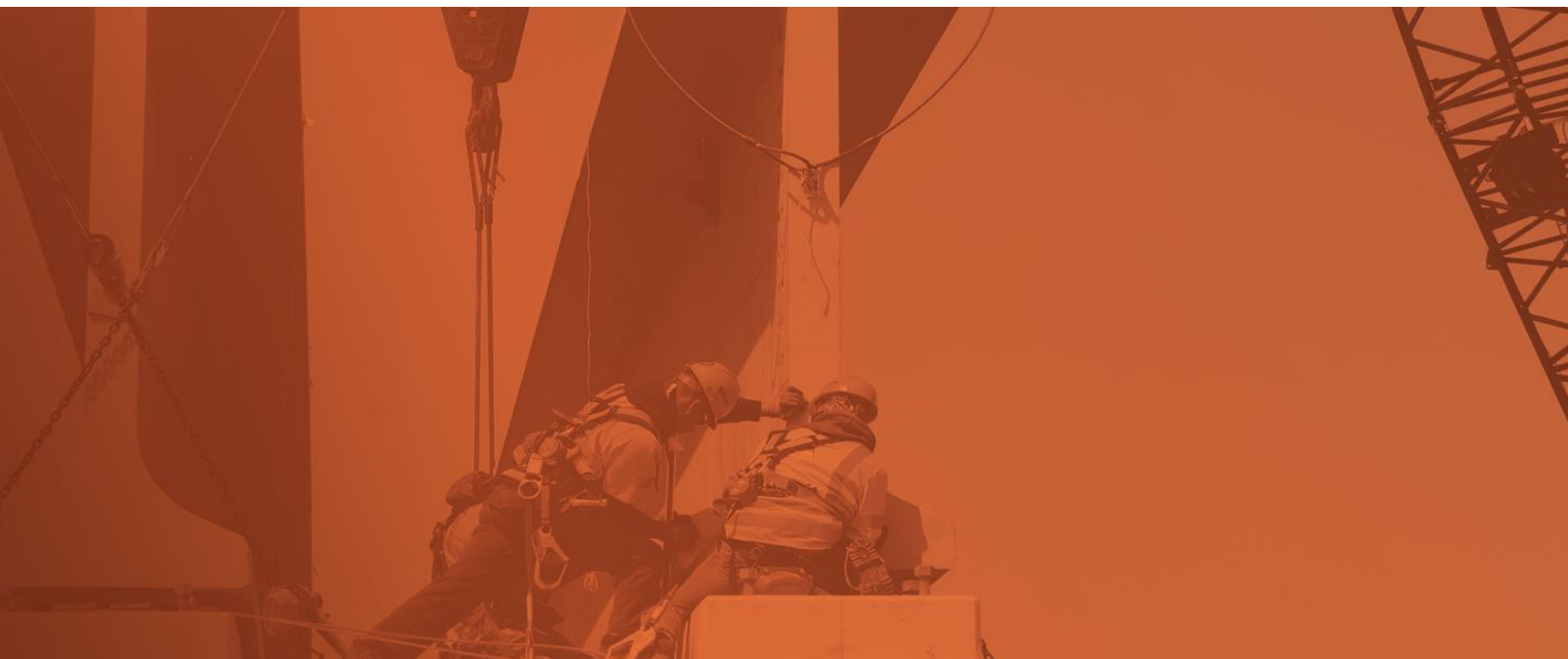




Document Reporting APIs – Power BI

Reporting APIs User Guide



InEight®
DOCUMENT

Changelog

This changelog contains only significant or other notable changes to the document revision. Editorial or minor changes that do not affect the context of the document are not included in the changelog.

Revision	Change Date	Description	Author
0		Original Draft	Andy Toraman
1	2/08/2021	Update to Add Portfolio Reporting Process	Richard Milligan
2	30/08/2021	Correct Example on page 17, syntax errors	Richard Milligan
3	01/02/2023	Update to Portfolio Reporting Process	Andy Toraman
4	21/08/2023	Update to EUI screenshots	Jo Vittiglia
5	12/03/2025	Update with informational notes	Andy Toraman
6	07/04/2025	Minor update to the informational notes	Johannes Sidarta

Important Notice to All InEight Customers

Please be advised that this document is provided as a guideline only and is offered "as-is."

Contents

Introduction.....	4
Using the InEight Document Reporting API Service.....	4
InEight DOCUMENT Reporting Function	5
Report Types.....	5
The Reporting APIs.....	7
Using the Reporting APIs in Swagger.....	7
Making API calls Step by Step	8
Ending the Session.....	13
Connecting Reporting APIs with Power BI.....	14
Project View Reporting Instructions	14
Portfolio View across all Projects.....	21

Introduction

The purpose of this document is to explain how to use InEight Document Reporting APIs in a step-by-step fashion and allow third party organizations to easily interface applications they use with InEight Document Reporting module.

Examples in this document are for demonstration purposes only.

It is essential that any third-party organization using the APIs first signs an agreement to keep this document confidential and what it can achieve. Furthermore, it is important that the APIs are only used for the purpose it is provided and that they are not misused to make spurious calls on the InEight Document Reporting API.

Using the InEight Document Reporting API Service

InEight Document always operates based on a single user's security settings as configured by the project administrators. The InEight Document Reporting APIs are no different. All InEight Document Reporting API functions require an active user session identifier to be passed in order to authorize any activity and will be limited by the user's access limitations.

This active user session identifier is known as the “**sessionKey**” which is returned when a user logs in using the following.

- UserID
- CompanyID
- Password
- Project Number

IMPORTANT: The sessionKey should be retained and re-used for any immediate API calls and then to have the session invalidated at the end of the application/process cycle.

The session initiate method is the only exception which does **not** require a session key authorization. See section 4.1.1 on how to authorize the InEight Document API requests with the session key.

InEight DOCUMENT Reporting Function

Report Types

To understand what the Reporting APIs will return the users need to be acquainted with the Reporting function in the DOCUMENT application. This is where all the available report types can be seen, and the APIs will work based on each individual report type.

Step 1: Open the Reports Module

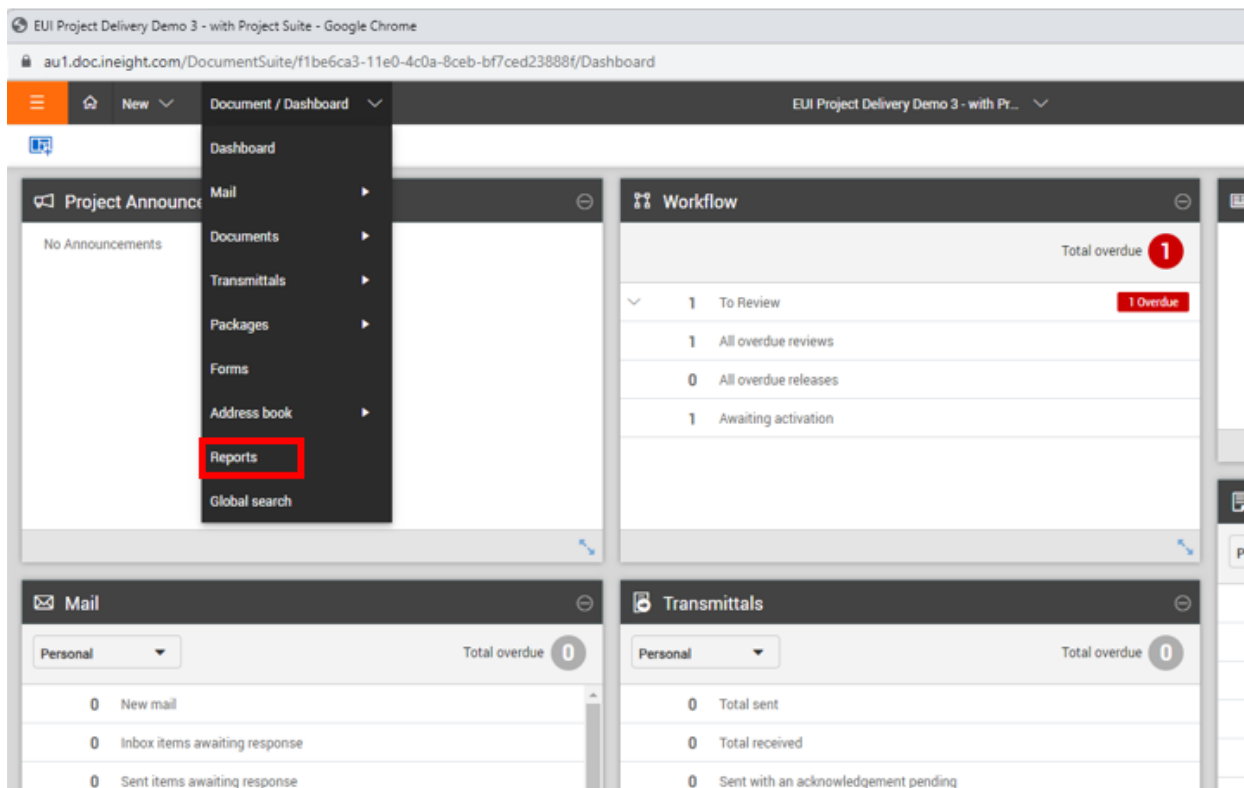


Figure 1: Click on “Reports” from DOCUMENT dropdown menu

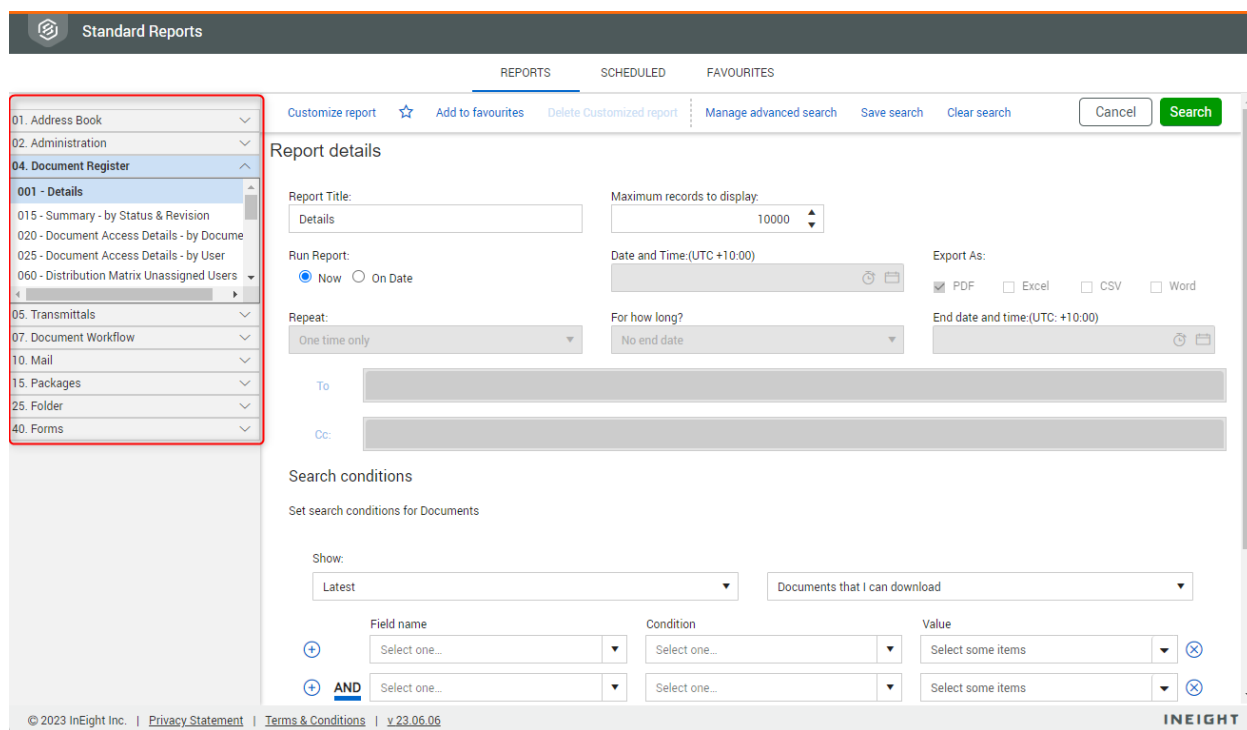
Step 2: Check Out the Standard Reports


Figure 2: Navigation Pane to run the Standard Reports

The available Report types are listed in the left-hand pane and note that the reports are categorised as Address Book, Administration, Document Register, Transmittals and so on. Under each category there are multiple different report types which users can run by selecting the filters and clicking on the “Search” button.

The format an individual report is numbered is XX.YYY where XX is the category number and YYY is the report identifier.

Some of the report types can be seen below.

Report No	Title
04.001	Document Register by Details
05.001	Transmittal Details by Transmittal No
40.950	Forms RFI Log
40.001	Forms Summary

For detailed information on the different report types please go to [REPORTS](#).

The Reporting APIs

Using the Reporting APIs in Swagger

Swagger is in essence a set of open-source tools to develop, document and use RESTful web services also known as RESTful APIs. The tool will serve the purpose of making the Reporting API calls which will return responses in a human-readable text format -JSON.

What is required?

- **URL to access Swagger:** <https://server/TBReportingAPI/>
 - The user needs to know which server the instance of DOC is hosted in.
 - This can be acquired from URL bar when logged in the DOCUMENT application.

