Rozwiązania szkieletowe w tworzeniu aplikacji www

Tomasz Łukaszuk

Katedra Oprogramowania
Wydział Informatyki
Politechnika Białostocka

Temat wykładu:

Dodatkowe moduły

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską
NCBR Narodowe Centrum Badań i Rozwoju

Materiały przygotowane w ramach projektu "PB 5.0 – dostosowanie oferty dydaktycznej Politechniki Białostockiej do potrzeb nowoczesnej gospodarki oraz zielonej i cyfrowej transformacji" (nr umowy FERS.01.05-IP.08-0327/23-00) w ramach programu Fundusze Europejskie dla Rozwoju Społecznego 2021-2027 współfinansowanego ze środków Europejskiego Funduszu Społecznego Plus.

Dodatkowe moduły: Agenda

1. Wprowadzenie
2. Generowanie pdf
   - RoR
   - Django
3. Przykłady ciekawych dodatkowych modułów
   - RoR
   - Django

Wprowadzenie

RoR - źródła dodatkowych modułów

- biblioteka Rails ActiveSupport
  ActiveSupport::Cache, ActiveSupport::Callbacks,
  ActiveSupport::CoreExtensions, ActiveSupport::Gzip,
  ActiveSupport::Inflector, ActiveSupport::JSON, ...
- dodatkowe biblioteki Ruby
  ruby gems
  gem install gem_name
  http://rubygems.org
- pluginy Rails
  aktualnie realizowane jako biblioteki gem

Django - źródła dodatkowych modułów

- 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/

Generowanie pdf

Prawn

- http://prawnpdf.org
- manual: prawnpdf.org/manual.pdf
- Prawn jest czystą biblioteką języka Ruby do generowania PDFów, która zapewnia wiele wspaniałych funkcji, starając się pozostać prostą i dość wydają.

