

MacRuby on Rails

～MacRubyから見たcRuby～

概要

2011年3月時点でのMacRubyはRuby処理系としての完成度は低く、Ruby on Railsが到底動作するようには思えない。しかしながら、MacRubyに対して改良に改良を重ねることできつとRailsを動作させることができる（はずである）。本発表ではMacRubyでRailsを動作させるために必要だった修正内容とともに、その過程であらためて知ることになったcRubyのすばらしさを説明します。

Abstract

MacRuby is an implementation of Ruby 1.9 that is directly on top of Mac OS X core technologies. Recently, MacRuby has become viable as a tool for developing useful desktop applications for Mac OS X. However, as of March 2011, MacRuby is still missing some functionality that is present in cRuby. Therefore, MacRuby is not able to run Ruby on Rails. In my presentation, I will explain how I modified MacRuby to make it a suitable foundation for running Rails. I would also like to explain some of technical intricacies that I discovered along the way.



Ruby会議2011 – 2011年7月17日
ネットワーク応用通信研究所
高尾宏治<kouji@netlab.jp>



高尾宏治

- 株式会社ネットワーク応用通信研究所
- cRubyのコミッタ (Readlineモジュール)
- MacRubyのコミッタ
- @takaokouji(@takaokouji_en)
- <http://facebook.com/takaokouji>

提供

ホーム ニュース オープンソース 業務内容 製品・サービス 会社情報 採用情報

株式会社ネットワーク応用通信研究所(NaCl)は、Rubyをはじめとするオープンソースソフトウェアを活用し、お客様にさまざまなソリューションを提供します。

・求人情報

2012年春の新規卒業者の追加募集

2012年春の新規卒業者の追加募集（第三期募集）を開始しました。第三期募集の締め切りは2011年8月31日(水)必着です。募集要件の詳細は[2012年新卒者向け採用情報](#)をご覧ください。

・講演・セミナー、メディア掲載、受賞に関する情報

NaCl及びNaClフェロー まつもとゆきひろ が関係する[「講演・セミナーの情報」](#)、[「メディア掲載情報」](#)、[「受賞に関する情報」](#)をご紹介します。

・新着情報

製品

- ▶ Ruby & Railsトレーニングプログラム

サービス

- ▶ 日医標準レセプトソフト導入
- ▶ RubyやRailsの企業向け講習
- ▶ 「Ruby + MySQL」コンサルティング

Ruby関連サイト

- ▶ Ruby Association

提供

ホーム ニュース オープンソース 業務内容 製品・サービス 会社情報 採用情報

株式会社ネットワーク応用通信研究所(NaCl)は、Rubyをはじめとするオープンソースソフトウェアを用いた技術研究を行なっています。

● ネットワーク応用通信研究所

・求人情報

2012年春の新規卒業者の追加募集

2012年春の新規卒業者の追加募集（第三期募集）を開始しました。第三期募集の締め切りは2011年8月31日(水)必着です。募集要件の詳細は[2012年新卒者向け採用情報](#)をご覧ください。

・講演・セミナー、メディア掲載、受賞に関する情報

NaCl及びNaClフェロー まつもとゆきひろ が関係する[「講演・セミナーの情報」](#)、[「メディア掲載情報」](#)、[「受賞に関する情報」](#)をご紹介します。

・新着情報

製品

- Ruby & Railsトレーニングプログラム

サービス

- 日医標準レセプトソフト導入
- RubyやRailsの企業向け講習
- 「Ruby + MySQL」コンサルティング

Ruby関連サイト

- Ruby Association

提供

ホーム ニュース オープンソース 業務内容 製品・サービス 会社情報 採用情報

株式会社ネットワーク応用通信研究所(NaCl)は、Rubyをはじめとするオープンソースソフトウェアを用いた技術研究を行なっています。

● **ネットワーク応用通信研究所**

・ [求人情報](#) **Network Applied Communicaiton Laboratory Co.,Ltd.**

2012年春の新規卒業者の追加募集

2012年春の新規卒業者の追加募集（第三期募集）を開始しました。第三期募集の締め切りは2011年8月31日(水)必着です。募集要件の詳細は[2012年新卒者向け採用情報](#)をご覧ください。

・ [講演・セミナー、メディア掲載、受賞に関する情報](#)

NaCl及びNaClフェロー まつもとゆきひろ が関係する[「講演・セミナーの情報」](#)、[「メディア掲載情報」](#)、[「受賞に関する情報」](#)をご紹介します。

・ [新着情報](#)

製品

- ▶ Ruby & Railsトレーニングプログラム

サービス

- ▶ 日医標準レセプトソフト導入
- ▶ RubyやRailsの企業向け講習
- ▶ 「Ruby + MySQL」コンサルティング

Ruby関連サイト

- ▶  [Ruby Association](#)

提供

ホーム ニュース オープンソース 業務内容 製品・サービス 会社情報 採用情報

株式会社ネットワーク応用通信研究所(NaCl)は、Rubyをはじめとするオープンソースソフトウェアを用いた技術研究を行なっています。

- **ネットワーク応用通信研究所**

・ 求人情報 Network Applied Communicaiton Laboratory Co.,Ltd.

● **自分を幸せにしてください。**

2012年春の新規卒業者の追加募集（第三期募集）を開始しました。第三期募集の締め切りは2011年8月31日(水)必着です。募集要件の詳細は2012年新卒者向け採用情報をご覧ください。

・ 講演・セミナー、メディア掲載、受賞に関する情報

NaCl及びNaClフェロー まつもとゆきひろ が関係する「講演・セミナーの情報」、「メディア掲載情報」、「受賞に関する情報」をご紹介します。

・ 新着情報

製品

- ▶ Ruby & Railsトレーニングプログラム

サービス

- ▶ 日医標準レセプトソフト導入
- ▶ RubyやRailsの企業向け講習
- ▶ 「Ruby + MySQL」コンサルティング

Ruby関連サイト

- ▶ Ruby Association

提供

ホーム ニュース オープンソース 業務内容 製品・サービス 会社情報 採用情報

株式会社ネットワーク応用通信研究所(NaCl)は、Rubyをはじめとするオープンソースソフトウェアを用いた技術研究を行なっています。

● **ネットワーク応用通信研究所**
リューションを提供します。

・ 求人情報 Network Applied Communicaiton Laboratory Co.,Ltd.

