

COEUS: “Semantic Web in a Box”

<http://bioinformatics.ua.pt/coeus>

In this tutorial you will learn:

- What is COEUS
- How to create a new COEUS instance
 - From GitHub download to web deployment
- How to integrate data from heterogeneous *omics sources
 - Exploring integration connectors
- How to use COEUS API to access aggregated data
 - Build new rich web information systems using the API
 - Access the SPARQL and LinkedData interfaces

Outline

Outline	2
1 Workshop Setup	3
1.1 Requirements.....	3
1.2 Get ready	3
2 News Reader	4
2.1 Requirements Analysis	5
2.2 Implementation/Configuration	6
2.3 Testing	23
3 Proteomics Knowledge Base	24
3.1 Requirements Analysis	24
3.2 Configuration	25
Testing	32
3.3 LinkedData browsing	32
4 Overview	33
4.1 Model	33
4.2 Integration.....	33
4.3 API.....	34
4.4 Documentation.....	34
5 Feedback	35

1 Workshop Setup

1.1 Requirements

- Application Server
 - Tomcat (.war deployment release, recommended)
 - <https://github.com/bioinformatics-ua/COEUS/releases/download/2.0/coeus.war>
 - Java environment with IDE, Maven, Tomcat (for developer release)
 - <https://github.com/bioinformatics-ua/COEUS/>
- Database
 - MySQL
 - Sample data: <http://bioinformatics.ua.pt/coeus/hummer.sql>

1.2 Get ready

To start this tutorial, you must configure your Tomcat application server authentication first. COEUS uses the Tomcat manager application (installed by default on “/manager”), which needs you to define the [authentication realm](#). This is done by simple adding a line on the [tomcat-users.xml](#) file of your Tomcat server: `<user name="admin" password="admin123" roles="manager-gui, manager-script" />`

To start, just deploy your COEUS instance to the application server. By default, it should be available at <http://localhost:8080/coeus/>.

Once it's online, you're ready to go!

2 News Reader

In this first example, we will build a simple news aggregator. The goal is to collect, merge and query news from multiple websites. This example highlights the workflow for creating new COEUS seeds.

Requirements Analysis

1. Identify data sources for integration
2. Draft internal seed model

Implementation

1. Setup COEUS seed
2. Create internal configuration
3. Build knowledge base

Test

1. Query data
2. Test APIs

Use (not covered in this tutorial)

- Use APIs to build client-side applications

2.1 Requirements Analysis

This News Reader will integrate sports news from four distinct sources. For each news outlet, we need to identify the required RSS feed:

- **Reuters:** <http://feeds.reuters.com/reuters/sportsNews>
- **BBC:** <http://feeds.bbc.co.uk/sport/0/rss.xml>
- **Marca:** <http://estaticos.marca.com/rss/portada.xml>
- **A Bola:** <http://www.abola.pt/rss/index.aspx>

Next, we need to create a semantic model to accommodate our news data. This model is mapped to COEUS' internal model, and defines the hierarchical structure organising our knowledge. For this scenario, we will draft a very simple model, with just one **entity** and one **concept** for each news source. This way, a single news entry matches a single **item** in the knowledge base (thus, belonging to a single **concept**). The final structure is the following:

- **Entity:** News
 - **Concept:** Reuters
 - **Concept:** BBC
 - **Concept:** Marca
 - **Concept:** A Bola

At this stage, we also identify what data we need to save for each news entry. We will not create a custom ontology to store these data, we will reuse existing ontologies/predicates:

- **dc:identifier:** entry id
- **dc:title:** entry title
- **dc:description:** entry text
- **rdfs:seeAlso:** entry link to original news
- **dc:date:** entry publication date

