

Ruby On Rails course

Tips, Mailers, Enumerations & Gems



Tips

Render @users

```
class UsersController < ApplicationController  
  
  def index  
    @users = User.all  
  end  
  
end
```



app/views/users/index.html.erb

```
<% @users.each do |user| %>
  <div>
    <p><%= prettify_name(user.name) %></p>
    <p><%= link_to 'Edit', edit_user_path(user) %></p>
    <p><%= link_to 'Delete', user_path(user), method: :delete %></p>
  </div>
<% end %>
```



app/views/users/_user.html.erb

```
<div>
  <p><%= prettify_name(user.name) %></p>
  <p><%= link_to 'Edit', edit_user_path(user) %></p>
  <p><%= link_to 'Delete', user_path(user), method: :delete %></p>
</div>
```



app/views/users/index.html.erb

```
<% @users.each do |user| %>  
  <%= render 'user', user: user %>  
  
<% end %>
```



app/views/users/index.html.erb

```
<%= render @users %>
```

Dependent destroy

```
class User < ActiveRecord::Base
  has_many :posts, dependent: :destroy
end
```



Dependent nullify

```
class User < ActiveRecord::Base
  has_many :posts, dependent: :nullify
end
```



Migration

```
class AddNameToUsers < ActiveRecord::Migration

  def change
    add_column :users, :name, :string
  end

end
```



Migration

```
class AddNameToUsers < ActiveRecord::Migration

  def change
    add_column :users, :full_name, :string
  end

end
```



Migration

```
class RenameUsersNameToFullName < ActiveRecord::Migration

  def change
    rename_column :users, :name, :full_name
  end

end
```



Content for

app/views/layouts/application.html.erb

```
<% if content_for?(:new_post_button) %>
```

```
  <%= yield :new_post_button %>
```

```
<% else %>
```

```
  <div class="btn">New post</div>
```

```
<% end %>
```



Content for

app/views/subreddits/show.html.erb

```
<%= content_for :new_post_button do %>
```

```
  <%= link_to 'New post', new_subreddit_post_path(@subreddit.id) %>
```

```
<% end %>
```



Rubocop

Ruby style guide

<https://github.com/bbatsov/ruby-style-guide>



Rubocop

```
def badName
  if something
    test
  end
end
```

Use snake_case for method names.

```
def badName
  ^^^^^^^
```



Rubocop

```
def bad_name
  if something
    test
  end
end
```

Use a guard clause instead of wrapping the code inside a conditional expression.

```
if something
  ^^
```

End is not aligned with if

```
  end
  ^^^
end
```



Rubocop

```
def bad_name  
  test if something  
end
```



Counter cache

Counter cache

```
@users = User.all
```

```
<% @users.each do |user| %>
```

```
  <h4>Posts (<%= user.posts.count %>)</h4>
```

```
  <% user.posts.each do |post| %>
```

```
    <p><%= post.title %></p>
```

```
    <p><%= link_to 'delete', user_post_path(user, post), method: :delete %></p>
```

```
  <% end %>
```

```
<% end %>
```



Counter cache

Started GET "/users" for 127.0.0.1 at 2015-07-17 14:23:25 +0200

Processing by UsersController#index as HTML

User Load (0.2ms) SELECT "users".* FROM "users"

(0.1ms) SELECT COUNT(*) FROM "posts" WHERE "posts"."user_id" = ? [["user_id", 12]]

Post Load (0.1ms) SELECT "posts".* FROM "posts" WHERE "posts"."user_id" = ? [["user_id", 12]]

(0.2ms) SELECT COUNT(*) FROM "posts" WHERE "posts"."user_id" = ? [["user_id", 14]]

Post Load (0.1ms) SELECT "posts".* FROM "posts" WHERE "posts"."user_id" = ? [["user_id", 14]]

(0.1ms) SELECT COUNT(*) FROM "posts" WHERE "posts"."user_id" = ? [["user_id", 15]]

Post Load (0.1ms) SELECT "posts".* FROM "posts" WHERE "posts"."user_id" = ? [["user_id", 15]]

(0.2ms) SELECT COUNT(*) FROM "posts" WHERE "posts"."user_id" = ? [["user_id", 16]]