2012年春の新規卒業者の追加募集（第三期募集）を開始しました。第三期募集の締め切りは2011年8月31日(水)必着です。募集要件の詳細は2012年卒業者向け採用情報をご覧ください。

● **自分を幸せにしてください。**

2012年春の新規卒業者の追加募集（第三期募集）を開始しました。第三期募集の締め切りは2011年8月31日(水)必着です。募集要件の詳細は2012年卒業者向け採用情報をご覧ください。

● **家族を幸せにしてください。**

・ 講演・セミナー、メディア掲載、受賞に関する情報

NaCl及びNaClフェロー まつもとゆきひろ が関係する「講演・セミナーの情報」、「メディア掲載情報」、「受賞に関する情報」をご紹介します。

・ 新着情報

製品

- Ruby & Railsトレーニングプログラム

サービス

- 日医標準レセプトソフト導入
- RubyやRailsの企業向け講習
- 「Ruby + MySQL」コンサルティング

Ruby関連サイト

- Ruby Association

提供

ホーム ニュース オープンソース 業務内容 製品・サービス 会社情報 採用情報

株式会社ネットワーク応用通信研究所(NaCl)は、Rubyをはじめとするオープンソースソフトウェアを用いたシステム開発を行なっています。

● **ネットワーク応用通信研究所**

・ 求人情報 Network Applied Communicaiton Laboratory Co.,Ltd.

・ **自分を幸せにしてください。**

2012年春の新規卒業者の追加募集（第三期募集）を開始しました。第三期募集の締め切りは2011年8月31日(水)必着です。募集要件の詳細は2012年卒業者向け採用情報をご覧ください。

・ 講演・セミナー、メディア掲載、受賞に関する情報

余った分で会社を幸せにしてください。

NaCl及びNaClフェロー まつもとゆきひろ が関係する「講演・セミナーの情報」、「メディア掲載情報」、「受賞に関する情報」をご紹介します。

・ 新着情報

製品

- Ruby & Railsト
レーニングプログラ
ム

サービス

- 日医標準レセプトソ
フト導入
- Ruby+Railsの企業
向け講習
- 「P

家族

自分



RubyKaigi2011

MacRuby on Rails

～MacRubyから見たcRuby～

Ruby会議2011 – 2011年7月17日
ネットワーク応用通信研究所
高尾宏治<kouji@netlab.jp>



RubyKaigi2011



MacRuby非公式イメージキャラクター
R u b y 忍 者 @ l r z

MacRuby on Rails

～MacRubyから見たcRuby～

Ruby会議2011 – 2011年7月17日
ネットワーク応用通信研究所
高尾宏治<kouji@netlab.jp>

MacRuby

Mac OS Xに特化したRuby



MacRuby



MacRuby

MacRuby is a unique blend of Ruby 1.9 and Objective-C.

→ Ruby 1.9とObjective-Cを融合させたもの



MacRuby

MacRuby is a unique blend of Ruby 1.9 and Objective-C.

→ Ruby 1.9とObjective-Cを融合させたもの

The goal of the MacRuby project is to be 100% compatible syntactically and behaviorally with Ruby 1.9.

→ Ruby 1.9との100%互換を目指す



Ruby 1.9

アプリケーション

Ruby 1.9

Ruby VM

組み込み
GC

組み込み
ライブラリ

添付
ライブラリ

ライブラリ

OS
(Linux,Windows,Mac OS X,etc...)

