Kurs programowania Ruby on Rails

Zagadnienia zaawansowane

Mailery

Szkolenie Rails

Wykonywane do wysyłania maili.

Mailery działają także w oparciu o wzorzec MVC.

Generacja pliku mailera

Szkolenie Rails

```
rails generate mailer UserMailer
```

Generacja pliku mailera

Szkolenie Rails

```
rails generate mailer UserMailer
```

Przykładowa akcja Mailera

Szkolenie Rails

```
class UserMailer < ActionMailer::Base
  default from: 'notifications@example.com'

  def welcome_email(user)
    @user = user
    @url  = 'http://example.com/login'
    mail(to: @user.email, subject: 'Welcome to My
Awesome Site')
  end
end
```

Przykładowy widok

Szkolenie Rails

```
<!DOCTYPE html>
<html>
  <head>
    <meta content='text/html; charset=UTF-8' http-equiv='Content-Type' />
  </head>
  <body>
    <h1>Welcome to example.com, <%= @user.name %></h1>
    <p>
      You have successfully signed up to example.com,
      your username is: <%= @user.login %>.<br/>
    </p>
    <p>
      To login to the site, just follow this link: <%= @url %>.
    </p>
    <p>Thanks for joining and have a great day!</p>
  </body>
</html>
```

Deliver

Szkolenie Rails

-aby wysłać musimy użyć metody Deliver:

UserMailer.welcome_email(@user).deliver

-więcej informacji na

http://guides.rubyonrails.org/action_mailer_basics.html

Konfiguracja mailera

Szkolenie Rails

-config/environments/development.rb

```
config.action_mailer.raise_delivery_errors = true

# Change mail delvery to either :smtp, :sendmail, :file, :test
  config.action_mailer.delivery_method = :smtp
  config.action_mailer.smtp_settings = {
      address: "smtp.gmail.com",
      port: 587,
      domain: "gmail.com",
      authentication: "plain",
      enable_starttls_auto: true,
      user_name: ENV["email"],
      password: ENV["password"]
  }

# Specify what domain to use for mailer URLs
  config.action_mailer.default_url_options = {host: "localhost:3000"}
```

Zmienne środowiskowe

Szkolenie Rails

- zmienne, które zależne są od środowiska, w którym pracujemy
- dzięki nim zabezpieczamy bezpieczeństwo
- ustalamy je na środowisku, na którym pracujemy
- może być sytuacja, że developer posiada dostęp do środowiska testowego, stagingowego , a może nie posiadać dostępu do produkcji
-z poziomu basha:

export EMAIL="makaruk.michal@gmail.com"

-odwołanie się z poziomu Railsów:
ENV['EMAIL"]

Maile dlaczego nie dochodzą??

-Protokoł SMTP: możemy podszyć się pod „Prezydenta USA”
-Serwer może nas zablokować możemy trafić na tzw. „black listy”
-firmy, które specjalizują się w wysyłce danych to np. FreshMail ,GetResponse
-mailing to ważny temat np. www.activeads.pl
-serwer smtp może nam nie umożliwić wysyłanie z zewnątrz maili, limity, potwierdzenia na gmailu.
-na heroku możemy korzystać np. z sendgrida do wysyłki maili.

Inne pytania.....

Szkolenie Rails

- do ilu osób dotarł mailing?
- ile osób odczytało maila?
- dlaczego mail nie dotarł?
- ile osób to kobiety?

Długie operacje

Szkolenie Rails

-jeśli operacja trwa bardzo długo to użytkownik musiałby czekać bardzo długo, aż zostanie wykonany dany proces

-w aplikacjach mobilnych np. przy pobieraniu zdjęcia użytkownik nie miałby wyświetlonego widoku, bo musiałby czekać na odpowiedź serwera

Wątek

Szkolenie Rails

Wątek (ang. *thread*) – część programu wykonywana współbieżnie w obrębie jednego procesu; w jednym procesie może istnieć wiele wątków.

Różnica między zwykłym procesem a wątkiem polega na współdzieleniu przez wszystkie wątki działające w danym procesie przestrzeni adresowej oraz wszystkich innych struktur systemowych (np. listy otwartych plików, gniazd itp.) – z kolei procesy posiadają niezależne zasoby.

