SET Dashboard

Administration Guide

# Overview

This document will give an overview of the SET Dashboard solution, designed for SharePoint 2013/Online environments.

# SharePoint Lists

The solution list configuration can be found in the src/cfg.ts source code file.

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| Announcements | Announcement | Used to display announcements above the dashboard. |
| Line of Effort | Generic | Defines the Focus Areas, Goals and Lines of Efforts. Referenced by the main solution. |
| Objectives | Document Set | Each folder in the document library is used to store information for each objective. |
| Tasks | Task | Stores the objective’s tasks. |
| Comments | Generic | Stores the comments for the objectives. |

# Deployment

## Solution Files

The solution files should be uploaded to the “Site Assets” folder, under the “set-dashboard” sub-folder.

|  |  |  |
| --- | --- | --- |
| Name | Description | Target Folder |
| index.html | The reference file for the content editor webpart. | [web]/siteassets/set-dashboard |
| set-config.json | The configuration file for the solution. | [web]/siteassets/set-dashboard |
| set-dashboard.js | The bundled un-minified solution JavaScript file. (Test Environment) | [web]/siteassets/set-dashboard |
| set-dashboard.min.js | The bundled minified solution JavaScript file. (Prod Environment) | [web]/siteassets/set-dashboard |
| af-logo-green.png | Referenced by the configuration file. | [web]/siteassets/set-dashboard/images |
| af-logo-red.png | Referenced by the configuration file. | [web]/siteassets/set-dashboard/images |
| af-logo-yellow.png | Referenced by the configuration file. | [web]/siteassets/set-dashboard/images |
| af-logo.png | Referenced by the configuration file. | [web]/siteassets/set-dashboard/images |
| AFGSC.png | Referenced by the configuration file. | [web]/siteassets/set-dashboard/images |

## Installation

### Upload Solution Files

Load the SharePoint site to deploy the solution to. Access the “Site Assets” library and create a folder called “set-dashboard”. Upload the files defined in the previous section.

### Install Solution (Automated)

The solution can be installed from the JavaScript library. This will require you to view a “classic” page (if you are in a SharePoint Online environment). Press F-12 or Ctrl+Shift+I to access the development tools. Click on the “Console” tab and load the library.

*var s = document.createElement(“script”); s.src = “[web]/siteassets/set-dashboard/set-dashboard.min.js”; document.head.appendChild(s);*

Once the library has loaded, you can run “SETDashboard.Configuration.install()” to Create the SharePoint lists and configure the dashboard. This command can be run again to ensure all fields and views are configured correctly.

