

What You Need to Know to be a Back-End Developer in 2018

[One speaker]

[Narrator spends most of the video reading information off the presentation]

Speaker: In this video you'll learn what a back-end developer does and what you'll need to learn in order to become one. But first we need to define what a back-end developer is. There are generally two parts of any web application: the front-end, or a client-side, and the backend, or a server-side. The front-end consists of the web pages you see and interact with in your browser. Those pages are usually designed by front-end developers. But how are those pages delivered to the browser, and when a user enters data into those pages, where does it go? When a user enters a URL, clicks a link, or submits a form, their browser connects to a web server. A web server is just a computer running software that delivers resources, like web pages and images.

When a web server receives a request for a resource, it needs to respond with that resource. Programming web servers to respond with the correct resources is the domain of back-end developers. Before a web page or other resource is delivered to a browser, a web application written by back-end developers fills that page with data. The data is pulled from a database which is a separate program that stores and retrieves data.

Back-end developers need to understand whatever programming language the web application is written in, as well as the database where the data is stored. So a front-end developer designs web pages, and a back-end developer serves up those web pages, customized with user data. You may have also heard the term full stack developer before. That just means someone with the skills of both front-end and back-end developers. Web developer is a term that isn't clearly defined. It could mean a front-end developer or it could mean a full stack developer.

Back-end developers typically work together with front-end developers when creating a web application. While the front-end developer focuses on the look and feel of a website and the user experience, back-end developers focus on the data of the site presents.

Most web apps present various forms where users can enter data to create profiles, share stories, keep track of appointments, and so on. Back-end developers are responsible for storing this data and ensuring it's only presented to users who are supposed to have access to it.

Many websites require paid subscriptions in order to access them. Some back-end developers are responsible for accepting data on credit cards and other methods of payment, securely storing that info and making charges to that payment at the right times and in the right amounts.

API's, or Application Programming Interfaces, are resources meant to be accessed, not by web browsers, but by mobile phone apps and other programs. There are many steps to responding to a request from a browser, each requiring a different type of software. The combination of technologies used by a particular web server is known as its text stack. Let's look at each layer of a typical text stack.

Transcript: What You Need to Know to be a Back-End Developer in 2018

When the browser connects to the server, web server software accepts the request. Examples of popular web server software include Apache, NGINX, and Microsoft IIS, short for Internet Information Services. By the way, just as you see several options for web servers for each layer of the tech stack, you'll find there are many choices between software and frameworks. There's no way to learn them all before you get your first development job, but you'll find that the different web servers, frameworks, and databases all work in much the same way.

If you learn a popular language or framework from each layer, you'll find there are plenty of companies ready to hire you and train you in their particular tech stack. The web service software then passes the request off to a web framework. Web frameworks are libraries for programming languages that help back-end developers handle web requests. There are many different programming languages out there, each with multiple frameworks to choose from; but again, they all work similarly. So after you've learned one language and framework, you'll have an easier time learning any of the others.

We're showing the most popular languages and their corresponding frameworks here, as those tend to be the ones with the most companies hiring developers. Javascript is a language originally developed to run inside web browsers, to make web pages more interactive. More recently, server software has been released that allows developers to write javascript code for the backend as well.

Popular JavaScript web frameworks include Node.js and Express. Java should not be confused with JavaScript. The Java language was designed from the ground up to be run on the server side. Java isn't as easy to pick up as JavaScript, but the resulting programs are fast enough to handle requests from thousands of users at once. Popular Java web frameworks include Spring and JavaServer Faces (JSF).

Python is a highly flexible programming language that's easy to learn and use. Python is also very popular for data analysis. Popular Python web frameworks include Django and Flask. C# was originally developed by Microsoft as an alternative to Java. If you're going to be working in the Microsoft ecosystem, you're going to want to learn C#. Popular C# web frameworks include ASP.NET MVC and ASP.NET Core.

The web framework then coordinates the rest of the process of handling the request. Code written by the back-end developer uses ORM, short for object relational mapping, to connect to a database and retrieve data that will be used in the finished webpage. Programming languages and web frameworks represent data as objects, and the ORM library is responsible for converting database records to objects.

