

# Ruby on Rails in Action

2015-08-08

李磊

# 大纲

- Setup a new project
- Model
- Controller
- View
- Routes
- 其他
- 幕后

# Setup a new project

```
# 创建一个新的rails项目
$ rails new hellorails -d mysql

# 创建数据库
$ rake db:create

# 创建model
$ rails g model user

# 创建数据库表(migration)
$ rake db:migration
























# 创建controller
$ rails g controller users
















# 编辑routes.rb和创建view

# 启动web server
$ rails s -p 3000

# 打开浏览器 http://localhost:3000/users
```

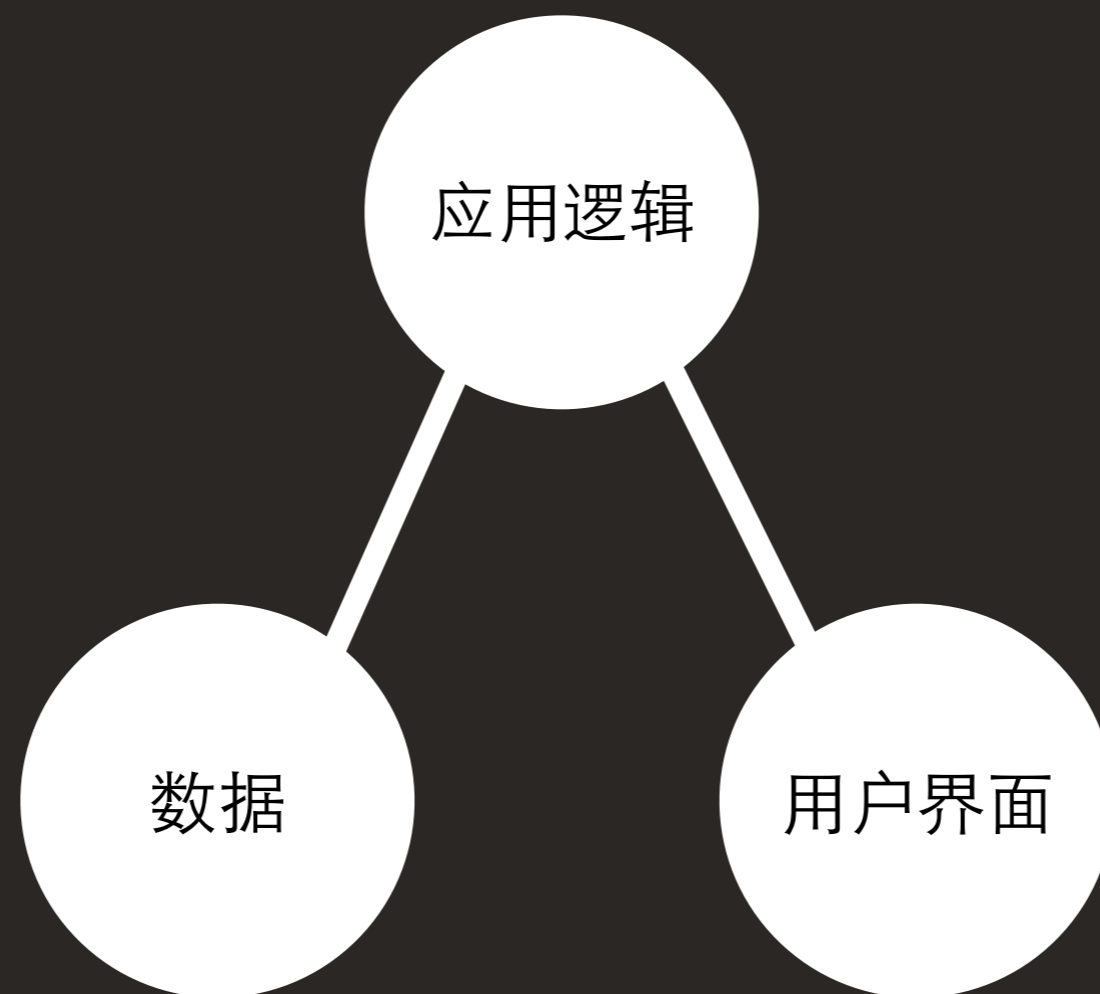
# 目录结构

- ▼  hellorails
  - ▼  app
    - ▶  assets
    - ▶  controllers
    - ▶  helpers
    - ▶  mailers
    - ▶  models
    - ▶  serializers
    - ▶  views
  - ▶  bin
  - ▼  config
    - ▼  environments
      -  development.rb
      -  production.rb
      -  test.rb
    - ▶  initializers
    - ▶  locales
      -  application.rb
      -  boot.rb
      -  database.yml
      -  environment.rb
      -  routes.rb
      -  secrets.yml

- ▼  db
  -  seeds.rb
- ▶  lib
- ▼  log
  -  .keep
  -  development.log
- ▶  public
- ▶  test
- ▶  vendor
-  .gitignore
-  config.ru
-  Gemfile
-  Gemfile.lock
-  Rakefile
-  README.md

