Datum360

Digital Adoption Platform

Using Microsoft Power Automate/Logic apps with PIM360

Steve Bayliss & Matthew Frost

Document ID: D360-TPRD-GUD-0010

Issue Date: 16-Mar-2023

Revision: 1

Datum360

Connected data

Revision History

Issue	Summary of changes
1	Issued for Use

1	16 Mar 23	Issued for Use	SB	16 Mar 23	MF	16 Mar 23	TP	16 Mar 23
Issue	Issue Date	Reason for Issue	Author	Date	Checked	Date	Approved	Date

Copyright ©2023 Datum360 Limited
All Rights Reserved

Contents

1 API Basics 4
1.1 API Documentation Online (Swagger UI) 4
1.2 Open API Endpoint 7
1.3 Authentication 8
1.4 Accounts and Permissions 10
2 Exporting Data from PIM360 with Power Automate 11
2.1 Create a Manual Flow 11
2.2 Bearer Token Request for Subsequent API Calls 12
2.3 Parse the Token Request Output into JSON 14
2.4 Make an API Request Using the HTTP + Swagger Action 15
2.5 Run the Flow 18
3 Listening to PIM360 Messages 19
3.1 Create an Automated Flow 19
3.2 Connection String and Subscriptions 20
3.3 Assign an Action when a Message is Received 21
3.4 Run the Flow 22
4 Uploading Data into PIM360 23
4.2 Get a File to Load into PIM360 23
4.3 Make an API Request using HTTP 24
4.4 Run the flow 28

Copyright ©2023 Datum360 Limited
All Rights Reserved

1 API Basics

1.1.1 The Datum360 Connected Data Platform API is a REST API providing JSON output.

1.1.2 The API is documented using the OpenAPI version 2 (Swagger) standard.

1.1.3 Human-readable documentation is provided within the application and the JSON description of the API is available to tools that can interpret OpenAPI version 2.

1.1.4 API authentication is via OAuth2.

1.1 API Documentation Online (Swagger UI)

1.1.1 The API documentation can be viewed online from within the Datum360 Connected Data Platform.

1.1.2 After logging into either CLS360, PIM360 or DDM360, click on the user account name at the top-right of the screen.

1.1.3 From the drop-down menu that appears, select API Documentation.

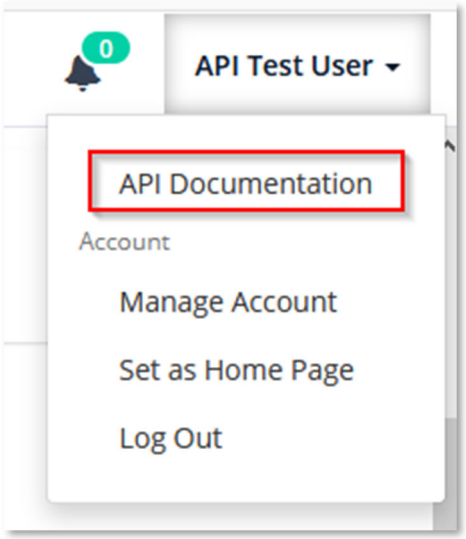


Fig 1. Accessing API Documentation

1.1.4 To execute API calls directly from the API Documentation (Swagger UI) page, API credentials must be generated.

1.1.5 To generate API credentials, select the Authorize button.

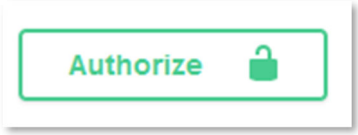


Fig 2. API Documentation Page Authorize button

1.1.6 From the Available authorizations dialogue box that appears populate the following:

a. Client_id= the username of the account that will access the API.

All Rights Reserved

b. Client_secret = the password of the account that will access the API.

c. Scopes = select at least one of the available scopes.

1.1.7 Click the Authorization button.

a. oAuth2SchemeAccessCode is the best option for SSO users.

b. oAuth2Scheme is the best option for users with a username and password.

c. Only one option needs to be used, as both authorizations achieve the same objective.

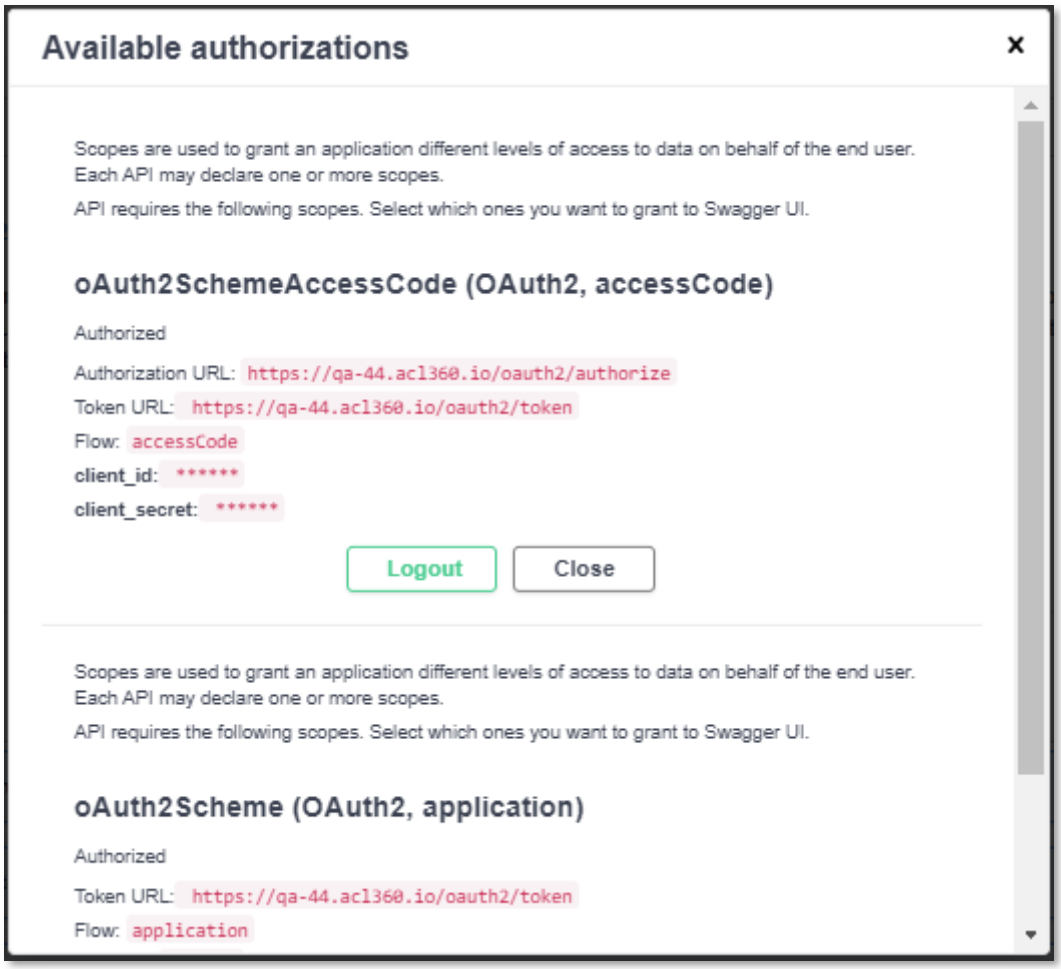


Fig 3. API Documentation Authorization Button

1.1.8 The system will now login. If the supplied credentials are incorrect, or the user account does not have sufficient permissions to access the API (see section 1.4), a Auth Error will be displayed, and access to the API will not be available until a successful login has been achieved.

1.1.9 SSO accounts will require a dedicated service account to be set up:

a. The service account name can be anything e.g., API User.

b. Only one capability "CanLogin" needs to be added (for each service requiring access).

Copyright ©2023 Datum360 Limited
All Rights Reserved

c. All users who require API access will need to have these credentials shared with them.

d. The users with the service account credentials will still be restricted by the API capabilities assigned to their account, i.e., the users person account will need to have the applicable capabilities added, as listed in Section 1.4.

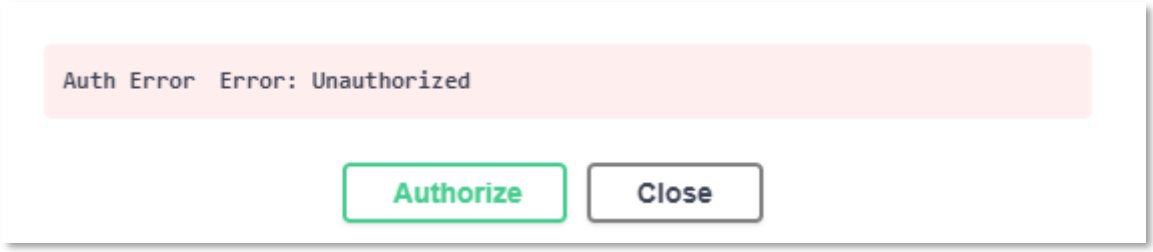


Fig 4. Authorization Error

All Rights Reserved

1.2 Open API Endpoint

1.2.1 A JSON document describing the API using the Open API 2 (Swagger) standard is available at a URL that does not require authentication.

