

ym4r – Google Maps w Ruby on Rails

Aleksander Pohl

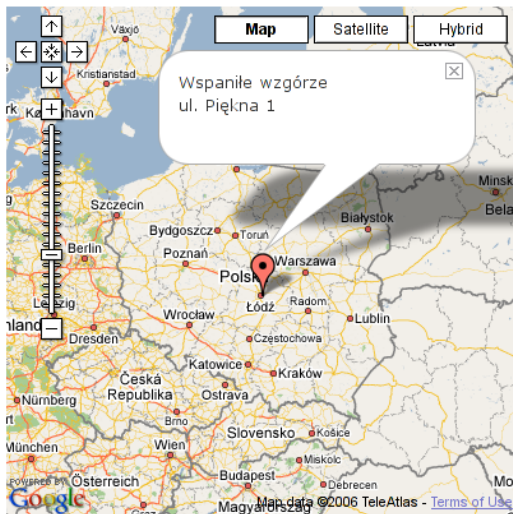
Krakow Ruby Users Group

25. listopada 2006

Kilka słów o Google Maps API

Google Maps API – interfejs programistyczny pozwalający na wyświetlenie na stronie WWW mapy wybranego rejonu Ziemi:

- ▶ Aktualna wersja: 2.0
- ▶ Google Maps vs. Google Maps for Enterprise
- ▶ Klucz licencyjny
- ▶ Ograniczenia: 500k wyświetleń, 50k geocodów
- ▶ ~ 30 klas



Integracja Google Maps z Ruby on Rails

- ▶ Zaimplementowanie własnego zestawu helperów – wykorzystanie JavaScriptowego API:
 - ▶ <http://www.google.com/apis/maps/documentation/reference.html>
 - ▶ czasochłonne
 - ▶ po co wymyślać od nowa koło?
- ▶ Cartographer:
 - ▶ <http://cartographer.rubyforge.org/>
 - ▶ kiepska i nieaktualna dokumentacja
 - ▶ niewielkie możliwości
 - ▶ kiepskie przykłady
 - ▶ projekt umarł?
- ▶ ym4r !

ym4r – ogólna charakterystyka

ym4r: yellow maps for rails:

- ▶ <http://thepochisuperstarmegashow.com/projects/#ym4r>
- ▶ dobra dokumentacja
- ▶ rozbudowane przykłady
- ▶ duże możliwości
- ▶ nie tylko GoogleMaps:
 - ▶ Mapstraction
 - ▶ Yahoo! Maps Building Block API
 - ▶ Clusterer, GeoRSS, WMS layer

Przykładowa aplikacja

Piszemy aplikację, dla sieci restauracji. Klient chce, aby użytkownicy w łatwy sposób mogli zlokalizować poszczególne restauracje. Dane adresowe to za mało. Dobrze byłoby dołączyć do danych adresowych interaktywną mapę, na której zaznaczone jest dokładne położenie wybranej restauracji.

Wymagania:

- ▶ restauracja – wskaźnik z danymi adresowymi
- ▶ możliwość określenia położenia wskaźnika za pomocą myszki

Określenie schematu bazy danych

```
db/migrate/001_create_restaurant.rb:  
create_table "restaurants" do |t|  
  t.column "name", :string  
  t.column "street", :string  
  t.column "number", :string  
  t.column "city", :string  
  t.column "latitude", :float  
  t.column "longitude", :float  
  t.column "zoom", :float  
end
```

Instalacja ym4r i scaffolding

```
$ ruby script/plugin install↵  
  svn://rubyforge.org/var/svn/ym4r/Plug↵  
  ins/GM/trunk/ym4r_gm
```

↵ oznacza kontynuację linii

```
$ ruby script/generate scaffold restaurant
```


Modyfikacje I: model

```
app/model/restaurant.rb:  
def address  
  city+"<br/>"+"ul. "+street+" "+number  
end  
def location  
  if(latitude.nil?)  
    [52,18]  
  else  
    [latitude, longitude]  
  end  
end  
def has_map?  
  ! latitude.nil?  
end
```

Modyfikacje II: widoki

```
app/views/layout/restaurants.rhtml:
```

```
<head>
```

```
  <% if ! @map.nil?  %>
```

```
    <%= GMap.header %>
```

```
    <%= @map.to_html %>
```

```
    <%= javascript_include_tag 'gmap_edit' %>
```

```
  <% end %>
```

```
</head>
```

```
app/views/restaurants/show.rhtml:
```

```
<% if @restaurant.has_map?  %>
```

```
  <%= @map.div(:width => 400, :height => 400) %>
```

```
<% end %>
```

Modyfikacje III: widoki (c.d.)

```
app/views/restaurants/_form.rhtml:  
<%= text_field 'restaurant', 'latitude',  
  :id => "latitude" %></p>  
<%= text_field 'restaurant', 'longitude',  
  :id => "longitude" %></p>  
<%= text_field 'restaurant', 'zoom',  
  :id => "zoom" %></p>  
<%= @map.div(:width => 400, :height => 400) %>
```

Modyfikacje IV: kontroler

```
app/controller/restaurants.rb (show):  
@restaurant = Restaurant.find(params[:id])  
if @restaurant.has_map?  
  loc = @restaurant.location  
  @map = GMap.new("map_div")  
  @map.control_init(:large_map => true,  
    :map_type => true)  
  @map.center_zoom_init(loc,@restaurant.zoom)  
  @map.overlay_init(GMarker.new(loc,  
    :title => @restaurant.name,  
    :info_window => @restaurant.address))  
end
```

Modyfikacje V: kontroler (c.d.)

```
Kontroler app/controller/restaurants.rb (edit):  
  @restaurant = Restaurant.find(params[:id])  
  @map = GMap.new("map_div")  
  loc = @restaurant.location  
  @map.control_init(:large_map => true,  
    :map_type => true)  
  @map.center_zoom_init(loc, @restaurant.zoom)  
  marker = GMarker.new(loc, :draggable => true)  
  @map.overlay_global_init(marker, "marker")  
  @map.record_init  
    marker.on_dragend("gmap_update_position")  
  @map.record_init  
    @map.on_zoomend("gmap_update_zoom")
```

Podsumowanie

Czynności wstępne:

- ▶ określenie schematu bazy danych (latitude, longitude, zoom)
- ▶ zainstalowanie pluginu
- ▶ dodanie skryptów inicjujących w nagłówku

Wyświetlenie mapy:

- ▶ dodanie mapy w ciele dokumentu
- ▶ inicjacja obiektu mapy w kontrolerze

Edycja mapy:

- ▶ dodanie skryptów modyfikujących
- ▶ modyfikacja formularza (identyfikatory pól)
- ▶ inicjacja mapy i markera w kontrolerze