# Components

The components used are:

1. Jersey JAX-RS (JSR 311 & JSR 339) Reference Implementation 2.28
2. Apache Tomcat 8.5
3. Eclipse WTP for a Gradle project
4. Gradle 5.4.1
5. MongoDB 4
6. Java 1.8.0\_211

# Architecture

Please see the application architecture below:

Source 1

RESTDataService

MongoDB

Gitter.com

HTTP

POST

Source 2

HTTP

GET

RESTQueryService (index.html)

RESTDataService (RESTDataService.java) implements the service endpoint. It has two methods, saveSource1Data() and saveSource2Data() to obtain and the save the JSON data in MongoDB from Source 1 and Source 2 respectively. The service endpoint is: http://localhost:8080/com.rest.test/rest/data

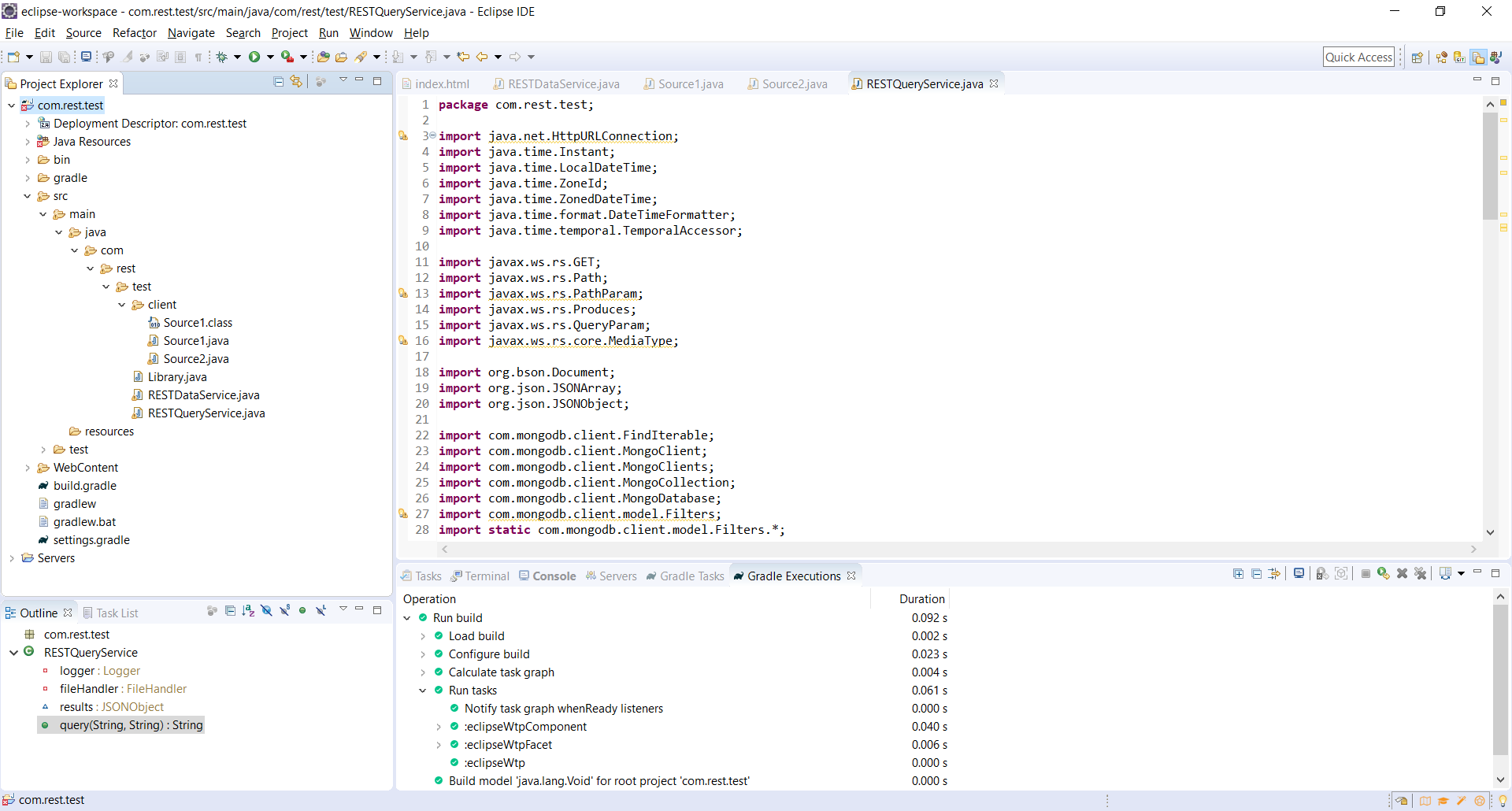
Source 1 (Source1.java) makes a POST request every 15 minutes with the following JSON object to the RESTDataService. {"source": "gitters", "updates": [], "timestamp": 1542912952203} where update format is {"name": "Bob", "git": "hello world!", "timestamp": 1542912641651}.

