

# Ruby on Rails Guides Guidelines

This guide documents guidelines for writing Ruby on Rails Guides. This guide follows itself in a graceful loop, serving itself as an example.

After reading this guide, you will know:

- About the conventions to be used in Rails documentation.
- How to generate guides locally.

## 1 Markdown

Guides are written in [GitHub Flavored Markdown](#). There is comprehensive [documentation for Markdown](#), as well as a [cheatsheet](#).

## 2 Prologue

Each guide should start with motivational text at the top (that's the little introduction in the blue area). The prologue should tell the reader what the guide is about, and what they will learn. As an example, see the [Routing Guide](#).

## 3 Headings

The title of every guide uses an `h1` heading; guide sections use `h2` headings; subsections use `h3` headings; etc. Note that the generated HTML output will use heading tags starting with `<h2>`.

```
Guide Title
=====

Section
-----

### Sub Section
```

When writing headings, capitalize all words except for prepositions, conjunctions, internal articles, and forms of the verb "to be".

```
#### Assertions and Testing Jobs inside Components
#### Middleware Stack is an Array
#### When are Objects Saved?
```

Use the same inline formatting as regular text:

```
#### The `:content_type` Option
```


## 4 Notes, Tips and Warnings

Sometimes a paragraph deserves a little more attention. For example, to clarify a common misunderstanding or warn about something that could break an application.

To highlight a paragraph, prefix it with `NOTE:`, `TIP:` or `WARNING::`

```
NOTE: Use `NOTE`, `TIP` or `WARNING` to highlight a paragraph.
```


This will wrap the paragraph in a special container resulting in the following:

```
 Use NOTE, TIP or WARNING to highlight a paragraph.
```

### 4.1 NOTE


Use `NOTE` to highlight something in relation to the subject and the context. Reading it will help your understanding of that subject or context, or clarify an important item.

For example, a section describing locale files could have the following `NOTE`:


```
 You need to restart the server when you add new locale files.
```

### 4.2 TIP

A `TIP` is just an additional bit of information regarding the subject, but not necessarily relevant to the understanding. It can point you to another guide or website:


```
 To learn more about routing, see Rails Routing from the Outside In.
```

Or show a helpful command to see more options to dig deeper:


```
 For further help with generators, run bin/rails generate --help.
```

### 4.3 WARNING

Use `WARNING` for things to avoid that could break the application:

```
 Refrain from using methods like update, save, or any other methods that cause side effects on the object within your callback methods.
```

Or warn about things that could compromise your application's security.

```
 Keep your master key safe. Do not commit your master key.
```

## 5 Links

Use descriptive links and avoid "here" and "more" links:

```
# BAD
See the Rails Internationalization (I18n) API documentation for [more details](i18n.html).

# GOOD
See the [Rails Internationalization (I18n) API documentation](i18n.html) for more details.
```

Use descriptive links for internal links as well:

```
# BAD
We will cover this [below](#multiple-callback-conditions).

# GOOD
We will cover this in the [multiple callback conditions section](#multiple-callback-conditions) shown below.
```

### 5.1 Linking to the API

Links to the API ([api.rubyonrails.org](#)) are processed by the guides generator in the following manner:

Links that include a release tag are left untouched. For example

```
https://api.rubyonrails.org/v5.0.1/classes/ActiveRecord/Attributes/ClassMethod
```

is not modified.

Please use these in release notes, since they should point to the corresponding version no matter the target being generated.

If the link does not include a release tag and edge guides are being generated, the domain is replaced by [edgeapi.rubyonrails.org](#). For example,

```
https://api.rubyonrails.org/classes/ActionDispatch/Response.html
```

becomes

```
https://edgeapi.rubyonrails.org/classes/ActionDispatch/Response.html
```

If the link does not include a release tag and release guides are being generated, the Rails version is injected. For example, if we are generating the guides for v5.1.0 the link

```
https://api.rubyonrails.org/classes/ActionDispatch/Response.html
```

becomes

```
https://api.rubyonrails.org/v5.1.0/classes/ActionDispatch/Response.html
```

Please don't link to [edgeapi.rubyonrails.org](#) manually.

## 6 Column Wrapping

Do not reformat old guides just to wrap columns. But new sections and guides should wrap at 80 columns.

## 7 API Documentation Guidelines

The guides and the API should be coherent and consistent where appropriate. In particular, these sections of the [API Documentation Guidelines](#) also apply to the guides:

- [Wording](#)
- [English](#)
- [Example Code](#)
- [Filenames](#)
- [Fonts](#)

## 8 HTML Guides

Before generating the guides, make sure that you have the latest version of Bundler installed on your system. To install the latest version of Bundler, run `gem install bundler`.

If you already have Bundler installed, you can update with `gem update bundler`.

### 8.1 Generation

To generate all the guides, just `cd` into the `guides` directory, run `bundle install`, and execute:

```
$ bundle exec rake guides:generate
```

or

```
$ bundle exec rake guides:generate:html
```

Resulting HTML files can be found in the `./output` directory.

To process `my_guide.md` and nothing else use the `ONLY` environment variable:

```
$ touch my_guide.md
$ bundle exec rake guides:generate ONLY=my_guide
```

By default, guides that have not been modified are not processed, so `ONLY` is rarely needed in practice.

To force processing all the guides, pass `ALL=1`.

If you want to generate guides in a language other than English, you can keep them in a separate directory under `source` (e.g. `source/es`) and use the `GUIDES_LANGUAGE` environment variable:

```
$ bundle exec rake guides:generate GUIDES_LANGUAGE=es
```

If you want to see all the environment variables you can use to configure the generation script just run:

```
$ rake
```

### 8.2 Validation

Please validate the generated HTML with:

```
$ bundle exec rake guides:validate
```

Particularly, titles get an ID generated from their content and this often leads to duplicates.

## 9 Kindle Guides

### 9.1 Generation

To generate guides for the Kindle, use the following rake task:

```
$ bundle exec rake guides:generate:kindle
```

### Chapters

1. [Markdown](#)
2. [Prologue](#)
3. [Headings](#)
4. [Notes, Tips and Warnings](#)
  - [NOTE](#)
  - [TIP](#)
  - [WARNING](#)
5. [Links](#)
  - [Linking to the API](#)
6. [Column Wrapping](#)
7. [API Documentation Guidelines](#)