

Execution of SPARQL Query using Apache Jena Fuseki Server in AISHE domain

Prof. Ronak Panchal¹ AND Dr. Priya R. Swaminarayan²

¹Vidyabharti Trust College of BCA, Bardoli, VNSGU, Surat

²Director of Parul Institute of Computer Applications, Parul University, Vadodara

Abstract:- To portray the status of higher education in the country, Ministry of Human Resource Development has endeavoured to conduct an annual web-based All India Survey on Higher Education (AISHE) since 2010-11. NIC – National Informatics Centre has developed AISHE portal with few limitations like no online system exists, it's neither a Mobile Application nor a Web Application, and retrieval of data doesn't support Semantic retrieval, so we proposed Semantic Web based AISHE system (AISHE Ontology) through which we can take advantages of Semantic Web Technology in AISHE system. The development of AISHE Ontology can be done by various Ontology Development Tools like Protégé, etc. but we designed AISHE Ontology using Protégé as it is the most popular Ontology Development tool as per survey conducted by JornKardoso. The major drawback of Protégé is we cannot enter triple using SPARQL in Protégé or we cannot manipulate triple directly using protégé tool and it can be overcome by using Apache Jena Fuseki Server.

Apache Jena is a free and open source Java framework for building Semantic Web and Linked Data applications. Fuseki is one of the components of Apache Jena which expose your triples as a SPARQL end-point accessible over HTTP. Fuseki provides REST-style interaction with your RDF data. In this paper, we have covered about Apache Jena Fuseki Server which includes benefits of Jena Fuseki Server, installation of Apache Jena Fuseki, how to upload dataset in a Server, how to write and implement SPARQL query including insertion and deletion. We used AISHE data as a domain for executing SPARQL query.

Keywords: Semantic Web, SPARQL, Apache Jena Fuseki, AISHE,

Introduction to AISHE (AISHE, 2010-11)

To portray the status of higher education in the country, Ministry of Human Resource Development has endeavoured to conduct an annual web-based All India Survey on Higher Education (AISHE) since 2010-11. The survey covers all the Institutions in the country engaged in imparting of higher education.

Data (AISHE, 2010-11) is being collected on several parameters such as teachers, student enrolment, programmes, examination results, education finance, and infrastructure. Indicators of educational development such as Institution Density, Gross Enrolment Ratio, Pupil-teacher ratio, Gender Parity Index, Per Student Expenditure will also be calculated from the data collected through AISHE. These are useful in making informed policy decisions and research for development of education sector.

NIC – National Informatics Center has developed AISHE portal with few limitations like no online system exists, it's neither a Mobile Application nor a Web Application, and retrieval of data doesn't support Semantic retrieval, so we proposed Semantic Web based AISHE system (AISHE Ontology) through which we can take advantages of Semantic Web Technology in AISHE system.

The development of AISHE Ontology can be done by various Ontology Development Tools like Protégé, SWOOP, OilED, WebODE, Ontolingua, OntoTrack etc. but we designed AISHE Ontology using Protégé as it is the most popular Ontology Development tool as per survey conducted by M. Rahamatullah Khondoker, Parul Muller [M. Rahamatullah Khondoker, Paul Mueller (2010), "Comparing Ontology Development Tools Based on an Online Survey", Proceedings of the World Congress on Engineering 2010 Vol 1 WCE 2010, June 30 – July 2, 2010, London, UK] and Jorge Cardoso [Jorge Cardoso (2007), "The Semantic Web Vision: Where are We?", IEEE Intelligent Systems, September/October 2007, pp.22-26]. The major drawback of Protégé is we cannot enter triple using SPARQL in Protégé or we cannot manipulate triple directly using protégé tool and it can be overcome by using Apache Jena Fuseki Server.

Introduction to Apache Jena Fuseki server

Apache Jena is a free and open source Java framework for building Semantic Web and Linked Data applications. It supports 1) RDF – Resource Description Framework: a) RDF API – Interact with the core API to create and read RDF graphs. Serialise your triples using popular formats such as RDF/CML or Turtle b) ARQ (SPARQL) – Query your RDF data using ARQ, a SPARQL 1.1 compliant engine. ARQ supports remote federated queries and free text search 2) Triple Store: a) TDB – Persist your data using TDB, a native high performance triple store. TDB supports the full range of Jena APIs b) Fuseki – Expose your triples as a SPARQL end-point accessible over HTTP. Fuseki provides REST-style interaction with your RDF data 3) OWL – Web Ontology Language: a) Ontology API – work with models, RDFS and

the OWL to add extra semantics to your RDF data. b) Inference API – reason over your data to expand and check the content of your triple store. Configure your own inference rules or use the build-in OWL and RDFS reasoners

In this paper, we have covered about Apache Jena Fuseki Server which includes installation of Apache Jena Fuseki, how to upload dataset in a Server, how to write and implement SPARQL query including insertion and deletion.

