

UDC 004.414.28

Dmytro Korenko, Andrii Boldak**APPROACH TO ORGANIZATION OF CLIENT-SERVER INTERACTION
FOR IMPLEMENTATION OF MODEL-VIEW-CONTROLLER PATTERN
IN DISTRIBUTED SYSTEMS**

The article considers the approach to organization of client-server interaction. This approach is used to implement the template model-view-controller in distributed systems. New concept organization WEB API implements ETL technology and allows you to move the controller functionality to the server side.

Keywords: client-server interaction, distributed systems, MVC, ETL technology.

Fig.: 1. Tabl.: 2.

У статті розглядається підхід до організації клієнт-серверної взаємодії. Даний підхід використовується для реалізації шаблону model-view-controller в розподілених системах. Нова концепція організації WEB API реалізує ETL-технологію та дає можливість перенести функціональність контролера на серверну сторону.

Ключові слова: клієнт-серверна взаємодія, розподілені системи, MVC, ETL-технологія,.

Fig.: 1. Fig.: 2.

Relevance of research topic. Modern WEB-applications are developed in accordance with the approach of SPA [1], first of all, those that apply processing large volumes of information, put forward fairly stringent requirements for the quality of communication channels. This is due to the fact that the generally accepted concepts of WEB API organization [2] are focused on the implementation of ELT-technology [3], the disadvantage of which is the need to transfer large volumes of intermediate data between a server whose API usually implements the CRUD-interface [4] model, and a client that implements a controller and a view as a MVVM template [5]. Studies related to the development of new concepts for the organization of the WEB API aimed at reducing the amount of intermediate data transmitted between the server and the client are relevant.

Target setting. The problem lies in the lack of concepts for organizing WEB APIs that implement ETL technology.

Actual scientific researches and issues analysis. Today, the generally accepted concepts of the WEB API organization are REST[6][7] and GraphQL[8], the essence of which is to implement the CRUD-interface on the server side.

Uninvestigated parts of general matters defining. In this paper, we explore a new concept for the organization of the WEB API, which implements ETL technology and enables the functionality of the controller to be transferred to the server side.

The research objective. The task is to develop a new concept of client-server interaction, which involves transferring to the server side and executing on it a script that defines the necessary actions related to the implementation of the functionality of the controller.

The statement of basic materials. The proposed concept of WEB API organization is to use the POST request, which transmits the data structure (script), which defines the sequence of calls implemented on the server side methods (commands) (see fig.1.a). On the server side, there should be an interpreter (similar to an orchestrator in GraphQL) that implements a sequence of calls in a single scope of script context, as depicted in fig.1.b).

It is clear that the composition of the teams that are interpreted on the server side is determined by the purpose of the service. In terms of the functional purpose associated with the processing of data, in our view, the system of commands should include:

- data definition and data manipulation commands with data loading from external sources;
- collection filtering, collection joining, collection mapping and collection reducing commands;
- statistic commands such as calculation of statistics, principal component analysis, clustering etc.

