

# Unit 10 Report

---

I started this report with the intention of looking more in-depth as to how scaffolding works in Rails. Scaffolding interested me because it made creating an entire structure (a controller, a helper, a model, multiple database changes, etc) a breeze with just a single command. In the end, my research led me to a more general topic – generators. It turns out that scaffolding is just one of many Rails generators that can be used on your application.

- A Rails generator can be run from the command line and be given a series of options
- A generator is written in Ruby and uses the command line options to create a series of file additions and updates within an application
- To see what generators are available for your app, you can run the command “rails generate”
- The most important generators are:
  - Controller
    - Ex. “rails generate controller People”
  - Helper
    - Ex. “rails generate helper Address”
  - Mailer
    - Ex. “rails generate mailer UserMailer”
  - Migration
    - Ex. “rails generate migration AddCityToPerson”
  - Model
    - Ex. “rails generate model Person name:string age:integer”
  - Scaffold
    - Ex. “rails generate scaffold User name:string email:string”
- In addition to these important generators, additional generators can be created and/or customized
  - Here is some example code of a simple custom generator named “initializer”

```
class InitializerGenerator < Rails::Generators::Base
  def create_initializer_file
    create_file "config/initializers/initializer.rb", "# Add
initialization content here"
  end
end
```
  - This also gives the ability to use more advanced generators, which be installed via a gem and then used while developing your application

Links used:

<http://guides.rubyonrails.org/generators.html>

[http://en.wikibooks.org/wiki/Ruby\\_on\\_Rails/Built-In\\_Rails\\_Tools/Generators](http://en.wikibooks.org/wiki/Ruby_on_Rails/Built-In_Rails_Tools/Generators)