

Програмні засоби оброблення інформації для Інтернету речей

Волосян Д. О.

e-mail kraine212@gmail.com

Кафедра звукотехніки та реєстрації інформації kaf-ztri.kpi.ua

Факультет електроніки fel.kpi.ua

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського» kpi.ua

Київ, Україна

Анотація—Описано аналіз методів створення програмних засобів оброблення інформації для Інтернету речей. Досліджено методи створення окремих етапів розробки додатків для зчитування інформації та керування Інтернет речами. Наведені порівняльні таблиці ефективності різних методів в залежності від критеріїв розробки. Також розроблено таблицю аналізу ефективності поєднання різних методів створення front-end та back-end частин програмних засобів оброблення інформації для Інтернету речей. Розглянуто лише найефективніші та найбільш розповсюджені методи розробки front-end та back-end частини додатку.

Бібл. 12, рис. 3, табл. 3.

Ключові слова — програмні засоби обробки інформації; веб-додаток; ефективність програмних засобів; front-end; back-end; веб-розробка; додаток IoT; сервер.

I. Вступ

На даному етапі розвитку сучасних технологій важливе місце займає молодий, але дуже перспективний напрямок – Інтернет речей. Зокрема, особлива увага приділяється технічній реалізації. Розробляються різні прилади, зокрема ті, що раніше існували, але без підключення до мережі Інтернет, підключаються до користувача через будь-яку безпроводову мережу та покращують користування цим пристроєм. Вони постійно покращуються та з'являються нові. Адже надзвичайно важливо забезпечити комфортне користування з максимальною дистанцією керування та кількістю функцій, також важливу роль відіграє швидкість та комфорт додатку керування даним пристроєм.

Насамперед Інтернет речей, як у сенсі їх створення, так і у самій назві говорять нам про те, що вони тим чи іншим чином під'єднані до мережі Інтернет. Їхня сутність полягає в тому, що з ними можна взаємодіяти за допомогою телефону/ПК або іншого пристрою, під'єданого до мережі Інтернет (або bluetooth).

Під час взаємодії з пристроєм через мережу Інтернет користувач може подивитись на стан приладу, його параметри та змінювати їх. Для цього потрібно використати розроблений для з цією метою мобільний або веб-додаток.

Оскільки розповсюдженість Інтернету речей стрімко зростає, тому і вимоги користувачів до зручності та функціональності додатків зростають також швидко. Тому перед розробниками постає складна, та на

даний момент нерозв'язна, задача: потрібно розробити додаток, зі зручним та приємним дизайном інтерфейсу користувача, максимальною швидкістю та реакцією на дії користувача, а також з широким функціоналом.

Тема є актуальною, тому що дана сфера дуже стрімко розвивається, в неї з кожним роком інвестуються все більші кошти. Напрацювання Інтернету речей використовуються у різних галузях як промисловості, так і повсякденного життя. Зокрема, в медицині; у військовій сфері; у сфері продажів; у сфері підбору персоналу, та інше. Дані технології спрощують життя людей та робить його комфортнішим.

Розглянемо загальну структурну схему IoT (рис.1).

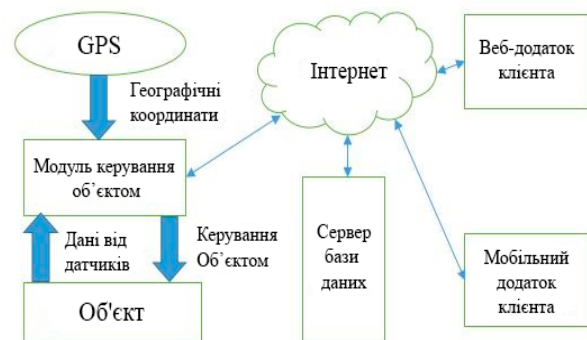


Рис. 1 Структурна схема IoT





Рис. 2 Структурна схема програмних засобів оброблення інформації для IoT

Відповідно до схеми: модуль керування об'єктом отримує дані від пристрою відстеження (якщо він передбачений) та від самого об'єкту, записуючи їх на сервер бази даних через мережу Інтернет. Клієнт, зайшовши у мобільний або веб-додаток може отримати інформацію з бази даних, переглянути її або змінити (якщо це можливо). При внесенні змін, інформація записується на сервер бази даних, звідки направляється до модуля керування об'єктом, котрий вносить зміни до самого об'єкту. Процес циклічно повторюється.

