NBN Web Services Workshop Exercises

October 2011

Jon Cooper

Resources

- soapUI: www.soapui.org
- EasyPHP: www.easyphp.org
- Nusoap php soap library: http://sourceforge.net/projects/nusoap
- NBN web services reference: www.nbn.org.uk > Guidebooks > Web services guide
- NBN web services WSDL: http://www.nbnws.net/ws_3_5/GatewayWebService?wsdl
- NBN Guide to web service registration keys: http://www.nbn.org.uk/Guidebooks/Web-services-documentation/Resources/registration.aspx
- NBN Web Services Forum: http://forums.nbn.org.uk/viewforum.php?id=15
- NBN site keys (including Vice Counties), designation keys, taxon group keys, etc can be found here: http://www.nbn.org.uk/Guidebooks/Web-services-documentation/Resources/download-documents.aspx
- Biological Records Centre Easy Maps documentation: http://www.brc.ac.uk/resources.htm

Registration keys

These are mandatory in all web service calls, here are some pre-prepared ones for use in different exercises:

- a85d4c129728e58da6ed1b9af84632e15e2b5927 – public access for exercises 1 & 2
- b2918589cb31065453af08d75b9a6f5f7c632bca – variable user access for exercise 3

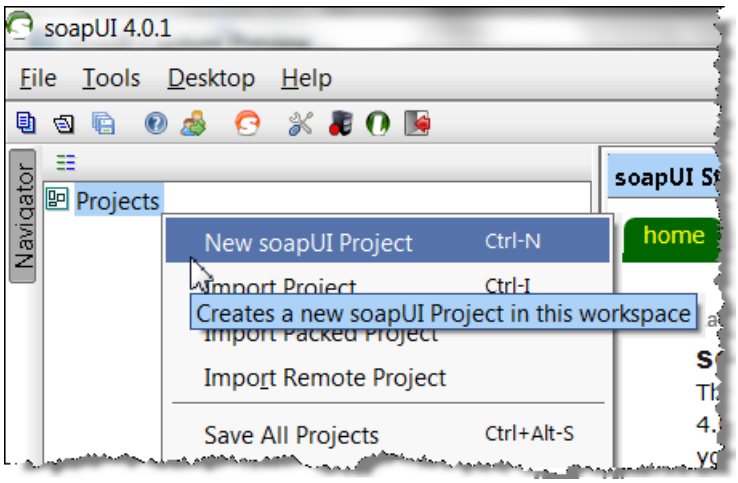
Exercises

Exercise 1. Use SOAPUI to create a Grid Map request

The purpose of this exercise is to become familiar with using soapUI to build and test requests to NBN web services. These requests will later be used in php pages. We will setup a soapUI project using the NBN's WSDL to create example requests for all NBN web services. Then we will focus on the Grid Map web service and create a working request.

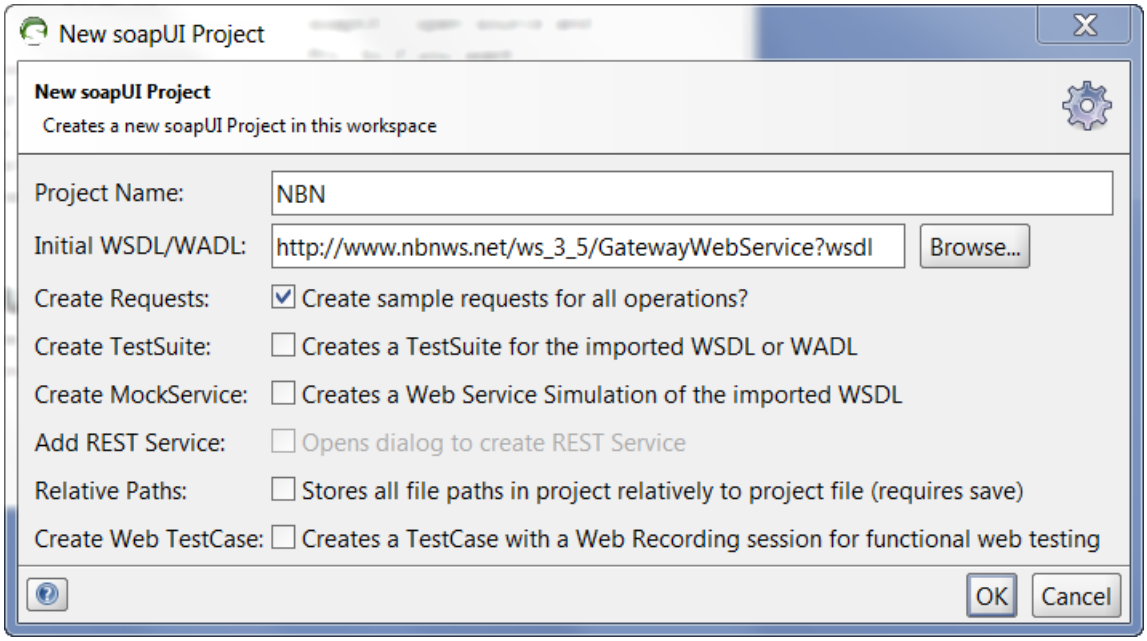
a. Open soapUI

b. Right click Projects and choose New soapUI Project

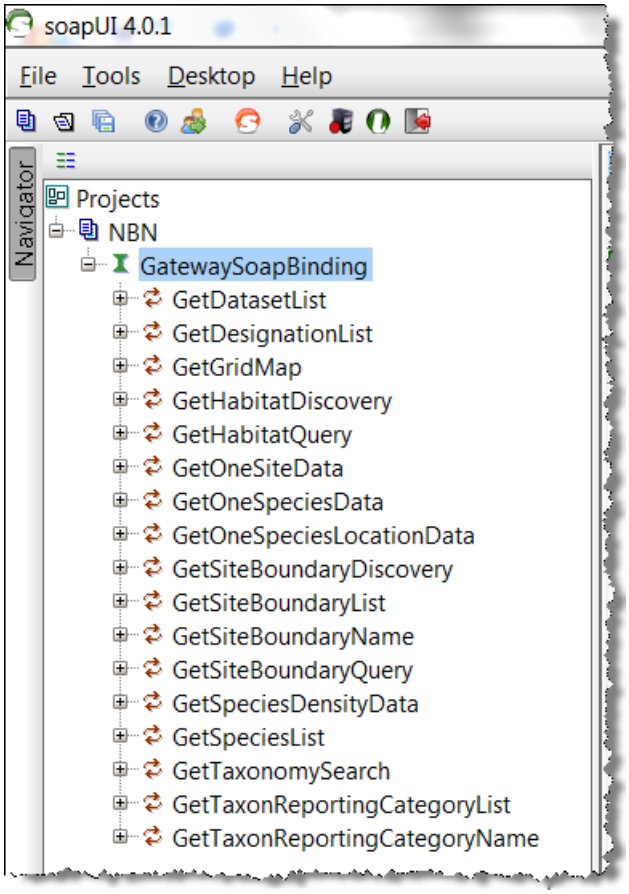


c. In the dialog:

