

Powerautomate Example - Poll For Change then Email

Thursday, 5 May 2022 2:06 PM

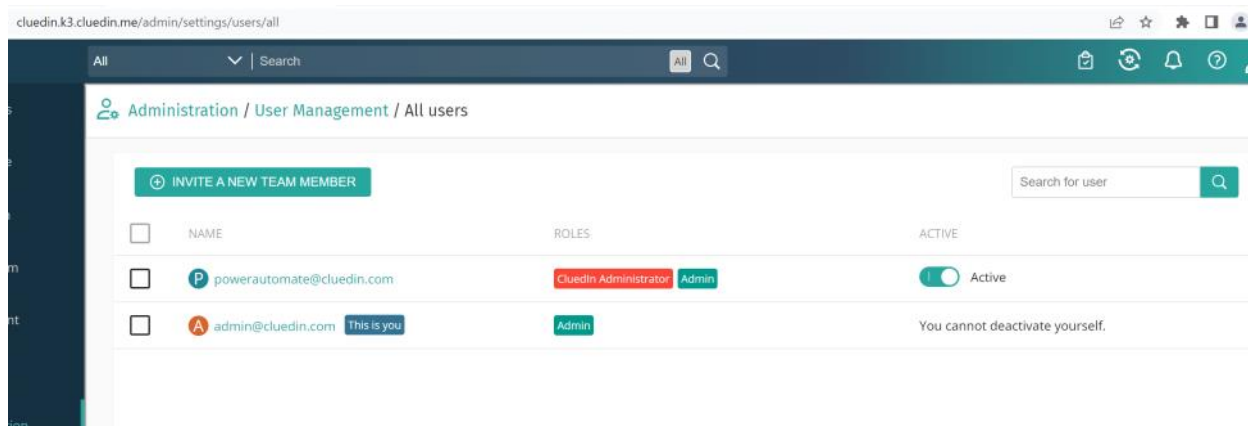
Purpose

Take the GraphQL calls worked as observed in CluedIn UI and turn it into powerautomate integration for workflows

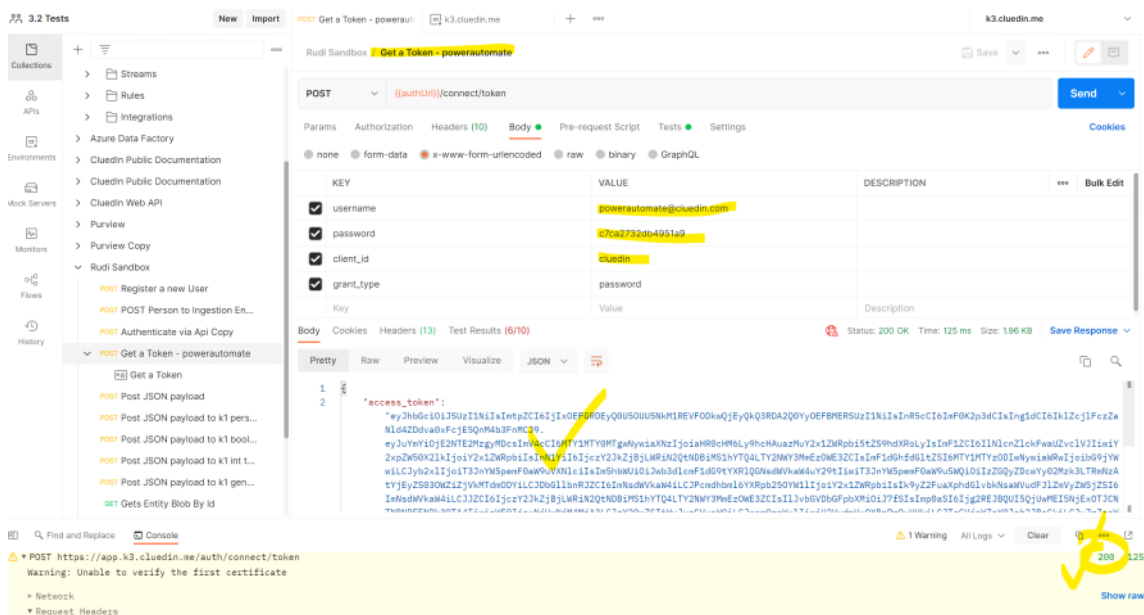
Method

Create the powerautomate@cluedin.com user as a service account

Let's update the user to admin level so we can make the GraphQL calls with it's token



