Ruby on Rails
Sustainable productivity for web-application development
EVANGELISM BY: JUSTIN PALMER (HTTP://ENCYTEMEDIA.COM)

Ostrya Labs

Clean Code Matters

Agenda – Day 3

- Theory -
    - ReSTful Architecture
    - Command lines
    - Database
    - Console Data Manipulations
    - Git Version Control
    - Deploy on Heroku

Agenda – Day 3

- Lab -
    - http://ruby.railstutorial.org/ruby-on-rails-tutorial-book
        - Chapter 2
        - Chapter 3
        - Chapter 4
        - Chapter 5
        - Chapter 6

REpresentational State Transfer

- Web Services are viewed as resources
- Can be uniquely identified by their URLs
- Explicit use of HTTP methods to denote the invocation of different operations
- highly reusable across platforms since they rely on basic HTTP protocol
- being preferred for integration with backend enterprise services

REpresentational State Transfer

CRUD Principle

- POST - Create a resource
- GET - Retrieve a resource
- PUT – Update a resource
- DELETE - Delete a resource

REpresentational State Transfer

CRUD Principle

- POST - ../products
- GET - ../products/:id
- PUT – ../products/:id
- DELETE - ../products/:id
- ../products/new
- ../products/id:/edit

Rails command lines

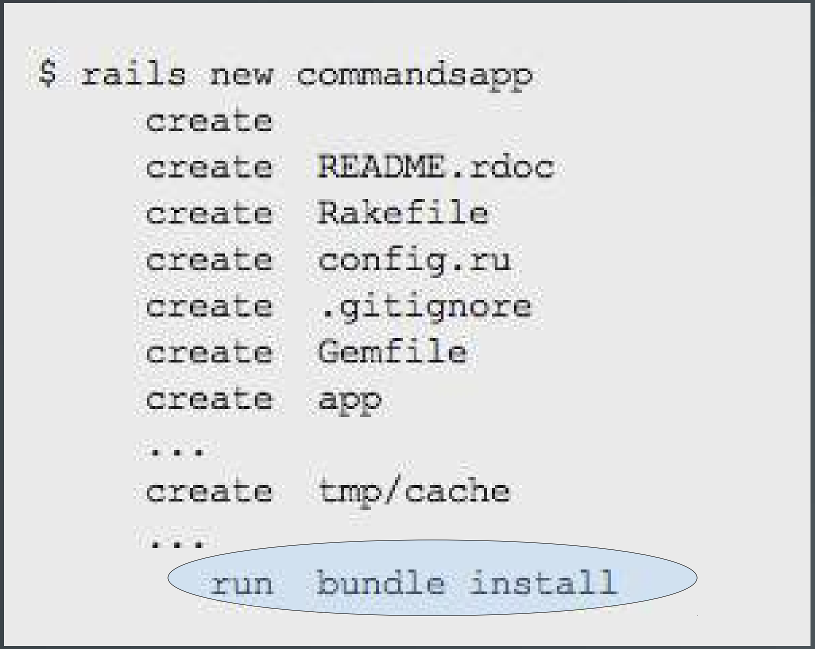
Command Line options

- rails new
- rails server
- rails generate and destroy
- rails console
- rake – ruby make

Command Line Basics

rails new

rails new commandsapp

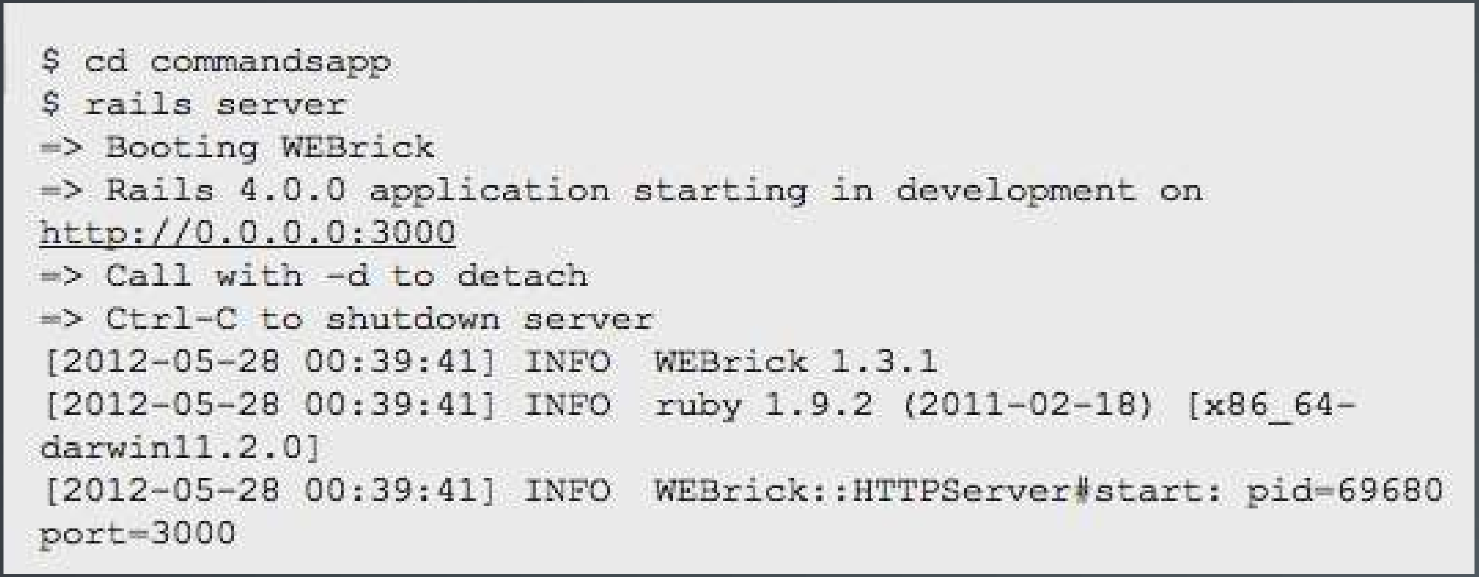


```
$ rails new commandsapp
     create
     create  README.rdoc
     create  Rakefile
     create  config.ru
     create  .gitignore
     create  Gemfile
     create  app
     ...
     create  tmp/cache
     ...
        run  bundle install
```

Command Line Basics

rails server

rails server [rails s]



```
$ cd commandsapp
$ rails server
=> Booting WEBrick
=> Rails 4.0.0 application starting in development on
http://0.0.0.0:3000
=> Call with -d to detach
=> Ctrl-C to shutdown server
[2012-05-28 00:39:41] INFO  WEBrick 1.3.1
[2012-05-28 00:39:41] INFO  ruby 1.9.2 (2011-02-18) [x86_64-
darwin11.2.0]
[2012-05-28 00:39:41] INFO  WEBrick::HTTPServer#start: pid=69680
port=3000
```

http://localhost:3000

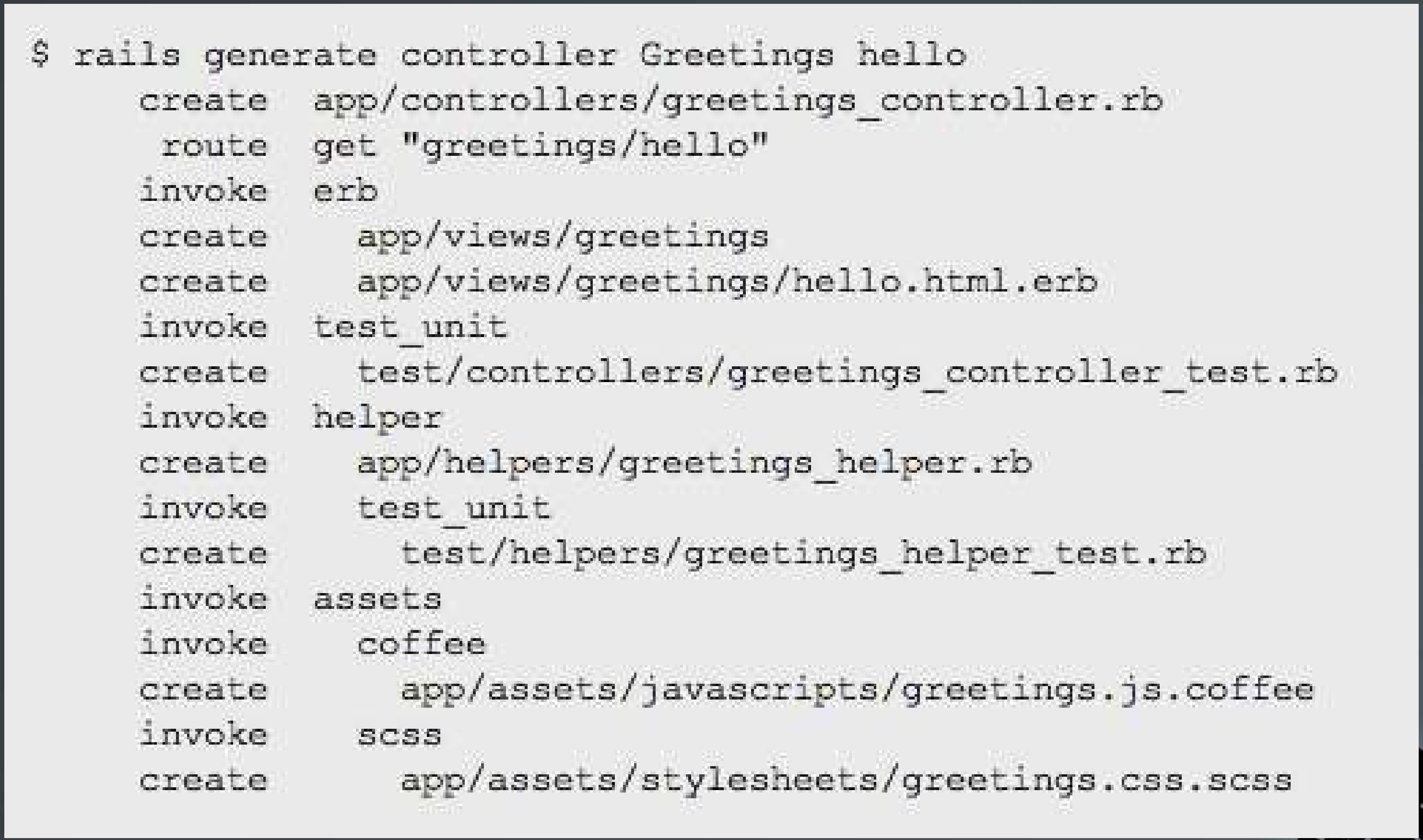
Command Line Basics

rails generate

- controller
- model
- scaffold
- migration

Command Line Basics

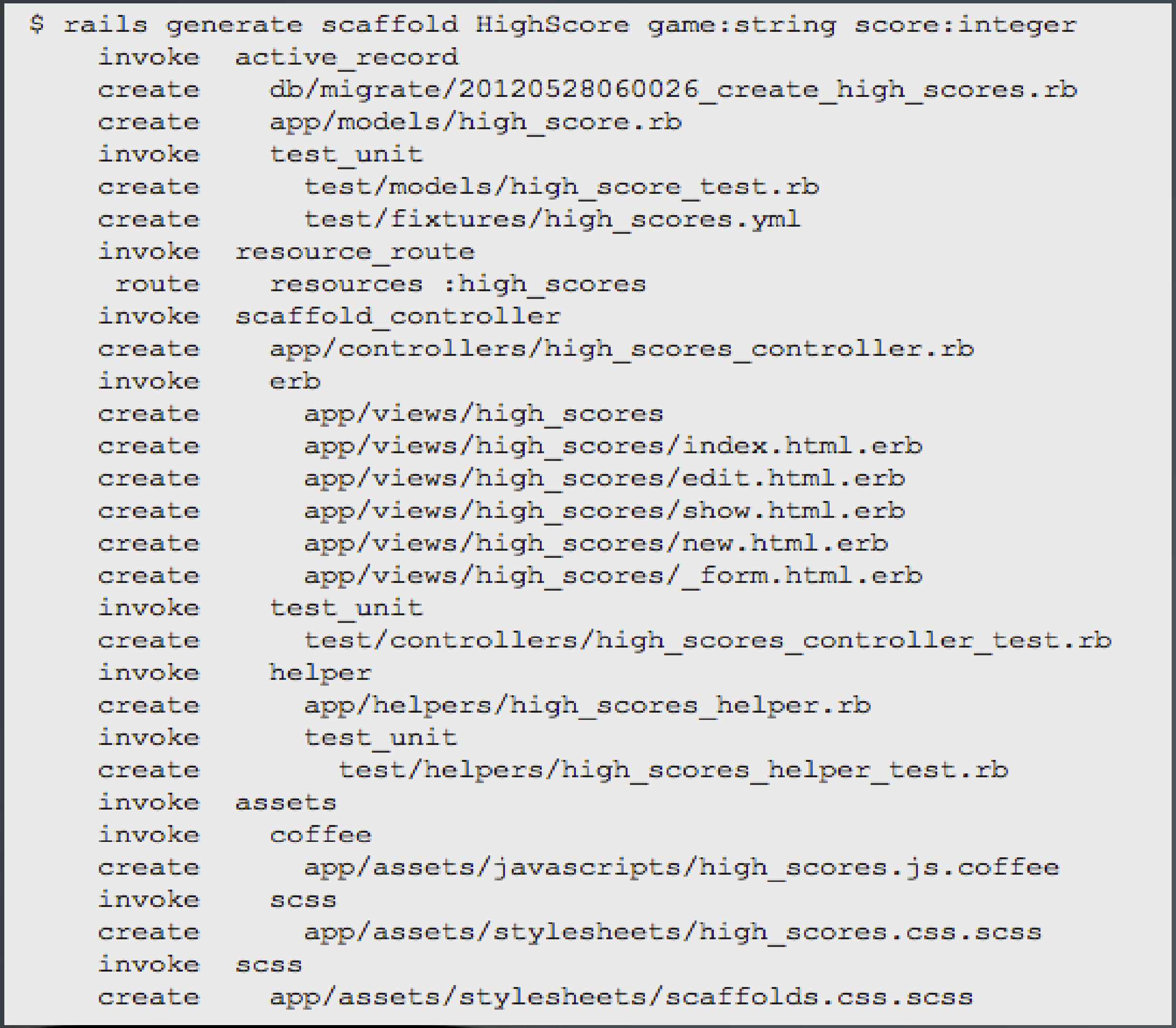
rails generate controller



```
$ rails generate controller Greetings hello
     create   app/controllers/greetings_controller.rb
      route   get "greetings/hello"
     invoke   erb
     create     app/views/greetings
     create     app/views/greetings/hello.html.erb
     invoke   test_unit
     create     test/controllers/greetings_controller_test.rb
     invoke   helper
     create     app/helpers/greetings_helper.rb
     invoke     test_unit
     create       test/helpers/greetings_helper_test.rb
     invoke   assets
     invoke     coffee
     create       app/assets/javascripts/greetings.js.coffee
     invoke     scss
     create       app/assets/stylesheets/greetings.css.scss
```

