

# Rails 5 brings consistency by wrapping all rake commands using rails

January 14, 2016 - Mohit Natoo

*This blog is part of our Rails 5 series  
(<http://blog.bigbinary.com/categories/Rails-5>).*

In Rails 4 some commands start with `rails` and some commands start with `rake`. This could be quite confusing for people new to Rails. Let's see an example.

Our task is to write a database migration and then to run that migration.

```
rails g migration create_users
```

Above command creates a migration. Now we need to run that migration.

```
rake db:migrate
```

As you can see first command starts with `rails` and second command starts with `rake`.

In order to consolidate them we can either use `rails` for everything or we can use `rake` for everything.

## Choosing rails command over rake command

Some favor using `rake` over `rails`. But an important feature missing in Rake is the ability to pass arguments.

```
rails console development
```

In order to execute the above command using `rake` we will have to pass `console` and `development` as arguments. We can pass these values using environment variables. That would mean adding additional code in Rake task to fetch right values and then only we will be able to invoke command `rails console development`.

## Rails 5 enables executing rake commands with rails

Rails core team decided (<https://github.com/rails/rails/pull/22288>) to have consistency by enabling `rails` command to support everything that `rake` does.

For example in Rails 5 commands like `db:migrate`, `setup`, `test` etc which are part of `rake` command in Rails 4 are now being supported by `rails` command. However you can still choose to use `rake` to run those commands similar to how they were run in Rails 4. This is because Rails community has introduced Rake Proxy ([https://github.com/rails/rails/blob/f718e52bcce02bc137263ead3a9d9f5df1c42c37/railties/lib/rails/commands/rake\\_proxy.rb](https://github.com/rails/rails/blob/f718e52bcce02bc137263ead3a9d9f5df1c42c37/railties/lib/rails/commands/rake_proxy.rb)) instead of completely moving the command options from `rake` to `rails`.

What happens internally is that when `rails db:migrate` command is executed, Rails checks if `db:migrate` is something that `rails` natively supports or not. In this case `db:migrate` is not natively supported by `rails` , so Rails delegates the execution to Rake via Rake Proxy.

If you want to see all the commands that is supported by `rails` in Rails 5 then you can get a long list of options by executing `rails --help` .

## Use app namespace for framework tasks in Rails 5

As `rails` command is now preferred over `rake` command, few rails namespaced framework tasks started looking little odd.

```
$ rails rails:update  
  
$ rails rails:template  
  
$ rails rails:templates:copy  
  
$ rails rails:update:configs  
  
$ rails rails:update:bin
```

So, Rails team decided to change the namespace for these tasks from `rails` to `app` .

```
$ rails app:update  
  
$ rails app:template  
  
$ rails app:templates:copy  
  
$ rails app:update:configs  
  
$ rails app:update:bin
```

Using `rails rails:update` will now give deprecation warning like:

```
DEPRECATION WARNING: Running update with the rails: namespace is  
deprecated in favor of app: namespace. Run bin/rails app:update  
instead .
```

## More improvements in pipeline

In Rails 4, the routes are usually searched like this.

```
$ rake routes | grep pattern
```

There is an effort underway

(<https://github.com/rails/rails/issues/18902>) to have a Rails command which might work as shown below.

```
$ rails routes -g pattern
```


There is also an effort to enable lookup for controller


(<https://github.com/rails/rails/pull/20420>) like this.

```
$ rails routes -c some_controller
```

---

Category: Ruby (/categories/Ruby) , Rails (/categories/Rails) , Rails 5 (/categories/Rails-5)

Share this article: 

([https://twitter.com/intent/tweet?text=Rails 5 brings consistency by wrapping all rake commands using rails&url=http://blog.bigbinary.com/2016/01/14/rails-5-supports-rake-commands-using-rails.html&via=bigbinary](https://twitter.com/intent/tweet?text=Rails+5+brings+consistency+by+wrapping+all+rake+commands+using+rails&url=http://blog.bigbinary.com/2016/01/14/rails-5-supports-rake-commands-using-rails.html&via=bigbinary))   
(<https://facebook.com/sharer.php?u=http://blog.bigbinary.com/2016/01/14/rails-5-supports-rake-commands-using-rails.html>)

Mohit Natoo

---

**Subscribe to our monthly newsletter.**

Follow @BigBinary



Start the discussion...

Be the first to comment.

« Previous (/2016/01/11/per-form-csrf-token-in-rails-5.html)

Next » (/2016/01/23/configure-postgresql-to-allow-remote-connection.html)

## BIG BINARY

Services (<http://bigbinary.com/services>)

Presentations (<http://bigbinary.com/presentations>)

Books (<http://bigbinary.com/books>)

Videos (<http://bigbinary.com/videos>)

Blog (<http://blog.bigbinary.com>)

## OUR SERVICES

Ruby on Rails (<http://bigbinary.com/ruby-on-rails-consulting>)

ReactJS (<http://bigbinary.com/reactjs-consulting>)

iOS & Android apps (<http://bigbinary.com/mobile-development-using-react-native>)

Workshops (<http://bigbinary.com/workshops>)

(<http://github.com/BigBinary>)

(<http://twitter.com/BigBinary>)

MIAMI ([HTTP://BIGBINARY.COM/MIAMI](http://bigbinary.com/miami))

SAN FRANCISCO ([HTTP://BIGBINARY.COM/SAN-FRANCISCO](http://bigbinary.com/san-francisco))

PUNE ([HTTP://BIGBINARY.COM/PUNE](http://bigbinary.com/pune))