UNIVERSIDADE
DE AVEIRO

COEUS: "Semantic Web in a Box"

http://bioinformatics.ua.pt/coeus

In this tutorial you will learn:

- – How to Generate Nanopublications from a GeneRIF dataset.

Outline

1 Integrate GeneRIF data

MAIN GOAL: Integrate **GeneRIF** dataset and make the linkage to **Pubmed** to retrieve extra information.

1.1 Requirements

1. Application Server

    –   Java environment with IDE, Maven, Tomcat (for developer release)

        •   https://github.com/bioinformatics-ua/COEUS/

2. Database

    –   MySQL

3. GeneRIF data: http://bioinformatics.ua.pt/coeus/generif_sample.csv

1.2 Configuration

There isn't any actual implementation to setup the backend for new COEUS seeds. Everything can be configured and managed from the seed setup web interface. This is where the fun starts: launch COEUS and start configuring your seed.
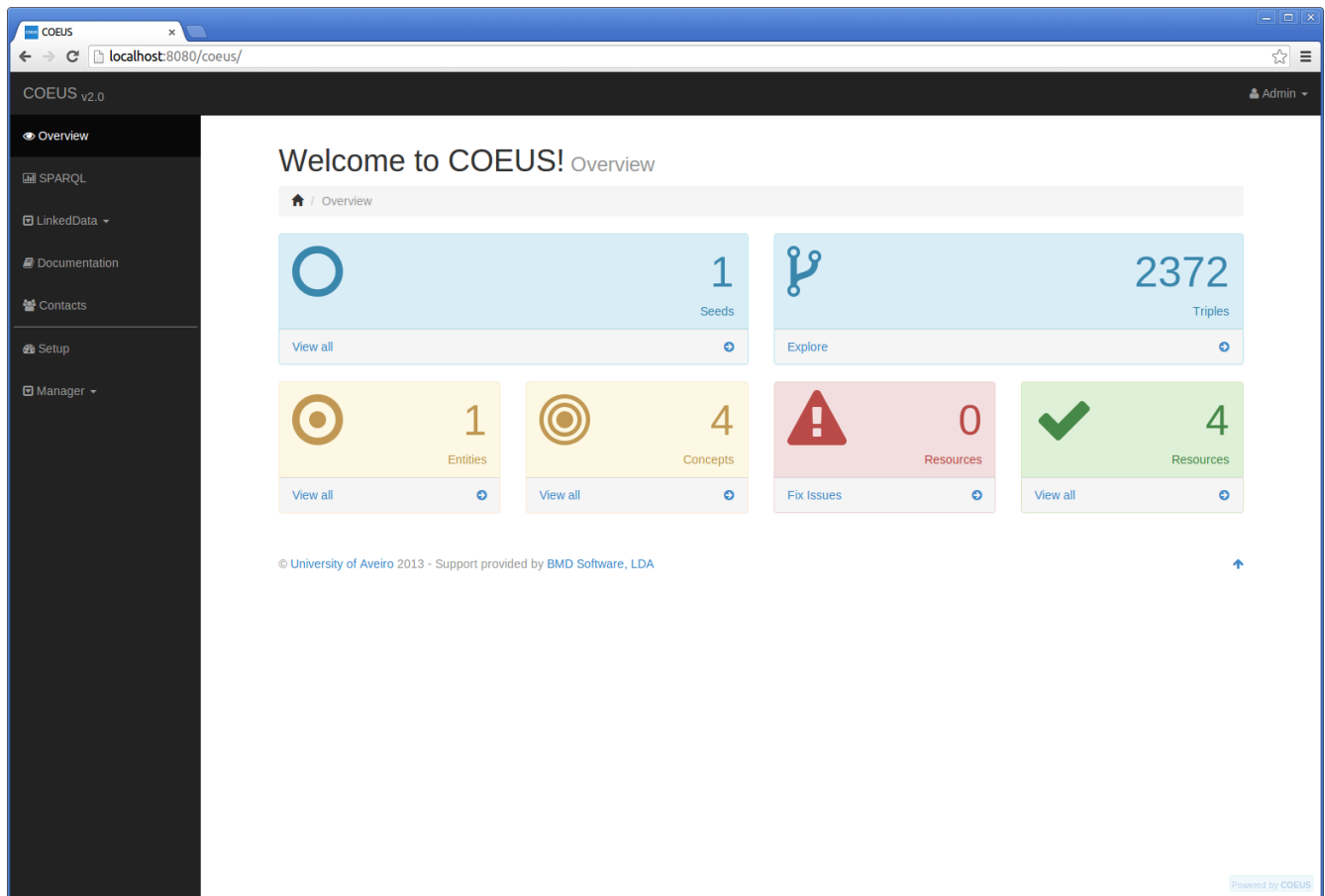
1.2.1 Setting up the environment

Notes

•   COEUS seeds are organised in environments

•   Environments define server properties

    –   MySQL connection

    –   Public/private LinkedData configuration

    –   Model

- Prefix list

- ...

