UNIVERSITY of WISCONSIN
UWMILWAUKEE

Integration of Home-Grown Apps and SharePoint for Digitizing Business Processes

Danny Harvey
Letters and Science Web and Data

Approach

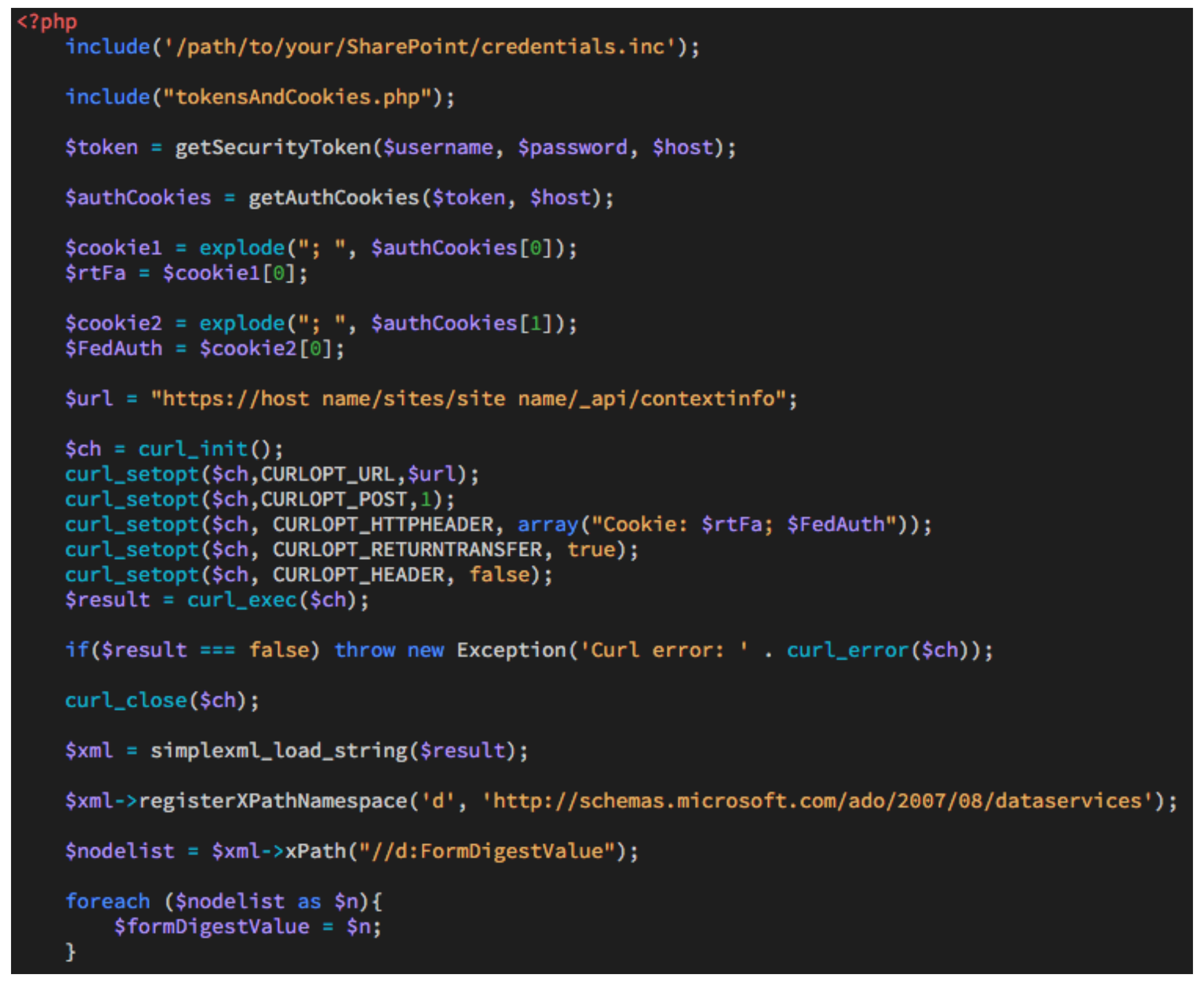
- Create forms, filter and combine data from multiple sources in PHP – MUCH easier than in SharePoint
- Use SharePoint for document and metadata storage
- Use SharePoint API to query Libraries and Lists and get data to PHP apps and WordPress
- Use Microsoft Flow to move files and data from PHP apps to SharePoint
- Can use Flow and file templates to take PHP form data and generate documents within SharePoint – but this is still a work in progress

