> learn-rails.com danielkehoe.com

Install Ruby on Rails 6.0 · macOS

BY DANIEL KEHOE

See the author's blog at danielkehoe.com

Last updated 16 May 2020

> This is an article from the RailsApps project. The RailsApps project provides Rails Example Applications that developers use as starter apps.

Install Ruby on Rails 6.0 on macOS 10.15 Catalina. Up-to-date and detailed instructions, plus troubleshooting, for the Rails newest release. How to set up and install Rails 6.0, the newest version of Rails, on macOS 10.15 Catalina.

The most complete and up-to-date installation guide for Ruby on Rails on macOS and a favorite of Rails developers. It explains how to set up a development environment for Rails, the same used by any professional Rails developer.

IF YOU ARE NEW TO RAILS

To use Rails on macOS, you'll need Ruby (an interpreter for the Ruby programming language) plus gems (software libraries) containing the Rails web application development framework. If you're new to Rails, see What is Ruby on Rails? or get the free book *Learn Ruby on Rails*.

RUBY VERSION MANAGERS

> You may hear about one-click installation programs such as RailsInstaller, Cinderella, and BitNami RubyStack. These installation programs are often outdated. Most developers install from scratch.

MacOS comes with a "system Ruby" pre-installed. MacOS Catalina includes Ruby 2.6.3 which is not the newest version. Do not use the system Ruby (see an article for why). You should install a Ruby version manager and update to the newest Ruby version. You'll find instructions below.

> Use Chruby to switch between Ruby versions. Some developers prefer RVM or rbenv but I now recommend Chruby.

A Ruby version manager makes it easy to switch between Ruby versions. I recommend Chruby. RVM was popular in the past but its additional features (its gemsets) are no longer necessary and it adds unnecessary complexity. Sam Stephenson's rbenv is also popular, but requires extra steps (the rehash command) and modifies gems when you install them.

Even if you are a student only building new Rails applications, you should be prepared to manage multiple versions of Ruby. In the future, you will need to install newer Ruby versions as they are released. And if you maintain older Rails applications, you will need a Ruby version manager.

You'll need to prepare your computer before installing Ruby on Rails.

Prepare Your Computer

> A Hosted Development Alternative
>
> You can use Ruby on Rails without actually installing it on your computer. Hosted development, using a service such as AWS Cloud9, means you get a computer "in the cloud" that you use from your web browser. Any computer can access the hosted development environment, though you'll need a broadband connection. Cloud9 is free for small projects.
>
> Using a hosted environment means you are no longer dependent on the physical presence of a computer that stores all your files. If your computer crashes or is stolen, you can continue to use your hosted environment from any other computer. Likewise, if you frequently work on more than one computer, a hosted environment eliminates the difficulty of maintaining duplicate development environments. However, most developers set up a local developmemt environment on their own computer.

These instructions are for macOS Catalina 10.15.

IF YOU UPDATED TO MACOS CATALINA

If you updated to Catalina from an earlier version of macOS (sometimes called an "over the top" installation), and you previously installed a Rails development environment, your earlier installation remains intact. You will need to install the new version of Xcode Command Line Tools (full instructions below). Though Rails is still intact after an update, read through this article and take this opportunity to update your development environment.

UPGRADE MACOS TO CATALINA

MacOS Catalina was released on October 7, 2019. Make sure you have the latest version of macOS. Under the Apple menu, check "About This Mac." It should show "Version 10.15.0" or newer. If you've owned your Mac for several years and haven't updated macOS, be prepared to spend several hours updating the operating system.

If you need to upgrade, see Apple's instructions Upgrade to macOS Catalina. You can install macOS 10.15 (Catalina) from the Mac App Store for free. Allow plenty of time for the download and installation (it may take several hours).

If you've already done development on earlier Mac versions, be aware of these changes in Catalina:

- Catalina uses Zsh as a default shell instead of Bash. You'll set configurations (such as PATH and aliases) in the *.zshrc* file.
- Apple has removed the */usr/include* folder making it difficult to compile some older system tools.