Ta cecha ma dwie ważne konsekwencje:

Wątki wymagają mniej zasobów do działania i też mniejszy jest czas ich tworzenia. Dzięki współdzieleniu przestrzeni adresowej (pamięci) wątki jednego zadania mogą się między sobą komunikować w bardzo łatwy sposób, niewymagający pomocy ze strony systemu operacyjnego. Przekazanie dowolnie dużej ilości danych wymaga przesłania jedynie wskaźnika, zaś odczyt (a niekiedy zapis) danych o rozmiarze nie większym od słowa maszynowego nie wymaga synchronizacji (procesor gwarantuje atomowość takiej operacji).

Aby uruchomić procesy w tle

Szkolenie Rails

rake jobs:work

-na heroku płacimy za workery;/

-jeden worker może wykonywać 1 proces

-na produkcji możemy mieć np. 8 workerów

-do zadań cyklicznych możemy użyć np. schedulera na heroku: https://devcenter.heroku.com/articles/scheduler

Sposób rozwiązania długich procesów w Rails

Szkolenie Rails

-gem delayed_job_active_record
-gem 'resque'

Gem delayed_job_active_record

-dane zapisywane w bazie danych o procesach

-rails g delayed_job:active_record
rake db:migrate
rake jobs:work

-delay.nazwa_metody wtedy dana metoda jest wykonywana w tle

Gem 'prawn'

Szkolenie Rails

-gem służy do generowania pdfów.
-jest dość skomplikowany w użyciu używa specjalnych metod z, których możemy korzystać, aby generować pdfa.
-możemy za pomocą niego wygenerować np. fakturę VAT.

Mechanizm 'counter_cache'

Szkolenie Rails

- gdy mamy 2 powiązane modele i chcemy wyświetlać liczności to musimy wykonywać zapytanie o ilość np. project.tasks.size w tym momencie wykonwane jest dodatkowe zapytanie do bazy danych, gdy mamy projektów np. milion to dodatkowo milion zapytań project.tasks.size

Migracja counter_cache

Szkolenie Rails

```
class AddTasksCount < ActiveRecord::Migration
  def change
    add_column :projects, :tasks_count, :integer, :default => 0

    Project.reset_column_information
    Project.find_each do |p|
      Project.update_counters p.id, :tasks_count => p.tasks.length
    end
  end
end
```

Asocjacja counter_cache

Szkolenie Rails

```
class Task < ActiveRecord::Base
  belongs_to :project, :counter_cache => true
end
```

Wyświetlanie Project name w Tasks

Szkolenie Rails

Task Load (0.5ms) SELECT "tasks".* FROM "tasks"
Project Load (0.3ms) **SELECT "projects".* FROM "projects" WHERE "projects"."id" = $1 ORDER BY "projects"."id" ASC LIMIT 1** [["id", 1]]
Project Load (0.4ms) SELECT "projects".* FROM "projects" WHERE "projects"."id" = $1 ORDER BY "projects"."id" ASC LIMIT 1 [["id", 2]]
CACHE (0.0ms) **SELECT "projects".* FROM "projects" WHERE "projects"."id" = $1 ORDER BY "projects"."id" ASC LIMIT 1** [["id", 1]]
CACHE (0.0ms) SELECT "projects".* FROM "projects" WHERE "projects"."id" = $1 ORDER BY "projects"."id" ASC LIMIT 1 [["id", 2]]

Joins

Szkolenie Rails

-WYKONUJE NAM INNER JOIN W BAZIE DANYCH SELECT

-przykładowo:

```
@products = Product.order("categories.name").joins(:category).select("products.*, categories.name as category_name")
```

Friendly_id

Szkolenie Rails

- Gem do generowania przyjaznych adresów url dla użytkownika zamiast articles/1 mamy np. articles/zoobmies

rails g migration add_slug_to_articles

add_column :articles, :slug, :string
add_index :articles, :slug

slug:string
rake db:migrate
rails c
rails g friendly_id

Kurs programowania Ruby on Rails

Sprawy organizacyjne

Koniec