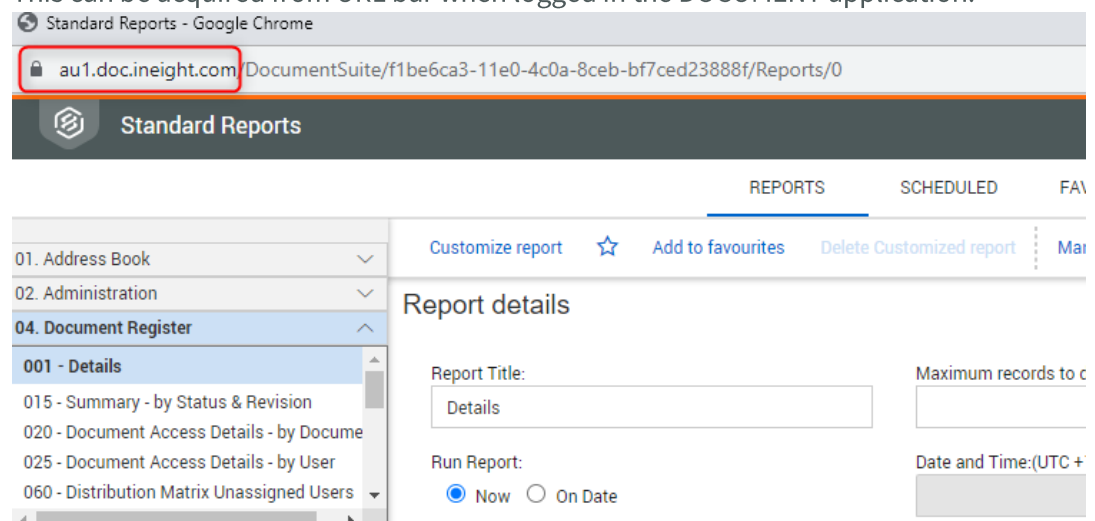


Figure 3: In this example, the URL will be <https://au1.doc.ineight.com/TBReportingAPI/>

- **UserID**
- **CompanyID**
- **Password**
- **ProjectNo**
 - It needs to be ensured that the Project is enabled for the Reporting APIs otherwise the API will return “invalid project id”. The system admin should be contacted when this error message pops up.

Making API calls Step by Step

Step 1: Open Swagger and click on Session

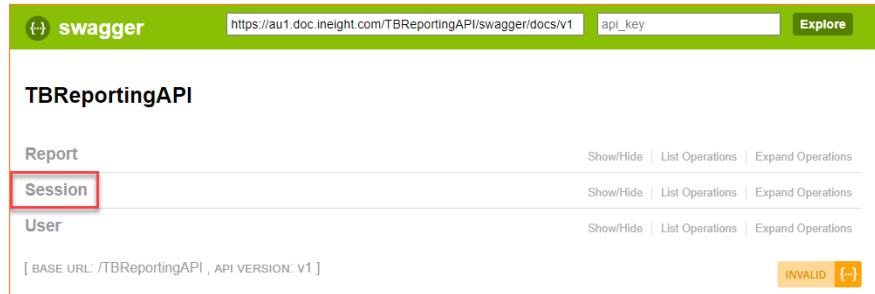


Figure 4: Swagger User Web Interface

Step 2: Click on POST /Session



Figure 5: Post /Session API Request to **log the user in** and generate the **sessionKey**.

Step 3: Click on JSON Code Example Value

TBReportingAPI

Report

Show/Hide | List Operations | Expand Operations

Session

Show/Hide | List Operations | Expand Operations

DELETE /Session /Session

POST /Session Session

Implementation Notes
Create a InEight Document session for use with the InEight Document Reporting API (Login)
Response Class (Status 200)
OK
Model | **Example Value**

{ }

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
userInfo	(required) Parameter content type: application/json Try it out!	The API access enabled user's InEight Document login information. oneTimePassword parameter is optional unless two-factor authentication enabled for user.	body	Model Example Value { "UserID": "string", "CompanyID": "string", "Password": "string", "ProjectNo": "string", "oneTimePassword": "string", "Application": "string" }

User

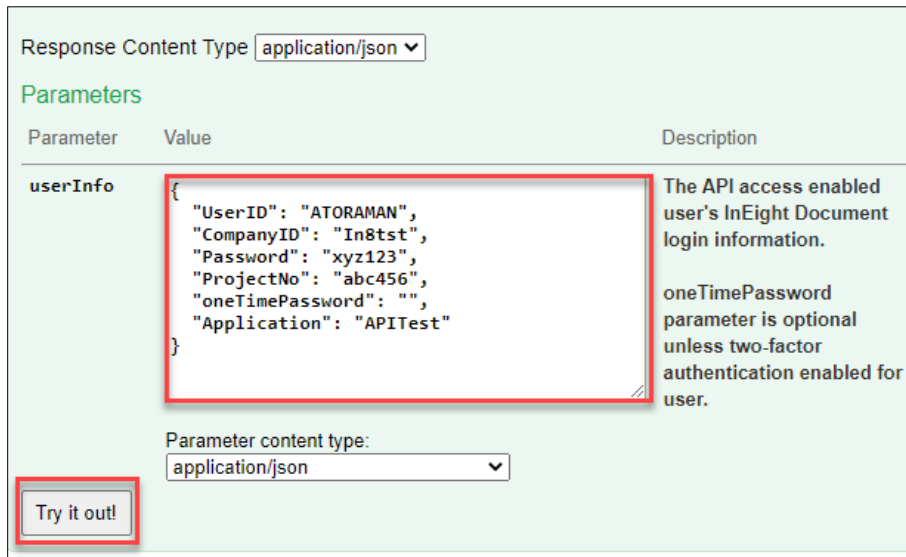
Show/Hide | List Operations | Expand Operations

[BASE URL: /TBReportingAPI , API VERSION: v1]

INVALID { }

Figure 6: Once clicked, the code template will be brought across into the blank userInfo field.

Step 4: Fill in the required fields and click on “Try it Out” button.



Response Content Type application/json

Parameters

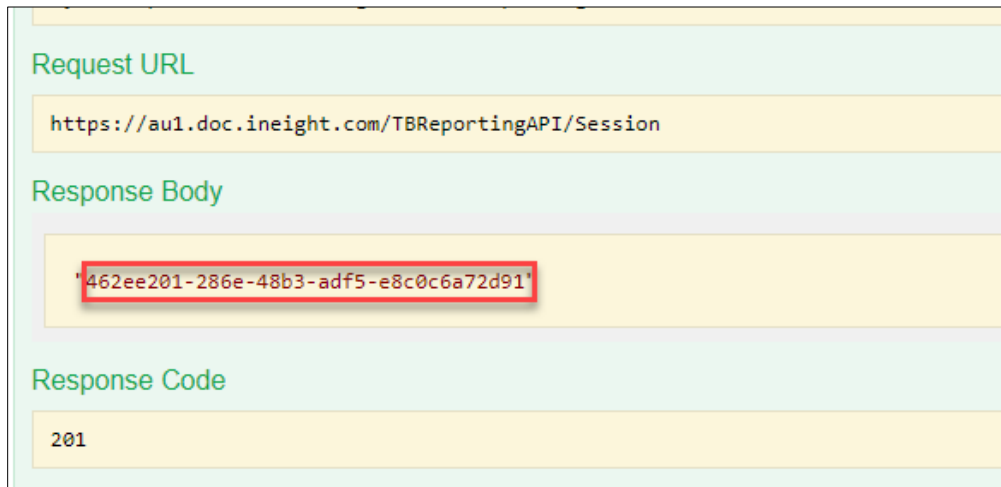
Parameter	Value	Description
userInfo	<pre>{ "UserID": "ATORAMAN", "CompanyID": "In8tst", "Password": "xyz123", "ProjectNo": "abc456", "oneTimePassword": "", "Application": "APITest" }</pre>	The API access enabled user's InEight Document login information. oneTimePassword parameter is optional unless two-factor authentication enabled for user.

Parameter content type: application/json

Try it out!

Figure 7: The users put in their credentials and specify the Project.

Step 5: Copy the sessionKey



Request URL

`https://au1.doc.ineight.com/TBReportingAPI/Session`

Response Body

`"462ee201-286e-48b3-adf5-e8c0c6a72d91"`

Response Code

`201`

Figure 8: The response code 201 is for the successful API calls.

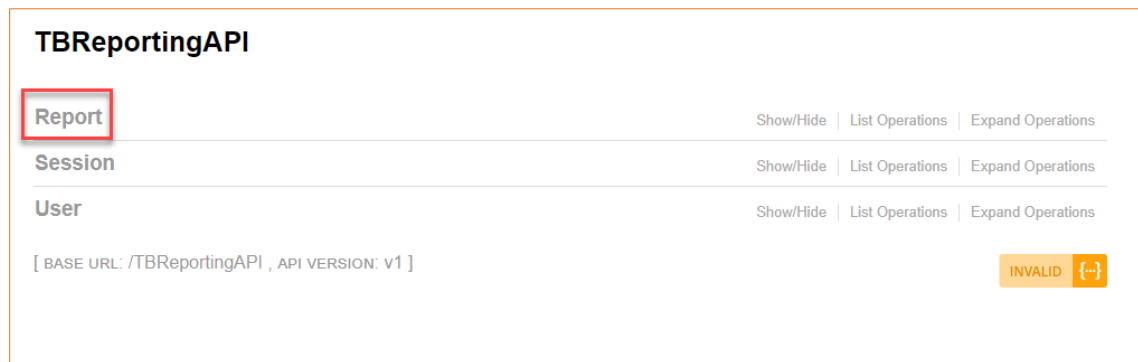
Step 6: Click on Report

Figure 9: Report is where all the Reporting APIs can be accessed.

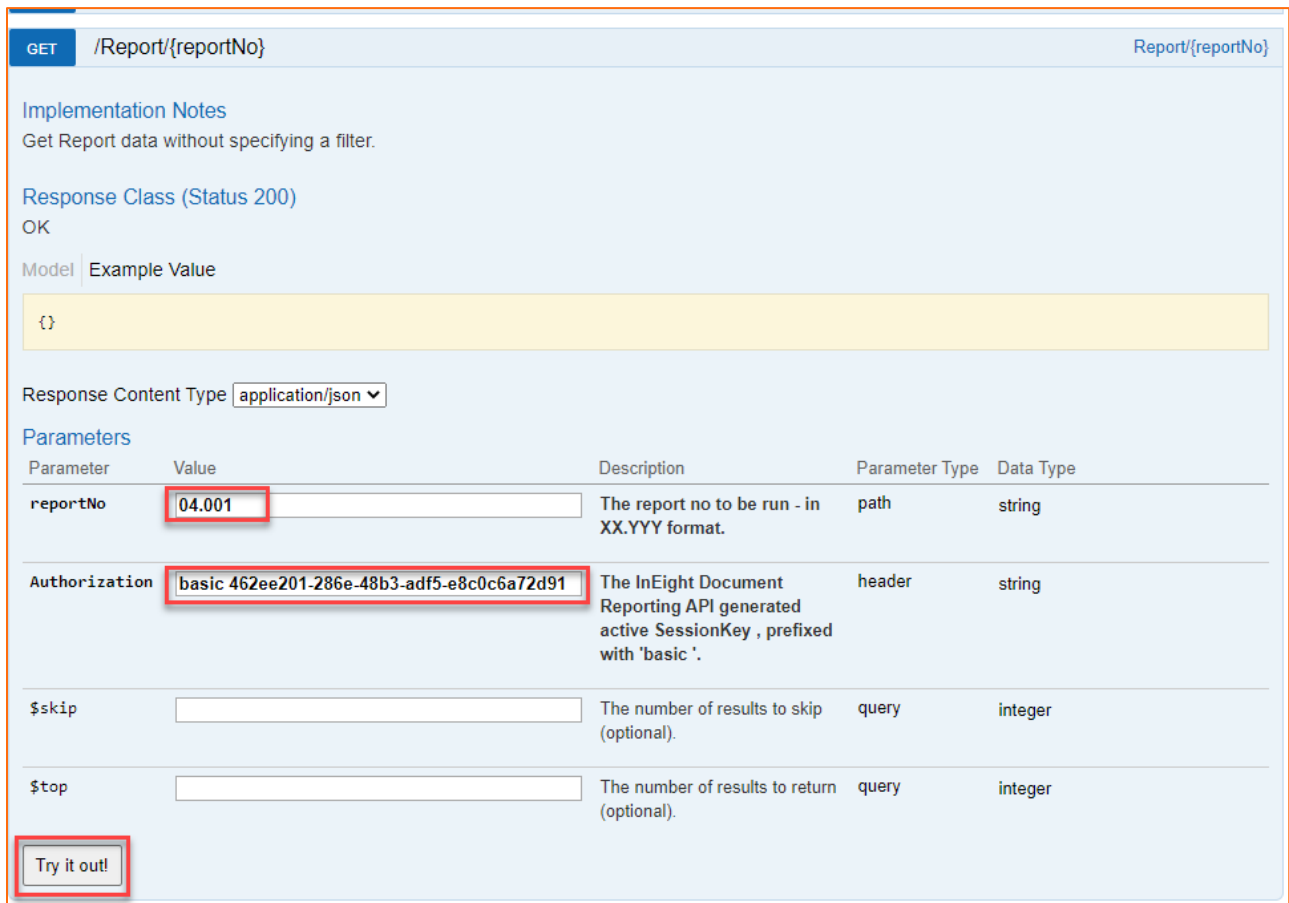
Step 7: Click on the API GET /Report/{reportNo}

Figure 10: This API returns the Report Data based on the Report Type **without** any Filtering.

Step 8: Specify the Report Type

- As can be seen from the Report function in the DOCUMENT web application, the report number is required to be put in “reportNo” field. Refer 3.1 in this document.

Step 9: Put in the sessionKey into the Authorization field preceded by “basic” and a space.

Step 10: Click on the “Try it Out”


GET /Report/{reportNo} Report/{reportNo}

Implementation Notes
Get Report data without specifying a filter.

Response Class (Status 200)
OK

Model | Example Value

```
{}
```

Response Content Type:

Parameters

Parameter	Value	Description	Parameter Type	Data Type
reportNo	04.001	The report no to be run - in XX.YYY format.	path	string
Authorization	basic 462ee201-286e-48b3-adf5-e8c0c6a72d91	The InEight Document Reporting API generated active SessionKey , prefixed with 'basic '.	header	string
\$skip		The number of results to skip (optional).	query	integer
\$top		The number of results to return (optional).	query	integer

Try it out!

Figure 11: reportNo and Authorization are the required fields.

NOTE: **\$skip** and **\$top** are the optional fields that allows for pulling in a specified number of results out of the whole Report data.

- The following example will return the top 10 records.
 - \$skip** = 0
 - \$top** = 10

Step 11: Check out the result

- It should be noted that the Response is in JSON format.
- The successful API call will return 200 as the response code.