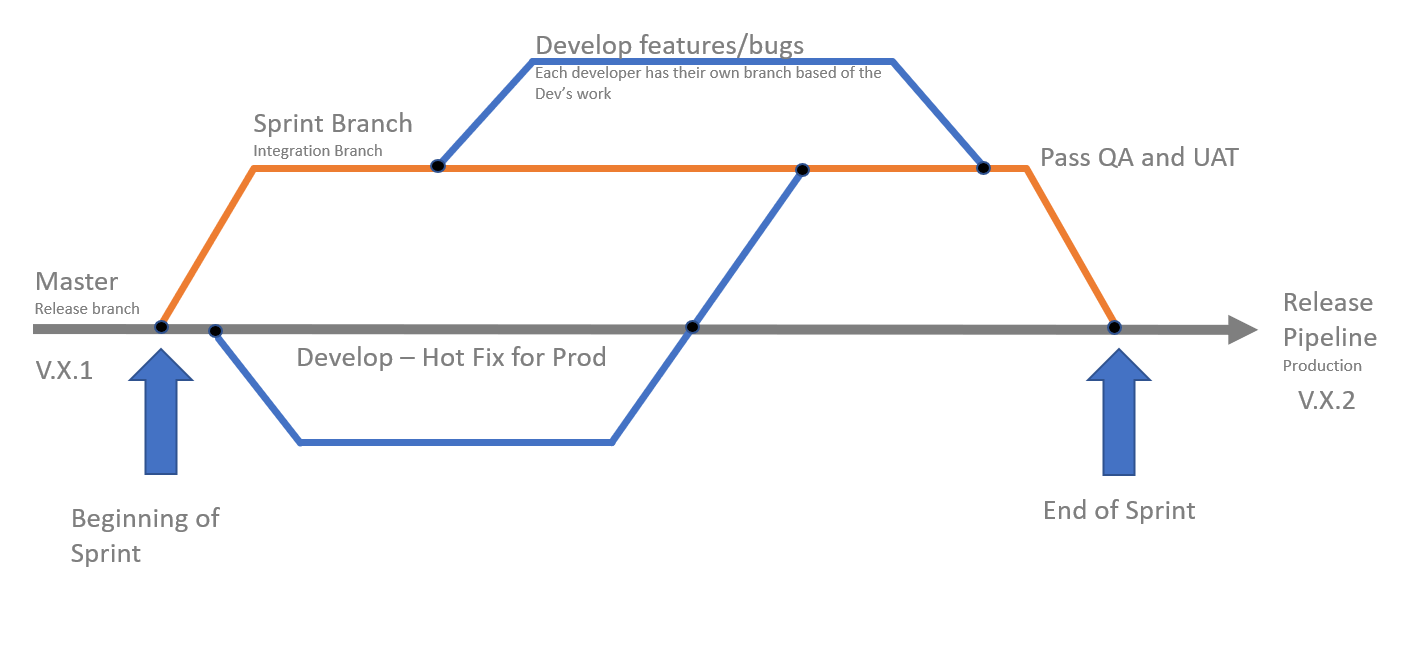
### Configure Document Set Dashboard

Access the “Objectives” document library settings and click on the advance settings. Select the “Classic” experience for the “List experience” setting and click “OK” to save the changes. Under “Content Types” select the “Workspace” link to view the content type details and click on the “Document Set settings”. Detailed instructions on how to customize the Document Set can be found in **Appendix D**

Upload files to the “Default Content” settings to be made available after creation of an objective. Click on the “Customize the Welcome Page” link at the bottom of the page to access the document set dashboard page. Once the dashboard page has loaded, edit the page and remove the default Logo and List Properties webparts.

# Azure DevOps Branch Strategy

The goal of the Azure DevOps Strategy is to ensure high quality code is being developed, and changes are tracked, and code review is performed before merging code. We are going to use the Sprint Branch strategy with a code release at the end of the sprint.



In the flow above we are going to do the following:

1. Start the sprint with a new branch based of Master and name it Sprint##. This branch will be created on the ADO site.
2. Based on that remote branch, every developer can create their own branch that will hold the code that they are developing this sprint. It can be features, bugs etc. To do so, see Appendix C. for Git Bash commands.
3. Once the developer is done with their work, they need to submit a Pull Request (PR) to Sprint branch and the code review will be scheduled with a lead developer. If code is good and integrates well, code is merged into the sprint branch.
4. Code is then deployed to test environment and, sprint demo is performed to the PO (Product Owner) that “accepts” the user stories.
5. At this point, UAT is done and code is ready to be promoted to Production. Once that is done Pull Request is submitted to master branch, and sprint branch is deleted.
6. New branch is created, to hold the following sprint, developers can create feature branches of the new sprint branch for that sprint’s features, bugs, etc.

# Solution

## Timeline

### Configuration

The TimelineJS library plugin has options available to better configure the solution. These options are available [here](https://timeline.knightlab.com/docs/options.html) and in the appendix. The set-config.json file has an “options” property which is an object corresponding to the available TimelineJS library plugin options. This will help the client team adjust the settings to ensure the solution renders appropriately in their environment.