1.2.2 This is visible towards the top-left of the API Documentation page.

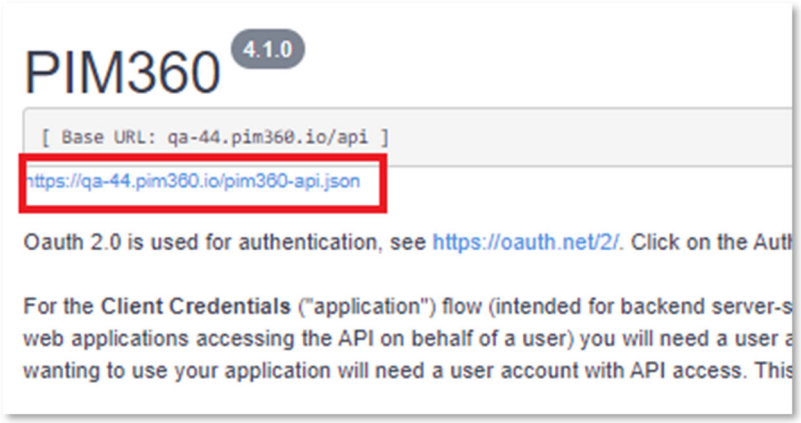


Fig 5. JSON Document Describing the API Access for the qa-4.4 System

1.2.3 This URL can be supplied to tools that understand OpenAPI v2 to autogenerate API connectors.

1.3 Authentication

1.3.1 The API uses bearer tokens for authentication.

1.3.2 Bearer tokens must be added to API requests as an Authorization header.

```
Authorization: Bearer <value of the token here>
```

Fig 6. Example Authorization Header with Bearer Token

1.3.3 The token is obtained via OAuth2.

1.3.4 The Datum360 Connected Data Platform API supports two- and three-legged OAuth2 flows.

a. for service interactions it is recommend using the 2-legged flow, known in OpenAPI/Swagger parlance as the "application" flow and commonly referred to as the OAuth2 "Client Credentials" flow.

1.3.5 A HTTP POST operation should be performed to the token endpoint (see below) specified in the OpenAPI JSON document with:

a. A Basic authorisation header containing the client id and client secret, i.e., `Authorization: Basic <base64 encoded version of client id:client secret>`

b. A content-type header of "application/x-www-form-urlencoded"

c. The body of the post should specify: `grant_type=client_credentials&scope=example system-pim360` replacing the scope with the one specified (see fig 8) in the OpenAPI JSON document.

1.3.6 A JSON document will be returned specifying the access token.

```
{
  "access_token": "eyJhb […] JFQ",
  "token_type": "Bearer"
}
```

Fig 7. Returned JSON Document with Access Token

1.3.7 The value of "access_token" should be used in subsequent API calls as a Bearer token (see 1.3.6).

Important: Bearer tokens expire after 30 minutes, long-running processes that makes a number of API calls would benefit from making a fresh bearer token for each request.

Copyright ©2023 Datum360 Limited
All Rights Reserved

```
{
    "swagger": "2.0",
  [...]
    "securityDefinitions": {
        "oAuth2SchemeAccessCode": {
            [...]
        },
        "oAuth2Scheme": {
            "type": "oauth2",
            "tokenUrl": "https://example-
system.acl360.io/oauth2/token",
            "flow": "application",
            "scopes": {
                "example-system-pim360": "PIM360"
            }
        }
    },
    "host": "qa-40.pim360.io",
    "schemes": ["https"]
}
```

Fig 8. Example JSON OpenAPI Document with the Token Endpoint and the Scope Highlighted.

1.3.8 Alternatively, this can be achieved via the API Documentation (Swagger UI) page, by selecting Authorize button (see Fig 2).

1.3.9 Scroll down to the Application flow (oAuth2Scheme (OAuth2, application), and populate the required information as in 1.1.6).

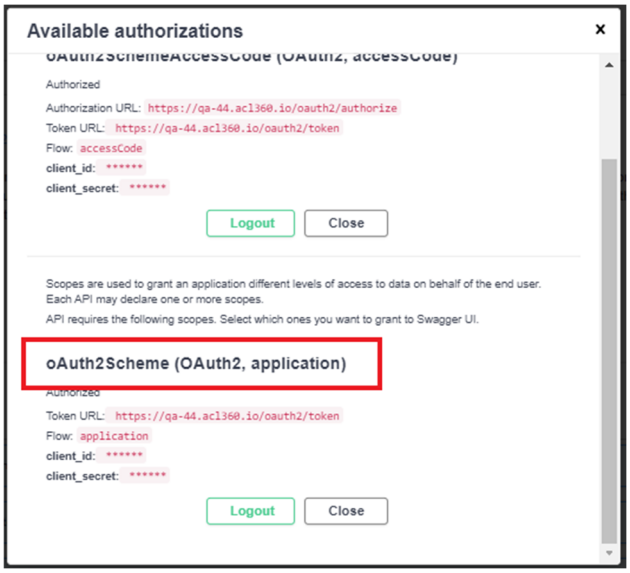


Fig 9. Application flow authorization

Copyright ©2023 Datum360 Limited
All Rights Reserved

1.4 Accounts and Permissions

1.4.1 The service account to be used should be created in ACL360 and be given the necessary permissions that the API service requires.

1.4.2 The service account must be given the capability “canUseApi”.

1.4.3 It is recommended to use separate service accounts rather than re-using an existing ACL360 user account.

All Rights Reserved

2 Exporting Data from PIM360 with Power Automate

2.1 Create a Manual Flow

2.1.1 From within Power Automate:

a. Click on + Create.

b. Click on Instant cloud flow.

c. In the dialog appears, select "Manually trigger a flow and give the Flow a name.

d. Click Create.