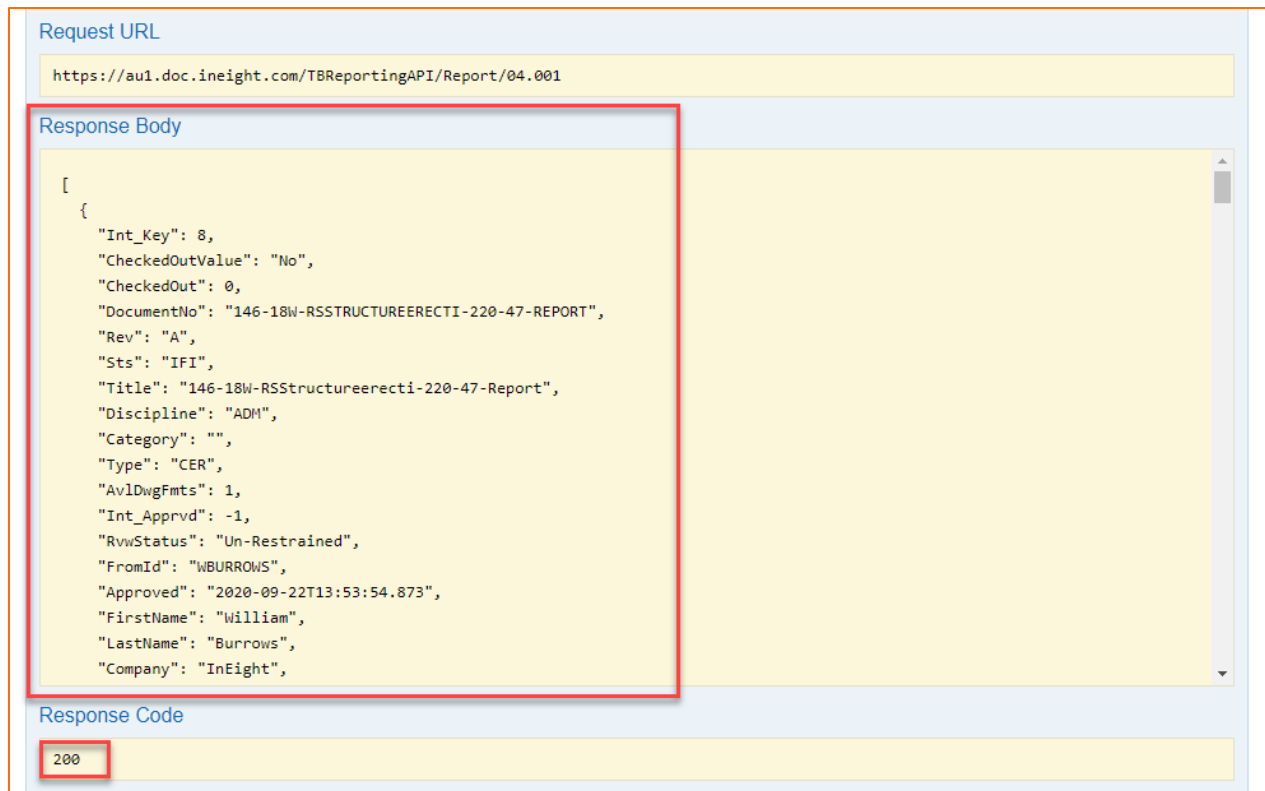


Figure 12: When the required fields are not correct, the Response Code will be 500 and an error message will pop up.

Ending the Session

It is critical that once the users are done with using the APIs, it is recommended that the active session is ended. The API type that will be used to log the user off is “DELETE”.

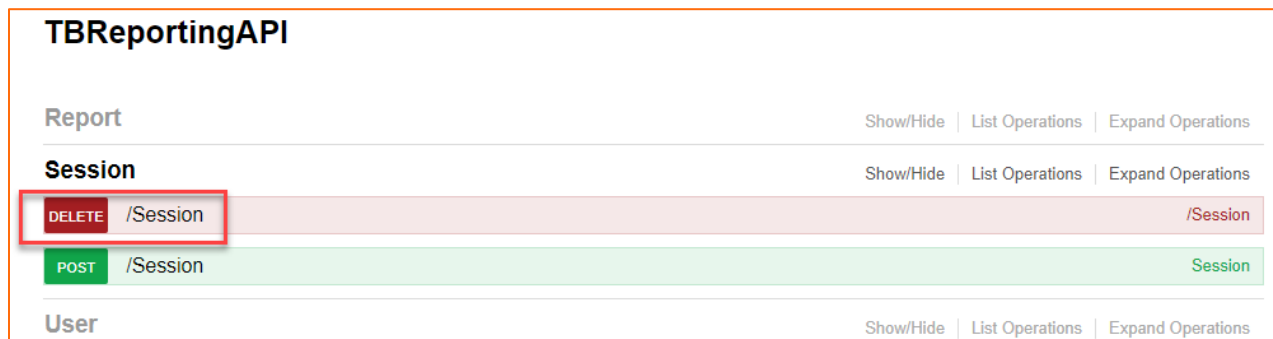


Figure 13: Delete can be found under Session.

Step 1: Click on Delete

Step 2: Paste the sessionKey in the Authorization field preceded by “basic” and a space.

Step 3: Click on “Try it Out”

Session Show/Hide List Operations Expand Operations

DELETE /Session /Session

Implementation Notes
Delete the session (Logoff)

Response Class (Status 200)
OK

Model **Example Value**

```
{}
```

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
Authorization	basic 1cb22815-53af-4984-aa99-401cc12545ad	The InEight Document Reporting API generated active SessionKey , prefixed with 'basic '.	header	string

Try it out! [Hide Response](#)

Figure 14: Delete method log the user off and make the sessionKey invalid.

Step 4: Check out the Response Code

Curl

```
curl -X DELETE --header 'Accept: application/json' --header 'Authorization: basic 1cb22815-53af-4984-aa99-401cc12545ad' 'https://a
```

Request URL

```
https://au1.doc.ineight.com/TBReportingAPI/Session
```

Response Body

```
no content
```

Response Code

```
200
```

Figure 15: The successful Delete request will return 200 as the response code.

Connecting Reporting APIs with Power BI

Project View Reporting Instructions

Power BI is in essence a tool to connect and visualize data and the following steps will explain how to bring the REPORT data from InEight DOCUMENT application into Power BI.

Step 1: Launch Power BI Desktop.

Step 2: Go to the Home tab and click on Get Data dropdown and select Blank Query.

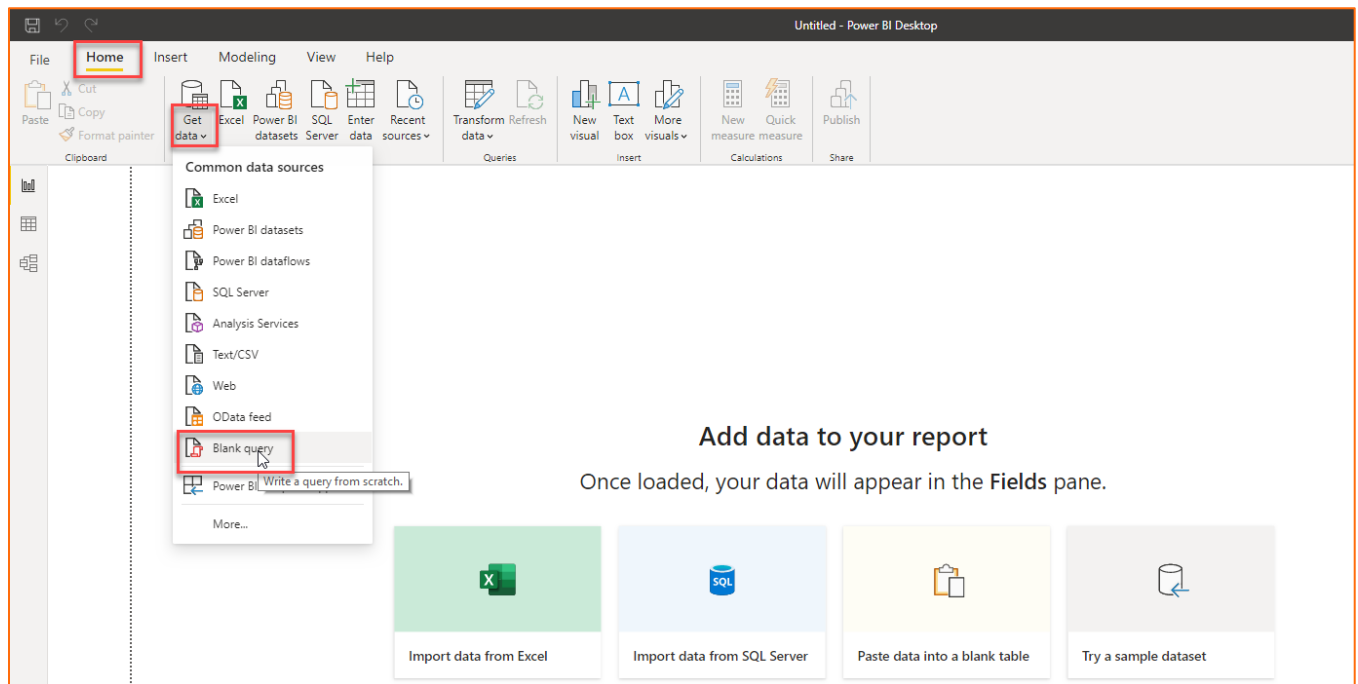


Figure 16: The query editor will pop up

Step 3: Go to Manage Parameters under the Home Tab and click on “New Parametres”

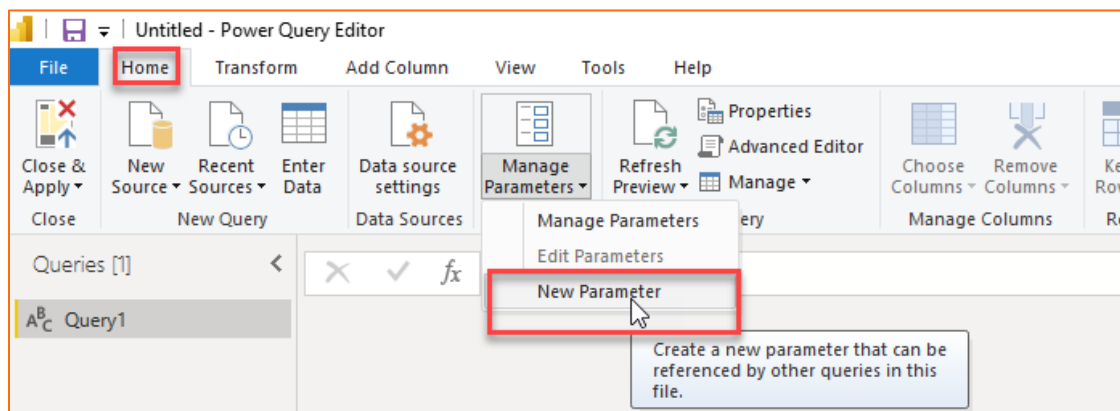


Figure 17: The same URL that allowed for accessing Swagger will be defined as the parameter

Step 4: Type “APIURL” in the Name field.

Step 5: Paste the url in the Current Value field.

- The url will be <https://server/TBReportingAPI/>
- It can be acquired once logged in the DOCUMENT web application.

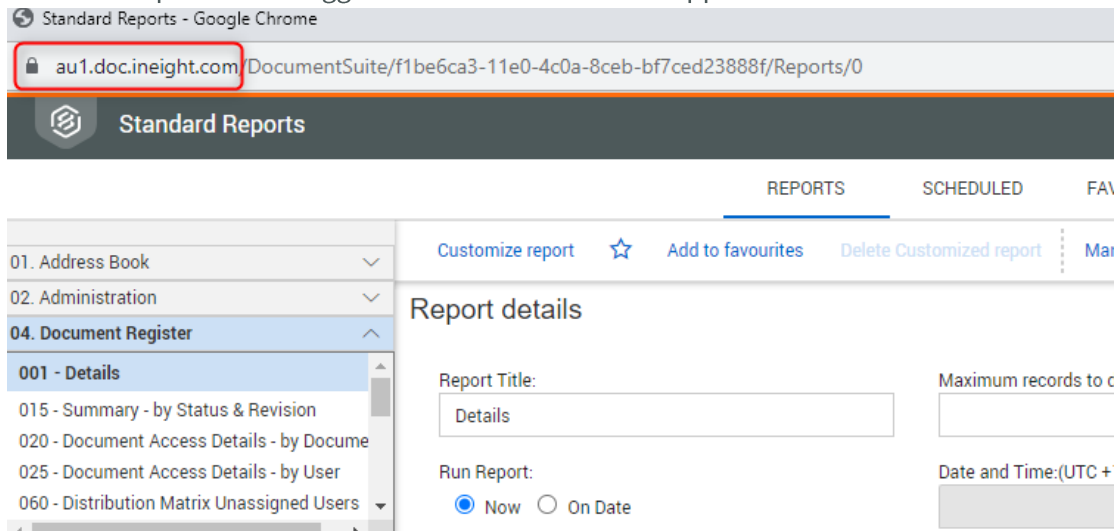


Figure 18: In this example, the URL will be <https://au1.doc.ineight.com/TBReportingAPI/>

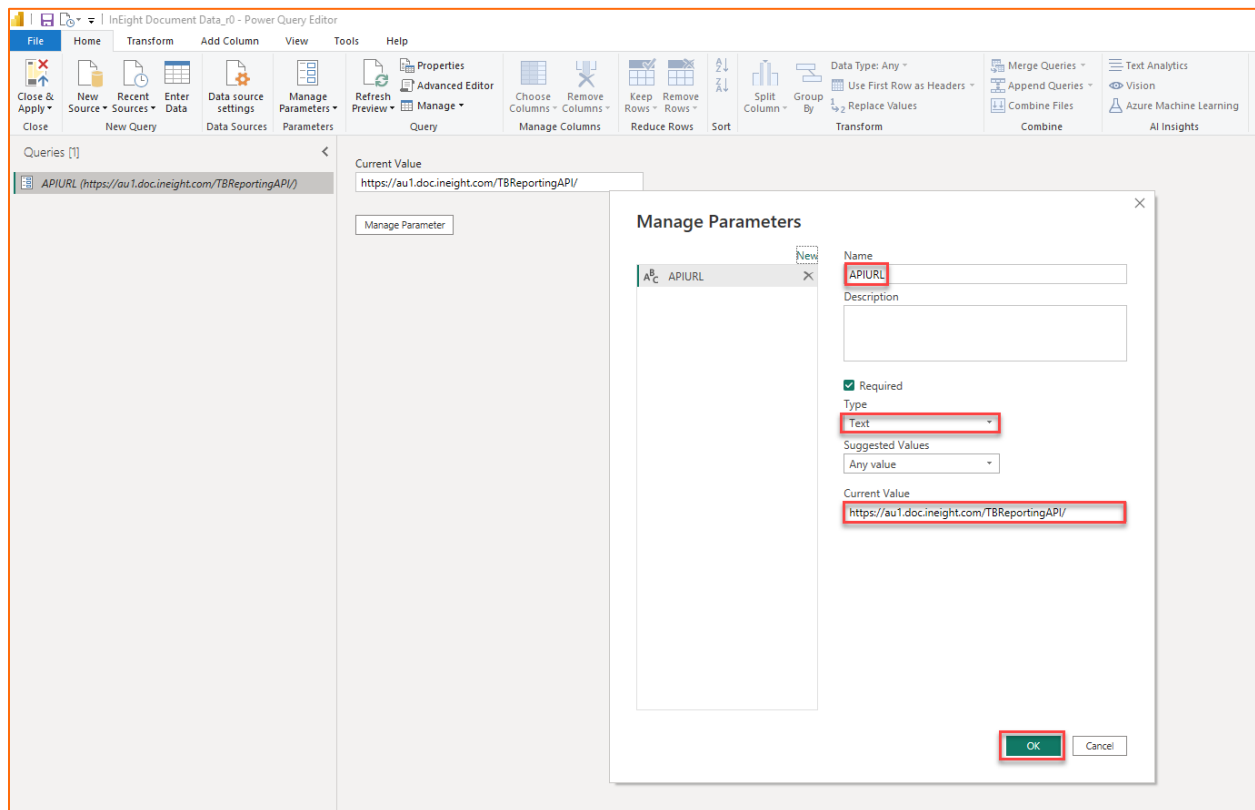
Step 6: Click OK.