Use Cases for SharePoint Integration

- WordPress Shortcode that dynamically populates a web page with links to documents in SharePoint
- Ingest Gravity Forms API, store in SharePoint List for additional workflow
- Upload Documents for Sabbatical Requests
- Data Report Mashups with Multiple SharePoint Lists and External Data Sources

SharePoint API Log In

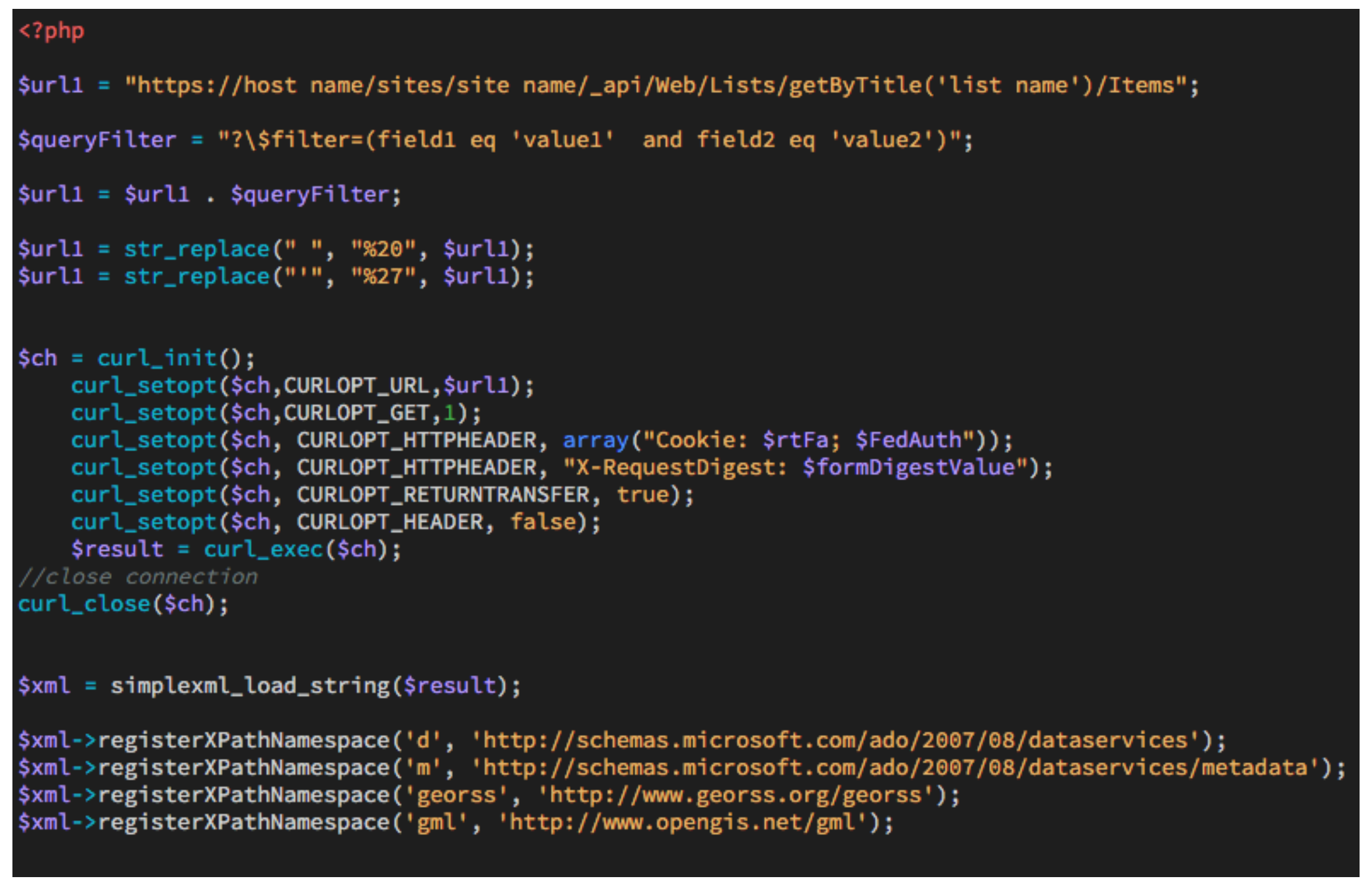
- Need username, password, and hostname for a SharePoint account – store this info in an include file OUTSIDE of the web root
- Get code from Hulette (2012) and place in a file called tokensAndCookies.php. This is used to:
  - Get a Security Token
  - Get Auth Cookies (rtFa, FedAuth)
- Use cURL to get a Form Digest Value



```php
<?php
    include('/path/to/your/SharePoint/credentials.inc');

    include("tokensAndCookies.php");

    $token = getSecurityToken($username, $password, $host);

    $authCookies = getAuthCookies($token, $host);

    $cookie1 = explode("; ", $authCookies[0]);
    $rtFa = $cookie1[0];

    $cookie2 = explode("; ", $authCookies[1]);
    $FedAuth = $cookie2[0];

    $url = "https://host name/sites/site name/_api/contextinfo";

    $ch = curl_init();
    curl_setopt($ch,CURLOPT_URL,$url);
    curl_setopt($ch,CURLOPT_POST,1);
    curl_setopt($ch, CURLOPT_HTTPHEADER, array("Cookie: $rtFa; $FedAuth"));
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_HEADER, false);
    $result = curl_exec($ch);

    if($result === false) throw new Exception('Curl error: ' . curl_error($ch));

    curl_close($ch);

    $xml = simplexml_load_string($result);

    $xml->registerXPathNamespace('d', 'http://schemas.microsoft.com/ado/2007/08/dataservices');

    $nodelist = $xml->xPath("//d:FormDigestValue");

    foreach ($nodelist as $n){
        $formDigestValue = $n;
    }
```

