



D04 – DELIVERABLE 5.1.2

Project Acronym: OpenUp!
Grant Agreement No: 270890
Project Title: Opening up the Natural History Heritage for Europeana

Data quality service for botanical names production version

D04 – Deliverable 5.1.2

Revision: 1.5a [Final]

Authors:

Nick Black **Royal Botanic Gardens, Kew (RBGK -16)**
Anna Saltmarsh **Royal Botanic Gardens, Kew (RBGK -16)**

Project co-funded by the European Commission within the ICT Policy Support Programme		
Dissemination Level		
P	Public	P
C	Confidential, only for members of the consortium and the Commission Services	

0 REVISION AND DISTRIBUTION HISTORY AND STATEMENT OF ORIGINALITY

Revision	Date	Author	Organisation	Description
1.1	11/11/2011	N. Black	RBGK	Detail of production version of botanical Name Service. Includes responses to issues raised at TMG stand-up calls on 22/09/11, 13/10/11, 27/10/11 and 10/11/11.
1.2	15/11/2011	A. Saltmarsh	RBGK	Proof-reading and minor corrections.
1.3	15/11/2011	S. Hargreaves	RBGK	Formatting.
1.4	16/11/2011	A. Saltmarsh	RBGK	Further minor corrections & queries.
1.5	16/11/2011	N. Black	RBGK	Corrections. This version approved by the TMG stand-up group on 17/11/11, including TMG chair.
1.5a	18/11/2011	Coordination Team	BGBM	Minor editing.

Statement of Originality:

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Distribution

Recipient	Date	Version	Accepted YES/NO
TMG	17/11/2011	1.5	YES
Work Package 5 Leader: Karol Marhold, IBSAS	17/11/2011	1.5	YES
Project Coordinator	20/11/2011	1.5a	YES



TABLE OF CONTENTS

0	REVISION AND DISTRIBUTION HISTORY AND STATEMENT OF ORIGINALITY	2
1	INTRODUCTION	4
2	NOMENCLATRURAL CHECK	4
2.1	Introduction	4
2.2	Metadata	4
2.3	Single query mode	5
2.4	Multiple query mode	6
3	TAXONOMIC CHECK.....	7
3.1	Introduction	7
3.2	Metadata	8
3.3	Single query mode	8
3.4	Multiple query mode	12
4	LIST OF REFERENCES	14



1 INTRODUCTION

The botanical data quality service has been subdivided into a *names* check and a *concept* check. These are each implemented as a HTTP accessed JSON format web-service, and are available from the following URL: <http://data.kew.org/openup-data-quality-service/>

2 NOMENCLATURAL CHECK

2.1 Introduction

The initial candidate service is focused on the nomenclatural check - accepting a name (or names) and returning a response to indicate if the name is found in the nomenclatural data source supporting the service.

We have investigated making this service compliant with the Google Refine¹ (previously known as Freebase Gridworks) reconciliation API. In summary this is an HTTP accessed service with JSON format request / response. The service must support callbacks (JSONP <http://en.wikipedia.org/wiki/JSONP>)²

2.2 Metadata

The service must support a **metadata request** - called by requesting the reconciliation service URL with no parameters or just a callback function name parameter.

2.2.1 Request

Sample URL: <http://data.kew.org/openup-data-quality-service/name-reconciler>

Response

```
{
  "name" : "IPNI name reconciliation service",
  "identifierSpace" : "http://lsid.ipni.org",
  "schemaSpace" : "http://lsid.ipni.org" ,
  "view" : {
    "url" : "http://ipni.org/{{id}}"
  }
}
```

2.2.2 Request (using callbacks)

Sample URL: http://data.kew.org/openup-data-quality-service/name-reconciler?callback=my_function

2.2.3 Response (using callbacks)

```
my_function(
{
```



```
"name" : "IPNI name reconciliation service",
"identifierSpace" : "http://lsid.ipni.org",
"schemaSpace" : "http://lsid.ipni.org" ,
"view" : {
  "url" : "http://ipni.org/{{id}}"
}
}
)
```

2.3 Single query mode

2.3.1 Request

Queries are specified in JSON. A single query can be specified as follows:

```
{
  "query" : "Ocimum americanum",
  "limit" : 3,
  "type" : "name",
}
```

The JSON query is specified as a URL parameter, and the service can respond to either GET or POST.

2.3.2 Response

```
{
  "result" : [
    {
      "id": "urn:lsid:ipni.org:names:452858-1"
      , "name": "Lamiaceae Ocimum americanum Auct. ex Benth."
      , "type": "name"
      , "score": "100"
      , "match": "true"
    }
    , {
      "id": "urn:lsid:ipni.org:names:316218-2"
      , "name": "Lamiaceae Ocimum americanum L."
      , "type": "name"
      , "score": "100"
      , "match": "true"
    }
    , {
      "id": "urn:lsid:ipni.org:names:452856-1"
      , "name": "Lamiaceae Ocimum americanum L."
      , "type": "name"
      , "score": "100"
      , "match": "true"
    }
  ]
}
```

The name input is validated and parsed using the ECAT Name Parser distributed by GBIF. If the name includes an author the service will try to match both the canonical name and the full name including the author string. It does this to avoid discarding records due to the different variations of author abbreviations. For names input with an authority the JSON output will score 100 for matching the Name+authority string and 90 for just matching the canonical name. If we added an authority to the input of the above example the following would be output.