Розглянемо алгоритм роботи програмних засобів оброблення інформації для Інтернету речей (рис.2).

Клієнт, зайшовши в додаток, бачить відображення front-end частини веб-додатку, тобто клієнтську сторону інтерфейсу користувача до програмно-апаратної частини сервісу. В ній знаходиться інформація про об'єкт, котра береться з бази даних через мережу Інтернет за допомогою back-end, тобто програмно-апаратної частини сервісу. Для зміни певних параметрів об'єкту, клієнт вносить корективи до відповідного поля додатку, ці дані потрапляють у клієнтську сторону інтерфейсу користувача, звідки перенаправляються у програмно-апаратну частину сервісу. Після обробки – дані через мережу Інтернет потрапляють до сервера (рис.3).

Перевіркою виконання вимог користувача щодо змін об'єкту є відображення стану об'єкту у додатку. Тобто після вводу даних змін та надсилання їх на сервер, модуль керування об'єктом вносить корективи в об'єкт. Після цього модуль зчитує інформацію з пристрою та направляє її на сервер. Програмно-апаратна частина сервісу приймає ці дані та за допомогою клієнтської сторони інтерфейсу користувача та відображає їх. Клієнт бачить, чи відповідає стан об'єкту внесеним користувачем змінам: якщо стан об'єкту змінився, тоді це буде відображено в додатку клієнта; якщо стан об'єкту залишився без змін, тоді інформація оновиться на ту ж саму. Тому складеться враження, що стан об'єкту у додатку клієнта залишився незмінним. Цей спосіб не єдиний, але самий простий та наглядний.

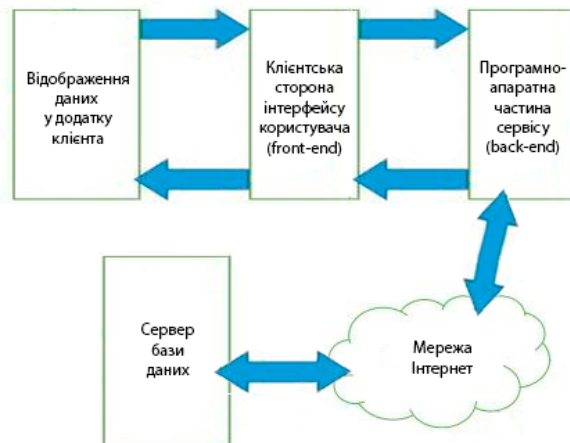


Рис. 3 Структурна схема роботи програмних засобів оброблення інформації для IoT

Для вирішення питання створення програмних засобів оброблення інформації для Інтернету речей потрібно:

- 1) Використати максимально відповідну до поставлених вимог Front-end частини створення додатку, тобто створити частину додатку, яка, на стороні клієнта, задовольняє таким вимогам: зручне розташування елементів та найкраще рішення графічного дизайну, швидкодія засобу оброблення інформації на стороні клієнта, коректне відображення на всіх видах пристроїв та браузерів, легкість читання та змінення коду, простота масштабованості та тестування проекту, ціна. Дана частина проекту повинна розроблюватись у відповідності до поставлених у проекті вимог та пріоритетів.
- 2) Обрати ефективну частину Back-end розробки, тобто створити частину додатку, яка, на стороні сервера, задовольняє таким вимогам: максимальна швидкодія додатку на стороні серверу, стабільність роботи, легкість читання та змінення коду, простота масштабованості та тестування проекту, ціна. Дана частина проекту повинна розроблюватись у відповідності до поставлених у проекті вимог та пріоритетів.
- 3) Вказані програмні частини розробки проекту мають бути підібрані таким чином, щоб між ними, також, була найбільш дійова взаємодія.
- 4) Вибір оптимального серверу для розміщення додатку.

