



Fenomen Ruby On Rails

Marek Kirejczyk

Część 2
Ruby on Rails

Framework

Eksplozja frameworków?

- Php
 - Zend
 - Symfony
- Python
 - Django
 - TurboGears
 - Pylons
- SmallTalk Seaside
- .NET
 - Ajax Extension
 - MonoRail
- Java
 - Hibernate
 - Spring
 - Guice
 - Tempesty
 - Struts
 - Stripes
- JavaScript
 - DoJo
 - Prototype
 - GWT

Ekspozja frameworków!

- Php

- EZ Publish , LogiCreate , Mojavi , Navigator , Phrame , ZNF , Achievo ATK , Akelos Framework , AModules3 , Ambivalence , Aukyla PHPFramework , Binarycloud , Biscuit , bitweaver , Booby , Cake , Castor , Cgiapp , CodeIgniter , Copix , Core Enterprise PHP , FastFrame , FuseLogic , Konstrukt , Kumbia , Krysalis , Inek , InterJinn , Ismo , Medusa , PHPon Trax , PhpMVC , Popoon , Prado , Qcodo , rwfphp , Seagull , Sitellite , SolarPHP , sQeletor , Studs , struts4php , symfony , TaniPHP , Tigermouse , web.framework , Wolfden CMF , Zephyr Framework , Zoop Framework

- Python

- Django, TurboGears, web.py, Pylons, Fanery, Webware, SkunkWeb, Zope

- Java

- Hibernate, Spring, Guice, Tempestry, Struts, Stripes, Wicket, Sails, Trails, Cocoon, WebWork, Turbine, Maverick, Echo, SOFIA, Verge, Anvil, Jaffa, Japple, JPublish, Jucas, WebOnSwing, Chrysalis, VRaptor, Millstone, Swinglets, Wicket, Millstone ...

- JavaScript

- DoJo, Prototype, GWT, TrimJunction, AJAX WebShop, ActiveWidgets , AjaxFace , Backbase , Bindows , Nitobi , Engine for Web Applications , Ext , Freja , Gravey , JackBe , Javeline FrameWork , JsLINB , JsRia , Macao , MochiKit , Mootools, Open Rico , OpenLink AJAX Toolkit, Plex Toolkit , Prototype , jQuery , qooxdoo , Script.aculo.us , SmartClient , Subsys_ JsHttpRequest, ThyAPI , Interactive Website Framework , TIBET , UI4W , YUI, inLIKE , Zimbra AjaxTK ,

How many software developers does it take to
change a light bulb?

Twenties?

Thirties?

Forties?

Frameworkties!

We are only one framework from the solution?

- Skąd taka popularność właśnie RoR przy takim zalewie frameworków?
- Nie warto być jednym z tysięcy studentów, ambitnych pracowników czy uczniów liceum, którzy właśnie dziś zaczynają pisać swój framework WWW inny niż wszystkie.
- Jeśli chce się nauczyć tylko jednego frameworku to RoR wydaje się dobrym pomysłem

Zasady RoR

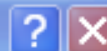
- DRY – Don't Repeat Yourself
- Convention over configuration

Instalacja

Instant Rails

- `students/~mk209471`
- Lub
`http://rubyforge.org/frs/download.php/18843/InstantRails-1.7-win.zip`
- Lub I'm feeling lucky: instant rails
- Rozpakowanie zajmie dłużej niż napisanie prostej aplikacji ;)
- Dowolny katalog, ale bez spacji w ścieżce!
 - Proponuje `c:\temp`

Zmienne środowiskowe



Zmienne użytkownika dla Marek

Zmienna	Wartość
iPlusPath	C:\Program Files\iPlus
PATH	C:\Code\InstantRails\ruby\bin

Edytowanie zmiennej użytkownika



Nazwa zmiennej:

Wartość zmiennej:

ComSpec	C:\WINDOWS\system32\cmd.exe
FP_NO_HOST_C...	NO
NUMBER_OF_P...	2
OS	Windows_NT

Właściwości sys

Aby móc przepro
Administrator.