# Model View Controller (MVC)

# Controller



**Model**

**View**

# Model

# Model

app/models/product.rb

```
class Product < ActiveRecord::Base
  validates_presence_of :name
  validates_uniqueness_of :serial_number
end
```



# Model

数据库表名



```
class Product < ActiveRecord::Base
  validates_presence_of :name
  validates_uniqueness_of :serial_number
end
```

# Model

数据库表名



```
class Product < ActiveRecord::Base
  validates_presence_of :name
  validates_uniqueness_of :serial_number
end
```

数据库字段校验

# Model

数据库表名

```
class Product < ActiveRecord::Base  
  validates_presence_of :name  
  validates_uniqueness_of :serial_number  
end
```

数据库字段名称

数据库字段校验

# Model

```
class Product < ActiveRecord::Base
  validates_presence_of :name
  validates_uniqueness_of :serial_number
end
```

# Model

```
class Product < ActiveRecord::Base
  validates_presence_of :name
  validates_uniqueness_of :serial_number
end
```

```
product = Product.new name: "PS3", category_id: 2
product.save
```

# Model

```
class Product < ActiveRecord::Base
  validates_presence_of :name
  validates_uniqueness_of :serial_number
end
```

```
product = Product.new name: "PS3", category_id: 2
product.save
```

```
product = Product.create name: "PS3", category_id: 2
```

# Model

```
class Product < ActiveRecord::Base
  validates_presence_of :name
  validates_uniqueness_of :serial_number
end
```

```
Product.find(1)
```

# Model

```
class Product < ActiveRecord::Base
  validates_presence_of :name
  validates_uniqueness_of :serial_number
end
```

```
Product.find(1)
Product.find_by_category_id(2)
```



# Model

```
class Product < ActiveRecord::Base
  validates_presence_of :name
  validates_uniqueness_of :serial_number
end
```

```
Product.find(1)
Product.find_by_category_id(2)
Product.find_all_by_category_id(2)
```

# Model

```
class Product < ActiveRecord::Base
  validates_presence_of :name
  validates_uniqueness_of :serial_number
end
```

```
Product.find(1)
```

```
Product.find_by_category_id(2)
```

```
Product.find_all_by_category_id(2)
```

```
Product.where(category_id: 1).limit(10).order("id desc")
```

# Model

```
class Product < ActiveRecord::Base
  validates_presence_of :name
  validates_uniqueness_of :serial_number
end
```

```
Product.find(1)
```

```
Product.find_by_category_id(2)
```

```
Product.find_all_by_category_id(2)
```

```
Product.where(category_id: 1).limit(10).order("id desc")
```

```
Product.paginate conditions: { category_id: [1,2,3] }, page: 1,
per_page: 10
```

# Model

```
class Product < ActiveRecord::Base
  validates_presence_of :name
  validates_uniqueness_of :serial_number
end
```

```
Product.find(1)
```

```
Product.find_by_category_id(2)
```

```
Product.find_all_by_category_id(2)
```

```
Product.where(category_id: 1).limit(10).order("id desc")
```

```
Product.paginate conditions: { category_id: [1,2,3] }, page: 1,
per_page: 10
```

```
Product.find_by_sql "select * from products where category_id =
'1'"
```

# Model

```
class Product < ActiveRecord::Base
  validates_presence_of :name
  validates_uniqueness_of :serial_number
end
```

# Model

```
class Product < ActiveRecord::Base
  validates_presence_of :name
  validates_uniqueness_of :serial_number

  belongs_to :category
end
```

# Model

```
class Product < ActiveRecord::Base
  validates_presence_of :name
  validates_uniqueness_of :serial_number

  belongs_to :category
end

class Category < ActiveRecord::Base
  has_many :products
end
```

# Model

```
class Product < ActiveRecord::Base
  validates_presence_of :name
  validates_uniqueness_of :serial_number

  belongs_to :category
end
```

```
class Category < ActiveRecord::Base
  has_many :products
end
```

```
product = Product.find(1)
category = product.category
puts category.name
```



# Model

```
class Product < ActiveRecord::Base
  validates_presence_of :name
  validates_uniqueness_of :serial_number

  belongs_to :category
end
```

```
class Category < ActiveRecord::Base
  has_many :products
end
```

```
product = Product.find(1)
category = product.category
puts category.name
```

```
category = Category.find(1)
products = category.products
```

# Model

```
class Product < ActiveRecord::Base
  validates_presence_of :name
  validates_uniqueness_of :serial_number

  belongs_to :category
end
```

```
class Category < ActiveRecord::Base
  has_many :products
end
```

```
product = Product.find(1)
product.price = 98.99
product.save
```

# Model

```
class Product < ActiveRecord::Base
  validates_presence_of :name
  validates_uniqueness_of :serial_number

  belongs_to :category
end
```

```
class Category < ActiveRecord::Base
  has_many :products
end
```

```
product = Product.find(1)
product.price = 98.99
product.save
```

```
product.update_attributes price: 98.99, name: "PS4"
```

# Model

```
class Product < ActiveRecord::Base
  validates_presence_of :name
  validates_uniqueness_of :serial_number

  belongs_to :category
end
```