Figure 19: This will define the URL as the parameter to be used in the query

Step 7: Click on New to add the following parameters and then click OK

- UserId
- Company
- Password
- Project

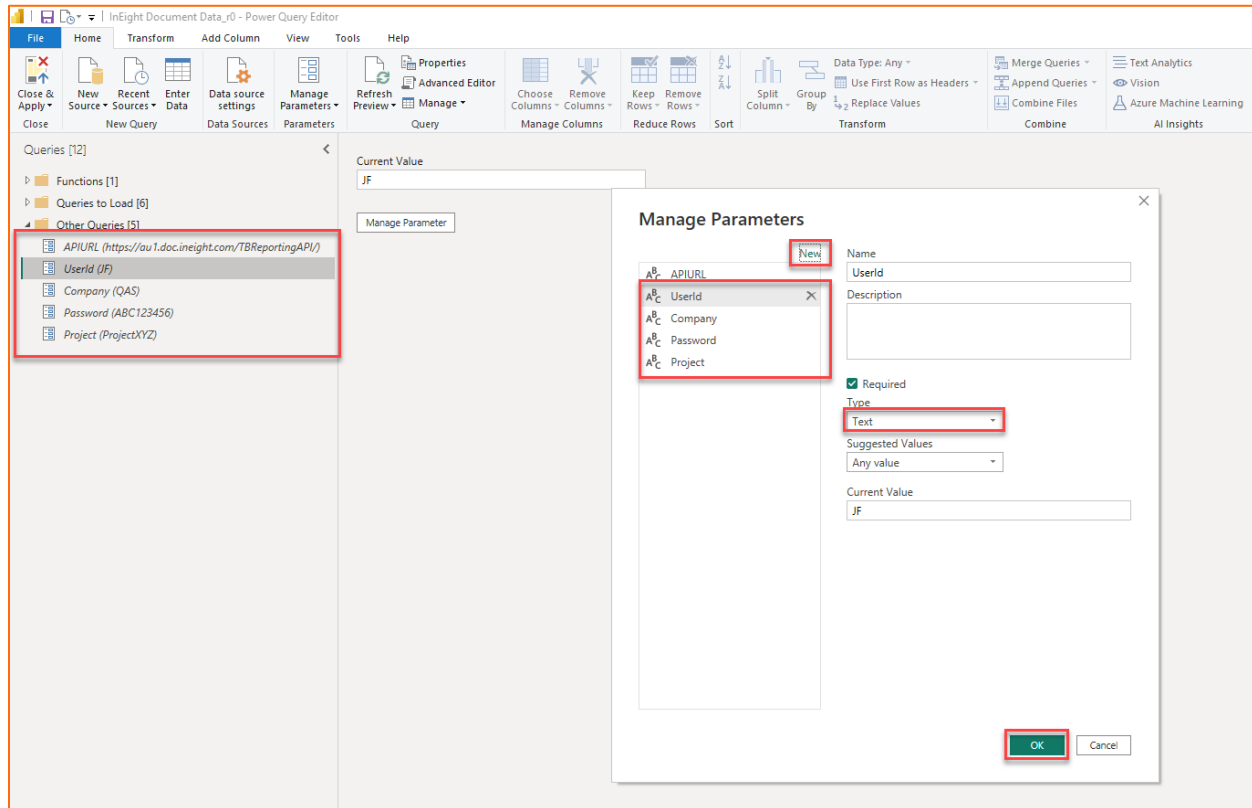


Figure 20: Adding other parameters to enable Reporting API calls

Step 8: Click on Query1 and then click on Advanced Editor under the home tab.

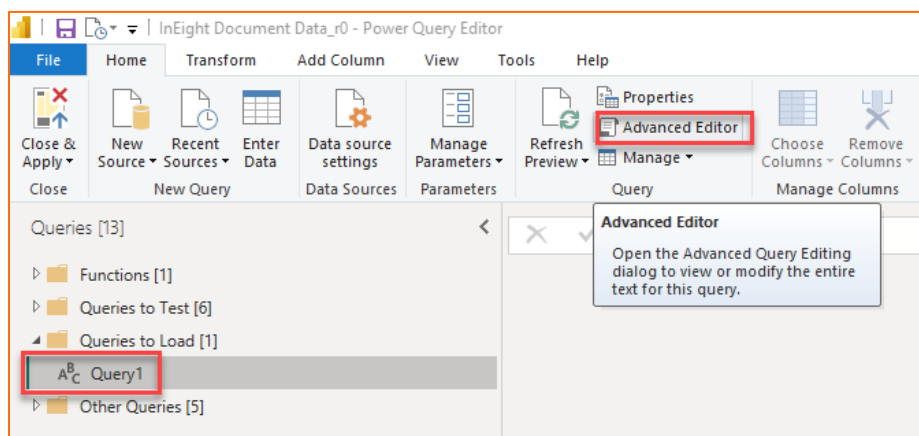


Figure 21: Query1 will be edited to enable extraction of Document data

Step 9: Copy and paste the following script and fill in the required fields.

Type in the Report number in the format of **XX.YYY**

- Report numbers can be found in the Report function in the DOCUMENT web application as well as via this link. [REPORTS](#)

let

Source =

```
(Web.Contents(APIURL, [Headers=[Accept="application/json", #"Content-Type"="application/x-www-form-urlencoded;charset=UTF-8"],
```

```
Content=Text.ToBinary("UserID=" & UserId &
"&CompanyID=" & Company & "&Password=" & Password & "&ProjectNo=" & Project & "&Application=APIDemo"),
RelativePath="session"))),
```

```
SessionKey = Json.Document(Source),
```

```
SessionKeyHeader = "Basic " & SessionKey,
```

```
Source1= Json.Document(Web.Contents(APIURL, [Headers=[Authorization=
SessionKeyHeader],RelativePath="Report/04.001",Timeout=#duration(0,0,30,0))])
```

in

Source1

IMPORTANT: Identifying which reports to call is driven by the organisation's own reporting requirements.

The script above will extract 04.001 Document Register against a single Project.

Step 10: Click on DONE.



Figure 22: Query1 script

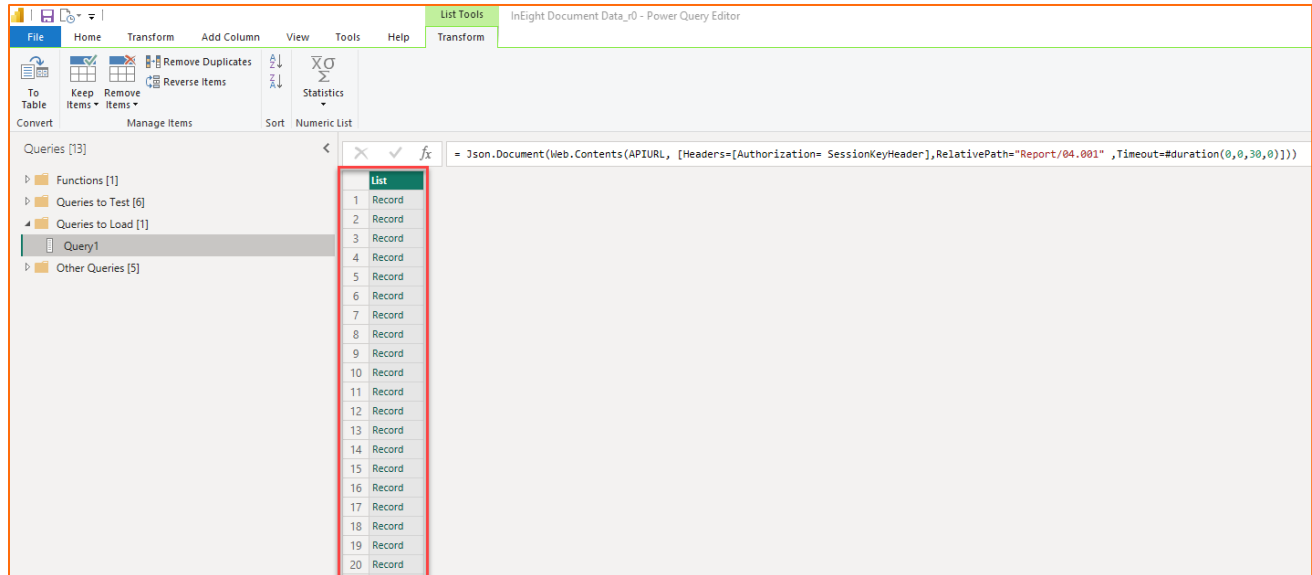


Figure 23: All individual Records will be pulled in. In this example, the report type is 04.001 i.e. Document List, every record from the Document Register.

Step 11: Click on “To Table” under the File tab and click on OK.

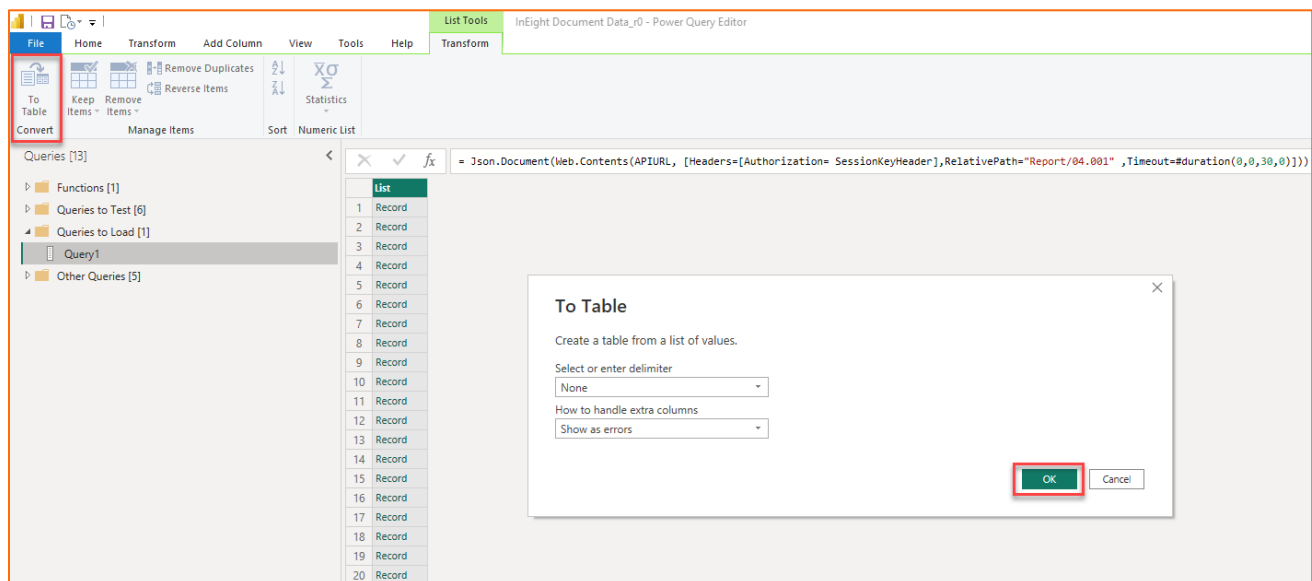


Figure 24: This will convert all the records into Table format.

Step 12: Click on Expand Columns button and then OK.