SharePoint API Queries

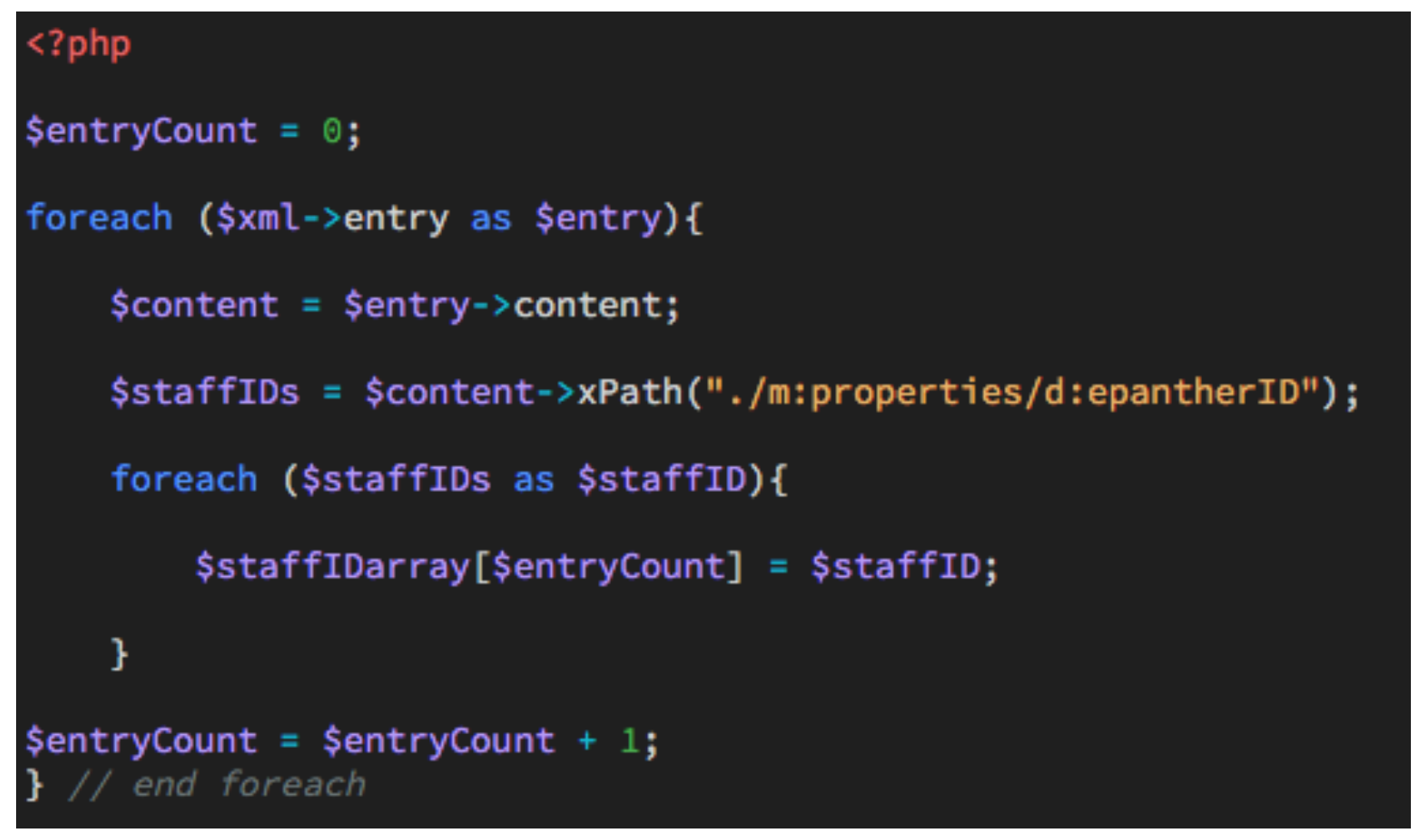
- Although the SharePoint REST API can support CRUD, our instance is Read Only.
- The Auth Cookies and Form Digest Value from the API Log In are needed to perform API queries
- Primary REST endpoint we use is retrieve items within a list (or library). (Microsoft, 2018).
- OData query operators used to filter data (Microsoft, 2017).
- Register XPath Namespaces to facilitate data parsing



```php
<?php

$url1 = "https://host name/sites/site name/_api/Web/Lists/getByTitle('list name')/Items";

$queryFilter = "?\$filter=(field1 eq 'value1'  and field2 eq 'value2')";

$url1 = $url1 . $queryFilter;

$url1 = str_replace(" ", "%20", $url1);
$url1 = str_replace("'", "%27", $url1);


$ch = curl_init();
    curl_setopt($ch,CURLOPT_URL,$url1);
    curl_setopt($ch,CURLOPT_GET,1);
    curl_setopt($ch, CURLOPT_HTTPHEADER, array("Cookie: $rtFa; $FedAuth"));
    curl_setopt($ch, CURLOPT_HTTPHEADER, "X-RequestDigest: $formDigestValue");
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_HEADER, false);
    $result = curl_exec($ch);
//close connection
curl_close($ch);


$xml = simplexml_load_string($result);

$xml->registerXPathNamespace('d', 'http://schemas.microsoft.com/ado/2007/08/dataservices');
$xml->registerXPathNamespace('m', 'http://schemas.microsoft.com/ado/2007/08/dataservices/metadata');
$xml->registerXPathNamespace('georss', 'http://www.georss.org/georss');
$xml->registerXPathNamespace('gml', 'http://www.opengis.net/gml');
```