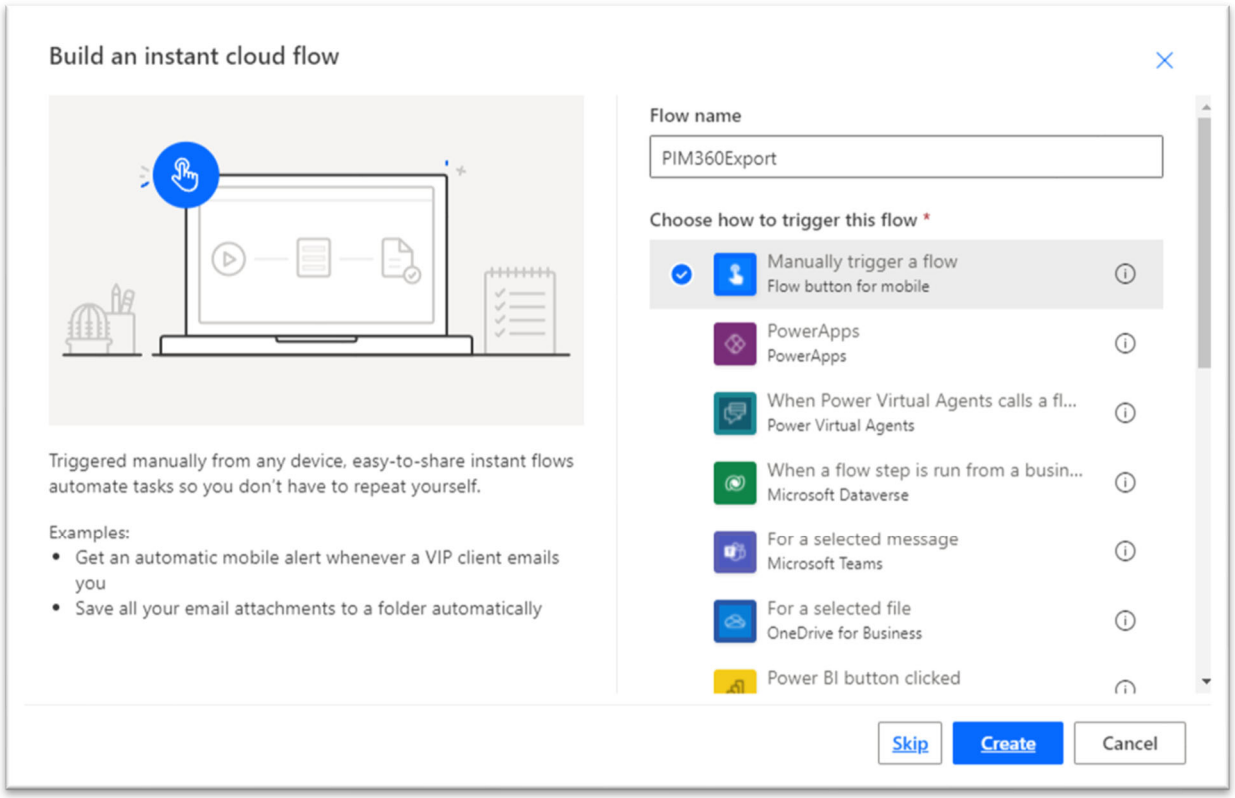


Fig 10. Creating a Manual Flow

Copyright ©2023 Datum360 Limited
All Rights Reserved

2.2 Bearer Token Request for Subsequent API Calls

2.2.1 Within the newly created Flow, click on the new step button.

2.2.2 In the choose an operation form that opens, search for HTTP, and select HTTP from the Actions section.

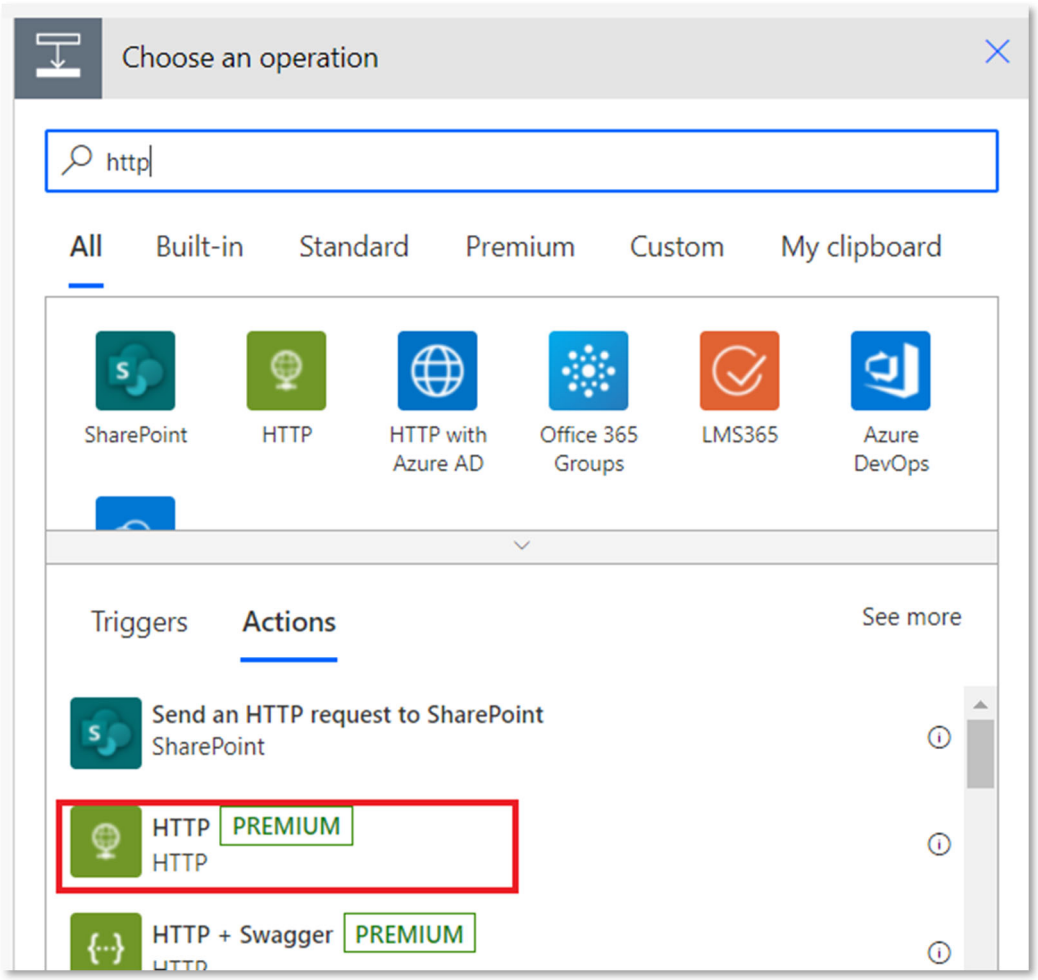


Fig 11. Select Standard HTTP Action

2.2.3 Populate the displayed parameters as follows:

a. Method: the API method being used, in this example use POST.

b. URI: the Application flow (oAuth2Scheme (OAuth2, application) Token URL (as seen in Fig 9).

c. Headers:

1. Key: content-type
2. Value: application/x-www-form-urlencoded

d. Queries:

1. Key: blank.
2. Value: blank.

Copyright ©2023 Datum360 Limited
All Rights Reserved

e. Body: grant_type=client_credentials&scope=<scope>

f. Cookie: blank.

g. Authentication: Basic.

h. Username: username for the account accessing the API.

i. Password: password for the account accessing the API.

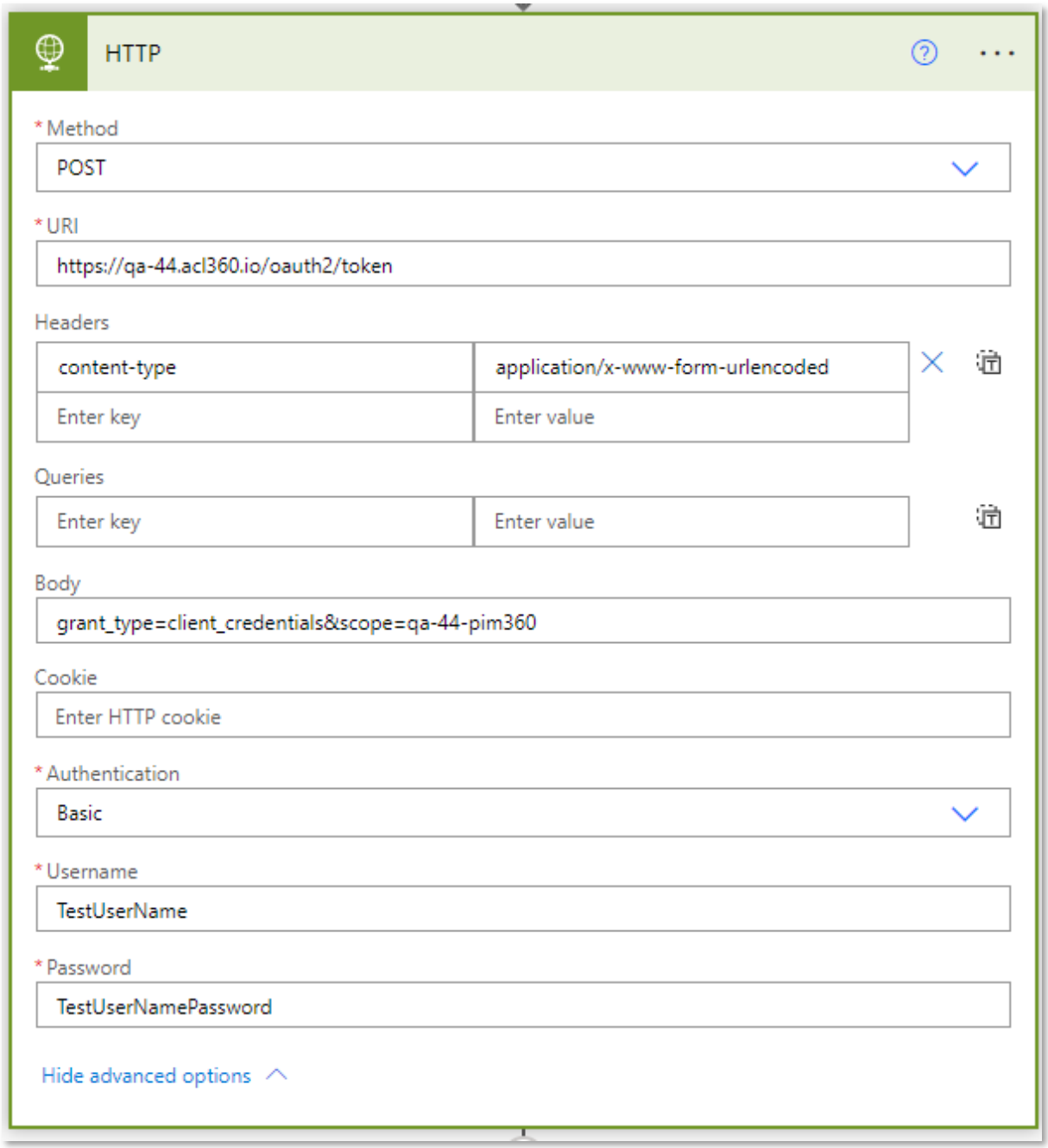


Fig 12. HTTP Action Parameters Example

2.3 Parse the Token Request Output into JSON

2.3.1 The HTTP request above will just provide raw data, the following action will transform it into JSON using the schema specified so that subsequent actions can reference the token easily.

2.3.2 Click on the new step button, below the HTTP step created in 0.

2.3.3 In the choose an operation form that opens, search for parse JSON, and select Parse JSON from the Actions section.

2.3.4 Populate the form that appears with the following parameters:

a. Content: Body (select from the available blocks).

b. Schema:

```
{
    "properties": {
        "access_token": {
            "type": "string"
        },
        "token_type": {
            "type": "string"
        }
    },
    "type": "object"
}
```

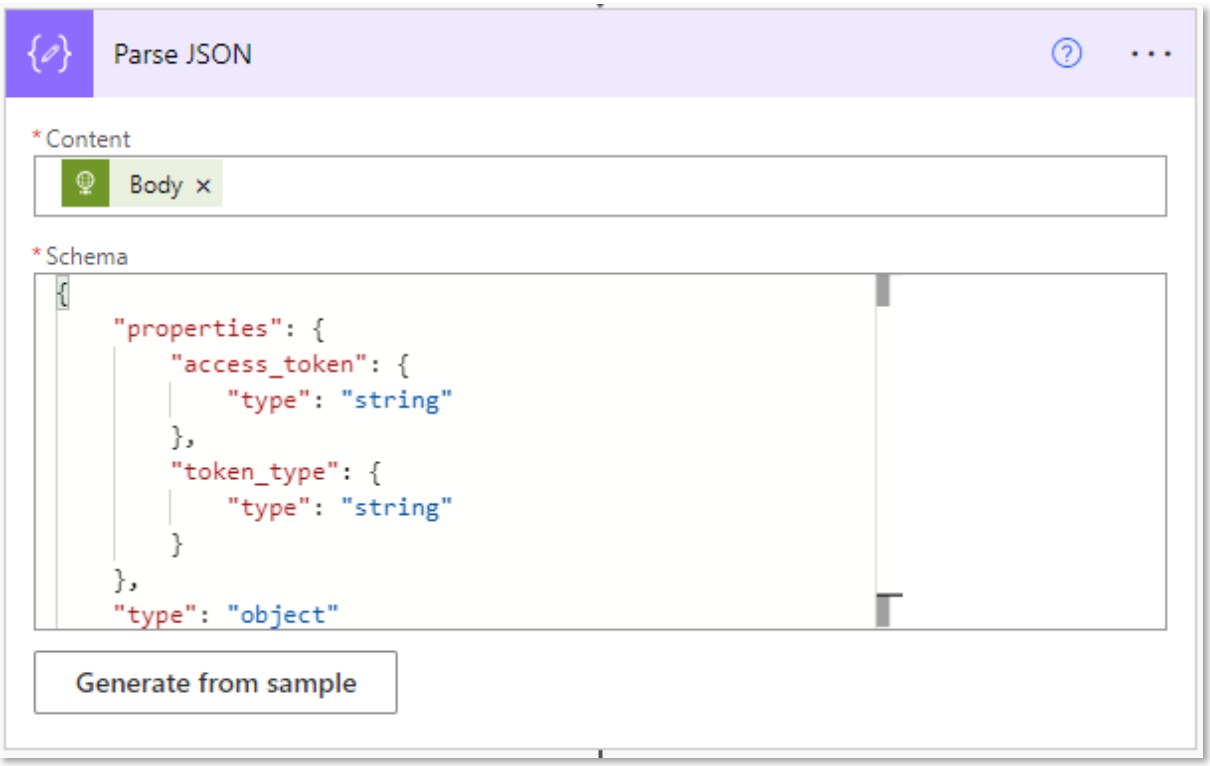


Fig 13. Parse JSON Step

2.4 Make an API Request Using the HTTP + Swagger Action

2.4.1 Click on the new step button, below the Parse JSON step created in 0.

2.4.2 In the choose an operation form that opens, search for HTTP +Swagger, and select HTTP +Swagger from the Actions section.

