Paris CESP Workshop April 2019 - Exercises

This document describes the exercises to be undertaken during the practical sessions of the workshop. The attendees of the workshop will be divided into groups. These tasks will be performed against the two virtual machines being used by each group. Each group will work together on their shared virtual machine to add data and change the configuration.

For each practical, the group should nominate a **single Driver for the session - the person enter commands over SSH or uses the UI.** The groups can then complete the exercises in a peer-programming style.

The virtual machines used where setup using the inventories here[1]. The virtual machines used for testing where two t2.xlarge (16GB, 4 CPU) with Ubuntu 16 from Amazon.

Monday Afternoon: Loading spatial layers

Prerequisite: A running spatial server with geoserver, postgres, spatial-service.

Check for signs of life - Spatial VM

Objective: Initial checks to see the spatial virtual machine is up and running ok.

1. Add entries in `/etc/hosts` on your laptop for your servers (see whiteboard for IP addresses)[2].
   ```
   123.123.123.123 spatial.livingatlas.org
   123.123.123.124 demo.livingatlas.org
   ```

2. Check you can SSH to spatial.livingatlas.org machine with the supplied PEM file (note: You may need to sudo chmod 400 <pem file> before you attempt to ssh.):

   ```
   ssh -i /path/to/key.pem ubuntu@spatial.livingatlas.org
   ```

[1] https://github.com/AtlasOfLivingAustralia/ala-install/tree/master/ansible/inventories/workshop

[2] These entries will also be in the /etc/hosts file of the virtual machines so they can see each other.

3. Check the follow are running:
    1. Postgres $ `ps -ef | grep postgres`
    2. Tomcat $ `ps -ef | grep tomcat`
    3. Nginx $ `ps -ef | grep nginx`
4. Check the following:
    a. Change to postgres account: $ `sudo su postgres`
    b. Postgres - SSH to the machine and open psql session (`$ psql`). Connect to the "`layersdb'` database (`\c layersdb`)
    c. Geoserver - http://spatial.livingatlas.org/geoserver
    d. Spatial Service - http://spatial.livingatlas.org/ws
    e. Spatial Portal - http://spatial.livingatlas.org/

Load Shapefile into spatial service

Objective: Load a shapefile (polygon) spatial layer, and access it via web services and the spatial portal UI.

1. Using a browser go to: http://spatial.livingatlas.org/ws
2. Upload shapefile (SHP) in zip file using the "Manage layer uploads". Your shapefile should be zipped and contain at least SHP, DBF, PRJ and SHX files. You can download an example shapefile here[3][4]
3. Create the "Layer" using the UI
4. View the DBF fields of the Shapefile by clicking on the map[5] - view the layers and click "edit layer"
5. Define one or "Fields" for your layer. These are the properties used in sampling. Make sure to:
    a. Select a sensible field from the layer.
    b. Change the display name. This is how the layer can be found in the spatial portal
    c. Edit the layer and set values for the properties `Classification1` and `Classification2`. This is visible in a tree hierarchy in the spatial portal.
6. Monitor progress of Field generation using the "Tasks" UI
7. Refresh intersect config: http://spatial.livingatlas.org/ws/intersect/reloadConfig
8. Verify the layer and fields in spatial service & geoserver and web services are working:
    a. Use the geoserver UI http://spatial.livingatlas.org/geoserver
        i. Preview the layer in OpenLayers
        ii. View configuration for layers (username/password = admin/geoserver)
    b. Test the following JSON services:

[3] https://spatial.ala.org.au/ws/layers/view/more/a2000_global
[4] Example polygon layer https://drive.google.com/open?id=1PEZbm03yWBeC-PRzhKJYaAACXGz-Sf79
[5] This requires the PRJ file to included

        i. Check your layer appears in the layer list: http://spatial.livingatlas.org/ws/layers
        ii. Check you field(s) appears in the fields list: http://spatial.livingatlas.org/ws/fields
        iii. List the objects in the field http://spatial.livingatlas.org/ws/objects/<FIELD_ID>
        iv. http://spatial.livingatlas.org/ws/objects/`cl10003`
        v. Retrieve the details of single object http://spatial.livingatlas.org/ws/object/<OBJECT_ID>
        vi. Download the KML for an object: http://spatial.livingatlas.org/ws/shape/kml/<OBJECT_ID>
        vii. Check point intersection works for your field(s) http://spatial.livingatlas.org/ws/intersect/<FIELD_ID>/<LATITUDE>/<LONGITUDE>

9. Test the uploaded layer in the spatial portal
    a. Use “Add to Map > Areas > Gazetteer Polygon”
    b. Use “Add to Map > Areas > Select from Polygon Layer”

Load a grid file into spatial service

Objective: Load a grid (raster) spatial layer, and access it via web services and the spatial portal UI.

