Intro to Rails

MIS 21

A.Navarro, R. Alampay

INFORMATION SYSTEMS &
COMPUTER SCIENCE

Learning Objectives

- Determine the steps to start up a new Rails application
- Become familiar with Rails app conventions
- Learn how routes work
- Learn how to use generators

Ruby on Rails (RoR)

- An open source web application framework written in Ruby
- Utilizes well known design patterns such as ActiveRecord
- Created initially by David Heinemeier Hansson in 2004 as a result of his work on Basecamp(http://basecamphq.com)
- In 2008, merged with another well known ruby application framework called Merb
- The current iteration is v4.1 and is actively maintained by a staff of Ruby programmers

Sites built on Rails

- http://basecamphq.com
- Early version of http://twitter.com
- http://github.com
- http://groupon.com
- http://hulu.com
- http://scribd.com
- http://assembla.com

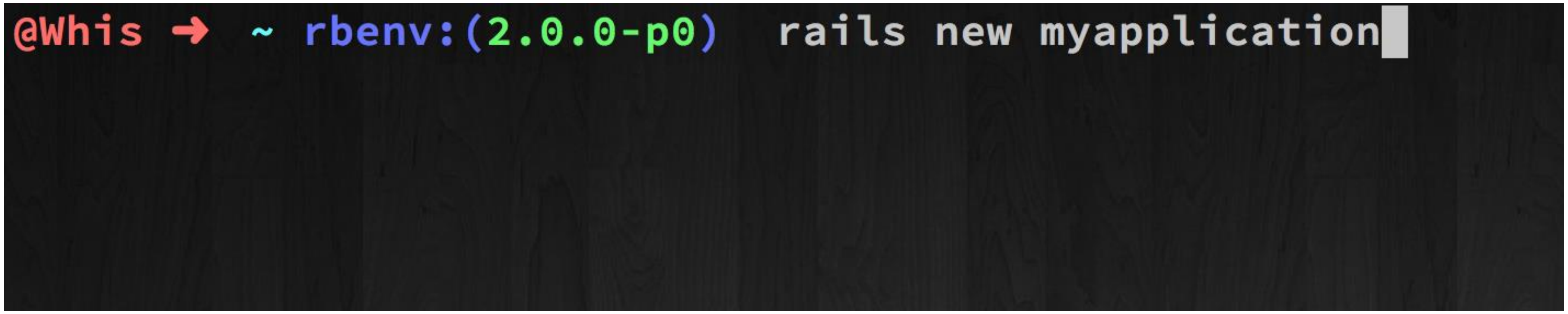
INFORMATION SYSTEMS &
COMPUTER SCIENCE

Ruby on Rails Core Principles

- Convention over configuration
- DRY – do not repeat yourself
- MVC – Model View Controller
- Fat Models, Skinny Controllers
- Agile Development

Creating a new rails application

- Creating an application is done by running the command



```
@Whis ➜  ~ rbenv:(2.0.0-p0)  rails new myapplication
```

- This creates a folder called “myapplication” which contains a rails project. This can be replaced with any app name.

INFORMATION SYSTEMS & COMPUTER SCIENCE

Javascript Runtime

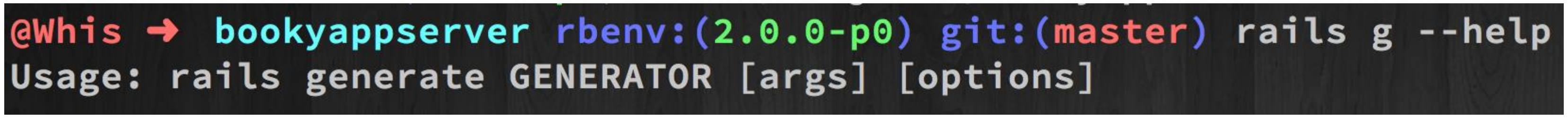
- Rails requires a javascript runtime in order to manipulate javascript and css files
- Find the line with "rubyracer" in your Gemfile and uncomment it then run bundle install
- Alternatively, you can install nodejs

rails folders

- app – contains your main application components
  - controllers – controllers handle web requests
  - models – models wrap the data found in our data stores
  - views – views contain our display templates
  - assets – contains files that are loaded by a browser (CSS, JS, Images)
- config – contains configuration code for our app
- db – contains code related to building the database
- lib – user created libraries/code (not a M,V or C)
- log – contains error logs
- public – contains any static files that don’t change
- test – contains unit, functional and integration tests

Generators

- Most application code is created using generators
- Some gems include generators to ease in bootstrapping of code
- Generators follow standard rails conventions



```
@Whis ➜  bookyappserver rbenv:(2.0.0-p0) git:(master) rails g --help
Usage: rails generate GENERATOR [args] [options]
```

INFORMATION SYSTEMS & COMPUTER SCIENCE

Models

- Models are classes used to represent rows in a database

Generator Syntax:

```
rails generate MODEL_NAME
fieldname1:fieldtype1 fieldname2:fieldtype2
```

Example:

```
rails generate Entry name:string address:string
number:string
```

- Fields are optional
- Field types are optional. If no field type is specified, the default is string
- Model generators also create a migration

Migrations

- Migrations are classes that define database structure.
- Saves user from writing SQL create statements for their database
- Found under the db/migrate folder
- To execute migrations run the command:

```
rake db:migrate
```

- To cancel the last executed migration run the command:

```
rake db:rollback
```

Controllers

- Controllers contain logic to handle web requests
- May use models to retrieve or save data
- Have actions (methods) that render corresponding views
  - A controller will have a corresponding folder under app/views

Generator Syntax:

```
rails generate CONTROLLER_NAME action1
action2
```

Example:

```
rails generate Items index new edit
```

INFORMATION SYSTEMS & COMPUTER SCIENCE