Parse Results of Query

- Output is XML
- If fieldnames in SharePoint have spaces or other "special characters", a randomly generated field name is used in the API
- It is necessary to comb through the XML manually to determine API fieldnames as well as the XML structure
- Use xPath in PHP to extract the data of interest



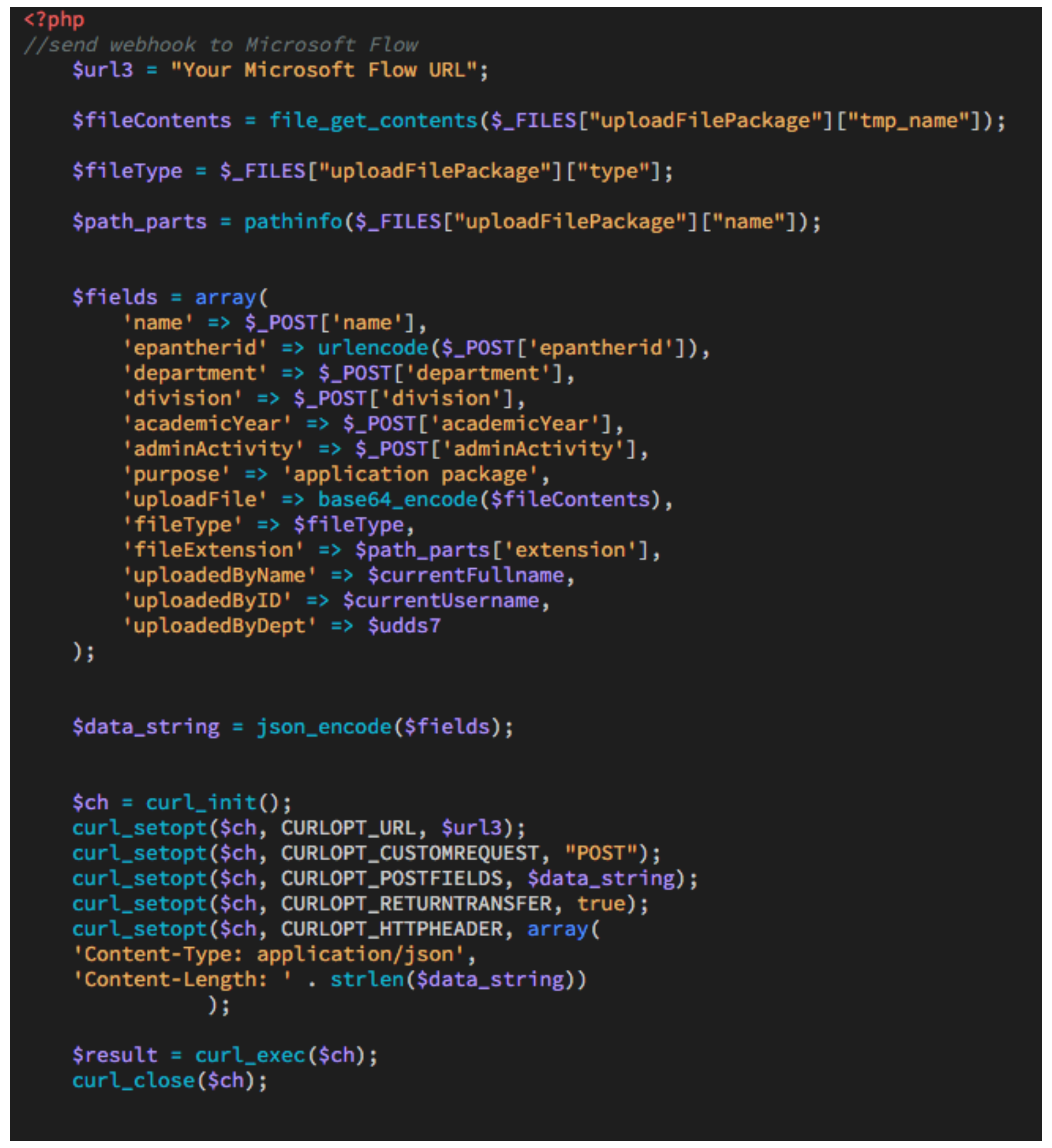
```php
<?php

$entryCount = 0;

foreach ($xml->entry as $entry){

    $content = $entry->content;

    $staffIDs = $content->xPath("./m:properties/d:epantherID");

    foreach ($staffIDs as $staffID){

        $staffIDarray[$entryCount] = $staffID;

    }

$entryCount = $entryCount + 1;
} // end foreach
```

Uploading Files into SharePoint via PHP

Okay. Fine. Boromir does have a point here. It takes a bit of processing in PHP to parse out the file contents, file type, and extension and then construct a cURL session to send to Flow.