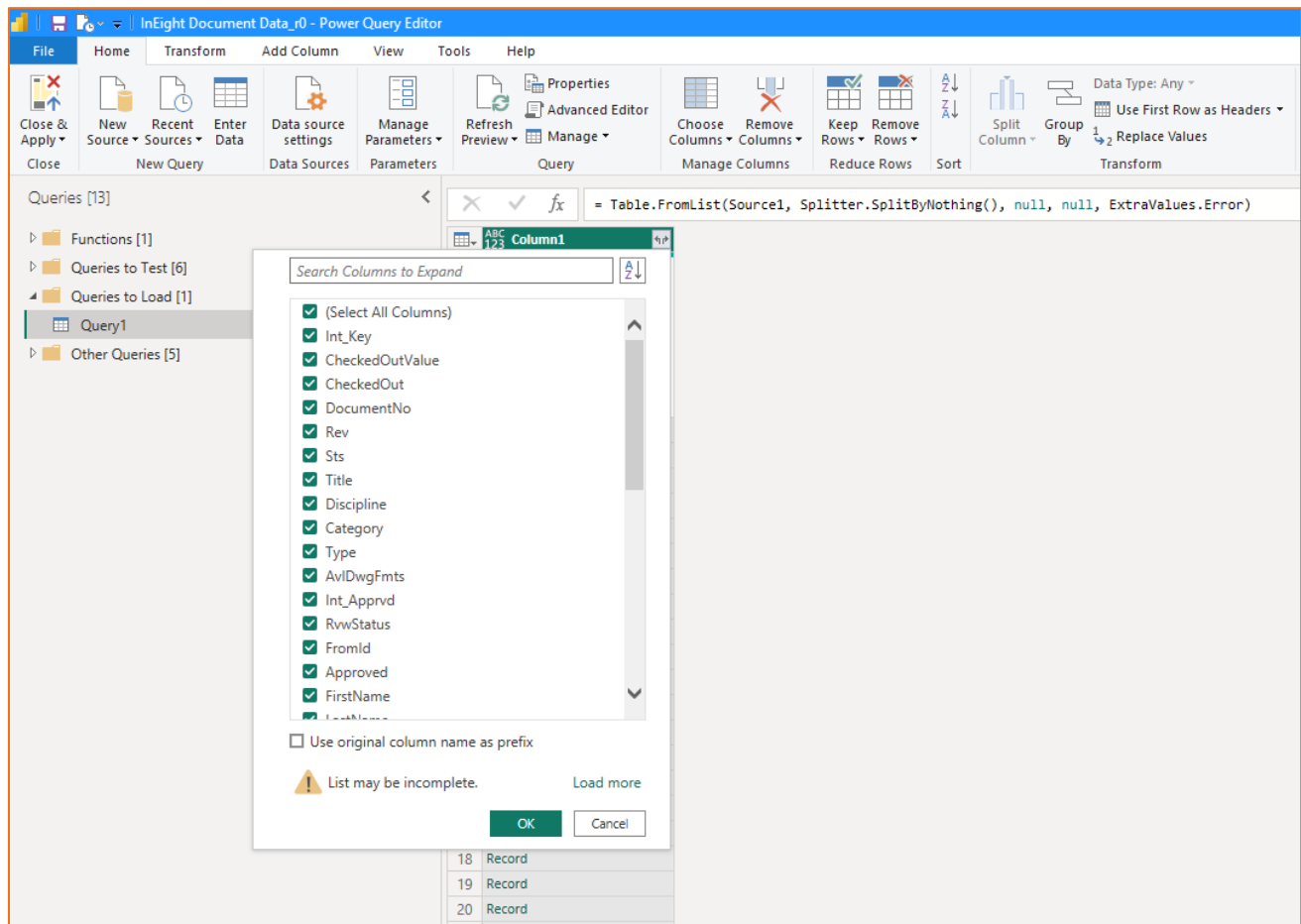


Figure 25: This will convert Records table into a Pivoted Table.

Step 13: Click on Close & Apply

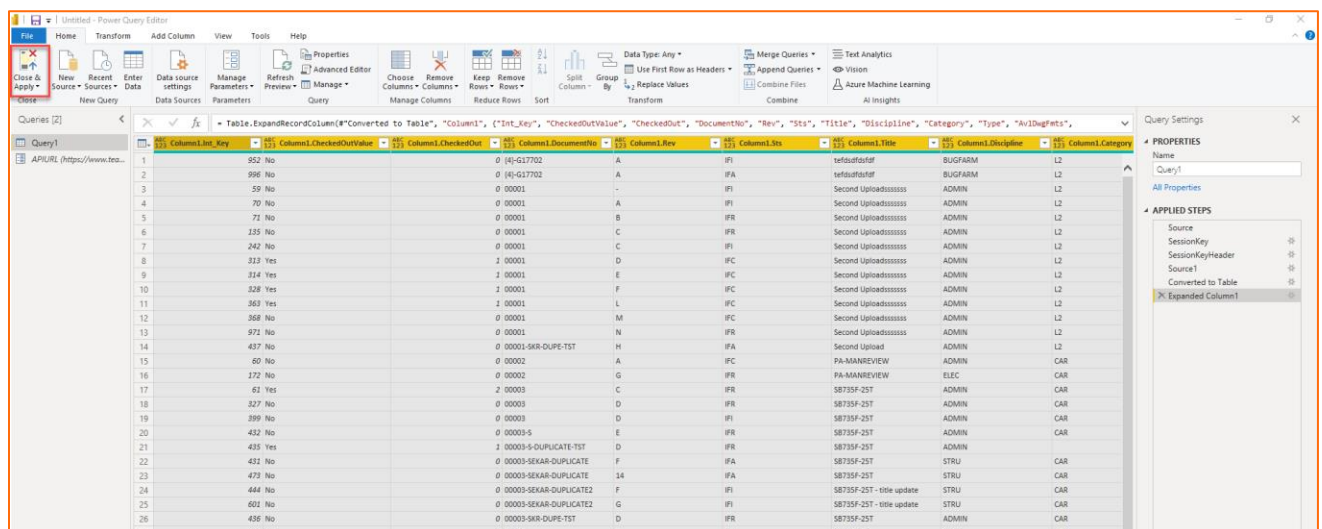
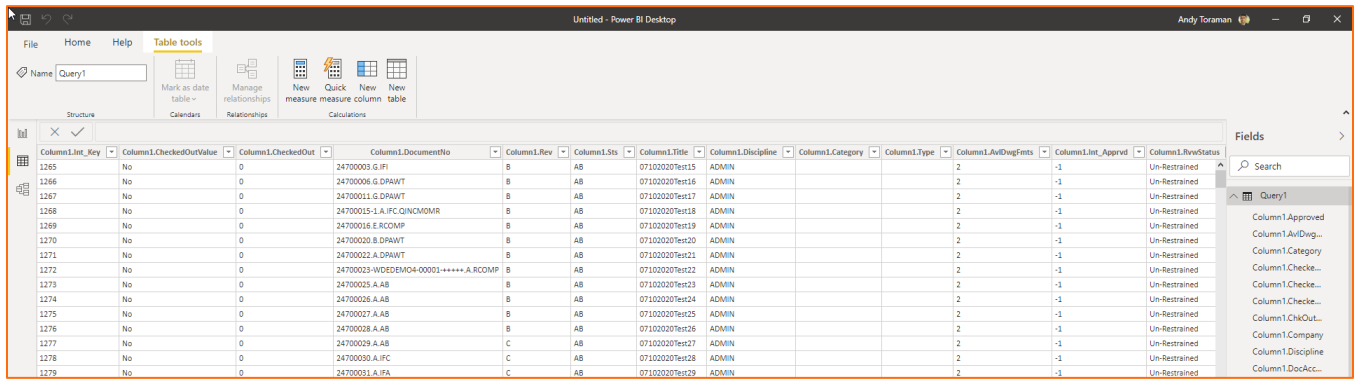


Figure 26: This will bring across the table into Power BI.

Step 14: Go to Data View and Check Out the Data.


Column1.Key	Column1.CheckedOutValue	Column1.CheckedOut	Column1.DocumentNo	Column1.Rev	Column1.Sts	Column1.Title	Column1.Discipline	Column1.Category	Column1.Type	Column1.AviDogFmts	Column1.Int_Approval	Column1.AviStatus
1265	No	0	24700003 G.IPL	B	AB	07102020Test15	ADMIN		2	-1	-1	Un-Released
1266	No	0	24700006 G.DPAWT	B	AB	07102020Test16	ADMIN		2	-1	-1	Un-Released
1267	No	0	24700011 G.DPAWT	B	AB	07102020Test17	ADMIN		2	-1	-1	Un-Released
1268	No	0	24700015-1.A.IFC.QINCMOMR	B	AB	07102020Test18	ADMIN		2	-1	-1	Un-Released
1269	No	0	24700016 E.RCOMP	B	AB	07102020Test19	ADMIN		2	-1	-1	Un-Released
1270	No	0	24700020 B.DPAWT	B	AB	07102020Test20	ADMIN		2	-1	-1	Un-Released
1271	No	0	24700022 A.DPAWT	B	AB	07102020Test21	ADMIN		2	-1	-1	Un-Released
1272	No	0	24700023-WDEDEM04-00001-++++A.RCOMP	B	AB	07102020Test22	ADMIN		2	-1	-1	Un-Released
1273	No	0	24700025 A.AB	B	AB	07102020Test23	ADMIN		2	-1	-1	Un-Released
1274	No	0	24700026 A.AB	B	AB	07102020Test24	ADMIN		2	-1	-1	Un-Released
1275	No	0	24700027 A.AB	B	AB	07102020Test25	ADMIN		2	-1	-1	Un-Released
1276	No	0	24700028 A.AB	B	AB	07102020Test26	ADMIN		2	-1	-1	Un-Released
1277	No	0	24700029 A.AB	C	AB	07102020Test27	ADMIN		2	-1	-1	Un-Released
1278	No	0	24700030 A.IFC	C	AB	07102020Test28	ADMIN		2	-1	-1	Un-Released
1279	No	0	24700031 A.IFA	C	AB	07102020Test29	ADMIN		2	-1	-1	Un-Released

Figure 27: The data table can be seen in the Data view.

Step 15: To bring in data from other reports into Power BI, repeat Step 2 and then from Step 8 onward.

Step 16: Go to the Report View and you can start building dashboards.

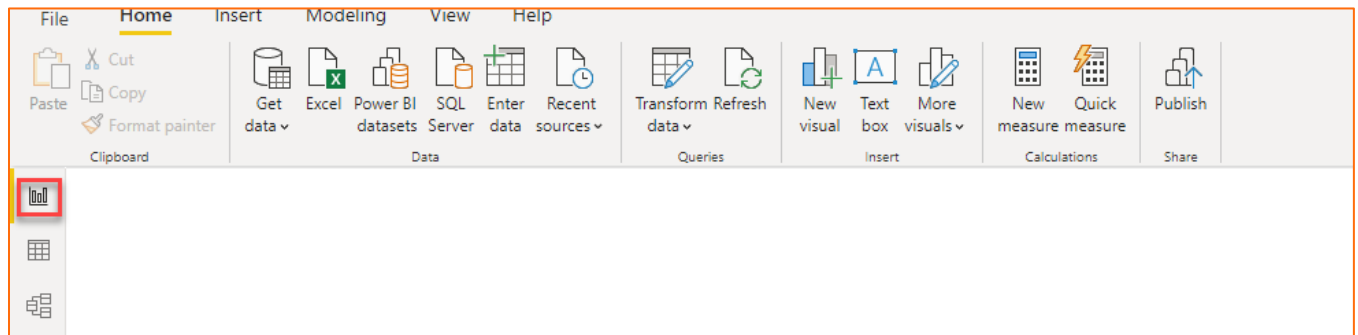


Figure 28: Dashboards can be built using the data fields i.e. columns in the data tables.

Portfolio View across all Projects

The following steps will build on the details described in the section above and will explain how to bring the REPORT data from each InEight DOCUMENT project into Power BI.

Step 1: Launch Power BI Desktop.

Step 2: Go to the Home tab and click on Get Data dropdown and select Blank Query.

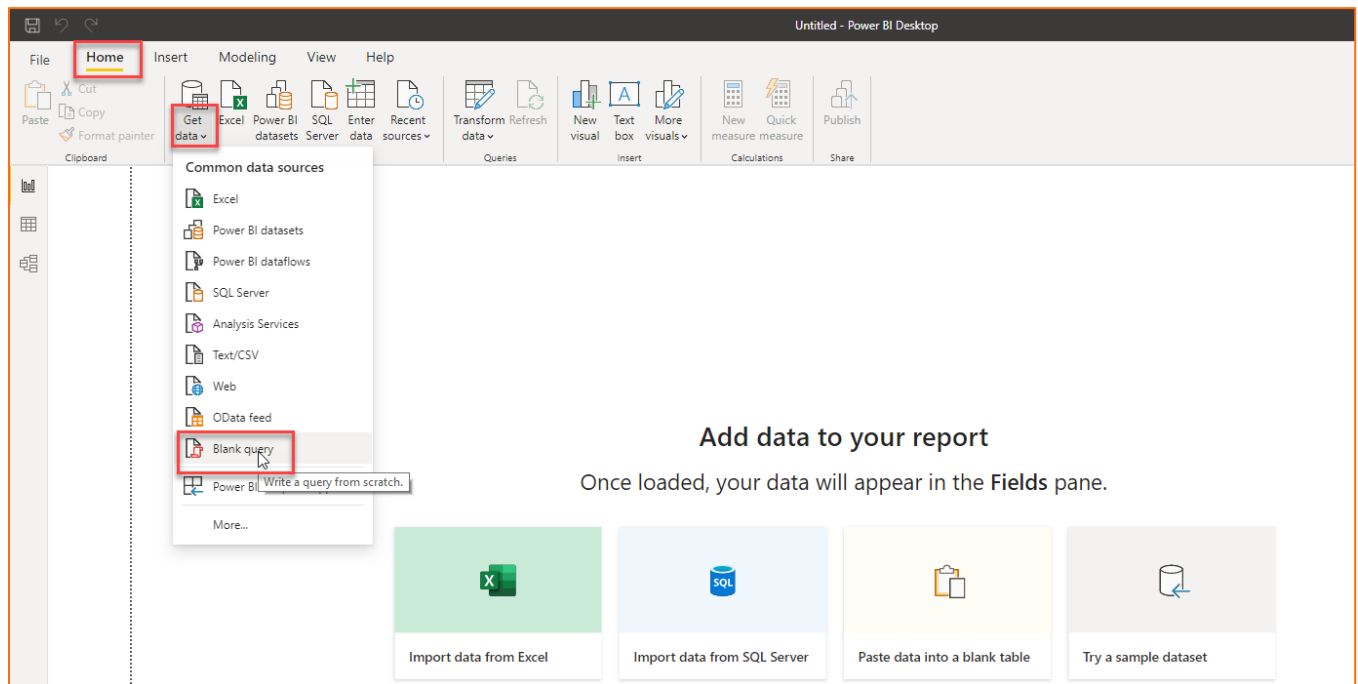


Figure 29: The query editor will pop up.

