

RubyOnRails

Jens Himmelreich

I. Was ist das?

II. Wie funktioniert das?

III. Wo funktioniert das nicht?

I.

Was ist das?

abstrakt

RubyOnRails

RubyOnRails

Ruby

Programmiersprache

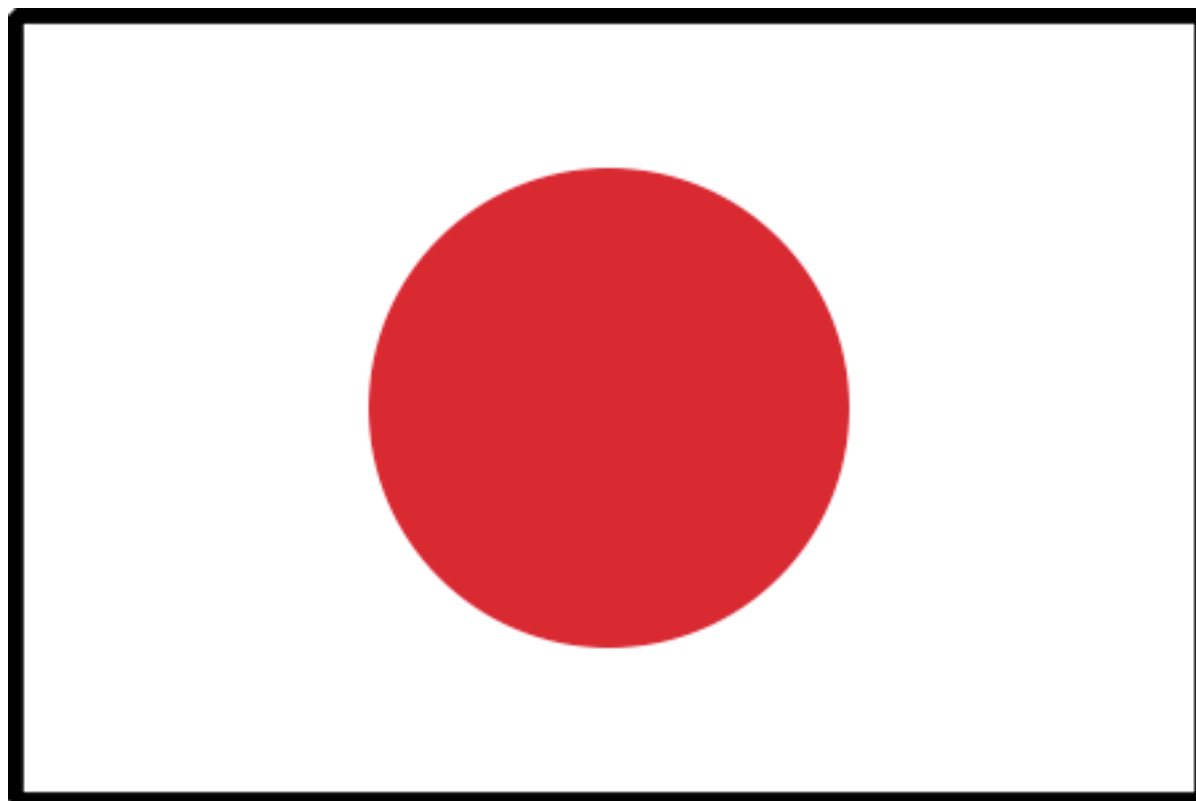


*** 24. 2. 1993**

geboren

Yukihiro
Matsumoto

Vater



Geburtsland

**interpretiert,
objektorientiert**

Perl
Smalltalk
Eiffel

RubyOn**Rails**

Rails

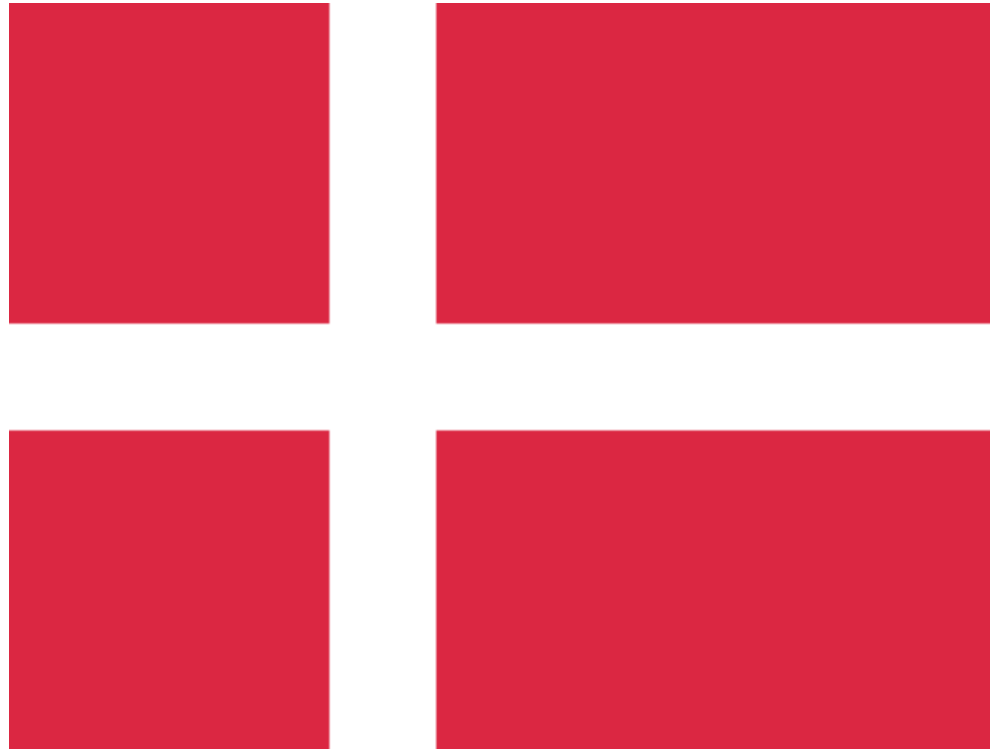
Webapplikationsframework



*** Juli 2004**
geboren

David
Heinemeyer
Hansson

Vater



Geburtsland

MVC

Model-View-Controller

ActiveRecord

,OR-Mapper‘

ActionPack

Controller/Template
Framework

Ruby**On**Rails

Schiene

Konvention

konkret

Hello World

```
$ rails hello_world
```

```
$ cd hello_world/
```

```
$ ls -1F
```

```
README
```

```
Rakefile
```

```
app/
```

```
components/
```

```
config/
```

```
db/
```

```
