WoRMS and OBIS services

Aleksandra Pawlik
myGrid Team
University of Manchester

VLIZ, 2014-10-06 / 2014-10-08
http://www.taverna.org.uk/

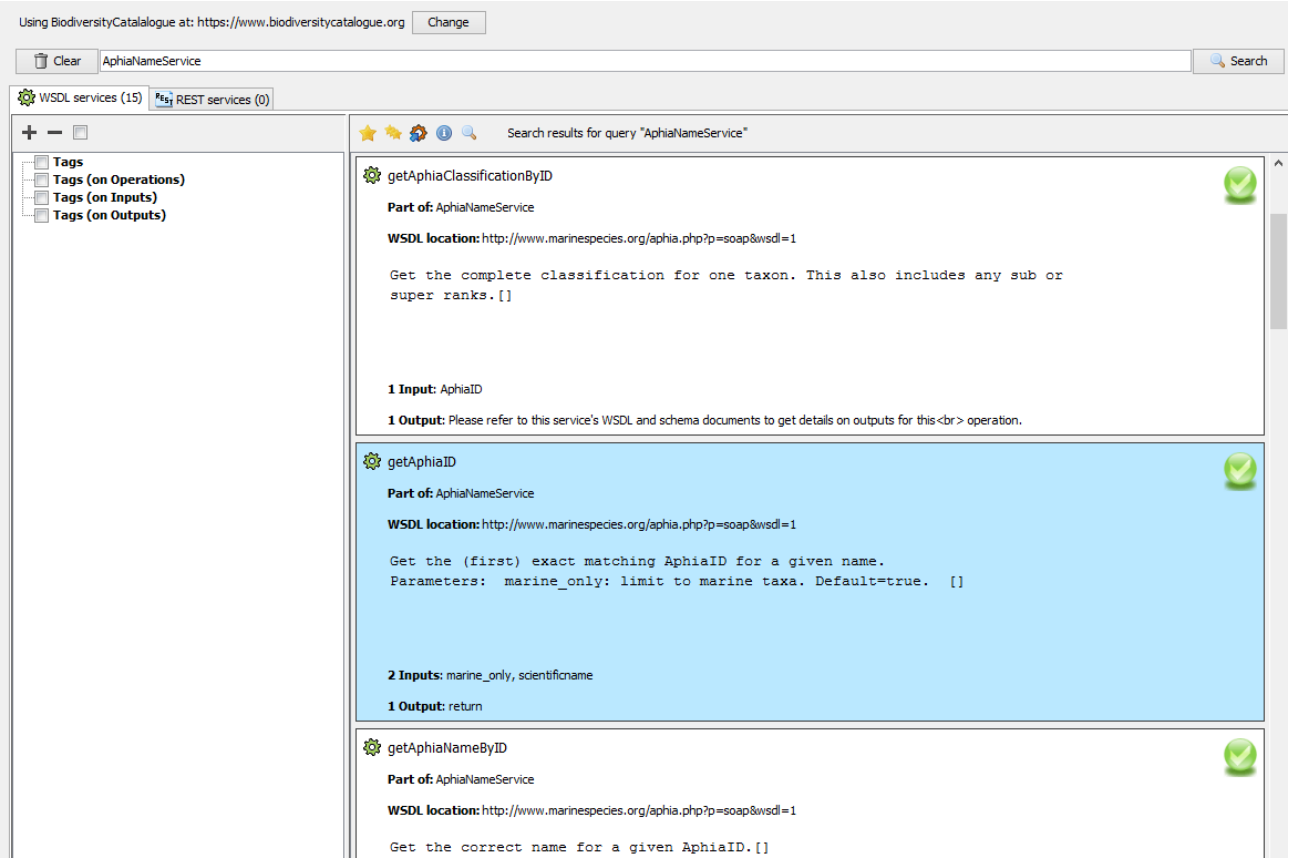
WoRMS services

- WoRMS is the World Register of Marine Species
- In this small exercise, we will use the WoRMS services to find the scientific names of species within a genus
- The naming tools are available as WSDL Web Services called the **AphiaNameService**
- Create a new workflow in Taverna
- In the **Service Catalogue** perspective, search for *AphiaNameService*