Apache Jena Fuseki (The_Apache_Software_Foundation, 2011–2017) is a SPARQL server. It can run as an operating system service, as a Java web application (WAR file), and as a standalone server. It provides security and has a user interface for server monitoring and administration. It provides the SPARQL 1.1 protocols for query and update as well as the SPARQL Graph Store protocol.

Fuseki (The_Apache_Software_Foundation, 2011–2017) is tightly integrated with TDB to provide a robust, transactional persistent storage layer, and incorporates Jena text query and Jena spatial query. It can be used to provide the protocol engine for other RDF query and storage systems.

Why do we need Apache Jena Fuseki Server?

- Apache Jena Fuseki Server need for SPARQL server which can present RDF data and answer SPARQL queries over HTTP.
- We cannot enter triple using SPARQL in Protégé or we cannot manipulate triple directly using protégé tool.
- There are following some alternatives but Apache Jena Fuseki is better way:
 - RDF4J (Jena vs. RDF4J Comparison.htm, 2016) is a Java framework for processing RDF data, supporting both memory-based and a disk-based storage.
 - CumulusRDF (Fuseki or cumulusRDF - Google Groups.htm, 2014) is an RDF store that uses Apache Cassandra as underlying storage. Apache Cassandra is a good and proven NoSql (specifically column-oriented) storage for managing huge volume of data.

What are the benefits of Jena Fuseki Server in Semantic Web?

- To execute federated queries (Semantic_Web_-_SPARQL[1].pdf, 2014) we need a SPARQL endpoint that “knows” how to distribute queries and join the results, the so called “mediator endpoint”. Fuseki server (Jena ARQ) can be used as a mediator endpoint.
- If none of the servers or the sparql library support federated queries, any Fuseki server can be used as a mediator.

Framework Architecture (architecture.html, 2017) with Fuseki Server

Jena (architecture.html, 2017) stores information as RDF triples in directed graphs, and allows your code to add, remove, manipulate, store and publish that information. We tend to think of Jena as a number of major subsystems with clearly defined interfaces between them.

Fuseki is a data publishing server, which can present, and update, RDF models over the web using SPARQL and HTTP.

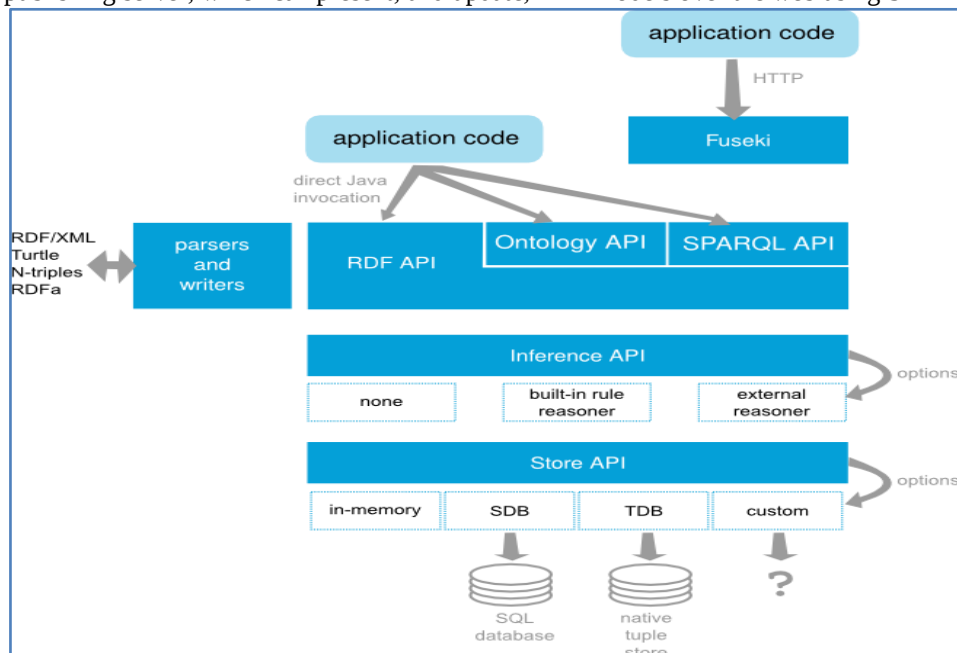


Figure 1: Framework Architecture with Fuseki Server (architecture.html, 2017)

