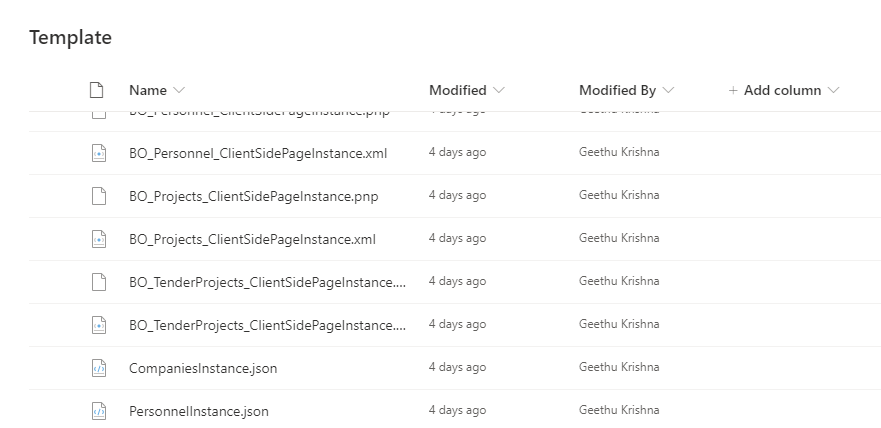
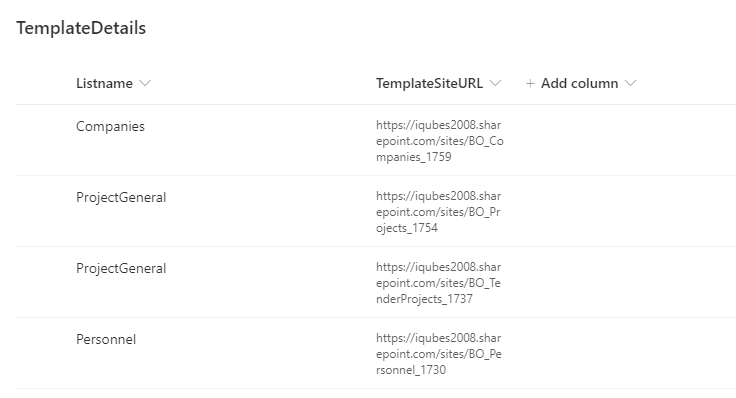
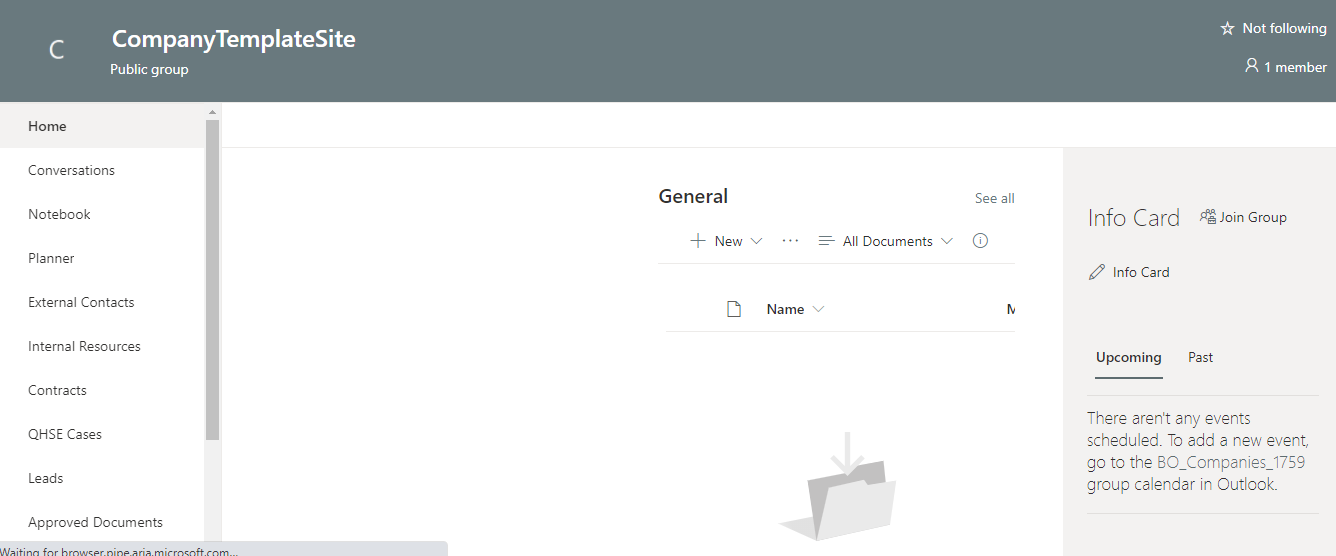
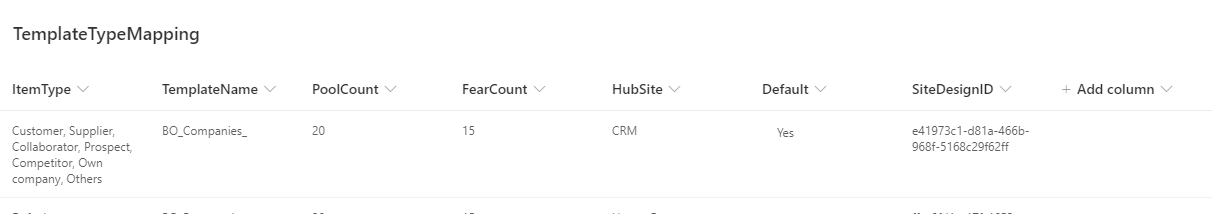
**Steps to Configure the Site pooling**

