

Open2view API - Documentation

If you have any questions regarding this interface please address them to webmaster@open2view.com

GET <https://api.open2view.com/nz>

A copy of this documentation is available at the above address.

Username and password

To use the API you need a username and password which will be given to you by Open2view.

Available Formats

Resources are available in either XML or JSON format.

Eg.

GET <https://api.open2view.com/nz/properties.xml?id=123456>

GET <https://api.open2view.com/nz/properties.json?id=123456>

Properties

Get a list of all your properties

GET https://api.open2view.com/nz/properties.xml?{search_criteria}

GET https://api.open2view.com/nz/properties.json?{search_criteria}

Get a list of all properties. Optional search criteria will limit the results returned.

Possible search criteria

Key	Example	Notes
id	/properties.xml?id=123456 /properties.xml?id[]=123456&id[]=111222	Open2view ID Could also use /properties/123456.xml which will return a single resource
agency_id	/properties.xml?agency_id=ABC123 /properties.xml?agency_id[]=ABC123&agency_id[]=ZYX987	Your agency's reference number for the property
agent_id	/properties.xml?agent_id=123 /properties.xml?agent_id[]=123&agent_id[]=987	Get properties for an agent with the given Open2view ID
street_number	/properties.xml?street_number=5	The property's street number
street_name	/properties.xml?street_name=Grey%20St	The property's street name
price_from	/properties.xml?price_from=200000	Find all properties with price greater than given price
price_to	/properties.xml?price_to=500000	Find all properties with price less than given price
category	/properties.xml?category=residential /properties.xml?category[]=residential&category[]=commercial	Find all properties with given category (residential, commercial or agricultural)
status	/properties.xml?status=sold /properties.xml?status[]=current&status[]=contract	Find all properties with given status
property_type	/properties.xml?property_type=house /properties.xml?property_type[]=house&property_type[]=apartment	Find all properties with given property type (eg. house, retail, vineyard)
price_type	/properties.xml?price_type=auction /properties.xml?price_type[]=auction&price_type[]=tender	Find all properties with given price_type (eg. auction, tender, pbn)
bedrooms	/properties.xml?bedrooms=3	Find all properties with at least the given number of bedrooms
bathrooms	/properties.xml?bathrooms=2	Find all properties with at least the given number of bathrooms
geo	/properties.xml?geo[lat]=-41.244772&geo[long]=172.617188& geo[rad]=5	Find all the properties within radius kilometres of the specified latitude and longitude

featured	/properties.xml?featured=true	Return only featured properties
approved_from	/properties.xml?approved_from=20091001	Find all properties approved after the specified date (yyyymmdd)
approved_to	/properties.xml?approved_to=20091031	Find all properties approved before the specified date (yyyymmdd)
detail	/properties.xml?detail=full	Get either a limited or full set of details (short or full). You need permission to access full details.
limit	/properties.xml?limit=10	Limit the number of results
offset	/properties.xml?offset=10	Offset from where the results should be returned.
order	/properties.xml?order=agency_id	The field the results should be ordered by. See appendix for possible options.

Notes:

1. If the Open2view ID is specified, it will be used and all other fields will be ignored
2. If the Agency ID is specified, only the property that matches the Agency ID will be returned. Other fields will be ignored.
3. Otherwise, if multiple conditions are supplied, only properties matching all conditions will be returned.

List of returned attributes for each property in the list

Key	Notes	Detail
id	Open2view ID	Short and full
agency_id	The agency's unique identifier for that property	Short and full
last_updated	Date property was last updated on open2view.com	Short and full
approved	Date property was approved (made live) on open2view.com	Short and full
address	Street number and name. Also a display element which will be either true or false.	Short and full
status	Open2view's status. One of CURRENT, CONTRACT, SOLD, RENTED	Short and full
latitude	Latitude of property	Full Only
longitude	Longitude of property	Full Only
price		Full Only
category	Either Residential, Commercial or Agricultural	Full Only
property_type	The type of property (eg. House, Retail, Vineyard)	Full Only
suburb	Eg. Te Aro	Full Only
area	Eg. Wellington Central	Full Only
region	Eg. Wellington	Full Only
bedrooms	The number of bedrooms	Full Only
bathrooms	The number of bathrooms	Full Only
floor_size	The floor size of the house in square metres	Full Only
lot_size	The area of the property in square metres	Full Only
built_in	The year the property was built (freeform field). Eg. 1998, 1950s, 1970 -1975	Full Only
description	The description of the property	Full Only
photos	Count of photos for this property	Short and full
	A list of the photos for that property. Each photo element gives a url for the photo and thumbnail, the photo type (Normal, Elevated or Night), caption, and position (the order of the photos).	Full Only