Source 2 (Source2.java): Responds to GET requests at https://gitter.com:9000/ (or another provider) on an hourly basis. The response format is {"status": "success", "updates": [], "timestamp": 1542912952203}, where update format is {"name": "Bob", "git": "hello world!", "timestamp": 1542912641651}.

Source1.java and Source2.java can be run as standalone Java applications.

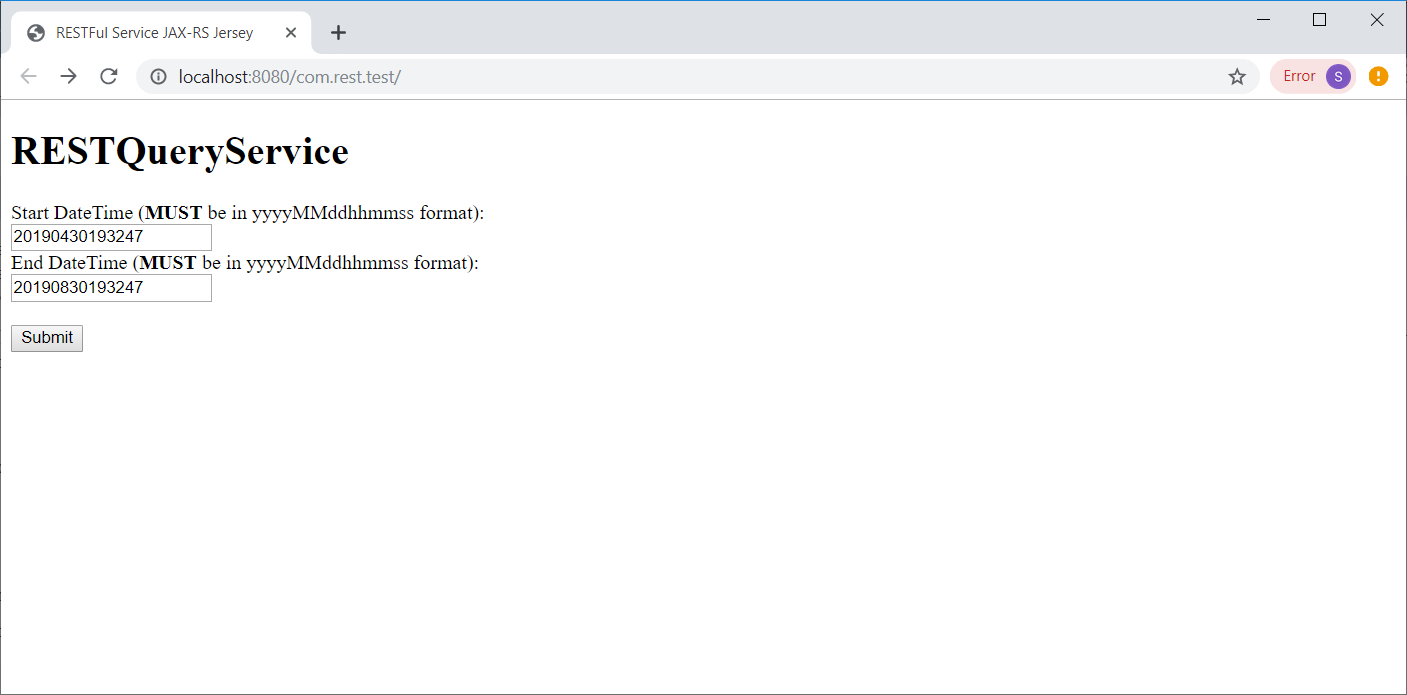
RESTQueryService (RESTQueryService.java) implements the query interface to query the JSON data. A user can query the endpoint with the following parameters: start, end. Both parameters accept only datetimes of the format yyyyMMddhhmmss. The query endpoint is: http://localhost:8080/com.rest.test/rest/query

The source directory structure is as below:



# Querying

The data querying can be done via the web interface below:



A sample result is below:

