



10. Dunlop G.R. and Jones T.P. Position analysis of a two DOF parallel mechanism- the Canterbury tracker. Mechanism and Machine Theory, May 1999. №34(4). P.599–614.
11. Su X.S. and others . Singularity analysis of fine-tuning Stewart platform for large radio telescope using genetic algorithm. Mechatronics, June 2003. №13(5) P.413–425.
12. Defendini A., Vaillon L., Trouve F., Rouse Th., Sanctorum B., Griseri G. Technology predevelopment for active control of vibration & very high accuracy pointing systems. Proceedings of 4th ESA International Conference on Spacecraft Guidance, Navigation and Control Systems (Proceedings to appear in spring 2000). 18–21 October 1999. Noordwijk: The Netherlands. 7 p.
13. Aliyev F.A., Larin V.B., Naumenko K.I., Suntsev V.N. Optimizatsiya lineynykh invariantnykh vo vremeni sistem upravleniya: Monografiya. In-t matematiki AN USSR.- K.: Nauk. dumka, 1978.- 327s.

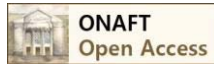
УДК 519.85:004.42

## АРХІТЕКТУРА КЛІЄНТ-СЕРВЕР НА ОСНОВІ ДОДАТКА ВІДДІЛУ АСПІРАНТУРИ ТА ДОКТОРАНТУРИ ОНАХТ

Д.О. Скалій<sup>1</sup>, М.В. Джиджула<sup>2</sup>, Ю.К. Корнієнко<sup>3</sup>, О.С.Бойцова<sup>4</sup>, О.О.Лівенцова<sup>5</sup>

<sup>1,2,3,4,5</sup>Одеська національна академія харчових технологій, Одеса, Україна  
ORCID: <sup>3</sup>0000-0003-3630-8384, <sup>4</sup>0000-0001-9994-587X, <sup>5</sup>0000-0001-7409-874X  
Scopus ID: <sup>3</sup>256578764800, <sup>5</sup>6507262177  
E-mail: <sup>1</sup>skaliy.d@gmail.com, <sup>3</sup>yurikkorn@gmail.com, <sup>4</sup>ola\_odessa@ukr.net

Copyright © 2018 by author and the journal “Automation technologies and business - processes.  
This work is licensed under the Creative Commons Attribution International License (CC BY).  
<http://creativecommons.org/licenses/by/4.0>



DOI: 10.15673/atbp.v10i2.974

*Веб-технології постійно розвиваються, відкриваються нові можливості створення сценарію іншими підходами. Веб-додаток можна розробляти як самостійно – full stack, або по частинах – front end (клієнтська частина) та back end (серверна частина) [1]. Клієнтська частина надає користувачеві візуальної інформації, а серверна частина обробляє, зберігає, захищає дані. Сьогодні для створення веб-серверу для веб-додатка є підхід до архітектури мережевих протоколів під назвою REST.*

*Веб-додаток, побудований з урахуванням REST, приймає термін RESTful. В такому веб-додатку проєктують спосіб з'єднання клієнтської та серверної частин за допомогою прикладного програмного інтерфейсу (API). Дані з серверної частини формуються посиланням необхідного HTTP-запиту і передаються у вигляді об'єкта, який клієнтська частина приймає та відображає у браузері користувача.*

*Веб-додаток Відділу аспірантури та докторантури ОНАХТ, розроблений для автоматизації процесів роботи, був заснований на архітектурі клієнт-сервер, де серверна частина розроблена саме як RESTful API, а для зниження навантаження на з'єднання з сервером та покращення роботи з користувачем клієнтська частина розроблена як односторінковий додаток (SPA).*

**Ключові слова:** веб, веб-додаток, REST, API, сервер, архітектура клієнт-сервер, односторінковий додаток, клієнтська частина, серверна частина

*Web technologies are constantly evolving, opening up new possibilities of creating a scenario by other approaches. The web application can be developed either on its own - full stack, or in parts - front end (client part) and back end (server part) [1]. The client part provides the user with visual information, and the server part processes, stores, protects the data. Today, in order to create a Web server for a Web application, there is an approach to the network protocols architecture called REST.*



*The REST-based web application accepts the term RESTful. In this web application, a method for connecting client and server parts using the Application Program Interface (API) is designed. Data from the server part is generated by the link of the required HTTP request and transmitted as an object that the client part receives and displays in the user's browser.*

*The online application of the Postgraduate and Doctoral Studies Department of ONAFT, designed to automate work processes, was based on client-server architecture, where the server part was developed exactly as the RESTful API, and to reduce the load on the connection to the server and improve the user experience, the client part was developed as a one-stop application (SPA).*

Keywords: web, web application, REST, API, server, client server architecture, one-page application, client part, server part

---

## 1. Вступ

Інформаційні технології в сучасному світі здатні значно вплинути на будь-яку область та об'єкти діяльності, полегшити процес пошуку інформації та підвищити рівень освіти. В століття розвитку інформаційних технологій їх неможливо не використовувати, особливо в таких областях, як освіта і форми підвищення кваліфікації осіб. Відтак, для комфорту в цій області потрібна автоматизація процесу, яка заснована на інформаційних технологіях, та система управління, яка б охоплювала діяльність відділу аспірантури та докторантури, а саме – інформаційно-управляюча система документообігу і моніторинг діяльності, вбудовані в веб-додаток.

В основному завдання діяльності відділів аспірантури і докторантури і вирішуються спеціалізованими інститутами, вченими та/або науковими радами та інформаційними службами. Освітня складова діяльності відділу включає чотири компоненти: здобуття глибоких знань із обраної спеціальності; оволодіння загальнонауковими (філософськими) компетентностями; набуття універсальних навичок дослідника та педагогічної майстерності; здобуття мовних компетентностей, в свою чергу наукова складова передбачає проведення власного наукового дослідження під керівництвом одного або двох наукових керівників та оформлення його результатів у вигляді дисертації. Підготовка здобувачів вищої освіти ступеня доктора філософії та доктора наук провадиться згідно правил, визначених Постановою Кабінету Міністрів України № 261 від 23.03.2016 р. «Про порядок підготовки здобувачів вищої освіти ступеня доктора філософії та доктора наук у вищих навчальних закладах (наукових установах)».

Відповідно до своїх функціональних обов'язків співробітники відділу: надають консультації з питань вступу до аспірантури, докторантури та прикріплення здобувачами університету; організують процес замовлення університетом державних місць в аспірантурі та докторантурі; проводять прийом документів вступників до аспірантури, докторантури та здобувачів; координують роботу екзаменаційної сесії; видають накази про зміни в контингенті аспірантів, докторантів та здобувачів; надають необхідні документи після закінчення аспірантури, докторантури та прикріплення здобувачем.

Метою і завданням розробки було створити інформаційну управляючу систему, яка забезпечує програмну підтримку у використанні журналу завдань та електронних документів відділом аспірантури і докторантури Одеської національної академії харчових технологій. Завдання потребує проаналізувати основні проблеми документообігу відділу, провести аналіз існуючих аналогів та обґрунтування засобів реалізації, а також виявити ряд проблем у сфері моніторингу діяльності здобувачів наукового ступеня.

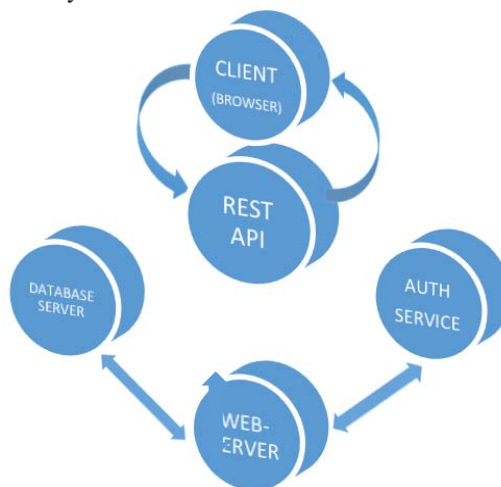


Рисунок 1 – Рівні веб-додатка



Для відділу аспірантури та докторантури потрібен якісно спроектований ресурс для документообігу та моніторингу діяльності, на якому би зручно відображалися ресурси та інформація про розділи в системі підвищення кваліфікації осіб та, крім цього, керування завданнями і роботою, ведення електронного журналу, контролювання електронного документообігу – завантаження-вивантаження, обмін, пошук, друк. Такий варіант використання системи є широко поширеним і зручним.

Розроблене програмне забезпечення для відділу аспірантури і докторантури має класифікацію клієнт-серверного додатку з трирівневою архітектурою: клієнт, сервер додатків та сервер бази даних. Кожний з цих рівнів має свою роль. Клієнт є інтерфейсом програми, який надається користувачеві через веб-браузер. Сервер додатків відповідає за зв'язок з мережею, клієнтською частиною, сервером базою даних та захист інформації. Клієнт надсилає до сервера запит, який той обробляє і надсилає запит до бази даних для отримання та відправлення назад потрібної інформації.