Try a get token call based on the postman collection using the service account



Let's capture the GraphQL call for the duplicates

Note: duplicate report was already created

cluedin.k3.cluedin.me/admin/management/duplicate/deduplication-projects/detail/d72d3b57-3911-4c88-ea28-08da2cc03103/duplicates

All Search

Management / ... / All De-duplication projects / Same ID

Same ID

Configuration Duplicates

Name	Number of Duplicates	
person.id: 1	2	View Entities
person.id: 10	2	View Entities
person.id: 100	2	View Entities
person.id: 1000	2	View Entities
person.id: 101	2	View Entities
person.id: 102	2	View Entities

See [Capture graphql calls and replay in Postman.pdf](#) for more details

Query / Mutation

- notifications
- notifications
- administration
- getDeDupProject
- getDeDupList**

Headers Request Response Response (Raw)

```

query getDeDupList($entityType: String, $vocabularyKeys: [String], $ruleType: Int) {
  management {
    id
    possibleDuplicates(
      entityType: $entityType
      vocabularyKeys: $vocabularyKeys
      ruleType: $ruleType
    ) {
      name
      count
      values
      __typename
    }
    __typename
  }
}

{
  "entityType": "/Person",
  "vocabularyKeys": [
    "person.id"
  ],
  "ruleType": 1
}

```

Headers Request Response **Response (Raw)**

```

{
  "data": {
    "management": {
      "id": "management",
      "possibleDuplicates": [
        {
          "name": "{\\"person.id\\":\\"1\\"}",
          "count": 2,
          "values": {
            "person.id": "1"
          },
          "__typename": "PossibleDuplicate"
        },
        {
          "name": "{\\"person.id\\":\\"10\\"}",
          "count": 2,
          "values": {
            "person.id": "10"
          },
          "__typename": "PossibleDuplicate"
        }
      ]
    }
  }
}

