
02NPYPD - LINGUAGGI E AMBIENTI MULTIMEDIALI A

USER POSTS

Creazione e gestione dei post degli utenti

Lo scopo di questa esercitazione è quello di fare ancora un po' di pratica con la creazione di modelli e delle pagine necessarie per creare e gestire i post degli utenti. Prerequisito è il set di slide intitolato "Lab 6 - Warm Up".

ESERCIZIO 1

Creare un modello per i post degli utenti che:

- abbia tre attributi, `title`, `content` (entrambi di tipo `string`) e `user_id` (di tipo `integer`);
- tramite un'indice sul database, assicurarsi di poter recuperare i post di un utente in ordine inverso di creazione;
- tra gli attributi accessibili del modello figurino solo `title` e `content`;
- `user_id` deve essere sempre presente nella creazione di un nuovo post;
- `title` deve essere sempre presente nella creazione di un post e deve avere una lunghezza superiore a 5 caratteri;
- `content` deve essere sempre presente nella creazione di un post e deve avere una lunghezza inferiore a 500 caratteri;
- utilizzare la funzionalità `belongs_to` e `has_many` per collegare post e utenti; tali metodi vanno usati nei modelli dei post e degli utenti, rispettivamente;
- ordinare i post in ordine decrescente di creazione (tramite la facility `default_scope`¹ di Rails);
- assicurarsi che, quando un utente viene cancellato dal social network, anche tutti i suoi post vengano eliminati (aggiungere a `has_many` nel modello degli utenti la proprietà `dependent: :destroy`).

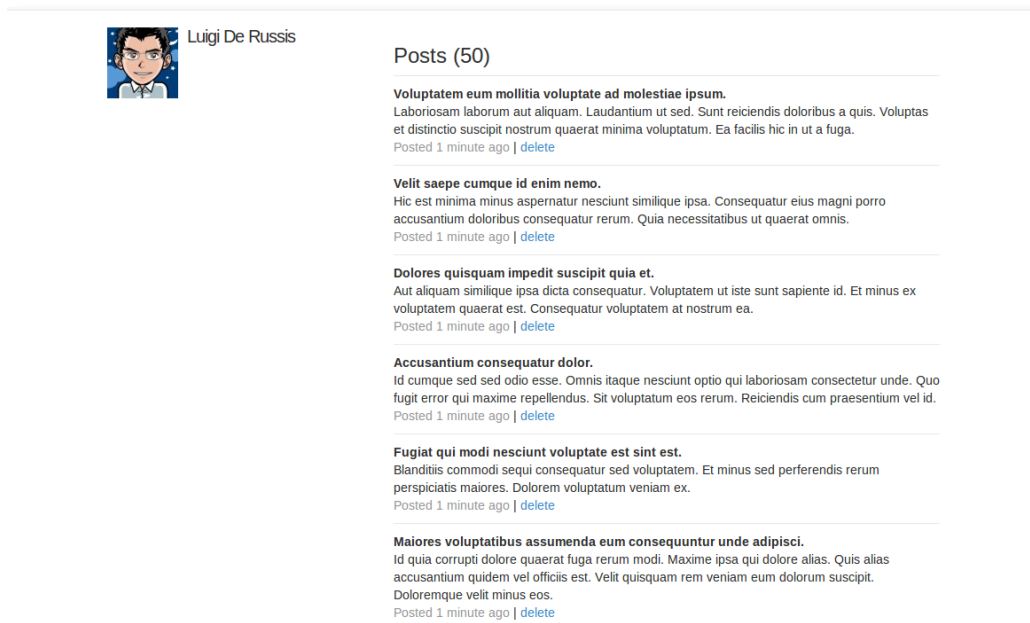
ESERCIZIO 2

Gestire la visualizzazione dei post creati con il modello precedente in modo che appaiano nella stessa pagina dell'utente (`show`).

In tale pagina, solo se ci sono post associati all'utente corrente, visualizzare il numero totale dei post dell'utente e il loro elenco, eventualmente su più pagine (chiamare `will_paginate` passandogli come parametro l'elenco paginato dei post nella variabile `@posts`).

Ogni post dovrà mostrare il proprio titolo, il contenuto e da quanto tempo è stato pubblicato (si può usare `time_ago_in_words`² con il campo `created_at` associato al modello). Vedi figura seguente per un esempio.

¹ http://api.rubyonrails.org/v3.2.21/classes/ActiveRecord/Scoping/Default/ClassMethods.html#method-i-default_scope



Luigi De Russis

Posts (50)

Voluptatem eum mollitia voluptate ad molestiae ipsum.
 Laboriosam laborum aut aliquam. Laudantium ut sed. Sunt reiciendis doloribus a quis. Voluptas et distinctio suscipit nostrum quaerat minima voluptatum. Ea facilis hic in ut a fuga.
 Posted 1 minute ago | [delete](#)

Velit saepe cumque id enim nemo.
 Hic est minima minus aspernatur nesciunt similique ipsa. Consequatur eius magni porro accusantium doloribus consequatur rerum. Quia necessitatibus ut quaerat omnis.
 Posted 1 minute ago | [delete](#)

Dolores quisquam impedit suscipit quia et.
 Aut aliquam similique ipsa dicta consequatur. Voluptatem ut iste sunt sapiente id. Et minus ex voluptatem quaerat est. Consequatur voluptatem at nostrum ea.
 Posted 1 minute ago | [delete](#)

Accusantium consequatur dolor.
 Id cumque sed sed odio esse. Omnis itaque nesciunt optio qui laboriosam consectetur unde. Quo fugit error qui maxime repellendus. Sit voluptatum eos rerum. Reiciendis cum praesentium vel id.
 Posted 1 minute ago | [delete](#)

Fugiat qui modi nesciunt voluptate est sint est.
 Blanditis commodi sequi consequatur sed voluptatem. Et minus sed perferendis rerum perspicatis maiores. Dolorem voluptatum veniam ex.
 Posted 1 minute ago | [delete](#)

Maiores voluptatibus assumenda eum consequuntur unde adipisci.
 Id quia corrupti dolore quaerat fuga rerum modi. Maxime ipsa qui dolore alias. Quis alias accusantium quidem vel officis est. Velit quisquam rem veniam eum dolorum suscipit. Doloremque velit minus eos.
 Posted 1 minute ago | [delete](#)

Modificare il task di Rake che permette di creare utenti “finti” in modo tale che possa generare anche post finti:

```
users = User.all(limit: 10)
50.times do
  post_title = Faker::Lorem.sentence(3)
  post_content = Faker::Lorem.paragraph
  users.each { |user| user.posts.create!(title: post_title, content:
post_content ) }
end
```

ESERCIZIO 3

Aggiungere la possibilità, solo per utenti loggati, di creare nuovi post tramite un apposito form collocato nella home page. Per farlo, utilizzare i controllori delle pagine e degli utenti. Ricordarsi di configurare correttamente le route, come già fatto in precedenza per gli utenti e per le sessioni.

Prevedere anche la possibilità di eliminare (“distruggere”) i post presenti nel profilo dell’utente: ispirarsi a quanto fatto per l’eliminazione di utenti sul sito.

² http://api.rubyonrails.org/v3.2.21/classes/ActionView/Helpers/DateHelper.html#method-i-time_ago_in_words