Prawn - użycie

- Gemfile

```
1  gem 'prawn'
```

- instalacja gemu

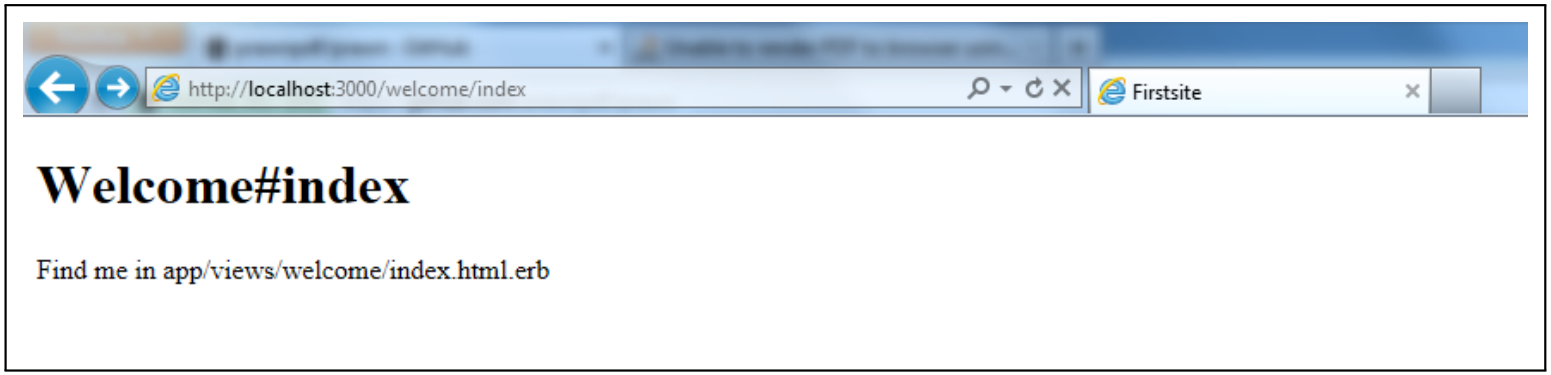
  `bundle install`

- controller

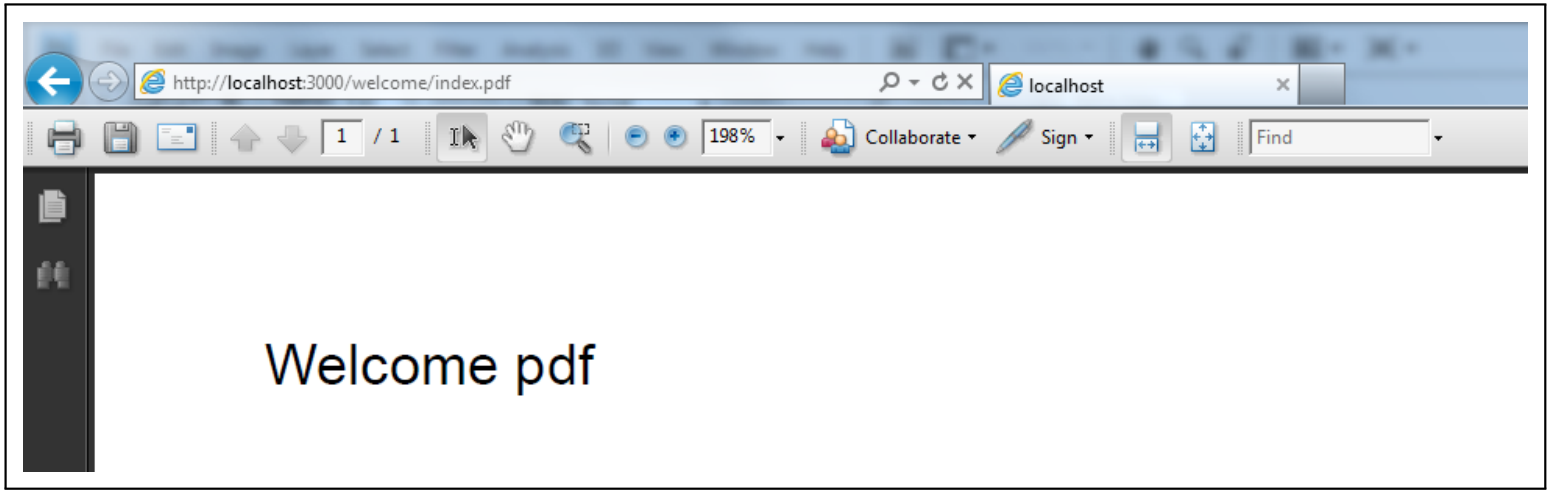
```
1  def index
2    respond_to do |format|
3     format.html
4     format.pdf do
5        pdf = Prawn::Document.new
6        pdf.text "Welcome pdf"
7        send_data pdf.render, type: "application/pdf", disposition: "inline"
8     end
9    end
10 end
```

Prawn - rezultat

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



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



HTMLDOC - przygotowanie I

- instalacja aplikacji HTMLDOC

```
1  wget http://www.msweet.org/files/project1/htmldoc-1.8.28-source.tar.gz
2  tar zxvf htmldoc-1.8.28-source.tar.gz
3  cd htmldoc-1.8.28-source
4  ./configure --prefix=/usr/local
5  make
6  sudo make install
```

