UDC 004.418

S.A. Kudrenko, PhD
(National Aviation University, Ukraine)

Automated Data Analysis System using React.js and GraphQL

This paper discusses frameworks and technologies for creating streamlined data analysis system. The selection of technology, library, architecture and layouts depends on the solved problems and their complexity are considered.

The general superiority of the automation will be that it allows you to reduce the size of sufficient memory, shorten the time for finishing provided and reduce the abundance of documents when updating information. Also, that touches the benefits, it allows you to find exciting information in a short time. The question for automated parsing of information technologies provided on the market is growing, which means that the supply is also growing.

The design of the data parsing provided is to ensure the differentiation of the information provided in relation to the role of the user, to exist stable against errors and negative input data.

To solve the problems of client application, the following technologies were preferred:

- React library for working with UI rendering [1];
- The GraphQL library as a library for making requests to the back end of a web application and creating an architecture for processing internal actions in the client application.

The next tasks of the server application when implementing the layout of a single-page web application are:

- coordination with the database;
- processing of subscriber addition requests and creation of application business logic.

The next technologies of the server application when implementing the layout of a single-page web application are:

- Node.js - a platform that allows JavaScript to be executed on the server and provides opportunities for interacting with the server's system resources;
- Express.js - a framework that allows to implement a web server, routing user requests, and decompose program logic into separate subsystems.

The coordination between the infobase and the web server is organized of the principle that the pattern is unanimously located in the code of the server application, and not in the data warehouse.

It is here that the base of the provided preserves the materials and guarantees an unconditional path to the data, and all the logic is implemented in the subsystems of the server application. The infobase allows transactions for the execution of atomic actions with data. One of the more monolithic automation of parsing provided - the creation of a single repository of information. The existence of a single center for storing information for the sake of minimizing the functions of separate users is no less a majestic problem in the creation of automated systems.

In addition, requirements for the developed organization seems to be the creation of a comfortable user interface, not overloaded with useless information and functionality.

It provides lightweight perception and processing of information, providing an unobstructed recipe for information in a more common form for enterprise employees, such as a document format.

The selection of technologies for the development of a system is a significant moment of the work. Correctly tucked together the intricacies of technologies must guarantee a convenient service at some time at all stages of the presence of the application:

- support convenience;
- scalability.

The choice of a technology stack must be approached infinitely intently and sensitively. Should look far into the future and predict the likely formation and fate of the project. The store is forced to exist freely scalable, functional, in line with the final trends of the market. It must be consistent with the most current characteristics. The most important thing, so that in the future it can be freely held by other developers. The existence in society of a large society of designers of such a product in the stack and an open primitive program are huge advantages that should be taken care of.

As follows from the above factors, and listening to the views of the main designers of the market of the web technology, the incoming technology stack was highlighted:

- React.js library;
- GraphQL library;
- Webpack - the utility is good for managing the modular construction of a interlacing operation;
- npm package is great for managing modules and dependencies.

React.js has a capacious and understandable Application Programming Interface (API). To exertion with React, it is necessary to understand a number of terms and the differences between them. Elements are JavaScript objects, whichelements are represented by HTML. Components are React.js elements., which are generated by the developer and can have any name. In most cases, they contain their environment and carry out various functions. React elements and components are constructed using JSX. JSX is a deep JavaScript syntax similar to XML [1].

React makes an analog of the real Document Object Model (DOM) tree of the document object model from components - VirtualDOM and presents it in the browser. The library tracks changes in the virtual tree and, if it changes, updates the real DOM, for the real and virtual trees were the same (fig.). The system will execute the application state through GraphQL. React and GraphQL are a pretty good match.

Components have life cycles such as mounting, updating, and unmounting. The library makes it possible to define different points in the life cycle of components and interact with them.

When you first use the component, call the lifecycle methods in the following order:

- constructor;
- getDerivedStateFromProps;
- render;
- componentDidMount.

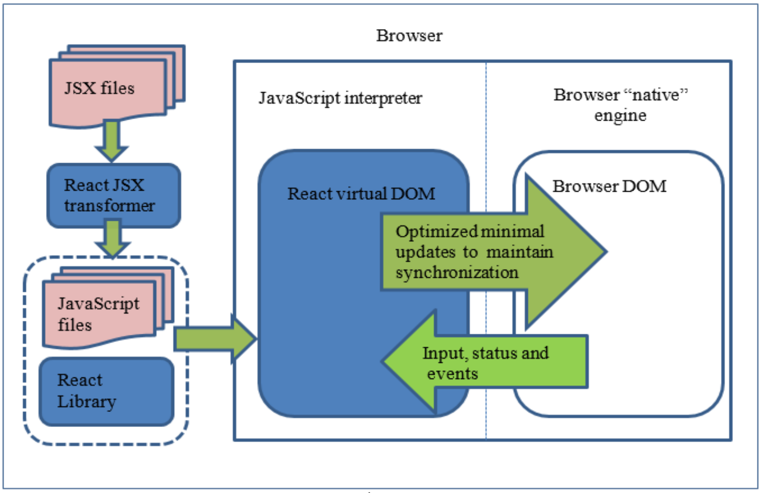

Fig. React functioning

GraphQL is a query and data manipulation speech for APIs [2-3], as well as an environment for making those queries. The speech was developed in two thousand-twelfth by Facebook for the internal needs of the company. In two thousand-fifteenth it was released to the public, and since November seven, two thousand eighteen, it's not Facebook that's been working on it, but the GraphQL Foundation.

System development using these applications was accessible and well-off. Their features are in correctness adequate to fabricate in correctness data analyzable system.

References

1. Johnson, Nicholas."Introduction to Flux - React Exercise".Nicholas Johnson. Retrieved 7 April 2018. Web-site: http://nicholasjohnson.com/react/course/exercises/flux/

2. G. Brito, T. Mombach, and M. T. Valente, “Migrating to GraphQL: A practical assessment,” in 26th International Conference on Software Analysis, Evolution and Reengineering (SANER), 2019, pp. 140–150.

3. E. Wittern, A. Cha, and J. A. Laredo, “Generating GraphQL-Wrappers for REST (-like) APIs,” in International Conference on Web Engineering, 2018, pp. 65–83.