

Εγκατάσταση Rails

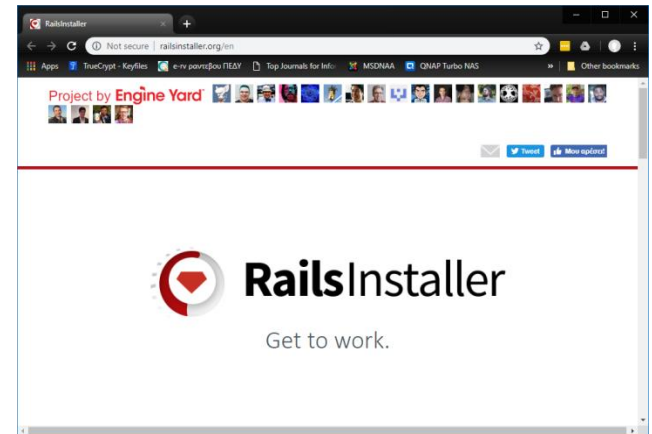
Υπηρεσιοστρεφές Λογισμικό 2020-2021

Ευάγγελος Σακκόπουλος

Τμήμα Πληροφορικής
Πανεπιστήμιο Πειραιώς

Συνοπτικά Εργαλεία

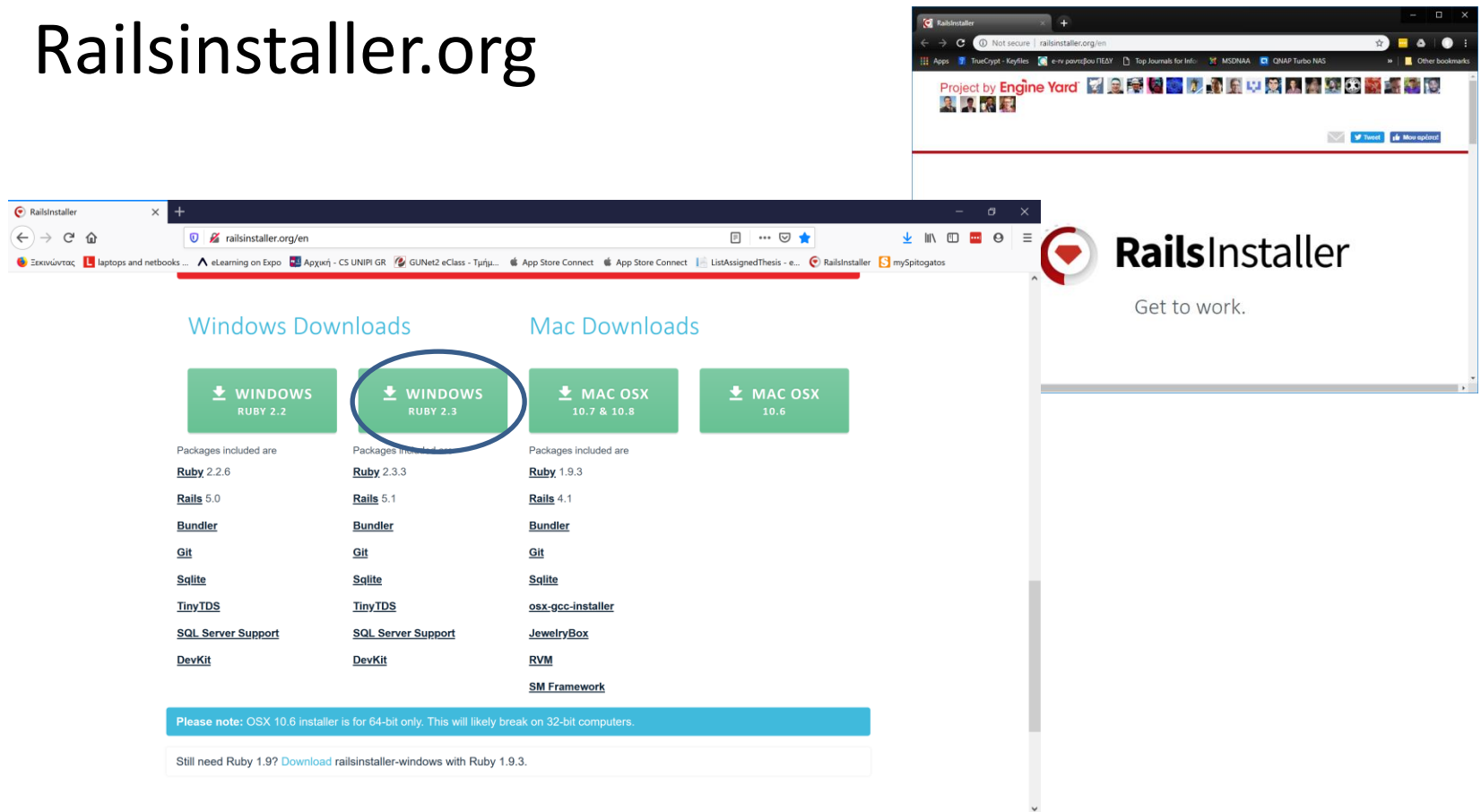
- Windows: Rails Installer
- macOS: Tokaido



- Railsinstaller.org
- For more installation methods for most Operating Systems take a look at ruby-lang.org

Συνοπτικά Εργαλεία

- Railsinstaller.org



Επιβεβαίωση

- `$ sqlite3 --version`

3.8.7.2 2014-11-18 20:57:56 2ab564bf9655b7c7b97ab85cafc8a48329b27f93

- `rails --version`

Rails 5.1.7

Νέο Blog

- rails new blog

Εγκατάσταση

rake, (αντίστοιχο του make της C για τη ruby)

rack, a Ruby Webserver Interface (<https://rack.github.io/>)

Το Rack έχει ένα "app« αντικείμενο που αντιστοιχεί στην κλήση call λαμβάνοντας παράμετρο εισόδου και επιστρέφει ένα Array με 3 στοιχεία:

HTTP response code

Hash of headers

The response body

Νέο Blog (2)

- rails new blog

Εγκατάσταση

- nio4r, New I/O for Ruby: Cross-platform asynchronous I/O primitives for scalable network clients and servers (<https://github.com/socketry/nio4r>)
(έκδοση =5.1.3)
- multi_json A generic swappable back-end for JSON handling (https://github.com/intridea/multi_json)
- Puma A Ruby/Rack web server built for concurrency (<http://puma.io>)



Rails command

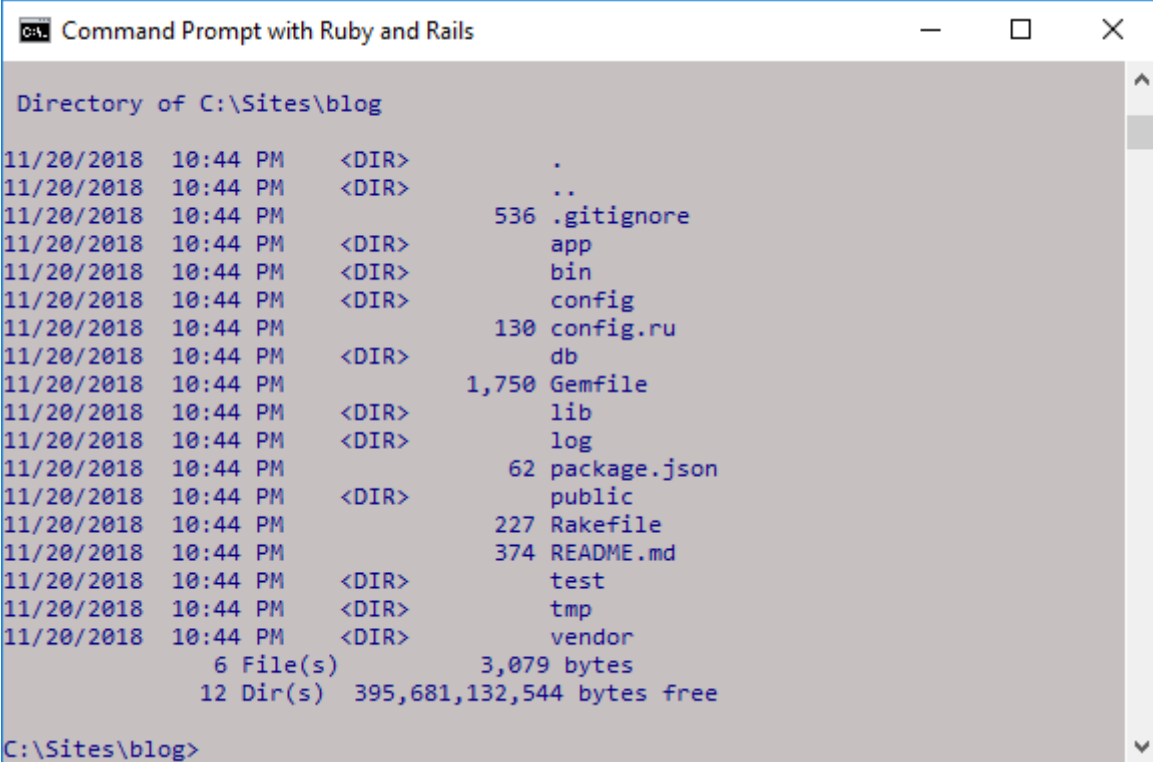
- command line options

Rails application builder

`rails new -h.`

Ο φάκελος blog

- `cd blog`



```
Command Prompt with Ruby and Rails

Directory of C:\Sites\blog

11/20/2018  10:44 PM  <DIR>      .
11/20/2018  10:44 PM  <DIR>      ..
11/20/2018  10:44 PM                536 .gitignore
11/20/2018  10:44 PM  <DIR>      app
11/20/2018  10:44 PM  <DIR>      bin
11/20/2018  10:44 PM  <DIR>      config
11/20/2018  10:44 PM                130 config.ru
11/20/2018  10:44 PM  <DIR>      db
11/20/2018  10:44 PM            1,750 Gemfile
11/20/2018  10:44 PM  <DIR>      lib
11/20/2018  10:44 PM  <DIR>      log
11/20/2018  10:44 PM                62 package.json
11/20/2018  10:44 PM  <DIR>      public
11/20/2018  10:44 PM                227 Rakefile
11/20/2018  10:44 PM                374 README.md
11/20/2018  10:44 PM  <DIR>      test
11/20/2018  10:44 PM  <DIR>      tmp
11/20/2018  10:44 PM  <DIR>      vendor
               6 File(s)                3,079 bytes
              12 Dir(s) 395,681,132,544 bytes free

C:\Sites\blog>
```

- `bundle install` (για επιβεβαίωση τυχόν ελλειπών gems)

Ρύθμιση για windows 10 (2018)

Μετά την εγκατάσταση του rails και τη δημιουργία της εφαρμογής σας (π.χ. blog)

Παρακαλώ προσέξτε την τροποποίηση που απαιτείται στο gem για τα windows:

στο αρχείο: c:\sites\blog\Gemfile

- # Use CoffeeScript for .coffee assets and views
gem 'coffee-rails', '~> 4.2' # <-- Αυτή η εγγραφή υπάρχει ήδη
gem 'coffee-script-source', '1.8.0' # <-- Προσθέστε την εγγραφή/γραμμή
- Διαγράψτε τυχόν αρχείο Gemfile.lock αν υπάρχει ήδη

Πύθμιση για sqlite3 if failed

Rails tries to automatically install the latest version of the sqlite3 gem (1.4), but that new version doesn't work with Rails yet. Specify a version that works like this:

Gemfile

```
gem 'sqlite3', '< 1.4'
```

And re-run bundle install.

