

Rails

This plugin adds completion for Ruby On Rails Framework and Rake commands, as well as some aliases for logs and environment variables.

To use it, add `rails` to the `plugins` array in your `zshrc` file:

```
plugins=(... rails)
```

List of Aliases

Rails aliases

Alias	Command	Description
<code>rc</code>	<code>rails console</code>	Interact with your Rails app from the CLI
<code>rccs</code>	<code>rails console</code> <code>--sandbox</code>	Test code in a sandbox, without changing any data
<code>rd</code>	<code>rails destroy</code>	Undo a generate operation
<code>rdb</code>	<code>rails dbconsole</code>	Interact with your db from the console
<code>rdc</code>	<code>rails db:create</code>	Create the database
<code>rdd</code>	<code>rails db:drop</code>	Delete the database
<code>rdm</code>	<code>rails db:migrate</code>	Run pending db migrations
<code>rdmd</code>	<code>rails db:migrate:down</code>	Undo specific db migration
<code>rdmr</code>	<code>rails db:migrate:redo</code>	Redo specific db migration
<code>rdms</code>	<code>rails</code> <code>db:migrate:status</code>	Show current db migration status
<code>rdmtc</code>	<code>rails db:migrate</code> <code>db:test:clone</code>	Run pending migrations and clone db into test database
<code>rdmu</code>	<code>rails db:migrate:up</code>	Run specific db migration
<code>rdr</code>	<code>rails db:rollback</code>	Roll back the last migration
<code>rdrs</code>	<code>rails db:reset</code>	Delete the database and set it up again
<code>rds</code>	<code>rails db:seed</code>	Seed the database
<code>rdsl</code>	<code>rails db:schema:load</code>	Load the database schema
<code>rdtc</code>	<code>rails db:test:clone</code>	Clone the database into the test database
<code>rdtp</code>	<code>rails db:test:prepare</code>	Duplicate the db schema into your test database
<code>rgen</code>	<code>rails generate</code>	Generate boilerplate code
<code>rgm</code>	<code>rails generate</code> <code>migration</code>	Generate a db migration
<code>rlc</code>	<code>rails log:clear</code>	Clear Rails logs
<code>rmd</code>	<code>rails middleware</code>	Interact with Rails middlewares
<code>rn</code>	<code>rails notes</code>	Search for notes (<code>FIXME</code> , <code>TODO</code>) in code comments
<code>rp</code>	<code>rails plugin</code>	Run a Rails plugin command
<code>rr</code>	<code>rails routes</code>	List all defined routes
<code>rrg</code>	<code>rails routes \</code> <code> grep</code>	List and filter the defined routes

Alias	Command	Description
<code>rs</code>	<code>rails server</code>	Launch a web server
<code>rsb</code>	<code>rails server --bind</code>	Launch a web server binding it to a specific IP
<code>rsd</code>	<code>rails server --debugger</code>	Launch a web server with debugger
<code>rsp</code>	<code>rails server --port</code>	Launch a web server and specify the listening port
<code>rsts</code>	<code>rails stats</code>	Print code statistics
<code>rt</code>	<code>rails test</code>	Run Rails tests
<code>ru</code>	<code>rails runner</code>	Run Ruby code in the context of Rails

Foreman

Alias	Command	Description
<code>fmns</code>	<code>foreman start</code>	Interact with your Rails app from the CLI

Utility aliases

Alias	Command	Description
<code>devlog</code>	<code>tail -f log/development.log</code>	Show and follow changes to the development log
<code>prodlog</code>	<code>tail -f log/production.log</code>	Show and follow changes to the production log
<code>testlog</code>	<code>tail -f log/test.log</code>	Show and follow changes to the test log

Environment settings

Alias	Command	Description
<code>RED</code>	<code>RAILS_ENV=development</code>	Sets <code>RAILS_ENV</code> to development
<code>REP</code>	<code>RAILS_ENV=production</code>	Sets <code>RAILS_ENV</code> to production
<code>RET</code>	<code>RAILS_ENV=test</code>	Sets <code>RAILS_ENV</code> to test

These are global aliases. Use in combination with a command or just run them separately. For example: `REP rake db:migrate` will migrate the production db.

Legacy

Rake aliases

The following commands are run using `rails` instead of `rake` since Rails v5, but are preserved under the prefix `rk` for backwards compatibility.

Alias	Command	Description
rkdc	<code>rake db:create</code>	Create the database
rkdd	<code>rake db:drop</code>	Delete the database
rkdm	<code>rake db:migrate</code>	Run pending db migrations
rkdms	<code>rake db:migrate:status</code>	Show current db migration status
rkdmtdc	<code>rake db:migrate db:test:clone</code>	Run pending migrations and clone db into test database
rkdr	<code>rake db:rollback</code>	Roll back the last migration
rkdrs	<code>rake db:reset</code>	Delete the database and set it up again
rkds	<code>rake db:seed</code>	Seed the database
rkdsl	<code>rake db:schema:load</code>	Load the database schema
rkdtc	<code>rake db:test:clone</code>	Clone the database into the test database
rkdtp	<code>rake db:test:prepare</code>	Duplicate the db schema into your test database
rkclc	<code>rake log:clear</code>	Clear Rails logs
rkmd	<code>rake middleware</code>	Interact with Rails middlewares
rkn	<code>rake notes</code>	Search for notes (FIXME, TODO) in code comments
rksts	<code>rake stats</code>	Print code statistics
rkt	<code>rake test</code>	Run Rails tests

Other

Alias	Command
sc	<code>ruby script/console</code>
sd	<code>ruby script/destroy</code>
sd	<code>ruby script/server --debugger</code>
sg	<code>ruby script/generate</code>
sp	<code>ruby script/plugin</code>
sr	<code>ruby script/runner</code>
ssp	<code>ruby script/spec</code>
sstat	<code>thin --stats "/thin/stats" start</code>

- `remote_console <server> <directory>`: runs `ruby script/console` production on a remote server.