розробки веб-додатків. Даний фреймворк має підтримку різних “Views and Templates”, на відміну JSF, який має свій власний шаблонізатор, власні View-елементи. Це такі шаблонізатори (Template Engine): Velocity, Freemarker, Mustache, Handlebars, Jade, Thymeleaf, Jetbrick, Pebble, Water. Це робить його універсальним, незалежним від певних технологій. Також найбільшою перевагою фреймворку, як засобу для побудови мікросервісів, є обробка HTTP запитів. Проте його гнучкість у використанні сторонніх компонентів для побудови представлення і для роботи з БД (оскільки стандартних засобів у Spark`у немає) може показатися недоліком, оскільки потрібно вивчати сторонні бібліотеки. Для розгортання проектів, Spark використовує вбудований Jetty веб-сервер, тому для цього достатньо лише встановленої JDK 8.

GoogleWebToolkit. GoogleWebToolkit представляє собою фреймворк для розробки RIA (Rich Internet Application) Ajax-додатків на основі Java-коду. За допомогою GWT веб-додаток розробляється на мові Java, і його код на стадії розробки містить визначення GUI-інтерфейсу, обробку його подій і роботу з даними. Потім Java-код GWT-додатку, компілюється в JavaScript-код Web-сторінки клієнта, а на стороні сервера залишається лише Java-код Web-сервісів, відповідаючих за роботу з даними. При цьому JavaScript-код веб-сторінки містить Ajax-клієнтів веб-сервісів. Таким чином, фреймворк GWT дозволяє написати клієнтську частину веб-додатка на мові Java, а потім скомпілювати її в оптимізований JavaScript-код, що коректно працює в більшості Web-браузерах, звільняючи розробника від завдань оптимізації та крос-браузерності додатка. Даний фреймворк є відмінним інструментом для побудови користувацьких інтерфейсів, проте для реалізації серверної логіки для масштабних веб-проектів, зазвичай використовують Spring з інтегрованим у нього GWT.

Apache Struts. Apache Struts – веб-фреймворк з відкритим кодом для розробки Java EE веб-додатків. Він використовує і розширює Java Servlet API, надаючи архітектуру MVC (Модель-Вид-Контролер). Struts був створений для того, щоб чітко відділяти модель (бізнес-логіку) від представлення (логіки відображення, у нашому випадку це JSP, хоча підтримуються і XML/XSLT і Velocity), і контролера (сутності, яка займається передачею інформації від моделі до представлення). Інформація передається між моделлю і представленням у вигляді особливих об'єктів типу JavaBeans. Багата бібліотека тегів дозволяє виводити і записувати вміст цих бінів на презентаційному рівні без включення Java-коду. Недоліком даного фреймворку є те, що створений він вузькоспеціалізованим – лише для написання веб-додатків.

Vaadin. Vaadin — ще один фреймворк, що використовує Java, як єдину мову для розробки представлення і веб-логіки. Це є одна із найзначніших функцій в Vaadin. Фреймворк використовує модель подій і певні елементи користувацького інтерфейсу – віджети, що робить його дуже близьким до моделі розробки настільних додатків на Java з використанням HTML і JavaScript. Vaadin побудований на основі Google Web Toolkit, а якщо бути точнішим, то розширює його, доповнюючи серверною частиною. Він використовує Google Web Toolkit для генерування за запитами користувацьких сторінок. Так як Google Web Toolkit функціонує тільки на стороні клієнта, Vaadin додає додаткову валідацію даних на стороні

сервера: це вирішує проблеми безпеки, пов'язані з можливістю підміни даних або коду JavaScript. Відповідно, при зміні і пошкодженні даних, що надходять від браузера, сервер, визначивши це, не пропускає запити. Це є єдиною перевагою Vaadin над GWT.

Vert.x. Ще одним фреймворком, який буде закривати даний список, стане Vert.x. Хоча це не зовсім суто Java веб-фреймворк, Vert.x — поліглот, універсальний фреймворк. Це означає, що ви можете реалізувати компоненти, які потрібно виконати Vert.x (так звані "Verticles") на різних мовах. Vert.x може виконувати Java, JavaScript, Ruby, Python, Groovy, Scala і Clojure. Vert.x – це асинхронний, event-driven фреймворк, ціль якого перетинається з node.js. Тобто це універсальний фреймворк-поліглот для JVM. Vert.x називають себе «реактивним» інструментарієм. Реактивні додатки складаються з компонентів, які посилають повідомлення або події один до одного. Це зовсім інший внутрішній дизайн, ніж Java EE. Ця внутрішня конструкція підходить для розробки таких типів додатків, як чат чи інші, побудовані на веб-сокетах додатки і ігрові сервери. Реалізація паралельної обробки покладена на фреймворк, тому при розробці не потрібно піклуватися про це (не потрібно використовувати `synchronized` or `volatile`, та блокування). Також не потрібно ніяких xml-конфігурацій. Даний фреймворк розроблявся під впливом відносно нової серверної технології `node.js` і дуже сподобається тим, хто знайомий з `node`. Отже, у висновку можна сказати, що даний фреймворк із вбудованим Netty-сервером є високопродуктивним рішенням для розробки Daemon-серверів, чи інших проектів з використанням технології веб-сокетів та асинхронної багатопотокової архітектури.