Ρύθμιση για sass-rails

Gemfile

```
gem 'sass-rails', '= 5.0.7'
```

And re-run bundle install.

selenium

Disbale selenium libraries

```
#group :test do#  
# Adds support for Capybara system testing and selenium  
driver  
# gem 'capybara', '>= 2.15'# gem 'selenium-webdriver'#  
# Easy installation and use of chromedriver to run system  
tests with Chrome  
# gem 'chromedriver-helper'  
#end
```

Nio4r για ruby <2.4

Για συμβατό nio4r επιπλέον προσθέστε επίσης στο Gemfile του φακέλου του blog που ξεκινήσατε τη γραμμή:

```
gem 'nio4r', '=2.3.1'
```

Επιβεβαίωση του αρχείου application.html.erb

Επιβεβαιώστε ότι το αρχείο application.html.erb είναι το εξής:

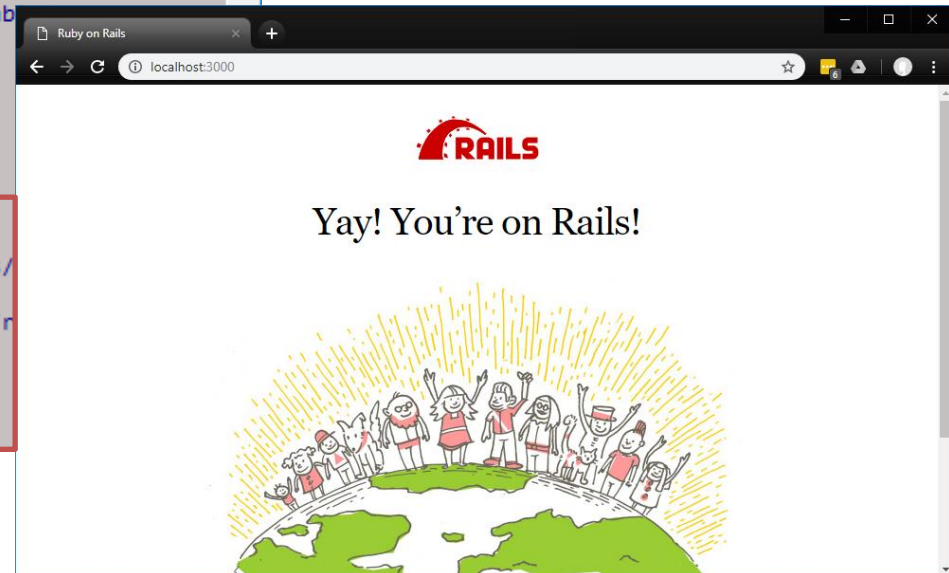
```
<!DOCTYPE html><html> <head>
<title>Blog</title>
<%= csrf_meta_tags %>
<%= csp_meta_tag %>
<%= stylesheet_link_tag 'application', media: 'all', 'data-turbolinks-
track': 'reload' %>
<%= javascript_include_tag 'application', 'data-turbolinks-track':
'reload' %>
</head> <body>
<%= yield %> </body></html>
```

Web Server READY!

- `ruby bin\rails server`

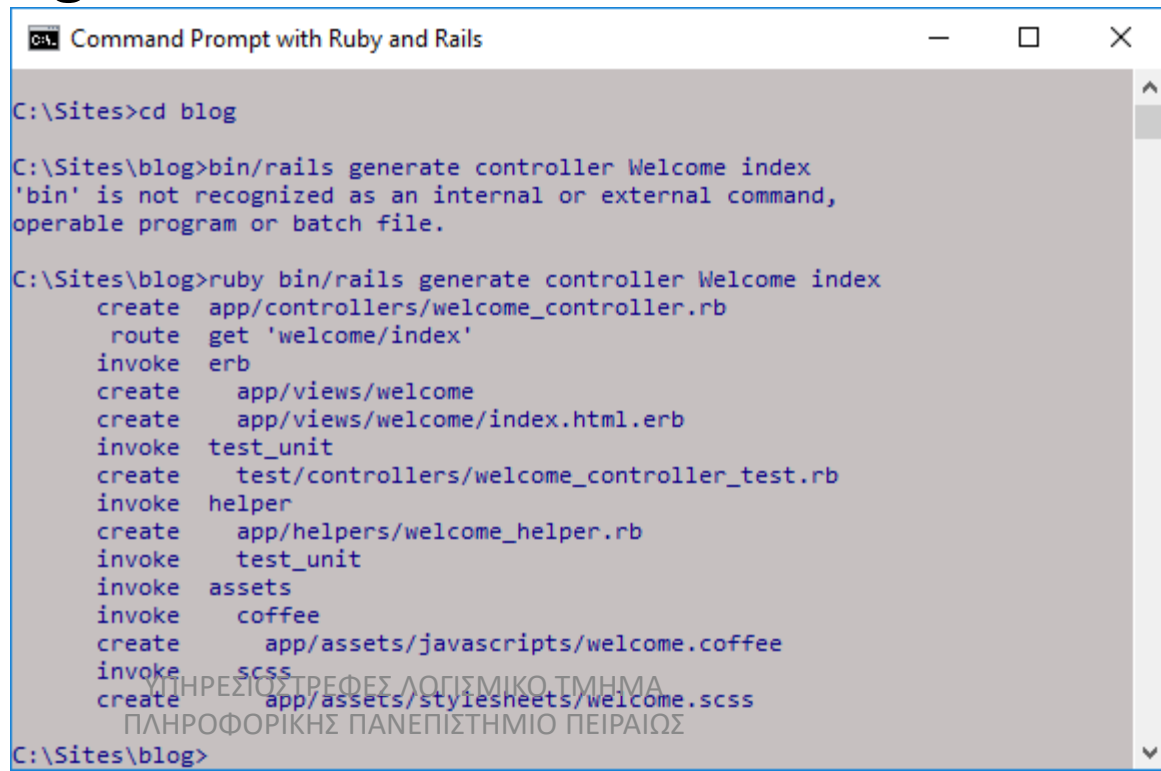
```
C:\Sites\blog> ruby bin\rails server
http://sass.logdown.com/posts/7081811

C:\Sites\blog> ruby bin\rails server
=> Booting Puma
=> Rails 5.1.6 application starting in development
=> Run `rails server -h` for more startup options
*** SIGUSR2 not implemented, signal based restart unavailable!
*** SIGUSR1 not implemented, signal based restart unavailable!
*** SIGHUP not implemented, signal based logs reopening unavailable!
Puma starting in single mode...
* Version 3.12.0 (ruby 2.3.3-p222), codename: Llamas in Pajamas
* Min threads: 5, max threads: 5
* Environment: development
* Listening on tcp://0.0.0.0:3000
Use Ctrl-C to stop
Started GET "/" for 127.0.0.1 at 2018-11-20 23:01:02 +0200
Processing by Rails::WelcomeController#index as HTML
  Rendering C:/RailsInstaller/Ruby2.3.3/lib/ruby/gems/2.3.0/gems/rails-5.1.6/lib/rails/templates/rails/welcome/index.html.erb
  Rendered C:/RailsInstaller/Ruby2.3.3/lib/ruby/gems/2.3.0/gems/rails-5.1.6/lib/rails/templates/rails/welcome/index.html.erb (5.0ms)
Completed 200 OK in 368ms (Views: 49.8ms)
```



controller

- Δημιουργία controller Welcome στο Index
- (παραμένουμε εντός του φακέλου blog)
- `ruby bin/rails generate controller Welcome index`



```
Command Prompt with Ruby and Rails

C:\Sites>cd blog

C:\Sites\blog>bin/rails generate controller Welcome index
'bin' is not recognized as an internal or external command,
operable program or batch file.

C:\Sites\blog>ruby bin/rails generate controller Welcome index
create  app/controllers/welcome_controller.rb
route   get 'welcome/index'
invoke  erb
create  app/views/welcome
create  app/views/welcome/index.html.erb
invoke  test_unit
create  test/controllers/welcome_controller_test.rb
invoke  helper
create  app/helpers/welcome_helper.rb
invoke  test_unit
invoke  assets
invoke  coffee
create  app/assets/javascripts/welcome.coffee
invoke  scss
create  app/assets/stylesheets/welcome.scss

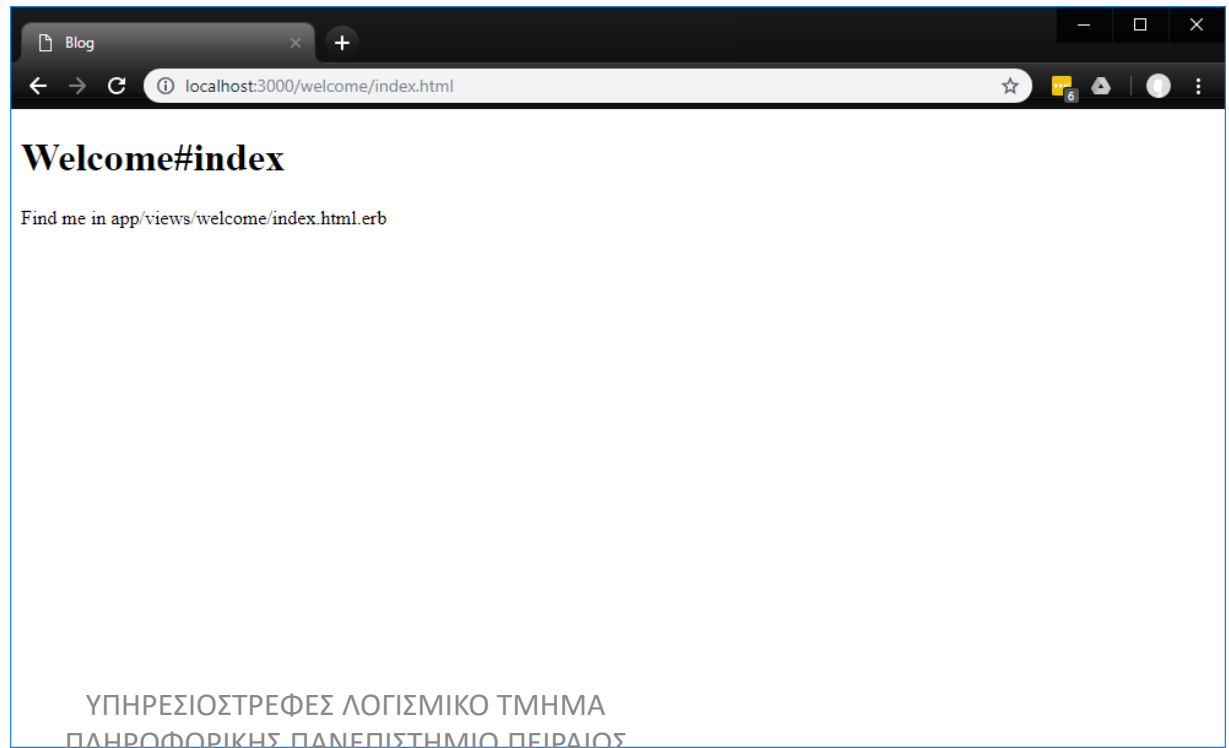
C:\Sites\blog>
```

ΥΠΗΡΕΣΙΟΓΡΑΦΕΣ ΛΟΓΙΣΜΙΚΟΥ ΤΜΗΜΑ
ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Προσθήκη HTML στο view

- `app/views/welcome/index.html.erb`

`<h1>Hello, Rails!</h1>`

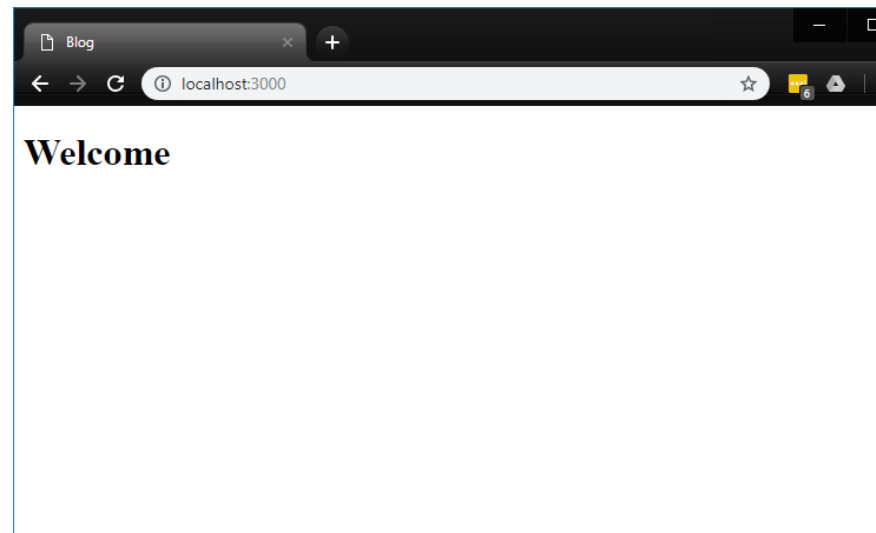


Ορισμός αρχικής σελίδας site

- Ανοίγουμε config/routes.rb

Rails.application.routes.draw do
 get 'welcome/index'

root 'welcome#index'
end



Routing

Routing:

- [Rails Routing from the Outside In.](#)

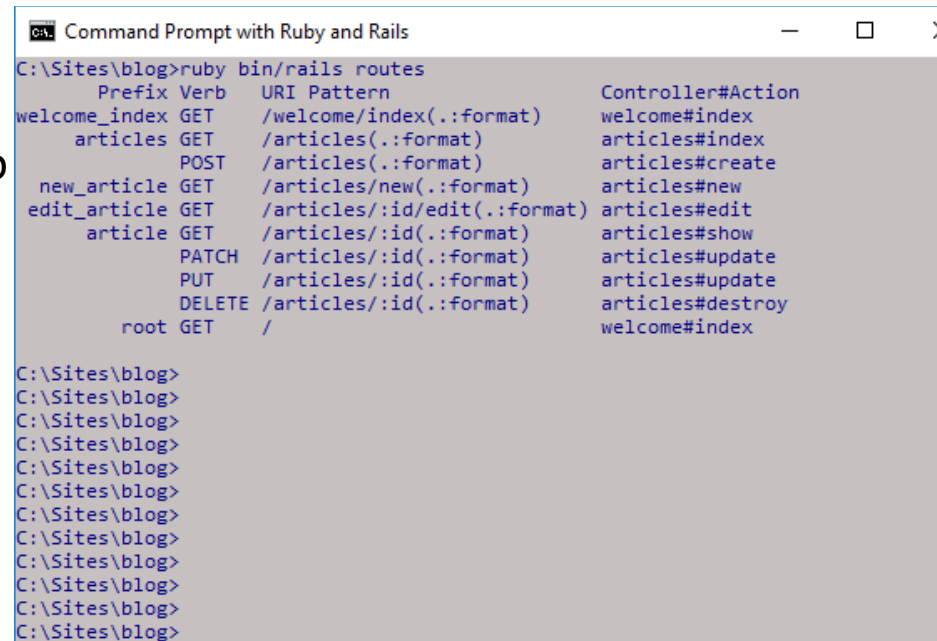
Νέο Resource

- Resource μπορεί να είναι ένα άρθρο
- Routes.rb

```
Rails.application.routes.draw do  
  get 'welcome/index'
```

```
  resources :articles
```

```
  root 'welcome#index'  
end
```



```
Command Prompt with Ruby and Rails  
C:\Sites\blog>ruby bin/rails routes  
Prefix Verb URI Pattern Controller#Action  
welcome_index GET /welcome/index(.:format) welcome#index  
articles GET /articles(.:format) articles#index  
articles POST /articles(.:format) articles#create  
new_article GET /articles/new(.:format) articles#new  
edit_article GET /articles/:id/edit(.:format) articles#edit  
article GET /articles/:id(.:format) articles#show  
article PATCH /articles/:id(.:format) articles#update  
article PUT /articles/:id(.:format) articles#update  
article DELETE /articles/:id(.:format) articles#destroy  
root GET / welcome#index  
C:\Sites\blog>  
C:\Sites\blog>  
C:\Sites\blog>  
C:\Sites\blog>  
C:\Sites\blog>  
C:\Sites\blog>  
C:\Sites\blog>  
C:\Sites\blog>  
C:\Sites\blog>  
C:\Sites\blog>  
C:\Sites\blog>  
C:\Sites\blog>
```

- `ruby bin\rails routes`

Δημιουργία του C & R

New Article

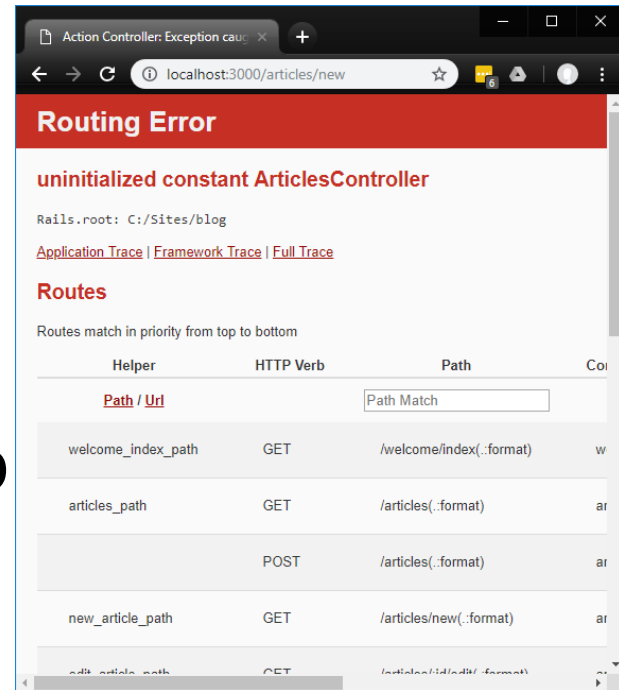
Title

Text

Save Article

New

- create a new article.
- Έστω ότι θέλουμε να πάμε από το `/articles/new`.
- Με το route ορισμένο
- <http://localhost:3000/articles/new>



Δημιουργία του controller για το new

- `ruby bin/rails generate controller Articles`



```
Command Prompt with Ruby and Rails
C:\Sites\blog>ruby bin/rails generate controller Articles
create  app/controllers/articles_controller.rb
invoke  erb
create  app/views/articles
invoke  test_unit
create  test/controllers/articles_controller_test.rb
invoke  helper
create  app/helpers/articles_helper.rb
invoke  test_unit
invoke  assets
invoke  coffee
create  app/assets/javascripts/articles.coffee
invoke  scss
create  app/assets/stylesheets/articles.scss
C:\Sites\blog>
```

Articles controller

- Edit:
- app/controllers/articles_controller.rb
- Θα δείτε:

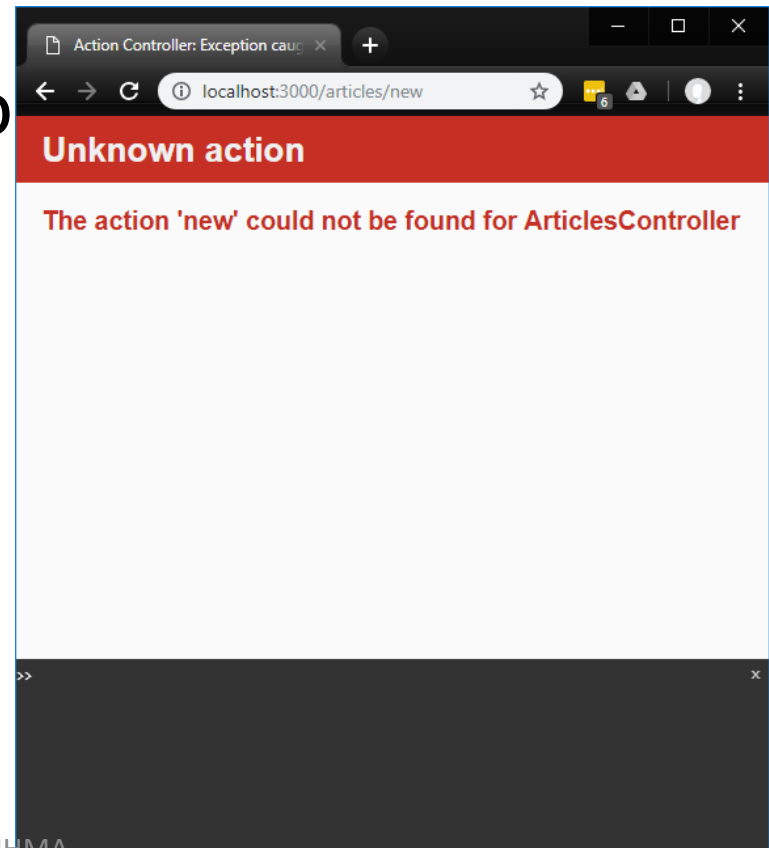
```
class ArticlesController < ApplicationController  
end
```

Για ανάλυση της Ruby: [Programming Ruby](#)

No new action!

- Για να διορθώσετε το
Λάθος δεξιά γράφετε εντός το
`app/controllers/
articles_controller.rb`

```
class ArticlesController < ApplicationController  
  def new  
  end  
end
```

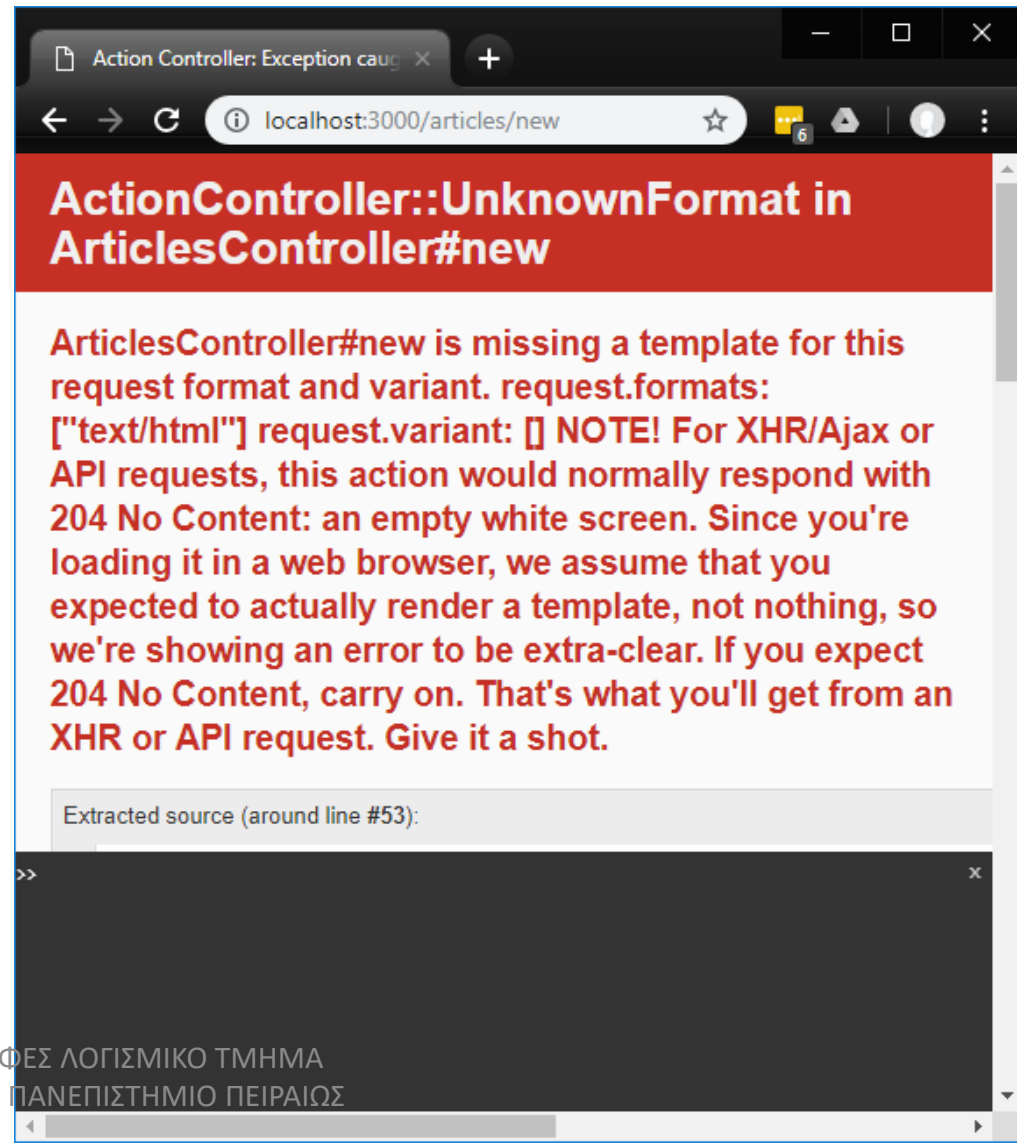


Δεν υπάρχει View

- Νέο αρχείο

app/views/articles/new.html.erb

<h1>New Article</h1>



Πλήρης Φόρμα New

```
<%= form_with scope: :article, local: true do |form| %>
  <p>
    <%= form.label :title %><br>
    <%= form.text_field :title %>
  </p>

  <p>
    <%= form.label :text %><br>
    <%= form.text_area :text %>
  </p>

  <p>
    <%= form.submit %>
  </p>
<% end %>
```

The image displays two browser screenshots of a web application running on localhost:3000/articles/new. The top screenshot shows the 'New Article' heading and a single text input field for the title. The bottom screenshot shows the same page but with an additional text area for the article content and a 'Save Article' button at the bottom.