MacRuby

アプリケーション

MacRuby

Ruby VM

組み込み
GC

組み込み
ライブラリ

添付
ライブラリ

ライブラリ

Mac OS X

MacRuby



MacRuby



MacRuby



Foundation

ライブラリ

組み込み

添付
ライブラリ

組み
GC

MacRuby

ケーション

Mac OS X

AutoZone

MacRuby

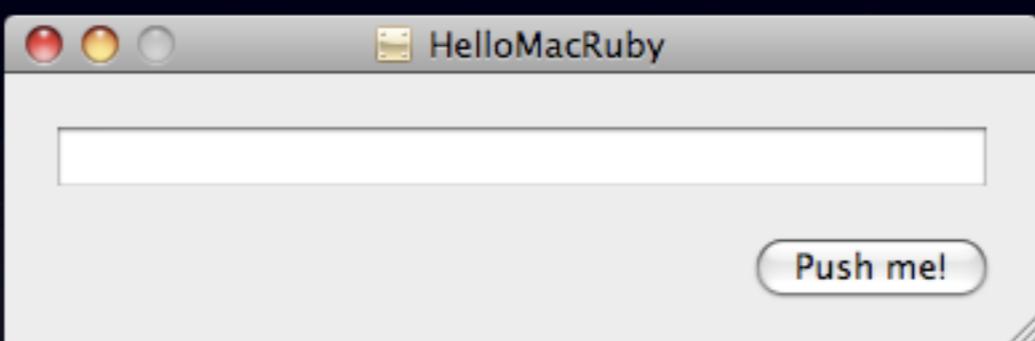


MacRuby×Cocoa

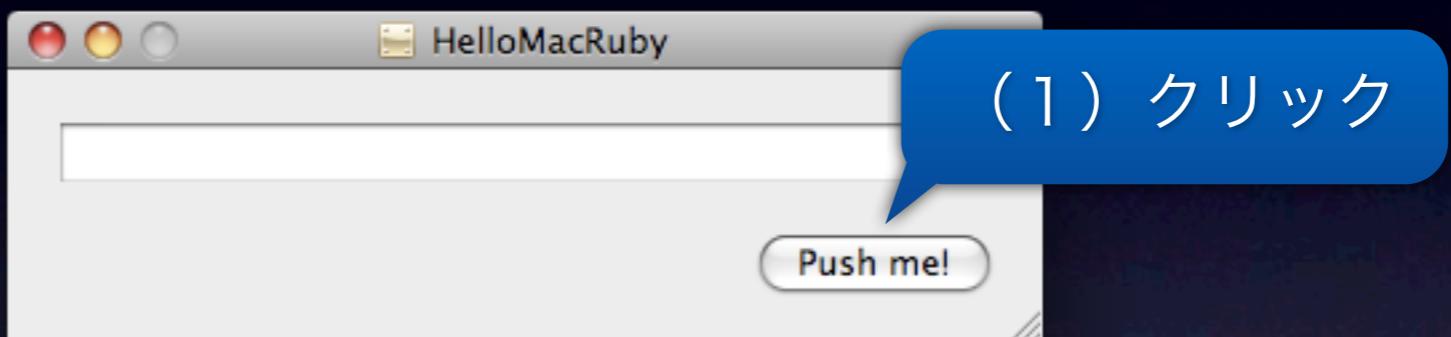
framework “cocoa”

```
def playSound(name)
    sound = NSSound.soundNamed(name)
    sound.play
end
```

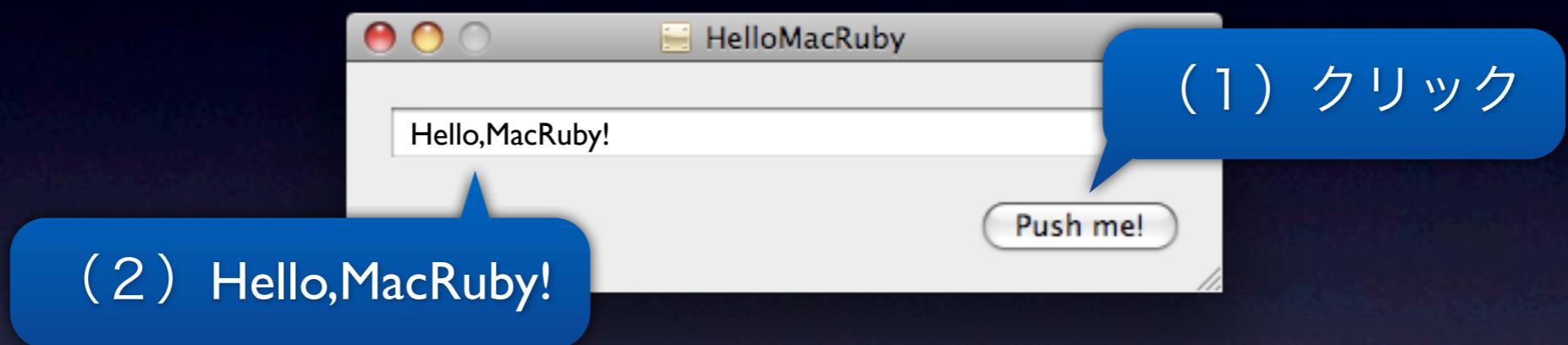
HelloMacRuby



HelloMacRuby



HelloMacRuby

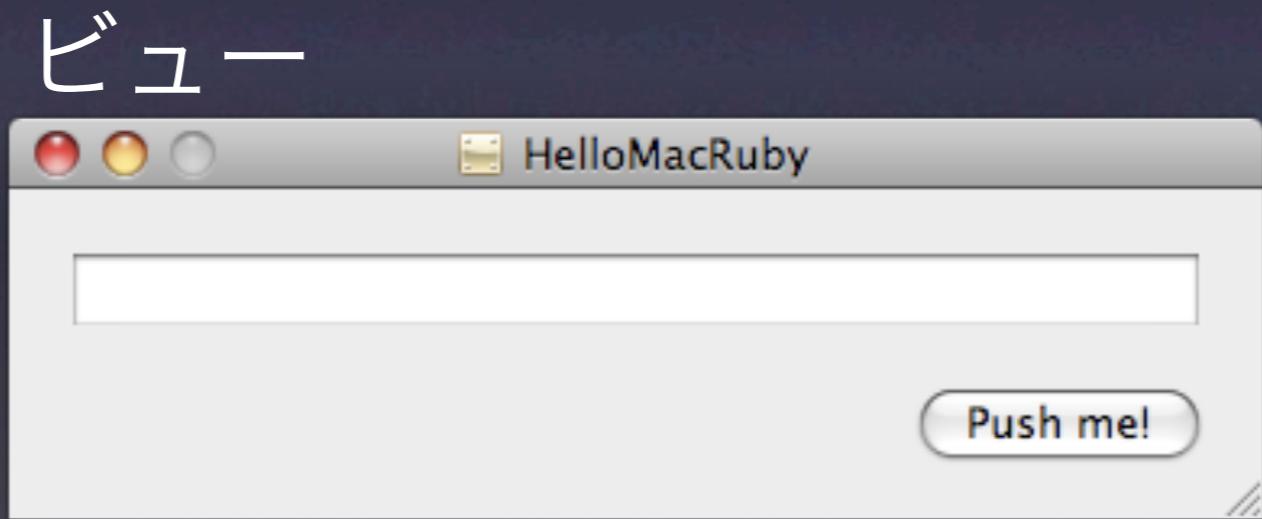


HelloMacRuby

Rubyスクリプト：コントローラ

```
class AppController
  attr_accessor :textField

  def hello(sender)
    @textField.stringValue = "Hello,MacRuby!"
  end
end
```