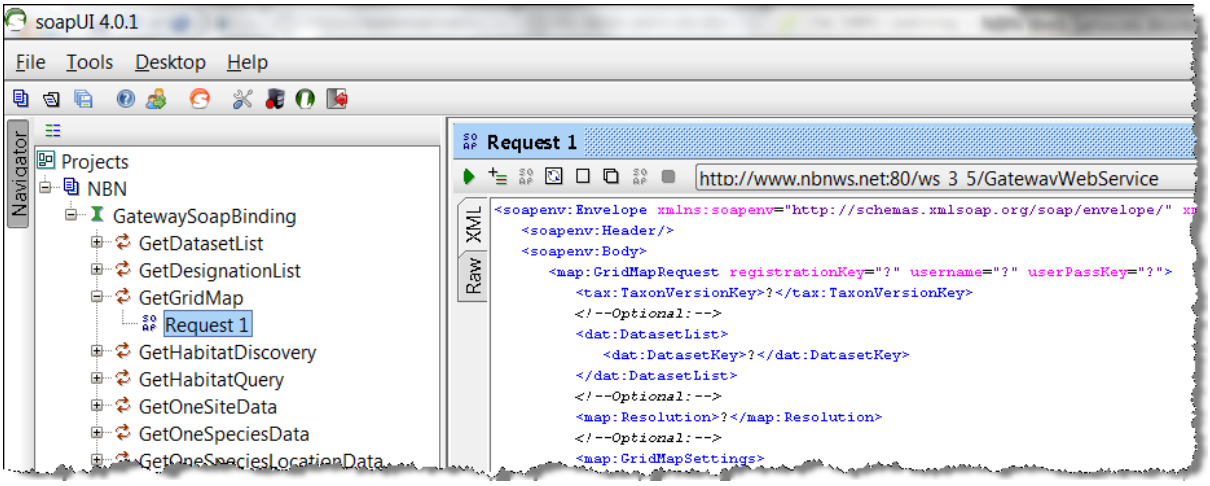
- give it a name (eg NBN)
- Initial WSDL = http://www.nbnws.net/ws_3_5/GatewayWebService?wsdl
- Make sure 'Create Requests' is ticked
- Press OK



d. You should now have a full list of all the NBN web services in the Navigator:



e. Expand GetGridMap and double click Request 1 to open the request in the editor. This displays the request we need to edit. Note that it indicates what elements are optional:



f. Create the simplest possible Grid Map request by making the following edits:

- <map:GridMapRequest registrationKey="a85d4c129728e58da6ed1b9af84632e15e2b5927">
- <tax:TaxonVersionKey>NBNSYS0000005629</tax:TaxonVersionKey>
- Delete all Optional elements (basically everything else except the SOAP elements)
- It should look like this:

g. This request will get a grid map response for the species with the key NBNSYS000005629 – the Norfolk Hawker (*Aeshna isosceles*) (to find other species keys go to the NBN Gateway (http://data.nbn.org.uk) and search for them in the search box). In soapUI, press the green arrow top left to run your request and get the response from the web service. Your result will look like this:

You can see the url to the map image, together with information about the species and the data providers. We will use this in the php example to display the map and acknowledge the data providers.

8. Finally, one of the reasons for using soapUI is to provide a reasonably easy way to create xml requests to copy-and-paste into your php code. The request in soapUI needs all the 'SOAP' bits removing from it before it is ready for use in php (eg in Exercise 2). Copy the request into Notepad++ and make these two edits:
    a. copy all the namespace attributes except 'xmlns:soapenv' from the <soapenv:Envelope> tag into the <map:GridMapRequest> tag
    b. delete the <soapenv:Envelope>, <soapenv:Header> and <soapenv:Body> tags

What you are left with is a request than can be copied directly into php in later exercises, here is an example of before and after the edit – keep it for Exercise 2:

Original soapUI request from before editing:

```
<soapenv:Envelope
   xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:map="http://www.nbnws.net/Map"
   xmlns:tax="http://www.nbnws.net/Taxon"
   xmlns:dat="http://www.nbnws.net/Dataset"
   xmlns:spat="http://www.nbnws.net/Spatial"
   xmlns:sit="http://www.nbnws.net/SiteBoundary">
   <soapenv:Header/>
   <soapenv:Body>
      <map:GridMapRequest registrationKey="a85d4c129728e58da6ed1b9af84632e15e2b5927">
         <tax:TaxonVersionKey>NBNSYS0000005629</tax:TaxonVersionKey>
      </map:GridMapRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

Request after editing, ready for php:

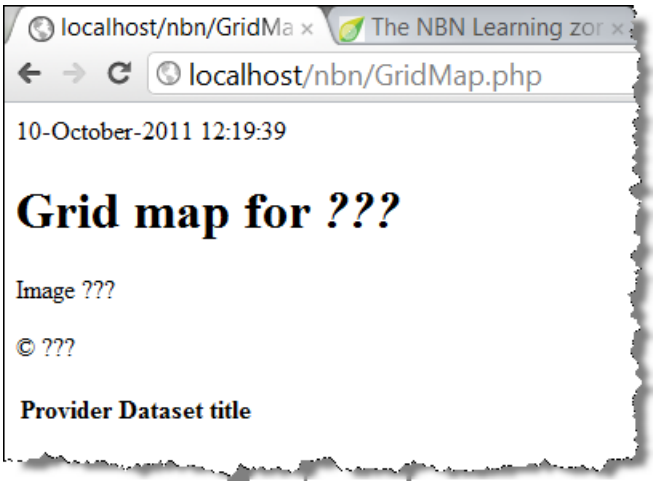
```
<map:GridMapRequest   xmlns:map="http://www.nbnws.net/Map"
        xmlns:tax="http://www.nbnws.net/Taxon"
        xmlns:dat="http://www.nbnws.net/Dataset"
        xmlns:spat="http://www.nbnws.net/Spatial"
        xmlns:sit="http://www.nbnws.net/SiteBoundary"
        registrationKey="a85d4c129728e58da6ed1b9af84632e15e2b5927">
          <tax:TaxonVersionKey>NBNSYS0000005629</tax:TaxonVersionKey>
</map:GridMapRequest>
```

Exercise 2. Step-by-step basic grid map in php

The purpose of this exercise is to use the request from Exercise 1 to create a grid map on a web page using php. It will also show you what needs to be present on the page to conform to the NBN's terms and conditions.

Note: save all your php files in the directory <EasyPHP installation dir>\www\nbn\. Where <EasyPHP installation dir> will be something like C:\Program Files\EasyPHP-5.3.8.0\. All php files can then be viewed in your browser at http://localhost/nbn/filename.php.

1. Make sure EasyPHP is running: Start > EasyPHP - you should see in your system tray.
2. Open Chrome and go to
3. http://localhost/nbn/GridMap.php - you should see something like this, if not something is wrong:



4. In Notepad++, open the source file for this page (<EasyPHP installation dir>\www\nbn\GridMap.php). We are now going to step through the commented sections of this code, adding the Grid Map web service.

   a. Import the nusoap library. This is the php SOAP library we use for all our communication with SOAP web services. It has already been added to the lib folder in EasyPHP and just needs to be imported into the page. It can be freely downloaded from http://sourceforge.net/projects/nusoap

