

Web application framework: What it is, how it works, and why you need it

Anastasia Yaskevich

Given the rich and wide functionality of modern web apps, the idea of creating one seems overwhelming. No matter how well informed you are about all the necessary steps, they are still the steps you have to follow by completing certain tasks. Yet you shouldn't necessarily do that on your own.

Thankfully, there are tools that can make [web application development](#) easier, and not through mere explanations but by providing a profound basis to build a web app on. A web application framework is one of such tools, and this article shows how it can help to create a web app that complies with the latest web standards.



Web application framework vs. content management system

Web app frameworks and content management systems (CMSs) are surrounded by confused questions from aspiring web developers. Do they differ? *How* do they differ? Which of the two is better? Written guides that start out with explaining the working principles of a web development framework and eventually give a list full of CMSs as examples just let the confusion linger.

In actuality, both frameworks and CMSs lay out a foundation for a future web app and refer to the same technologies; for instance, both Symfony (a web application framework) and Joomla (a CMS) are powered by PHP. Their possibilities are very similar, and you can create equally powerful web apps with either of the two.

The main difference is in the approach. Imagine navigating your computer via a command line and an explorer. The former is closer to creating a web app with a web application framework, the latter – with a CMS.

Since a framework is usually a set of libraries and tools that help to build a web app, it requires higher programming skills. Offering more freedom for implementation from scratch, a web application framework is a perfect fit for experienced developers.

Still, you shouldn't belittle a developer who prefers creating web apps in a CMS. One can argue that with a CMS, a person doesn't need programming language at all, but this is true only if we're talking about *managing* an already existing website. To *set up* a web app via a CMS, a person has to know how to work with a server and be able to read/edit various chunks of code.

Model View Controller architecture

More than 80% of all web app frameworks rely on the Model View Controller architecture. The secret of this pattern's popularity is in how rationally it separates the app logic from the interface forming the 3 components reflected in the architecture's name.

Model

The Model knows all about the content and the structure of an app. Upon receiving user input data from the Controller, it communicates the way an updated interface should look directly to the View.

View

This is the app's frontend. It knows the layout and the way a user can interact with any of its parts. The View receives user input, communicates it to the Controller for analysis and updates or reassembles itself according to the Model's instructions (or the Controller's, if a change is minor).

Controller

The Controller is an intermediary between the Model and the View. It receives user input from the View, processes it and informs the Model (or the View) what changes should be made.

Some people advocate that the Controller isn't always necessary and what matters most is to separate the logic from the interface, that is, the Model and the View. Yet, assigning input processing to either Model or View disrupts the pattern's initial ideology of Separated Presentation, where tasks are distributed based on their type.

When each component in an architecture is responsible for a single line of tasks, the project is transparent, flexible and easy to maintain. Besides, the MVC architecture allows:

- Parallel development (less time to deliver)
- Code reuse
- Fixing or modifying one of the components without having to update the others
- Setting SEO-friendly URLs.

Many web frameworks have incorporated the MVC pattern, so if you're interested in it, make sure the framework of your choice relies on this architecture.

Types of web application frameworks

In the Web 1.0 era, all web apps were mainly built around servers. Such apps still exist and are highly secure, since their entire app logic is stored on the backend.

But as web standards began to change, app logic started to shift toward the client, which helped to ensure a smarter interaction between a user and a web app. With the logic on its side, a client can instantly react to user input. What's more, client-side logic makes apps responsive, so they are easy to navigate on any device.

This way, we now have two groups of web application frameworks: one helps to set up app logic on the server, the other – on the client. To create a powerful web app, you can use both of them simultaneously.

Server-side web application framework

Although front end has evolved, it's first and foremost job is to display an interface, and without app logic any UI/UX is irrelevant. That's why server-side frameworks are important.

Among the most popular MVC-based server-side web frameworks are:

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

By using either of these server-side web application frameworks, you let it handle HTTP requests, database control and management, as well as URL mapping. You can also render view data with a server, like in the Web 1.0 era, but consider using client-side frameworks instead to introduce more user-engaging features and responsiveness.

Client-side web application framework

In the previous article on [web app architecture](#), we discussed the Single-Page Web App concept, where one of the main components of the architecture is a JavaScript client layer. To properly set it up, you need client-side frameworks, such as:

- Bootstrap
- React.js
- Angular.js
- Backbone
- Semantic-UI.

While with server-side frameworks your choice mainly depends on the language you feel comfortable to develop in, here you should mind specific capabilities of different client-side frameworks. Since they vary in the scope of functionality supported, look for the one that fits the needs of your future web app.

Summary

If you want to develop a web app, a web application framework isn't a must. Yet, it can significantly speed up and simplify your work, as well as help you create a responsive and more engaging web app. Since many web frameworks are now based on the MVC principle, you can enjoy the perks of this architecture pattern and have double the advantage. Carefully assess your skills, revise the goals of your project, and choose your server-side and/or client-side frameworks wisely. <https://www.scnsoft.com/blog/web-application-framework>

Top 10 Frameworks for Web Applications

Web Application Framework or simply “web framework” is a software framework that is designed to support the development of web applications including web services, web resources, and web APIs. Frameworks are, in short, libraries that help you develop your application faster and smarter!

Nowadays, the number of Web Frameworks has increased greatly. To help you pick up the most suitable one for your Web Application, we have compiled a list of 10 best frameworks available online, in your preferred language.