Installation of Apache Jena Fuseki Server:

First of all download apache jena fuseki server for windows from given link <https://jena.apache.org/download/index.cgi> then extract it. To make fuseki folder in c:\ drive and paste in it. Now you start server batch file in it. Now run apache jena fuseki server in browser as shown in figure-2. Then create dataset like AISHE as shown in below figure.

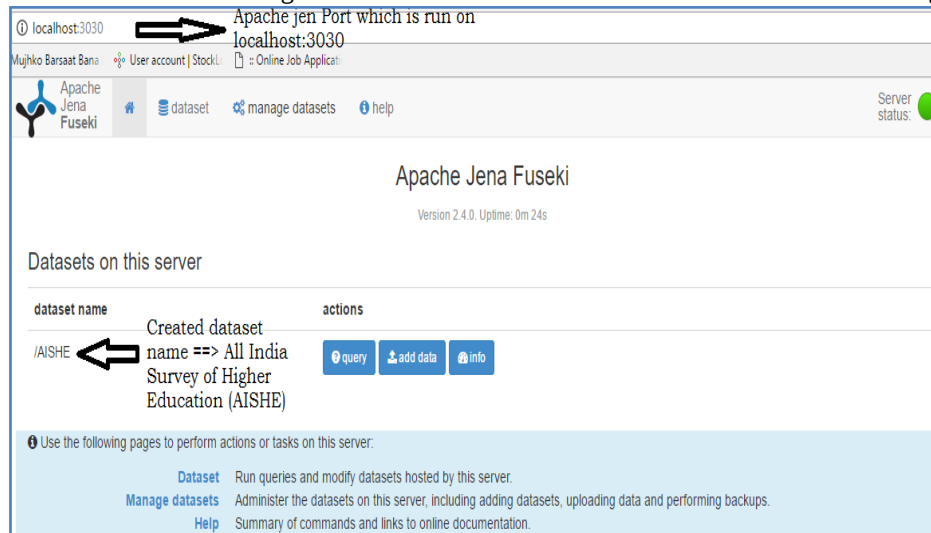


Figure-2: Run Apache Jena Fuseki Server in browser & Add Dataset

How to manage Dataset in Apache Jena Fuseki?

Step 1: To create dataset like AISHE.

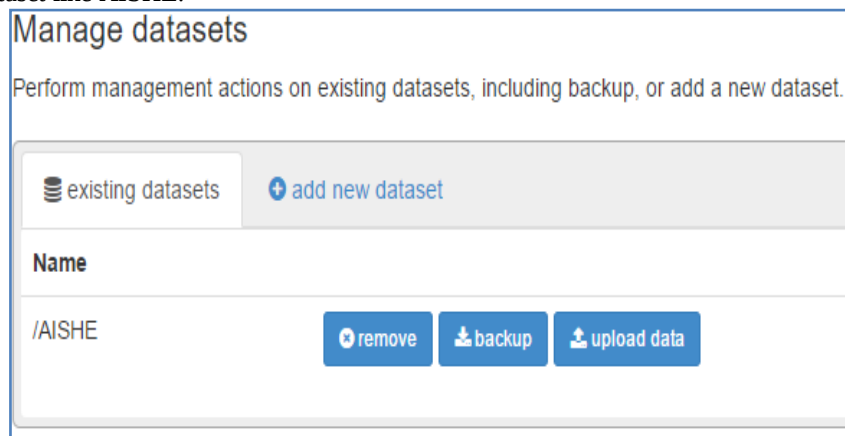


Figure-3: Creation of Dataset

Step 2: Then select any RDF format file like .owl,.rdf,.nq etc. and upload it. In this system, we have created AISHE Ontology using Protégé with the extension .owl which includes classes, object properties, datatype properties and instances.

