Aleksandra Sagitova

From concept till prototype

https://github.com/aleksa0809/Business-from-scratch/pull/new/january2021

This task demonstrates online shop (web application) with a dynamic user interface.

The submitted work is continued from the same source code project (repo). It comes from the brach proof_of_concept. The REST API is used in the project. The special server provides data about goods according with user's requests. The API is described in app.md file. The files that do not contain source code of the project are listed in .gitignore file (node_modules, build and so on).

There is asynchronous function for fetching data from special REST server. The application retrieves and displays information for the user that is retrieved via calls to an external REST API (the Showcase component). The application sends information created from interaction with the user (links of the LeftMenu) to an external REST API. The communication to external Web APIs in the application is delimited from the application's UI: web requests located in the Context component that has no UI and affects on visible components via state API.

Documentation

Name and briefly describe 3 different places for data storage that are available in a web application on the client side.

There are some abilities for storing data on client side:

1. localStorage – simple key/value storage where data can be saves as strings between user sessions;
2. IndexedDB another variant of key/value storage;
3. cookies - traditional way of saving information that can be transmitted between client and server.

What is important to consider when choosing a data storage location from a developer's perspective?

It is important to consider choosing a data storage security, simplicity, ability to work on different browsers. In the application local storage is used as a place where users basket can be saved.

Information about the basket does not leave client's side, so the necessary security level is achieved. For transformation the basket from and to javascript objects json format is used.

When we make changes to our application, what is important to keep in mind if we used the browser address (URL) as a data storage location for the client?

There are insecure ways to store client-server related information in browser. If user inputs a password into a form and the form data are sent via GET method, the data can appear in the browser’s address bar.

Reflection

What design patterns / structures have you used in the code to separate the application's UI from the communication with REST APIs in the source code, why is this a suitable structure?

The REST API is a common solution for web applications.

It is useful because there is ability to separate data structure and user’s dependent components (UI). This approach allows to a programmer to concentrate on abstract scheme of application, to increase its stability and scalability (robustness). UI is important but not fundamental part of application. Graphic interface must not influence on business logic of client – server application.

The data about user’s basket, showcase content and so on go from server in json format. They are not connected with UI. Such a connection is established on client side via React.

What syntax and method / mindset have you used to ensure that the asynchronous code is mishandled correctly. Describe which potential errors and problems you have identified and how these are handled in the code, why is this appropriate error handling?

During the fetch requests some network errors can take place: server can be unavailable, request data can contain some mistakes and so on. For dealing with such errors the special method catch can be used in the method chain. When the program analyzes a server response, it can use status number for decision is the response correct or not. For example if status is not equal to 200, the exception must be thrown.

The exception should describe an error for a programmer (for example to print it in console.log). Moreover it should be visible for a user as an understandable message (for example to show it as a toast).

What locations do you use for local data storage in your application, why are these suitable choices? If you have not implemented any local data storage, what do you see as potential uses in your application?

In the application the localStorage is used for saving information about user’s basket. The basket has relatively complex structure but converting it to and from json format is easy. So, basket information can be stored as a simple string with the key “basket” in the localStorage.

By what method have you excluded files that can be generated or installed in your source code project, for what reason have you chosen to exclude these files?

The standard method to exclude generated or installed files is using .gitignore file, it contains records about files and directories that do not belong to specific source code of the project. Such files, that are useless in repository, can be downloaded or generated by scripts (for example: node_modules, build). Moreover such files can take large amount of space on hard drive. So, the manual efforts of the programmer are not needed and saving such files in a repository is impractical.

There are ready-to-use templates of .gitignore files on github and other resources (particularly for JavaScript and React projects).

.