Автор: Abdirasulov J.U.,Qodirov F.E.
29.03.2019 12:40 -

ANGULAR JS EFFECTIVE FOR CREATING DYNAMIC WEBSITES AND OPTIMIZE THEIR PERFORMANCE

Abdirasulov Javohir Ulugbekovich,

Qodirov Farrux Ergashevich

Karshi branch of the Tashkent University of Information Technologies named after Mohammed al-Khorezmi

Angular is an open source JavaScript framework for developing mobile and web applications. It began as a client-side JavaScript platform for writing the best interface applications that run in a web browser. Today, Angular uses the achievements of modern web development to create SPA (single page applicatiion - single-page applications), cross-platform and native mobile applications. Angular 2 is built on TypeScript, which uses ES6 syntax and compiles to plain javascript. ES5 JavaScript is also suitable for TypeScript, so you can still use your existing code. Angular is a framework for dynamic web applications that allows you to use HTML as a template language and then extend the HTML syntax to express application components. Using data binding and dependency injection, you can eliminate most of the code that would have to be written. Components are the building blocks of an angular application. A component is a combination of an HTML template and a class written in TypeScript (or JavaScript). To create a component in Angular with Typscript, you need to create a class in the file with the .ts extension and decorate it with the component decorator.

Автор: Abdirasulov J.U.,Qodirov F.E.
29.03.2019 12:40 -

For programming Angulyar applications, it is recommended to use the TypeScript language. This is a JavaScript add-in that supports static types. If you declare variables with types (although this is not necessary), the IDE will offer autocomplete and highlight syntax errors as soon as you make them, without waiting for any compilation or program launch. According to the experience of our company, the productivity of developers is significantly increased compared to programming with JavaScript.

Typescript is easy to learn for anyone who knows Java, C #, PHP or Python. This language supports classes, interfaces, inheritance, generics, and more. You write in TypeScript and compile JavaScript code that all browsers understand. Take a look at this screenshot and compare the TypeScript code on the left with the generated JavaScript equivalent (ES5) on the right. I do not know about you, but I prefer the TypeScript version.

In addition, it is important to understand that Angular is used in Google. Most Google applications use Angular from the main repository branch on GitHub. Whenever an update occurs in the master branch, everything is immediately integrated into Google's main internal repository (the company uses one monolithic repository), where products such as Google Maps, Adsense and others are located. As a result, all projects that use Angular fall, if they have critical changes, so you have to rewrite everything, which, in turn, is inconvenient. Thus, it is best to have a fully unified version that would be easier to maintain over time, which would help the team be more productive in releasing new features. AngularJS offers a data binding concept that works in both directions. However, it should be noted that the center object of the data binding is always the scope. Only objects within the scope can participate in data binding. Dependency injection (Dependency Injection, DI) is the third cornerstone of AngularJS, but it does not relate directly to the logic of the library, but rather to code organization.

As such, the models in AngularJS are not. The ngModel used above is nothing more than a directive that changes the behavior of the DOM, and model is only a convenient name. You can

use Backbone.js models for example, or you can manually request data from the server manually or use the ngResource module (read the $ http and $ resource documentation).

It is important to remember that all data binding occurs solely on scope. Put the model in scope? Everything works fine. Templates in AngularJS are nothing more than plain HTML, extended with directives and expressions. Expressions are used to display data, directives - to extend the functionality of HTML, turning it into a DSL (Domain Specific Language) for your specific application. You can add a directive that creates a tabbed interface, you can add a directive that loop the contents of an array with data (such a directive is built into AngularJS), you can output HTML depending on the value of a variable, and so on. Everything is limited only by your imagination. The true power of AngularJS lies precisely in the directives, and the usual , extended directive can turn into anything. Angular.js is currently the most popular framework developed by guys from Google. Most frameworks are overconfident now. They have their own philosophy of how a web application should be built. You will have to take the time to learn basic concepts. Other Backbone solutions do not tell developers how to build a project. Therefore, some even call such technologies simply libraries, not frameworks.

There are different approaches to make your application look good on mobile devices, for example:

• Use the same code for desktops, tablets and smartphones, but implement responsive web design techniques to change the content and layout of the screen depending on its size.

• Use one of the libraries (for example, Native Script), which offer a set of tags for use in templates of Angulyar components with their autochange to native components of iOS or Android.

In fact, JS frameworks appeared not so long ago. I remember when sites were written with poorly structured JS code (in most cases, under the control of jQuery). However, client UIs became increasingly difficult, and JS lost its reputation as a “toy” language. Modern websites rely heavily on JS, so the need for proper organization (and testing!) Of the code has increased. Therefore, client frameworks have become popular, and at the moment there are at least a dozen of them.

The framework on pure JS should have separated the application logic and DOM manipulation. His goal was to update dynamic pages. But it was not so obtrusive: under the control of AngularJS there could be only a part of the page. This framework introduced many powerful features that allow developers to easily create rich, single-page applications. AngularJS became popular very quickly and received great support. Nevertheless, the developers decided to go even further and proceeded to create a new version of Angular 2 (later just Angular without the JS part). The framework received a new name for a reason: it was completely rewritten and redesigned, and many concepts were revised.

REFERENCES:

1. Angular для профессионалов. Адам Фримен, 2018

2. Изучаем Angular 2. Пабло Дилеман, 2017

3. Angular 4. Быстрая разработка сверхдинамических Web-сайтов на TypeScript и PHP. Владимир Дронов, 2018

4. Angular. Сборник рецептов. Мэтт Фрисби, 2018

5. https://angularjs.org/