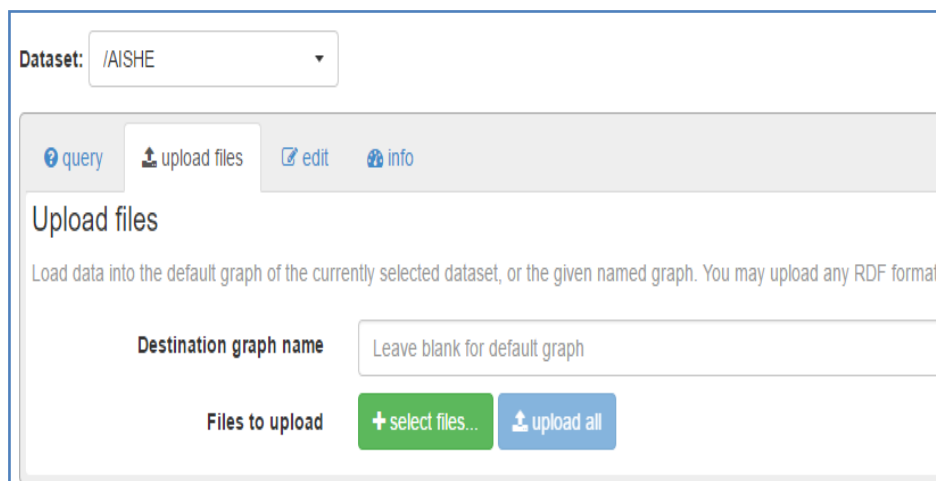
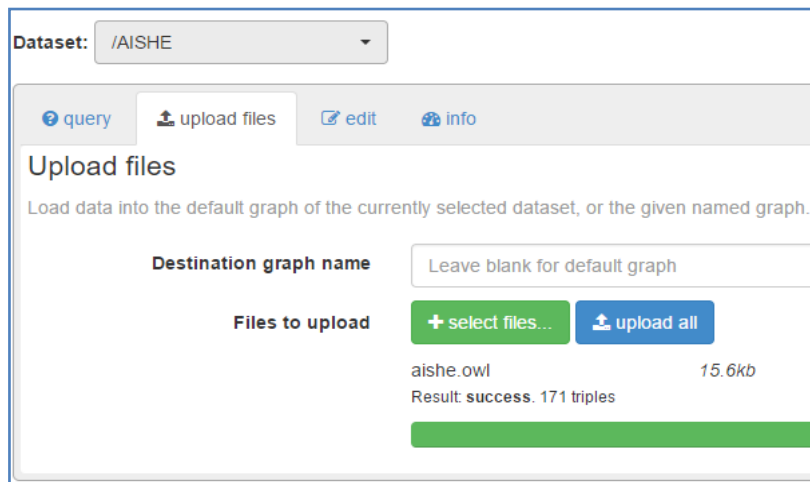


Figure-4: Loading data into the dataset

Step 3: Then system upload file on apache jena fuseki server with appropriate message and no. of triples.