## 2.Архітектура «клієнт-сервер»

Клієнт-серверна архітектура передбачає сервер, який надає інформацію клієнту; клієнт, який запрошує і використовує інформацію з сервера; мережу, яка забезпечує взаємодію між клієнтом та сервером [2]. Завдяки технології побудови веб-додатків SPA та REST архітектури, сервер і клієнт можуть бути настільки незалежними один від одного, наскільки це можливо.

Серверна сторона програми в основному забезпечує зберігання і генерацію відображення даних для кінцевого користувача. На ній знаходяться база даних для збереження інформації необхідної для роботи додатку.

Клієнтська сторона програми містить маніпуляції з даними, навігаційні служби та будь-які інші модулі, необхідні для роботи клієнтської сторони програми. Доступ до даних відбувається за допомогою запитів до REST API. Навігація - тут в основному міститься бізнес логіка додатку, прописані відповідні дії додатку на дії які виконує користувач. Також на клієнтській частині містяться сторінки, які будуть відображатись відповідно до запитів користувача.

Під час роботи системи користувач використовує клієнтську програму, яка відправляє запити на сервер, який обробляє ці запити відповідно до бізнес-логіки програми та відправляє/приймає оброблені дані на сервер з базою даних. Авторизація здійснюється за допомогою сервісу авторизації і взаємодіє з клієнтською і серверною частиною програми.

SPA (single-page application) це односторінковий додаток, який взаємодіє з користувачем, динамічно переписуючи поточну сторінку, тобто такий додаток буквально має одну сторінку і використовує тільки її, а не завантажує нові сторінки [3]. Цей підхід забезпечує роботу користувача між послідовними сторінками без переривань.

REST (Representational State Transfer) являє собою архітектурний стиль, який визначає набір обмежень і властивостей на основі HTTP [4]. RESTful сервіси, які відповідають стилю REST, дозволяють запитувачим системам отримувати доступ до текстуальних уявлень і керувати ними, використовуючи єдиний та попередньо визначений набір операцій. HTTP запит (GET, POST, PUT, DELETE тощо), зроблений на URI ресурсу, викликає відповідь, яка може бути у вигляді HTML, XML, JSON або іншому вигляді. Системи REST прагнуть до швидкої продуктивності, надійності та здатності зростати.

API (Application Programming Interface) (прикладний програмний інтерфейс) це набір чітко визначених способів зв'язку між різними програмними компонентами [5]. Хороший API спрощує розробку комп'ютерної програми, надаючи всі будівельні блоки, які потім об'єднуються програмістом. API може бути для веб-системи, операційної системи, системи баз даних, комп'ютерного обладнання або бібліотеки програмного забезпечення. Оскільки графічний інтерфейс користувача спрощує використання програм, інтерфейси прикладного програмування допомагають розробникам використовувати певні технології для створення додатків. Абстракуючи базову реалізацію і лише піддаючи дії розробника необхідним об'єктам або діям, API спрощує програмування.

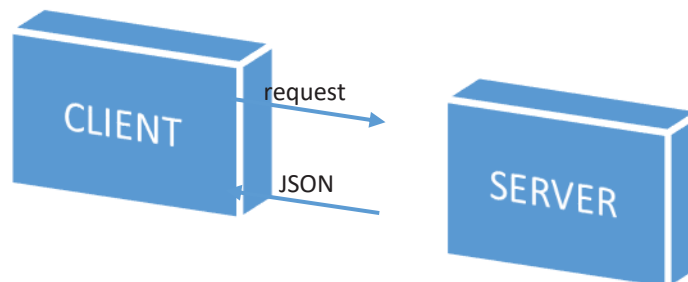


Рисунок 2 – Запит та отримання даних клієнтом від сервера



### 3. Розроблений додаток

Додаток відділу аспірантури та докторантури розроблявся виключно для потреб відділу, він досить складний з технічного боку, досить легкий з боку використання, має великий функціонал та використовує велику кількість даних. Для відділу є актуальним якісний моніторинг діяльності та електронний документообіг здобувачів наукового ступеня. Моніторинг діяльності може проявлятися у зборі, обробці та збереженні, а також розповсюдженні інформації щодо діяльності науковця, орієнтованої на забезпечення управління, яке дозволяє оцінювати стан справ у будь-який момент часу та може забезпечувати прогноз його розвитку. До задач моніторингу у системі відноситься представлення об'єктивних та достовірних даних про хід науково-дослідного процесу та його результатів.