- Usage

    - Configure multiple setups in a single machine

    - Setup "development", "test" and "production" environments

1.2.2   Launch COEUS web app

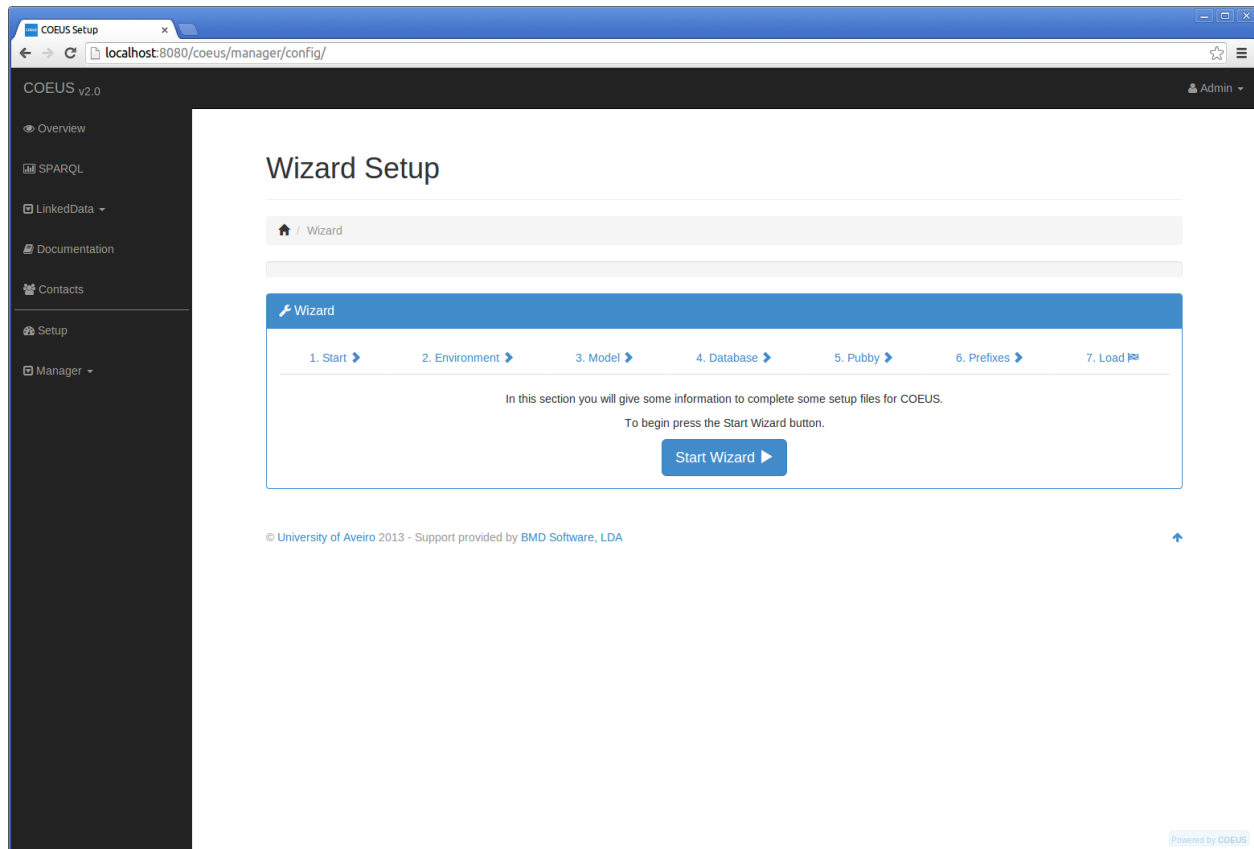- Go to /coeus on your browser of choice

1.2.3 Go to setup page

- Click the Setup link in the left menu bar

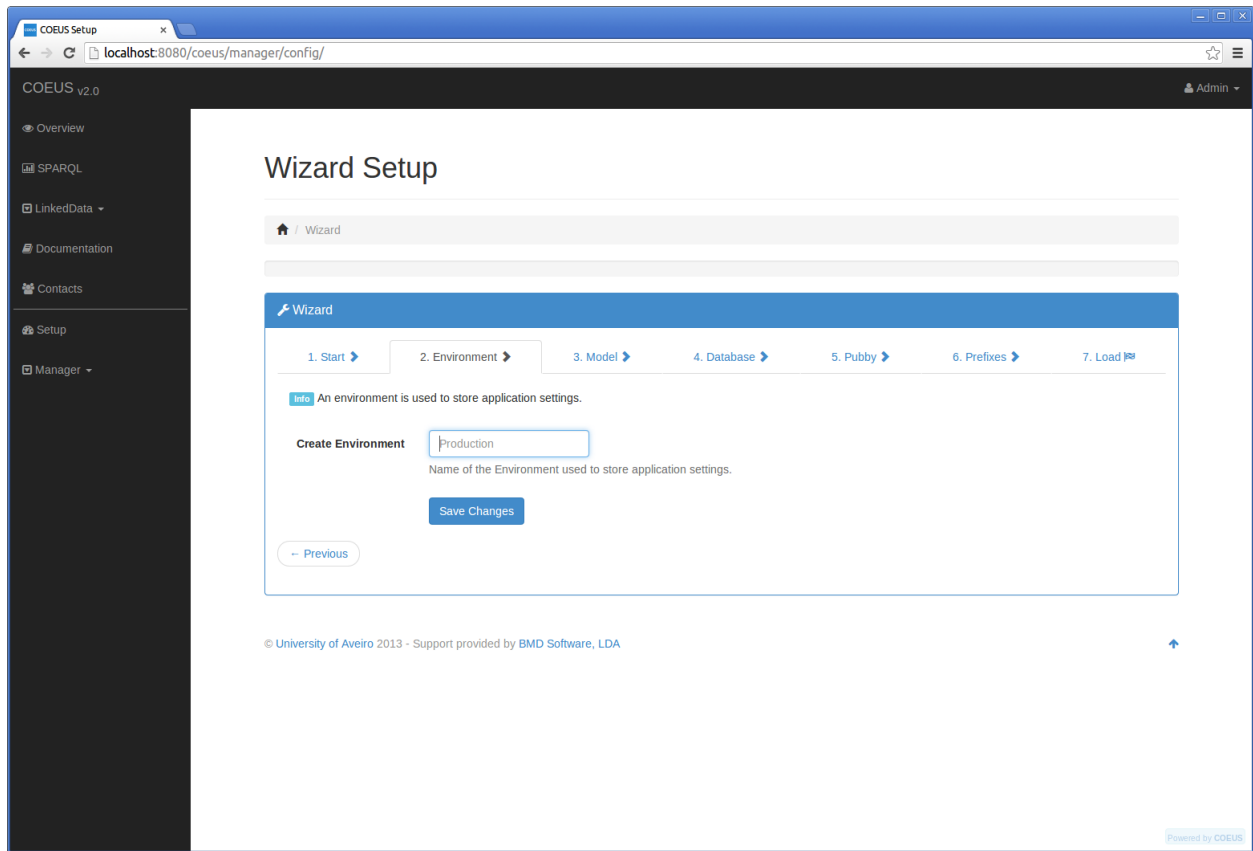 – If authentication is enabled use your tomcat credentials.

