International ALA Workshop

Santiago Martínez de la Riva Vivanco
Unidad de Coordinación GBIF España
sama@gbif.es
www.gbif.es
Real Jardín Botánico 03/09/16

INDEX

- Current situation of GBIF Spain data portal.
- Biocache-hub:
    - Configuration files.
    - Customizations.
    - Practical session.
- Taxonomy:
    - Working with Taxonomy:
        - How to configure.
        - Different taxonomies. Is it possible?
    - Creating our own taxonomy.
    - Vernacular names and Homonyms
    - National species checklist.
    - GBIF Taxonomy Backbone.
- Security in GBIF Spain data portal.
- Internationalization:
    - About the ALA Portal User Interface.
    - Step-by-step guide UI translation.
    - Language configuration.
    - Muti-languages capability.
- Future steps.

CURRENT SITUATION

- Home page: http://datos.gbif.es
- Information that's open in the Data Portal of GBIF Spain:

- REGISTROS: **14.375.846**
- JUEGOS DE DATOS: **220**
- INSTITUCIONES: **86**

- Hosting by Institute of Physics of Cantabria (Super-computational centre)
  - Infrastructure
- Helping users to use the data portal – youtube guide
- Generic-hub
  - Implementations:
    - New layer on the map: Protected natural spaces.
    - Solved i18n problems in the JavaScript files.
    - Solved problem navigation in the graphic of Taxonomy tree.
  - Customizations:
    - New header with selector of languages.
    - New footer.
    - Clearing fields in the list of records (New structure).
    - Clearing facets that we don't need: IBRA, IMCRA,…
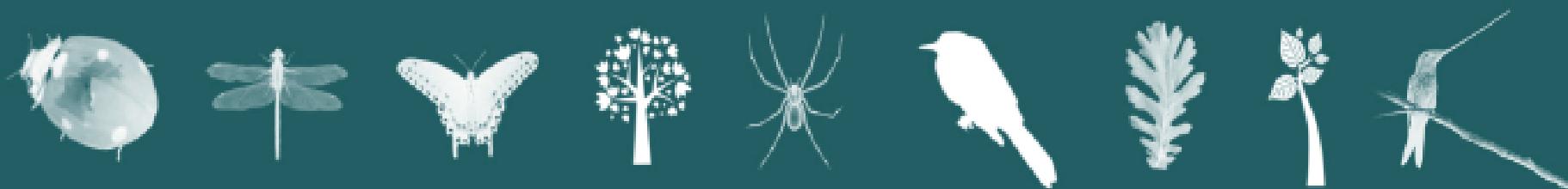
CURRENT SITUATION

- Generic-hub
  - Customizations:
    - New structure of the record page (3 blocks):
    Origin of data, Taxonomy and Geospatial information.
    - Included more fields in the record page and more specific information.

- Collectory-hub (yet to be uploaded)
  - Implementations:
    - Solved problem with institutions/collections map.
    - Customizations:
    - New header with selector of languages.
    - New footer.
    - Adapting fields of MySQL database to data from Spanish institutions/collections/data resources.

- Biocache
  - Changed vocabs: basis of record (HUMAN_OBSERVATION) and countries (e.g.: Republic of Venezuela)
 - Indexing with our owns taxonomies...