Post Load (0.2ms) SELECT "posts".* FROM "posts" WHERE "posts"."user_id" = ? [["user_id", 16]]



Counter cache

```
class User < ActiveRecord::Base  
  has_many :posts, dependent: :destroy  
end  
  
class Posts < ActiveRecord::Base  
  belongs_to :user, counter_cache: true  
end
```

Counter cache

```
class AddCounterCacheStatus < ActiveRecord::Migration

  def change
    add_column :users, :posts_counter, :integer
  end

end
```



Counter cache

```
@users = User.all
```

```
<% @users.each do |user| %>
```

```
  <h4>Posts (<%= user.posts.count %>)</h4>
```

```
  <% user.posts.each do |post| %>
```

```
    <p><%= post.title %></p>
```

```
    <p><%= link_to 'delete', user_post_path(user, post), method: :delete %></p>
```

```
  <% end %>
```

```
<% end %>
```



Counter cache

```
@users = User.all
```

```
<% @users.each do |user| %>
```

```
  <h4>Posts (<%= user.posts_count %>)</h4>
```

```
  <% user.posts.each do |post| %>
```

```
    <p><%= post.title %></p>
```

```
    <p><%= link_to 'delete', user_post_path(user, post), method: :delete %></p>
```

```
  <% end %>
```

```
<% end %>
```



Without counter cache

Started GET "/users" for 127.0.0.1 at 2015-07-17 14:23:25 +0200

Processing by UsersController#index as HTML

User Load (0.2ms) SELECT "users".* FROM "users"

(0.1ms) SELECT COUNT(*) FROM "posts" WHERE "posts"."user_id" = ? [["user_id", 12]]

Post Load (0.1ms) SELECT "posts".* FROM "posts" WHERE "posts"."user_id" = ? [["user_id", 12]]

(0.2ms) SELECT COUNT(*) FROM "posts" WHERE "posts"."user_id" = ? [["user_id", 14]]

Post Load (0.1ms) SELECT "posts".* FROM "posts" WHERE "posts"."user_id" = ? [["user_id", 14]]

(0.1ms) SELECT COUNT(*) FROM "posts" WHERE "posts"."user_id" = ? [["user_id", 15]]

Post Load (0.1ms) SELECT "posts".* FROM "posts" WHERE "posts"."user_id" = ? [["user_id", 15]]

(0.2ms) SELECT COUNT(*) FROM "posts" WHERE "posts"."user_id" = ? [["user_id", 16]]

Post Load (0.2ms) SELECT "posts".* FROM "posts" WHERE "posts"."user_id" = ? [["user_id", 16]]



With counter cache

Started GET "/users" for 127.0.0.1 at 2015-07-17 14:23:25 +0200

Processing by UsersController#index as HTML

User Load (0.2ms) SELECT "users".* FROM "users"

Post Load (0.2ms) SELECT "posts".* FROM "posts" WHERE "posts"."user_id" = ? [["user_id", 12]]

Post Load (0.1ms) SELECT "posts".* FROM "posts" WHERE "posts"."user_id" = ? [["user_id", 14]]

Post Load (0.1ms) SELECT "posts".* FROM "posts" WHERE "posts"."user_id" = ? [["user_id", 15]]

Post Load (0.2ms) SELECT "posts".* FROM "posts" WHERE "posts"."user_id" = ? [["user_id", 16]]



```
@users = User.all
```

```
<% @users.each do |user| %>
```

```
  <h4>Posts (<%= user.posts_count %>)</h4>
```

```
  <% user.posts.each do |post| %>
```

```
    <p><%= post.title %></p>
```

```
    <p><%= link_to 'delete', user_post_path(user, post), method: :delete %></p>
```

```
  <% end %>
```

```
<% end %>
```



Lazy loading

```
@users = User.includes(:posts)
```

```
<% @users.each do |user| %>
```

```
  <h4>Posts (<%= user.posts_count %>)</h4>
```

```
  <% user.posts.each do |post| %>
```

```
    <p><%= post.title %></p>
```

```
    <p><%= link_to 'delete', user_post_path(user, post), method: :delete %></p>
```

```
  <% end %>
```

```
<% end %>
```