If it is the first time running COEUS' setup, the *Configuration Wizard* will be displayed automatically.

1. Press the **Start Wizard** button to begin.

1.2.4   Environment Configuration

Set the environment name. For instance, "tutorial".

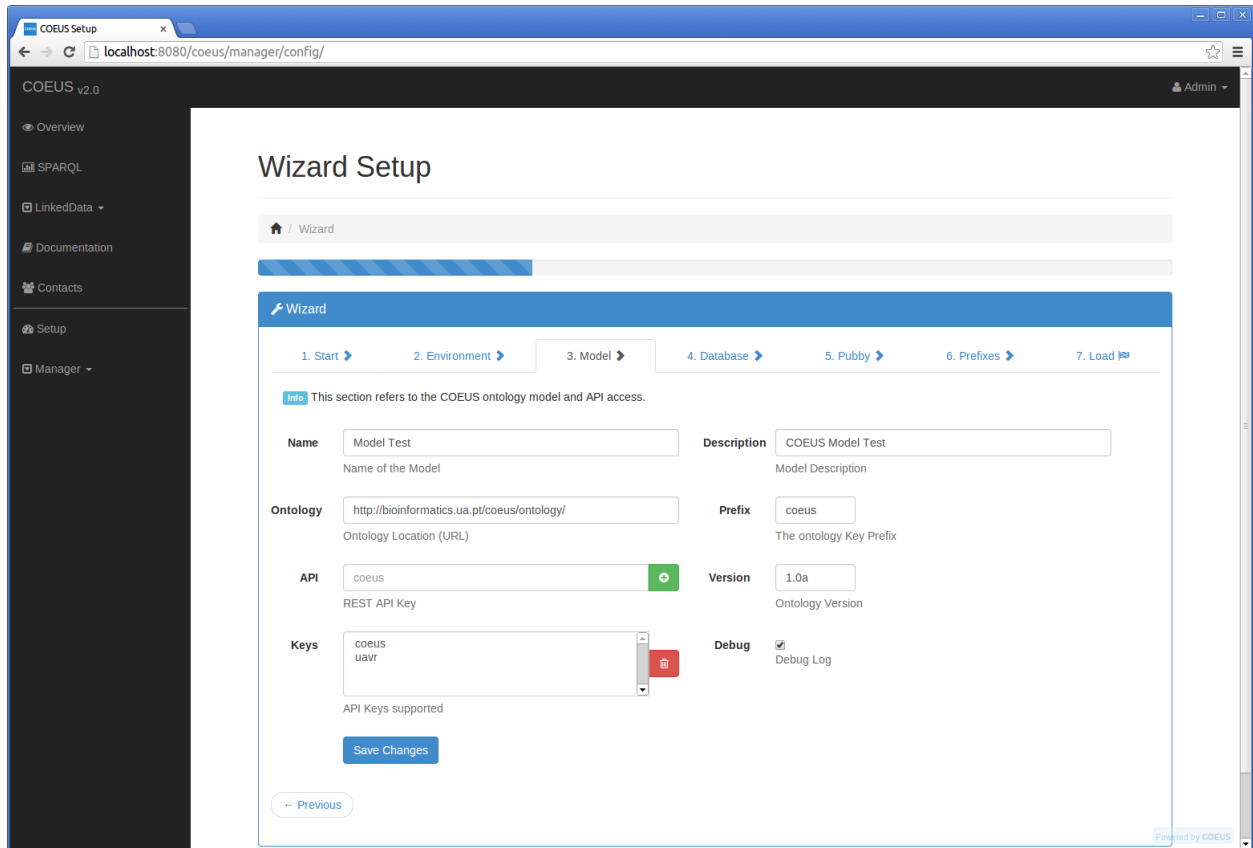Save the changes and press *Next*.

