FUSION NODE

Version 1.0.x

FUSION NODE – DATA REPORTING
VERSION 1

Node Overview

The Fusion Node Data Reporting is run locally by data providers to report data to a central HUB. The node exposes SDMX compliant web services for the HUB to obtain data on demand. The node can connect to a local database to load data into the node on demand.

Fusion Node: 1.0.x
Document Last Updated: June 27th 2017

Version 20170627

1 Contents

2 Overview

The Fusion Node provides a standalone web application for the purpose of automatic data validation, transformation, and reporting to a central Fusion Registry Hub.

The node launch is able to pull metadata from the Hub relevant to the data reporter that is running the node. This information includes datasets that the data reporter has been configured to report data for and any related metadata for data validation of these datasets.

The node generates a folder structure on the client system based on the datasets the data reporter is able to report data for. The data reporter can optionally link a node to a MySQL, SQL Server, or Oracle database for the purpose of data reporting.

To load data into the node, the data reporter simply places a SDMX, CSV, or Excel file into the relevant folder on the file system, the node validates the data, and loads it into memory for dissemination. On successful load the node automatically registers the data with the Hub. The Hub is then able to obtain the data from the node on demand.

If a database connection is defined, the data reporter can also pull data from their database into the node. On database pull, the node exports the table contents for the dataset, and writes the dataset to the node's file system, in the relevant folder. The file is then validated, and loaded into the node, and the data is automatically registered with the Hub.

The node runs a highly optimised data store which provides fast access to data, exposed via SDMX compliant web services, and as such the node can be used to provide data to a Registry hub as well as support any other data service built on the node, including web dissemination, or internal processes.

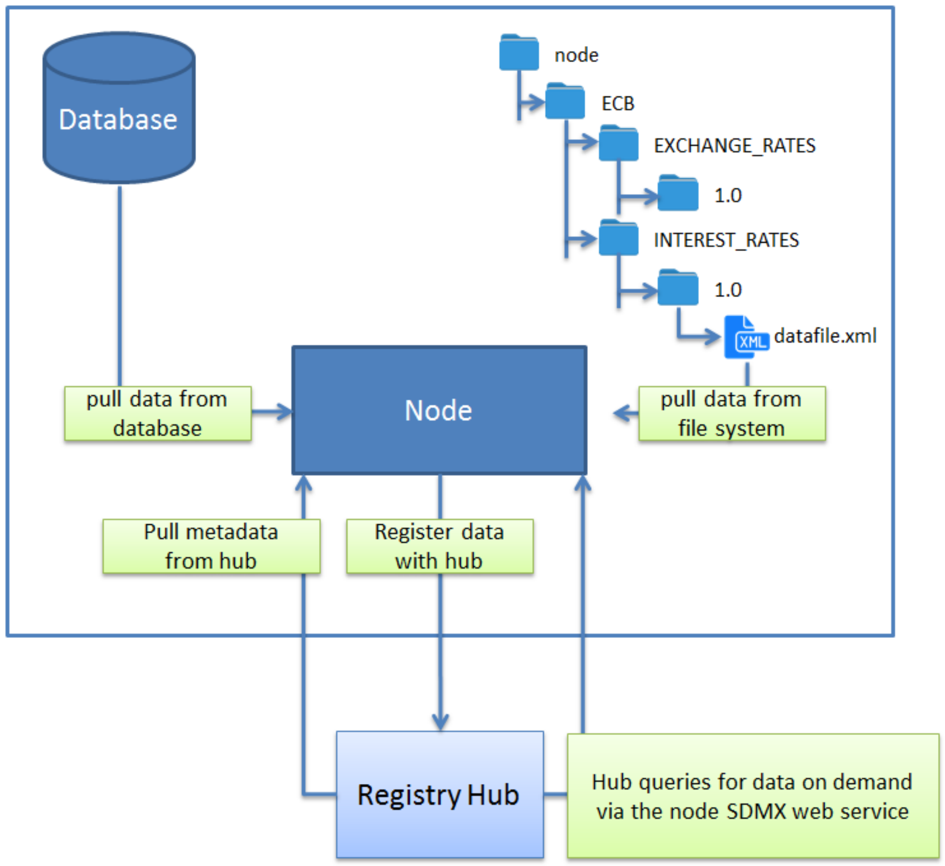


Figure 1 showing the node architecture

3 Deployment

3.1 Deployment Using Tomcat

Fusion Node must be run within a Java Servlet Container. Metadata Technology recommends using Apache Tomcat as the Java Servlet Container.