# Model

```
class Product < ActiveRecord::Base
  validates_presence_of :name
  validates_uniqueness_of :serial_number

  belongs_to :category

  after_create :refresh_category_counter

  private
  def refresh_category_counter
    Category.increment_counter :product_quantity, self.category_id
  end
end
```

# Q & A

# Controller & View

# Controller & View

app/controllers/products\_controller.rb

```
class ProductsController < ApplicationController
  def index
    @products = Product.paginate conditions: { category_id:
params[:category_id] },
    page: params[:page] || 1,
    per_page: params[:per_page] || 20,
    order: "id desc"
  end
end
```



# Controller & View

app/controllers/products\_controller.rb

```
class ProductsController < ApplicationController
  def index
    @products = Product.paginate conditions: { category_id:
params[:category_id] },
    page: params[:page] || 1,
    per_page: params[:per_page] || 20,
    order: "id desc"
  end
end
```

app/views/products/index.html.erb

```
<h1>Products</h1>
<% @products.each do |product| %>
  <li>
    <a href="<%= product_path(product) %>">
      <%= product.name %>
    </a>
  </li>
<% end -%>
```

# Controller & View

app/controllers/products\_controller.rb

```
class ProductsController < ApplicationController
  def index
    @products = Product.paginate conditions: { category_id:
params[:category_id] },
    page: params[:page] || 1,
    per_page: params[:per_page] || 20,
    order: "id desc"
  end
end
```

# Controller & View

app/controllers/products\_controller.rb

```
class ProductsController < ApplicationController
  def index
    @products = Product.paginate conditions: { category_id:
params[:category_id] },
    page: params[:page] || 1,
    per_page: params[:per_page] || 20,
    order: "id desc"

    render json: @products
  end
end
```

# Controller & View

app/controllers/products\_controller.rb

```
class ProductsController < ApplicationController
  def index
    @products = Product.paginate conditions: { category_id:
params[:category_id] },
    page: params[:page] || 1,
    per_page: params[:per_page] || 20,
    order: "id desc"

    respond_to do |format|
      format.html do
        @categories = Category.all
      end
      format.json do
        render json: @products
      end
    end
  end
end
```

# Controller & View

app/controllers/products\_controller.rb

```
class ProductsController < ApplicationController
  def index
    @products = Product.paginate conditions: { category_id:
params[:category_id] },
    page: params[:page] || 1,
    per_page: params[:per_page] || 20,
    order: "id desc"
  end
end
```

# Controller & View

app/controllers/products\_controller.rb

```
class ProductsController < ApplicationController

  before_filter :authenticate_user

  def index
    @products = Product.paginate conditions: { category_id:
params[:category_id] },
    page: params[:page] || 1,
    per_page: params[:per_page] || 20,
    order: "id desc"
  end

  private
  def authenticate_user
    unless logged_in?
      redirect_to login_path
    end
  end
end
```

# Routes

config/routes.rb

```
Rails.application.routes.draw do
  # GET /products => products#index
  # GET /products/2 => products#show
  # GET /products/2/edit => products#edit
  # POST /products => products#create
  # PUT /products/2 => products#update
  # DELETE /products/2 => products#destroy
  resources :products
end
```

# Routes

config/routes.rb

```
Rails.application.routes.draw do
  # GET /products => products#index
  # GET /products/2 => products#show
  # GET /products/2/edit => products#edit
  # POST /products => products#create
  # PUT /products/2 => products#update
  # DELETE /products/2 => products#destroy
  resources :products

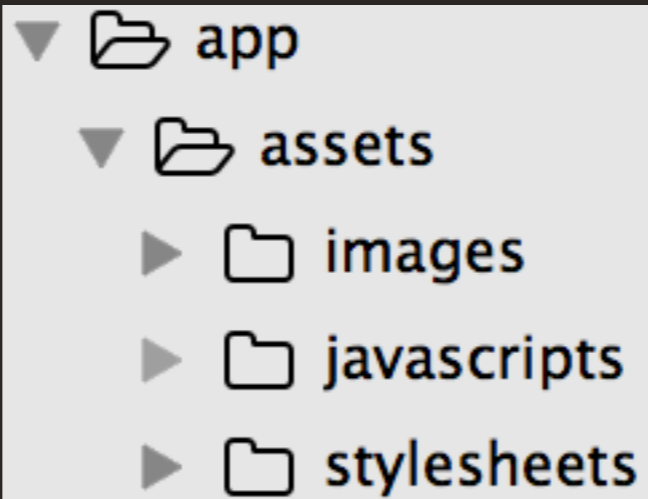
  match '/search' => 'products#search'
end
```



# 其他

- assets
- sass / scss
- rake tasks
- tests
- configurations

# Assets



# 幕后

- Rails其实是有多个Gem组成的一个大框架
  - rails
  - activemodel
  - activerecord
  - activesupport
  - actionmailer
  - actionpack
  - actionview
  - rack

# 其他Gems

- 以及其他第三方Gems
  - mysql2
  - sprockets-rails
  - sass-rails
  - uglifier
  - protected\_attributes
  - will\_paginate
  - etc.

# Q & A

谢谢!