Examples can be downloaded from: http://www.worldclim.org/current. Look for the BioClim layers. [6] For the purposes of the practical, use a 10 minute grid (smaller). Note: the zips from this website will fail. Uploads require a zip with a single set of HDR, BIL, PRJ and bil.aux.xml files).

But you can for example download the http://biogeo.ucdavis.edu/data/climate/worldclim/1_4/grid/cur/prec_10m_bil.zip, unzip a single set of .hdr and .bil files, such as prec1.hdr and prec1.bil and then zip that up and use that file.

1. Using a browser go to: http://spatial.livingatlas.org/ws for the admin screen
2. Upload grid file (must contain HDR, BIL and PRJ files) in zip file using the “Manage layer uploads”. For GeoTIFF files, use gdal_translate before hand[7].
3. Create the “Layer” using the UI

[6] Example grid layer: https://drive.google.com/open?id=1Slquk2wAR_tc0_xq4Mb9YtjzzSWH0Pnh

[7] `gdal_translate -of EHdr -ot Float32 wc2.0_bio_10m_01.tif wc2.0_bio_10m_01.bil`

4. Create a “Field” associated with this layer using the UI
5. Monitor progress of Field generation using the “Tasks” UI
6. Refresh intersect config: http://spatial.livingatlas.org/ws/intersect/reloadConfig
7. Verify the layer and fields in spatial service & geoserver and web services are working:
    a. Use the geoserver UI http://spatial.livingatlas.org/geoserver
        i. Preview the layer in OpenLayers
    b. Test the following JSON services:
        i. Check you layer appears in the layer list:
        http://spatial.livingatlas.org/ws/layers
        ii. Check you field(s) appears in the fields list:
        http://spatial.livingatlas.org/ws/fields
        iii. Check point intersection works for your field(s)
        http://spatial.livingatlas.org/ws/intersect/<FIELD_ID>/<LAT>/<LNG>

Tuesday Morning: Sampling and UI

Prerequisite: A running demo server with collectory, biocache tools and services.

Check for signs of life - Demo VM

Objective: Initial checks to see the demo virtual machine is up and running ok.

1. Add entries in `/etc/hosts` for your ALA demo server
2. Check you can SSH to this machine with the supplied PEM file
3. Check the following is running:
   a. Demo landing page - http://demo.livingatlas.org/
   b. Collectory - http://demo.livingatlas.org/collectory
   c. Biocache service - http://demo.livingatlas.org/biocache-service
   d. Biocache UI - http://demo.livingatlas.org/ala-hub
4. Check the biocache CLI tool is working. To do this, SSH to the machine and run the command “`biocache --help`” up

Sample and Index against layers

Objective: Sample against the shape and raster layers loaded in the previous steps.

1. Check the configuration for biocache-store is pointing to sampling URL:
   a. $ `/data/biocache/config`
   b. $ `more biocache-config.properties`
2. Look for `spatial.layers.url=http://spatial.livingatlas.org/ws/fields`
   a. `You can do this by connecting with ssh to the livingatlas-demo server you are using and issue command “biocache config | grep -e ".*layers.*url"`
3. Load the DwCA into the collectory, For purposes of the workshop, please choose a small dataset (<50k records) just for speed, preferably Mammals (as this affects a later step with taxonomy).
   a. For IPT users:
      i. Start here: http://demo.livingatlas.org/collectory/admin/
      ii. Create a data provider and point at IPT instance by setting website URL to IPT URL e.g. https://ipt.gbif.es
      iii. Click “Update data resources” button
      iv. Note: check the unique fields. Typical values are “`catalogNumber`” or “`occurrenceID`”. The default is “`catalogNumber`”

        v. Find a UID e.g. “dr123” to load
    b. For Non-IPT users:
        i. Start here: http://demo.livingatlas.org/collectory/admin/
        ii. Create a data resource
        iii. Upload your DwCA
        iv. Note: check the unique fields. Typical values are “`catalogNumber`” or “`occurrenceID`”. The default is “`catalogNumber`”
4. Load a DwCA into the biocache using command line tool
    a. Use the command “`biocache load dr123`”
    b. Validate the data has been loaded using Cassandra command line tool. Use the tool “`cqlsh`” on the command line.
        i. Connect to “`occ`” keyspace using “`use occ;`”,
        ii. Run “`select * from occ;`”.
5. Process the data resource - Use the command “`biocache process -dr dr123`”
6. Sampling - Use the command “`biocache sample -dr dr123`”
7. Indexing - Use the command “`biocache index -dr dr123`”
8. Test the indexing was successful by:
    a. Viewing the SOLR admin console[8] http://demo.livingatlas.org:8983 - this might be blocked by the firewall, so check the footnote :) for the tunnel command to run first and then use http://localhost:8983 to view the SOLR UI
    b. View the results in biocache services
        i. http://demo.livingatlas.org/biocache-service/occurrences/search?q=*:*
    c. Test with an Area Report in the Spatial Portal
        i. Search for Gazetteer Polygon e.g. “Queensland”
        ii. Tools > Area Report - and follow wizard

Configure Biocache UI

Objective: Expose the sampled field as a facet in the biocache UI. Alter the UI to change map and charts.

1. Add facet in biocache-hubs to expose sampling - see `/data/ala-hub/config/grouped_facets_default.json` (add an entry to one of the sections)
2. Add a chart for your facet (see `/data/ala-hub/config/charts.json`
3. ~~Add a WMS overlay to the map (see `/data/ala-hub/config/overlays.json`) - use this file as a template~~
4. Restart tomcat and test with search

[8] If viewing 8983 is an issue, we can use SSH tunnel like so: `ssh -i ~/.ssh/cesp-2019.pem -Nf -L 8983:127.0.0.1:8983 ubuntu@demo.livingatlas.org”.`. With this, we can view the SOLR admin console on your laptop on http://localhost:8983

Setup Regions

Objective: Install the Regions App and expose the sampled polygon layers in menu.

1. Install Regions app with ansible on the demo server. You'll need to produce an inventory to do this. There is an inventory here for use in the workshop for installing on demo.livingatlas.org:
https://github.com/AtlasOfLivingAustralia/ala-install/blob/master/ansible/inventories/workshop/regions-livingatlas.yml

2. Due to an ansible script bug, you'll need to manually add the following into `/etc/nginx/sites-enabled/demo.livingatlas.org.conf` on demo.livingatlas.org:

```
location /regions {
  if ($blocked_user_agent) {
    return 444;
  }
  # proxy_set_header Host $host;
  proxy_set_header Host demo.livingatlas.org;
  proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
  proxy_set_header X-Forwarded-Proto $scheme;
  proxy_set_header X-Forwarded-Port $server_port;an



  proxy_read_timeout 10m;
   proxy_pass http://127.0.0.1:8080/regions;
}
```

3. You need to add CORS headers for spatial.livingatlas.org in the geoserver 'location'. Edit the file `` `/etc/nginx/sites-enabled/spatial.livingatlas.org.conf` `` and modify the block (add text in bold):

```
location /geoserver {
        if ($blocked_user_agent) {
            return 444;
        }
        # proxy_set_header Host $host;
        proxy_set_header Host spatial.livingatlas.org;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header X-Forwarded-Port $server_port;
        proxy_read_timeout 10m;
        proxy_pass http://127.0.0.1:8080/geoserver;
        add_header 'Access-Control-Allow-Origin' '*' always;
        add_header 'Access-Control-Allow-Methods' 'GET, OPTIONS' always;
        add_header 'Access-Control-Allow-Headers' 'Accept,Authorization,Cache-Control,Content-Type,DNT,If-Modified-Since,Keep-Alive,Origin,User-Agent,X-Requested-With' always;
    }
```

And restart Nginx with `sudo service nginx restart`.

