

Ruby on Rails Tutorial

Jieun Oh

February 16, 2011

For Tom's & my examples, see: <https://ccrma.stanford.edu/~jieun5/TA/256b/tutorial3/>

1. Installation - for Mac OSX (Snow) Leopard

In Terminal:

<pre>sudo gem update sudo gem update --system sudo gem install rails sudo gem install sqlite3</pre>	<pre>// you may see "ERRORs" but they are really ok // this updates to latest version of ruby // updates Rails (should be 3.0 or higher) // install Sqlite3</pre>
---	---

2. Create a Rails project

In Terminal:

- Navigate to where you want your project to be. Then type:

```
rails new <projectName>
// ex: rails new ro/tutorial
// You should see many new files automatically created for you.
```

3. Create a Controller

In Terminal (in your Rails project):

```
rails generate controller <ControllerName> [options]
// ex: rails generate controller Simple post get
```

Note:

- This command generates a "Simple" **controller** with template for "submit" and "get" **actions** inside of it.
- Also generated are a test file, view helper, and view files (for each action), but we don't need to deal with them for our purpose.
- In fact, you should get rid of the files under app/views/simple, because they will be called from the controller by default.

4. Create a Model

In Terminal:

```
rails generate model <ModelName> [options]
// ex: rails generate model event
```

Note:

This command generates a **model** named "event", and a migration file <timestamp>_create_events.rb (inside db/migrate), among other things.

5. Specifying the Columns of a DB table

In db/migrate/<timestamp>_create_events.rb:

```
class CreateEvents < ActiveRecord::Migration
  def self.up
    create_table :events do |t|
      t.column :udid, :string
      t.column :latitude, :float
      t.column :longitude, :float
      t.column :fav_number, :int
      t.column :fav_phrase, :string
      t.column :audio_filename, :string
      t.column :picture_filename, :string
    end
    t.timestamps
  end
end

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

6. Run migration

In Terminal:

```
rake db:migrate
```

Runs all migrations to bring the database up to date.

7. Implement Controller

In app/controllers/simple_controller.rb:

```
class ProcessController < ApplicationController

  # Posts information from client to table
  def post
    @event = Event.new
    @event.udid = params[:udid]
    @event.latitude = params[:latitude]
    @event.longitude = params[:longitude]
    @event.fav_number = params[:fav_number]
    @event.fav_phrase = params[:fav_phrase]

    if @event.save then
      render :text => "success!"
    else
      render :text => "failed..."
    end
  end
end
```

```

# Gets information from table to client
def get
  @event = Event.find(:last)
  # raw formatting by hand (alternative: JSON, XML, plist...)
  returnText = "<udid: " + @event.udid + ">" +
    "<latitude" + @event.latitude.to_s + ">" +
    "<longitude" + @event.longitude.to_s + ">" +
    "<favorite number" + @event.fav_number.to_s + ">" +
    "<favorite phrase" + @event.fav_phrase + ">"
  puts returnText
  render :text => returnText
end
end

```

8. Circumventing “Invalid Authenticity Token”

In app/controller/application_controller.rb:

delete the line “protect_from_forgery”

(Generally, not a good thing to do...)

9 Set Routes

- Default URL is http://localhost:3000/<controllerName>/<actionName>
- Let's just use that from our client. :)
- In routes.rb, uncomment the last line: `match ':controller(/:action(/:id.:format))'`

10. Start Server

In Terminal:

```
rails server
```

A small web server named “WEBrick” comes bundled with Ruby...

11. Testing with Client: Variables

Download iPhone client code example, and test GET/POST on variables

https://ccrma.stanford.edu/~jieun5/TA/256b/tutorial3/jieun/web_final.zip

Now, onto files! How do we GET/POST files (i.e. audio, picture)?

12. Sending & Receiving Files: Server-Side

In app/controllers/simple_controller.rb

```
class SimpleController < ApplicationController
```

```

# Posts information from client to table
def post
  #handle variables
  @event = Event.new
  @event.udid = params[:udid]
  @event.latitude = params[:latitude]
  @event.longitude = params[:longitude]
  @event.fav_number = params[:fav_number]
  @event.fav_phrase = params[:fav_phrase]

