

A large, light pink spiral graphic that starts from the center and expands outwards, framing the title text.

# **Rails 快速上手攻略**

潘旻琦(孝达)





# 日程安排

- Rails 核心技术
- 15 分钟创建博客演示
- 深入学习路线图



# Rails 核心技术

- 模型：Active Record
- 视图：Action View
- 控制器：Action Controller
- 路由：Action Dispatch



# Rails 核心技术

- 辅助：Active Support
- 邮件：Action Mailer
- 前端：sprockets
- 第三方扩展包



# 开始之前的提问时间

- 安装有没有问题？
- 创建项目有没有问题？
- 创建的目录结构看过了吗？



# Active Record

- 对象关系映射
- 数据库迁移
- 数据校验
- 对象生命周期管理
- 对象关联管理
- 数据查询



# 对象关系映射

类名	表名
Post	posts
LineItem	line_items
Mouse	mice
STD	猜一猜
B	猜一猜
A::B	猜一猜



# 对象关系映射

类名	表名
Post	posts
LineItem	line_items
Mouse	mice
STD	stds
B	bs
A::B	bs





# 特殊字段

- id、\*\_id
- created\_at、updated\_at
- lock\_version
- type、\*\_type
- \*\_count
- 保留字 (DangerousAttributeError)



# CRUD

- new、create、create!
- all、first、find\_by、where
- update、update\_all、update\_attribute
- destroy、delete



# 数据库迁移

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

      t.timestamps
    end
  end
end
```



# 数据库迁移

```
class ChangeProductsPrice < ActiveRecord::Migration
  def up
    change_table :products do |t|
      t.change :price, :string
    end
  end

  def down
    change_table :products do |t|
      t.change :price, :integer
    end
  end
end
```



# 数据库迁移

```
$ bin/rails generate migration \  
AddPartNumberToProducts  
  
$ bin/rails generate migration \  
AddPartNumberToProducts part_number:string  
  
$ bin/rails generate migration \  
AddPartNumberToProducts part_number:string:index  
  
$ bin/rake db:rollback  
  
$ bin/rake db:rollback STEP=3
```



# 数据库迁移

```
create_table(:apples) { |t| }  
add_column :products, :part_number, :string  
add_index :products, :part_number  
  
add_reference :products, :supplier,  
              polymorphic: true, index: true  
  
create_join_table :products, :categories,  
                  table_name: :categorization
```



# 数据校验

```
validates :name, presence: true
validates :password, confirmation: true
validates :size, inclusion:
  { in: %w(small medium large) }
validates :name, length: { minimum: 2 }
validates :email, uniqueness: true
validates :password, length: { in: 6..20 }
validates :points, numericality: true
validates :terms_of_service, acceptance: true
validates_associated :books
```



# 对象生命周期管理

```
class User < ActiveRecord::Base
  validates :login, :email, presence: true

  before_validation :ensure_login_has_a_value

  protected
  def ensure_login_has_a_value
    if login.nil?
      self.login = email unless email.blank?
    end
  end
end
```





# 对象生命周期管理 (C)

■ before\_validation

■ after\_validation

■ before\_save

■ around\_save

■ before\_create

■ around\_create

■ after\_create

■ after\_save



# 对象生命周期管理 (R)

■ after\_initialize

■ after\_find



# 对象生命周期管理 (U)

■ before\_validation

■ after\_validation

■ before\_save

■ around\_save

■ before\_update

■ around\_update

■ after\_update

■ after\_save



# 对象生命周期管理 (D)

- before\_destroy
- around\_destroy
- after\_destroy



# 对象关联管理

- belongs\_to
- has\_one
- has\_many
- has\_many :through
- has\_one :through
- has\_and\_belongs\_to\_many



# 数据查询