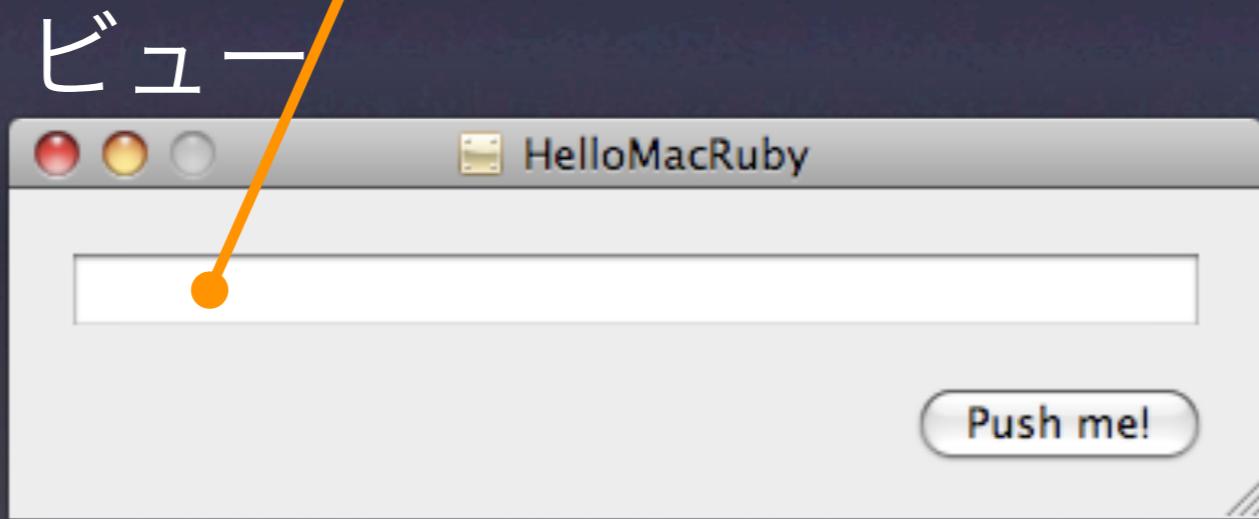


HelloMacRuby

Rubyスクリプト：コントローラ

```
class AppController
  attr_accessor :textField

  def hello(sender)
    @textField.stringValue = "Hello,MacRuby!"
  end
end
```

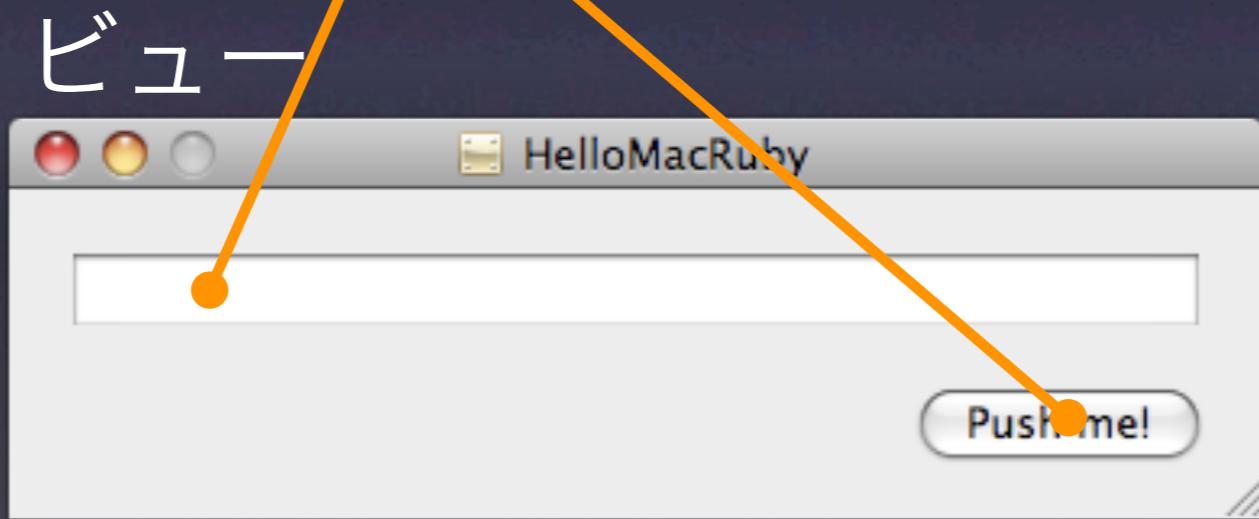


HelloMacRuby

Rubyスクリプト : コントローラ

```
class AppController
  attr_accessor :textField

  def hello(sender)
    @textField.stringValue = "Hello,MacRuby!"
  end
end
```



MacRuby on Rails

MacRubyでRuby on Railsを動作させる



MacRuby on Rails



MacRuby on Rails

```
$ rails new demo
kouji:/Users/kouji/work/MacRuby/MacRuby/issues/work:0
dhcp219$ env VM_DISABLE_RBO=1 macruby -S rails new demo
  create
  create README
  create Rakefile
  create config.ru
  create .gitignore
  create Gemfile
  create app
  create app/controllers/application_controller.rb
  create app/helpers/application_helper.rb
  create app/mailers
  create app/models
  create app/views/layouts/application.html.erb
  create config
  create config/routes.rb
  create config/application.rb
  create config/environment.rb
  create config/environments
  create config/environments/development.rb
  create config/environments/production.rb
  create config/environments/test.rb
  create config/initializers
  create config/initializers/backtrace_silencers.rb
```



