

DEPARTMENT OF PLANNING, INDUSTRY & ENVIRONMENT

Air Quality Application Programming Interface (API) User Guide

Climate & Atmospheric Science Branch



© 2021 State of NSW and Department of Planning, Industry and Environment

With the exception of photographs, the State of NSW and Department of Planning, Industry and Environment are pleased to allow this material to be reproduced in whole or in part for educational and non-commercial use, provided the meaning is unchanged and its source, publisher and authorship are acknowledged. Specific permission is required for the reproduction of photographs.

The Department of Planning, Industry and Environment (DPIE) has compiled this report in good faith, exercising all due care and attention. No representation is made about the accuracy, completeness or suitability of the information in this publication for any particular purpose. DPIE shall not be liable for any damage which may occur to any person or organisation taking action or not on the basis of this publication. Readers should seek appropriate advice when applying the information to their specific needs.

All content in this publication is owned by DPIE and is protected by Crown Copyright, unless credited otherwise. It is licensed under the <u>Creative Commons Attribution 4.0 International</u> (<u>CC BY 4.0</u>), subject to the exemptions contained in the licence. The legal code for the licence is available at <u>Creative Commons</u>.

DPIE asserts the right to be attributed as author of the original material in the following manner: © State of New South Wales and Department of Planning, Industry and Environment 2021.

Cover photo: Darling River campground (Yapara Paaka Thuru), Toorale National Park. Joshua Smith/DPIE

Published by:

Environment, Energy and Science Department of Planning, Industry and Environment 4 Parramatta Square, 12 Darcy Street, Parramatta NSW 2150 Phone: +61 2 9995 5000 (switchboard) Phone: 1300 361 967 (Environment, Energy and Science enquiries) TTY users: phone 133 677, then ask for 1300 361 967 Speak and listen users: phone 1300 555 727, then ask for 1300 361 967 Email: <u>info@environment.nsw.gov.au</u> Website: <u>www.environment.nsw.gov.au</u>

Report pollution and environmental incidents Environment Line: 131 555 (NSW only) or <u>info@environment.nsw.gov.au</u> See also <u>www.environment.nsw.gov.au</u>

ISBN 978 1 922318 71 8 EES 2021/0346 First published in June 2020; second edition published in August 2021

Find out more about your environment at:

www.environment.nsw.gov.au

Contents

1.	Scope		1
2.	Introd	uction	1
3.	Overv	view of the DPIE Air Quality API data service	2
	3.1	Air Quality API data sets	2
4.	Searc websi	h and retrieve air quality data using Air Quality Data API te (swagger)	3
	4.1	Setting up a query in the Air Quality Data API website (swagger)	4
5.	Searc	h and retrieve air quality data using Excel and API	19
	5.1	Setting up a Power Query in Excel using the API	19
6.	Searc langu	h and retrieve air quality data using the R programming age and API	36
	6.1	Deploying the API using the R programming language	36
7.	Searc langu	h and retrieve air quality data using Python programming age and API	41
	7.1	Deploying the API using the Python programming language	41
8.	Apper using	ndix 1: worked examples Extract Historical air quality data Excel and API	47
	8.1	Example 1	47
	8.2	Example 2	48
9.	Apper using	ndix 2: worked examples Extract Historical air quality data R programming language and API	49
	9.1	Example 1.	49
	9.2	Example 2	51
10.	Apper using	ndix 3: worked examples Extract Historical air quality data Python programming language and API	53
	10.1	Example 1.	53
	10.2	Example 2	55
11.	Apper	ndix 4: Data dictionary for air quality data	57

List of tables

Table 1	Category and Subcategory combinations	35
Table 2	Air Quality data set and descriptions	57

List of figures

Figure 1	Air Quality API Architecture	2
Figure 2	Air Quality Data API - swagger	3
Figure 3	Screenshot of Air Quality Data API website (swagger) interface	4
Figure 4	Excel – Enable all Data Connections (not recommended)	19
Figure 5	Excel – Blank query	20

1. Scope

This document describes the process for searching and retrieving Department of Planning, Industry and Environment (DPIE) air quality, meteorological and monitoring site data from the DPIE Azure Cloud Data Warehouse using an Application Programming Interface (API) data service and:

- Excel Power Query
- R programming language
- Python programming language

Although this document describes how to extract DPIE air quality data using the Air Quality API and Excel software, R and Python programming languages, the main purpose of an API is to connect software applications and stream data.

2. Introduction

Currently the DPIE Air Quality webpages provide a facility to search and retrieve static air quality data. Manual intervention is required to retrieve data from the DPIE Air Quality web site. The DPIE Air Quality API is a data service providing an open access Application Programming Interface (API), to enable IT application developers and end-users to integrate current real-time and historical DPIE air quality data, with a variety of software applications.

The scope of applications for which the DPIE Air Quality API data service can provide data for, spans the full breadth of potential uses, from mobile apps, organisational decision-management business systems to dynamic streaming of online air quality data as outlined in Figure 1.

The DPIE Air Quality API data service does not replace the existing DPIE air quality search and retrieve facility, as it is intended for use by software applications, not as a user interface. The DPIE air quality data available through the API, has been fully quality assured, in accordance with DPIE quality assurance procedures.

Further air quality information in relation to air quality data and monitoring sites can be found on the DPIE Air Quality web site:

https://www.environment.nsw.gov.au/topics/air

This Air Quality Application Interface (API) User Guide is an updated version to the document issued in June 2020. The main change has been the replacement of Air Quality Index (AQI) with Air Quality Categories (AQC). This has resulted in Air Quality Index values being removed from the Air Quality API. The reason for the implementation of Air Quality Categories can be found on the DPIE Air Quality web site:

https://www.environment.nsw.gov.au/topics/air/understanding-air-quality-data/air-qualitycategories

3. Overview of the DPIE Air Quality API data service

The DPIE Air Quality API (the API) data service provides a standardised RESTful (REpresentational State Transfer) protocol for searching and retrieving data. The Air Quality API architecture is shown in Figure 1. The Client software applications is not exhaustive and is only provided as an example.





Air Quality API Architecture

3.1 Air Quality API data sets

The Air Quality API provides three data end-points (data sets):

1. Sites

Monitoring site data including the site name, longitude, latitude and region. (url = "https://data.airquality.nsw.gov.au/api/Data/get SiteDetails")

2. Parameters

Details the air pollutants and meteorological parameters, monitored by the DPIE including parameter averaging periods and exceedances.

(url = "https://data.airquality.nsw.gov.au/api/Data/get_ParameterDetails")

3. Observations

Air quality and meteorological observational data collected from the air quality monitoring network, can be searched and downloaded as current real-time hourly data or as historical data. Current observational data is refreshed every hour.

(url = "https://data.airquality.nsw.gov.au/api/Data/get_Observations")

4. Search and retrieve air quality data using Air Quality Data API website (swagger)

The DPIE API web service can be used by a variety of third-party software applications as identified in Figure 1, to search and retrieve DPIE air quality data. The following outlines how to use the Air Quality Data API website (swagger) (below) to search and retrieve air quality and related data stored in the DPIE Azure Cloud Data Warehouse. The data is returned as a **json file format**.

To go to the Air Quality Data API website (swagger) click on the link below: <u>https://data.airquality.nsw.gov.au/docs/index.html</u>

NSW	Planning, Industry & Environment	Air Quality Data API			
			🔒 swagger	Select a spec Air Quality API V1	~
			Air Quality Data API Contemporation April		
			Sites		~
			Parameters		~
			CAT /api/Data/get_ParameterDetails Retrieve Parameters available for filtering		
			Observations		\sim
			POST /api/Data/get_Observations Retrieve Observation data		
			Models		~
			Site >		
			ErrorResponse >		
			Parameter >		
			ObservationRequest >		
			ObservationResult >		
		Diadomer: We strive t	comply with MICs Web Content Accountility Edification. If you are under to access content on this which due to a disability, plans request as observative former by con	tecting newski371@enviburement.new.gov.au. Where possible, we are committed to prov	riding alternative versions on request.

Figure 2 Air Quality Data API - swagger

The Air Quality API provides three data sets:

- Site details data
- Parameter details data
- Observational data. The observational data can be extracted as current real-time hourly data or as historical data by querying data through the body command.

The information under Models provides details on what data is retrieved from Sites, Parameters and Observations API. This is also a Data Dictionary which details the technical details of the API.

4.1 Setting up a query in the Air Quality Data API website (swagger)

The Air Quality Data API website (swagger) provides three data sets:

- Site details data
- Parameter details data
- Observational data.

The observational data can be extracted as current real-time hourly data or as historical data by querying data through the body query command.

A screenshot of the Air Quality Data API website (swagger) interface is below:



The Site Details and Parameter Details can also be queried through the body command in the body query command. However, since this is static data it is easier to search and filter using a third-party application like Excel. Using the body query command to filter site and parameter details, is not possible in the Air Quality Data API website

The get_SiteDetails and get_ParameterDetails use the **GET** method to retrieve data.

The get_Observations use the **POST** method to retrieve data.

4.1.1 Extract Monitoring Site data using the Air Quality Data API website (swagger)

The following instructions outline how to extract air quality monitoring Site data from the DPIE Azure Cloud Data Warehouse using the Air Quality API website. It is not possible to add parameters to filter the Site data.



1. Click on the **GET** button.

Sites						
GET /api/Data/get_SiteDetails Retrieve Sites available for filtering						
Retrieve Sites available for filtering						
Parameters			Try it out			
No parameters						
Responses		Response content type	application/json ~			
Code	Description					
200	Returns list of Sites					
500	Example Value Model					

- 2. Click on Try it out
- 3. Click on Execute.

Sites		\checkmark			
GET /api/Data/get_S	iteDetails Retrieve Sites available for filtering				
Retrieve Sites available for filtering					
Parameters		Cancel			
No parameters	No parameters				
	Execute				
Responses		Response content type application/json			
Code	Description				
200	Returns list of Sites				
	Example Value Model				
	[{ "Site_Id": 39, "SiteMame": "Mozelle", "Longtude": -33.864353, "Laftude": 151.86333, "Region": "Sydney central-east" }]				
500	Error with processing the request				
	Example Value Model				
	{ "Message": "Error message from server", "RequestId": "b5e9816c-131d-41c2-b057-89391b3d9ele" }				

4. Click on Download.

Responses		Response content type	application/json	~
Curl curl -X GET "https://deta.airqu	wality.nsw.gov.au/api/Data/get_SiteDatails" -H "accept: application/json"			
Request URL				
https://data.airquality.nsw.gov	.au/api/Data/get_SiteDetails			
Server response				
Code	Details			
200	<pre>Response body [{ "Site_Id": 33, "SiteName": "FMOWICK", "Latitude": 151.242802, "Latitude": 151.242802, "Latitude": 151.250, "SiteName": "FMORELLE", "SiteName": FMORELLE", "SiteLat": 151.15093, "Latitude": -33.804353, "Latitude": -33.804353, "Latitude": -33.804353, "Latitude": -33.804353, "Kegion": "Sydney East" " }, "SiteLat": 101.1509, "Latitude": -33.70113, "Megion": "Sydney East" }, "SiteLat": 101.9071, "SiteLate": 100.9073, "Latitude": -33.3013, "SiteLate": -33.804354, "SiteLate": -33.804355, "Latitude": -33.804355, "Latitude": -33.804354, "SiteLate": -33.804355, "Latitude": -33.80435, "Latitude": -33.80435, "Latitude": -33.80435, "Latitude": -33.80435, "Latitude": -33.80435, "Latitude": -33.80435, "Latitude": -33.804, "Latitude": -33.8</pre>		Do	wnload
	Response headers content-encoding: grip content-type: application/json; charset=utf-8 dete: Sats, 30 May 2020 10:09:42 GWT server: Kestrel transfer-encoding: chunked vary: Accept-Encoding x-powered-by: ASP.NET			

5. Click on response.json.



6. Save response.json.

🗐 response_1590833383093.json - Notepad — 🗆 🛛 🔿
File Edit Format View Help
{ "Site_Id": 33, "SiteName": "RANDWICK", "Longitude": 151.242802, "Latitude": -33.931776, "Region": "Sydney East" }, { "Site_Id": 39,
iteName": "WYONG", "Longitude": 151.432414, "Latitude": -33.278966, "Region": "Central Coast" }, { "Site_Id": 287, "SiteName": "WALLSEND",
IAMWORTH", "Longitude": 150.914496, "Latitude": -31.109909, "Region": "North-west Slopes" }, { "Site_Id": 500, "SiteName": "WOLLONGONG",
', "Longitude": 150.766799, "Latitude": -33.795155, "Region": "Sydney North-west" }, { "Site_Id": 765, "SiteName": "VINEYARD", "Longitud€
[d": 2921, "SiteName": "ALBION PARK STH", "Longitude": 150.782437, "Latitude": -34.577804, "Region": "Illawarra" }, { "Site_Id": 333, "Si
'Site_Id": 4330, "SiteName": "BULGA", "Longitude": 151.03602, "Latitude": -32.648705, "Region": "Upper Hunter" }, { "Site_Id": 1650, "Sit
, { "Site_Id": 8330, "SiteName": "WARKWORTH", "Longitude": 151.02721, "Latitude": -32.57401, "Region": "Upper Hunter" }, { "Site_Id": 34
ast" }, { "Site_Id": 9880, "SiteName": "Tibooburra", "Longitude": 142.057, "Latitude": -29.4448, "Region": "Western LLS" }, { "Site_I(
'SiteName": "Cobar", "Longitude": 145.829, "Latitude": -31.4837, "Region": "Western LLS" }, { "Site_Id": 9877, "SiteName": "Condobolin",
).269, "Latitude": -31.0261, "Region": "North West LLS" }, { "Site_Id": 9711, "SiteName": "Hay", "Longitude": 144.829, "Latitude": -34.5
Western LLS" }, { "Site_Id": 92551, "SiteName": "Loddon Plains", "Longitude": 144.319, "Latitude": -36.4538, "Region": "North Central CMA"
2, "SiteName": "Walgett", "Longitude": 148.123, "Latitude": -30.0359, "Region": "North West LLS" }, { "Site_Id": 9507, "SiteName": "Walp∈
<pre>\LE", "Longitude": 151.661658, "Latitude": -30.50845, "Region": "Northern Tablelands" }, { "Site_Id": 1800, "SiteName": "ORANGE", "Longit</pre>
{ "Site_Id": 155, "SiteName": "ROUSE HILL", "Longitude": 150.903713, "Latitude": -33.682762, "Region": "Sydney North-west" }, { "Site_]
'Bushfire Monitoring" }, { "Site_Id": 450, "SiteName": "COFFS HARBOUR", "Longitude": 153.11808, "Latitude": -30.29831, "Region": "Bushfire [
L4185, "Latitude": -33.794246, "Region": "Sydney North-west" }, { "Site_Id": 10000000, "SiteName": "Central Coast", "Longitude": null, "L
ill, "Latitude": null, "Region": "Northern Tablelands" }, { "Site_Id": 80000000, "SiteName": "South-west slopes", "Longitude": null, "Lat
mme": "Upper Hunter - Singleton", "Longitude": null, "Latitude": null, "Region": "Upper Hunter - Singleton" }, { "Site_Id": 150000000, "Site
Save As X

Organise 🔻 🛛 N	iew folder					?
🗎 Documents		^ Name	^	Date modified	Туре	51
🕹 Downloads			N 3			
Music		~ <				>
File name	response_159083338309	3.json				~
Save as type	Text Documents (*.txt)					~

To import a .json file into Excel

- 1. Data/Get Data/From File/From JSON
- 2. Get response.json file.
- 3. Click on *To Table*.
- 4. Click on OK

5. Click on the double headed arrow to expand the record set

	ABC 123	Column1	† 1₽
--	------------	---------	-------------

- 6. Click OK
- 7. Click *Close & Load*. The data is imported into the Excel worksheet.

