PivotalResearch

Requirements

Module: ppr.reporting

Author: Qia Chen

Date: January 27, 2014

Contents

Overview

Requirements

1.Create models ReportType, Report

2.Allow privileged users to upload Fund report

3.Users can view the existing types of reports

4.Create module ppr.util.pdf to handle with the PDF file displaying

5.Create configurable watermark for PDF file

Sign-Off

Overview

The ppr.reporting module is a Django “app” that contains the configuration and generation of the reports that will be accessible to the clients.

Create a ReportType model that will allow different report types per Fund

Requirements

1.Create models ReportType, Report

**Description**

ReportType model allows privileged user (Administrator, Support, Analyst T2) to create any numbers of different report types for each Fund instance.

Report model contains foreign keys ‘fund’ and ‘report\_type’ and file field ‘report\_file’. The file field points to the directory storing the real report file.

**Functional Tests**

**Manual Tests**

2.Allow Privileged users to upload Fund report

**Description**

The users that have the permission to upload Fund reports would be able to see the ‘upload’ link and access the upload page on the PivotalPath client site. The ‘upload’ link is invisible for any other users without the permission.

The privileged users (Administrator, Analyst T2) can select one report type for the specific Fund instance and upload the corresponding report. The only allowed file format for uploading is PDF.

The uploaded files will be stored in /media folder using Django File System Storage. The privileged users are allowed to re-upload any specific type of report. The new report will be stored in the same directory with the existing report. When the new PDF file is uploaded and stored, the original one will be removed.

**Functional Tests**

**Manual Tests**

3.Users can view the existing types of reports

**Description**

For each Fund instance, users would be able to choose a specific report type to view the corresponding PDF file. Only the report types on which there is report existing will show up and be clickable for users to view the PDF file.

**Functional Tests**

**Manual Tests**

4.Create module ppr.util.pdf to handle with the PDF file displaying

**Description**

The module ppr.util.pdf is an app used as utility for PDF generation and displaying.

Django template tag needs to be created for embedding PDF viewer in the base template.

Users would be able to view the PDF file, change the zoom and select page. The users with permission can download the PDF.

**Functional Tests**

**Manual Tests**

5.Create configurable watermark for PDF file

**Description**

The PDF app should allow for a watermark to be applied. This should be a dynamic process where the original PDF is "merged" with the watermark and served to the front end (via the template tag).

**Functional Tests**

**Manual Tests**

Sign-Off

**Name**

**Signature**

**Date**

Wyatt Anderson