AphiaNameService

- You should see 15 WSDL services
- The easiest way to use the Aphia services is by the AphiaID.
- So we will first get the AphiaID for the genus
- Scroll down to getAphiaID
- Right-click on it and select **Add to workflow**
- You will switch back to the design perspective

AphiaNameService search

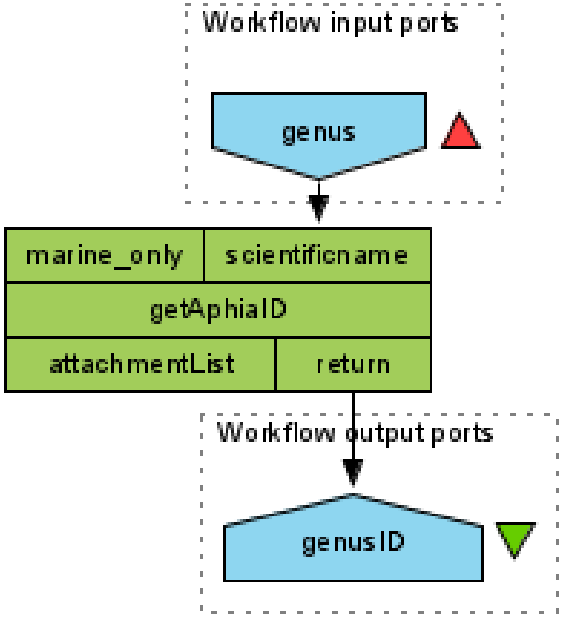


Finding the Aphia ID of the genus

- Connect the input port *scientificname* of the service to a workflow input called *genus*
- Connect the output port return of the service to a workflow output called *genusID*
- Run the workflow with *Isurus*
- You should get a genusID of *105743*

❑Isurus is a type of mackerel shark

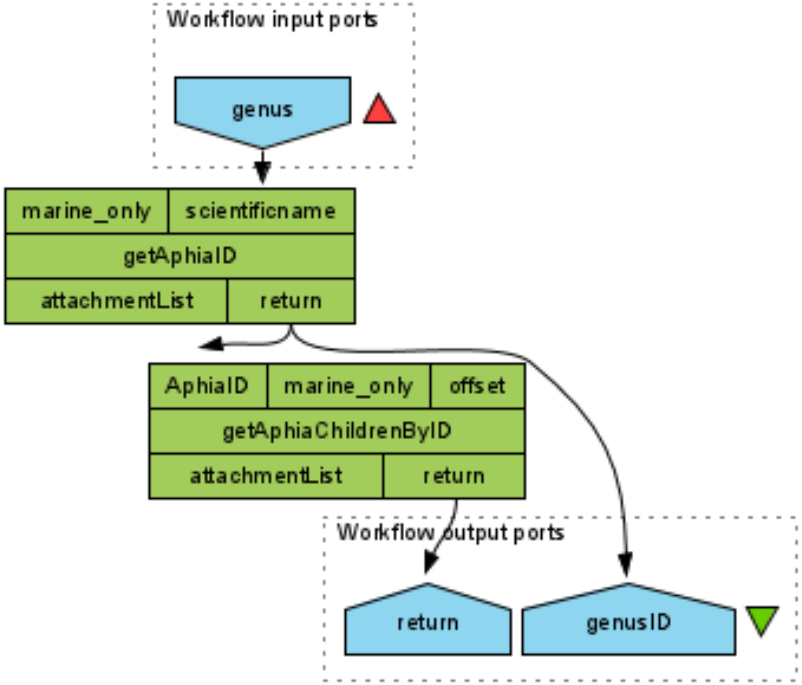
Genus ID workflow



Finding the genus's children

- We now want to find the "children" of the genus
- To do this, switch to the **Service Catalogue** perspective and find the service *getAphiaChildrenByID*
- Add it to the workflow
- Connect the *return* port of *getAphiaID* to the *AphiaID* port of *getAphiaChildrenByID*
- Connect the *return* port of *getAphiaChildrenByID* to a new workflow output port
- Run the workflow