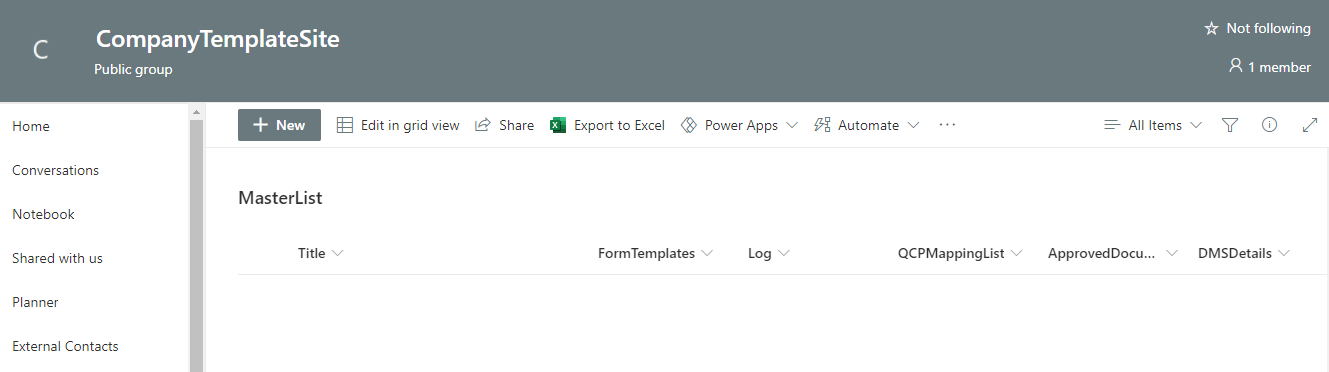
* List Creation Details are listed below  
  **Template**This list is used for holding the pnp files, latest clientsidepage xml files and json files  
    
  **Template Details**This list is used for getting the template site URL details



* **Take the latest site from the pool and set as template site. Do all the template changes and this WS is used for the generation of site design id and pnp files.**
* **Add column in “TemplateTypeMapping” list named “SiteDesignID”. Set the corresponding site design ID here**



* **“Master” list created in all template WS.** **List contains a number of lookup columns (lookup to all lists in WS).** **This master list is used to create copies of the lookup lists in New WS. Created all the lookup columns of lists that you want to add to the new sites. List Schema is generated based on this master list**



* **Take the client side page xml from template sites using below code**

#connecting the template site  
Connect-PnPOnline -Url <https://iqubes2008.sharepoint.com/sites/BO_Companies_1759>  
#Getting the xml from templates WS and saved into a local path