International ALA Workshop
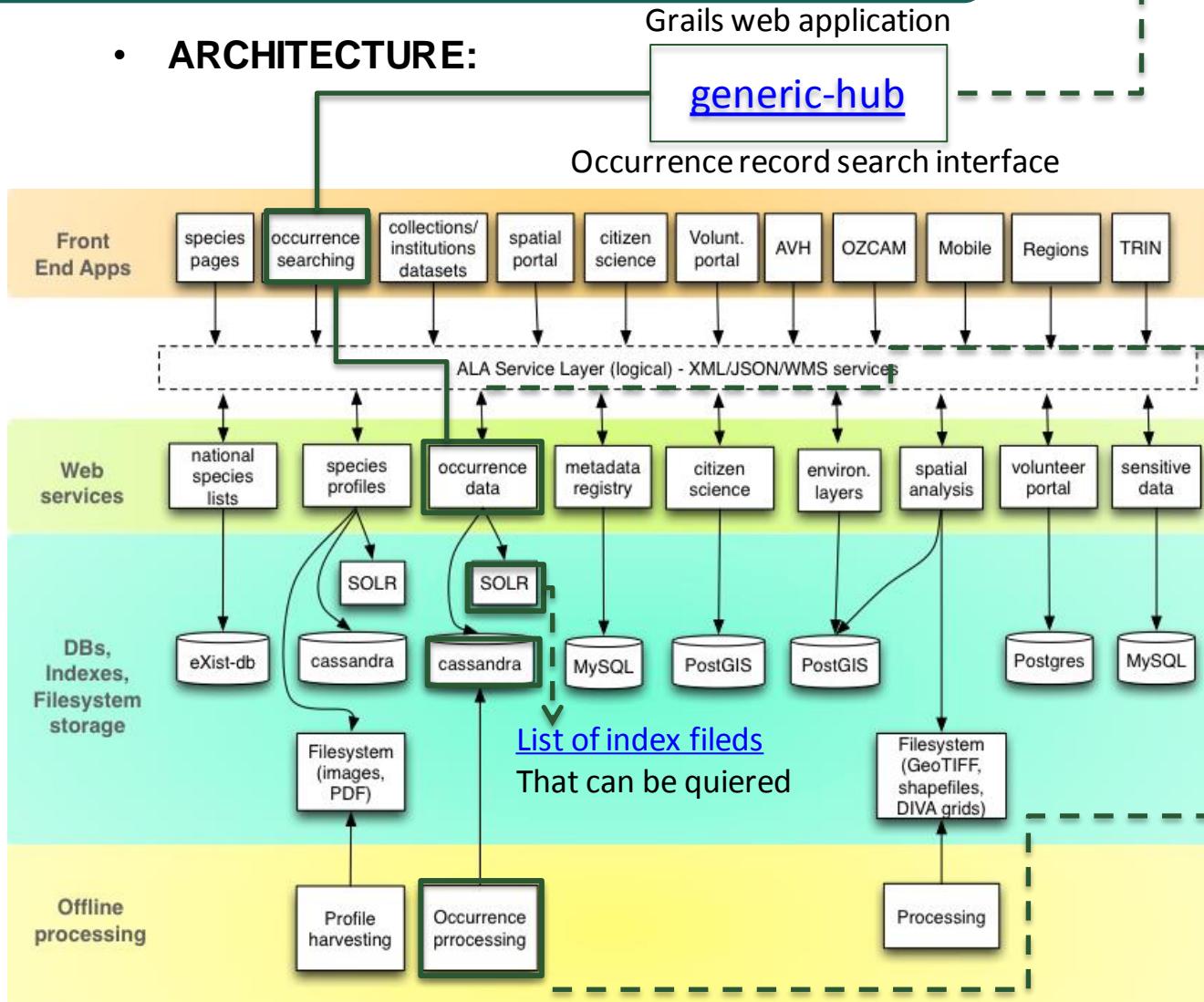
Biodiversity platform of GBIF Spain based on ALA

- BIOCACHE-HUBS -

BIOCACHE-HUBS

- **ARCHITECTURE:**

Grails plugin – core functionalities

biocache-hubs

Grails web application

generic-hub

Occurrence record search interface

Occurrence search portal front-end

provides

Web UI for a back-end service

biocache-service
(old API | new API)

Occurrence and mapping serv
that provides us
a full-text search and record
retrieval for occurrence data
records, using JSON data
format.