1.2.5   Model Setup

Each COEUS environment has an internal model. Besides the integration structure, the model has several properties:

- *Name*: The model name

- *Ontology*: External ontology URL, if used

- *Description*: The model description

- *Key Prefix*: The base semantic prefix to store the data

- *API Key*: List of API keys for usage in client applications (delimited by |)

- *Version*: Model version

- *Debug*: Debug mode generates a verbose server output

No changes are required for the *GeneRIF* example. Feel free to update it according to your setup (change the name or the API keys).

Save the changes and press *Next*.

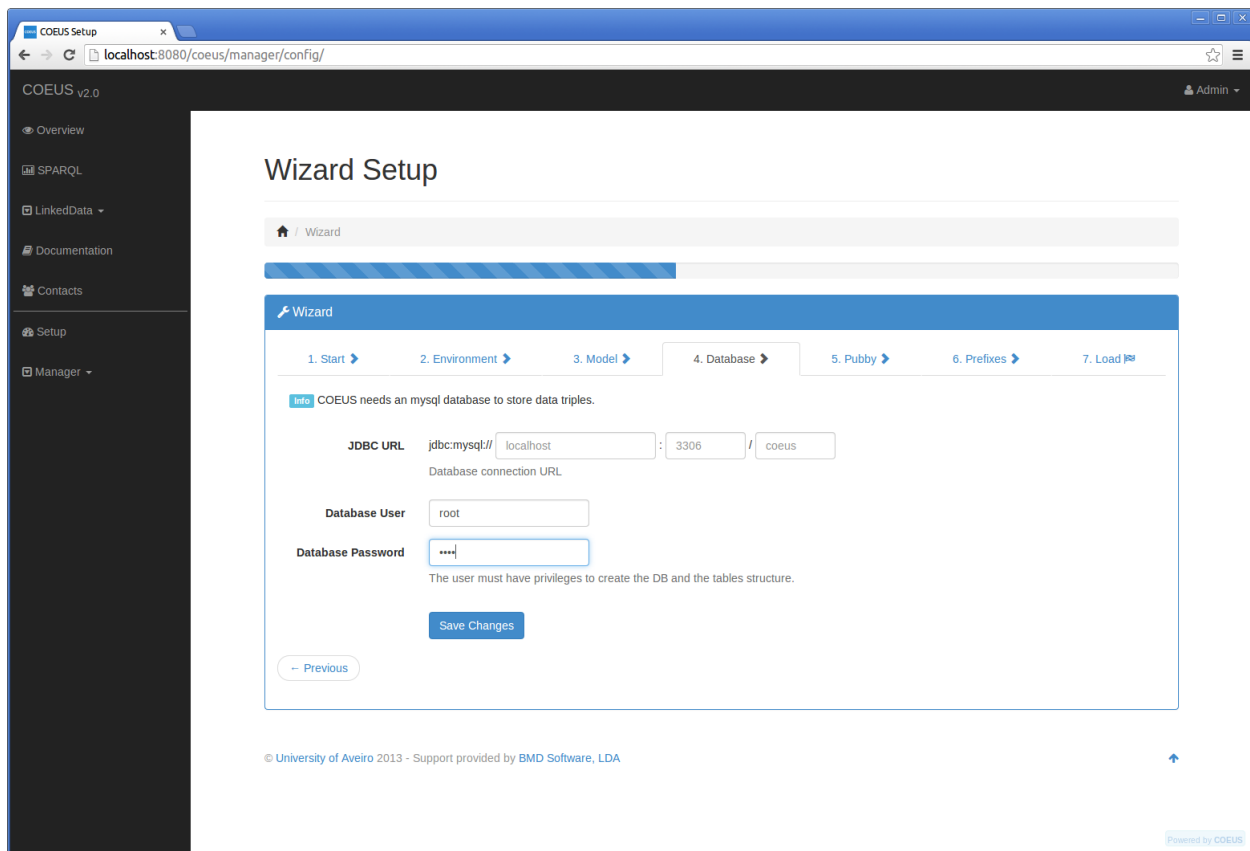## 1.2.6   Database Setup

Notes

- COEUS requires a MySQL database

  - with a user with root privileges or other similar to create the DB and tables structure.

- Used to

  - Support Jena triplestore

- Store temporary configuration properties

- Extend client-side applications

Configure the connections properties of the environment database. If you don't have MySQL installed, you can use the following details:

- *Host*: localhost

- *Port*: 3306

- *Database*: tutorial_xyz (change *xyz* to any other word...)