- instalacja gem'a HTMLDOC

```
1  sudo gem install htmldoc
```

- PDF Renderer - app/controllers/application_controller.rb

HTMLDOC - przygotowanie II

```
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 - kontroler

- 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 - widok

- 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 - wynik

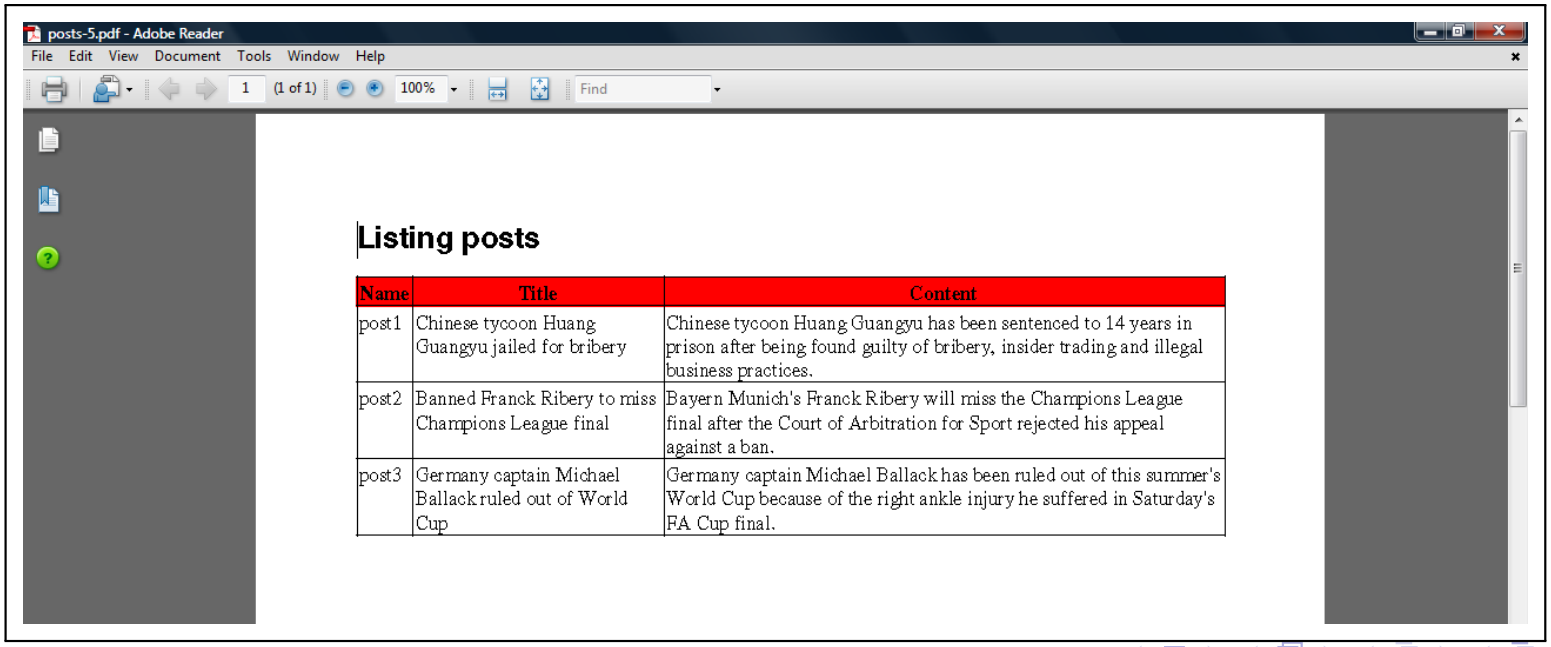
- http://example.com/posts

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



Przygotowania i kod aplikacji

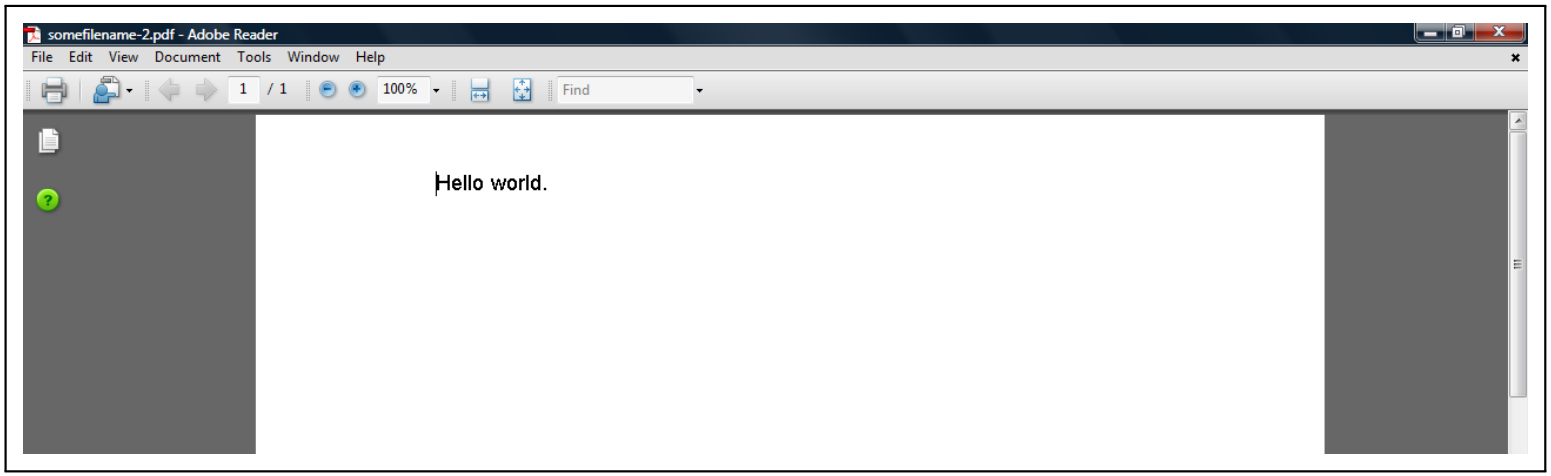
```
1   # python -m pip install reportlab
2
3   import io
4   from django.http import FileResponse
5   from reportlab.pdfgen import canvas
6
7   def show_pdf(request):
8       # Create a file-like buffer to receive PDF data.
9       buffer = io.BytesIO()
10
11      # Create the PDF object, using the buffer as its "file."
12      p = canvas.Canvas(buffer)
13
14      # Draw things on the PDF. Here's where the PDF generation happens.
15      p.drawString(100, 100, "Hello world.")
16
17      # Close the PDF object cleanly, and we're done.
18      p.showPage()
19      p.save()
20
21      # FileResponse sets the Content-Disposition header.
22      buffer.seek(0)
23      return FileResponse(buffer, as_attachment=True, filename='hello.pdf')
```

ReportLab - wynik

- urls.py

