

Веб приложение на Ruby on Rails

КОПЫЛОВ АНТОН
anton@kopylov.net



Ruby on Rails

- Ruby – язык программирования
- Rails – фреймворк



Ruby

- интерпретируемый
- объектно-ориентированный
- динамический
- рефлексия



Стиль Rails

- меньше кода
- простота настройки и установки
- соглашения вместо конфигураций
- фокус на решаемой задаче



Основные концепции

- Model View Controller
- Object Relational Mapping
- Test Driven Development



Rails - всё в одном

- MVC
- ActiveRecord, ActiveMailer
- AJAX
- Migration
- Plugins
- ЧПУ
- TestUnit
- Встроенный вебсервер webrick
- и др.



Структура приложения

```
app/  
  - controllers/  
  - helpers/  
  - models/  
  - views/  
components/  
config/  
db/  
  - migrate/  
doc/  
lib/  
public/  
  - images/  
  - javascripts/  
script/  
test/  
  - fixtures/  
  - functional/  
  - integration/  
  - mocks/  
  - unit/  
tmp/  
vendor/
```

создание приложения

```
$ rails myapp  
  create  
  create  app/controllers  
  create  app/helpers  
  create  app/models  
  ...
```

приложение можно сразу запустить

```
$ ruby script/server
```

и посмотреть

```
http://localhost:3000
```



ActiveRecord

файл `app/models/partner.rb`

```
class Partner < ActiveRecord::Base
  has_many :advertisements
  belongs_to :rate

  validates_uniqueness_of :email
  validates_presence_of :email, :name
end
```

```
partner = Partner.find(23)
partner.balance += 100
partner.save
```

```
SELECT * FROM partners WHERE id=23
```



Плагины

Установка плагина `attachment_fu`

```
$ script/plugin install http://svn.techno-weenie.net/projects/plugins/attachment_fu/
```


Модель

```
class Asset < ActiveRecord::Base
  belongs_to :advertisement

  has_attachment :storage => :file_system, :max_size => 5.megabytes,
    :resize_to => '640x480>',
    :thumbnails => {:thumb => '80x80>', :tiny => '40x40>'},
    :processor => :ImageMagick
end

@asset = Asset.new(params[:asset])
@asset.advertisement_id = params[:id]
@asset.save

<%= image_tag(@item.public_filename(:thumb)),
@item.public_filename)%>
```



Migration

db/migrate/011_create_streets.rb

```
class CreateStreets < ActiveRecord::Migration
  def self.up
    create_table :streets do |t|
      t.column "name", :string, :limit=>50
    end

    add_index "streets", ["name"], :name => "name", :unique => true
  end

  def self.down
    drop_table :streets
  end
end
```

ВЫПОЛНИТЬ МИГРАЦИЮ

```
$ rake db:migrate VERSION=11
```



ЧПУ

файл config/routes.rb

```
ActionController::Routing::Routes.draw do |map|
  map.index '', :controller=>'main', :action => 'index'
  map.feed 'rss', :controller=>'main', :action => 'feed'
  map.adv ':id', :controller=>'main', :action=>'show', :id=>/\d+/
  ...
  map.connect ':controller/:action/:id'
end
```

```
<%= link_to "Новости RSS", feed_url %>
```

```
<%= link_to "#", adv_url(:id=>@advertisement.id) %>
```



Кеширование

кеширование страницы целиком

```
class ContentController < ApplicationController
  before_filter :verify_user, :except => :public_content
  caches_page   :public_content
  caches_action :premium_content
```

кеширование фрагментов

```
<% cache(:action => 'list', :part => 'articles') do %>
  <ul>
    <% for article in @articles %>
      <li><%= h(article.body) %></li>
    <% end %>
  </ul>
<% end %>
```

```
expire_fragment(:action => 'list', :part => 'articles')
```



Текстовое поле с выпадающей подсказкой

Форма с текстовым полем ввода

```
<%= text_field_with_auto_complete :street, :name %>
```

простой вариант

```
class MainController < ApplicationController
  auto_complete_for :street, :name
end
```

или вариант сложнее

```
class MainController < ApplicationController
  def auto_complete_for_street_name
    s = params[:street][:name]
    @streets = Street.find :all, :select=>'name', :order=>'name',
      :conditions => "name LIKE '#{s}%"
    render :partial => "street"
  end
end
```

app/views/main/_street.rhtml

```
<ul class="autocomplete_list">
  <% for s in @streets -%>
    <li class="autocomplete_item"><%= s.name %></li>
  <% end -%>
</ul>
```



Capistrano

установка

```
$ gem install capistrano
```

файл с настройками config/deploy.rb

```
set :application, "myapp"  
set :repository, "svn://myapp.ru/myapp/trunk"  
set :deploy_to, "/www/myapp/"  
  
role :app, "myapp.ru"  
role :web, "myapp.ru"  
role :db, "myapp.ru", :primary => true
```

deploy на сервер одной командой

```
$ cap deploy
```

рестарт приложения

```
$ cap deploy:restart
```



Веб сервер Mongrel

установка


```
$ gem install mongrel  
$ gem install mongrel_cluster
```

запуск

```
$ mongrel_rails cluster::start
```

файл настройки mongrel_cluster.yml

```
---  
user: mongrel  
cwd: /www/myapp/current  
log_file: log/mongrel.log  
port: "8000"  
environment: production  
group: mongrel  
address: 127.0.0.1  
pid_file: tmp/pids/mongrel.pid  
servers: 2
```



Развёртывание приложения



- гибкость
- масштабируемость
- требует меньше ресурсов



IDE

для Windows

Aptana Studio

aptana.com

для Linux

Aptana Studio

aptana.com

EasyEclipse for Ruby on Rails

easyeclipse.org



Что дальше

Текущая версия

Ruby – 1.8.6

Rails – 1.2.5

Скоро ожидается (конец 2007-начало 2008)

Rails – 2.0

Ruby – 1.9



Спасибо!

rubyonrails.org

Вопросы?

КОПЫЛОВ АНТОН
anton@kopylov.net
gorodpenza.ru