```
//a. Import the nusoap library
require_once('../lib/nusoap.php');
```

   b. Create the client to the NBN web services – the 'true' argument in the constructor tells nusoap that this is a WSDL file:

```
//b. Create the client to the gateway web services
$client = new soapclient('http://www.nbnws.net/ws_3_5/GatewayWebService?wsdl',true);
```

   c. If there is a SOAP fault then print it out:

```
//c. SOAP fault handling
if($client->fault){
        echo "FAULT:  <p>Code: {$client->faultcode} >br />";
        echo "String: {$client->faultstring} </p>";
}
```

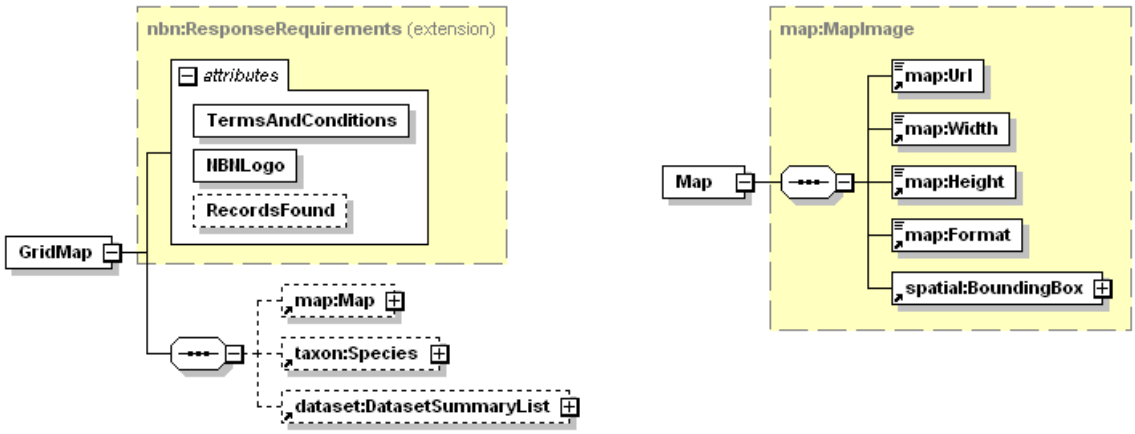
d. Add the 'php ready' xml request we created in exercise 1 step 8:

```
//d. Create the request
$gridMapQuery =
'<map:GridMapRequest registrationKey="a85d4c129728e58da6ed1b9af84632e15e2b5927"
        xmlns:map="http://www.nbnws.net/Map"
        xmlns:tax="http://www.nbnws.net/Taxon"
        xmlns:dat="http://www.nbnws.net/Dataset"
        xmlns:spat="http://www.nbnws.net/Spatial"
        xmlns:sit="http://www.nbnws.net/SiteBoundary">
                <tax:TaxonVersionKey>NBNSYS0000005629</tax:TaxonVersionKey>
</map:GridMapRequest>';
```

e. Add the code that sends the request and gets the response. The name used to identify the web service 'GetGridMap' is the same as we saw in soapUI in exercise 1 just after consuming the WSDL:

```
//e. Send the request and get the response
$response = $client->call('GetGridMap', $gridMapQuery);
```

f. Now retrieve data from the response, ready for displaying on our page. The structure of the response can be reviewed in the web services documentation http://www.nbn.org.uk/Guidebooks/Web-services-documentation/the-web-services/Grid-map/response.aspx:




```
//f. Extract the information from the response that we need for our web page
//See http://www.nbn.org.uk/Guidebooks/Web-services-documentation/the-web-services/Grid-map/response.aspx for info. on how the response is structured

//Map url
$Map = $response['Map'];
$MapUrl = $Map['Url'];

//Species name
$Species = $response['Species'];
$SpeciesName = $Species['ScientificName'];

//List of datasets contributing to the map
$DatasetSummaryList = $response['DatasetSummaryList'];

//The NBN logo and terms and conditions
//Note: attributes of xml elements are represented in php with a leading exclamation mark
$NBNLogo = $response['!NBNLogo'];
$TermsAndConditions = $response['!TermsAndConditions'];
```

g. At last, it is time to display the map. It is required that an OS copyright statement is displayed next to any map images that contain OS data. The text for the copyright statement is in the Terms and Conditions, which is referenced in the response or can be found at http://data.nbn.org.uk/help/popups/generalTerms.jsp. Add the species name, map image and copyright like this:

```
<!--g. Print title and map image with copyright.
Note: the copyright text can be found in the Terms and Conditions retrieved either
from the response or from http://data.nbn.org.uk/help/popups/generalTerms.jsp-->
<h1>Grid map for <i><?php print $SpeciesName; ?></i></h1>
<div>
        <img src="<?php print $MapUrl; ?>" />
        <p>&copy; Crown copyright and database rights 2011 Ordnance Survey [100017955]</p>
</div>
```

In Chrome refresh your page (localhost/nbn/GridMap.php) and make sure you get a map image.

h. The Terms and Conditions also require that the data providers are acknowledged. The following code loops through this information pulling out the provider name and dataset name and putting them into a table. The way it is handled depends on whether just one dataset has been returned, or many.

i.