TERMINAL APPLICATION (FOR BEGINNERS)

> To learn more about Unix shell commands, read The Command Line Crash Course or Learn Enough Command Line to Be Dangerous.

You'll need to use the Terminal application to install Ruby and develop Rails applications. The Terminal application or *console* gives us access to the Unix command line, or *shell*. We call the command line the shell because it is the outer layer of the operating system's internal mechanisms (which we call the kernel). Search for the macOS Terminal application by pressing the Command-Spacebar combination (which Apple calls "Spotlight Search") and searching for "Terminal." Or look in the **Applications/Utilities/** folder for the Terminal application.

You'll also need a text editor (typically VS Code or Atom) to configure your Mac (and for writing code).

Launch the Terminal application. Then try out the terminal application by entering a shell command:

```
$
```

```
$ whoami
```

Don't type the `$` character. The `$` character is a cue that you should enter a shell command. This is a longtime convention that indicates you should enter a command in the terminal application. The Unix shell command `whoami` returns your username.

Just a reminder to absolute beginners: Press "Enter" after you type the command.

IS XCODE ALREADY INSTALLED?

You need to install Apple's Xcode Command Line Tools to get the Unix tools needed to install Ruby and develop with Rails. Xcode is Apple's software library for macOS developers. You don't need all of Xcode for Rails development. You just need the Xcode Command Line Tools. Only install the full Xcode package if you are doing development of applications for the Apple operating systems.

Check if you have previously installed the full Xcode package:

```
$ xcode-select -p
```

SCENARIO 1

If you see:

```
xcode-select: error: unable to get active developer directory...
```

The Xcode package is not installed. Jump to the next section and install only the Xcode Command Line Tools.

SCENARIO 2

If you see:

```
/Applications/Xcode.app/Contents/Developer
```

The full Xcode package is already installed. Perhaps you installed it previously. If Xcode is installed, you will need to update Xcode to the newest version (Xcode 11 or newer). Go to the App Store application and check "Updates." After updating Xcode, be sure to launch the Xcode application and accept the Apple license terms.

SCENARIO 3

If you see:

```
/Library/Developer/CommandLineTools
```

The Xcode Command Line Tools may be installed *or an older version* is installed. Here's how to test:

```
$ /usr/sbin/pkgutil --packages | grep CL
```

You should see:

```
com.apple.pkg.CLTools_Executables
```

Also try:

```
/usr/sbin/pkgutil --pkg-info com.apple.pkg.CLTools_Executables
```

You should see:

```
package-id: com.apple.pkg.CLTools_Executables
```

If you get a response from the test, you're done. The Xcode Command Line Tools are installed and you can skip ahead to install Homebrew.

If you get no response from the test, you have an older version of Xcode Command Line Tools and you must remove it.

Remove the old version:

```
$ sudo rm -rf /Library/Developer/CommandLineTools
```

You are using `sudo` for admin access so you must enter the password you use to log in to your computer (you will not see the password after entering it). After removing the old version, jump to the next section and install the Xcode Command Line Tools.

SCENARIO 4

If you see a file location that contains spaces in the path:

```
/Applications/Apple Dev Tools/Xcode.app/Contents/Developer
```

you will have problems installing Ruby. You should delete and reinstall Xcode.

INSTALL XCODE COMMAND LINE TOOLS

The Xcode Command Line Tools provide a C language compiler needed to install Ruby. For many Rails projects, you will need the C language compiler to install gems that use native extensions (such as Nokogiri).

Here's how to install Apple's Xcode Command Line Tools.