## Dashboard

## Workspace

# Appendix

## SharePoint Lists

## TimelineJS Options

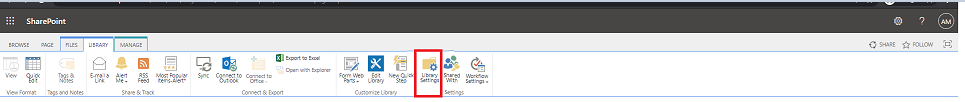
|  |  |  |
| --- | --- | --- |
| Name | Default Value | Description |
| debug | FALSE | If true, copious console logging will be enabled. |
| height | this.\_el.container.offsetHeight | The height of the timeline. |
| width | this.\_el.container.offsetWidth | The width of the timeline. |
| is\_embed | FALSE | If true, the class tl-timeline-embed is added to the outer Timeline container. Typically only used to support Timeline iframe embeds. |
| hash\_bookmark | FALSE | If set to true, TimelineJS will update the browser URL each time a slide advances, so that people can link directly to specific slides. |
| default\_bg\_color | white | RGB values to use for slide backgrounds. Specify as hex code, CSS named color, or a Javascript object with r, g, and b properties from 0-255. |
| scale\_factor | 2 | How many screen widths wide the timeline should be at first presentation. |
| initial\_zoom | | The position in the zoom\_sequence series used to scale the Timeline when it is first created. Takes precedence over scale\_factor. |
| zoom\_sequence | [0.5, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89] | Array of values for TimeNav zoom levels. Each value is a scale\_factor, which means that at any given level, the full timeline would require that many screens to display all events. |
| timenav\_position | bottom | Display the timeline nav on the top or bottom. |
| optimal\_tick\_width | 100 | Optimal distance (in pixels) between ticks on axis. |
| base\_class | tl-timeline | Removing the tl-timeline base class will disable all default stylesheets. |
| timenav\_height | 150 | The height in pixels of the timeline nav. Takes precedence over timenav\_height\_percentage. |
| timenav\_height\_percentage | 25 | Specify the timeline nav height as a percentage of the screen instead of in pixels. |
| timenav\_mobile\_height\_percentage | 40 | Specify the timeline nav height as a percentage of a mobile device screen. |
| timenav\_height\_min | 150 | The minimum timeline nav height (in pixels). |
| marker\_height\_min | 30 | The minimum marker height (in pixels). |
| marker\_width\_min | 100 | The minimum marker witdh (in pixels). |
| marker\_padding | 5 | Top and bottom padding (in pixels) for markers. |
| start\_at\_slide | 0 | The first slide to display when the timeline is loaded. |
| start\_at\_end | FALSE | If true, loads timeline on last slide. |
| menubar\_height | 0 |  |
| use\_bc | FALSE | Use declared suffix on dates earlier than 0. |
| duration | 1000 | Animation duration (in milliseconds). |
| ease | TL.Ease.easeInOutQuint | |
| dragging | TRUE |  |
| trackResize | TRUE |  |
| slide\_padding\_lr | 100 | Padding (in pixels) on the left and right of each slide. |
| slide\_default\_fade | 0% |  |
| language | en |  |
| ga\_property\_id | null | Google Analytics ID. |
| track\_events | ['back\_to\_start', 'nav\_next', 'nav\_previous', 'zoom\_in', 'zoom\_out'] | |
| script\_path | | Can be used to help Timeline load related resources such as CSS themes and language files. Rarely needs to be set. |

