Web Technologies

# What is a web technology?

Web technology is a collective name for technology primarily for the world wide web. This does, however, tend to focus more specifically towards technology that aides in the creation, maintenance and development of web-based applications.

From a developer’s point of view, these are things like languages and frameworks, data storage mechanisms, services and products to allow us to create platforms and applications for the web.

Dan Jackson. (2017). What is meant by web technology?. Retrieved from https://www.quora.com/What-is-meant-by-web-technology

# Web Development Frameworks

Here are the five web technologies currently used: Angular, Ruby on Rails, YII, Meteor JS and Express.js.

Milos Timotic. (2018). 9 Web Technologies Every Web Developer Must Know in 2019. Retrieved from https://tms-outsource.com/blog/posts/web-technologies/

## Detail about web development frameworks

### Angular

AngularJS is a structural framework for dynamic web applications. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application components clearly and succinctly. Its data binding and dependency injection eliminate much of the code you currently have to write. And it all happens within the browser, making it an ideal partner with any server technology.

TutorialsPoint. AngularJS – Overview. Retrieved from https://www.tutorialspoint.com/angularjs/angularjs\_overview.htm

Applicability

Ease of use

Reliability of the technology

#### Pros of AngularJS

⊕ Two-way data binding. AngularJS was built with Model-View-Controller architecture. And the framework synchronized the Model and the View. As the data in the Model changes, the View does too. Two-way data binding allowed engineers to reduce development time as it didn’t require writing additional code to provide continual View and Model synchronization.

⊕ Directives. This feature enabled the HTML extension mentioned above. Directives allowed developers to assign special behaviours to the Document Object Model (DOM), permitting engineers to create dynamic and rich content with HTML.

⊕ Dependency injection. Dependencies define how different pieces of code interact with each other and how the changes in one component impact the other ones. Usually, dependencies are directly defined in the components themselves. So that every change in dependency requires changing components as well. With AngularJS, you could use injectors that defined dependencies as external elements decoupling components from their dependencies. Dependency injection made components more reusable, easier to manage and test.

⊕ Community. Right from the beginning, AngularJS became extremely popular among engineers. A strong community provided enough training materials, discussions, and third-party tools to embark on using AngularJS as well as find a solution to nearly every arising issue.

#### Cons of AngularJS

Θ Performance. Dynamic applications didn’t always perform that well. Complex SPAs could be laggy and inconvenient to use due to their size.

Θ Steep learning curve. As AngularJS is a versatile instrument, there is always more than one way to complete any task. This has produced some confusion among engineers. However, the abundance of tutorials and issue discussions allowed for resolving most of the problems.

AltexSoft. (2019). The Good and the Bad of Angular Development. Retrieved from https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-angular-development/

### Ruby on Rails

Ruby On Rails is one of the most popular Web development frameworks and the one that has boosted the Ruby language itself. It has been initially designed and brought to life by David Heinemeier Hansson (DHH). Now it has a core team and many contributors around the world. Huge number of projects are spawned daily on Web using Ruby on Rails. Its main objective being to relief the developer from dealing with the technology issues and concerns, and focus on solving the business problem.

RoR follows the Model-View-Controller (MVC) architectural pattern. Also, it follows the rule of convention over configuration, (also known as coding by convention) which means that it has already taken some decisions for you and you don't have to worry about these things as long as you follow RoR conventions.

Panos M. (2017). Ruby on Rails - Architecture Overview For Beginners. Retrieved from https://www.techcareerbooster.com/blog/ruby-on-rails-architecture-overview-for-beginners