4.1.2 Extract Parameter data using the Air Quality Data API website

The following instructions outline how to Parameter data from the DPIE Azure Cloud Data Warehouse using the Air Quality API website. It is not possible to add parameters to filter the Parameter data.

Parame	ters
GET	/api/Data/get_ParameterDetails Retrieve Parameters available for filtering

1. Click on the **GET** button.

Parameters ×						
GET /api/Data/get_ParameterDetails Retrieve Parameters available for filtering						
Retrieve Parameters available for filtering						
Parameters		Try it out				
No parameters						
Responses		Response content type application/json v				
Code	Description					
200	Returns list of Parameters Example Value Model [

- 2. Click on *Try it out*.
- 3. Click on *Execute*.

Air Quality Application Programming Interface (API) User Guide

Parameters		\checkmark						
GET /api/Data/get_F	arameterDetails Retrieve Parameters available for filtering							
Retrieve Parameters available for filtering								
Parameters		Cancel						
No parameters								
	Execute							
Responses		Response content type application/json v						
Code	Description							
200	Returns list of Parameters Example Value Model [
	Error with processing the request Example Value Model { "Measage": "Error measage from server", "RequestId": "b5e9816c-131d-41c2-b057-83391b3d9e1e" } }							

4. Click on Download.

Responses		Response content type application/json V
Cord		
curl -X GET "http	ps://data.wirquality.now.gov.au/opi/Data/get_ParameterDetails" -H "accept: application/json"	
Request URL		
https://dete.eir	quality.nsw.gov.au/api/Data/get_ParameterDetails	
Server response		
Code	Details	
200	Response body	
	<pre>[</pre>	Download
	Response headers	
	content-encoding: grip content-type: spylication/json; charset=utf-8 date: Sat, 30 May 2020 10:19:45 GMT server: Kestrei transfer-encoding: chunked vary: Accept-Encoding x-powered-by: ASP.NET	

5. Click on response.json.



6. Save response.json.

response 1590833986673.ison - Notepad				- 🗆 ×
File Edit Format View Help				
File Edit Format View Help [{ "ParameterCode": "AQI", "Para ount", "UnitsDescription": "number of oxide", "Units": "count", "UnitsDe iption": "Carbon monoxide", "Units": "ParameterCode": "CO", "ParameterDes ge" }, { "ParameterCode": "CO", "Carbon monoxide", "Units": "ppm", tion": "10^-4 m^-11", "Category": "Ave "Monthly", "Frequency": "Hourly avera age" }, { "ParameterCode": "NEPH", "ParameterDescription": "Ammonia", hm", "UnitsDescription": "Parts per h ion": "parts per hundred million", "C ", "Category": "Exceedences", "Soub SubCategory": "Daily", "Frequency": "Ho ily", "Frequency": "An rolling average	<pre>meterDescription": "AQI", "Uni 'exceedances", "Category": "Ex 'scription": "number of exceedance 'ppm", "UnitsDescription": "pr cription": "Carbon monoxide", "ParameterDescription": "Carbon of 'UnitsDescription": "Parts per of 'UnitsDescription": "Darts per of "BarameterDescription": "Neph "Units": "pphm", "UnitsDescri undred million", "Category": 'ategory": "Maximums", "SubCat Category": "Monthly", "Freque '24h average derived from 1h average 'a derived from 1h average" }, { "Parameter e derived from 1h average" }, { "Parametrer e derived from 1h average" }, { "Parametrerererererererererererererererererer</pre>	<pre>its": "index", "UnitsDescriptic kceedences", "SubCategory": "Da ss", "Category": "Exceedences", arts per million", "Category": "Da monxide", "Units": "ppm", "UnitsDescript monxide", "Units": "ppm", " and the state of the state of the state of the state "ParameterDescription": elometer ", "Units": "index", ription": "parts per hundred milli "Averages", "SubCategory": "Dai ggory": "Annual", "Frequency": "Dai ggory": "Hourly average" }, { "ParameterCode": "No erCode": "NO2", "ParameterDescription": "Dai "Date the state of the state of the state "ParameterCode": "OZONE",</pre>	<pre>n": "index", "Category" illy", "Frequency": "8h "SubCategory": "Monthl "Averages", "SubCategor ion": "parts per million", UnitsDescription": "parts ", "SubCategory": "Mont fee" }, { "ParameterCod "Nephelometer ", "Units "UnitsDescription": "air on", "Category": "Avera ()", "Frequency": "Avera "Hourly average" }, { "ParameterCode": "NO2", IO2", "ParameterDescript iption": "Nitrogen Dioxide "ParameterDescription": "C"</pre>	: "Site AQI", "SubCatt rolling average derived i y", "Frequency": "Hour "Category": "Maxinum er million", "Categor hly", "Frequency": "Ht le": "NEPH", "Parameter "; "count", "UnitsDesc quality index", "Cat ges", "SubCategory": " "ParameterDescription": " "ParameterDescription": " tion": "Nitrogen Dioxide", ", "Units": count
", "Frequency": "Hourly average") }, { "ParameterCode": "OZONE".	Save As	"D . D		<pre>"" scription": "air</pre>
<pre>h rolling average derived from 1h aver s", "SubCategory": "Monthly", "F</pre>	$\leftarrow \rightarrow$ \checkmark \bigstar \bigstar This PC \Rightarrow Downloads		✓ ひ Search Downloads	<pre></pre>
", "Frequency": "Hourly average"] rom 1h average" }. { "ParameterCo	Organise 🔻 New folder			 cription": "parts per cubic meter'
ency": "Hourly average" }, { "Par	Documents	Name	Date modified Type	<pre>icrogram per cubi Category": "Annus</pre>
ode": "PM10", "ParameterDescriptior	Downloads Music	K1 - 74 - 14 - 1		> ences", "SubCa
<pre>nits": "index", "UnitsDescription": , "Units": "µg/m³", "UnitsDescri </pre>	File name: response_1590833986673.jsc	on and a second s		v cy": "24h average
"ParameterDescription": "PM2.5",	Save as type: Text Documents (*.txt)			SubCategory": "/
cription": "number of exceedances", "UnitsDescription": "number of exceed				average" }, { { "Parameter(
<pre>uality index", "Category": "Indexes QI", "SubCategory": "Hourly", "F ourly average" }. { "ParameterCode</pre>	∧ Hide Folders	Encoding: UTF-8	✓ Save Ca	"ParameterDescr Units": "index",
: "SO2", "ParameterDescription": "Sul	phur Dioxide", "Units": "pphm"	", "UnitsDescription": "parts p	per hundred million", "C	ategory": "Averages",

Importing a .json file into Excel

See: To import a .json file into Excel.

4.1.3 Extract Current Observed air quality data using the Air Quality Data API website.

The following instructions outline how to extract current observed real time hourly air quality data from the DPIE Azure Cloud Data Warehouse using the Air Quality API website.

Observations	
POST /api/Data/get_Observations Retrieve Observation data	

1. Click on the **POST** button.

Observations		\checkmark					
POST /api/Data/get_Obser	POST /api/Data/get_Observations Retrieve Observation data						
Query Observation data based on provided filter(s). If none is provided, latest data for all available Sites will be return							
Parameters		Try it out					
Name	Description						
request	Filters to query data						
(body)	Example Value Model						
	<pre>{ "Parameter content type " " " " " " " " " " " " " "</pre>						
	application/json 🗸						

Note the Observations API has parameters (in the black rectangle).

The query is an example.

The query extracts PM10 hourly averages for 24 hour rolling average derived from 1 hour average for Aberdeen (336), Bulga (4330), Camberwell (2330), Jerry's Plains (7330), Maison Dieu (3330), Merriwa (329), Mount Thorley (5330) for the previous 2 days.

- 2. Click on *Try it out*
- Edit the query in the Example value field. To get current observed real time hourly data the body command is equal to empty quotations (body = "",).
- 4. Click on Execute

Air Quality Application Programming Interface (API) User Guide

Observations	servations ×								
POST /api/Data/get_Observations Retrieve	Observation data								
Query Observation data based on provided filter(s). If none is provided, latest data for all available Sites will be return									
Parameters		Cancel							
Name	Description								
request	Filters to query data								
(body)	Example Value Model								
	x === }								
	Cancel Parameter content type application/json								
	Execute								

5. Click on Download.

Responses	Respon	se content type	application/json	~
Curl curl -X POST "https://	/deta.airquality.nsw.gov.au/api/Data/get_Observations" -H "accept: application/json" -H "Content-Type: application/json" -d '	'{ \"\"}"		
Request URL				
https://data.airquali1	ty.nsw.gov.au/api/Data/get_Observations			
Server response				
Code	Details			
200	Response body			
	<pre>[{ "Site_Id": 33, "Parameter": { "Thermeter": { "Thermeter: { "Thermeter: { "Thermeter: { "Thermeter: { "Thermeter: { "Thermeter: { "Therm</pre>		Da	beolmw

6. Click on response.json.



response_159083....json ^

7. Save response.json.

	respons	se_159083	4489165	(2).json - Notepad	
File	Edit	Format	View	Help	

- 🗆 ×

File Edit Format View Help					
<pre>[{ "Site_Id": 33, "Parameter": "Hourly average" }, "Date": " "Hitric Oxide", "Units": "ophm", "Hour": 19, "HourDescription": "UnitsDescription": "parts per hundree 30", "Hour": 19, "HourDescription": "air of "HourDescription": "6pm - 7 pm", "UnitsDescription": "6fgree", "Cat ". "UnitsDescription": "6gree", "Cat ". 0.1 }, { "Site_Id": 33, "Pa bCategory": "Hourly average" }, "Date" ". "UnitsDescription": "Garbon monoxide", ency": "Hourly average" }, "Date" "HourDescription": "Garbon monoxide", "." "Units": "Garbon monoxide", "." "Units": "idex", "Units" "HourDescription": "6 pm - 7 pm", "Value": 2.1 : "air quality index", "Category": scription": "6 pm - 7 pm", "Value": 2.1 </pre>	<pre>{</pre>	"ParameterDescription": "A HourDescription": "6 pm - 7 pm" per hundred million", "Cat }, { "Site_Id": 33, "Pa rerages", "SubCategory": "H 17.3 }, { "Site_Id": 33, "Indexes", "SubCategory": Id": 33, "Parameter": { "Category": "Hourly", "Freque ": "IEMP", "ParameterDescr se": "2020-05-30", "Hour": 19 Jescription": "air quality index "HourDescription": "6 pm - 7 xx", "Category": "Indexes", 39, "Parameter": { "ParameterCode" "Hourly", "Frequency": "2 "Hourly", "Frequency": "2	QI", "Units": ' , "Value": 68.3 egory: "Averages", rameter": { "Pa- ourly", "Freque "ParameterCode": "Fn- ncy": "Hourly average iption": "Temperatur , "HourDescriptic "HourDescriptic", "Category": pm", "Value": 0. "SubCategory": rameterCode": "NO2", "Frequency": "Hour 4h rolling average of the second (Sate Sate Sate Sate Sate Sate Sate Sate	<pre>index", "Uni }, { "Site_IC "SubCategory unameterCode": "02 marameterCode": "02 marameterCode" "ParameterCode" quency": "Hourly", "Para e", "Units", "Indexes", "Onit "Indexes", "Site "Hourly", " "Indexes", "Site "Hourly", " "Indexes", 1 "meterDescription" lerived from 1h av "Dataset of the set of the set "Dataset of the set of the set "Dataset of the set of the set of the set "Dataset of the set of t</pre>	tsDescription": "ir ": "Alourly", " ONE", "Paramete verage derived fr : "PM10", "Par average derived fr : "PM10", "Par average" }, "extra strategister : "2020-85-30", "Onits! ": "2020-85-30", "Units! ": "2020-85-30", "Units! "subCategory": "Hourl Id": 39, "Paras "Frequency": "Hourl) bescription": "Nitce "PM10", "Date": "2020-6 : "PM10", "Date": "2020-6 : "PM10", "Date": "Site SubCategory", "Date": "Site SubCategory", "Date": "Site SubCategory", "Date": "Site SubCategory, "Date": "Site SubCategory, "Date": Site SubCategory, "Date": "Site SubCategory, "Date": "Site SubCategory, "Date": Site Site Site Site Site Site Site Site Site Site Site Site Site Site Site Site Site
<pre>{ "Site_Id": 39, "Parameter": {</pre>					"UnitsDescript
s", "SubCategory": "Hourly",	$\leftarrow \rightarrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \downarrow $ This PC > Downlo	ads	V O Search D	ownloads	/ m - 7 pm", "\
: "WSP", "ParameterDescription":	0 i N (1)			0	: "Averages",
5-30", "Hour": 19, "HourDescript	Organise V New folder			8== •	NEPH", "Pa
, "UnitsDescription": "parts per	Documents	 Name 	Date modified	Туре	<pre>s^ average" },</pre>
"HourDescription": "6 pm - 7 pm",	Downloads				terDescription":
"Units": "°", "UnitsDescription":	Downloads	11 M			"], "Date
", "Value": 16.6 }, { "Site Id	h Music	v <		· · · · · · · · · · · · · · · · · · ·) ", "Units
Category": "Hourly", "Frequency":	File name: response 15908344891	65 (2).ison			<pre>Value": 26 },</pre>
eter": {					million",
"Frequency": "Hourly average" },	Save as type: Text Documents (*.txt)				"Site Id": 107.
rDescription": "Nitric Oxide", "U					"SubCategory":
ate": "2020-05-30", "Hour": 19.					ameterCode": "02
"Units": "pphm", "UnitsDescription					ency": "4h rolli
}, "Date": "2020-05-30", "Hour":	∧ Hide Folders	Encoding: UTF-8	✓ Sa	ve Cancel	"ParameterCoc
M10", "Units": "index", "Unit					equency": "Hour]