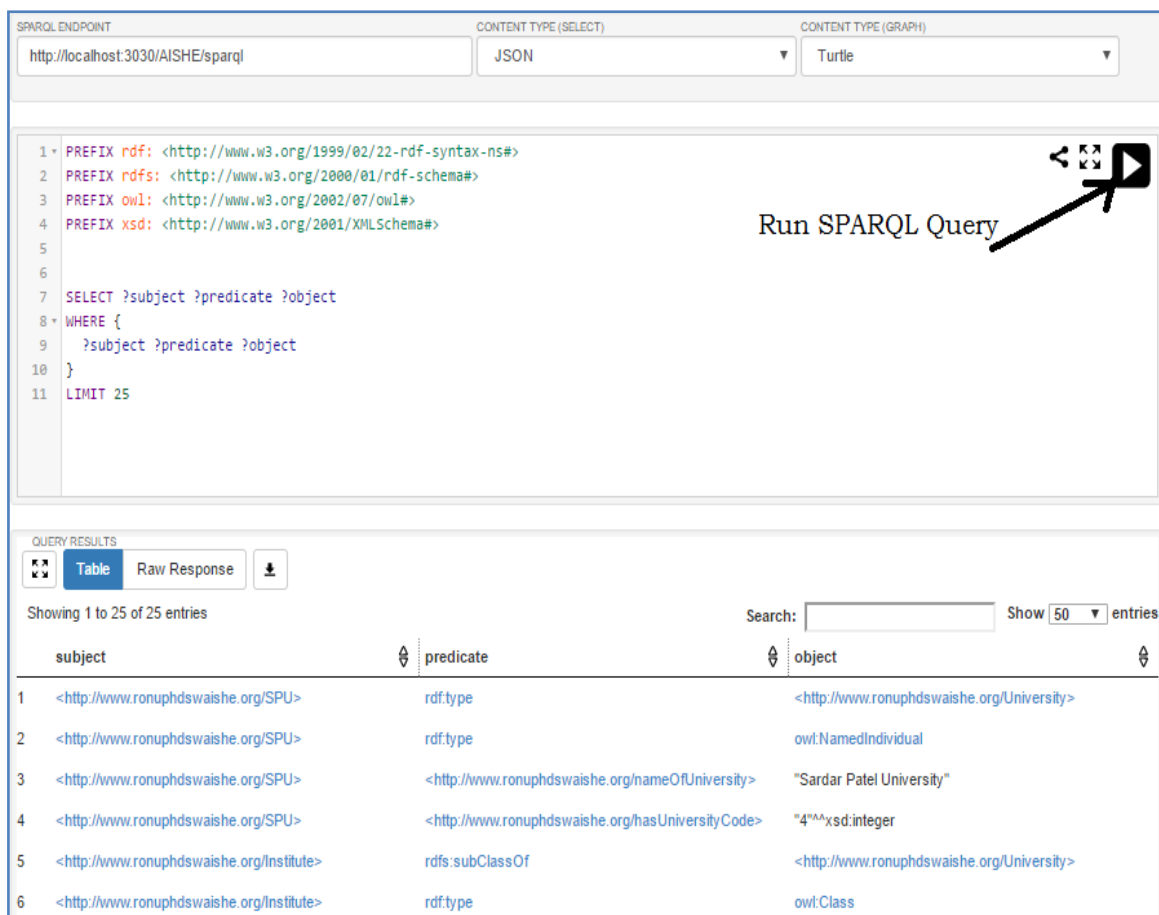
How to run SPARQL in Apache Jena Fuseki?(SPARQL, 2011)

After uploading AISHE Ontology (aishe.owl) file in Apache Jena Fuseki, we can directly execute SPARQL query without using Protégé Tool and Jena Fuseki will not only display the result of SPARQL query but we can edit and delete record directly from Jena Fuseki.

In this section, we have executed some sample SPARQL queries in Apache Jena Fuseki. First of all, set SPARQL endpoint like <http://localhost:3030/AISHE/sparql> or <http://localhost:3030/AISHE/query>

There are few examples of SPARQL queries in Apache Jena Fuseki. To set SPARQL endpoint like <http://localhost:3030/AISHE/sparql> or <http://localhost:3030/AISHE/query> (here AISHE is dataset), when we execute basic SPARQL query in apache jena fuseki server,

1. To find basic subject, predicate and object from AISHE dataset



Run SPARQL Query

	subject	predicate	object
1	<http://www.ronuphdswaishe.org/SPU>	rdf:type	<http://www.ronuphdswaishe.org/University>
2	<http://www.ronuphdswaishe.org/SPU>	rdf:type	owl:NamedIndividual
3	<http://www.ronuphdswaishe.org/SPU>	<http://www.ronuphdswaishe.org/nameOfUniversity>	"Sardar Patel University"
4	<http://www.ronuphdswaishe.org/SPU>	<http://www.ronuphdswaishe.org/hasUniversityCode>	"4"^^xsd:integer
5	<http://www.ronuphdswaishe.org/Institute>	rdfs:subClassOf	<http://www.ronuphdswaishe.org/University>
6	<http://www.ronuphdswaishe.org/Institute>	rdf:type	owl:Class

2. Display statutory wise approved university

SPARQL ENDPOINT

CONTENT TYPE (SELECT)

JSON

CONTENT TYPE (GRAPH)

Turtle

```

1 PREFIX my: <http://www.ronuphdswaishe.org/>
2 SELECT ?collegename ?uniname
3 WHERE
4 {
5   ?collegename my:isAffiliatedTo ?uniname .
6 }
7
8
        
```