Command Line Basics

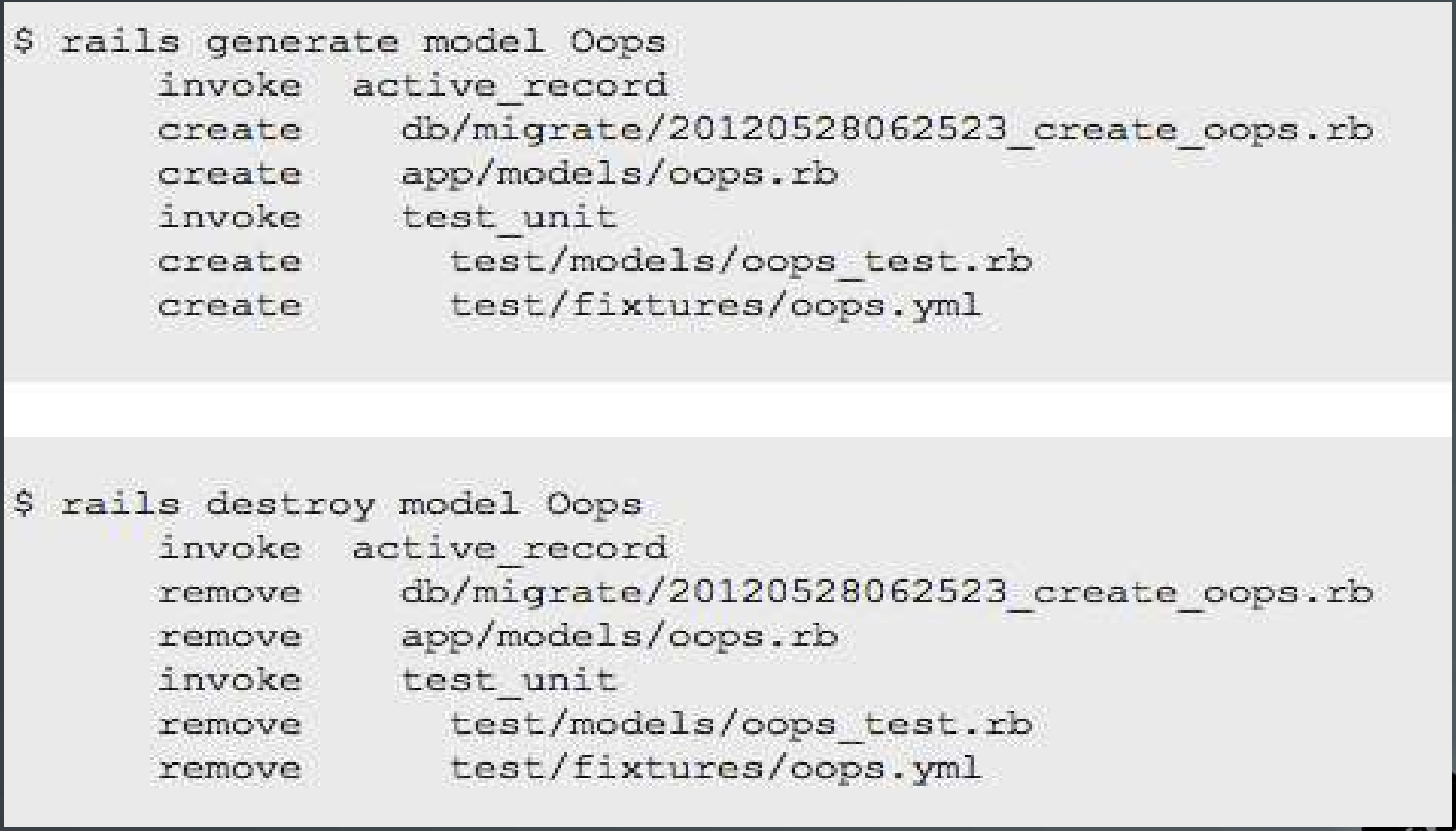
rails generate model[scaffold]



```
$ rails generate scaffold HighScore game:string score:integer
    invoke  active_record
    create    db/migrate/20120528060026_create_high_scores.rb
    create    app/models/high_score.rb
    invoke    test_unit
    create      test/models/high_score_test.rb
    create      test/fixtures/high_scores.yml
    invoke  resource_route
     route    resources :high_scores
    invoke  scaffold_controller
    create    app/controllers/high_scores_controller.rb
    invoke    erb
    create      app/views/high_scores
    create      app/views/high_scores/index.html.erb
    create      app/views/high_scores/edit.html.erb
    create      app/views/high_scores/show.html.erb
    create      app/views/high_scores/new.html.erb
    create      app/views/high_scores/_form.html.erb
    invoke    test_unit
    create      test/controllers/high_scores_controller_test.rb
    invoke    helper
    create      app/helpers/high_scores_helper.rb
    invoke      test_unit
    create        test/helpers/high_scores_helper_test.rb
    invoke  assets
    invoke    coffee
    create      app/assets/javascripts/high_scores.js.coffee
    invoke    scss
    create      app/assets/stylesheets/high_scores.css.scss
    invoke  scss
    create    app/assets/stylesheets/scaffolds.css.scss
```

Command Line Basics

rails destroy   [rails d]

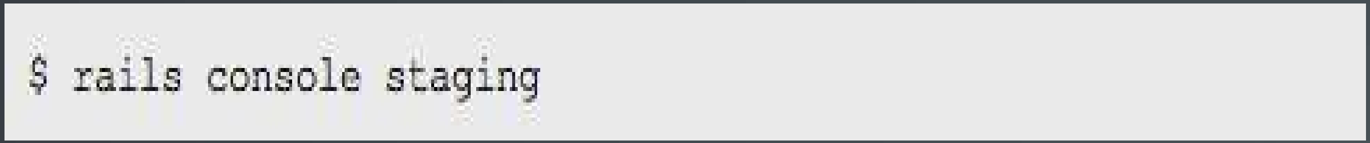


```
$ rails generate model Oops
      invoke  active_record
      create    db/migrate/20120528062523_create_oops.rb
      create    app/models/oops.rb
      invoke    test_unit
      create      test/models/oops_test.rb
      create      test/fixtures/oops.yml
```

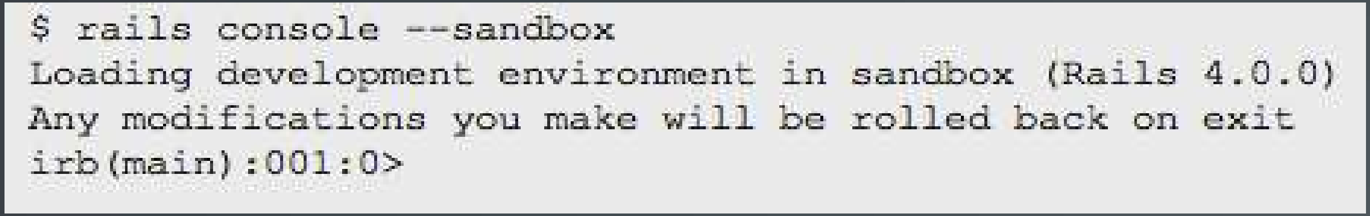
```
$ rails destroy model Oops
      invoke  active_record
      remove    db/migrate/20120528062523_create_oops.rb
      remove    app/models/oops.rb
      invoke    test_unit
      remove      test/models/oops_test.rb
      remove      test/fixtures/oops.yml
```

Command Line Basics

rails console



```
$ rails console staging
```



```
$ rails console --sandbox
Loading development environment in sandbox (Rails 4.0.0)
Any modifications you make will be rolled back on exit
irb(main):001:0>
```

Command Line Basics

rake – ruby make

- rake about
- rake assets
- rake db
- rake doc

Database

Selecting a database

/config/database.yml

- By default has sqlite3

```
rails new myapp --database=postgresql
```

Model

- Default datastructure for data model is called a **model**.
- Default library for interacting with a database is called **Active Record**.
- **Migrations** allow data definitions to be written in Ruby.
- No use of SQL, DDL

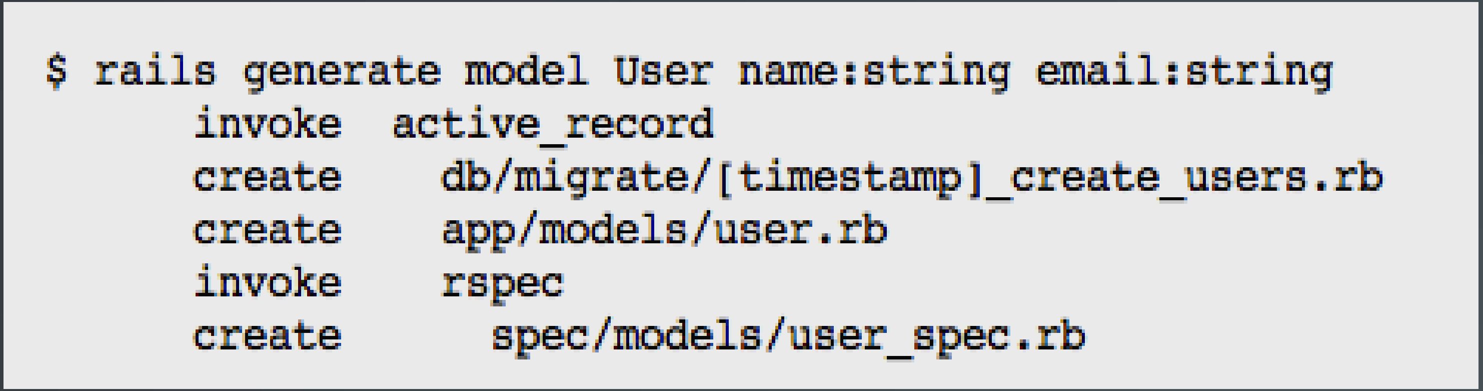
Rake commands

rake db commands

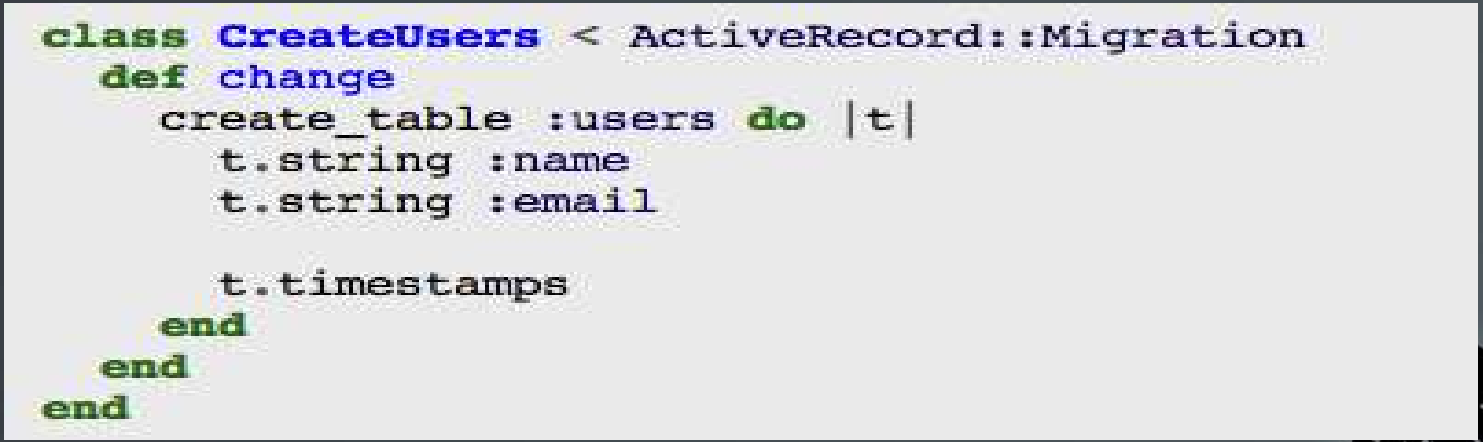
- rake db:create /db:create:all
- rake db:migrate
- rake db:rollback
- rake db:drop / db:drop:all
- rake db:seed

Model

rails generate model



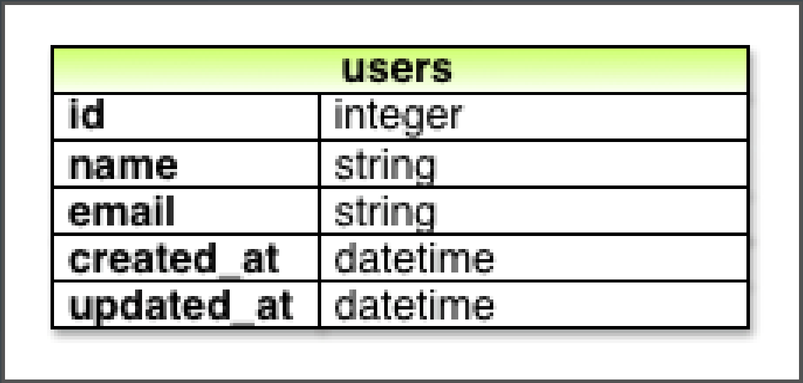
```
$ rails generate model User name:string email:string
      invoke  active_record
      create    db/migrate/[timestamp]_create_users.rb
      create    app/models/user.rb
      invoke    rspec
      create      spec/models/user_spec.rb
```



```
class CreateUsers < ActiveRecord::Migration
  def change
    create_table :users do |t|
      t.string :name
      t.string :email

      t.timestamps
    end
  end
end
```

Model



users	
id	integer
name	string
email	string
created_at	datetime
updated_at	datetime

```
rake db:migrate
```

