

УДК 004.415.3

Пех П.А. к.т.н. доц., Бондарчук В.О.

Луцький національний технічний університет

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ СИНХРОННИХ ТА АСИНХРОННИХ ТЕХНОЛОГІЙ ПРОГРАМУВАННЯ У ПРОЦЕСІ РОЗРОБКИ ВЕБ ДОДАТКІВ

Пех П.А., Бондарчук В.О. Дослідження ефективності синхронних та асинхронних технологій програмування у процесі розробки веб додатків. В даній статті відображено концепт взаємодії серверних додатків із CMS системами на прикладі асинхронної технології Node.js та CMS Drupal.

Ключові слова: Drupal, Node.js, Open source, API, асинхронність, Javascript, PHP, real-time apps.

Пех П.А., Бондарчук В.А. Исследование эффективности синхронных и асинхронных технологий программирования в процессе разработки веб приложений. В данной статье отражен концепт взаимодействия серверных приложений с CMS системами на примере асинхронной технологии Node.js и CMS Drupal.

Ключевые слова: Drupal, Node.js, Open source, API, асинхронность, Javascript, PHP, real-time apps.

Pekh P., Bondarchuk V. The efficacy of sync and async programming technologies in the web applications development. This article shows the concept of integration server applications with CMS systems on example async technology Node.js and CMS Drupal.

Keywords: Drupal, Node.js, Open source, API, async, Javascript, PHP, real-time apps.

Вступ. Drupal є прекрасною платформою, але він не може зробити все. Зі зростанням об'ємів навантажень на сайт, зростають і випадки коли движок (Drupal) не справляється. На щастя, всі ці проблеми, пов'язані з оптимізацією і зменшенням навантаження на систему, можна розв'язати за допомогою сторонніх сервісів, такі як Node.js.

Node.js - це платформа з відкритим вихідним кодом для написання серверних і мережеских додатків. В її основі лежить подієво-орієнтоване та асинхронне програмування з неблокуючим введенням/виведенням. Платформа призначена для відокремленого виконання веб-додатків на мові JavaScript. Для виконання JavaScript-коду використовується рушій V8, розроблений компанією Google. Інтеграцію Node.js з Drupal 7 можна провести за допомогою модуля Node.js integration.

Все більш поширеними засобами для оптимізації сайтів стають сторонні API. В свою чергу, API (Application Programming Interface) - це зазвичай (але не обов'язково) метод абстракції між низькорівневим та високорівневим програмним забезпеченням. Одним з найпоширеніших призначень API є надання набору широко використовуваних функцій, наприклад для малювання вікна чи іконок на екрані. Програмісти використовують переваги API у функціональності, таким чином їм не доводиться розробляти все з нуля. API є абстрактним поняттям — програмне забезпечення, що пропонує деякий API, часто називають реалізацією (implementation) даного API. У багатьох випадках API є частиною набору розробки програмного забезпечення, водночас, набір розробки може включати як API, так і інші інструменти/апаратне забезпечення, отже ці два терміни не є взаємозамінювані.

Інтеграція з сторонніми сервісами являє собою досить гнучке рішення, яке дозволяє робити сайти більш потужними без використання купи нового функціоналу всередині веб додатка. Головна проблема – це те, більшість сторонніх API сервісів дуже повільні.

Node.js призначений для відокремленого виконання високопродуктивних мережеских застосунків на мові JavaScript. Функції платформи не обмежені створенням серверних скриптів для веб, платформа може використовуватися і для створення звичайних клієнтських і серверних мережеских програм. Для забезпечення виконання JavaScript-коду використовується розроблений компанією Google рушій V8. Для забезпечення обробки великої кількості паралельних запитів Node.js задіює асинхронну модель запуску коду, засновану на обробці подій в неблокуючому режимі і визначенні обробників зворотніх викликів (callback). В якості способів мультиплексування з'єднань підтримується epoll, kqueue, /dev/poll і select. Для мультиплексування з'єднань використовується бібліотека libev, для створення пулу нитей (thread pool) задіяна бібліотека libeio, для виконання DNS-запитів у неблокуючому режимі інтегрований c-ares. Всі системні виклики, що викликають блокування, виконуються всередині пула нитей і потім, як і обробники сигналів, передають результат своєї роботи назад через неіменовані канали (pipe).