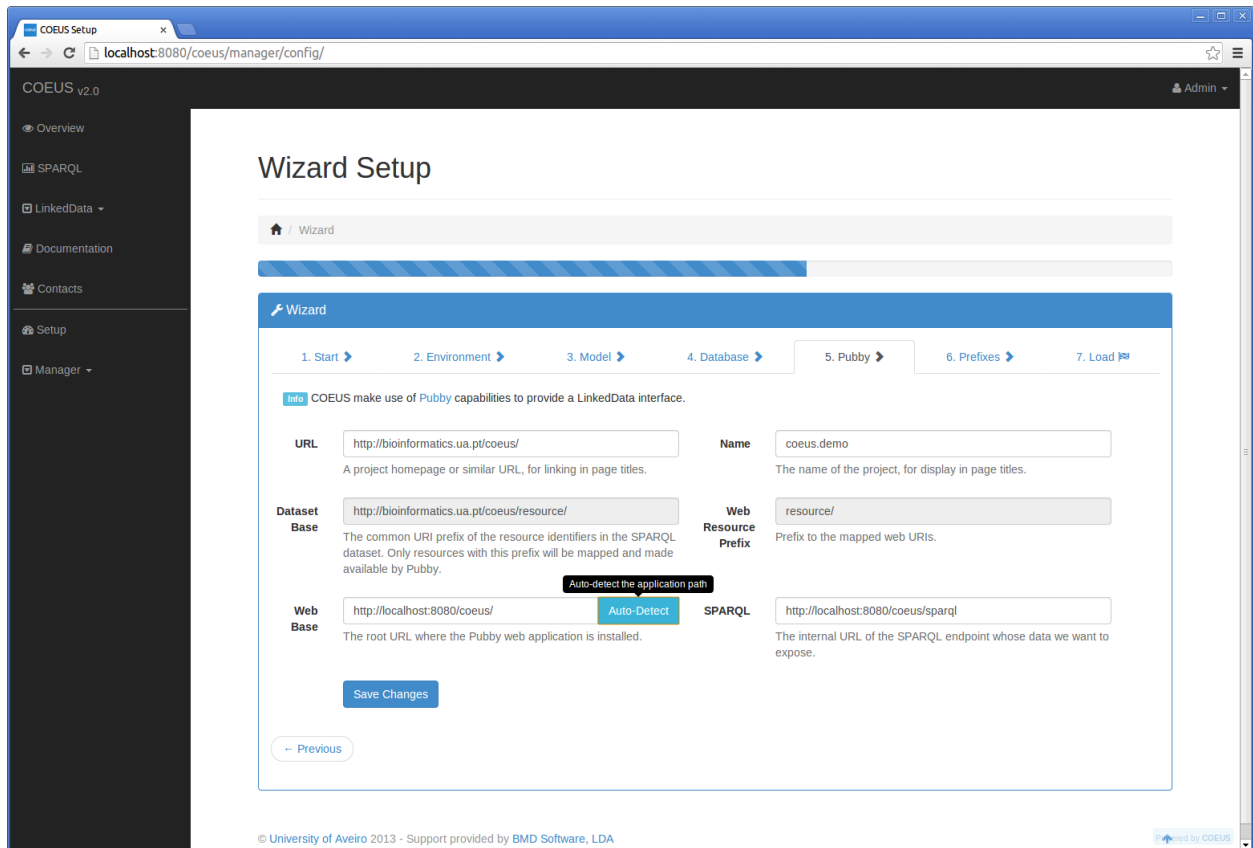- *User*: tutorial

- *Password*: coeus

Save the changes and press *Next*.

1.2.7   Pubby Setup

Pubby manages the LinkedData publishing interfaces. On the Pubby configuration screen you can define seed-specific properties to be used by the Pubby engine, such as the project homepage, the SPARQL endpoint address or the base knowledge base URI.

For now, we can leave the Pubby setup with the default values by only clicking on the auto-detect button that ensures that your app has the correct location.
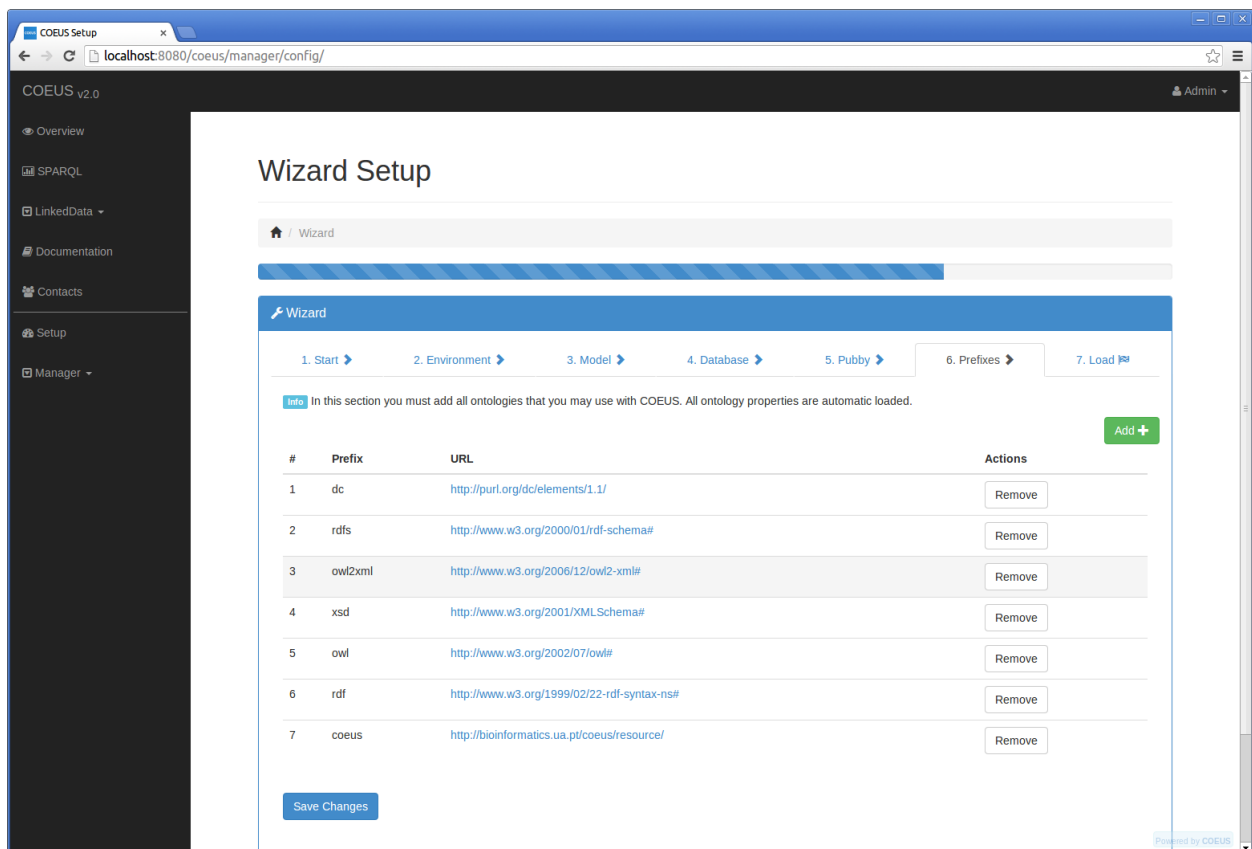