## Git Commands

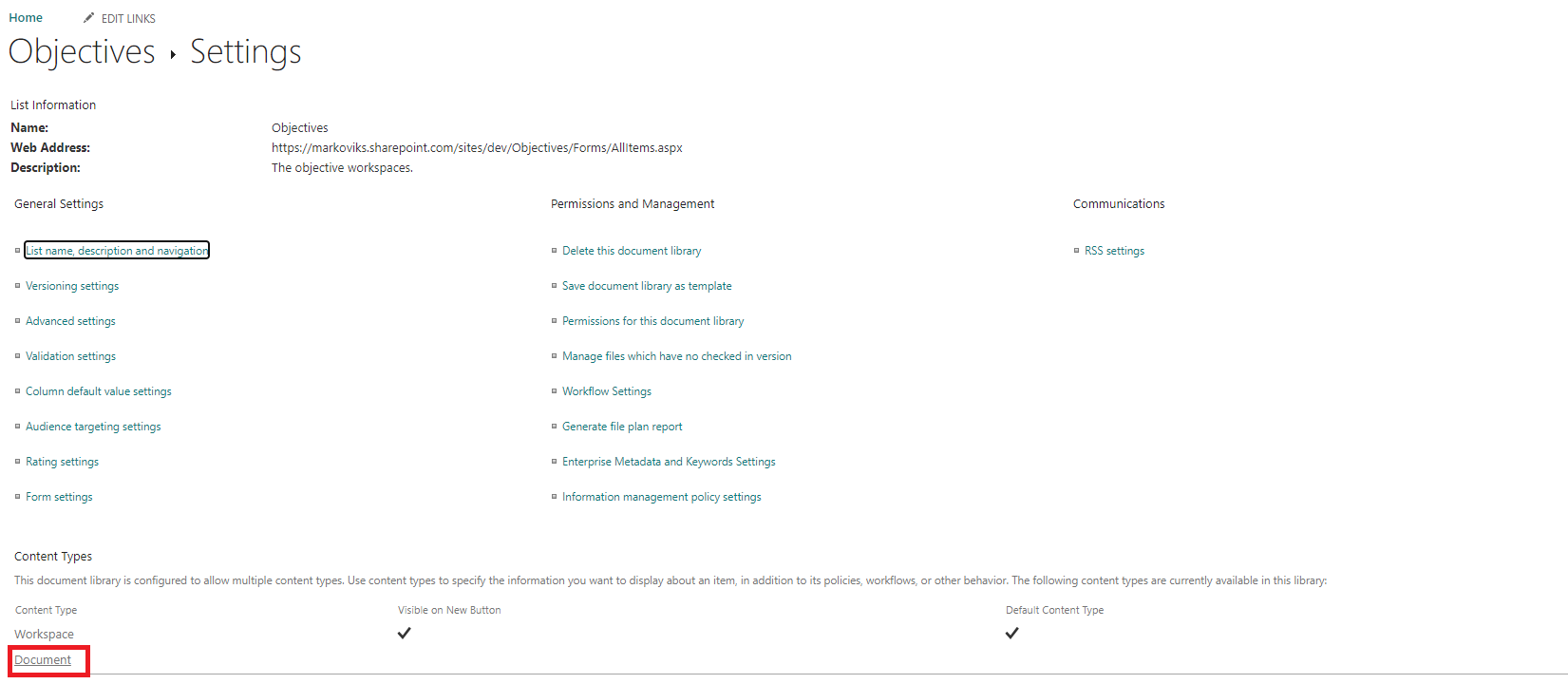
|  |  |  |
| --- | --- | --- |
| Git task | Notes | Git commands |
| Tell Git who you are | Configure the author name and email address to be used with your commits.  Note that Git (for example trailing periods) from user.name. | git config --global user.name "John Doe"  git config --global user.email jdoe@example.com |
| Create a new local repository |  | git init |
| Clone repository  (Creates all branch from remote) | For a remote server, use: URL from the Clone button in ADO | git clone username@host:/path/to/repository |
| List all branches | List all the branches in your repo, and tells you what branch you're currently in: | git branch |
| Change branch | Switch from one branch to another. Example: sprint\_branch\_feature\_a | git checkout <branchname> |
| Add files | Add one or more files to staging | git add <filename>  git add \* |
| Commit | Commit changes to local (but not yet to the remote repository): | git commit -m "Commit message" |
| Push | Send changes to the sprint\_branch\_feature\_a branch of your | git push origin <here goes the branch name> |
| Fetch | Downloads all the branches and changes from the remote to local repo | git fetch origin |
| Status | List the files you've changed and those you still need to add or commit: | git status |
| Delete Branch | Delete the feature branch: | git branch -d <branchname> |
|  |  |  |
|  |  |  |

Configure Doc Set – Upload Documents Metadata and View:

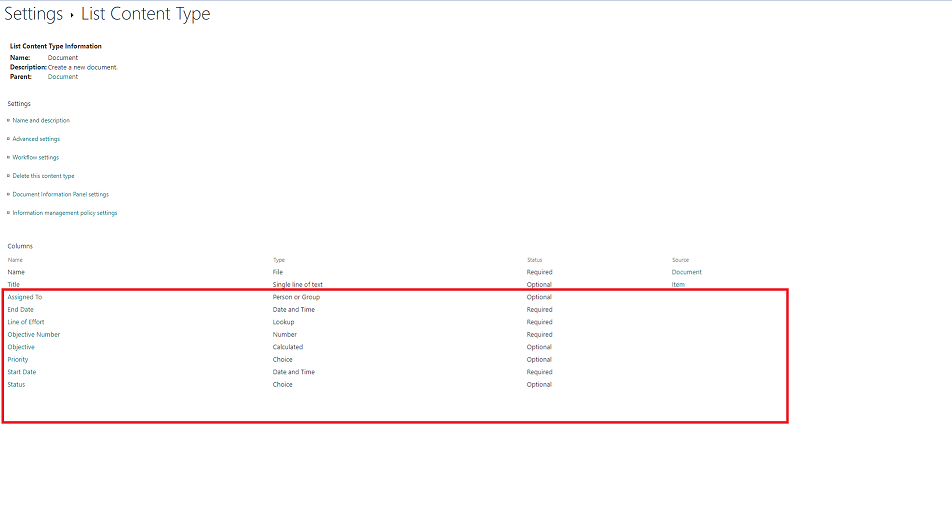
1. From the menu bar on the Objectives Library, select Library Settings