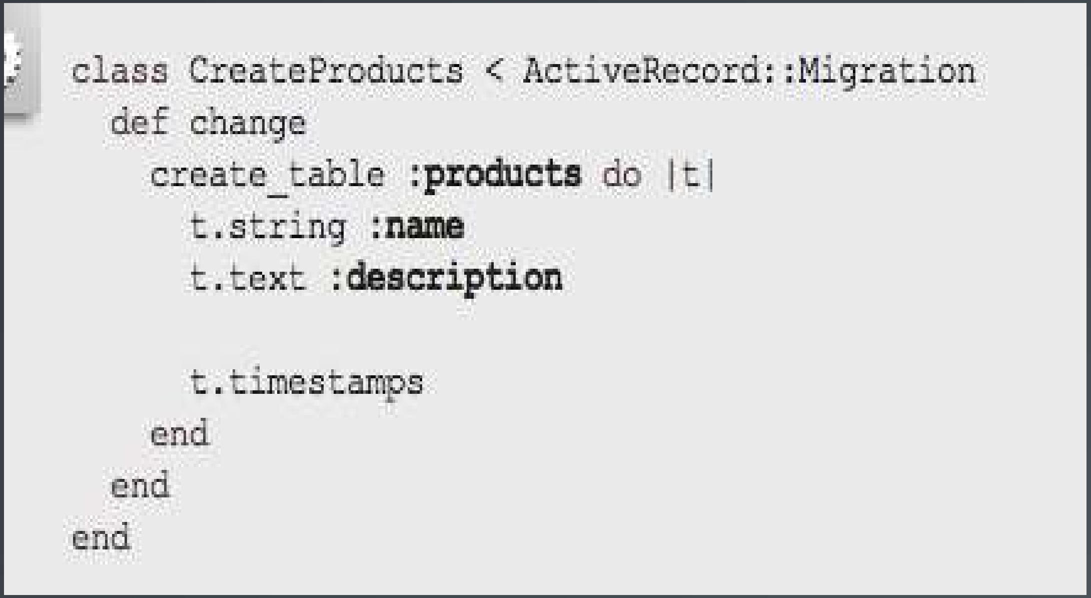
Model

- Controller name is Plural
- Model name is Singular
- Table name is Plural again...

Database manipulations

--- No DMLs

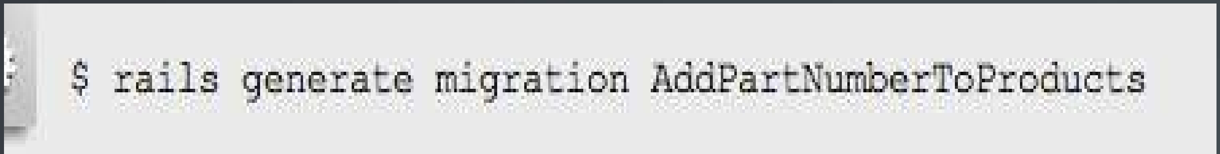
Model



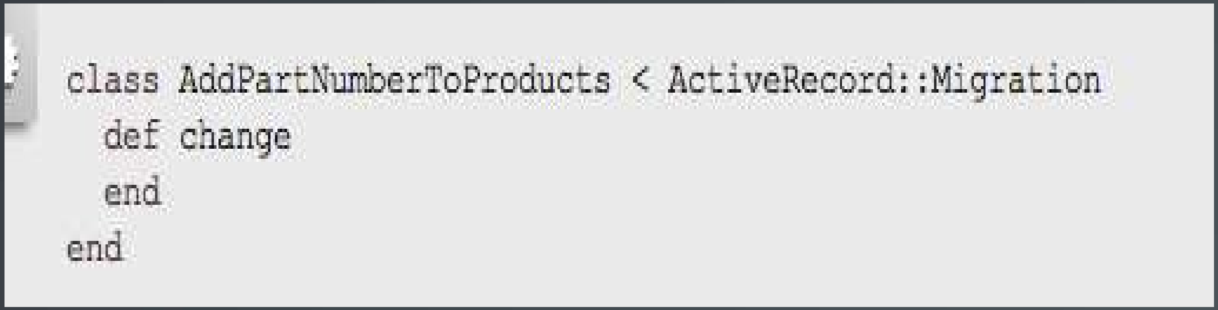
```
class CreateProducts < ActiveRecord::Migration
  def change
    create_table :products do |t|
      t.string :name
      t.text :description

      t.timestamps
    end
  end
end
```

Model - Change

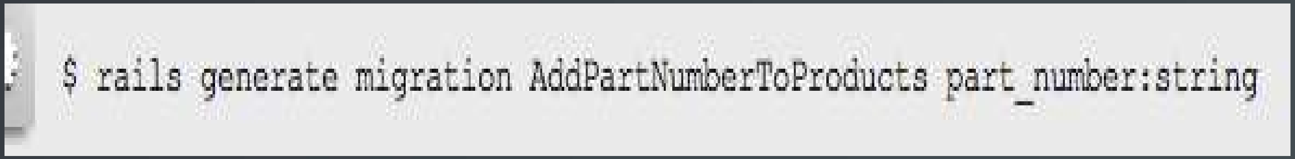


```
$ rails generate migration AddPartNumberToProducts
```

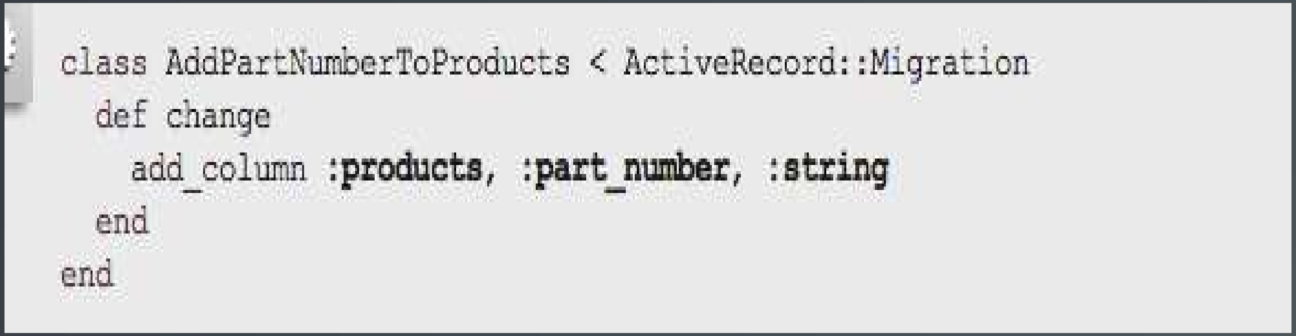


```
class AddPartNumberToProducts < ActiveRecord::Migration
  def change
  end
end
```

Model – Add Column

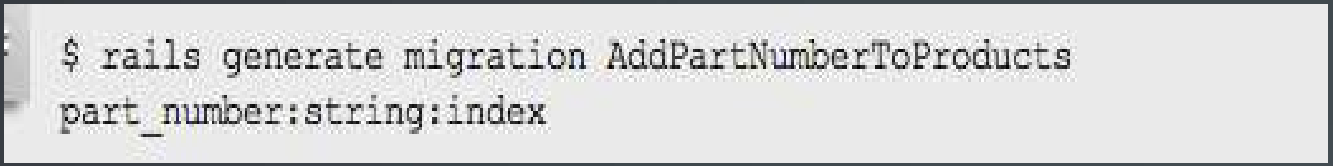


```
$ rails generate migration AddPartNumberToProducts part_number:string
```

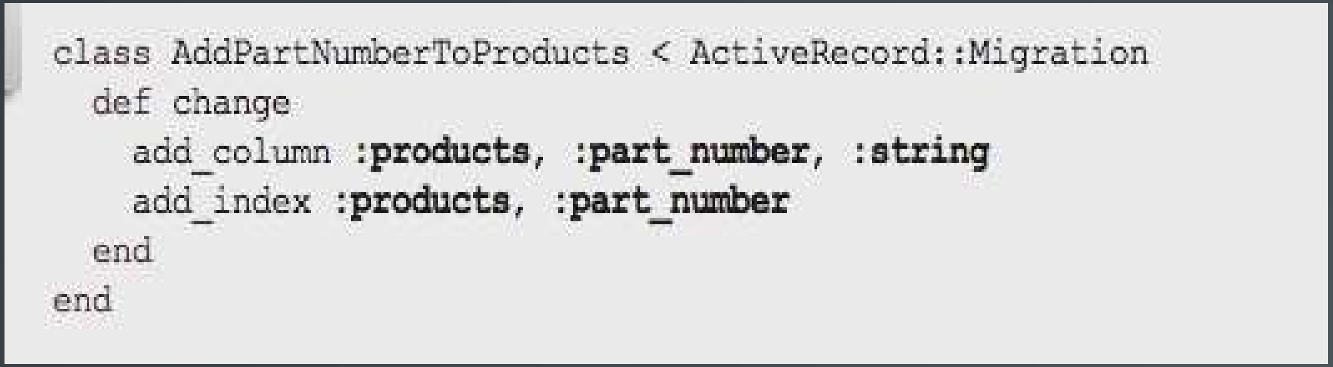


```
class AddPartNumberToProducts < ActiveRecord::Migration
  def change
    add_column :products, :part_number, :string
  end
end
```

Model – Add Column, Index

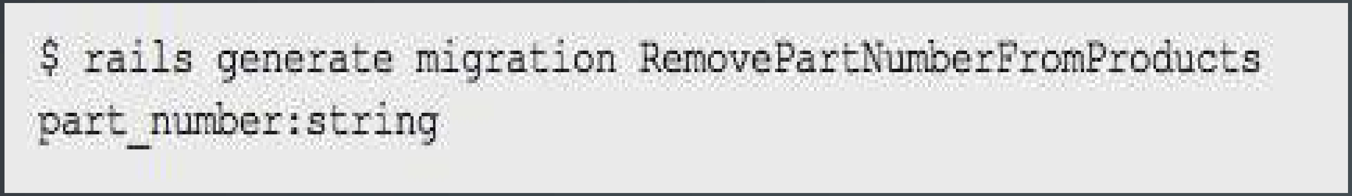


```
$ rails generate migration AddPartNumberToProducts
part_number:string:index
```

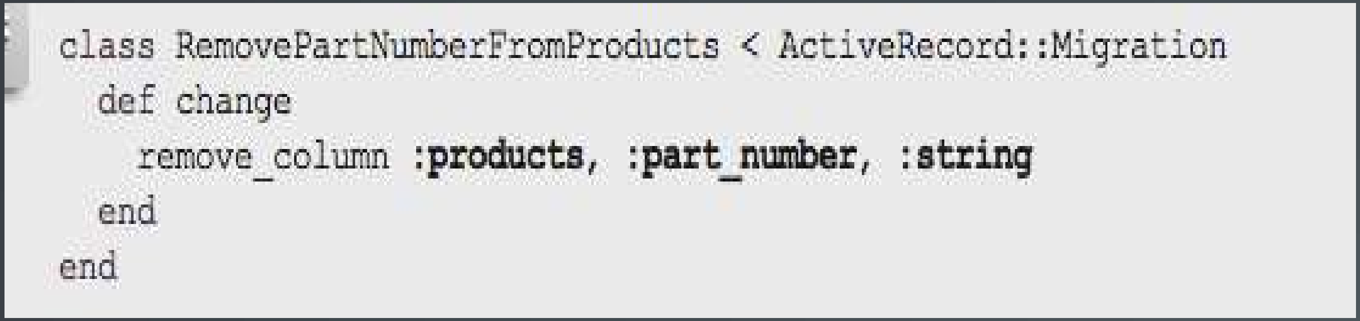


```
class AddPartNumberToProducts < ActiveRecord::Migration
  def change
    add_column :products, :part_number, :string
    add_index :products, :part_number
  end
end
```

Model – Remove Column

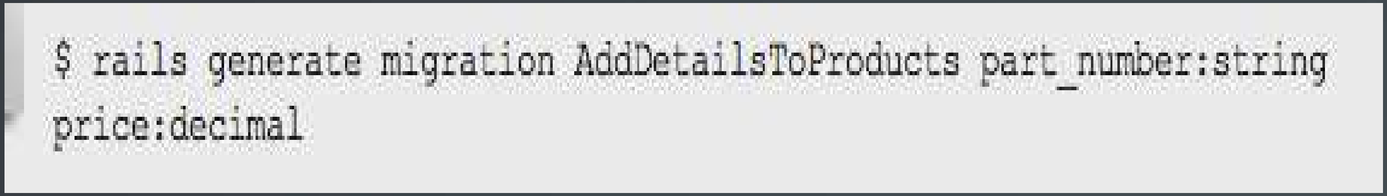


```
$ rails generate migration RemovePartNumberFromProducts
part_number:string
```

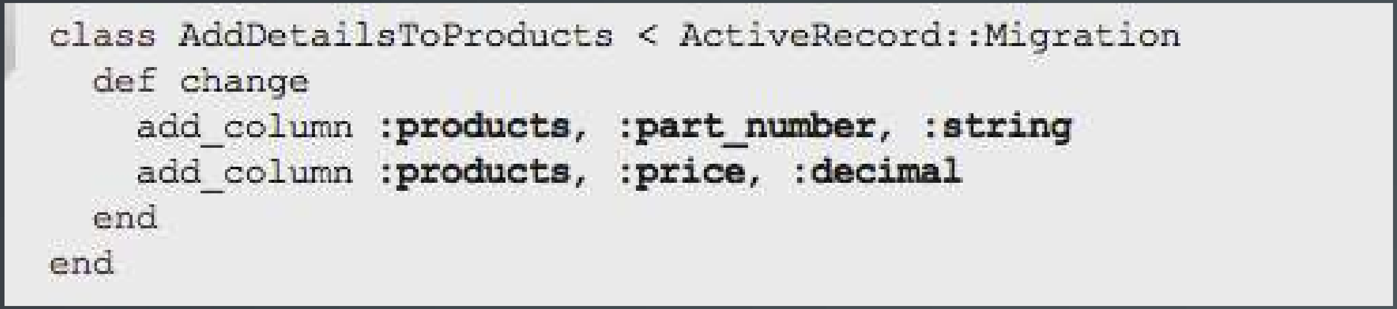


```
class RemovePartNumberFromProducts < ActiveRecord::Migration
  def change
    remove_column :products, :part_number, :string
  end
end
```

