

Images haven't loaded yet. Please exit printing, wait for images to load, and try to print again.



Kate Ganiukova [Follow](#)

Mar 6 · 6 min read



<https://www.eventshigh.com/>

The year of 2017 has created the highest demand for full stack developers ever. Why is it happening? In a nutshell, the more skills you have in your profession nowadays, the more valuable you become. What a benefit it is to have someone who knows everything about app development! But who are those people in reality?

A question which can't be answered in a few simple words. If you look for a jack-of-all-trades developer or want to become one, here is a guide on how-to.

What Makes Full Stack Developers Special in 2018

If we look through some theoretical aspects, we'll learn the most widespread meaning of "the full stack developer". It is a specialist who prefers working equally good in back-end and front-end.

This term is far from those you may meet in the standard software theory. Such universal specialists appeared thanks to evolution

within the technical development. They never stop learning new technologies and employ them in projects.

And the evolutionary process bears fruit to the labor market.

Developer Survey held by Stack Overflow in 2016 shows us that the majority of developers consider themselves as full-stack technicians.



<https://insights.stackoverflow.com/>

Back-end developers are in the second place by a big margin. Thanks to this research, we know that full-stack developers can work with 5–6 languages and frameworks, while other specialists choose 4 of them.

Personal Traits of Full Stack Developers

If you decide to focus on full stack development, you should first understand whether you have necessary personal characteristics to cope with the job. These are requirements you may meet in the job offers:

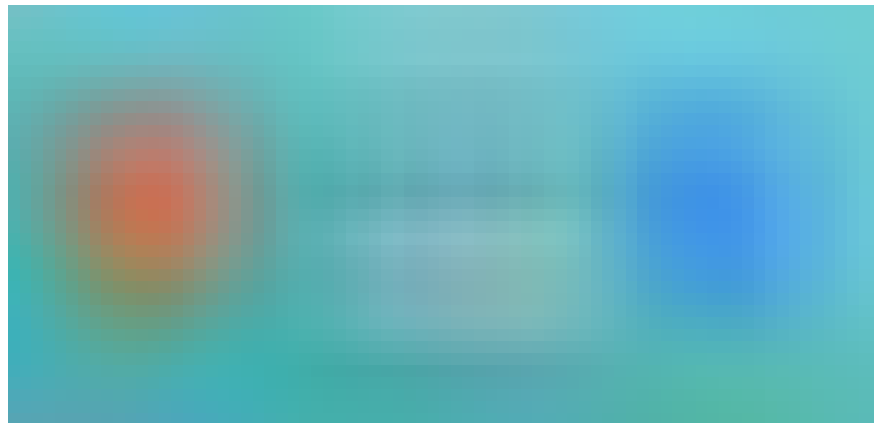
- *Analytical mind*
- *Patience*
- *Love for learning*
- *Attention to detail*
- *Creative vision*
- *Discipline*

If you feel alright with this list and your confidence becomes even stronger, it's time to talk about the technical side of the issue.

Technical Skills of Full Stack Developers in 2018

You should not necessarily become a master in each technology if you concentrate on both back-end and front-end. But what really matters is your ability and passion to understand everything that's going on and make it work. This is a list of things to learn if you're a technician or things to ask for if you're a business owner.

HTML and CSS



<https://www.sololearn.com/>

HTML and CSS are two fundamentals, which you meet first at the very beginning of any programming course. It works this way because thanks to HTML you can add materials to a web page, and CSS helps to give it a certain style.

Start with these things:

- *Semantic HTML.*
- *Learning the CSS BoxModel so you could know its principles and distinctive traits.*
- *CSS preprocessors—their benefits for software development.*
- *Bootstrap.*

JavaScript

<https://insights.stackoverflow.com/>

Why exactly JavaScript? As it was mentioned in [the Stack Overflow's survey](#), this programming language is in the first place for three types of developers (including full stack). New tools appear regularly so you have to be always on guard.