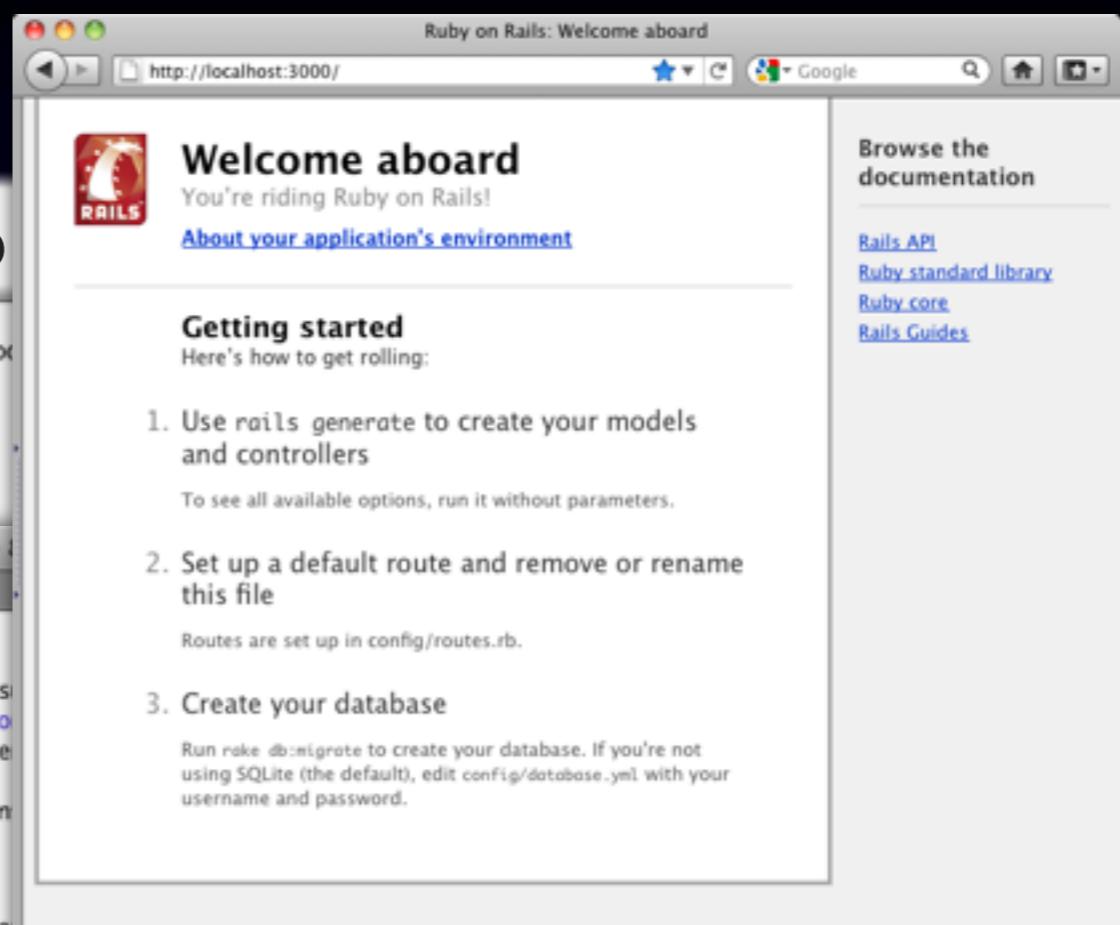
MacRuby on Rails



MacRuby on Rails



MacRuby on Rails



4ヶ月前...



4ヶ月前...

2011年3月中旬



4ヶ月前...

2011年3月中旬

- MacRubyでRailsは動くの？



4ヶ月前...

2011年3月中旬

- MacRubyでRailsは動くの？
- Railsのインストールはできる

```
$ sudo macgem install --verbose rails --no-ri --no-rdoc  
7 gems installed
```

```
$ sudo macgem install --verbose sqlite3-ruby --no-ri --no-rdoc  
2 gems installed
```



もしかして...



もしかして…

- すでにRailsって動作する！？



もしかして…

- すでにRailsって動作する！？
- プロジェクトが作成できた！

```
$ macruby -S rails new demo  
  create  
  create README  
  create Rakefile  
  
...  
  create vendor/plugins  
  create vendor/plugins/.gitkeep
```



まさか！？



まさか！？

- scaffoldは...



まさか！？

- scaffoldは...
- abort ! ?やっぱりだめじゃん

```
$ macruby -S rails generate scaffold Bookmark title:string  
description:text url:string  
...  
Assertion failed: ((size_t)pos < current_exceptions.size()),  
function pop_current_exception, file vm.cpp, line 3434.  
zsh: abort    macruby -S rails generate scaffold Bookmark  
title:string description:text
```



まさか！？

- scaffoldは...
- abort ! ?やっぱりだめじゃん

```
$ macruby -S rails generate scaffold Bookmark title:string  
des...  
$ macruby -S rails server  
...  
Ass...  
func...  
zsh:...  
title
```

Assertion failed: ((size_t)pos < current_exceptions.size()),
function pop_current_exception, file vm.cpp, line 3434.
zsh: abort macruby -S rails server



イテレーション

Railsの実行

SEGV・abort

原因調査

コーディング

RubySpec

Railsの実行



イテレーション

Railsの実行

SEGV・abort

原因調査

コーディング

RubySpec

Railsの実行

必要な知識や技術

- cRuby
- Objective-C
- C++
- LLVM



#860 catch/throw

Assertion fails in pop_current_exception
when catch/throw used in rescue

➡ rescue節の中でcatchとthrowを行うとabortしてしまう



#860 catch/throw

Assertion fails in pop_current_exception
when catch/throw used in rescue

→ rescue節の中でcatchとthrowを行うとabortしてしまう

```
begin
  raise
rescue
  catch(:ambiguous) { throw :ambiguous }
  p :ok
end
```



#860 catch/throw

Assertion fails in pop_current_exception
when catch/throw used in rescue

➡ rescue節の中でcatchとthrowを行うとabortしてしまう

```
begin
  raise
rescue
  catch(:ambiguous) { throw :ambiguous }
  p :ok
end
```

```
:ok
Assertion failed: ((size_t)pos < current_exceptions.size()),
function pop_current_exception, file vm.cpp, line 3448.
```



#860 catch/throw

Assertion fails in pop_current_exception
when catch/throw used in rescue

→ rescue節の中でcatchとthrowを行うとabortしてしまう

```
begin
  raise
rescue
  catch(:ambiguous) { throw :ambiguous }
  p :ok
end
```

throwで例外をスタックから
取り除いてしまっていた

:ok

Assertion failed: ((size_t)pos < current_exceptions.size()),
function pop_current_exception, file vm.cpp, line 3448.