```

View possible duplicate entities

 SELECT FOR MERGE



Estella Everson

No snippet available

Created a day ago
Last changed a day ago
Found a day ago





Estella

No snippet available

Created a day ago
Last changed a day ago
Found a day ago

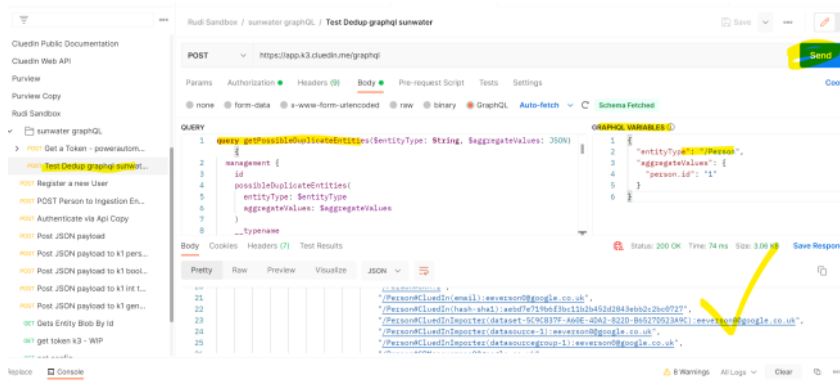


Query / Mutation	Headers	Request	Response	Response (Raw)
 getPossibleDuplicateEntities		<pre>query getPossibleDuplicateEntities(\$entityType: String, \$aggregateValues: JSON) { management { id possibleDuplicateEntities(entityType: \$entityType aggregateValues: \$aggregateValues) __typename } }</pre>		
 notifications				

```

{
  "data": {
    "management": {
      "id": "management",
      "possibleDuplicateEntities": {
        "TotalResults": 2,
        "NextCursor": null,
        "Entries": [
          {
            "entity": {
              "attribute-id": "38b382dd-2384-53c7-b40b-e0462f4f6974",
              "attribute-organization": "3dd2d70c-6397-4f70-b12e-79fbf5d17f86",
              "processedData": {
                "entityType": "/Person",
                "name": "Estella Everson",
                "aliases": [
                  "Estella Everson"
                ],
                "codes": [
                  "/Person#CRM:1",
                  "/Person#CluedIn(email):eeverson@google.co.uk",
                  "/Person#CluedIn(hash-sha1):aebd7e719b6f3bc11b2b452d2043ebb2c2bc0727",
                  "/Person#CluedInImporter(dataset-5C9C837F-A60E-4DA2-822D-B6527D523A9C):eeverson@google.co.uk",
                  "/Person#CluedInImporter(datasource-1):eeverson@google.co.uk",
                  "/Person#CluedInImporter(datasourcegroup-1):eeverson@google.co.uk",
                  "/Person#CRM:eeverson@google.co.uk",
                  "/Person#File Data Source:eeverson@google.co.uk"
                ]
              },
              "provider-origins": [
                "CRM",

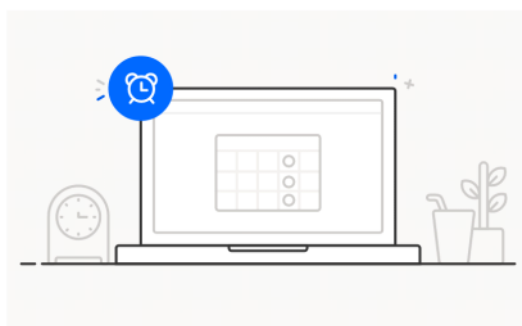
```



go to powerautomate

create new one flow to poll

Build a scheduled cloud flow



Stay on top of what's important without the effort—you choose

Flow name

check for duplicate records example

Run this flow *

Starting

5/31/22

at

10:00 AM

Repeat every

1

Month

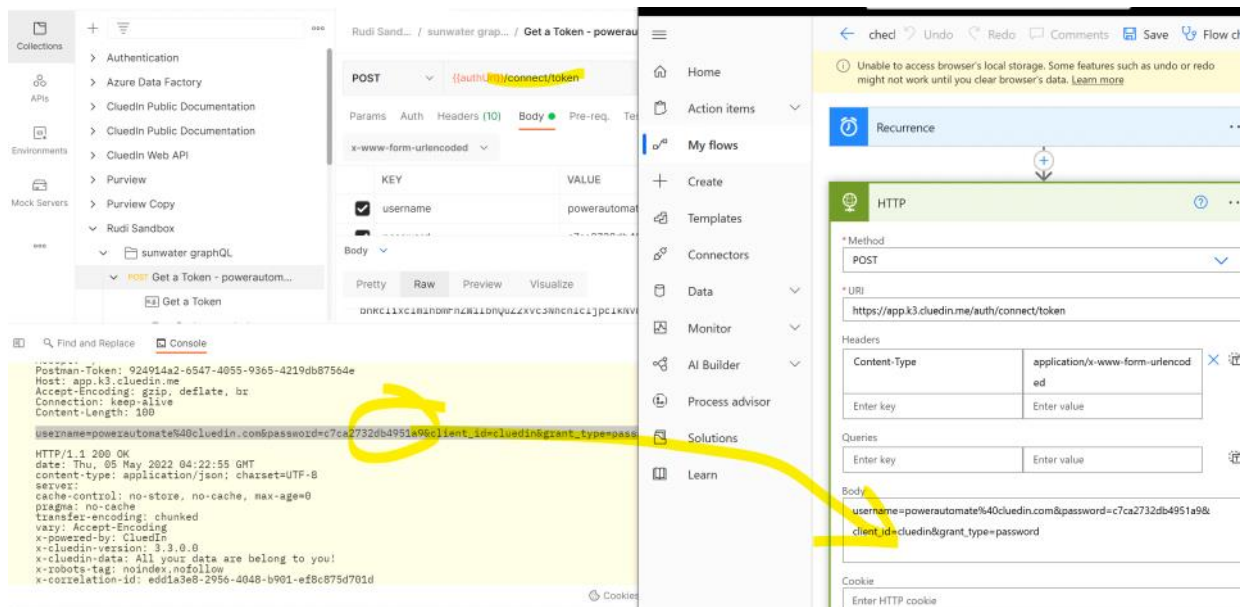
This flow will run:

Every month

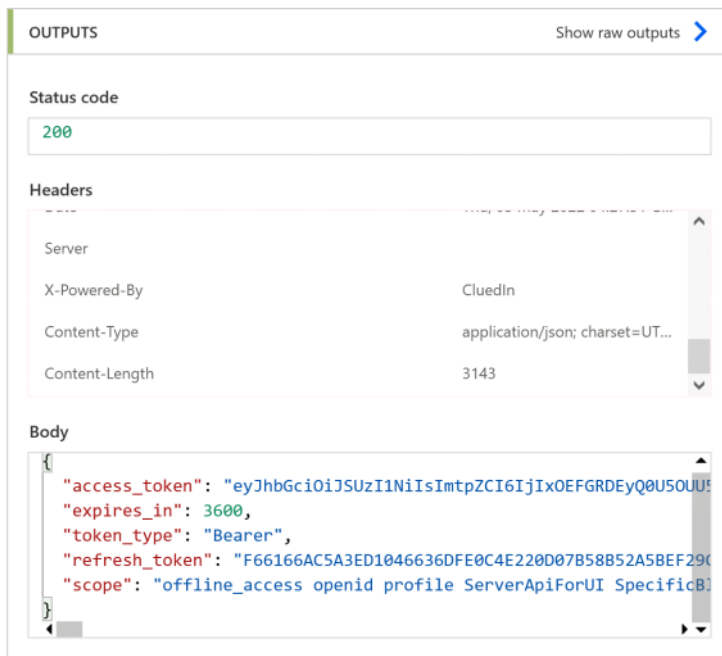
start building the flow

use the /connect/token endpoint to login to CluedIn as a service account

use postman to convert to the name/value pairs to the url post format as shown



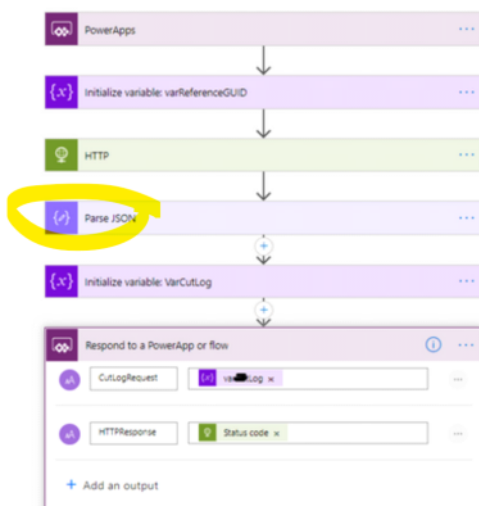
run the flow for the first get token call... success!



how do we refer to these returned json?

<https://powerusers.microsoft.com/t5/Building-Power-Apps/Formatting-a-JSON-response-from-Power-Automate-flow-on-PowerApps/td-p/907563>

3. Send back to Powerapp a string with the JSON text



then we refer to the parsed json variables in later actions

create http graphql

The screenshot shows the Power Automate interface for a GraphQL test action. The test is named "Test Dedup graphql sunwat...". The URL is "https://app.k3.cluedin.me/graphql". The method is "POST". The body is a JSON object with a query and variables. The console shows the raw response, which is a JSON object with "data" and "errors" fields. A yellow circle highlights the "data" field in the console output.

```
POST https://app.k3.cluedin.me/graphql

{
  "query": "query getPossibleDuplicateEntities($entityType: String, $aggregateValues: JSON!) {\n  management {\n    possibleDuplicateEntities(entityType: $entityType, aggregateValues: $aggregateValues) {\n      __typename\n      person.id\n    }\n  }\n}"
}
```

```
{
  "data": {
    "management": {
      "possibleDuplicateEntities": [
        {
          "__typename": "Person",
          "person.id": "1"
        }
      ]
    }
  }
}
```

test - fail

The screenshot shows the "HTTP - GraphQL - Get Dup List" test action. The method is "POST". The URI is "https://app.k3.cluedin.me/graphql". The body is a JSON object with a query and variables. The headers include "Bearer" token and "Content-Type: application/json".

```
POST https://app.k3.cluedin.me/graphql

{
  "query": "query getPossibleDuplicateEntities($entityType: String, $aggregateValues: JSON!) {\n  management {\n    possibleDuplicateEntities(entityType: $entityType, aggregateValues: $aggregateValues) {\n      __typename\n      person.id\n    }\n  }\n}"
}
```

```
{
  "Bearer": "eyJhbGciOiJIUzI1NiIsImtpZCI6IjIwOEFGRDEyQ0U5OUU5NkM1RE...",
  "Content-Type": "application/json"
}
```

```
{
  "errors": [
    {
      "message": "401: Unauthorized",
      "locations": [
        {
          "line": 4,
          "column": 2
        }
      ],
      "path": [
        "management",
        "possibleDuplicateEntities"
      ],
      "extensions": {
        "code": "UNAUTHENTICATED",
        "response": {
          "url": "http://cluedin-server:9000/api/duplicates/potential-dynamic-duplicates-entities",