За своєю суттю Node.js схожий на фреймворки Perl AnyEvent, Ruby Event Machine і Python Twisted, але цикл обробки подій (event loop) у Node.js прихований від розробника і нагадує обробку подій у веб-застосунку, що працює в браузері. При написанні програм для Node.js необхідно враховувати специфіку подієво-орієнтованого програмування, наприклад, замість виконання.

Node.js це асинхронна мова програмування, іншими словами, поки в одному потоці виконується `db_select()`, програма починає виконання іншого шматка коду в іншому потоці, наприклад `db_query`, а після опрацювання обох потоків і отримання `response` (відповіді), повертає дані, які використовуються у програмі далі.

Це єдиний процес, але він багатопоточний - у порівнянні з PHP, який створює процес для кожного користувача, в Node.js один процес може виконувати паралельні дії для кожного користувача. Це набагато швидше, тому що сервери не повинні витрачати ресурси і час для створення процесів для кожного запита (request), витягати якісь дані з бази даних чи щось подібне. Потрібно просто запустити сервер, завантажити всі дані і чекати на запит від користувача.

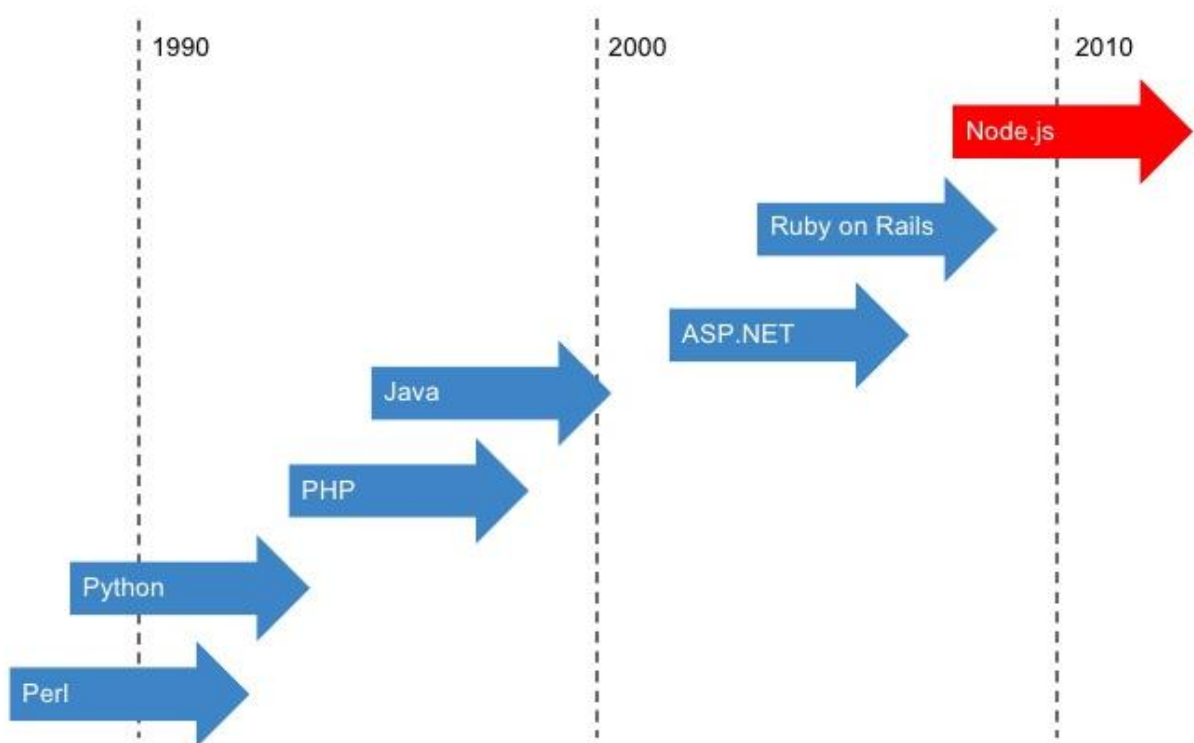


Рис. 1 – Хронологія появи деяких мов програмування та фреймворків.

