Running KLD on OS X

Table of Contents

Install MongoDB Community Edition on OS X

Install Homebrew

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Update Homebrew

```
brew update
```

Install MongoDB

```
brew install mongodb
```

Install Node.js on OS X

https://nodejs.org/en/download/

Install JDK on OS X

https://docs.oracle.com/javase/8/docs/technotes/guides/install/mac_jdk.html

Install KLD on OS X

Download GIT client for OS X at https://git-scm.com/download/mac

```
git clone https://github.com/koneksys/KLD.git
cd KLD
cd api
npm install
cd ../middleware
npm install
cd ../client
npm install
```

Run KLD on OS X

Instruction: Run each process in different terminal

Create directory for MongoDB (KLD user database)

```
mkdir –p ~/database
```

Run MongoDB

```
sudo mongod ––dbpath ~/database
```

Run KLD Middleware

```
cd KLD/middleware
npm start conf/conf.js
```

Run KLD API Server

```
cd KLD/api
npm start conf/conf.js
```

Run KLD Client. ***There are 3 ways you can run KLD client in different contexts.***

Run KLD Client in Development Mode

With Development Mode, you will run a simple web server that watches all KLD client source files. Each source file will be watched by the KLD build system and entire KLD client web app will be built automatically and the result of the change will be updated on the web browser in real-time when developers make a change on files under src folder.

```
cd KLD/client

npm start

Open http://localhost:9000
```

Run KLD Client using Built-In Node.js web server

With Built-In Node.js web server, you will run KLD client using a built-In Node.js based web server. Yuri have the build KLD client source files using “npm run build”, which the built files will be saved in asset.default folder. You can start the web server using “npm run server” command line.

```
cd KLD/client

npm run build

npm run server

Open http://localhost:3000
```

Run KLD Client on an Apache Web Server

You can reuse Apache Web Server to run KLD client which allows you configure advanced web application deployment.

```
cd KLD/client

npm run build

use “asset.default” folder as an Apache document root folder
```

If you want to setup an Apache virtual host, the following is the example of KLD client virtual host setting. To assume that your server IP is 192.168.0.100 and your KLD GIT repo is relocated at /var/www/KLD

```
NameVirtualHost 192.168.0.100
<VirtualHost 192.168.0.100:80>
        ServerName 192.168.0.100
        DocumentRoot /var/www/KLD/client/asset.default
        RewriteEngine On
        RewriteCond %{DOCUMENT_ROOT}%{REQUEST_URI} -f [OR]
        RewriteCond %{DOCUMENT_ROOT}%{REQUEST_URI} -d
        RewriteRule ^ - [L]
        RewriteRule ^ /index.html
</VirtualHost>
```

Create sample organization and user account database

```
cd KLD/api

npm run orgadd ExampleCompany http://example.com info@example.com "We are ExampleCompany"

npm run useradd  demo@example.com demo ExampleCompany true
```

Open KLD Client

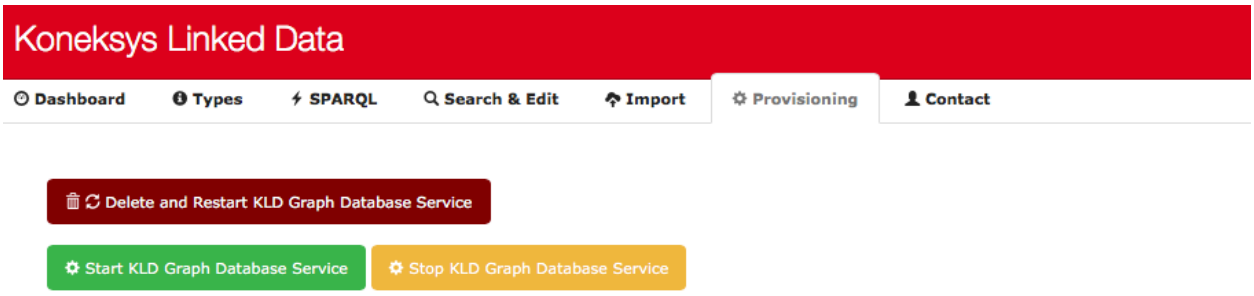
http://localhost:9000/ for development mode
http://localhost:3000/ for built-in Node.js web server

Login account
user: demo@example.com
password: demo

Before using KLD client, initialize KLD Graph Database Service is required.