Σελίδα html source

```
<form action="/articles/new" accept-charset="UTF-8" method="post">
```

```
<input name="utf8" type="hidden" value="✓"><input type="hidden" name="authenticity_token"
value="GYe9414DWIm6BNmDPEHPMD/1q7Ow2t7++/w4IU2HQJ2XjUd5zxDgp5cUEdLjtmruOmlEbQeHCSlldrwriTKg6g=="
">
```

```
&nbsp;&nbsp;&nbsp;<p>
```

```
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<label for="article_title">Title</label><br>
```

```
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="text" name="article[title]">
```

```
&nbsp;&nbsp;&nbsp;</p>
```

```
&nbsp;
```

```
&nbsp;&nbsp;&nbsp;<p>
```

```
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<label for="article_text">Text</label><br>
```

```
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<textarea name="article[text]"></textarea>
```

```
&nbsp;&nbsp;&nbsp;</p>
```

```
&nbsp;
```

```
&nbsp;&nbsp;&nbsp;<p>
```

```
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="submit" name="commit" value="Save Article" data-disable-with="Save
Article">
```

```
&nbsp;&nbsp;&nbsp;</p>
```

```
</form>
```

Μετάβαση σε άλλη σελίδα

Αντικαταστήστε:

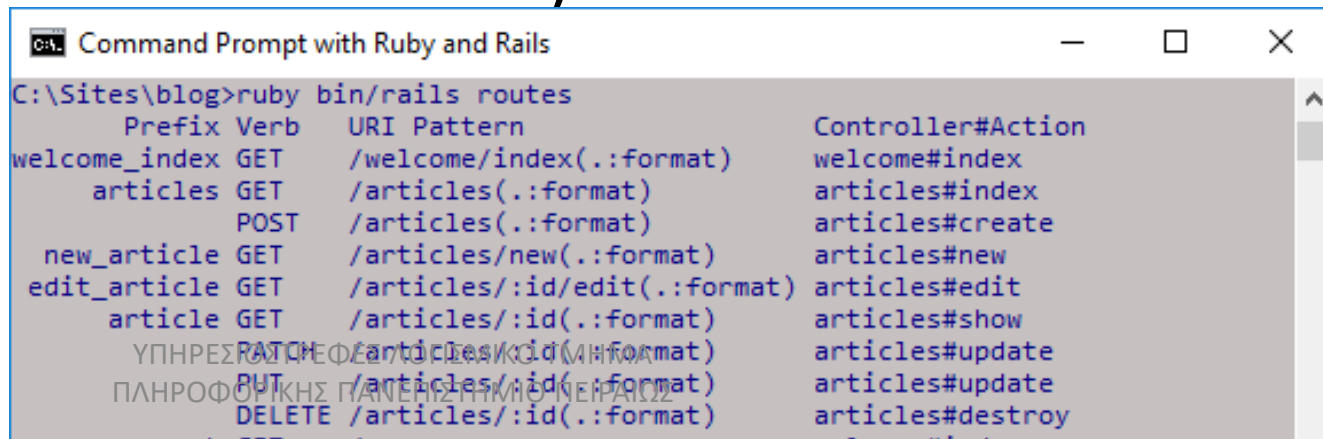
```
<%= form_with scope: :article, local: true do  
|form| %>
```

→ (με το)

```
<%= form_with scope: :article, url: articles_path,  
local: true do |form| %>
```

Rails routes

- `ruby bin/rails routes`
- το `articles_path` helper λέει στο Rails να πάει την φόρμα στο URI Pattern που συσχετίζεται με το πρόθεμα `articles`
- Η φόρμα θα (by default) στείλει ένα POST request σε αυτό το route.
- Αυτό συσχετίζεται με το `create` action του current controller (`ArticlesController`).



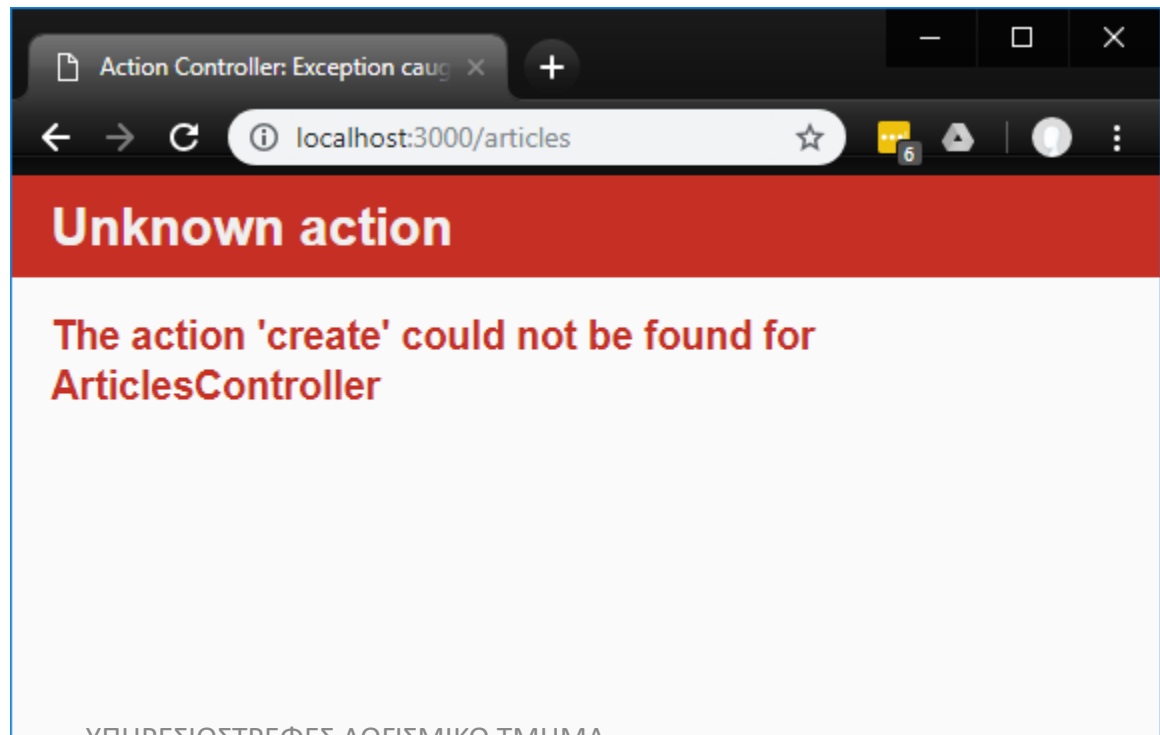
```
Command Prompt with Ruby and Rails

C:\Sites\blog>ruby bin/rails routes

  Prefix Verb   URI Pattern               Controller#Action
welcome_index GET    /welcome/index(.:format) welcome#index
articles GET    /articles(.:format)      articles#index
          POST   /articles(.:format)      articles#create
new_article GET    /articles/new(.:format)  articles#new
edit_article GET    /articles/:id/edit(.:format) articles#edit
article GET    /articles/:id(.:format)  articles#show
          PATCH  /articles/:id(.:format)  articles#update
          PUT    /articles/:id(.:format)  articles#update
          DELETE /articles/:id(.:format)  articles#destroy
```

create create action στο ArticlesController

- By default form_with submits forms Ajax
- disabled local: true



"Unknown action"

- Δημιουργία της ArticlesController class
- Στο αρχείο
app/controllers/articles_controller.rb

```
class ArticlesController < ApplicationController
  def new
  end

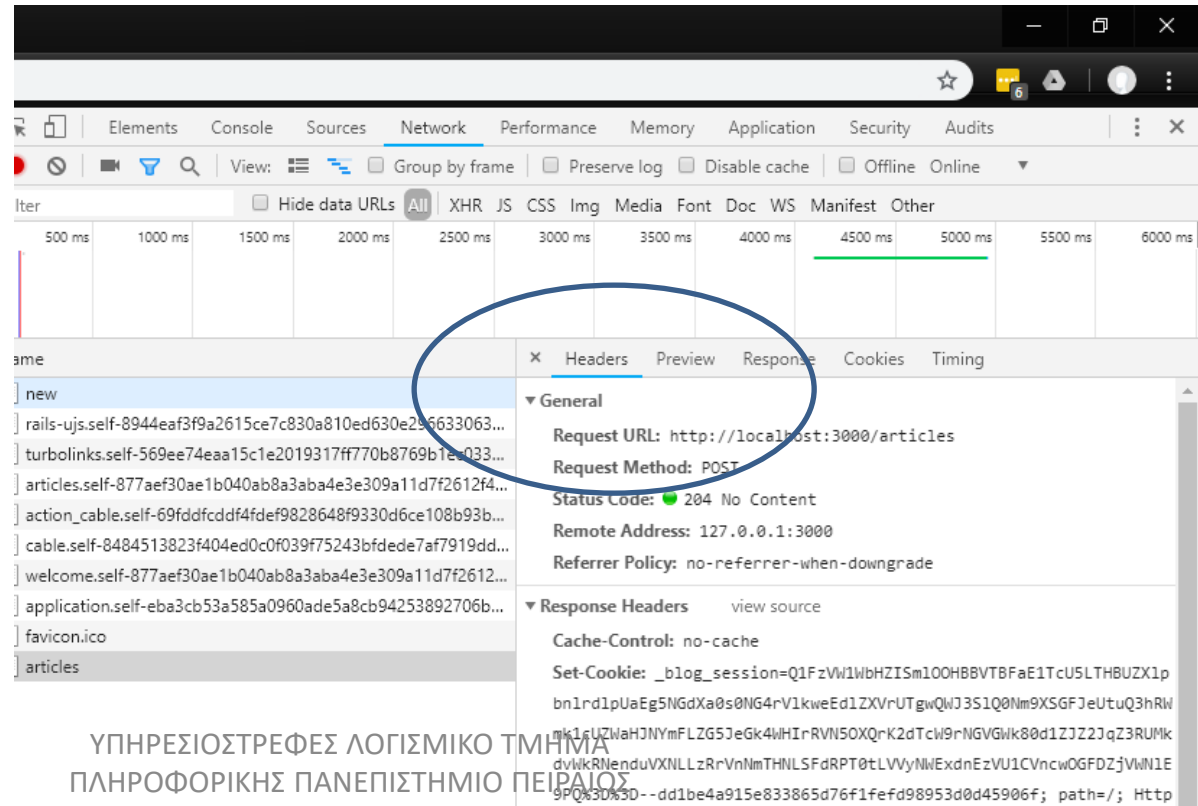
  def create
  end
end
```


Δοκιμή του action!