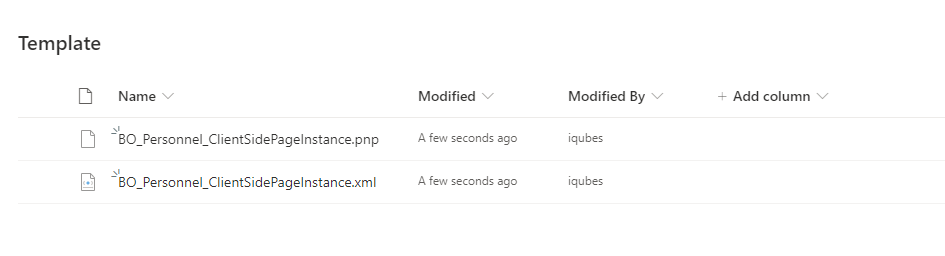
Get-PnPProvisioningTemplate -Out   
C:\Users\Geethu\Desktop\SiteScript\iQubeS\TemplateXmls\BO\_Companies\_ClientSidePageInstance.xml –IncludeAllClientSidePages

* **Generated Xml files upload to the Templates DL in admin Site**
* **Below code is used to Converted xml files to pnp in the local path and uploaded to template document library**

**#**Creates a pnp package file of an existing template xml and includes all files in the C:\ProjectXml folder

Convert-PnPFolderToProvisioningTemplate -Out C:\ProjectXml\BO\_Projects\_ClientSidePageInstance.pnp -Folder C:\ProjectXml

* **XML and PNP Files uploaded to the Template Library in Admin Site**



**SiteScripGeneration**

* **Below code is used to generate site script from list  
  #Initialize the basic details**

$username = [system.account.prd@iqubes.no](mailto:system.account.prd@iqubes.no)

$password = "hrmkzcppfvmdxfkp"

$tenantAdminUrl = <https://iqubes2008-admin.sharepoint.com>

$encPwd = ConvertTo-SecureString -String $password -AsPlainText –force

$newCred = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList $username,$encPwd  
**#Connecting the admin url**

Connect-SPOService -Url $tenantAdminUrl -Credential $newCred

**#Get the master list content and extracted to a json file**

$listUrl = <https://iqubes2008.sharepoint.com/sites/BO_Companies_1759/Lists/MasterList/AllItems.aspx>

$extracted = Get-SPOSiteScriptFromList -ListUrl $listUrl

**#set the above content to a local path**

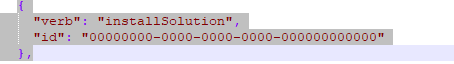
Set-Content -Path C:\Users\Geethu\Desktop\SiteScript\iQubeS\SiteDesignJson\Companies\CompaniesListInstance.json -Value $extracted  
**The below picture shows the site script of list**



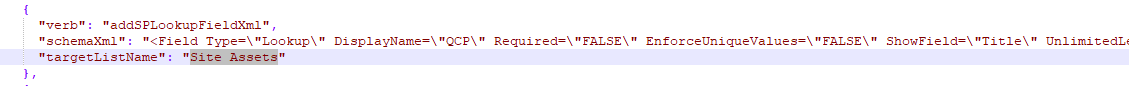
**Remove the spaces between the List Names**

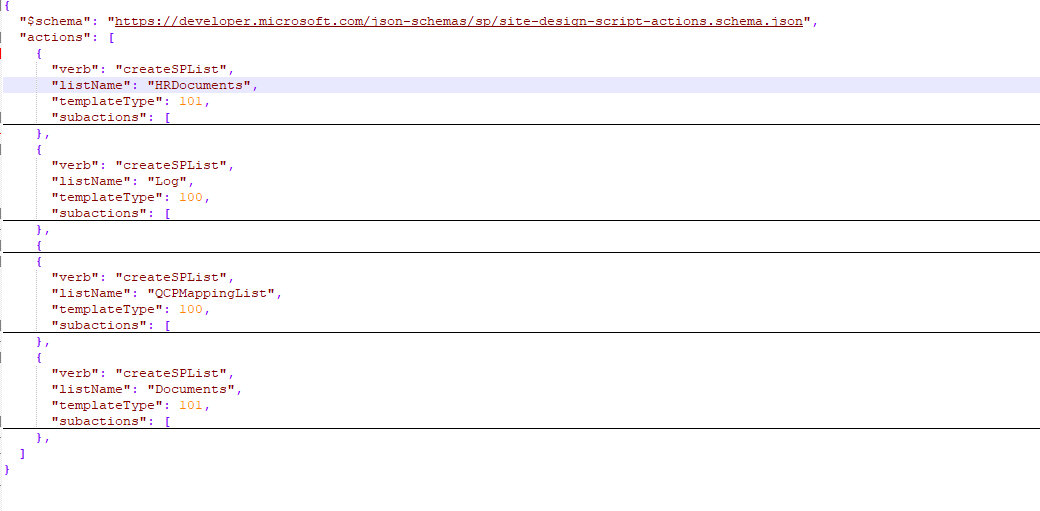
**Removed the master list content details from json**

