

6.170 Tutorial 0 - Development Environment Setup and First App

Warning: Installation can be quite time consuming as each system/laptop has a slightly different environment and you may have to install some software in addition to those listed below. Get started with this soon and contact the TAs if you have questions.

Prerequisites

1. A laptop (or a desktop) with Windows 7, MAC OS, or Linux (Ubuntu 12.04) installed.

Goals of this Tutorial

1. Install Ruby on Rails on your computer.
2. Install Git on your computer.
3. Signup for GitHub account. (<http://www.github.com>)
4. Signup for a Heroku account. (<http://www.heroku.com>)
5. Create a “Hello World” app using Ruby on Rails and deploy it to Heroku

Step 1: For Windows Users Only - Install Ruby and Rails

Ruby Installation

Install the latest version of Ruby (1.9.3) by downloading the installer at <http://cdn.rubyinstaller.org/archives/1.9.3-p327/rubyinstaller-1.9.3-p327.exe>

This is the official Ruby installer for Windows system. After downloading, double-click on it and follow the wizard. Using the default options will be fine. After the install, add the bin directory to your path environment variable(Default options => add “C:\Ruby193\bin” to system Path variable).

Rails Installation

Open the command prompt window and install rails by executing the command: “gem install rails”. This installs Rails and the dependent gems. (“Gems” are ruby packages/libraries)

Step 1: For MAC OSX Users Only - Install Ruby and Rails

Make sure XCode is installed, we will need the gcc compiler and it comes with a plethora of command line tools. Expect to run into a lot of issues, set aside substantial time to install Ruby/Rails. Contact TAs with your questions.

The Long

Follow instructions at:

<http://ruby.railstutorial.org/ruby-on-rails-tutorial-book#sec-rubygems>

The Short

Install XCode (app store) → XCode → Preferences → Downloads → Command Line Tools

```
$ sudo \curl -L https://get.rvm.io | bash -s stable --ruby  
$ rvm get head && rvm reload
```

Make sure you have gcc-4.2 with `which gcc-4.2`

```
$ rvm install 1.9.3
```

Download RubyGems (http://rubyforge.org/frs/?group_id=126)

```
$ ruby setup.rb  
$ gem update --system 1.8.24  
$ gem install rails -v 3.2.11
```

Note: If you encounter the error “There has been an error while running make. Halting the installation,” install HomeBrew using:

```
$ ruby -e "$(curl -fskL raw.github.com/mxcl/homebrew/go)"
```

Step 1: For Linux (Ubuntu 12.04) Users Only - Install Ruby and Rails

Follow instructions at:

<http://blog.sudobits.com/2012/05/02/how-to-install-ruby-on-rails-in-ubuntu-12-04-lts/>

Step 2: Testing Rails Installation

Test rails is installed: open a command window and go to the directory you want your projects and run the following command:

```
$ rails new helloworld
```

This command will create a new directory named “helloworld” and bunch of directories like “app”, “config”, “db”, “doc”, “lib”, “log” etc under it. You will become very familiar with these directories and their contents over the course of the semester.

Start the web server by running the following commands in a terminal window:

```
$ cd helloworld  
$ rails server
```

You should see an output line like: “WEBrick::HTTPServer#start: pid=3433 port=3000” which confirms that the web server is started. Now open your web browser and go to “http://127.0.0.1:3000/”. If you see the “Welcome aboard” page, Ruby on Rails is successfully installed on your computer!

Step 3: Install and Setup Git, GitHub

Install Git

For Windows Users

Download the installer from <http://code.google.com/p/msysgit> and run it

For MAC and Linux Users

Check if you already have Git installed by running the command:

```
$ which git
```

If you do not have Git, install it using:

```
$ sudo apt-get install git
```

Note: Details of Git, GitHub, and repository creation will be covered later in the recitation for Week #1. Please have Git installed on your laptops and be signed up on GitHub in preparation.

Create a GitHub Repository

Create a GitHub repository named “projectzero”.

Important: The GitHub repository name must be “projectzero” for the instructions in this tutorial to work successfully.

More details on Git and GitHub are covered in recitation #1

Clone the GitHub Repository

```
git clone <url_of_your_github_repository>
```

More details on Git and GitHub are covered in recitation #1

Step 4: Create Your Rails App for “Project P0”

Now we will create a new rails app called “projectzero”, add the required code, push it to GitHub and to heroku. Before running the commands shown below, make sure that you have:

1. Created a GitHub repository for this assignment
2. Cloned it to a local directory (referred as <cloned_repository> below)

The details of cloning a GitHub repository will be covered in recitation #1.