virtualtours	Count of virtual tours for this property	Short and full
	A list of the virtual tours for that property. Each virtualtour element gives a url for the virtual tour and thumbnail, caption, and position (the order)	Full Only
floorplans	Count of floorplans for this property.	Short and full
	A list of the floorplans for that property. Each floorplan element gives a url for the floorplan and thumbnail, caption, and position (the order).	Full Only
agents	A list of the agents for that property. Includes a count. Each agent element gives their name and Open2view ID.	Short and full
openhomes	A list of the current openhomes for the property. Includes a count. Each openhome element gives the start and end of the open home as datetimes.	Full Only

XML Response example

```

HTTP/1.1 200 OK
Date: Thu, 07 Feb 2008 00:15:32 GMT
Content-Type: application/xml
Encoding: UTF-8
Content-Length: 13219

<?xml version="1.0" encoding="UTF-8"?>
<properties count="10">
  <property last_updated="Fri Jan 04 19:01:48 +1300 2008" id="162528">
    <address display="true">614 Moonshine Hill Road</address>
    <agency_id>57886</agency_id>
    <status>CURRENT</status>
    <latitude>-41.0949013784599</latitude>
    <longitude>175.005694627762</longitude>
    <photos count="8">
      <photo type="Normal" caption="" position="1"
thumbnail="http://www.nz.open2view.com/assets/property/100/c/9/c98845b78d081624a130
3c322522edd01f5df0da.jpeg">http://www.nz.open2view.com/assets/property/XL/c/9/c9884
5b78d081624a1303c322522edd01f5df0da.jpeg</photo>
      ...
    </photos>
    <virtualtours count="8">
      <virtualtour caption="Big Front lawn" position="0"
thumbnail="http://www.nz.open2view.com/assets/property/66/7/e/7e4ef33ae024d687d8eb0
6af315efd776e0dbb0a.jpeg">http://www.nz.open2view.com/tour/virtual/162528?num=0</vi
rtualtour>
      ...
    </virtualtours>
    <floorplans count="0">
    </floorplans>
  </property>
  ...
</properties>

```

JSON Response example

```

{
  "properties":
  {
    "count": "10",
    "property":
    [
      {
        "id": "162528",

```



```
<?xml version="1.0" encoding="UTF-8"?>
<errors count="2">
  <error field="approved_from">
    Invalid date supplied for Approved From. Please refer to documentation appendix
    for correct date format.
  </error>
  <error field="category">
    Category must be one of the following: Residential, Agricultural, and
    Commercial.
  </error>
</errors>
```

JSON Response example

```
{
  "errors":
  {
    "count": "2",
    "error":
    [
      {
        "field": "approved_from",
        "message": "Invalid date supplied for Approved From. Please refer to
        documentation appendix for correct date format."
      },
      {
        "field": "category",
        "message": "Category must be one of the following: Residential, Agricultural,
        and Commercial."
      }
    ]
  }
}
```