2.2 Implementation/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.

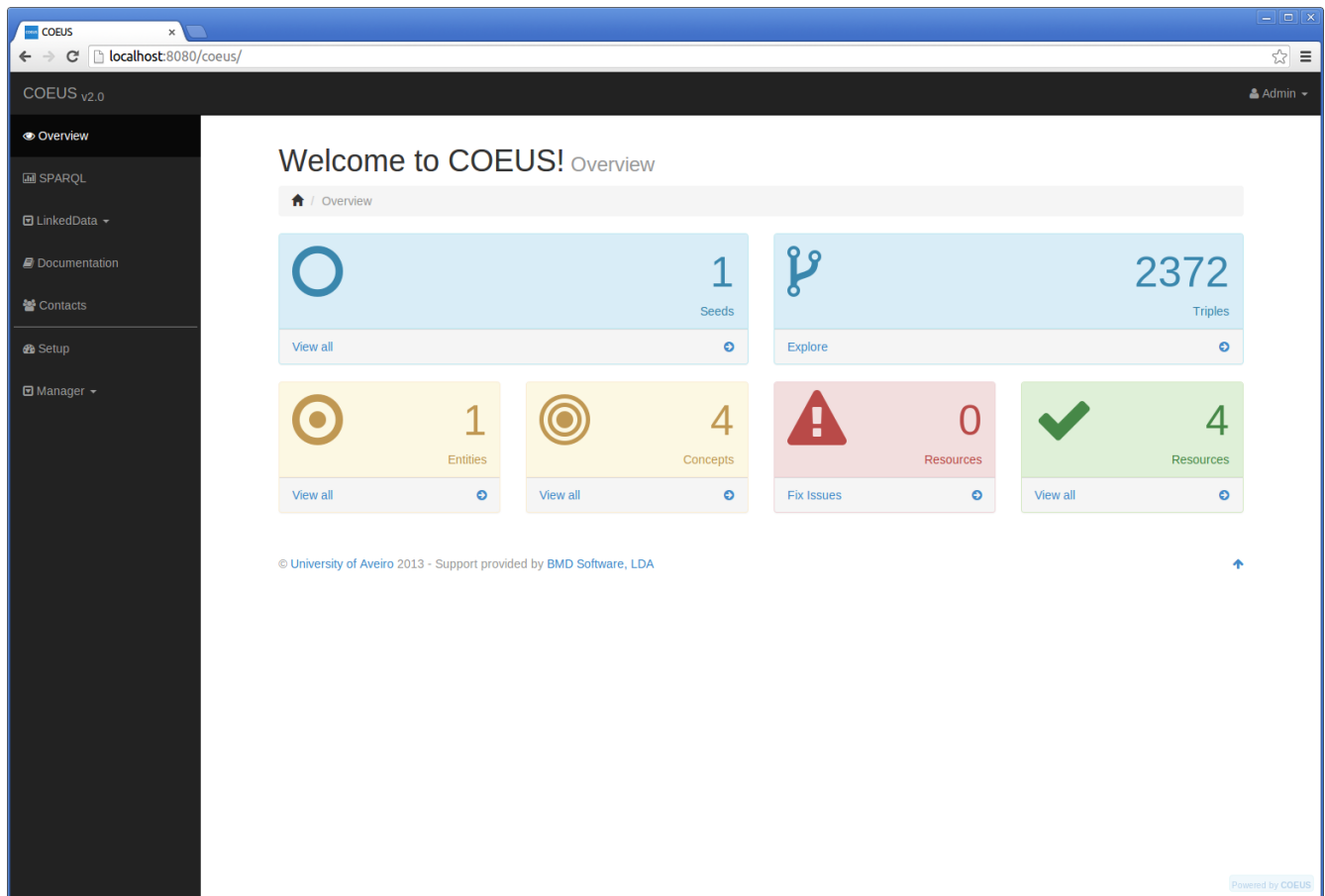
2.2.1 Setting up the environment

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

2.2.1.1 Launch COEUS web app

- Go to /coeus on your browser of choice

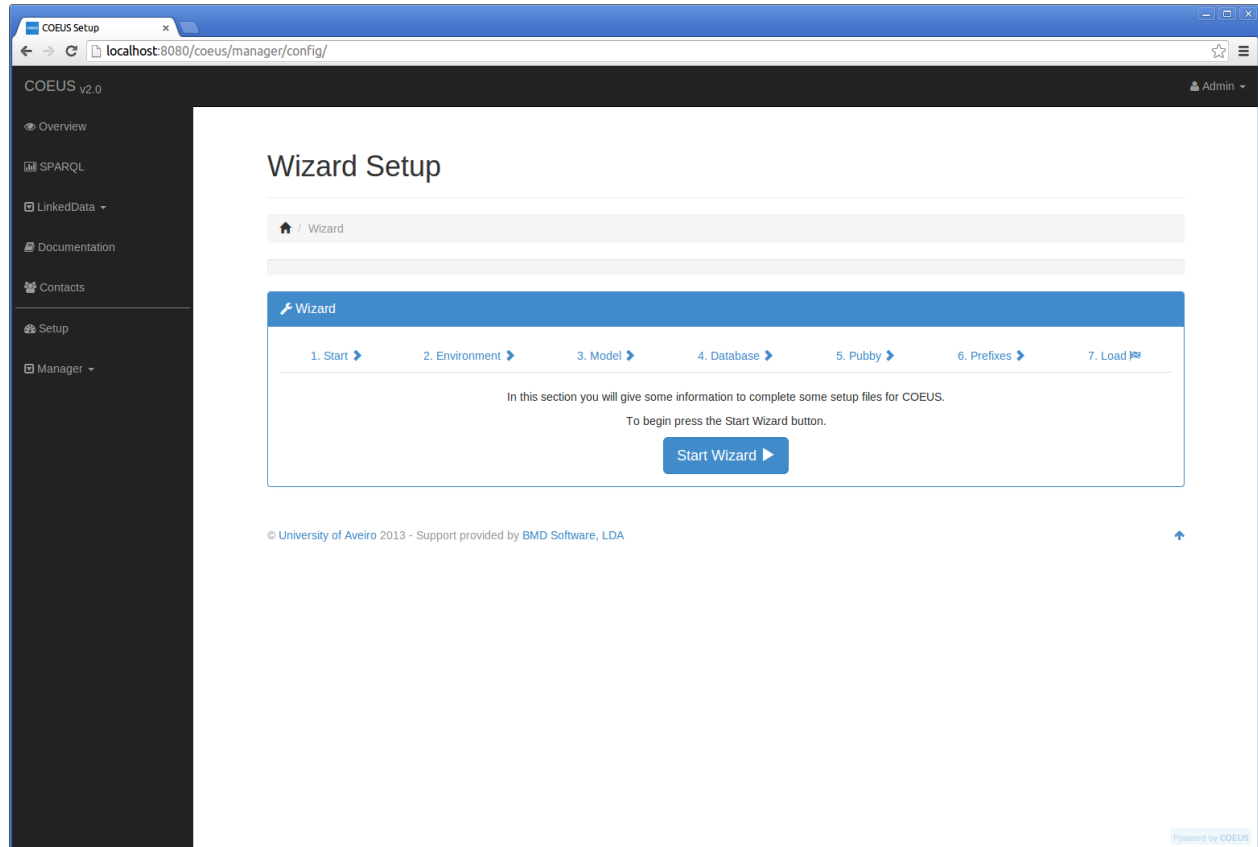


2.2.1.2 Go to setup page

- Click the Setup link in the left menu bar
 - If authentication is enabled use your tomcat credentials.

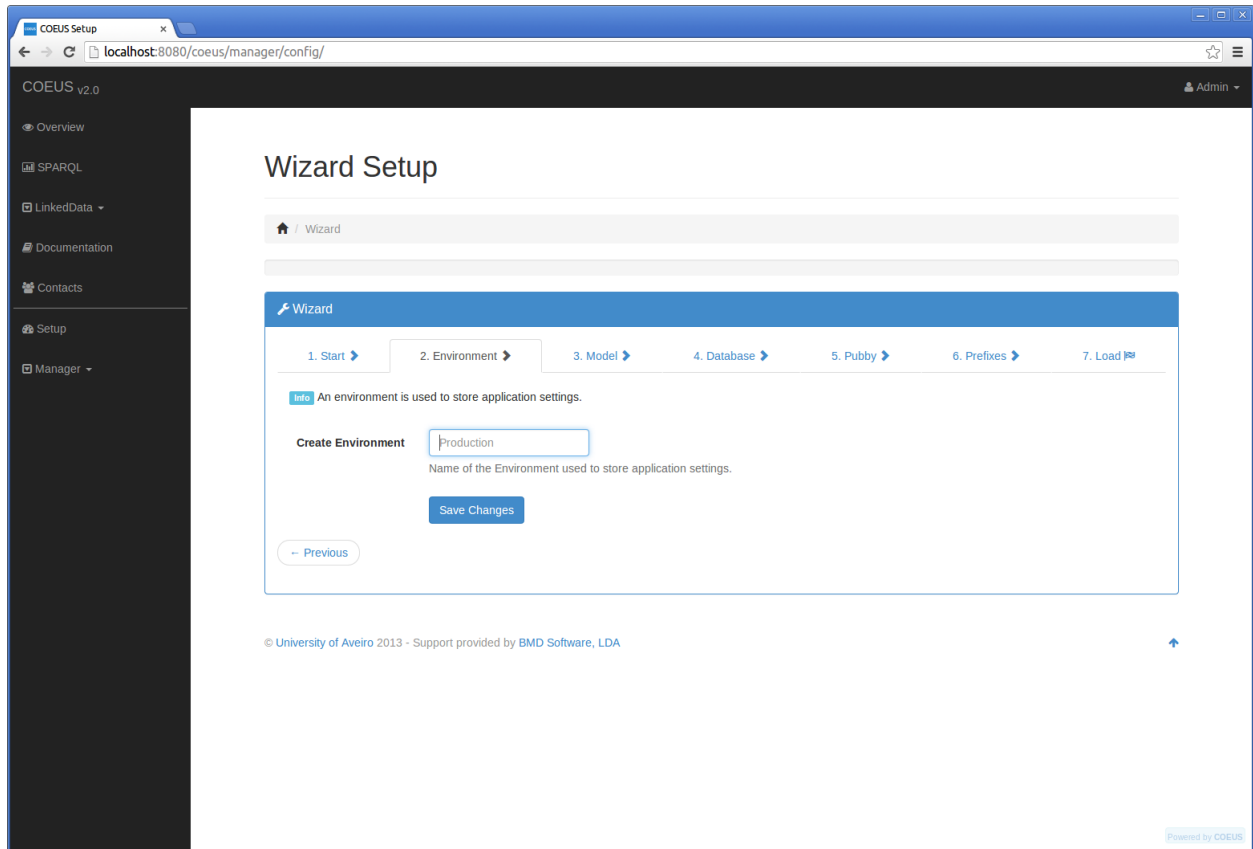
If it is the first time running COEUS' setup, the *Configuration Wizard* will be displayed automatically.

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



2.2.1.3 Environment Configuration

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



Save the changes and press *Next*.

2.2.1.4 Model Setup

Notes

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

The screenshot shows the 'Wizard Setup' page for COEUS v2.0. The left sidebar contains navigation links: Overview, SPARQL, LinkedData, Documentation, Contacts, Setup, and Manager. The main content area is titled 'Wizard Setup' and shows a progress bar with steps 1 to 7. The '3. Model' step is active. The form contains the following fields:

- Name**: Model Test (Name of the Model)
- Description**: COEUS Model Test (Model Description)
- Ontology**: http://bioinformatics.ua.pt/coeus/ontology/ (Ontology Location (URL))
- Prefix**: coeus (The ontology Key Prefix)
- API**: coeus (REST API Key)
- Version**: 1.0a (Ontology Version)
- Keys**: coeus, uavt (API Keys supported)
- Debug**: ☒ Debug Log

A 'Save Changes' button is located at the bottom of the form. A 'Previous' button is also visible at the bottom left.