- <http://localhost:3000/articles/new>
- Save article
- Τι παρατηρείτε;

204 No content Response

- Επιτυχημένο (οικογένεια http code 20x)
- Θα πρέπει τελικά να αποθηκεύει το article



Παράμετροι

- Υποβολή φόρμας: τα πεδία περνούν ως παράμετροι
- Οι παράμετροι μπορούν να εντοπίζονται εντός των ενεργειών του controller
- Γράψτε στην create action :
 - `def create`
 - `render plain: params[:article].inspect`
 - `End`

Παράμετροι (2)

- Εδώ η render method λαμβάνει ένα hash
- Κλειδί :plain
- τιμή params[:article].inspect

Params method είναι ένα object των παραμέτρων της φόρμας

Επιστρέφει έναν ActionController::Parameters object

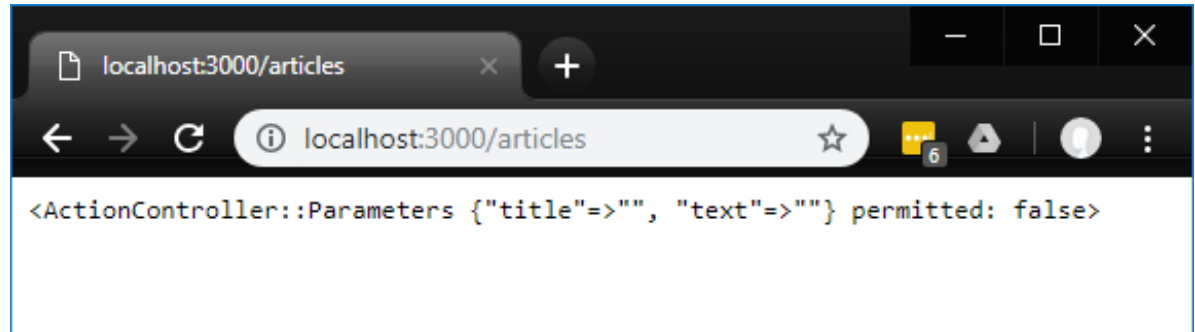
Με το οποίο προσπελάζουμε τα κλειδιά του hash με strings or symbols.

app/controllers/articles_controller.rb

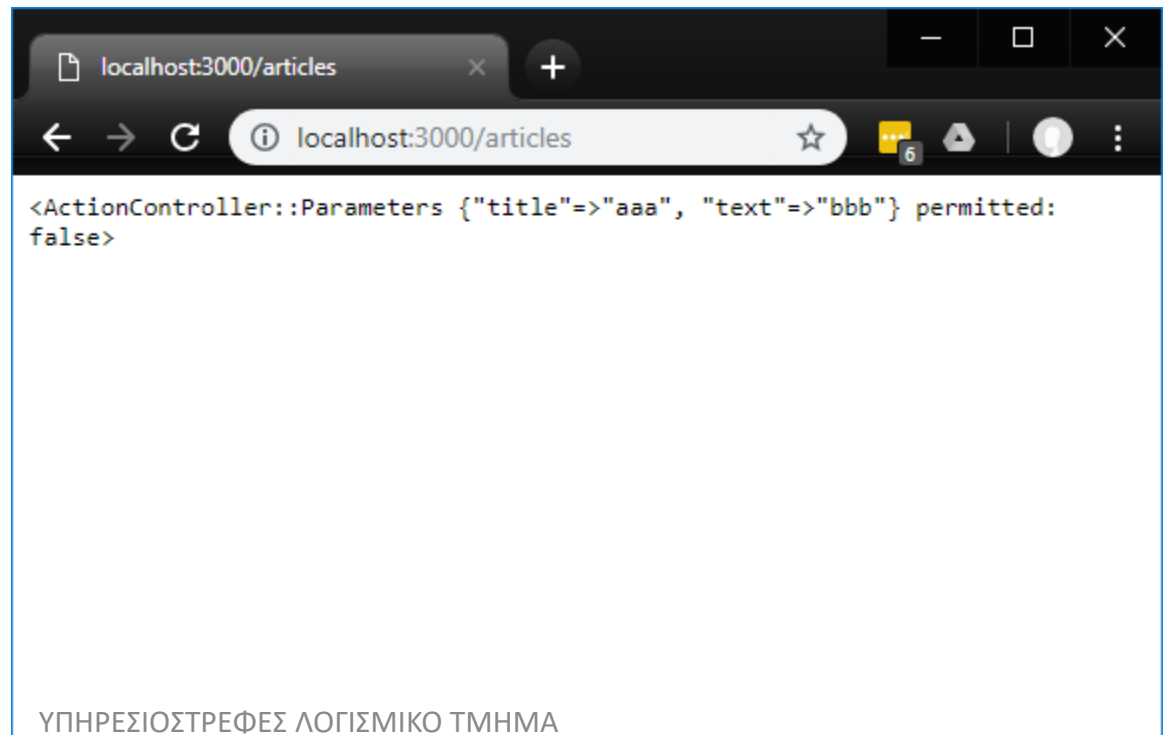
```
class ArticlesController < ApplicationController
  def new
  End
  def create
  render plain: params[:article].inspect
  End
end
```

Δοκιμή με παραμέτρους

- Χωρίς τιμές:



- Με τιμές:



Παράμετροι (3)

- params method πολύ σημαντική!

- URL:

www.example.com/?username=dhh&email=dhh@email.com

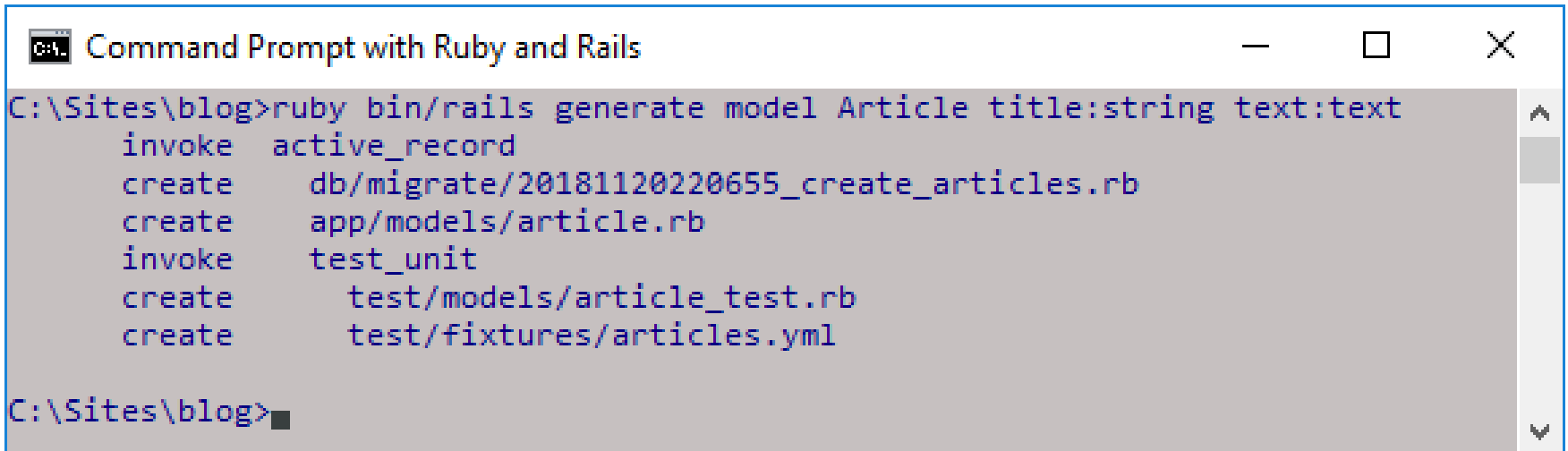
params[:username] → "dhh"

params[:email] → "dhh@email.com".

Μοντέλο

- Τα μοντέλο στο rails τα ονομάζουμε στον ενικό
- Τους πίνακες του μοντέλου τους ονομάζουμε με πληθυντικό αριθμό
- (επομένως ένα Article model)
- ***Ruby bin/rails generate model Article title:string text:text***

Δημιουργία Μοντέλου



```
Command Prompt with Ruby and Rails

C:\Sites\blog>ruby bin/rails generate model Article title:string text:text
  invoke  active_record
  create   db/migrate/20181120220655_create_articles.rb
  create   app/models/article.rb
  invoke   test_unit
  create   test/models/article_test.rb
  create   test/fixtures/articles.yml

C:\Sites\blog>
```

- Δημιουργήθηκε model με ένα *database migration* file στο db/migrate directory

Αρχείο Μοντέλου

- Ανοίξτε το αρχείο

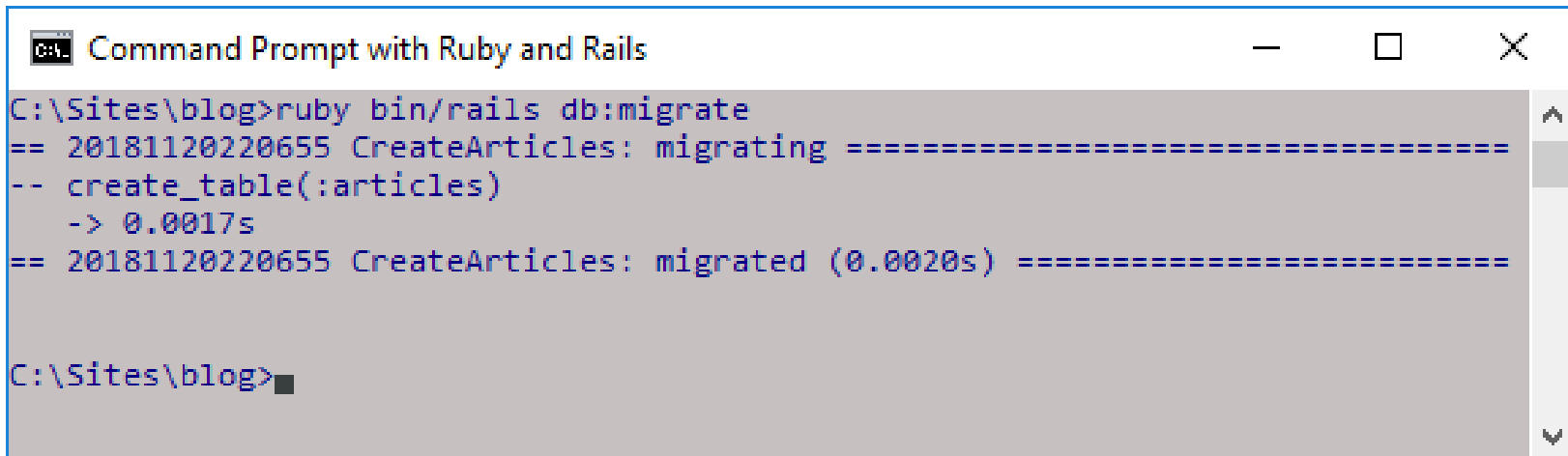
db/migrate/YYYYMMDDHHMMSS_create_articles.rb

```
class CreateArticles < ActiveRecord::Migration[5.1]  
  def change  
    create_table :articles do |t|  
      t.string :title  
      t.text :text  
      t.timestamps  
    end  
  end  
end
```

Δείτε για περισσότερα στο: [Active Record Migrations](#)

Κατασκευή Μοντέλου

- `ruby bin/rails db:migrate`



```
Command Prompt with Ruby and Rails

C:\Sites\blog>ruby bin/rails db:migrate
== 20181120220655 CreateArticles: migrating =====
-- create_table(:articles)
   -> 0.0017s
== 20181120220655 CreateArticles: migrated (0.0020s) =====

C:\Sites\blog>
```

ανάπτυξη/ παραγωγή

- Για την ανάπτυξη η ΒΔ που χρησιμοποιείται ορίζεται στο
- `config/database.yml`
- Για αλλού περάστε σχετική παράμετρο
`bin/rails db:migrate RAILS_ENV=production`

Αποθήκευση δεδομένων στον controller

- Αλλαγή του create στον


app/controllers/articles_controller.rb

```
def create
  @article = Article.new(params[:article])

  @article.save
  redirect_to @article
end
```

Αποθήκευση δεδομένων στον controller

- Κάθε Rails model μπορεί να αρχικοποιείται με τις αντίστοιχες παραμέτρους
- Που αντιστοιχίζονται σε στήλες της ΒΔ
- Στην πρώτη γραμμή αυτό συμβαίνει (params[:article] περιέχει τις παραμέτρους)

```
def create
   ΚΛΑΣΗ
  @article = Article.new(params[:article])

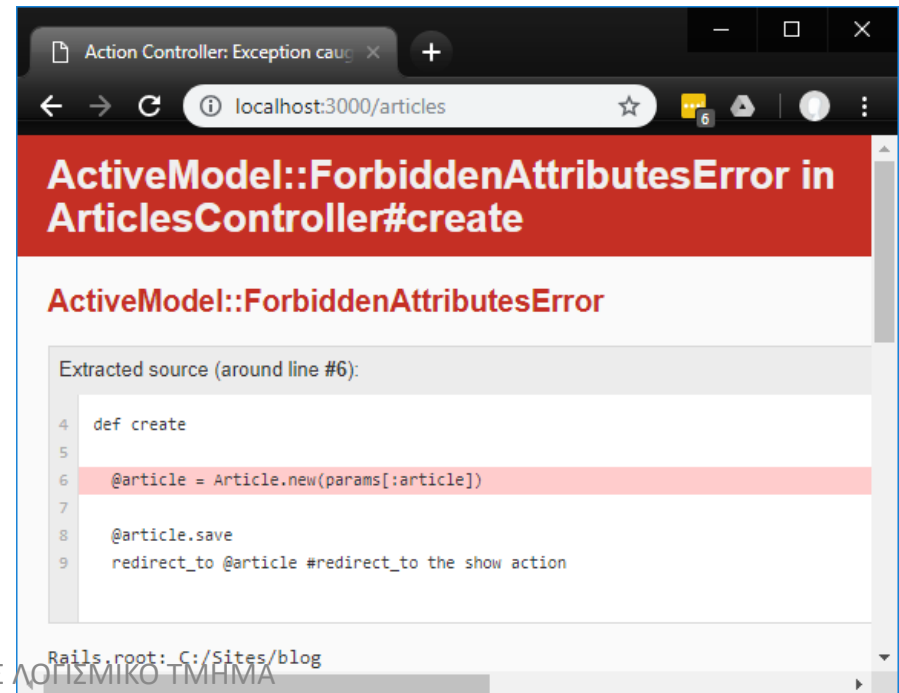
  @article.save #αποθηκεύει στη ΒΔ
  redirect_to @article #επιστρέφει στο show
end
```

Δοκιμή αποθήκευσης (1)

- [strong parameters](#)
- Καθορισμός ποιες ακριβώς παραμέτρους πρέπει να περαστούν στον controller

- require
- Permit

[Περισσότερα για Strong Parameters.](#)



The screenshot shows a web browser window with the address bar displaying 'localhost:3000/articles'. The page content is a red error banner with the text 'ActiveModel::ForbiddenAttributesError in ArticlesController#create'. Below the banner, the error details are shown, including the extracted source code around line #6:

```
4 def create
5
6   @article = Article.new(params[:article])
7
8   @article.save
9   redirect_to @article #redirect to the show action
```

At the bottom of the browser window, the Rails root path is indicated as 'C:/Sites/blog'.