Save the changes and press *Next*.

1.2.8   Prefix Setup

Notes

For improved performance, COEUS includes an internal semantic name resolution engine. This tools translates long-form URIs into short-form strings and vice-versa. For instance, http://www.w3.org/1999/02/22-rdf-syntax-ns#label to rdf:label. These prefixes are prepoluted on the COEUS engine. Hence, in this step, you have to include any external ontology that will be used in the COEUS environment.

6.   If not available, add the following schema to the prefix list.

• *Prefix*: dc

• *URI*: http://purl.org/dc/elements/1.1/

Save the changes and press *Next*.

1.2.9　　Finish Environment Setup

If all went well, pressing the *Load* button will load the new settings to start using COEUS.

Press the *Load* button and wait a few seconds. After success response you must click on "Start with COEUS" button.

1.3 Create Ontology Model for GeneRIF data

1.3.1　　Configuring your first seed

On the COEUS homepage, press the left *Setup* link.

- In this screen you can manage all the COEUS seeds under a specific environment.
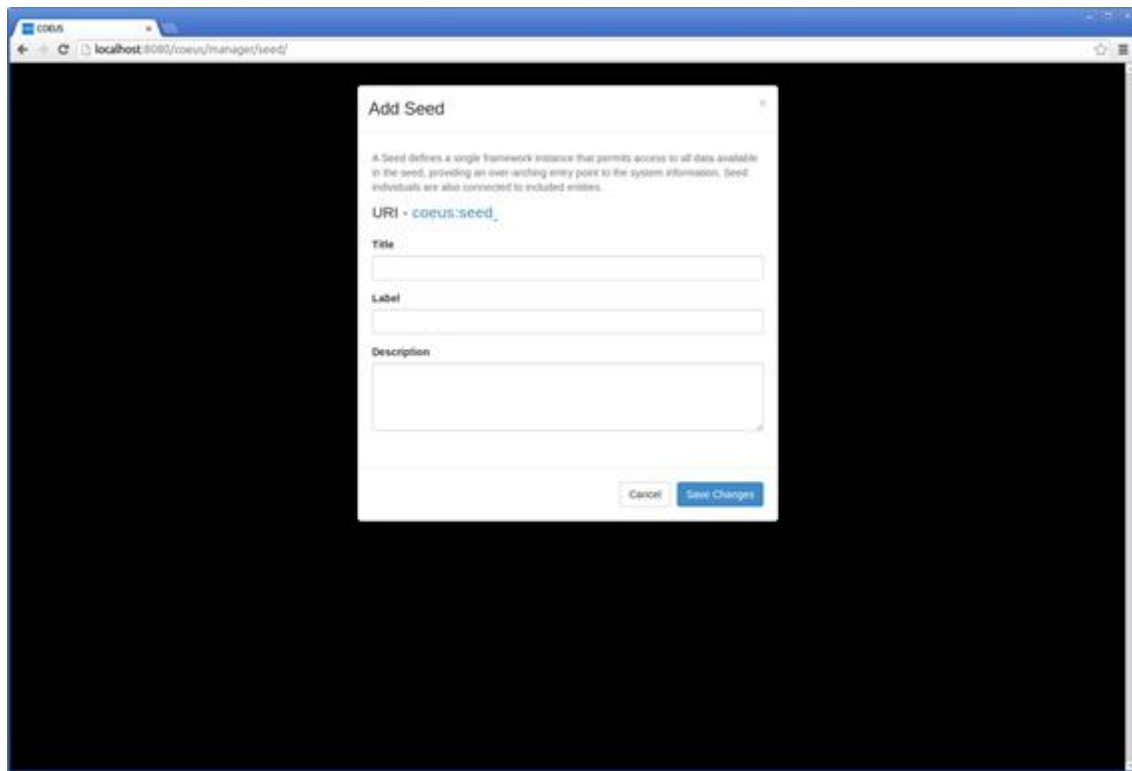
Press the *Add Seed* button. On the seed configuration form, set the Seed title (e.g. Nanopub), label and a description. The new seed will appear on the seed-listing interface.

To start configuring the internal data structure, press the *Choose* button on the new seed. COEUS will display the Seed dashboard.

1.3.2   Dashboard

The seed dashboard provides a general overview of the entire seed content. The structure tree display the entire hierarchical structure configured in the seed (Entity-Concept-Resource). The action toolbox provides direct access buttons to the most common seed actions. The next image provides an overview of the final ontology model that we intend to build.