Fig 14. HTTP + Swagger Action

2.4.3 A prompt for the JSON OpenAPI endpoint will appear, enter in the endpoint url, as shown in Fig 5).

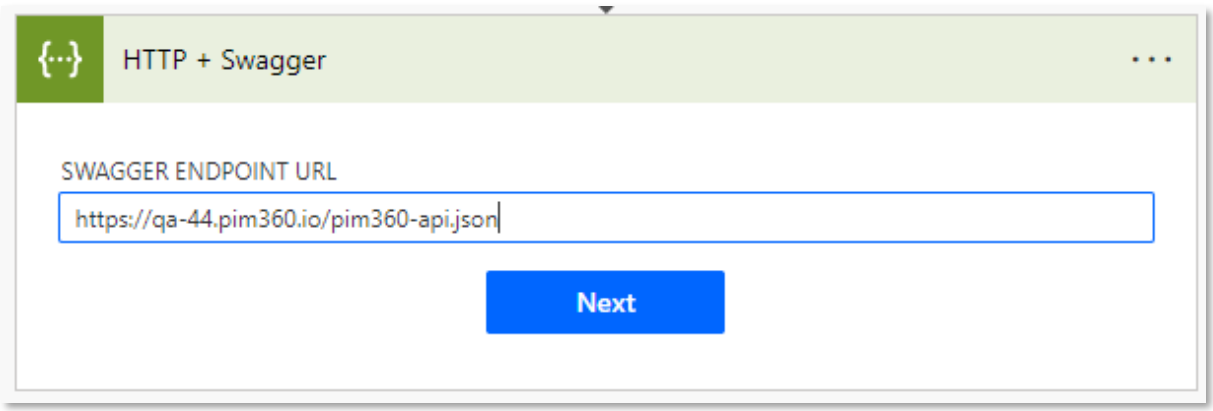


Fig 15. HTTP + Swagger URL Endpoint Example

2.4.4 Click Next and the HTTP + Swagger action will retrieve the JSON OpenAPI documentation and will list possible API calls described in the documentation.

Copyright ©2023 Datum360 Limited
All Rights Reserved

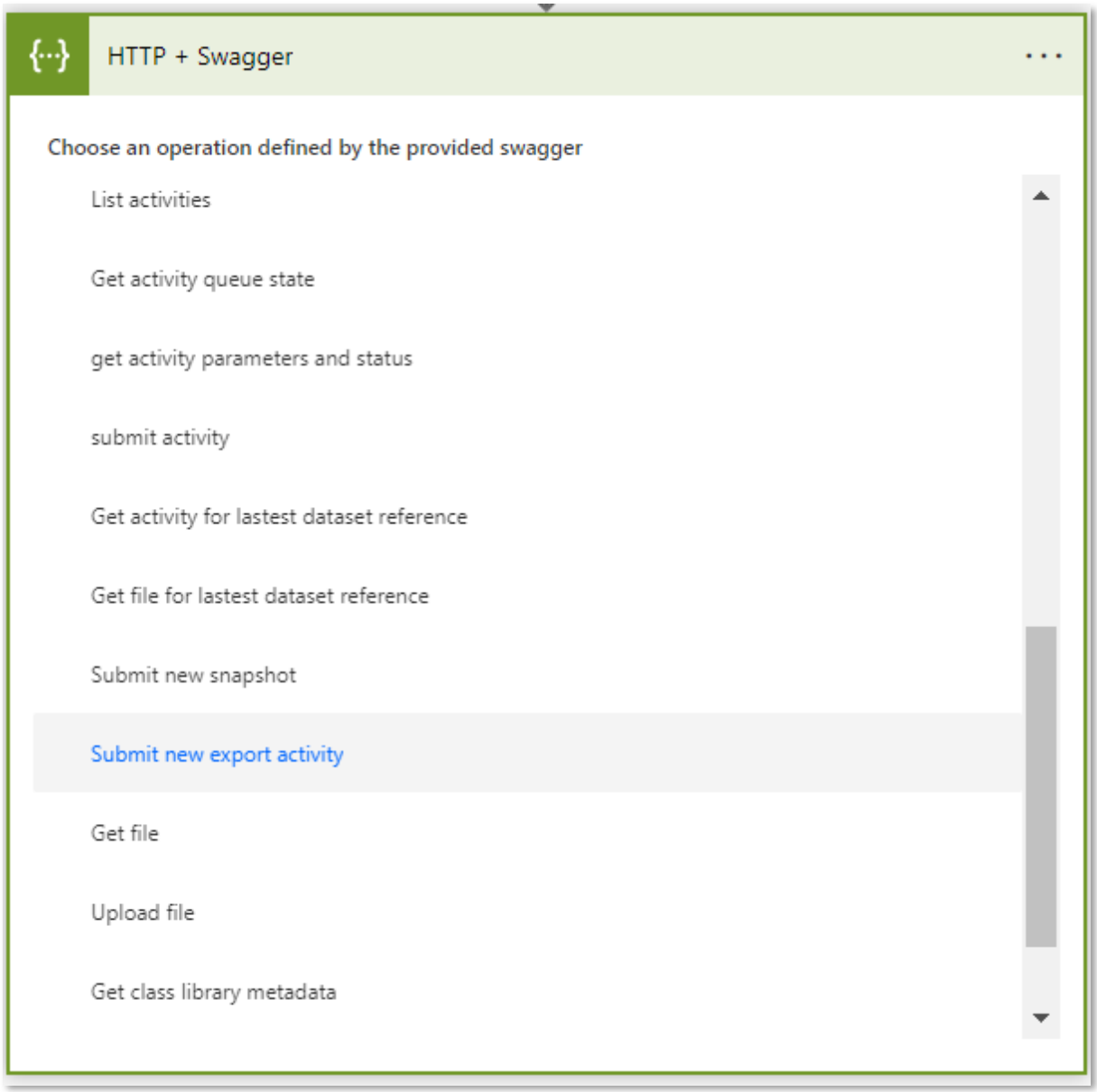


Fig 16. All Possible API Calls for Example

2.4.5 Select Submit new export activity for this worked example.

2.4.6 In the form that appears populate the following parameters:

a. liveviewHdl: the handle (unique ID) of the saved LiveView from PIM360. (handle can be obtained using the PIM360 GET /customviews call).

b. etlTargetHdl: the handle (unique ID) of the CLS360 Data target. (handle can be obtained by using the CLS360 GET /domains/{domHdl}/classes, using [Object Type] eq [Data Target] as the parameter. This call also requires the class library (domain) as a parameter, this can be obtained by using the CLS360 GET /domains).

c. outputFormat: select the output format of the file. xlsx is recommended.

d. eicHdl: the handle (unique ID), of the EIC to get the data from. Leave blank if the

Copyright ©2023 Datum360 Limited
All Rights Reserved

data is coming from active values. (handle can be obtained by using the PIM360 Get /eic/list call).

e. concatUoms: if units of measure should be shown in the same field as the measure attributes value, select Yes, otherwise No (and the unit of measure will be listed in the adjacent field to the measure attribute).

f. missingValue: the string value that should be used to represent a required attribute that is currently not in the data set. Leave blank if the field should be empty.

g. datasetReference: the name this export will be given to enable it to be accessible via API calls. Leave blank if not required.

h. Authentication:

```
{
"type": "raw",
"value": "Bearer 'access_token'"
}
```

1. 'access_token' should be replaced with the dynamic content access_token. The dynamic content access_token is the result of the OAuth2 bearer token request and using the bearer token in the authentication header.