```

```

        "status": 401,
        "statusText": "Unauthorized",
        "body": ""
    }
  },
  ],
  "data": {
    "management": {
      "id": "management",
      "possibleDuplicateEntities": null,
      "__typename": "Management"
    }
  }
}

```

mmmm, re-run then trace through server logs

```
[06:17:05 ERR] Credentials validated for username: powerautomate@cluedin.com in organization cluedin
```

mmm, I'm using the wrong auth header format, fixing:

URI	https://app.k3.cluedin.me/graphql	
Headers	Authorization	Bearer access_token x
	Content-Type	application/json

re-test - success!!

OUTPUTS
[Show raw outputs](#)

Status code
200

Headers

Key	Value
access-control-allow-origin	*
keep-alive	timeout=5
strict-transport-security	max-age=15768000

Body

```

{
  "data": {
    "management": {
      "id": "management",
      "possibleDuplicateEntities": {
        "TotalResults": 2,
        "NextCursor": null,
        "Entries": [

```

```

{
  "data": {
    "management": {
      "id": "management",
      "possibleDuplicateEntities": {
        "TotalResults": 2,
        "NextCursor": null,
        "Entries": [
          {
            "entity": {
              "attribute-id": "38b382dd-2384-53c7-b40b-e0462f4f6974",
              "attribute-organization": "3dd2d70c-6397-4f70-b12e-79fbf5d17f86",
              "processedData": {
                "entityType": "/Person",
                "name": "Estella Everson",
                "aliases": [
                  "Estella Everson"
                ],
                "codes": [
                  "/Person#CRM:1",
                  "/Person#CluedIn(email):eeverson0@google.co.uk",
                  "/Person#CluedIn(hash-sha1):aebd7e719b6f3bc11b2b452d2043ebb2c2bc0727",
                  "/Person#CluedInImporter(dataset-5C9C837F-A60E-4DA2-822D-B6527D523A9C):eeverson0@google.co.uk",
                  "/Person#CluedInImporter(datasource-1):eeverson0@google.co.uk",
                  "/Person#CluedInImporter(datasourcegroup-1):eeverson0@google.co.uk",
                  "/Person#CRM:eeverson0@google.co.uk",
                  "/Person#File Data Source:eeverson0@google.co.uk"
                ],
                "provider-origins": [
                  "CRM",

```

```

        "CluedIn",
        "CluedInImporter",
        "File Data Source"
    ],
    "createdDate": "2022-05-03T04:43:50.0000000+00:00",
    "modifiedDate": "2022-05-03T04:43:50.0000000+00:00",
    "discoveryDate": "2022-05-03T04:43:58.8650047+00:00",
    "properties": {
        "attribute-type": "/Metadata/KeyValue",
        "property-person.city": "Potosí",
        "property-person.id": "1",
        "property-person.street": "8 Di Loreto Place",
        "property-user.email": "eeverson0@google.co.uk",
        "property-user.firstName": "Estella",
        "property-user.gender": "Male",
        "property-user.lastName": "Everson"
    },
    "propertyGroups": [
        {
            "name": "Metadata",
            "source": "CluedIn",
            "keys": [
                "person.street",
                "person.id",
                "person.city"
            ]
        },
        {
            "name": "Metadata",
            "source": "CluedIn",
            "keys": [
                "user.firstName",
                "user.lastName",
                "user.gender",
                "user.email"
            ]
        }
    ],
    "attribute-origin": "/Person#CRM:1",
    "sortDate": "2022-05-03T04:43:50.0000000+00:00",
    "timeToLive": "0",
    "isShadowEntity": "False",
    "dataDescription": {
        "dataClasses": [
            "/Email",
            "/Person/FirstName",
            "/Person/LastName",
            "/Person/Gender"
        ]
    }
},
{
    "entity": {
        "attribute-id": "ccae4835-ebaf-5069-b9df-7cd48671e6f1",
        "attribute-organization": "3dd2d70c-6397-4f70-b12e-79fbf5d17f86",
        "processedData": {
            "entityType": "/Person",
            "name": "Estella",
            "codes": [
                "/Person#ERP:eeverson0@google.co.uk",
                "/Person#CluedIn(hash-sha1):563618fdd85ed711c0f5d73aa305d074421af404"
            ],
            "provider-origins": [
                "ERP",
                "CluedIn"
            ],
            "createdDate": "2022-05-03T04:47:51.0000000+00:00",
            "modifiedDate": "2022-05-03T04:47:51.0000000+00:00",
            "discoveryDate": "2022-05-03T04:47:51.0000000+00:00",
            "properties": {
                "attribute-type": "/Metadata/KeyValue",
                "property-person.city": "Potosí",
                "property-person.emailaddress": "eeverson0@google.co.uk",
                "property-person.id": "1",
                "property-person.last": "Everson",
                "property-person.name": "Estella"
            },
            "propertyGroups": [
                {
                    "name": "Metadata",
                    "source": "CluedIn",
                    "keys": [
                        "person.last",
                        "person.id",
                        "person.emailaddress",
                        "person.city",
                        "person.name"
                    ]
                }
            ],
            "isSensitiveInformation": "False",
            "attribute-origin": "/Person#ERP:eeverson0@google.co.uk",
            "sortDate": "2022-05-03T04:47:51.0000000+00:00",
            "timeToLive": "0",