Genus children workflow



Children return value

- If you look at the value returned by the *getAphiaChildrenByID*, it is an XML file - not very friendly
- We are just interested in the scientific names of the children
- There are two ways to extract this data, XML splitters or an XPath service – see the XPath service tutorial
- XML Splitters are used for WSDL services and they understand the XML documents produced/consumed by a service

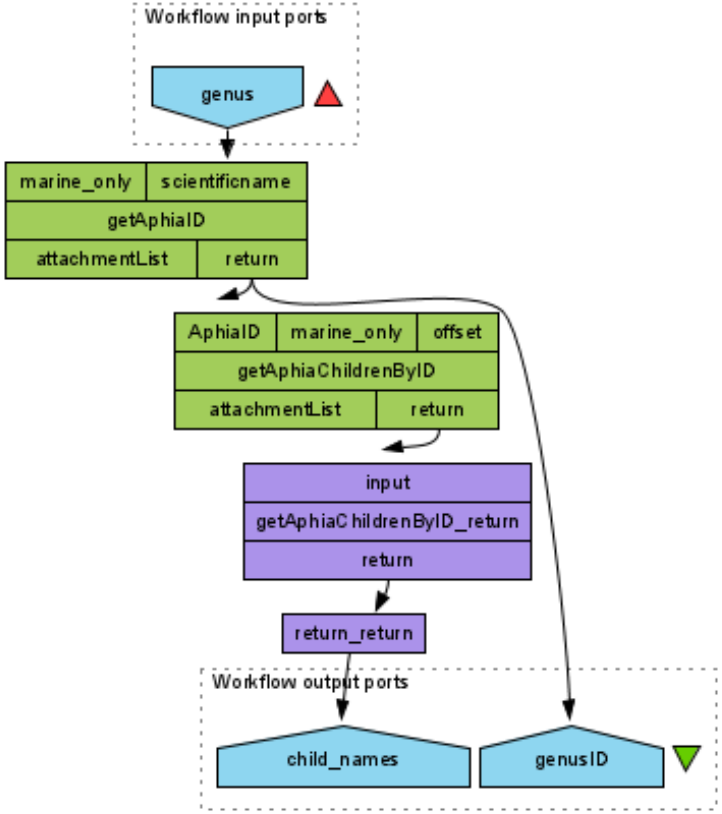
XML Splitters

- Delete the workflow *return* port
- Right-click on *getAphiaChildrenByID* and select **Add XML Output Splitter** – choose *return* to split
- If you connect the *return* port of *getAphiaChildrenByID_return* to a workflow output and run the workflow, you can see it is still an XML document
- Right-click on *getAphiaChildrenByID_return* and add an XML Output Splitter for its *return* port

Obtaining the children names

- Add a new workflow output port called *child_names*
- Connect the *scientificname* port of *return_return* to the *child_names* port
- *Return_return* has a lot of ports so it is easier to make the connection in the workflow explorer or to right-click on the service and choose **Link from output…** and pick *scientificname*
- Run the workflow
- *Child_names* should now get a list of 15 scientific names

Genus children names workflow



WoRMS services

- Congratulations – you have now created a workflow using the WoRMS web services

OBIS Services

- OBIS provides a set of Open Geospatial Consortium (OGC) Web Services at http://www.iobis.org/geoserver
- Access given to
  - Maps
  - Features – “things” on maps, including observations
  - Information about features
  - Legends to maps

Getting a map

- The OBIS services are registered in BiodiversityCatalogue
- We will get a map of the counties in the world
- In Taverna, create a new workflow
- In the Service Catalogue perspective, search for *OBIS*
- Click on **REST services**

GetMap

- Scroll down to *GetMap*
- Right-click and choose **Add to workflow**
- We need to set values for the ports of the *GetMap* service

GetMap parameters

- Set the parameter values for the service
- Right-click on the port name and choose **Constant value**, enter the text value in the dialog.
- Make sure you do not have a newline in the value

Port name	Text value
bbox	-180,-90,180,90
format	image/jpeg
height	330
layers	OBIS:country
styles	country
width	660

What do the parameters mean?