doc/
```

```
lib/
```

```
log/
```

```
public/
```

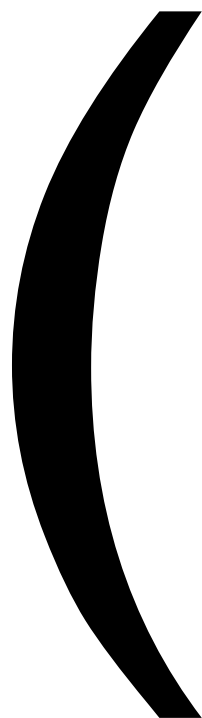
```
script/
```

```
test/
```

```
tmp/
```

```
vendor/
```

```
$
```



```
$ mysqladmin -u root create hello_world_development
```

```
$ ./script/generate migration CreateMessages  
  create db/migrate  
  create db/migrate/001_create_messages.rb  
$  
$ mate db/migrate/001_create_messages.rb
```

```
1 class CreateMessages < ActiveRecord::Migration
  def self.up
  end

  def self.down
  end
end
```

```
1 class CreateMessages < ActiveRecord::Migration
  def self.up
    create_table "messages" do |t|
      t.column :text, :string
    end
  end

  def self.down
    drop_table "messages"
  end
end
```

```
$ rake db:migrate
== CreateMessages: migrating
=====
-- create_table("messages")
   -> 0.0618s
== CreateMessages: migrated (0.0620s)
=====

$
```