Model – Add Columns

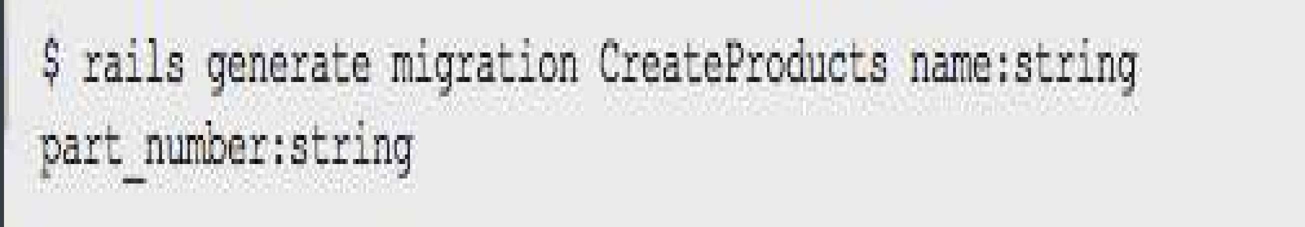


```
$ rails generate migration AddDetailsToProducts part_number:string
price:decimal
```

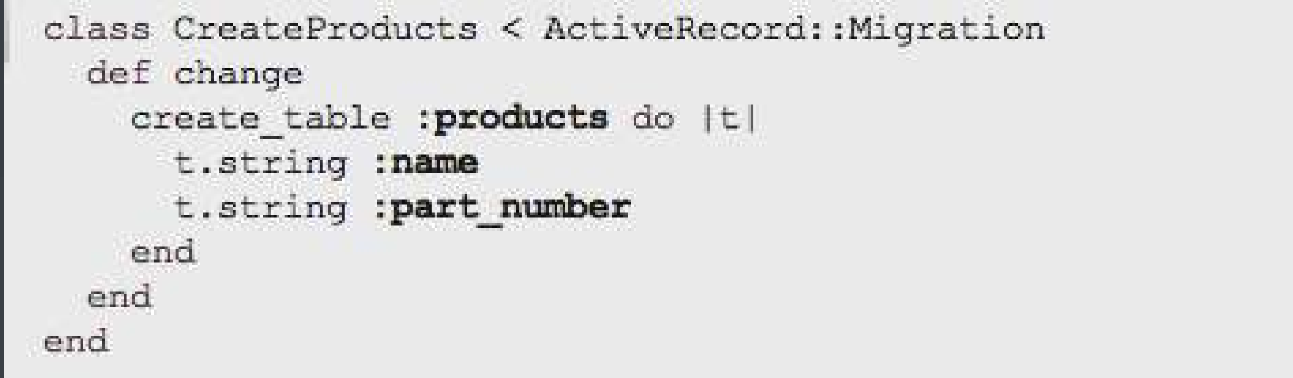


```
class AddDetailsToProducts < ActiveRecord::Migration
  def change
    add_column :products, :part_number, :string
    add_column :products, :price, :decimal
  end
end
```

Model – Create table

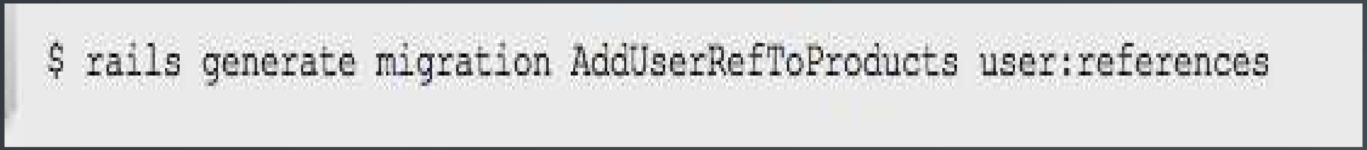


```
$ rails generate migration CreateProducts name:string
part_number:string
```

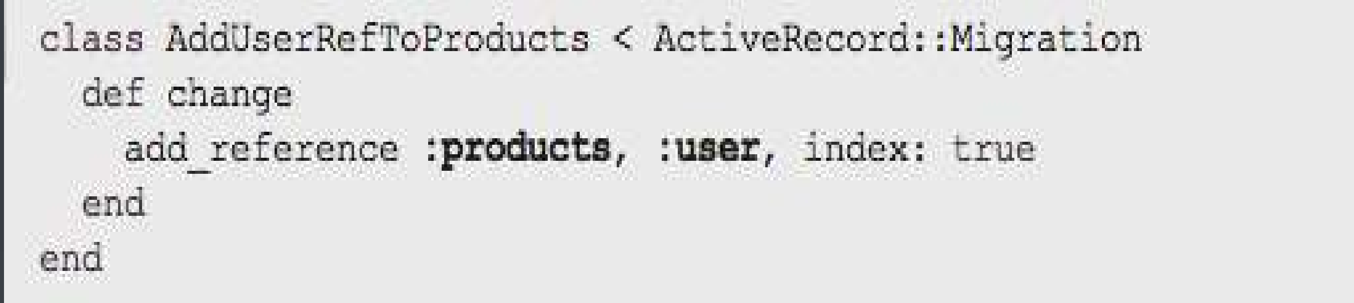


```
class CreateProducts < ActiveRecord::Migration
  def change
    create_table :products do |t|
      t.string :name
      t.string :part_number
    end
  end
end
```

Model – Add belongs_to

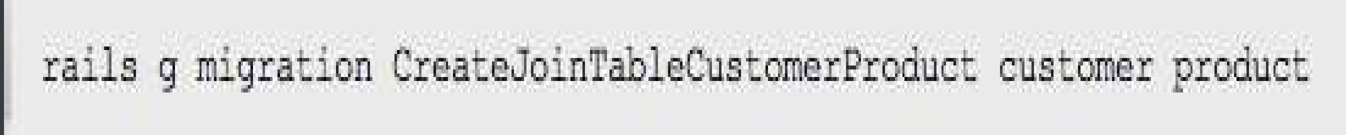


```
$ rails generate migration AddUserRefToProducts user:references
```

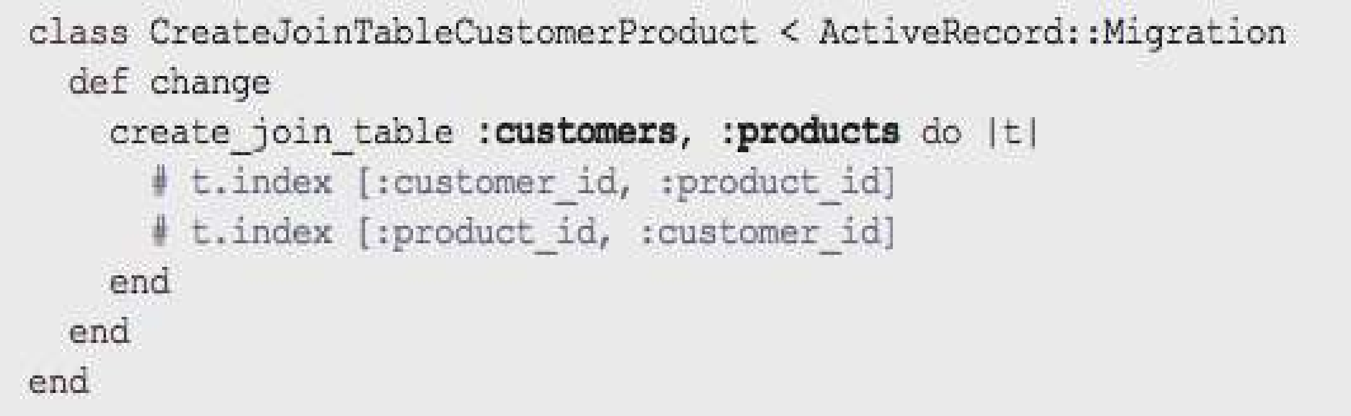


```
class AddUserRefToProducts < ActiveRecord::Migration
  def change
    add_reference :products, :user, index: true
  end
end
```

Model – Create Join tables



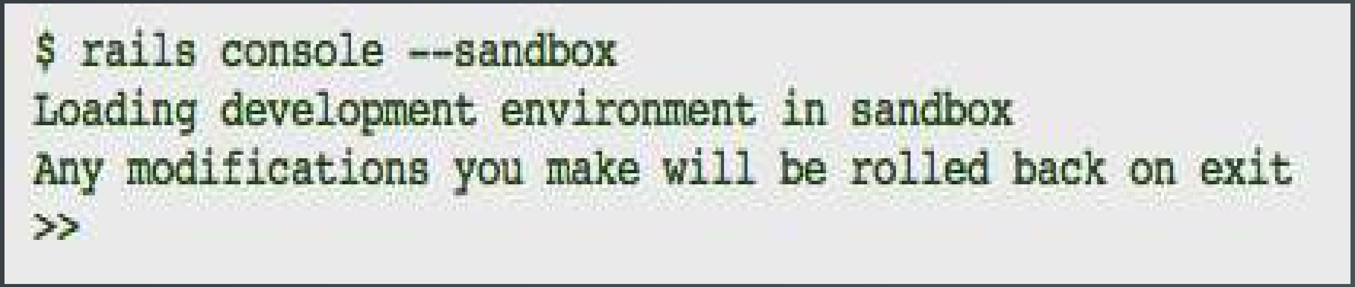
```
rails g migration CreateJoinTableCustomerProduct customer product
```



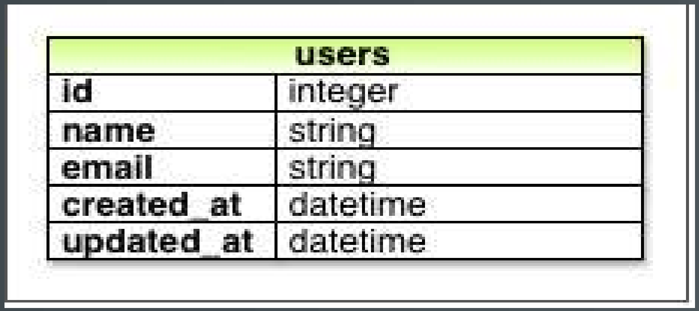
```
class CreateJoinTableCustomerProduct < ActiveRecord::Migration
  def change
    create_join_table :customers, :products do |t|
      # t.index [:customer_id, :product_id]
      # t.index [:product_id, :customer_id]
    end
  end
end
```

Console Data Manipulations

Console tricks

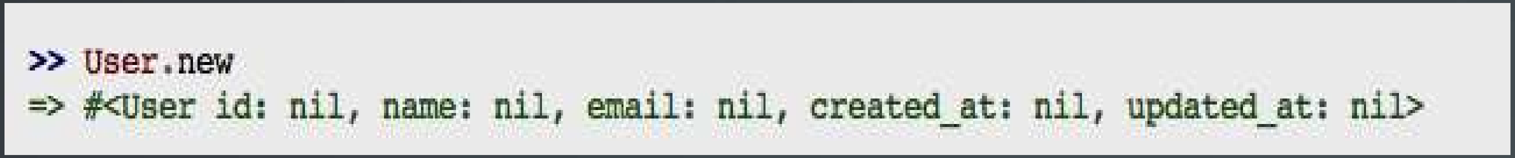


```
$ rails console --sandbox
Loading development environment in sandbox
Any modifications you make will be rolled back on exit
>>
```

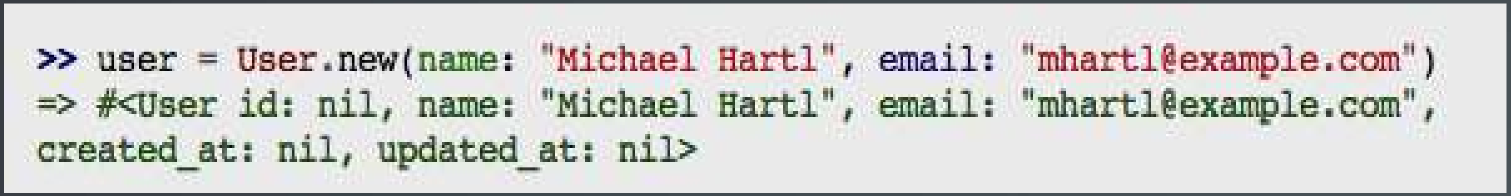


users	
id	integer
name	string
email	string
created_at	datetime
updated_at	datetime

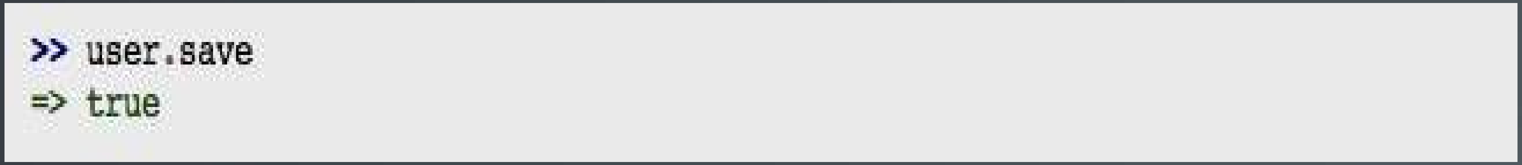
Console tricks



```
>> User.new
=> #<User id: nil, name: nil, email: nil, created_at: nil, updated_at: nil>
```

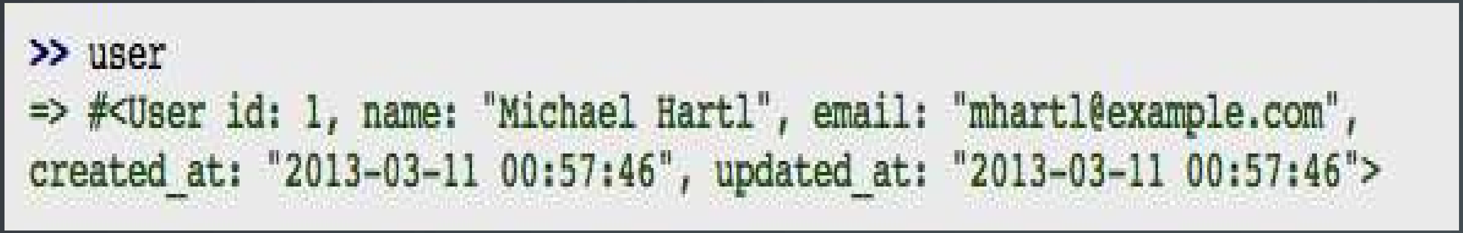


```
>> user = User.new(name: "Michael Hartl", email: "mhartl@example.com")
=> #<User id: nil, name: "Michael Hartl", email: "mhartl@example.com",
created_at: nil, updated_at: nil>
```