- *bbox* is the minimum longitude, minimum latitude, maximum longitude and maximum latitude as a comma separated list
- *format* says how we want the results returned – image/jpeg means as a JPEG image
- *height* and *width* specify the size of the JPEG image
- *layers* is a comma separated list of layers known to OBIS – OBIS:country is a map of countries

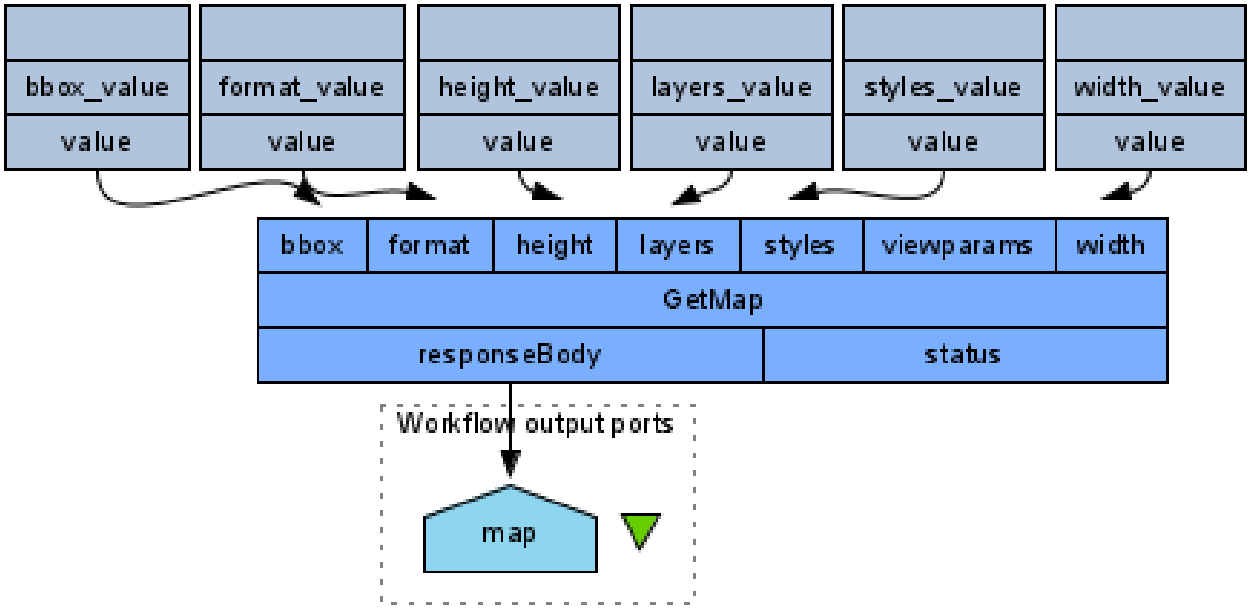
❑You can find the possible layers and their formats at http://www.iobis.org/geoserver/web/ under Layer Preview

