Installing Rails Core Development Dependencies

This guide covers how to set up an environment for Ruby on Rails core development.

After reading this guide, you will know:

- How to set up your machine for Rails development

1 Other Ways to Set Up Your Environment

If you don't want to set up Rails for development on your local machine, you can use Codespaces, the VS Code Remote Plugin, or rails-dev-box. Learn more about these options here.

2 Local Development

If you want to develop Ruby on Rails locally on your machine, see the steps below.

2.1 Install Git

Ruby on Rails uses Git for source code control. The Git homepage has installation instructions. There are a variety of resources online that will help you get familiar with Git.

2.2 Clone the Ruby on Rails Repository

Navigate to the folder where you want to download the Ruby on Rails source code (it will create its own `rails` subdirectory) and run:

```
$ git clone https://github.com/rails/rails.git
$ cd rails
```

2.3 Install Additional Tools and Services

Some Rails tests depend on additional tools that you need to install before running those specific tests.

Here's the list of each gems' additional dependencies:

- Action Cable depends on Redis
- Active Record depends on SQLite3, MySQL and PostgreSQL
- Active Storage depends on Yarn (additionally Yarn depends on Node.js), ImageMagick, libvips, FFmpeg, muPDF, Poppler, and on macOS also XQuartz.
- Active Support depends on memcached and Redis
- Railties depend on a JavaScript runtime environment, such as having Node.js installed.

Install all the services you need to properly test the full gem you'll be making changes to. How to install these services for macOS, Ubuntu, Fedora/CentOS, Arch Linux, and FreeBSD are detailed below.

Redis' documentation discourages installations with package managers as those are usually outdated. Installing from source and bringing the server up is straight forward and well documented on Redis' documentation.

Active Record tests *must* pass for at least MySQL, PostgreSQL, and SQLite3. Your patch will be rejected if tested against a single adapter, unless the change and tests are adapter specific.

Below you can find instructions on how to install all of the additional tools for different operating systems.

2.3.1 macOS

On macOS you can use Homebrew to install all of the additional tools.

To install all run:

```
$ brew bundle
```

You'll also need to start each of the installed services. To list all available services run:

```
$ brew services list
```

You can then start each of the services one by one like this:

```
$ brew services start mysql
```

Replace `mysql` with the name of the service you want to start.

2.3.2 Ubuntu

To install all run:

```
$ sudo apt-get update
$ sudo apt-get install sqlite3 libsqlite3-dev mysql-server libmysqlclient-dev
postgresql postgresql-client postgresql-contrib libpq-dev redis-server memcached
imagemagick ffmpeg mupdf mupdf-tools libxml2-dev libvips42 poppler-utils libyaml-
dev libffi-dev

# Install Yarn
# Use this command if you do not have Node.js installed
# ref: https://github.com/nodesource/distributions#installation-instructions
$ sudo mkdir -p /etc/apt/keyrings
$ curl --fail --silent --show-error --location
https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key | sudo gpg --dearmor -o
/etc/apt/keyrings/nodesource.gpg
$ echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg]
https://deb.nodesource.com/node_20.x nodistro main" | sudo tee
/etc/apt/sources.list.d/nodesource.list
$ sudo apt-get update
$ sudo apt-get install -y nodejs

# Once you have installed Node.js, install the yarn npm package
$ sudo npm install --global yarn
```

2.3.3 Fedora or CentOS

To install all run:

```
$ sudo dnf install sqlite-devel sqlite-libs mysql-server mysql-devel postgresql-
server postgresql-devel redis memcached ImageMagick ffmpeg mupdf libxml2-devel
vips poppler-utils

# Install Yarn
# Use this command if you do not have Node.js installed
# ref: https://github.com/nodesource/distributions#installation-instructions-1
$ sudo dnf install https://rpm.nodesource.com/pub_20/nodistro/repo/nodesource-
release-nodistro-1.noarch.rpm -y
$ sudo dnf install nodejs -y --setopt=nodesource-nodejs.module_hotfixes=1

# Once you have installed Node.js, install the yarn npm package
$ sudo npm install --global yarn
```

2.3.4 Arch Linux

To install all run:

```
$ sudo pacman -S sqlite mariadb libmariadbclient mariadb-clients postgresql
postgresql-libs redis memcached imagemagick ffmpeg mupdf mupdf-tools poppler yarn
libxml2 libvips poppler
$ sudo mariadb-install-db --user=mysql --basedir=/usr --datadir=/var/lib/mysql
$ sudo systemctl start redis mariadb memcached
```

If you are running Arch Linux, MySQL isn't supported anymore so you will need to use MariaDB instead (see this announcement).

2.3.5 FreeBSD

To install all run:

```
$ sudo pkg install sqlite3 mysql80-client mysql80-server postgresql11-client
postgresql11-server memcached imagemagick6 ffmpeg mupdf yarn libxml2 vips
poppler-utils
# postmaster databases/redis
```

Or install everything through ports (these packages are located under the `databases` folder).

If you run into problems during the installation of MySQL, please see the MySQL documentation.

2.3.6 Debian

To install all dependencies run:

```
$ sudo apt-get install sqlite3 libsqlite3-dev default-mysql-server default-
libmysqlclient-dev postgresql postgresql-client postgresql-contrib libpq-dev
redis-server memcached imagemagick ffmpeg mupdf mupdf-tools libxml2-dev libvips42
poppler-utils
```

If you are running Debian, MariaDB is the default MySQL server, so be aware there may be differences.

2.4 Database Configuration

There are couple of additional steps required to configure database-engines required for running Active Record tests.

PostgreSQL's authentication works differently. To set up the development environment with your development account, on Linux or BSD, you just have to run:

```
$ sudo -u postgres createuser --superuser $USER
```

and for macOS:

```
$ createuser --superuser $USER
```

MySQL will create the users when the databases are created. The task assumes your user is `root` with no password.

Then, you need to create the test databases for both MySQL and PostgreSQL with:

```
$ cd activerecord
$ bundle exec rake db:create
```

You can also create test databases for each database engine separately:

```
$ cd activerecord
$ bundle exec rake db:mysql:build
$ bundle exec rake db:postgresql:build
```

and you can drop the databases using:

```
$ cd activerecord
$ bundle exec rake db:drop
```

Using the Rake task to create the test databases ensures they have the correct character set and collation.

If you're using another database, check the file `activerecord/test/config.yml` or `activerecord/test/config.example.yml` for default connection information. You can edit `activerecord/test/config.yml` to provide different credentials on your machine, but you should not push any of those changes back to Rails.

2.5 Install JavaScript Dependencies

If you installed Yarn, you will need to install the JavaScript dependencies:

```
$ yarn install
```

2.6 Installing Gem Dependencies

Gems are installed with Bundler which ships by default with Ruby.

To install the Gemfile for Rails run:

```
$ bundle install
```

If you don't need to run Active Record tests, you can run:

```
$ bundle config set without db
$ bundle install
```

2.7 Contribute to Rails

After you've set up everything, read how you can start contributing.

Feedback

You're encouraged to help improve the quality of this guide.

Please contribute if you see any typos or factual errors. To get started, you can read our documentation contributions section.

You may also find incomplete content or stuff that is not up to date. Please do add any missing documentation for main. Make sure to check Edge Guides first to verify if the issues are already fixed or not on the main branch. Check the Ruby on Rails Guides Guidelines for style and conventions.

If for whatever reason you spot something to fix but cannot patch it yourself, please open an issue.

And last but not least, any kind of discussion regarding Ruby on Rails documentation is very welcome on the official Ruby on Rails Forum.

Chapters