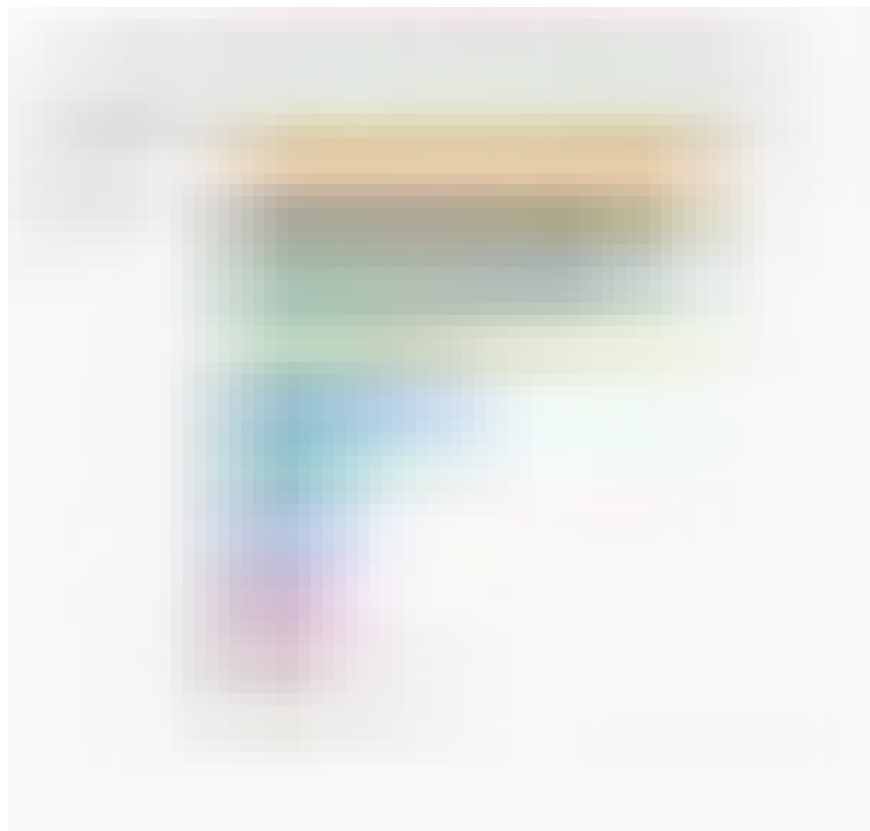
Start with these things:

- *Learning working principles of DOM and JSON.*
- *Focusing on the language's important features: prototypal inheritance, event delegation, etc.*
- *Paying much of your attention to one framework of your choice.*

- *Having a common understanding of testing frameworks.*

Back-End Language

The important point here is that you have to choose one language and focus on it. In other words, you should know in detail how it works and what exactly you can do. Complete understanding—that's what an employer would prefer. We have some stats for you to ease your considerations:



Indeed.com

Databases and Storage

How to store data wisely? You should know a definite answer to this question. Take care of your knowledge on this aspect as data is your main equipment.

Start with these things:

- *Getting to know NoSQL databases and situations when to choose a certain one.*

- *Peculiarities of relational data.*
- *Understanding how to link a database and a back-end language you've chosen.*
- *Web storage.*

HTTP

The application protocol HTTP allows clients and servers to stay in a constant connection. For example, you can create a request via the JavaScript code, which will be sent to the back-end code. It will happen thanks to HTTP.

Not so long time ago, software specialists began facing lots of security issues with HTTP protocol. The thing is HTTP data is not encrypted creating an extremely unsafe environment for website users. Now, we have a secure version which is called HTTPS. No matter what website you work with, knowing HTTPS is a crucial matter. If your website stores sensitive data, than this type of protocol is of the biggest importance for you.

Start with these things:

- *Learning fundamental points about REST.*
- *Designing RESTful API.*
- *Using Chrome DevTools.*
- *Learning more about SSL Certificates.*

Architecture

If you deal with everything above, here comes an advanced part. Knowing app architecture is crucial when you concentrate on the complex project. You should know how to put your code and the data in order, where to store large files, how to separate certain files, and a lot more aspects.

Start with these things:

- *Gathering information about platforms as a service.*

- *Working on optimization of the app.*
- *MVC.*
- *Learning codebases from the GitHub large projects—people there know what they're doing.*

Git



<https://think201.com/blog/>

Git is the most well-known version control system. If you understand how it works, you will successfully cooperate with other developers working on the same project. When any changes are made, you will hold a finger on the app's pulse: get the latest data and update it, make necessary fixes, etc.

Start with these things:

- *Learning the full list of Git commands.*
- *Examining a basic tutorial on Git usage.*

Where to Learn the Basic Skills

Constant learning—that's what makes success for full-stack developers. As there are lots of spheres to embrace, being alert about the latest tools and technologies updates is your daily concern. But where should you start? Here are some sources to help you:

- *Mozilla Developer Network*

- *MySQL Documentation*
- *W3Schools Tutorials*
- *Hypertext Preprocessor*
- *Online training from the web*
- *Helping non-profit organizations*

No theoretical knowledge can replace regular practice. Being a part of the complex project will allow you to work with all sides of the product and learn more about app architecture.

Drawbacks That Should Be Mentioned

At the very beginning, it's useful to know some negative points you will face inevitably. Some people would say that knowing everything means knowing nothing. So any full-stack developer should track his knowledge to keep up with the technology improvements. Any tricks you've learned should be written in a blog or a personal planner. Such a meticulous work is not for everyone.

Don't expect that you'll be given lots of responsibilities when you make the first steps in full-stack development. Any specialist of such kind should prove his skills, so you'll be probably given only a part of your future autonomy. Be ready to fight for more tasks on the way.

Knowledge is Full Stack Power

Full-stack development is great for those who don't like to focus on one field and prefer complicated tasks. Your own database of knowledge will constantly grow, so you should take care of tracking it.

At the same time, businesses can benefit from working with full-stack technicians. They know all aspects of software building no matter how the project changes.

