Technical Design Document

55495 - Settings menu for Content Viewer

Contents

[1 Introduction 2](#_Toc523840627)

[1.1 Purpose 2](#_Toc523840628)

[1.2 Scope 2](#_Toc523840629)

[Tchnical Details 2](#_Toc523840630)

[2.1 Client Side Implementation 2](#_Toc523840631)

[2.2.1 Unit Test 3](#_Toc523840632)

# 1 Introduction

## 1.1 Purpose

This document provides technical details of user stories. Purpose of this document is to simplify implementation of technical stories for developer.

## 1.2 Scope

The scope of this document is to provide **server and client side technical details of “Setting” of Standalone Document WorkflowItemViewer.**

Server side mostly all methods are already available related workflowuser and userconfig. Below is link of FD.

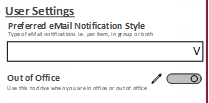
<https://operations.cloud.papersave.com/PaperSave.WebModule/DocumentDisplay.aspx?DocIDs=260608>

# Tchnical Details

## 2.1 Client Side Implementation

To achieve this design we have to render “settings” component. “Setting” component is container component of various component. For this page we will rendered following component.

1. workflowusersetting.component (this one is already there. No implementation required)
2. tablefieldsetting.component
3. annotationsetting.component
4. Render workflowusersetting component.



**No implementation required to achieve this from client and server side. It is already there we just need to render this. It will automatically represents value of workflowuser notification style and Out of office settings**.

1. Create component called tablefieldsetting.component.ts and html under shared\settings folder.



Component contains header and option button only. User can select any option at a time.

1. Add following two UserConfigConstant in constants. Class is already exists there so, just need to add this property.

export module UserConfigConstants {

export class Global{

static readonly TableFieldView: string = "tablefieldview";

}

1. While rendering we need to check if UserConfigs contains TableFieldView value in it or not. application\_id will be null. If it doesn’t contain any values then we need to use “Card Style” as default value.

On change of option button we need to call updateconfig call.

//application id will be null

//config name will be UserConfigConstants.Global.TableFieldView

//transaction type id would be null in this case.

//config value will be “Card” or “Grid”

saveUserConfigByName(applicationID,configName,configValue,null)

1. Create following class to support annotation options

export class AnnotationOptions {

annotationtype: AnnotationType;

fillcolor:string;

bordercolor:string;

borderthickness:number;

fontname:string;

fontsize:number;

textbackground:string;

textforeground:string;

defaulttext:string;

verticalalignment : AnnotationAlignment.Vertical;

horizontalalignment:AnnotationAlignment.Horizontal;

wordwrap:boolean;

}

export enum AnnotationType {

Text,

Note,

Pen,

Circle,

Line,

Polygone,

Rectangle,

Arrow,

Highlighter

}

export module AnnotationAlignment{

export enum Vertical{

Top,

Center,

Bottom

}

export enum Horizontal{

Left,

Center,

Right

}

}

1. Create annotationoption.component.ts and html. This component will have ability to show any annotation properties based on annotation type.

<https://ej2.syncfusion.com/javascript/demos/#/material/color-picker/default.html>

<https://ej2.syncfusion.com/javascript/demos/#/material/dropdownlist/default.html>

in HTML we need to render control based on annotation type. i.e. see below example

<app-annotationoption annotationType=AnnotationType.Text> </app-annotationoption>

Here annotation type is “Text” so, we need to rendered this control by folowing (a).

* 1. Text Annotation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Papersave | LeadTools PropertyName | Render | Default | Notes |
| Fill | Fill | False | Transperent | This three property we will not render in Text annotation. |
| Color | Strock | False | Transperent |
| Thickness | Thickness | False | 0 |
| Font | FontFamilyName | True | Calibri | Size will render right side of font. Check FD for image |
| Size | SizeInPoints | True | 10 |
| Color | Foreground | True | Black |  |
| Background | Background | True | Transperent |  |
| Text | Text | True | Blank Value |  |
| Vertical | VerticalAlignment | True | Top |  |
| Horizontal | HorizontalAlignment | True | Left |  |