Wydajność

Efekty wizualn
pamięć wirtualn

Profile użytkow

Ustawienia pul

Uruchamianie i

Informacje o ur

InstantRails Applications [X]

Rails Applications

- cookbook
- typo-2.6.0

Web Server

Configure Startup Mode...

Start with Mongrel

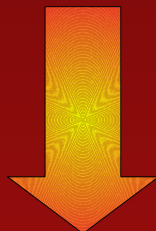
Open a Rails Console...

Check one or more Rails applications and then click a button above to perform that action on the selected applications.

To create a new Rails app, click the button below to open a console window where you can run the 'rails' command.

Unless you configure the startup mode, the default is to start a Rails app in development mode on port 3000.

Create New Rails App... Refresh List Close



InstantRails Applications [X]

Rails Applications

- cookbook
- demo
- typo-2.6.0

Web Server

Configure Startup Mode...

Start with Mongrel

Open a Rails Console...

Check one or more Rails applications and then click a button above to perform that action on the selected applications.

To create a new Rails app, click the button below to open a console window where you can run the 'rails' command.

Unless you configure the startup mode, the default is to start a Rails app in development mode on port 3000.

Create New Rails App... Refresh List Close



```
C:\InstantRails\rails_apps>ruby ..\ruby\bin\rails demo
```

http://127.0.0.1:3000/



Welcome aboard

You're riding the Rails!

[About your application's environment](#)

Getting started

Here's how to get rolling:

- 1. Create your databases and edit `config/database.yml`**
Rails needs to know your login and password.
- 2. Use `script/generate` to create your models and controllers**
To see all available options, run it without parameters.
- 3. Set up a default route and remove or rename this file**
Routes are setup in `config/routes.rb`.

Search the Rails site

Join the community

[Ruby on Rails](#)

[Official weblog](#)

[Mailing lists](#)

[IRC channel](#)

[Wiki](#)

[Bug tracker](#)

Browse the documentation

[Rails API](#)

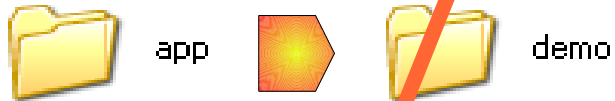
[Ruby standard library](#)

[Ruby core](#)

Krok 1: Kontroler i widok

- `ruby script/generate controller Blog`
- `app\views\blog\index.rhtml`
- `ew. controllers\blog_controller`
 `def index`
 `render :text => "Hello z kontrolera"`
 `end`