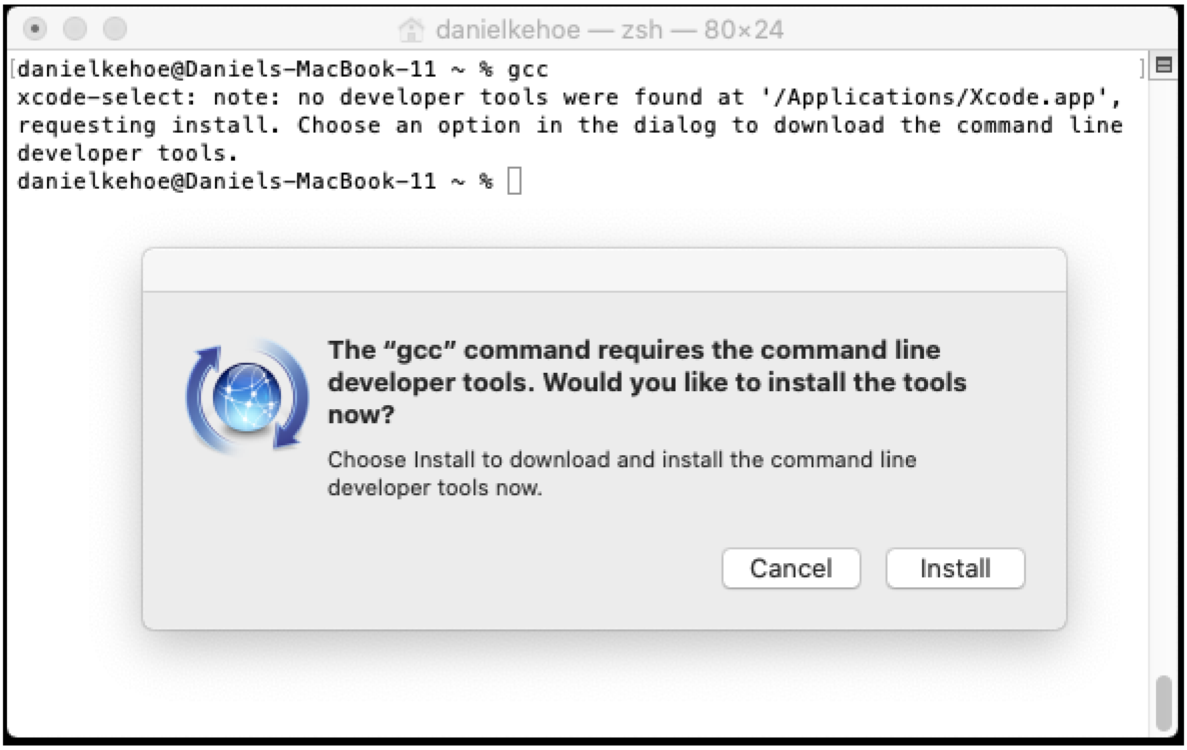
MacOS Catalina will alert you when you enter a command in the terminal that requires Xcode Command Line Tools. For example, you can enter `gcc`, `git`, or `make`.

Try it. Enter:

```
$
```

```
$ gcc
```

If Xcode Command Line Tools are not installed, you'll see an alert box:



```
danielkehoe@Daniels-MacBook-11 ~ % gcc
xcode-select: note: no developer tools were found at '/Applications/Xcode.app',
requesting install. Choose an option in the dialog to download the command line
developer tools.
danielkehoe@Daniels-MacBook-11 ~ %
```

The "gcc" command requires the command line developer tools. Would you like to install the tools now?

