ΜΥΕ042 - Τεχνολογίες Διαδικτύου

Διδάσκων: Στέργιος Αναστασιάδης

Αναφορά 1ης Εργαστηριακής Άσκησης:

Εφαρμογή ιστού για κοινοχρησία φωτογραφιών στοRuby-on-Rails

Ομάδα:

Κονδυλία Βέργου 4325

Παναγιώτης Βουζαλής 2653

Χειμερινό Εξάμηνο2022

Πίνακας Περιεχομένων

[1. Προϋπάρχων Error: Δημιουργία Null Photo 3](#_Toc120641460)

[2. Προσθήκη Τίτλου σε Φωτογραφία 6](#_Toc120641461)

[2.1. Δημιουργία Migration add\_title\_to\_photos 6](#_Toc120641462)

[2.2. Επεξεργασία Ελεγκτή photos\_controller 7](#_Toc120641463)

[2.3. Επεξεργασία Προβολής new 8](#_Toc120641464)

[2.4. Επεξεργασία Προβολής show 8](#_Toc120641465)

[2.4. Παράδειγμα Χρήσης Title σε Φωτογραφίες 9](#_Toc120641466)

[3. Προσθήκη λειτουργίας follow 10](#_Toc120641467)

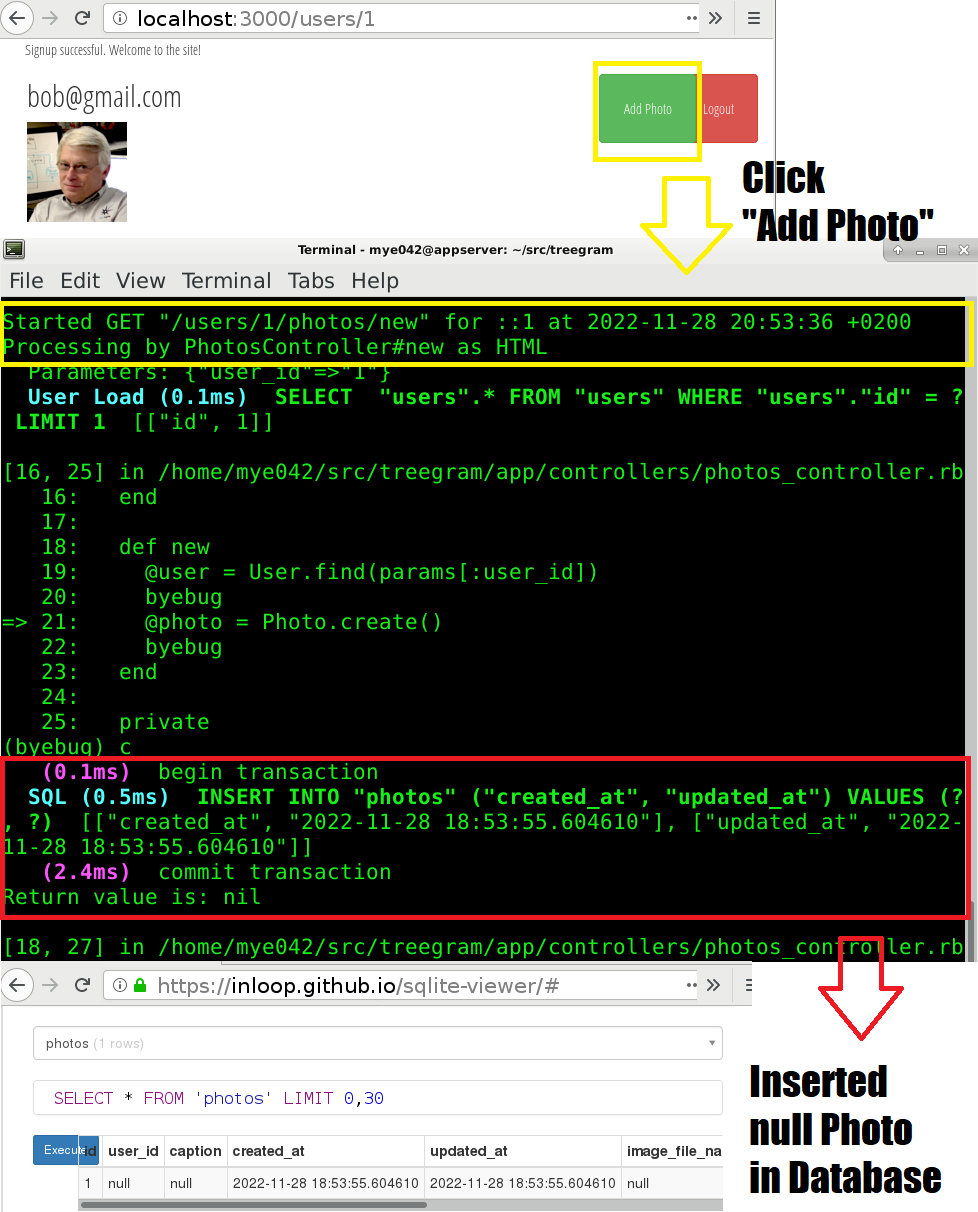
[3.1. Δημιουργία Migration create\_follows 10](#_Toc120641468)