#860 catch/throw

Assertion fails in pop_current_exception
when catch/throw used in rescue

→ rescue節の中でcatchとthrowを行うとabortしてしまう

```
begin
  raise
rescue
  catch(:ambiguous) { throw :ambiguous }
  p :ok
end
```

throwで例外をスタックから
取り除いてしまっていた

:ok

ここでも例外をスタックから
取り除こうとしてabort

Assertion failed: (assertions.size() == exceptions.size()),
function pop_current_exception, file vm.cpp, line 3448.



#860 catch/throw

修正前(60723bf~)だとthrow時に例外があれば必ずスタックから取り除いてしまう



#860 catch/throw

修正前(60723bf~)だとthrow時に例外があれば必ずスタックから取り除いてしまう

```
catch(:ambiguous) {  
  begin  
    raise  
  rescue  
    throw :ambiguou  
  end  
}
```



#860 catch/throw

修正前(60723bf~)だとthrow時に例外があれば必ずスタックから取り除いてしまう

```
catch(:ambiguous) {  
  begin  
    raise  
  rescue  
    throw :ambiguou  
  end  
}
```

ここには到達しないためendで例外を取り除くだけではダメ



#860 catch/throw

修正前(60723bf~)だとthrow時に例外があれば必ずスタックから取り除いてしまう

```
catch(:ambiguous) {  
  begin  
    raise  
  rescue  
    throw :ambiguou  
  end  
}
```

throwで例外をスタックから取り除かないといけない

ここには到達しないためendで例外を取り除くだけではダメ



#860 catch/throw

修正前(60723bf~)だとthrow時に例外があれば必ずスタックから取り除いてしまう

```
vm.cpp:4321:  
VALUE  
RoxorVM::ruby_throw(VALUE tag, VALUE value)  
{  
...  
    // We must pop the current VM exception in case we are in a rescue handler,  
    // since we are going to unwind the stack.  
    if (current_exception() != Qnil) {  
        pop_current_exception();  
    }  
...  
}
```



#860 catch/throw

→ 必要なときだけ例外を取り除くように
修正(60723bf)



#860 catch/throw

→ 必要なときだけ例外を取り除くように
修正(60723bf)

```
vm.cpp:4318:  
VALUE  
RoxorVM::ruby_throw(VALUE tag, VALUE value)  
{  
...  
    rb_vm_catch_t *catch_ptr = iter->second;  
    while (catch_ptr->current_exception != current_exception()) {  
        pop_current_exception();  
    }  
...}
```



#860 catch/throw

→ 必要なときだけ例外を取り除くように

修正(60722bf)

vm.cpp:4318:

VALUE

RoxorVM::ruby_

{

...

rb_vm_catch_t *catch_ptr = iter->second;

while (catch_ptr->current_exception != current_exception()) {

 pop_current_exception();

}

...

catchを処理したときの例外(catch_ptr->current_exception)
と現在の例外(current_exception())が等しくなければ例外
をスタックから取り除く



#860 catch/throw

```
begin
  raise
rescue
  catch(:ambiguous) {
    begin
      raise
    rescue
      begin
        raise
      rescue
        throw :ambiguous
      end
    end
  }
  p :ok
end
```



#860 catch/throw

```
begin          例外Aが発生
 raise
rescue
catch(:ambiguous) {
  begin
    raise
  rescue
    begin
      raise
    rescue
      throw :ambiguous
    end
  end
}
p :ok
end
```



#860 catch/throw

```
begin          例外Aが発生
 raise
 rescue
 catch(:ambiguous) {
 begin
 raise
 rescue
 begin
 raise
 rescue
 throw :ambiguous
 end
 end
 }
 p :ok
end          例外Aを記録
catch_ptr->current_exception = current_exception();
```



#860 catch/throw

```
begin      例外Aが発生
 raise
rescue
catch(:ambiguous) {
begin      例外Bが発生
 raise
rescue
begin
 raise
rescue
 throw :ambiguous
end
end
}
p :ok
end
```

例外Aを記録
catch_ptr->current_exception = current_exception();



#860 catch/throw

```
begin      例外Aが発生
 raise
rescue
catch(:ambiguous) {
begin      例外Aを記録
 raise
      catch_ptr->current_exception = current_exception();
rescue
begin      例外Bが発生
 raise
rescue
begin      例外Cが発生
 raise
rescue
 throw :ambiguous
end
end
}
p :ok
end
```



#860 catch/throw

```
begin      例外Aが発生
raise
rescue
catch(:ambiguous) {
    begin      例外Aを記録
    raise
    rescue
        begin      例外Bが発生
        raise
        rescue
            begin      例外Cが発生
            raise
            rescue
                throw :ambiguous
            end
        end
    end
}
p :ok
end
```

例外Aを記録
catch_ptr->current_exception = current_exception();

例外C、B、Aの順に記録しておいた例外Aと比較して、例外CとBをスタックから取り除く



#860 catch/throw

```
begin      例外Aが発生
raise
rescue
catch(:ambiguous) {
    begin      例外Bが発生
        raise
    rescue
        begin      例外Cが発生
            raise
        rescue
            throw :ambiguous
        end
    end
}
P :ok      例外Aをスタックから取り除く
end
```

例外Aを記録
catch_ptr->current_exception = current_exception();

例外C、B、Aの順に記録しておいた例外Aと比較して、例外CとBをスタックから取り除く



#1192 定数探索

Did not find nested constants.

→ ネストした定数の探索に失敗する



#1192 定数探索

Did not find nested constants.

→ ネストした定数の探索に失敗する

```
module A
  B = 10
  Object.class_eval { B }
end
```



#1192 定数探索

Did not find nested constants.

→ ネストした定数の探索に失敗する

```
module A
  B = 10
  Object.class_eval { B }
end
```

```
reduction.rb:3:in `block': uninitialized constant B (NameError)
from reduction.rb:1:in `<main>'
```



#1192 定数探索

Did not find nested constants.