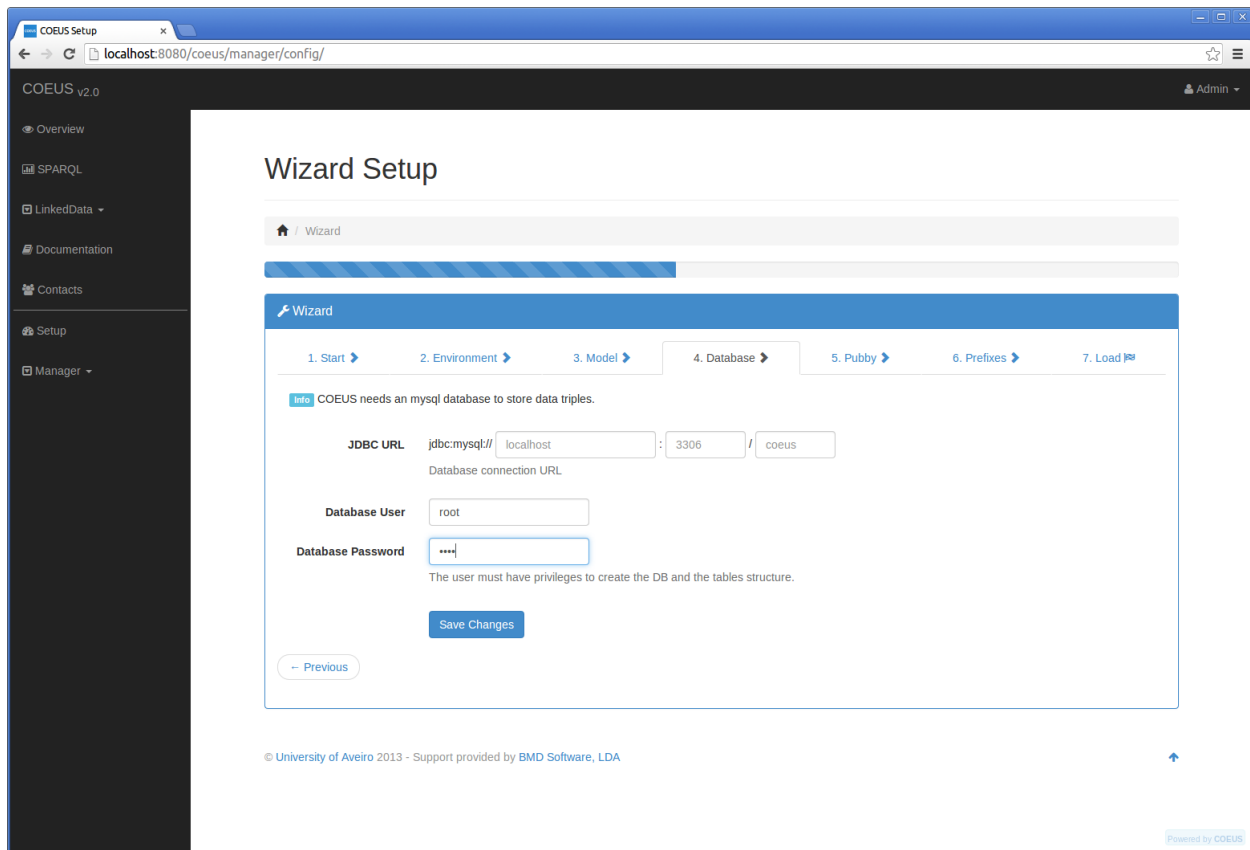
Save the changes and press *Next*.

2.2.1.5 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
4. 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



The screenshot shows the COEUS Setup Wizard in a web browser. The browser address bar shows 'localhost:8080/coeus/manager/config/'. The left sidebar contains navigation links: Overview, SPARQL, LinkedData, Documentation, Contacts, Setup, and Manager. The main content area is titled 'Wizard Setup' and shows a progress bar with steps: 1. Start, 2. Environment, 3. Model, 4. Database (current), 5. Pubby, 6. Prefixes, and 7. Load. An information box states: 'COEUS needs an mysql database to store data triples.' The 'JDBC URL' field is filled with 'jdbc:mysql://localhost:3306/tutorial_xyz'. Below it, the 'Database User' field is filled with 'root' and the 'Database Password' field is filled with 'coeus'. A 'Save Changes' button is visible. At the bottom, there is a 'Previous' button and a footer: '© University of Aveiro 2013 - Support provided by BMD Software, LDA'.

Save the changes and press *Next*.

2.2.1.6 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.

5. 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.

The screenshot shows the 'Wizard Setup' interface for COEUS v2.0. The browser address bar shows 'localhost:8080/coeus/manager/config/'. The left sidebar contains navigation links: Overview, SPARQL, LinkedData, Documentation, Contacts, Setup, and Manager. The main content area is titled 'Wizard Setup' and shows a progress bar with steps: 1. Start, 2. Environment, 3. Model, 4. Database, 5. Pubby (current), 6. Prefixes, and 7. Load. Below the progress bar, an 'Info' box states: 'COEUS make use of Pubby capabilities to provide a LinkedData interface.' The configuration fields are as follows:

Field	Value	Description
URL	http://bioinformatics.ua.pt/coeus/	A project homepage or similar URL, for linking in page titles.
Name	coeus.demo	The name of the project, for display in page titles.
Dataset Base	http://bioinformatics.ua.pt/coeus/resource/	The common URI prefix of the resource identifiers in the SPARQL dataset. Only resources with this prefix will be mapped and made available by Pubby.
Web Resource Prefix	resource/	Prefix to the mapped web URIs.
Web Base	http://localhost:8080/coeus/	The root URL where the Pubby web application is installed.
SPARQL	http://localhost:8080/coeus/sparql	The internal URL of the SPARQL endpoint whose data we want to expose.

An 'Auto-detect the application path' button is positioned above the 'Web Base' field. At the bottom of the form are 'Save Changes' and 'Previous' buttons. The footer of the page reads: '© University of Aveiro 2013 - Support provided by BMD Software, LDA'.

Save the changes and press *Next*.

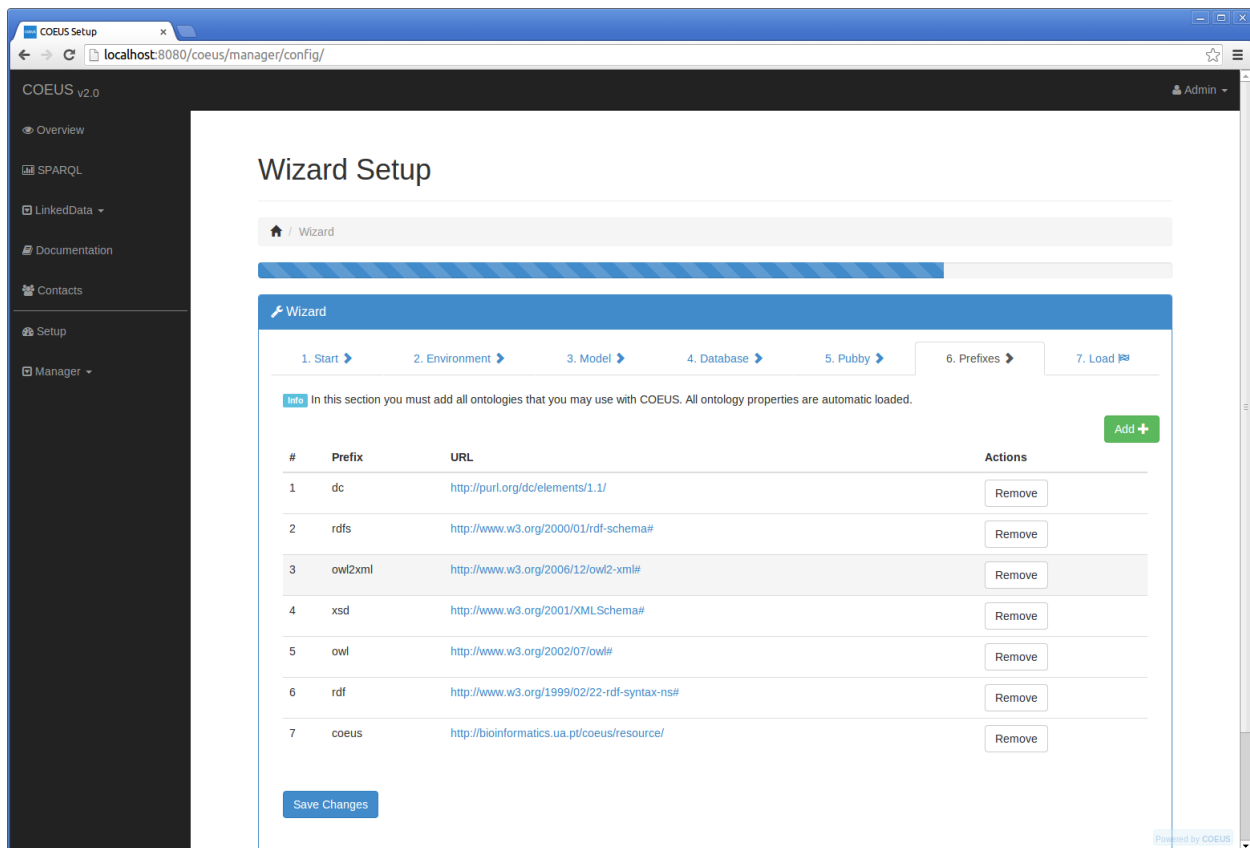
2.2.1.7 Prefix Setup

Notes

For improved performance, COEUS includes an internal semantic name resolution engine. This tool 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/>