```
1   path('pdf', 'views.show_pdf'),
```

- http://example.com/pdf



Przykłady ciekawych dodatkowych modułów

RoR - Calendar Helper I

- instalacja

```
1  gem install calendar_helper
```

- Gemfile

```
1  gem 'calendar_helper', '~> 0.2.6'
```

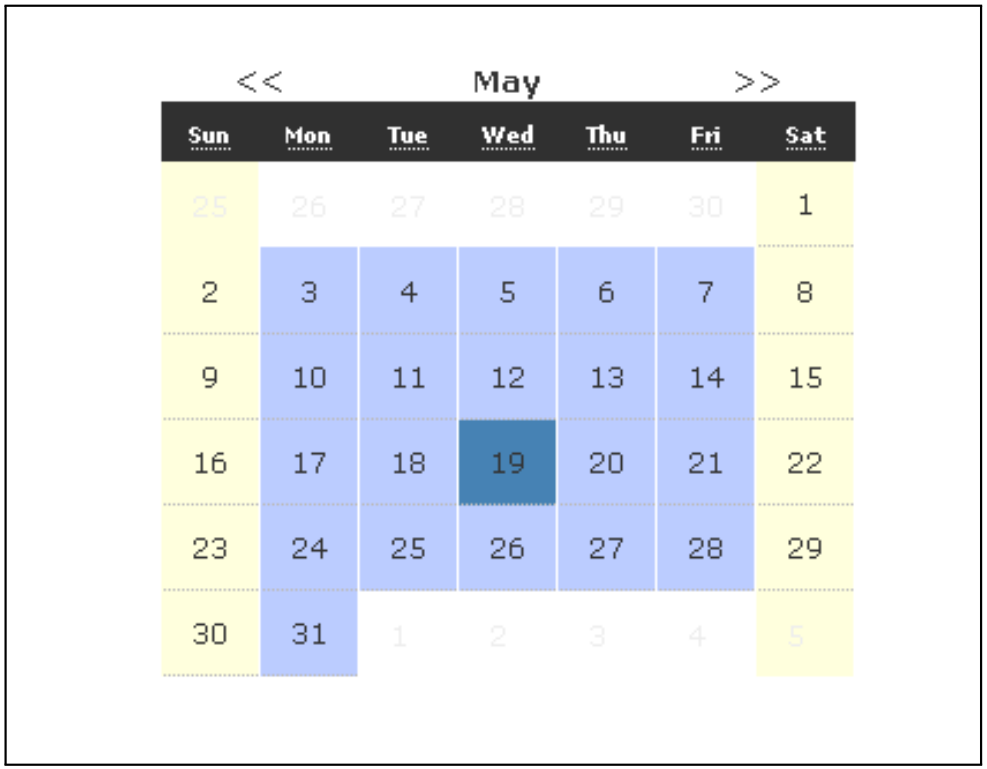
- konfiguracja `app/assets/stylesheets/application.css`