- *styles* says for each layer how we want it displayed in the result

Setting the return type

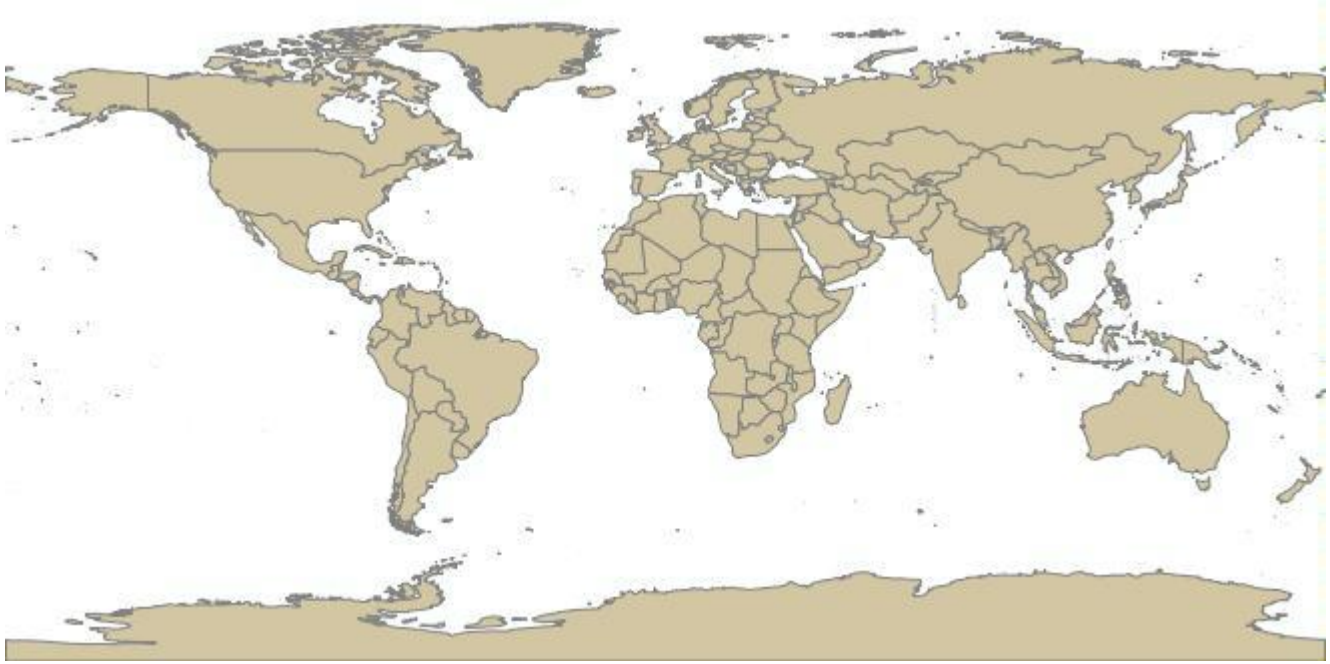
- Although we have set the format parameter to *image/jpeg*, Taverna does not know to expect image/jpeg.
- Right-click on *GetMap* and choose **Configure REST service**
- Change the 'Accept header' to image/jpeg and click **Apply** then **Close**
- Connect the *responseBody* port of *GetMap* to a new workflow output port

GetMap workflow



GetMap results

- When you run the workflow, you should see a world map:



Save GetMap

- Save your workflow – *get_map* is a good name
- We will use the workflow later on for more complicated maps
- You can now close the workflow – **Close workflow** under the **File** or **Taverna** menu

GetFeature

- We are now going to get some species observations
- Create a new workflow
- Add the *GetFeature* service from the service
- In the workflow, merge in the Construct_viewparams workflow from myExperiment – **Insert** -> **Merge workflow**
  - **See: http://www.myexperiment.org/workflows/4483**

❑ We covered merging workflows in the *Nested workflows* tutorial

- Connect the *viewparams* ports

GetFeature parameters

- Set constant values for these ports of *GetFeature* and *construct_viewparams*

Port name	Text value
maxfeatures	20
outputformat	csv
typename	OBIS:drs_with_woa

- Connect the *tname* port of *construct_viewparams* to a new workflow input port

