



Rapid Development with Ruby, Jruby and Rails

Joey Shen

Technology Evangelist

Sun Microsystems Inc.



Agenda

- Introduction of Ruby
- Introduction of JRuby
- Get on Rails
- JRuby on Rails
- Summary

Agenda

- Introduction of Ruby
- Introduction of JRuby
- Get on Rails
- JRuby on Rails
- Summary

What is Ruby?

- Dynamically-typed pure object-oriented programming language
 - > Focus on simplicity, productivity, and “fun” especially
 - > Interpreted
 - > Open source, written in C
 - > Implementation is the Specification
- Created 1993 by Yukihiro “Matz” Matsumoto
 - > “More powerful than Perl and more OO than Python”
- Very active community, wide range of apps

Ruby Quick Tour: Pure OO

- Everything is an Object
- Even a number
 - > 7.class => Fixnum
 - > "hello".length => 5
 - > 3.times { |i| puts "Number #{i}" }
 - => Number 1
 - => Number 2
 - => Number 3

Ruby Quick Tour: Simplicity

- Most likely your intuition is right
- Speaking Ruby rather than Programming Ruby
 - > a = :red; b = :green; c = :blue
a, b, c = b, c, a
puts a, b, c
=>green
=>blue
=>red
 - > 1.upto 3 { | i | puts "Number #{i}" }
=>Number 1
=>Number 2
=>Number 3

Ruby Quick Tour: Dynamic Typing

- No need to declare variables
- Duck-typing
 - > If it quacks like a duck and waddles like a duck, then...

```
def make_it_waddle(waddler)
  waddler.waddle
end
```

```
make_it_waddle(Duck.new)
make_it_waddle(Penguin.new)
make_it_waddle(Octopus.new)
```

Ruby Quick Tour: Blocks

- Anonymous methods you can pass around and invoke
 - > Add block to be invoked on end of method call
 - > Removes repetitive code
 - `list.each { |element| ... do something ... }`
 - > Internalizes transactional code
 - `File.open(fname) { |file| line = file.gets ... }`
 - > Makes nice APIs

Ruby Quick Tour: Modules

- Modules mix-in behavior

```
module ShapeStuff
  def diameter
    return 2 * @radius
  end
end
```

```
class Circle
  include ShapeStuff
  ...
end
```

```
Circle.new(4).diameter
```

=> 8

Ruby Quick Tour: Metaprogramming

- Meta Programming
 - > Dynamically generate classes
 - > Modify existing classes
 - Add/Alias/Remove/Replace methods
 - Include a module
 - Add instance/class variable
- Special dispatching to pretend things exist
 - > `method_missing`, `const_missing`

Ruby Quick Tour: Method Aliasing

```
class Multiplier
  def double_your_pleasure(pleasure)
    return pleasure * 3 # FIXME: Actually triple
  end
end
```

```
m = Multiplier.new
m.double_your_pleasure(6)           # => 18
```

```
class Multiplier
  alias :double_your_pleasure_BUGGY :double_your_pleasure
  def double_your_pleasure(pleasure)
    return pleasure * 2
  end
end
```

```
m.double_your_pleasure(6)           # => 12
m.double_your_pleasure_BUGGY(6)     # => 18
```

Ruby Quick Tour: Method missing

- One way to respond infinite methods call on the instance

```
class MyClass
  def defined_method
    'This method is defined.'
  end

  def method_missing(m, *args)
    "Sorry, I don't know about any #{m} method."
  end
end

o = MyClass.new
o.defined_method      # => "This method is defined."
o.undefined_method   # => "Sorry, I don't know about
                      any undefined_method method."
```

Agenda

- Introduction of Ruby
- Introduction of JRuby
- Get on Rails
- JRuby on Rails
- Summary

What is JRuby?

- Started in 2002
- Java platform implementation of Ruby language
- Open source, many active contributors
- Aiming for compatibility with current Ruby version
- Integrates with Java technology
 - > Call to Ruby from Java technology via JSR 223
 - > Use Java class files from Ruby
- Growing set of external projects based on JRuby
 - > Jruby-extras (ActiveRecord-JDBC,...)

JRuby: Java Technology Integration

- Script Java
 - > Power of Java technology with the Syntax of Ruby

```
require 'java'
```

```
list = java.util.ArrayList.new
```

```
list << 1
```

```
list << 3
```

```
list << 2
```

```
list.sort
```

Why JRuby?

- JRuby over Ruby
 - > Strong likelihood JRuby will be faster soon
 - > Better scalability with native threading
 - > Native unicode support
 - > Compilation
 - > Integration with Java libraries
 - > Easier path to getting Ruby in the enterprise
- JRuby over Java
 - > Language features
 - Blocks, modules, metaprogramming, dynamic-typing
 - > Ruby applications/libraries
 - Rails, Rake, Raven,...

Agenda

- Introduction of Ruby
- Introduction of JRuby
- **Get on Rails**
- JRuby on Rails
- Summary

What is Ruby on Rails

- A full-stack MVC web development framework
- Open source (MIT), many contributors
- Written in Ruby
- Single-threaded design
- First released in 2004 by David Heinemeier Hansson
- Ruby and Rails book sales are outselling Perl book sales
- Shipping with “Leopard”

Ruby on Rails Precepts

- Convention over configuration
 - > Why punish the common cases?
 - > Encourages standard practices
 - > Everything simpler and smaller
- Don't Repeat Yourself
 - > Reamework witten around minimizing repetition
 - > Repetitive code harmful to adaptability
- Agile development enviroment
 - > No recompile, deploy, restart cycles
 - > Simple tools to generate code quickly
 - > Testing built into framework