4. Configure sensible defaults for the initial maps (the default is scoped to Australia), on the parameter `map.bounds` at `/data/regions/config/regions-config.properties`
   Exemple : `map.bounds=[38,-9,42,-7]`
5. Configure `menu-config.json` to use new layer in the menu. This file is found in `/data/regions/config`. E.g.

```
[
  {
    "label": "China National Boundaries",
    "fid": "cl10004"
  },
  {
    "label": "Australian States",
    "fid": "cl10006"
  },
  {
    "label": "Canadian National Parks",
    "fid": "cl0007"
  }
]
```

Configure the Spatial Portal

Objective: Configure the spatial portal to change menu items, base layers, WMS servers

1. Configuration options
   a. Change default bounding box (see ‘defaultareas” in YML in `/data/spatial-hub/config/spatial-hub-config.yml`) -> see the

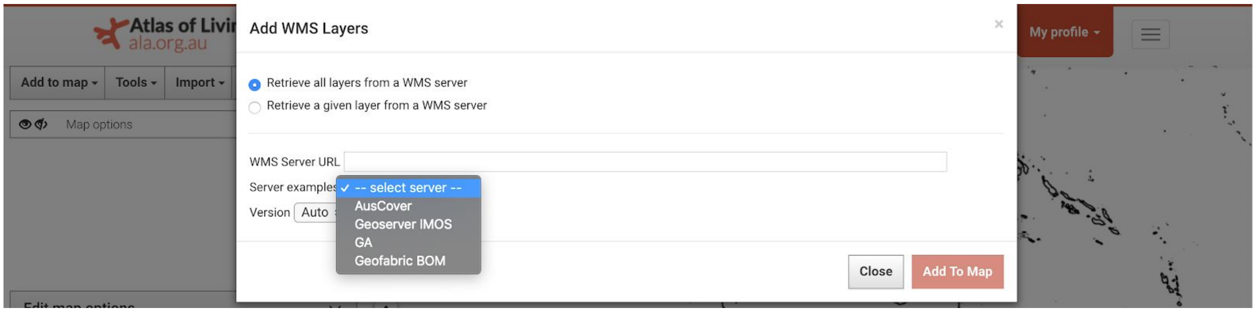
default_areas section and specify your own coordinates. Then restart tomcat7. You can see the result by clicking on Tools -> Add area report.

b. Change the default base layer.
(Copy-paste the outline block, and change name, url and layer name)

c. Add your own WMS servers - (see ’presetWMSServers’ in YML in `/data/spatial-hub/config/spatial-hub-config.yml`)

If your json request ends with a 401-Unauthorized error, add your server host to the allowproxy line in `spatial-hub-config.yml :`

```
allowProxy:
    server:"spatial.livingatlas.org;data.auscover.org.au;data.auscover.org.au:80;geoserver.imos.org.au;www.ga.gov.au/gis;geofabric.bom.gov.au"
```



2. Skinning the Spatial Portal - see here[9] for skinning options for the portal.

[9] https://github.com/AtlasOfLivingAustralia/spatial-hub

Tuesday Afternoon: Using your own Taxonomy with the Atlas

Build a name index

Objective: Building your own name matching index, to use your own taxonomy.

1. Details of the required structure are here
2. SCP the zipped DwCA to the demo.livingatlas.org into the directory `/data/lucene`, and expand the zip.
3. Build the lucene index from DwCA using the nameindexer command line tool install on the demo server. Type "`nameindexer --help`" to get some instructions.
4. Test the new name index with "`nameindexer -testSearch Sorex`"[10]
5. Test the biocache by processing a single record using "`biocache process-single <Record UUID>`"
6. Re-process your data in the biocache "`biocache process -dr dr123`"
7. Re-index your data to see the changes "`biocache index -dr dr123`"

Wednesday Afternoon: Species pages & species-level traits

Objective: Load data into BIE and Lists tool.

1. Load DwCA with an extension into BIE using the admin interface: http://demo.livingatlas.org/bie-index/admin. An example DwCA archive is here[11]. This is s subset of GBIF BackBone containing only Mammals. To load the archive, scp the file to `/data/bie/import/`, and extract. Note: you made to change the permissions of the uploaded archive[12]. Once the DwC archive has loaded, you need to swap the SOLR cores using the SOLR web interface (see above for how to create SSH tunnel to see SOLR interface). In SOLR admin, chose "Core admin" and then select the BIE core and then the "swap" button.
2. Load CSV of trait information into lists tool. Upload a CSV of data into http://demo.livingatlas.org/specieslists. Mark the list as "Authoritative" for it to appear on species pages.
3. Use the list in the spatial portal to load a number of species in one layer.

[10] Sorex is a genus name for shrews

[11] https://drive.google.com/open?id=1zWsClOR9BLxt1owI2DRrFtwz0nG9nQIS

[12] `chown -R tomcat7:tomcat7 /data/bie/import/dwc-a`

4. Index against the list.