```
client = Client.find(10)
# SELECT * FROM clients WHERE (clients.id = 10) LIMIT 1

client = Client.take(2)
# SELECT * FROM clients LIMIT 2

client = Client.first
# SELECT * FROM clients ORDER BY clients.id ASC LIMIT 1

client = Client.last
# SELECT * FROM clients ORDER BY clients.id DESC LIMIT 1
```



# 数据查询

```
Post.none
Client.find_by first_name: 'Lifo'
Client.select(:name).distinct
Client.order(created_at: :asc)
Client.limit(5).offset(30)

Order.select(
  "date(created_at) as ordered_date,
  sum(price) as total_price").
group("date(created_at)").
having("sum(price) > ?", 100)
```



# 数据查询

```
Client.find_by first_name: 'Lifo'  
Client.select(:name).distinct  
Client.limit(5)  
  
Order.select(  
  "date(created_at) as ordered_date,  
  sum(price) as total_price").  
group("date(created_at)").  
having("sum(price) > ?", 100)
```





# Action View

- \* ERB / jbuilder
- \* 主模板 / 部分模板
- \* content\_for / yield
- \* Remote



# ERB

```
<h1>Names of all the people</h1>
<% @people.each do |person| %>
  Name: <%= person.name %><br>
<% end %>
```



# jbuilder

```
json.content format_content(@message.content)
json.(@message, :created_at, :updated_at)

json.comments @message.comments, :content, :created_at

json.attachments @message.attachments do |attachment|
  json.filename attachment.filename
  json.url url_for(attachment)
end
```



# 主模板 / 部分模板

```
<%= render "menu" %>

<%= render partial: 'post',
      layout: 'box', locals: {post: @post} %>

<% @products.each do |product| %>
  <%= render partial: "product",
    locals: {product: product} %>
<% end %>
```



# content\_for / yield

```
<html>
  <head>
    <title>Welcome!</title>
    <%= yield :special_script %>
  </head>
  <body>
    <p>Welcome! The date and time is <%= Time.now %></p>
  </body>
</html>
```



# content\_for / yield

```
<p>This is a special page.</p>

<% content_for :special_script do %>
  <script>alert('Hello!')</script>
<% end %>
```



# Remote

```
<%= link_to "a post", @post, remote: true %>
```



# Remote

```
$(function() {  
  return $("a[data-remote]").on(  
    "ajax:success", function(e) {  
      return alert("The post was deleted.");  
    });  
});
```





# Remote

```
<%= form_for(@post, remote: true) do |f| %>  
  ...  
<% end %>
```



# Remote

```
$(document).ready(function() {  
    return $("#new_post").on("ajax:success",  
    function(e, data, status, xhr) {  
        return $("#new_post").append(xhr.responseText);  
    }).on("ajax:error",  
    function(e, xhr, status, error) {  
        return $("#new_post").append("<p>ERROR</p>");  
    });  
});
```



# Action Controller

- 请求参数处理
- 过滤器
- 渲染管理
- 会话管理



# 请求参数处理

```
class ClientsController <
  ApplicationController
  def index
    if params[:status] == "activated"
      @clients = Client.activated
    else
      @clients = Client.inactivated
    end
  end
end
```



# 过滤器

```
class ApplicationController <
  ActionController::Base
  before_action :require_login

  private

  def require_login
    unless logged_in?
      flash[:error] = "You must be logged"
      redirect_to new_login_url
    end
  end
end
```



# 渲染管理

```
class UsersController <
  ApplicationController
  def index
    @users = User.all
    respond_to do |format|
      format.html # index.html.erb
      format.xml { render xml: @users}
      format.json { render json: @users}
    end
  end
end
```



# 会话管理

```
YourApp::Application.config.secret_key_base =  
  '49d3f3de9ed86c74b94ad6bd0...'  
  