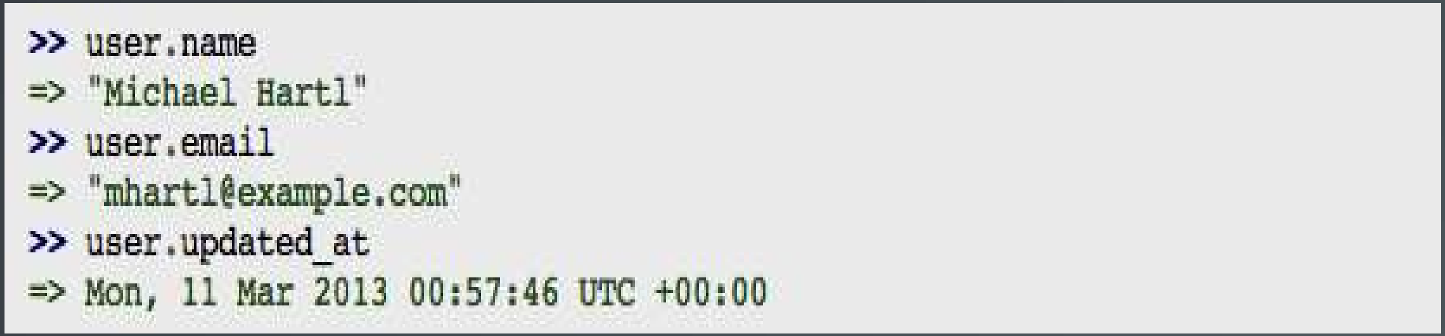


```
>> user.save
=> true
```

Console tricks

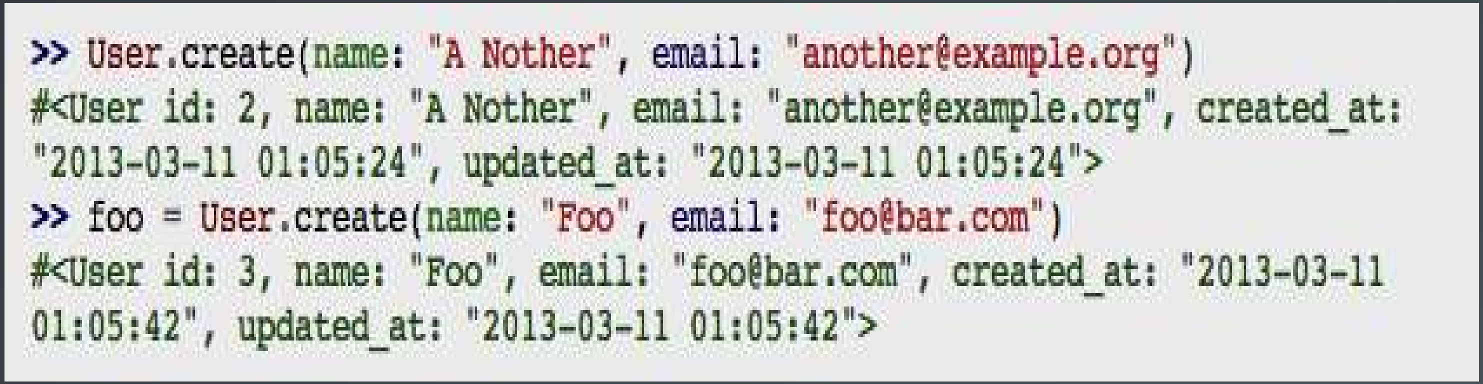


```
>> user
=> #<User id: 1, name: "Michael Hartl", email: "mhartl@example.com",
created_at: "2013-03-11 00:57:46", updated_at: "2013-03-11 00:57:46">
```

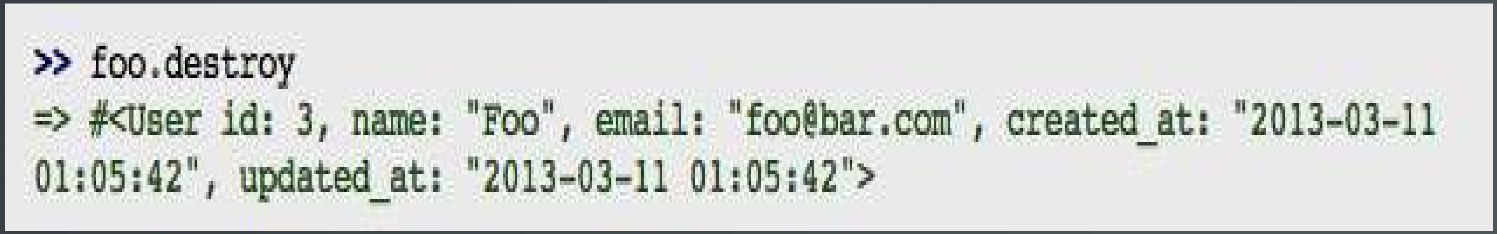


```
>> user.name
=> "Michael Hartl"
>> user.email
=> "mhartl@example.com"
>> user.updated_at
=> Mon, 11 Mar 2013 00:57:46 UTC +00:00
```

Console tricks

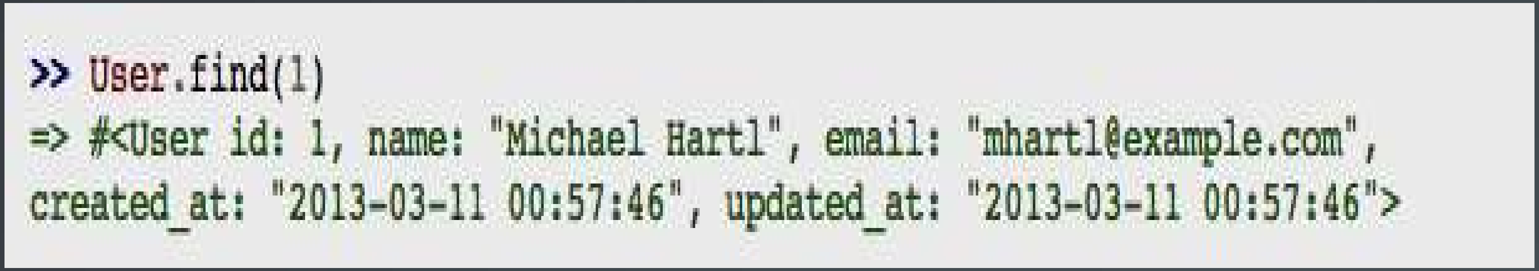


```
>> User.create(name: "A Nother", email: "another@example.org")
#<User id: 2, name: "A Nother", email: "another@example.org", created_at:
"2013-03-11 01:05:24", updated_at: "2013-03-11 01:05:24">
>> foo = User.create(name: "Foo", email: "foo@bar.com")
#<User id: 3, name: "Foo", email: "foo@bar.com", created_at: "2013-03-11
01:05:42", updated_at: "2013-03-11 01:05:42">
```

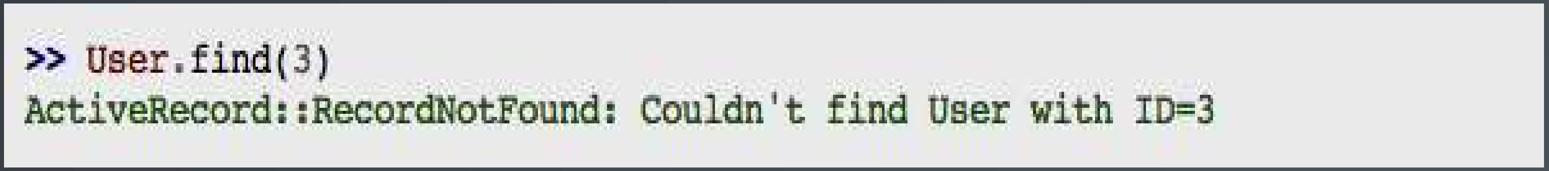


```
>> foo.destroy
=> #<User id: 3, name: "Foo", email: "foo@bar.com", created_at: "2013-03-11
01:05:42", updated_at: "2013-03-11 01:05:42">
```

Console tricks

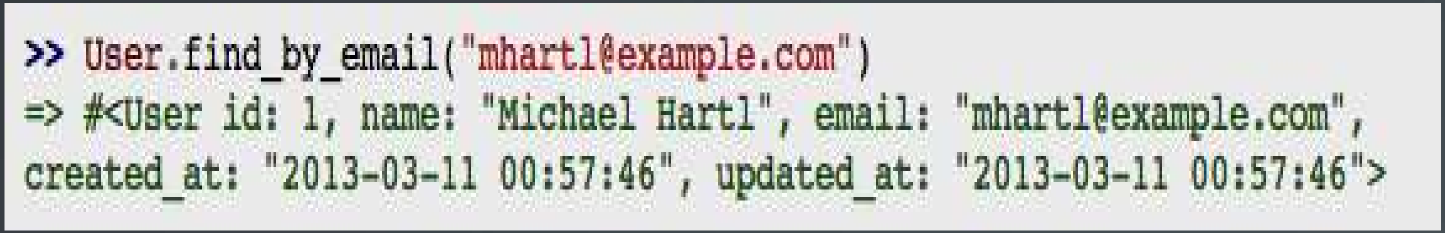


```
>> User.find(1)
=> #<User id: 1, name: "Michael Hartl", email: "mhartl@example.com",
created_at: "2013-03-11 00:57:46", updated_at: "2013-03-11 00:57:46">
```

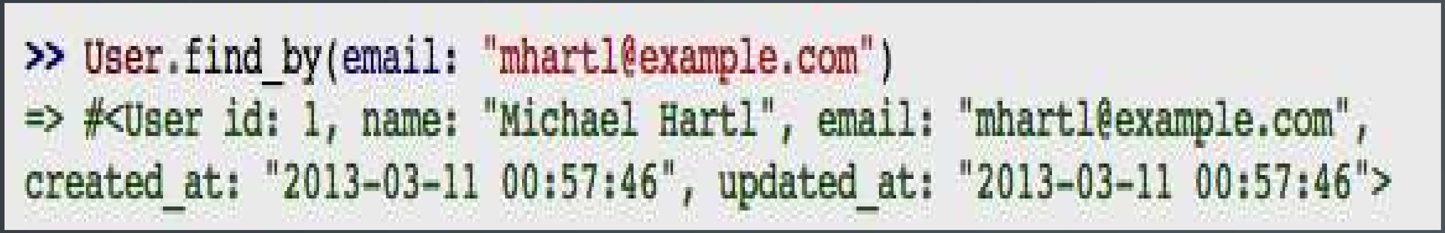


```
>> User.find(3)
ActiveRecord::RecordNotFound: Couldn't find User with ID=3
```

Console tricks

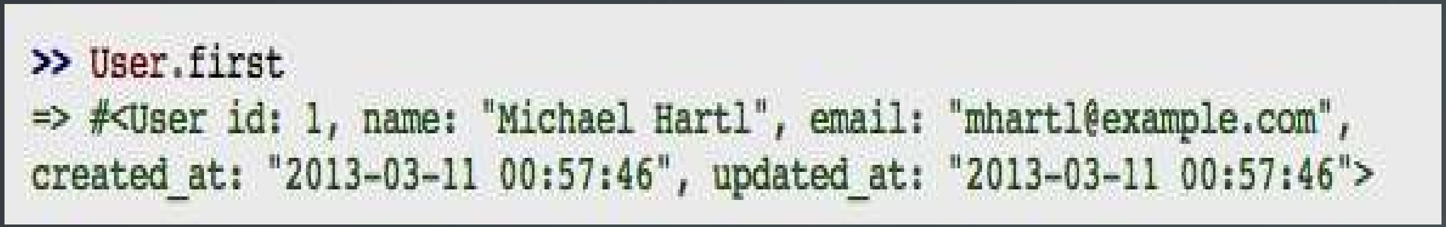


```
>> User.find_by_email("mhartl@example.com")
=> #<User id: 1, name: "Michael Hartl", email: "mhartl@example.com",
created_at: "2013-03-11 00:57:46", updated_at: "2013-03-11 00:57:46">
```

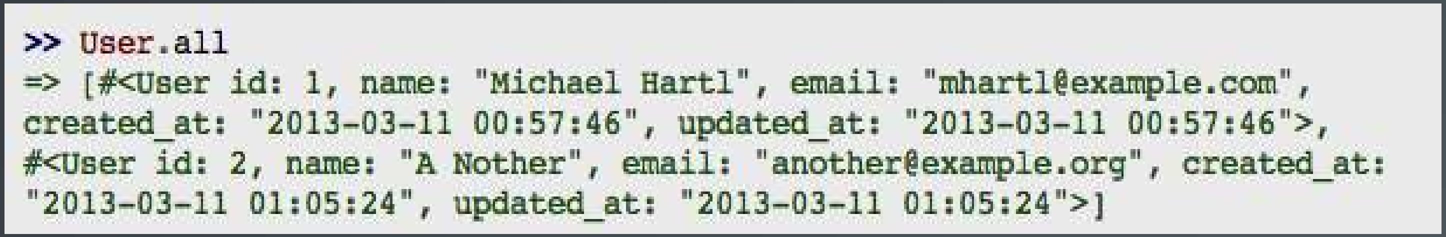


```
>> User.find_by(email: "mhartl@example.com")
=> #<User id: 1, name: "Michael Hartl", email: "mhartl@example.com",
created_at: "2013-03-11 00:57:46", updated_at: "2013-03-11 00:57:46">
```