1.3.3   Create Ontology for GeneRIF integration

1. Create a "generif" and "pubmed" entity (title). Complete remain fields (Label, Comment) according to your needs.

2. Create also a "generif" and "pubmed" concepts (title) under the respective entities according to the previous image. Complete remain fields (Label, Comment) according to your needs.

3. The next step is the creation of the Resources. Resources store the metadata configuring the ETL being performed for each Concept. One resource has only one concept but one concept can have many resources.

   1) Create a "generif" resource. Complete it:

   a. Title, Label, Comment: generif

   b. Order: 1

   c. Extends: concept_generif

   d. Method: cache

   e. Publisher: csv

   f. Endpoint: http://bioinformatics.ua.pt/coeus/generif_sample.csv

   g. Csv Quotes: "

   h. Cvs Starting line: 1

2) Create the selectors to the "generif" resource (click on the wrench icon):

a. Create a selector to parse the ID column:

      i. Title, Label: id

      ii. Key: yes

      iii. Query: 0

      iv. Property: dc:identifier

b. Create a selector to parse the text column:

      i. Title, Label: text

      ii. Key: no

      iii. Query: 5

      iv. Property: dc:description

c. Create a selector to parse the date column:

      i. Title, Label: date

      ii. Key: no

      iii. Query: 4

      iv. Property: dc:date

d. Create a selector to parse the gene id column:

      i. Title, Label: gene_id

      ii. Key: no

      iii. Query: 2

      iv. Property: dc:relation

e. Create a selector to parse the pubmed id column:

      i. Title, Label: pub_id

      ii. Key: no

      iii. Query: 3

      iv. Property: dc:source

3) Create a "pubmed_article" resource. Complete it:

a. Title, Label, Comment: pubmed

b. Order: 2

c. Extends: concept_generif

d. Method: cache

e. Property Extension: dc:source

f. Publisher: xml

g. Endpoint:

http://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=pubmed&id=#replace

#&rettype=xml&retmode=text

h. Query: //PubmedArticle

4) **Create the selectors to the "pubmed_article" resource (click on the wrench icon):**

    a. Create a selector to parse the ID field:

       i. Title, Label: id

       ii. Key: yes

       iii. Query: //PMID

       iv. Property: dc:identifier

    b. Create a selector to parse the title field:

       i. Title, Label: title

       ii. Key: no

       iii. Query: //ArticleTitle

       iv. Property: dc:title

    c. Create a selector to parse the abstract field:

       i. Title, Label: abstract

       ii. Key: no

       iii. Query: //AbstractText

       iv. Property: dcterms:abstract

    d. Create a selector to parse the journal field:

       i. Title, Label: journal

       ii. Key: no

       iii. Query: //Title

       iv. Property: dc:subject

5) **Create a "pubmed_link" resource. Complete it:**

    a. Title, Label, Comment: pubmed_link

    b. Order: 3

    c. Extends: concept_pubmed

    d. Method: cache

    e. Publisher: linkeddata

    f. Endpoint: http://www.ncbi.nlm.nih.gov/pubmed/#replace#

    g. Query: linkeddata

6) **Create the selectors to the "pubmed_link" resource (click on the wrench icon):**

a. Create a selector to make the connection to pubmed link:
  i. Title, Label: link
  ii. Key: yes
  iii. Query: link
  iv. Property: rdfs:seeAlso

4. Click the "Build" button to start the integration process.

2 Nanopublications Generation Process

2.1 Requirements

- COEUS Knowledge Base successfully built with the respective data concept items.

  - This tutorial contain information about the "GeneRIF" example. This information is used in the figures to demonstrate one example scenario. However, you can apply this tutorial to any example that you want because the steps are the same.

2.2 How to Generate Nanopubs

COEUS has now the capability to share your data in the Nanopublication format. With this new plugin you can transform your integrated data in this prominent format by following the next steps.