QUERY RESULTS

Table

Raw Response

Download

Showing 1 to 5 of 5 entries

Search:

	collegename	uniname
1	my:VPSscience	my:SPU
2	my:VTCBBA	my:VNSGU

3. Display universities wise college names

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX my: <http://www.ronuphdswaishe.org/>
SELECT ?collegename ?uniname
WHERE
{
  ?collegename my:isAffiliatedTo ?uniname .
}
    
```

4. Display College or Institute only

SPARQL ENDPOINT

CONTENT TYPE (SELECT)

JSON

CONTENT TYPE (GRAPH)

Turtle

```

1 PREFIX my: <http://www.ronuphdswaishe.org/>
2 SELECT ?c WHERE
3 {
4   { ?c a my:College . }
5   union
6   { ?c a my:Institute . }
7 }
8
        
```

QUERY RESULTS

Table Raw Response

Showing 1 to 5 of 5 entries

Show 50 entries

Search:

c
1 my:VTCBBA
2 my:VPScience
3 my:VTCBCA
4 my:VTCMCA
5 my:SNPIT

Showing 1 to 5 of 5 entries

5. Count Colleges in University

SPARQL ENDPOINT:

CONTENT TYPE (SELECT):

CONTENT TYPE (GRAPH):

```

1 PREFIX my: <http://www.ronuphdswaishe.org/>
2 SELECT ?uname (count(?collegename) AS ?count)
3 {
4   ?c my:isAffiliatedTo ?u .
5   ?c my:nameOfCollege ?collegename .
6   ?u my:nameOfUniversity ?uname .
7 }
8 Group By ?uname
9

```

QUERY RESULTS

Table Raw Response

Showing 1 to 3 of 3 entries

Search:

Show 50 entries

uname	count
1 "Gujarat Technology University"	"1"^^xsd:integer
2 "Sardar Patel University"	"1"^^xsd:integer

6. Display Code wise College name list

SPARQL ENDPOINT: CONTENT TYPE (SELECT): CONTENT TYPE (GRAPH):

```
1 PREFIX my: <http://www.ronuphdswaishe.org/>
2 SELECT ?c ?collegename ?uname WHERE
3 {
4   ?c my:isAffiliatedTo ?uname .
5   ?c my:nameOfCollege ?collegename .
6   ?uname my:hasUniversityCode 6 .
7 }
```

QUERY RESULTS

Table Raw Response

Showing 1 to 2 of 2 entries

	collegename	uname
my.VTCBBA	"Vidyabharti Trust College of BBA"	my.VNSGU
my.VTCBCA	"Vidyabharti Trust College of BCA"	my.VNSGU

How to run update (insert and delete) SPARQL query in Apache Jena Fuseki?

Here first query is following, we have change name of university like Auro University now we change ABC University Instead of Auro University. First we have search it from file.

QUERY RESULTS

Table Raw Response

Showing 1 to 4 of 4 entries (filtered from 171 total entries)

Search Auro University

Search: Show 50 entries

subject	predicate	object
<http://www.ronuphdswaishe.org/AuroUni>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://www.ronuphdswaishe.org/University>
<http://www.ronuphdswaishe.org/AuroUni>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://www.w3.org/2002/07/owl#NamedIndividual>
<http://www.ronuphdswaishe.org/AuroUni>	<http://www.ronuphdswaishe.org/nameOfUniversity>	"Auro University"
<http://www.ronuphdswaishe.org/AuroUni>	<http://www.ronuphdswaishe.org/hasUniversityCode>	"1"^^xsd:integer

Showing 1 to 4 of 4 entries (filtered from 171 total entries)

Now you can check SPARQL Endpoint with <http://localhost:3030/AISHE/update> instead of <http://localhost:3030/AISHE/query> or <http://localhost:3030/AISHE/sparql> we have delete record from file through sparql update.

SPARQL ENDPOINT: CONTENT TYPE (SELECT):

```
1 PREFIX my: <http://www.ronuphdswaishe.org/>
2 DELETE
3 where
4 {
5   <http://www.ronuphdswaishe.org/University> my:nameOfUniversity "Auro University"
6 }
7
```

here is update query of SPARQL Update (SPARQL_1_1_Update, 2013).