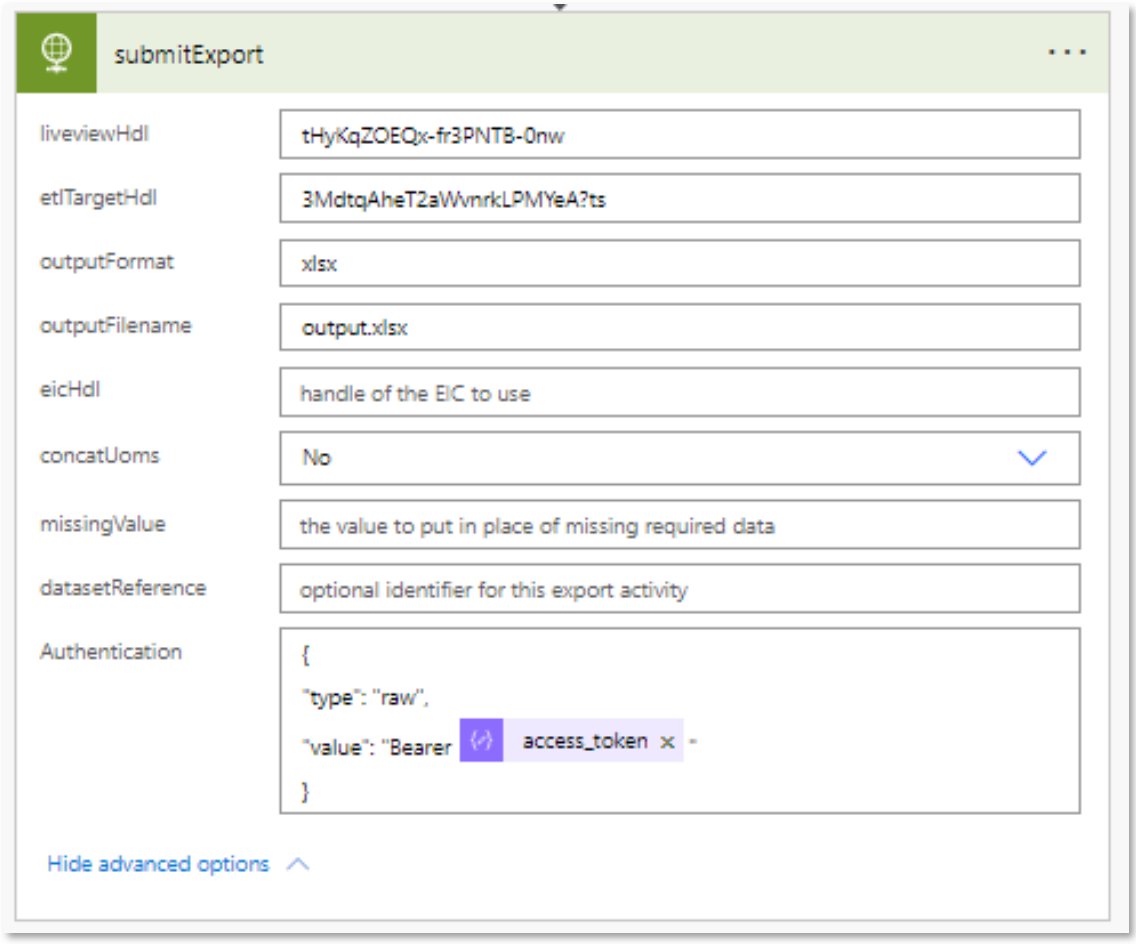


Fig 17. HTTP + Swagger Example

All Rights Reserved

2.5 Run the Flow

2.5.1 Click Test on the top right of the page and select manually from the Test Flow menu.

Fig 18. Test Flow Menu

2.5.2 Click the Save & Test button.

2.5.3 Click the Run flow button. A message will appear to confirm the flow has started.

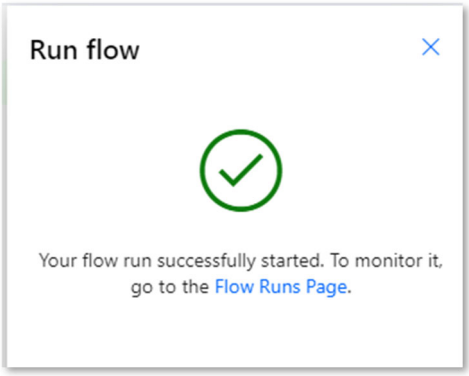


Fig 19. Confirmation of Flow Starting to Run

2.5.4 Click the Done button. The screen will refresh and redirect to the Flow Runs page.

2.5.5 The Flow run page will indicate if each step has completed successfully. Common reasons for non execution are incorrect parameters and token expiry.

2.5.6 Click on the Edit button in the top right-hand corner to return to the editable Flow.

2.5.7 Open PIM360 and see the export has run.

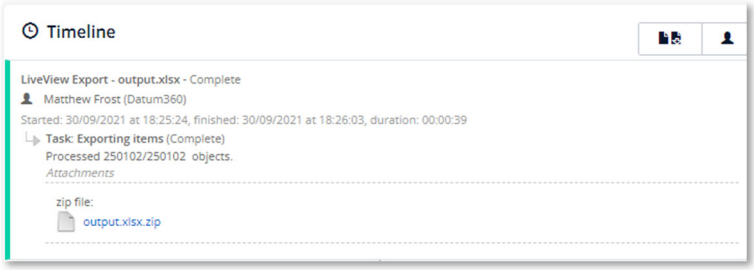


Fig 20. PIM360 Administration Timeline Showing Successful Export

All Rights Reserved

3 Listening to PIM360 Messages

Important: The system must be configured to utilise this function. Contact Datum60 via Zendesk to confirm configuration settings.

3.1 Create an Automated Flow

3.1.1 From within Power Automate:

a. Click on + Create.

b. Click on Automated cloud flow.

c. In the dialog appears, select When a message is received in a topic from the triggers and give the Flow a name.

d. Click Create.

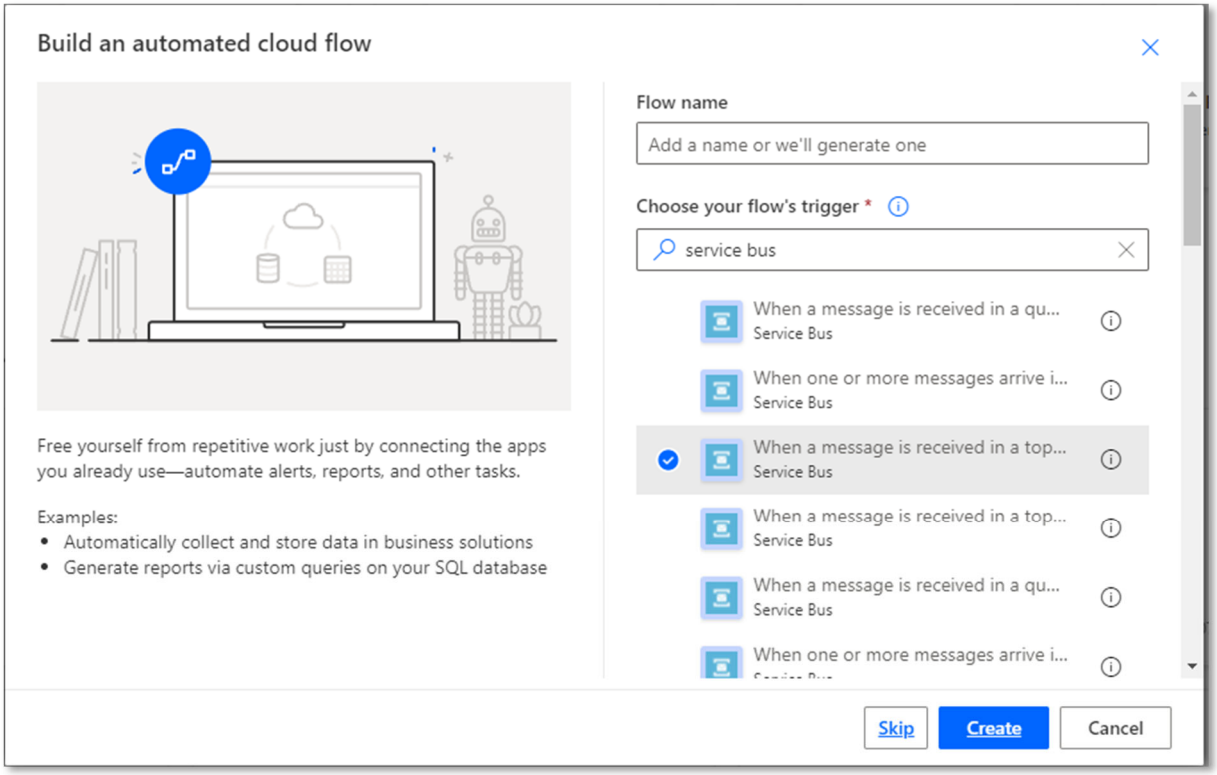


Fig 21. Automated Cloud Flow Set Up

3.2 Connection String and Subscriptions

3.2.1 The next dialogue that appears will request the connection details, populate as follows:

a. Connection name: name the connection (this can be anything).

b. Connection string: the connection sting from a 'shared access policy' on the service bus to be used in Azure.

1. The Azure service is normally centrally managed by an IT / Technical department from within a company, this department should be able to supply the connection string.

2. See Datum360s Bitbucket documentation 'Setting up Azure Service Bus for Datum360 services' for guidance if this has not been set up yet.

c. Click Create.

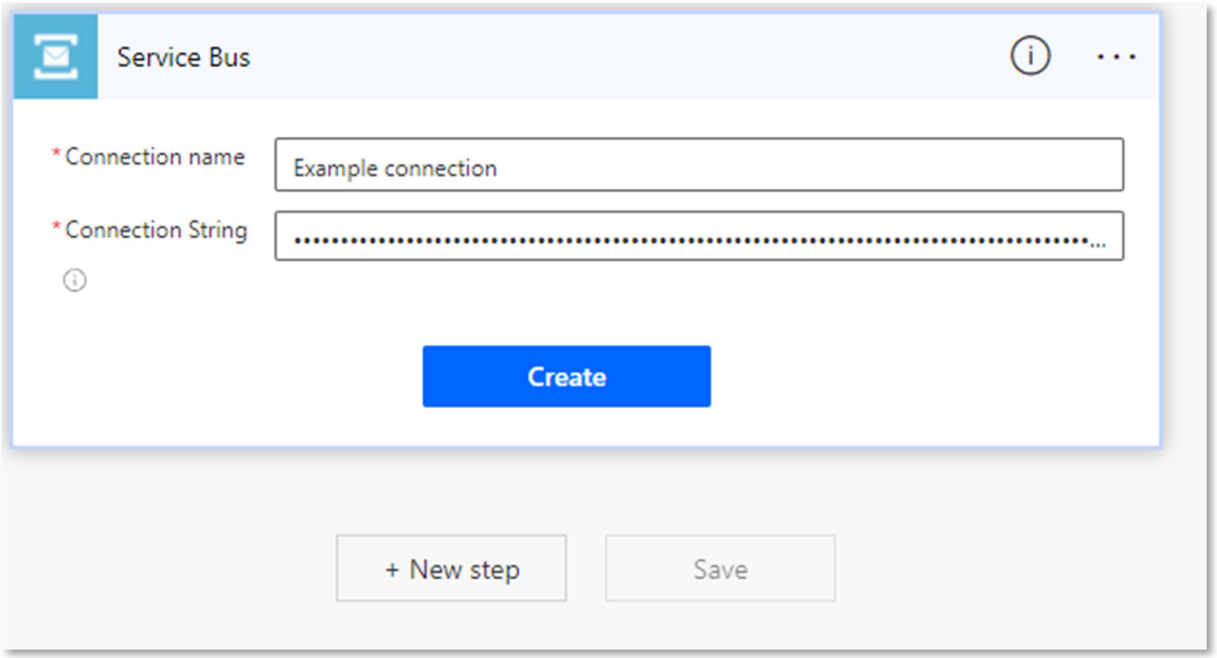


Fig 22. Connection to Service Bus

3.2.2 The 'When a message is received in a topic subscription' dialogue will now appear.

3.2.3 Within the service bus set up in Azure, there will be topics for cls360, ddm360 and pim360, which will be available in the Topic name field drop down.