1. Scroll down, and edit the columns in the Content Type:

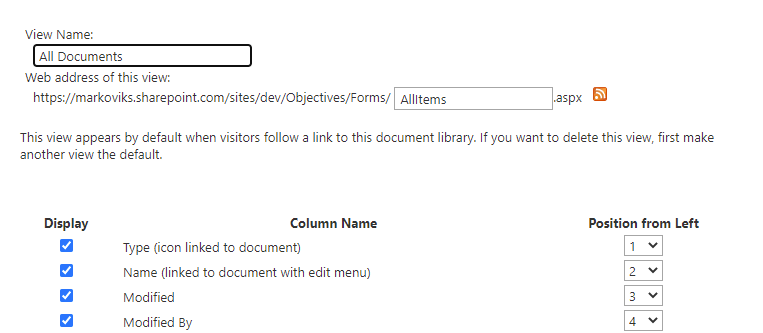


1. Remove all columns except, Title and Name

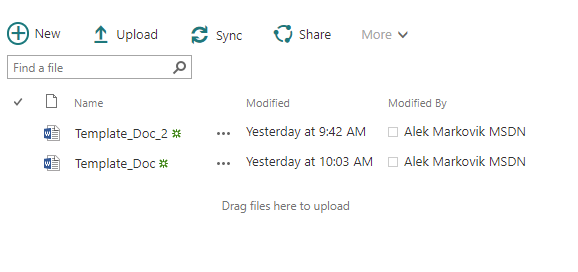




1. Fix, default Objectives Library default view to show only: **Type (icon linked to document) Name (linked to document with edit menu), Modified and Modified By**.



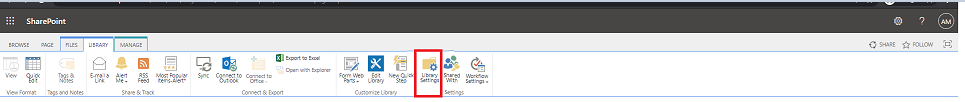
1. Final View, will be as shown below from the Workspace of an Objective.