With CC & without lazy loading

Started GET "/users" for 127.0.0.1 at 2015-07-17 14:23:25 +0200

Processing by UsersController#index as HTML

User Load (0.2ms) SELECT "users".* FROM "users"

Post Load (0.2ms) SELECT "posts".* FROM "posts" WHERE "posts"."user_id" = ? [["user_id", 12]]

Post Load (0.1ms) SELECT "posts".* FROM "posts" WHERE "posts"."user_id" = ? [["user_id", 14]]

Post Load (0.1ms) SELECT "posts".* FROM "posts" WHERE "posts"."user_id" = ? [["user_id", 15]]

Post Load (0.2ms) SELECT "posts".* FROM "posts" WHERE "posts"."user_id" = ? [["user_id", 16]]



With CC & with lazy loading

Started GET "/users" for 127.0.0.1 at 2015-07-17 14:23:25 +0200

Processing by UsersController#index as HTML

User Load (0.2ms) SELECT "users".* FROM "users"

Post Load (0.5ms) SELECT "posts".* FROM "posts" WHERE "posts"."user_id" IN (12, 14, 15, 16)



Enumerations

ActiveRecord enums

Migration

```
class AddStatus < ActiveRecord::Migration

  def change
    add_column :users, :state, :integer
  end

end
```



Model

```
class User < ActiveRecord::Base  
  enum state: [:active, :blocked]  
  
end
```



Methods

```
user = User.new  
user.state  
# => nil
```

```
user.active?  
# => false
```

```
user.state = :blocked  
user.blocked?  
# => true
```



Methods

```
user = User.new  
user.active!
```

```
user.persisted?  
# => true
```

```
user.active?  
# => true
```

```
user[:state]  
# => 0
```



Scopes

`User.blocked`

`# => #<ActiveRecord::Relation []>`

`User.active`

`# => #<ActiveRecord::Relation [#<User id: 7, status: 0...]>`



Reserved words

```
class User < ActiveRecord::Base
```

```
  enum state: [:logger]
```

```
end
```

```
# => ArgumentError: You tried to define an  
enum named "state" on the model "User", but  
this will generate a class method "logger",  
which is already defined by Active Record.
```



Form

```
User.states
```

```
# => {"active"=>0, "blocked"=>1}
```

```
<%= form_for @user do |f| %>
```

```
  <%= f.select :state, User.states.keys %>
```

```
<% end %>
```



Mailers

Migration

```
rails g mailer Notifications
```

```
create app/mailers/notifications.rb
create app/mailers/application_mailer.rb
invoke erb
create app/views/notifications
create app/views/layouts/mailer.text.erb
create app/views/layouts/mailer.html.erb
invoke test_unit
create test/mailers/notifications_test.rb
create test/mailers/previews/notifications_preview.rb
```

app/mailers/application_mailer.rb

```
class ApplicationMailer < ActionMailer::Base  
  
  default from: "from@example.com"  
  layout 'mailer'  
  
end
```



app/mailers/notifications.rb

```
class Notifications < ApplicationMailer

  def comments(user)
    @user = user

    mail(to: @user.email, subject: 'New comment')
  end

end
```



