

Open Source Frameworks for Rapid Application Development

Marek Krętowski
Krzysztof Bandurski, Tomasz Łukaszuk, Tomasz Rybak

Software Departament
Faculty of Computer Science
Bialystok University of Technology

m.kretowski@pb.edu.pl
k.bandurski@pb.edu.pl, t.lukaszuk@pb.edu.pl, t.rybak@pb.edu.pl

Lecture topic

RoR and Django additional modules

RoR and Django additional modules: Table of content

- 1 Rails additional modules
- 2 Django additional modules
- 3 Rails - Creating Pdf documents
- 4 Django - Creating Pdf documents
- 5 Examples of rails plugins
- 6 Examples of django applications

Rails additional modules

- **ActiveSupport Rails library**

ActiveSupport::Cache, ActiveSupport::Callbacks,
ActiveSupport::CoreExtensions, ActiveSupport::Gzip,
ActiveSupport::Inflector, ActiveSupport::JSON, ...

- **additional Ruby libraries**

ruby gems

gem install gem_name

<http://rubygems.org>

- **Rails plugins**

RAILS_ROOT/vendor/plugins

rails plugin install plugin_url

<http://agilewebdevelopment.com/>

Django additional modules

- **Django Standard Library**

`django.contrib`

admin, admindocs, auth, contenttypes, csrf, databrowse, flatpages, formtools, gis, humanize, localflavor, markup, redirects, sessions, sitemaps, sites, syndication, webdesign

- **Python libraries**

e.g. Python Imaging Library (PIL), Matplotlib, ReportLab

- **ready-made Django applications**

`settings.py` `INSTALLED_APPS`

<http://www.django-apps.com/>

Prawn

- Gemfile

```
1 gem 'prawn'
```

- /config/initializers/mime_types.rb

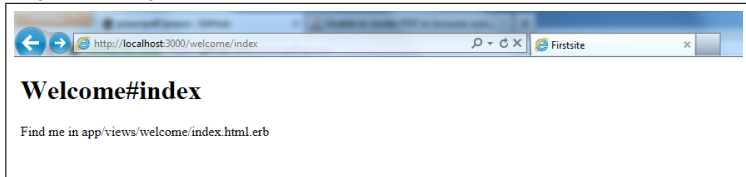
```
1 Mime::Type.register "application/pdf", :pdf
```

- controller

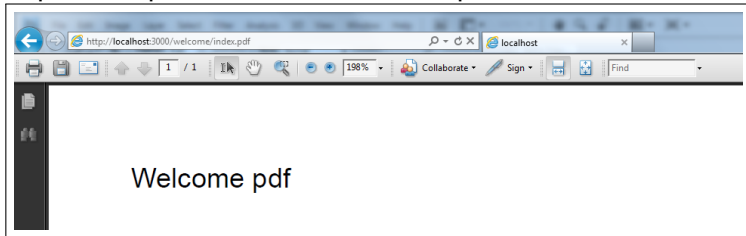
```
1 def index
2   respond_to do |format|
3     format.html
4     format.pdf do
5       pdf = Prawn::Document.new
6       pdf.text "Welcome pdf"
7       send_data pdf.render, type: "application/pdf", disposition: "inline"
8     end
9   end
10 end
```

Prawn - Result

- <http://example.com/welcome/index>



- <http://example.com/welcome/index.pdf>



Prawn - Documentation

- source:
<https://github.com/prawnpdf/prawn>
- manual:
<http://prawn.majesticseacreature.com/manual.pdf>

HTMLDOC - Preparation I

● Install HTMLDOC

```
1 wget http://ftp.easysw.com/pub/htmldoc/snapshots/htmldoc-1.9.x-r1629.tar.gz
2 tar zxvf htmldoc-1.9.x-r1629.tar.gz
3 cd htmldoc-1.9.x-r1629
4 ./configure --prefix=/usr/local
5 make
6 sudo make install
```

● Install the HTMLDOC Gem

```
1 sudo gem install htmldoc
```

● Configuring Your Application - config/...

```
1 Mime::Type.register 'application/pdf', :pdf
2 require 'htmldoc'
```

HTMLDOC - Preparation II

● PDF Renderer - app/controllers/application_controller.rb

```
1  def render_to_pdf(options = nil)
2    data = render_to_string(options)
3    pdf = PDF::HTMLDoc.new
4    pdf.set_option :bodycolor, :white
5    pdf.set_option :toc, false
6    pdf.set_option :portrait, true
7    pdf.set_option :links, false
8    pdf.set_option :webpage, true
9    pdf.set_option :left, '2cm'
10   pdf.set_option :right, '2cm'
11   pdf << data
12   pdf.generate
13 end
```