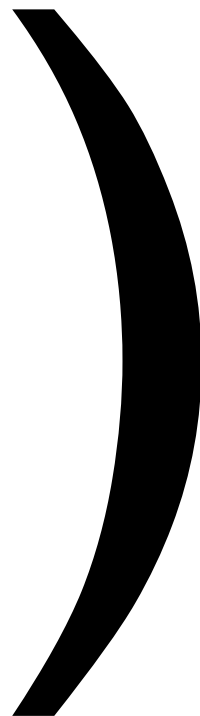
```
$ mysql -u root hello_world_development
```

```
mysql> desc messages;
```

Field	Type	Null	Key
id	int(11)	NO	PRI
text	varchar(255)	YES	

```
2 rows in set (0.07 sec)
```

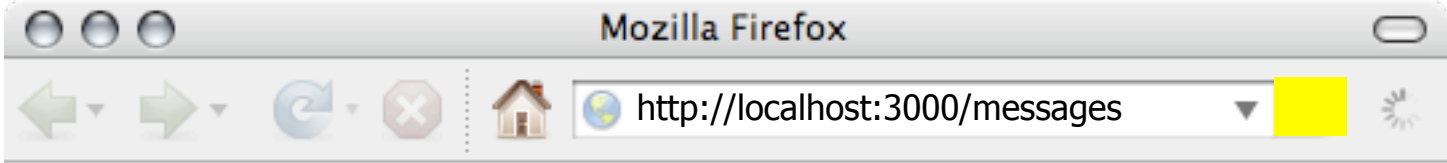
```
mysql>
```

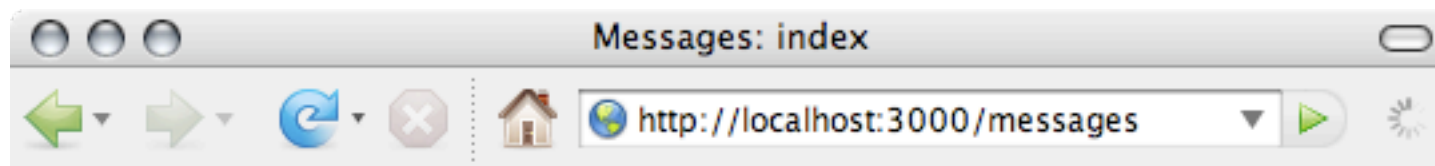


```
$ ./script/generate scaffold Message
create app/models/message.rb
create test/unit/message_test.rb
create test/fixtures/messages.yml
create app/views/messages/_form.rhtml
create app/views/messages/list.rhtml
create app/views/messages/show.rhtml
create app/views/messages/new.rhtml
create app/views/messages/edit.rhtml
create app/controllers/messages_controller.rb
create test/functional/messages_controller_test.rb
create app/helpers/messages_helper.rb
create app/views/layouts/messages.rhtml
create public/stylesheets/scaffold.css
```

```
$
```

```
$ ./script/server webrick
=> Booting WEBrick...
=> Rails application started on http://0.0.0.0:3000
=> Ctrl-C to shutdown server; call with --help for
options
[2007-05-07 10:23:25] INFO WEBrick 1.3.1
[2007-05-07 10:23:25] INFO ruby 1.8.5 (2006-08-25)
[i686-darwin8.7.1]
[2007-05-07 10:23:25] INFO
WEBrick::HTTPServer#start: pid=6136 port=3000
```