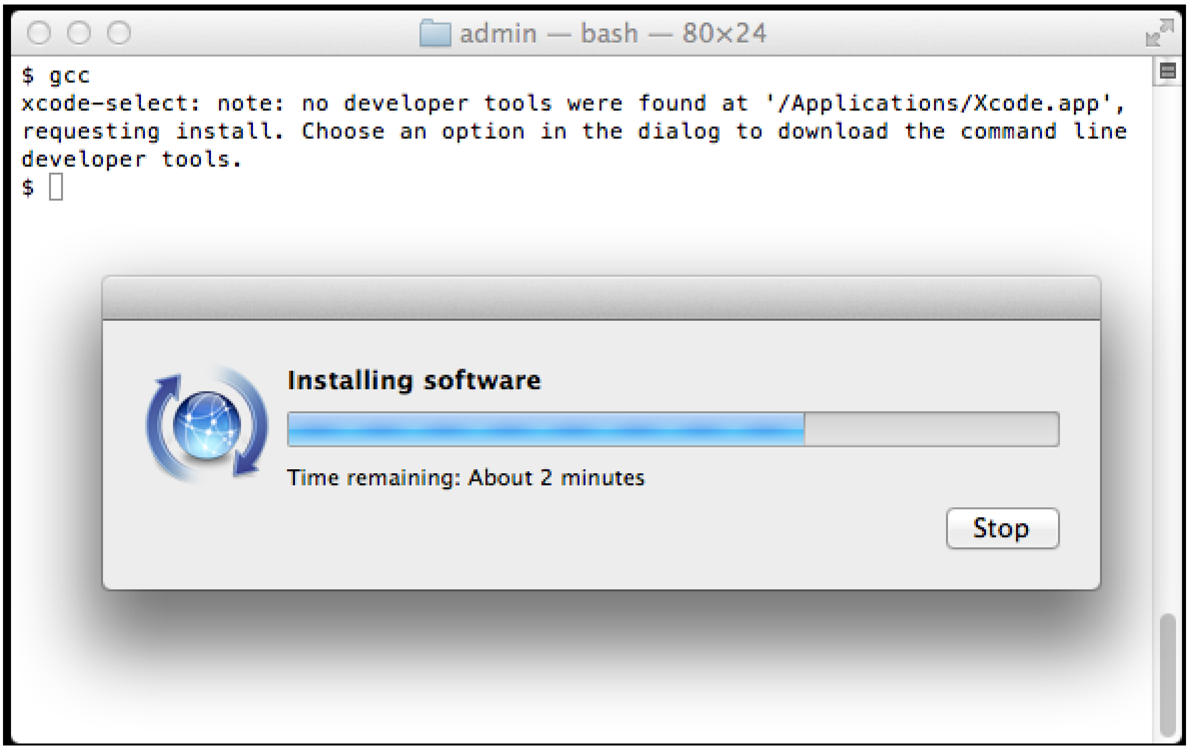
Choose Install to download and install the command line developer tools now.

Alternatively, you can use a command to install Xcode Command Line Tools. It will produce a similar alert box. Note the double hyphen:

```
$ xcode-select --install
```

Click "Install" to download and install Xcode Command Line Tools. If you have a slow Internet connection, it may take many minutes.