8. Import response.json into Excel (screenshot below)

1	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Column1.Site	_Id Column1.Pa	ra Column1.Paramet	t Column1.Pa	r: Column1.Parameter.U	Column1.Paramete	Column1.Par	a Column1.Pa					Column1.AirQualityCategory	Column1.DeterminingPollutant
2		33 AQC	AQC	category	category	Hourly average	Site AQC	Hourly	2021-07-16	18	5 pm - 6 pm		GOOD	PM2.5
3		33 HUMID	Humidity	%	percent	Hourly average	Averages	Hourly	2021-07-16	18	5 pm - 6 pm	44.726		
4		33 NEPH	Nephelometer	10^-4 m^-1	10^-4 m^-1	Hourly average	Averages	Hourly	2021-07-16	18	5 pm - 6 pm	0.036		
5		33 NO	Nitric Oxide	pphm	parts per hundred mill	Hourly average	Averages	Hourly	2021-07-16	18	5 pm - 6 pm	0.012603		
6		33 NO2	Nitrogen Dioxide	pphm	parts per hundred mill	i Hourly average	Averages	Hourly	2021-07-16	18	5 pm - 6 pm	0.353501	GOOD	
7		33 OZONE	Ozone	pphm	parts per hundred mill	4h rolling average	Averages	Hourly	2021-07-16	18	5 pm - 6 pm	2.634975	GOOD	
8		33 OZONE	Ozone	pphm	parts per hundred mill	i Hourly average	Averages	Hourly	2021-07-16	18	5 pm - 6 pm	2.617125	GOOD	
9		33 PM10	PM10	µg/m³	microgram per cubic m	24h rolling average	Averages	Hourly	2021-07-16	18	5 pm - 6 pm	8.014125		
10		33 PM10	PM10	µg/m³	microgram per cubic m	Hourly average	Averages	Hourly	2021-07-16	18	5 pm - 6 pm	11.693	GOOD	
11		33 PM2.5	PM2.5	µg/m³	microgram per cubic m	24h rolling average	Averages	Hourly	2021-07-16	18	5 pm - 6 pm	3.850783		
12		33 PM2.5	PM2.5	µg/m³	microgram per cubic m	Hourly average	Averages	Hourly	2021-07-16	18	5 pm - 6 pm	6.634	GOOD	
13		33 SD1	Wind Direction Si	•	degree	Hourly average	Averages	Hourly	2021-07-16	18	5 pm - 6 pm	23.049		
14		33 SO2	Sulphur Dioxide	pphm	parts per hundred mill	Hourly average	Averages	Hourly	2021-07-16	18	5 pm - 6 pm	0.02479	GOOD	
15		33 TEMP	Temperature	°C	degree Celsius	Hourly average	Averages	Hourly	2021-07-16	18	5 pm - 6 pm	14.553		
16		33 WDR	Wind Direction (1	(*	degree	Hourly average	Averages	Hourly	2021-07-16	18	5 pm - 6 pm	287.177		
17		33 WSP	Wind Speed (10m	m/s	meter per second	Hourly average	Averages	Hourly	2021-07-16	18	5 pm - 6 pm	4,555		

Importing a .json file into Excel

See: To import a .json file into Excel.

4.1.4 Extract Historical air quality data using the Air Quality Data API website.

The following instructions outline how to extract historical air quality data from the DPIE Azure Cloud Data Warehouse using the Air Quality API website.



1. Click on the **POST** button.

Observations		\checkmark
POST /api/Data/get_Observ	vations Retrieve Observation data	
Query Observation data based on provid	ied filter(s). If none is provided, latest data for all available Sites will be return	
Parameters		Try it out
Name	Description	
request	Filters to query data	
(body)	Example Value Model	
	<pre>{ "Parameters": ["Parameters": ["Parameters": ["Sites": [336, 435, 435, 2350, 320, 320, 320, 320, 320, 320, 320, 320, 300, "EndDate": "2000-06-20", "EndDate": "2000-06-30", "Categories": ["founty" "founty" "founty" "founty" "founty" "routy" "routy" "founty" "Townorts": ["parameter content type application/json </pre>	

Note: the Observations API has parameters. (in the black rectangle) The query is an example of extracting historical air quality data.

The query extracts PM10 hourly averages for 24 hour rolling average derived from 1 hour average for Aberdeen (*336*), Bulga (*4330*), Camberwell (*2330*), Jerry's Plains (*7330*), Maison Dieu (*3330*), Merriwa (*329*), Mount Thorley (*5330*) for the previous 2 days.

Note: the parameter body query command has only single quotes. Excel Power query has double quotes.

- 2. Click on Try it out
- 3. Click on Execute.

Air Quality Application Programming Interface (API) User Guide

Observations ×						
POST /api/Data/get_Observations Retrieve	Observation data					
Query Observation data based on provided filter(s). If none is provided, latest data for all available Sites will be return						
Parameters		Cancel				
Name	Description					
request	Filters to query data					
(body)	Example Value Model					
	<pre>{ "Parameters": ["PATS0" "Attorn [states": [states": [states": "2020-05-20", "fandbates": "2020-05-30", "categories": ["Averages" "Averages" "Averages" "Hourty" </pre>	•				
	Cancel Parameter content type application/json					
	Execute					

4. Click on Download.

Responses		Response content type	application/json ~
Curl curl -X POST "htt	ps://deta.eirquality.nsw.gov.au/api/Data/get_Observations" ~H "accept: application/json	" -H "Content-Type: application/json" -d "{ \"\"}"	
Request URL	uality.nsw.gov.au/api/Data/get_Observations		
Server response			
Code	Details		
200	Response body [{ ""Site_Id": 33, ""Parameter": { }		Â
	"Parameter-Code": "AQT", "Parameter-Code": "AQT", "UnitalBacoription": "index", "UnitalBacoription": "index", "Subctategory": "Site AQT", "Frequency": "Hourly average" } "bacee": "3030-03-30", "Mourle-Code: "Site AQT", "Value": Site AQT", "Value": Site AQT", }		
	<pre>{ "Site_Id": 33, "Parameter": { "Parameter": { "Parameter".code": "AQI", "Parameter Description": "AQI (based on rolling 24h PM)", "UnitalEscription": "index", "UnitalEscription": "index", "Category:": "Site AQI", "SideStegory:": "Nourly", "SideStegory:": "Nourly", "Encourse:" "Nourly", "SideStegory:": "Nourly", "Encourse:" "Nourly", "SideStegory:": "Nourly", "SideStegory:": "Nourly", "SideStegory:": "Nourly", "SideStegory:": "Nourly", "Encourse:" "Nourly", "SideStegory:": "Nourly", "Encourse:" "Nourly", "SideStegory:": "Nourly", "SideStegory:": "Nourly", "Encourse:" "Nourly", "Encourse:" "Nourly", "Encourse:" "Nourly", "SideStegory:": "Nourly", "Encourse:" "Nourly", "Enco</pre>		Download
	Response headers		
	content-encoding: grip content-type: splication/jaon; charact=utf=8 date: Sat, 39 way 2020 10:41:00 GMT arrier1 Keatrel transfer-encoding; x-powerd-by: ASP.NET		

5. Click on response.json.



response_159083....json ^

6. Save response.json.

{ "Site_Id": 39, rarameter...,
s", "SubCategory": "Hounly",
:"WSP", "ParameterDescription":
5-30", "Hour": 19, "HourDescript
"HourDescription": "parts per
"HourDescription": "6 pm - 7 pm",
"Units": "o", "UnitsDescription":
", "Value": 16.6 }, { "Site_Id
Category": "Hounly", "Frequency":
eter": { "ParameterCode": "CO",
"Frequency": "Hounly average" },
rDescription": "Nitric Oxide", "Units": "2020-05-30", "Hour": 19,
"Units": "2020-05-30", "Hour":
M10", "Units": "index", "Units" terDescription": " }, "Date)", "Units "Value": 26 }, million", "Site_Id": 107, "SubCategory": " ameterCode": "Oi consu": "Oi consultation": "Oi co h Music File name: response_1590835267504.json Save as type: Text Documents (*.txt) ency": "4h rolli "ParameterCoc requency": "Hour] Encoding: UTF-8 V Save Cancel ∧ Hide Folders

7. Import response.json into Excel (screenshot below)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Column1.Site_Id (olumn1.Pa	Column1.P	a Column1.P	Column1.Parameter.UnitsD	Column1.P	a Column1.Para	r Column1.Parameter.Frequency	Column1.Date	Column1. Colu	ımn1.HourDe	e Column1.Value	Column1.AirQuality	Column1.Determinir
2	329 F	M10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	24h rolling average derived from 1h average	2020-05-29	1 12 a	m - 1 am	10.13775		
3	336 F	M10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	24h rolling average derived from 1h average	2020-05-29	1 12 a	m - 1 am	11.403333		
4	2330 F	M10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	24h rolling average derived from 1h average	2020-05-29	1 12 a	m - 1 am	20.156625		
5	3330 F	M10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	24h rolling average derived from 1h average	2020-05-29	1 12 a	m - 1 am	18.515125		
6	4330 F	M10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	24h rolling average derived from 1h average	2020-05-29	1 12 a	m - 1 am	12.319708		
7	5330 F	M10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	24h rolling average derived from 1h average	2020-05-29	1 12 a	m - 1 am	23.92475		
8	7330 F	M10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	24h rolling average derived from 1h average	2020-05-29	1 12 a	m - 1 am	13.20275		
9	329 F	M10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	24h rolling average derived from 1h average	2020-05-29	2 1 am	n - 2 am	9.9615		
10	336 F	M10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	24h rolling average derived from 1h average	2020-05-29	2 1 am	n - 2 am	11.454292		
11	2330 F	M10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	24h rolling average derived from 1h average	2020-05-29	2 1 am	n - 2 am	20.32025		
12	3330 F	M10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	24h rolling average derived from 1h average	2020-05-29	2 1 am	n - 2 am	18.785333		
13	4330 F	M10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	24h rolling average derived from 1h average	2020-05-29	2 1 am	n - 2 am	12.373		
14	5330 F	M10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	24h rolling average derived from 1h average	2020-05-29	2 1 an	n - 2 am	22.4385		
15	7330 F	M10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	24h rolling average derived from 1h average	2020-05-29	2 1 am	n - 2 am	13.223667		
16	329 F	M10	PM10	ug/m³	microgram per cubic meter	Averages	Hourly	24h rolling average derived from 1h average	2020-05-29	3 2 an	n - 3 am	9.859792		

Importing a .json file into Excel

See: To import a .json file into Excel.

5. Search and retrieve air quality data using Excel and API

The DPIE Air Quality API data service can be used by a variety of third-party software applications as identified in Figure 1, to search and retrieve DPIE air quality data. The following instructions outline how to deploy the API using Excel Power Query to search and retrieve air quality and related data stored in the DPIE Azure Cloud Data Warehouse. The remainder of Section 5 is in the downloadable air-quality-api-excel-power-query.xlsx Excel file, if you do not want to go through the set-up.

5.1 Setting up a Power Query in Excel using the API

The API provides three data sets:

- Site details data
- Parameter details data
- Observational data. The observational data can be extracted as current real-time hourly data or as historical data by querying data through the body command in the power query.

The Site details and Parameter details can also be queried through the body command in the Power Query in Excel. However, since this is primarily static data it is easier to search using the filter in Excel. Using the body command to filter Site and Parameter details, will not be considered in this document.

Before extracting air quality data, using Excel and the API

- 1. Open a blank workbook in Excel.
- 2. Go to File, Options and Trust Centre.
- 3. Click on the Trust Centre Settings button.
- Set the Trust Centre settings by clicking the radio button next to Enable all Data Connections (not recommended) as shown in Figure 4. This enables Excel to connect to the API.





Excel – Enable all Data Connections (not recommended)

5.1.1 Extract Monitoring Site data using Excel and API

The following instructions outline how to extract monitoring site data from the DPIE Azure Cloud Data Warehouse using Excel Power Query and the DPIE Air Quality API.

- 1. Open a blank Excel workbook.
- 2. Click on Data on the menu bar
- 3. Select Get Data.
- 4. Select From Other Sources.
- 5. Select then Blank Query. See Figure 5 below.

	AutoSave 💽 017 🔒 🐬 12 - 12 - 12 - 12 - 12 - 12 - 12 - 12												Book1 - Ex	cel
	File	Home	Insert	Page	Layout	Forn	nulas Data	Rev	iew View	Dev	eloper	Help	HPE Conten	t Manager
¢	Get Data V	From Text/C From Web From Table/F	SV [Range	👌 Rece	ent Sources ing Connections Refresh All - & Ed Queries			Queries Properti Edit Link	& Connection es ks	s 2↓ Z↓	Z A A Z Sort	Filter	Clear Reapply Advanced	Text to Column
	Fre	om <u>F</u> ile		•	£		Quen		incetions					
	Fre	om <u>D</u> atabase		×	<i>J</i> .x									
	Fre	om <u>A</u> zure		×	В		С		D		E	Ξ	F	F
	Fre	om Online S <u>e</u>	rvices	×										
	Fre	om <u>O</u> ther So	urces	Þ		From Ta	able/Range							
	G 60	mbine <u>Q</u> ueri	es	×		From <u>W</u>	eb							
	Launch Power Query Editor			From Microsoft Query										
- G	C Data	<u>C</u> atalog Searc	n Jueries			From St	arePoint List							
	👌 Data	Source <u>S</u> ettin	gs		5	FTOIL SI	larer olift <u>L</u> ist							
	🖹 Quen	y O <u>p</u> tions				From <u>O</u> l	Data Feed							
	8					From <u>H</u> a	adoop File (HDF	S)						
	9					From <u>A</u> o	ctive Directory							
	10				ES	From M	icrosoft <u>E</u> xchar	ge						
	11				•	From O <u>I</u>	<u>D</u> BC							
	12				 []]	From O	LEDR							
	13	3												
	14 Blank Query													
	15													
Figure 5	gure 5 Excel – Blan				uery									

- 6. The Power Query Editor opens.
- 7. Click on Advanced Editor.