#### Advantages of Ruby on Rails

* Tooling — Rails provides fantastic tooling that helps you to deliver more features in less time. It provides a standard structure for web apps, where all the common patterns are taken care of for you.
* Libraries — There’s a gem (3rd party module) for just about anything you can think of. They are all publicly available and searchable through https://rubygems.org/.
* Code Quality — Generally, we find the quality of third-party Ruby code to be significantly higher than their PHP or NodeJS equivalents.
* Test Automation — The Ruby community is big into testing and test automation. We believe this is incredibly valuable in helping to deliver good quality software and is one of the reasons the Ruby libraries are so great.
* Large Community — Pretty much every major city in the world has a Ruby community that runs regular meetups. It’s one of the most popular languages on social coding site Github.
* Popular in The Valley — History has shown that technology that’s been popular within Silicon Valley has gradually been adopted across the world. If you look at the big startup successes of recent years, such as Airbnb, Etsy, GitHub & Shopify — they are are all on Ruby on Rails.
* Responsible Developers — You tend to find Ruby developers are more closely aligned around the rules of responsible development. If you start small, communicate well, tackle vertical slices, write simple code over the smart code, share ownership etc, you tend to find your project ends up in better shape.
* Productivity — Ruby is an eloquent and succinct language, which when combined with the plethora of 3rd party libraries, enables you to development features incredibly fast. I would say it’s the most productive programming language around.
* Next Generation — Ruby on Rails seems to be the language of choice for a number of the popular online code schools, such as Makers Academy, Steer and CodeCademy. This should mean an increase in talented programmers joining the Ruby community over the coming years.

#### Disadvantages of Ruby on Rails

* Runtime Speed — The most cited argument against Ruby on Rails is that it’s “slow”. We would agree, certainly when compared to the runtime speed of NodeJS or GoLang. Though in reality, the performance of a Ruby application is incredibly unlikely to be a bottleneck for a business. In 99% of cases, the bottleneck is going to be elsewhere, such as within the engineering team, IO, database or server architecture, etc. When you get to a significant enough scale to have to worry about Rails runtime speed, then you’re likely to have an incredibly successful application (think Twitter volume) and will have many scaling issues to deal with.
* Boot Speed — The main frustration we hear from developers working in Rails is the boot speed of the Rails framework. Depending on the number of gem dependencies and files, it can take a significant amount of time to start, which can hinder developer performance. In recent versions of Rails this has been somewhat combatted by the introduction of Spring, but we feel this could still be faster.
* Documentation — It can be hard to find good documentation. Particularly for the less popular gems and for libraries which make heavy use of mixins (which is most of Rails). You’ll often end up finding the test suite acts as documentation and you’ll rely on this to understand behavior. This isn’t itself a bad thing, as the test suite should be the most up-to-date representation of the system, however, it can still be frustrating having to dive into code when sometimes written documentation would have been much quicker.
* Multi-Threading — Rails supports multi-threading, though some of the IO libraries do not, as they keep hold of the GIL (Global Interpreter Lock). This means if you’re not careful, requests will get queued up behind the active request and can introduce performance issues. In practice, this isn’t too much of a problem as, if you use a library that relies on GLI, you can switch to multi-process setup. The knock-on effect of this is your application ends up consuming more compute resources than necessary, which can increase your infrastructure costs.
* Active Record — AR is used heavily within the Ruby on Rails world and is a hard dependency for many of the Ruby Gems. Although we think it’s a great design pattern, the biggest drawback we see is that your domain becomes tightly coupled to your persistence mechanism. This is far from ideal and can lead to bad architecture decisions.

RORY MACDONALD. (2015). Pros and Cons of Ruby on Rails. Retrieved from https://www.madetech.com/blog/pros-and-cons-of-ruby-on-rails

### YII

The Yii[ji:] framework is an open-source PHP framework for rapidly-developing, modern Web applications. It is built around the Model-View-Controller composite pattern.

Yii provides secure and professional features to create robust projects rapidly. The Yii framework has a component-based architecture and a full solid caching support. Therefore, it is suitable for building all kinds of Web applications: forums, portals, content managements systems, RESTful services, e-commerce websites, and so forth. It also has a code generation tool called Gii that includes the full CRUD(Create-Read-Update-Delete) interface maker.

TutorialsPoint. Yii – Overview. Retrieved from https://www.tutorialspoint.com/yii/yii\_overview.htm