**Removed the Form Templates Library Details**

**Also Removed the Install Solution Verb**

* **Search all the list name in the json file and removed all the space between the list names**



* **This is the sample we getting after removing spaces between the list names** **The below code fetch the WS regional settings,theme,branding etc.. and added this content to json files.**Get-SPOSiteScriptFromWeb –WebUrl "https://iqubes2008.sharepoint.com/sites/BO\_Companies\_1759/" -IncludeTheme -IncludeBranding -IncludeSiteExternalSharingCapability –IncludeRegionalSettings -IncludeLinksToExportedItems
* {
* "verb": "setSiteBranding",
* "navigationLayout": "Cascade",
* "headerLayout": "Standard",
* "headerBackground": "Strong",
* "showFooter": false
* },
* {
* "verb": "applyTheme",
* "themeJson": {
* "version": "2",
* "isInverted": false,
* "palette": {
* "themePrimary": "#ff69797e",
* "themeLighterAlt": "#fff8f9fa",
* "themeLighter": "#ffe4e9ea",
* "themeLight": "#ffcdd5d8",
* "themeTertiary": "#ff9fadb1",
* "themeSecondary": "#ff78888d",
* "themeDarkAlt": "#ff5d6c70",
* "themeDark": "#ff4f5b5f",
* "themeDarker": "#ff3a4346",
* "accent": "#ff0078d4",
* "neutralDark": "#ff201f1e",
* "neutralPrimary": "#ff323130",
* "neutralPrimaryAlt": "#ff3b3a39",
* "neutralSecondary": "#ff605e5c",
* "neutralSecondaryAlt": "#ff8a8886",
* "neutralTertiary": "#ffa19f9d",
* "neutralTertiaryAlt": "#ffc8c6c4",
* "neutralQuaternary": "#ffd2d0ce",
* "neutralQuaternaryAlt": "#ffe1dfdd",
* "neutralLight": "#ffedebe9",
* "neutralLighter": "#fff3f2f1",
* "neutralLighterAlt": "#fffaf9f8",
* "black": "#ff000000",
* "white": "#ffffffff",
* "primaryBackground": "#ffffffff",
* "primaryText": "#ff333333",
* "HyperlinkActive": "#ff3a4346",
* "CommandLinksPressed": "#ff3a4346",
* "NavigationPressed": "#ff3a4346",
* "EmphasisHoverBorder": "#ff3a4346",
* "TopBarPressedText": "#ff3a4346",
* "HeaderNavigationPressedText": "#ff3a4346",
* "Hyperlinkfollowed": "#ff3a4346",
* "EmphasisHoverBackground": "#ff4f5b5f",
* "EmphasisBorder": "#ff4f5b5f",
* "AccentText": "#ff69797e",
* "CommandLinksHover": "#ff69797e",
* "RowAccent": "#ff69797e",
* "NavigationAccent": "#ff69797e",
* "NavigationHover": "#ff69797e",
* "EmphasisBackground": "#ff69797e",
* "HeaderNavigationHoverText": "#ff69797e",
* "HeaderNavigationSelectedText": "#ff69797e",
* "SuiteBarBackground": "#ff69797e",
* "Hyperlink": "#ff69797e",
* "ContentAccent1": "#ff69797e",
* "AccentLines": "#ff78888d",
* "HeaderAccentLines": "#ff78888d",
* "ButtonPressedBorder": "#ff78888d",
* "SuiteBarHoverBackground": "#ff9fadb1",
* "StrongLines": "#ffcdd5d8",
* "HeaderStrongLines": "#ffcdd5d8",
* "SuiteBarHoverText": "#ffcdd5d8",
* "ButtonPressedBackground": "#ffcdd5d8",
* "ButtonHoverBorder": "#ffcdd5d8",
* "ButtonHoverBackground": "#ffe4e9ea",