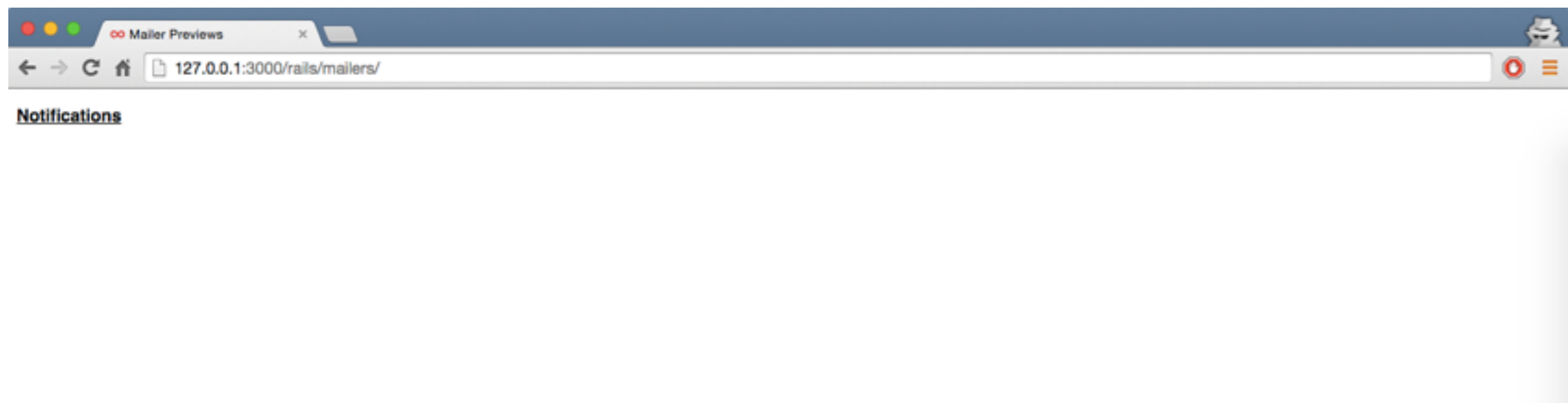
app/views/notifications/comments.html.erb

```
<%= "Hi, #{@user.name} left a comment on your post" %>
```



Testing

<http://127.0.0.1:3000/rails/mailers/>



test/mailers/previews/notifications_preview.rb

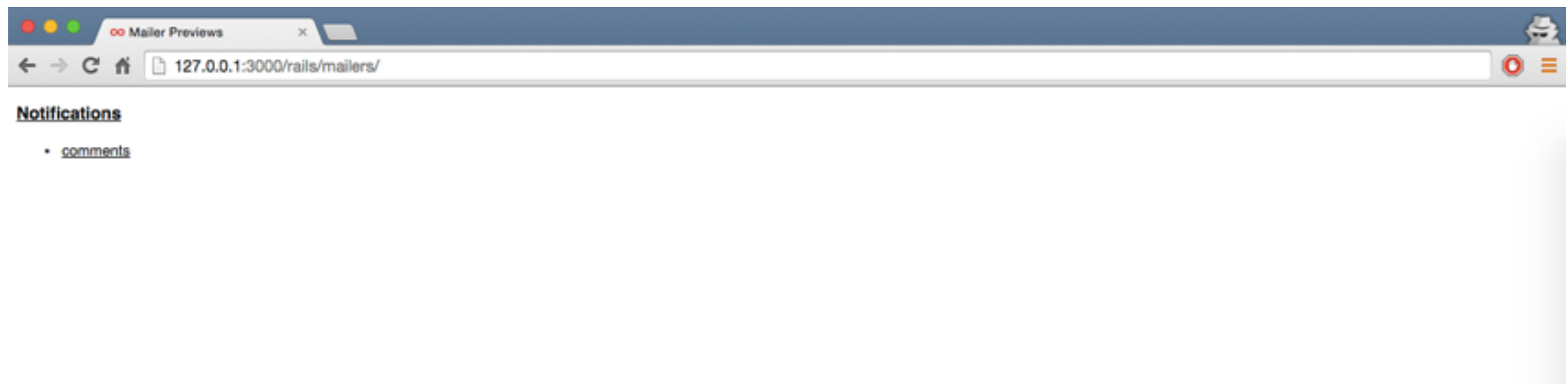
```
# Preview all emails at http://localhost:3000/rails/  
mailers/notifications
```

```
class NotificationsPreview < ActionMailer::Preview  
  
  def comments  
    Notifications.comments(User.first)  
  end  
  
end
```