a. The Topics will need to be configured by a user/department that has write access to the Azure service bus.

3.2.4 Select pim360.

3.2.5 Within the service bus set up in Azure there will be subscriptions to these topics set up, which will be available in the Topic subscription name field.

a. The Topic subscriptions will need to be configured by a user/department that has write access to the Azure service bus.

Copyright ©2023 Datum360 Limited
All Rights Reserved

3.2.6 Select the required subscription.

a. The Topic subscription name will need to be configured by a user/department that has write access to the Azure service bus.

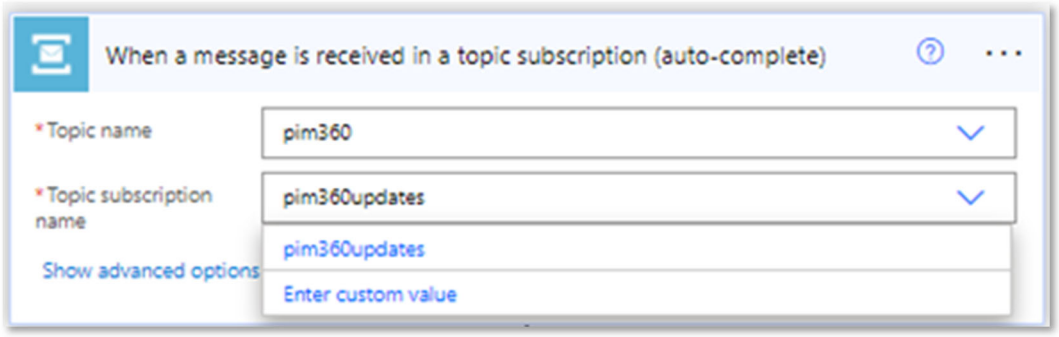


Fig 23. Selecting Topics and Subscriptions

3.3 Assign an Action when a Message is Received

3.3.1 Click the + New Step button below the 'When a message is received in a topic subscription (auto complete)' step.

a. The new step being added will be the action that completes once a message is received. In this example a message will be sent to a Microsoft Teams channel.

3.3.2 In the new step, choose an operation box, search for and select Post a message in a chat or channel.

3.3.3 Populate the dialogue that appears as follows:

a. Post as: Flow bot or User

b. Post in: Channel

c. Team: The Teams channel that the message should be posted in.

d. Channel: The team with in the above selected channel that the message will be posted.

e. Message: What the post should say, dynamic content can also be added here.

Copyright ©2023 Datum360 Limited
All Rights Reserved

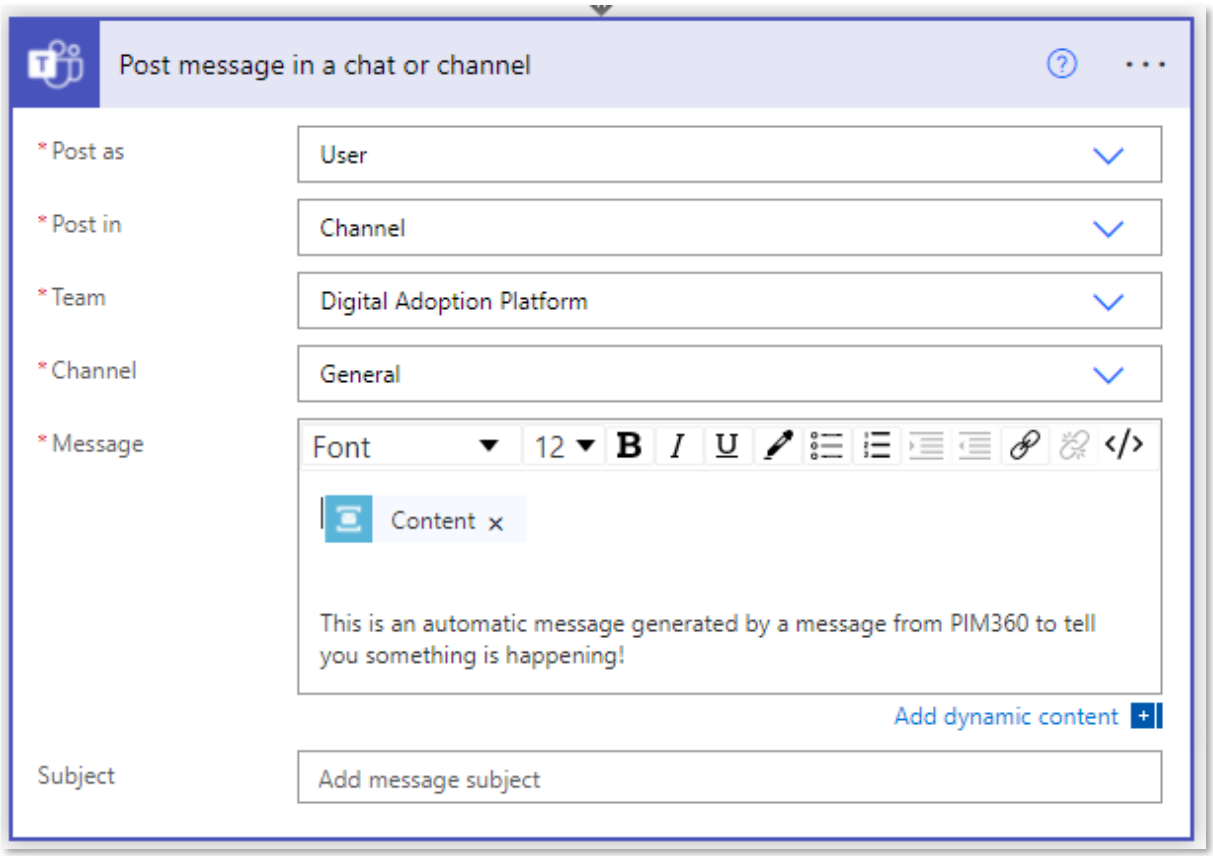


Fig 24. Post a Message to Teams Example

3.4 Run the Flow

3.4.1 Save the flow.

3.4.2 Run an export in PIM360 and see the automatic message appear in the selected Teams channel.

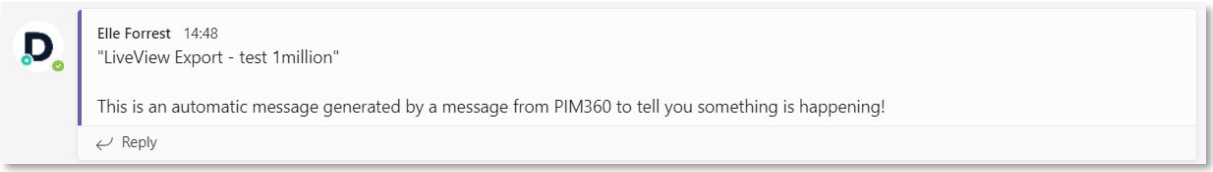


Fig 25. Example Teams Message

Copyright ©2023 Datum360 Limited
All Rights Reserved

4 Uploading Data into PIM360

4.1.1 This example illustrates how to upload a file from a cloud service (in this example OneDrive is used) and load it into PIM360.

4.1.2 Follow the steps in 2.1, 0 and 0 to make a token request and parse the JSON.

4.2 Get a File to Load into PIM360

4.2.1 Click on the new step button

4.2.2 In the dialogue that appears search for, Get file metadata path, and select it from the results in the Actions.

4.2.3 In the step that appears, navigate to the load file that will be used, and select it from the list. The file name and path will now appear in the *File field. In this test a tab delimited text file, a wide import format is being loaded.