II. FRONT-END

Front-end (Фронтенд) - клієнтська сторона інтерфейсу користувача до програмно-апаратної частини сервісу. Це частина додатку, яку безпосередньо бачить та взаємодіє користувач.

Вимоги:

- 1) Максимально зрозуміла інформаційна архітектура проектування взаємодії графічного



- дизайну та контенту, найкраще рішення графічного оформлення технічних елементів.
- 2) Максимальна швидкодія: швидкість завантаження сторінки та її відгук на дії користувача;
 - 3) Стабільність роботи додатку на стороні клієнта;
 - 4) Адаптація – коректне відображення інформації на всіх видах пристроїв;
 - 5) Кроссбраузерність – правильне відтворення елементів у всіх браузерах;
 - 6) Код повинен легко читатись та змінюватись;
 - 7) Масштабованість проекту;
 - 8) Простота тестування;
 - 9) Ціна: розробки, оновлення, підтримки.

Варіанти рішення:

- 1) HTML, CSS, JS;
- 2) React;
- 3) Angular;
- 4) Backbone.js;
- 5) Vue.js.

Рейтинг ефективності варіантів front-end частин розробки, без урахування вимог та пріоритетностей проекту, згідно з даною таблицею:

1 місце – React – 37;

- 2 місце – Angular – 36;
3 та 4 місце - Backbone.js та Vue.js – 35;
5 місце - HTML, CSS, JS – 27.

III. BACK-END

Back-end (Бекенд) - програмно-апаратна частина сервісу. Вона виконує певні дії на сервері та з'єднує front-end з ним. За допомогою back-end здійснюється зв'язок додатку клієнта та речі, якою він керує через мережу інтернет.

Вимоги:

- 1) Максимальна швидкодія проекту на стороні серверу;
- 2) Стабільність роботи;
- 3) Код повинен легко читатись та змінюватись;
- 4) Простота масштабування серверної частини проекту;
- 5) Легкість тестування;
- 6) Ціна: розробки, оновлення, підтримки.

Варіанти рішення:

- 1) PHP;
- 2) Java;
- 3) Node.js;
- 4) Redux.

ТАБЛИЦЯ 1 ВАРІАНТИ FRONT-END ЧАСТИНИ РОЗРОБКИ ТА ЇХ ЕФЕКТИВНІСТЬ

Показники оцінювання	Front-end				
	HTML, CSS, JS	React	Angular	Backbone.js	Vue.js
Дизайн	2	5	5	5	5
Швидкодія	2	5	4	4	4
Стабільність	4	5	5	5	5
Адаптація	4	5	5	5	5
Кроссбраузерність	4	5	5	5	5
Масштабованість	3	5	4	4	4
Тестування	3	4	5	4	4
Ціна	5	3	3	3	3

ТАБЛИЦЯ 2 ВАРІАНТИ BACK-END ЧАСТИНИ РОЗРОБКИ ТА ЇХ ЕФЕКТИВНІСТЬ

Показники оцінювання	Back-end			
	PHP	Java	Node.js	Redux
Швидкодія	4	4	5	5
Стабільність	4	5	5	5
Читаємість	4	5	5	5
Масштабованість	3	5	5	5
Ціна	5	4	4	4



ТАБЛИЦЯ 3 ЕФЕКТИВНІСТЬ ПОЄДНАННЯ FRONT-END ТА BACK-END ЧАСТИН РОЗРОБКИ

Front-end	Back-end			
	PHP	Java	Node.js	Redux
HTML,CSS,JS	5	3	3	3
React	3	4	5	5
Angular	3	4	5	4
Backbone.js	3	4	5	4
Vue.js	3	4	5	4

Рейтинг ефективності варіантів back-end частин розробки, без урахування вимог та пріоритетностей проекту, згідно з даною таблицею:

1-2 місце – Node.js та Redux – 24;

3 місце - Java – 23;

4 місце - PHP – 20.

