УДК 519.713
Vitalii Bilyi[1], Svitlana Voitenko[2]
[1]PhD student, CST Dpt., NU "Zaporizhzhia Polytechnic"
[2]senior teacher NU «Zaporizhzhia Polytechnic»

MODERN METHODS OF WEB-SERVICES DESIGN AND DEVELOPING

Modern web-services operate according to the principles of "client-server" architecture, where the browser acts as the client, and the web-server acts as the server. The peculiarity of such approach is that the application itself is located and executed on the server - the client receives only the results of the work. The main task of the client is correct displaying of the received information, as well as sending requests and user data to the server. Transfering of the requests and results of their processing are going through the Internet.

According to the development methods, web-services are divided into two main types: single-page (SPA) and multi-page (MPA). Let's consider the differences between them.

"Single-page" services allow you to imitate the work of desktop programs, as they respond to user actions without delay. The concept of SPA is that all resources which are required for work (CSS elements, scripts, etc.) are located on one page and loaded during initialization. With such approach, there is no need to reload or load additional pages - the content is updated dynamically using JavaScript (AJAX requests).

"Multi-page" services work according to the traditional scheme. This means that when the data changes slightly or when you go to another page, the browser makes a new request to the server and re-loads all resources, even those components that are repeated on every page (header, footer, etc.). Thus, performance is wasted loading the same elements.

Summing up, we can conclude that each architecture has its advantages for certain types of projects. So, the MPA approach is well suited for those cases when you need to display a large amount of content, for example, when creating an online store, business site, catalog. However, due to the higher work speed, the SPA concept is gradually replacing the traditional approach, in which the page is reloaded after each user action.

Developing of the web-services from scratch requires a lot of work. At the same time, in most cases, you have to spend time reproducing functions that have already been performed thousands of times. Frameworks help to deal with this problem by providing a kind of skeleton for the future application and a set of additional tools designed to speed up the product creation.

When developing a web-service based on the SPA concept, which involves a clear separation of responsibilities between the external presentation (front-end) and internal implementation (back-end), you will need to choose an individual framework for both the server and client parts.

The most popular frameworks for the server side at the moment are: Django, Laravel, Ruby on Rails. For the client side: Vue.js, React, Angular.

In web-services built on the SPA architecture, the server side is limited to the backend through some API (Application Programming Interface) entry points. The client sends requests to these endpoints and the server returns a response. The most common approach of API interfaces design is REST (Representational State Transfer). It is not a standard, but it defines limitations such as statelessness, client-server communication, and a unified interface. With such approach, the body of the request and response from the server is represented as JSON object.

According to the architectural principles of the REST, there are at least 4 HTTP methods corresponding to basic operations on entities: loading data (GET), saving (POST), editing (PUT/PATCH) and deleting (DELETE). This list is also accompanied by such operations as handling errors in the request, delimiting access and validating input data.

In any web-service the database has the key role. Therefore, you should spend enough time designing its structure, especially if there is a need to work with large amounts of information. When working with relational databases, normalization mechanisms can help you with it.

Normalization consists in transformation of the data storage structure to normal forms. Normal form is a set of requirements for the structure of tables to ensure the database integrity and eliminate excessive functional dependencies. Usually, when designing a database, it is sufficient to transform it to the first three normal forms: first normal form: each table must have a primary key; each attribute should consist only of elementary (indivisible) values and not contain repeating groups (atomicity); the second normal form: data that repeatedly appear in several rows should be stored in separate tables; third normal form: data should not be stored in a table if it can be obtained from non-key fields.