

# Using Rdf4j libraries and workbench application to perform SPARQL federated queries

## Introduction

This quick guide explains how to use the Rdf4j library and workbench to perform federated queries. This functionality is available by the federation with FedX , as explained [here](#) .

## Workbench installation

The Rdf4j workbench is a simple webapp that provides a graphical interface to define single external SPARQL endpoints and also federate them into a unique federated endpoint, and run sparql queries. It can be easily deployed in a Tomcat environment as described [here](#) .

Once deployed, the environment is accessible through <http://localhost:8080/rdf4j-workbench/> .



RDF4J Server

Repositories

New repository

Delete repository

Explore

Summary

Namespaces

Contexts

Types

Explore

Query

Saved Queries

Export

Modify

SPARQL Update

Add

Remove

Clear

System

Information

## List of Repositories

		Id	Description	Location
✓	✓	<a href="#">intermine</a>	Intermine endpoint proxy	<a href="http://localhost:8080/rdf4j-server/repositories/intermine">http://localhost:8080/rdf4j-server/repositories/intermine</a>
✓	✓	<a href="#">uniprot</a>	Uniprot endpoint	<a href="http://localhost:8080/rdf4j-server/repositories/uniprot">http://localhost:8080/rdf4j-server/repositories/uniprot</a>
✓	✓	<a href="#">intermine_uniprot</a>	Intermine and uniprot federated service	<a href="http://localhost:8080/rdf4j-server/repositories/intermine_uniprot">http://localhost:8080/rdf4j-server/repositories/intermine_uniprot</a>

Copyright © 2015 Eclipse RDF4J Contributors

By clicking in “New repository”, we can define a new single sparql endpoint, of type “SPARQL endpoint proxy”. The definition is simple, the only information needed is the URI of the external sparql endpoint we want to define.

Once we have defined the single endpoints, we can federate them by defining a new repository of type “Federation” in the workbench. The application allows us to select all the single repositories that will be part of the federation:

The screenshot shows the 'New Repository' dialog in the RDF4J Workbench. On the left is a sidebar with a tree view containing 'RDF4J Server', 'Repositories' (with sub-items 'New repository' and 'Delete repository'), 'Explore' (with sub-items 'Summary', 'Namespaces', 'Contexts', 'Types', 'Explore', 'Query', 'Saved Queries', 'Export'), 'Modify' (with sub-items 'SPARQL Update', 'Add', 'Remove', 'Clear'), and 'System' (with sub-item 'Information'). The main area is titled 'New Repository' and contains three input fields: 'Type' set to 'Federation', 'ID' set to 'int-uniprot-fed', and 'Title' set to 'Intermine and uniprot federated service'. Below these fields are 'Next' and 'Cancel' buttons. At the bottom right, a copyright notice reads 'Copyright © 2015 Eclipse RDF4J Contributors'.