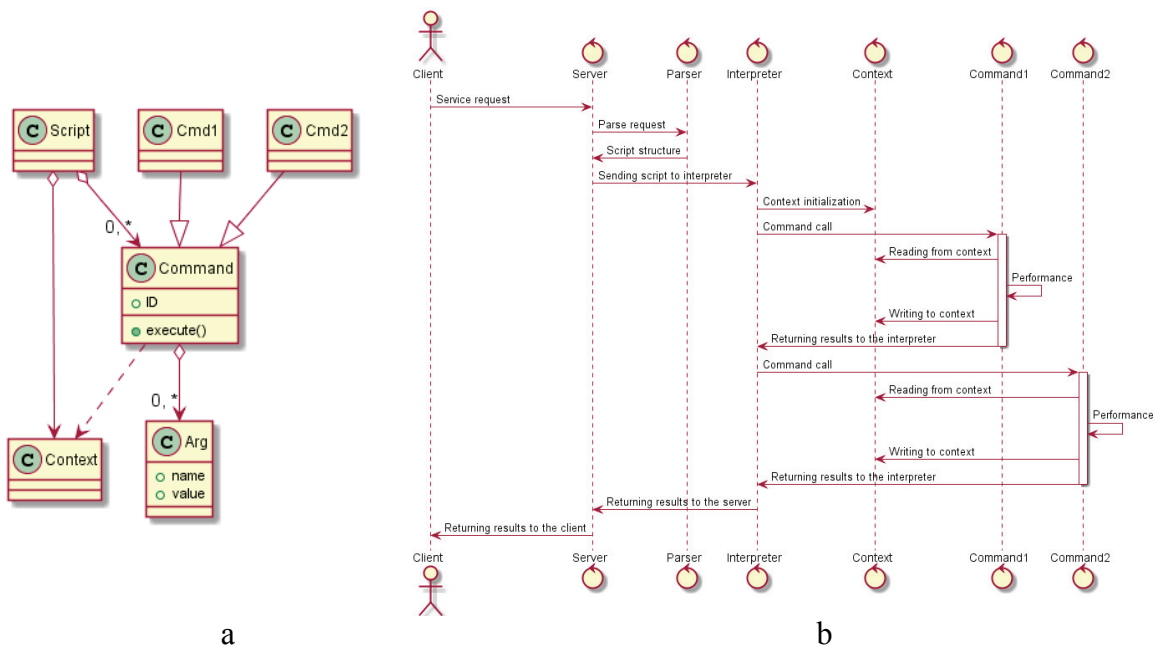


Fig.1. Script based WEB API: a. structure of script; b. request life cycle

The basis of the integral integrity of the distributed system is the implementation of the unified interface subsystems access to its resources. Remote Assignment Tools (RPCs) implemented at the level of underlying assets allow the delegation of data

processing to another resource that supports this protocol. In this case, the integration is implemented on the server side (fig.2.a), which can act as a proxy.

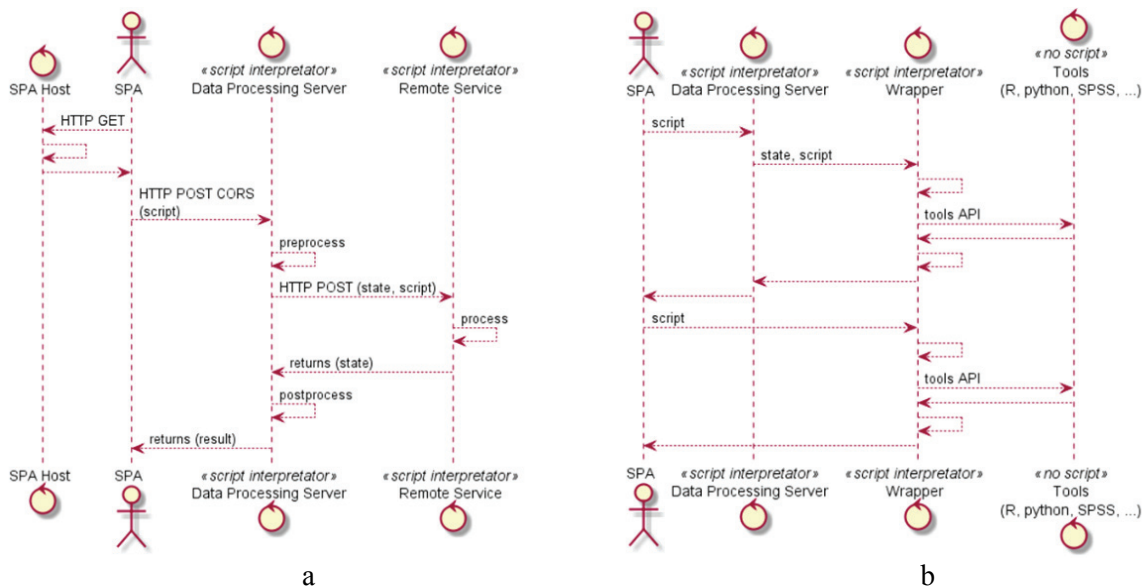


Fig.2. Integration of system resources: a. with RPC and CORS; b. with wrappers

Using Cross Origin Resource Sharing (CORS) [9] lies in the integration of resources such as one-page portal and data processing services, and enables the integration of client-side resources.

In case when there is a need to use resources that do not implement the above interface, it is advisable to use adapters (wrappers). This shell, as shown in fig.2.b, implements a unified interface for the subsystem and uses resource-specific software interfaces.

Thus, the proposed model implements a flexible scheme for organizing the interaction of various resources, including heterogeneous, both on the server side and on the client side.

Conclusions. The proposed concept of organization of the WEB API, in contrast to the known, such as REST and GraphQL, allows, in the framework of the SPA approach, to reduce the amount of intermediate data transferred between the server and the client by moving the implementation of actions related to the implementation of the controller on the server side. The implementation of such a concept involves the presence on the server side of a special software script interpretation. This approach to the organization of the WEB API has been successfully applied to the development of the data processing service [10][12], which is part of a distributed data processing system used by the World Data Center for Geoinformatics and Sustainable Development [11][13].