Get a single property

GET <https://api.open2view.com/nz/properties/{id}.xml>

GET <https://api.open2view.com/nz/properties/{id}.json>

Get full details of a single property with id of {id}

XML Response example

```
HTTP/1.1 200 OK
Date: Thu, 07 Feb 2008 00:15:32 GMT
Content-Type: application/xml
Encoding: UTF-8
Content-Length: 13219

<?xml version="1.0" encoding="UTF-8"?>
<property last_updated="Fri Jan 04 19:01:48 +1300 2008" id="162528">
  <address display="true">614 Moonshine Hill Road</address>
  <agency_id>57886</agency_id>
```

```

<status>CURRENT</status>
<latitude>-41.0949013784599</latitude>
<longitude>175.005694627762</longitude>
<price>BBO $1,150,000</price>
<category>Residential</category>
<property_type>Lifestyle Property</property_type>
<suburb>Pauatahanui</suburb>
<area>Porirua</area>
<region>Wellington</region>
<bedrooms>5</bedrooms>
<bathrooms>2</bathrooms>
<floor_size>275</floor_size>
<lot_size>158600</lot_size>
<built_in>2006</built_in>
<description>Rarely do you find ...</description>
<photos count="8">
  <photo type="Normal" caption="" position="1"
thumbnail="http://www.nz.open2view.com/assets/property/100/c/9/c98845b78d081624a130
3c322522edd01f5df0da.jpeg">http://www.nz.open2view.com/assets/property/XL/c/9/c9884
5b78d081624a1303c322522edd01f5df0da.jpeg</photo>
  ...
</photos>
<virtualtours count="8">
  <virtualtour caption="Big Front lawn" position="0"
thumbnail="http://www.nz.open2view.com/assets/property/66/7/e/7e4ef33ae024d687d8eb0
6af315efd776e0dbb0a.jpeg">http://www.nz.open2view.com/tour/virtual/162528?num=0</vi
rtualtour>
  ...
</virtualtours>
<floorplans count="0">
</floorplans>
<openhomes count="1">
  <openhome>
    <start>2009-10-31 13:00:00</start>
    <end>2009-10-31 13:30:00</end>
  </openhome>
</openhomes>
<agents count="1">
  <agent id="123">Brent Cottle</agent>
</agents>
</property>

```

JSON Response example

```

{
  "property":
  {
    "id": "162528",
    "last_updated": "Fri Jan 04 19:01:48 +1300 2008",
    "status": "CURRENT",
    "address": { "display": "true", "address": "614 Moonshine Hill Road" },
    "agency_id": "57886",
    "latitude": "-41.0949013784599",
    "longitude": "175.005694627762",
    "floorplans": { "count": "0" },
    "price": "BBO $1,150,000",
    "category": "Residential",
    "property_type": "Lifestyle Property",

```

```

"suburb": "Pauatahanui",
"area": "Porirua",
"region": "Wellington",
"bedrooms": "5",
"bathrooms": "2",
"floor_size": "275",
"lot_size": "158600",
"built_in": "2006",
"description": "Rarely do you find ... ",
"photos":
{
  "count": "8",
  "photo":
  [
    {
      "thumbnail": "http://www.nz.open2view.com/assets/property/100/c/9/c98845b78d081624a1303c322522edd01f5df0da.jpeg",
      "type": "Normal",
      "caption": "",
      "url": "http://www.nz.open2view.com/assets/property/XL/c/9/c98845b78d081624a1303c322522edd01f5df0da.jpeg"
    },
    ...
  ]
},
"virtualtours":
{
  "count": "8",
  "virtualtour":
  [
    {
      "thumbnail": "http://www.nz.open2view.com/assets/property/66/7/e/7e4ef33ae024d687d8eb06af315efd776e0dbb0a.jpeg",
      "caption": "Big Front lawn",
      "url": "http://www.nz.open2view.com/tour/virtual/162528?num=0"
    },
    ...
  ]
},
"openhomes":
{
  "count": "1",
  "openhome":
  {
    "start": "2009-10-31 13:00:00",
    "end": "2009-10-31 13:30:00"
  }
},
"agents":
{
  "count": "1",
  "agent":
  [
    {
      "id": "123",
      "name": "Brent Cottle"
    }
  ]
}

```

```
}  
}
```

Errors

If the property can't be found, a single error will be returned.

XML Response example

```
HTTP/1.1 200 OK  
Date: Thu, 07 Feb 2008 00:15:32 GMT  
Content-Type: application/xml  
Encoding: UTF-8  
Content-Length: 13219  
  
<?xml version="1.0" encoding="UTF-8"?>  
<error>  
  Couldn't find a property with that ID  
</error>
```

JSON Response example

```
{  
  "error": " Couldn't find a property with that ID"  
}
```

Agents

Get a list of all your agents

GET https://api.open2view.com/nz/agents.xml?{search_criteria}

GET https://api.open2view.com/nz/agents.json?{search_criteria}

Get a list of all agents. Optional search criteria will limit the results returned.