Основна частина. Для налаштування Node.js в Drupal потрібно:

- 1) Встановити всі необхідні пакети для роботи з Node.js та сам Node.js
`$ sudo apt-get install build-essential git curl openssl libssl-dev`
- 2) Компіляція його вихідних кодів
`sudo make install`
- 3) Якщо все пройшло без помилок, то вам відкриється доступ до оболонки node.js.
`$ node`
`> console.log('Hello world');`
Результатом має бути:
`Hello world`
- 4) Наступним нашим кроком буде встановлення модуля node.js на Drupal
`drush en nodejs`
- 5) Встановлення всіх необхідних залежностей
`$ sudo npm install socket.io`

```
$ sudo npm install request
$ sudo npm install express
$ sudo npm install connect
```

- б) Далі необхідно зайти на сторінку налаштування node.js, скопіювати вміст поля “Suggested configuration” і створити файл sites/all/modules/nodejs/nodejs.config.js та додати туди скопійовані налаштування.

Node.js server configuration builder

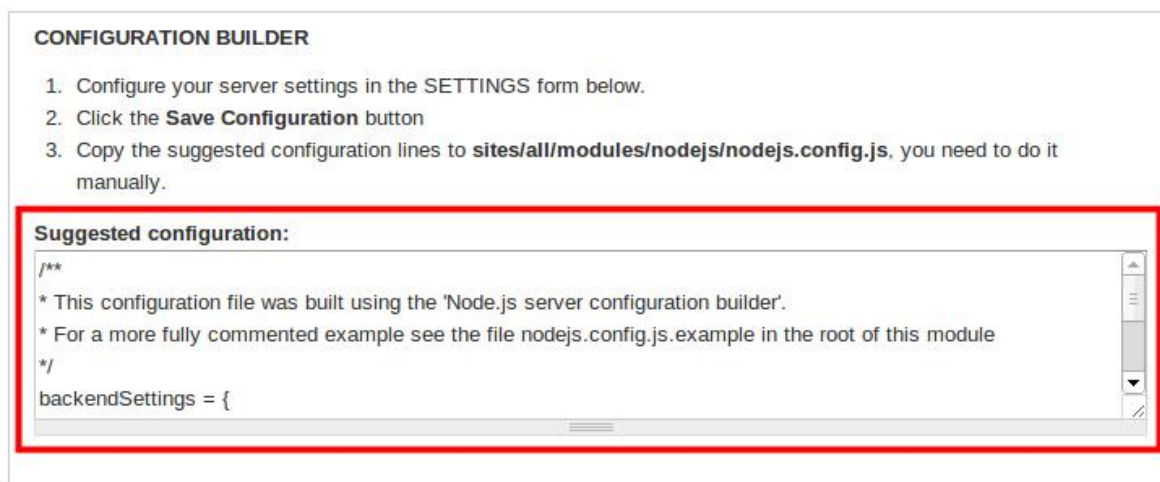


Рис. 2 – Видгляд вікна конфігурації Node.js Integration module у Drupal.

Все, платформу node.js налаштовано. Перевірити її роботу можна, включивши модуль “Nodejs Watchdog” (він входить в пакет модуля Node.js integration).