```
1  //= require 'calendar_styles/grey'
2  //= require 'calendar_styles/red'
3  //= require 'calendar_styles/blue'
```

- przykład

```
1  #view
2  <%= calendar(:year => 2010, :month => 5,
3       :previous_month_text => "<<", :next_month_text => ">>") %>
```

RoR - Calendar Helper II



RoR - Simple Captcha I

- instalacja

```
1  #Gemfile
2  gem "galetahub-simple_captcha", :require => "simple_captcha"
3  #command
4  bundle install
```

- konfiguracja

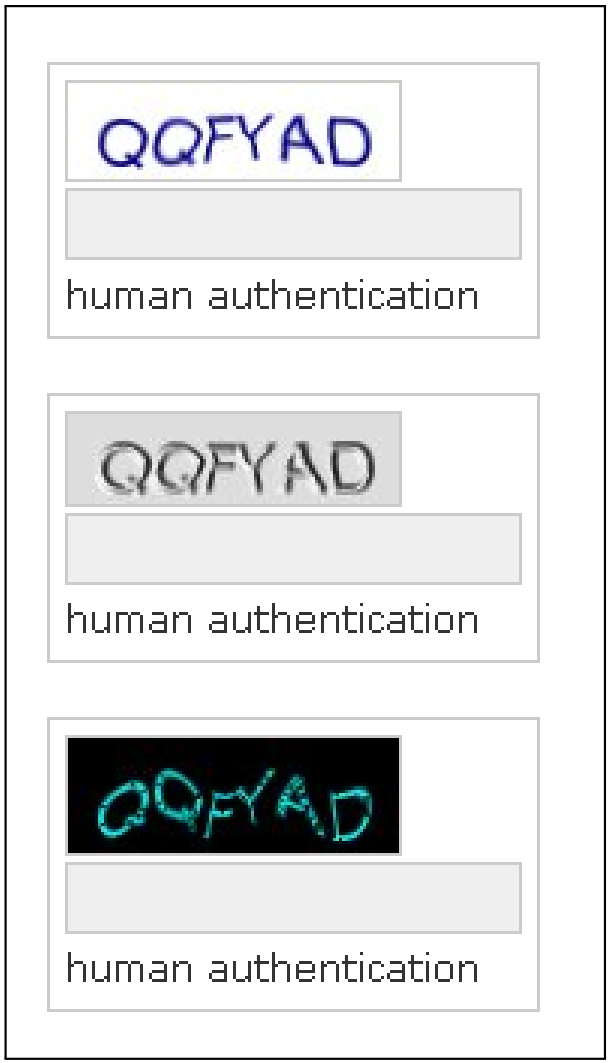
```
1  rails generate simple_captcha
2  rails db:migrate
```

- przykład

RoR - Simple Captcha II

```
1  #view
2  <%= show_simple_captcha(:label => "human authentication") %>
3  <br>
4  <%= show_simple_captcha(:label => "human authentication",
5       :image_style => 'embosed_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
```

RoR - Simple Captcha III



RoR - BackgrounDRb I

- instalacja

```
1  #Gemfile
2  gem 'backgroundrb-rails3', :require => 'backgroundrb'
3  #command
4  bundle install
```

- konfiguracja

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

- użycie

RoR - BackgrounDRb II

```
1  rails generate worker longtask
2
3  #lib/workers/longtask_worker.rb
4  class LongtaskWorker < BackgrounDRb::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")
```

RoR - BackgrounDRb III

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

RoR - Restful-authentication I

- instalacja