2.3.3 Request

Query includes name and authority:

```
{
  "query" : "Ocimum americanum Auct. ex Benth.",
  "limit" : 3,
  "type" : "name",
}
```

2.3.4 Response

```
{
  "result" : [
    {
      "id": "urn:lsid:ipni.org:names:452858-1"
      , "name": "Lamiaceae Ocimum americanum Auct. ex Benth."
      , "type": "name"
      , "score": "100" -- match found for name+authority
      , "match": "true"
    }
    , {
      "id": "urn:lsid:ipni.org:names:316218-2"
      , "name": "Lamiaceae Ocimum americanum L."
      , "type": "name"
      , "score": "90" -- Canonical match found
      , "match": "true"
    }
    , {
      "id": "urn:lsid:ipni.org:names:452856-1"
      , "name": "Lamiaceae Ocimum americanum L."
      , "type": "name"
      , "score": "90"
      , "match": "true"
    }
  ]
}
```

As per the metadata request, a callback function name can be specified - this will have the effect of wrapping the JSON response in a function call to the specified function name.

2.4 Multiple query mode

Here, multiple queries are passed in a single request. The individual queries are specified as above, but are keyed. These keys are used to link the results to the queries.

2.4.1 Request

```
{ "q1" : {
  "query" : "Ocimum americanum",
  "limit" : 3,
  "type" : "name",
}
, "q2" : {
  "query" : "Ocimum africanum",
```



```
"limit" : 3,  
"type" : "name",  
} }
```

2.4.2 Response

```
{  
  "q2" : {  
    "result" : [  
      {  
        "id": "urn:lsid:ipni.org:names:452853-1"  
        , "name": "Lamiaceae Ocimum africanum Lour."  
        , "type": "name"  
        , "score": "100"  
        , "match": "true"  
      }  
    ]  
  }  
  , {  
    "q1" : {  
      "result" : [  
        {  
          "id": "urn:lsid:ipni.org:names:452857-1"  
          , "name": "Lamiaceae Ocimum americanum Jacq."  
          , "type": "name"  
          , "score": "100"  
          , "match": "true"  
        }  
        , {  
          "id": "urn:lsid:ipni.org:names:452858-1"  
          , "name": "Lamiaceae Ocimum americanum Auct. ex Benth."  
          , "type": "name"  
          , "score": "100"  
          , "match": "true"  
        }  
        , {  
          "id": "urn:lsid:ipni.org:names:452856-1"  
          , "name": "Lamiaceae Ocimum americanum L."  
          , "type": "name"  
          , "score": "100"  
          , "match": "true"  
        }  
      ]  
    }  
  }  
}
```

As per the metadata and single query requests, a callback function name can be specified.

3 TAXONOMIC CHECK

3.1 Introduction

This service has much the same interface as the nomenclatural check - accepting a name (or names) and returning a response to indicate the taxonomic status of the name in the underlying concept store, and either a list of synonyms or accepted names.

The **status** flag will be one of A (accepted), S (synonym), U (unplaced).



3.2 Metadata

The service must support a **metadata request** - called by requesting the reconciliation service URL with no parameters or just a callback function name parameter.

3.2.1 Request

Sample URL: <http://data.kew.org/openup-data-quality-service/concept-reconciler>

3.2.2 Response

```
{
  "name" : "Kew concept reconciliation service",
  "identifierSpace" : "http://data.kew.org",
  "schemaSpace" : "http://data.kew.org" ,
}
```

Request (using callbacks)

Sample URL: http://data.kew.org/openup-data-quality-service/concept-reconciler?callback=my_function

3.2.3 Response (using callbacks)

```
my_function(
{
  "name" : "Kew concept reconciliation service",
  "identifierSpace" : "http://data.kew.org",
  "schemaSpace" : "http://data.kew.org" ,
}
)
```

3.3 Single query mode

3.3.1 Request

Queries are specified in JSON. A single query can be specified as follows:

```
{
  "query" : "Ocimum americanum",
  "limit" : 3,
  "type" : "concept",
}
```

The JSON query is specified as a URL parameter, and the service can respond to either GET or POST.

