



Office of
Environment
& Heritage

Developer Guidelines

BioNet Web Services

© 2016 State of NSW and Office of Environment and Heritage

With the exception of photographs, the State of NSW and Office of Environment and Heritage 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 Office of Environment and Heritage (OEH) has compiled this guideline 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. OEH 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. This document may be subject to revision without notice and readers should ensure they are using the latest version

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

OEH asserts the right to be attributed as author of the original material in the following manner:
© State of New South Wales and Office of Environment and Heritage 2016.

Published by:

Office of Environment and Heritage

59 Goulburn Street, Sydney NSW 2000

PO Box A290, Sydney South NSW 1232

Phone: +61 2 9995 5000 (switchboard)

Phone: 131 555 (environment information and publications requests)

Phone: 1300 361 967 (national parks, general environmental enquiries, and publications requests)

Fax: +61 2 9995 5999

TTY users: phone 133 677, then ask for 131 555

Speak and listen users: phone 1300 555 727, then ask for 131 555

Email: info@environment.nsw.gov.au

Website: www.environment.nsw.gov.au

Report pollution and environmental incidents

Environment Line: 131 555 (NSW only) or info@environment.nsw.gov.au

See also www.environment.nsw.gov.au

ISBN 978-1-76039-416-5

OEH 2016/0404

July 2016

Contents

Contents	iii
List of figures	iii
1. Introduction	1
1.1 What data is available?	1
2. Before you begin	2
2.1 How do I go about developing an application using data from BioNet?	2
2.2 What support do you provide?	2
2.3 What is your SLA?	2
2.4 Where do I find documentation on the data available via the Biodiversity Web Services?	2
2.5 Are there any security restrictions placed on the data?	3
2.6 Are there any terms and conditions I should be aware of when using data from the web service?	3
2.7 Are there any restrictions on your implementation of OData that I should be aware of?	3
2.8 What best practice guidance should I be aware of when creating OData queries?	4
2.9 How should I acknowledge BioNet Web Services in my application?	5
3. Getting started	6
3.1 BioNet Web Services URI	6
3.2 Architecture	6
3.3 Reference material for developers	6
4. Samples	8
4.1 Sample OData queries	8
4.2 Sample OData client	9
4.3 Sample output	10
5. Data we collect	11

List of figures

Figure 1: Overview of Bionet Web Service implementation	1
Figure 2: Service architecture	6
Figure 3: Species sightings core data record	9
Figure 4: Species sightings measurements or facts	9

1. Introduction

BioNet is the trusted source of biodiversity data for the state of NSW and a resource for all. The BioNet Web Service provides an application level Open API enabling developers to directly integrate biodiversity data into their software systems and unlock the innovation potential of this valuable data set.

1.1 What data is available?

Release 2.0 of the BioNet Web Services makes the following data available:

- **Species Sightings** data currently available through the [BioNet Atlas of NSW Wildlife](#) user interface.
- **Vegetation Classification** data on Plant Community Types currently available through [BioNet VIS Classification](#) user interface
- **NSW Landscapes** data on Mitchell Landscapes (Over Cleared Landscapes) currently available as a downloadable report through the [BioNet VIS Classification](#) user interface.

The data available via the web service is updated daily from the source data (see Figure 1 below).