Step 3: Go to Manage Parameters under the Home Tab and click on “New Parametres”

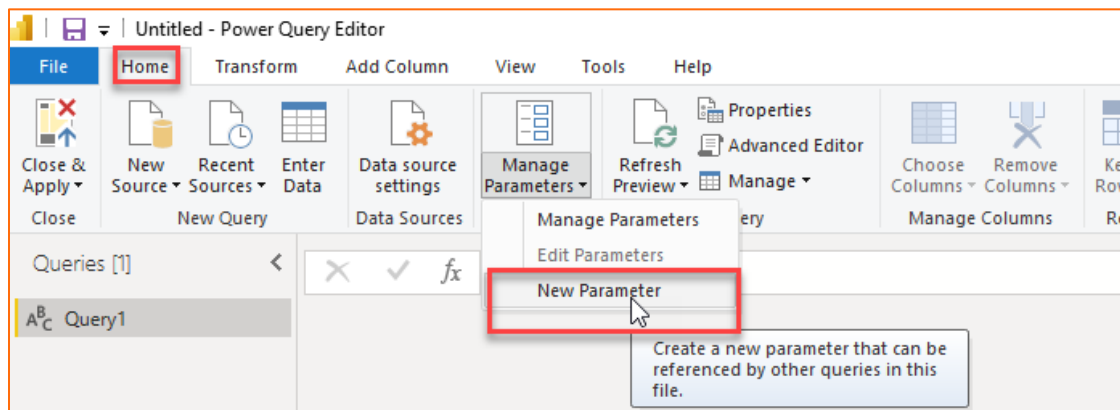


Figure 30: The same URL that allowed for accessing Swagger will be defined as the parameter.

Step 4: Type “APIURL” in the Name field.

Step 5: Paste the url in the Current Value field.

- The url will be <https://server/TBReportingAPI/>

- It can be acquired once logged in the DOCUMENT web application.

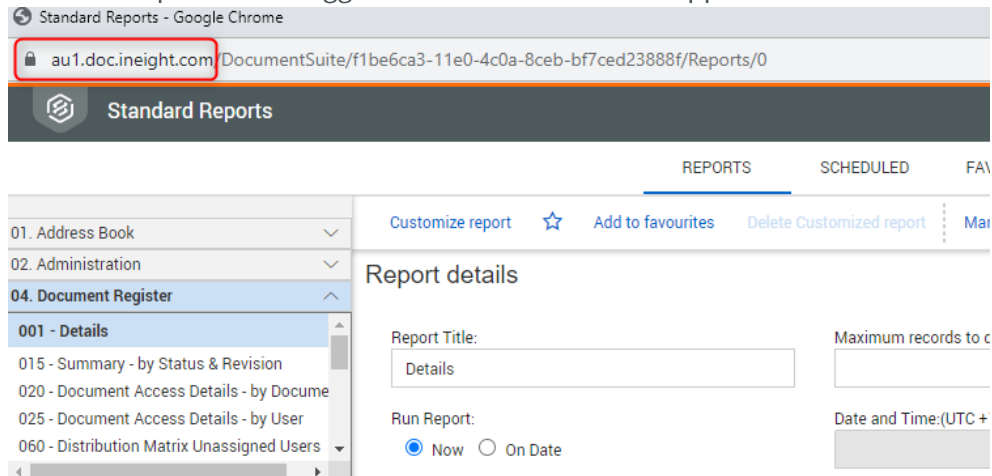


Figure 31: In this example, the URL will be <https://au1.doc.ineight.com/TBReportingAPI/>

Step 6: Click OK.

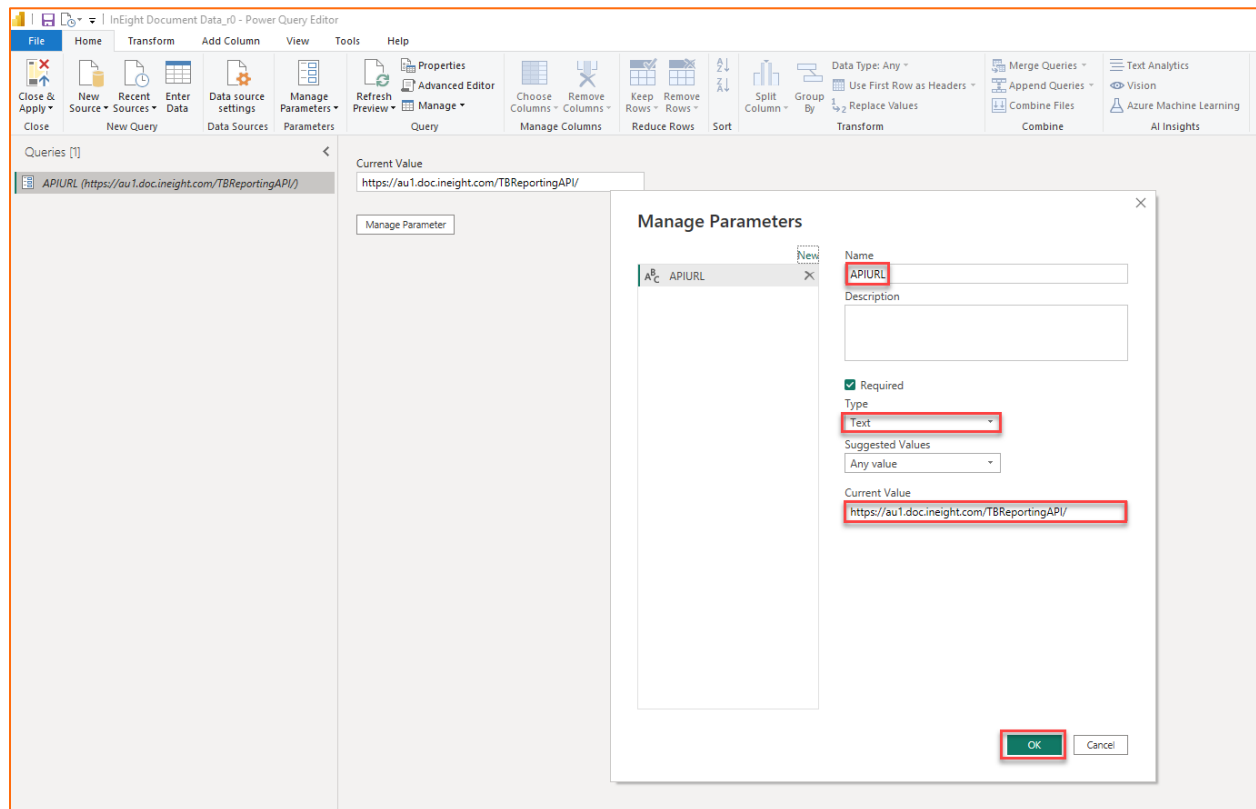


Figure 32: This will define the URL as the parameter to be used in the query.

Step 7: Click on New to add the following parametres and then click OK

- Userld
- Company
- Password

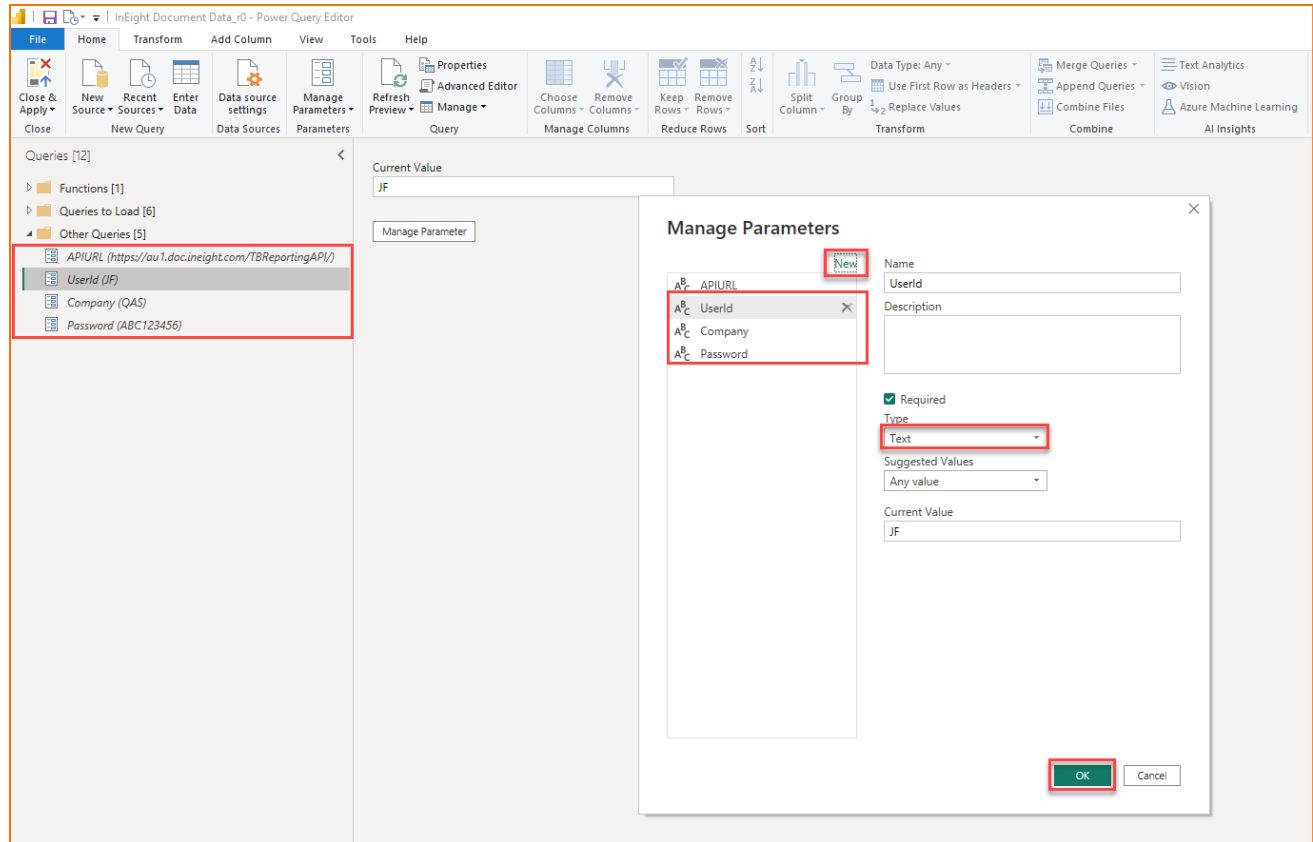


Figure 33: Adding other parameters to enable Reporting API calls.

Step 8: Click on New Source dropdown and select Blank Query

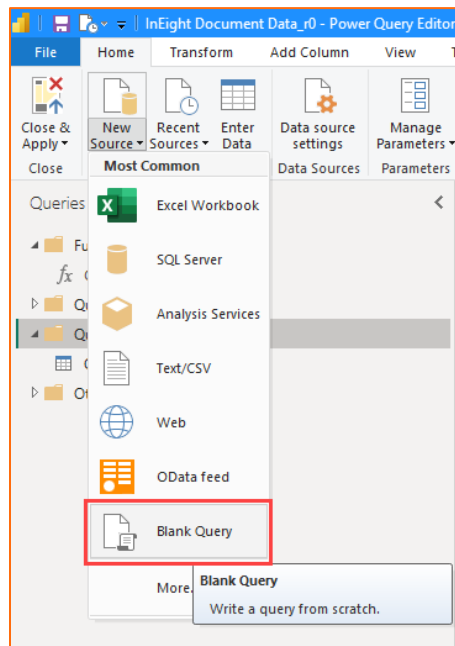


Figure 34: Adding a new query to convert into a function

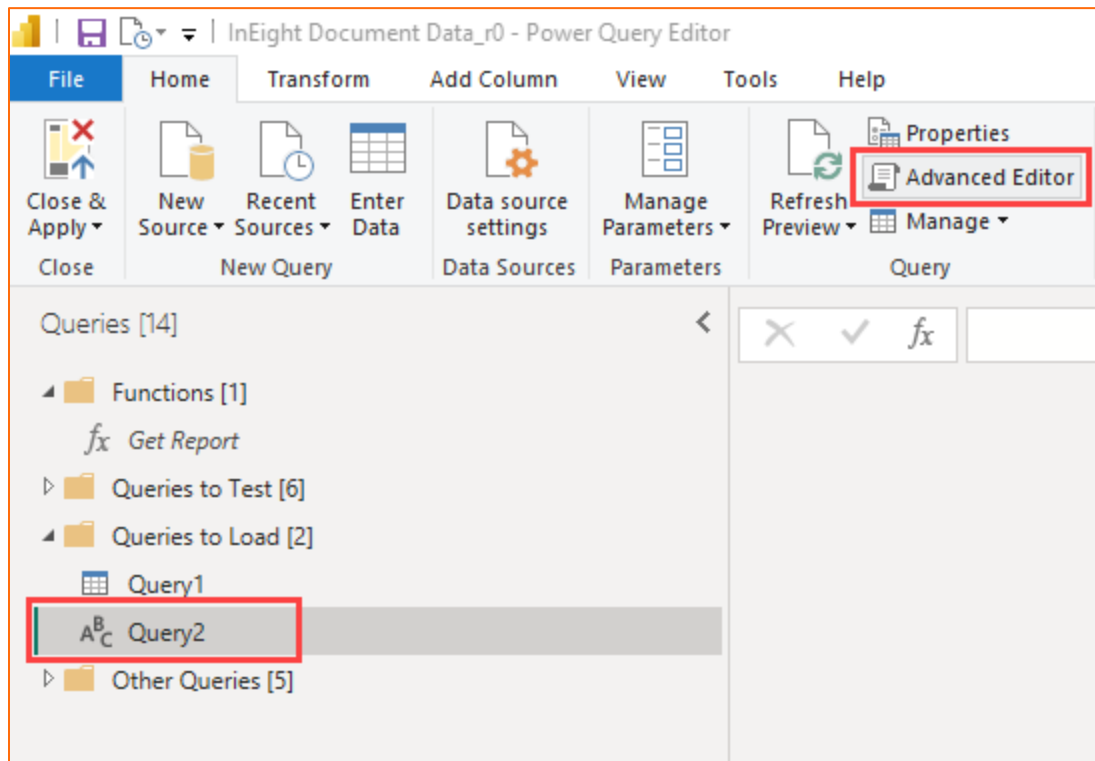
Step 8: Open Advanced Editor

Figure 35: Open Advanced Editor to convert Query 2 to a function

Step 9: Copy and Paste the following script and click DONE

(ProjectNoFromQuery as text, reportNo as text) =>

let

Source =

(Web.Contents(APIURL, [Headers=[Accept="application/json", #"Content-Type"]="application/x-www-form-urlencoded; charset=UTF-8"],