Why Ruby on Rails

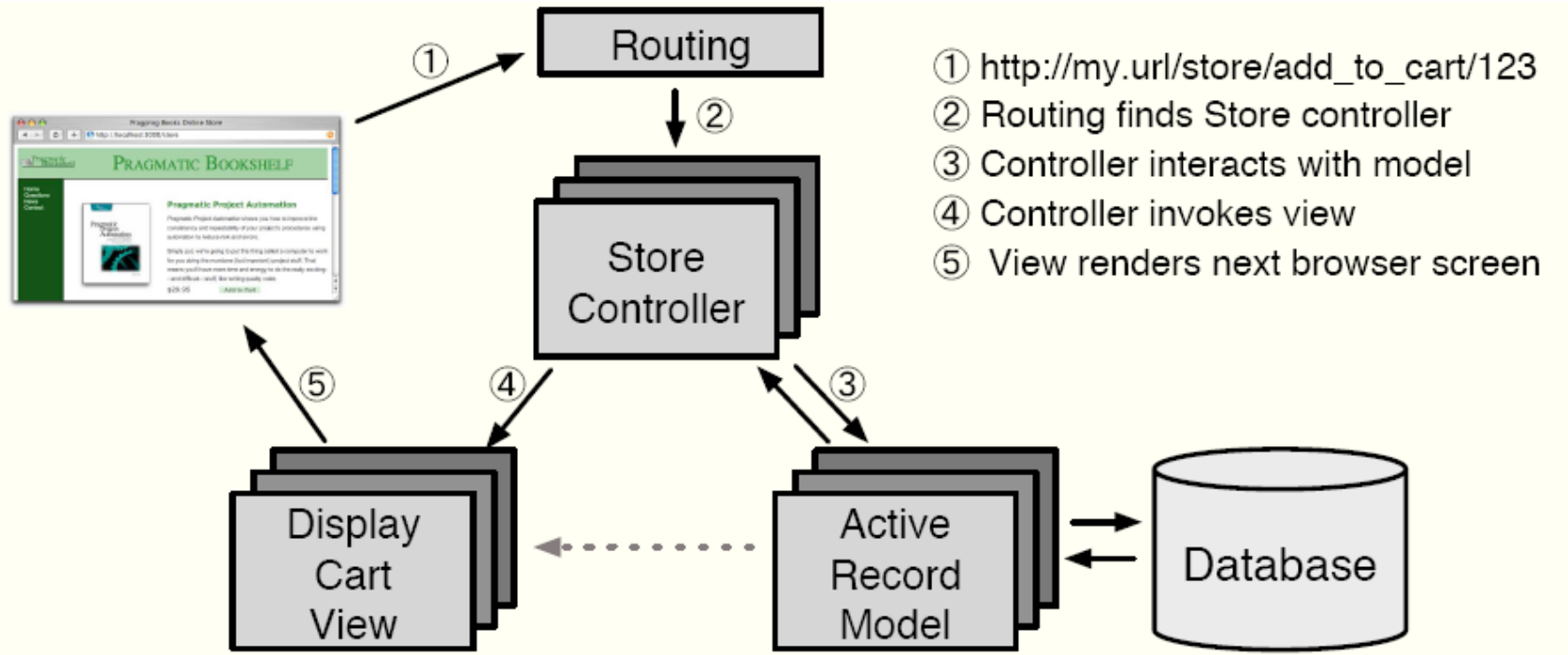
- Greatly simplified web development
 - > “Less Rails code than Java application configuration”
 - > Instant applications: working code in minutes
- Growing excited community of developers
- Makes small apps trivial to create
- Ruby in an excellent language

Rails Quick Tour: Project Layout

- All Rails apps have same layout

```
myapp
  + app
    + controllers
    + helpers
    + models
    + views
  + config
  + db
  + lib
  + public
  + test
```

Rails Quick Tour: MVC



Rails Quick Tour: Tools

- 'rails myapp' create new application
- 'ruby script/generate' generates:
 - > Models, Views, and Controllers
 - > Migrations (Ruby DB-agnostic DDL management)
 - > Scaffolding (quick CRUD maker)
 - > Tests (unit and functional)
- 'rake' runs tasks
 - > Perform migration
- 'ruby script/server' starts application
- Many more ...



Demo

Ruby on Rails with Netbeans

Agenda

- Introduction of Ruby
- Introduction of JRuby
- Get on Rails
- JRuby on Rails
- Summary

Why Jruby on Rails?

- Deployment to Java application servers
- Java technology production environments pervasive
 - > Easier to switch framework vs. whole architecture
 - > Lower barrier to entry
- Broader, more scalable database support
- Integration with Java technology libs, legacy services

JRuby on Rails: Java EE Platform

- Pool database connections
- Access any Java Naming and Directory Interface (J.N.D.I) API resource
- Access any Java EE platform:
 - > Java Persistence API (JPA)
 - > Java Management Extensions (JMX)
 - > Enterprise JavaBeans (EJB)
 - > Java Message Service (JMS) API
 - > SOAP/WSDL/SOA

JRuby On Rails: Deployment

- Mongrel
- Warfile deployment to
 - > GlassFish v2 +
 - > Servlet-based app server

Agenda

- Introduction of Ruby
- Introduction of JRuby
- Get on Rails
- JRuby on Rails
- **Summary**

Summary

- Ruby is elegant programming language
- Ruby on Rails makes web development really easy
- JRuby is Ruby, which leverages Java platforms
- Jruby: www.jruby.org
- Ruby: www.ruby-lang.org
- Rails: www.rubyonrails.org
- Netbeans IDE: www.netbeans.org
-



Q&A Thank You!

Joey Shen
Technology Evangelist
Sun Microsystems Inc.