```
PREFIX my: <http://www.ronuphdswaishe.org/>
DELETE
where
{
<http://www.ronuphdswaishe.org/AuroUni>my:nameOfUniversity "Auro University" .
};
Insert data
{
<http://www.ronuphdswaishe.org/AuroUni>my:nameOfUniversity "ABC University" .
}
```

Now you can again check update record through simple query with the SPARQL ENDPOINT

<http://localhost:3030/AISHE/query>

QUERY RESULTS

Table

Raw Response

Showing 1 to 4 of 4 entries (filtered from 171 total entries)

Search:

Auro

Show

50

entries

subject	predicate	object
1 <http://www.ronuphdswaishe.org/AuroUni>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://www.ronuphdswaishe.org/University>
2 <http://www.ronuphdswaishe.org/AuroUni>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://www.w3.org/2002/07/owl#NamedIndividual>
3 <http://www.ronuphdswaishe.org/AuroUni>	<http://www.ronuphdswaishe.org/nameOfUniversity>	"ABC University"
4 <http://www.ronuphdswaishe.org/AuroUni>	<http://www.ronuphdswaishe.org/hasUniversityCode>	"1"^^xsd:integer

Showing 1 to 4 of 4 entries (filtered from 171 total entries)

Second query is following;Delete “ABC University” based triple from owl file using Apache Jena Fuseki Server.

SPARQL ENDPOINT		CONTENT TYPE (SELECT)	
http://localhost:3030/AISHE/update		JSON	
<pre>1 PREFIX my: <http://www.ronuphdswaishe.org/> 2 DELETE 3 where 4 { 5 <http://www.ronuphdswaishe.org/University> my:nameOfUniversity "ABC University" . 6 } 7</pre>			

Delete ABC University record from owl file

Conclusion

In this paper, we have used Protégé as Ontology development tools but the most important weakness of Protégé is we cannot enter triple using SPARQL in Protégé or we cannot manipulate triple directly using protégé tool. All these drawbacks can be overcome by using Apache Jena Fuseki Server. With the help of Jena Fuseki Server, we can add new triple and modify the existing triple; and Ontology file can be easily converted into other format as per our requirement. This paper will be helpful to Ontology developer as we have covered benefits of Jena Fuseki Server, installation of Apache Jena Fuseki, how to upload dataset in a Server, how to write and implement SPARQL query including insertion and deletion. We used AISHE data as a domain for executing SPARQL query.

Reference

<https://jena.apache.org/documentation/fuseki2/>
<https://jena.apache.org/download/index.cgi>
<http://aishe.nic.in/aishe/home>
<https://jena.apache.org/tutorials/sparql.html>

Bibliography

- [1] AISHE. (2010-11). *home.htm*. Retrieved may 12, 2017, from <http://aishe.nic.in>: <http://aishe.nic.in/aishe/home>
- [2] *Apache Jena - Getting started with Apache Jena.htm*. (2017). Retrieved 6 19, 2017, from <https://jena.apache.org>: https://jena.apache.org/getting_started/
- [3] *architecture.html*. (2017). Retrieved 6 19, 2017, from https://jena.apache.org/about_jena/: https://jena.apache.org/about_jena/architecture.html
- [4] *Fuseki or cumulusRDF - Google Groups.htm*. (2014, 4 25). Retrieved 6 19, 2017, from <https://groups.google.com/forum>: https://groups.google.com/forum/#!topic/cumulusrdf-list/x0K5Cg_bsdg
- [5] *Jena vs. RDF4J Comparison.htm*. (2016). Retrieved 6 19, 2017, from <https://db-engines.com/en/system/>: <https://db-engines.com/en/system/Jena%3BRDF4J>
- [6] *Semantic_Web_-_SPARQL[1].pdf*. (2014). Retrieved 6 19, 2017, from <https://www.fer.unizg.hr/>: [https://www.fer.unizg.hr/_download/repository/11._Semantic_Web_-_SPARQL\[1\].pdf](https://www.fer.unizg.hr/_download/repository/11._Semantic_Web_-_SPARQL[1].pdf)
- [7] SPARQL. (2011). *sparql.html*. Retrieved may 12, 2017, from <https://jena.apache.org/tutorials/>: <https://jena.apache.org/tutorials/sparql.html>
- [8] SPARQL_1_1_Update. (2013). *SPARQL 1.1 Update.htm*. Retrieved may 12, 2017, from <https://www.w3.org/>: <https://www.w3.org/TR/sparql11-update/>
- [9] The_Apache_Software_Foundation. (2011–2017). *Apache Jena - Apache Jena Fuseki.html*. Retrieved from <https://jena.apache.org>: <https://jena.apache.org/documentation/fuseki2/>