Fusion Node consists of a single .war file called *FusionNode.war*. This file needs to be copied into the directory: *<TOMCAT_HOME>/webapps* then the Tomcat server should be started. As the Tomcat application server starts, the contents of the Fusion Registry war file will be unpacked into the directory:

> *<TOMCAT_HOME>/webapps/ FusionNode*

Please check the Tomcat log files to ensure that Fusion Node has deployed correctly. Once it has then you may navigate to the URL:

> *http://[server]:[port]/ FusionNode /*

The values for server and port must be replaced with the IP address and port number that the web application server is running on. For example, if the web browser is running on the same machine as the web application server and the Apache Tomcat has not had its default port settings modified, then the following address can be used:

> *http://localhost:8080/ FusionNode /*

3.2 Memory Requirements

The node memory requirements depend on the volumes of data to be held on the node. A minimum of 2Gb memory should be allocated. An extra 100Mb being allocated for every 2-3million observations should be sufficient.

3.3 Configuring Tomcat Memory – From File System

It is important to override the default Tomcat server memory settings as the default Tomcat settings will not be adequate to run the Fusion Node. Unless you are running Tomcat as a Windows service (see next section) overriding the memory settings can be achieved by placing a *setenv.bat* (Windows) or *setenv.sh* (Unix) file into the Apache Tomcat *bin* folder. The recommended minimum settings are:

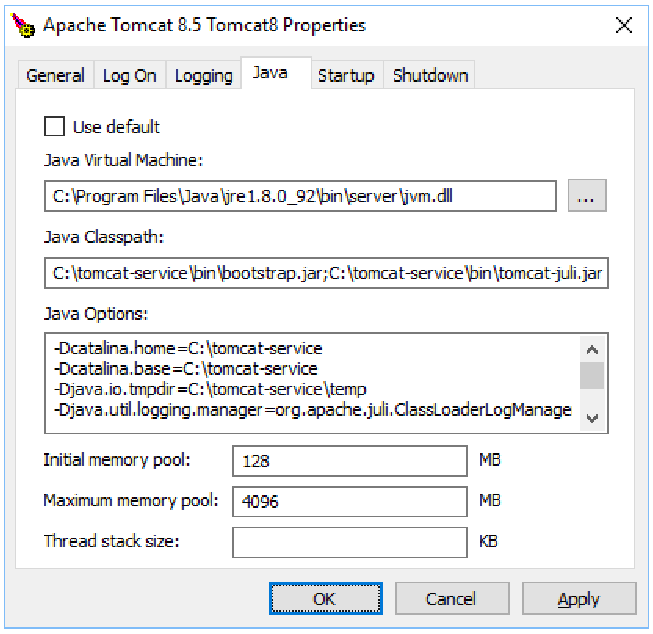
- **2Gb Heap Memory**
- **256M PermGen Space**

The Fusion Node distribution contains a *setenv.bat* (Windows) and *setenv.sh* (Unix) with the recommended minimum settings configured. These files can be copied (and optionally modified) to the Tomcat/bin folder before starting the Tomcat instance.

Version 1.8 and later versions of Java removed the concept of PermGen Space. If you are using a Java runtime of version 1.8 and start the Fusion Node with the PermGen setting, a warning will be issued, but this will not stop the Node from starting successfully.

3.4 Configuring Tomcat Memory – From Service

When Tomcat is running as a service the Apache Configuration window must be used to configure memory settings. This window has a tab labelled "Java". From this tab the appropriate settings can be applied.


Apache Tomcat 8.5 Tomcat8 Properties

☐ Use default

Java Virtual Machine: C:\Program Files\Java\jre1.8.0_92\bin\server\jvm.dll

Java Classpath: C:\tomcat-service\bin\bootstrap.jar;C:\tomcat-service\bin\tomcat-juli.jar

Java Options:

```
-Dcatalina.home=C:\tomcat-service
-Dcatalina.base=C:\tomcat-service
-Djava.io.tmpdir=C:\tomcat-service\temp
-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManage
```

Initial memory pool: 128 MB

Maximum memory pool: 4096 MB

Thread stack size: KB

By default, the Initial memory pool Maximum Memory Pool is set to 128Mb and the Maximum memory pool is set to 256Mb. This Maximum Memory Pool will almost certainly not be sufficient to run Fusion Node so it is recommended to increase this value to at least 2048 Mb. These two settings are the equivalent of settings –Xms and –Xmx in the setenv.bat file.

