

Different Back-end Stacks

2 min

Unlike the front-end, which must be built using HTML, CSS, and JavaScript, there's a lot of flexibility in which technologies can be used in order to create the back-end of a web application. Developers can construct back-ends in many different languages like PHP, Java, JavaScript, Python, and more.

You don't need to reinvent the wheel to create a robust back-end. Instead, most developers make use of *frameworks* which are collections of tools that shape the organization of your back-end and provide efficient ways of accomplishing otherwise difficult tasks.

There are numerous [back-end frameworks](#) from which developers can choose. Here are a few examples:

Framework	Language
Laravel	PHP
Express.js	JavaScript (runs in the Node environment)
Ruby on Rails	Ruby
Spring	Java
JSF	Java
Flask	Python
Django	Python
ASP.NET	C#

The collection of technologies used to create the front-end and back-end of a web application is referred to as a *stack*. This is where the term *full-stack developer* comes from; rather than working in either the front-end or the back-end exclusively, a full-stack developer works in both.

For example, [the MEAN stack](#) is a technology stack for building web applications that uses **M**ongoDB, **E**xpress.js, **A**ngularJS, and **N**ode.js: MongoDB is used as the database, Node.js with Express.js for the rest of the back-end, and Angular is used as a front-end framework. While the [LAMP Stack](#), sometimes considered the archetypal stack, uses **L**inux, **A**pache, **M**ySQL, and **P**HP.