If it is the first time to run KLD on your machine, you have to initialize KLD graph database by clicking "Delete and Restart KLD Graph Database Service" button.

If you already initialized KLD Graph Database Service, you can start and stop KLD Graph Database Service using the two buttons "Start" and "Stop"

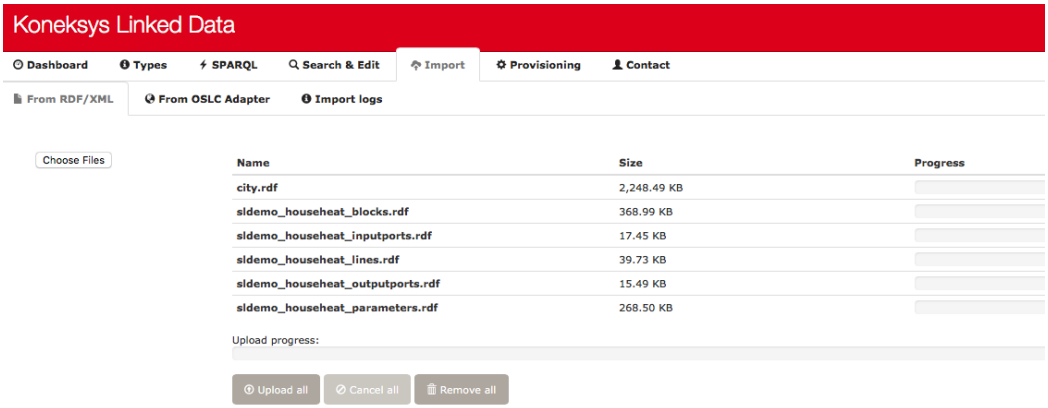


Generic Linked Data Use Cases in KLD

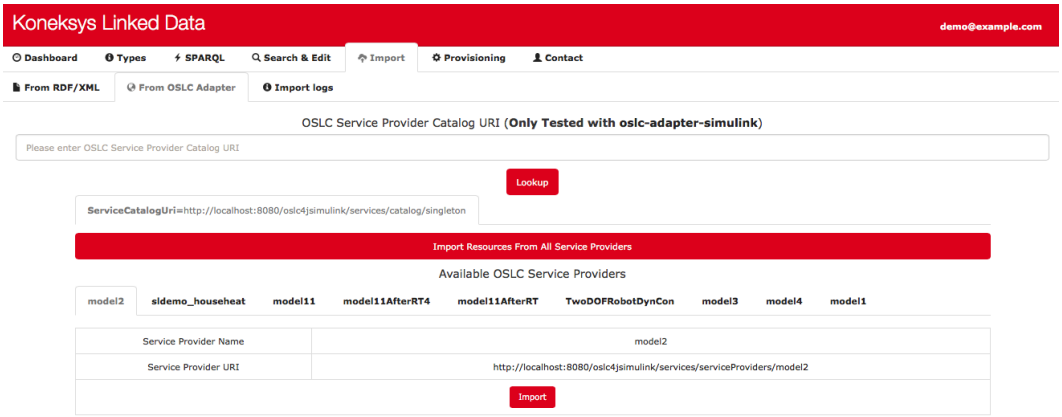
KLD provides generic use cases of using linked data for your data. The following are 4 basic use cases you use KLD to manage your data in a semantic form.

Import RDF resources

You can import RDF/XML files by going to Import > From RDF/XML. There is a number of test RDF/XML files available in *KLD/middleware/exampleRDF,* which allows you can experience KLD features.