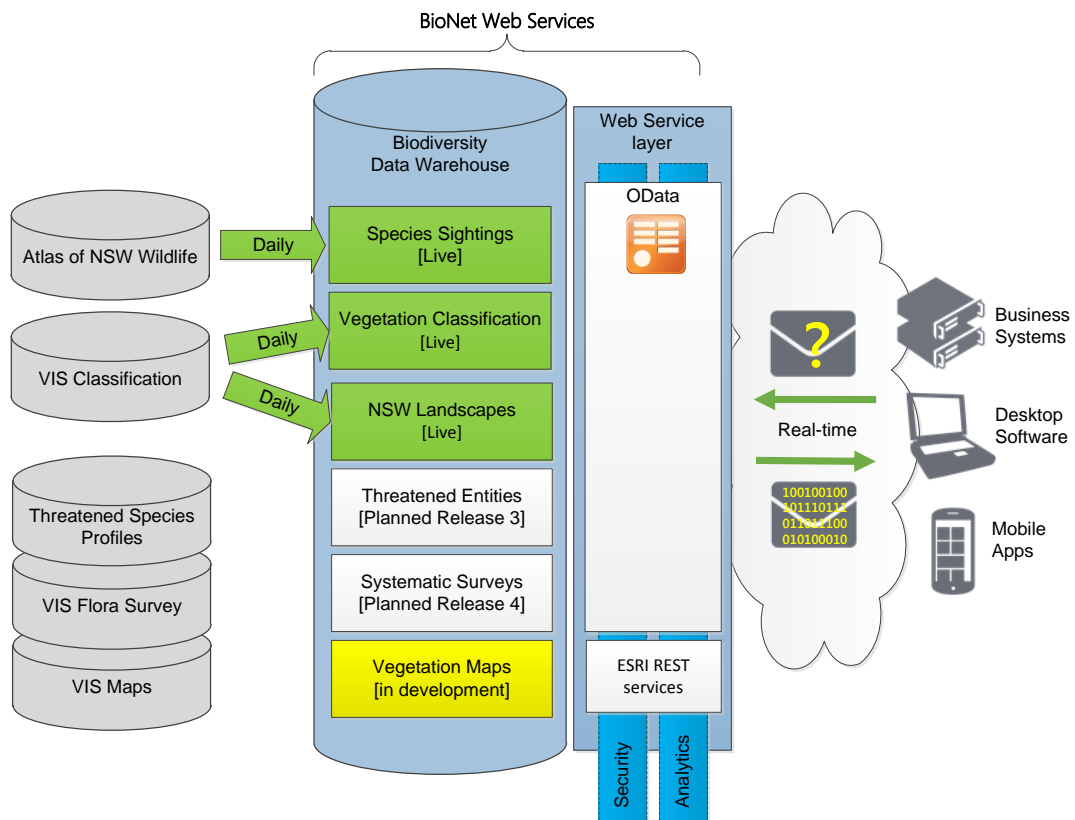


Figure 1: Overview of BioNet Web Service implementation

2. Before you begin



This section contains important information you need to understand and consider before you start the development of your application.

2.1 How do I go about developing an application using data from BioNet?

Download and fill out the Data Use Case form from [BioNet for developers](#) and submit to bionet@environment.nsw.gov.au. We will use the information provided to support you through the process and to issue you with an App Token.

Once you have received your App Token, read this guide and start developing.

2.2 What support do you provide?

We will provide advice to help guide you in your development decision making, particularly around the use and interpretation of data made available via the service. This advice will be provided based on the information you supply us in the Data Use Case form.

We will provide technical support for the server side services; that is for the correct functioning and availability of the API itself and for the data that is served through that API.

We do not provide technical support for client side applications, development toolkits or libraries. Support of these should be sourced from the vendor/supplier of the software, or in the case of open source solutions, from the developer community.

For support, email bionet@environment.nsw.gov.au

2.3 What is your SLA?

Our target for the web service is to support 50 simultaneous connections and process a well formed request retrieving 1000 records or less within one second of the web service receiving the request. The service will be unavailable from 3am to 6am to allow us to run the daily data update.

For more details on a well formed request, see section 2.8 What best practice guidance should I be aware of when creating OData queries? Where do I find documentation on the OData protocol?

The service uses the OASIS OData v4.0 Protocol. Documentation on the protocol is available at [OData](#).

2.4 Where do I find documentation on the data available via the Biodiversity Web Services?

Release 2.0 of the BioNet Web Services implements the Species Sighting, Vegetation Classification and NSW Landscapes Web Services. The data available via these services is documented in the following standards which are available at [BioNet for developers](#):

- *BioNet Web Services Species Sightings Web Service Data Standard*
- *BioNet Web Services Vegetation Classification Web Service Data Standard*
- *BioNet Web Services NSW Landscapes Web Service Data Standard*

2.5 Are there any security restrictions placed on the data?

Yes, the Species Sightings web service implements and uses the existing security model applied to the BioNet Atlas of NSW web application. For more details on the levels of access please refer to section 3 of the [BioNet Atlas User Manual](#).