Δοκιμή αποθήκευσης (1)

Αλλαγή του articles_controller.rb →

```
def create
  @article = Article.new(article_params)

  @article.save
  redirect_to @article
end

private
def article_params
  params.require(:article).permit(:title,
:text)
end
```


Δοκιμή αποθήκευσης (2)

- Δεν υπάρχει η show όπου γίνεται redirect
- Τι παρατηρείτε στο URL ;;;



Εμφάνιση άρθρων

- `ruby bin/rails routes:`
- Παρατηρείστε τη γραμμή:

`article GET /articles/:id(.:format) articles#show`

- ΠΡΟΣΟΧΗ περιλαμβάνει **`:id`**
- Επομένως αλλαγή στο `articles_controller.rb` να περιλαμβάνει **`SHOW`**

Προβολή στον controller

```
def show
  @article = Article.find(params[:id])
end
```

- `@article` (μεταβλητή στιγμιοτύπου) το Rails θα περάσει όλες τις μεταβλητές στιγμιοτύπου στο View
- `Article.find` εντοπίζει το άρθρο περνώντας παράμετρο στο `params[:id]` για να λάβει το `:id` από το request

!!Σειρά: index, show, new, edit, create, update and destroy.

Show.html.erb

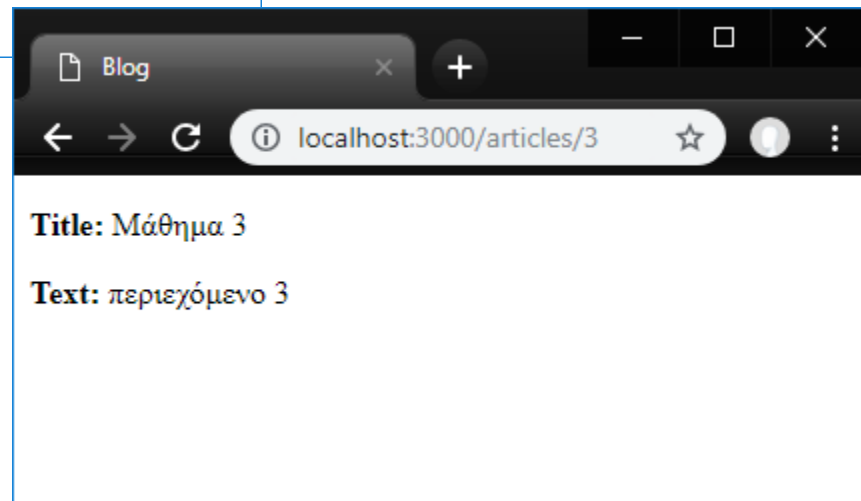
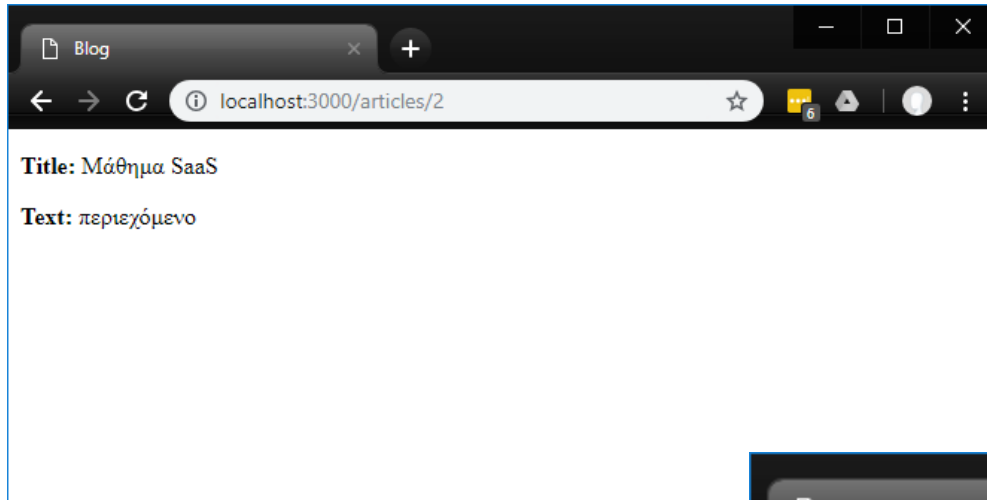
- Δημιουργία αρχείου

app/views/articles/show.html.erb

```
<p>  
  <strong>Title:</strong>  
  <%= @article.title %>  
</p>
```

```
<p>  
  <strong>Text:</strong>  
  <%= @article.text %>  
</p>
```

Δημιουργία εγγραφών articles!



Αλλαγή παραμέτρου στο URL

- Δοκιμάστε να αλλάξετε το id στο URL
- Θα προσπελάσετε το αντίστοιχο άρθρο από τη ΒΔ!!

Προβολή όλων των άρθρων

- `ruby bin/rails routes:`
- Παρατηρείστε τη γραμμή:

`articles GET /articles(.:format) articles#index`

- Προσθήκη στον controller του article: (πρώτο στη σειρά): (προσοχή πληθυντικός στην παράμετρο!)

```
def index
  @articles = Article.all
end
```

Προβολή για το index

- Δημιουργία

app/views/articles/index.html.erb

```
<h1>Listing articles</h1>
```

```
<table>
```

```
<tr>
```

```
<th>Title</th>
```

```
<th>Text</th>
```

```
<th></th>
```

```
</tr>
```

```
<% @articles.each do |article| %>
```

```
<tr>
```

```
<td><%= article.title %></td>
```

```
<td><%= article.text %></td>
```

```
<td><%= link_to 'Show', article_path(article) %></td>
```

```
</tr>
```

```
<% end %>
```

```
</table>
```



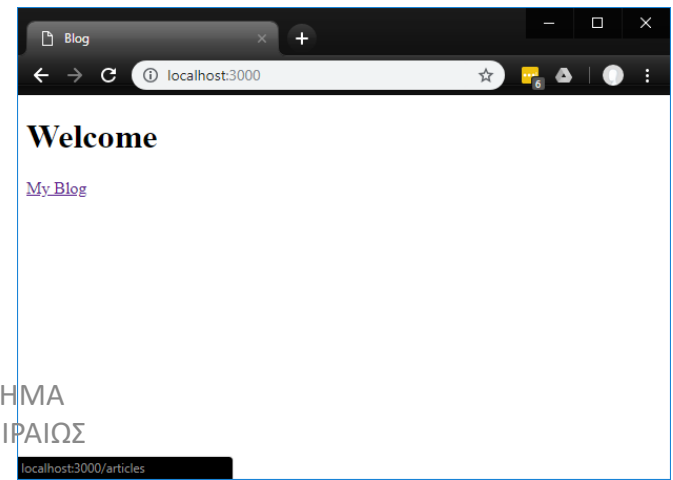
Προσθήκη links

- Στο αρχείο

app/views/welcome/index.html.erb

```
<h1>Hello, Rails!</h1>
```

```
<%= link_to 'My Blog', controller: 'articles' %>
```



Προσθήκη links (2)

- Στο αρχείο
app/views/articles/index.html.erb

`<%= link_to 'New article', new_article_path %>`

!!! Για σύνδεσμο στον ίδιο controller, δεν απαιτείται το :controller option
χρησιμοποιείται το τρέχον

`<%= link_to 'My Blog', controller: 'articles' %>`



Προσθήκη links (3)

- Στο αρχείο

app/views/articles/new.html.erb

```
<%= link_to 'Back', articles_path %>
```



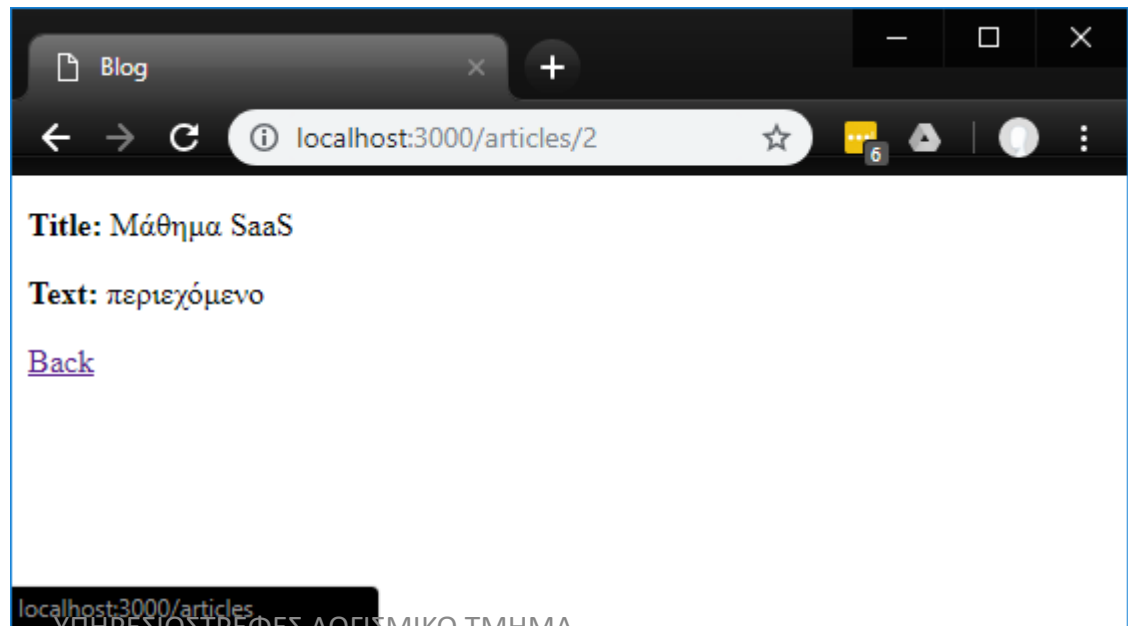
A screenshot of a web browser window. The address bar shows 'localhost:3000/articles/new'. The page title is 'New Article'. The form contains two input fields: 'Title' and 'Text'. Below the 'Text' field is a 'Save Article' button. At the bottom of the form, there is a 'Back' link.

Προσθήκη links (4)

- Στο αρχείο

app/views/articles/show.html.erb

```
<%= link_to 'Back', articles_path %>
```



Έλεγχοι

- app/models/article.rb

```
class Article < ApplicationRecord
  validates :title, presence: true,
                length: { minimum: 5 }
end
```

- Τίτλος τουλάχιστον 5 χαρακτήρων
- Όλοι οι έλεγχοι παρουσιάζονται στο [Active Record Validations](#).