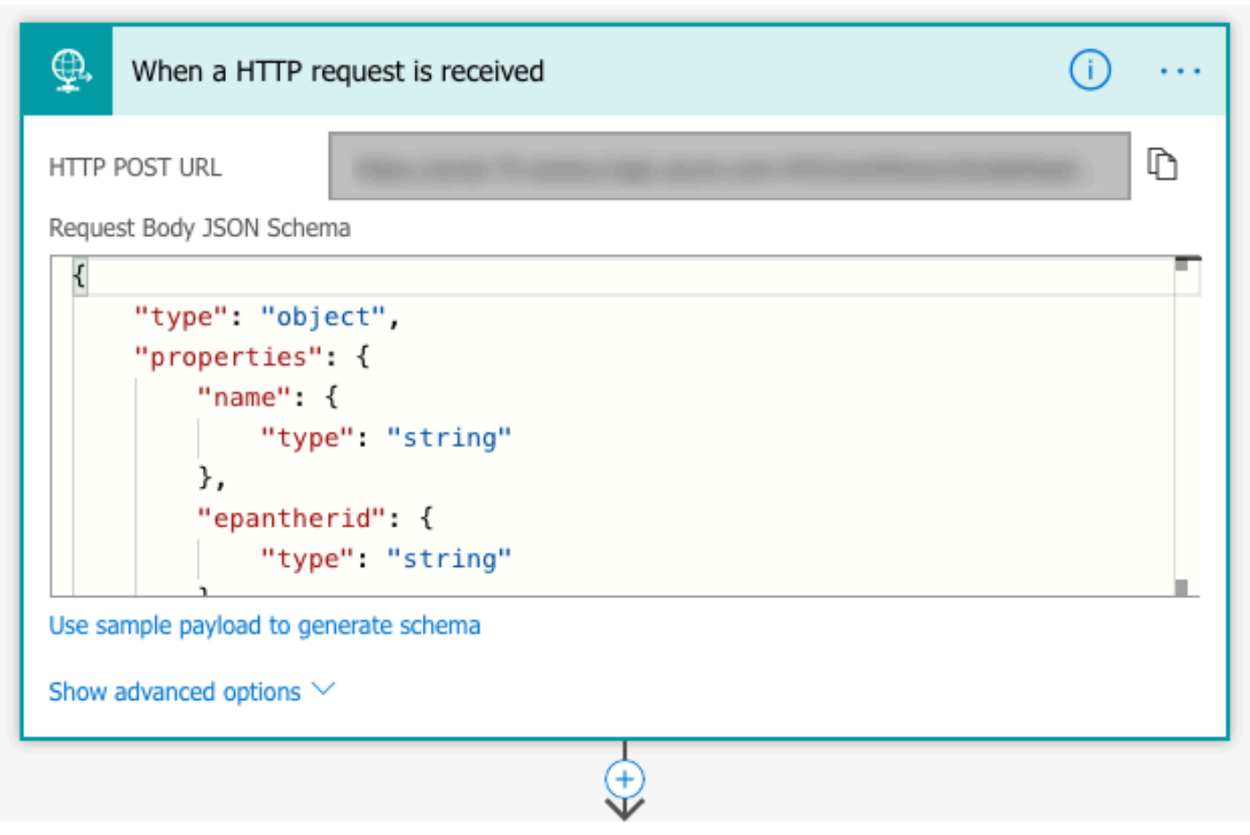
PHP

- Use a File Upload form element
- Use file_get_contents to obtain contents in binary
- base64_encode($fileContents)
  No decoding needed in Flow
- Use $_FILES[] to get File Type and File Extension
- Create a fields array and json_encode it
- Send cURL to Flow



```php
<?php
//send webhook to Microsoft Flow
    $url3 = "Your Microsoft Flow URL";

    $fileContents = file_get_contents($_FILES["uploadFilePackage"]["tmp_name"]);

    $fileType = $_FILES["uploadFilePackage"]["type"];

    $path_parts = pathinfo($_FILES["uploadFilePackage"]["name"]);


    $fields = array(
        'name' => $_POST['name'],
        'epantherid' => urlencode($_POST['epantherid']),
        'department' => $_POST['department'],
        'division' => $_POST['division'],
        'academicYear' => $_POST['academicYear'],
        'adminActivity' => $_POST['adminActivity'],
        'purpose' => 'application package',
        'uploadFile' => base64_encode($fileContents),
        'fileType' => $fileType,
        'fileExtension' => $path_parts['extension'],
        'uploadedByName' => $currentFullname,
        'uploadedByID' => $currentUsername,
        'uploadedByDept' => $udds7
    );


    $data_string = json_encode($fields);


    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url3);
    curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
    curl_setopt($ch, CURLOPT_POSTFIELDS, $data_string);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Content-Type: application/json',
    'Content-Length: ' . strlen($data_string))
               );

    $result = curl_exec($ch);
    curl_close($ch);
```

Process Uploaded File in Flow



When a HTTP request is received

HTTP POST URL [redacted]

Request Body JSON Schema

```
{
    "type": "object",
    "properties": {
        "name": {
            "type": "string"
        },
        "epantherid": {
            "type": "string"
```