Content=Text.ToBinary("UserID=" & UserId &
"&CompanyID=" & Company & "&Password=" & Password & "&ProjectNo=" & ProjectNoFromQuery &
"&Application=APIDemo"), RelativePath="session"))),

SessionKey = Json.Document(Source),

SessionKeyHeader = "Basic " & SessionKey,

Source1= Json.Document(Web.Contents(APIURL, [Headers=[Authorization=
SessionKeyHeader], RelativePath="Report/" & reportNo, Timeout=#duration(0,0,30,0))])

in

Source1

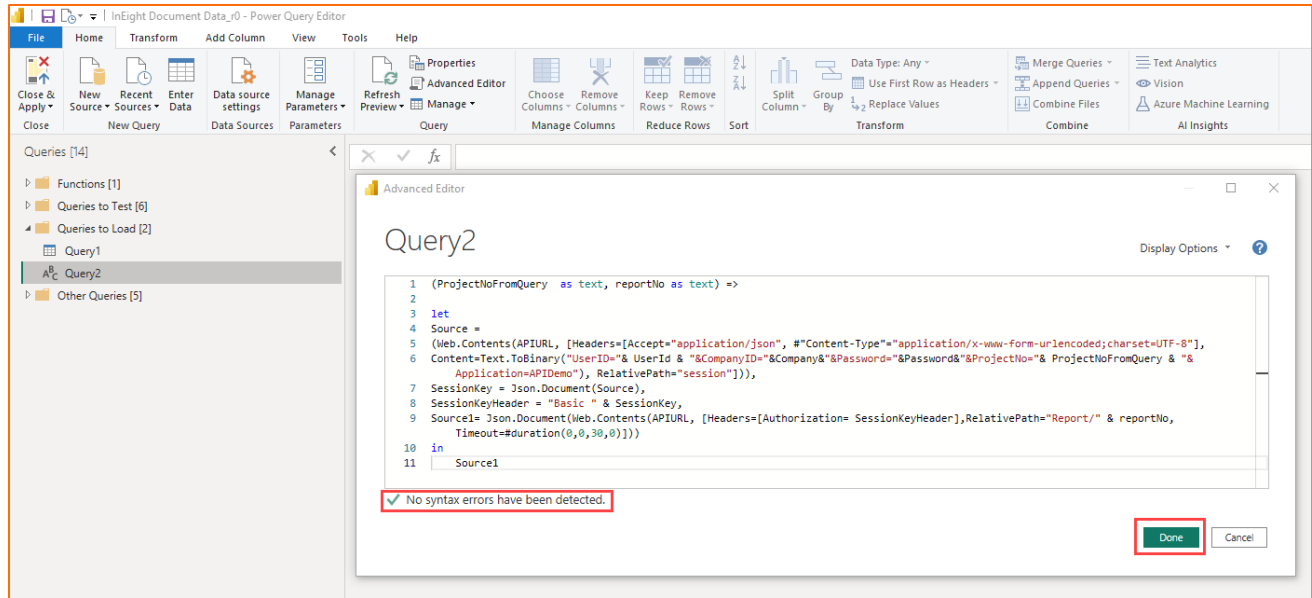


Figure 36: Creating power query function to invoke against all projects.

Step 10: Give your Function the name 'Get Report' then press 'OK'

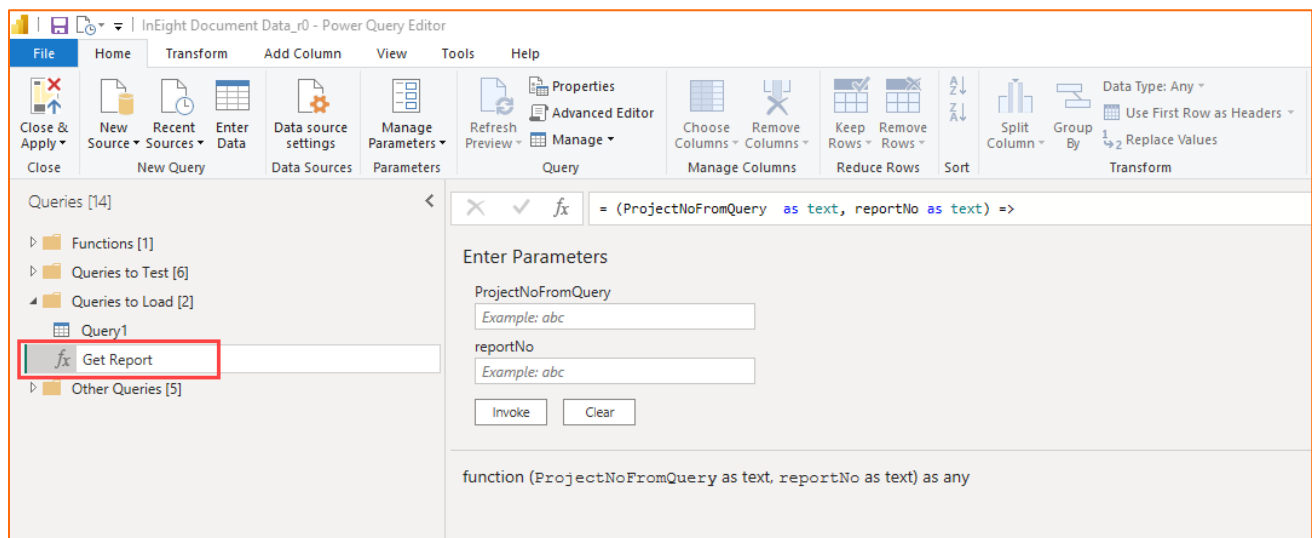


Figure 37: Rename the function

Step 11: Repeat Step 8 to add another new query

Step 12: Open Advanced Editor

Step 13: Copy and Paste the following script.

```
let
```

```
    Query1 = let
```

```
        Source = (Web.Contents(APIURL, [
```

```
            Headers=[Accept="application/json", Authorization="Basic " & Password],
```

```
            RelativePath="Company/" & Company & "/User/" & UserId & "/Project"])),
```

```
        #"Imported JSON" = Json.Document(Source,65001)
```

```
in
```

```
    Source,
```

```
        #"Imported JSON" = Json.Document(Query1,65001),
```

```
        #"Converted to Table" = Table.FromList(#"Imported JSON", Splitter.SplitByNothing(), null, null, ExtraValues.Error),
```

```
        #"Expanded Column1" = Table.ExpandRecordColumn(#"Converted to Table", "Column1", {"Int_Proj",
"MailNewCount", "DocsForReview", "DocsForRelease", "MailUnregisteredCount", "OutstMLCnt", "ProjNo", "ProjTitle",
"ServerUrl", "Int_Server", "Int_Key"}, {"Int_Proj", "MailNewCount", "DocsForReview", "DocsForRelease",
"MailUnregisteredCount", "OutstMLCnt", "ProjNo", "ProjTitle", "ServerUrl", "Int_Server", "Int_Key"}),
```

```
        #"Removed Other Columns" = Table.SelectColumns(#"Expanded Column1",{"ProjNo"}),
```

```
        #"Changed Type" = Table.TransformColumnTypes(#"Removed Other Columns",{{"ProjNo", type text}})
```

```
in
```

```
    #"Changed Type"
```

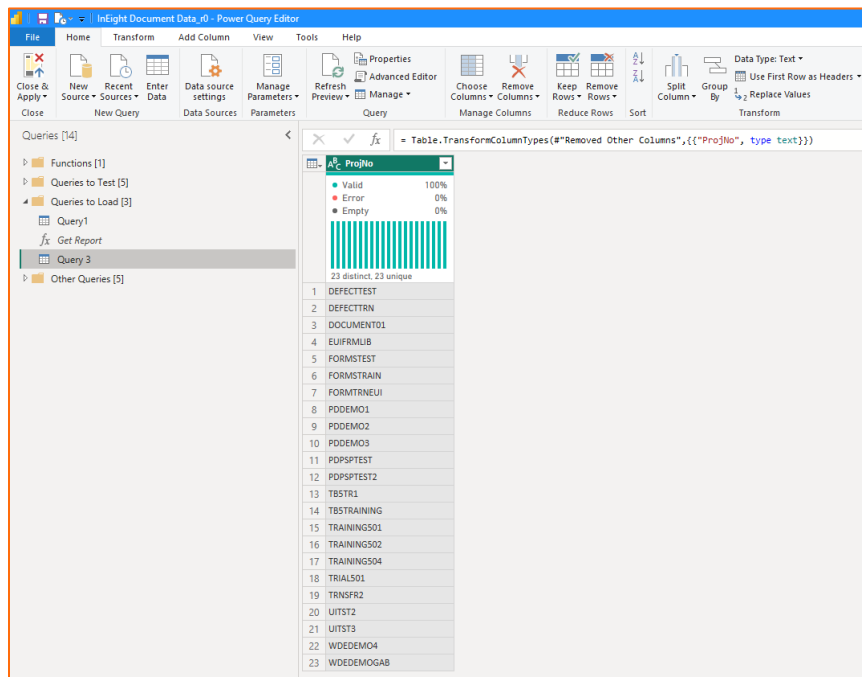


Figure 38: Project numbers to be used to invoke the function

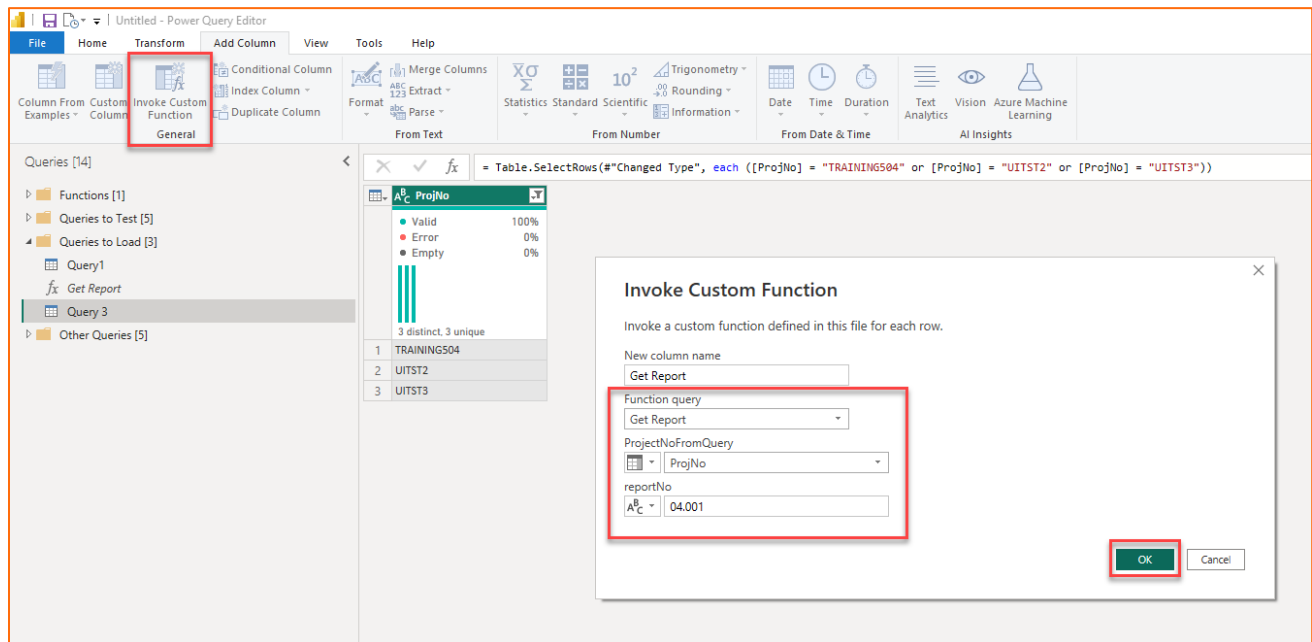
Step 14: Select Add Column tab and Invoke the Get Report function

Figure 39: Report Number 04.001 is manually entered.

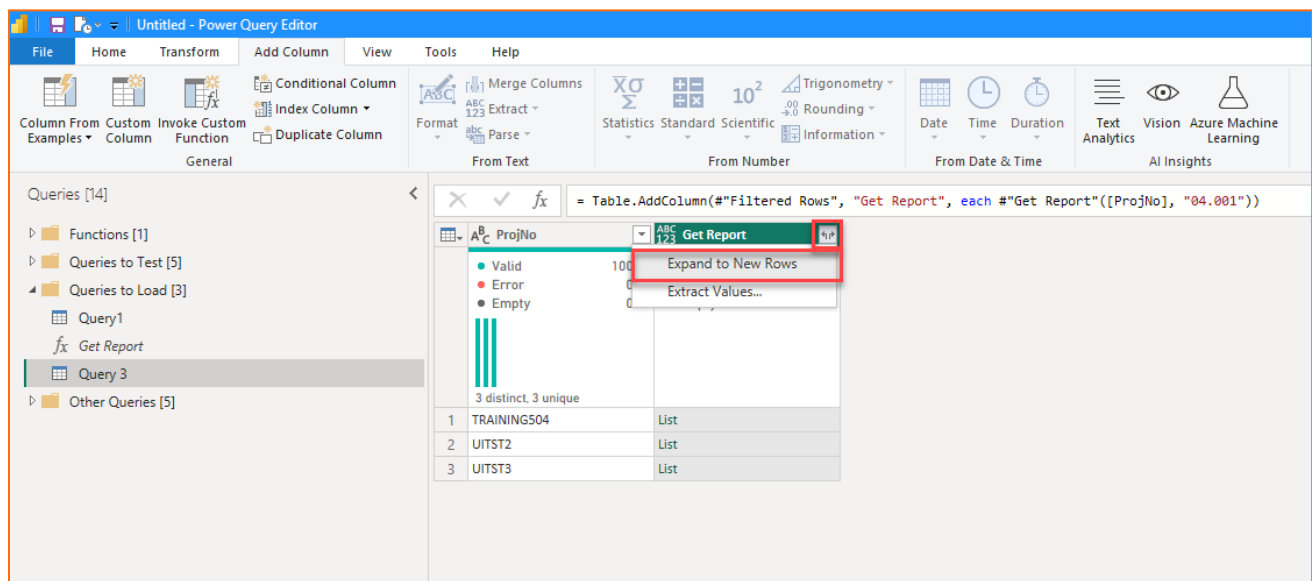
Step 15: Expand the Get Report column to New Rows