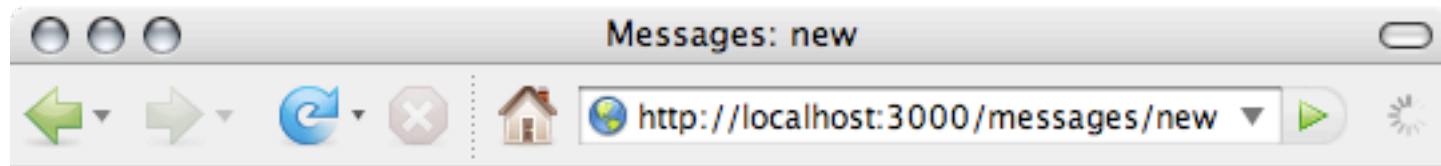
Listing messages

Text



Fertig





New message

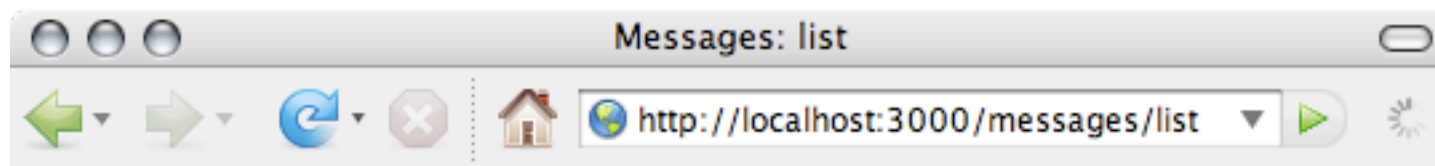
Text



Back

Fertig





Message was successfully created.

Listing messages

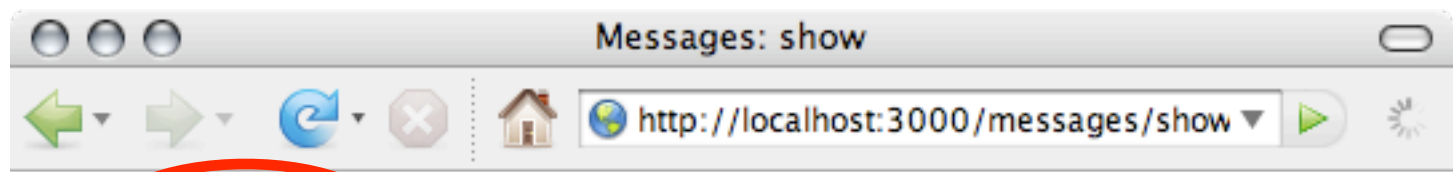
Text

Hello World [] [Edit](#) [Destroy](#)

[New message](#)

Fertig





Text: Hello World

Edit | Back

Fertig



```
$ rails hello_world
```

+

```
$ ./script/generate scaffold Message
```

=

```
Text: Hello World
```

II.

**Wie funktioniert
das?**

**Convention
over
Configuration**

Struktur

statisch

Verzeichnis- struktur

```
$ ls -1F
README
Rakefile
app/
components/
config/
db/
doc/
lib/
log/
public/
script/
test/
tmp/
vendor/
```

```
$
```

```
$ ls -1F app  
controllers/  
helpers/  
models/  
views/
```

```
$
```

```
$ ls -lF app/models/  
message.rb
```

```
$
```

```
$ ls -1F app/controllers/  
application.rb  
messages_controller.rb
```

```
$
```

```
$ ls -1F app/views/  
layouts/  
messages/
```

```
$ ls -1F app/views/messages/  
_form.rhtml  
edit.rhtml  
list.rhtml  
new.rhtml  
show.rhtml
```

```
$
```

```
$ ls -1F app/helpers/  
application_helper.rb  
messages_helper.rb
```

```
$
```

```
$ ls -1F
README
Rakefile
app/
components/
config/
db/
doc/
lib/
log/
public/
script/
test/
tmp/
vendor/
```

```
$
```

```
$ ls -1F config/  
boot.rb  
database.yml  
environment.rb  
environments/  
routes.rb
```

```
$ ls -1F config/environments/  
development.rb  
production.rb  
test.rb
```

```
$
```

```
$ ls -1F db  
migrate/  
schema.rb
```

```
$ ls -1F db/migrate/  
001_create_messages.rb
```

```
$
```

```
$ ls -1F
README
Rakefile
app/
components/
config/
db/
doc/
lib/
log/
public/
script/
test/
tmp/
vendor/
```

```
$
```

```
$ ls -1F test
```

```
fixtures/
```

```
functional/
```

```
integration/
```

```
mocks/
```

```
test_helper.rb
```

```
unit/
```

```
$
```