```
1  #Gemfile
2  gem 'restful-authentication', '~> 1.2.1'
3  #command
4  bundle install
```

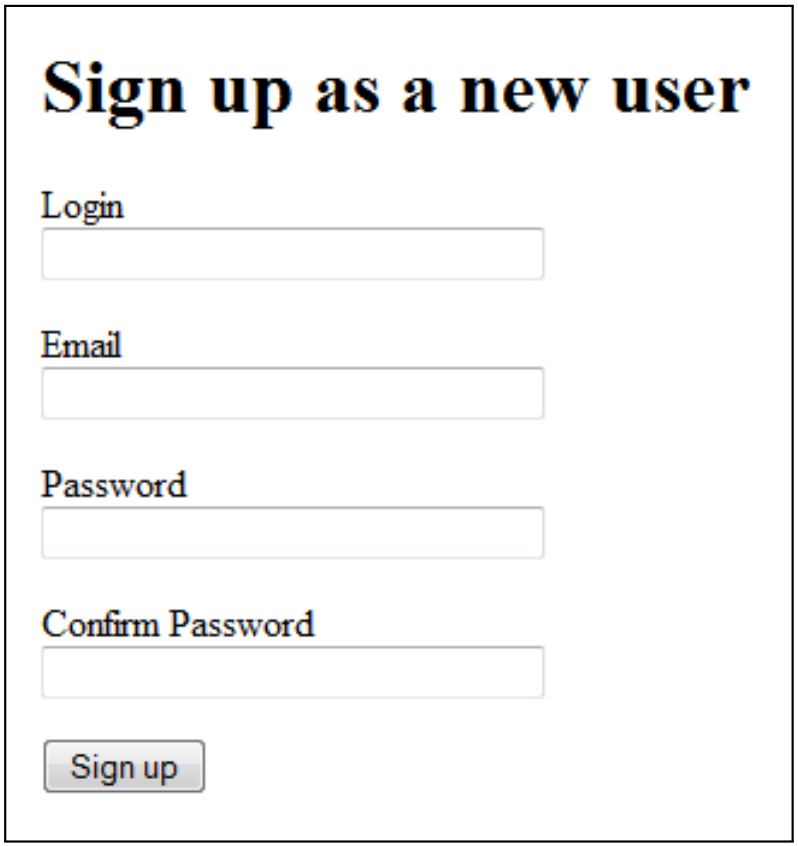
- konfiguracja

```
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  rails db:migrate
```

RoR - Restful-authentication II

- użycie

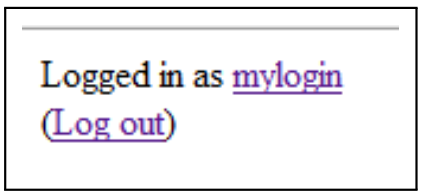
http://example.com/signup



http://example.com/login

RoR - 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" %>
```



Inne plugin'y Rails

- **Redbox** - prosta bibliotka lightbox
- **Attachement Fu** - ładowanie plików, manipulowanie obrazkami graficznymi
- **CSS Graphy** - tworzenie wykresów graficznych
- **Progress Bar Helper** - pasek postępu, używa javascript

repozytoria plugin'ów

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

Simple Captcha I

- Instalacja

```
1  pip install django-simple-captcha
2
3  # add 'captcha' to the INSTALLED_APPS in your settings.py
4
5  python manage.py migrate
6
7  # add to urls.py
8  path('captcha/', include('captcha.urls')),
```