→ ネストした定数の探索に失敗する

```
module A
  B = 10
  Object.class_eval { B }
end
```

Objectから定数Bを探索して
いるため失敗する

```
reduction.rb:3:in `block': uninitialized constant B (NameError)
from reduction.rb:1:in `<main>'
```



#1192 定数探索

修正前 (6b101bd~) だとレキシカルな定数探索が実装されていなかった。



#1192 定数探索

修正前 (6b10lbd~) だとレキシカルな定数探索が実装されていなかった。

```
module A
  module B
    CONST = "B's Const"
    module ::A
      p CONST
    end
  end
end
```



#1192 定数探索

修正前 (6b101bd~) だとレキシカルな定数探索が実装されていなかった。

```
module A
  module B
    CONST = "B's Const"
    module ::A
      p CONST
    end
  end
end
```

-e:5:in `block': uninitialized constant A::CONST (NameError)



#1192 定数探索

修正前 (6b101bd~) だとレキシカルな定数探索が実装されていなかった。

```
module A
  module B
    CONST = "B's Const"
    module ::A
      p CONST
    end
  end
end
```

::A、A::B、Aの順に定数探索する必要があるが、::Aしか探索していない。

-e:5:in `block': uninitialized constant A::CONST (NameError)



定数探索むずい

module_eval(文字列)

```
module A
  CONST = "A's CONST"
  def f
    class_eval <<-EOS
      p CONST
    EOS
  end
end
```

```
class K
  CONST = "K's CONST"
  extend A
  f
end
```

module_eval ブロック

```
module A
  CONST = "A's CONST"
  def f
    class_eval {
      p CONST
    }
  end
end
```

```
class K
  CONST = "K's CONST"
  extend A
  f
end
```



定数探索むずい

module_eval(文字列)

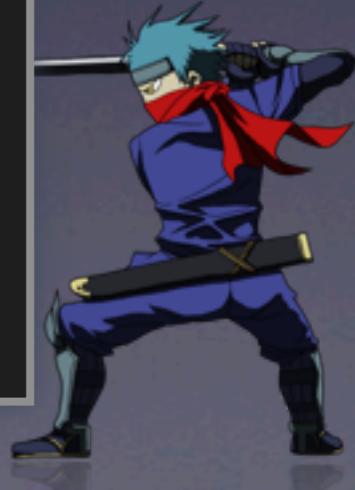
```
module A
  CONST = "A's CONST"
  def f
    class_eval <<-EOS
      p CONST
    EOS    #=> "K's CONST"
  end
end

class K
  CONST = "K's CONST"
  extend A
  f
end
```

module_eval ブロック

```
module A
  CONST = "A's CONST"
  def f
    class_eval {
      p CONST
    }
  end
end

class K
  CONST = "K's CONST"
  extend A
  f
end
```



定数探索むずい

module_eval(文字列)

```
module A
  CONST = "A's CONST"
  def f
    class_eval <<-EOS
      p CONST
    EOS      #=> "K's CONST"
  end
end

class K
  CONST = "K's CONST"
  extend A
  f
end
```

module_eval ブロック

```
module A
  CONST = "A's CONST"
  def f
    class_eval {
      p CONST
    }          #=> "A's CONST"
  end
end

class K
  CONST = "K's CONST"
  extend A
  f
end
```



定数探索むずい

module_eval(文字列)

```
module A
  CONST = "A's CONST"
  def f
    class_eval <<-EOS
      p CONST
    EOS      #=> "K's CONST"
  end
end
```

K、A、Objectの順に
定数CONSTを探索する。

```
class K
  CONST = "K's CONST"
  extend A
  f
end
```

module_eval ブロック

```
module A
  CONST = "A's CONST"
  def f
    class_eval {
      p CONST
    }          #=> "A's CONST"
  end
end
```

```
class K
  CONST = "K's CONST"
  extend A
  f
end
```



定数探索むずい

module_eval(文字列)

```
module A
  CONST = "A's CONST"
  def f
    class_eval <<-EOS
      p CONST
    EOS      #=> "K's CONST"
  end
end
```

K、A、Objectの順に
定数CONSTを探索する。

```
class K
  CONST = "K's CONST"
  extend A
  f
end
```

module_eval ブロック

```
module A
  CONST = "A's CONST"
  def f
    class_eval {
      p CONST
    }          #=> "A's CONST"
  end
end
```

A、Objectの順に
定数CONSTを探索する。
Kからは探索しない。

```
class K
  CONST = "K's CONST"
  extend A
  f
end
```



困難

かなり実装が難しい振る舞いを発見



困難

かなり実装が難しい振る舞いを発見

```
module A
  B = 42
end
```

```
A.class_eval do
  def self.f
    p B
  end
end

A.f
```



困難

かなり実装が難しい振る舞いを発見

```
module A
  B = 42
end
```

```
A.class_eval do
  def self.f
    p B
  end
end

A.f
```

```
Ruby 1.9
42
```



困難

かなり実装が難しい振る舞いを発見

```
module A  
  B = 42  
end
```

```
A.class_eval do  
  def self.f  
    p B  
  end  
end  
  
A.f
```

Ruby 1.9
42

Ruby 1.8.7
-e:8:in `f': uninitialized constant B (NameError)



困難

かなり実装が難しい

```
module A  
  B = 42  
end
```

```
end  
end
```

```
A.f
```

cRubyのbug

. uninitialized constant B (NameError)



#?

abort when I pressed “About your application’s environment” in welcome page.

➡ Welcome aboardのページにあるリンクをクリックする
とabortする



#?

abort when I pressed “About your application’s environment” in welcome page.