3.5 Permissions

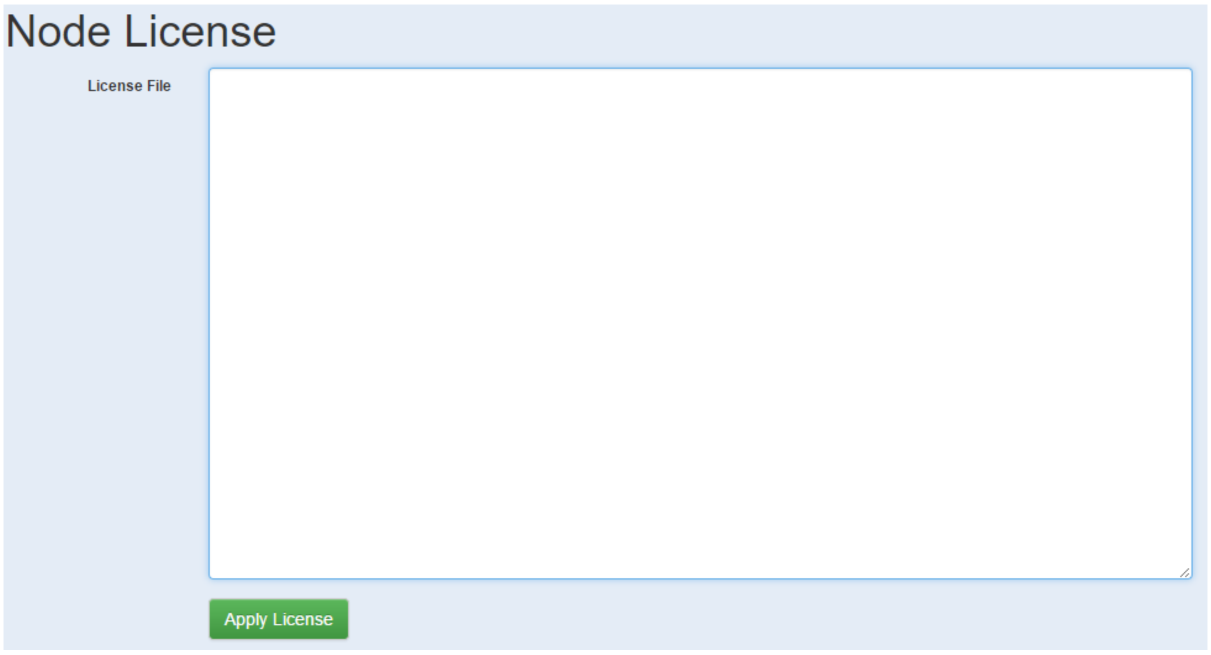
The node must have permission to both read and write to its own WEB-INF folder, as this is where the node file system is managed containing the node structures and data files. The WEB-INF folder is located under tomcat/webapps/FusionNode.

4 Configuring the Node

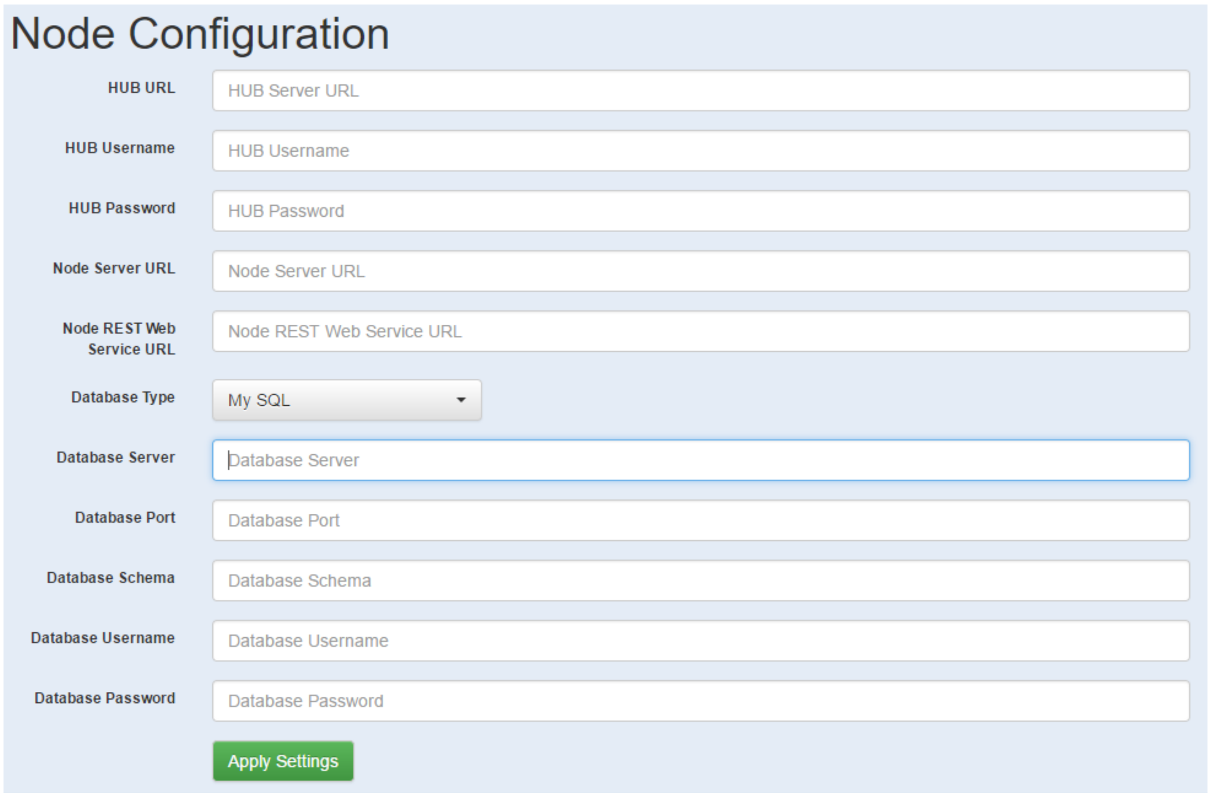
After launching Tomcat use the web browser to open the Node home page