Simple Captcha II

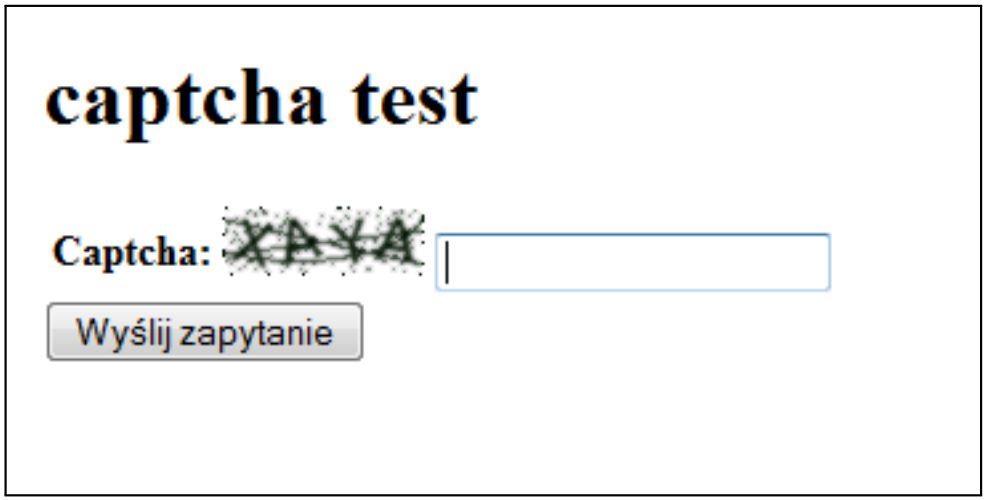
- Użycie

```
# forms.py
from django import forms
from captcha.fields import CaptchaField

class CaptchaTestForm(forms.Form):
    myfield = AnyOtherField()
    captcha = CaptchaField()

# views.py
def some_view(request):
    if request.POST:
        form = CaptchaTestForm(request.POST)

        # Validate the form: the captcha field will automatically
        # check the input
        if form.is_valid():
            human = True
    else:
        form = CaptchaTestForm()

    return render(request, 'template.html', {'form': form})
```

Simple Captcha III

- Użycie 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>
```



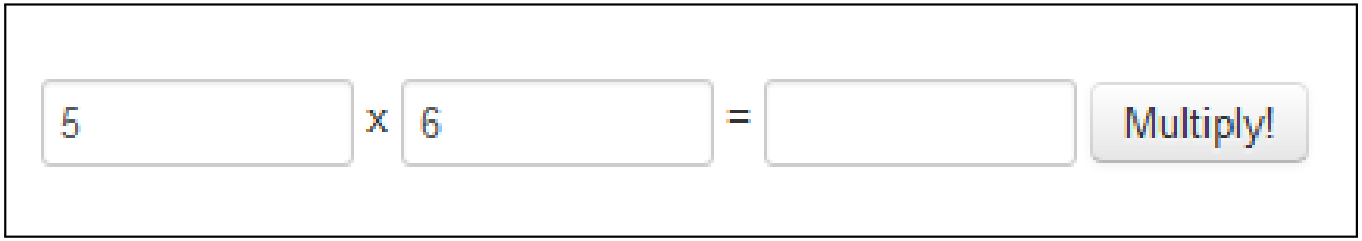
Dajax, Dajaxice

http://django-dajax.readthedocs.org
http://django-dajaxice.readthedocs.org

- Instalacja

```
1  pip install django_dajax
2
3  #add dajax and dajaxice in your project settings.py inside INSTALLED_APPS
4  INSTALLED_APPS = (
5      'django.contrib.auth',
6      'django.contrib.contenttypes',
7      'django.contrib.sessions',
8      'django.contrib.sites',
9      'dajaxice',
10     'dajax',
11     ...
12 )
```

Dajax, Dajaxice - przykład I



- ajax.py

```
1  from dajax.core import Dajax
2  from dajaxice.decorators import dajaxice_register
3
4  @dajaxice_register
5  def multiply(request, a, b):
6      dajax = Dajax()
7      result = int(a) * int(b)
8      dajax.assign('#result','value',str(result))
9      return dajax.json()
```

- html

Dajax, Dajaxice - przykład II