➡ Welcome aboardのページにあるリンクをクリックする
とabortする

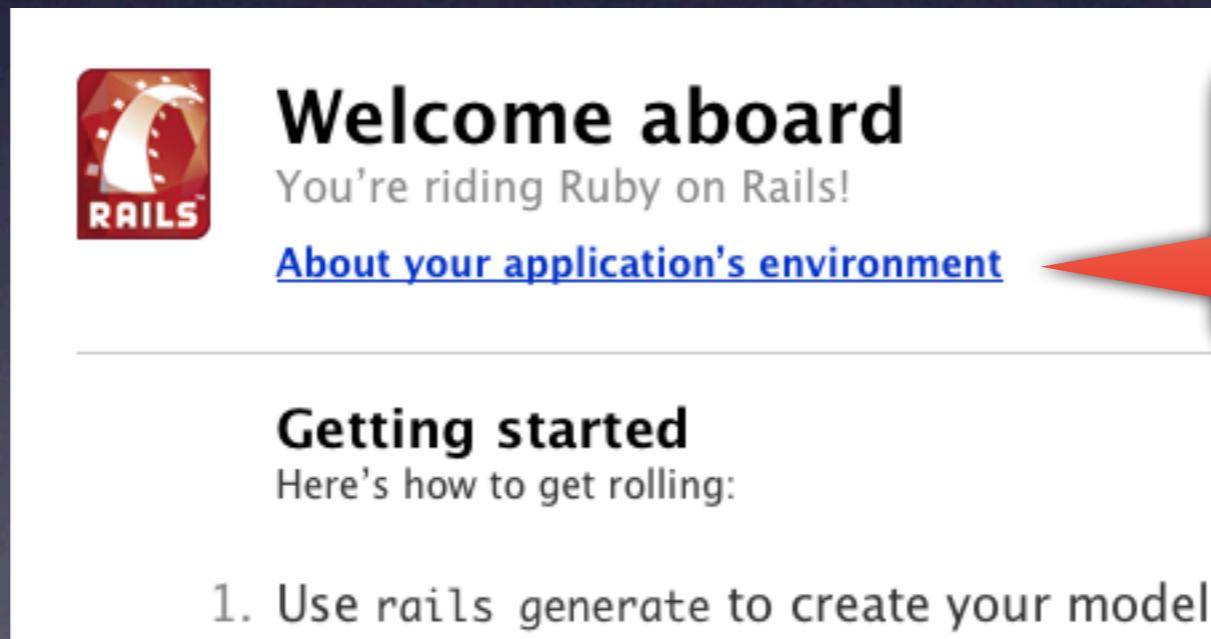
The screenshot shows a web page titled "Welcome aboard" with the subtitle "You're riding Ruby on Rails!". Below the title is a blue underlined link labeled "About your application's environment". A horizontal line separates this section from the "Getting started" section below. The "Getting started" section contains the text "Here's how to get rolling:" followed by a numbered list: "1. Use rails generate to create your model".



#?

abort when I pressed “About your application’s environment” in welcome page.

→ Welcome aboardのページにあるリンクをクリックする
とabortする



Welcome aboard
You're riding Ruby on Rails!
[About your application's environment](#)

Getting started
Here's how to get rolling:

1. Use rails generate to create your model

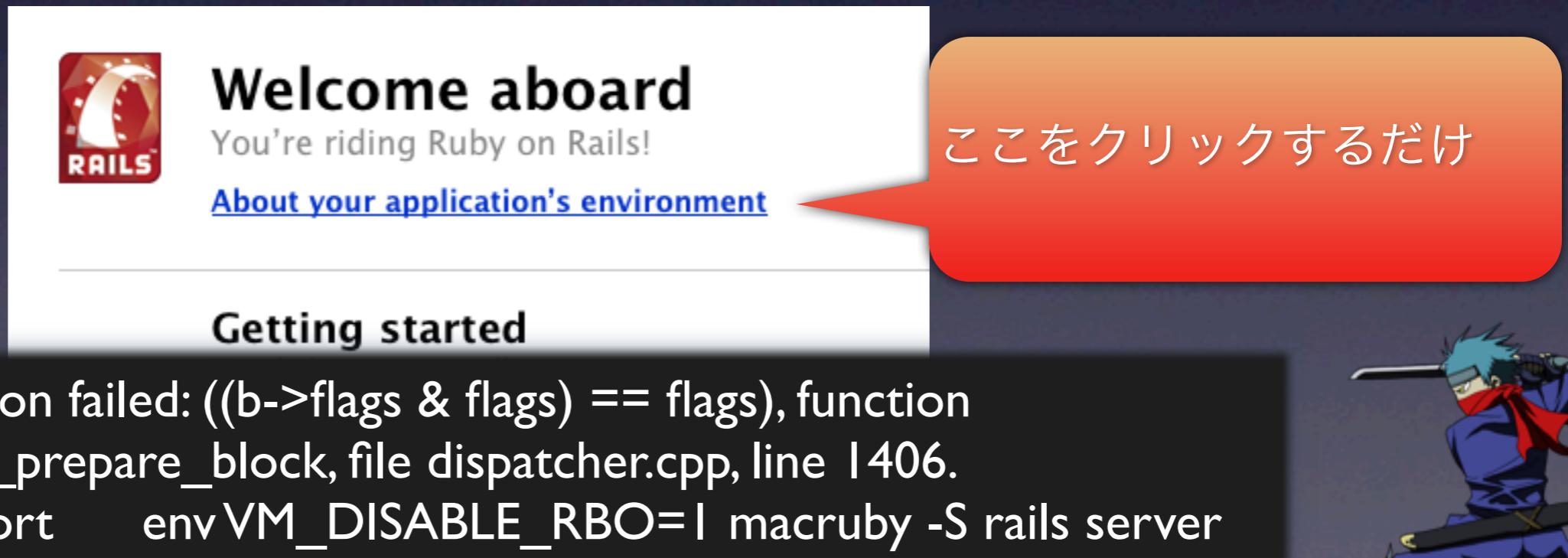
ここをクリックするだけ



#?

abort when I pressed “About your application’s environment” in welcome page.

→ Welcome aboardのページにあるリンクをクリックする
とabortする



それって…



それって…

MacRubyでRailsが動いていないのでは…



それって…

MacRubyでRailsが動いていないのでは…

→ すみません。

実はまだ動いていませんでした。



それって…

MacRubyでRailsが動いていないのでは…

- ➡ すみません。
実はまだ動いていませんでした。
- ➡ @lrz
「Railsが動作すればMacRuby 1.0をリリースする」



つづきはRubyConfで...



To be continued...

イラスト：諸星佑樹<morohoshi@netlab.jp>