Console tricks



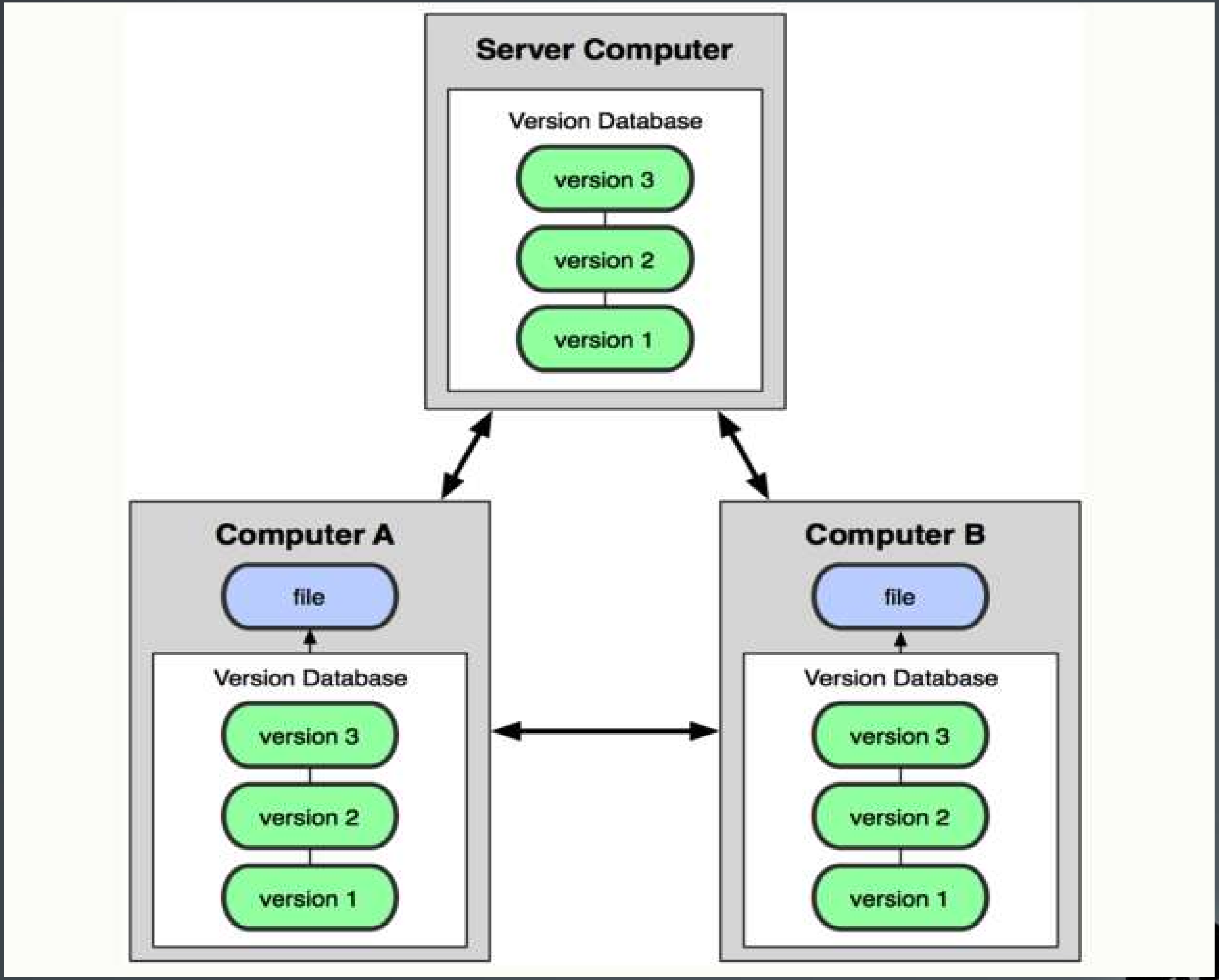
```
>> User.first
=> #<User id: 1, name: "Michael Hartl", email: "mhartl@example.com",
created_at: "2013-03-11 00:57:46", updated_at: "2013-03-11 00:57:46">
```



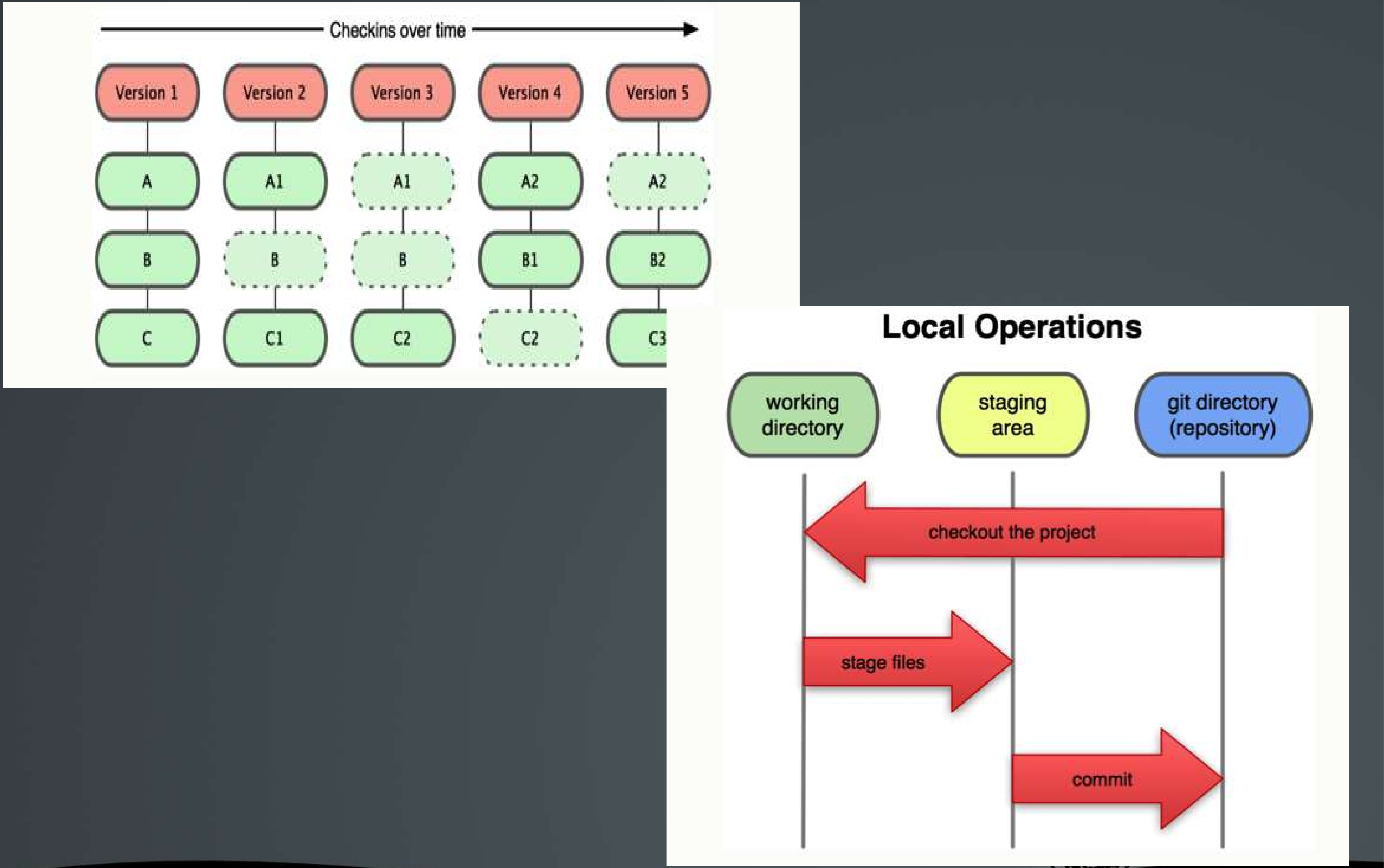
```
>> User.all
=> [#<User id: 1, name: "Michael Hartl", email: "mhartl@example.com",
created_at: "2013-03-11 00:57:46", updated_at: "2013-03-11 00:57:46">,
#<User id: 2, name: "A Nother", email: "another@example.org", created_at:
"2013-03-11 01:05:24", updated_at: "2013-03-11 01:05:24">]
```

Git Version Control

Git Version Control

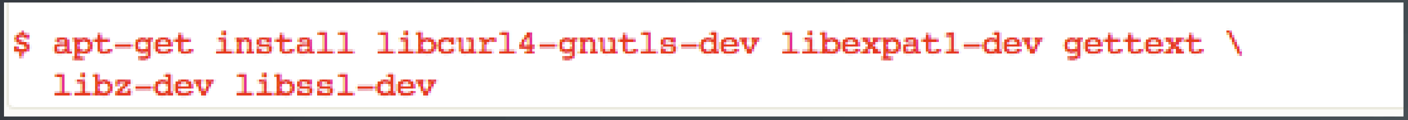


Git Version Control

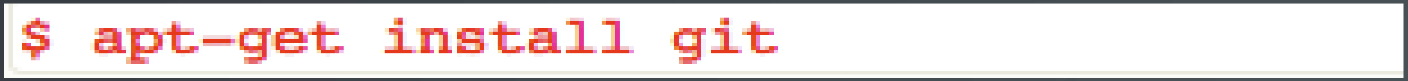


Git Version Control

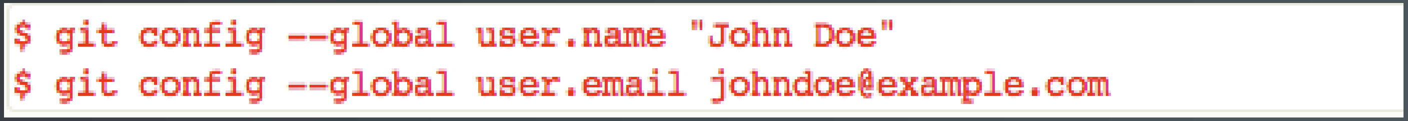
Installation



```
$ apt-get install libcurl4-gnutls-dev libexpat1-dev gettext \
  libz-dev libssl-dev
```



```
$ apt-get install git
```

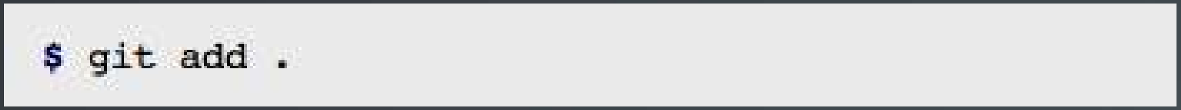


```
$ git config --global user.name "John Doe"
$ git config --global user.email johndoe@example.com
```

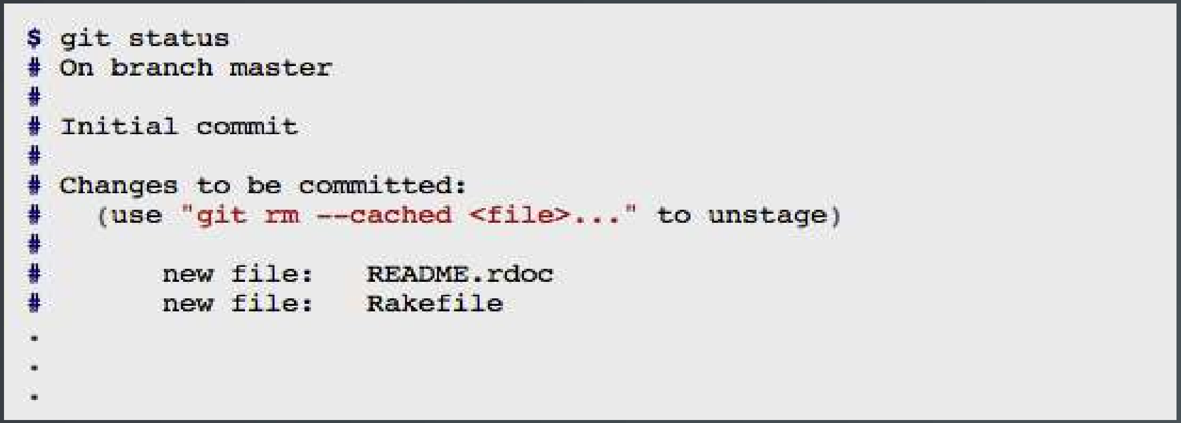
Gitignore

- Not all changes need be tracked.
  - Bundle config files
  - Database journal files
  - Log files
  - Temp files
  - Swap files……

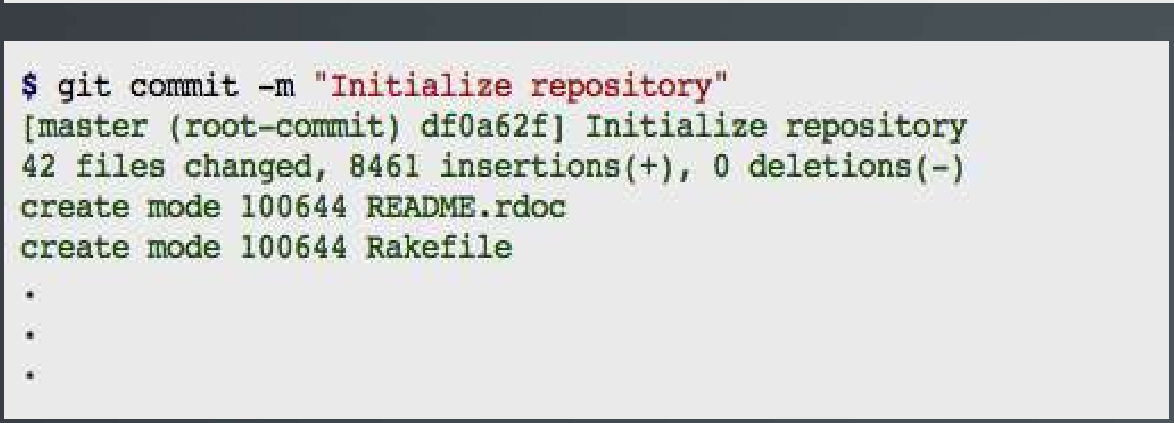
Git Version Control



```
$ git add .
```

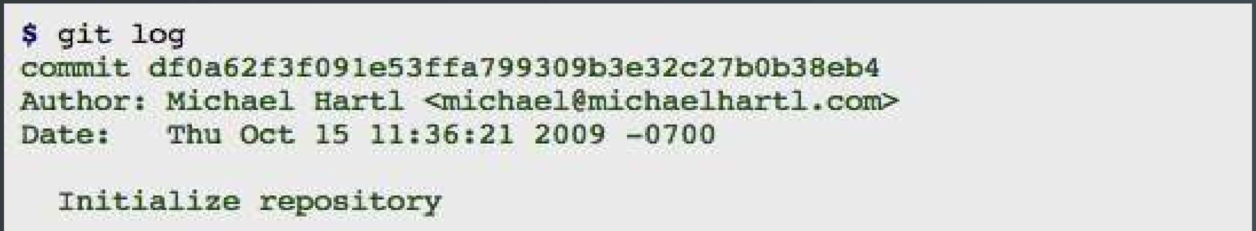


```
$ git status
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file:   README.rdoc
#       new file:   Rakefile
.
.
.
```



```
$ git commit -m "Initialize repository"
[master (root-commit) df0a62f] Initialize repository
42 files changed, 8461 insertions(+), 0 deletions(-)
create mode 100644 README.rdoc
create mode 100644 Rakefile
.
.
.
```