```
1  <input type="text" value="5" id="a"> x
2  <input type="text" value="6" id="b"> =
3  <input type="text" value="" id="result">
4  <input type="button" value="Multiply!" onclick="calculate();">
```

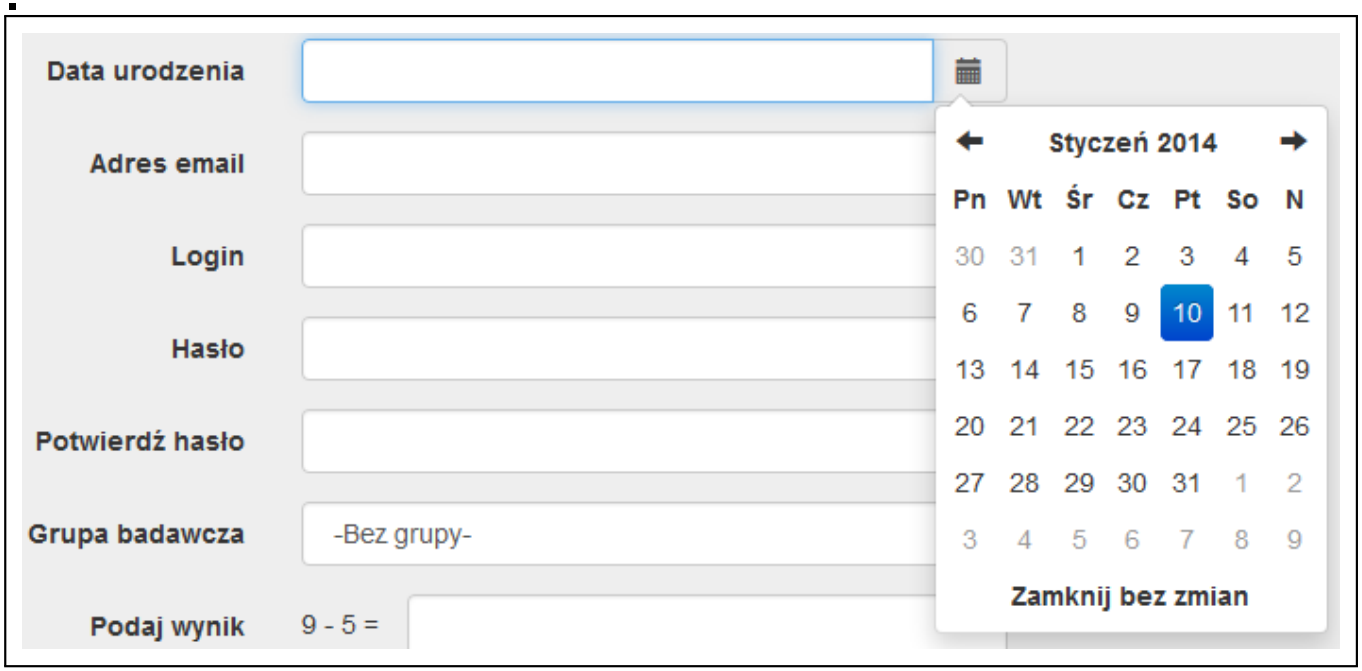
- javascript

```
1  function calculate(){
2      Dajaxice.examples.multiply(
3              Dajax.process,{'a':$('#a').val(),'b':$('#b').val()}
4      )
5  }
```

Datetimewidget

`github.com/asaglimbeni/django-datetime-widget`

.



Datetimewidget - instalacja i użycie I

- Instalacja

```
pip install django-datetime-widget
#Add datetimewidget to your INSTALLED_APPS
#Set USE_L10N = True, USE_TZ = True and USE_I18N = True in settings.py
#Add 'django.middleware.locale.LocaleMiddleware' to MIDDLEWARE_CLASSES
```

- Formularz

```
from datetimewidget.widgets import DateTimeWidget

class yourForm(forms.ModelForm):
    class Meta:
        model = yourModel
        widgets = {
            'datetime': DateTimeWidget(attrs={'id':"yourdatetimeid"}, usel10r
        }
```

- template

Datetimewidget - instalacja i użycie II

```
<head>
....
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.mi
    <link href="{% static 'css/bootstrap.css' %}" rel="stylesheet" type="text
    <script src="{% static 'js/bootstrap.js' %}"></script>
    {{ form.media }}

        ....
</head>
<body>
    <form  action="" method="POST">
    {% csrf_token %}
    {{ form.as_table }}
    <input id="submit" type="submit" value="Submit">
    </form>
</body>
```