References

1. David Flanagan (2006). *JavaScript - The Definitive Guide*, 5th ed., O'Reilly, Sebastopol, CA, p.497.
2. E. Michael Maximilien, Ajith Ranabahu, Karthik Gomadam (Sep–Oct 2008). *An Online Platform for Web APIs and Service Mashups*. In Proceedings of the IEEE conference on computer vision and pattern recognition (Volume: 12 , Issue: 5).
3. Clive Skinner. (Jan 2018). *Using Redshift Spectrum to load data pipelines*. URL: <https://www.dativa.com/using-amazon-redshift-spectrum-data-pipelines> (Access time: 12.04.2019)
4. James Martin (1983). *Managing the Data-base Environment*. Englewood Cliffs, New Jersey: Prentice-Hall. p. 381. ISBN 0-135-50582-8.
5. John Gossman. "Introduction to Model/View/ViewModel pattern for building WPF apps ". URL:<https://blogs.msdn.microsoft.com/johngossman/2005/10/08/introduction-to-modelviewviewmodel-pattern-for-building-wpf-apps> (Access time: 08.04.2019)
6. Fielding, Roy (June 2014). *"Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content, Section 4"*. IETF. Internet Engineering Task Force (IETF). RFC 7231. Retrieved 2018-02-14.
7. Fielding, Roy (June 2014). *"Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content, Section 4"*. IETF. Internet Engineering Task Force (IETF). RFC 7231. Retrieved 2018-02-14.
8. Lee Byron (SEP 2015). *GraphQL: A data query language*. URL: <https://code.fb.com/core-data/graphql-a-data-query-language> (Access time: 08.04.2019)
9. Rick Anderson (2019). *Enable Cross-Origin Requests (CORS) in ASP.NET Core*. URL: <https://docs.microsoft.com/en-us/aspnet/core/security/cors?view=aspnetcore-2.2> (Access time: 12.04.2019)
10. Andrii Boldak (Dec 2017). *Basic DJ Data Processing Server*. URL: <https://github.com/boldak/dj-dps-server> (Access time: 18.04.2019)
11. *World Data Center for Geoinformatics and Sustainable Development*. URL: <http://wdc.org.ua/> (Access time: 18.04.2019)
12. Власов М. Д., Болдак А. О. Оптимізація конфігурацій розподілених інформаційних систем URL: http://sait.kpi.ua/media/filer_public/6e/80/6e804b3f-ae13-4899-b336-4daddbd45584/sait2018ebook.pdf (Access time: 18.04.2019)
13. Єфремов К. В., Болдак А. О. Предметно-орієнтована мова аналітичної обробки даних. Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. пр. – К.: Век+, – 2012. – № 55. - с. 50-55.– 212 с

Autors

Boldak Andrii – Candidate of Technical Sciences, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: boldak.andrey@gmail.com

Болдак Андрій Олександрович – кандидат технічних наук, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

Korenko Dmytro – Student, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: korenko.dima98@gmail.com

Коренко Дмитро Володимирович – студент, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

РОЗШИРЕНА АНОТАЦІЯ

А. О. Болдак, Д. В. Коренко

ПІДХІД ДО ОРГАНІЗАЦІЇ КЛІЄНТ-СЕРВЕРНОЇ ВЗАЄМОДІЇ ДЛЯ РЕАЛІЗАЦІЇ ШАБЛОНУ MODEL-VIEW-CONTROLLER В РОЗПОДІЛЕНИХ СИСТЕМАХ

Актуальність теми дослідження. Сучасні WEB-застосунки розроблені у відповідності до підходу SPA, в першу чергу ті, що застосовують опрацювання великих обсягів інформації, висувають доволі жорсткі вимоги до якості каналів зв'язку. Це пов'язано з тим, що загально прийняті концепції організації WEB API є орієнтованими на реалізацію ETL-технології, недоліком якої є необхідність передачі великих обсягів проміжних даних між сервером, API якого, зазвичай, реалізує CRUD-інтерфейс моделі, та клієнтом, який реалізує контролер та зовнішній вигляд як шаблон MVVM.

Постановка проблеми. Проблема полягає у відсутності концепцій організації WEB API, які реалізують ETL-технологію.

Аналіз останніх досліджень і публікацій. Сьогодні загальноприйнятими концепціями організації WEB API є REST та GraphQL, сутність яких полягає у реалізації CRUD-інтерфейсу на серверній стороні.

Виділення недосліджених частин загальної проблеми. В цій роботі досліджуються нова концепція організації WEB API, яка реалізує ETL-технологію та дає можливість перенести функціональність контролера на серверну сторону.

Постановка завдання. Завдання полягає в розробці нової концепції клієнт-серверної взаємодії, яка передбачає передачу на серверну сторону та виконання на ній скрипту, що визначає необхідні дії, пов'язані з реалізацією функціональності контролера.

Викладення основного матеріалу. Запропонована концепція організації WEB API полягає у використанні POST-запиту, в якому передається структура даних (скрипт), що визначає послідовність викликів реалізованих на серверній стороні методів (команд). На серверній стороні повинен бути інтерпретатор (схожий на оркестратора в GraphQL), який реалізує послідовність викликів в єдиній області видимості змінних.

Висновки. Запропонована концепція організації WEB API, на відміну від відомих, дозволяє в рамках підходу SPA зменшити об'єм переданих між сервером і клієнтом проміжних даних за рахунок переміщення виконання дій, пов'язаних з реалізацією контролера на серверну сторону.

Ключові слова: клієнт-серверна взаємодія, розподілені системи, MVC, ETL-технологія.