As the same model is used, existing user credentials for the BioNet Atlas of NSW Wildlife or the VIS-C web apps will work for the Web Services. New user credentials should also be requested via the existing application process for login access to either the BioNet Atlas of NSW Wildlife or VIS-C.

However, authentication is not mandatory. If you do not pass a user name and password to the web service, it will return the publicly available data as per the query submitted. It should be noted that for the Species Sightings Web Service, this data will contain obfuscated data for species listed in the Sensitive Species Policy. This will be made clear in the dataGeneralizations field and will withhold some fields as listed in the dataWithheld field. There are also records which we do not have permission to share publicly. These are withheld and will not appear in the data returned.

2.6 Are there any terms and conditions I should be aware of when using data from the web service?

Yes, there are specific [terms and conditions](#) for the data shared via the Species Sightings web service which should be read and accepted before development of your application.

Development of an application that uses data from the Species Sightings web service will be considered acceptance of the terms and conditions by the developer of the application.

2.7 Are there any restrictions on your implementation of OData that I should be aware of?

2.7.1 \$orderby query option

To prevent sorting on properties that are not indexed in the database, \$orderby has been restricted to the following fields:

- catalogNumber
- scientificNameID
- consequence
- locationID
- PNFFilter
- occurrenceStatus
- taxonRank
- stateProvince
- coordinatePrecision
- datasetID
- occurrenceID

2.7.2 any and all functions

The any () and all () functions have been disabled to mitigate the risk of slow query performance and enable the service to enforce the Page limits. This will not restrict the data that a user can extract; rather it limits data served to 100,000 rows per page.

2.7.3 Filtering of navigation properties

Support for filtering on navigation properties will not be implemented. Filtering on navigation properties can result in a join, which will impact on the performance of the service.

This will not affect the usability of the service for users as the underlying data model does not support this type of filtering.

2.7.4 Server-side queryable attributes

We have implemented server-side queryable attributes to safe guard against large data returns and provide query optimisations. The attributes that have been implemented include:

- PageSize=100,000
- MaxNodeCount=50

2.8 What best practice guidance should I be aware of when creating OData queries?

2.8.1 \$select operator

All queries that an application submits should use the `$select` operator to return only the fields you require. Using the `$select` operator is critical to shield your application from changes to the data standard. It means that new fields added to an entity set, or removal of fields not used by your application will not affect your application's ability to process the data returned by the web service.

2.8.2 \$filter operators

The best practice is to use the `eq` operator in preference to other operators such as `startswith` or `endswith`. This is because the `eq` operator is the most efficient from a server resource consumption point of view, and will help ensure that performance of the service remains high for all users.

For example if you know you are searching for Square-tailed Kite you should use `?$filter=vernacularName eq 'Square-tailed Kite'`.

The `contains` operator should be avoided as this operator is very resource intensive and can result in query time outs. In this case it would be preferable to use the `startswith` operator.

2.8.3 \$orderby query option

The `$orderby` query option sorts result returned in either descending or ascending order. Best practice is to not use this option and undertake sorting if needed on the client side in your application.

However, if sorting on the server side is required it should be noted that ordering is very resource intensive and can result in query time outs. In this instance, best practice is to combine the `$orderby` with a `$filter` operator to limit the result set that is being sorted. In testing it was found that sorting result sets of +/- 500,000 records performed acceptably.

For example: `?$filter=vernacularName eq 'Square-tailed Kite'&$orderby=locationID desc`

2.8.4 Which fields should my query target?

The `eq` operator can be used on any field.

The following fields have been tuned to enable their use with the `startswith`, `endswith` and `contains` operators if necessary:

- `scientificName`
- `vernacularName`
- `family`
- `datasetName`
- `county`.