http(s)://[server]:[port]/ FusionNode

When the node is first launched, the Node License page will be displayed. The license file should be copied and pasted into the text area followed by clicking the Apply License button.



After successfully applying the license, the Node Configuration page will be displayed, node configuration must be completed before the node can be used.



HUB URL	HUB Server URL
HUB Username	HUB Username
HUB Password	HUB Password
Node Server URL	Node Server URL
Node REST Web Service URL	Node REST Web Service URL
Database Type	My SQL
Database Server	Database Server
Database Port	Database Port
Database Schema	Database Schema
Database Username	Database Username
Database Password	Database Password

Figure 2 showing the node configuration page

The Node Configuration requires the following settings, note all settings are encrypted in the node server using an asymmetric cryptography algorithm, and therefore security settings are kept safe.

Setting	Definition
HUB URL	This is the URL of the Fusion Registry HUB For example: https://sdmxcentral.imf.org/ The node will establish a connection with the Hub and use this service to obtain its metadata. The node will periodically poll the hub to check for updates to metadata. The node will use this service for submitting data registrations each time data in the node is updated.
Hub Username	This is the username the node will use to authenticate itself with the Hub for the process of data registration. This should be the username of the data provider.
Hub Password	This is the password the node will use to authenticate itself with the Hub for the process of data registration. This should be the password of the data provider.
Node Server URL	This is the URL that the node is installed on. This URL is only used internally in the node for the purpose of redirection.
Node REST Web Service URL	This is the URL of the nodes public SDMX web service. This is required by the node to report to the Hub on data registration, and must resolve to the web service of the node. The node web service URL is *http(s)://[server]:[port]/ FusionNode/ws/public/sdmxapi/rest* If using a Proxy server such as Apache (recommended) to route public traffic then this URL must be the public facing URL, as the Hub must be able to communicate with it.
Database Settings	Only required if using a local database for data reporting

On applying these settings, the node will communicate the license server for verification. The license server will either by the Hub or the Metadata Technology license server, depending on which application created the license.

On successful install, the node will obtain metadata from the hub, and create it's folder structure. The node will then redirect to the home page.

Node settings can be redefined at any time by clicking the Settings button which is located in the header, at the top right, of the home page.

5 Loading Data from File System

5.1 Node Folder Structure

The node file structure is created under the folder

<Tomcat Home>/webapps/[FusionNode]/WEB-INF/node

The folder structure is based on the Provision Agreement defined in the Hub. A Provision Agreement is an item of metadata which links a Data Provider to a Dataflow. A data reporter will always report data against the provision agreement, and from this the Hub is able to identify who the Data Provider is and which Dataflow they are reporting data for.