```



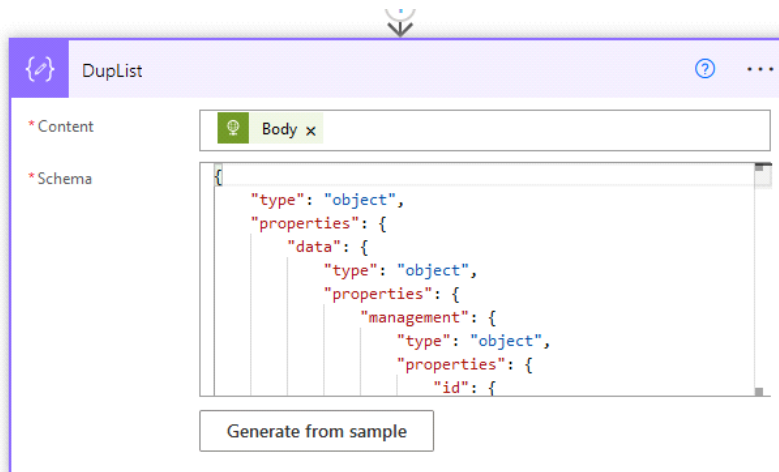
```

        "isShadowEntity": "False",
        "dataDescription": {
          "dataClasses": [
            "/Email",
            "/"
          ]
        }
      }
    ]
  },
  "_typename": "Management"
}
}
}

```

let's try parsing it

add a new parse json action, format results into a string, email it!