# rdf4j / workbench

**RDF4J Server**

**Repositories**

New repositoryDelete repository

**Explore**

SummaryNamespacesContextsTypesExploreQuerySaved QueriesExport

**Modify**

SPARQL UpdateAddRemoveClear

**System**

Information

## New Repository

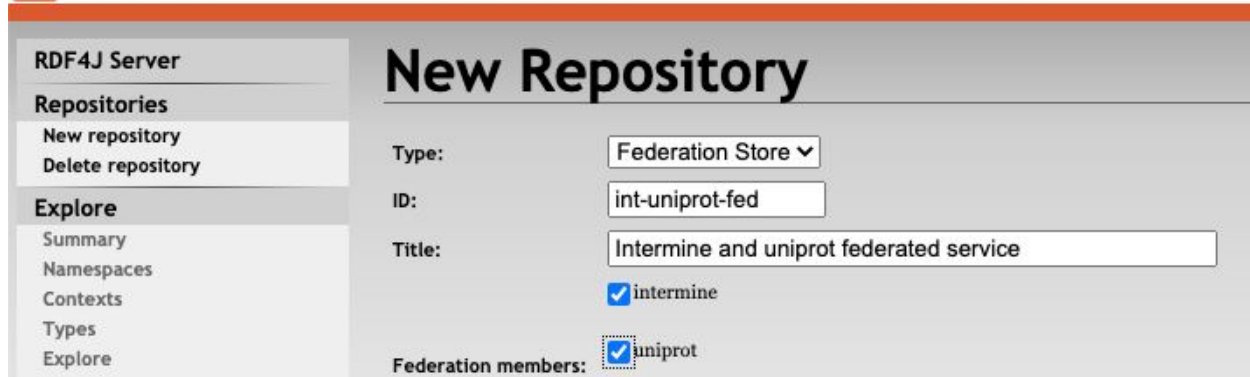
Type: Federation▼

ID: int-uniprot-fed

Title: Intermine and uniprot federated service

NextCancel

Copyright © 2015 Eclipse RDF4J Contributors



**RDF4J Server**

**Repositories**

- New repository
- Delete repository

**Explore**

- Summary
- Namespaces
- Contexts
- Types
- Explore

## New Repository

Type:

ID:

Title:

☒ intermine

Federation members: ☒ uniprot

Once defined, we can select the federated repository to use by the workbench for the SPARQL queries. The application allows us to make the selection of the reference repository in the home page, and it can be changed at every time also by clicking on the “change” link in the up right corner of the page.

## Intermine-Uniprot federated query

In this use case we have defined the intermine and uniprot repositories, where:

- [Intermine](#) SPARQL endpoint has been deployed in the context of this hackathon, in project 22, by using Jena, RDF4J, R2RML and [Ontop](#) to transform the Intermine relational database into a SPARQL one. The intermine test SPARQL endpoint is available on a localhost port;
- [Uniprot](#) is the related project public endpoint.

We have performed a simple federated query, where Protein is a resource that is present in both environments. Then, we have asked for two properties, one coming from Intermine, the other one coming from Uniprot. Rdf4j federates the query, returning in one row, for the matching Protein. Basically, allowing an user to query all of the InterMine instance and UniProt as if they were a single data warehouse.

## Query Repository

Query Language: SPARQL ▼

Query:

```
1 PREFIX up: <http://purl.uniprot.org/core/>
2 PREFIX voc: <http://intermine.org/vocabulary/>
3 SELECT
4   ?protein ?citation ?sequence WHERE {
5     ?protein a up:Protein .
6     ?protein up:citation ?citation .
7     ?protein voc:hasSequence ?sequence .
8
9     FILTER(?protein = <http://purl.uniprot.org/uniprot/Q8ILG7>)
10  }
```

Results per page: 100 ▼

Action Options: ☒ Include inferred statements ☐ Save privately (do not share)

Actions:

Clear

Execute

Save query

## Query Result (1-1 of 1)

Download format: SPARQL/CSV ▼

Download

Results per page: 100 ▼

Results offset:

Previous 100

Next 100

Show data types & language tags: ☒

Protein	Citation	Sequence
<a href="http://purl.uniprot.org/uniprot/Q8ILG7">http://purl.uniprot.org/uniprot/Q8ILG7</a>	<a href="http://purl.uniprot.org/citations/12368864">http://purl.uniprot.org/citations/12368864</a>	<a href="http://intermine.org/biotestmine/intermine_Sequence/a807cf4f6ac3889631171a9854595ff3">http://intermine.org/biotestmine/intermine_Sequence/a807cf4f6ac3889631171a9854595ff3</a>

Obviously, this is a very simple query, in order to demonstrate federation. More complex queries have to be performed in order to test the system.

## Defining a Federated SPARQL environment by Java code

Through the usage of the Rdf4j libraries it is possible to define a federated repository and perform queries via java code. Some examples can be found [here](#) (section 2.1) .

A simple main class defining a federated environment between Intermine and Uniprot, performing the SPARQL query defined above, has been released in the Biohackathon code for project 22.