

ThoughtWorks®

JRuby and luke

On Google AppEngine



Ola Bini

ola.bini@gmail.com
<http://olabini.com/blog>

Vanity slide

ThoughtWorks consultant/developer/programming language geek

JRuby Core Developer

From Stockholm, Sweden

Programming language nerd (Lisp, Ruby, Java, Smalltalk, Io, Erlang, ML, C/C++/C#, etc)

Java +12 years

Ruby +5 years, JRuby +3 years

Author of Practical JRuby on Rails (APress)

Creator of the loke language

ThoughtWorks

Global consulting firm

US, Canada, UK, Australia, India, China - and Sweden

1100 people worldwide

40% of 2007 US revenue from Ruby work

Agile

Open Source

Martin Fowler



jruby

What is Ruby?

Dynamic, strongly typed, pure object oriented language

Interpreted

Open Source

Default implementation in C (called MRI)

Current versions: 1.8.6, 1.8.7 and 1.9.1

Created in 1993 by Yukihiro 'Matz' Matsumoto

“More powerful than Perl and more object oriented than Python”

“The principle of least surprise”

Ruby features

Object orientation - everything is an object

Literals

Blocks

Modules - as namespaces

Modules - as mixins - Enumerable - Comparable

Metaprogramming

Ruby applications/libraries

RubyGems

Rake

RSpec

Rails/Merb

Ramaze

Sinatra

What is JRuby

Implementation of the Ruby language

Java 1.5+

Open Source

Compatible with Ruby 1.8.6 p114

Current version 1.2

Why JRuby?

Threading

Unicode

Performance

Memory

C Extensions

Libraries

Politics

Legacy systems

Getting started

Install Java

Download <http://dist.codehaus.org/jruby/jruby-bin-1.2.0.tar.gz>

This includes JRuby, Ruby stdlib, RubyGems and rake

Unpack

Multiple copies are fine

Add `<jruby-dir>/bin` to PATH

Install gems

```
gem install rspec
```

```
jruby -S gem install rspec
```

Calling Ruby from Java

```
// One-time load Ruby runtime
ScriptEngineManager factory = new ScriptEngineManager();

ScriptEngine engine = factory.getEngineByName("jruby");

// Evaluate JRuby code from string.
try {
    engine.eval("puts('Hello')");
} catch (ScriptException exception) {
    exception.printStackTrace();
}
```

Java Integration

Java types == Ruby types

Call methods, construct instances, pass objects around

camelCase or snake_case both valid

Interfaces can be implemented

Classes can be inherited from

Implicit closure conversion

Extra added features to Rubyfy Java classes and Interfaces

Demo

Java Integration



rails

Rails

Opinionated software

Convention over configuration

Optimized for programmer productivity

Open Source

Rich community, plugins, extensions...

Rails + JRuby

Easy setup

Excellent performance

Thousands of libraries

Easy deployment

Wide database support

Platform independence

JRuby on Rails on AS/400, calling DB2.

Less political resistance in Java orgs

JRuby compatibility

Most compatible alternative implementation

Native threads

vs green threads

Native (C) Ruby extensions not supported

No continuations

No fork (daemonize, etc.)

ObjectSpace disabled by default

Warbler

Gem for packaging Rails apps in a WAR file

Self-contained

Includes JRuby and servlet adapter

```
jruby -S gem install warbler
```

```
jruby -S warble config
```

```
jruby -S warble
```


Rack

A well defined interface for dispatching web applications

Supported by basically all Ruby web frameworks

JRuby-rack

- A RackServlet

- Pooling of JRuby runtimes

- Handles dispatch to the real application

Reggae

Ruby Enabled Google App Engine

Provides simplified support for

- Configuration

- Gem-to-Jar bundler

- Simplify so that Rack applications run on local dev server

- Simplifies deployment

It is not completely done yet...

DataMapper

Ruby ORM tool

Has an adapter for DataSource

`dm-datastore-adapter`

Supports simple queries through DataMapper interface

BeeU

Extremely simple integration with UserService

To have person and admin status automatically set

Add to application.rb:

```
include BeeU
before_filter :assign_user
before_filter :assign_admin_status
```

To protect an action, add to the controller in question:

```
require_admin :remove
```

Demo

A JRuby on Rails GAE
application

Sinatra

Sinatra is a small DSL for writing Ruby web applications

The simplest app looks like this:

```
require 'rubygems'  
require 'sinatra'  
get '/' do  
  'Hello world!'  
end
```

Sinatra works fine with JRuby-rack on Google App Engine

Ramaze

A modular web framework

Runs under Rack

Works with DataMapper

Keep it simple

Needs a simple patch to get around thread usage

Limitations/things to note

File limits

- < 1 000 files

- < 10mb jar-files

Image APIs - no javax.image

- There is something called image-gae to get around this

No threads

No writing to files

Some reflection is turned off

Startup time

Testing

Local dev server

Doesn't duplicate the GAE environment well enough. =(

Local testing is hard

Unit testing possible, but need more extensive functional tests

You need to run the functional tests on GAE/J to be sure

So, either build in testing triggers in your app

Or use something like Selenium

And deploy to a specific "test" version, that will never be the default version

ioke





What is Ioke?

An experiment

A programming language

Dynamic and strong typing

Prototype based object orientation

Homoiconic

Inspirations: Io, Ruby, Self/Smalltalk, Lisp

Hosted on the JVM

Current versions: Ioke E ikj 0.3.1, Ioke E ikc 0.1.1



Demo

loke in action



IKanServe

Ioke equivalent of Rack

Small IokeServlet that dispatches to Ioke code

You include a file called `iks_application.ik`

Define your actions there

Right now, very Sinatra like - but not fully featured at all



HTML Builder

```
use("blank_slate")
```

```
IKanServe HtmlBuilder = BlankSlate create(fn(bs,  
  bs pass = method(+args, +:attrs,  
    args "<%s%:[ %s=\"%s\"%]>%[%s%]</%s>\n" format(  
      currentMessage name, attrs, args, currentMessage name))))
```

```
h = IKanServe HtmlBuilder  
title = "a title"  
h html(  
  h head(h title(title)),  
    h body(  
      h h1(  
        style: "font-size: 1.5em",  
        "Isn't this nice?"  
      )  
    )  
  )  
)
```



Demo

loke on GAE/J

Alternative languages on GAE

Automatic scaling

Harder to test

Security restrictions

Reflection

Threads

File system

But they all work well

Conclusions

GAE/J is very well suited for alternative languages

Ruby web frameworks

Ruby + DataStore vs JDO/JPA + DataStore

GAE/J enables polyglot programming

And thinking service oriented

It is easy to get started

The platform is very powerful

Q

and

A



Q



A



>

<