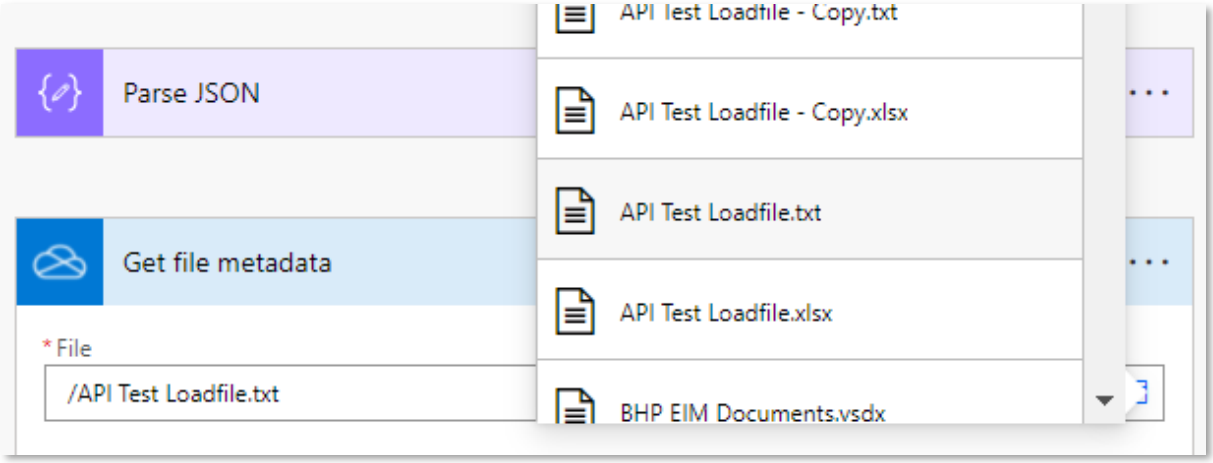


Fig 26. Selecting File

4.2.4 Click on the new step button.

4.2.5 In the dialogue that appears search for Get file content and select it from the results in the Actions.

4.2.6 In the step that appears, populate the *File field with the dynamic content from the previous step, Id, by placing the cursor in the field and clicking on the Id parameter in the list that appears.

Copyright ©2023 Datum360 Limited
All Rights Reserved

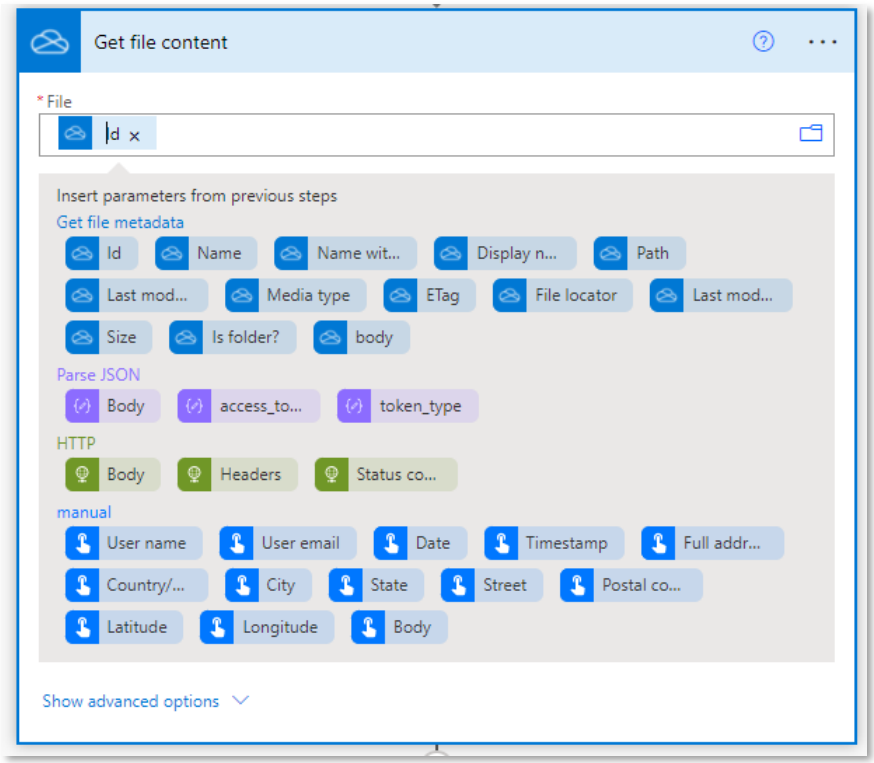


Fig 27. Get File Content

4.3 Make an API Request using HTTP

4.3.1 Click on the New step button

4.3.2 In the dialogue that appears search for HTTP and select it from the results in the Actions. This will be added as HTTP 2, as it is the same Action as the first step added.

4.3.3 Populate the step that appears with the following:

a. *Method: POST

b. *URI: the address to the post call. An example is shown below, the words highlighted in yellow should be replaced with the name (scope) of the PIM360 environment being used.

1. https://qa-master.pim360.io/api/etl_queue/activities/wide_import

c. Headers: blank

d. Queries: blank

e. Body:

```
{
  "$content-type": "multipart/form-data",
  "$multipart": [
    {
      "body": "Pi6oMK48SlyGfbVmOWoxbQ",
```


All Rights Reserved

```
    "headers": {
      "Content-Disposition": "form-data; name=\"eicHdl\""
    }
   },
   {
    "body": "jTtqE13-SRi0Gk30LIc9xQ",
    "headers": {
      "Content-Disposition": "form-data; name=\"etlSourceHdl\""
    }
   },
   {
    "body": "TAGGED_ITEM",
    "headers": {
      "Content-Disposition": "form-data; name=\"objectType\""
    }
   },
   {
    "body": "cls",
    "headers": {
      "Content-Disposition": "form-data; name=\"classification\""
    }
   },
   {
    "body": @{body('Get_file_content')},
    "headers": {
      "Content-Disposition": "form-data; name=\"upfile\";
filename=\"@{outputs('Get_file_metadata')?['body/Name']}\""
    }
   }
  ]
}
```

f. Observe the following:

1. The value that is in the “body”: field is the value that will be used in the Content-Disposition identified form field in the following header section.

All Rights Reserved

2. The string "Content-Disposition": "form-data; name=\"etlSourceHdl\"" identifies which field the following data will be populating. The field names can be found in the Swager API documentation.

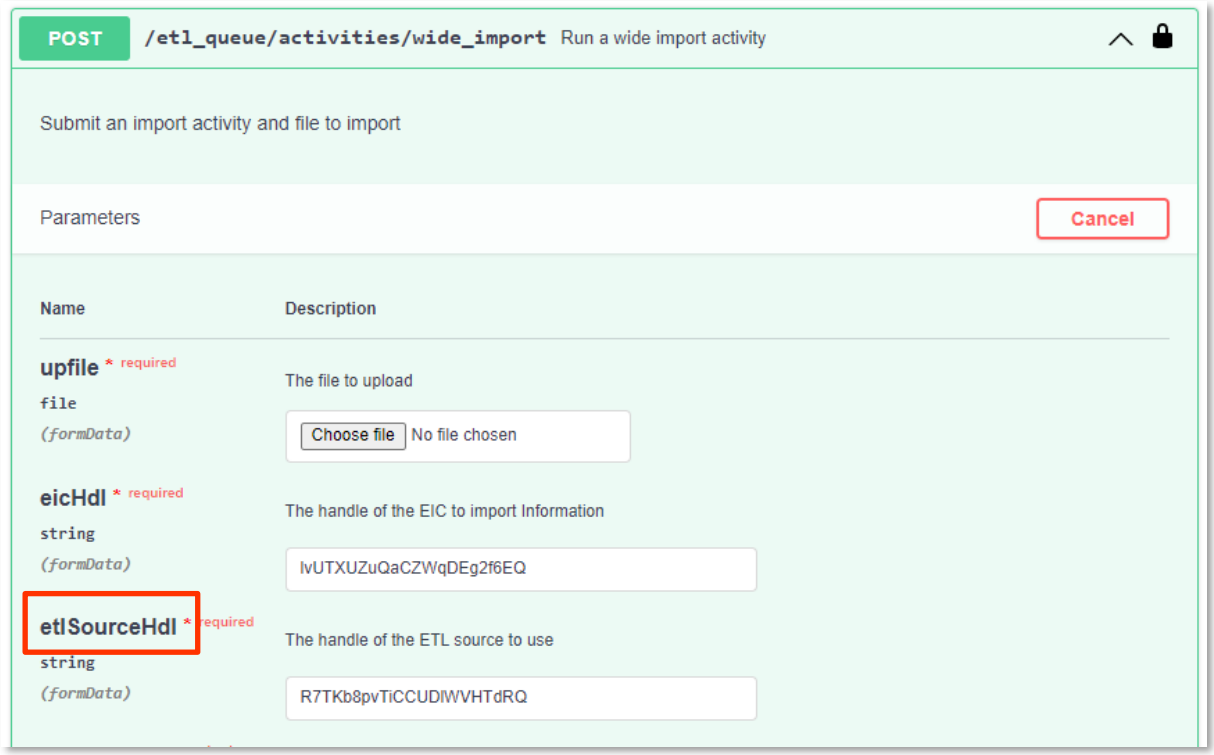


Fig 28. Field Names for Web Forms in Swagger API Documentation

3. Values with @ are dynamic values from previous steps, these should convert to visual representations, see diagram below. If the dynamic contents don't convert to visual representations and stay as text, the text should be removed, and the dynamic content inserted manually.

