## Rails: Convert HTML to PDF?

What is the best and easiest way of taking HTML and converting it into a PDF, similar to use CFDOCUMENT on ColdFusion?

**UPDATE:** I really appreciate all the comments and suggestions that people have given so far, however I feel that people leaving their answers are missing the point.

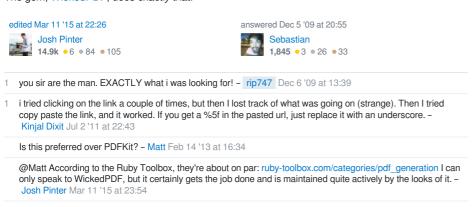
- 1) the solution has to be free or open sourced. one person suggested using pricexml and the other pd4ml. both of these solutions costs money (pricexml costing an arm and a leg) which i'm not about the fork over.
- 2) they must be able to take in html (either from a file, url or a string variable) and then produce the pdf. libraries like prawn, rprf, rtex are produced using their own methods and not taking in html.

please don't think i'm ungrateful for the suggestions, it's just that pdf generation seems like a really problem for people like me who use ColdFusion but want to convert to Rails.



## 14 Answers

The gem, WickedPDF, does exactly that.



The greatest thing nowaday is PDFKit for this job:

- https://github.com/pdfkit/PDFKit
- http://railscasts.com/episodes/220-pdfkit



Try WickedPDF, PDF generator (from HTML) plugin.

Here is example

```
gem 'wicked_pdf'
gem 'wkhtmltopdf-binary'
#wicked_pdf is a wrapper for wkhtmltopdf, you'll need to install that, too
```

In Your Controller(I supposed you are in show method):

In your view for PDF link

```
= link_to 'Download Pdf', you_show_path(@uour_object, format: :pdf)
```

If you want to send pdf in email attachment, first save your pdf file in public or temp folder then in mailer class (e.g. in mailers/e\_mailer.rb)

```
 attachments["test.pdf"] = File.read(Rails.root.join('public', "test.pdf")) \\ mail(:to => to, :cc => cc , :subject => "subject")
```

After sending email you can delete file

File.delete(Rails.root.join("public", "test.pdf"))

edited Aug 4 '16 at 6:26

answered Oct 7 '15 at 12:26

Sajjad Murtaza
715 • 6 • 14

This is the true answer - Abs Jan 31 '16 at 0:58

I know this question is almost 5 years old at this point. However, new options are available.

Try Shrimp! https://github.com/adeven/shrimp

answered Jun 5 '13 at 17:30

Michael Irey
1,573 • 2 • 22 • 45

After After a lot of blood sweat and tears I managed to write a pretty simple rails app that allows the user to write letter templates via TinyMce, these are then saved and can be viewed as pdf documents.

I decided to restrict the options in the wysiwyg editor as much as possible as some of the options don't work exactly as expected, but that's nothing a little gsub-ing couldn't solve if needed.

There is a ruby gem that wraps HTMLDOC which you'll need: PDF::HTMLDoc

Once you've got that, register the mime type, then you can do something like:

```
@letter_template = LetterTemplate.find(params[:id])
respond_to do |format|
format.html
format.pdf { send_data render_to_pdf({:action => 'show.rpdf', :layout => 'pdf_report'}), :filename =>
@letter_template.name + ".pdf", :disposition => 'inline' }
end
```

In the application controller i've added the render\_to\_pdf method like:

```
def render_to_pdf(options =nil)
    data = render_to_string(options)
    pdf = PDF::HTMLDoc.new
    pdf.set_option :bodycolor, :white
    pdf.set_option :toc, false
    pdf.set_option :portrait, true
    pdf.set_option :links, false
    pdf.set_option :webpage, true
    pdf.set_option :left, '2cm'
    pdf.set_option :right, '2cm'
```

```
pdf.set_option :footer, "../"
pdf.set_option :header, "..."
pdf.set_option :bottom, '2cm'
pdf.set_option :top, '2cm'
pdf.generate
end
```

You'll find more documentation on HTMLDOC site that Travis Beale Linked to. Hope this helps get you on your way and unless your documents are really complicated it should suffice. Let us know how you get on.

answered May 8 '09 at 14:43



- There is RPDF in which you describe your pdf's in as a
- There also is RTEX which uses (La)TeX to generates pdf's.