1. Ruby on Rails

Ruby on Rails is an extremely productive web application framework written by David Heinemeier Hansson. One can develop an application at least ten times faster with Rails than a typical Java framework. Moreover, Rails includes everything needed to create a database-driven web application, using the Model-View-Controller pattern.

- **Language:** [Ruby](#)
- **Latest Version:** Rails 5.0.0.beta2
- **Framework Link:** <http://rubyonrails.org>
- **Github Link:** <https://github.com/rails/rails>

Websites using Ruby on Rails are GroupOn, UrbanDictionary, AirBnb, Shopify, Github

2. Django

Django is another framework that helps in building quality web applications. It was invented to meet fast-moving newsroom deadlines while satisfying the tough requirements of *experienced Web developers*. Django developers say the applications are it's ridiculously fast, secure, scalable, and versatile.

- **Language:** Python
- **Latest Version:** Django 1.9.2
- **Framework Link:** <https://www.djangoproject.com>
- **Github Link:** <https://github.com/django/django>

Websites using Django are Disqus, Pinterest, Instagram, Quora, etc.

3. [Angular\(Also, know as Angular JS\)](#)

Angular is a framework by Google (originally developed by Misko Hevery and Adam Abrons) which helps us in building powerful Web Apps. It is a framework to build large scale and high-performance web applications while keeping them as easy-to-maintain. There are a huge number of web apps that are built with Angular.

- **Language:** JavaScript
- **Latest Version:** Angular 7.1.5
- **Framework Link:** <https://angular.io/>
- **Github Link:** <https://github.com/angular/angular>

Websites using Angular are Youtube on PS3, Weather, Netflix, etc.

4. [ASP.NET](#)

ASP.NET is a framework developed by Microsoft, which helps us to build robust web applications for PC, as well as mobile devices. It is a high performance and lightweight framework for building Web Applications using .NET. All in all, a framework with Power, Productivity, and Speed.

- **Language:** [C#](#)
- **Latest Version:** ASP.NET 5 (ASP.NET Core 1.0)
- **Framework Link:** <http://www.asp.net/>

Websites using ASP.NET are GettyImages, TacoBell, StackOverflow, etc.

5. [METEOR](#)

Meteor or MeteorJS is another framework that gives one a radically simpler way to build realtime mobile and web apps. It allows for rapid prototyping and produces cross-platform (Web, Android, iOS) code. Its cloud platform, Galaxy, greatly simplifies deployment, scaling, and monitoring.

- **Language:** JavaScript

- **Latest Version:** Meteor 1.2.1
- **Framework Link:** <https://www.meteor.com/>
- **Github Link:** <https://github.com/meteor/meteor>

Websites using Meteor are HagggleMate, WishPool, Telescope, etc.

6. Laravel

Laravel is a framework created by Taylor Otwell in 2011 and like all other modern frameworks, it also follows the MVC architectural pattern. Laravel values Elegance, Simplicity, and Readability. One can right away start learning and developing Laravel with Laracasts which has hundreds of tutorials in it.

- **Language:** PHP
- **Latest Version:** Laravel 5.2
- **Framework Link:** <https://laravel.com/>
- **Github Link:** <https://github.com/laravel/laravel>

Websites using Laravel are Deltanet Travel, Neighbourhood Lender, etc.

7. Express

Express or Expressjs is a minimal and flexible framework that provides a robust set of features for web and mobile applications. It is relatively minimal meaning many features are available as plugins. Express facilitates the rapid development of Node.js based Web applications. Express is also one major component of the MEAN software bundle.

- **Language:** JavaScript
- **Framework Link:** <http://expressjs.com/>
- **Github Link:** <https://github.com/strongloop/express>

Websites using Express are Storify, Myspace, LearnBoost, etc.

8. Spring

Spring, developed by Pivotal Software, is the most popular application development framework for enterprise Java. Myriads of developers around the globe use Spring to create high performance and robust Web apps. Spring helps in creating simple, portable, fast, and flexible JVM-based systems and applications.

- **Language:** Java
- **Latest Version:** Spring 4.3.0
- **Framework Link:** <http://projects.spring.io/spring-framework/>
- **Github Link:** <https://github.com/spring-projects/spring-framework>

Websites using spring are Mascus, Allocine, etc.

9. **PLAY**

Play is one of the modern web application framework written in Java and Scala. It follows the MVC architecture and aims to optimize developer productivity by using convention over configuration, hot code reloading, and display of errors in the browser. Play quotes itself as “The High-Velocity Web Framework”.

- **Language:** Scala and Java
- **Latest Version:** Play 2.4.6
- **Framework Link:** <https://www.playframework.com/>
- **Github Link:** <https://github.com/playframework/playframework>

Websites using PLAY are LinkedIn, Coursera, LendUp, etc.

10. CodeIgniter

CodeIgniter, developed by EllisLab, is a famous web application framework to build dynamic websites. It is loosely based on MVC architecture since Controller classes are necessary but models and views are optional. CodeIgnitor promises with exceptional performance, nearly zero-configuration, and no large-scale monolithic libraries.

- **Language:** PHP
- **Latest Version:** CodeIgnitor 3.0.4
- **Framework Link:** <https://codeigniter.com/>
- **Github Link:** <https://github.com/EllisLab/CodeIgniter>

Websites using CodeIgnitor are Bufferapp, The Mail and Guardian, etc. Apart from these 10 frameworks, others like [Symphony](#) , [Ember.js](#), [Sails.js](#), [React.js](#) are also worth mentioning.