

Rails CRUD

Goals

- Build a Model
- Build a Controller that creates, reads, updates and destroys
- Display the data in a basic HTML view

Rails Migrations

Migrations

- Manage Database in the project
- Migration DSL
- How do I migrate?

Database Version Management

Migration DSL

- Migrations use a DSL for creating DB structure
- Migrations are built so that they can be reversed

Example Migration

```
class CreateQuestions < ActiveRecord::Migration
  def self.up
    create_table :questions do |t|
      t.string :title
      t.text :body

      t.timestamps
    end
  end

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

create_table/drop_table

```
def up
  create_table :items do |t|
    t.string :name, :null => false
    t.string :description, :default => "Best in class!"
    t.timestamps
  end
end

def down
  drop_table :item
end
```


add/remove column

```
def up  
  add_column :items, :price, :float  
  add_column :items, :premium, :boolean  
end
```

```
def down  
  remove_column :items, :price  
  remove_column :items, :premium  
end
```

execute & others

- execute
- rename_column
- change_column
- add_index
- remove_index

Using Migrations

- rake db:migrate
 - Updates the database to the latest version
- rake db:rollback
 - Rollback the latest migration
- rake db:reset
 - Drop the database and recreate it from migrations

ActiveRecord

ActiveRecord

- Object-Relational Model that ships with Rails
- Table-Oriented

Conventions

- Singular Model class names
- Plural table names

Example Model

```
class User < ActiveRecord::Base
  def self.with_handle(handle)
    where("handle = ?", handle)
  end

  def full_name
    "#{first_name} #{last_name}"
  end
end
```

find

- Search for an id or list of ids

User.find(1)

select * from users where users.id = 1

User.find(1,2,3,5)

select * from users where users.id in (1,2,3,5)

Using Models

- Methods are automatically generated for each column

```
user = User.find(1)
```

```
# Print the user's name to the screen
```

```
puts user.first_name
```

```
# Set the user's age to 10
```

```
user.age = 10
```

Creating

Create John Doe age 18

```
User.create(:first_name => "John",  
            :last_name => "Doe", :age => 18)
```

Also creates John Doe age 18

```
user = User.create do |u|  
  u.first_name = "John"  
  u.last_name = "Doe"  
  u.age = 18  
end
```

```
puts user.id
```

Updating a Record

```
user = User.find(1)
```

```
user.first_name = "Jim"
```

```
user.save
```

Controller Actions

Actions

- Actions are typically the first point in the request cycle you have control
- All instance variables (@variables) are passed to the view

Params Hash

- Params hash has all parameters passed to action
- Query params and other sorts
 - /foo?bar=baz

Rendering

- Render the view named after the method by default
- If the render method is called return the result of that instead
- render is smart about the common formats
 - html
 - json
 - xml

Redirecting

- Sometimes it makes sense to forward the user to another action (object creation or editing)
- You can accomplish this with the `redirect_to` method

`redirect_to :action => :index`

Action Example

```
def index
```

```
  @users = User.limit(params[:page_count])
```

```
  respond_to do |format|
```

```
    format.html #=> render index.html.erb
```

```
    format.json { render :json => @users }
```

```
    format.xml { render :xml => @users }
```

```
  end
```

```
end
```

Views

Views

- Embedded Ruby (erb)
 - Templating language with inline ruby
- Layouts
- Helpers
 - Modules that contain code for views
 - Rails has many built in helpers

ERB Templates

```
<% if @foo %>
```

```
  <span>@foo is truthy!</span>
```

```
<% end %>
```

```
<% @users.each do |user| %>
```

```
  <div>Name: <%= user.full_name %></div>
```

```
  <div>Age: <%= user.age %></div>
```

```
  <div><%= image_tag "user_icon.png" %></div>
```

```
<% end %>
```

Helpers

- Asset Helpers
 - javascript_include_tag
 - stylesheet_include_tag
 - image_tag

Asset Helpers

```
<%= javascript_include_tag "underscore" %>  
<script type="text/javascript" src="javascripts/underscore.js"  
></script>  
  
<%= stylesheet_include_tag "style" %>  
<link rel="stylesheet" type="text/css" src="stylesheets/style.css"  
></script>  
  
<%= image_tag "icon.png" %>  

```