YourApp::Application.config.session_store(  
  :cookie_store, key: '_your_app_session',  
  domain: ".example.com"  
)
```



# Action Dispatch

- 资源路由
- 非资源路由





# 资源路由

resources :photos 生成：

- GET /photos photos#index
- GET /photos/new photos#new
- POST /photos photos#create
- GET /photos/:id photos#show
- GET /photos/:id/edit photos#edit
- PATCH/PUT /photos/:id photos#update
- DELETE /photos/:id photos#destroy



# 资源路由

```
resources :photos do
  collection do
    get 'search'
  end

  member do
    get 'preview'
  end
end
```



# 非资源路由

```
get ':controller(/:action(/:id))'
get ':controller/:action/:id/:user_id'
get ':controller/:action/:id/with_user/:user_id'
get 'photos/:id', to: 'photos#show'
get 'exit', to: 'sessions#destroy', as: :logout
match 'photos', to: 'photos#show', via: :all
root to: 'pages#main'
```



# Active Support

```
def set_conditional_cache_control!  
  return if self["Cache-Control"].present?  
  ...  
end
```



# Active Support

```
host = config[:host].presence || 'localhost'
"foo".duplicable? # => true
false.duplicable? # => false
@number.try(:next)
[0, true, String].to_param # => "0/true/String"
quietly { system 'bundle install' }
alias_attribute :login, :email
```



# Action Mailer

```
class UserMailer < ActionMailer::Base
  default from: 'notifications@example.com'

  def welcome_email(user)
    @user = user
    @url  = 'http://example.com/login'
    mail(to: @user.email,
         subject: 'Welcome to My')
  end
end
```



# Action Mailer

```
app/views/user_mailer/  
welcome_email.html.erb
```



# Action Mailer

```
UserMailer.welcome_email(@user).deliver
```





# sprockets

- 静态文件组织结构
- 清单文件结构
- 编译与部署



# 静态文件组织结构

- app/assets
- lib/assets
- vendor/assets



# 清单文件结构

```
// ...  
//= require jquery  
//= require jquery_ujs  
//= require_tree .
```



# 清单文件结构

例如：

- `app/assets/javascripts/home.js`
- `lib/assets/javascripts/moovinator.js`
- `vendor/assets/javascripts/slider.js`
- `vendor/assets/somepackage/phonebox.js`



# 清单文件结构

```
// = require home  
// = require moovinator  
// = require slider  
// = require phonebox
```



# 编译与部署

- `config.assets.precompile += ...`
- `RAILS_ENV=production bin/rake assets:precompile`
- 编译成例如  
`g-908e25f4bf641868d8683022a5b62f54.css`
- 更改`config.action_controller.asset_host`



# 第三方扩展包

- 分页：kaminari
- 用户认证：devise
- 用户授权：cancan
- 上传：paperclip



# 第三方扩展包

- 后台任务：sidekiq
- 全文搜索：sunspot
- 测试统计：simplecov
- 性能统计：rack-mini-profiler





# 第三方扩展包

- 代码高亮：coderay
- 监控：god
- 服务器：thin
- 动作流：public\_activity



# 15 分钟创建博客演示

Demo



# 深入学习路线图：入门

- 学 Rails Guides : <http://guides.rubyonrails.org/>
- 看 <http://railscasts.com> 的视频教程
- 看 《应用 Rails 进行敏捷 Web 开发》
- 阅 ihowe 写的 《Ruby on Rails 實戰聖經》
- 读 《Ruby on Rails Tutorial》 : <http://www.railstutorial.org/book>



# 深入学习路线图：进阶

- 阅读《Why's (Poignant) Guide to Ruby》了解社区文化
- 用 Rails 开发一个完整的项目
- 阅读《Ruby 风格指导》和《Rails 风格指导》
- 用 “Rails for Zombies”：<http://railsforzombies.org/>
- 上 Ruby-China 参与讨论：<https://ruby-china.org/>



# 深入学习路线图：高级

- 阅读工程、扩展包、框架三类项目的源代码
- 为工程项目提交代码，如 `ruby-china`
- 为扩展包项目提交代码，如 `warden`
- 为 `rails` 框架提交代码
- 加入 `rails core team`