# Web Cmdlets

**TRAIN**SIGNAL
THE GLOBAL LEADER IN PROFESSIONAL COMPUTER TRAINING

---

## PowerShell and the Cloud

| | | | |
|---|---|---|---|
| V3 designed with the web and cloud in mind | New cmdlets for working with web services and data | No more parsing and "screen-scraping" | May require some developer-based background or assistance |

**TRAIN**SIGNAL

---

## Invoke-WebRequest

| | | |
|---|---|---|
| No more .NET WebClient wrappers | Retrieve web pages via HTTP or HTTPS | Parses the response into an object you can work with in PowerShell |

Save results to a text file right from the cmdlet or -Passthru to a file or the console

**TRAIN**SIGNAL

**Invoke-RestMethod**

| Similar to Invoke-WebRequest | Send request to RESTful web services | Writes HTML documents and JSON converted objects |
|---|---|---|

**Example:**
```
PS C:\> Invoke-RestMethod -Uri
"http://search.twitter.com/search.json?q=PowerShell&lang=en" |
Select-Object -ExpandProperty results |
Select Created_At,From_User,Text | out-gridview
```

---

**ConvertTo-Json**

JSON = Javascript Object Notation
  • A common format style for web data

Although, you can convert anything to this format
  • Could be used for serializing objects
  • No methods
  • Restore with ConvertFrom-Json

---

**ConvertFrom-Json**

Convert a JSON string into a custom object
  • Can be used with Invoke-RestMethod to transform results into PowerShell objects

**New-WebServiceProxy**

| Create an object to manage your connection to a web service | Instead of invoking RESTMethods, use the object's methods | Data returned as an XML object |
|---|---|---|

TRAINSIGNAL

---

**New-WebServiceProxy**

Very, very easy way to use a web service

- PS C:\> $URI = "http://www.webservicex.net/uszip.asmx?WSDL"

- PS C:\> $zip = New-WebServiceProxy -uri $URI

- PS C:\> $results = $zip.GetInfoByZIP("60195")

- PS C:\> $results.table

TRAINSIGNAL

---

**Web Cmdlets Demo**

**Summary**

    – Web cmdlets will be new to many IT
       Pros

    – Explore what internal web services
       you can leverage

    – Expect a steeper learning curve

    – Read full help and examples for these
       cmdlets

**TRAIN**SIGNAL