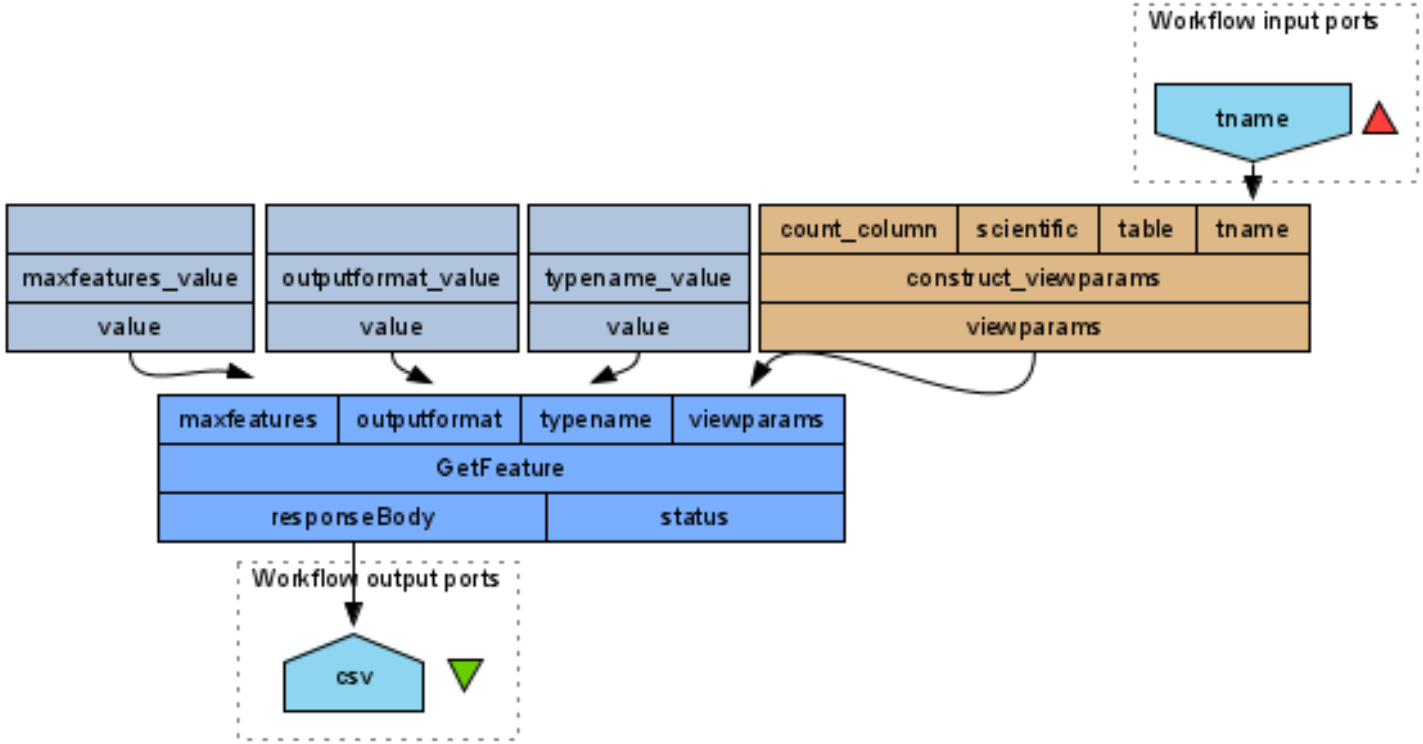
What do the parameters mean?

- *maxfeatures* is the maximum number of features we want to return – so we will return at most 20
- *outputformat* says how we want the results returned – csv means as comma separated values
- *typename* is the layer we want to query – OBIS:drs_with_woa has occurrence information

❑You can find the possible layers and their formats at http://www.iobis.org/geoserver/web/ under Layer Preview

- *viewparams* is an SQL query into the table storing the layer

GetFeature workflow



GetFeature results

- Run the workflow with *tname* as Kogia breviceps
- The results of the GetFeature workflow are (in this case) csv data
- This data can be used, for example, to produce ecological niche models

Gridded maps

- GetMap can be used to retrieve species distribution (gridded) maps
- Open the GetMap workflow you saved before
- We are now going to retrieve data from OBIS:dist_sp – a species distribution layer - as well as OBIS:country
- We need to specify *viewparams* - an SQL query into the table storing the layer