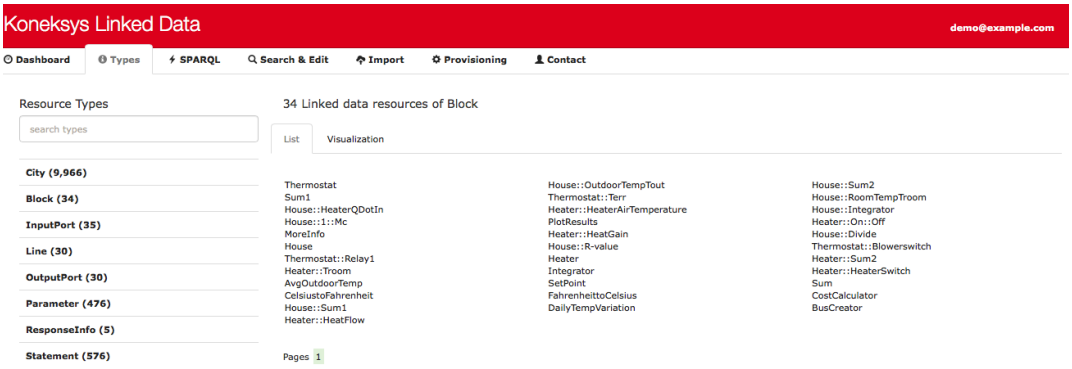


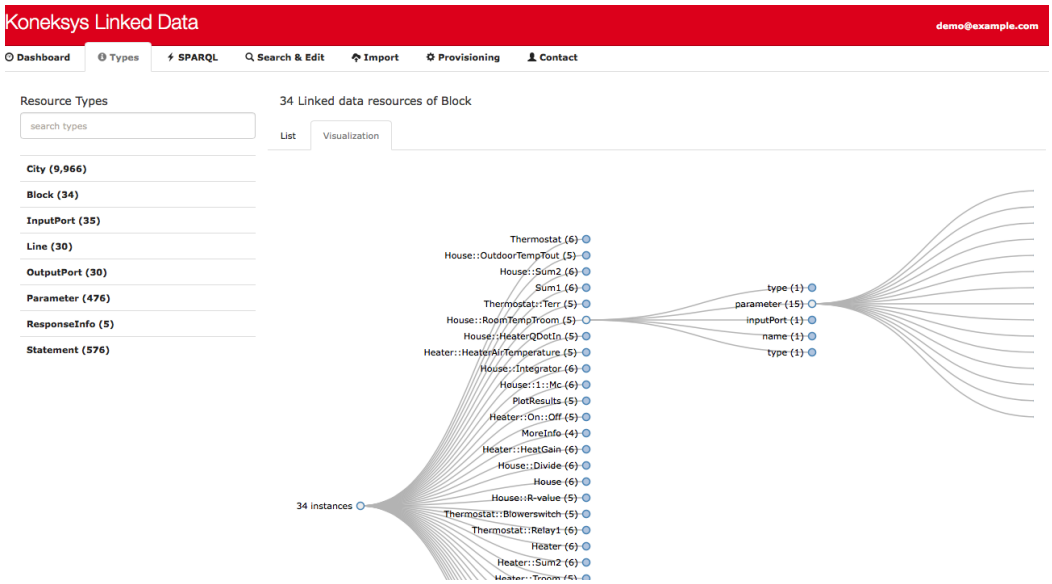
In addition, importing RDF resources from an OSLC adapter is supported in KLD but the current KLD implementation ONLY TESTED with OSLC Adapter Simulink (https://github.com/ld4mbse/oslc-adapter-simulink). You have to install and run OSLC Adapter Simulink on the same machine for experiencing this feature. Other OSLC adapter supports will be updated on KLD Github.