* "SelectionBackground": "#7fcdd5d8",
* "HoverBackground": "#7fe4e9ea",
* "NavigationHoverBackground": "#7fe4e9ea",
* "PageBackground": "#ffffffff",
* "EmphasisText": "#ffffffff",
* "SuiteBarText": "#ffffffff",
* "TileText": "#ffffffff",
* "BackgroundOverlay": "#d8ffffff",
* "HeaderBackground": "#d8ffffff",
* "FooterBackground": "#d8ffffff",
* "DisabledBackground": "#fffaf9f8",
* "HeaderDisabledBackground": "#fffaf9f8",
* "ButtonBackground": "#fffaf9f8",
* "ButtonDisabledBackground": "#fffaf9f8",
* "SubtleEmphasisBackground": "#fff3f2f1",
* "DialogBorder": "#fff3f2f1",
* "NavigationSelectedBackground": "#c6edebe9",
* "TopBarBackground": "#c6edebe9",
* "DisabledLines": "#ffedebe9",
* "HeaderDisabledLines": "#ffedebe9",
* "ButtonDisabledBorder": "#ffedebe9",
* "SuiteBarDisabledText": "#ffedebe9",
* "SubtleLines": "#ffc8c6c4",
* "HeaderSubtleLines": "#ffc8c6c4",
* "ButtonGlyphDisabled": "#ffc8c6c4",
* "DisabledText": "#ffa19f9d",
* "CommandLinksDisabled": "#ffa19f9d",
* "HeaderDisableText": "#ffa19f9d",
* "ButtonDisabledText": "#ffa19f9d",
* "Lines": "#ffa19f9d",
* "HeaderLines": "#ffa19f9d",
* "ButtonBorder": "#ffa19f9d",
* "SubtleBodyText": "#ff8a8886",
* "HeaderSubtleText": "#ff8a8886",
* "CommandLinks": "#ff605e5c",
* "Navigation": "#ff605e5c",
* "SubtleEmphasisText": "#ff605e5c",
* "TopBarText": "#ff605e5c",
* "HeaderNavigationText": "#ff605e5c",
* "ButtonGlyph": "#ff605e5c",
* "BodyText": "#ff323130",
* "WebPartHeading": "#ff323130",
* "HeaderText": "#ff323130",
* "ButtonText": "#ff323130",
* "ButtonGlyphActive": "#ff323130",
* "TopBarHoverText": "#ff323130",
* "StrongBodyText": "#ff201f1e",
* "SiteTitle": "#ff201f1e",
* "CommandLinksSecondary": "#ff201f1e",
* "SubtleEmphasisCommandLinks": "#ff201f1e",
* "HeaderSiteTitle": "#ff201f1e",
* "TileBackgroundOverlay": "#7f000000",
* "ContentAccent2": "#ff00485b",
* "ContentAccent3": "#ff288054",
* "ContentAccent4": "#ff767956",
* "ContentAccent5": "#ffed0033",
* "ContentAccent6": "#ff682a7a"
* }
* }
* },
* {
* "verb": "setSiteExternalSharingCapability",
* "capability": "ExternalUserSharingOnly"
* },
* {
* "verb": "setRegionalSettings",
* "timeZone": 4,
* "locale": 1044,
* "sortOrder": 9,
* "hourFormat": "24"
* }

* **Add new verb to the json file for connecting site to hub. Below code is used to get the hub site id**  
  #Connecting the Hubsite and get the hubsiteID  
  $hubSite = Get-SPOHubsite <https://iqubes2008.sharepoint.com/sites/CRM>  
  $hubSite.Id

Guid

----

2a10e36f-960a-4877-b47d-58b043193af0

* **Below verb is added to the json file for connecting the hub site and hubsiteId is replaced with new one.**  
  {

"verb": "joinHubSite",

"hubSiteId": "f66a8a89-3a0e-435f-93d2-f5530399a53d"

},

* **Verb is used to remove items from quick launch.Below code is added for removing items from QL.**