2.8.5 Are there any specific fields you recommend I should use?

We recommend that you should return the `dcterms_bibliographicCitation` and `catagloueNumber` fields. These fields enable us to know when the record was retrieved in case there have been changes to the data since the time of retrieval, and which record exactly the data refers to.

2.9 How should I acknowledge BioNet Web Services in my application?

Please contact us for a logo that you can use in your application to acknowledge the source of the data.

3. Getting started

Make sure you have received your App Token before proceeding further. See Section 2.1 How do I go about developing an application using data from BioNet? for more about the App Token.

3.1 BioNet Web Services URI

<https://data.bionet.nsw.gov.au/biosvcapp/odata>

3.2 Architecture

All requests are made using https to the relevant service URI.

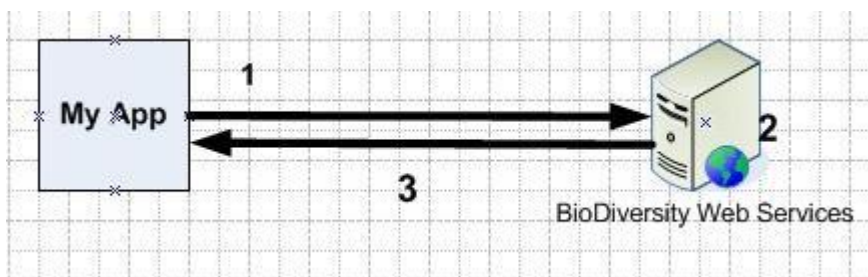


Figure 2: Service architecture

1. HTTPS request with App Token in header and conforming to the OData protocol. Supply your app token in your request header with the header name “access_token”. For example;

```
access_token: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

2. Web service processes request and sends response in json format. No other formats are supported at this time.
3. OData response contains results. Results are paged at 100,000 rows per page with a "NEXTLINK" link embedded in the data. Note that this only occurs if the resultant record count is greater than 100,000. For example
`"@odata.nextLink": "https://data.bionet.nsw.gov.au/biosvcapp/odata/SpeciesSightings_CoreData?$skip=100000"`

3.3 Reference material for developers

The following material is given to get you started. Please note that it is not exhaustive. It is suggested you Google OData and your development platform to find specific reference material, if available, for your project.

- About OData:
 - <http://www.software-architects.com/devblog/2014/09/12/10-OData-FAQs>
- OData tutorials:
 - <http://www.odata.org/getting-started/basic-tutorial/>
 - <https://damienbod.wordpress.com/2014/07/09/web-api-odata-v4-using-an-odata-t4-generated-client-part-8/>

- <http://blogs.msdn.com/b/odatateam/archive/2014/03/11/how-to-use-odata-client-code-generator-to-generate-client-side-proxy-class.aspx>
- <http://www.asp.net/web-api/overview/odata-support-in-aspnet-web-api/odata-v4/create-an-odata-v4-client-app>
- http://help.sap.com/saphelp_hanaplatform/helpdata/en/8d/78eb6f6e8f4307be939a6bc684b9ca/content.htm
- <http://www.ibm.com/developerworks/data/library/techarticle/dm-1205odata/index.html>
- <http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/dotnet/EntityFrameworkWCF/WCFEntityFramework.htm>
- OData libraries and code:
 - <http://www.odata.org/libraries/>
 - <http://stackoverflow.com> (but make sure you use only approved solutions)
 - <http://blogs.msdn.com/b/odatateam/>
 - <https://components.xamarin.com/gettingstarted/simple.odata.client>
 - <http://www.pluralsight.com/courses/#/table-of-contents/odata-clients>
 - <https://msdn.microsoft.com/en-us/library/ff478141.aspx>
- Vendor products supporting OData:
 - <http://www.odata.org/ecosystem/>