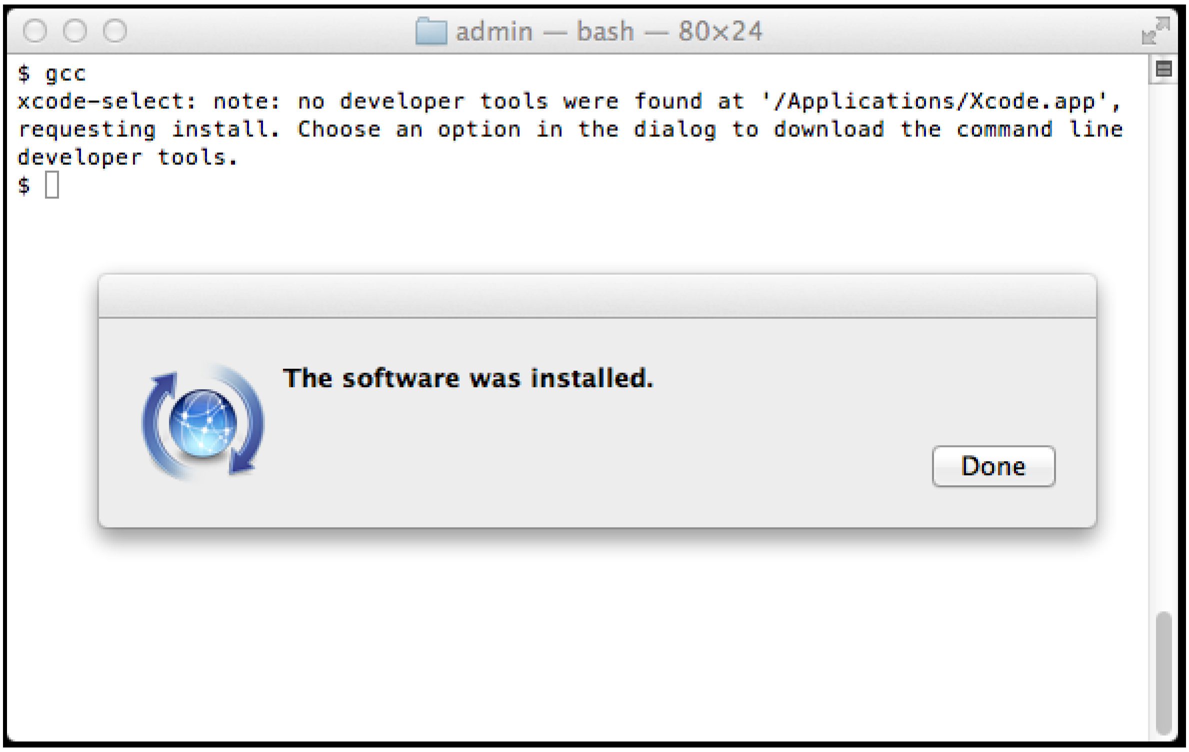
If the download takes a very long time (over an hour) or fails, you can try an alternative. Go to https://developer.apple.com/downloads/more and enter your Apple ID and password. You'll see a list of software packages you can download. Look for the latest version of *Command Line Tools* and click to download the *.dmg* file. Downloading the *.dmg* file is much faster than waiting for the command-line-based download. Install the *.dmg* file by clicking on the package icon. You'll need about 1.46GB of storage on your computer.



```
$ gcc
xcode-select: note: no developer tools were found at '/Applications/Xcode.app',
requesting install. Choose an option in the dialog to download the command line
developer tools.
$ 
```

Installing software

Time remaining: About 2 minutes



```
$ gcc
xcode-select: note: no developer tools were found at '/Applications/Xcode.app',
requesting install. Choose an option in the dialog to download the command line
developer tools.
$ 
```

The software was installed.

Verify that you've successfully installed Xcode Command Line Tools.

```
$ xcode-select -p
```

You should see:

```
/Library/Developer/CommandLineTools
```

The Xcode Command Line Tools must be installed before you go to the next steps. Just to be certain, verify that the Command Line Tools executables are available:

`$ /usr/sbin/pkgutil --packages | grep CL`

You should see:

`com.apple.pkg.CLTools_Executables`

Also try:

`/usr/sbin/pkgutil --pkg-info com.apple.pkg.CLTools_Executables`

You should see:

`package-id: com.apple.pkg.CLTools_Executables`

If the Command Line Tools executables are available, you'll have no problem compiling Nokogiri or any other gems that require native extensions.

Install Homebrew

Developers use Homebrew to install various Unix software packages, including chruby and ruby-install.

If you did not use a password to log in to your Mac (that is, if your password is blank), you cannot install Homebrew.

```
$
             command
```

`$ brew`

Check if Homebrew is installed.

If Homebrew is not installed, install it.

```
$                 "$(                                                                                            )"
```

`$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"`

The Homebrew installation script will ask you to enter your Mac password.

`Password:`

The Homebrew installation script may display a warning:

`WARNING: Improper use of the sudo command could lead to data loss...`

`To proceed, enter your password...`

Ignore the scary warning and enter your Mac password. You won't see the characters as you type. Press enter when you are done.

If you've installed Homebrew, you can check that Homebrew is installed properly.

```
$
```

`$ brew doctor`

Next we'll configure Git, an essential tool for any developer.

Configure Git

Before installing Ruby on Rails, you should configure Git. Git is automatically installed as part of the Xcode Command Line Tools.

Git provides a source control repository. Developers use Git to roll back code changes when needed, to collaborate with others, and deploy applications for hosting with a service such as Heroku. As a Rails developer, you'll use Git with a GitHub account for remote backup and collaboration. See the article Rails with Git and GitHub for more background.