2.2.1   Go to Dashboard

- Go to the seed dashboard that you want to generate nanopublications. You must verify at your dashboard seed to see the total of concept data items (e.g. in the next figure the number of total concept items is 19608).
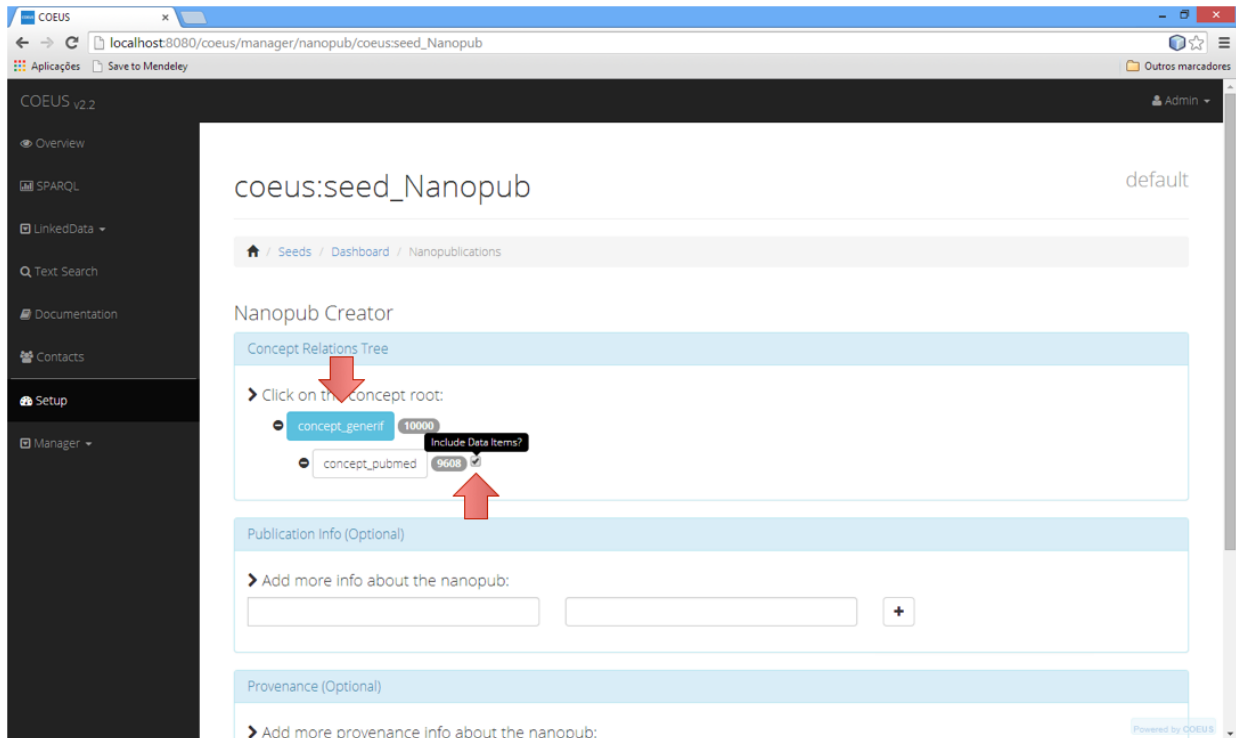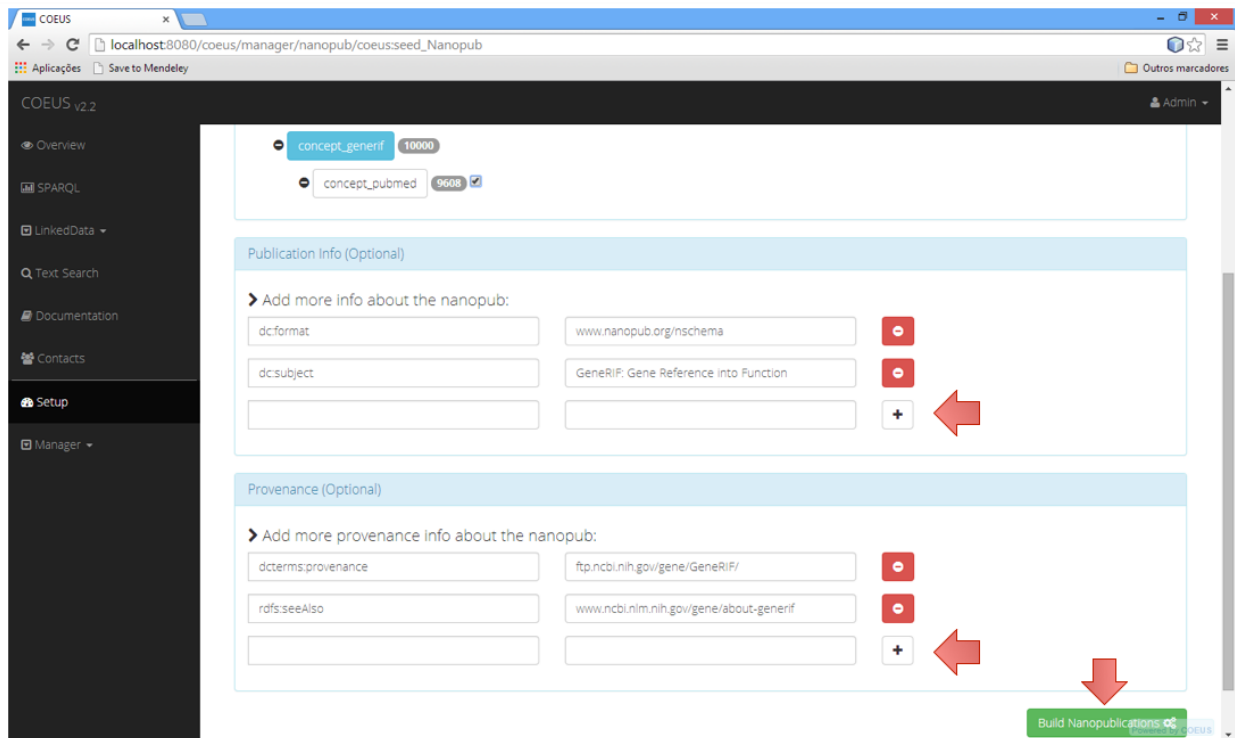- Click on the Nanopublications Button.

2.2.2   Choose the concept root

- Select the root concept that will generate the nanopublications. The number of generated nanopublications number will be equal to the number of concept data items root.
- Optionally you can select others concepts related data and include it in the final nanopublication.

2.2.3   Optional Information and Build Process

- Fill out the additional and provenance information about the nanopublications (Optional).
  - Try for example:
    - Provenace:
      - dcterms:provenance   ftp.ncbi.nih.gov/gene/GeneRIF/
      - ...
    - PublicationInfo:
      - dc:subject   http://www.ncbi.nlm.nih.gov/gene/about-generif
      - ...
- Press the "Build Nanopublications" button to start the process.

2.2.4   Finish

- When finished you can view the concept data items according to the nanopublication format.