Possible search criteria

Key	Example	Notes
id	<code>/agents.xml?id=123</code> <code>/agents.xml?id[]=123&id[]=222</code>	Open2view ID Could also use <code>/agents/123.xml</code> which will return a single resource with full details
name	<code>/agents.xml?name=John%20Smith</code> <code>/agents.xml?name=Bob</code>	Search on the agent's name. Will perform partial matches. So <code>name=John%20Smith</code> will match John Smith and John Smithson.
username	<code>/agents.xml?username=jsmith</code> <code>/agents.xml?username[]=jsmith&username[]=johns</code>	The agent's Open2view login name
detail	<code>/agents.xml?detail=full</code>	Get either a limited or full set of details (short or full)
limit	<code>/agents.xml?limit=10</code>	Limit the number of results
offset	<code>/properties.xml?offset=10</code>	Offset from where the results should be returned.
order	<code>/properties.xml?order=name</code>	The field the results should be ordered by. See appendix for possible options.

Notes:

1. If the Open2view ID is specified, it will be used and all other fields will be ignored
2. Otherwise, if multiple conditions are supplied, only agents matching all conditions will be returned.

List of returned attributes for each agent

Key	Notes	Detail
id	Open2view ID	Short and full
name	Agent's full name	Short and full
username	Agent's Open2view login name	Full Only
phone	Agent's phone number	Full Only
after_hours_phone	Agent's after hours phone number	Full Only
mobile_phone	Agent's mobile phone number	Full Only
email	Agent's email address	Full Only
description	Agent's description	Full Only
photo	URL of agent's photo. (Empty if no photo exists.)	Full Only
website_url	URL of agent's website.	Full Only

properties	Count of agent's current properties	Short and full
	List of agent's current properties. Includes Open2view ID, Agency ID, address.	Full Only

XML Response example

```

HTTP/1.1 200 OK
Date: Thu, 07 Feb 2008 00:15:32 GMT
Content-Type: application/xml
Encoding: UTF-8
Content-Length: 13219

<?xml version="1.0" encoding="UTF-8"?>
<agents count="10">
  <agent id="123">
    <name>John Smith</name>
    <username>jsmith</username>
    <phone>555 1234</phone>
    <mobile_phone>021 555 1234</mobile_phone>
    <email>jsmith@virtualrealty.co.nz</email>
    <properties count="8">
      <property last_updated="Fri Jan 04 19:01:48 +1300 2008" id="162528">
        <address>614 Moonshine Hill Road</address>
        <agency_id>57886</agency_id>
      </property>
      ...
    </properties>
  </agent>
  ...
</agents>

```

JSON Response example

```

{
  "agents":
  {
    "count": "10",
    "agent":
    [
      {
        "id": "123",
        "name": "John Smith",
        "username": "jsmith",
        "phone": "555 1234",
        "mobile_phone": "021 555 1234",
        "email": "jsmith@virtualrealty.co.nz",
        "properties":
        {
          "count": "8",
          "property":
          [
            {
              "id": "162528",
              "last_updated": " Fri Jan 04 19:01:48 +1300 2008",
              "address": "614 Moonshine Hill Road",
              "agency_id": "57886",
            },
            ...
          ]
        }
      },
      ...
    ]
  }
}

```

```
    ]
  }
},
...
]
}
}
```

Errors

If there are any problems with the supplied search criteria then a list of errors will be returned instead of a list of agents.

XML Response example

```
HTTP/1.1 200 OK
Date: Thu, 07 Feb 2008 00:15:32 GMT
Content-Type: application/xml
Encoding: UTF-8
Content-Length: 13219

<?xml version="1.0" encoding="UTF-8"?>
<errors count="1">
  <error field="id">
    ID is not a valid number
  </error>
</errors>
```

JSON Response example

```
{
  "errors":
  {
    "count": "1",
    "error":
    {
      "field": "id",
      "message": " ID is not a valid number"
    }
  }
}
```