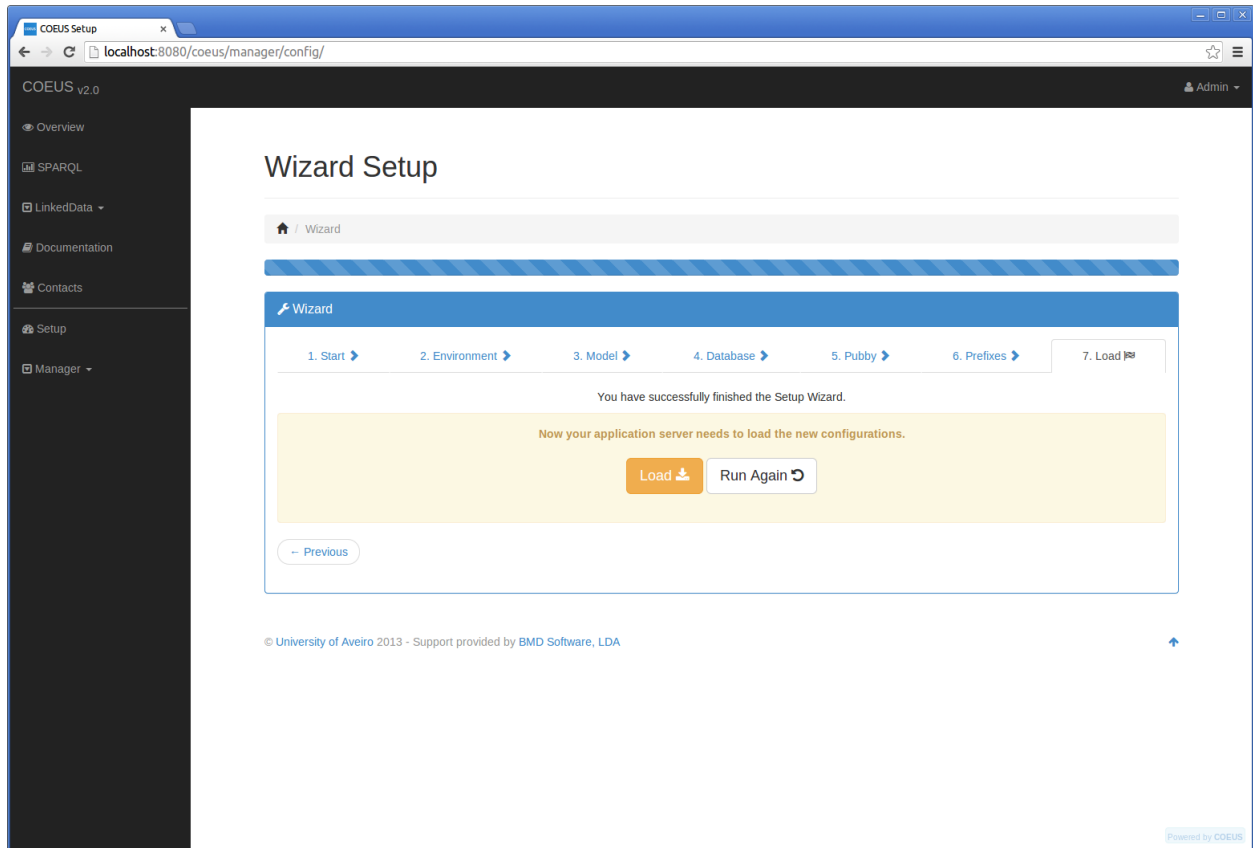


Save the changes and press *Next*.

2.2.1.8 Finish Environment Setup

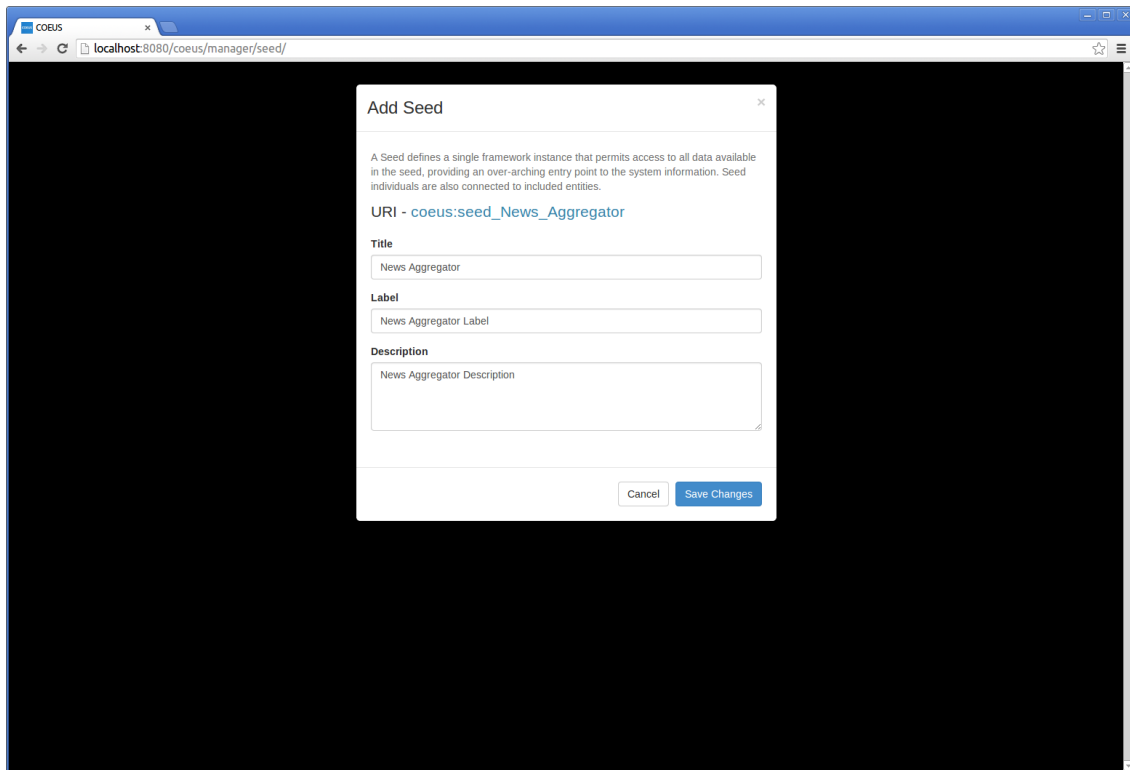
If all went well, pressing the *Load* button will load the new settings to start using COEUS.

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



2.2.2 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.
- 1. Press the *Add Seed* button. On the seed configuration form, set the Seed title, label and a description. The new seed will appear on the seed-listing interface.