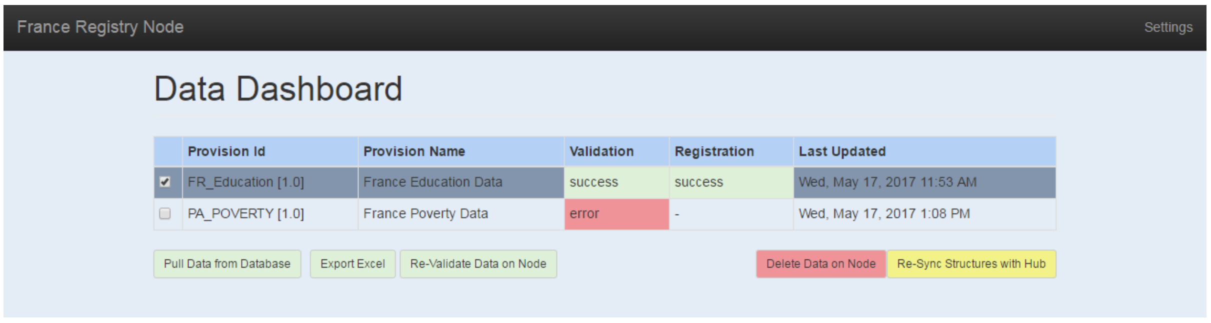
Folders are created for each provision agreement that the node's data reporter has Provision Agreements for, as defined in the Hub. The folder structure is as follows:

Provision Agreement Owning Agency Id → Provision Agreement Id → Provision Agreement Version

5.2 Loading New Data

To load new data, copy the data file into the leaf folder (Provision Agreement Version). The file can be in any supported node format.

The node will detect the presence of a new file and read it for validation. If it passes the validation stage, it will be loaded into the node's data store, and the data will be available from the node's web service. When this process is complete, the node will register the web service with the Hub, against the relevant Provision Agreement.



	Provision Id	Provision Name	Validation	Registration	Last Updated
✓	FR_Education [1.0]	France Education Data	success	success	Wed, May 17, 2017 11:53 AM
	PA_POVERTY [1.0]	France Poverty Data	error	-	Wed, May 17, 2017 1:08 PM

5.3 Updating Data

To update data from the file system, place the new data file in the folder. If the data file is an update to the existing data file (i.e it contains more data) then it is possible to keep both the data files in the folder. The node will read all data files in the provision agreement folder, and consolidate them into a single dataset. If there are no duplications and the files are valid, then all data sets will be loaded into the node store.

If the date modified date of the new data file is later than any other file in the directory, then the node will automatically revalidate all datafiles for import. Otherwise files marked as .fds and .err should be deleted to force a revalidation. Alternatively, the web interface can be used to force a revalidation, by clicking on the 'Re-Validate Data on Node' button.

5.4 Delete

Data can be deleted by either using the web interface to delete the data on the node, for the selected provision agreement. Alternatively if all the files in the provision agreement folder are deleted, the node will update its definitions. In both cases, the node will register the data deletion with the Hub by de-registering its data web service for the provision agreement.

6 Loading Data from Database

6.1 Overview

If a database connection has been defined, it will be possible to pull data from the database for the selected provision agreement. The 'Pull Data from Database' button on the user interface, deletes all existing data in the node, for the provision agreement, and pulls the data from the database into the node's folder system for import.

6.2 Table or View Naming Convention

The database table or view must follow the naming convention of:

[Provision Agency Id]_[Provision Id]_[Provision Version]

The table name must be in upper case, and any '.' or '-' characters must be replaced with underscores '_'. For example:

Provision Agency	Provision Id	Provision Version	Expected Table/View Name
WB	POVERTY	1.0	WB_POVERY_1_0
ONS.DEPT1	GDP-DATA	2.1.1	ONS_DEPT1_GDP_DATA_2_1_1

The Provision Agreement agency, id, and version can be determined by looking at the folder structure in the node, or by visiting the Hub Fusion Registry.

6.3 Column Naming Convention

The database table or view is expected to contain a column for each of the Components of the Data Structure Definition that the Provision Agreement conforms to. For example:

DSD Component ID	Component Type	Column Id
FREQ	Dimension	FREQ
REF_AREA	Dimension	REF_AREA
SERIES	Dimension	SERIES
Series_Title	Series Attribute	SERIES_TITLE
OBS_VALUE	Observation Attribute	OBS_VALUE
OBS_CONF	Observation Attribute	OBS_CONF
OBS_VALUE	Observation Value	OBS_VALUE
TIME_PERIOD	Time Dimension	TIME_PERIOD

The value in the TIME_PERIOD column must represent the same date, formatted according to the frequency. The following table shows an example of the observation frequency vs the expected date format.