X ∎ -	Query1 - I	Power Query Editor			
•	Home	Transform Add	d Column View		
Close & Load *	Refresh Preview	Advanced Editor	Image: Constraint of the second se		
Close		Query			_
ies ×	×	fx	Query1	0	ier
Queri			let Source = "" in Source ✓ No syntax errors have been detected. Dore	Cancel	OP me Uer Prc Sc

- 8. Remove paragraph markers from this document by going to *File/Options/Display* and unselecting *Show all formatting marks*. (you need to do this since the Power Query Editor cannot interpret paragraph markers).
- 9. Copy and paste the code below into Query 1. Copy the code between the horizontal lines below.

let

```
Source =
Json.Document(Web.Contents("https://data.airquality.nsw.gov.au/api/D
ata/get_SiteDetails")),
#"Converted to Table" = Table.FromList(Source,
Splitter.SplitByNothing(), null, null, ExtraValues.Error),
#"Expanded Column1" = Table.ExpandRecordColumn(#"Converted to
Table", "Column1", {"Site_Id", "SiteName", "Longitude", "Latitude",
"Region"}, {"Column1.Site_Id", "Column1.SiteName",
"Column1.Longitude", "Column1.Latitude", "Column1.Region"})
in
```

#"Expanded Column1"

10. Click on Done.

11. The Site details data are extracted into the Excel Power Query Editor.

File		Home Transform Add	Column View			
Close & Load • Close	k F	Properties ☐ Advanced Editor Refresh review → Manage → Query	Manage Columns ▼ Reduce Rows ▼ 2↓ 素↓ Col Sort	Data Type: Any ▼ iplit Group umn ▼ By ¹ → 2 Replace Valu Transform	w as Headers • les • Pa	Manage rameters Data source arameters Data Sources
>	×	✓ f _X = Table.So	rt(#"Expanded Column1",{	{"Column1.Site_Id", Order	Ascending}})	
iries		ABC 123 Column1.Site_Id	ABC 123 Column1.SiteName	ABC Column1.Longitude	ABC 123 Column1.Latitude	ABC 123 Column1.Region
Que	1	33	RANDWICK	151.24278	-33.93175	Sydney East
	2	39	ROZELLE	151.16395	-33.86433	Sydney East
	3	61	BRADFIELD HIGHWAY	151.21109	-33.84327	Roadside Monitoring
	4	70	LINDFIELD	151.1509	-33.78113	Sydney East
	5	107	LIVERPOOL	150.90727	-33.93132	Sydney South-west
	6	113	MACQUARIE PARK	151.1178	-33.76524	Sydney East
	7	155	ROUSE HILL	150.90366	-33.68275	Sydney North-west
	8	171	BRINGELLY	150.76192	-33.91766	Sydney South-west
	9	190	CHULLORA	151.0461	-33.89156	Sydney East
	10	205	EARLWOOD	151.13577	-33.91619	Sydney East
	11	259	WYONG	151.43239	-33.27891	Central Coast
	12	264	MORISSET	151.55256	-33.107815	Lake Macquarie
	13	287	WALLSEND	151.67021	-32.89435	Lower Hunter
	14	294	CARRINGTON	151.763302	-32.909718	Newcastle Local
	15	295	STOCKTON	151.78429	-32.90201	Newcastle Local
	16	300	NEWCASTLE	151.75965	-32.9312	Lower Hunter
	17	304	MAYFIELD	151.72842	-32.88488	Newcastle Local
	18	322	BERESFIELD	151.66099	-32.79677	Lower Hunter
	19	329	MERRIWA	150.45824	-32.12665	Upper Hunter
	20	330	SINGLETON	151.17707	-32.55734	Upper Hunter
	21	333	MUSWELLBROOK	150.88563	-32.27152	Upper Hunter
	22	336	ABERDEEN	150.88415	-32.17718	Upper Hunter
	23	340	TAMWORTH	150.91451	-31.1099	North-west Slopes
	24	380	GUNNEDAH	150.26069	-30.98178	North-west Slopes
	25 390		NARRABRI	149.82932	-30.31842	North-west Slopes
	26	450	COFFS HARBOUR	153.11811	-30.29828	Mid-North Coast

🚺 | 🗢 | SiteDetails - Power Query Editor

12. Click on Close & Load. The Site Details data are extracted into an Excel worksheet.

	А	В	с	D	E
1	Column1.Site_Id	Column1.SiteName	Column1.Longit	Column1.Lat	Column1.Region
2	33	RANDWICK	151.24278	-33.93175	Sydney East
3	39	ROZELLE	151.16395	-33.86433	Sydney East
4	61	BRADFIELD HIGHWAY	151.21109	-33.84327	Roadside Monitoring
5	70	LINDFIELD	151.1509	-33.78113	Sydney East
6	107	LIVERPOOL	150.90727	-33.93132	Sydney South-west
7	113	MACQUARIE PARK	151.1178	-33.76524	Sydney East
8	155	ROUSE HILL	150.90366	-33.68275	Sydney North-west
9	171	BRINGELLY	150.76192	-33.91766	Sydney South-west
10	190	CHULLORA	151.0461	-33.89156	Sydney East
11	206	EARLWOOD	151.13577	-33.91619	Sydney East
12	259	WYONG	151.43239	-33.27891	Central Coast
13	264	MORISSET	151.55257	-33.10801	Lake Macquarie
14	287	WALLSEND	151.67021	-32.89435	Lower Hunter
15	294	CARRINGTON	151.76335	-32.90964	Newcastle Local
16	295	STOCKTON	151.78429	-32.90201	Newcastle Local
17	300	NEWCASTLE	151.75965	-32.9312	Lower Hunter
18	304	MAYFIELD	151.72842	-32.88488	Newcastle Local
19	322	BERESFIELD	151.66099	-32.79677	Lower Hunter
20	329	MERRIWA	150.45824	-32.12665	Upper Hunter
21	330	SINGLETON	151.17707	-32.55734	Upper Hunter
22	333	MUSWELLBROOK	150.88563	-32.27152	Upper Hunter
23	336	ABERDEEN	150.88415	-32.17718	Upper Hunter
24	340	TAMWORTH	150.91451	-31.1099	North-west Slopes
25	380	GUNNEDAH	150.26069	-30.98178	North-west Slopes
26	390	NARRABRI	149.82932	-30.31842	North-west Slopes
27	450	COFFS HARBOUR	153.11811	-30.29828	Mid-North Coast
28	500	WOLLONGONG	150.88733	-34.41706	Illawarra

Air Quality Application Programming Interface (API) User Guide

13. Change the name of Query1 and Sheet1 to *SiteDetails*. This is the name in the airquality-api-excel-power-query.xlsx Excel workbook.

	1	2	3	4	5	
1	Column1.Site_Id Colu	mn1.SiteName	Column1.Longitude	Column1.Latitude	Column1.Region	Queries & Connections
2	33 RANI	DWICK	151.24278	-33.93175	Sydney East	Question Commutitions
3	39 ROZE	ELLE	151.16395	-33.86433	Sydney East	Queries Connections
4	61 BRAD	DFIELD HIGHWAY	151.21109	-33.84327	Roadside Monitoring	1 query
5	70 LIND	FIELD	151.1509	-33.78113	Sydney East	CharD and the
6	107 LIVER	RPOOL	150.90727	-33.93132	Sydney South-west	
7	113 MAC	QUARIE PARK	151.1178	-33.76524	Sydney East	126 rows loaded.
8	155 ROU	SE HILL	150.90366	-33.68275	Sydney North-west	
9	171 BRIN	IGELLY	150.76192	-33.91766	Sydney South-west	
10	190 CHUI	LLORA	151.0461	-33.89156	Sydney East	
11	206 EARL	WOOD	151.13577	-33.91619	Sydney East	
12	259 WYO	NG	151.43239	-33.27891	Central Coast	
13	264 MOR	ISSET	151.55257	-33.10801	Lake Macquarie	
14	287 WAL	LSEND	151.67021	-32.89435	Lower Hunter	
15	294 CARF	RINGTON	151.76335	-32.90964	Newcastle Local	
16	295 STOC	CKTON	151.78429	-32.90201	Newcastle Local	
17	300 NEW	CASTLE	151.75965	-32.9312	Lower Hunter	
18	304 MAY	FIELD	151.72842	-32.88488	Newcastle Local	
19	322 BERE	SFIELD	151.66099	-32.79677	Lower Hunter	
20	329 MERI	RIWA	150.45824	-32.12665	Upper Hunter	
21	330 SING	LETON	151.17707	-32.55734	Upper Hunter	
22	333 MUS	WELLBROOK	150.88563	-32.27152	Upper Hunter	
23	336 ABEF	RDEEN	150.88415	-32.17718	Upper Hunter	
24	340 TAM	WORTH	150.91451	-31.1099	North-west Slopes	
25	380 GUN	NEDAH	150.26069	-30.98178	North-west Slopes	
26	390 NARI	RABRI	149.82932	-30.31842	North-west Slopes	
27	450 COFF	FS HARBOUR	153.11811	-30.29828	Mid-North Coast	
28	500 WOL	LONGONG	150.88733	-34.41706	Illawarra	
29	526 KEMI	BLA GRANGE	150.81913	-34.47408	Illawarra	
30	573 RICH	MOND	150.74731	-33.61641	Sydney North-west	
					· · · ·	

14. To refresh the data, go to Data/RefreshAll/Refresh.

Auto	oSave 💽 🖁 🖌	9• (?- ÷		D	PIEExcelAPI.	xlsx - Excel			Tat
File	Home Insert	Page Layout Formu	as Data	a Review	View	Developer	Help H	HPE Content Mar	nager D
Get Data	From From From Text/CSV Web F	m Table/ Recent Existing Existing	Refresh All -	Queries & Co Properties Co	onnections	E Stocks	Geography	$ \begin{array}{c} \bullet \\ \bullet \\ \bullet \\ \hline \end{array} \\ \begin{array}{c} \bullet \\ \bullet \\ \hline \end{array} \\ \begin{array}{c} \bullet \\ \bullet \\ \bullet \\ \bullet \\ \end{array} \\ \begin{array}{c} \bullet \\ \bullet \\ \bullet \\ \bullet \\ \bullet \\ \bullet \\ \end{array} \\ \begin{array}{c} \bullet \\ \bullet $	Filter F
	Get & Tra	nsform Data	Re Re	fresh <u>A</u> ll		D	ata Types		Sort & Filter
E32	- i ×	√ f _x	🔁 Be	fresh					
1	А	В	Re	fresh Status Incel Refresh			D		E
1 0	olumn1.Site_Id	Column1.SiteName		nnection Droner	tier le	Column	1.Latitude	Column1.Re	gion
2	33	RANDWICK		10	1.242802	2 -	33.931776	Sydney East	
3	39	ROZELLE		15	1.163933	3 -	33.864353	Sydney East	
4	61	BRADFIELD HIGHWAY		15	1.211422	2 -	33.843426	Roadside Mo	onitoring
5	70	LINDFIELD			151.1509	Э	-33.78113	Sydney East	
6	107	LIVERPOOL			150.9073	3	-33.9313	Sydney South	n-west

5.1.2 Extract Parameter data using Excel and API

The following instructions outline how to extract Parameter data from the DPIE Azure Cloud Data Warehouse using Excel Power Query and the DPIE API.

- 1. Create a new blank query by following the procedure as described in section 5.1.1, up to after clicking on the Advanced editor.
- 2. Replace the code in Query 1 with the code below: (remove paragraph markers)
- 3. Copy the code between the horizontal lines below.

let

```
Source =
Json.Document(Web.Contents("https://data.airquality.nsw.gov.au/api/D
ata/get_ParameterDetails")),
#"Converted to Table" = Table.FromList(Source,
Splitter.SplitByNothing(), null, null, ExtraValues.Error),
#"Expanded Column1" = Table.ExpandRecordColumn(#"Converted to
Table", "Column1", {"ParameterCode", "ParameterDescription",
"Units", "UnitsDescription", "Category", "SubCategory",
"Frequency"}, {"Column1.ParameterCode",
"Column1.ParameterDescription", "Column1.Units",
"Column1.SubCategory", "Column1.Frequency"})
```

in

#"Expanded Column1"

- 4. Click on Done.
- 5. The Parameter details data are extracted into the Excel Power Query Editor.
- 6. Click on Close & Load. The Parameter Details data are extracted into an Excel worksheet.
- 7. Change the name of Query1 and Sheet1 to *ParameterDetails*. This is the name in the airquality-api-excel-power-query.xlsx Excel workbook.

1	Α	В	С	D	E	F	G	н	·
1	Column1.ParameterCode	Column1.ParameterDescriptio	Column1.Units	Column1.UnitsDescription	Column1.Category	Column1.SubCategory	Column1.Frequency		Queries & Connections
2	AQC	AQC	category	category	Site AQC	Daily	Hourly average		Queries Connections
3	AQC	AQC	category	category	Site AQC	Hourly	Hourly average		
4	CO	Carbon monoxide	count	number of exceedances	Exceedences	Annual	8h rolling average derived from 1h average		2 queries
5	CO	Carbon monoxide	count	number of exceedances	Exceedences	Daily	8h rolling average derived from 1h average		III SiteDetails
6	CO	Carbon monoxide	count	number of exceedances	Exceedences	Monthly	8h rolling average derived from 1h average		126 rows loaded.
7	CO	Carbon monoxide	count	number of exceedances	Exceedences	Annual	Hourly average		ParameterDetails
8	CO	Carbon monoxide	count	number of exceedances	Exceedences	Daily	Hourly average		126 rows loaded
9	CO	Carbon monoxide	count	number of exceedances	Exceedences	Monthly	Hourly average		120 TOWS TOBUEU.
10	CO	Carbon monoxide	ppm	parts per million	Averages	Daily	24h average derived from 1h average		
11	CO	Carbon monoxide	ppm	parts per million	Averages	Annual	8h rolling average derived from 1h average		
12	CO	Carbon monoxide	ppm	parts per million	Averages	Daily	8h rolling average derived from 1h average		
13	CO	Carbon monoxide	ppm	parts per million	Averages	Hourly	8h rolling average derived from 1h average		
14	CO	Carbon monoxide	ppm	parts per million	Averages	Monthly	8h rolling average derived from 1h average		
15	CO	Carbon monoxide	ppm	parts per million	Maximums	Annual	8h rolling average derived from 1h average		

8. To refresh the data, go to Data/RefreshAll/Refresh.

5.1.3 Extract Current Observed air quality data using Excel and API

The following instructions outline how to extract current observed real time hourly air quality data from the DPIE Azure Cloud Data Warehouse using Excel Power Query and the DPIE API.