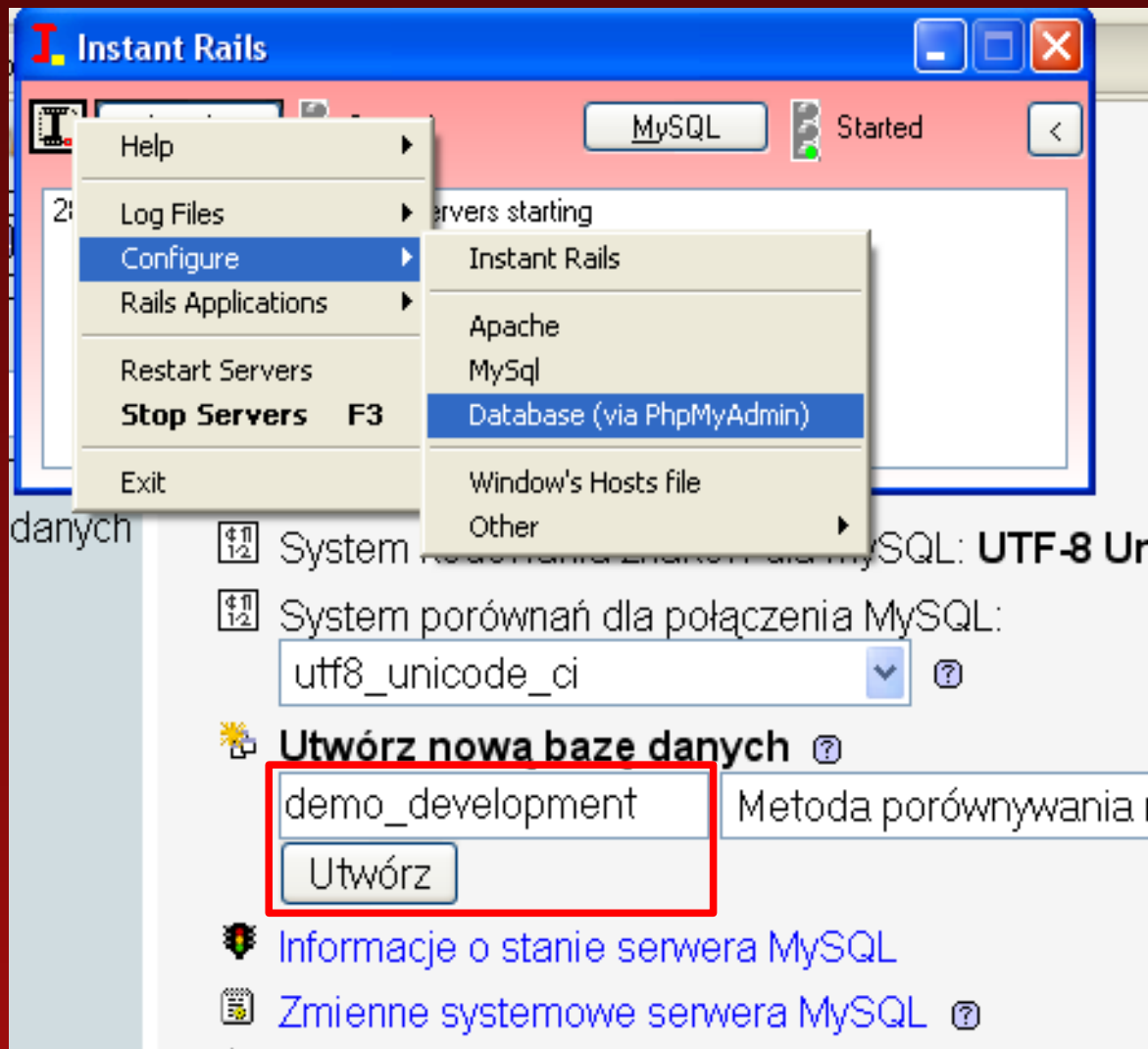
http://127.0.0.1:3000/blog/index



```
1 class BlogController < ApplicationController
2
3   def index
4     |end
5
6   end
```

1 hello

Krok 2: Baza danych







demo\config:

```
14 development:
15   adapter: mysql
16   database: demo_development
17   username: root
18   password:
19   host: localhost
20
21 test:
22   adapter: mysql
23   database: demo_test
24   username: root
25   password:
26   host: localhost
27
28 production:
29   adapter: mysql
30   database: demo_production
31   username: root
32   password:
33   host: localhost
34
```

Dodajemy tabele posts

Server: localhost ▶ Baza danych: demo_development ▶ Tabela: posts

Pole	id	title
Typ ?	INT	VARCHAR
Długość/Wartości* ¹		255
Metoda porównywania napisów		
Atrybuty		
Null	not null	not null
Domyślnie ²		
Dodatkowo	auto_increment	
	<input checked="" type="radio"/>	<input type="radio"/>
	<input type="radio"/>	<input type="radio"/>
	<input type="radio"/>	<input type="radio"/>
---	<input type="radio"/>	<input checked="" type="radio"/>
	<input type="checkbox"/>	<input type="checkbox"/>
Komentarze		