{

"verb": "removeNavLink",

"url": "/SitePages/Forms/ByAuthor.aspx",

"displayName": "Pages",

"isWebRelative": true

},

* **Verb is used to add.Below code is added for Navigating items from QL.**

{

"verb": "addNavLink",

"url": "/SitePages/Leads.aspx",

"displayName": "Leads",

"isWebRelative": true

},

* **Below verb used for Site Column Creation. ID and Source ID varies in different tenant site**

{

"verb": "createSiteColumnXml",

"schemaXml": "<Field Type=\"Text\" DisplayName=\"DocumentCategoryText\" Required=\"FALSE\" EnforceUniqueValues=\"FALSE\" Indexed=\"FALSE\" MaxLength=\"255\" Group=\"Custom Columns\" ID=\"{42ffe208-51d2-40b8-b106-d22f0aee0818}\" SourceID=\"{fc94c3b8-71b0-4215-aea3-0dc03816832a}\" StaticName=\"DocumentCategoryText\" Name=\"DocumentCategoryText\"></Field>"

},

* **Source ID added for Site Columns in All Library’s (AuthorText,CreatedText,ProjectRevisionText,DocumentIdText).**

{

"verb": "addSPFieldXml",

"schemaXml": "<Field Type=\"Text\" DisplayName=\"CreatedText\" Required=\"FALSE\" EnforceUniqueValues=\"FALSE\" Indexed=\"FALSE\" MaxLength=\"255\" Group=\"Custom Columns\" ID=\"{36ee6ab2-7efb-4753-9db3-e9a2f5fd486c}\" SourceID=\"{5dd42ed1-3617-4210-ba77-fc775114bedc}\" StaticName=\"CreatedText\" Name=\"CreatedText\" AllowDeletion=\"TRUE\" />"

},

* **After updating json files then read the content of json files from local path and site scrip is generated. Below code is used for reading the json content and Site script generation**

$extracted = Get-Content -Path C:\Users\Geethu\Desktop\SiteScript\iQubeS\SiteDesignJson\Companies\CompaniesInstance.json | Out-String

Add-SPOSiteScript -Title "CompaniesInstance" -Content $extracted -Description "Creates the Companies WS"

Id : ee035d71-306d-4fb7-a480-c114fd9cb59a

Title : CompaniesInstance

Description : Creates the Comanies WS

Content :

Version : 0

IsSiteScriptPackage : False

* **Below script is used for generating site design id.Here site script is replaced with the new one**

Add-SPOSiteDesign -Title "CRM" -WebTemplate "64" -SiteScripts 18274cba-f393-4ad6-ad0e-df7c645cb607 -Description "Template for CRM Hubsite"

Id : 0b6bb07f-0635-476e-b893-d0c3d6750c2f

Title : CRM

WebTemplate : 64

SiteScriptIds : {ee035d71-306d-4fb7-a480-c114fd9cb59a}

Description : Template for CRM Hubsite

PreviewImageUrl :

PreviewImageAltText :