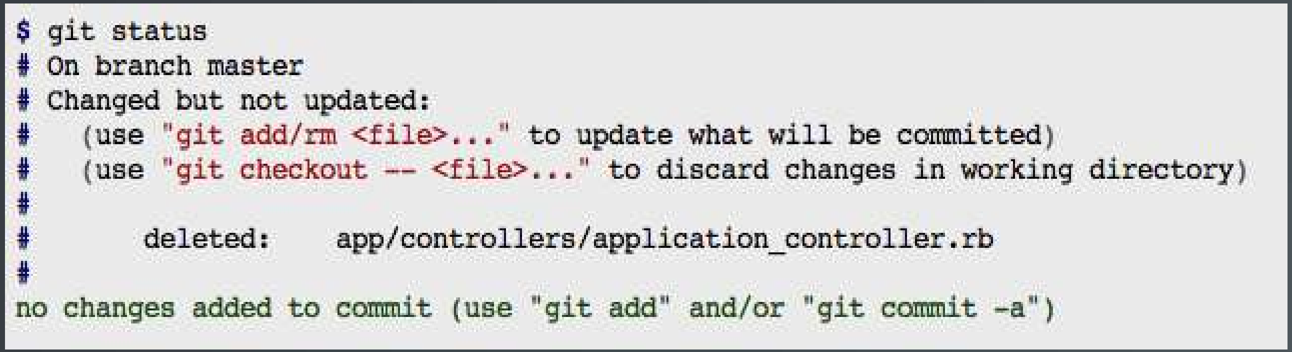
Git Version Control



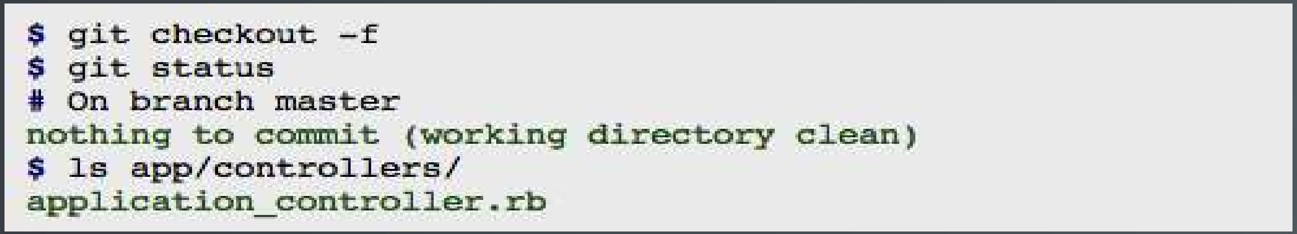
```
$ git log
commit df0a62f3f091e53ffa799309b3e32c27b0b38eb4
Author: Michael Hartl <michael@michaelhartl.com>
Date:   Thu Oct 15 11:36:21 2009 -0700

  Initialize repository
```

For discarding changes – before commiting

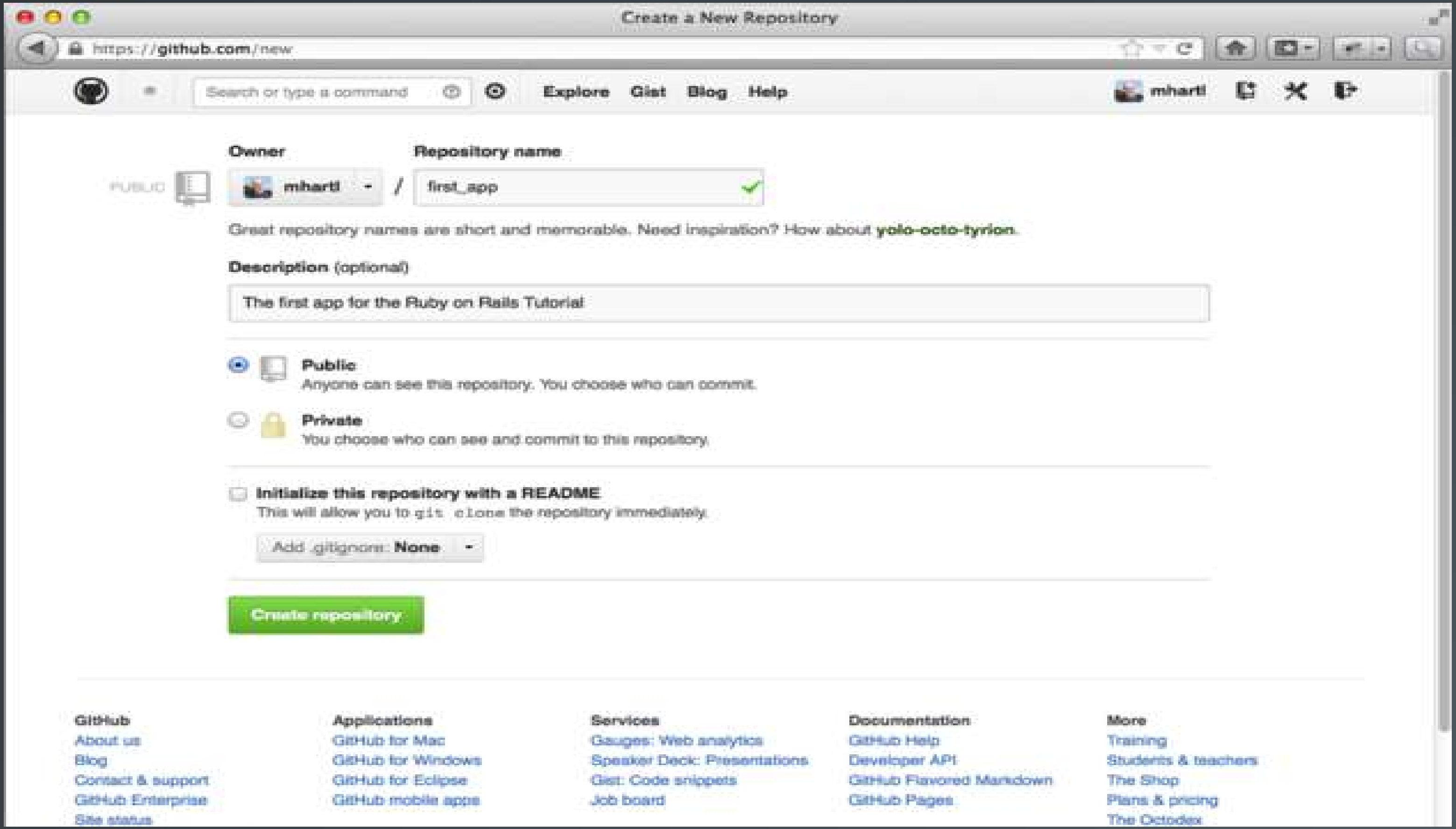


```
$ git status
# On branch master
# Changed but not updated:
#   (use "git add/rm <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       deleted:    app/controllers/application_controller.rb
#
no changes added to commit (use "git add" and/or "git commit -a")
```



```
$ git checkout -f
$ git status
# On branch master
nothing to commit (working directory clean)
$ ls app/controllers/
application_controller.rb
```

Github



Create a New Repository

https://github.com/new

Owner: mhartl / Repository name: first_app

Great repository names are short and memorable. Need inspiration? How about **yolo-octo-tyrion**.

Description (optional)

The first app for the Ruby on Rails Tutorial

● **Public**
Anyone can see this repository. You choose who can commit.

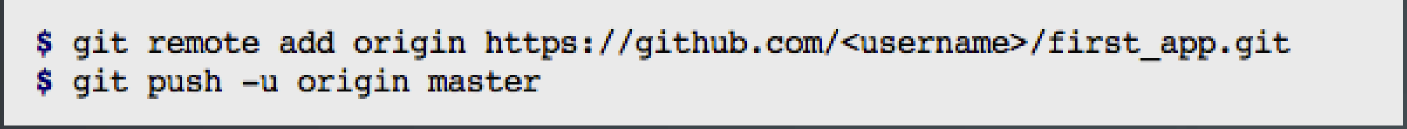
○ **Private**
You choose who can see and commit to this repository.

□ **Initialize this repository with a README**
This will allow you to `git clone` the repository immediately.

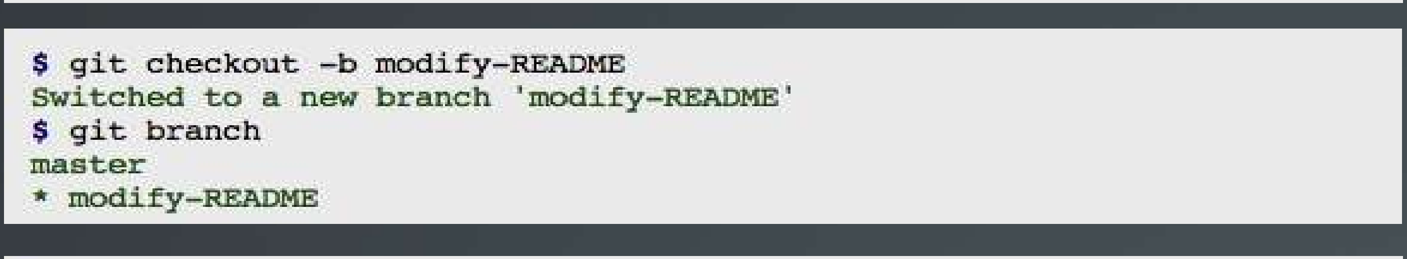
Add .gitignore: **None**

Sign up for a Github account
Setup SSH Keys

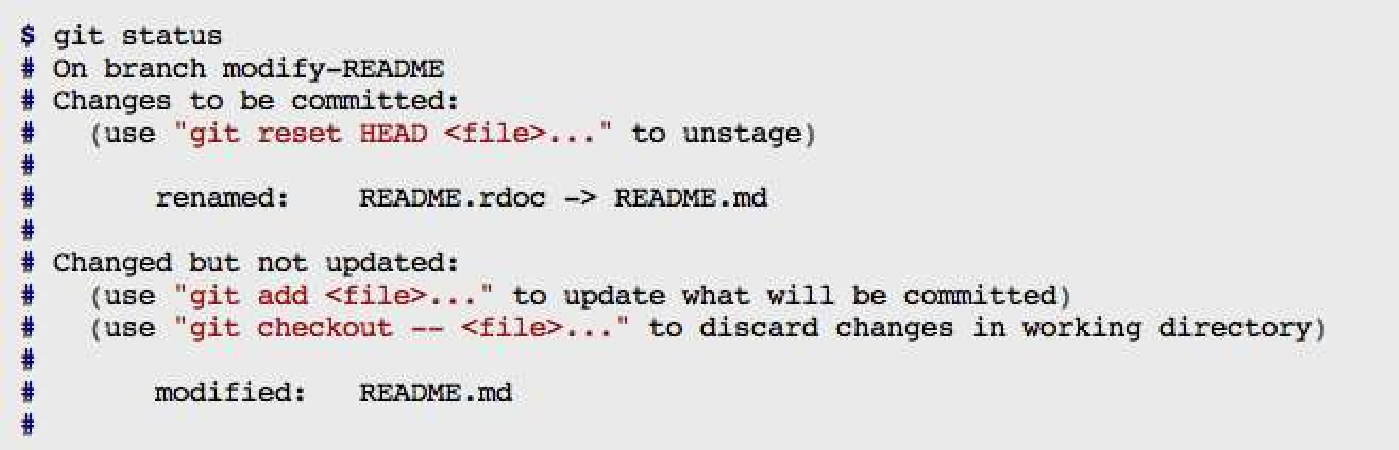
Git Version Control



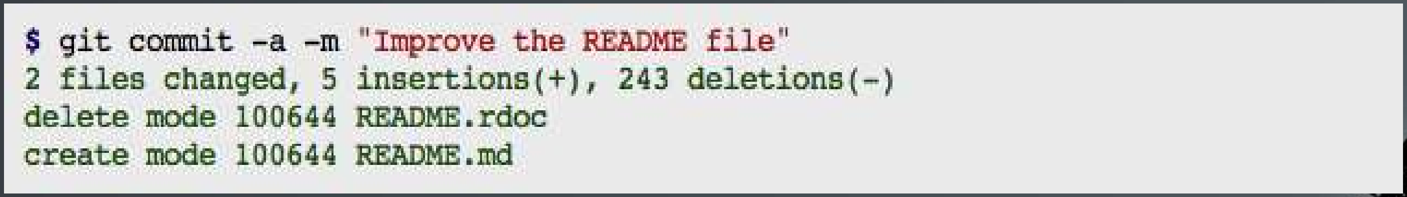
```
$ git remote add origin https://github.com/<username>/first_app.git
$ git push -u origin master
```



```
$ git checkout -b modify-README
Switched to a new branch 'modify-README'
$ git branch
master
* modify-README
```

