

Take a **REST** with Rails

Rob Lee

KodeFoo

(chief monkeyhelper)

What is REST ?

- **R**epresentational **S**tate **T**ransfer
 - An architectural style

What is **REST** ?

- Not just POX
- Mapping HTTP verbs to CRUD actions

RESTful Design

- Identify resources
- Create representation of resource
- Define how we use the resources

Representations

- A resource has_many representations

(X)HTML

RSS

JSON

Resource Mapping

Resource	Method	Action	URI
article	GET	Show articles index	/articles
article	POST	Create new article	/articles
article	GET	Show new article form	/articles/new
article	GET	Show article number 1	/articles/1
article	PUT	Update article number 1	/articles/1
article	GET	Show edit article form	/articles/1;edit
article	DELETE	Delete article number 1	/articles/1

REST design notes

- GET requests are idempotent
 - PUT vs POST
 - Method support varies

Why use REST ?

- Improved response times due to support for caching
 - Scales Horizontally
 - Easy to debug
- Low barrier to entry

REST and Rails – Why ?

DHH says :

“... start thinking about how your application could become more RESTful. How you too can transform that 15-action controller into 2-3 new controllers each embracing a single resource with CRUDing love. This is where the biggest benefit is hidden: A clear approach to controller-design that'll reduce complexity for the implementer and result in an application that behaves as a much better citizen on the general web.”

<http://weblog.rubyonrails.org/2007/1/19/rails-1-2-rest-admiration-http-lovefest-and-utf-8-celebrations>

REST and Rails – When ?

Rails 1.2 :

- `simple_restful` integration
 - `map.resource`

REST and Rails - How ?

```
$rails blog
```

```
$cd blog
```

```
$script/generate scaffold_resource post \
```

```
  title:string body:string created_at:datetime \
```

```
  updated_at:datetime
```

RESTful routing

- Resource based routing : `map.resource`
 - 7 new routes
 - 4 new route helpers

Adding extra actions

```
map.resources :posts, :collection =>  
  { :recent => :get }
```

Method	URL	Action	Helper
GET	/posts;recent	recent	recent_posts_url

Nested Resources

```
map.resources :posts do |post|  
  post.resources :comments  
end
```

Method	URL	Action	Route
GET	/posts/1/comments/5	show	comment_url(:post_id => 1, :id => 5)

Multiple representations

```
respond_to do |format|  
  format.html  
  
  format.xml { render :xml =>  
    @post.to_xml }  
  
  format.yaml { render :text =>  
    @post.to_yaml }  
  
end
```

Consuming **REST**ful services

ActiveResource in edge Rails

config/environment.rb :

```
config.load_paths += %W(  
  #{RAILS_ROOT}/vendor/rails/activeresource/lib )
```

Resource Model

app/models/blog_resource.rb

```
class BlogResource < ActiveRecord::Base
  self.site = 'http://localhost:3000'
end
```

app/models/post.rb

```
class Post < BlogResource
end
```

Yummy !

```
newpost = Post.new(:title => 'Take a REST with Rails',  
  :body => 'A look at REST and Rails in Rails 1.2')  
newpost.save
```

```
post = Post.find_by_title('Take a REST with Rails')  
puts post.description  
=> 'A look at REST and Rails in Rails 1.2'
```

Summary

- REST is here to stay
- Rails REST conventions ease adoption
 - ActiveRecord = ActiveRecord
- Why not SOAP ?