

Chapter 1 - From Zero to Deploy

What we'll learn this chapter:

- Ruby on Rails is a **web development framework** written in **Ruby**.
- We create a Rails **Hello, world** application.
- Use **Rails commands** to do things like create new applications (rails new) or test your code on a local server (rails server).
- We'll use **Cloud9** as a sweet cloud IDE (c9.io); the Rails tutorial is pre-installed!
- We'll use **Git** for source code control (versions and sharing).
- We'll store our git repo at **Bitbucket** (bitbucket.org).
- We'll deploy our application using **Heroku** (www.heroku.com).

That's 3 new website accounts: Cloud9, Bitbucket, and Heroku. Each accounts mean another new password. I use a password algorithm, so I only have to remember that.

1.1 Introduction

"Ruby on Rails is 100% open-source"; 1000's of developers creating gems

1.2 Up and running

Yikes! "Even for experienced Rails developers, installing Ruby, Rails, and all the associated supporting software can be an exercise in frustration."

Create a C9 account. Cloud IDE with editor, files, and Unix command shell. Rails tutorial setup already installed. Set 2 spaces as default in editor. Idiom!

You can always install locally on your machine: <http://installrails.com/>

Install rails: `gem install rails -v 5.0.1`

1.3 The first application

Table 1.1 - common Unix commands

Create new Rails app: `rails _5.0.1_ new hello_app`

Table 1.2 - default Rails directories

"After creating a new Rails application... use **Bundler** to install and include the gems needed by the app"

Grabbed his real Gemfile from gemfiles-4th-ed.railstutorial.org.

Install gems: `bundle install`; it failed and I needed to do `bundle update`; and that worked

Run locally in Cloud9: `rails server -b $IP -p $PORT`

MVC = Model View Controller; the core of how Rails apps are organized

Model = database representation

View = template converted dynamically to web page (HTML)

Controller = the logic/code of your app

routes.rb file defines web page routing:

```
root 'application#hello'
```

1.4 Version control with Git

git - distributed source code control; means sharing + version control

Some commands:

- `git init` # first step for new repo
- `git add -A` # add all files
- `git status` # status of staging area
- `git commit -m "Initialize repository"` # commit to keep changes

bitbucket - hosts git repos; create account; connect to git

- `git remote add origin git@bitbucket.org:<username>/hello_app.git`
- `git push -u origin --all` # push files to bitbucket repo

Hartl's **git workflow**: branch, edit, commit, merge

- `git checkout -b modify-README` # create a new branch
- `git branch` # list/report git branches
- # edit the README or whatever files you want
- `git commit -a -m "Improve the README file"` # commit to accept changes
- `git checkout master` # switch back to 'master'
- `git merge modify-README` # merge changes from old branch
- `git push` # push changes to bitbucket

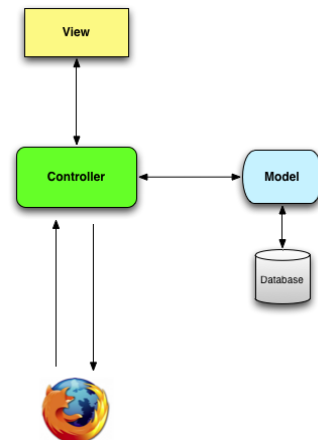
1.5 Deploying

Create heroku account

`bundle install --without production` # to prevent install of production gems locally (?)

- `heroku login` # username + password
- `heroku keys:add` # establish ssh key link
- `heroku create` # create new app at Heroku
- `git push heroku master` # deploy

For a nice url, I used: `heroku rename profbill-hello`



Heroku is cool. Hartl uses it for the Rails tutorial.

“In addition to supporting subdomains, Heroku also supports custom domains. (In fact, the Ruby on Rails Tutorial site lives at Heroku; if you’re reading this book online, you’re looking at a Heroku-hosted site right now!)”

1.6 Conclusion

Done. My notes went just a little over 2 pages. Whatev.

My hello URL is:

profbill-hello.herokuapp.com/