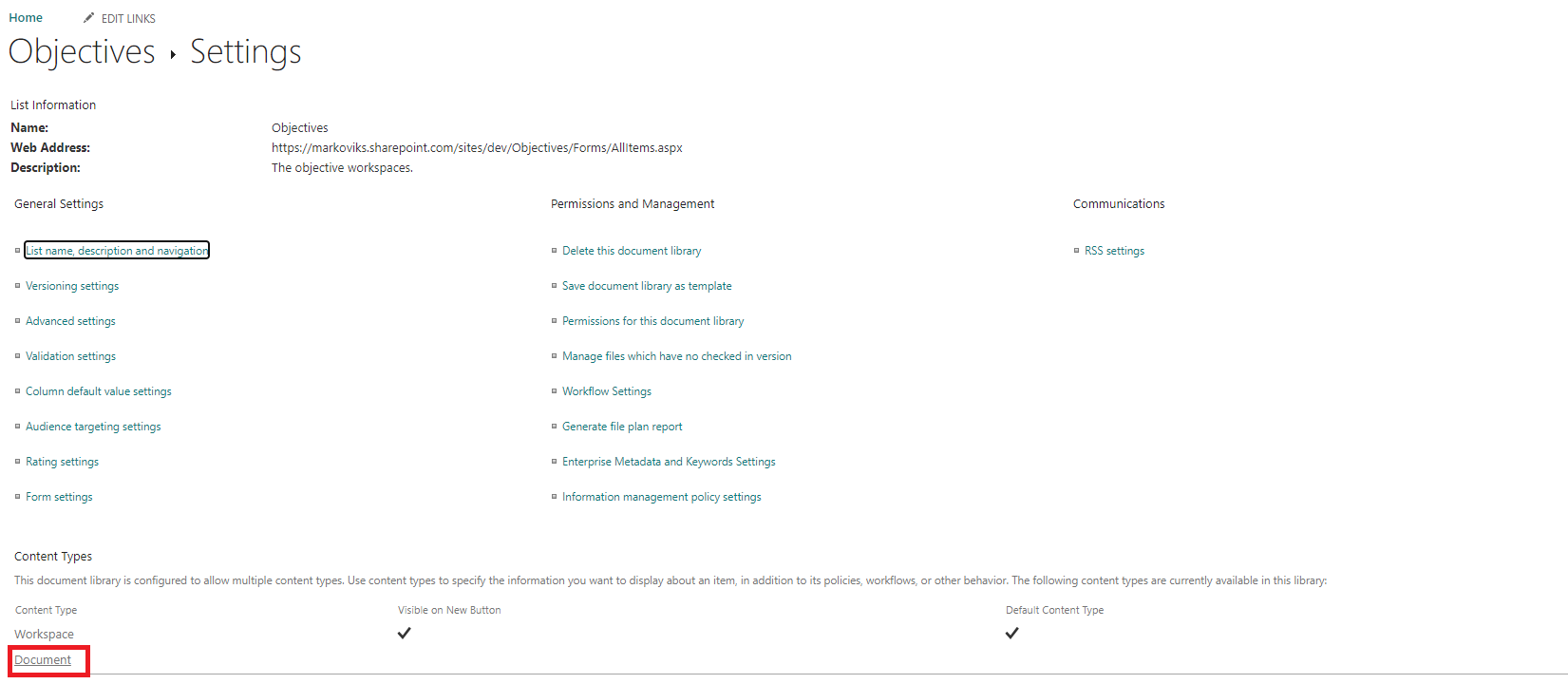
## Configure Doc Set

### Adjust Documents view

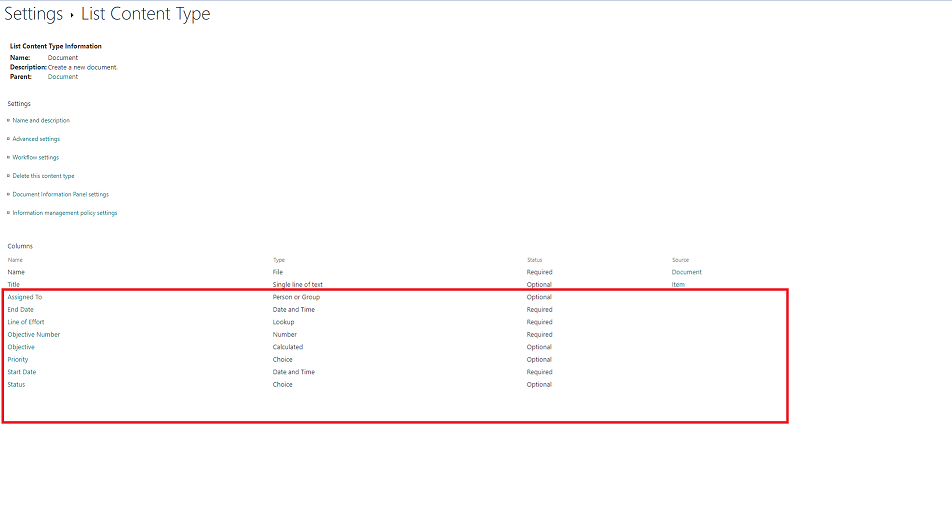
1. From the menu bar on the **Objectives Library**, select **Library Settings**