HTMLDOC - Controller

- app/controllers/posts_controller.rb

```
1  class PostsController < ApplicationController
2    def index
3      @posts = Post.all
4      respond_to do |format|
5        format.html # index.html.erb
6        format.xml { render :xml => @posts }
7        format.pdf { send_data render_to_pdf() }
8      end
9    end
10  end
```

HTMLDOC - View

- app/views/posts/index.pdf.erb

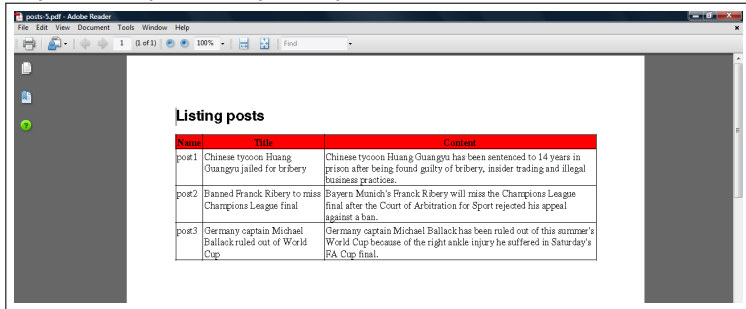
```
1  <h1>Listing posts</h1>
2  <table border="1">
3    <tr bgcolor="red">
4      <th>Name</th>
5      <th>Title</th>
6      <th>Content</th>
7    </tr>
8
9    <% @posts.each do |post| %>
10      <tr>
11        <td><%=h post.name %></td>
12        <td><%=h post.title %></td>
13        <td><%=h post.content %></td>
14      </tr>
15    <% end %>
16  </table>
```

HTMLDOC - Result

- <http://example.com/posts>

Listing posts		
Name	Title	Content
post1	Chinese tycoon Huang Guangyu jailed for bribery	Chinese tycoon Huang Guangyu has been sentenced to 14 years in prison after being found guilty of bribery, insider trading and illegal business practices. Show Edit Destroy
post2	Banned Franck Ribery to miss Champions League final	Bayern Munich's Franck Ribery will miss the Champions League final after the Court of Arbitration for Sport rejected his appeal against a ban. Show Edit Destroy
post3	Germany captain Michael Ballack ruled out of World Cup	Germany captain Michael Ballack has been ruled out of this summer's World Cup because of the right ankle injury he suffered in Saturday's FA Cup final. Show Edit Destroy
PDF New post		

- <http://example.com/posts.pdf>



The screenshot shows a web browser window with the title 'posts-5.pdf - Adobe Reader'. The document content is a table titled 'Listing posts' with three columns: Name, Title, and Content. The table contains three rows of data, each representing a post. The first row is for 'post1' (Chinese tycoon Huang Guangyu), the second for 'post2' (Banned Franck Ribery), and the third for 'post3' (Germany captain Michael Ballack). Each row includes a 'Show', 'Edit', and 'Destroy' link. The table is displayed within a PDF viewer interface, which includes a sidebar on the left and a toolbar at the top.

Listing posts		
Name	Title	Content
post1	Chinese tycoon Huang Guangyu jailed for bribery	Chinese tycoon Huang Guangyu has been sentenced to 14 years in prison after being found guilty of bribery, insider trading and illegal business practices. Show Edit Destroy
post2	Banned Franck Ribery to miss Champions League final	Bayern Munich's Franck Ribery will miss the Champions League final after the Court of Arbitration for Sport rejected his appeal against a ban. Show Edit Destroy
post3	Germany captain Michael Ballack ruled out of World Cup	Germany captain Michael Ballack has been ruled out of this summer's World Cup because of the right ankle injury he suffered in Saturday's FA Cup final. Show Edit Destroy

Preparation and application code

- Install ReportLab Python PDF library

<http://www.reportlab.com/software/opensource/rl-toolkit/download/>

- View

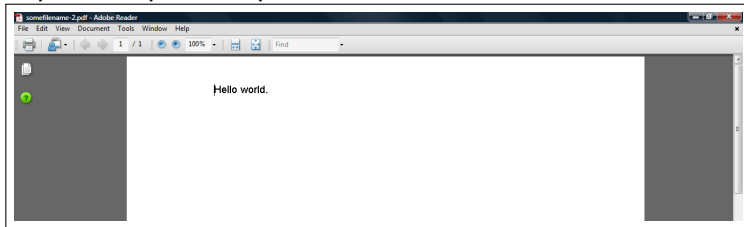
```
1  from reportlab.pdfgen import canvas
2  from django.http import HttpResponse
3
4  def show_pdf(request):
5      # Create the HttpResponse object with the appropriate PDF headers.
6      response = HttpResponse(mimetype='application/pdf')
7      response['Content-Disposition'] = 'attachment; filename=somefilename.pdf'
8
9      # Create the PDF object, using the response object as its "file."
10     pdf = canvas.Canvas(response)
11
12     # Draw things on the PDF. Here's where the PDF generation happens.
13     pdf.drawString(100, 800, "Hello world.")
14
15     # Close the PDF object cleanly, and we're done.
16     pdf.showPage()
17     pdf.save()
18     return response
```