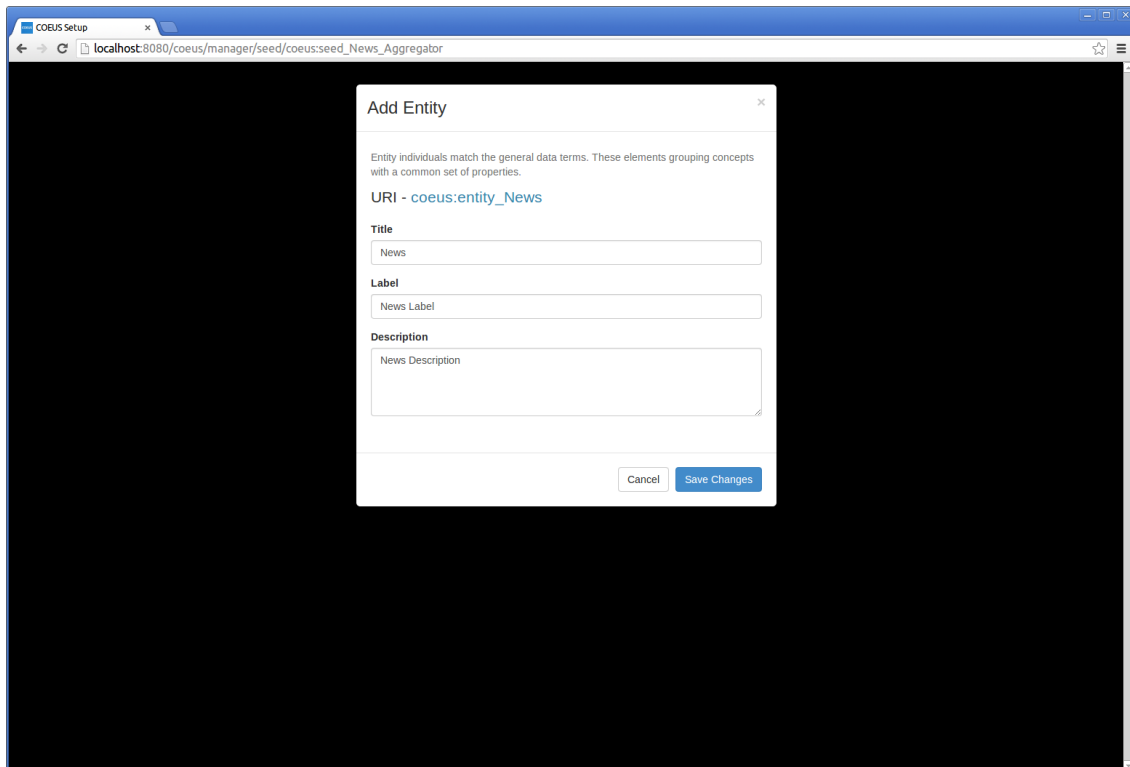
A screenshot of a web browser window showing the COEUS application. The browser's address bar displays 'localhost:8080/coeus/manager/seed/'. The main content area is dark, and a white modal dialog titled 'Add Seed' is centered on the screen. The dialog contains a close button (X) in the top right corner. Below the title, there is a descriptive paragraph: 'A Seed defines a single framework instance that permits access to all data available in the seed, providing an over-arching entry point to the system information. Seed individuals are also connected to included entities.' Below this text, the 'URI' is set to 'coeus:seed_News_Aggregator'. The form includes three input fields: 'Title' with the value 'News Aggregator', 'Label' with the value 'News Aggregator Label', and 'Description' with the value 'News Aggregator Description'. At the bottom of the dialog are two buttons: 'Cancel' and 'Save Changes'.

2. To start configuring the internal data structure, press the *Choose* button on the new seed. COEUS will display the Seed 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. The action toolbox provides direct access buttons to the most common seed actions.

2.2.3 Adding the News Entity

With the environment and seed already configured, we can transpose our integration setup, from the requirements analysis to the actual seed.

- On the seed dashboard, press the *Add Entity* button.
 - Fill in the required details, Title, Label and Description, and save the changes.

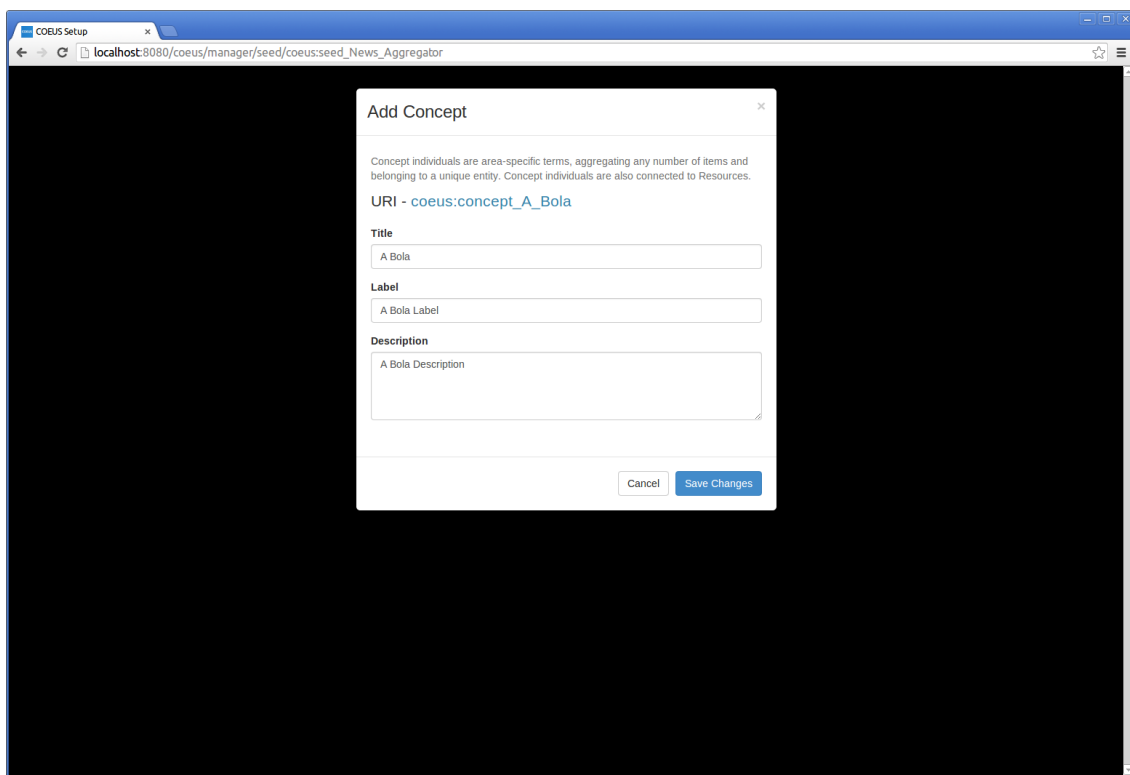


The screenshot shows a web browser window titled "COEUS Setup" with the address bar displaying "localhost:8080/coeus/manager/seed/coeus:seed_News_Aggregator". The main content area is dark blue. A white "Add Entity" dialog box is centered on the screen. The dialog box has a title bar with a close button (X). Below the title bar, there is a text area with the text: "Entity individuals match the general data terms. These elements grouping concepts with a common set of properties." Below this text, the URI "coeus:entity_News" is displayed. The dialog box contains three input fields: "Title" with the value "News", "Label" with the value "News Label", and "Description" with the value "News Description". At the bottom of the dialog box, there are two buttons: "Cancel" and "Save Changes".

2.2.4 Adding the news Concepts

New concepts can be added directly from the seed dashboard knowledge base tree, from the entities list or from the concept list.

- We need to create a new concept for each news outlet.
- Press the *Concept +* button and fill in the required details in the displayed modal box (Title, Label and Description).
 - Repeat for the tree remaining news outlets.



The screenshot shows a web browser window titled "COEUS Setup" with the address bar displaying "localhost:8080/coeus/manager/seed/coeus:seed_News_Aggregator". A modal box titled "Add Concept" is open in the center. The modal contains the following text and fields:

Concept individuals are area-specific terms, aggregating any number of items and belonging to a unique entity. Concept individuals are also connected to Resources.

URI - [coeus:concept_A_Bola](#)

Title

Label

Description

At the bottom right of the modal are two buttons: "Cancel" and "Save Changes".