1. Scroll down, and edit the columns in the Content Type:

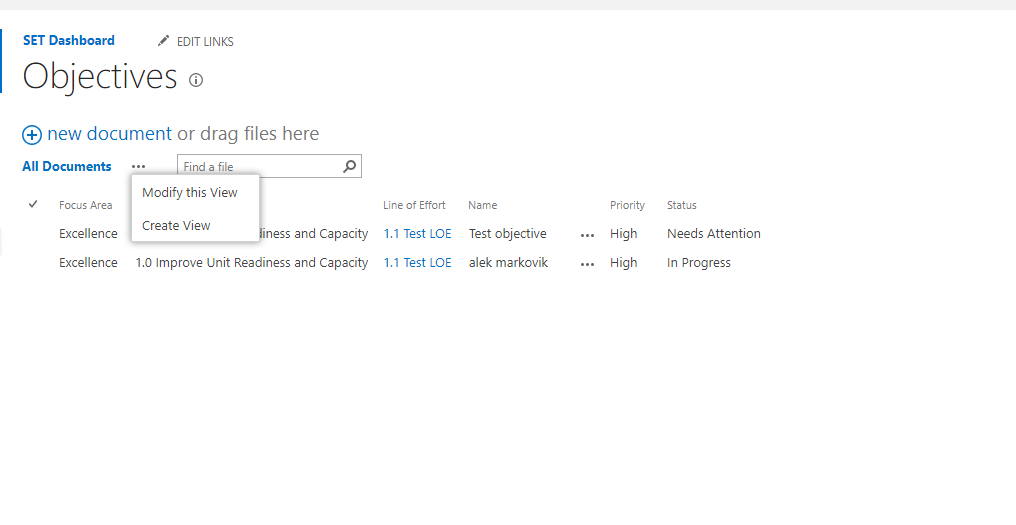


1. Remove all columns except, **Title and Name**

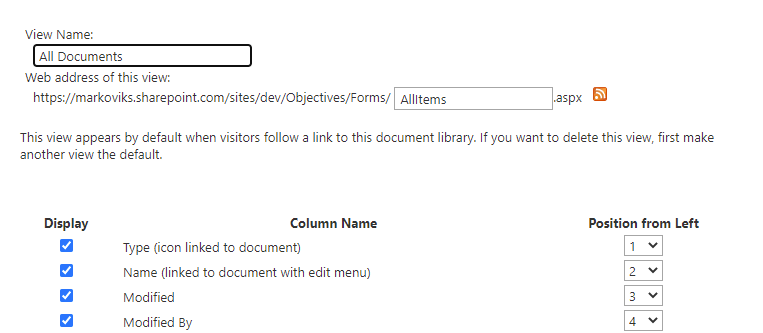




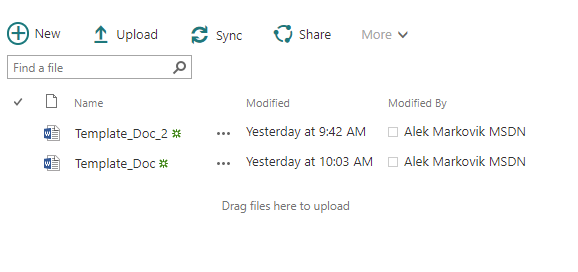
1. Fix, default Objectives Library default view to show only: **Type (icon linked to document) Name (linked to document with edit menu), Modified and Modified By**.
   1. Go to Objectives Library and press modify this view



* 1. Set the view as below



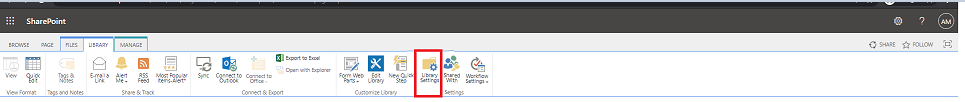
1. Final View, will be as shown below from the Workspace of an Objective.