[3.2. Δημιουργία Μοντέλου follow 11](#_Toc120641469)

[3.3. Δημιουργία Ελεγκτή follows\_controller 11](#_Toc120641470)

1. Προϋπάρχων Error: Δημιουργία Null Photo   
Όταν πατούμε το κουμπί Add Photo, δημιουργείται μια null Photo πριν καν φορτωθεί η προβολή new.html.haml. Αυτό γίνεται, διότι η συνάρτηση Ruby που υλοποιεί την ενέργεια new στον PhotoController καλεί την Photo.create().

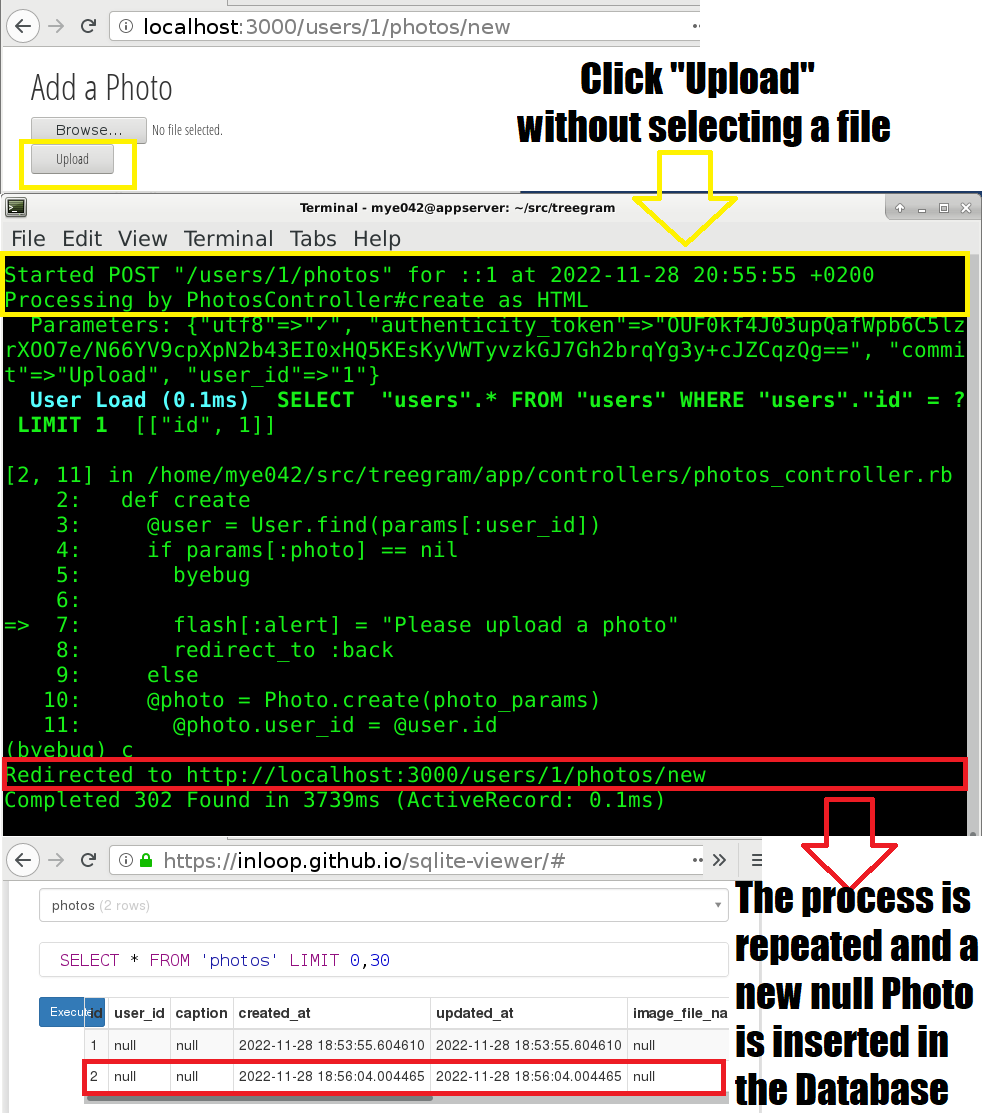
Επιβεβαιώσαμε τα παραπάνω, χρησιμοποιώντας τον διαδραστικό αποσφαλματωτή byebug και τον online [SQLite Viewer](https://inloop.github.io/sqlite-viewer).



