PaperSave 7.0

Technical Design Document

View layout and resize options - [48883](https://tfs.papersave.com/tfs/PaperSave/PaperSave5.3/_workitems?id=48883&_a=edit)

(Zoom In & Zoom Out, Actual Size, Fit Visible, Fit Width, Fit Height, Custom Zoom)

Contents

[1 Introduction 2](#_Toc519792479)

[1.1 Purpose 2](#_Toc519792480)

[1.2 Scope 2](#_Toc519792481)

[2 Technical Story : Implementation of content view and resize options. 3](#_Toc519792482)

[2.1 Clint Side Implementation 3](#_Toc519792483)

[2.1.1 Implementation of Unit Test 4](#_Toc519792484)

# 1 Introduction

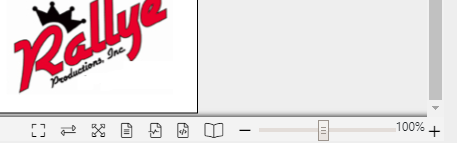
## 1.1 Purpose

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

## Scope

The scope of this document is to provide technical details for following features.

* + implementation of Actual Size
  + Implementation of Fit Width
  + Implementation of Best Fit
  + Implementation of Single Page View
  + Implementation of Horizontal Page View
  + Implementation of Vertical Page View
  + Implementation of Double Page View
  + Implementation of Zoom In, Zoom Out slider. (Custom Zoom)



Changes are almost same for each feature. So, I have merge those in one technical story. We should import leadtools package in angular client. They provides client side component which will communicates to Content API. **There Is no server side change require for this technical story.**

# 2 Technical Story : Implementation of content view and resize options.

## 2.1 Clint Side Implementation

In earlier version of PaperSave, we were having these buttons in ribbon bar. In PaperSave 7.0, these features should be part of content viewer web app. So, if we bind this image viewer in some other app it still works.

|  |  |  |  |
| --- | --- | --- | --- |
| Button | Tool Tip | Control Type |  |
| Actual Size | Show the actual size of the page | State Button | Group 1 |
| Fit Width | Fit the image width into the window | State Button |
| Best Fit | Fit the image into the window | State Button |
| Custom Zoom | Zoom | Slider Bar Control |
| Single Page | Single Page Display | State Button | Group 2 |
| Horizontal Page | Horizontal Page Display | State Button |
| **Vertical Page** | Vertical Page Display | State button |
|  |  |  |  |

* + Create state button in exactly in same order as mention in Image(under scope section).
  + There will be no **Fit height** button as lead tools don’t have it separately. Best fit will coverup that requirement.
  + Create slider bar by combining Zoom In, Zoom Out & Custom Zoom. After creating slider bar there is no requirement of separate Zoom In, Zoom out buttons.
  + There should be two group basically for state buttons. So, user can select any one option from each group.
  + Actual Size, Fit Width & Best Fit will be group 1. User can select any one state from this group. Default will be best fit. So, in absense of userconfig bestfit will be default behaviour.
  + Single Page, Horizontal Page, Vertical page will be Group 2. User can select any one state from this group. Default will be “Vertical Page”. So, in absense of userconfig it will render in vertical page view.
  + Button state should be stored in database by PaperSave API under UserConfig table.
  + Following code is sample that how to bind events for any of these state button. I.e. BestFit

btnBestFit.addEventListener('click', function () { documentViewer.commands.run(lt.Document.Viewer.DocumentViewerCommands.viewFitPage, null);

setviewfocus(this);

});

Enum “lt.Document.Viewer.DocumentViewerCommands” contains all action which needs to be performed. All the events needs to handle in contentviewer.ts file under PaperSave.Server.ContentAPI project.

* + We need to preserve state in UserConfig. To achieve that we should post “action” from contentviewer.ts to contentviewer.component.ts(angular) . “shared\component\contentviewer.component.ts”.
  + Based on incoming “Action” we need to call api for Add or Update UserConfig. This api will add or update based on action clicked. In Angular app, Get, Insert, Update methods are already available for UserConfig under userconfig.service.ts file. So, just need to pass value & configName. In case of CustomZoom need to pass Zoom level as well. While retrival we will split string to get zoom value.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Value | Application\_Id | TransactionType\_Id |
| ZoomMode | BestFit  ActualSize  FitWidth  CustomZoom:127 | 1 | Null |
| PageLayout | SinglePage  HorizontalPage  VerticalPage | 1 | null |

* + Once userconfigs get saved it will be available in userconfig.service.ts in angular app. So, next time while setting iframe we will pass these values to contentviewer page. Content viewer should render document based on these values.

## 2.1.1 Implementation of Unit Test

* + Write unit test for contentviewer.component.ts(angular) & contentviewer.ts(content api).

Every component will contains spec.ts file.

Angular project has already configuration of Karma & Jasmin for unit test.

For, ContentAPI project, we need to use jasmin as well.

<https://angular.io/guide/testing>