3.3.2 Response

```
{
```




```
"result" : [
{
  "id": "kew-136802"
  , "name": "Ocimum americanum L."
  , "type": "concept"
  , "score": "100"
  , "match": "true"
  , "status": "Accepted"

  , "synonyms": [

    "Ocimum album Roxb."
    ,

    "Ocimum brachiatum Blume"
    ,

    "Ocimum canum Sims"
    ,

    "Ocimum dichotomum Hochst. ex Benth."
    ,

    "Ocimum dinteri Briq."
    ,

    "Ocimum fluminense Vell."
    ,

    "Ocimum fruticosum Burch."
    ,

    "Ocimum hispidulum Schumach. & Thonn."
    ,

    "Ocimum incanescens Mart."
    ,

    "Ocimum stamineum Sims"
    ,

    "Ocimum thymoides Baker"

  ]

}
,

{
  "id": "kew-136804"
  , "name": "Ocimum americanum var. pilosum (Willd.) A.J.Paton"
  , "type": "concept"
  , "score": "100"
  , "match": "true"
}
```



```
, "status": "Synonym"

, "accepted_names": [

    "Ocimum × africanum Lour."

]

}

]
```

The name input is validated and parsed using the ECAT Name Parser distributed by GBIF. If the name includes an author the service will try to match both the canonical name and the full name including the author string. It does this to avoid discarding records due to the different variations of author abbreviations. For names input with an authority the JSON output will score 100 for matching the Name+authority string and 90 for just matching the canonical name. If we added an authority to the input of the above example changing the input query to: -

```
{
  "query" : "Ocimum americanum L.",
  "limit" : 3,
  "type" : "concept",
}
```

The output would now have the following scores:-

```
{
  "result" : [
    {
      "id": "kew-136802"
      , "name": "Ocimum americanum L."
      , "type": "concept"
      , "score": "100"
      , "match": "true"
      , "status": "Accepted"

      , "synonyms": [

          "Ocimum album Roxb."
          ,

          "Ocimum brachiatum Blume"
          ,

          "Ocimum canum Sims"
          ,

          "Ocimum dichotomum Hochst. ex Benth."
          ,

          "Ocimum dinteri Briq."
        ]
    }
  ]
}
```



```
,
  "Ocimum fluminense Vell."
,
  "Ocimum fruticulosum Burch."
,
  "Ocimum hispidulum Schumach. & Thonn."
,
  "Ocimum incanescens Mart."
,
  "Ocimum stamineum Sims"
,
  "Ocimum thymoides Baker"
]
}
,
{
  "id": "kew-136804"
  , "name": "Ocimum americanum var. pilosum (Willd.) A.J.Paton"
  , "type": "concept"
  , "score": "90"
  , "match": "true"
  , "status": "Synonym"

  , "accepted_names": [
    "Ocimum × africanum Lour."
  ]
}
]
}
```

As per the metadata request, a callback function name can be specified - this will have the effect of wrapping the JSON response in a function call to the specified function name.

3.4 Multiple query mode

Here, multiple queries are passed in a single request. The individual queries are specified as above, but are keyed. These keys are used to link the results to the queries.

3.4.1 Request

```
{ "q1" : {
  "query" : "Ocimum americanum L.",
  "limit" : 3,
  "type" : "concept",
}
, "q2" : {
  "query" : "Ocimum africanum",
  "limit" : 3,
  "type" : "concept",
} } }
```

3.4.2 Response

```
{
  "q2" : {
    "result" : [
      {
        "id": "kew-136798"
        , "name": "Ocimum × africanum Lour."
        , "type": "concept"
        , "score": "100"
        , "match": "true"
        , "status": "Accepted"
      }
      , "synonyms": [
        "Ocimum americanum var. pilosum (Willd.) A.J.Paton"
        ,
        "Ocimum basilicum var. anisatum Benth."
        ,
        "Ocimum basilicum var. pilosum (Willd.) Benth."
        ,
        "Ocimum × citriodorum Vis."
        ,
        "Ocimum × graveolens A.Br."
        ,
        "Ocimum × petitianum A.Rich."
        ,
        "Ocimum × pilosum Willd."
      ]
    ]
  }
}
```



```
    ]
  }
]
}
,
"q1" : {
  "result" : [
    {
      "id": "kew-136802"
      , "name": "Ocimum americanum L."
      , "type": "concept"
      , "score": "100"
      , "match": "true"
      , "status": "Accepted"
      , "synonyms": [
          "Ocimum album Roxb."
          ,
          "Ocimum brachiatum Blume"
          ,
          "Ocimum canum Sims"
          ,
          "Ocimum dichotomum Hochst. ex Benth."
          ,
          "Ocimum dinteri Briq."
          ,
          "Ocimum fluminense Vell."
          ,
          "Ocimum fruticosum Burch."
          ,
          "Ocimum hispidulum Schumach. & Thonn."
          ,
          "Ocimum incanescens Mart."
          ,
          "Ocimum stamineum Sims"
          ,
          "Ocimum thymoides Baker"
        ]
    }
  ]
}
```



```
    ]
  }
  ,
  {
    "id": "kew-136804"
    , "name": "Ocimum americanum var. pilosum (Willd.) A.J.Paton"
    , "type": "concept"
    , "score": "90"
    , "match": "true"
    , "status": "Synonym"
    , "accepted_names": [
      "Ocimum × africanum Lour."
    ]
  }
]
}
}
```

As per the metadata and single query requests, a callback function name can be specified.

4 LIST OF REFERENCES

1. Google Refine: <http://code.google.com/p/google-refine/>
2. Wikipedia entry for JSONP: <http://en.wikipedia.org/wiki/JSONP>
3. ECAT Name Parser: <http://tools.gbif.org/nameparser/>