2.2.5 Configuring Resources

Notes

- Resources are related to Concepts
 - **Many-to-One:** *one* resource has *one* concept, *one* concept can have many *resources*
- Resources store the metadata configuring the **ETL** being performed for each Concept
- Examples
 - Endpoint `coeus:endpoint`
 - **CSV:** `http://bioinformatics.ua.pt/coeus/list.csv`
 - **SQL:** `jdbc:mysql://localhost:3306/?user=&password=`
 - Query `coeus:query`
 - **SQL:** `SELECT * FROM genes;`
 - **XML:** `//entry`

We need to define the resource integration properties for each of the four Concepts:

- Select a Concept from the list
- Press the *Resource +* button to display the new Resource modal
- Fill in the resource metadata
 - Title, Label and Comment: matching the given news outlet
 - Order: the integration order for this resource (irrelevant for this example)
 - Extends: when using an integration graph the Concept data extraction requires data from other Concept (previously loaded)
 - Method: the type of data integration being performed, select *cache* to copy all data to the COEUS knowledge base
 - Publisher: the resource data format, select *XML*

- Endpoint: resource URI, insert each news item respective RSS/Atom URL
- Query: the base query that will provide the integration list, insert `//item`

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/coeus/manager/seed/coeus:seed_News_Aggregator'. The main content area is a dark blue background with a white 'Add Resource' dialog box centered on the screen. The dialog box has a title bar with 'COEUS Setup' and a close button. Inside the dialog, there is a paragraph explaining that resource individuals are used to store external resource integration properties and that the configuration is specialized with CSV, XML, JSON, RDF, TTL, SQL, and SPARQL classes. Below this, the 'Resource URI' is set to 'coeus:resource_A_Bola'. The form includes several fields: 'Title' (A Bola), 'Order' (1), 'Label' (A Bola Label), 'Description' (A Bola Description), 'Extends' (concept_Abola), 'Method' (cache), 'Publisher' (xml), 'Endpoint' (http://www.abola.pt/rss/index.aspx), and 'Query' (//item). At the bottom right of the dialog are 'Cancel' and 'Save Changes' buttons.

In this simple News Reader example, the process is similar to all concepts and resources. With all resources introduced, we must specify their selectors to finish the configuration.

2.2.6 Configuring Selectors

Notes

- Selectors specify the mapping between predicates and the extraction being performed
 - Applied to the Resource output list
- Examples
 - **CSV**, column number
 - dc:identifier, 0 (matches column zero)
 - **XML**, XPath query
 - dc:title, //title
 - **JSON**, JSONPath query
 - dc:description, entry.description
 - **SQL**, column name
 - dc:identifier, "id" (matches "id" column from SELECT)
 - **SPARQL**, variable name
 - dc:description, d (matches ?d variable)

On the resources listing interface, press the *Configuration* button to setup the selectors.

- Press the *New* button to add a selector
- Let's start with the news identifier selector
 - Key: Select this checkbox to use this selector as the identity generator for the new Items being generated
 - Query: the selector query to extract data from the resource list, in this case use the XPath *guid*
 - Regex: an optional regular expression to filter or validate values, use *[0-9]{5,}*

- Property: selectors can import data for one or multiple semantic predicates, add *dc:identifier* to the list (use the Search Property input to search and press *Enter* to add to the list)

The screenshot shows a web browser window with the URL `localhost:8080/coeus/manager/selector/coeus.resource_Abola`. A modal dialog titled "Add Selector" is open. It contains the following fields and controls:

- Selector URI**: `coeus.selector_Abola_xml_id`
- Key**: A checkbox that is checked.
- Title**: A text input field containing `xml id`.
- Label**: A text input field containing `xml_id`.
- Query**: A text input field containing `guid`.
- Regex**: A text input field containing `[0-9][.5]`.
- Search Property**: A text input field with a green search button.
- Property List**: A list box containing `dc:identifier` with a red remove button.
- Buttons**: "Close" and "Save Changes" at the bottom right.

- Fill in the remaining selectors
 - **dc:title**: entry title - title
 - **dc:description**: entry text - description
 - **rdfs:seeAlso**: entry link to original news - link
 - **dc:date**: entry publication date - pubDate

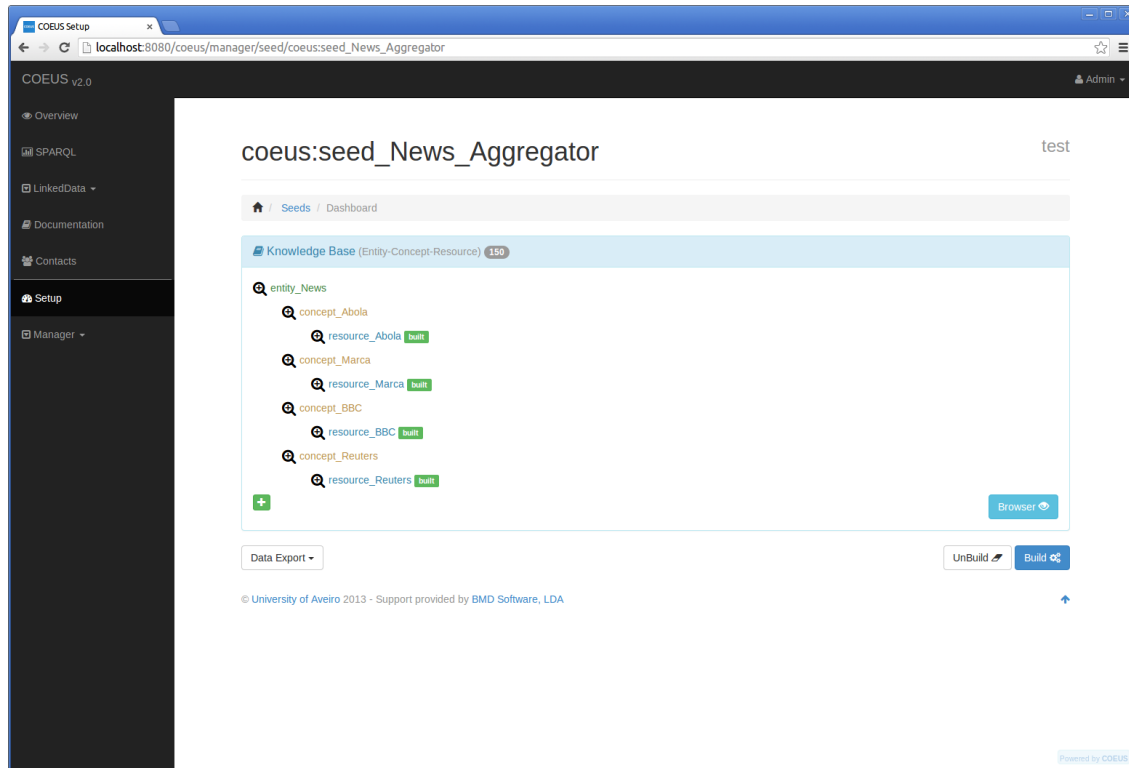
2.2.6.1 Reusing Selectors

Since we are working with similar data sources (i.e., using the same configurations), we do not need to define new selectors for each Resource. We can reuse existing selectors, making the integration process much simpler.

- On the Resource configuration interface, press the "Existing +" button to associate a previously configured selector with the resource.

2.2.7 Building the knowledge base

With all the entities, concepts, resources and selectors configured, you can go back to the seed dashboard. The seed structure tree, should list a single entity (News), with four concepts each with its own resource (BBC, Reuters, Marca and A Bola).



If you are using an IDE, view the server output to check what's happening behind the scenes:

- Traverse Resources list
 - Check dependencies
 - Concept Extends
 - Process Resource data
 - Apply Resource query
 - Process selectors
 - Apply selectors' queries
 - Build axioms with properties

2.3 Testing

With the knowledge base built, COEUS provides multiple methods to access the aggregated dataset. In this first example, we will just use the SPARQL endpoint to performs simple queries.