List of index fileds
That can be quiered

Scala implementation of biocache
backend

biocache-store

This code base manages the
loading, sampling, processing and
indexing of occurrence records

BIOCACHE-HUB

- **CONFIGURATION FILES:**

    - Where is located the config file of generic-hub?
        - Depends on the configuration that we do in Grails Framework:
            - grails-app/conf/config.groovy:
            - grails.config.locations – Externalized configuration
            - default_config = "/data/${appName}/config/${appName}-config.properties"

            /data/generic-hub/config/generic-hub-config.properties

    - Changing config directory:

        We must set the new directory in the config.groovy file and also we have to change the "data_dir" variable of our ansible inventory (ala_install/inventories/our_inventory)

    - List of properties - https://goo.gl/ZSk0XU

BIOCACHE-HUB

- How can we read the value of the properties that the config file has ?

  Ultimately all configuration files get merged into the config property of the GrailsApplication object and are hence obtainable from there.

  <g:set var="orgNameShort" value="**${**grailsApplication.config.skin.orgNameShort**}**"**/>**

  Property

- Other configuration files in grails-app/conf directory:

  - Config.groovy
  - BuildConfig.grooy
  - ApplicationResource.groovy
  - UrlMappings.groovy
  - BootStrap.groovy
  - DataSource.groovy

- appplication.properties

BIOCACHE-HUB

- **CUSTOMIZATIONS:**

  - Repository: https://github.com/santiagomtzdelariva/generic-hub-es

1. New header and footer:

| Portal de Datos | Inicio | Instituciones, colecciones y proyectos | Juegos de datos | Datos georeferenciados | Buscar | www.gbif.es ▾ | Idioma ▾ | Ayuda |

Català
English
Español

Búsqueda avanzada

Registros

Búsqueda rápida

What we have to do if we want to change the header and footer(github wiki):

- Create/Modify our own layout: gbifes.gsp. (copy/paste generic.gsp)
- Define in the gbifes.gsp the list of module that are required for the new layout: gbifes
- Add js, css and images files that we need for the this new module in ApplicationResource.groovy file.
- Define in the Config.groovy the value of skin.layout.

Practical session: include a new basic header/footer of our organization with images and new styles.

BIOCACHE-HUB

- **CUSTOMIZATIONS:**

  2. I18n: (?lang=es/en)

    - message.properties files located in /grails-app/i18n/
    - How to read the properties that are included in the message.properties:
      - .gsp files: (gbifes.gsp)

`<g:message code="footer.menu.02.01" default="Institutions, collections and projects"`**`/>`**

      - .js files: (charts2.js) - jQuery.i18n.prop('charts2.js.before')
      - .groovy files: (OccurrenceTagLib.groovy) - ${message(code: 'occurrence.raw_date')}
      - How to manage the message.properties files throw IntelliJ
    - How can we detect the language that's enabled? **${**lang == 'es'**}**
    - How can we include the lang_selector plugin?
      1. Modify BuildConfig.groovy to include runtime ":lang-selector:0.3"
      2. Modify your header to include the selector: <langs:selector langs="ca"/>

  Practical session: Modify the plugin: Display the name of the language.

BIOCACHE-HUB

- **CUSTOMIZATIONS:**

    3. Advance_search: (advanced.gsp)

        - Included customize list provinces of Spain and Portugal.
        - Deleted IBRA, IMCRA and lga filters:
            ```
            <select class="biogeographic_region" name="ibra" id="ibra">
            <select class="biogeographic_region" name="imcra" id="imcra">
            <select class="lga" name="lga" id="lga">
            ```

        Practical session: Delete the filters that you don't need.

    4. recordsView (grails-app/taglib/au/org/ala/biocache/hubs/OcurrenceTagLib.groovy)
        - http://datos.gbif.es/biocache-service/occurrences/68b1b58a-a35b-45d5-8ea3-6d9d04c4d9a5
        - def JSONObject occurrence = attrs.occurrence / def record = attrs.record

        Practical session: Create your own way to structure the list of records.