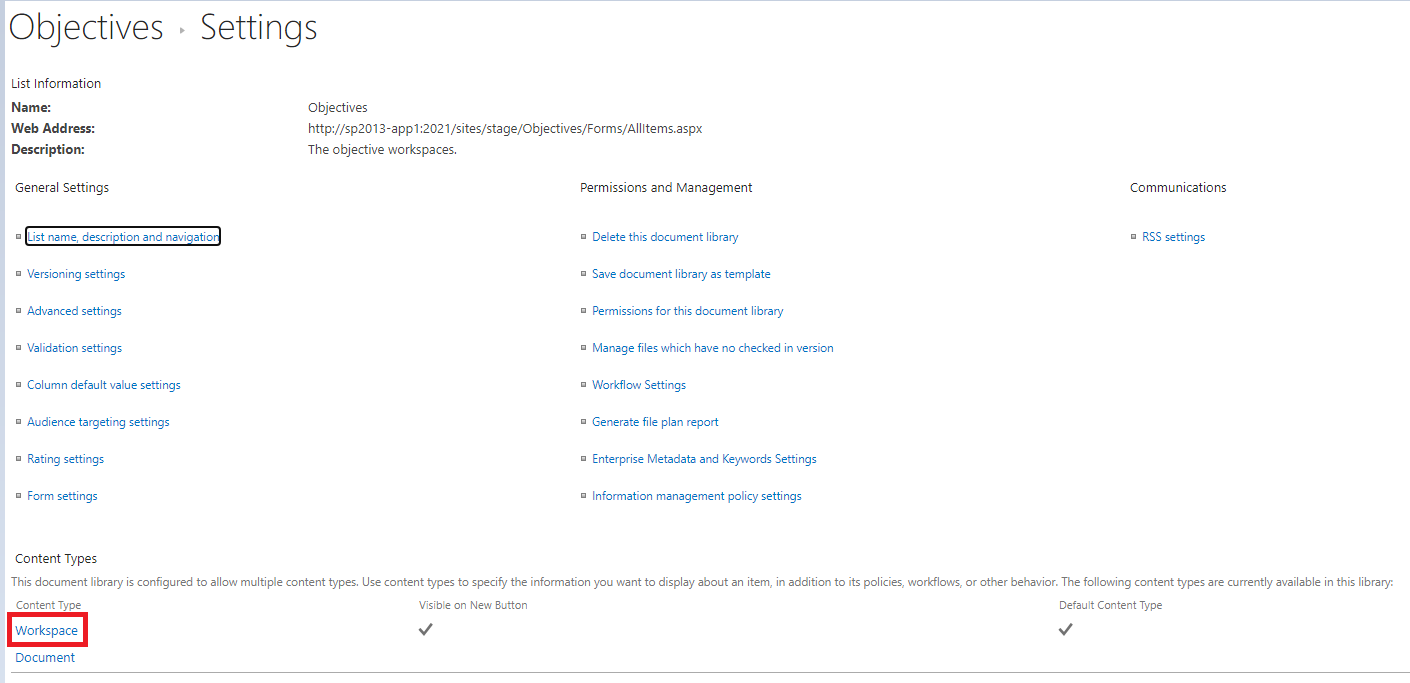
### Configure Template Files and Folders

To provide default documents to users in a workspace, we need to upload them to the Workspace Content type on the Objectives library. To that we need to:

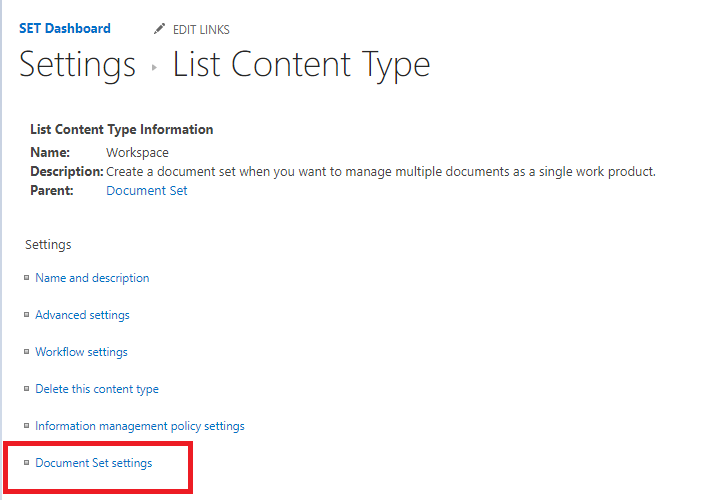
1. From the menu bar on the **Objectives Library**, select **Library Settings**



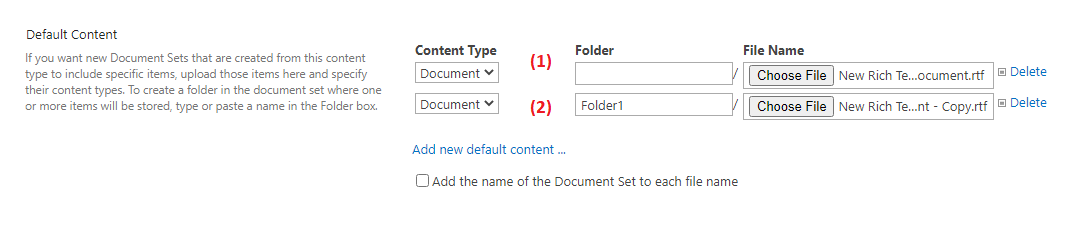
1. Select the Workspace Content Type



1. Select Document Set settings



1. On the Default Content Set the documents that you want appear in the workspace, for each Objective.



Note: You can add to the root by uploading to Folder empty (1), or you can create folders you typing the name of the folder in Folder Box (2)