- 1. Create a new blank query by following the procedure as described in 5.1.1 (above), up to after clicking on the Advanced editor.
- 2. Replace the code in Query 1 with the code below: (remove paragraph markers) Copy the code between the horizontal lines below.

let

```
url = "https://data.airquality.nsw.gov.au/api/Data/get_Observations",
```

body = "",

Parsed_JSON = Json.Document(body),

BuildQueryString = Uri.BuildQueryString(Parsed_JSON),

Source = Json.Document(Web.Contents(url,[Headers = [#"Content-Type"="application/json"], Content = Text.ToBinary(body)])),

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

#"Expanded Column1" = Table.ExpandRecordColumn(#"Converted to Table", "Column1", {"Site_Id", "Parameter", "Date", "Hour", "HourDescription", "Value", "AirQualityCategory", "DeterminingPollutant"}, {"Column1.Site_Id", "Column1.Parameter", "Column1.Date", "Column1.Hour", "Column1.HourDescription", "Column1.Value", "Column1.AirQualityCategory", "Column1.DeterminingPollutant"}),

#"Expanded Column1.Parameter" = Table.ExpandRecordColumn(#"Expanded Column1", "Column1.Parameter", {"ParameterCode", "ParameterDescription", "Units", "UnitsDescription", "Enguappy", "Catagony", "SubCatagony")

"UnitsDescription", "Frequency", "Category", "SubCategory"},

{"Column1.Parameter.ParameterCode", "Column1.Parameter.ParameterDescription",

"Column1.Parameter.Units", "Column1.Parameter.UnitsDescription",

"Column1.Parameter.Frequency", "Column1.Parameter.Category",

"Column1.Parameter.SubCategory"})

in

#"Expanded Column1.Parameter"

- 3. Click on *Done*. The Current Observed air quality details data, for the current hour, are extracted into the Excel Power Query Editor.
- 4. Click on *Close & Load.* The Current Observed details data are extracted into an Excel worksheet.
- 5. Change the name of Query 1 and Sheet1 to *CurrentObserved*. This is the name in the air-quality-api-excel-power-query.xlsx Excel workbook.

	1	2 3	4	5	6	7	8	9	10	11	12	13	14	
1 Co	lumn1.Site_Id Colun		et Column1.Par						Column1.H (Column1.AirQualityCategory	Column1.DeterminingPollutant	Queries & Connecti
2	33 AQC	AQC	category	category	Hourly average	Site AQC	Hourly	2021-07-16	18 5	pm - 6 pm		GOOD	PM2.5	Quarter Connections
3	33 HUMI	Humidity	%	percent	Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	44.726			Queries Connections
4	33 NEPH	Nephelometer	10^-4 m^-1	10^-4 m^-1	Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	0.036			3 queries
5	33 NO	Nitric Oxide	pphm	parts per hundred mil	Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	0.012603			Cha Danta la
6	33 NO2	Nitrogen Dioxide	e pphm	parts per hundred mil	li Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	0.353501	GOOD		LIII SiteDetails
7	33 OZON	E Ozone	pphm	parts per hundred mil	I 4h rolling average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	2.634975	GOOD		126 rows loaded.
8	33 OZON	E Ozone	pphm	parts per hundred mil	li Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	2.617125	GOOD		III ParameterDetails
9	33 PM10	PM10	μg/m³	microgram per cubic r	n 24h rolling averag	e Averages	Hourly	2021-07-16	18 5	pm - 6 pm	8.014125			126 rows loaded.
10	33 PM10	PM10	µg/m³	microgram per cubic r	n Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	11.693	GOOD		The common second
11	33 PM2.5	PM2.5	μg/m³	microgram per cubic r	n 24h rolling averag	e Averages	Hourly	2021-07-16	18 5	pm - 6 pm	3.850783			ta currentObserved
12	33 PM2.5	PM2.5	µg/m³	microgram per cubic r	n Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	6.634	GOOD		1,005 rows loaded.
13	33 SD1	Wind Direction S	sig*	degree	Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	23.049			
14	33 SO2	Sulphur Dioxide	pphm	parts per hundred mil	li Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	0.02479	GOOD		
15	33 TEMP	Temperature	°C	degree Celsius	Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	14.553			
16	33 WDR	Wind Direction (1(*	degree	Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	287.177			
17	33 WSP	Wind Speed (10r	n m/s	meter per second	Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	4.555			
18	39 AQC	AQC	category	category	Hourly average	Site AQC	Hourly	2021-07-16	18 5	pm - 6 pm		GOOD	PM2.5	
19	39 CO	Carbon monoxid	le ppm	parts per million	8h rolling average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	0.096951	GOOD		
20	39 CO	Carbon monoxid	le ppm	parts per million	Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	0.07088			
21	39 HUMI	Humidity	%	percent	Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	47.177			
22	39 NEPH	Nephelometer	10^-4 m^-1	10^-4 m^-1	Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	0.048			
23	39 NO	Nitric Oxide	pphm	parts per hundred mil	I Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	-0.030016			
24	39 NO2	Nitrogen Dioxide	e pphm	parts per hundred mil	I Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	0.213251	GOOD		
25	39 OZON	Ozone	pphm	parts per hundred mil	4h rolling average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	2.717575	GOOD		
26	39 OZON	Ozone	pphm	parts per hundred mil	I Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	2.738525	GOOD		
27	39 PM10	PM10	µg/m³	microgram per cubic r	n 24h rolling averag	e Averages	Hourly	2021-07-16	18 5	pm - 6 pm	9.0895			
28	39 PM10	PM10	µg/m³	microgram per cubic r	n Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	12.897	GOOD		
29	39 PM2.5	PM2.5	µg/m³	microgram per cubic r	n 24h rolling averag	e Averages	Hourly	2021-07-16	18 5	pm - 6 pm	2.990333			
30	39 PM2.5	PM2.5	µg/m³	microgram per cubic r	n Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	2.089	GOOD		
31	39 RAIN	Rainfall	mm/m²	millimetre rainfall	Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	0			
32	39 SD1	Wind Direction S	ile °	degree	Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	28.782			
33	39 SO2	Sulphur Dioxide	pphm	parts per hundred mil	I Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	0.029025	GOOD		
34	39 SOLAR	Solar radiation	W/m²	watt per square mete	r Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	-7.204			
35	39 TEMP	Temperature	°C	degree Celsius	Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	14.457			
36	39 WDR	Wind Direction (1(°	degree	Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	309.335			
37	39 WSP	Wind Speed (10r	n/m/s	meter per second	Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	4.439			
38	61 CO	Carbon monoxid	le ppm	parts per million	8h rolling average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	0.042781	GOOD		
39	61 CO	Carbon monoxid	le ppm	parts per million	Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	0.025171			
40	61 HUMI	Humidity	%	percent	Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	44.949			
41	61 NEPH	Nephelometer	10^-4 m^-1	10^-4 m^-1	Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm				
42	61 NO	Nitric Oxide	pphm	parts per hundred mil	li Hourly average	Averages	Hourly	2021-07-16	18 5	pm - 6 pm	2.690619			
	SiteDetails	ParameterDetails	CurrentObser	ved 🕀										C

6. To refresh the data, go to Data/RefreshAll/Refresh.

Current Observed air quality data

In the code above, the body command is equal to empty quotations (body = "",). This is important, since this is the **default state** for extracting DPIE air quality data and means that only the current hourly data for all sites and all parameters will be extracted from the Azure Cloud Data Warehouse. If historical or filtered air quality data is required than the body needs to be filled with the specific parameters and this is detailed in section 5.1.4.

1	1 2	3	4	5	6	7	8	9	10	11	12	13	14
1	Column1.Site_Id_Column1.Para	a Column1.Paramet	Column1.Par	Column1.Parameter.U	Column1.Paramet	e Column1.Par	a Column1.Par					Column1.AirQualityCategory	Column1.DeterminingPollutant
2	33 AQC	AQC	category	category	Hourly average	Site AQC	Hourly	2021-07-16	18 .	5 pm - 6 pm		GOOD	PM2.5
3	33 HUMID	Humidity	%	percent	Hourly average	Averages	Hourly	2021-07-16	18	5 pm - 6 pm	44.726		
4	33 NEPH	Nephelometer	10^-4 m^-1	10^-4 m^-1	Hourly average	Averages	Hourly	2021-07-16	18 .	5 pm - 6 pm	0.036		
5	33 NO	Nitric Oxide	pphm	parts per hundred mill	Hourly average	Averages	Hourly	2021-07-16	18	5 pm - 6 pm	0.012603		
6	33 NO2	Nitrogen Dioxide	pphm	parts per hundred mill	i Hourly average	Averages	Hourly	2021-07-16	18 3	5 pm - 6 pm	0.353501	GOOD	
7	33 OZONE	Ozone	pphm	parts per hundred mill	4h rolling average	Averages	Hourly	2021-07-16	18	5 pm - 6 pm	2.634975	GOOD	
8	33 OZONE	Ozone	pphm	parts per hundred mill	Hourly average	Averages	Hourly	2021-07-16	18 3	5 pm - 6 pm	2.617125	GOOD	
9	33 PM10	PM10	µg/m³	microgram per cubic m	24h rolling average	e Averages	Hourly	2021-07-16	18 .	5 pm - 6 pm	8.014125		
10	33 PM10	PM10	µg/m³	microgram per cubic m	Hourly average	Averages	Hourly	2021-07-16	18 !	5 pm - 6 pm	11.693	GOOD	
11	33 PM2.5	PM2.5	µg/m²	microgram per cubic m	24h rolling average	e Averages	Hourly	2021-07-16	18	5 pm - 6 pm	3.850783		
12	33 PM2.5	PM2.5	µg/m³	microgram per cubic m	Hourly average	Averages	Hourly	2021-07-16	18 .	5 pm - 6 pm	6.634	GOOD	
13	33 SD1	Wind Direction Sig	•	degree	Hourly average	Averages	Hourly	2021-07-16	18	5 pm - 6 pm	23.049		
14	33 SO2	Sulphur Dioxide	pphm	parts per hundred mill	i Hourly average	Averages	Hourly	2021-07-16	18 3	5 pm - 6 pm	0.02479	GOOD	
15	33 TEMP	Temperature	°C	degree Celsius	Hourly average	Averages	Hourly	2021-07-16	18	5 pm - 6 pm	14.553		
16	33 WDR	Wind Direction (10	•	degree	Hourly average	Averages	Hourly	2021-07-16	18 3	5 pm - 6 pm	287.177		
17	33 WSP	Wind Speed (10m	m/s	meter per second	Hourly average	Averages	Hourly	2021-07-16	18 .	5 pm - 6 pm	4.555		
18	39 AQC	AQC	category	category	Hourly average	Site AQC	Hourly	2021-07-16	18 .	5 pm - 6 pm		GOOD	PM2.5
19	39 CO	Carbon monoxide	ppm	parts per million	8h rolling average	Averages	Hourly	2021-07-16	18 3	5 pm - 6 pm	0.096951	GOOD	
20	39 CO	Carbon monoxide	ppm	parts per million	Hourly average	Averages	Hourly	2021-07-16	18 .	5 pm - 6 pm	0.07088		
21	39 HUMID	Humidity	%	percent	Hourly average	Averages	Hourly	2021-07-16	18	5 pm - 6 pm	47.177		
22	39 NEPH	Nephelometer	10^-4 m^-1	10^-4 m^-1	Hourly average	Averages	Hourly	2021-07-16	18 .	5 pm - 6 pm	0.048		
23	39 NO	Nitric Oxide	pphm	parts per hundred mill	Hourly average	Averages	Hourly	2021-07-16	18	5 pm - 6 pm	-0.030016		
24	39 NO2	Nitrogen Dioxide	pphm	parts per hundred mill	Hourly average	Averages	Hourly	2021-07-16	18 .	5 pm - 6 pm	0.213251	GOOD	
25	39 OZONE	Ozone	pphm	parts per hundred mill	4h rolling average	Averages	Hourly	2021-07-16	18 .	5 pm - 6 pm	2.717575	GOOD	
26	39 OZONE	Ozone	pphm	parts per hundred mill	Hourly average	Averages	Hourly	2021-07-16	18 .	5 pm - 6 pm	2.738525	GOOD	
27	39 PM10	PM10	µg/m³	microgram per cubic m	24h rolling average	e Averages	Hourly	2021-07-16	18 5	5 pm - 6 pm	9.0895		
28	39 PM10	PM10	µg/m³	microgram per cubic m	Hourly average	Averages	Hourly	2021-07-16	18 .	5 pm - 6 pm	12.897	GOOD	
29	39 PM2.5	PM2.5	µg/m²	microgram per cubic m	24h rolling average	e Averages	Hourly	2021-07-16	18 3	5 pm - 6 pm	2.990333		
30	39 PM2.5	PM2.5	µg/m³	microgram per cubic m	Hourly average	Averages	Hourly	2021-07-16	18 5	5 pm - 6 pm	2.089	GOOD	
31	39 RAIN	Rainfall	mm/m²	millimetre rainfall	Hourly average	Averages	Hourly	2021-07-16	18	5 pm - 6 pm	0		
32	39 SD1	Wind Direction Sig	٠	degree	Hourly average	Averages	Hourly	2021-07-16	18 3	5 pm - 6 pm	28.782		
33	39 SO2	Sulphur Dioxide	pphm	parts per hundred mill	Hourly average	Averages	Hourly	2021-07-16	18	5 pm - 6 pm	0.029025	GOOD	
34	39 SOLAR	Solar radiation	W/m²	watt per square meter	Hourly average	Averages	Hourly	2021-07-16	18 .	5 pm - 6 pm	-7.204		
35	39 TEMP	Temperature	°C	degree Celsius	Hourly average	Averages	Hourly	2021-07-16	18 .	5 pm - 6 pm	14.457		
36	39 WDR	Wind Direction (10	٠	degree	Hourly average	Averages	Hourly	2021-07-16	18 .	5 pm - 6 pm	309.335		
37	39 WSP	Wind Speed (10m	m/s	meter per second	Hourly average	Averages	Hourly	2021-07-16	18	5 pm - 6 pm	4.439		

5.1.4 Extract Historical air quality data using Excel and API

The following instructions outline how to extract historical air quality data from the DPIE Azure Cloud Data Warehouse using Excel Power Query and the DPIE API.