Result

- urls.py

```
1 (r'^pdf$', 'show_pdf'),
```

- http://example.com/pdf



Calendar Helper I

- installation

```
1 wget http://github.com/topfunky/calendar_helper/tarball/rails3
2 # unpack to /venders/plugins
3 # rename to calendar_helper
```

- configuration

```
1 rails generate calendar_styles
```

- example

```
1 #layout
2 <head>
3   ...
4   <%= stylesheet_link_tag 'calendar/blue/style' %>
5 </head>
6
7 #view
8 <%= calendar(:year => 2010, :month => 5,
9   :previous_month_text => "<<", :next_month_text => ">>") %>
```

Calendar Helper II

<< May >>						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Simple Captcha I

• installation

```
1 wget http://github.com/galeki/SimpleCaptcha3/tarball/master
2 # unpack to /vendors/plugins
3 # rename to SimpleCaptcha3
```

• configuration

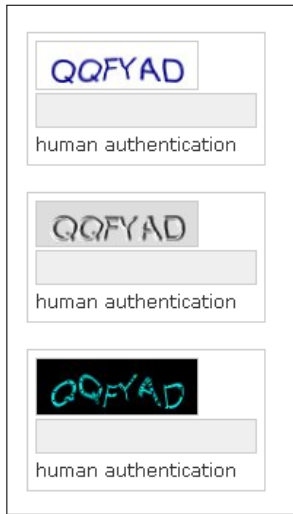
```
1 rake simple_captcha:setup
2 rake db:migrate
3 #config/routes.rb
4 match '/simple_captcha(/:action)' => 'simple_captcha', :as => :simple_captcha
5 #config/application.rb
6 include SimpleCaptcha::ControllerHelpers
```

• example

Simple Captcha II

```
1  #view
2  <%= show_simple_captcha(:label => "human authentication") %>
3  <br>
4  <%= show_simple_captcha(:label => "human authentication",
5      :image_style => 'embossed_silver') %>
6  <br>
7  <%= show_simple_captcha(:label => "human authentication",
8      :image_style => 'almost_invisible', :distortion => 'high') %>
9
10 #controller
11 if simple_captcha_valid?
12   do this
13 else
14   do that
15 end
```

Simple Captcha III



BackgroundDRb I

• installation

```
1  gem install chronic packet
2  wget https://github.com/mtylty/backgroundrb-rails3/tarball/master
3  # unpack to /vendors/plugins
4  # rename to backgroundrb-rails3
```

• configuration

```
1  rake backgroundrb:setup
2  rake db:migrate
```

• usage

BackgroundDRb II

```

1 rails generate worker longtask
2
3 #lib/workers/longtask_worker.rb
4 class LongtaskWorker < BackgroundDRb::MetaWorker
5   set_worker_name :longtask_worker
6   set_no_auto_load true
7
8   def create(args = nil)
9     # this method is called, when worker is loaded for the first time
10  end
11
12  def do_work(param)
13    `long_running_process_name`    #<-----
14  end
15
16 end
17
18 #controller
19 MiddleMan.new_worker(:worker => :longtask_worker, :worker_key => "4")
20 MiddleMan.worker(:longtask_worker, "4").async_do_work(:arg => "arg_value")

```

BackgroundDRb III

```
1  #run backgroundrb server  
2  script/backgroundrb start
```

Restful-authentication I

● installation

```
1 wget http://github.com/Satish/restful-authentication/tarball/rails3
2 # unpack to /vendors/plugins
3 # rename to restful_authentication
```

● configuration

```
1 rails generate authenticated user sessions \
2   --include-activation \
3   --stateful \
4   --rspec \
5   --skip-migration \
6   --skip-routes \
7   --old-passwords
8 #creates users_controller (registration), sessions_controller (login, logout)
9 rake db:migrate
```

Restful-authentication II

- usage

<http://example.com/signup>

Sign up as a new user

Login

Email

Password

Confirm Password

Sign up

<http://example.com/login>

Log In

Login

Password

Log in

Restful-authentication III

```
1  # in app/controllers/application_controller.rb,
2  include AuthenticatedSystem
3
4  #forcing logging
5  before_filter :find_user, :except => [:index]
6
7  #logged_in?
8  def action_name
9    if logged_in?
10      #do something for logged users
11    else
12      #do something for non-logged users
13    end
14  end
15
16  # user bar - view or layout
17  <% render :partial => "users/user_bar" %>
```

