

РОЗРОБКА СЕРВЕРНОЇ ЧАСТИНИ РОЗПОДІЛЕНОЇ СИСТЕМИ КОНТРОЛЮ ПАРАМЕТРІВ ТЕХНОЛОГІЧНОГО ПРОЦЕСУ

Паньків Ю.В., Михайлів В. В.

*Івано-Франківський національний технічний університет нафти і газу,
вул. Карпатська, 15, м. Івано-Франківськ, 76019, vitoksmile@gmail.com*

При проектуванні інтернет-систем більшу частину розробки займає написання саме серверної частини проекту, адже там знаходиться основна логіка управління та маніпулювання даними. Розробка серверних частин веб-проектів є досить актуальним питанням на даний час, яке потребує постійного аналізу і вдосконалення.

Основна мережева архітектура взаємодії у інтернеті отримала назву “клієнт-сервер” [1]. У даній концепції беруть участь дві сторони: клієнт (замовник певної послуги) і сервер (постачальник цієї послуги).

Розробка серверної частини полягає у створенні певних публічних методів API (Application Programming Interface – прикладний програмний інтерфейс), до яких матимуть доступ клієнти системи. Клієнт за допомогою мережі інтернет звертається до певних методів, які надають певну інформацію користувачу чи вносять нові дані у систему.

Клієнтами системи виступають плати ESP8266 [2] мобільні додатки та веб-браузери, які за допомогою API звертаються до сервера та маніпулюють інформацією.

API – частина програмного забезпечення, яка дозволяє викликати методи низького рівня програмного забезпечення на високому рівні [3].

Розробка серверної частини розподіленої системи збору даних полягає у створенні певних публічних методів API, до яких матимуть доступ клієнти системи. Клієнт за допомогою мережі інтернет звертається до певних методів, які надають певну інформацію користувачу чи вносять нові дані у систему. На рисунку 1 наведено загальну структуру схеми взаємодії основних компонентів системи збору даних.

Серверна частина системи збору даних міститиме у собі Application Programming Interface (прикладний програмний інтерфейс), який розділятиме користувачів системи (клієнтів) від внутрішніх компонентів серверу (програмний код, база даних).

Система збору даних включає у себе інформацію про користувачів даної системи, канали, які включають у себе значення певних величин, які прив’язані до дат (наприклад, 13 °С, 4 жовтня 2017 року о 20:32, географічне розміщення каналів (довгота та широта), параметри, які використовуються у каналах (температура, вологість). Для доступу до даної інформації можна використовувати певні методи API, які поділимо на наступні групи методів:

- Авторизація – група, яка дозволяє авторизуватися у системі, вийти із системи (використовується для анулювання дійсності ключа авторизації), реєстрація користувачів у системі;
- Користувач – маніпулювання даними поточного користувача (редагування інформації про користувача, зміна паролю);
- Канали – група методів, яка дозволяє додавати нові канали, редагувати та видаляти існуючі, отримувати інформацію про створені канали поточного користувача, шукати публічні канали інших користувачів, та підписуватися

«Методи та засоби неруйнівного контролю промислового обладнання»

чи відписуватися на ці канали;

- Значення – методи, які дозволяють додати нове чи отримати доступні значення величин для певного каналу;

- Розташування – додавання нових чи редагування існуючих розташувань каналів, шукати доступні місця для створення каналу для даного місця;

- Параметри - додавання нових чи редагування існуючих параметрів каналів, шукати доступні параметри для створення каналу із цими параметрами.

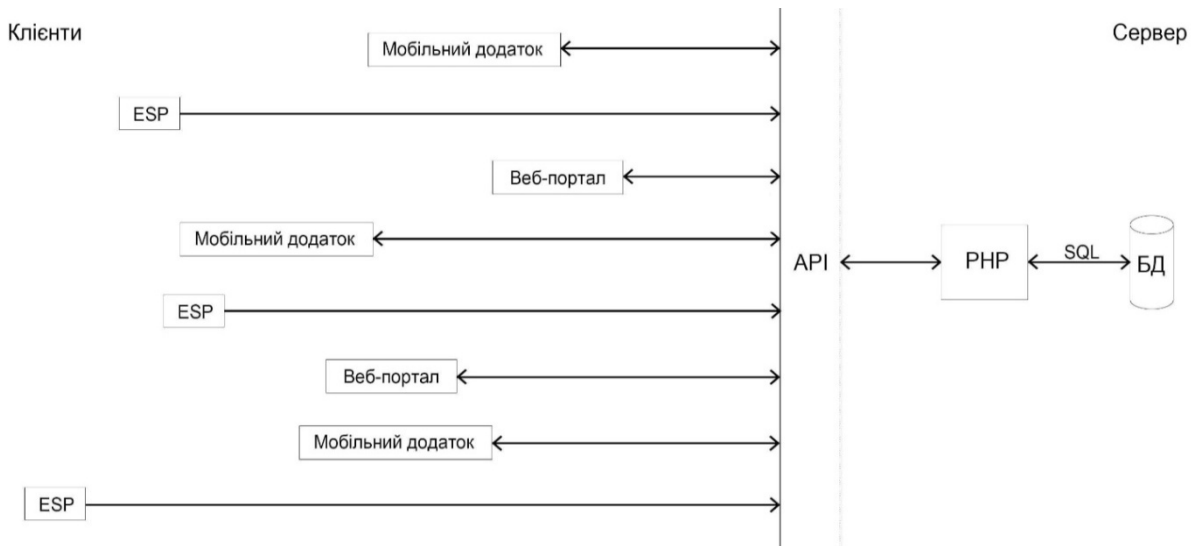


Рисунок 1 – Схема взаємодії компонентів системи збору даних

База даних містить у собі наступні таблиці:

- Користувачі системи (users);
- Ключі авторизації, які потрібні для виклику методів API (tokens);
- Місця розташування каналів (locations);
- Параметри, які використовуються у каналах (parameters);
- Канали (channels), які зв'язані із користувачем, що створив їх, параметром та місцем розташування;
- Підписки (subscriptions) – перелік публічних каналів інших користувачів, на які підписався певний користувач для спостережень за зміною значень цього каналу;
- Значення (datas), які прив'язані до певного каналу.

Розбиття вихідного коду проекту на декілька модулів підвищить надійність системи в цілому, адже при появі помилки у одному з модулів це не призведе до відмови у функціонуванні цілого проекту. Для гнучкої взаємодії із серверною частиною системи використовують API, який дозволяє розділити користувачів системи (клієнтів) від внутрішніх компонентів серверу (програмний код, база даних).

1. *Клиент-серверное взаимодействие [Электронный ресурс]. – Режим доступа: <http://4stud.info/networking/lecture5.html>.* 2. *ESP8266 [Электронный ресурс]. – Режим доступа: <https://en.wikipedia.org/wiki/ESP8266>.* 3. *Прикладний програмний інтерфейс [Электронный ресурс]. – Режим доступа: <https://goo.gl/ZubjLk>.*