```
<!--h. Acknowledge the dataset providers-->
<table>
<tr><th>Provider</th><th>Dataset title</th></tr>
<?php
        //Note: if multiple DatasetSummary elements are returned they will be presented
        //as an array.  This needs handling differently from when just one
        //DatasetSummary is returned, which won't be in an array.
        if (isset($DatasetSummaryList['DatasetSummary'][0])) {
                foreach ($DatasetSummaryList['DatasetSummary'] as $DatasetSummary) {
                        $ProviderMetadata = $DatasetSummary['ProviderMetadata'];
                        print '<tr><td>'.$ProviderMetadata['DatasetProvider'].'</td>';
                        print '<td>'.$ProviderMetadata['DatasetTitle'].'</td></tr>';
                }
        } else {
                $DatasetSummary = $DatasetSummaryList['DatasetSummary'];
                $ProviderMetadata = $DatasetSummary['ProviderMetadata'];
                print '<tr><td>'.$ProviderMetadata['DatasetProvider'].'</td>';
                print '<td>'.$ProviderMetadata['DatasetTitle'].'</td></tr>';
        }
?>
</table>
```

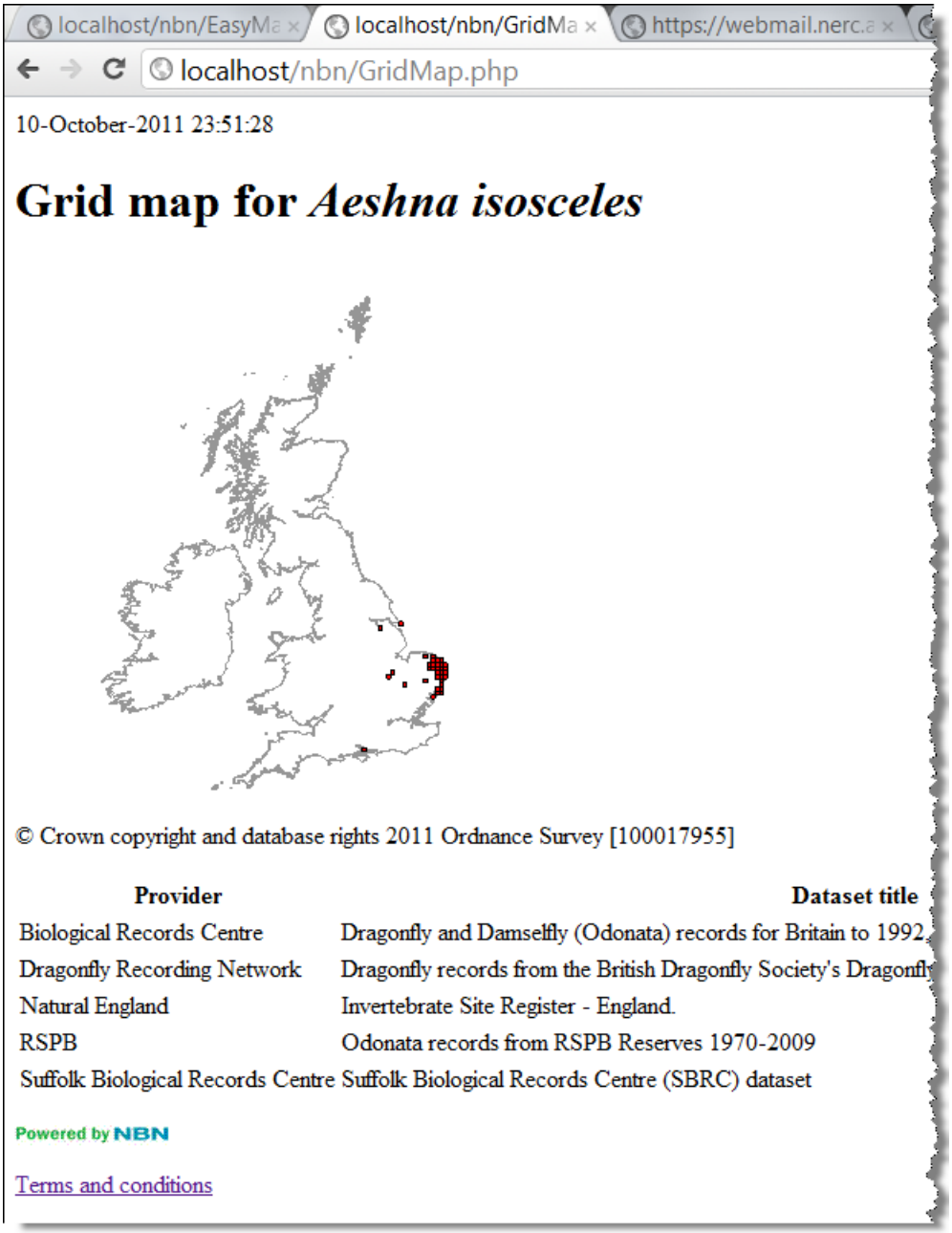
j. It is also a requirement that the NBN logo is displayed and a link to the NBN's terms and conditions:

```
<!--i. Add the NBN logo and Terms and Conditions-->
<p><a href="http://data.nbn.org.uk"><img src="<?php print $NBNLogo ?>"/></a></p>
<p><a href="<?php print $TermsAndConditions ?>">Terms and conditions</a></p>
```

k. Finally, just for development, if you want to look at the request and response you can use this – just change the Boolean flag accordingly:

```
//j. Display the XML of the request and response if you like
$showXML = false;
if($showXML){
        echo '<h2>Request</h2>';
        echo '<pre>' . htmlspecialchars($client->request, ENT_QUOTES) . '</pre>';
        echo '<h2>Response</h2>';
        echo '<pre>' . htmlspecialchars($client->response, ENT_QUOTES) . '</pre>';
}
```

Refresh http://localhost/nbn/GridMap.php and make sure you get something like this:



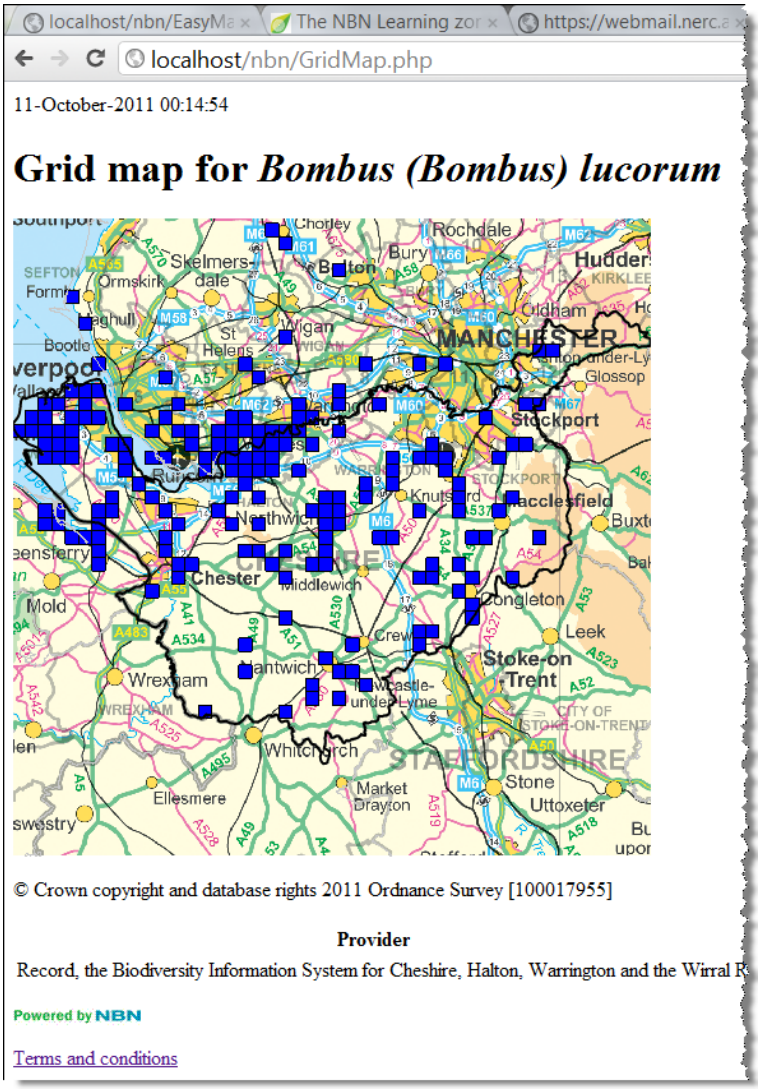
l. You now have the most basic grid map conforming to the NBN terms and conditions. However, if you review the optional query elements that were available to you in soapUI, or take a look at the grid map web services documentation, there are plenty of customisations you can do in the query. Here is an example:

- Change species to *Bombus (Bombus) lucorum* (NHMSYS0000875473)
- zoom to Cheshire
- add an Ordnance Survey background
- display 2km squares
- change the colour of the squares to blue
- only show records for one specific datasets (GA000917)
- make image 500 by 500 pixels

Here is the request:

```
        //d. Create the request
        $gridMapQuery =
      '<map:GridMapRequest
          registrationKey="b2918589cb31065453af08d75b9a6f5f7c632bca"
          xmlns:map="http://www.nbnws.net/Map"
          xmlns:tax="http://www.nbnws.net/Taxon"
          xmlns:dat="http://www.nbnws.net/Dataset"
          xmlns:spat="http://www.nbnws.net/Spatial"
          xmlns:sit="http://www.nbnws.net/SiteBoundary">
        <tax:TaxonVersionKey>NHMSYS0000875473</tax:TaxonVersionKey>
        <dat:DatasetList>
           <dat:DatasetKey>GA000917</dat:DatasetKey>
        </dat:DatasetList>
        <map:Resolution>_2km</map:Resolution>
        <map:GridMapSettings>
           <map:Width>500</map:Width>
           <map:Height>500</map:Height>
           <map:Background>OSMap</map:Background>
           <map:ViceCounty>58</map:ViceCounty>
           <map:FillColour>#0000ff</map:FillColour>
        </map:GridMapSettings>
      </map:GridMapRequest>';
```

And here is the map you should see:



Exercise 3: Use Variable User Access key to send credentials to web service

The purpose of this exercise is to allow a username and password to be entered on the web page and sent to the web service. If the credentials match an NBN Gateway login then that user's access to data is used, otherwise a SOAP fault is returned.

1. In Notepad++ open <EasyPHP installation dir>\www\nbn\ GridMapLogin.php. This is how exercise 2 should have finished, but with some very minor changes and comments. We are going to edit it to add credential handling.
2. Just beneath the body tag, add a form for collecting the username and password:

```
<!—Exercise 3: Add form for entering username and password-->
<form action="http://localhost/nbn/GridMapLogin.php" method="post">
        <p>Username: <input type="text" name="username" /></p>
        <p>Password: <input type="password" name="password" /></p>
        <p><input type="submit" /></p>
</form>
```

3. Just before you create the xml request string, extract the username and password and prepare them for appending to the request. The attribute names used in the request are always 'username' and 'userPassKey', where the 'userPassKey' is an md5 hash of the password. In this example I simply create an empty string if no credentials are found, which will have no effect on the request (see step 4):

```
//Exercise 3
//If a username and password have been entered then create a string
//ready for the request containing the password and hashed password
$username = (is_null($_POST['username']) ? '' : $_POST['username']);
$password = (is_null($_POST['password']) ? '' : $_POST['password']);
$hasCredentials = (($username != '') && ($password != ''));
$credentialsForQuery = '';
if($hasCredentials){
        $credentialsForQuery = ' username="' . $username . '" userPassKey="' . md5($password) . '"';
}
```

4. Now alter your request so that it has a valid 'variable user access' registration key, and also has the credentials from step 3:

```
//Exercise 3: Extend the GridMapRequest to contain the credentials
//Note: the registrationKey must be a valid 'Variable user access' one
$gridMapQuery =
'<map:GridMapRequest registrationKey="b2918589cb31065453af08d75b9a6f5f7c632bca"
        ' . $credentialsForQuery . '
        xmlns:map="http://www.nbnws.net/Map"
        xmlns:tax="http://www.nbnws.net/Taxon"
        xmlns:dat="http://www.nbnws.net/Dataset"
        xmlns:spat="http://www.nbnws.net/Spatial"
        xmlns:sit="http://www.nbnws.net/SiteBoundary">
                <tax:TaxonVersionKey>NBNSYS0000005629</tax:TaxonVersionKey>
</map:GridMapRequest>';
```

5. In Chrome you should see the page shown below. If you don't put any credentials in it will still work, but you will only have access to publicly available data. If you have an NBN Gateway login you can add that to get your enhanced access to data.

localhost/nbn/GridMa × localhost/nbn/GridMa × https://webmail.nerc.a ×

localhost/nbn/GridMapLogin.php

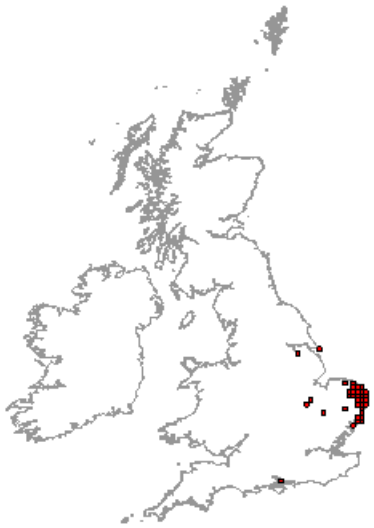
Username: TestUser

Password: ••••••••

Submit

10-October-2011 22:40:57

Grid map for *Aeshna isosceles*



Provider	Dataset title
Biological Records Centre	Dragonfly and Damselfly (Odonata) records for Britain to 1992, fr
Dragonfly Recording Network	Dragonfly records from the British Dragonfly Society's Dragonfly R
Natural England	Invertebrate Site Register - England.
RSPB	Odonata records from RSPB Reserves 1970-2009
Suffolk Biological Records Centre	Suffolk Biological Records Centre (SBRC) dataset

Powered by NBN

Terms and conditions

Exercise 4 EasyMaps

The Biological Records Centre had developed a wrapper around the grid map service to make it very easy for you to obtain a grid map in your web page. This wrapper takes care of acknowledging data providers and adhering to the terms and conditions. All you have to do is include an iFrame to the map in your web page. A link to its documentation can be found here: http://www.brc.ac.uk/resources.htm

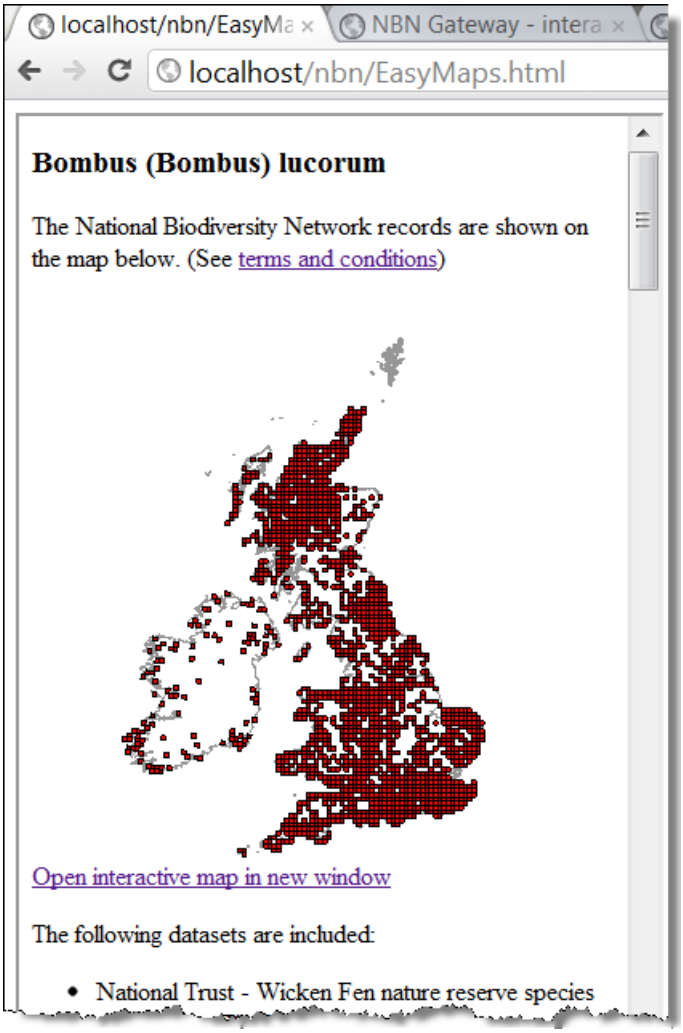
Note: the NBN Gateway team is soon to take over the running of EasyMaps from the Biological Records Centre. So all documentation and hosting will take place through the NBN website.

In this exercise we will create the simplest Easy Map first, and then customise it using the additional parameters available in the web service.

1. In Notepad++ create the new web page <EasyPHP installation dir>\www\nbn\EasyMaps.html and add the following content to it:

```
<html>
<head></head>
<body>
<iframe width="400" height="600"
src="http://www.brc.ac.uk/schemes/NBNWidget/tvkGridmap.aspx?tvk=NHMSYS0000875473"</if
rame>
</body>
</html>
```

2. View the map in Chrome (http://localhost/nbn/EasyMaps.html), which should look like the one below. Note that the species key can be changed to whatever species you require:



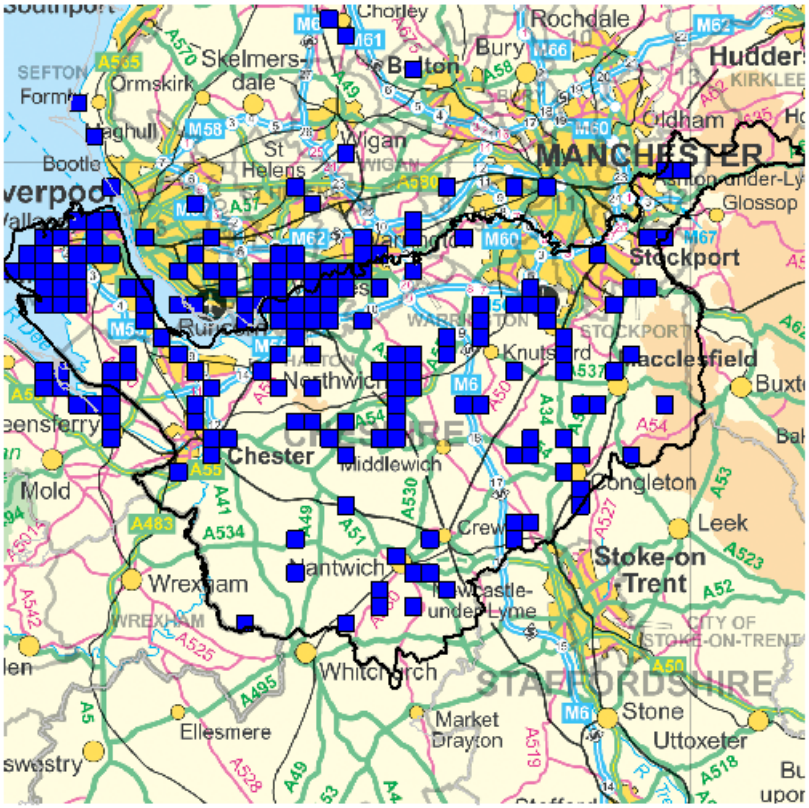
3. There are many options for customising the map as shown in the Easy Maps documentation. Here is an example illustrating some customisations:
    - zoom to Cheshire
    - add an Ordnance Survey background
    - display 2km squares
    - change the colour of the squares to blue
    - only show records for one specific datasets (GA000917)
    - make image 500 by 500 pixels

```
<html>
<head></head>
<body>
<iframe width="600" height="800"
src="http://www.brc.ac.uk/schemes/NBNWidget/tvkGridmap.aspx?tvk=NHMSYS0000875473&res=
2km&vc=58&bg=OS&w=500&h=500&b0fill=0000ff&ds=GA000917"</iframe>
</body>
</html>
```

Your map should look like this:

Bombus (Bombus) lucorum

The National Biodiversity Network records are shown on the map below. (See terms and conditions)



Open interactive map in new window

The following datasets are included:

- Record, the Biodiversity Information System for Cheshire, Halton, Warrington and the Wirral - RECORD Aculeate Hymenoptera Data up to 01/04/2011

Powered by NBN

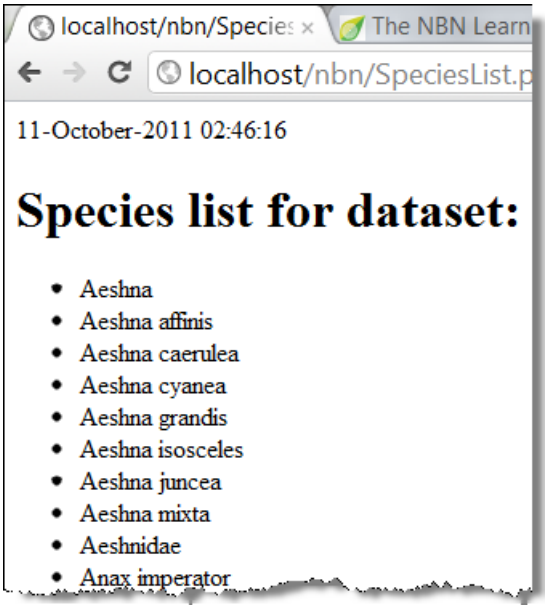
Exercise 5 Species list for a dataset and adding to grid map

The purpose of this exercise is to introduce the Species List web service. In this example a species list will be generated for a single dataset. This will then be integrated into the grid map example to provide a selectable list of species maps.

1. Open soapUI and look at the request for GetSpeciesList. Edit it so that it just provides a list of species in the Dragonfly Recorder Network dataset (datasetKey = GA000012). Note: just use the registration key that gives you public access (a85d4c129728e58da6ed1b9af84632e15e2b5927). You should end up with a request that looks like this:

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tax="http://www.nbnws.net/Taxon"
xmlns:spat="http://www.nbnws.net/Spatial"
xmlns:sit="http://www.nbnws.net/SiteBoundary"
xmlns:map="http://www.nbnws.net/Map"
xmlns:dat="http://www.nbnws.net/Dataset"
xmlns:tax1="http://www.nbnws.net/TaxonReportingCategory">
   <soapenv:Header/>
   <soapenv:Body>
      <tax:SpeciesListRequest
registrationKey="a85d4c129728e58da6ed1b9af84632e15e2b5927">
         <dat:DatasetList>
            <dat:DatasetKey>GA000012</dat:DatasetKey>
         </dat:DatasetList>
      </tax:SpeciesListRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

2. Having got that request working, now open <EasyPHP installation directory>\www\nbn\SpeciesList.php. Each comment in the code indicates what you need to add to make the page produce a species list. Try working through it and getting it displaying a species list for your request. This is what the web page should look like:



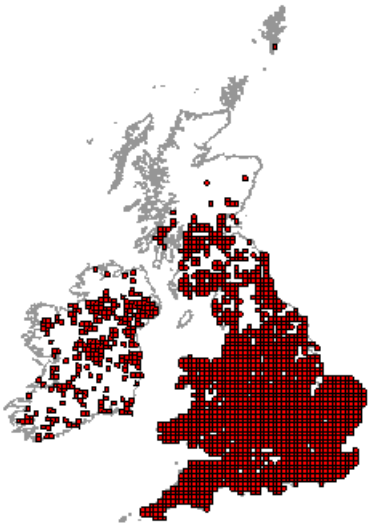
And my working php code looks like this:

```
<html>
<head>
</head>
<body>

<?php
print(date("d-F-Y H:i:s"));
error_reporting(0);

        require_once('../lib/nusoap.php');

        $client = new soapclient('http://www.nbnws.net/ws_3_5/GatewayWebService?wsdl',true);

        if($client->fault){
                echo "FAULT:  <p>Code: {$client->faultcode} >br />";
                echo "String: {$client->faultstring} </p>";
        }

        //Paste your edited species list request from soapUI
        $speciesListQuery =
        '<tax:SpeciesListRequest
                registrationKey="a85d4c129728e58da6ed1b9af84632e15e2b5927"
                xmlns:tax="http://www.nbnws.net/Taxon"
                xmlns:spat="http://www.nbnws.net/Spatial"
                xmlns:sit="http://www.nbnws.net/SiteBoundary"
                xmlns:map="http://www.nbnws.net/Map"
                xmlns:dat="http://www.nbnws.net/Dataset"
                xmlns:tax1="http://www.nbnws.net/TaxonReportingCategory">
          <dat:DatasetList>
             <dat:DatasetKey>GA000012</dat:DatasetKey>
          </dat:DatasetList>
      </tax:SpeciesListRequest>
        ';

        //Add the name of the web service (see list of web services in soapUI navigator)
        $response = $client->call('GetSpeciesList', $speciesListQuery);

        //Extract the information from the response that we need for our web page
        //See http://www.nbn.org.uk/Guidebooks/Web-services-documentation/the-web-
services/Species-List/response.aspx for info. on how the response is structured
        $SpeciesList = $response['SpeciesList'];
        $DatasetTitle =
$resonse['DatasetSummaryList']['DatasetSummary']['ProviderMetadata']['DatasetTitle'];
?>

<!--Add a heading to the page, naming the dataset this species list is from-->
<h1>Species list for dataset: <?php print $DatasetTitle ?></h1>

<!--Iterate through the species list and display their scientific name-->
<ul>
<?php
foreach($SpeciesList['Species'] as $Species){
        print '<li>' . $Species['ScientificName'] . '</li>';
}
?>
</ul>

</body>
</html>
```

3. *If there is time to spare* you could try integrating the species list into the grid map from exercise 2. You are aiming to have a selectable list of species that changes what is displayed in the map. It will look something like this:

11-October-2011 03:33:37

Grid map for *Coenagrion puella*



- Aeshna
- Aeshna affinis
- Aeshna caerulea
- Aeshna cyanea
- Aeshna grandis
- Aeshna isosceles
- Aeshna juncea
- Aeshna mixta
- Aeshnidae
- Anax imperator

And here is the code from my working example:

```
<html><head></head><body>
<?php
        print(date("d-F-Y H:i:s"));
        error_reporting(1);

        require_once('../lib/nusoap.php');

        $client = new soapclient('http://www.nbnws.net/ws_3_5/GatewayWebService?wsdl',true);

        if($client->fault){
                echo "FAULT:  <p>Code: {$client->faultcode} >br />";
                echo "String: {$client->faultstring} </p>";
        }

        //Get the species key, if there isn't one then default to the Norfolk Hawker (NBNSYS0000005629)
        $taxonkey = (is_null($_GET['taxonkey']) ? 'NBNSYS0000005629' : $_GET['taxonkey']);

        //Do the grid map
        $gridMapQuery =
        '<map:GridMapRequest registrationKey="a85d4c129728e58da6ed1b9af84632e15e2b5927"
        xmlns:map="http://www.nbnws.net/Map"
        xmlns:tax="http://www.nbnws.net/Taxon"
        xmlns:dat="http://www.nbnws.net/Dataset"
        xmlns:spat="http://www.nbnws.net/Spatial"
        xmlns:sit="http://www.nbnws.net/SiteBoundary">
                <tax:TaxonVersionKey>' . $taxonkey . '</tax:TaxonVersionKey>
                <dat:DatasetList>
                        <dat:DatasetKey>GA000012</dat:DatasetKey>
                </dat:DatasetList>
        </map:GridMapRequest>';
        $gridMapResponse = $client->call('GetGridMap', $gridMapQuery);
        $Map = $gridMapResponse['Map'];
        $MapUrl = $Map['Url'];
        $Species = $gridMapResponse['Species'];
        $SpeciesName = $Species['ScientificName'];
        $DatasetSummaryList = $gridMapResponse['DatasetSummaryList'];
        $NBNLogo = $gridMapResponse['!NBNLogo'];
        $TermsAndConditions = $gridMapResponse['!TermsAndConditions'];

        //Do the species list query
                $speciesListQuery = '<tax:SpeciesListRequest
        registrationKey="a85d4c129728e58da6ed1b9af84632e15e2b5927"
        xmlns:tax="http://www.nbnws.net/Taxon"
        xmlns:spat="http://www.nbnws.net/Spatial"
        xmlns:sit="http://www.nbnws.net/SiteBoundary"
        xmlns:map="http://www.nbnws.net/Map"
        xmlns:dat="http://www.nbnws.net/Dataset"
        xmlns:tax1="http://www.nbnws.net/TaxonReportingCategory">
                <dat:DatasetList>
                        <dat:DatasetKey>GA000012</dat:DatasetKey>
                </dat:DatasetList>
        </tax:SpeciesListRequest>';
        $speciesListResponse = $client->call('GetSpeciesList', $speciesListQuery);
        $SpeciesList = $speciesListResponse['SpeciesList'];
        $DatasetTitle =
$speciesListResponse['DatasetSummaryList']['DatasetSummary']['ProviderMetadata']['DatasetTitle'];
?>
<h1>Grid map for <i><?php print $SpeciesName; ?></i></h1>
<div>
        <img src="<?php print $MapUrl; ?>" />
        <p>&copy; Crown copyright and database rights 2011 Ordnance Survey [100017955]</p>
</div>