- 1. Create a new blank query by following the procedure as described in section 5.1.1, up to after clicking on the Advanced editor.
- 2. Replace the code in Query 1 with the code below: (remove paragraph markers). Copy the code between the horizontal lines below.

let

```
url = " https://data.airquality.nsw.gov.au/api/Data/get_Observations",
```

body = " $\{$

```
""Parameters"": [""PM10""],
```

""Sites"": [329, 336, 2330, 3330, 4330, 5330, 7330],

""StartDate"": ""2018-12-05"",

```
""EndDate"": ""2018-12-06"",
```

""Categories"": [""Averages""],

""Subcategories"": [""Hourly""],

""Frequency"": [""Hourly average""]

}",

Parsed_JSON = Json.Document(body),

BuildQueryString = Uri.BuildQueryString(Parsed_JSON),

Source = Json.Document(Web.Contents(url,[Headers = [#"Content-Type"="application/json"], Content = Text.ToBinary(body)])),

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

#"Expanded Column1" = Table.ExpandRecordColumn(#"Converted to Table", "Column1", {"Site_Id", "Parameter", "Date", "Hour", "HourDescription", "Value", "AirQualityCategory", "DeterminingPollutant"}, {"Column1.Site_Id", "Column1.Parameter", "Column1.Date", "Column1.Hour", "Column1.HourDescription", "Column1.Value", "Column1.AirQualityCategory", "Column1.DeterminingPollutant"}),

#"Expanded Column1.Parameter" = Table.ExpandRecordColumn(#"Expanded Column1", "Column1.Parameter", {"ParameterCode", "ParameterDescription", "Units", "UnitsDescription", "Category", "SubCategory", "Frequency"}, {"Column1.Parameter.ParameterCode", "Column1.Parameter.ParameterDescription", "Column1.Parameter.Units", "Column1.Parameter.UnitsDescription", "Column1.Parameter.Category", "Column1.Parameter.SubCategory", "Column1.Parameter.Frequency"})

in

#"Expanded Column1.Parameter"

- 3. Click on *Done.* The Historical air quality details data are extracted into the Excel Power Query Editor.
- 4. Click on Close & Load. This extracts the Historical details data into an Excel worksheet.
- 5. Change the name of Query 1 and Sheet1 to *Historical*. This is the name in the airquality-api-excel-power-query.xlsx Excel workbook.

1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	*	
1		ite_Id Column		.F Colum	n Column1.Parameter.Uni	Column1.Para	m(Column1.Para	ar Column1.Parame	el Column1.Date	Column1		Column1.Value Column	1.AirQualityCategory	Column1.DeterminingPollutant		Queries & Connections
2		329 PM10	PM10	µg/m³	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05	1	l 12 am - 1 am	25.758 GOOD				Quarter Connections
3		336 PM10	PM10	µg/m²	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05	1	l 12 am - 1 am	25.665 GOOD				Quenes Connections
4		2330 PM10	PM10	µg/m²	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05	1	l 12 am - 1 am	26.773 GOOD				4 queries
5		3330 PM10	PM10	μg/m³	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05	1	l 12 am - 1 am	23.073 GOOD				- C. D. J.
6		4330 PM10	PM10	µg/m³	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05	1	l 12 am - 1 am	19.366 GOOD				IIII SiteDetails
7		5330 PM10	PM10	µg/m²	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05	1	l 12 am - 1 am	21.853 GOOD				126 rows loaded.
8		7330 PM10	PM10	µg/m*	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05	1	l 12 am - 1 am	31.56 GOOD				III ParameterDetails
9		329 PM10	PM10	µg/m³	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05	1	2 1 am - 2 am	23.681 GOOD				126 rows loaded.
10		336 PM10	PM10	µg/m³	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05		2 1 am - 2 am	25.124 GOOD				
11		2330 PM10	PM10	µg/mª	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05	1	2 1 am - 2 am	18.82 GOOD				LIII CurrentObserved
12		3330 PM10	PM10	µg/m³	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05	2	21am - 2am	10.487 GOOD				1,005 rows loaded.
13		4330 PM10	PM10	µg/m³	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05	1 1	2 1 am - 2 am	17.567 GOOD				III Historical
14		5330 PM10	PM10	µg/m³	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05		21am-2am	13.102 GOOD				168 rows loaded.
15		7330 PM10	PM10	ug/m ^a	microgram per cubic me	Averages	Hourly	Hourly average	2018-12-05		1 am - 2 am	26.403 GOOD				

6. To refresh the data, go to Data/RefreshAll/Refresh.

Historical Observed air quality data

The only **difference between the CurrentObserved and Historical data sets** is the body of the text in the Power Query. To get the current observed hourly data, that is refreshed every hour, the body is empty between the double quotes. (body = "",). To extract historical or filter data, then specific parameter information is required between the quotes of the body command. For instance:

body = "{
""Parameters"": [""PM10""],
""Sites"": [336, 4330, 2330, 7330, 3330, 329, 5330],
""StartDate"": ""2018-12-05"",
""EndDate"": ""2018-12-06"",
""Categories"": [""Averages""],
""Subcategories"":[""Hourly""],
""Frequency"":[""Hourly average""],
}",

In the example above:

Parameters: **PM10.** Full list in the Parameter Details worksheet.

Sites: Aberdeen (336), Bulga (4330), Camberwell (2330), Jerry's Plains (7330), Maison Dieu (3330), Merriwa (329), Mount Thorley (5330). Full list in the Site Details worksheet.

StartDate: 2018-12-05 5th Dec 2018. The start date of the filter

EndDate: 2018-12-05 5th Dec 2018. The end date of the filter.

Category: Averages. The statistical value. Full list in the Parameter Details worksheet.

Subcategory: *Hourly*. The time-period. Full list in the Parameter Details worksheet.

Frequency: <u>*Hourly average*</u>. The frequency time-period. Full list in the Parameter Details worksheet.

For the query above the following data will be extracted.

1	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Column1.Site_l	l Columr	n 1. Column1	1.F Columr	n Column1.Parameter.Uni	i Column1.Param	Column1.Para	r Column 1 .Parame	l Column1.Date	Column1	l. Column1.Hou	Column1.Value	Column1.AirQualityCategory	Column1.DeterminingPollutant
2	32	9 PM10	PM10	µg/m³	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05	1	l 12 am - 1 am	25.76	GOOD	
3	33	6 PM10	PM10	µg/m³	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05	1	l 12 am - 1 am	25.67	GOOD	
4	233	0 PM10	PM10	µg/m³	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05	1	l 12 am - 1 am	26.77	GOOD	
5	333	0 PM10	PM10	µg/m³	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05	1	l 12 am - 1 am	23.07	GOOD	
6	433	0 PM10	PM10	µg/m³	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05	1	l 12 am - 1 am	19.37	GOOD	
7	533	0 PM10	PM10	µg/m³	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05	1	l 12 am - 1 am	21.85	GOOD	
8	733	0 PM10	PM10	µg/m³	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05	1	l 12 am - 1 am	31.56	GOOD	
9	32	9 PM10	PM10	µg/m³	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05	2	2 1 am - 2 am	23.68	GOOD	
10	33	6 PM10	PM10	µg/m³	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05	2	21 am - 2 am	25.12	GOOD	
11	233	0 PM10	PM10	µg/m³	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05	2	21 am - 2 am	18.82	GOOD	
12	333	0 PM10	PM10	µg/m³	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05	2	21 am - 2 am	10.49	GOOD	
13	433	0 PM10	PM10	µg/m³	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05	2	2 1 am - 2 am	17.57	GOOD	
14	533	0 PM10	PM10	µg/m³	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05	2	21 am - 2 am	13.10	GOOD	
15	733	0 PM10	PM10	µg/m³	microgram per cubic me	t Averages	Hourly	Hourly average	2018-12-05	2	2 1 am - 2 am	26.40	GOOD	

To retrieve the hourly averages for PM10 remove the Frequency parameter. This will give all the hourly averages including the *Hourly average* and the *24h rolling average derived from 1h average*.

```
body = "{

""Parameters"": [""PM10""],

""Sites"": [336, 4330, 2330, 7330, 3330, 329, 5330],

""StartDate"": ""2018-12-05"",

""EndDate"": ""2018-12-06"",

""Categories"": [""Averages""],

""Subcategories"": [""Hourly""]

}",
```

The data extracted from the query above is shown below.

1	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1 0	olumn1.Site_I	d Column1.Pa	ra Column1.Param	Column1.F	P. Column1.Parameter.UnitsDe	Column1.Par	a Column1.P	a Column 1. Parameter. Frequency	Column1.Date	Column1	1 Column1.HourDescription	Column1.Value	Column1.AirQualityCategory	Column1.DeterminingPollutant
2	33	29 PM10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	24h rolling average derived from 1h average	2018-12-05	1	1 12 am - 1 am	30.55425		
3	33	29 PM10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	Hourly average	2018-12-05	1	1 12 am - 1 am	25.758	GOOD	
4	33	36 PM10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	24h rolling average derived from 1h average	2018-12-05	1	1 12 am - 1 am	27.8925		
5	3	36 PM10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	Hourly average	2018-12-05	1	1 12 am - 1 am	25.665	GOOD	
6	233	30 PM10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	24h rolling average derived from 1h average	2018-12-05	1	1 12 am - 1 am	48.39175		
7	23	30 PM10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	Hourly average	2018-12-05	1	1 12 am - 1 am	26.773	GOOD	
8	333	30 PM10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	24h rolling average derived from 1h average	2018-12-05	1	1 12 am - 1 am	39.499167		
9	333	30 PM10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	Hourly average	2018-12-05	1	1 12 am - 1 am	23.073	GOOD	
10	433	30 PM10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	24h rolling average derived from 1h average	2018-12-05	1	1 12 am - 1 am	27.333125		
11	433	30 PM10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	Hourly average	2018-12-05	1	1 12 am - 1 am	19.366	GOOD	
12	533	30 PM10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	24h rolling average derived from 1h average	2018-12-05	1	1 12 am - 1 am	46.7175		
13	533	30 PM10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	Hourly average	2018-12-05	1	1 12 am - 1 am	21.853	GOOD	
14	73	30 PM10	PM10	µg/m³	microgram per cubic meter	Averages	Hourly	24h rolling average derived from 1h average	2018-12-05	1	1 12 am - 1 am	39.079		
15	73	30 PM10	PM10	ug/m ³	microgram per cubic meter	Averages	Hourly	Hourly average	2018-12-05	1	12 am - 1 am	31.56	6000	

Note: the 24h rolling average derived from 1h average does not have an Air Quality Category.

5.1.5 Explanation of the parameters within the body command

The ParameterDetails data will identify the **correct combination** of Parameters, Categories, Subcategory and Frequency that can be added to the body command. If the correct combination of Parameters, Categories, Subcategory and Frequency is not used then an error message will result.

Worked examples of extracting historical data are detailed in Appendix 1 and 2. The following examples highlight the need to have the correct combination of parameters, categories, subcategory and frequency.

Example 1

All the meteorological parameters (temperature, humidity, rain, wind speed and wind direction) **only have hourly averages** as shown in the data set below.

Column1.P	Column1.ParameterDescription	Column1.Units 🔽	Column1.UnitsDescription 💌	Column1.Frequency	Column1.Category	Column1.SubCategory 💌
HUMID	Humidity	%	percent	Hourly average	Averages	Hourly
RAIN	Rainfall	mm/m²	millimetre rainfall	Hourly average	Averages	Hourly
SD1	Wind Direction Sigma Theta	0	degree	Hourly average	Averages	Hourly
TEMP	Temperature	°C	degree Celsius	Hourly average	Averages	Hourly
WDR	Wind Direction (10m)	0	degree	Hourly average	Averages	Hourly
WSP	Wind Speed (10m)	m/s	meter per second	Hourly average	Averages	Hourly

Example 2

The parameters PM10 and PM2.5 are the only parameters that have 24 hour rolling averages

Column1.ParameterCc+1	Column1.Parar 🕶	Column1.Ur 🔻	Column1.UnitsDescription	Column1.Frequency	🕶 Column1.Categ 🖬	Column1.SubCateg
PM10	PM10	µg/m³	microgram per cubic meter	24h rolling average derived from 1h average	Averages	Hourly
PM2.5	PM2.5	µg/m³	microgram per cubic meter	24h rolling average derived from 1h average	Averages	Hourly

Detailed below is a brief summary of the parameters that can be added to the body command.

Parameters

The list below identifies the possible parameters that can be placed in the Parameters command. *(in italics)*

Air Quality

Gas

CO, NEPH, NH3, NO, NO2, OZONE, SO2

Particles

NEPH, PM10, PM10d, PM2.5, PM2.5d, TSPd

Meteorological

HUMID, RAIN, SOLAR, TEMP, SD1, WDR, WSP

Sites

The complete list of Sites is in the SiteDetails data.

To extract regional AQC (RAQC) data you need to use the site id for the regions.

Category and Subcategory

Table 1 below identifies the various Category and Subcategory combinations that can be in the Category and Subcategory command.

Category	Subcategory	Comment
Averages	Hourly	Hourly average
	Daily	Daily average
	Monthly	Monthly average
	Annual	Annual average
Exceedances	Daily	Number of exceedances in a day
	Monthly	Number of exceedances in a month
	Annual	Number of exceedances in a year
	Yearly	Number of exceedances over years.
Maximums	Daily	Daily maximum
	Monthly	Monthly maximum
	Annual	Annual maximum
Site AQC	Hourly	Hourly site Air Quality Category (AQC)
	Daily	Daily site Air Quality Category (AQC)
Regional AQC	Hourly	Hourly regional Air Quality Category (RAQC)
	Daily	Daily regional Air Quality Category (RAQC)

 Table 1
 Category and Subcategory combinations

6. Search and retrieve air quality data using the R programming language and API

The DPIE Air Quality API data service can be used by a variety of third-party software applications as identified in Figure 1, to search and retrieve DPIE air quality data. The following outlines how to deploy the API using the R programming language. Instructions on installing the R programming language can be found by clicking the link below:

https://rstudio.com/

To extract the Air Quality data using the API you must be familiar with the R programming language environment. The code below is written in RStudio Version 1.2.1335.

