Ruby On Rails

What is Ruby On Rails?

- Known as RoR, or Rails
- http://www.rubyonrails.org
- Web Application Framework
- Developed on Ruby Language
- Developed by David Heinemeier Hansson
- Released under MIT-License

Ruby On Rails (RoR)

- Why Ruby and RoR?
  - dynamic
  - domain specific language
  - simple
  - metaprogramming features

- Why not PHP?
  - several short comings
  - Namespacing
  - Messy

Advantages

- Design Patterns
  - Active Record
  - Model View Controller (MVC)
  - Object Relational Model (ORM)
- Don't Repeat Yourself (DRY)
- Convention over Configuration
- Built in Testing
  - For all elements of MVC, functional and unit
- Ruby
- Rapid Iterative Development
- Database Support:
  - MySQL, PostgreSQL, SQLite, Oracle, SQL Server, DB2

Disadvantages

- Design Patterns
  - Active Record
  - Isn't a Pure MVC
  - Object Relational Model (ORM)
- Lack of (Free?) Documentation
  - for both Ruby and Ruby On Rails

- Not the Silver Bullet
- Needs more support from ISPs
- Tab indents are 2 spaces!

Similar Frameworks

- Some are based on RoR, some not
- PHP
  - Symfony
  - CodeIgnitor
- Java
  - JSP/Struts
- Python
  - Django
  - Turbogears
  - WebPy

Simplistic

```
class Article < ActiveRecord::Base
    has_many :comment
    has_many :category
end

Class Comment < ActiveRecord::Base
    belongs_to :article
end

Class  Category< ActiveRecord::Base
    belongs_to :article
end
```

Ruby On Rails

- Blogging: Typo
- Wiki: Iteraski

Demonstration

Creating A Rails Application

- Create A Rails Application by typing:
  - *rails myoss*
  - This creates a folder called 'myoss' and populates it with required files
- Enter the new directory
- Console: *./script/console*

- Most important folders are
  - app, test, script
- Test out rails by running the Webbrick server
  - *./script/server*
  - Open your browser and go to *http://localhost:3000/*

Databases and Environments

- Enviroments:
  - Development
  - Test
  - Production
- Should be pretty straight forward

- Open
  - *./config/database.yml*
- Edit appropriately

ActiveRecord / Model

- Generate a Model by running
  - *./scipt/generate model post*
  - where 'post' is the name of the model

- Creates:
  - Model: *./app/models/post.rb*
  - Migration: *./db/migration/001-create_post.rb*
  - Unit Test: *./test/unit/post_test.rb*
  - Fixtures: *./test/fixtures/posts.rb*

ActiveRecord / Model Sample

```
class Post < ActiveRecord::Base
end
```

Migration

- Helps to ease 'migration' between different versions of the database
- You can move 'up' and 'down' a migration
- Open and Edit:
  - *./db/migrate/001-create_post.rb*

Migrate my DB!

```
class CreatePost < ActiveRecord::Migration
    def self.up
        create_table :posts do |t|
            t.create :id, :integer
            t.create :title, :string
            t.create :created_at, :datetime
        end
    end

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

Run:

rake migrate version=1

*notice that the table is called 'posts', plural for 'post'

Controller

- What is a Controller/View?
- Run
  - *./scripte/generate controller article*
- Creates
  - Controller: *./app/controller/article_controller.rb*
  - Function Test *./test/functional/article_controller_test.rb*
  - Views:
  - *./app/views/article/* (empty folder)

Controller Sample!

```
class ArticleController < ApplicationController
    def show
        @text = “A surprise will be shown later”
    end
end
```

Views

- ERb: Embedded Ruby
  - Allows you to embed Ruby in files
  - Similar to <?php ?>
- Uses <% %>
  - <% : Normal open Tag
  - <%= : Echo, Print, Puts the result
  - %> : Close Tag
  - -%> : Close Tag with no End of Line Char
    - deprecated?
- Controller instance variables exist in View

View Sample

app/views/article/show.rb

```
<html>
     <head>
          <title>My First Ruby Page</title>
     </head>
</html>
<body>
     <h1>Hello World</h1>
     <p><%= @text %></p>
</body>
</html>
```

Scaffolding

- Rapid CRUD
  - Create Retrieve Update Destroy
- *./script/generate 'model' 'controller'*
- Generates
  - Model, Controllers, Helpers
- Create your own Scaffold!

Testing

- Unit: /test/unit
  - Test out your models
  - Locate problems with your models
- Functional: /test/functional
  - Test our your controllers

Others

- Filtering
- Sweepers
- Helpers
- Ajax/Web2.0
- URL Rewrite
- Deployment
- Extending Rails

- Validations
- Callbacks
- Fixtures
- Associations
- Aggregations

Resources for Ruby and RoR

- Agile Web Development with Rails
  - By Dave Thomas, David Heinemeier Hansson
- Programming With Rails
- Ruby On Rails: Up and Runnning
- Rails Recipe
- Official Rails Resources
  - http://api.rubyonrails.org
  - http://wiki.rubyonrails.org