Using PowerShell to Call Web Services

Eric Kool-Brown, Software Engineer (kool@uw.edu)
UW-IT Infrastructure

INFORMATION TECHNOLOGY

UNIVERSITY *of* WASHINGTON

Agenda

> **What is a Web Service**

> **REST Details**

> **Example Web Services**

> **PowerShell Web Service Commandlets**

> **Web Service AuthN/AuthZ**

> **Reading the Data – JSON or XML**

> **Demos**

What is a Web Service?

DCE, RPC, DCOM, CORBA, CRUD, oh my!

> **In the beginning, non-standard and proprietary formats/protocols**
>   – **open a socket and send binary packets**
> **Remote Procedure Calls – making function calls over the network.**
>   – **E.g. connecting to SQL Server over a named pipe**
>   – **Not firewall-friendly!**
> **SOAP (simple object access protocol) – transport-independent, can use SMTP, HTTP, UDP, etc.**
>   – **XML as the payload encoding, POST as the HTTP verb**

What is a Web Service Part 2

One HTTP Protocol to Rule Them All!

> **Representational State Transfer (REST)**
>   – **leveraging, rather than abusing, the features of HTTP**
>   – **Roy Fielding dissertation published in 2000**

> **REST architecture**
>   – **Resources uniquely named using a URL**
>   – **CRUD translates directly to the HTTP verbs PUT, GET, POST, and DELETE**
>   – **Stateless – facilitates Internet-scale services**
>   > **HTTP GETs have no side-effects, allows for caching**
>   > **PUTs and DELETEs are idempotent**
>   – **Payload encoding uses HTTP media types (JSON, XML, etc)**

REST Details

> **Multiple API description languages**
> - **OData**
>   - > **$metadata = service data description in XML**
> - **OpenAPI (was: Swagger)**

> **HTTP URL Components**
> - **Each resource has a unique, hierarchical URL path**
>   - > **https://example.org/stuff/thingys/thingy22**
> - **Searching rooted in container paths using URL parameters to specify query filters, paging, etc.**
>   - > **https://example.org/stuff/thingys?$filter=thingnum lt 22**
> - **Some function-like URL paths**
>   - > **https://example.org/foobar(2)/allfoo**

Examples of Web Services

> **UW Groups Web Service (GWS)**
  - **Manage your UW groups programmatically**
  - **Data is sent in XHTML**
  - **Requires UWCA certificate auth**

> **Azure Active Directory Graph API**
  - **Apply CRUD operations to AAD objects**
  - **Traverse links to follow relationships**
  - **Uses OAuth authentication/authorization, must have an AAD application object with appropriate permissions**
  - **Superseded by Microsoft Graph API?**

> **AWS, Facebook, Google, etc.**
  - **UW Enterprise Integration Platform (EIP)**

PowerShell Web Service Commandlets

> **Invoke-WebRequest** – **parses response into HtmlWebResponseObject with the following elements**
>   – **StatusCode – Http response status**
>   – **Content – the response body**
>   – **RawContent – the entire, unparsed response**
>   – **Forms, Headers, Images, Links, ParsedHtml**

> **Invoke-RestMethod** – **Can parse JSON, XML, and HTML forms**
>   – **Returns corresponding object types**
>   – **Gets thoroughly confused by GWS XHTML**

> **Both throw WebException for failure**

Web Service Authentication and Authorization

> **SSL/TLS almost always used for confidentiality**

> **Certificate-based auth – passed as part of SSL setup to validate the client**

> **OAuth – obtain an authorization token, attach it to the web service request**

  – **OAuth tokens on Azure are obtained via an Azure AD Application object**

  > **Confusingly named, it is actually an application identity**

  > **Permissions are assigned to the app object that determine what the token can be used for; e.g. read all Azure AD user objects**

  > **Perms: direct vs. delegated, user vs. admin**

Processing the Returned Data

> **XML – PowerShell's XML parsing is complicated**
  - **XPath: PS cmdlet Select-XML, does not return an XML node object, must extract it from the SelectXmlInfo object**
  - **Can use dotted notation if all of the elements exist**

> **JSON – PS "deserializes" JSON into custom objects**
  - **Returned JSON can be a single long line, machine readable but not easily human readable**
  - **Use ConvertFrom-Json to convert a JSON string into a PSCustomObject**
    > **A "NoteProperty" will be created or each JSON element**
  - **Trick: to format the JSON string with indented multiple lines:**

```
$Json = $groupInfo.Content | ConvertFrom-Json | ConvertTo-Json
```

Demos

> **Example PS scripts:**
> - **New-UwGsGroup.ps1**
> - **Get-UwGsGroup.ps1**
> - **Get-AzureADGraphInfo.ps1**
> - **Get-AADGraphRestMethod.ps1**

Resources

> **Example PS scripts: https://github.com/erickool/ws-powershell**

> **Blog post: https://blogs.uw.edu/kool**

> **Groups Web Service: https://wiki.cac.washington.edu/display/infra/Groups+Web+Service+REST+API**