Αφού ολοκληρωθεί η κλήση της new, φορτώνεται η προβολή στον διακομιστή μας.

Σε αυτό το σημείο, μπορούμε να ξαναδημιουργήσουμε μια null Photo, αν πατήσουμε στην φόρμα προσθήκης Photo το Upload χωρίς να έχουμε προσθέσει φωτογραφία.

Αυτό γίνεται καθώς η συνάρτηση Ruby που υλοποιεί την ενέργεια create στον PhotoController, αν δεν έχει επιλεχθεί φωτογραφία, κάνει redirect στην new και επαναλαμβάνονται όσα αναφέραμε παραπάνω.



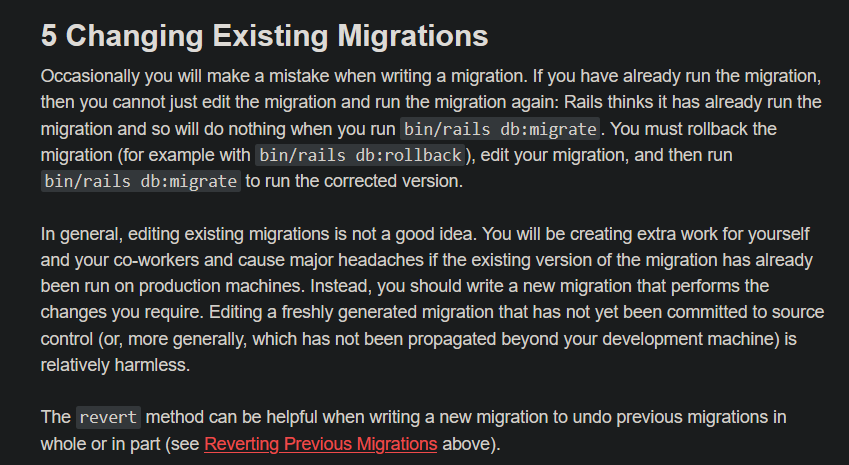
Για να εξαλείψουμε αυτό το πρόβλημα στην υλοποίησή μας, αφαιρέσαμε στην συνάρτηση new του αρχείου “app/controllers/photos\_controller.rb” τις δυο γραμμές κώδικα που είχε:  
$ @user = User.find(params[:user\_id])  
$ @photo = Photo.create()

Η πρώτη γραμμή έκανε απλά GET έναν User που δεν τον χρησιμοποιούσε και η δεύτερη γραμμή δημιουργούσε τη null Photo.

# 2. Προσθήκη Τίτλου σε Φωτογραφία

## 2.1. Δημιουργία Migration add\_title\_to\_photos

Από το [guides.rubyonrails.org](https://guides.rubyonrails.org/active_record_migrations.html#changing-existing-migrations) διαβάσαμε ότι είναι προτιμότερο να δημιουργήσουμε καινούργιο migration παρά να τροποποιήσουμε ένα ήδη υπάρχον.



Για αυτόν τον λόγο θα φτιάξουμε ένα καινούργιο migration έτσι ώστε να προσθέσουμε τον τίτλο στις φωτογραφίες.