При розробці веб-додатка поставало завдання створити інформаційну управляючу систему, яка забезпечує програмну підтримку у використанні журналу завдань та електронних документів відділом аспірантури і докторантури Одеської національної академії харчових технологій, вирішення проблеми із зменшення паперового документообігу та моніторингу діяльності здобувачів наукового ступеня. На практиці це програмне забезпечення повинно не тільки представляти собою гнучкий та зручний засіб для моніторингу діяльності, а також мати можливість зберігати їх наукові праці, відстежувати та фіксувати всі етапи діяльності, представляти статистичні дані, тощо.

Додаток має надійний захист. Доступ до будь-якої інформації має тільки авторизований користувач, якого зареєстрував один із адміністраторів. Усі запити до URI ресурсу проходять перевірку прав доступу користувача. При першому відвідуванні ресурсу відображається форма авторизації.

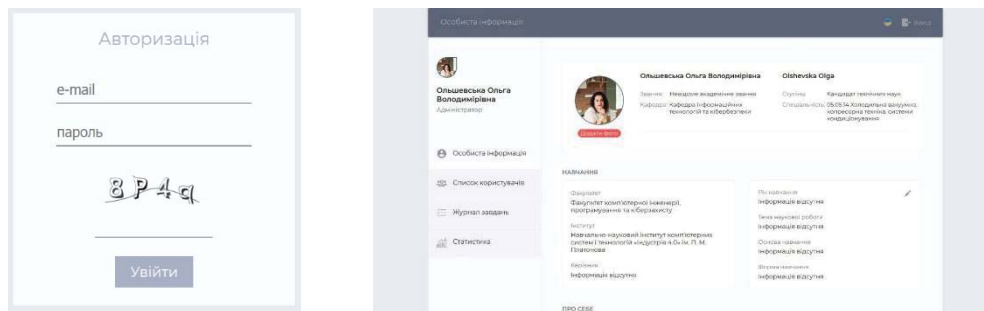


Рисунок 3 – Процес отримання доступу до ресурсу

### Висновок

Більшість програм нині доступні в Інтернеті у вигляді веб-додатків. Це позбавляє користувача від необхідності встановлювати програми на робочому столі, додаток завжди буде доступний в Інтернеті.

Результатом розробки є програмний продукт, який має впроваджувати доступ до можливостей моніторингу діяльності роботи співробітника ОНАХТ. Розроблений програмний продукт впроваджено у робочий процес ОНАХТ.

### Література

- [1] Веб-застосунок [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Веб-застосунок>  
 [2] Клієнт-серверна архітектура [Електронний ресурс] – Режим доступу до ресурсу: [https://uk.wikipedia.org/wiki/Клієнт-серверна\\_архітектура](https://uk.wikipedia.org/wiki/Клієнт-серверна_архітектура)  
 [3] Односторінковий застосунок [Електронний ресурс] – Режим доступу до ресурсу: [https://uk.wikipedia.org/wiki/Односторінковий\\_застосунок](https://uk.wikipedia.org/wiki/Односторінковий_застосунок)  
 [4] REST [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/REST>  
 [5] Application programming interface [Електронний ресурс] – Режим доступу до ресурсу: [https://en.wikipedia.org/wiki/Application\\_programming\\_interface](https://en.wikipedia.org/wiki/Application_programming_interface)

### References

- [1] “Web application,” *Wikipedia*, 28-Jan-2018. [Online]. Available: [https://en.wikipedia.org/wiki/Web\\_application](https://en.wikipedia.org/wiki/Web_application). [Accessed: 30-Jan-2018].  
 [2] “Client–server model,” *Wikipedia*, 29-Mar-2018. [Online]. Available: [https://en.wikipedia.org/wiki/Client–server\\_model](https://en.wikipedia.org/wiki/Client–server_model). [Accessed: 30-Apr-2018].  
 [3] “Single-page application,” *Wikipedia*, 30-Jun-2017. [Online]. Available: [https://en.wikipedia.org/wiki/Single-page\\_application](https://en.wikipedia.org/wiki/Single-page_application). [Accessed: 30-Jun-2017].  
 [4] “Representational state transfer,” *Wikipedia*, 12-Jul-2017. [Online]. Available: [https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer). [Accessed: 30-Jan-2018].  
 [5] “Application programming interface,” *Wikipedia*, 9-Jan-2018. [Online]. Available: [https://en.wikipedia.org/wiki/Application\\_programming\\_interface](https://en.wikipedia.org/wiki/Application_programming_interface). [Accessed: 10-Jan-2018].