IV. АНАЛІЗ ТА РЕКОМЕНДАЦІЇ

При створенні додатку важливу роль відіграють не окремо вибрані front-end та back-end варіанти розробки, а ефективність їх поєднання. Також важливо правильно вибрати сервер розташування додатку.

Вибір серверу:

- 1) Сервер повинен підтримувати та виконувати весь потрібний додатку функціонал;
- 2) Стабільність роботи;
- 3) Локальне розміщення серверу (наприклад, якщо і клієнт і річ, підключена до додатку, знаходяться в Києві, то потрібно вибрати сервер в Києві, а не в Чикаго, щоб забезпечити максимальну швидкість передачі даних);
- 4) Також для підвищення швидкодії потрібно обирати сервер з SSD (solid-state drive, або твердотільний накопичувач), а не HDD (hard (magnetic) disc drive, або накопичувач на жорстких магнітних дисках);
- 5) Під час вибору розміру SSD на хостингу потрібно врахувати розмір проекту, данні які будуть зберігатись під час роботи проекту та можливість його масштабування;

Таблиця 3 показує тільки ефективність поєднання методів, ефективність самих методів розглянуті в Таблиці 1 та Таблиці 2.

Максимальну ефективність поєднання front-end та back-end частин розробки, без урахування вимог та пріоритетностей проекту, згідно з даною таблицею мають:

- 1) PHP:
 - HTML, CSS, JS.
- 2) Node.js:
 - React;
 - Angular;
 - Backbone.js;
 - Vue.js.
- 3) Redux:
 - React.

ПЕРЕЛІК ПОСИЛАНЬ

- [1]. X. Cui, «The Internet of Things», Ethical Ripples of Creativity and Innovation, Palgrave Macmillan, London, p. 61-68, 2016. ISBN: 978-1-349-70059-2 DOI: [10.1057/9781137505545_7](https://doi.org/10.1057/9781137505545_7)
- [2]. J. Holler, V. Tsiatsis, C. Mulligan, S. Karnouskos, S. Avesand, D. Boyle «Internet of Things», Academic Press, 2014. ISBN: 9780124076846
- [3]. A. Mohammad, «Middleware Internet of Things», IEEE Internet of Things Journal, vol. 3, no. 1, pp. 70-95, 2015. DOI: [10.1109/JIOT.2015.2498900](https://doi.org/10.1109/JIOT.2015.2498900)
- [4]. J. Sauro, J. Lewis «Quantifying the User Experience», 2nd ed., Morgan Kaufmann, 2016. ISBN: 9780128023082
- [5]. E. Tittel, J. Noble «HTML, XHTML & CSS For Dummies, 7th Edition», 2011. ISBN: 9780470916599 DOI: [10.1002/9781118257586](https://doi.org/10.1002/9781118257586)
- [6]. A. White, «JavaScript Programmer's Reference», Wiley Publishing, Inc., 2009. ISBN: 978-0470344729
- [7]. Official site Angular. Available: <https://angularjs.org/> (accessed 2 March 2019).
- [8]. Official site React. Available: <https://reactjs.org/> (accessed 2 March 2019).
- [9]. K. Arto, «From the front end of projects to the back end of operations: Managing projects for value creation throughout the system lifecycle», International Journal of Project Management, 34(2), 2015. DOI: [10.1016/j.ijproman.2015.05.003](https://doi.org/10.1016/j.ijproman.2015.05.003)
- [10]. Official site PHP. Available: <https://php.net/> (accessed 2 March 2019).
- [11]. Official site Java. Available: <https://www.java.com/> (accessed 2 March 2019).
- [12]. Official site Redux. Available: <https://redux.js.org/> (accessed 2 March 2019).