Krok 3: Dodajemy model

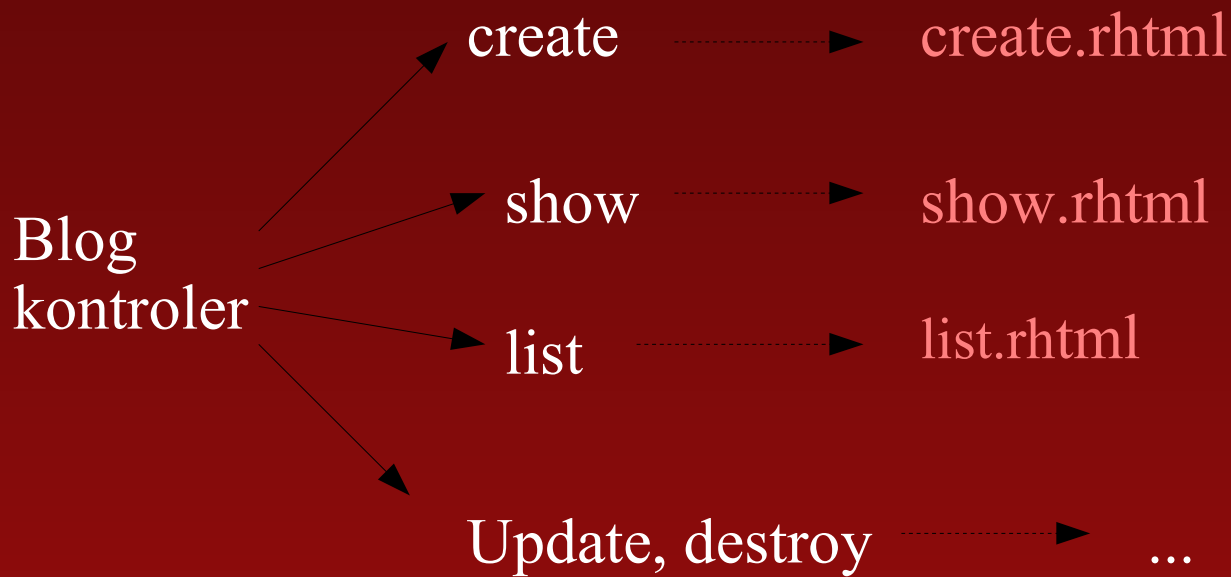
- `ruby script/generate model Post`
- Dodajemy do kontrolera scaffold `:post`
 - Dostajemy CRUD
- Dodajmy jeszcze pola:
 - `body :text`
 - `created_at:datetime`
- Wprowadźmy walidacje do modelu
 - `validates_presence_of :title`

Działa!

- Dodajmy kilka postów

Co się stało?

Wygenerował się kod w czasie kompilacji.



A co z validate?

- `validates_presence_of :title,`
 - Wywołanie metody w definicji klasy
 - Wygenerował metodę `validate`

Model

- `class Post < ActiveRecord::Base`
 - Generuje akcesory na podstawie bazy danych
 - Metody statyczne np.: `find`

Czego nie robimy!

- Nie kompilujemy
- Nie ładujemy aplikacja
- Nie konfigurujemy
 - np.: nie przypisujemy widoków akcjom

Urozmaićmy nasz blog!

- Wygenerujemy scaffolda
- `ruby script/generate scaffold Post Blog`
- Poprawki:
 - Usuwamy przewijanie
 - Kontroler `posts = find(:all)`
 - Widok `<%= link_to post.title, ...`
 - Odwróćmy kolejność postów

Poprawmy strukturę

- Tworzymy Partial
 - Wydzielamy `_post.rhtml` z `list.rhtml`
 - `<%= render :partial => "post", :collection => @posts.reverse %>`
- Ponowne użycie
 - `show.rhtml`
 - `<%= render :partial => "post", :object => @post %>`

Ruby On Rails - części

- ActiveRecord
- ActionPack
 - ActionController
 - ActiveSupport
- ActiveSupport
- Action Mailer
- Active Support