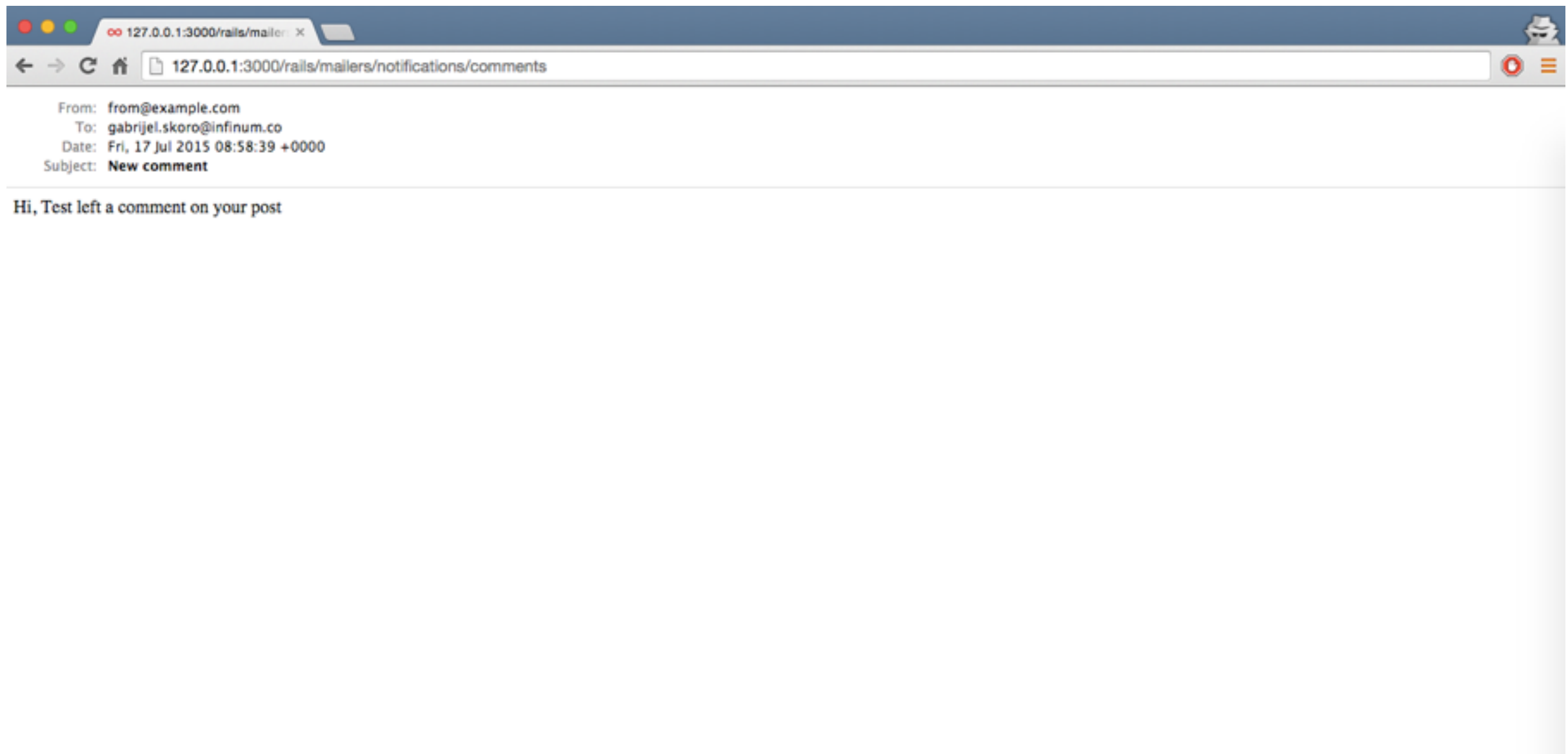


Testing

<http://127.0.0.1:3000/rails/mailers/>



Preview



Sending mail

```
class PostsController < ApplicationController

  def send_notification
    @user = User.find(params[:id])

    Notification.comments(@user).deliver_now
  end

end
```



Letter opener

```
group :development do  
  gem "letter_opener"  
end
```

```
config.action_mailer.delivery_method = :letter_opener  
# => config/environments/development.rb
```



Gems

Better errors, simple_form, paperclip

Better errors

Better Errors replaces the standard Rails error page with a much better and more useful error page

https://github.com/charliesome/better_errors

```
group :development do  
  gem "better_errors"  
  gem "binding_of_caller"  
  
end
```



Simple form

`https://github.com/plataformatec/simple_form`

`gem "simple_form"`

Paperclip

`https://github.com/thoughtbot/paperclip`

`gem "paperclip"`



Utorak, 21.07 – Infinum 7. kat

Cetvrtak 23.07 – Zgrada, –1. kat