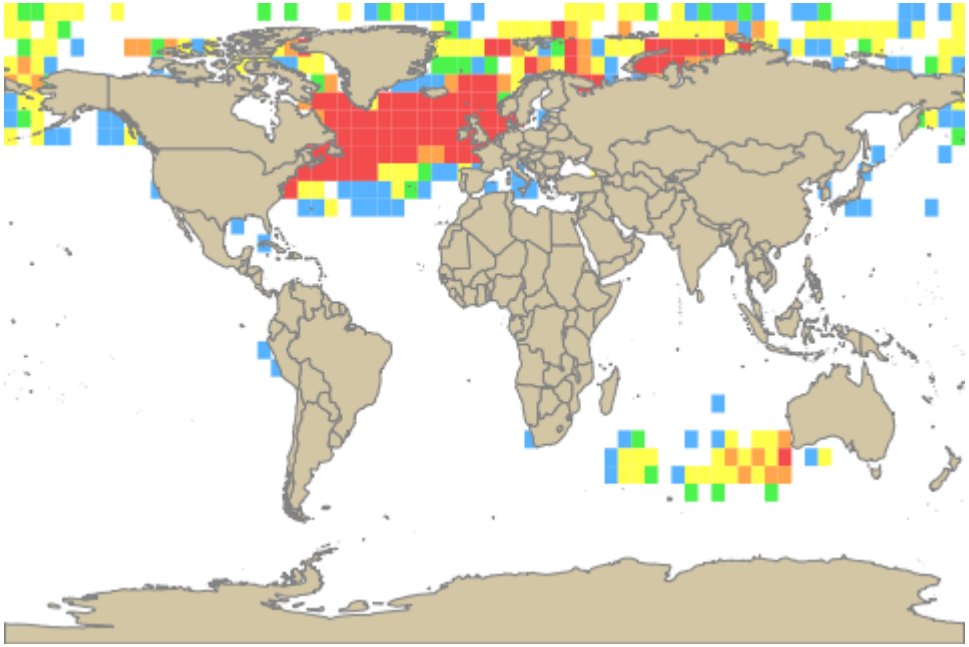
GetMap species distribution parameter changes

- We will need to edit the value for *layers* and also for *styles* – right-click on *layers_value* and select **Edit value…**
- Also change *styles_value*
- Add a constant value for the *viewparams* port

Port name	Text value
layers	OBIS:country,OBIS:dist_sp
style	country, grid_rainbow
viewparams	where:scientific='Calanus finmarchicus';table:dist_sp_5deg;

GetMap species distribution results

- When you run the workflow, you should get a result like:



Species distribution - issues

- The viewparams is difficult to understand
- What do the colours mean?

Constructing viewparams

- To make it easier to construct the *viewparams*, merge in the Construct_viewparams workflow from myExperiment – **Insert** -> **Merge workflow**
  - **See:** http://www.myexperiment.org/workflows/4483