 - <http://www.odata.org/libraries/>
 - <http://stackoverflow.com> (but make sure you use only approved solutions)
 - <http://blogs.msdn.com/b/odatateam/>
 - <https://components.xamarin.com/gettingstarted/simple.odata.client>
 - <http://www.pluralsight.com/courses/#/table-of-contents/odata-clients>
 - <https://msdn.microsoft.com/en-us/library/ff478141.aspx>
- Vendor products supporting OData:
 - <http://www.odata.org/ecosystem/>

4. Samples

4.1 Sample OData queries

This link provides the structure information.

[https://data.bionet.nsw.gov.au/biosvcapp/odata/\\$metadata](https://data.bionet.nsw.gov.au/biosvcapp/odata/$metadata)

Calling the SpeciesSightings service returns the first 100,000 rows of the dataset. All queries return data paged at 100,000 rows at a time, the NEXT link is at the bottom of the output.

https://data.bionet.nsw.gov.au/biosvcapp/odata/SpeciesSightings_CoreData

Server-side queryable attributes. Note this list is not exhaustive. For a full list of attributes use the metadata query link above.

?\$select=dynamicProperties

?\$filter=catalogNumber eq '065000001'

?\$select=catalogNumber,scientificName,vernacularName

?\$filter=contains(vernacularName,'River Red')

?\$filter=contains(vernacularName,'River Red')&\$count=true

?\$filter=startswith(vernacularName,'Red')

?\$filter=startswith(toupper(vernacularName),'RED')

?\$filter=endswith(scientificName,'camaldulensis')

?\$filter=indexof(vernacularName,'blue') gt -1

?\$select=dynamicProperties&\$filter=contains(dynamicProperties,'CMA=Northern Rivers')

?\$select=dynamicProperties&\$filter=contains(dynamicProperties,'CMA=Sydney')

?\$select=countryConservation&\$filter=countryConservation ne null&\$count=true

?\$select=decimalLatitude,decimalLongitude&\$filter=contains(dynamicProperties,'CMA=North ern Rivers')

?\$filter=(PNFFilter eq

'Y')&\$select=catalogNumber,recordNumber,scientificNameID,scientificName,vernacularName,PNFFilter&\$orderby=scientificNameID

?\$select=catalogNumber,scientificNameID,scientificName,vernacularName,PNFFilter,decimalLatitude,decimalLongitude,geodeticDatum&\$filter=((decimalLongitude ge 142.0) and (decimalLongitude le 142.5)) and ((decimalLatitude lt -32.000000001) and (decimalLatitude gt -32.499999999))&\$orderby=scientificNameID

4.2 Sample OData client

4.2.1 C# .NET

Resources

- <http://www.nuget.org/packages/Microsoft.OData.Client/>
- <https://visualstudiogallery.msdn.microsoft.com/9b786c0e-79d1-4a50-89a5-125e57475937>

Sample C# snippet

```
Uri svcURL = new
Uri(ConfigurationManager.AppSettings["metadataURL"].ToString());
ODataServicesTestingApp.Default.Container container = new
ODataServicesTestingApp.Default.Container(svcURL);
....
if (!string.IsNullOrEmpty(ODataQuery.Text.Trim()))
{
    string queryString = string.Format("/SpeciesSightings{0}",
ODataQuery.Text.Trim());
    dataGridView1.DataSource =
container.Execute<TbICUBE_SpeciesSightings>(new Uri(queryString,
UriKind.Relative)).ToList();
}
....
```

4.2.2 HTML5 and JavaScript

Resources

- <http://codesmartinc.com/2011/01/31/leveraging-odata-end-points-in-json-format-with-jquery/>
- <http://www.odata.org/libraries/#javascript>

4.3 Sample output

Fig.3. SpeciesSightings_CoreData record json format