Before you configure Git, it is a good idea to create an account on GitHub. It is important to use the same email address for Git and for GitHub. After you create an account on GitHub, follow their instructions to set up an SSH key so you don't have to enter a username and password every time you connect to GitHub from the Terminal.

```
$
                   (               )
```

Check that Git is installed:

`$ git version`

You should see `git version 2.21.1` (or newer).

If the Git version installed with Xcode Command Line Tools is old, you can use Homebrew to install the latest version of Git:

`$ brew install git`

You can use Homebrew to upgrade to the newest version of Git:

`$ brew upgrade git`

```
$
                 read             '/Users/.../.gitconfig'
$                               "Your Real Name"
$
$
         =
          =
```

Configure Git if you haven't used it before. First, list the current settings with the `git config -l --global` command. Then set `user.name` and `user.email` if necessary. Don't just copy and paste the code you see here. Edit it to add your name and the email address you've used for GitHub.

Now you'll be ready to use Git when you need it.

Install Ruby-install and Chruby

```
$
==>
######################################################################## 100.0%
==>
🍺
==>
######################################################################## 100.0%
==>
==>

  source

   enable

  source
==>
🍺
```

Use Homebrew to install the ruby-install utility and the chruby Ruby version manager.

`$ brew install ruby-install chruby`

You'll see beer mug emojis (part of the Emoji 1.0 standard released in 2015).

CONFIGURE CHRUBY

You must update your Unix *.zshrc* file so that the chruby program can be found.

The *.zshrc* file is in your user home folder. It is hidden in the file browser. You can enter Command + Shift + . to show hidden files. If you want to hide those files again, you can enter the same keystrokes.

You can force the Mac to always display hidden files by entering the following command in the Terminal application:

```
$
```

`$ defaults write com.apple.finder AppleShowAllFiles TRUE; killall Finder`

Hidden files will appear in gray in the Finder window.

Use your text editor (typically VS Codeor Atom) to open the *.zshrc* file. Check the instructions for your text editor to learn how to open files from the terminal. In VS Code, open the Command Palette (⇧⌘P) and type 'shell command' to locate and click "Shell Command: Install 'code' command in PATH command."

To open the *.zshrc* file with VS Code:

`$ code ~/.zshrc`

In Unix, the squiggle (tilde) character is a shortcut to your user home folder.

Add two lines anywhere in your *.zshrc* file. Your *.zshrc* file may not exist (or may be empty) if you have not used it before. Saving the file will create it.

```
### ~/.zshrc
source
source
```

`source /usr/local/share/chruby/chruby.sh`

`source /usr/local/share/chruby/auto.sh`

The first line makes the chruby program available in the shell. The second line will automatically switch the current version of Ruby when you change directories if a hidden file indicates a specific Ruby version.

You must close and reopen the Terminal window for the changes to the *.zshrc* file to be recognized.

Check that each program was installed successfully.

```
$

$
              [                      [
```

```
$ ruby-install -V
```

```
$ chruby -h
```

Now you can install the newest Ruby version.

Install Ruby

After installing the ruby-install utility, install the newest version of Ruby.

The Chruby version manager will use your shell to intercept any calls to Ruby. There's no need to remove the system Ruby. The system Ruby will remain on your system and your preferred version will take precedence.

You can check for the current recommended version of Ruby. Tell the ruby-install utility to check for the newest Ruby version.

```
$
```

```
$ ruby-install --latest
```

Then install the newest Ruby version.

```
$
```

```
$ ruby-install --latest ruby
```

Add a line to the *.zshrc* file to set the new Ruby version as a default.

```
chruby ruby-2.7.0
```

You'll need to close and re-open the terminal window.

Then verify that the newest version of Ruby is installed:

```
$
```

```
$ ruby -v
```

Now you can update the development environment. It is likely some gems have been updated since the newest version of Ruby was released. It's best if everything is up to date before you start a new project.

Check the Gem Manager

RubyGems is the package manager in Ruby. We use it to install software packages that add functionality to Ruby.