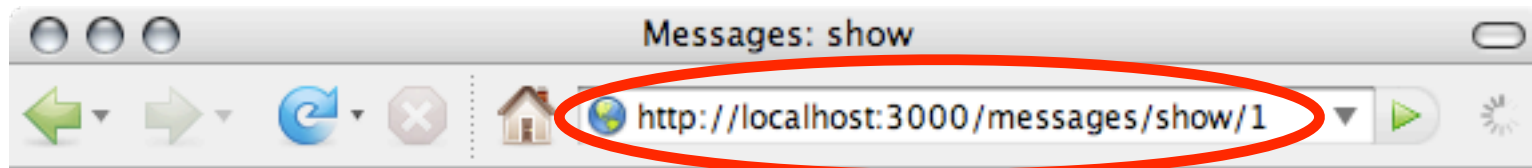
dynamisch

Namens- konventionen

```
$ rails hello_world
```

```
$ ./script/generate scaffold Message
```

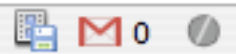
```
$
```



Text: Hello World

Edit | Back

Fertig



Zeitüberschreitung

<http://localhost:3000/messages/show/>

/messages/show/ |

```
$ mate config/routes.rb
```

```
1 ActionController::Routing::Routes.draw do |map|
```

```
  # ...
```

```
  map.connect ':controller/:action/:id'
```

```
end
```

`/messages/show/1`

`:controller = messages`

`:action = show`

`:id = 1`

```
$ mate app/controllers/messages_controller.rb
```

```
1 class MessagesController < ApplicationController  
  
  # ...  
  
  def show  
    @message = Message.find(params[:id])  
  end  
  
  # ...  
  
end
```

```
$ mate app/models/message.rb
```

```
1 class Message < ActiveRecord::Base  
end
```

```
$ ls -1F app/views/messages/
```

```
_form.rhtml
```

```
edit.rhtml
```

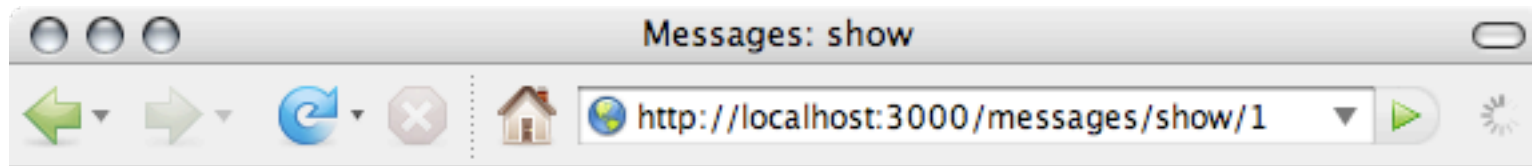
```
list.rhtml
```

```
new.rhtml
```

```
show.rhtml
```

```
1 <% for column in Message.content_columns %>
  <p>
    <b><%= column.human_name %>:</b>
    <%=h @message.send(column.name) %>
  </p>
<% end %>

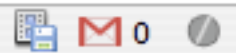
<%= link_to 'Edit',
  :action => 'edit',
  :id => @message %> |
<%= link_to 'Back',
  :action => 'list' %>
```