6.1 Deploying the API using the R programming language

The API provides three data sets:

- Site details data
- Parameter details data
- Observational data. The observational data can be extracted as current real-time hourly data or as historical data by querying data through the **POST** function in R.

The Site Details and Parameter Details can be queried through setting the body value in the **GET** function to list().

6.1.1 Extract Monitoring Site data using R programming language and API

The following instructions outline how to extract monitoring Site data from the DPIE Azure Cloud Data Warehouse using R programming language and the API.

- 1. Reproduce the R code below, that is between the horizontal lines, into the source/workspace area in the R Studio software.
- 2. Highlight the code.
- 3. Click Run.

#_

The Site Details data is written into the df_site_details datafame and then to the SiteDetails.csv file, in the working directory.

6.1.2 Extract Parameter data using R programming language and API

The following instructions outline how to extract Parameter data from the DPIE Azure Cloud Data Warehouse using R programming language and the DPIE API.

- 1. Reproduce the R code below, that is between the horizontal lines, into the source/workspace area in the R Studio software.
- 2. Highlight the code.
- 3. Click Run.

The Parameter Details data is written to the df_param_details dataframe and then to the ParameterDetails.csv file, in the working directory.

#___

6.1.3 Extract Current Observed air quality data using R programming language and API.

The following instructions outline how to extract current observed hourly air quality data from the DPIE Azure Cloud Data Warehouse using R programming language and the DPIE API.

- 4. Reproduce the R code below, that is between the horizontal lines, into the source/workspace area in the R Studio software.
- 5. Highlight the code.
- 6. Click Run.

The Current Observed air quality data, for the current hour, is written to the df_current_obs dataframe and then to the CurrentObs.csv file in the working directory.

#

6.1.4 Extract Historical air quality data using R programming language and API

The following instructions outline how to extract historical air quality data from the DPIE Azure Cloud Data Warehouse using R programming language and the DPIE API.

- 1. Reproduce the R code below, that is between the horizontal lines, into the source/workspace area in the R Studio software.
- 2. Highlight the code.
- 3. Click Run.

#

The Historical Observed PM₁₀ air quality data, is written to the df_historical_obs dataframe and then to the <code>HistoricalObs.csv</code> file in the working directory.

```
#Install 'dplyr', 'httr', and 'jsonlite' libraries to extract Air Quality API data.
library(dplyr)
library(httr)
library(jsonlite)
# Historical air quality data
path_historical_obs <- 'https://data.airquality.nsw.gov.au/api/Data/get_Observations'</pre>
bodyparam <- '{ "Parameters": [ "PM10" ],</pre>
Dodyparam <- `{ "Parameters": [ "PM10" ],
"sites": [ 336, 4330, 2330, 7330, 3330, 329, 5330 ],
"startDate": "2018-12-05",
"EndDate": "2018-12-06",
"Categories": [ "Averages" ],
"subCategories": [ "Hourly" ],
"Frequency": [ "Hourly average" ]}'
df_historical_obs <- path_historical_obs %>%
   POST(body = bodyparam,
         ))) %>%
   content(as = "text", encoding = "UTF-8") %>%
fromJSON(flatten = TRUE) %>%
   data.frame()
df_historical_obs
# Save Historical Observations in a csv file in the working directory
write.csv(df_historical_obs, file="HistoricalObs")
```

Citation.

#

R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

7. Search and retrieve air quality data using Python programming language and API

The DPIE Air Quality API data service can be used by a variety of third-party software applications as identified in Figure 1, to search and retrieve DPIE air quality data. The following outlines how to deploy the API using the Python programming language.

To extract the Air Quality data using the API you must be familiar with the Python programming language environment. The code below is written in Python 3.8.2.

7.1 Deploying the API using the Python programming language

The API provides three data sets:

- Site details data
- Parameter details data
- Observational data.

The observational data can be extracted as current real-time hourly data or as historical data by querying data through the **POST** function in Python.

The Site Details and Parameter Details can be queried through the **GET** function in Python.

7.1.1 Extract Monitoring Site data using Python programming language and API

The following instructions outline how to extract monitoring Site data from the DPIE Azure Cloud Data Warehouse using Python programming language and the API.

- 1. Create a text file from the code below.
- 2. Open the file in the Python shell environment.
- 3. Click F5 to run the code.

The Site details data is written to the SiteDetails.txt file in the working directory.

```
by thon 3.8.2 Shell
                                                                ×
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
import os
import svs
import requests
import logging
import urllib
import json
class aqms_api_class(object):
   This class defines and configures the API to query the Azure DataWarehouse
   .....
   def __init__(self, ):
      self.logger = logging.getLogger( file )
      self.url api = "https://data.airquality.nsw.gov.au"
      self.headers = {'content-type': 'application/json', 'accept': 'application/json'}
      self.get site url = "/api/Data/get SiteDetails"
      return
def get_site_details(self, ):
      Build a query to return all the sites details
      query = urllib.parse.urljoin(self.url api, self.get site url)
      response = requests.get(url = query, data = '')
      return response
_name__ == '__main__':
AQMS = aqms_api_class()
if _
   AllSites = AQMS.get site details()
   Save Site Details to a text file
   f = open('SiteDetails.txt', 'w') # open a file in write mode
   for item in AllSites: # iterate over the list items
      item = item.decode("ISO-8859-1") # remove b from check process
      f.write(str(item) + '\n') # write to the file in working directory
   f.close()
```

7.1.2 Extract Parameter data using Python programming language and API

The following instructions outline how to the Parameter data from the DPIE Azure Cloud Data Warehouse using Python programming language and the DPIE API.

- 1. Create a text file from the code below.
- 2. Open the file in the Python shell environment.
- 3. Click F5 to run the code.

The Parameter details data is written to the ParameterDetails.txt file in the working directory.

```
A Python 3.8.2 Shell
                                                                 Х
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
import os
import svs
import requests
import logging
import urllib
import datetime as dt
import json
class agms api class(object):
   This class defines and configures the API to query the Azure DataWarehouse
   .....
   def init (self, ):
      self.logger = logging.getLogger(__file_
      self.url api = "https://data.airquality.nsw.gov.au"
      self.headers = {'content-type': 'application/json', 'accept': 'application/json'}
      self.get parameters = "/api/Data/get ParameterDetails"
      retur
def get_parameters_details(self, ):
      . . .
      Build a query to return all the parameter details
      111
      query = urllib.parse.urljoin(self.url_api, self.get_parameters)
      response = requests.get(url = query, data = '')
      return response
if __name__ == '__main__':
    AQMS = aqms_api_class()
   Allparameters = AQMS.get parameters details()
   1.1.1
   Save Parameter Details to a text file
   f = open('ParameterDetails.txt', 'w') # open a file in write mode
   for item in Allparameters:
      item = item.decode("ISO-8859-1") # iterate over the list items
      f.write(str(item) + '\n') # write to the file in working directory
   f.close()
```

7.1.3 Extract Current Observed air quality data using Python programming language and API

The following outlines how to extract current observed hourly air quality data from the DPIE Azure Cloud Data Warehouse using Python programming language and the DPIE API.

- 1. Create a text file from the code below.
- 2. Open the file in the Python shell environment.
- 3. Click F5 to run the code.

The current hourly observed air quality data, for the current hour, is written to the CurrentObs.txt file in the working directory file.

Python 3.8.2 Shell \times File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AM D64)] on win32 Type "help", "copyright", "credits" or "license()" for more information. >>> import os import sys import requests import logging import urllib import datetime as dt import json class aqms_api_class(object): This class defines and configures the api to query the aqms database def _init_(self,): self.logger = logging.getLogger(file self.url api = "https://data.airquality.nsw.gov.au" self.headers = {'content-type': 'application/json', 'accept': 'application/json'} self.get site url = 'api/Data/get SiteDetails' self.get parameters = 'api/Data/get ParameterDetails' self.get_observations = 'api/Data/get_Observations' return def get_Obs(self,): Build a query to return all the Current hourly observations query = urllib.parse.urljoin(self.url api, self.get observations) response = requests.post(url = query, data = '', headers = self.headers) return response if __name__ == '__main__': AQMS = aqms_api_class() AllCurrentObs = AQMS.get Obs() # Save Current hourly observations to a text file f = open('CurrentObs.txt', 'w') # open a file in write mode for item in AllCurrentObs: item = item.decode("ISO-8859-1") # iterate over the list items f.write(str(item) + '\n') # write to the file in working directory f.close()

7.1.4 Extract Historical air quality data using Python programming language and API

The following outlines how to extract current historical air quality data from the DPIE Azure Cloud Data Warehouse using Python programming language and the DPIE API.

- 1. Create a text file from the code below.
- 2. Open the file in the Python shell environment.
- 3. Click F5 to run the code.

The Historical Observed PM10 air quality data, is written to the HistoricalObs.txt file in the working directory file.

```
Python 3.8.2 Shell
                                                                    \times
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
import os
import sys
import requests
import logging
import urllib
import datetime as dt
import json
class aqms_api_class(object):
  This class defines and configures the api to query the agms database
       init_(self, ):
   def
      self.logger = logging.getLogger( file )
      self.url api = "https://data.airquality.nsw.gov.au"
     self.headers = {'content-type': 'application/json', 'accept': 'application/json'}
     self.get_observations = 'api/Data/get_Observations'
def get Obs(self, ObsRequest):
     Build a guery to return all the observation details
      query = urllib.parse.urljoin(self.url_api, self.get_observations)
      response = requests.post(url = query, data = json.dumps(ObsRequest), headers = self.headers)
      return response
*****************
```

```
def ObsRequest_init(self, ):
       1.1.1
       Build a query to return all historical observations
       111
       ObsRequest = {}
       ObsRequest['Parameters'] = ['PM10']
       ObsRequest['Sites'] = [336, 4330, 2330, 7330, 3330, 329, 5330]
       StartDate = dt.date(2018,12,5)
       EndDate = dt.date(2018,12,6)
       ObsRequest['StartDate'] = StartDate.strftime('%Y-%m-%d')
      ObsRequest['EndDate'] = EndDate.strftime('%Y-%m-%d')
ObsRequest['Categories'] = ['Averages']
       ObsRequest['SubCategories'] = ['Hourly']
       ObsRequest['Frequency'] = ['Hourly average']
       return ObsRequest
if __name__ == '__main__':
    AQMS = aqms_api_class()
    ObsRequest = AQMS.ObsRequest_init()
   AllHistoricalObs = AQMS.get_Obs(ObsRequest)
   Save Historical Obserations to a text file
   111
   f = open('HistoricalObs.txt', 'w') # open a file in write mode
for item in AllHistoricalObs: # iterate over the list items
      item = item.decode("ISO-8859-1") # remove b from check process
       f.write(str(item) + '\n') # write to the file in working directory
   f.close()
```

8. Appendix 1: worked examples Extract Historical air quality data using Excel and API

8.1 Example 1

How do I extract Daily Maximum Values and the Air Quality Category for Ozone 4 hour rolling averages at Chullora monitoring site for December 2018?

Step 1

Open the ParameterDetails worksheet and filter on OZONE, Maximums and Daily. This filtering verifies this is the correct combination of Parameter, Category, SubCategory and Frequency that can be added to the body command.

Column1.ParameterCod	e 🚽 Column1.Frequency	T Column1.Category	T Column1.SubCategory								
OZONE	4h rolling average derived from 1h average	Maximums	Daily								
Parameter: O	ZONE										
Sites: 190 (Ch	nullora).										
Start date: 20	18-12-01										
End date: 201	End date: 2019-01-01										
Categories: N	Categories: Maximums										
Subcategories	s: Daily										
Frequency: 4ł	Frequency: 4h rolling average derived from 1h average										
Step 2											
Copy the code from	m 5.1.4 and replace the body of the	command with									
body = "{											
""Parameters'	": [""OZONE""],										
""Sites"": [190	<mark>].</mark>										
""StartDate"":	<mark>""2018-12-01"",</mark>										
""EndDate"": '	"2019-01-01"",										
""Categories"	': [""Maximums""],										
""Subcategori	es"": [""Daily""],										
""Frequency""	: [""4h rolling average derived from	1h average""]									

}",

Below is the data set for the query in the body command. The query will extract the ozone 4hour rolling average daily maximum value and air quality category. Below is a screenshot for 1 and 2 December 2018. The name of the worksheet in the air-quality-api-excelpower-query.xlsx Excel workbook is <u>ChulloraOzone</u>.

Column1.Site_Id	Column1.	Column1	Column1.P	Colu: Column1.Pa	Colum	Column1.Parameter.Frequency	Column1.Date	Col Column1.Hou	Colum Column1.
190	OZONE	Ozone	pphm	parts Maximums	Daily	4h rolling average derived from 1h average	2018-12-01	1 12 am - 1 am	4.01 GOOD
190	OZONE	Ozone	pphm	parts Maximums	Daily	4h rolling average derived from 1h average	2018-12-02	1 12 am - 1 am	3.80 GOOD

8.2 Example 2

How do I extract Annual averages for PM10 and PM2.5 at Earlwood monitoring site from 2015 to 2018?

Step 1

Open the ParameterDetails worksheet and filter on PM10 and PM2.5, Averages and Annual. This filtering verifies this is the correct combination of Parameter, Category, SubCategory and Frequency that can be added to the body command.

Column1.ParameterCo	Column1.Frequency	. T	Column1.Categ 🎦	Column1.SubCateg
PM10	24h average derived from 1h average		Averages	Annual
PM2.5	24h average derived from 1h average		Averages	Annual

Parameter: PM10, PM2.5

Sites: 206 (Earlwood).