Для того, щоб постійно не прописувати ‘node server.js’ можна його запустити в фоновому режимі за допомогою пакета Forever. Forever вміє керувати фоновими процесами, а саме: перезапускати їх після збою; перенаправляти стандартний вивід помилок у файли журналів, та виконувати інші корисні функції. Отже, для того, щоб встановити його на сервері потрібно:

```
$ sudo npm install -g forever
```

Тепер замість 'node server.js' в консолі можна виконувати 'forever start server.js'. Для того, щоб зупинити - прописати 'forever stop server.js'. Щоб переглянути, які процеси запущено через forever, можна скористатись 'forever list'.

Далі буде продемонстровано приклад відображення “Hello World!” за допомогою Node.js без участі Drupal CMS.

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(8000);
```

- Переваги Node.js над прост Javascript:
- 1) не вибагливий в ресурсах;
 - 2) немає стандартних особливостей веб сервера;
 - 3) не потрібна аутентифікація;
 - 4) не потребує сесій для користувачів;
 - 5) не потребує cookies;
 - 6) не потребує емейлів;
 - 7) не потребує шаблонів (templates);
 - 8) не містить в собі MVC чи інший інструкцій фреймворка;

Node.js і Drupal

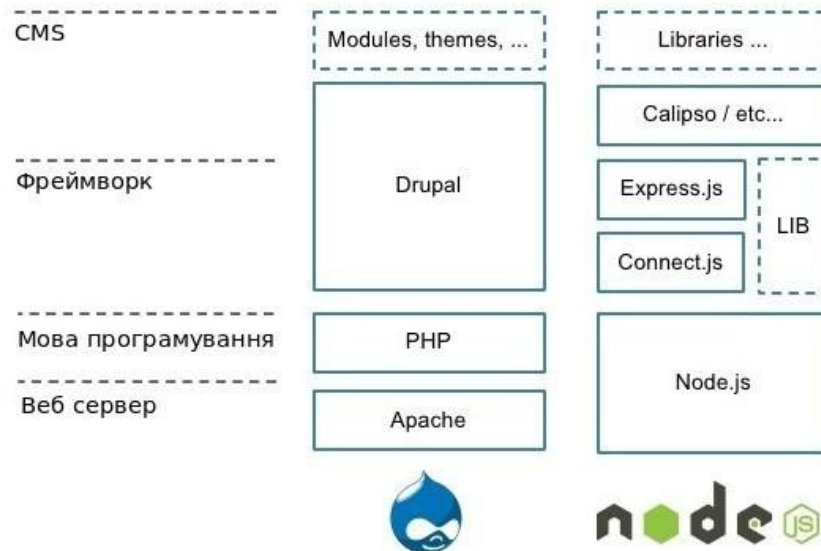


Рис. 3 – Схема інтеграції і взаємодії Drupal з Node.js.

У наступному прикладі буде продемонстровано приклад виведення тексту “Hello World” методами Drupal через взаємодію Node.js API.

```
$message = (object) array(  
  'broadcast' => TRUE,  
  'data' => (object) array(  
    'subject' => 'buu',  
    'body' => 'Hellow World!',  
  ),  
);  
nodejs_enqueue_message($message);
```

Більш складний приклад можна проглянути в модулі “Nodejs Watchdog”, який входить в пакет модуля Node.js Integration.

Висновки

У роботі досліджено перспективність і аспекти використання асинхронних та синхронних технологій на прикладі серверних скриптів, побудованих за допомогою технології Node.js та показано приклад інтегрування Node.js API з CMS Drupal.

1. Кантелон М.. Node.js in Action. –М.:«Manning Publications Co», 2014.-417 с.
2. Херрон Д. Node.js Разработка серверных веб-приложений на JavaScript. – М.:Издательство ДМК Пресс, 2012.-946 с.
3. Пауерс Ш. Изучаем Node.js С. – Питер: Орияна-Нова, 2014. -400 с.