Create SharePoint fields, views & pages from XML schema file programmatically

```
<?xml version="1.0" encoding="utf-8"?>
<Elements xmlns="http://schemas.microsoft.com/sharepoint/">
    <Module Name="Home" Url="$Resources:osrvcore,List_Pages_UrlName;">
    <File Url="Podprojekt7.aspx" Type="GhostableInLibrary" Level="Published" Path="xyz\Podprojekt7.aspx">
      <Property Name="Title" Value="Podprojekt 7" />
      <Property Name="PublishingPageLayout" Value="~SiteCollection/_catalogs/masterpage/xyz.aspx, xyz Template" />
      <Property Name="ContentType" Value="xyz" />
      <View List="$Resources:core,lists_Folder;/OsnovnaInformacija" BaseViewID="9" WebPartZoneID="Main2Top" WebPartOrder="0">
        <![CDATA[
                    <webParts>
                        <webPart xmlns="http://schemas.microsoft.com/WebPart/v3">
                            <metaData>
                                <type name="Microsoft.SharePoint.WebPartPages.XsltListViewWebPart,Microsoft.SharePoint,Version=15.0.0.0,Culture=neutral,PublicKeyToken=71e9bce111e9429c
                                <importErrorMessage>$Resources:osrvcore,WebPartImportError;</importErrorMessage>
                            </metaData>
                            <data>
                                <properties>
                                    <property name="Title" type="string">Osnovna informacija</property>
                                    <property name="DisableViewSelectorMenu" type="bool">True</property>
                                    <property name="InplaceSearchEnabled" type="bool">False</property>
                                    <property name="ShowToolbarWithRibbon" type="bool">False</property>
                                    <property name="ChromeType" type="chrometype">Default</property>
                                </properties>
                            </data>
                        </webPart>
                    </webParts>
            ]]>
      </View>
      <View List="$Resources:core,lists_Folder;/Porocila" BaseViewID="57" WebPartZoneID="Main2Top" WebPartOrder="1">
        <![CDATA[
                    <webParts>
                        <webPart xmlns="http://schemas.microsoft.com/WebPart/v3">
                            <metaData>
                                <type name="Microsoft.SharePoint.WebPartPages.XsltListViewWebPart,Microsoft.SharePoint,Version=15.0.0.0,Culture=neutral,PublicKeyToken=71e9bce111e9429c
                                <importErrorMessage>$Resources:osrvcore,WebPartImportError;</importErrorMessage>
                            </metaData>
                            <data>
                                <properties>
                                    <property name="Title" type="string">Tehnologija</property>
                                    <property name="DisableViewSelectorMenu" type="bool">True</property>
                                    <property name="InplaceSearchEnabled" type="bool">False</property>
                                    <property name="ShowToolbarWithRibbon" type="bool">False</property>
                                    <property name="ChromeType" type="chrometype">Default</property>
                                </properties>
                            </data>
                        </webPart>
                    </webParts>
```

Let's imagine that we have **existing WSP solution** for one of our projects. We want to update one of **web template** with additional **pages**. On this pages we have list view web parts with specific newly added **list views**. We have to update our list with this additional list views and we have to add some additional **fields** too.

In WSP solution we already have **module** file for publishing custom pages to SharePoint **Pages** library, **fields** definition file for publishing list fields and **list schema** for our list and views definition. All this **features** are automatically deployed to newly created sites within custom **web template**.

This WSP solution is already deployed to SharePoint farm and we have sites already created with this web template. We know that in our case we have no problem with new pages created from updated web template. The problem arises when we update our SharePoint farm with solution where we have updated pages module, list fields and list schema and we want to apply this changes to **existing sites** which are created from our custom web template before that.

Because we have module file with list view web part definitions, list schema with field and view definitions we want to use this files for our existing sites too.
So I decided to create **PowerShell Script** which read all this definition files and create belonging SharePoint element from it.

In script below we define Site URL in we go through all subsites which were created from our custom Web Template. Firstly, we create fields (**CreateField** PS function), then we create views (**CreateViews** PS function) and at last we create additional three pages (**ReadPodprojektTemplate** PS function).

```
$siteUrl = "{site-url}"

$site = Get-SPSite $siteUrl
foreach ($web in $site.AllWebs)
{
    if (!$web.IsRootWeb)
    {
        try
        {
            $webUrl = $web.Url

            Write-Host $webUrl

            $pages = $web.Lists["Strani"]
            if (-not($pages)) {
                $pages = $web.Lists["Pages"]
            }

            $items = $pages.Items

            CreateField $web $webUrl

            $IDs = CreateViews $web $webUrl "C:\Solutions\2017-12-19\Views\TPPorocila_Views7-
9.xml" "Lists/Porocila"
            $IDsOI2 = CreateViews $web $webUrl "C:\Solutions\2017-12-
19\Views\TPOsnovnaInformacija_Views7-9.xml" "Lists/OsnovnaInformacija"

            if ($items.Title -notcontains "Podprojekt 7") {
                ReadPodprojektTemplate $siteUrl $web $webUrl "Podprojekt7" "Podprojekt 7" $IDs
$IDsOI2
            }

            if ($items.Title -notcontains "Podprojekt 8") {
                ReadPodprojektTemplate $siteUrl $web $webUrl "Podprojekt8" "Podprojekt 8" $IDs
$IDsOI2
            }

            if ($items.Title -notcontains "Podprojekt 9") {
                ReadPodprojektTemplate $siteUrl $web $webUrl "Podprojekt9" "Podprojekt 9" $IDs
$IDsOI2
            }
        }
        catch
        {
            Write-Host $_.Exception.Message
        }
    }
}
```

