

لطفا توجه فرمایید که جالب‌ترین قسمت این مقاله قابلیت استفاده از کلاس‌های دات نت در دل PowerShell می‌باشد. که در قسمت چهارم کدها مشاهده می‌فرمایید.

حذف تمام رکوردهای یک لیست شیرپوینت از طریق رابط کاربری SharePoint مسیر نمی‌باشد و لازم است برای آن چند خط کد نوشته شود که می‌توانید آن را با console و جالب‌تر از آن با PowerShell اجرا کنید. 1- ساده‌ترین روش حذف رکوردهای شیرپوینت را در روبرو مشاهده می‌فرمایید که به ازای حذف هر رکورد یک رفت و برگشت به پایگاه انجام می‌شود

```
SPList list = mWeb.GetList(strUrl);
if (list != null)
{
    for (int i = list.ItemCount - 1; i >= 0; i--)
    {
        list.Items[i].Delete();
    }
    list.Update();
}
```

2- با استفاده از [SPWeb.ProcessBatchData](#) در کد زیر می‌توانیم با سرعت بیشتر و هوشمندانه‌تری، حذف تمام رکوردها را در یک عمل انجام دهیم

```
public static void DeleteAllItems(string site, string list)
{
    using (SPSite spSite = new SPSite(site))
    {
        using (SPWeb spWeb = spSite.OpenWeb())
        {
            StringBuilder deletebuilder = BatchCommand(spWeb.Lists[list]);
            spSite.RootWeb.ProcessBatchData(deletebuilder.ToString());
        }
    }
}

private static StringBuilder BatchCommand(SPList spList)
{
    StringBuilder deletebuilder = new StringBuilder();
    deletebuilder.Append("<?xml version='1.0' encoding='UTF-8'><Batch>");
    string command = "<Method><SetList Scope='Request'>" + spList.ID +
        "</SetList><SetVar Name='ID'>{0}</SetVar><SetVar Name='Cmd'>Delete</SetVar></Method>";

    foreach (SPListItem item in spList.Items)
    {
        deletebuilder.Append(string.Format(command, item.ID.ToString()));
    }
    deletebuilder.Append("</Batch>");
    return deletebuilder;
}
```

3- در قسمت زیر همان روش batch قبلی را مشاهده می‌فرمایید که با تقسیم کردن batch ها به تکه‌های 1000 تایی کارایی آن را بالا برده ایم

```
// We prepare a String.Format with a String.Format, this is why we have a {{0}}
string command = String.Format("<Method><SetList Scope='Request'>{0}</SetList><SetVar Name='ID'>{0}</SetVar><SetVar Name='Cmd'>Delete</SetVar><SetVar Name='owsfileid'>{1}</SetVar></Method>", list.ID);
// We get everything but we limit the result to 100 rows
SPQuery q = new SPQuery();
q.RowLimit = 100;

// While there's something left
while (list.ItemCount > 0)
{
    // We get the results
    SPListItemCollection coll = list.GetItems(q);
```

```

StringBuilder sbDelete = new StringBuilder();
sbDelete.Append("<?xml version='1.0' encoding='UTF-8'><Batch>");

Guid[] ids = new Guid[coll.Count];
for (int i=0;i<coll.Count;i++)
{
    SListItem item = coll[i];
    sbDelete.Append(string.Format(command, item.ID.ToString(), item.File.ServerRelativeUrl));
    ids[i] = item.UniqueId;
}
sbDelete.Append("</Batch>");

// We execute it
web.ProcessBatchData(sbDelete.ToString());

//We remove items from recyclebin
web.RecycleBin.Delete(ids);

list.Update();
}
}

```

4- در این قسمت به جالبترین و آموزندهترین قسمت این مطلب میپردازیم و آن import کردن namespaces ها و ساختن object های دات نت در دل PowerShell هست که می‌توانید به راحتی با مقایسه با کد قسمت قبلی که در console نوشته شده است، آنرا فرا بگیرید.

برای فهم script پاور شل زیر کافیت به چند نکته ساده زیر دقت کنید
ایجاد متغیرها به سادگی با شروع نوشتن نام متغیر با \$ و بدون تعریف نوع آنها انجام می‌شود
write-host حکم write را دارد و واضح است که نوشتن تنهای آن برای ایجاد یک line break می‌باشد.

کامنت کردن با #

عدم وجود semi colon برای اتمام فرامین

```

[System.Reflection.Assembly]::Load("Microsoft.SharePoint, Version=12.0.0.0, Culture=neutral,
PublicKeyToken=71e9bce111e9429c")
[System.Reflection.Assembly]::Load("Microsoft.SharePoint.Portal, Version=12.0.0.0, Culture=neutral,
PublicKeyToken=71e9bce111e9429c")
[System.Reflection.Assembly]::Load("Microsoft.SharePoint.Publishing, Version=12.0.0.0, Culture=neutral,
PublicKeyToken=71e9bce111e9429c")
[System.Reflection.Assembly]::Load("System.Web, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a")

write-host

# Enter your configuration here
$siteUrl = "http://mysharepointsite.example.com/"
$listName = "Name of my list"
$batchSize = 1000

write-host "Opening web at $siteUrl..."

$site = new-object Microsoft.SharePoint.SPSite($siteUrl)
$web = $site.OpenWeb()
write-host "Web is: $($web.Title)"

$list = $web.Lists[$listName];
write-host "List is: $($list.Title)"

while ($list.ItemCount -gt 0)
{
    write-host "Item count: $($list.ItemCount)"

    $batch = "<?xml version='1.0' encoding='UTF-8'><Batch>"
    $i = 0

    foreach ($item in $list.Items)
    {
        $i++
        write-host "`rProcessing ID: $($item.ID) ($i of $batchSize)" -nonewline
    }
}

```

```
$batch += "<Method><SetList Scope=`Request`">${$list.ID}</SetList><SetVar  
Name=`ID`">${$item.ID}</SetVar><SetVar Name=`Cmd`">Delete</SetVar><SetVar  
Name=`owsfileref`">${$item.File.ServerRelativeUrl}</SetVar></Method>"  
  
    if ($i -ge $batchSize) { break }  
}  
  
$batch += "</Batch>"  
  
write-host  
  
write-host "Sending batch..."  
  
# We execute it  
$result = $web.ProcessBatchData($batch)  
  
write-host "Emptying Recycle Bin..."  
  
# We remove items from recyclebin  
$web.RecycleBin.DeleteAll()  
  
write-host  
  
$list.Update()  
}  
  
write-host "Done."
```