Rails Routes and Actions

Rendering a Response

1. Define a **route** that maps a url path to a specific **controller** and **action method.**

2. Define a **controller class.**

3. Inside the controller class, define an **action method.**

4. Call the **render** method to respond with a **status** code along with a **text** or **inline** body; or other headers.

```ruby
class MyController < ApplicationController

  def hello
    render :text => "Hello!", :status => 200
  end

  def goodbye
    render :status => 302,
           :location => "http://apple.com"
  end

end
```

MVC Basics

4-Step Web Page Recipe

1. Define a **route** that maps a url path to a specific **controller** and **action.**

2. Define a **controller class.**

3. Inside the controller class, define an **action method.** This method is the best place to create **instance variables** that contain the data you want on your page. (You can use your **models** to retrieve the data you want.)

4. Write a **view template** that generates the HTML for your web page. You can use any instance variables you defined in Step 3. .

How To Use Route Placeholders

```
# config/routes.rb
MyApp::Application.routes.draw do

  get “products/:id”, :controller => “products”,
                      :action => “show”

end
```

Placeholder

```
# app/controllers/products_controller.rb
class ProductsController < ApplicationController

  def show
    product_id = params[:id]
    @the_product = Product.find(product_id)
  end

end
```

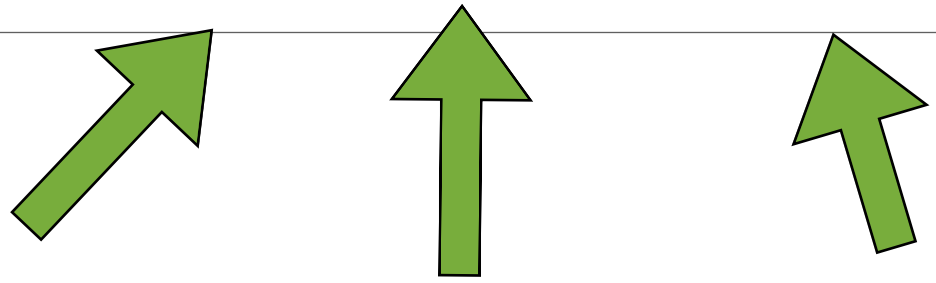
Helpful Command-Line Tasks

```
rake
rake about
rake db:create
rake db:drop
rake db:migrate
rake db:rollback
rake db:seed
rake db:version
rake notes
rake routes
rake stats
```

```
rails new
rails server
rails console
rails dbconsole
rails runner
```

ActiveRecord
Generating a New Model

```
rails generate model product name:string color:string
```



Generator

Model name

Model Attributes

products		
id	integer	autoincrement
name	string	
color	string	

ActiveRecord Basics

CRUD Methods

```
new                            all
save                           first
create                         last
                               count
update_attribute(...)
update_attributes(...)         valid?

destroy                        find(id)
```

Refining Your Query

`where`	SQL fragment (string or array)
`order`	SQL order clause
`limit`	Limit the number of rows returned
`joins`	SQL fragment (string) or named associations
`include`	Joins and eager-loads from named associations

Examples

```
Product.all

Product.find_by_color ‘blue’
Product.find_all_by_color ‘blue’
Product.where(:color => ‘blue’)

Product.where(“price > ?”, price)

Product.order(‘price asc’)
       .where(“color = ‘blue’”)
       .limit(10)
```