Check the installed gem manager version.

```
$
```

`$ gem -v`

You can check if a newer RubyGems version is available. Use `gem update --system` to upgrade the Ruby gem manager:

```
$
```

`$ gem update --system`

FASTER GEM INSTALLATION

By default, when you install gems, documentation files will be installed. Developers seldom use gem documentation files (they'll browse the web instead). Installing gem documentation files takes time, so many developers like to toggle the default so no documentation is installed.

Here's how to speed up gem installation by disabling the documentation step:

```
$ echo "gem: --no-document"
```

`$ echo "gem: --no-document" >> ~/.gemrc`

This adds the line `gem: --no-document` to the hidden **.gemrc** file in your home directory.

INSTALL NOKOGIRI

Nokogiri is a gem that is a dependency for many other gems. Nokogiri is a gem that requires compilation for your specific operating system. As such, if your system environment doesn't match Nokogiri's requirements, compilation of Nokogiri will fail. If your system is configured properly, you'll be able to compile Nokogiri. However, compilation takes time. Every time you install the Nokogiri gem, you'll wait (as long as five minutes).

To save time, install the Nokogiri gem:

```
$
```

`$ gem install nokogiri`

During installation, Nokogiri will display two lengthy messages in the console. It will also pause without displaying any progress for as long as five minutes. Don't assume installation has failed unless you see an error message or you've waited more than ten minutes.

If Nokogiri installation fails, it's likely your Xcode Command Line Tools are not installed correctly. Try installing the Xcode Command Line Tools from the Apple developer site, as described above in the section "Install Xcode Command Line Tools." The maintainers of the node-gyp build tool have provided Installation notes for macOS Catalina that are very helpful in resolving Xcode Command Line Tools installation issues. The Nokogiri maintainers also provide instructions for Installing Nokogiri.

Install Rails

Check for the current version of Rails. Rails 6.0.2 was current when this was written.

You can install Rails into the current Ruby environment.

Here are the options you have for installing Rails.

If you want to install the current stable release:

```
$
```

`$ gem install rails`

If you want the newest beta version or release candidate, you can install with `--pre`.

```
$
```

`$ gem install rails --pre`

Or you can get a specific version.

For example, if you want the Rails 5.2.3 release:

```
$                          =
```

`$ gem install rails --version=5.2.3`

Verify that the correct version of Rails is installed:

```
$
```

`$ rails -v`

`Rails 6.0.2`

Rails is now installed. If you want, you can close the Terminal window. Everything is installed, so you won't lose anything by closing the Terminal.

Next, try building a Rails application.

Create a Workspace Folder

You'll need a convenient folder to store your Rails projects. You can give it any name, such as **code/** or **projects/**. For this tutorial, we'll call it **workspace/**.

Create a projects folder and move into the folder:

```
$
$ cd
```

`$ mkdir workspace`

`$ cd workspace`

This is where you'll create your Rails applications.

New Rails Application

Here's how to create a new Rails application.

```
$
```

```
$ rails new myapp
```

We'll name the new application "myapp." Obviously, you can give it any name you like.

The `rails new` command generates the default Rails starter app.

QUICK TEST

For a "smoke test" to see if everything runs, display a list of Rake tasks.

```
$ cd
$
```

```
$ cd myapp
```

```
$ rails -T
```

Run the application with the Rails server command:

```
$
```

```
$ rails server
```

Use your web browser to visit the application at http://localhost:3000/.

Use Control-c to stop the server.

This concludes the instructions for installing Ruby and Rails. Read on for additional advice and tips.

Terminal Application and Text Editor

The built-in Apple Terminal application is adequate but many developers install iTerm 2. It has more options for customization.

For examining and writing code, you'll also need a text editor program such as VS Code or Atom. Of course, some programmers will suggest you try Vim or Emacs.

Databases for Rails

Rails uses the SQLite database by default. MacOS come with SQLite pre-installed and there's nothing to configure.