#### Pros

* Yii framework has a large community support and since it’s an open-source platform, it allows developers to connect from around the world and share their suggestions and solutions to some of the key problems. This makes it undoubtedly one of the best e-learning platforms out there.
* Yii framework’s rigid, well-planned architecture assists the developer to devise and develop a technically advanced website. It’s database servers specifically isolated from the users and helps the developer to create effective and cleaner web applications.
* This advanced framework provides the users with a lot of options for layouts and themes for a unique designing aspect of the website. It’s well designed unique controller helps in linking different libraries and packages which will eventually decrease the programming load of the developer.
* Yii development company designed the framework in such a way that it frees the developers from repetitive code writing. Apart from that the developers also ensured that changes implemented in one line are reflected throughout the program. This saves a lot of time and hard work is not necessary.
* This framework provides web developers with neat Ajax helpers, and that too especially in a grid format. Also, the CRUD generator uses Gii. It also offers ActiveRecord for different SQL databases and the smart caching feature supports systems such as XCache and Memcache.
* One of the unique aspects of the Yii framework is it’s added security features preserve and protect sensitive data against outside threats and hence maintaining user credentials and data becomes easier. Some of the security features integrated into this framework are the cross-site forgery, scripting and cookies attack.
* Another astounding aspect with Yii is it’s easy to debug the program because of its simple code format. It has additional features such as a role-based and built-in user authentication process which enables the developing part easier and faster than developed.

#### Cons

* Yii framework is sort of over-reliant on static methods which certain web developers might not be comfortable with.
* The Ajax features embedded in the system is not well-drafted or built smartly as the developers have to use Java language wherever necessary.
* This framework doesn’t allow the build-up of multiple relations and also doesn’t provide AR queries.
* Yii framework’s unique design means it requires greater attention from the developers while developing an application. Commit one mistake and the entire program can get over-sized.
* For young developers, Yii framework might be a little more challenging as they require more knowledge on PHP language, have enough mastery over programming skills and the different add-ons required on it.
* This framework has a steep learning curve making it difficult for new web developers to start with.

Agriya. (2017). Yii Framework – The Major Pros and Cons. Retrieved from https://www.agriya.com/blog/yii-framework-the-major-pros-and-cons/

### Meteor JS

Meteor is a full-stack JavaScript platform for developing modern web and mobile applications. Meteor includes a key set of technologies for building connected-client reactive applications, a build tool, and a curated set of packages from the Node.js and general JavaScript community.

TutorialsPoint. Meteor – Overview. Retrieved from https://www.tutorialspoint.com/meteor/meteor\_overview.htm

#### Pros

* Simplicity
  + If you are a novice in the world of JS frameworks, Meteor is the better option to start your learning process. The reason to opt for the very framework is its full-stack characteristic. Otherwise speaking the comprehension of both client and server-side web creation process will be complete and extensive on the example of a single platform and a single language.
  + Beginners are free of possible complex concepts that may cause the biggest trouble in early stages of learning. The framework offers a clear documentation in addition to well-established programming conventions.
* Abundance of Packages and Libraries
  + An extended functionality of the described framework is achieved due to its foundation represented by the jQuery JS library in combination with Underscore.js. Such libraries as Tracker and Blaze make their contribution to frontend elaboration.
  + Third-party packages won’t deliver any troubles either. A complete set of the Meteor packages is represented by Atmosphere.js. All of them are reliable, up-to-date and highly-sought among numerous engineers working with the framework.
* Seamless Client-Server Communication
  + We have already mentioned that jQuery is responsible for client part of web development. The server side is provided by Node.js. In this structure, Meteor ensures a seamless real-time communication between both sides. If you got used to relying upon the representational state transfer APIs or publish-subscribe code, you can enjoy the automation provided by Meteor.
* Debugging
  + It would be wonderful to have a tool to debug apps built with Meteor, as well as to perform such functions as removal of collections, display of client-side docs, modification of documents, and management of subscriptions. In fact, such tool exists and is called WebStorm. Besides, a server-side debugging is also supported.
* Real-time Testing Tool
  + Velocity is the name of the Meteor’s testing tool. In addition to its core functions, Velocity enables integration with Mocha or Jasmine syntax. As soon as the code is saved, the testing procedure starts. Red or green dots found in the upper right corner indicated the results.