Copyright ©2023 Datum360 Limited
All Rights Reserved

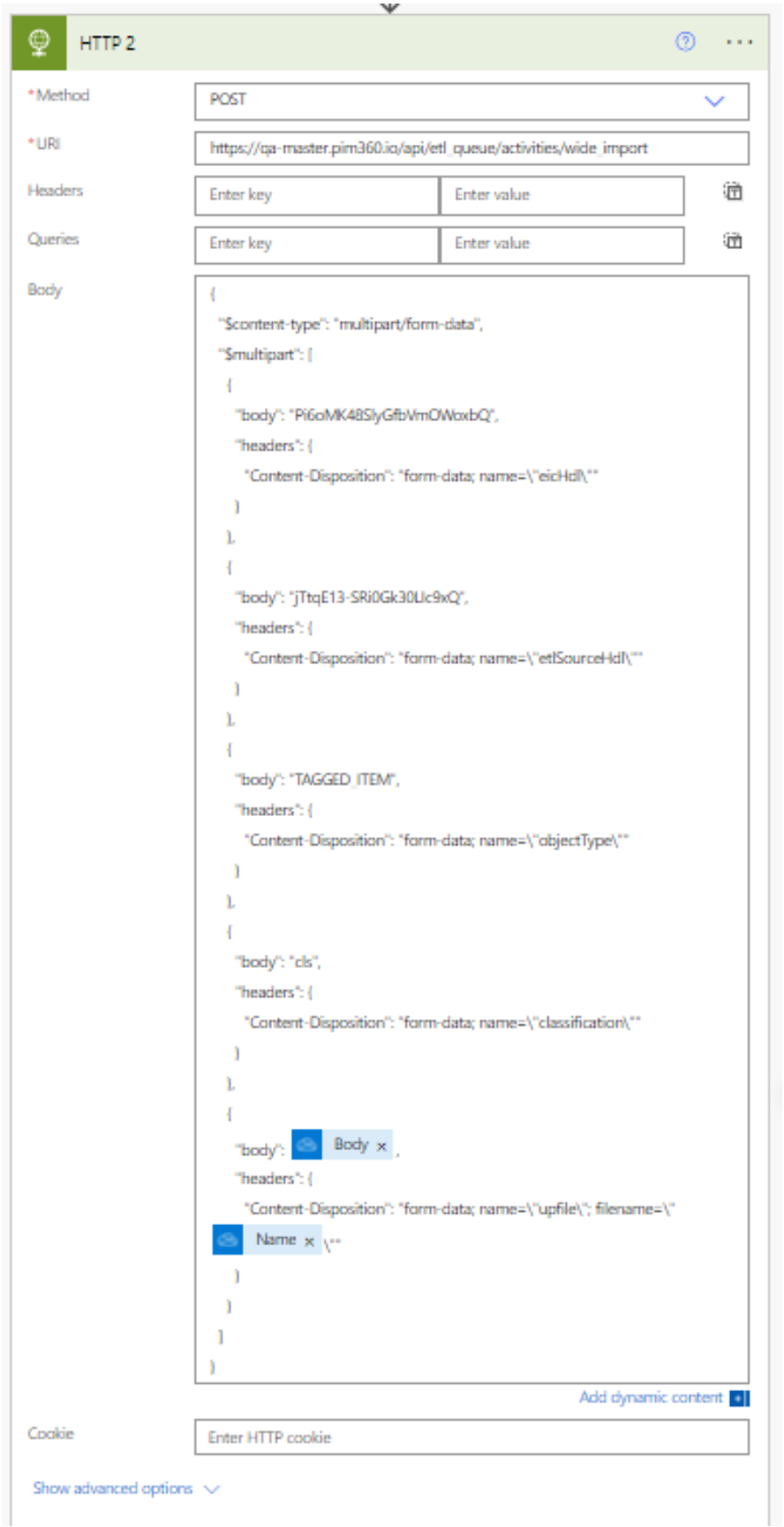


Fig 29. Examples of Dynamic Content

4. The handles for the eic and deliverable can be obtained from GET requests to PIM360 (see Swagger API documentation, eic =/objects/{type}/handles, deliverable = /eic/{eicHdl}/deliverables).

5. The handle for etlSourceHdl can be obtained from GET requests to CLS360, (see Swagger API Documentation, /domains/{domHdl}/classes [Object Type] eq [Data Source].

g. Cookie: blank

h. Authentication: Raw

i. Value: Bearer @{body('Parse_JSON')?['access_token']}

4.3.4 Click Save.

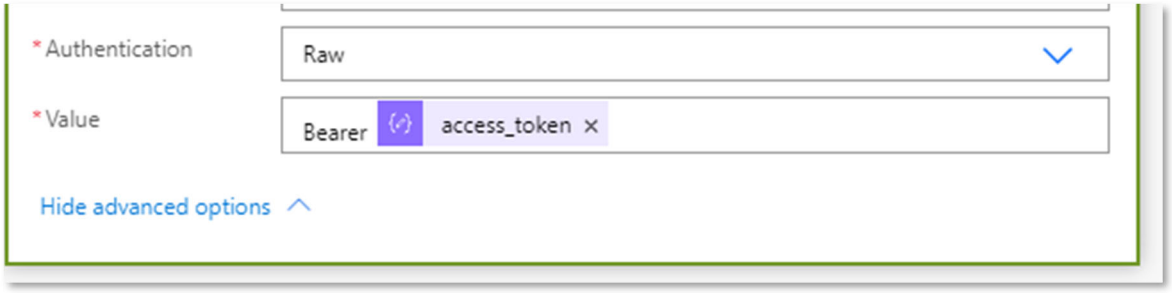


Fig 30. Authorisation for POST Request

4.4 Run the flow

4.4.1 From the edit flow area, select Test. The button is located in the top right-hand corner of the screen.

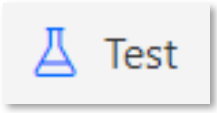


Fig 31. Test Button

4.4.2 In the right had side menu that appears, select Manually.

4.4.3 Select the blue Test button.

4.4.4 Depending on previous set up, MS Flow may now require sign into the document management system being used (in this instance OneDrive), to confirm the flows permissions to access the named files in the flow. If required sign in.

4.4.5 If permissions have already been granted, a green tick will show against the service requiring permission.

4.4.6 Click the blue Continue button.

4.4.7 Click the Run flow button.

4.4.8 A message will show indicating that the flow run successfully started. Click the blue Done button to be redirected to the Flow runs page, to see the if the flow completed successfully. If it didn't the step it failed on will be highlighted and error checking / fixes can be applied be going back to the edit area.

Copyright ©2023 Datum360 Limited
All Rights Reserved

4.4.9 On successful completion the loaded file will be shown in the EIC timeline

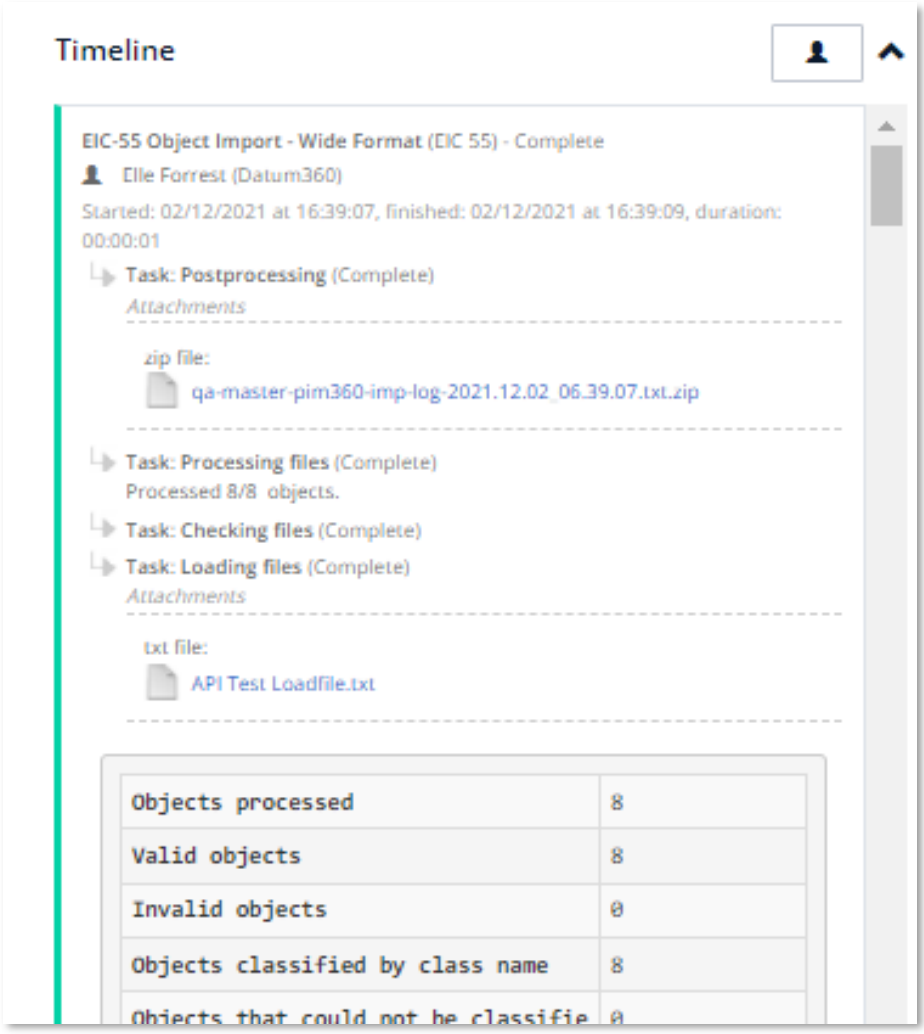
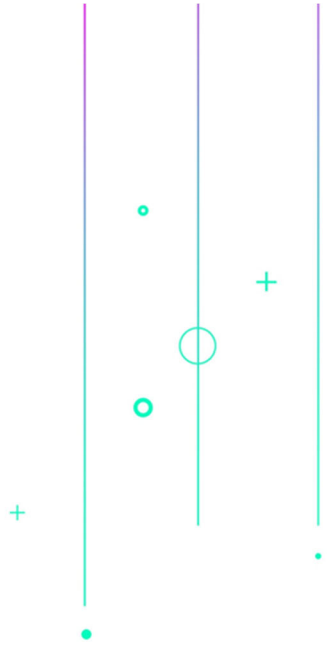


Fig 32. Flow Load File Successfully Loaded in PIM360

All Rights Reserved



Datum360
Connected data
Datum360 Limited / Fusion Hive / North Shore Road / Stockton on Tees / TS18 2NB / UK /
T: +44 3333 441 882 / E: info@datum360.com / W: www.datum360.com