Надійшла до редакції 10 квітня 2019 р.



Программные средства обработки информации для интернета вещей

Волосян Д. О.

e-mail kraine212@gmail.com

Кафедра звукотехники и регистрации информации kaf-ztri.kpi.ua

Факультет электроники fel.kpi.ua

Национальный технический университет Украины

«Киевский политехнический институт имени Игоря Сикорского» kpi.ua

Киев, Украина

Аннотация—Выполнен анализ методов создания программных средств обработки информации для интернета вещей. Исследованы методы создания отдельных этапов разработки приложений для считывания информации и управления интернет вещами. Приведенные сравнительные таблицы эффективности различных методов в зависимости от критериев разработки. Также разработаны аналитические таблицы эффективности сочетания различных методов создания front-end и back-end частей программных средств обработки информации для интернета вещей. Рассмотрены только самые эффективные и наиболее распространенные методы разработки front-end и back-end частей приложения.

Библ. 12, рис. 3, табл. 3.

Ключевые слова — программные средства обработки информации, веб-приложение, эффективность программных средств, front-end, back-end, веб-разработка, приложение IoT, сервер.

UDC 791.44

Software Tools for Handling Information for the Internet of Things

D. O. Volosian

e-mail kraine212@gmail.comDepartment of Sound Engineering and Information Registration kaf-ztri.kpi.uaFaculty of Electronics fel.kpi.uaNational technical university of Ukraine «Igor Sikorsky Kyiv polytechnic institute» kpi.ua
Kyiv, Ukraine

Abstract—The analysis of methods of creation of software for information processing for the Internet of things is described. The methods of creating separate stages of the development of applications for reading information and managing Internet speeches are explored. Comparative tables of efficiency of different methods are given, depending on development criteria. A table for analyzing the effectiveness of a combination of different methods for creating front-end and back-end parts of information processing tools for the Internet of Things has also been developed.

In this stage of the development of modern technologies, the important place does young, but a very prospective direction is the internet. Specifically, special warnings a technical realization is provided. Differential developments, including the following, before connecting to the network, connect to the user through any wireless network, and enhanced the use of this device. They are constantly improving and appearing new. Urgently important to provide comfort use with a maximum remote control and number of functions, an important role will play the speed and comfort the data management additional.

Software internet risk is strongly risks, and that and users 'requirements to contribute and functionality of applications rise the quantity of fast. That the developer appears complex, currently intractable, problem: you need to develop application with a convenient and pleasant user interface designs, maximum performance and response to user actions, and with broad functionality.

Theme is actual, that this sphere is very strongly developing, in all your funds investigated in that foreign year. Internet range development is used in different industries, as industry, and sustainable life. Specially in medicine; in the military sphere; in the sales area; in the field of personnel selection, and others. This technology makes life of people and makes it faster.

The client, going into the application, sees the display of the front-end parts of web application, that is, the client side of the user interface to the software and hardware part of the service. It contains information about the object, which is taken from the database through the Internet through back-end, that is, software-hardware part of the service. To change certain parameters of the object, the client makes adjustments to the corresponding field of the application, this data fall into the client side of the user interface, from which are redirected to the software and hardware part of the service. After processing – data through the Internet goes to the server. Checking whether the user's requirements for object changes are to display the status of the object in the application. This method is not the only, but the simplest and most visual.

In the research component of this article, only the most effective and most common methods of developing the front-end and back-end parts of the application are considered.

Ref. 12, fig. 3, tabl. 3.

Keywords — software tools, web application, software efficiency, front-end, back-end, web development, IoT application, server.