❑ We covered merging workflows in the *Nested workflows* tutorial

❑ Construct_viewparams is a helpful workflow that we have created for you

- Delete the *viewparams_value* service

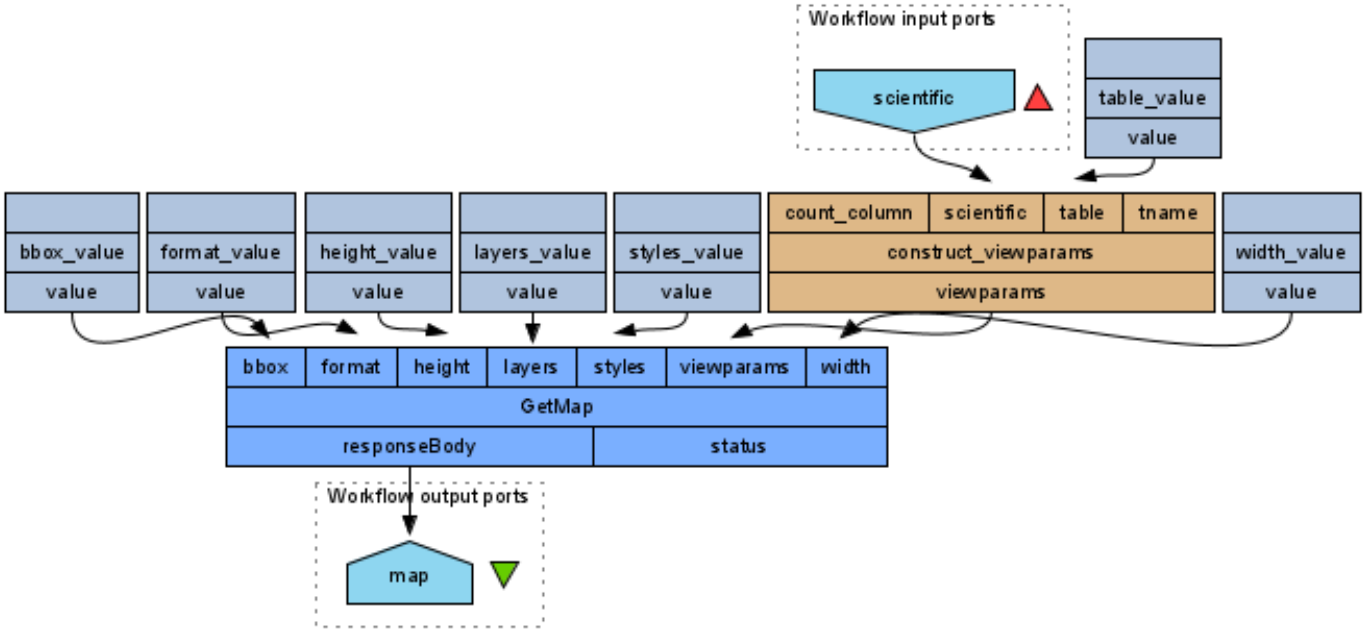
Constructing viewparams - 2

- Set a constant value for the table port of *construct_viewparams* service to

Port name	Text value
table	dist_sp_5deg

- Connect the *scientificname* port of viewparams to a new workflow input port
- Connect the two *viewparams* ports

Species distribution workflow



Obtaining species distribution for a species

- You can now run the workflow and specify the species
- Try Kogia breviceps – you should get the same result as before
- Try Limulus polyphemus to see different results

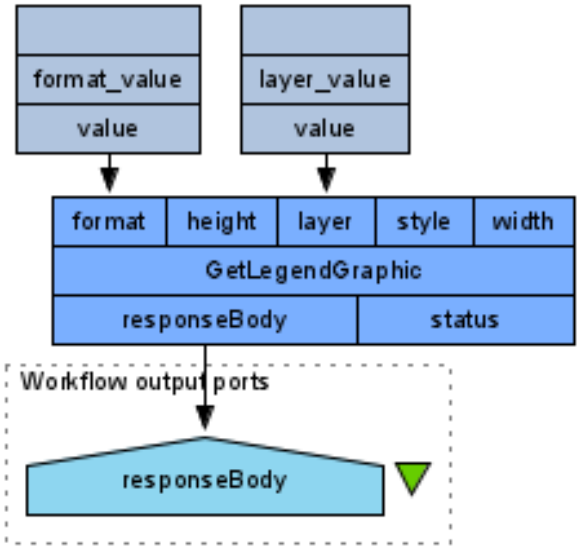
GetLegendGraphic

- We can find out what the colours mean with the *GetLegendGraphic* service
- Create a new workflow and add GetLegendGraphic from the Service Catalogue perspective
- Set constant values for its input ports

Port name	Text value
layer	OBIS:dist_sp
format	image/png

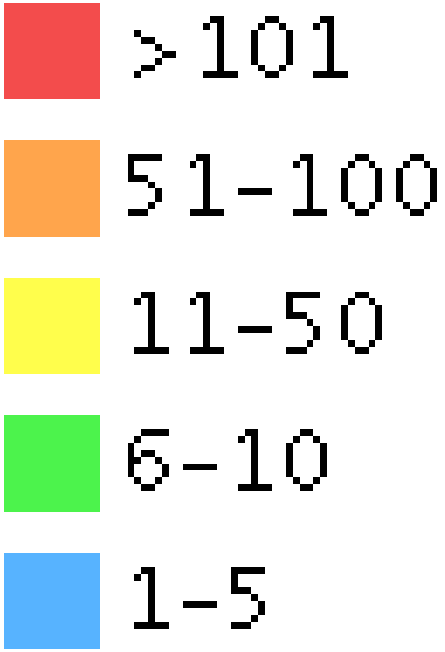
- Connect the responseBody port of GetLegendGraphic to a new workflow output port

GetLegendGraphic workflow



GetLegendGraphic - results

- Running the workflow will give:



OBIS services

- Congratulations – you are now able to
  - retrieve simple maps
  - retrieve species occurrence information
  - retrieve gridded maps for specific species
  - see what the colours on a map mean