Back-end Frameworks

An overview of back-end frameworks that are commonly used.

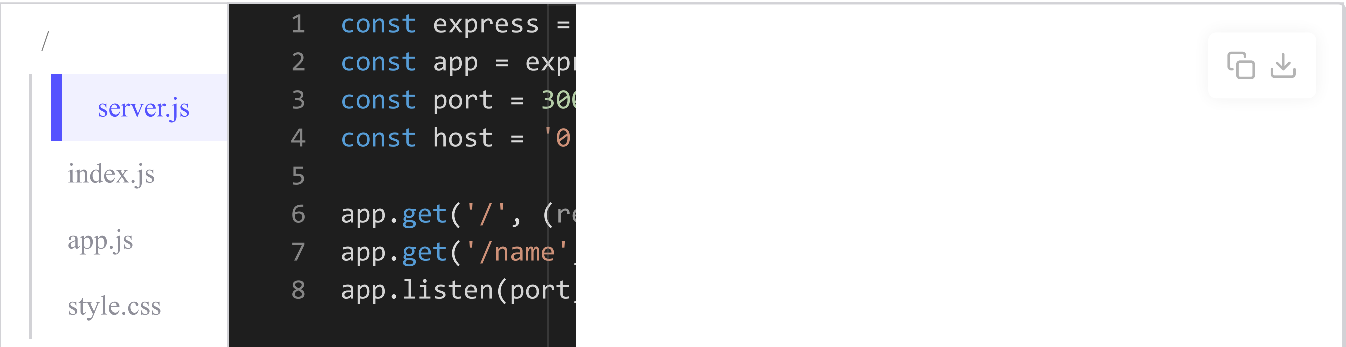
We'll cover the following

- Express (JavaScript)
- Symfony (PHP)
- Django (Python)
- Ruby on Rails (Ruby)
- ASP .NET (C#)

The back-end, as we have already seen, refers to the server-side of a web application. The back-end spans everything from the initial creation of the server to the handling of user requests on the application. Let us take a look at the various back-end frameworks that exist to make these tasks much easier!

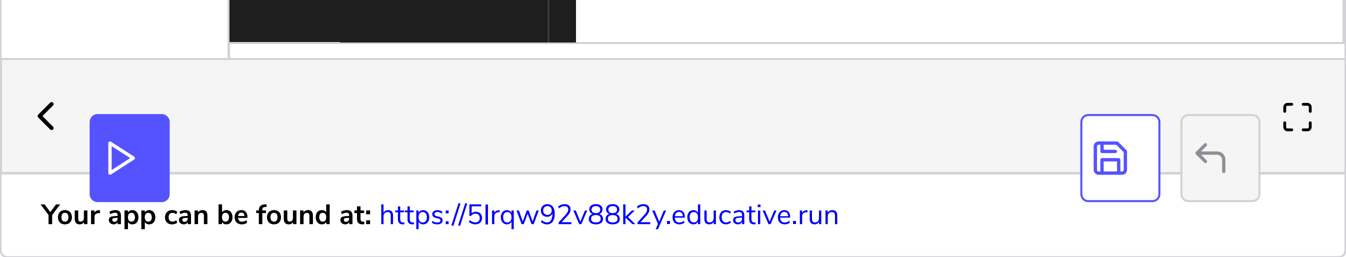
Express (JavaScript)

Express.js often referred to as the “de facto standard server framework” for Node.js, is a minimal and highly flexible Node.js web application framework that provides a robust set of features for web applications. The idea behind Express is to simplify the back-end development process enough to reduce basic back-end features such as creating an HTTP server from multiple, unintelligible lines of Node.js code, to a single instruction, while simultaneously keeping the core syntactical and logical characteristics of Node.js intact.



```
const express =
const app = expr
const port = 30(
const host = '0

app.get('/', (re
app.get('/name'_
app.listen(port_
```



Your app can be found at: https://5lrqw92v88k2y.educative.run

Symfony (PHP)

Symfony is a web development framework for PHP. The key idea behind Symfony is that it is essentially a set of reusable components. The benefits of this are obvious; with over 30 standalone components at your disposal, the process of writing an application becomes considerably simplified and, consequently, much easier and more efficient. Symfony, like the previous frameworks we have discussed, allows for more compact, readable, and simple code. This is beneficial because it allows you to focus on the higher-level functionality you want to implement as the more mundane tasks become confined to much smaller pieces of code that do not need to be thought about from scratch.

Django (Python)

Django is a high-level Python web framework that enables the rapid development of secure and maintainable websites. Django, like Express, takes care of reducing code for basic functionalities to simple instructions, so you can focus on writing your application and implementing more specific functionality without needing to reinvent the wheel and getting caught up in needlessly long bouts of code in doing so!

Ruby on Rails (Ruby)

Ruby on Rails, also known as Rails, is a server-side web application framework written in Ruby. Rails is built on the Model View Controller architecture that we have previously studied, and it provides default structures for everything that goes into each component, including databases, several common web services as well as web pages themselves. Ruby on Rails, therefore, separates the process of web development into simplified components and provides a basic structure for each element any given component comprises of, thus making it immensely convenient for users to build their own specific functionality on top of these provisional structures without having to worry about starting from the very basics.

ASP .NET (C#)

ASP.NET is an open-source, server-side web development framework for building modern web applications and services using .NET or any language supported by .NET. ASP .NET was developed by Microsoft to facilitate the process of developing dynamic web pages, and it has, therefore, considerably simplified the process of creating websites based on HTML5, CSS, and JavaScript. As a result, it has made it possible for users to create websites that are not only simple and fast but also have the capacity to scale to millions of users.

That covers the variety of both back-end and front-end frameworks you can choose from to build your first website. However, these aren’t all the frameworks you can consider. In the next lesson, we will discuss the third type of framework you can choose from called isomorphic frameworks.