Για την προσθήκη του column “title” στον πίνακα “photos” της βάσης μας εκτελούμε τις εξής εντολές στo root directory της εφαρμογής:  
$ rails generate migration add\_title\_to\_photos title:string  
$ rake db:migrate  
  
Graphical user interface, text, application

Description automatically generated

A screenshot of a computer screen

Description automatically generated

## 2.2. Επεξεργασία Ελεγκτή photos\_controller

Στο αρχείο “app/controllers/photos\_controller.rb” προσθέτουμε τους εξής ελέγχους, για να αναγκάζεται ο χρήστης να έχει επιλέξει φωτογραφία προς ανέβασμα και να έχει προσθέσει τίτλο, πριν του επιτραπεί να προχωρήσει:  
$ if params[:photo][:image] == nil  
$ flash[:alert] = "Please upload a photo."  
$ redirect\_to :back  
$ elsif params[:photo][:title] == ""  
$ flash[:alert] = "You forgot to add a title!"  
$ redirect\_to :back

Επίσης βάζουμε το argument :title στη γραμμή   
$ params.require(:photo).permit(:image, :title)  
Text

Description automatically generated

## 2.3. Επεξεργασία Προβολής new

Στο αρχείο “app/views/photos/new.html.haml” προσθέτουμε την παρακάτω γραμμή:  
$ %h2 Add a title: #{form.text\_field :title, placeholder:"Look at my dog!"}  
Συνεπώς, το prompt για το upload φωτογραφίας θα φαίνεται έτσι:  
Graphical user interface, application

Description automatically generated

## 2.4. Επεξεργασία Προβολής show

Στο αρχείο “app/views/users/show.html.haml” προσθέτουμε την εξής γραμμή για να υποδείξουμε στον κώδικά μας να εμφανίζει τον τίτλο κάθε φωτογραφίας:  
$ %h2= photo.title

Επίσης προσθέτουμε την εξής γραμμή για να εμφανίζεται το όνομα και το catchphrase της εφαρμογής όσο τη χρησιμοποιούμε:  
$ %h1.title Treegram - Show off to the world!  
Timeline

Description automatically generated

## 2.4. Παράδειγμα Χρήσης Title σε Φωτογραφίες

Επομένως, η εφαρμογή μας φαίνεται έτσι, όταν έχουμε κάνει log σε ένα user και έχουμε ανεβάσει φωτογραφίες.

Graphical user interface, application

Description automatically generated

# 3. Προσθήκη λειτουργίας follow

## 3.1. Δημιουργία Migration create\_follows

Δημιουργούμε το migration create\_follows και μέσα του ορίζουμε τη δομή του πίνακα με τα αναγνωριστικά του follower (αυτός που ακολουθεί τα post κάποιου) και του followee (αυτός του οποίου τα post ακολουθούν άλλοι):  
Text

Description automatically generated

Εκτελούμε $ rake db:migrate για να δημιουργήσουμε τον πίνακα των follow:  
Text

Description automatically generated

Ο οποίος σε αυτό το σημείο είναι άδειος:  
Graphical user interface, application

Description automatically generated

## 3.2. Δημιουργία Μοντέλου follow

Δημιουργούμε το μοντέλο follow με την παρακάτω εντολή:  
$ rails generate model follow  
Text

Description automatically generated

Τροποποιούμε το αρχείο “app/models/follow.rb” προσθέτοντας τη γραμμή  
$ validates :follower\_id, uniqueness: {scope: :followee\_id}  
Graphical user interface, text

Description automatically generated

## 3.3. Δημιουργία Ελεγκτή follows\_controller

Σε αυτό το σημείο πρέπει να δημιουργήσουμε τη σχέση μεταξύ των πινάκων follow και user. Αυτό θα το κάνουμε με έναν follows controller.

Δημιουργούμε τον follows controller με την εξής εντολή:  
$ rails generate controller follows  
Text

Description automatically generated

Τροποποιούμε το αρχείο “app/controllers/follows\_controller.rb” προσθέτοντας λειτουργία follow/unfollow:  
Text

Description automatically generated

## 3.3. Επεξεργασία Αρχείου Διαδρομών routes.rb

Δημιουργούμε τις διαδρομές στις συναρτήσεις που δημιουργήσαμε, προσθέτοντας στο αρχείο “config/routes.rb” τη γραμμή:  
$ resources :follows  
A screenshot of a computer

Description automatically generated with medium confidence

## 3.4. Δημιουργία Προβολής index

Δημιουργούμε το αρχείο “/app/views/follows/index.html.haml” για προβολή όλων των χρηστών που μπορεί να ακολουθήσει ένας χρήστης:  
Text

Description automatically generated

## 3.5. Επεξεργασία Προβολής show

Τροποποιούμε το αρχείο “/app/views/users/show.html.haml” για να προσθέσουμε:  
- Κουμπί για εμφάνιση εγγεγραμμένων χρηστών  
- Εμφάνιση φωτογραφιών των ατόμων που ακολουθεί ο χρήστης (folowees)