Отже веб-фреймворк – це набір бібліотек (класів та інтерфейсів) для швидкої розробки веб-орієнтованих програмних систем, який звільняє програміста від рутинних задач і дозволяє зосередитись на програмуванні бізнес-логіки системи. Також перевагами фреймворків є те, що вони вже надають структуровану архітектуру програми, наслідуючи певні патерни проектування, та надають нам вже вбудовані класи для роботи з базою даних та багато іншого. Головна ціль фреймворку – забезпечити програміста зручним середовищем для розробки з великими та розширюваним функціоналом. На нашу думку, програміст, який вибирає веб-фреймворк для вивчення і використання, в першу чергу має звернути увагу на каркаси Spring MVC та JSF. Обидва рішення є найбільш популярними серед спільноти веб-програмістів на java, мають розширену екосистему, потужну спільноту розробників, вичерпну документацію. Spring MVC є більш гнучким, тому що має орієнтованість на запити. Коли розробник чітко розуміє, які запити і куди відправляються з боку клієнта, то це приводить до більш раціонального проектування програми, до більш оптимізованого і швидкого коду. Імовірність неочевидних помилок, з якими доводиться стикатися в роботі з JSF, мінімальна. Підтримувати додаток на Spring MVC простіше, тому що все є більш прозорим. Але для дуже швидкої розробки веб-орієнтованого інтерфейсу певних категорій (корпоративних систем), JSF є ідеальним рішенням, бо має потужні бібліотеки графічних компонентів, продуманий та перевірений життєвий цикл каркасу.

### Література

1. Бурлаков А.А. Організація гнучкого доступу до даних в додатках на java-платформі / А.А. Бурлаков // Вісник ХНУ. – 2016. – № 2. – С. 30–36.
2. The official website of the online resource [Електронний ресурс]. – Режим доступу : <http://zeroturnaround.com/>
3. Офіційна документація. Spring [Електронний ресурс]. – Режим доступу : <http://spring.io/docs>
4. Офіційна документація. Spark [Електронний ресурс]. – Режим доступу : <https://sparktutorials.github.io/>
5. Офіційна документація. Apache Struts [Електронний ресурс]. – Режим доступу : <https://struts.apache.org/docs/guides.html>
6. Офіційна документація. Google Web Toolkit [Електронний ресурс]. – Режим доступу : <http://www.gwtproject.org/doc/latest/tutorial/index.html>
7. Какой Java Web фреймворк выбрать? [Електронний ресурс] // Спільнота програмістів. DOU : портал. – Режим доступу : <https://dou.ua/forums/topic/11227/>
8. Выбор Java фреймворка для веб-разработки [Електронний ресурс] // Тостер — вопросы и ответы для IT-специалистов : портал. – Режим доступу : <https://toster.ru/q/9571>
9. Нужен веб-фреймворк для java [Електронний ресурс] // LINUX.ORG.RU: Русская информация об ОС Linux : портал. – Режим доступу : <http://www.linux.org.ru/forum/web-development/9069828>
10. Java. Фреймворки для веба [Електронний ресурс] // askdev.ru : социальный сайт вопросов и ответов для IT-специалистов. – Режим доступу : <http://www.askdev.ru/java/4479/>
11. Find your new favorite web framework [Електронний ресурс] // Web framework rankings | HotFrameworks : портал. – Режим доступу : <http://hotframeworks.com/languages/java>
12. Sparkjava is an amazing Java web framework. Do you really need it? [Електронний ресурс] // zeroturnaround.com : портал. – Режим доступу : <http://zeroturnaround.com/rebellabs/sparkjava-is-an-amazing-java-web-framework-do-you-really-need-it/>
13. Inside Vert.x. Comparison with Node.js. [Електронний ресурс]. – Режим доступу : <http://www.cubrid.org/blog/dev-platform/inside-vertx-comparison-with-nodejs/>

Рецензія/Peer review : 5.4.2016 р. Надрукована/Printed : 6.6.2016 р.  
Рецензент: д.т.н., проф., Марченко В.Л.