Έλεγχοι

- Με τους ελέγχους όταν πραγματοποιούνται αν τυχόν αποτύχουν επιστρέφουν false

Παράδειγμα:

```
@article.save
```

```
  redirect_to @article
```

```
  → (μετατροπή για χρήση του true/false
```

```
  if @article.save
```

```
    redirect_to @article
```

```
  else
```

```
    render 'new'
```

```
  end
```

Έλεγχοι

- Το πρότυπο του new επιστρέφεται αν γίνει false
- Αν δοκιμάσουμε την

<http://localhost:3000/articles/new>

Στην περίπτωση λάθους τιτλου < 5 χαρακτήρες

Επιστροφή στην new (δεν έχει μήνυμα λάθους)

Έλεγχοι

- Ενεργοποίηση μηνυμάτων λαθών
- app/views/articles/new.html.erb

Προσθήκη στη φόρμα :

```
<% if @article.errors.any? %>
  <div id="error_explanation">
    <h2>
      <%= pluralize(@article.errors.count, "error") %> prohibited
      this article from being saved:
    </h2>
    <ul>
      <% @article.errors.full_messages.each do |msg| %>
        <li><%= msg %></li>
      <% end %>
    </ul>
  </div>
<% end %>
```


Έλεγχοι



Blog

localhost:3000/articles

New Article

1 error prohibited this article from being saved:

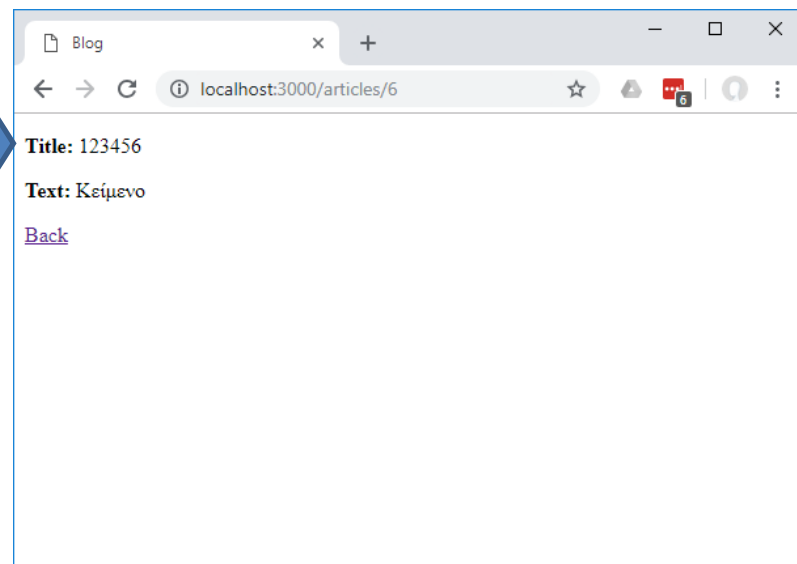
- Title is too short (minimum is 5 characters)

Title

Text

Save Article

[Back](#)



Blog

localhost:3000/articles/6

Title: 123456

Text: Κείμενο

[Back](#)

Έλεγχοι

- Έλεγχος αν υπάρχουν λάθη
`@article.errors.any?`
- Αν υπάρχουν μπορείτε να παρουσιάζετε μία λίστα όλων των λαθών
`@article.errors.full_messages`

Έλεγχοι

- Έγινε προσθήκη του
- `@article = Article.new`

Στον `ArticlesController`

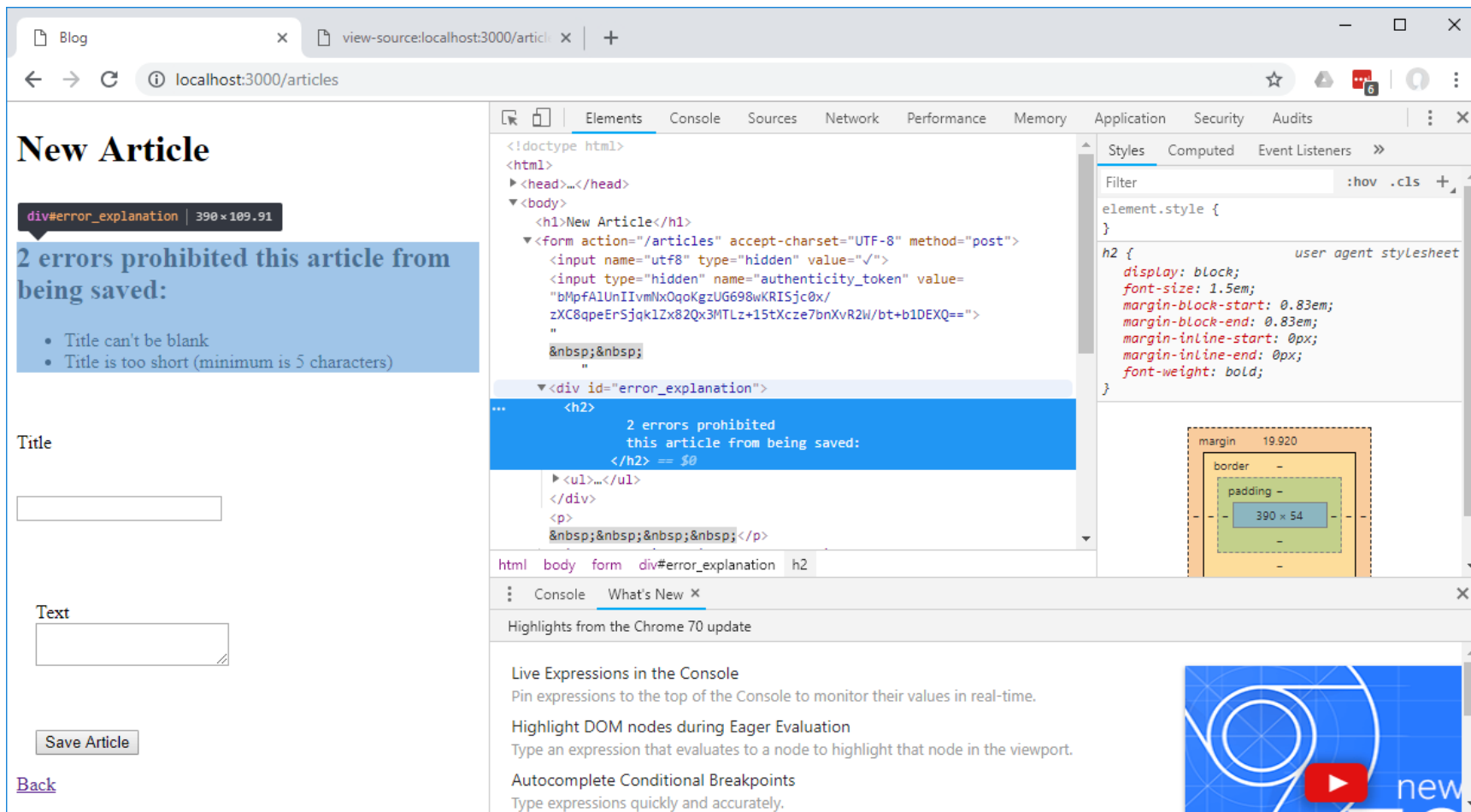
Αλλιώς το `@article` θα ήταν `nil`

Και η `@article.errors.any?` δε θα δράσει

CSS και Έλεγχοι

- `<div id="error_explanation">`
- `<h2>`
- 2 errors prohibited
- this article from being saved:
- `</h2>`
- ``
- `Title can't be blank`
- `Title is too short (minimum is 5`
characters)``
- ``
- `</div>`

CSS και Έλεγχοι



Ενημέρωση Άρθρου

- Ολοκληρώθηκε το CRUD
- Τώρα προχωρούμε για την ενημέρωση Update
- Προσθήκη edit action στον ArticlesController (μετά το new και πριν το create)
app/views/articles/edit.html.erb

Ενημέρωση (Edit)

```
<h1>Edit article</h1>
<%= form_with(model: @article, local: true) do |form| %>
  <% if @article.errors.any? %>
    <div id="error_explanation">
      <h2>
        <%= pluralize(@article.errors.count, "error") %> prohibited
        this article from being saved:
      </h2>
      <ul>
        <% @article.errors.full_messages.each do |msg| %>
          <li><%= msg %></li>
        <% end %>
      </ul>
    </div>
  <% end %>
  <p>
    <%= form.label :title %><br>
    <%= form.text_field :title %> </p>
  <p>  <%= form.label :text %><br>
    <%= form.text_area :text %>
  </p> <p>
    <%= form.submit %>
  </p>
<% end %>
<%= link_to 'Back', articles_path %>
```

Ενημέρωση (Edit) Controller

- update action
στο `app/controllers/articles_controller.rb`

```
def edit
```

```
  @article = Article.find(params[:id])
```

```
end
```


Ενημέρωση (Edit) Controller με το πέραςμα παραμέτρων

```
def create
  @article = Article.new(article_params)

  if @article.save
    redirect_to @article
  else
    render 'new'
  end
end
```

```
def update
  @article = Article.find(params[:id])

  if @article.update(article_params)
    redirect_to @article
  else
    render 'edit'
  end
end
```

```
private
def article_params
  params.require(:article).permit(:title, :text)
end
```

Ενημέρωση (Edit) Controller

Update μόνο ένα πεδίο:

```
@article.update(title: 'A new title')
```

Ενημέρωση (Edit) Controller

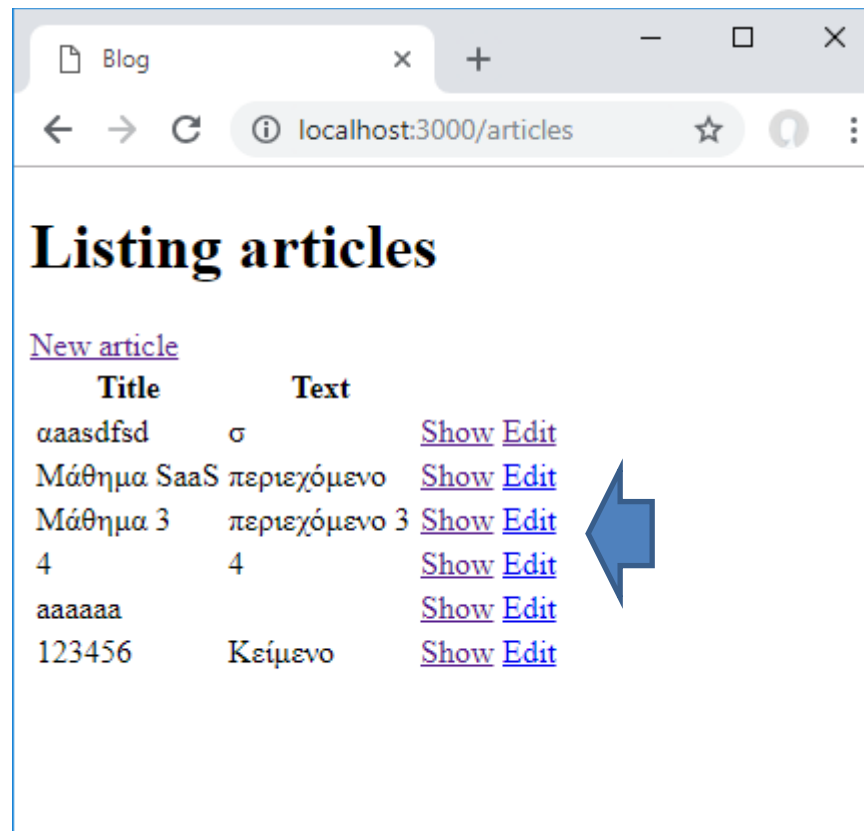
Προσθήκη του edit στη
λίστα των άρθρων!

app/views/articles/index.html.erb

```
<table>
  <tr>
    <th>Title</th>
    <th>Text</th>
    <th colspan="2"></th>
  </tr>

  <% @articles.each do |article| %>
    <tr>
      <td><%= article.title %></td>
      <td><%= article.text %></td>
      <td><%= link_to 'Show',
article_path(article) %></td>
      <td><%= link_to 'Edit',
edit_article_path(article) %></td>
    </tr>
  <% end %>
</table>
```

Ενημέρωση (Edit) Controller



Ενημέρωση (Edit) Controller

- app/views/articles/show.html.erb

Edit:

```
<%= link_to 'Edit', edit_article_path(@article) %>
```

```
|
```

```
<%= link_to 'Back', articles_path %>
```