Browse RDF resources

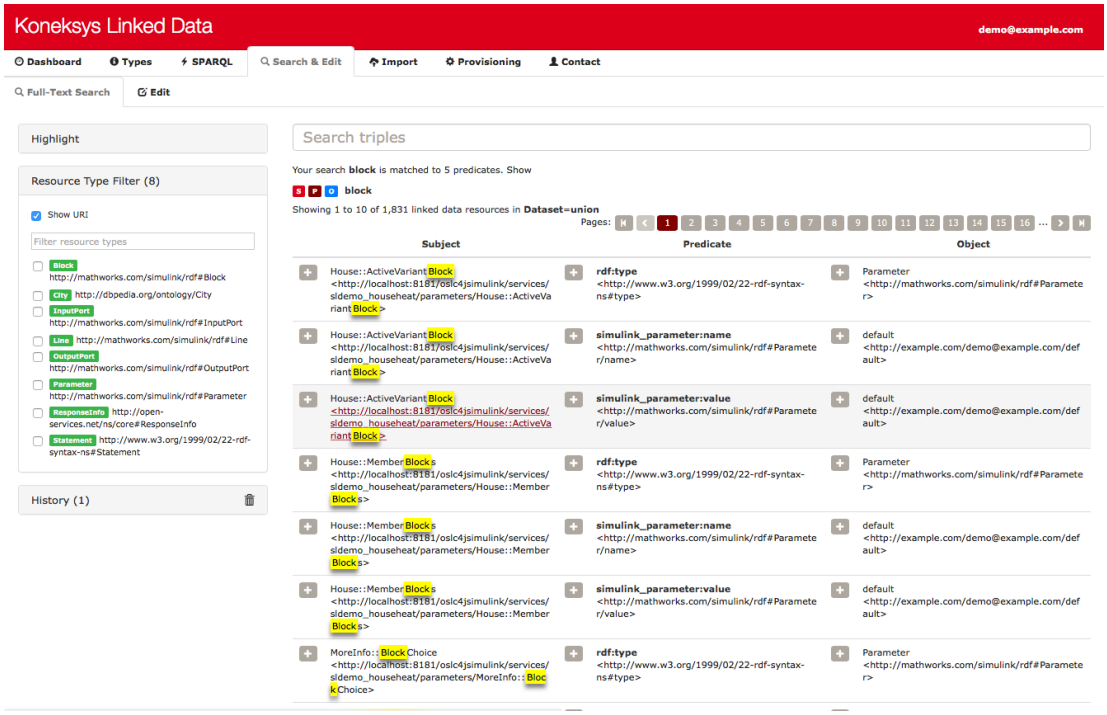
Browsing RDF resources can be used when you have RDF resources in the KLD graph database. Browsing RDF resources with KLD will start at available RDF types in the system, which you can see al types at the left side in Types tab, and then you can browse all instances of each type through list and visualization interface.





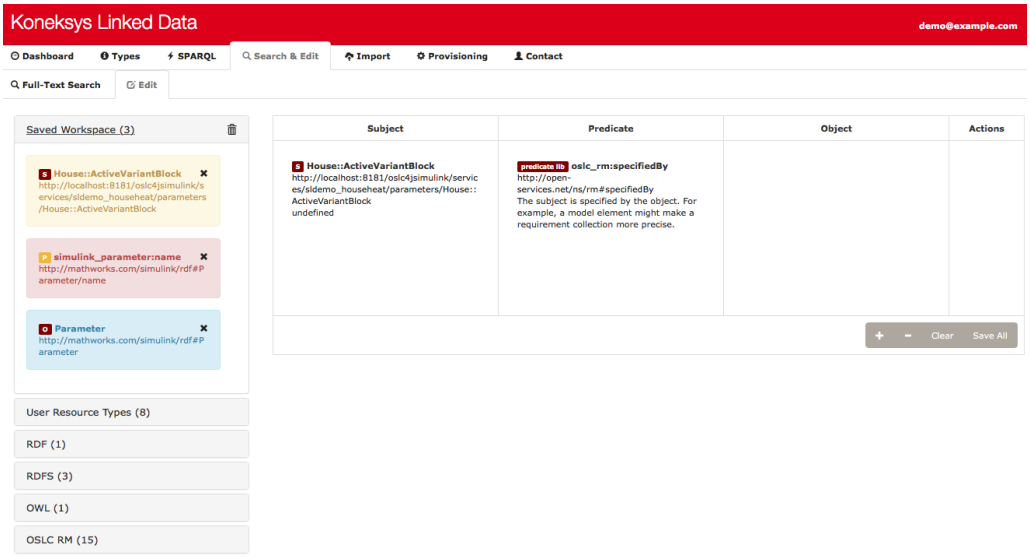
Search RDF resources

Searching RDF resources with your specific words can be done by Search feature which allows you to perform full-text search over entire graph database in the system. The scope of the full-text search to find a matched word in resource URIs and literal values using a very fast search technique using SOLR search server.



Edit RDF resources

KLD allows you edit RDF resources using drag and drop interface.



Query RDF resources using SPARQL

A semantic analysis using SAPRQL can be done by the built-in KLD SPARQL client. You can query RDF resource using SPARQL that will performed the query on the KLD graph database directly.