  # handle file
  name = params[:audio_filename].original_filename
  directory = "public/audio"
  path = File.join(directory, name)
  @event.audio_filename = name

  if @event.save then
    File.open(path, "wb") do |f|
      f.write( params[:audio_filename].read )
    end
    render :text => "success!"
  else
    render :text => "failed..."
  end
end

# Gets information from table to client
def get
  @event = Event.find(:last)
  # raw formatting by hand
  returnText = "udid: " + @event.udid + "\n" +
    "latitude: " + @event.latitude.to_s + "\n" +
    "longitude: " + @event.longitude.to_s + "\n" +
    "favorite number: " + @event.fav_number.to_s + "\n" +
    "favorite phrase: " + @event.fav_phrase + "f\n" +
    "audio filename: " + @event.audio_filename
  puts returnText
  render :text => returnText
end
end

```

Inside “public” folder, **create a folder named “audio”** where all audio files from clients will be stored.

13. Sending & Receiving Files: Client Side

POST:

```

- (IBAction) submitData
{
    NSLog(@"about to send data to server...");

```

```

NSString *server = [[NSString alloc] initWithUTF8String:serv];
NSURL *url = [NSURL URLWithString:
               [server stringByAppendingString:@"simple/post"]];

ASIDataRequest *request = [ASIDataRequest requestWithURL:url];

// Setup file name from the application sandbox
NSString * myFile;
if( g_wav == 0 )
    myFile = [[NSBundle mainBundle] pathForResource:@"kick" ofType:@"wav"];
else if( g_wav == 1)
    myFile = [[NSBundle mainBundle] pathForResource:@"duck" ofType:@"wav"];
else if( g_wav == 2)
    myFile = [[NSBundle mainBundle] pathForResource:@"dogbun" ofType:@"wav"];
else
    myFile = [[NSBundle mainBundle] pathForResource:@"baby" ofType:@"wav"];

// Setup the file for the request
[request setFile:myFile forKey:@"audio_filename"];

// Setup other variables for the request
request = setupPostVars(request);

[request startSynchronous];
NSError *error = [request error];
if (!error) {
    NSString *response = [request responseString];
    [status setText:response];
}
}

```

GET:

```

// parse string to figure out filename
string filename = s.substr(int(found) + toFind.length() + 1);

// get data at URL
NSString *URLString = [NSString stringWithFormat:@"%s%@", serv, @"audio/",
                       filename.c_str()];
NSURL * URL = [NSURL URLWithString: URLString];
NSData * audioData = [NSData dataWithContentsOfURL:URL];

if( audioData == nil ) {
    NSLog(@"Error fetching audio data!");
}

// write data to sandbox
NSArray *paths = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory,
                                                     NSUserDomainMask, YES);
NSString *documentsDirectory = [paths objectAtIndex:0];
NSString * filePath = [NSString stringWithFormat:@"%s/%s",
                      documentsDirectory,
                      [audioData writeToFile: filePath atomically: YES]];

```

```
// Play using STK FileReader. Hint:
NSString *file = @"/Documents/file.wav";
NSString *path = [NSHomeDirectory() stringByAppendingString: file];
const char *cPath = [path UTF8String];
reader->open(cPath);
```

Download complete iPhone client code here: https://ccrma.stanford.edu/~jjeun5/TA/256b/tutorial3/jjeun/web_file_final.zip

(It's a bit messy, but the functionality is there...)

Helpful links and documentations

- Rails Guides:
<http://guides.rubyonrails.org/>
- Ruby on Rails API:
<http://api.rubyonrails.org/>
- Rails ActiveRecord Class Documentation: for working with database tables in Rails
<http://api.rubyonrails.org/classes/ActiveRecord/Base.html>
- SQLite Manager Firefox Add-On: one of ways to browse contents of SQLite3 file, located in your Rails project inside "db" folder
<https://addons.mozilla.org/en-US/firefox/addon/sqlite-manager/>
- ASIHTTPRequest: to make HTTP requests from iPhone clients (alternative: NSURLConnection, see Tom's example)
<http://allseeing-i.com/ASIHTTPRequest/>