Start date: 2015-01-01

End date: 2019-01-01

Categories: Averages

Subcategories: Annual

Frequency: 24h average derived from 1h average

Step 2

Copy the code from 5.1.4 and replace the body of the command with:

body = "{
""Parameters"": [""PM10"", ""PM2.5""],
""Sites"": [206],
""StartDate"": ""2015-01-01"",
""EndDate"": ""2019-01-01"",
""Categories"": [""Averages""],
""Subcategories"":[""Annual""],
""Frequency"": [""24h average derived from 1h average""]
} <mark>"</mark> ,

Below is the data set for the query in the body command. The query will extract the annual averages for PM10 and PM2.5 at Earlwood for the years from 2015 to 2018. The name of the worksheet in the air-quality-api-excel-power-query.xlsx Excel workbook is *EarlwoodPM10PM25AnnualAvg*.

Column1.Site_	Id Column1.Parameter	. Column1.Parameter.Pa		a Column1.Parameter.UnitsDescription	Column1.Parameter.Frequency	Column1.Par	a Column1.Parameter.S	ul Column1.Date	Column1.Ho: Column1.HourD	Column1.
	206 PM10	PM10	µg/m³	microgram per cubic meter	24h average derived from 1h average	Averages	Annual	2015-12-31	1 12 am - 1 am	17.15
	206 PM2.5	PM2.5	µg/m³	microgram per cubic meter	24h average derived from 1h average	Averages	Annual	2015-12-31	1 12 am - 1 am	8.54
	206 PM10	PM10	µg/m³	microgram per cubic meter	24h average derived from 1h average	Averages	Annual	2016-12-31	1 12 am - 1 am	17.59
	206 PM2.5	PM2.5	µg/m³	microgram per cubic meter	24h average derived from 1h average	Averages	Annual	2016-12-31	1 12 am - 1 am	8.08
	206 PM10	PM10	µg/m³	microgram per cubic meter	24h average derived from 1h average	Averages	Annual	2017-12-31	1 12 am - 1 am	18.02
	206 PM2.5	PM2.5	µg/m³	microgram per cubic meter	24h average derived from 1h average	Averages	Annual	2017-12-31	1 12 am - 1 am	7.28
	206 PM10	PM10	µg/m³	microgram per cubic meter	24h average derived from 1h average	Averages	Annual	2018-12-31	1 12 am - 1 am	19.77
	206 PM2.5	PM2.5	ug/mª	microgram per cubic meter	24h average derived from 1h average	Averages	Annual	2018-12-31	1 12 am - 1 am	7.84

9. Appendix 2: worked examples Extract Historical air quality data using R programming language and API

9.1 Example 1.

How do I extract Daily Maximum Values and the Air Quality Category for Ozone 4 hour rolling averages at Chullora monitoring site for December 2018?

Step 1

Import the ParameterDetails.csv file that was created in section 5.1.2 by opening Excel; and clicking on the Data and From Text/CSV on the Excel menu. This will import the Parameter data into an Excel worksheet.

File Home	Insert	Page Layou	t Formulas	Data
Get From	From From	Table/ Recent	Existing	Refresh
Data * Text/CSV	Web Rar	nge Sources	Connections	All +

Filter on OZONE, Maximums and Daily. This filtering verifies this is the correct combination of Parameter, Category, SubCategory and Frequency that can be added to the body command.

Column1.ParameterCode 🖃	Column1.Frequency	Ţ,	Column1.Category	îΤ	Column1.SubCategory	Û	r
OZONE	4h rolling average derived from 1h average		Maximums		Daily		

Parameter: OZONE Sites: 190 (Chullora). Start date: 2018-12-01 End date: 2019-01-01 Categories: Maximums Subcategories: Daily Frequency: 4h rolling average derived from 1h average

Step 2

#

Reproduce the R code below, that is between the horizontal lines, into the source/workspace area in the R Studio software, highlight the code, and click run.

The Historical Observed Ozone air quality data, is written to the df_historical_ozone dataframe and then to the HistoricalOzone.csv file in the working directory of R Studio.

9.2 Example 2

How do I extract Annual averages for PM10 and PM2.5 at Earlwood monitoring site from 2015 to 2018?

Step 1

Import the ParameterDetails.csv file that was created in section 5.1.2 by opening Excel; and clicking on the Data and From Text/CSV on the Excel menu. This will import the Parameter data into an Excel worksheet.



Filter on PM10 and PM2.5, Averages and Annual. This filtering verifies this is the correct combination of Parameter, Category, SubCategory and Frequency that can be added to the body command.

Column1.ParameterCo	Column1.Frequency	🕶 Column1.Categ 🖬	Column1.SubCateg
PM10	24h average derived from 1h average	Averages	Annual
PM2.5	24h average derived from 1h average	Averages	Annual
		1	

Parameter: PM10, PM2.5

Sites: 206 (Earlwood).

Start date: 2015-01-01

End date: 2019-01-01

Categories: Averages

Subcategories: Annual

Frequency: 24h average derived from 1h average

Step 2

#

#

Reproduce the R code below, that is between the horizontal lines, into the source/workspace area in the R Studio software, highlight the code, and click run.

The Historical Observed Particle Matter (PM10 and PM2.5) air quality data, is written to the df_historical_pm dataframe and then to the <code>HistoricalPM.csv</code> file in the working directory of R Studio.

```
#Install 'dplyr', 'httr', and 'jsonlite' libraries to extract Air Quality API data.
library(dplyr)
library(httr)
library(jsonlite)
# Historical air quality data - Example 2
# How do I extract Annual averages for PM10 and PM2.5 at Earlwood monitoring site from 2015 to 2018?
path_historical_pm <- 'https://data.airguality.nsw.gov.au/api/Data/get_Observations'</pre>
bodyparam <- '{ "Parameters": [ "PM10", "PM2.5"],
"sites": [ 206 ],
"startDate": "2015-01-01",
"EndDate": "2019-01-01",
"Categories": [ "Averages" ],
"subCategories": [ "Annual" ],
"Frequency": [ "24h average derived from 1h average" ]}'
df_historical_pm <- path_historical_pm %>%
  `Content-Type` = 'application/json'
        ))) %>%
  content(as = "text", encoding = "UTF-8") %>% fromJSON(flatten = TRUE) %>%
  data.frame()
df_historical_pm
# Save Historical Particle Matter Observations in a csv file in the working directory
write.csv(df_historical_pm, file="HistoricalPM")
```

10. Appendix 3: worked examples Extract Historical air quality data using Python programming language and API

10.1 Example 1.

How do I extract Daily Maximum Values and the Air Quality Category for Ozone 4 hour rolling averages at Chullora monitoring site for December 2018?

Step 1

Import the ParameterDetails.csv file that was created in section 5.1.2 by opening Excel; and clicking on the Data and From Text/CSV on the Excel menu. This will import the Parameter data into an Excel worksheet.



Filter on OZONE, Maximums and Daily. This filtering verifies this is the correct combination of Parameter, Category, SubCategory and Frequency that can be added to the body command.

Column1.ParameterCode 星	Column1.Frequency	T Column1.Category	I Column1.SubCategory	ĵΤ
OZONE	4h rolling average derived from 1h average	Maximums	Daily	
Parameter: OZO	NE			
Sites: 190 (Chulle	ora).			
Start date: 2018-	12-01			
End date: 2019-0	01-01			
Categories: Maxi	mums			
Subcategories: D	Daily			
Frequency: 4h ro	lling average derived from 1h av	/erage		

Step 2

Create a text file from the code below and then open the file in the Python shell environment and click F5 to run the code. The Historical Observed Ozone air quality data, is written to the HistoricalOzone.txt file in the working directory file.

```
import sys
import requests
import logging
import urllib
import datetime as dt
import json
*****
class aqms_api_class(object):
   This class defines and configures the api to query the aqms database
   .....
   def
       init (self, ):
      self.logger = logging.getLogger( file
                                     )
      self.url api = "https://data.airquality.nsw.gov.au"
      self.headers = {'content-type': 'application/json', 'accept': 'application/json'}
      self.get observations = 'api/Data/get_Observations'
      retur
def get_Obs(self, ObsRequest):
      Build a query to return all the observation details
      query = urllib.parse.urljoin(self.url_api, self.get_observations)
      response = requests.post(url = query, data = json.dumps(ObsRequest), headers = self.headers)
      return response
def ObsRequest init(self, ):
      Build a query to return all historical observations
      ....
      ObsRequest = {}
      ObsRequest['Parameters'] = ['OZONE']
      ObsRequest['Sites'] = [190]
      StartDate = dt.date(2018,12,1)
      EndDate = dt.date(2019,1,1)
      ObsRequest['StartDate'] = StartDate.strftime('%Y-%m-%d')
      ObsRequest['EndDate'] = EndDate.strftime('%Y-%m-%d')
ObsRequest['Categories'] = ['Maximums']
      ObsRequest['SubCategories'] = ['Daily']
      ObsRequest['Frequency'] = ['4h rolling average derived from 1h average']
      return ObsRequest
__name__ == '__main__':
AQMS = aqms_api_class()
if _
   ObsRequest = AQMS.ObsRequest init()
  AllHistoricalOzone = AQMS.get Obs(ObsRequest)
  Save Historical Ozone obserations to a text file
   f = open('HistoricalOzone.txt', 'w') # open a file in write mode
   for item in AllHistoricalOzone: # iterate over the list items
      item = item.decode("ISO-8859-1") # remove b from check process
      f.write(str(item) + '\n') # write to the file in working directory
   f.close()
```

10.2 Example 2

How do I extract Annual averages for PM10 and PM2.5 at Earlwood monitoring site from 2015 to 2018?

Step 1

Import the ParameterDetails.csv file that was created in section 5.1.2 by opening Excel; and clicking on the Data and From Text/CSV on the Excel menu. This will import the Parameter data into an Excel worksheet.



Filter on PM10 and PM2.5, Averages and Annual. This filtering verifies this is the correct combination of Parameter, Category, SubCategory and Frequency that can be added to the body command.

Column1.ParameterCo	Column1.Frequency	🕶 Column1.Categ 🖬	Column1.SubCateg
PM10	24h average derived from 1h average	Averages	Annual
PM2.5	24h average derived from 1h average	Averages	Annual
			_

Parameter: PM10, PM2.5

Sites: 206 (Earlwood).

Start date: 2015-01-01

End date: 2019-01-01

Categories: Averages

Subcategories: Annual

Frequency: 24h average derived from 1h average

Step 2

Create a text file from the code below and then open the file in the Python shell environment and click F5 to run the code. The Historical Observed Particle Matter (PM10 and PM2.5) air quality data, is written to the Historical PM.txt file in the working directory file.

```
import os
import sys
import requests
import logging
import urllib
import datetime as dt
import ison
*****
class aqms_api_class(object):
   This class defines and configures the api to query the aqms database
   def
       init (self, ):
      self.logger = logging.getLogger(__file__)
      self.url_api = "https://data.airquality.nsw.gov.au"
      self.headers = {'content-type': 'application/json', 'accept': 'application/json'}
      self.get observations = "/api/Data/get Observations"
def get_Obs(self, ObsRequest):
      Build a guery to return all the observation details
      query = urllib.parse.urljoin(self.url_api, self.get_observations)
      response = requests.post(url = query, data = json.dumps(ObsRequest), headers = self.headers)
      return response
def ObsRequest_init(self, ):
      1.1.1
      Build a query to return all historical PM10 and PM2.5 annual averages
      at Earlwood (site id 206)
      ObsRequest = {}
      ObsRequest['Parameters'] = ['PM10', 'PM2.5']
      ObsRequest['Sites'] = [206]
      StartDate = dt.date(2015, 1, 1)
      EndDate = dt.date(2019,1,1)
      ObsRequest['StartDate'] = StartDate.strftime('%Y-%m-%d')
      ObsRequest['EndDate'] = EndDate.strftime('%Y-%m-%d')
      ObsRequest['Categories'] = ['Averages']
      ObsRequest['SubCategories'] = ['Annual']
      ObsRequest['Frequency'] = ['24h average derived from 1h average']
      return ObsRequest
__name__ == '__main__':
AQMS = aqms_api_class()
   ObsRequest = AQMS.ObsRequest init()
  AllHistoricalPM = AQMS.get Obs(ObsRequest)
  Save Historical PM10 and PM2.5 obserations at Earlwood to a text file
   f = open('HistoricalPM.txt', 'w') # open a file in write mode
   for item in AllHistoricalPM:
                           # iterate over the list items
      item = item.decode("ISO-8859-1") # remove b from check process
      f.write(str(item) + '\n') # write to the file in working directory
   f.close()
```

11. Appendix 4: Data dictionary for air quality data

Data set	Field name	Description	Example	Data type
Site details				
get_SiteDetails	Site ID	Unique ID for each site. To be used as a filter on get_Observations	39	integer
	Site Name	The name of the site.	Rozelle	string
	Longitude	The longitude of the site.	-33.864353	number
	Latitude	The latitude of the site.	151.163933	number
	Region	The region where the site belongs.	Sydney central-east	string
Parameter details				
get_ParameterDetails	Parameter Code	The code for each parameter	SO2	string
	Parameter Description	The description of the parameter.	Sulphur Dioxide	string
	Parameter Units	The units used to measure the parameter	pphm	string
	Unit Description	The description of the unit of measurement	parts per hundred million	string
	Frequency	The frequency at which the parameter is recorded or calculated	Hourly average	string
	Category	The category to which the parameter belongs	Averages	string
Observations				
get_Observations	Site ID	The ID of the site where the value was recorded.	39	integer
	Parameter Code	The code for each parameter	SO2	string
	Parameter Description	The description of the parameter.	Sulphur Dioxide	string
	Parameter Units	The units used to measure the parameter	pphm	string

Table 2 Air Quality data set and descriptions

Data set	Field name	Description	Example	Data type
	Unit Description	The description of the unit of measurement	parts per hundred million	string
	Parameter Frequency	The frequency at which the parameter is recorded or calculated	Hourly average	string
	Parameter Category	The category to which the parameter belongs	Averages	string
	Parameter Sub Category	The sub-category to which the parameter belongs.	Hourly	string
	Date	The date the observed data is collected	2019-10-18	string
	Hour	The hour the observed data is collected. The values are collected every minute and averaged over an hour.	8	string
	Hour Description	The description of the hour. This identifies which hour the data has been collected.	7am – 8am	string
	Value	The recorded/calculated value	0.4	number
	Air Quality Category	Categorisation of each pollutant into Good, Fair, Poor, Very Poor or Extremely Poor.	Fair	string
	Determining pollutant	The pollutant that determines the Site or Regional Air Quality Category.	PM2.5	string