Logged in as [mylogin](#)
([Log out](#))

[Not logged in](#)
[Log in](#) / [Sign up](#)

Other Rails plugins

- **Searchlogic** - provides object based searching, common named scopes, and other useful named scope tools
- **Redbox** - a very simple lightbox library
- **Attachement Fu** - uploading files, image manipulating
- **CSS Graphy** - a simple-to-use bar-graphing view helper plugin that creates horizontal and vertical graphs only using CSS and markup
- **Progress Bar Helper** - animates the progress bar if javascript activated

plugin repositories

- <http://agilewebdevelopment.com/plugins>
- <http://dev.rubyonrails.org/svn/rails/plugins/>
- <http://topfunky.net/svn/plugins/>
- <http://github.com>

Simple Captcha I

● Installation

```
1  wget http://django-simple-captcha.googlecode.com/files/  
2      django-simple-captcha-0.1.7.tar.gz  
3  # unpack captcha folder into PythonPath place  
4  
5  # add 'captcha' to the INSTALLED_APPS in your settings.py  
6  
7  python manage.py syncdb  
8  
9  # add to urls.py  
10 url(r'^captcha/', include('captcha.urls'))
```

Simple Captcha II

- Usage
view.py

```
1  from django import forms
2  from captcha.fields import CaptchaField
3
4  class CaptchaTestForm(forms.Form):
5      captcha = CaptchaField()
6
7  def captchatest(request):
8      if request.POST:
9          form = CaptchaTestForm(request.POST)
10         if form.is_valid():
11             human = True
12     else:
13         form = CaptchaTestForm()
14     return render_to_response('captchatest.html', locals())
```

Simple Captcha III

- Usage cd
templates/captchatest.html

```
1 <h1>captcha test</h1>
2 <form action="." method="POST">
3   <table>
4     {{ form.as_table }}
5   </table>
6   <input type="submit" name="Submit!" />
7 </form>
```

captcha test

Captcha: 

Wyślij zapytanie

Django Countries I

● Installation

```
1  svn export http://django-countries.googlecode.com/svn/trunk/ \  
2    django-countries  
3  # copy countries folder into PythonPath place  
4  
5  # add 'countries' to the INSTALLED_APPS in your settings.py  
6  
7  python manage.py syncdb
```

● Usage

```
1  from countries.models import *  
2  
3  Countries.objects.all  
4  UsStates.object.filter(...)
```

Django Countries II

Django administration
Welcome, root. Change password / Log out

Home > Countries > Countries

Select Country to change

Add Country +

Action: Go

<input type="checkbox"/>	Country
<input type="checkbox"/>	India
<input type="checkbox"/>	Indonesia
<input type="checkbox"/>	Iran, Islamic Republic of
<input type="checkbox"/>	Iraq
<input type="checkbox"/>	Ireland
<input type="checkbox"/>	Isle of Man
<input type="checkbox"/>	Israel
<input type="checkbox"/>	Italy
<input type="checkbox"/>	Jamaica
<input type="checkbox"/>	Japan
<input type="checkbox"/>	Jersey

Django administration
Welcome, root. Change password / Log out

Home > Countries > Countries > Poland

Change Country

History

ISO alpha-2: PL

Official name (CAPS): POLAND

Country name: Poland

ISO alpha-3: POL

ISO numeric: 616

✖ Delete

Save and add another

Save and continue editing

Save

django-tracking

● Installation

```

1  svn co http://django-tracking.googlecode.com/svn/trunk/ \
2      django-tracking
3  # copy tracking folder into PythonPath place
4
5  # add 'tracking' to the INSTALLED_APPS in your settings.py
6
7  # Add tracking.middleware.VisitorTrackingMiddleware
8  #to your MIDDLEWARE_CLASSES in settings.py
9
10 python manage.py syncdb
11
12 # add to urls.py
13 (r'^tracking/', include('tracking.urls')),

```

● Usage

```

1  {% visitors_on_site as visitors %}
2  <p>
3      {{ visitors }} active user{{ visitors|pluralize }}
4  </p>

```

Other Django applications

- **Django-ckeditor** - application that makes it easy to use CKEditor for form textareas
- **Django evolution** - extension to Django that allows you to track changes in your models over time
- **django-sslauth** - SSL client certificate authentication and validation
- **django-googlemap** - get your site GoogleMaps-enabled

django application repositories

- <http://www.django-apps.com/>
- <http://code.djangoproject.com/wiki/DjangoResources>
- <http://github.com>