2.3.1 List all news items

```
PREFIX coeus: <http://bioinformatics.ua.pt/coeus/resource/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>

SELECT ?item ?title ?description {
  ?item a coeus:Item .
  ?item dc:title ?title .
  ?item dc:description ?description .
  ?item dc:date ?date
}
```

2.3.2 List all news items with specific outlet

```
PREFIX coeus: <http://bioinformatics.ua.pt/coeus/resource/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>

SELECT ?item ?title ?description {
  ?item a coeus:Item .
  ?item dc:title ?title .
  ?item dc:description ?description .
  ?item dc:date ?date .
  ?item coeus:hasConcept coeus:concept_BBC
}
```

2.3.3 List all news items ordered by date

```
PREFIX coeus: <http://bioinformatics.ua.pt/coeus/resource/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>

SELECT ?date ?item ?title ?description {
  ?item a coeus:Item .
  ?item dc:title ?title .
  ?item dc:description ?description .
  ?item dc:date ?date
} ORDER BY DESC(?date)
```

3 Proteomics Knowledge Base

With second example we aim to build a proteomics knowledge base, aggregating miscellaneous data for associated with proteins. This scenario is more complex than the first one, as we will integrate and combine data from distinct heterogeneous resources.

The Proteomics KB will feature a mashup of data from HGNC, UniProt, InterPro, and OMIM. We will adopt the same workflow for this tutorial: (1) requirements analysis, (2) configuration & building, (3) testing.

3.1 Requirements Analysis

The first step is to think about a shared model to organize these data. There are lots of distinct solutions, we are proposing only one that fits COEUS' internal model. This model is a subset of the one used in [Diseasecard](#).

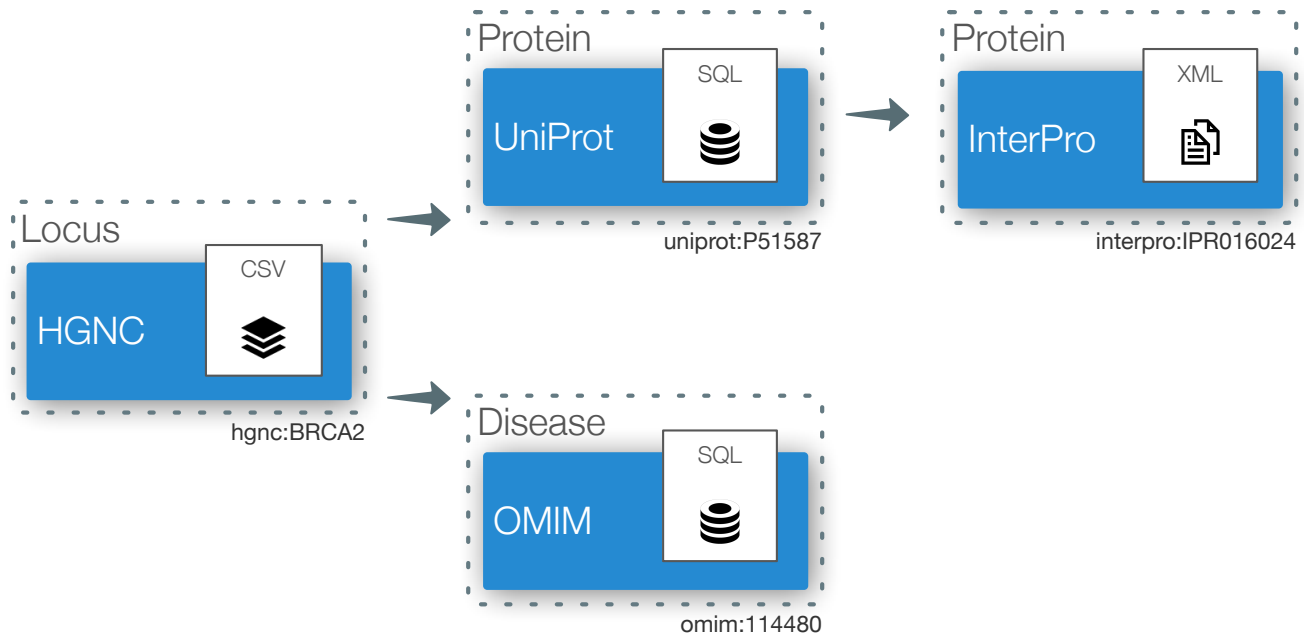
- **Entity:** Gene
 - **Concept:** HGNC
- **Entity:** Protein
 - **Concept:** UniProt
 - **Concept:** InterPro
- **Entity:** Disease
 - **Concept:** OMIM

To simplify this model, we will focus on loading unique identifiers for each data source (**dc:identifier**).

More important than just importing data, we need to establish new relationship. Hence, data will be loaded from our own "dependency graph". This graph can take many shapes and distinct structures. The following data-loading map relates the data from the multiple sources being integrated, which will result in a new semantic network in the proteomics knowledge base.

- **HGNC** (starting seed, loaded from file, *CSV*)

- **UniProt** (matches from UniProt query, *SQL*)
 - **InterPro** (from individual UniProt entries, *XML*)
- **OMIM** (from OMIM genemap, *SQL*)
 - **Orphanet** (from OrphaData, *SQL*)



Further details regarding the resources and selectors configurations for each Concept are described in the **Configuring Resources & Selectors** activity.

3.2 Configuration

COEUS allows multiple seeds in a single environment. As such, the Proteomics KB will be a new seed within the already set environment. (If you have not setup the COEUS environment with the News Reader example, go back and do the initial configuration tasks.)

3.2.1 Creating the Proteomics KB Seed

- Go back to the seeds list and create the new *Proteomics* seed.

3.2.2 Adding Entities & Concepts

- Replicate the seed organisation obtained in the requirements analysis using the COEUS GUI.

- All steps were covered in the News Reader example.

3.2.3 Configuring Resources & Selectors

In this more complex scenario, the integration is based on a dependency graph: to discover information about a certain concept, we need details (in this case, identifiers) of a concept that was previously loaded. Hence, on the resource configuration tasks, some resources extend concepts beyond their original pair.

3.2.3.1 HGNC

HGNC data is easily obtained in CSV format from the [GeneNames website](http://www.genenames.org/). The download can be customized to feature multiple columns, including gene symbol, names, chromosome locations or reference sequence identifiers, among others. For instance, the following CSV file includes the approved gene symbol, HGNC identifier and name:

- http://www.genenames.org/cgi-bin/hgnc_downloads.../

To reduce the tutorial server load, a smaller listing will be used.

3.2.3.1.1 Resource

- Add a new Resource to the HGNC Concept
 - Order: 1 (*this is the starting point*)
 - Extends: Concept HGNC (*has no dependencies*)
 - Method: cache (*copy everything*)
 - Publisher: CSV
 - Endpoint: <http://bioinformatics.ua.pt/diseasecard/hgnc.csv>
 - CSV Starting Line: 0 (*no headers*)

3.2.3.1.2 Selectors

For HGNC data, we will extract the gene symbol as the identifier and the gene name as a simple description. Hence, we need two distinct selectors.

- Id
 - Query: 1 (*column number 1*)

- Property: dc:identifier
- Name
 - Query: 2 (*column number 2*)
 - Property: dc:description

3.2.3.2 UniProt

According to the Proteomics Knowledge Base graph, UniProt entries will be obtained based on the HGNC concept items. This has three main consequences for the UniProt Resource:

- it must be loaded after the HGNC resource (higher order property)
- it must extend the HGNC resource (to use the loaded identifiers)
- it must use a template endpoint URI or query to use the HGNC identifiers (apply the keyword *#replace#* with the gene symbol)

3.2.3.2.1 Resource

- Add a new Resource to the UniProt Concept
 - Order: 2 (*higher than HGNC*)
 - Extends: Concept HGNC (*the dependency*)
 - Method: cache (*copy everything*)
 - Publisher: SQL
 - Host: localhost
 - Port: 3306
 - Database: hummer
 - Login/Password: tutorial / coeus
 - Query: SELECT uniprot FROM uniprot WHERE hgnc = '*#replace#*';

3.2.3.2.2 Selectors

For UniProt data, we will extract the identifier.

- Id
 - Query: uniprot (*the SQL SELECT column name*)
 - Property: dc:identifier

COEUS doesn't just copy data into its knowledge base. You can configure a model to add properties to items that are already in the triplestore. To show this, we will import the UniProt items name. COEUS will use the dependency concept identifier to perform the query/endpoint composition.