Use sample payload to generate schema

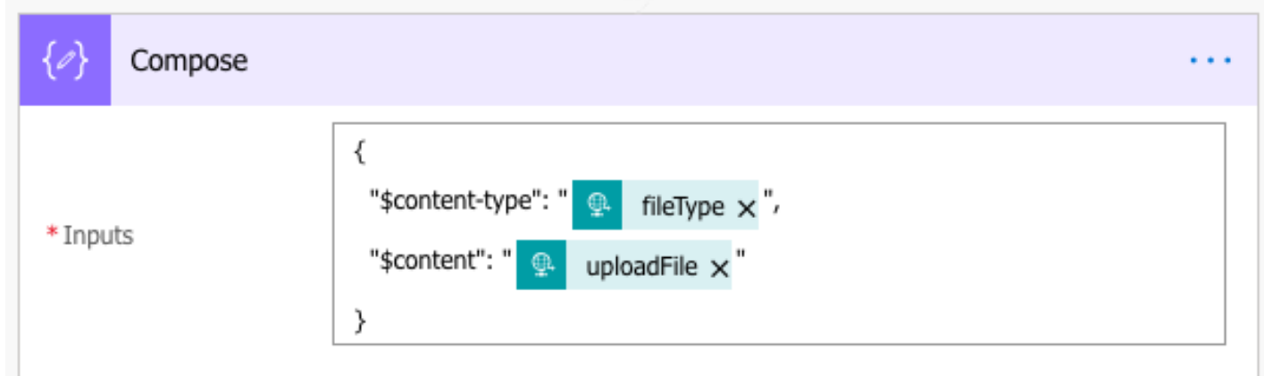
Show advanced options

Step 1: Read in JSON Object sent via cURL request

JSON Schema

```json
{
    "type": "object",
    "properties": {
        "name": {
            "type": "string"
        },
        "epantherid": {
            "type": "string"
        },
        "AY": {
            "type": "string"
        },
        "uploadFile": {
            "type": "string"
        },
        "fileType": {
            "type": "string"
        },
        "fileExtension": {
            "type": "string"
        }
    }
}
```

Your JSON schema will vary according to the metadata you need for your library.

