

Create and print PDF files in rails

Add gem 'prawn' to gemfile

```
gem 'prawn'
```

Bundle install. Add to the file: config/initializers/mime_types.rb

```
Mime::Type.register "application/pdf", :pdf
```

Create a new file in the Model folder named invoice_pdf.rb. In this file, first we create the new class.

```
class InvoicePdf < Prawn::Document  
end
```

Then we add the initialize method. Later if we create an instance of the class we pass the invoice.id as a parameter.

```
def initialize(invoice)  
  super()  
  @invoice = invoice  
end
```

Then we start adding the content of the PDF file. First we add the logo on the left upper corner. The cursor starts on the left upper corner.

```
def image  
  image "#{Rails.root}/public/images/logo.jpg", :width => 120  
end
```

The route is where the image is stored. :width => 120 defines the size of the image. Next we add a text below the image.

```
def title_text  
  move_down 10  
  text "Fake Company\n" +  
    "Fake Street\n" +  
    "Fake City" , :size => 10  
end
```

move_down is to move the cursor from the current position down. \n is next line. :size => 10 defines the size of the text. Next we add a text on the right on a specific position.

```
def customer
  text_box "<u><font size='20'>Invoice</font></u>\n" +
    "Customer No: #{@invoice.customer_number}" , :at => [350, 680],
    :inline_format => true
end
```

Here we see that I am using inline format. We have to tell the text box that we are doing this with :inline_format => true. With :at => [] we are giving the position for the text box. The start position [0, 0] is on the corner at the bottom left.

Next we add a table. For that we need another Gem Add code below to your gemfile

```
gem 'prawn-table'
```

Bundle install. Add on the top of the invoice_pdf.rb require 'prawn/table' First we create the table header.

```
def table_head
  move_down 30
  data = [ ["Item", "Qty", "Price"] ]
  table(data, :column_widths => [200, 50, 50],
    :cell_style => {size: 7, :font_style => :bold})
end
```

With data we define the first row of the table. :column_widths like the name it is the size of each column and with cell_style we add the design of the text in the table.

Next we add the other rows:

```
def items
  @invoice.items.each do |i|
    data = [ [i.name, i.count, i.price] ]
    table(data, :column_widths => [200, 50, 50])
  end
end
```

And now we add a barcode. For this we need another Gem.

```
gem 'barby'
```

Bundle install. Add on the top of the invoice_pdf.rb

```
require 'barby'
require 'barby/barcode/code_39'
require 'barby/outputter/prawn_outputter'
```

Then we add the barcode function:

```
def barcode
  bacode= Barby::Code39.new @invoice.id
  barcode.annotate_pdf(self, :x => 350, :y => 700, :height => 25)
end
```

x and y are the position of the barcode. height is the height of the barcode (50 standard). At the end we call all this functions in the InvoicePdf class:

```
def initialize(invoice)
  super()
  @invoice = invoice
  image
  title_text
  customer
  table_head
  items
  barcode
end
```

Now we need the link for the pdf in the show of the invoice.

```
<a href="<%= url_for :action => 'view_invoice_pdf',
                  :id => @invoice,
                  :format => 'pdf' %>" target="_blank">View Invoice as PDF
</a>
end
```

Add the function view_invoice_pdf to the controller:

```
def view_invoice_pdf
  @invoice = Invoice.find(params[:id])
  respond_to do |format|
    format.html
    format.pdf do
      pdf = InvoicePdf.new(@invoice)
      send_data pdf.render, filename: "Invoice.pdf",
        type: 'application/pdf',
        disposition: "inline"
    end
  end
end
```

And for the end we add a function to print the invoice PDF without opening it.

```

def print_invoice
  @invoice = Invoice.find(params[:id])
  pdf = PdfInvoice.new(@invoice)
  save_path = Rails.root.join('app/pdfs','invoice.pdf')
  File.open(save_path, 'wb') do |file|
    file << pdf.render
  end
  system "lpr", "app/pdfs/invoice.pdf"
  File.delete("app/pdfs/invoice.pdf")
  render :nothing => true, :status => 200, :content_type => 'text/html'
end

```

First we save the created PDF file to the folder pdfs, then we print the created PDF file with the system command, and then we delete the PDF file.

And we add the link to print the PDF automatically:

```

<a href="<%= url_for :action => 'print_invoice', :id => @invoice %>">Print Invoice</a>

```

For more information about prawn <http://prawnpdf.org/manual.pdf>

For more information about the tables with prawn <http://prawnpdf.org/prawn-table-manual.pdf>