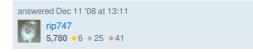
answered Dec 11 '08 at 17:16



8.082 • 2 • 31 • 42

The closest thing I found as a "solution" to this is by using JRuby and then using The Flying Saucer Project. I just wish that it would get ported over to Ruby so it could be a native solution. God I wish I was better at Java.

edited Dec 12 '08 at 18:30



acts\_as\_flying\_saucer is a ruby port of flying saucer library - Amar Jun 13 '11 at 9:46

The project may have stagnated, but Pisa (aka XHTML2PDF) is a Python lib which does CSS+HTML to PDF. used it in a (low-traffic) Rails app a while back by driving it as a commandline tool -- worked pretty well.

answered Jun 5 '10 at 23:54



**Thomas 41** • 1

If you want to have really good quality (think, proper layout/CSS etc) you need to create a process embedding a real webbrowser rendering it offscreen and printing it via PDF printer driver.

answered Dec 10 '08 at 19:21



actually I don't think would be the best solution as most websites have a "print" style to them that remove elements of the site. - rip747 Dec 10 '08 at 19:30

There is for sure no direct way to do this in Ruby (at least there wasn't when I was trying to do it

couple of months ago). If you really can't avoid generating your pdfs from html, you should probably look around for some external tools or services for this purpose.

Good luck!

answered Dec 10 '08 at 19:44



12.7k • 6 • 44 • 60

I am trying the library from www.pd4ml.com

here is a code snip....

I pull in the html into the String content. Then I do a couple things like replace the radio buttons with checkboxes and then run it through the pd4ml library to render the pdf. The result looks pretty much like the original page....

```
String content = nextPage.generateResponse().contentString();
content = content.replace("Print", "");
content = content.replace("Back", "");
content = content.replace("border=\"1\"", "border=\"0\"");
content = content.replace("radio", "checkbox");
java.net.InetAddress\ i=java.net.InetAddress.getLocalHost();\\ String\ address=\ i.getHostAddress()+":53000";
content = content.replace("img src=\"/cgi-bin", "img src=\"http://"+address+"/cgi-bin");
System.out.println(content);
PD4ML html = new PD4ML();
html.setPageSize( new java.awt.Dimension(650, 700) );
html.setPageInsets( new java.awt.Insets(30, 30, 30, 30) );
html.setHtmlWidth( 750 );
html.enableImgSplit( false );
html.enableTableBreaks(true);
StringReader isr = new StringReader(content);
baos = new ByteArrayOutputStream();
html.render(isr, baos);
PDFRegForm pdfForm = (PDFRegForm)pageWithName("PDFRegForm");
pdfForm.baos = baos;
pdfForm.generateResponse();
```

answered Dec 17 '08 at 19:15 Adam Yocum

Have you tried looking at html2pdf? the home page is http://html2pdf.seven49.net/Web/

There is also a sourceforge project of it.

Alan

answered Dec 17 '08 at 20:09



## HTMLDOC?

- converts HTML to PDF;
- Free and Open Source (commercial support available);
- · can be used as...
  - ...standalone application,
  - ...for batch document processing,
  - ...or run as a web-based service;
- interface is CLI or GUI.

edited Oct 7 '15 at 12:06

answered Jun 10 '10 at 19:56





Not specific to Ruby, but the best solution I've found to this is the open source HTMLDOC.

edited Oct 7 '15 at 12:09 Aleksandr Kovalev **1,145** • 2 • 12 • 25 answered May 8 '09 at 13:26 Travis Beale 3,004 • 5 • 24 • 33