Frequency	Format	Example
Annual	YYYY	2010
Daily	YYYY-MM-DD	2010-01-01
Date Time	YYYY-MM-DD-Thh:mm:ss	2010-01T20:22:00
Monthly	YYYY-MM	2010-01
Quarterly	YYYY-Qn	2010-Q1
Semester	YYYY-Sn	2010-S1
Trimester	YYYY-Tn	2010-T1

7 Data Reporting Errors

A dataset can result in error if either the database pull/data validation fails on the node, or if the data registration process failed. In both cases, clicking on the table cell with label 'error' will display the error reason. If the data validation failed, the data can be deleted on the node. If data registration failed, check the node settings correctly specify the node data web service, and then re-register.

8 Data view and web services

8.1 SDMX Web Services

The data held in the node is made available via SDMX compliant web services. The web service entry point for the node is available at:

http(s)://[server]:[port]/ FusionNode/ws/public/sdmxapi/rest

Node web service documentation is provided separately

8.2 Excel Export

Data can be exported from the node in Excel format, note for larger datasets this may be an expensive operation. To export data in excel, use the node web service to select the provision agreement, and then click the 'Export Excel' button.

8.3 Building a Custom Web User Interface

In addition the IMJS JavaScript library can be used to build upon the node web services in order to build a web user interface for data exploring, visualisation, and export.

IMJS documentation is available on the Metadata Technology website

http://www.metadatatechnology.com/FusionRegistry9/IMJS/

8.4 Performance

In terms of performance, the node web services can support over 200 requests per second, depending on the size of data query. The node supports query piggybacking, which prevents the same query running concurrently multiple times on the node. In addition the node provides a file system caching solution to ensure peak performance is met.

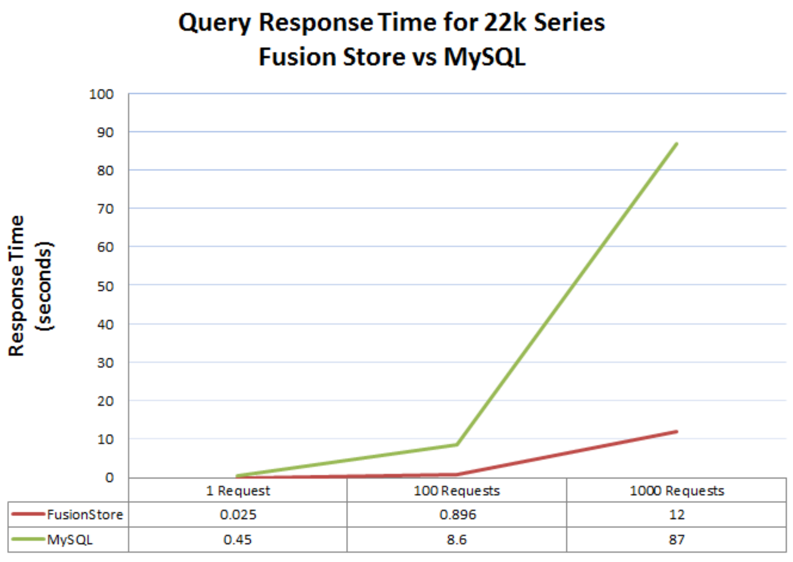
Benchmarking the node's internal database (Fusion Store) against a relational database using an optimised star schema design, for the same dataset, the node provides a much improved service, even with all caching disabled.

Dataset Size:	500Mb SDMX file with 2.1 Million Observations
Tomcat Max Memory:	2Gb
Data Load:	58 Seconds

Query 22k Observations:

Num Requests	Concurrency	Time (s)	Requests Per Second
1	1	0.025	40
100	10	0.872	114
100	50	0.896	111
1000	50	12	82

When benchmarked against the same query for a MySQL instance (on a desktop PC) the graph looks as follows. **Note:** All caching is disabled:



	1 Request	100 Requests	1000 Requests
🟥 FusionStore	0.025	0.896	12
🟩 MySQL	0.45	8.6	87

9 Resync structures with Hub

The node will periodically poll the hub service to ensure it is up to date, if any failures occur the node should manage these by automatically re-syncing with the hub. However if for any reason the structures in the node are not in sync with the hub, they can be resynced by clicking the 'Re-Sync Structures with Hub' button in the web interface.

Alternatively the structure file can be deleted to force a re-sync. The structure file can be found in the root folder of the node file system.

<Tomcat Home>/webapps/[FusionNode]/WEB-INF/node