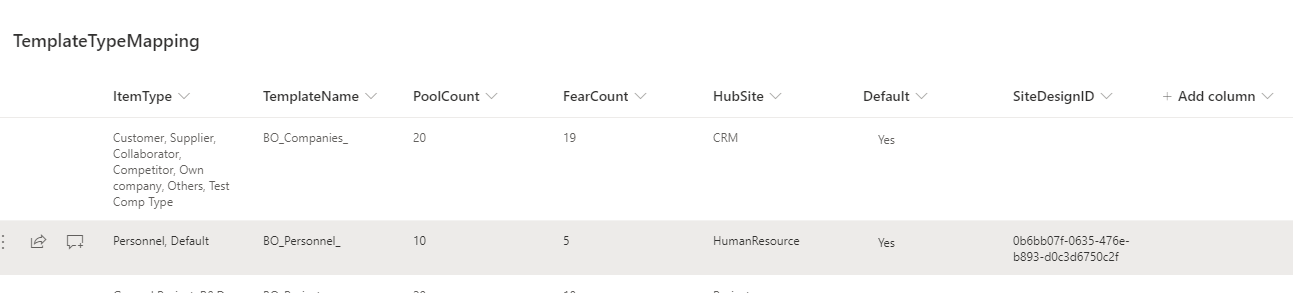
IsDefault : False

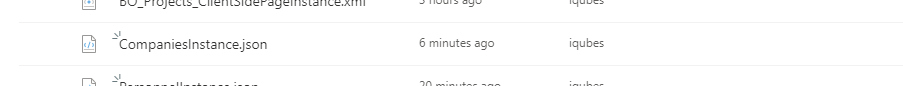
Version : 1

DesignPackageId : 00000000-0000-0000-0000-000000000000

DesignType : Site

* **Generated SiteDesignID value is updated in the template type mapping list.**



* **Json Files Added to the template library**
* **Delete the existing pools and run the site pooling scheduler**
* **Console Execution**

**For Updating Existing Sites  
Site Assets**

$username = 'system.account.stg@bostg01.onmicrosoft.com'

$password = 'wnkfsvgvgkqwpwnr'

$SiteUrl = 'https://bostg01.sharepoint.com/Sites/crm'

$ListName = "SiteAssets"

$NewListURL = "SiteAssets"

$encPwd = ConvertTo-SecureString -String $password -AsPlainText -force

$newCred = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList $username,$encPwd

#connecting to hub site

$connection = Connect-PnPOnline -Url $SiteUrl -Credentials $newCred -ReturnConnection

#fetching all items from list

$items = Get-PnPListItem -List 'Companies'

#looping on item & fetching Sub sitecollection link

foreach($item in $items)

{

#fetching subsite collection link

$url=$item["Link"].Url;

$title = $item["Title"];

$ID = $item["ID"];

#connecting to subsite collection

$username = 'system.account.stg@bostg01.onmicrosoft.com'

$password = 'wnkfsvgvgkqwpwnr'

#checking condition whether site link is blank or not

if($url -ne $Null)

{

try{

$subsite = Connect-PnPOnline -Url $url -Credentials $newCred -ReturnConnection

#Get the List

$List= Get-PnPList -Identity $ListName -Includes RootFolder -Connection $subsite

#sharepoint online powershell change list url

$List.Rootfolder.MoveTo($NewListURL)

Invoke-PnPQuery

#this will change library title

Set-PnPList -Identity $ListName -Title $NewListURL

Write-host "List Name Updated for" $ID

}

catch{

#if site link not connection or broken or webpart not present

Write-host "Site not connected for" $ID

}

}

else

{

#incase site link blank

Write-host "Site url not present for" $ID

}

}

**QCPMappingList**  
  
$username = 'system.account.stg@bostg01.onmicrosoft.com'

$password = 'wnkfsvgvgkqwpwnr'

$SiteUrl = 'https://bostg01.sharepoint.com/Sites/crm'

$ListName = "QCP Mapping List"

$NewListURL = "Lists/QCPMappingList"

$encPwd = ConvertTo-SecureString -String $password -AsPlainText -force

$newCred = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList $username,$encPwd

#connecting to hub site

$connection = Connect-PnPOnline -Url $SiteUrl -Credentials $newCred -ReturnConnection

#fetching all items from list

$items = Get-PnPListItem -List 'Companies'

#looping on item & fetching Sub sitecollection link

foreach($item in $items)

{

#fetching subsite collection link

$url=$item["Link"].Url;

$title = $item["Title"];

$ID = $item["ID"];

#connecting to subsite collection

$username = 'system.account.stg@bostg01.onmicrosoft.com'

$password = 'wnkfsvgvgkqwpwnr'

#checking condition whether site link is blank or not

if($url -ne $Null)

{

try{

$subsite = Connect-PnPOnline -Url $url -Credentials $newCred -ReturnConnection

#Get the List

$List= Get-PnPList -Identity $ListName -Includes RootFolder -Connection $subsite

#sharepoint online powershell change list url

$List.Rootfolder.MoveTo($NewListURL)

Invoke-PnPQuery

#this will change library title

Set-PnPList -Identity $ListName -Title "QCPMappingList"

Write-host "List Name Updated for" $ID

}

catch{

#if site link not connection or broken or webpart not present

Write-host "Site not connected for" $ID

}

}

else

{

#incase site link blank

Write-host "Site url not present for" $ID

}

}

Remove-SPOSiteDesign 344a2e8d-688a-474a-b5aa-613528b87081