BIOCACHE-HUB

- **CUSTOMIZATIONS:**

   5. Added new WMS layer:

        - Layer capabilities:
          http://wms.magrama.es/sig/Biodiversidad/ENP/wms.aspx?request=getcapabilities

        - _map.gsp: Define a new overlay in a MAP_VAR.

        We have to define the next properties: layers, format, transparent, version, attribution. Alll this properties are based on the capabilities of the WMS layer.

        Practical session: Include a new layer. (* Config.groovy)

        Note: Cartographic server (WMS)

BIOCACHE-HUB

- **CUSTOMIZATIONS:**
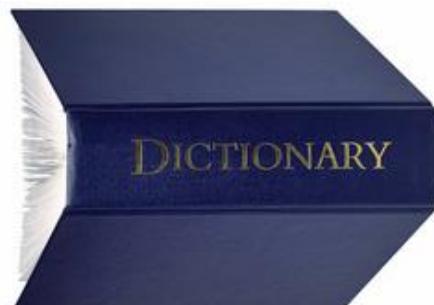
 6. I18n of vocabs: Adding new properties in message.properties.

     - type_status and basis_of_record: (_advance.gsp)
     - speciesGroup (yourAreaMap.js)

     Practical session: What is the best way to include this i18n?
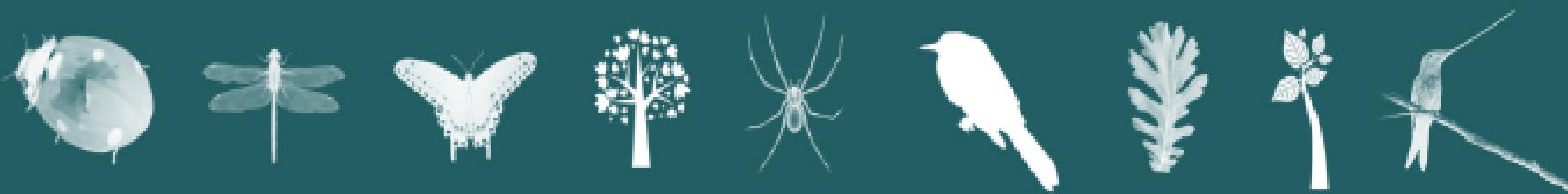
     Note: Compare ALA bor with GBIF.org bor.

     BIOCACHE-STORE. How can we define new terms in our vocabs??