1. Create fields with PowerShell script from Fields XML file

We have XML file with multiple Field definition like this below:

```
<Fields>
    <Field Name="Jakost" ID="{cdfe0a2b-eca0-4ce5-a507-0986e00727e2}" DisplayName="Jakost"
Type="MultiChoice" Required="TRUE">
        <Default>/</Default>
      <CHOICES><CHOICE>/</CHOICE></CHOICES></Field>
    ...
</Fields>
```

And here is **CreateField** function which read this XML file and create fields in SharePoint List named **Porocila**.

```
function CreateField($web, $webUrl)
{
    [xml]$fieldXml = Get-Content "C:\Solutions\2017-12-19\Fields\TPPorocila_Jakost.xml" -encoding
UTF8

    $lPorocila = $web.GetList($webUrl + "/Lists/Porocila")

    foreach ($field in $fieldXml.Fields.Field)
    {
        $spField = $lPorocila.Fields.TryGetFieldByStaticName($field.Name)
        if ($spField -eq $null)
        {
            $lPorocila.Fields.AddFieldAsXml($field.OuterXml)
        }
    }
}
```

2. Create views with PowerShell script from Views XML file

We have XML file with multiple View definition like this below:

```
<Views>
    <View BaseViewID="57" Name="636b3f07-f60b-4616-b54b-08544053473e" DisplayName="Tehnologija_P7"
Type="HTML" WebPartZoneID="Main" SetupPath="pages\viewpage.aspx" Url="Tehnologija_P7.aspx">
<RowLimit>10</RowLimit>
        <ViewFields>
          <Field Name="Jakost" />
          <FieldRef Name="DocIcon" /><FieldRef Name="LinkFilename" />...</ViewFields>
        <Query>
          <Where>
            ...
          </Where>
          <OrderBy><FieldRef Name="Modified" Ascending="FALSE"></FieldRef></OrderBy>
        </Query><Toolbar Type="Standard" /><XslLink Default="TRUE">main.xsl</XslLink>
<JSLink>clienttemplates.js</JSLink>
    </View>
    ...
</Views>
```

And here is **CreateViews** function which read this XML file and create views in specific SharePoint List.

```
function CreateViews($web, $webUrl, $filePath, $listPath)
{
    $tempIDs = @{}

    $lPorocila = $web.GetList($webUrl + "/" + $listPath)

    [xml]$viewXml = Get-Content $filePath -encoding UTF8

    foreach ($view in $viewXml.Views.View)
    {
        $spView = $lPorocila.Views[$view.DisplayName]
        if(($spView -eq $null) -and (($view.DisplayName -notlike "All Documents") -or
($view.DisplayName -notlike "All items")))
        {
            $viewFields = New-Object System.Collections.Specialized.StringCollection
            foreach ($column in $view.ViewFields.Field)
            {
                $viewFields.Add($column.Name) > $null
            }
            foreach ($column in $view.ViewFields.FieldRef)
            {
                $viewFields.Add($column.Name) > $null
            }

            $query = $view.Query.InnerXml

            $rowLimit = $view.RowLimit

            $newview = $lPorocila.Views.Add($view.DisplayName, $viewFields, $query, $rowLimit,
$true, $false)

            $tempIDs.Add($view.BaseViewID, $newview.ID)

            $lPorocila.Update()
        }
        else
        {
            $tempIDs.Add($view.BaseViewID, $spView.ID)
        }
    }

    return $tempIDs
}
```