Get a single agent

GET <https://api.open2view.com/nz/agents/{id}.xml>

GET <https://api.open2view.com/nz/agents/{id}.json>

Get full details of a single agent with id of {id}

XML Response example

```
HTTP/1.1 200 OK
Date: Thu, 07 Feb 2008 00:15:32 GMT
Content-Type: application/xml
Encoding: UTF-8
Content-Length: 13219
```

```
<?xml version="1.0" encoding="UTF-8"?>
<agent id="123">
  <name>John Smith</name>
  <username>jsmith</username>
  <phone>555 1234</phone>
  <mobile_phone>021 555 1234</mobile_phone>
  <email>jsmith@virtualrealty.co.nz</email>
  <description>John Smith works hard to ...</description>
  <photo/>
  <properties count="8">
    <property last_updated="Fri Jan 04 19:01:48 +1300 2008" id="162528">
      <address>614 Moonshine Hill Road</address>
      <agency_id>57886</agency_id>
    </property>
    ...
  </properties>
</agent>
```

JSON Response example

```
{
  "agent":
  {
    "id": "123",
    "name": "John Smith",
    "username": "jsmith",
    "phone": "555 1234",
    "mobile_phone": "021 555 1234",
    "email": "jsmith@virtualrealty.co.nz",
    "description": "John Smith works hard to ...",
    "photo": "",
    "properties":
    {
      "count": "8",
      "property":
      [
        {
          "id": "162528",
          "last_updated": " Fri Jan 04 19:01:48 +1300 2008",
          "address": "614 Moonshine Hill Road",
          "agency_id": "57886",
        },
        ...
      ]
    }
  }
}
```

Errors

If the agent can't be found, a single error will be returned.

XML Response example

```
HTTP/1.1 200 OK
Date: Thu, 07 Feb 2008 00:15:32 GMT
Content-Type: application/xml
Encoding: UTF-8
Content-Length: 13219

<?xml version="1.0" encoding="UTF-8"?>
<error>
  Couldn't find an agent with that ID
</error>
```

JSON Response example

```
{
  "error": " Couldn't find an agent with that ID"
}
```

Appendix

Search criteria values

Date format

Dates should be in the format:

yyyymmdd

E.g. 20091031 for the 31st of October 2009

Category

residential
commercial
agricultural

Status

current
contract
sold
rented

Property Type

Residential		Commercial		Agricultural	
Name	URL encoded	Name	URL encoded	Name	URL encoded
Apartment	apartment	Accommodation	accommodation	Aquaculture	aquaculture
Flat	flat	Business for sale	business%20for%20sale	Crop	crop
Holiday Home	holiday%20home	Commercial Land	commercial%20land	Farm	farm
House	house	Hospitality	hospitality	Forestry	forestry
Investment	investment	Industrial	industrial	Grazing	grazing
Lifestyle Property	lifestyle%20property	Investment	investment	Horticulture	horticulture
Section	section	Office	office	Nurseries	nurseries
Studio	studio	Retail	retail	Orchard	orchard
Subdivision	subdivision	Tourism	tourism	Rural House	rural%20house
Terrace House	terrace%20house	Warehouse	warehouse	Specialised	specialised
Town House	town%20house			Vacant Land	vacant%20land
Unit	unit			Vineyard	vineyard
Villa	villa				

Price Type

Name	URL Encoded
Auction	auction

BBO	bbo
BEO	beo
Deadline Private Treaty	deadline%20private%20treaty
Expressions of Interest	expressions%20of%20interest
Mortgagee Auction	mortgagee%20auction
MWP	mwp
PBN	pbn
Per annum	per%20annum
Per Month	per%20month
Per sq.ft	per%20sq.ft
Per sq.metre	per%20sq.metre
Per week	per%20week
POA	poa
Price	price
Price + GST (if any)	price%20%2B%20gst%20%28if%20any%29
Rateable Value	rateable%20value
Set Date of Sale	set%20date%20of%20sale
Tender	tender

Order

Use {field}_desc to get the results in reverse order.

Properties	
id	Open2view ID
agency_id	Agency's reference number
address	Street name and number
approved	When property was approved (made live) on open2view.com
price	Price
status	Property's status

Agents	
id	Open2view ID
name	Agent's name
properties	Number of properties