<ul>
<?php
foreach($SpeciesList['Species'] as $Species){
        print '<li><a href="/nbn/GridMapSpeciesList.php?taxonkey=' . $Species['!taxonVersionKey'] . '">' .
$Species['ScientificName'] . '</a></li>';
}
?>
</ul>
<table>
<tr><th>Provider</th><th>Dataset title</th></tr>
<?php
        if (isset($DatasetSummaryList['DatasetSummary'][0])) {
                foreach ($DatasetSummaryList['DatasetSummary'] as $DatasetSummary) {
                        $ProviderMetadata = $DatasetSummary['ProviderMetadata'];
                        print '<tr><td>'.$ProviderMetadata['DatasetProvider'].'</td>';
                        print '<td>'.$ProviderMetadata['DatasetTitle'].'</td></tr>';
                }
        } else {
                $DatasetSummary = $DatasetSummaryList['DatasetSummary'];
                $ProviderMetadata = $DatasetSummary['ProviderMetadata'];
                print '<tr><td>'.$ProviderMetadata['DatasetProvider'].'</td>';
                print '<td>'.$ProviderMetadata['DatasetTitle'].'</td></tr>';
        }
?>
</table>
<p><a href="http://data.nbn.org.uk"><img src="<?php print $NBNLogo ?>"/></a></p>
<p><a href="<?php print $TermsAndConditions ?>">Terms and conditions</a></p>
</body></html>
```

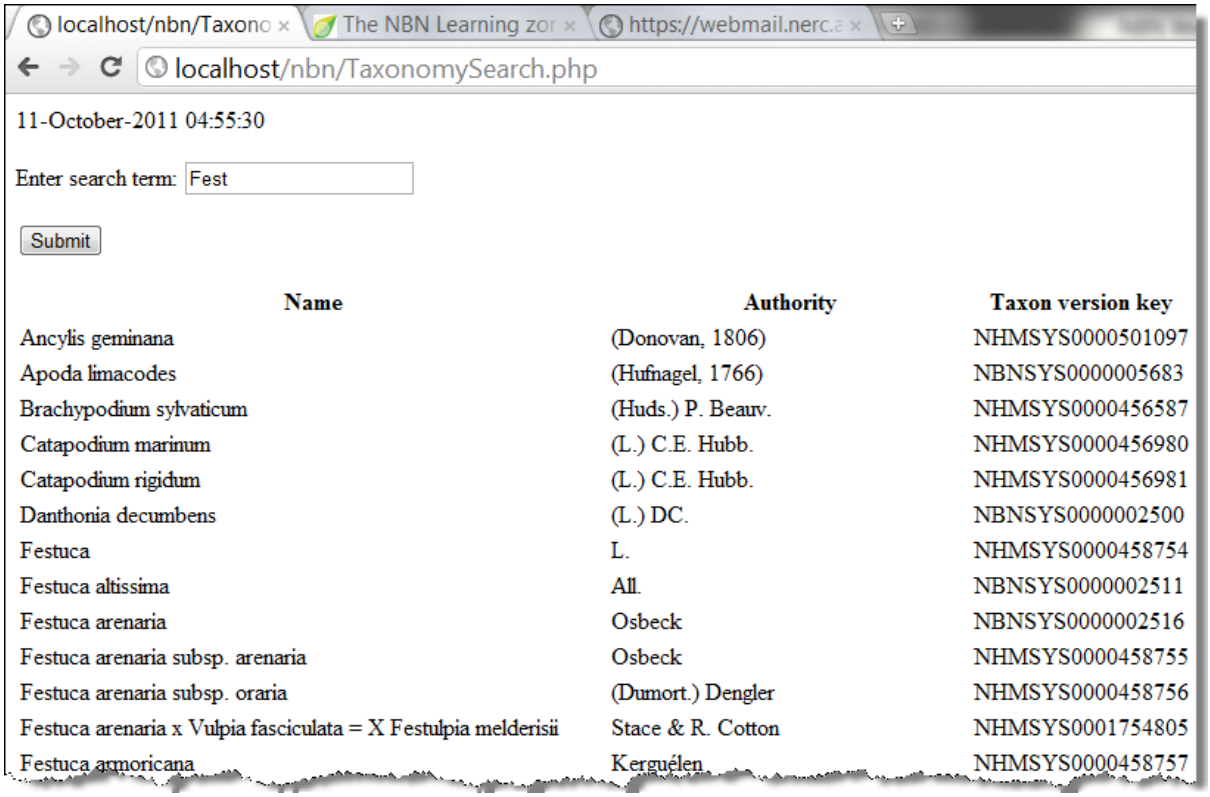
Exercise 6 The taxonomic search web service

The purpose of this exercise is to demonstrate how to search for species and display taxonomic information for them.

1. Open soapUI and create a working GetTaxonomySearch query that uses the SearchTerm element
2. In Notepad++ create the file <EasyPHP installation dir>\www\nbn\TaxonomySearch.php
3. Try to write the code that uses your query from soapUI. One solution is shown below:

```
<html>
<head>
</head>
<body>
<?php   print(date("d-F-Y H:i:s")); ?>

<form action="http://localhost/nbn/TaxonomySearch.php" method="post">
        <p>Enter search term: <input type="text" name="term" /></p>
        <p><input type="submit" /></p>
</form>

<?php
        error_reporting(1);

        require_once('../lib/nusoap.php');

        $client = new soapclient('http://www.nbnws.net/ws_3_5/GatewayWebService?wsdl',true);

        if($client->fault){
                echo "FAULT:  <p>Code: {$client->faultcode} >br />";
                echo "String: {$client->faultstring} </p>";
        }

        //Get the search term
        $term = (is_null($_POST['term']) ? '' : $_POST['term']);

        $taxonomySearchQuery =
        '<tax:TaxonomySearchRequest
                registrationKey="a85d4c129728e58da6ed1b9af84632e15e2b5927"
                xmlns:tax="http://www.nbnws.net/Taxon/Taxonomy"
                xmlns:tax1="http://www.nbnws.net/Taxon">
                        <tax:SearchTerm>' . $term . '</tax:SearchTerm>
        </tax:TaxonomySearchRequest>';

        $response = $client->call('GetTaxonomySearch', $taxonomySearchQuery);

        $nothingFound = ($response['Taxa'] == '');

        if($nothingFound){
                print '<p>Nothing found, please try again</p>';
        }else{
                print '<table><tr><th>Name</th><th>Authority</th><th>Taxon version key</th>';
                if (isset($response['Taxa']['Taxon'][0])) {
                        foreach($response['Taxa']['Taxon'] as $taxon){
                                print '<tr><td>' . $taxon['TaxonName']['!'] . '</td><td>' .
$taxon['Authority'] . '</td><td>' . $taxon['TaxonVersionKey'] . '</td></tr>';
                        }
                }else{
                        $taxon = $response['Taxa']['Taxon'];
                        print '<tr><td>' . $taxon['TaxonName']['!'] . '</td><td>' .
$taxon['Authority'] . '</td><td>' . $taxon['TaxonVersionKey'] . '</td></tr>';
                }
                print '</table>';
        }
?>

</body>
</html>
```

This produces a web page that looks like this:



Name	Authority	Taxon version key
Ancylis geminana	(Donovan, 1806)	NHMSYS0000501097
Apoda limacodes	(Hufnagel, 1766)	NBNSYS0000005683
Brachypodium sylvaticum	(Huds.) P. Beauv.	NHMSYS0000456587
Catapodium marinum	(L.) C.E. Hubb.	NHMSYS0000456980
Catapodium rigidum	(L.) C.E. Hubb.	NHMSYS0000456981
Danthonia decumbens	(L.) DC.	NBNSYS0000002500
Festuca	L.	NHMSYS0000458754
Festuca altissima	All.	NBNSYS0000002511
Festuca arenaria	Osbeck	NBNSYS0000002516
Festuca arenaria subsp. arenaria	Osbeck	NHMSYS0000458755
Festuca arenaria subsp. oraria	(Dumort.) Dengler	NHMSYS0000458756
Festuca arenaria x Vulpia fasciculata = X Festulpia melderisii	Stace & R. Cotton	NHMSYS0001754805
Festuca armoricana	Kerguélen	NHMSYS0000458757