3.2.3.2.3 Resource

- Add a new Resource to complete the Uniprot Concept
 - Order: 3 (*higher than UniProt*)
 - Extends: Concept UniProt (*the dependency*)
 - Method: complete (*add statements*)
 - Publisher: SQL
 - Host: localhost
 - Port: 3306
 - Database: hummer
 - Login/Password: tutorial / coeus
 - Query: SELECT name FROM uniprot WHERE uniprot = '#replace#';

3.2.3.2.4 Selectors

- Name
 - Query: name
 - Property: dc:title

3.2.3.3 InterPro

InterPro items are also discovered through UniProt. In this case, we will use UniProt entries' XML to extract InterPro identifiers.

3.2.3.3.1 Resource

- Add a new Resource to the InterPro Concept
 - Order: 3 (*higher than UniProt*)
 - Extends: Concept UniProt (*the dependency*)
 - Method: cache (*copy everything*)
 - Publisher: XML
 - Endpoint: `http://uniprot.org/uniprot/#replace#.xml`
 - Query: `//entry`

3.2.3.3.2 Selectors

For InterPro data, we will only extract the identifier.

- Id
 - Query (XPath): `//dbReference[@type='InterPro']/@id`
 - Property: `dc:identifier`

3.2.3.4 OMIM

OMIM items are obtained from an external database mapping in HUMMER, based on HGNC gene symbols.

3.2.3.4.1 Resource

- Add a new Resource to the OMIM Concept
 - Order: 2 (*higher than HGNC*)
 - Extends: Concept HGNC (*the dependency*)
 - Method: cache (*copy everything*)
 - Publisher: SQL
 - Host: localhost
 - Port: 3306
 - Database: hummer
 - Login/Password: tutorial / coeus
 - Query: SELECT omim FROM omim WHERE hgnc = '#replace#';

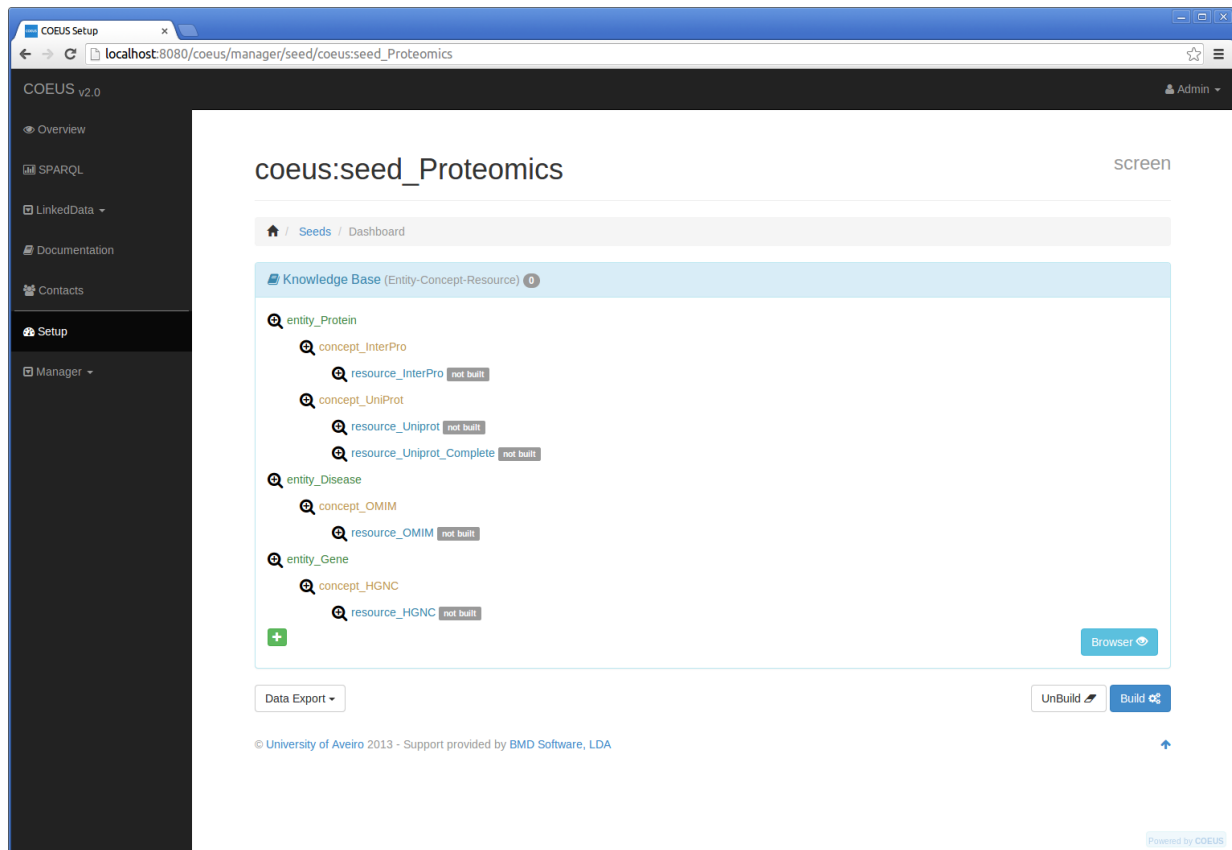
3.2.3.4.2 Selectors

For OMIM data, we will only extract the identifier.

- Id
 - Query: omim
 - Property: dc:identifier

3.2.4 Building the knowledge base

As previously, the final step is to integrate the data. Go to the seed dashboard and make sure the knowledge base structure matches what was obtained in the requirements analysis. To start the integration process you must click on *Build* button.



Testing

With the knowledge base built, COEUS provides multiple methods to access the aggregated dataset.

3.2.5 List all UniProt individuals

```
PREFIX coeus: <http://bioinformatics.ua.pt/coeus/resource/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>

SELECT * {
  ?uniprot coeus:hasConcept coeus:concept_Uniprot .
  ?uniprot dc:identifier ?id
}
```

3.2.6 List protein info for BRCA2

```
PREFIX coeus: <http://bioinformatics.ua.pt/coeus/resource/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>

SELECT ?name ?uniprot ?interpro {
  ?g dc:description ?name .
  ?g dc:identifier "BRCA2" .
  ?g coeus:isAssociatedTo ?up .
  ?up coeus:hasConcept coeus:concept_Uniprot .
  ?up dc:identifier ?uniprot .
  ?up coeus:isAssociatedTo ?ip .
  ?ip coeus:hasConcept coeus:concept_InterPro .
  ?ip dc:identifier ?interpro
}
```

3.2.7 List protein info for all genes (where available)

```
PREFIX coeus: <http://bioinformatics.ua.pt/coeus/resource/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>

SELECT ?id ?name ?uniprot ?interpro {
  ?g dc:description ?name .
  ?g dc:identifier ?id .
  ?g coeus:isAssociatedTo ?up .
  ?up coeus:hasConcept coeus:concept_Uniprot .
  ?up dc:identifier ?uniprot .
  ?up coeus:isAssociatedTo ?ip .
  ?ip coeus:hasConcept coeus:concept_InterPro .
  ?ip dc:identifier ?interpro
}
```

3.3 LinkedData browsing

- Go to: http://localhost:8080/coeus/resource/seed_Proteomics

4 Overview

4.1 Model

- Seed – Entity – Concept – Item
 - Resource – Selector

Notes

- Internal structure
- Can accommodate multiple data models

4.2 Integration

- Multiple entry points (connectors)
 - CSV
 - XML
 - JSON (not covered)
 - SQL
 - SPARQL (not covered)
 - LinkedData (not covered)
 - RDF (not covered)
- Flexible
 - Map to any ontology
 - Create complex axioms

4.3 API

- Multiple access points
 - SPARQL
 - LinkedData
 - Java
 - Javascript
 - “REST”
 - Ruby gem (gem install coeus)
 - Python package (pip install coeus, by Luís Bastião)
- Flexible
 - “Anything” can be used for client-side development

4.4 Documentation

Full information for everything covered (and not covered) available online at <http://bioinformatics.ua.pt/coeus/documentation/>.

5 Feedback

Your feedback is highly appreciated; please submit your opinions on the form at <http://goo.gl/gcphs3>.