Though SQLite is adequate for development (and even some production applications), a new Rails application can be configured for other databases. The command `rails new myapp --database=` will show you a list of supported databases.

```
Supported for preconfiguration are: mysql, oracle, postgresql, sqlite3, frontbase, ibm_db, sqlserver, jdbcmysql, jdbcsqlite3, jdbcpostgresql, jdbc.
```

POSTGRESQL

Use the easy-to-install macOS Postgres.app if you'd like to use PostgreSQL.

To create a new Rails application to use PostgreSQL:

```
$ rails new myapp --database=postgresql
```

The `--database=postgresql` parameter will add the pg database adapter gem to the Gemfile and create a suitable config/database.yml file.

Deployment

Most developers deploy their applications with a "platform as a service" (PaaS) provider such as Heroku.

- Heroku
- DigitalOcean
- EngineYard
- Amazon AWS EC2

See a helpful comparison of Rails hosting services.

For deployment on Heroku, see the article:

- Rails Deploy to Heroku

Security

By design, Rails encourages practices that avoid common web application vulnerabilities. The Rails security team actively investigates and patches vulnerabilities. If you use the most current version of Rails, you will be protected from known vulnerabilities. See the Ruby On Rails Security Guide for an overview of potential issues and watch the Ruby on Rails Security Mailing List for announcements and discussion.

YOUR APPLICATION'S SECRET TOKEN

Rails uses a session store to provide persistence between page requests. The default session store uses cookies. To prevent decoding of cookie data and hijacking a session, Rails encrypts cookie data using a secret key. When you create a new Rails application using the `rails new` command, a unique secret key is generated.

The file **config/master.key** contains a secret token used to decrypt credentials (and other encrypted files).

Take care to hide the secret token you use in production. Don't expose it in a public GitHub repo, or people could change their session information, and potentially access your site without permission. It's best to set the secret token in a Unix shell variable.

If you need to create a new secret token:

```
$ rails secret
```

The command `rake secret` generates a new random secret you can use. The command won't install the key; you have to copy the key from the console output to the appropriate file.

Troubleshooting

CONFLICTING BUNDLER VERSIONS

If you install a version of Bundler that is newer than the default installed with Ruby, you will get a message "Warning: the running version of Bundler (2.1.2) is older than the version that created the lockfile (2.1.4)" when you run a Rails command. You can avoid the warning by prepending `bundle exec`, for example `bundle exec rails -T`. It's not possible to remove a default gem with `gem uninstall bundler` so it's advisable not to install a newer version of Bundler unless absolutely necesary.

PROBLEMS WITH "GEM::REMOTEFETCHER::FETCHERROR: SSL_CONNECT"

Ruby and RubyGems (starting with Ruby 1.9.3p194 and RubyGems 1.8.23) require verification of server SSL certificates when Ruby makes an Internet connection via https. If you run `rails new` and get an error "Gem::RemoteFetcher::FetchError: SSL_connect returned=1 errno=0 state=SSLv3 read server certificate" see this article suggesting solutions: OpenSSL errors and Rails.

PROBLEMS WITH "CERTIFICATE VERIFY FAILED"

Are you getting an error "OpenSSL certificate verify failed" when you try to generate a new Rails app from an application template? See this article suggesting solutions: OpenSSL errors and Rails.

INSTALLING IN A SECOND ACCOUNT

If you've created a second user account on your Mac, and the first user has already installed Homebrew, the first user may have to change permissions for the **/usr/local/bin** and **/usr/local/Cellar** folders:

```
$ sudo chmod g+w /usr/local/bin
```

```
$ sudo chmod g+w /usr/local/Cellar
```

The second user account should be given admin privileges using the System Preferences "Users and Groups" setting.

Where to Get Help

Your best source for online help with problems is Stack Overflow. Your issue may have been encountered and addressed by others.

To find a real person who can help, attend a Ruby meetup in your city. Google for "ruby meetup" with your city name.

Comments by CommentBox

Have a problem with Ruby on Rails installation? Got a suggestion for others?

Comments (0) best

Leave a comment

Sign In

Please sign in to create comments.

CommentBox