Figure 40: Transforming the table by creating new rows based on the data against each project

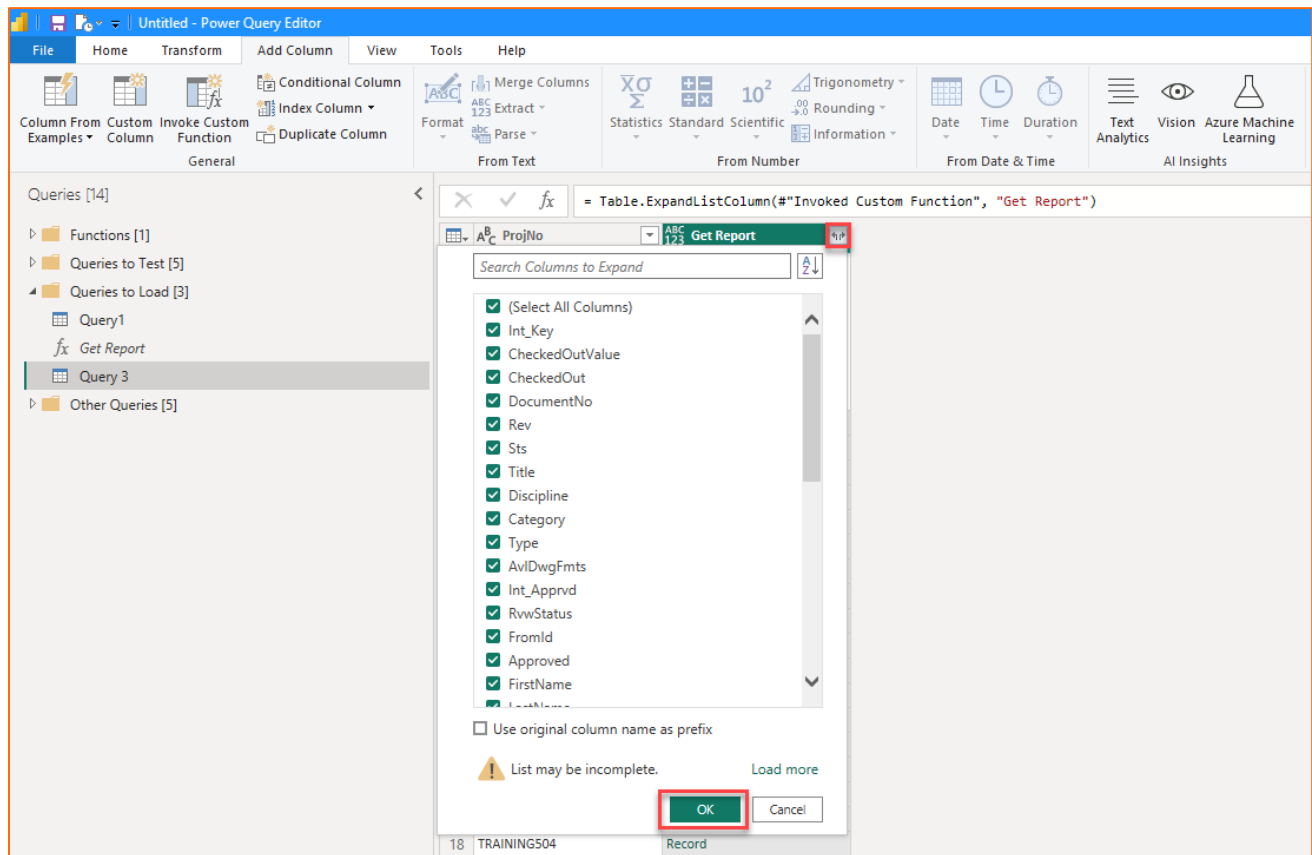
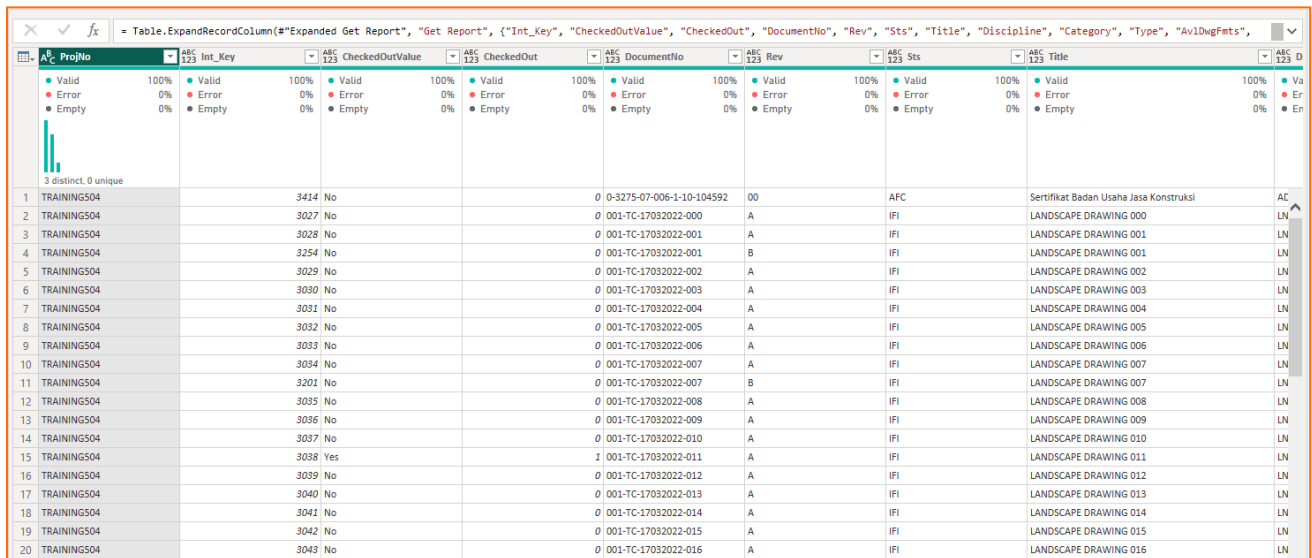
Step 16: Expand the records in the Get Report column

Figure 41: Expanding each record and populating the table

Step 17: Expand the records in the Get Report column


The screenshot shows the Power Query Editor interface with the expanded table. The formula bar displays the expression: `= Table.ExpandRecordColumn(#"Expanded Get Report", "Get Report", {"Int_Key", "CheckedOutValue", "CheckedOut", "DocumentNo", "Rev", "Sts", "Title", "Discipline", "Category", "Type", "AvlDwgFmts", "Int_Apprvd", "RvwStatus", "FromId", "Approved", "FirstName", "LastModified"})`. The table has 20 rows of data for 'TRAINING504'. The columns are: ProjNo, Int_Key, CheckedOutValue, CheckedOut, DocumentNo, Rev, Sts, Title, Discipline, Category, Type, AvlDwgFmts, and Int_Apprvd. The table is populated with data for 'TRAINING504'.

ProjNo	Int_Key	CheckedOutValue	CheckedOut	DocumentNo	Rev	Sts	Title	Discipline	Category	Type	AvlDwgFmts	Int_Apprvd	RvwStatus	FromId	Approved	FirstName	LastModified
1	TRAINING504	3414	No	0	0-3275-07-006-1-10-104592	00	AFC	Sertifikat Badan Usaha Jasa Konstruksi	AC								
2	TRAINING504	3027	No	0	001-TC-17032022-000	A	IFI	LANDSCAPE DRAWING 000	LN								
3	TRAINING504	3028	No	0	001-TC-17032022-001	A	IFI	LANDSCAPE DRAWING 001	LN								
4	TRAINING504	3254	No	0	001-TC-17032022-001	B	IFI	LANDSCAPE DRAWING 001	LN								
5	TRAINING504	3029	No	0	001-TC-17032022-002	A	IFI	LANDSCAPE DRAWING 002	LN								
6	TRAINING504	3030	No	0	001-TC-17032022-003	A	IFI	LANDSCAPE DRAWING 003	LN								
7	TRAINING504	3031	No	0	001-TC-17032022-004	A	IFI	LANDSCAPE DRAWING 004	LN								
8	TRAINING504	3032	No	0	001-TC-17032022-005	A	IFI	LANDSCAPE DRAWING 005	LN								
9	TRAINING504	3033	No	0	001-TC-17032022-006	A	IFI	LANDSCAPE DRAWING 006	LN								
10	TRAINING504	3034	No	0	001-TC-17032022-007	A	IFI	LANDSCAPE DRAWING 007	LN								
11	TRAINING504	3201	No	0	001-TC-17032022-007	B	IFI	LANDSCAPE DRAWING 007	LN								
12	TRAINING504	3035	No	0	001-TC-17032022-008	A	IFI	LANDSCAPE DRAWING 008	LN								
13	TRAINING504	3036	No	0	001-TC-17032022-009	A	IFI	LANDSCAPE DRAWING 009	LN								
14	TRAINING504	3037	No	0	001-TC-17032022-010	A	IFI	LANDSCAPE DRAWING 010	LN								
15	TRAINING504	3038	Yes	1	001-TC-17032022-011	A	IFI	LANDSCAPE DRAWING 011	LN								
16	TRAINING504	3039	No	0	001-TC-17032022-012	A	IFI	LANDSCAPE DRAWING 012	LN								
17	TRAINING504	3040	No	0	001-TC-17032022-013	A	IFI	LANDSCAPE DRAWING 013	LN								
18	TRAINING504	3041	No	0	001-TC-17032022-014	A	IFI	LANDSCAPE DRAWING 014	LN								
19	TRAINING504	3042	No	0	001-TC-17032022-015	A	IFI	LANDSCAPE DRAWING 015	LN								
20	TRAINING504	3043	No	0	001-TC-17032022-016	A	IFI	LANDSCAPE DRAWING 016	LN								

Figure 42: Table populated

Connecting Reporting APIs with Power BI

Step 18: Transform data as required.

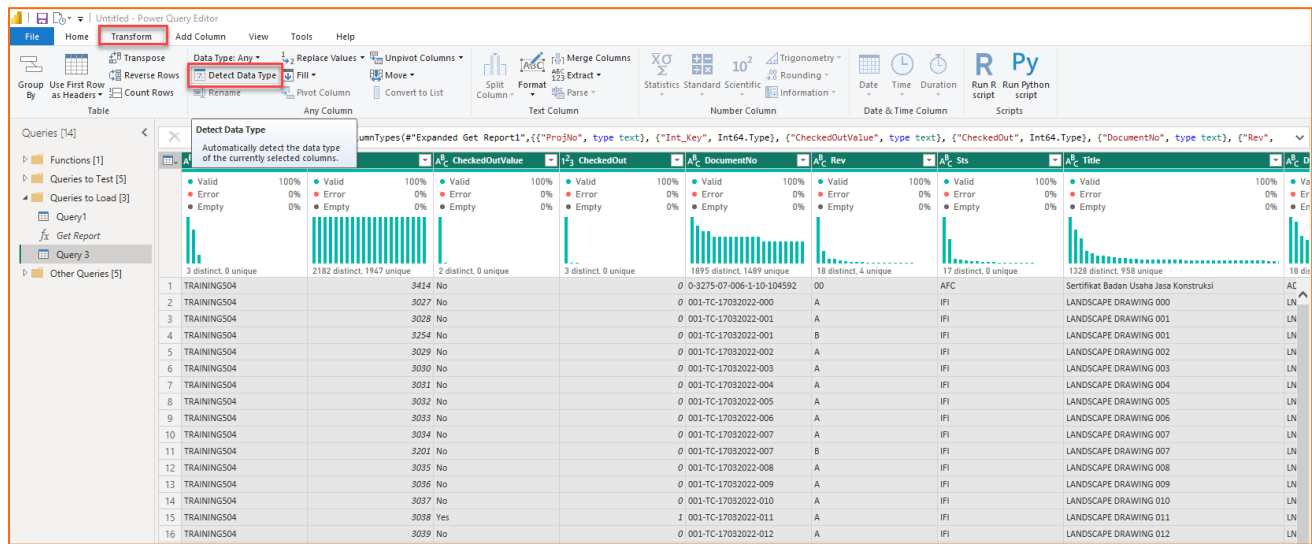


Figure 43: Select All columns and get Power Query to detect the data type based on the date in each column. (example)

Step 19: Rename the query

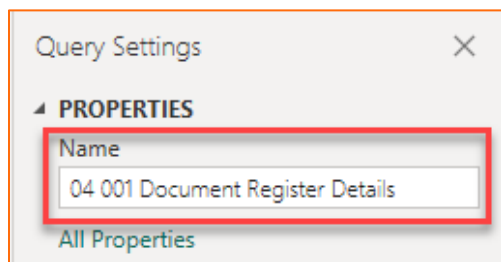


Figure 44: Re naming the query as per the title of the report

Step 20: Load the data into Power BI

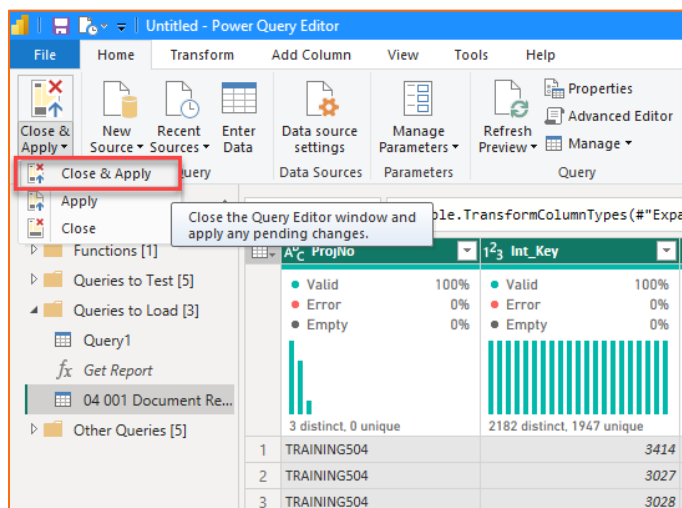


Figure 45: This will load the table into the power bi data model with import storage mode