3. Create pages with PowerShell script from Module XML file

We have XML file with multiple File (representing Pages) and Views definition like this below:

```
<Module Name="Home" Url="$Resources:osrvcore,List_Pages_UrlName;">
    <File Url="Podprojekt7.aspx" Type="GhostableInLibrary" Level="Published"
Path="xyz\Podprojekt7.aspx">
      <Property Name="Title" Value="Podprojekt 7" />
      <Property Name="PublishingPageLayout" Value="~SiteCollection/_catalogs/masterpage/xyz.aspx,
xyz Template" />
      <Property Name="ContentType" Value="xyz" />
      <View List="$Resources:core,lists_Folder;/Porocila" BaseViewID="72" WebPartZoneID="Main2Top"
WebPartOrder="8">
        <![CDATA[
                    <webParts>
                        <webPart xmlns="http://schemas.microsoft.com/WebPart/v3">
                            <metaData>
                                <type
name="Microsoft.SharePoint.WebPartPages.XsltListViewWebPart,Microsoft.SharePoint,Version=15.0.0.0,
Culture=neutral,PublicKeyToken=71e9bce111e9429c" />
                                <importErrorMessage>$Resources:osrvcore,WebPartImportError;
</importErrorMessage>
                            </metaData>
                            <data>
                                <properties>
                                    <property name="Title" type="string">Surovina</property>
                                    <property name="DisableViewSelectorMenu"
type="bool">True</property>
                                    <property name="InplaceSearchEnabled"
type="bool">False</property>
                                    <property name="ShowToolbarWithRibbon"
type="bool">False</property>
                                    <property name="ChromeType"
type="chrometype">Default</property>
                                </properties>
                            </data>
                        </webPart>
                    </webParts>
                ]]>
      </View>
      ...
    </File>
    ...
</Module>
```

And here is **ReadPodprojektTemplate** function which read this XML file and create pages from specific **page layout** with multiple list view web parts (type of **XsltListViewWebPart)** from **cdata-section** of input XML file.

```
function ReadPodprojektTemplate($siteUrl, $web, $webUrl, $name, $title, $IDs, $IDsOI2)
{
    [xml]$pageXml = Get-Content "C:\Solutions\2017-12-19\Pages\TPPodprojekt7-9.xml" -encoding UTF8

    $fileXml = $pageXml.Elements.Module.File | ?{$_.Url -eq "$name.aspx"}
    if (-not($fileXml)) {
        throw "Page definition missing"
    }

    $plDefinition = $fileXml.Property | Where { $_.Name -eq "PublishingPageLayout" }
    if (-not($plDefinition)) {
        throw "Page layout missing"
    }

    $plUrl = New-Object Microsoft.SharePoint.SPFieldUrlValue($plDefinition.Value)
    $plName = $plUrl.Url.Substring($plUrl.Url.LastIndexOf('/') + 1)

    $psite = New-Object Microsoft.SharePoint.Publishing.PublishingSite($siteUrl)
    $pl = $psite.GetPageLayouts($false) | Where { $_.Name -eq $plName }
    if (-not($pl)) {
        throw "Page layout not found"
    }

    $pweb = [Microsoft.SharePoint.Publishing.PublishingWeb]::GetPublishingWeb($web)
    $page = $pweb.AddPublishingPage("$name.aspx", $pl)
    $page.Title = $title
    $page.Update()

    $views = $fileXml.View
    foreach ($view in $views)
    {
        [xml]$cdata = $view | select -expand "#cdata-section"
        $wpProperties = $cdata.webParts.webPart.data.properties

        $listUrl = [Microsoft.SharePoint.Utilities.SPUtility]::GetLocalizedString($view.List,
$null, 1033)
        $list = $web.getList($webUrl + "/" + $listUrl)

        if ($view.BaseViewID -eq 1)
        {
            $listView = $list.DefaultView
        }
        else
        {
            if ($listUrl -like "*OsnovnaInformacija")
            {
                $tempGUID = $IDsOI2[$view.BaseViewID]
            }
            else
            {
                $tempGUID = $IDs[$view.BaseViewID]
            }

            if (-not($tempGUID)) {
                throw "List view GUID not found"
            }

            $listView = $list.GetView($tempGUID)
            #$listView = $list.Views | Where { $_.BaseViewID -eq $view.BaseViewID }
        }

        if (-not($listView)) {
            throw "List view not found"
```

```
        }

        $wpmgr = $web.GetLimitedWebPartManager($page.Url,
[System.Web.UI.WebControls.WebParts.PersonalizationScope]::Shared)
        $listviewwebpart = New-Object Microsoft.SharePoint.WebPartPages.XsltListViewWebPart
        $listviewwebpart.Title = ($wpProperties.property | Where { $_.Name -eq "Title" }).'#text'
        $listviewwebpart.ChromeType = ($wpProperties.property | Where { $_.Name -eq "ChromeType"
}).'#text'
        $listviewwebpart.ListName = $list.ID.ToString("B")
        $listviewwebpart.ViewGuid = $listView.ID.ToString("B")
        $listviewwebpart.ExportMode = "All"
        $wpmgr.AddWebPart($listviewwebpart, $view.WebPartZoneID, $view.WebPartOrder)
    }

    $page.CheckIn("")
    $page.ListItem.File.Publish("");
}
```

[Complete code on GitHub]

Cheers!
Gašper Rupnik

{End.}

27/12/2017
SharePoint

Published by rasper87

View all posts by rasper87