Rails and REST

Peter Battaglia

12/16/08

Topics

- Rails Overview
  - Features
- Rails support for REST
  - Rails Routing
- Rails support for Web Services
  - Active Resource
- Example of Routes in simple Rails app

Rails Overview

- Open source web application framework for the Ruby programming language
- Creator: David Heinemeier Hansson
- MVC Pattern
- Favors Convention over Configuration

Rails Features

- Active Record – O/R Mappings - maps model classes to database tables
- Action Controller – handles all routing and URLs
- Action View – encapsulates all functionality needed to render views
- Active Resource – used for creating and consuming web services
- Database Migrations – makes for easy database management
- Various Generators
  - Project
  - Scaffolding
- Tons of Helpers…
- Etc…

Rails and REST

- Routing
  - 2 Conventions used for routing in Rails
    - Direct mapping of URLs to actions (methods) based on pattern matching, requirements and conditions
      - Example(routes.rb):
        - map.connect ':controller/:action/:id‘
        - map.connect ':controller/:action/:id.:format‘
        - Format can be HTML, XML, JSON, etc…
    - map.resources :model
      - Creates a set of seven new routes and 4 route helpers to the application

Rails and REST

- **Routes and Named Routes Example (rake routes)**
    - **scores** GET /scores {:action=>"index", :controller=>"scores"}
    - **formatted_scores** GET /scores.:format {:action=>"index", :controller=>"scores"}
    - POST /scores {:action=>"create", :controller=>"scores"}
    - POST /scores.:format {:action=>"create", :controller=>"scores"}
    - **new_score** GET /scores/new {:action=>"new", :controller=>"scores"}
    - **formatted_new_score** GET /scores/new.:format {:action=>"new", :controller=>"scores"}
    - **edit_score** GET /scores/:id/edit {:action=>"edit", :controller=>"scores"}
    - **formatted_edit_score** GET /scores/:id/edit.:format {:action=>"edit", :controller=>"scores"}
    - **score** GET /scores/:id {:action=>"show", :controller=>"scores"}
    - **formatted_score** GET /scores/:id.:format {:action=>"show", :controller=>"scores"}
    - PUT /scores/:id {:action=>"update", :controller=>"scores"}
    - PUT /scores/:id.:format {:action=>"update", :controller=>"scores"}
    - DELETE /scores/:id {:action=>"destroy", :controller=>"scores"}
    - DELETE /scores/:id.:format {:action=>"destroy", :controller=>"scores"}

Rails and REST

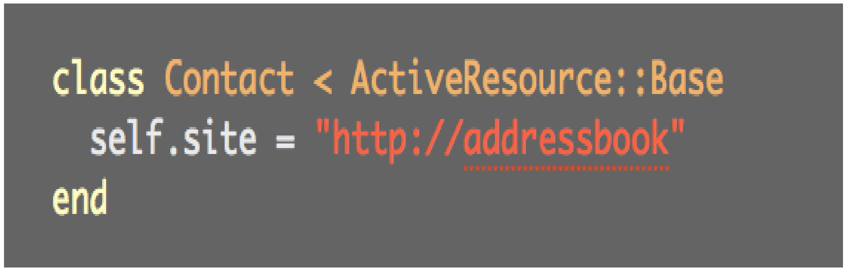
- **Custom Routes**
  - **map.resources :scores**
    - create the 7 Rails standard controller actions of index, create, new, show, update, edit, destroy…..
    - but what if you need custom actions??
  - **Create custom routes….**
    - map.resources :scores, :collection => { :recent => :get }
    - Read as “Add a new action, called “recent” which is invoked via GET which applies to a collection”
    - Adds new route:
      - /scores/recent → recent_scores_url

Rails and Web Services

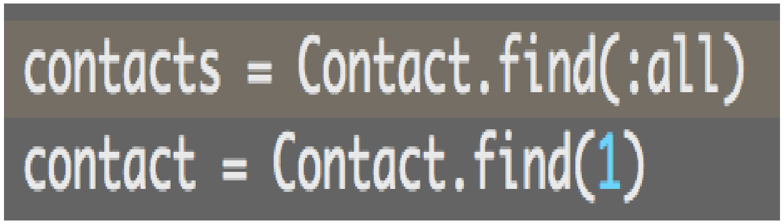
- Active Resource
  - Builds URI
  - Sends and Receives XML Data
  - Processes the responses
  - Provides O-O access to the response details

Rails and Web Services

- Active Resource Example (Ryan Daigle)
  - □ Create a class based on the URL in AR
  - □ Make calls to the services provided by addressbook same as you would with Active Record



```
class Contact < ActiveResource::Base
  self.site = "http://addressbook"
end
```



```
contacts = Contact.find(:all)
contact = Contact.find(1)
```

Questions?