Views

- Η new και η index έχουν σχεδόν τον ίδιο κώδικα και κάνουν την ίδια λειτουργία.
- View partial ([Layouts and Rendering in Rails](#))

Απλοποίηση του edit

- app/views/articles/_form.html.erb

```
<%= form_with model: @article, local: true do |form| %>
```

```
<% if @article.errors.any? %> <div id="error_explanation">
  <h2>
    <%= pluralize(@article.errors.count, "error") %> prohibited
    this article from being saved:
  </h2>
  <ul>
    <% @article.errors.full_messages.each do |msg| %>
      <li><%= msg %></li>
    <% end %>
  </ul> </div>
<% end %>
<p> <%= form.label :title %><br>
  <%= form.text_field :title %> </p>
<p> <%= form.label :text %><br>
  <%= form.text_area :text %> </p>
<p> <%= form.submit %> </p>
<% end %>
```

Update του edit

- update the app/views/articles/new.html.erb

```
<h1>New article</h1>
```

```
<%= render 'form' %>
```

```
<%= link_to 'Back', articles_path %>
```


Διαγραφή

- "D" του (delete) CRUD

bin/rails routes

DELETE /articles/:id(.:format) articles#destroy

`look at this cat!`

app/controllers/articles_controller.rb

def destroy

@article = Article.find(params[:id])

@article.destroy

redirect_to articles_path

end

Διαγραφή

'Destroy' link : app/views/articles/index.html.erb

```
<h1>Listing Articles</h1>
<%= link_to 'New article', new_article_path %>
<table>
  <tr>
    <th>Title</th>
    <th>Text</th>
    <th colspan="3"></th>
  </tr>

  <% @articles.each do |article| %>
    <tr>
      <td><%= article.title %></td>
      <td><%= article.text %></td>
      <td><%= link_to 'Show', article_path(article) %></td>
      <td><%= link_to 'Edit', edit_article_path(article) %></td>
      <td><%= link_to 'Destroy', article_path(article),
        method: :delete,
        data: { confirm: 'Are you sure?' } %></td>
    </tr>
  <% end %>
</table>
```

Listing Articles

[New article](#)

Title	Text
Rails is awesome! It really is.	Show Edit Destroy

The page at localhost:30...



Are you sure?

Cancel

OK

ΣΧΟΛΙΑ: Επιπλέον Μοντέλο (2^ο)

- `ruby bin/rails generate model Comment
commenter:string body:text article:references`

`app/models/comment.rb:`

```
class Comment < ApplicationRecord  
  belongs_to :article  
end
```

```
C:\Sites\blog>ruby bin/rails generate model Comment commenter:string body:text article:references
  invoke  active_record
  create  db/migrate/20181205084753_create_comments.rb
  create  app/models/comment.rb
  invoke  test_unit
  create  test/models/comment_test.rb
  create  test/fixtures/comments.yml

C:\Sites\blog>
```

- Δημιουργεί το comment.rb (models)
class Comment < ApplicationRecord
belongs_to :article
end

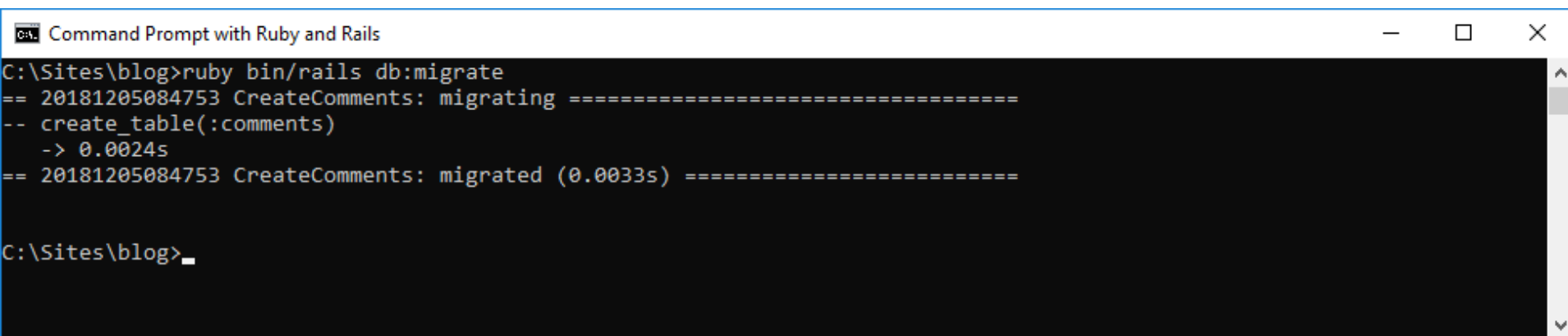
Database Comment Table

```
class CreateComments <
  ActiveRecord::Migration[5.0]
  def change
    create_table :comments do |t|
      t.string :commenter
      t.text :body
      t.references :article, foreign_key: true

      t.timestamps
    end
  end
end
```

Δημιουργία του Comment πίνακα

`ruby bin/rails db:migrate`



```
Command Prompt with Ruby and Rails
C:\Sites\blog>ruby bin/rails db:migrate
== 20181205084753 CreateComments: migrating =====
-- create_table(:comments)
   -> 0.0024s
== 20181205084753 CreateComments: migrated (0.0033s) =====

C:\Sites\blog>_
```

Associations

- Κάθε άρθρο έχει πολλά σχόλια
- Κάθε σχόλιο ανήκει σε ένα ακριβώς άρθρο

Σύνδεση με το σχόλιο

- Τα άρθρα δεν γνωρίζουν το σχόλιο ως τώρα

- app/models/article.rb

```
class Article < ApplicationRecord
```

```
  has_many :comments
```

```
  validates :title, presence: true,
```

```
            length: { minimum: 5 }
```

```
end
```


Περισσότερα για συσχετίσεις

- Active Record Associations

Το route για το comments

- config/routes.rb
- Προσοχή στην ένθεση!

```
resources :articles do  
  resources :comments  
end
```

Πλήρης οδηγός του routing: [Rails Routing](#) guide

Δημιουργία του Comment Controller

- Δημιουργήσαμε το model
- Συνδέσαμε το comments με το articles στο config
- Παράγουμε τον controller που θα χειριστεί όλο το CRUD του comments

`ruby bin/rails generate controller Comments`

Command Prompt with Ruby and Rails

```
C:\Sites\blog>ruby bin/rails generate controller Comments
create  app/controllers/comments_controller.rb
invoke  erb
create  app/views/comments
invoke  test_unit
create  test/controllers/comments_controller_test.rb
invoke  helper
create  app/helpers/comments_helper.rb
invoke  test_unit
invoke  assets
invoke  coffee
create  app/assets/javascripts/comments.coffee
invoke  scss
create  app/assets/stylesheets/comments.scss
```

```
C:\Sites\blog>_
```

Show comments

- app/views/articles/show.html.erb

```
<p>  
  <strong>Title:</strong>  
  <%= @article.title %>  
</p>
```

```
<p>  
  <strong>Text:</strong>  
  <%= @article.text %>  
</p>
```

(συνεχίζεται!)

Show comments (2)

```
<h2>Add a comment:</h2>
```

```
<%= form_with(model: [ @article, @article.comments.build ], local: true) do |form| %>
```

```
<p>
```

```
<%= form.label :commenter %><br>
```

```
<%= form.text_field :commenter %>
```

```
</p>
```

```
<p>
```

```
<%= form.label :body %><br>
```

```
<%= form.text_area :body %>
```

```
</p>
```

```
<p>
```

```
<%= form.submit %>
```

```
</p>
```

```
<% end %>
```

```
<%= link_to 'Edit', edit_article_path(@article) %> |
```

```
<%= link_to 'Back', articles_path %>
```

Show comments (3)

- Παρουσιάζεται το άρθρο
- Παρουσιάζεται φόρμα για εισαγωγή νέου σχολίου
- Κλήση από τη φόρμα του:
`/articles/1/comments`

Create comment

- app/controllers/comments_controller.rb

```
def create
```

```
  @article = Article.find(params[:article_id])
```

```
  @comment = @article.comments.create(comment_params)
```

```
  redirect_to article_path(@article)
```

```
end
```

```
private
```

```
  def comment_params
```

```
    params.require(:comment).permit(:commenter, :body)
```

```
  end
```


Nesting

- app/views/articles/show.html.erb

```
<p>
  <strong>Title:</strong>
  <%= @article.title %>
</p>
```

```
<p>
  <strong>Text:</strong>
  <%= @article.text %>
</p>
```

```
<h2>Comments</h2>
<% @article.comments.each do |comment| %>
  <p>
    <strong>Commenter:</strong>
    <%= comment.commenter %>

    <strong> Comment:</strong>
    <%= comment.body %>
  </p>
<% end %>
```

```
<h2>Add a comment:</h2>
<%= form_with(model: [ @article, @article.comments.build ], local: true) do |form| %>
  <p>
    <%= form.label :commenter %><br>
    <%= form.text_field :commenter %>
  </p>
  <p>
    <%= form.label :body %><br>
    <%= form.text_area :body %>
  </p>
  <p>
    <%= form.submit %>
  </p>
<% end %>
```

```
<%= link_to 'Edit', edit_article_path(@article) %> |
<%= link_to 'Back', articles_path %>
```

Title: Rails is Awesome!

Text: It really is.

Comments

Commenter: A fellow dev

Comment: I agree!!!

Add a comment:

Commenter

Body

Create Comment

[Edit](#) | [Back](#)

Refactoring

- `app/views/articles/show.html.erb`
- Πολύ μεγάλο!

Μερικές συλλογές

- app/views/comments/_comment.html.erb

```
<p>  
  <strong>Commenter:</strong>  
  <%= comment.commenter %>  
</p>
```

```
<p>  
  <strong>Comment:</strong>  
  <%= comment.body %>  
</p>
```

Μερικές συλλογές

- app/views/articles/show.html.erb

<p>

Title:

<%= @article.title %>

</p><p>

Text:

<%= @article.text %>

</p><h2>Comments</h2>

<%= render @article.comments %>

Μερική Φόρμα

- app/views/comments/_form.html.erb

```
<%= form_with(model: [ @article, @article.comments.build ], local: true) do
|form| %>
  <p>
    <%= form.label :commenter %><br>
    <%= form.text_field :commenter %>
  </p>
  <p>
    <%= form.label :body %><br>
    <%= form.text_area :body %>
  </p>
  <p>
    <%= form.submit %>
  </p>
<% end %>
```

Μερική Φόρμα

- app/views/articles/show.html.erb

```
<p>
  <strong>Title:</strong>
  <%= @article.title %>
</p>
```

```
<p>
  <strong>Text:</strong>
  <%= @article.text %>
</p>
```

```
<h2>Comments</h2>
<%= render @article.comments %>
```

```
<h2>Add a comment:</h2>
<%= render 'comments/form' %>
```

```
<%= link_to 'Edit', edit_article_path(@article) %> |
<%= link_to 'Back', articles_path %>
```

comments/form (smart
Rails!!!)

forward slash →
render →
_form.html.erb file

app/views/comments directory

Διαγραφή σχολίων

- app/views/comments/_comment.html.erb

```
<p>
  <strong>Commenter:</strong>
  <%= comment.commenter %>
</p>
```

```
<p>
  <strong>Comment:</strong>
  <%= comment.body %>
</p>
```

```
<p>
  <%= link_to 'Destroy Comment', [comment.article, comment],
    method: :delete,
    data: { confirm: 'Are you sure?' } %>
</p>
```

DELETE /articles/:article_id/comments/:id (CommentsController)

Διαγραφή σχολίων

- app/controllers/comments_controller.rb

```
class CommentsController < ApplicationController
  def create
    @article = Article.find(params[:article_id])
    @comment = @article.comments.create(comment_params)
    redirect_to article_path(@article)
  end

  def destroy
    @article = Article.find(params[:article_id])
    @comment = @article.comments.find(params[:id])
    @comment.destroy
    redirect_to article_path(@article)
  end

  private
  def comment_params
    params.require(:comment).permit(:commenter, :body)
  end
end
```

Διαγραφή σχολίων

- Διαγραφή άρθρου και των σχολίων του!

app/models/article.rb

```
class Article < ApplicationRecord
  has_many :comments, dependent: :destroy
  validates :title, presence: true,
                  length: { minimum: 5 }
end
```