```
{
  "@odata.context": "http://webdev.environment.nsw.gov.au/BioSvcApp/odata/$metadata#SpeciesSightings_CoreData", "value": [
    {
      "PKID": 36438659, "basisOfRecord": "HumanObservation", "collectionCode": "BioNet Atlas of NSW Wildlife",
      "dataGeneralizations": "The observer name has been changed to a unique User ID",
      "datasetID": 1, "datasetName": "OEH Default Sightings", "dcterms_bibliographicCitation": "BioNet Atlas of NSW Wildlife 28/04/2015 12:00 AM",
      "dcterms_language": "en", "dcterms_modified": "1996-01-01T00:00:00 is Jan 1 1996 12:00AM UTC",
      "dcterms_rights": "Creative Commons by Attribution", "dcterms_rightsHolder": "Office of Environment and Heritage",
      "dcterms_type": "Event", "dynamicProperties": "BFMCS=N/A, botanicalDivision=N/A, CMA=Southern Rivers,
      CMAsubregion=Southern Rivers - East Gippsland Lowlands (Part C); floraReserve=N/A, geogExtent=N/A;
      IBRA=South East Corner (NSW); mechanicalHRDescription=N/A;
      potentialImpact=N/A; profileID=N/A; reserve=N/A; speciesFireDescription=N/A; stateForest=N/A; vulnerability=N/A",
      "informationWithheld": "The following fields have been withheld and are only available to
      licensed or OEH staff: locality, locationRemarks, occurrenceRemarks",
      "institutionCode": "NSW Office of Environment and Heritage", "ownerInstitutionCode": "Office of Environment and Heritage",
      "associatedReferences": null, "countryConservation": null, "catalogNumber": "75080-035", "consequence": null,
      "establishmentMeans": "Alive in NSW Native", "estimateTypeCode": null, "individualCount": null,
      "individualID": null, "observationType": "Observed", "occurrenceID": "urn:catalog:NSW Office of Environment and Heritage:BioNet Atlas of NSW
      Wildlife:75080-035", "occurrenceRemarks": "occurrenceRemarks withheld", "occurrenceStatus": "Present",
      "otherCatalogNumbers": null, "PNFFilter": "N", "recordedBy": "HUTKE", "recordNumber": null, "reproductiveCondition": null,
      "sex": null, "stateConservation": "Protected", "status": "Valid and accepted without modification", "eventDate": "1983-09-06", "eventID": null, "eventRemarks": null,
      "eventTime": null, "habitat": null, "samplingEffort": null, "samplingProtocol": null, "coordinatePrecision": "9",
      "coordinateUncertaintyInMeters": 1000.0000, "country": "Australia", "countryCode": "AU", "county": "BEGA
      VALLEY", "decimalLatitude": -37.246482589, "decimalLongitude": 149.921284892, "easting": 759113,
      "geodeticDatum": "GDA94", "georeferenceProtocol": null, "locality": "locality withheld", "locationRemarks": "locationRemarks
      withheld", "mapSheetNumber": "8823 - EDEN", "maximumElevationInMeters": null, "minimumElevationInMeters": null,
      "northing": 5873784, "stateProvince": "NSW", "zone": "55", "class": "Aves", "family": "Meliphagidae",
      "genus": "Caligavis", "infraspecificEpithet": null,
      "kingdom": "Fauna", "nomenclaturalCode": "ICZN", "order": "Passeriformes", "populationName": null,
      "scientificName": "Caligavis chrysops", "scientificNameAuthorship": "(Latham,
      1802)", "scientificNameID": "0614", "sortOrder": 2638, "locationID": "2620-035", "specificEpithet": "chrysops",
      "taxonRank": "Species", "vernacularName": "Yellow-faced Honeyeater"
    }
  ]
}
```

Fig.4. SpeciesSightings_MeasurementsOrFacts record json format

```
{
  "@odata.context": "http://webdev.environment.nsw.gov.au
  /BioSvcApp/odata/$metadata#MeasurementOrFact", "value": [
    {
      "DatasetID": 1, "OccuranceID": "urn:catalog:Office
      of Environment and Heritage, Department of Planning
      and Environment representing the State of New South
      Wales:BioNet Atlas of NSW
      Wildlife:065000001", "measurementID": "405", "measurementT
      ype": "SurfaceGeology", "measurementValue": "BA", "measurmentUnit": "Basalt"
    }
  ]
}
```

5. Data we collect

The service logs request details for internal analytics usage. The following information is captured:

- Web Service Name
- Request parameters
- Date and Time of request
- Data Size of response
- Geographic Location of requestor
- User name
- Application name.