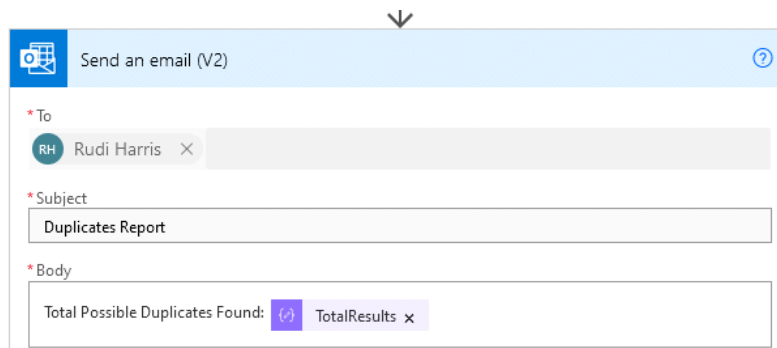


The screenshot shows a workflow step named "DupList". It has two main fields: "Content" and "Schema". The "Content" field is set to "Body". The "Schema" field contains a JSON structure that defines the data types and properties for the workflow. Below the "Schema" field is a button labeled "Generate from sample".

```

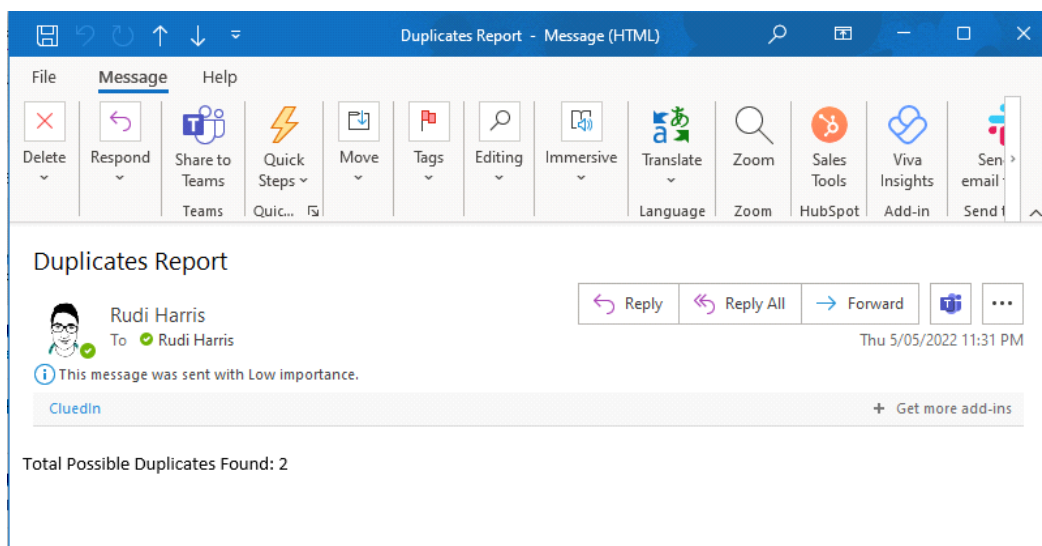
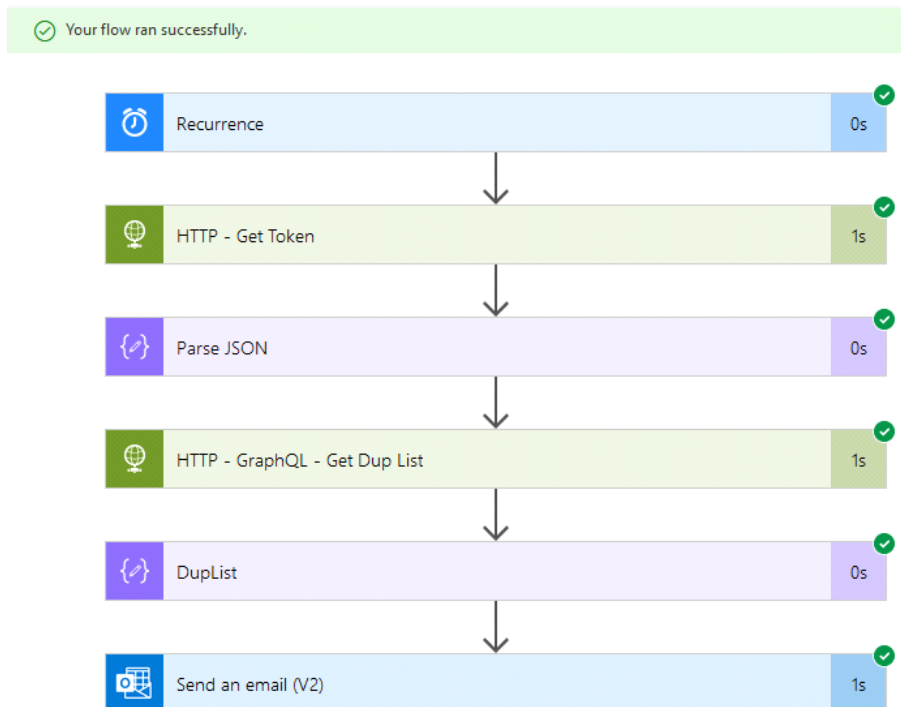
{
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "management": {
          "type": "object",
          "properties": {
            "id": {

```



The screenshot shows a workflow step named "Send an email (V2)". It has three main fields: "To", "Subject", and "Body". The "To" field is set to "Rudi Harris". The "Subject" field is set to "Duplicates Report". The "Body" field contains a placeholder for the JSON results, showing "Total Possible Duplicates Found:" followed by a JSON icon and "TotalResults".

test it



further example

a much better email would be if the JSON was parsed and turned into html for the email output (say by an Azure Function)

