

# Chapter 3 Notes:

## CREATING THE `sample_app`:

- Make sure you're in the `workspace`
- `rails _5.0.1_ new sample_app`
- `cd sample_app`
- Edit Gemfile
- `bundle update`
- `bundle install --without production`
- `git init`
- `git add -A`
- `git commit -m "Initialize repository"`
- Edit the README.md
- `git commit -am "Improve the README"` ← -am = -a (all) and -m (message) tags combined
- **Manually** Create Repository on BitBucket (ideally named the same as the app name in Cloud9)

### THEN DO THIS: (author of the tutorial recommends)

- `git remote add origin git@bitbucket.org:<username>/sample_app.git`
- `git push -u origin --all`

### OR DO THIS: (BitBucket recommends)

- `git remote add origin ssh://git@bitbucket.org/<username>/sample_app.git`
- `git push -u origin master`

Both ways work, but don't just believe me, test both! The differences are **in red!**

### NEXT:

- Add the "Hello world!" function in `app/controllers/application_controller.rb` (Listing 3.4)
- Edit the `config/routes.rb` to include `'application#hello'`
- `git commit -am "Add hello"`
- `git push`
- `heroku create`
- `git push heroku master`
- **`heroku logs` - if you run into problems at Heroku, use this command to start fixing them!**

### NEW Material:

**Static pages** - web pages with only *static* HTML, which means the web page's content will not change (as opposed to *dynamic* HTML which allows the web page to change when certain events occur)

- `git checkout -b static-pages` (Makes a new BRANCH called "static-pages")
- `rails generate controller StaticPages home help` (Generate StaticPages controller with Home and Help actions, which we'll turn into web pages)
- `git add -A`
- `git commit -m "Add a Static Pages controller"`
- `git push -u origin static-pages`
- `git push`
- Edit HTML in created Home and Help pages found in `app/views/static_pages`

### Writing Your First Test And Finishing the App: (and creating the About page the hard way)

- `ls test/controllers/` (Should list the test file)
- Add this to `test/controllers/static_pages_controller_test.rb`:

- test "should get about" do
- get static\_pages\_about\_url
- assert\_response :success
- end
- **rails test** (Will tell you how many tests the program is passing or failing. The author uses this command to tell us when we've finished fully implementing the About page.)
- Add `get 'static_pages/about'` to `config/routes.rb`
- Add empty `about` function to `app/controllers/static_pages_controller.rb`
- Create `about.html.erb` file in `app/views/static_pages`.  
**`touch app/views/static_pages/about.html.erb`** is the command to create the file if you don't manually create it in the Cloud9 file editing system.
- `mv app/views/layouts/application.html.erb layout_file` (m for move; won't normally do this)
- Add `assert_select "title", "Home | Ruby on Rails Tutorial Sample App"` to each test
- `rails test` (Should return 3 tests and 3 failures.) Pass the tests by editing the HTML of the three web pages.
- Change the `root` in `routes.rb` to `'static_pages#home'` to make the `home` the homepage.
- *NOTE: I did leave out some HTML editing and testing. It isn't new information necessarily if you're familiar with HTML. If you are not familiar with HTML, the author suggests a site to check out!*
- `mv layout_file app/views/layouts/application.html.erb` (Don't normally use this again)
- `git add -A`
- `git commit -m "Finish static pages"`
- `git checkout master`
- `git merge static-pages`
- `git push`
- `rails test`
- `git push heroku`
- The app is deployed!

**My deployed sample\_app: (Chapter 4 overwrites the Chapter 3 sample\_app)**

<https://apple-crisp-25148.herokuapp.com/> (I like the standard URLs for some reason!)

#### **HOW TO RUN APP LOCALLY without git pushing to BitBucket and Heroku:**

`rails server -b $IP -p $PORT`

Click on Window → click on Share → click on the Application link under "Links to Share" → Open  
**Don't forget to Ctrl-C in the terminal when you're done!** Otherwise, you'll get many errors later!

#### **HOW TO UNDO CODE GENERATION:**

To undo a "`rails generate ...`" call, use "`rails destroy ...`"

`rails generate controller StaticPages home help`

`rails destroy controller StaticPages home help`

AND/OR:

`rails generate model User name:string email:string`

`rails destroy model User`

To undo a "`rails db:migrate`" call, use "`rails db:rollback`" to undo 1 migration or "`rails db:migrate VERSION=0`" to undo all migrations. Note: the '0' can be changed to 1 or 2 or... based on how many migrations you've done. The counter starts at 0 and is incremented every time you do a migrate.