In a command window, run the following commands:

```
$ cd <parent_of_projectzero>
```

Warning: If you are not in the parent directory of “projectzero” before executing the instructions shown below, you will very likely end up with a directory structure which Heroku does not understand. This will result in a “pre-receive hook failure” error when pushing your app to Heroku.

```
$ rails new projectzero
```

Note: You might be asked for permission to overwrite the .gitignore file. You can say NO to this request.

```
$ cd projectzero
$ git add .
$ git commit -m s"Created app for Project Zero"
$ git push origin
```

In a new command window, start the app as follows:

```
$ rails server
```

Go to <http://localhost:3000/> in your web browser and you should see the generic page for a new rails app.

In the first command window, execute the following:

```
$ rails generate controller hello sayhi
```

This will generate the necessary controller and view files. The "projectzero/app/controller/hello_controller.rb" file should look like:

```
class HelloController < ApplicationController
  def sayhi
  end
end
```

Edit "projectzero/app/views/hello/sayhi.html.erb" file so that it looks like:

```
<h1>Hello World!</h1>
```

In "projectzero/config/routes.rb", find line beginning with "# root :to =>".

Replace that line with:

```
root :to => "hello#sayhi"
```

Delete "projectzero/public/index.html", which is the default Welcome to Ruby on Rails page. Otherwise this will override our routing.

Now when you access http://localhost:3000, you should see a new welcome page which says "Hello World!"

Step 5: Setup Heroku

Signup for a Heroku account at <http://www.heroku.com> Install required packages by running the

following commands in a command window:

For mac and linux users:

```
$ sudo gem install heroku foreman
```

For windows users:

```
C:\> gem install heroku foreman
```

A few more configuration steps to work with heroku:

```
$ ssh-keygen -t rsa
```

```
$ heroku keys:add
```

The above commands create RSA keys to authenticate you whenever you use the heroku command line and to establish a secure communication channel between your computer and heroku servers. You can find more details at: <https://devcenter.heroku.com/articles/keys>

To create your first app on heroku, run the following command:

```
$ heroku create
```

You will see output similar to:

```
$ Creating afternoon-waters-7298... done, stack is cedar
```

```
$ http://afternoon-waters-7298.herokuapp.com/
```

```
$ git@heroku.com:afternoon-waters-7298.git
```

You can now view your app at a URL like <http://afternoon-waters-7298.herokuapp.com/>.

Important: Remember to include this URL of your app in the "projectzero/README.md" file before you turn in your project.

When heroku created the app for you, it also created a remote git repository to which you can push your code. Make your local git aware of this remote repository by running the commands:

```
$ cd projectzero
```

```
$ git remote add projectzero git@heroku.com:afternoon-waters-7298.git
```

Note: Replace "afternoon-waters-7298" in the command with the name of your heroku app

Step 6: Deploy your Project Zero app to Heroku

Make the following changes to "projectzero/Gemfile"

Delete the line "gem 'sqlite3'" from this file.

Add the following section at the end of the file:

```
group :development, :test do
  gem 'sqlite3'
```

```
end

group :production do
  gem 'pg'
end
```

The reason for doing this change is that “sqlite3” is the default development database used by rails. However, it is a very lightweight database with limited features and is not suitable for real world applications. So Heroku expects you to use a more powerful database system (like postgres). The above lines instruct rails to use sqlite3 database in development environment and to use postgres in production environment. If you don't make this change to the Gemfile, your deployment to heroku in the next step will fail.

Install all required gems mentioned in this Gemfile:

```
$ cd projectzero
$ bundle install
```

Lets check the status of the local Git repository before pushing it to heroku:

```
$ git status
```

This will list your modified files as well as new files which are in your directory but not in Git repository. Add required files to the repository using:

```
$ git add <path_to_file>
```

Now commit your changes to the repository using:

```
$ git commit -am "Added 'Hello World' code"
```

To deploy the app to heroku, run the following commands:

```
$ cd projectzero
$ git push projectzero master
```

This will push your code to the git repository on heroku deploy it. Once successfully deployed, you can now open a web browser and view your app at your app URL like <http://afternoon-waters-7298.herokuapp.com/>

Step 7: Push your app to GitHub

Once you have your app working locally and also on heroku, you are almost ready to push it to GitHub. In a command window, run the following commands:

```
$ cd projectzero
$ git push origin master
```

This command will ask you for your GitHub username and password and then upload your code to the github repository.

Step 8: Submit a “Pull Request” on GitHub

Note: This part will be covered in recitation #1.

Once you successfully push your code to GitHub, you must submit a “pull request” via GitHub. This will inform your TAs that your project code is uploaded to Github.

Additional Notes:

1. An excellent tutorial for installing RoR on Mac is available at:
<http://www.moncebelyamani.com/how-to-install-xcode-homebrew-git-rvm-ruby-on-mac/>
2. For Windows,
 - a. Download and install the DevKit from: <http://rubyinstaller.org/downloads>
 - b. Follow the instructions at:
<http://github.com/oneclick/rubyinstaller/wiki/Development-kit>

MIT OpenCourseWare
<http://ocw.mit.edu>

6.170 Software Studio
Spring 2013

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.