Ch 2 notes - toy_app

My tutorial notes… be brief, H3 for each section, bold terms etc, code is italics
The tutorial: www.railstutorial.org/. Chapter 2 builds an application called **toy_app**.

2.1 Planning the application

Can do multiple apps in one C9 workspace.
Start new Rails app: *rails _5.0.0.1_ new toy_app*
Install gems locally: *bundle install…*
Init git; create repo on bitbucket; push to bitbucket repo;
deploy "hello" dummy to heroku because "it's never too early to deploy"

2.2 The Users resource

Create User with scaffolding: *rails generate scaffold User name:string email:string*
Update database to reflect User: *rails db:migrate*
Very important - MVC in Action, Figure 2.11
resources command creates Restful connections between URL and controller method:
resources users # in config/routes.rb
Table 2.2 is **important** too. URL + HTTP request => Action in Rails

HTTP request	URL	Action	Purpose
GET	/users	index	page to list all users
GET	/users/1	show	page to show user with id 1
GET	/users/new	new	page to make a new user
POST	/users	create	create a new user
GET	/users/1/edit	edit	page to edit user with id 1
PATCH	/users/1	update	update user with id 1
DELETE	/users/1	destroy	delete user with id 1

Table 2.2: RESTful routes provided by the Users resource in Listing 2.5.

Representational State Transfer (REST) - app resources can be created, read, updated and deleted (CRUD); corresponds to 4 HTTP requests: POST, GET, PATCH, and DELETE.
Instance variables start with the @ sign and are automatically available in views.

2.3 The Microposts resource

Create MicroPost with scaffolding:

rails generate scaffold Micropost content:text user_id:integer
rails db:migrate

First validations in data model: *validates :content, length: { maximum: 140 }*
First data model associations:

has_many :microposts # in User data model
belongs_to :user # in micropost data model

Rails command line access: *rails console*
Inheritance through < operator: *class User < ApplicationRecord*
Remember this **deploy command sequence!**

git status
git add -A
git commit -m "Finish toy app"
git push # push changes up to bitbucket

git push heroku # push changes up to heroku and deploy!
heroku run rails db:migrate # must migrate DB changes too!

Notice on heroku… your data model is there/changed, but your data was **not** sent over. Only the data model was migrated, not the data itself.

2.4 Conclusion

Run it: http://toyapp694.herokuapp.com/
This chapter introduced: scaffolding, MVC, REST architecture, some data modeling with validations and associations.
thanks… yow, bill