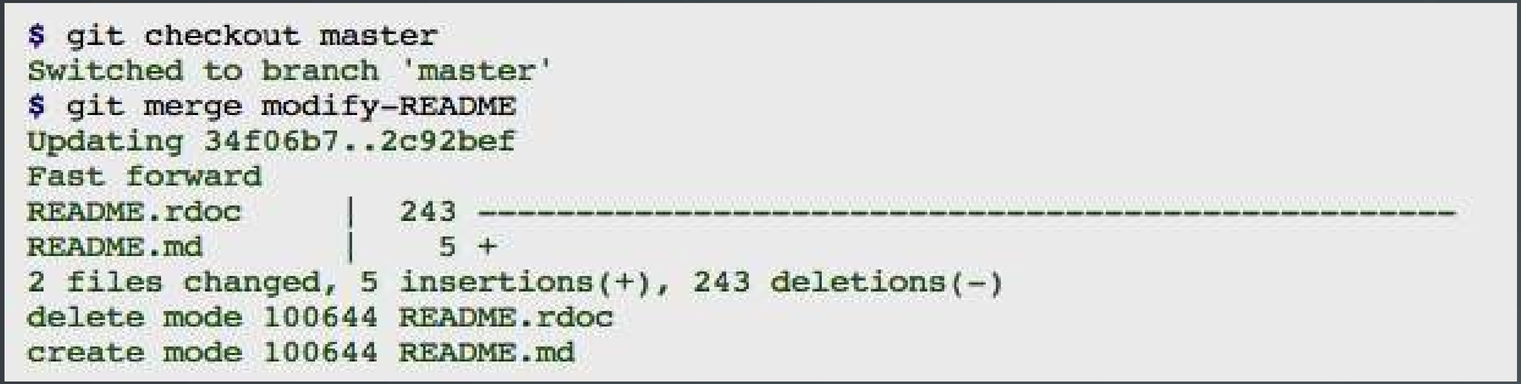


```
$ git status
# On branch modify-README
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       renamed:    README.rdoc -> README.md
#
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   README.md
#
```

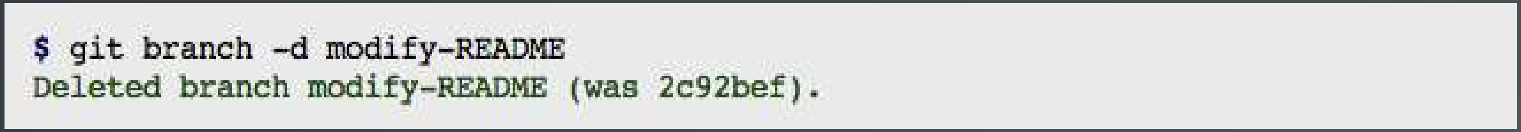


```
$ git commit -a -m "Improve the README file"
2 files changed, 5 insertions(+), 243 deletions(-)
delete mode 100644 README.rdoc
create mode 100644 README.md
```

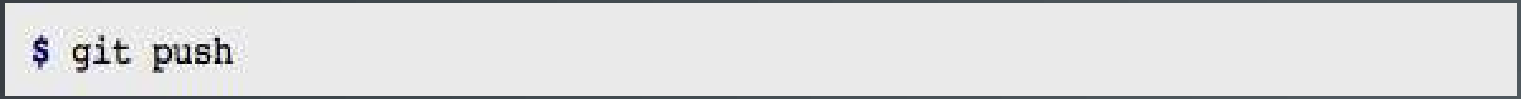
Git Version Control



```
$ git checkout master
Switched to branch 'master'
$ git merge modify-README
Updating 34f06b7..2c92bef
Fast forward
README.rdoc     |  243 --------------------------------------------------
README.md       |    5 +
2 files changed, 5 insertions(+), 243 deletions(-)
delete mode 100644 README.rdoc
create mode 100644 README.md
```



```
$ git branch -d modify-README
Deleted branch modify-README (was 2c92bef).
```

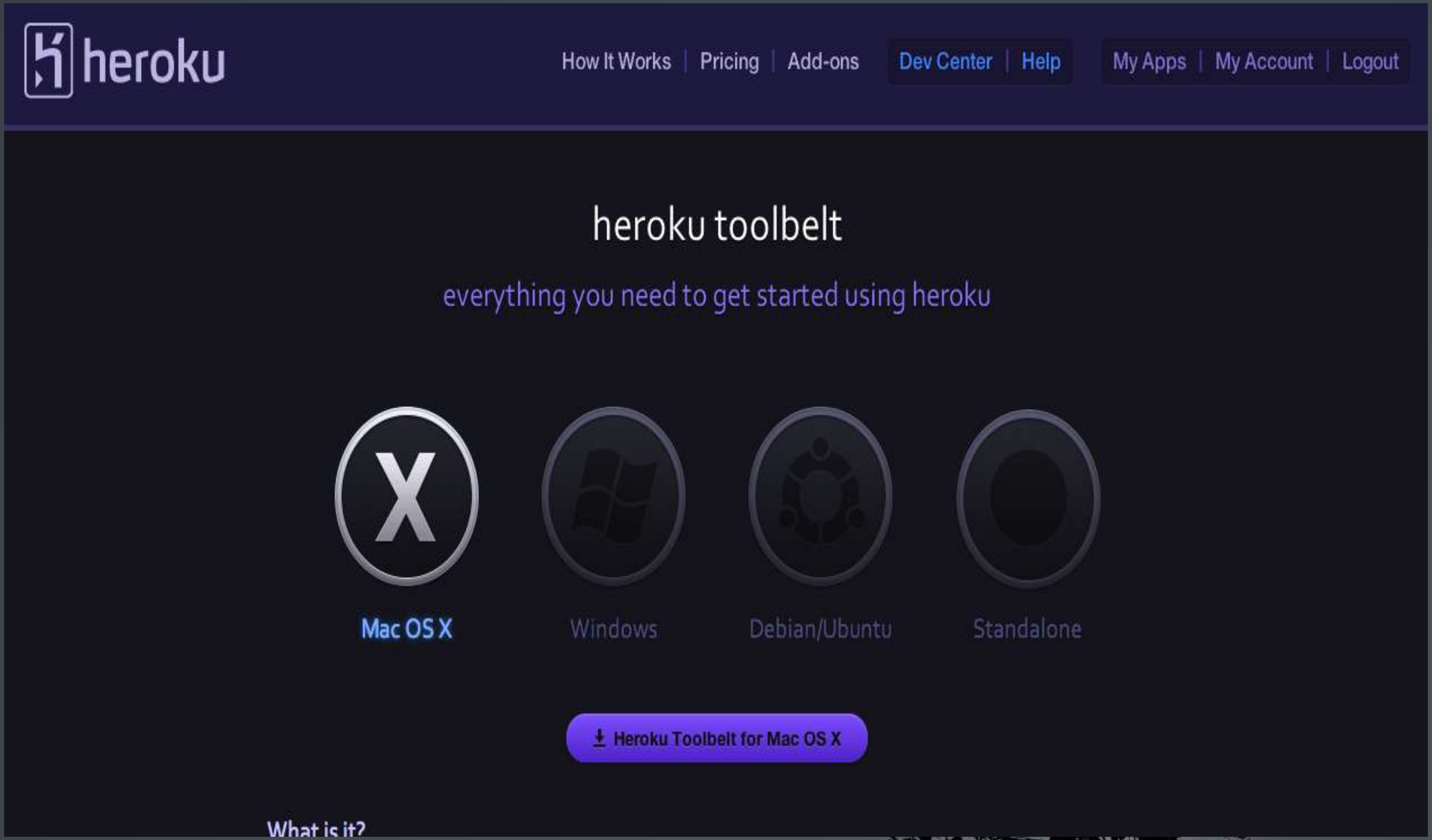


```
$ git push
```

Deploy on Heroku

Heroku

Sign Up for a Heroku account
Install Heroku Toolbelt



heroku
How It Works | Pricing | Add-ons
Dev Center | Help
My Apps | My Account | Logout

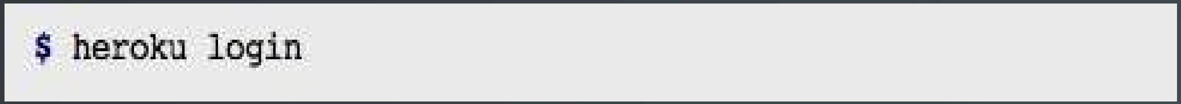
heroku toolbelt

everything you need to get started using heroku

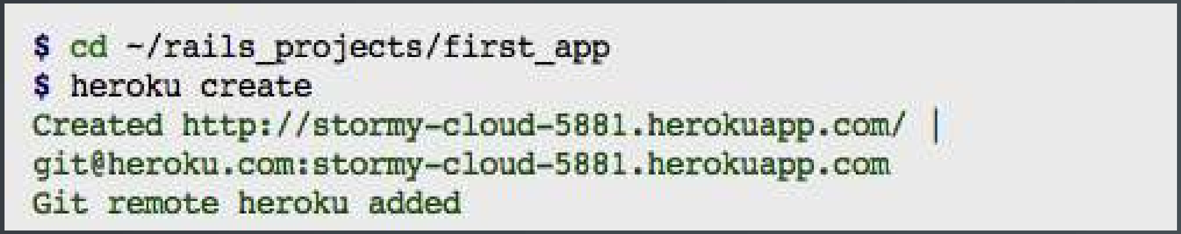
Mac OS X
Windows
Debian/Ubuntu
Standalone
Heroku Toolbelt for Mac OS X

What is it?

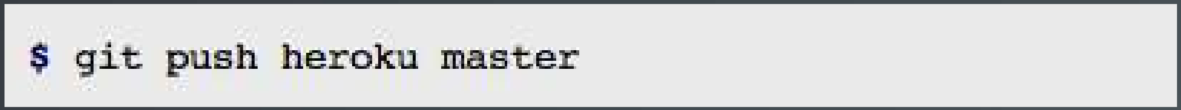
Heroku Deployment



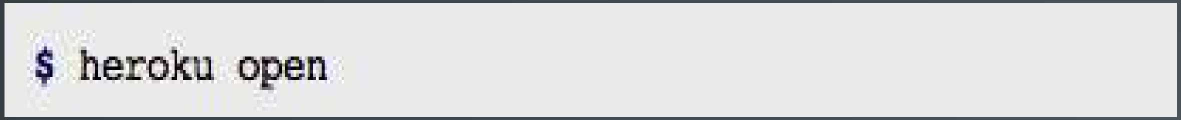
```
$ heroku login
```



```
$ cd ~/rails_projects/first_app
$ heroku create
Created http://stormy-cloud-5881.herokuapp.com/ |
git@heroku.com:stormy-cloud-5881.herokuapp.com
Git remote heroku added
```



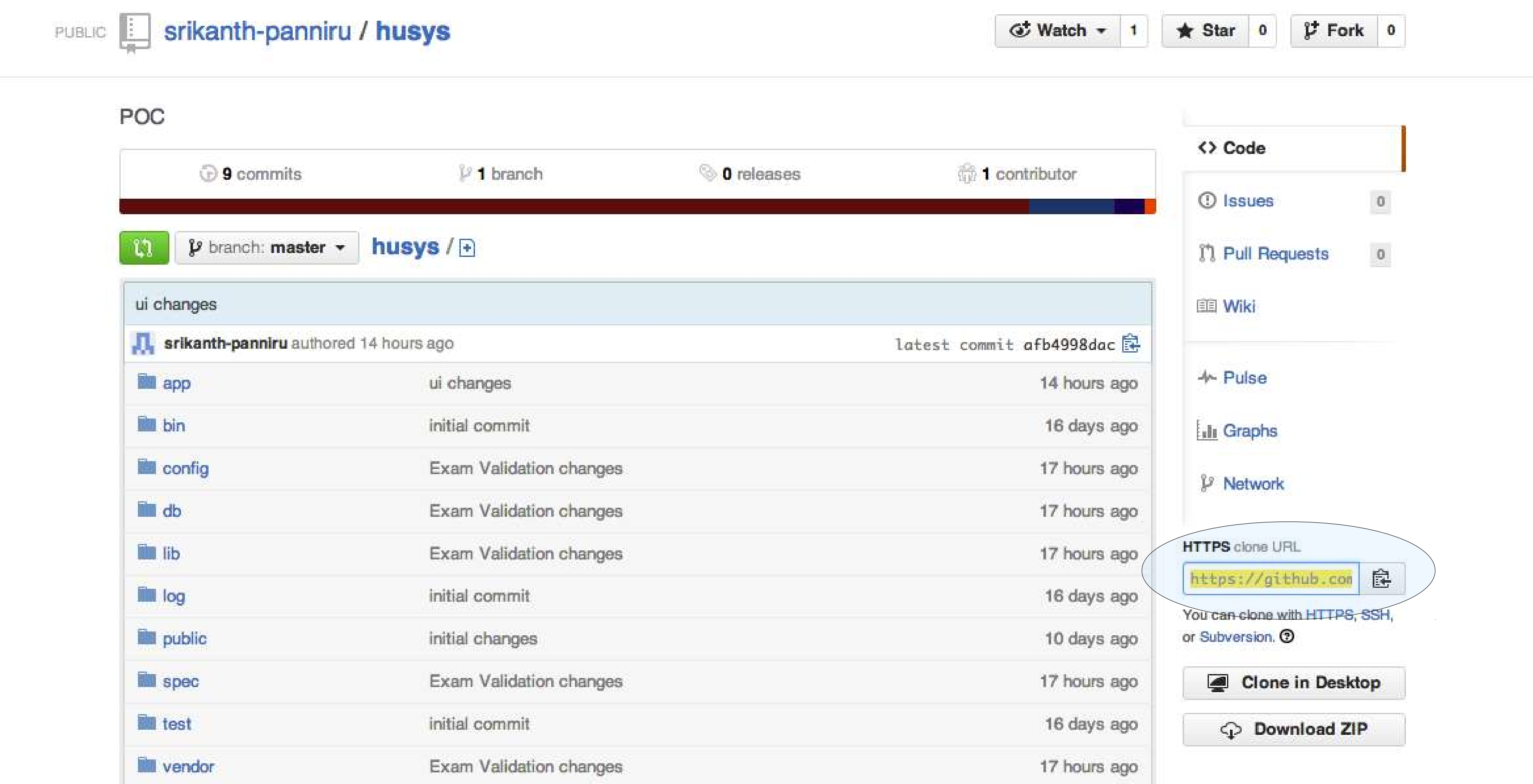
```
$ git push heroku master
```



```
$ heroku open
```

Heroku – Other's application

git clone <path>



POC

ui changes

srikanth-panniru authored 14 hours ago — latest commit afb4998dac

app	ui changes	14 hours ago
bin	initial commit	16 days ago
config	Exam Validation changes	17 hours ago
db	Exam Validation changes	17 hours ago
lib	Exam Validation changes	17 hours ago
log	initial commit	16 days ago
public	initial changes	10 days ago
spec	Exam Validation changes	17 hours ago
test	initial commit	16 days ago
vendor	Exam Validation changes	17 hours ago

Heroku – Other's application

Only fetch and test

```
git pull
```

Fetch and modify

```
git fetch
git merge origin/master
```

Thank You!