

Jakub - PDF

Have used two gems to date

1. gem 'prawn'
2. gem 'wicked_pdf'

I think wicked_pdf is better. I have given up using 'prawn'.

They both need the <https://wkhtmltopdf.org> library being installed.

We need an improved method for generating PDFs. Generally they take a long time to run.

Current action is,

1. produce a view
2. view contains a 'report' button to create the equivalent PDF
3. click report
4. action in controller that responds to HTML or PDF
5. PDF page is displayed in new browser tab

Might be nice to have a new tab open, spinner to show PDF being produced then replaced by PF when it is ready. Use the same mechanism for all PDFs. Note the PDF is the equivalent of a common front page (same for all reports with small differences like a title) with the pages being what was displayed in the view.

So updated action

1. produce a view
2. view contains a 'report' button to create the equivalent PDF
3. click report
4. action in controller that directs to busy page in new tab
5. 'Busy' view with spinner, start report
6. action in controller that responds to produce report
7. PDF page is displayed in the busy browser tab

The code below is an example of the current implementation. It may not be the best way, it was put together quickly but gives you an example. All reports based on the Report::WickedCore class which provides common functionality. The pdf.css file provides the bootstrap styling for the PDFs

Example controller action

```
def changes_report
  authorize CdiscTerm, :view?
```

```
results = CdiscCtChanges.read(CdiscCtChanges::C_ALL_CT)
cls = CdiscTerm::Utility.transpose_results(results)
respond_to do |format|
  format.pdf do
    @html = Reports::CdiscChangesReport.new.create(results, cls, current_user)
    render pdf: "cdisc_changes.pdf", page_size: current_user.paper_size, orientation:
'Landscape'
  end
end
end
```

View

```
<%= wicked_pdf_stylesheet_link_tag "pdf.css" %>
<%= @html.html_safe %>
```

Other Files



Generic File



Generic File



Generic File