Most databases use a separate language called SQL, short for a structured query language, to retrieve data. The ORM library automatically writes most of the SQL queries to retrieve the needed data. Sometimes, though, a query is so complex that the ORM library can't handle it. Back-end developers need to know SQL so they can write their own queries in these situations. Once the data has been retrieved, the back-end developer needs to insert it in the pages that were designed by the front-end developers. Web pages are written using HTML, short for hypertext markup language, and a back-end developer needs to know enough HTML to be able to update the page. Or, if you're maintaining an API, the server might send a response that's meant for use by a program other than a browser.

The most common formats for API responses are JSON, which stands for JavaScript Object Notation, and XML, which stands for Extensible Markup Language. A back-end developer needs

Transcript: What You Need to Know to be a Back-End Developer in 2018

to be able to convert data to JSON or XML for the server to respond with. Once code has been written to carry out all these steps for a request, the back-end developers work is done. The web framework will automatically deliver the completed response to the web server, and the web server will respond to the users browser.

If the back-end developer has written their code carefully and cleanly, avoiding slower inefficient operations, then this whole request and response process usually takes less than a second. Successful back-end developers rely on other tools besides their programming language and web framework. You'll need to be familiar with the database software itself so you can create the tables and columns that will hold user data. Fortunately, most databases work similarly, and the ORM library will handle most of the work of managing the database for you.

Popular database software includes MySQL and SQL Server, as well as PostgreSQL. You'll need to have one version of your code running on your webserver and another version on your own computer while you work on new features. Version control software can help you manage these different versions of your code.

Popular version control software includes Git and Subversion. You'll also need computers to run your finished web app. Gone are the days when most companies ran giant server computers in their own buildings. Nowadays, most apps are hosted on cloud hosting platforms, which can deploy copies of your app to as many or as few servers as you need.

Popular cloud hosting platforms include Amazon Web Services, Heroku, and Google Cloud Platform. Learning new technology is hard, and one of the most difficult parts is figuring out what you need to know. I'm a teacher at Treehouse, an online school where you learn at your own pace. If you're new to software development, Treehouse is the best place for you to learn. Although we have lots of content from professional developers, we also offer carefully designed video tutorials for beginners that don't assume you already know a lot about development. But Treehouse offers lots of content. What's the best place to start?

All the data the back-end developers manage needs to be presented in HTML web pages. So even if you're certain your goal is to be a back-end developer it's a good idea to learn the basics of front-end development first. Front-end developers need to know HTML to give pages structure, cascading style sheets to let you add colors and other styling to pages, and JavaScript programming to make pages interactive.

A great way to learn all that is to take Treehouse's front-end web development track. By the time you've finished, you'll know HTML, CSS, and JavaScript, and you'll have some experience using Node.js as a web server. The track covers other technologies you'll need to know too, like Version Control with Git.

From there, you can join a track for whatever back-end programming language interests you.

You might choose a language because it's used by companies you want to work for. Java is heavily used at Google and Amazon for example. Or you might pick a language because it has a tool or framework you're in. Python, for example, has a lot of great data analysis libraries. So if you're interested in both web development and data analysis, Python is a great choice.

Regardless of what programming language you want to learn, Treehouse has you covered, with tracks for all these languages and more. You can get access to Treehouse tracks for just \$25 a month. Each track is a guided curriculum, designed by industry experts, and it contains everything you need to know to get a successful start as a developer. Tracks are designed to

Transcript: What You Need to Know to be a Back-End Developer in 2018

help you learn effectively, with content including videos, quizzes, and code challenges you can complete right in your web browser.

Picking a programming language track might seem like a tough choice, but in the end, it doesn't matter which language you select. All of Treehouse's tracks are set up to help someone who's learning to code for the first time, and there are lots of similarities between the different programming languages. Once you've learned one language, you'll be able to switch to any other language much more easily. We can't wait to work with you! Click the link in the description to start your free 7-day trial at Treehouse and get started on your journey as a back-end developer.

[End of video]