Active Record

- Problem: Nie można wyciągnąć relacji z bazy danych
- Rozwiązanie: `has_many`, `belongs_to` ...
 - Zapisujemy w modelu

Active Record

- Problem: Część aplikacji (definicje danych) trzymamy w bazie danych
- Rozwiązanie: Migracje
 - Napisane w railsach skrypty migrujące przez kolejne wersje bazy danych
 - rake db migrate

Widok i Kontroler

- Akcje automatycznie renderują widoki
- Środowisko akcji jest kopiowane do widoku
- Wszelkie pomocnicze funkcje dla widoku pisze się w helperach
 - Automatycznie wygenerowane w katalogu helpers
 - Wiele gotowych helperów, np.: do obsługi formularzy
- layouts

Scaffolding

- Szybkie prototypowanie
- Tworzenie paneli administracyjnych
 - Customizowane scuffoldy
 - Technika programowania:
 - Generowanie i edycja
- Edukacyjna rola
 - Szybkie powstawanie prostych serwisów
 - Scaffoldy pokazują wiele typowych technik RoR

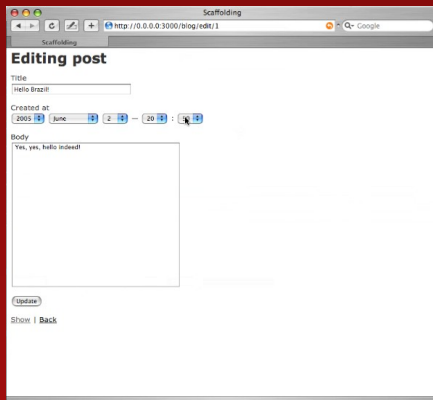
Scaffoldy

```
class BlogController < ApplicationController  
  scaffold :post  
end
```

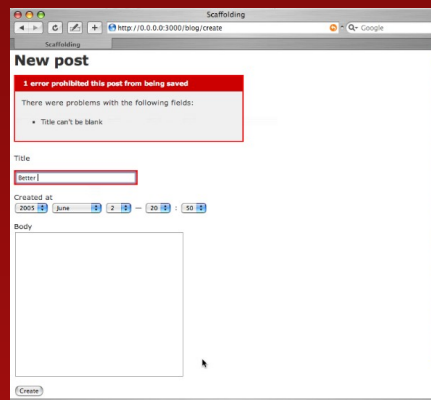
+

Field	Type	Length	unsigned	zerofill	binary	Null	Key	Default	Extra
id	int	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NO	PRI	NULL	auto_increment
title	varchar	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	YES		NULL	None
created_at	datetime		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	YES		NULL	None
body	text		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	YES		NULL	None

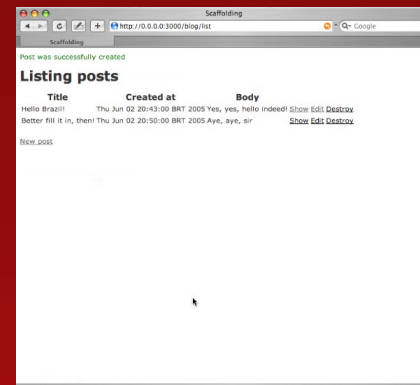
=



+



+



Testowanie

- Automatycznie generują się testy jednostkowe
- Do każdego modułu generuje się prosty fixture, automatycznie ładujący się przed testem
- Wsparcie dla testów funkcjonalnych i integracyjnych
- Specjalny katalog na mocks objects ;)

AJAX

- Rails Way: `link_to_remote`
 - Link z dodatkowym argumentem - miejscem na „wklejenie” requestu
- Bindowania do prototype'a
- Biblioteki, puginy, dodatki...

Wiecej, więcej, więcej

- Ruby i Rails 2.0
 - Intensywna praca nad spełnieniem wymogów enterprise
- Pluginy (gems)
 - np.: do obsługi i18n

Literatura

- „Agile Web Development with Rails” -
Dave Thomas, David Heinemeier Hansson