

Programación con Ruby on Rails

Tienda (VIII)

Delio Tolivia





Indice

Configurando email

Creando un mailer

Enviado el email

Configurando email

- Para que nuestra aplicación envíe email debemos configurarlo en config/enviroment.rb.

```
Shop::Application.configure do  
  config.action_mailer.delivery_method = :smtp #Ponemos en :test y vemos en consola  
  config.action_mailer.smtp_settings = {  
    adress: "smtp.gmail.com",  
    port: 587,  
    domain: "talentocorporativo.com",  
    authentication: "plain"  
    user_name: "delio.tolivia@talentocorporativo.com",  
    password: "xxxxxxxxxx",  
    enable_starttls_auto: true  
  }  
end
```

Creando un mailer(I)

- Rails tiene un script para generar mailers. Son clases que se guardan en app/mailers y contienen uno o más métodos que cada uno se corresponderá con una plantilla de email. Vamos a crear un mailer con dos métodos, uno para cuando se hace el pedido y otro para cuando es enviado.

rails generate mailer OrderNotifier received shipped

Creando un mailer(II)

- Modificamos la plantilla usada app/views/order_notifier_mailer/received.text.erb

Dear <%= @order.name %>

Thank you for your recent order from The Pragmatic Store.

You ordered the following items:

<%= render @order.line_items %>

We'll send you a separate e-mail when your order ships.

Creando un mailer(III)

- El partial template para cada línea es app/views/line_item/_line_item.text.erb

```
<%= sprintf("%2d x %s",  
            line_item.quantity,  
            truncate(line_item.product.title, length:50)) %>
```

- Y el método received de app/mailers/order_notifier.rb

```
def received (order)  
  @order = order
```

```
  mail to: order.email, subject: 'Pragmatic Store Order Confirmation'  
end
```

Enviando el email (I)

- En app/controllers/orders_controller.rb en el método **create()**

```
def create
```

```
  @order = Order.new(order_params)
```

```
  @order.add_line_items_from_cart(@cart)
```

```
  respond_to do |format|
```

```
    if @order.save
```

```
      Cart.destroy(session[:cart_id])
```

```
      session[:cart_id]=nil
```

```
      OrderNotifier.received(@order).deliver # en Rails 5 la clase es OrderNotifierMailer
```

```
      format.html { redirect_to store_url, notice: 'Thank you for your order.' }
```

```
      format.json { render :show, status: :created, location: @order }
```

```
    else
```

```
      format.html { render :new }
```

```
      format.json { render json: @order.errors, status: :unprocessable_entity }
```

```
    end
```

```
  end
```

```
end
```

Enviando el email (II)

- Además de las plantillas en texto plano podemos hacerlas en html. Se encuentran en `app/views/order_notifier/received.html.erb` y el partial en `app/cIEWS/line_items/_line_item.html.erb`