International ALA Workshop

## Biodiversity platform of GBIF Spain based on ALA
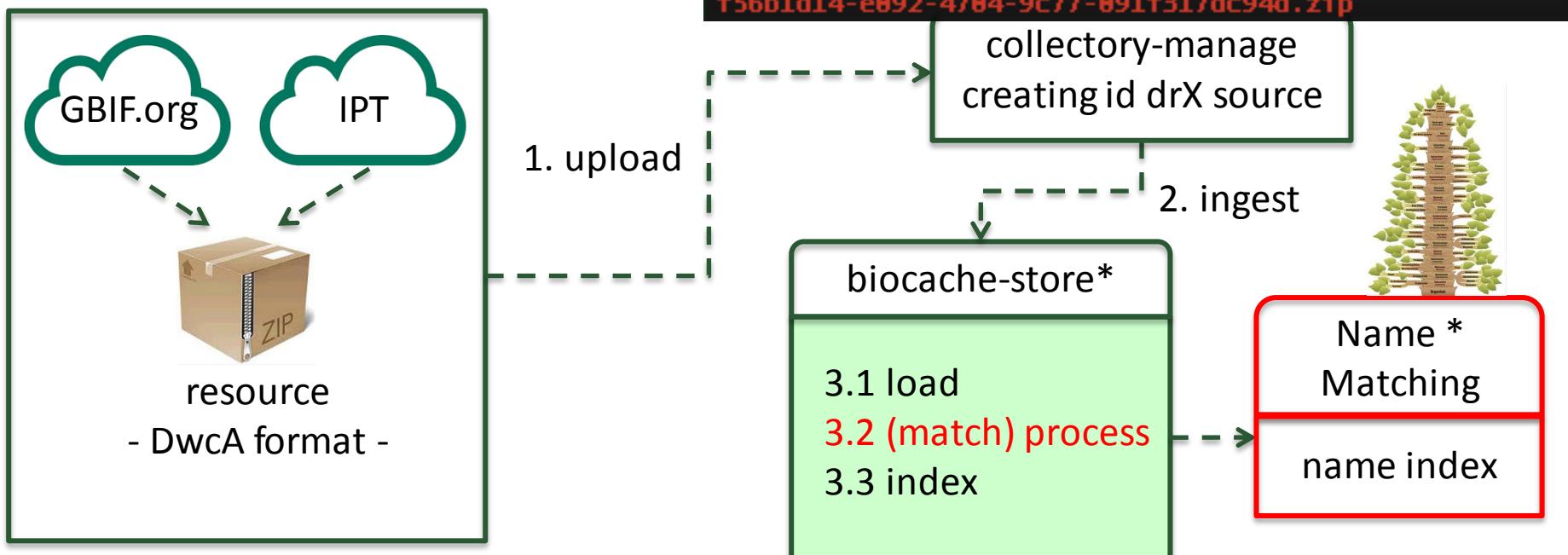
- TAXONOMY -

TAXONOMY

- **WORKING WITH TAXONOMIES:**

  - **Taxonomy** is the science of defining groups of biological organism on the basis of shared characteristics and giving names to those groups. Organisms are grouped together into taxa and these groups are given a taxonomic rank.

  - Understanding the process:


```
Lsama@gbif-websrv:/data/collectory/upload/14743589021421
f56b1d14-e092-4704-9c77-891f317dc94d.zip
```

collectory-manage
creating id drX source

GBIF.org   IPT

1. upload

2. ingest

resource
- DwcA format -

biocache-store*

3.1 load
3.2 (match) process
3.3 index

Name *
Matching

name index

TAXONOMY

- **NAME MATCHING:** is the API in use by the data portal to match scientific name to taxon concepts

    We have two phases in the match process:

    1. Finding the potential match candidate
        a. Attempt to match the exact the scientific name that is supplied.
        b. Parse the supplied scientific name.
    2. Validating the match:
        a. Check if the results are "excluded"
        b. Check if the result represent a "Split species"
        c. Check if the result is a "misapplied name"

    - When the name matching is created the names is stored in several formats:
        a. Raw scientific name including authorship.
        b. Canonical scientific name, without authorship
        c. In "taxa match" form that will allow fuzzy matches.
        d. In component to allow for phrases name patching.

