

# What is Ember.js?

Ember.js is a front-end JavaScript framework that provides shared solutions to tough problems, where development is guided by the community rather than any one company.

To build for production, you'll need a whole lot more than view layer tools! Ember takes care of the hard parts so you can ship new features quickly:

- zero config package management
- deployment pipelines
- testing framework
- data fetching, serialization, and local storage
- command line tools for building and deploying

Users get to concentrate on the fun part of building web apps - the code, user interface, and interactions.



# Community

Since so many developers use the same tools to handle the infrastructure of their apps, it's easy to get help and advice.

## Slack

[ember-community-slackin.herokuapp.com](https://ember-community-slackin.herokuapp.com)

Check out the #-help channel to get answers to your questions, day or night

## Meetups

[emberjs.com/community/meetups](https://emberjs.com/community/meetups)

Join other devs in major cities throughout the world

## Discussion Forum

[discuss.emberjs.com](https://discuss.emberjs.com)

For questions that need more context or input than chat

## See Who Else Uses Ember

[emberjs.com/ember-users](https://emberjs.com/ember-users)

## For more resources

[learn.emberjs.com](https://learn.emberjs.com)

# Get Started

```
npm install -g ember-cli
ember new my-app-name
cd my-app-name
ember server
```

For free, full-length, official tutorials, visit:

[guides.emberjs.com](https://guides.emberjs.com)

# The Shortest ember Book

Everything you need  
to get started with a  
front-end framework  
for ambitious developers.



# Routing

Have you ever used an app or site where a refresh makes you lose your place? URLs are important! Ember comes with some routing tools out of the box. You can create content that users can easily bookmark, share, and come back to later.

Routing consists of three parts:

- Router listing the urls that should be available
- Template what the user sees when they visit a route, which often includes Components

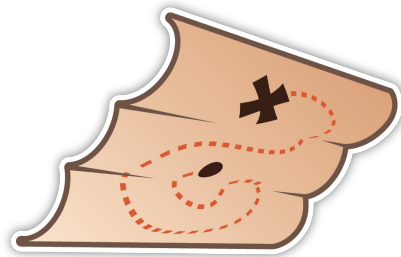


- Route JavaScript functions to load data from APIs

For example, one route might be a full list of blog articles, and each article might have its own route. New routes can be created straight from the data. It's easy to automatically load the correct article based on the url.

[emberjs.com/blog](http://emberjs.com/blog)

Major decisions are made through an RFC (Request for Comments) process, where community feedback determines which features get built or change.



Ember's direction and development are guided by the developers who use it. Whether you're a consultant, a startup dev, or you work for a Fortune 500 company, you have a say.

# Development Roadmap

## Dependencies

There are thousands of libraries that are designed especially for Ember. We call them addons. Many of your favorite JavaScript libraries are available inside addons that bring you some Ember conveniences:

- D3
- Bootstrap
- moment

Some other libraries are unique to Ember:

- ember-concurrency
- ember-power-select
- liquid-fire

You can also use regular npm packages, though addons are often more convenient.

To see what's available, rankings, and reviews, visit:

[EmberObserver.com](http://EmberObserver.com)

## Components

Components are reusable pieces of JavaScript code and markup that form the user interface. Here are some examples of features that a developer might create a component for:



Components in Ember render lightning-fast as data in your app changes. There are straightforward ways to organize components and connect them together, so that they can be independently changed and rearranged when project requirements shift.

ember-cli.com

All this functionality comes ready to go with zero configuration (but if you like configuring things by hand, you still can :P)

Update app versions and syntax

Run tests

Build for production and deploy

Serve the app locally

Install dependencies

Generate new files including boilerplate

Create your app

do things like:

The ember-cli is a command line tool that helps you

## Command Line Tools

## Data

Ember works with data sources of all different kinds:

- REST APIs
- GraphQL/Apollo
- Websockets
- JSONAPI
- ...and more!

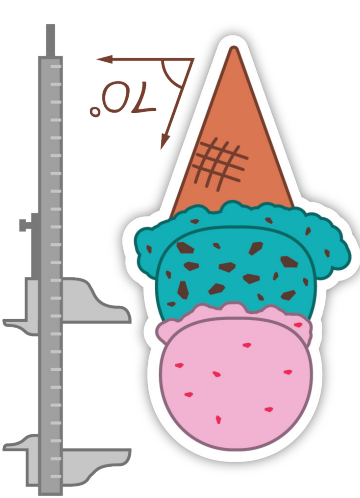


Developers have the option of using Ember Data to make API requests. Ember Data is an official library that includes a local data store, serializers that turn data into searchable objects, async relationship loading tools, and some helpful wrappers for making requests.

Ember is also an ideal candidate for building Progressive Web Apps (PWAs), which are web apps that look and feel like native mobile apps. They often allow for use online, offline, and over shaky internet connections.

## Testing

A maintainable app needs tests, so Ember makes testing as easy as possible.



ember test builds the app and runs tests in the browsers of your choice. For example, you can watch the tests running right in the browser and pause them to inspect the DOM. You could also run them in headless browsers for maximum speed. Tests come in three flavors! You can test user interactions, individual functions, and everything in between using built-in helpers.

By default, Ember uses QUnit, and has full official support for Mocha. You can use your own testing tools if you'd like.

## Deployment

Yay, deploying to the cloud is sunshine and rainbows! Right?

Whether it's your first deployment ever or you deploy every day, Ember has tools to help you out.

There are many community plugins for deploying to different platforms, including:

- Heroku
- AWS S3 or Cloudfront
- Redis
- GitHub pages
- and more.

It can be as simple as filling in some credentials and typing "ember deploy"

### Continuous Integration/Continuous Deployment

(aka CI/CD) is a common strategy some teams use to deploy code automatically when it is merged into their main branch. When you combine a solid testing suite with the CLI, you get a clear path to CI/CD too.