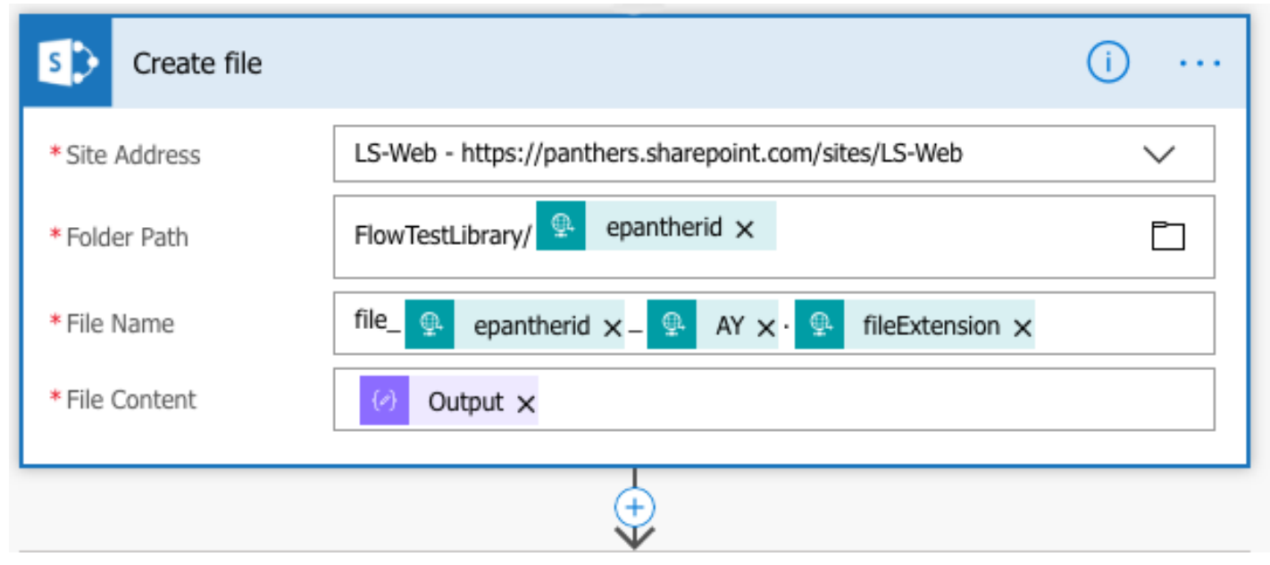


Compose

* Inputs

```
{
  "$content-type": "fileType",
  "$content": "uploadFile"
}
```

Step 2: Compose a Data Object that SharePoint can read and use to create a file



Create file

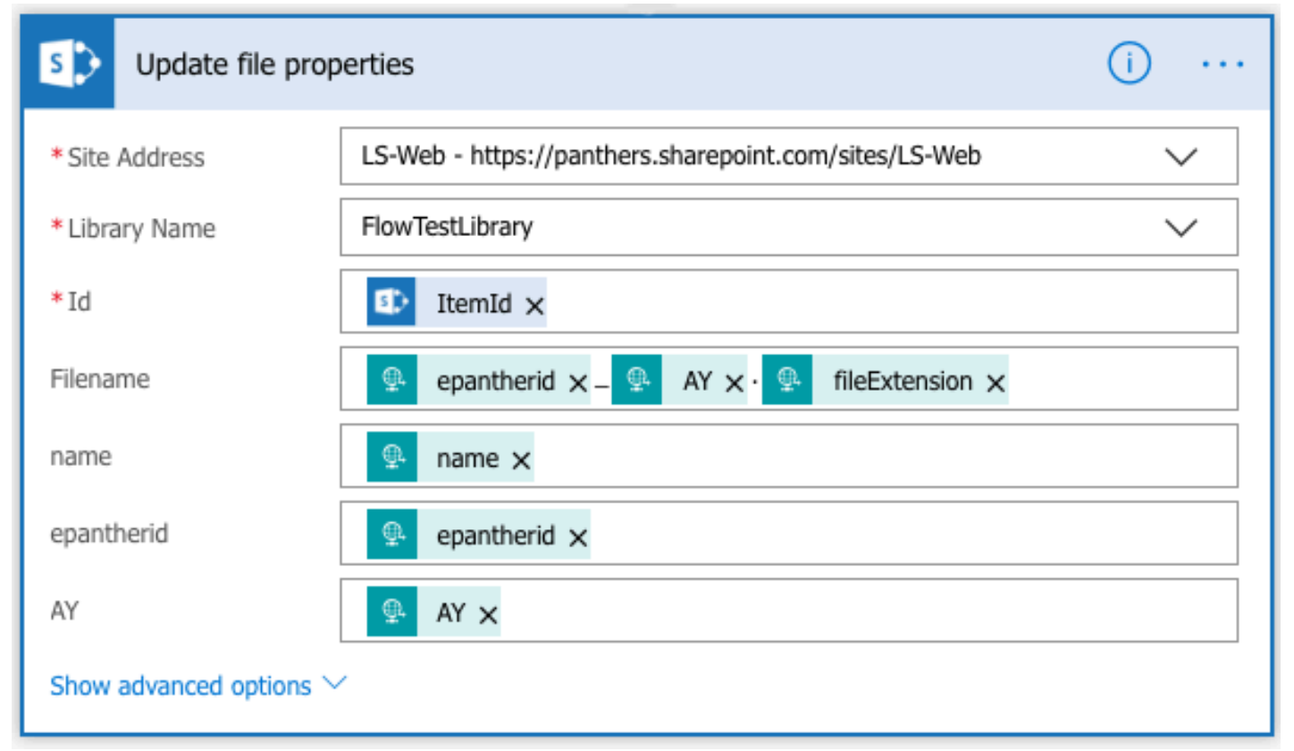
* Site Address: LS-Web - https://panthers.sharepoint.com/sites/LS-Web

* Folder Path: FlowTestLibrary/ epantherid

* File Name: file_ epantherid _ AY . fileExtension

* File Content: Output

Step 3: Create File
a) Set Folder Path
b) Rename File Systematically
c) Input File Content



Update file properties

* Site Address: LS-Web - https://panthers.sharepoint.com/sites/LS-Web

* Library Name: FlowTestLibrary

* Id: ItemId

Filename: epantherid _ AY . fileExtension

name: name

epantherid: epantherid

AY: AY

Show advanced options

Step 4: Update File Metadata

References

Hulette, J. Wesley. (2012, June 23). How to log into Office 365 or SharePoint using PHP. [Blog post]. Retrieved from https://macfoo.wordpress.com/2012/06/23/how-to-log-into-office365-or-sharepoint-online-using-php/

Microsoft Corporation. (2017, December 13). Use OData query operations in SharePoint REST requests. Retrieved from https://docs.microsoft.com/en-us/sharepoint/dev/sp-add-ins/use-odata-query-operations-in-sharepoint-rest-requests

Microsoft Corporation. (2018, February 6). Get to know the SharePoint REST service. Retrieved from https://docs.microsoft.com/en-us/sharepoint/dev/sp-add-ins/get-to-know-the-sharepoint-rest-service

Contact Info

Danny Harvey
Letters and Science Web and Data
Merrill Hall, G34
University of Wisconsin-Milwaukee
Milwaukee, WI 53211

Email: harveyd@uwm.edu