TAXONOMY

- **NAME MATCHING:**

   - **Built a DwCA(s)** that contains all the scientific names that we want to add**:**
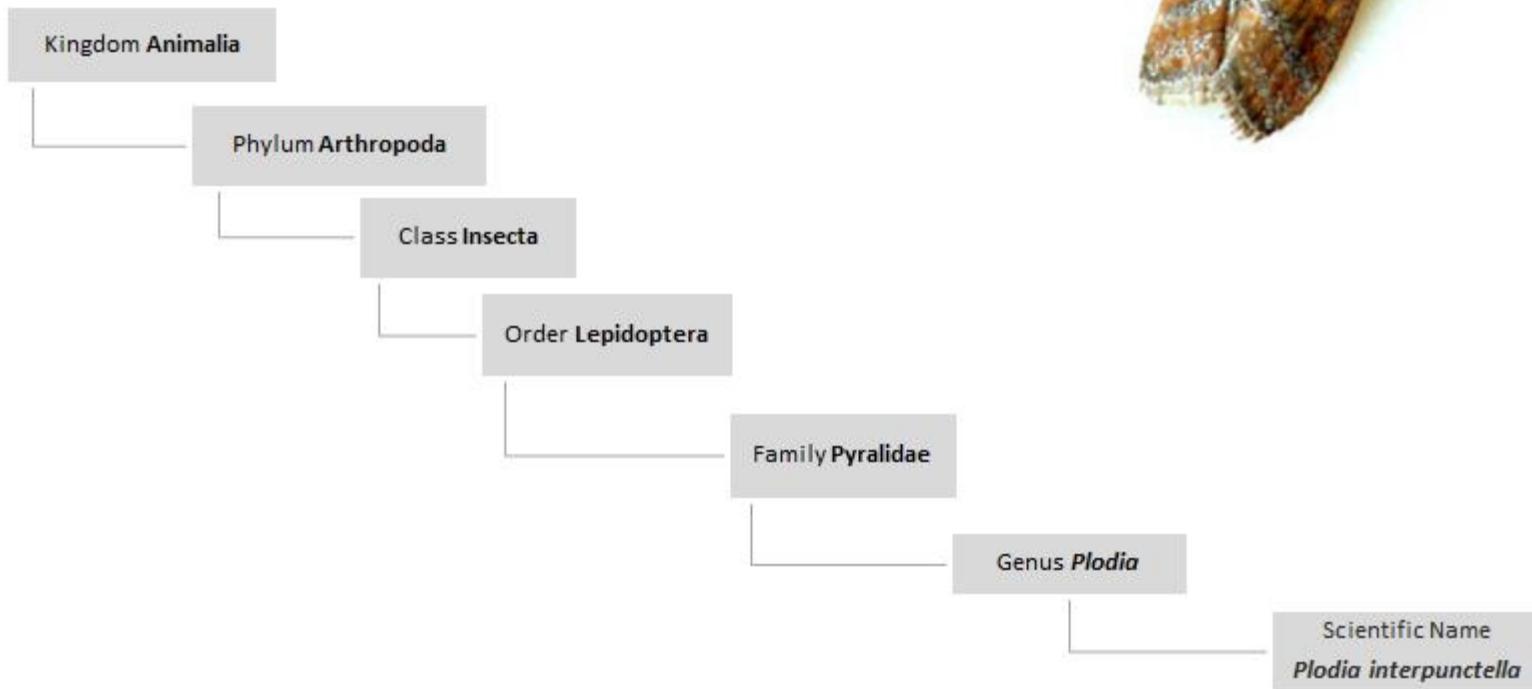
- eml.xml	- Spain_species_taxa_2016.txt
- meta.xml	- vernacular.txt

    Notes:
    - The taxon DwCA must have a core row type of
      http://rs.tdwg.org/dwc/terms/Taxon
    - The vernacular name extension must have a row type of
      http://rs.gbif.org/terms/1.0/VernacularName

      - **Relevant fields** of Taxon core and Vernacular Names extension:
        - Taxon: id, taxonID, parentNameUsage, acceptedNameUsage, taxonomicStatus...
        - Vernacular names: id, taxonID, vernacularName, language.

TAXONOMY

- **NAME MATCHING:**

  - How do you have to be build your occurrence.txt ?.

Kingdom **Animalia**

Phylum **Arthropoda**

Class **Insecta**

Order **Lepidoptera**

Family **Pyralidae**

Genus *Plodia*

Scientific Name
Plodia interpunctella

TAXONOMY

- **CONFIGURATION - NAME MATCHING**:

  - Where do you have to specify the location of your
  name-index directory? /data/biocache/config/biocache-config.properties

  You have to set the value of the next property:

```
# Lucene indexes for name matching
name.index.dir=/data/lucene/namematching13
#name.index.dir=/data/lucene/col_namematching
#name.index.dir=/data/lucene/namematching_plantae
#name.index.dir=/data/lucene/namematching_animalia_plantae
#name.index.dir=/data/lucene/namematching_animalia
#name.index.dir=/mnt/col_namematching_20160226
```

TAXONOMY

- **NAME MATCHING**:

    - **Searching sources** to create our own Name Index:

Nº Names

1. National Species List (NSL)
2. Catalogue of Life
3. GBIF Backbone Taxonomy (54 sources)

    - 2) **Creating the name index**:
        a. Fork ala-name-matching in our repository or
        b. Clone it in our machine.
        c. Create new module from existing files with IntelliJ
        d. Select ala-name-matching directory
        e. Import module from external model: Maven
        f. Next / Next / Next / Finish
        g. Specify the SDK (Open module settings)
        h. Right click on /src/main/java/search/DwcaNameIndexer - Make module
        i. Run and edit configurations:
            a. VM options: - Xmx4g - Xms4g
            b. Program arguments: -dwca /dwca_folder

TAXONOMY

- **NATIONAL CHECKLIST (SPECIESLIST-WEBAPP):**

This is a grails application for managing species list. It supports:

- Upload of species profile data in CSV format
- UI for viewing and faceting species
- Supports any number of key/value pairs
- JSON services used by
    - BIE - species pages
    - Biocache - indexing occurrences
- Supports single owner & multiple editors of a single list
- Lists can be tagged as:
    - authoritative
    - conservation
- Supports sensitive species lists

Architecture: Grails web application and MySQL database

TAXONOMY

- **NATIONAL CHECKLIST (SPECIESLIST-WEBAPP):**

  - **Installation:** There is an Ansible playbook for this application ala-install/ansible/species-list-standalone.yml
  See example inventory ala-install/ansible/inventories/vagrant/species-list-vagrant

  - **Steps:**
  1. Run the species list playbook to install on vagrant VH or in the local machine

ansible-playbook -i inventories/vagrant/species-list-vagrant species-list-standalone.yml --private-key ~/.vagrant.d/insecure_private_key -u vagrant -s

ansible-playbook -I inventories/vagrant/species-list-vagrant species-list-standalone.yml -u root -s –connection=local

  2. Edit /etc/hosts file:

10.1.1.3   vagrant1 ala.vagrant1.dev lists.vagrant1.ala.org.au vagrant1.ala.org.au demo.vagrant1.ala.org.au

TAXONOMY

- **NATIONAL CHECKLIST (SPECIESLIST-WEBAPP):**

  - **Steps:**
  3. Check the service: http://lists.vagrant1.ala.org.au
  4. Logged in with ALA account.
  5. Download checklist or built it. e.g.: http://www.gbif.org/dataset/888b4cc1-7342-4d6c-ba5c-c1b29c8d27fd
  6. Unzip the archive
  7. Upload the DarwinCoreA_Pli.txt from the DwC-A using the form
  8. Fill in fields (Title, Description, URL and type (List of species with characters)
  9. We reprocessed the file to do the following:
     - remove taxon remarks
     - make a UTF-8 tab delimited file
  10. Edit the config file specieslist-webapp-config.properties

  bie.nameIndexLocation= /data/lucene/col_namematching

  11. ran out of time. Should have matched and provided and example like:

  http://lists.ala.org.au/ws/species/urn:lsid:biodiversity.org.au:afd.taxon:5b72fb29-0318-43f8-bc0a-58b879b17601

TAXONOMY

- **NATIONAL CHECKLIST (SPECIESLIST-WEBAPP):**

  - **Example of how to link:**

Webpage for an occurrence page (has no species IDs)http://datos.gbif.es/generic-hub/occurrences/c4abc981-6cf7-413a-9085-c8195ccccf4f

Has an example API call which has the taxa IDs: http://datos.gbif.es/biocache-service/occurrences/c4abc981-6cf7-413a-9085-c8195ccccf4f

TAXONOMY

- **QUESTIONS**:

  - Can we work with different names index or taxonomies? (by Jörg)

  Yes, but It's no recommended. We have to have one name index for BIE-index and Biocache. In this way we can relate the names that we have indexed with biocache with the related species pages that are indexed with BIE-index, because they will have the same TaxonID.
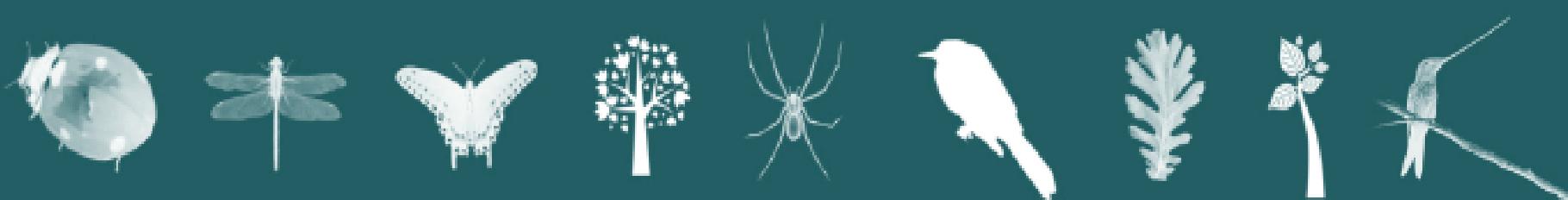
  - Is it possible use empty taxonomies? (by Jörg)

  I don't understand this question.

International ALA Workshop

Biodiversity platform of GBIF Spain based on ALA

- SECURITY -

SECURITY

- **CASSANDRA:**

  - Hosted in a virtual machine with Private IP, 14 GB RAM and 190 GB HD.

  - Access to Cassandra using "Cassandra-cli" directly on VM.

  - The occurrence data is stored in: "/data/cassandra/data/occ".

  - Connection to Cassandra throw: $ cassandra-cli

- **SOLR:**

  - Hosted in a virtual machine with Public IP, 32 GB RAM and 50 GB HD.
  - Access to SOLR interface throw SSH.

SECURITY

- **ADMINSTRATION PAGES:**

  - Filter access by IP - Modify Apache2 configuration file:

```
<Location /generic-hub/admin>
        Order Deny,Allow
        Deny from all
        Allow from
</Location>


<Location /collectory/manage>
        Order Deny,Allow
        Deny from all
        Allow from
</Location>

<Location /collectory/collection>
        Order Deny,Allow
        Deny from all
        Allow from
</Location>

<Location /collectory/dataProvider>
        Order Deny,Allow
        Deny from all
        Allow from
</Location>

<Location /collectory/institution>
        Order Deny,Allow
        Deny from all
        Allow from
</Location>
```

International ALA Workshop

Biodiversity platform of GBIF Spain based on ALA

- INTERNATIONALIZATION -

- **INTERNATIONALIZATION:**

  - Where are the message.properties located in the modules?
  - How can you manage the message.properties?
    - How can add/delete/modify new properties.
    - Collaborative work using Crowdin platform
      - Project name: ALA Internationalization
        1. Sign up or sign in with one of possible social networks.
        2. Go to ALA internationalization project.
        3. Select the language that you want to translate. E.g.: German
        4. Select the message.properties that you want to translate
        5. Translate all messages
        6. When you finish you have to ask to one manager that build the project and send you the zip with with the message.properties that you translated

FUTURE STEPS

- Solve problems with collectory module.

- Implement BIE and Species modules in the production environment.

- Implement logger-service.

- Upgrade biocache-service

- Upgrade biocache-store.

- Upgrade taxonomy.

- Include data resource with images.

- Test Spatial Portal module.

Santiago Martínez de la Riva Vivanco
Unidad de Coordinación GBIF España
sama@gbif.es
www.gbif.es
Real Jardín Botánico 03/09/16