#### Cons

* Rendering on the Server Side
  + SSR is extremely important for search engine optimization tasks. Unfortunately, Meteor lacks the SSR support. The problem is solved via a downloadable third-party package. Besides, the Meteor developers claim that this flaw will be fixed in the nearest release.
* MongoDB Support
  + When it refers to backend options, there is only one official solution represented by MongoDB. In addition to the offered scalability, this document database is known for problems with data integrity. Though you may select other backend options, no official analogs are stipulated.
* Native Widget Library
  + The absence of a native widget library is recompensed by numerous client libraries that will be useful for a certain project. E.g., Bootstrap is a fit for general web apps design and user experience. Such set of user interface components like Webix UI is also the option to build ERP and admin apps.
* Network Connection Flexibility
  + The framework has a severe requirement for the internet connection. If it is weak, the possibility to lose synchronization between frontend and backend sides is rather high. It is rather inconvenient for chat apps where a regular page refresh is required to obtain updated data.
* Inbuilt Support for PWA
  + PWA refers to Progressive Web Applications. Meteor lacks the inbuilt support for creating such products. The structure of such apps lies in combining the best of web and device applications. Unfortunately, such essential technologies as Service Worker, HTTP/2 + Server Push and others are not supported. As a result, despite the small number of cons of Meteor JS development, apps built with it have a common problem of a low-speed first-page load.

Binariks. (2017). 10 Pros and Cons of Meteor JS development. Retrieved from https://binariks.com/blog/tips/10-pros-cons-meteor-js-development/

### Express.js

ExpressJS is a web application framework that provides you with a simple API to build websites, web apps and back ends. With ExpressJS, you need not worry about low level protocols, processes, etc.

TutorialsPoint. ExpressJS – Overview. Retrieved from https://www.tutorialspoint.com/expressjs/expressjs\_overview.htm

#### Pros

* Fast app development
  + Express.js framework allows you to use the same language which is JavaScript both on the back-end and front-end. It gives JavaScript developers the opportunity to become full-stack. As a result, the development process is much faster and easier as one person can manage both presentation and data access layer.
* I/O request handling
  + Express JS is a great choice for applications that handle a lot of requests and notifications from users. Exactly because of this reason Uber has chosen Node.js. This is what the Software Engineer of Uber said:
  + Node.js is particularly well-suited to writing systems that have all their state in memory. They do not have to externalize the concerns of a distributed system. Therefore, the systems can be more available, and they can respond more quickly to requests by eliminating the reading/writing and the serialization of state into a database”.
  + To conclude, Node.js together with Express.js is capable of supporting thousands of concurrent actions.
* Minimizing the expenses spent on hiring the development team
  + Node.js simplifies the process of developing mobile apps and makes it more efficient. You can use the same language to develop your server and mobile app.
* Open-source community
  + Express.js is one of the most supported Node.js frameworks. It has an open-source community, so the code is always reviewed and improved. TJ Holowaychuk, the founder of the Express.js stated:
  + There is no way I could have learned everything that I have without open-source. Contributions and feedback, you get on your own code are really helpful. Naturally it feels great to have people enjoy something you have created as well, so that can be fun”
* Easy integration of third-party services and middleware
  + Express.js is a minimalist framework, so Express development team has created middleware packages for solving different development problems. They include URL parameters, sessions, POST data, security headers, libraries etc.
  + Express.js is unopinionated framework, which means that there is no strict and determined rules on how to deal with particular tasks or what components to choose. You can use any middleware in the order that is convenient for you. Also, you can decide how to structure your app, as there is no one right way to do it.
* Easy to learn
  + Most developers say that Express.js is easy to learn. It is well-documented and supported with large community. Moreover, if you are already familiar with JavaScript and Backed Architecture, you will easily get started with it.

#### Cons

* Event-driven nature (callbacks)
  + Node.js is a single threaded framework with event loop that listens to various events and executes registered callbacks. Developers who have worked with other programming languages may find it difficult to understand the callback nature. After reading several instructions, they start to write all the code in callbacks. Such approach will only entangle the code, so it will be impossible to maintain it in future. This is called a callback hell and we hope you will never experience it! The last version of Node.js gives you the opportunity to use async/await , that will increase the code quality and prevent you from the problems with callbacks.
* Philosophy of plugins known as middleware
  + Express.js is built on this philosophy, that is why it is important to understand its main concepts. In short, middleware is a subset of chained functions that run between the client request and the server answer.
* Code organization
  + Note that the code organization in Express.js is represented by patterns that make your code easier to maintain. Here are some useful tips for simplifying the development process.
    - Choose one approach and stick to it during the whole process of development
    - Control the size of your files. If they are too big, divide them into several (100 lines)
    - Stay tuned about the latest Node.js updates and Express.js tendencies

Tymets Volodymyr. (2017). Express.js Mobile App Development: Pros and Cons for Developers. Retrieved from https://apiko.com/blog/express-mobile-app-development/