* 1. Note Annotation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Papersave | LeadTools PropertyName | Render | Default | Notes |
| Fill | Fill | True | Transperent | This three property we will not render in Text annotation. |
| Color | Strock | True | Transperent |
| Thickness | Thickness | True | 1 |
| Font | FontFamilyName | True | Calibri | Size will render right side of font. Check FD for image |
| Size | SizeInPoints | True | 10 |
| Color | Foreground | True | Black | **This information is missing from FD**. So, get it from leadtools. It may change later whenever related FD comes for this. |
| Background | Background | True | Yellow |
| Text | Text | True | Blank Value |
| Vertical | VerticalAlignment | True | Top |
| Horizontal | HorizontalAlignment | True | Left |

* 1. Pen Annotation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Papersave | LeadTools PropertyName | Render | Default | Notes |
| Color | Strock | True | Blue | only this two property should rendered. Rest of property will be false. |
| Thickness | Thickness | True | 1 |

* 1. Circle Annotation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Papersave | LeadTools PropertyName | Render | Default | Notes |
| Fill | Fill | True | Transperent | Only this three property should rendered. Rest of property will be false. |
| Color | Strock | True | Red |
| Thickness | Thickness | True | 1 |

* 1. Line Annotation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Papersave | LeadTools PropertyName | Render | Default | Notes |
| Color | Strock | True | Blue | only this two property should rendered. Rest of property will be false. |
| Thickness | Thickness | True | 1 |

* 1. Polygone Annotation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Papersave | LeadTools PropertyName | Render | Default | Notes |
| Fill | Fill | True | Transperent | Only this three property should rendered |
| Color | Strock | True | Blue |
| Thickness | Thickness | True | 1 |

* 1. Rectangle Annotation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Papersave | LeadTools PropertyName | Render | Default | Notes |
| Fill | Fill | True | Transperent | Only this three property should rendered |
| Color | Strock | True | Blue |
| Thickness | Thickness | True | 1 |

* 1. Arrow Annotation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Papersave | LeadTools PropertyName | Render | Default | Notes |
| Fill | Fill | True | Transperent | Only this three property should rendered |
| Color | Strock | True | Green |
| Thickness | Thickness | True | 1 |

* 1. Highlighter Annotation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Papersave | LeadTools PropertyName | Render | Default | Notes |
| Hilite (Fill) | Hilite | True | Yellow | Only this three property should rendered |

1. annotationoption will have method for for getting updated annotationOption object. So, In case of any change in value that method will return dirty object.
2. json text of that object will be our userConfig Value and annotation type will be userConfig Name. Application\_ID will be null.
3. We can keep single config for all annotation setting but then object will be very heavy when we introduce signature, stamp & redaction annotation detail. So, better to keep it isolated. By that way we can add/update property later on whenever its required.
4. Create annotationsettings.component.ts and html under shared\settings folder. Annotation settings is container component for annotationoptions. As of now we will render following annotation options. Based ony type it will rendered control in slider view. Annotationsetting will be slider view which will be open from link of setting component. It will contain following instance of annotation option in it. By following this way we are utilizing same code for multiple annotation option.

<app-annotationoption annotationType=AnnotationType.Text> </app-annotationoption>

<app-annotationoption annotationType=AnnotationType.Note> </app-annotationoption>

<app-annotationoption annotationType=AnnotationType.Pen> </app-annotationoption>

<app-annotationoption annotationType=AnnotationType.Circle> </app-annotationoption>

<app-annotationoption annotationType=AnnotationType.Line> </app-annotationoption>

<app-annotationoption annotationType=AnnotationType.Polygone> </app-annotationoption>

<app-annotationoption annotationType=AnnotationType.Arrow> </app-annotationoption>

<app-annotationoption annotationType=AnnotationType.Highlighter> </app-annotationoption>

1. Annotationsettings will contain two more button at the bottom “Save” & “Cancel”.
2. Save will get modified userconfigs from component and will call userconfig Addupdate method. Method is already there so No code change require there.
3. Cancel will close slider and does nothing.