Text: Hello World

Edit | Back

Fertig



Zeitüberschreitung

ActiveRecord

messages

id

text

```
1 class Message < ActiveRecord::Base  
end
```

class Message

≈

Tabelle messages

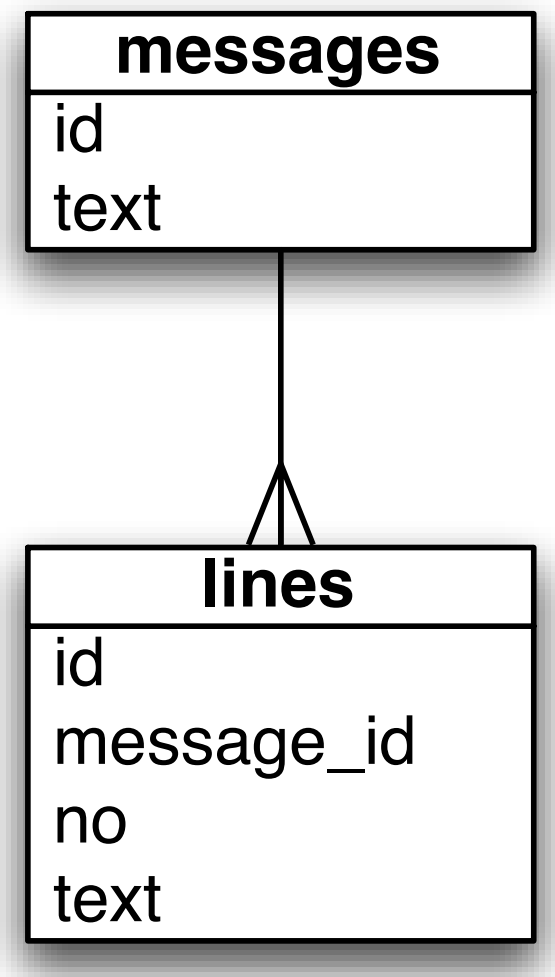
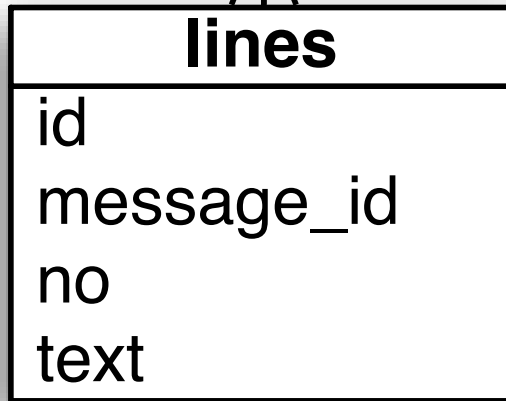
Property text

≈

Feld text

```
$ ./script/console
Loading development environment.
>> m = Message.new(:text => 'hello')
>> m.save
>> m.id
>> 1
>> exit
```

```
$ ./script/console
Loading development environment.
>> m = Message.find 1
>> m.text
>> "hello"
```



```
1 class Message < ActiveRecord::Base  
  has_many :lines  
end
```

```
$ ./script/console
Loading development environment.
>> m = Message.new :text => 'abc'
>> m.save
>> m.id
>> 2
>> m.lines << Line.new(:text => 'def')
>> exit
```

```
$ ./script/console
Loading development environment.
>> m = Message.find 2
>> m.lines.first.text
>> "def"
```

Convention over Configuration

statisch - Verzeichnisstruktur

dynamisch - Namenskonvention

**No
Impedance
Mismatch**

Browser

Applikationsserver

Datenbank

HTML, JavaScript

Java, Ruby, Python

SQL

HTML, JavaScript

Java - Hibernate

RubyOnRails

rhtml, rjs

Ruby

ActiveRecord

AJAX

integriert

link_to **link_to_remote**

form_tag **form_remote_tag**

rjs-Template

```
page.replace_html
```

```
"cart", :partial => "cart"
```

Railsentwicklung

Prototype-,
script.aculo.us-
Entwicklung

III.

**Wo funktioniert
das nicht?**

**Convention
over
Configuration**

ActiveRecord

Zusammengesetzte Schlüssel

Klassenstruktur
≠
Tabellenstruktur

Application- Database

SingleTableInheritance

Rails

Deployment

Webkonnektor

webrick

fastcgi

lighttpd

mod_ruby

apache

mongrel

Ruby

Bibliotheken

Threading

Garbage Collection

Performance

Antworten

Rails + rBatis

Rails + SQL

Plugins

Performance

Page-Caching

Action-Caching

Fragment-Caching

YARV

jruby

xruby



Ende

**Vielen Dank für
ihre Aufmerksamkeit ...**

... und noch ein
Veranstaltungshinweis in
eigener Sache ...

Agilität und Mikropolitik

Dienstag 29. Mai

20:00 Lagerhaus

Agile Gruppe Bremen

agb.wikipaces.com

Fragen?