Data Federation Engine

About Data Federation Engine

This project is part of Google Summer of Code 2013 with the partnership with Emory University, Department of Biomedical.
Currently there are different Restful services that provide data and it is common now to have Restful API for retrieving health data. This have solved a lot of time from the traditional method where user have to retrieval records and now the next step to solve this problem is that, user have to combine different results from various data source. The purpose of this project is to create a middleware between various different data source and using Map-Reduce to combine the results before returning to the user.

Data Federation Architectural design

This is the overview of how Data Federation architectural design:

Software / Application Needed

In this documnentation, it will cover on how to setup Data Federation Engine. Before setting up Data Federation Engine, these are the things needed to be installed:

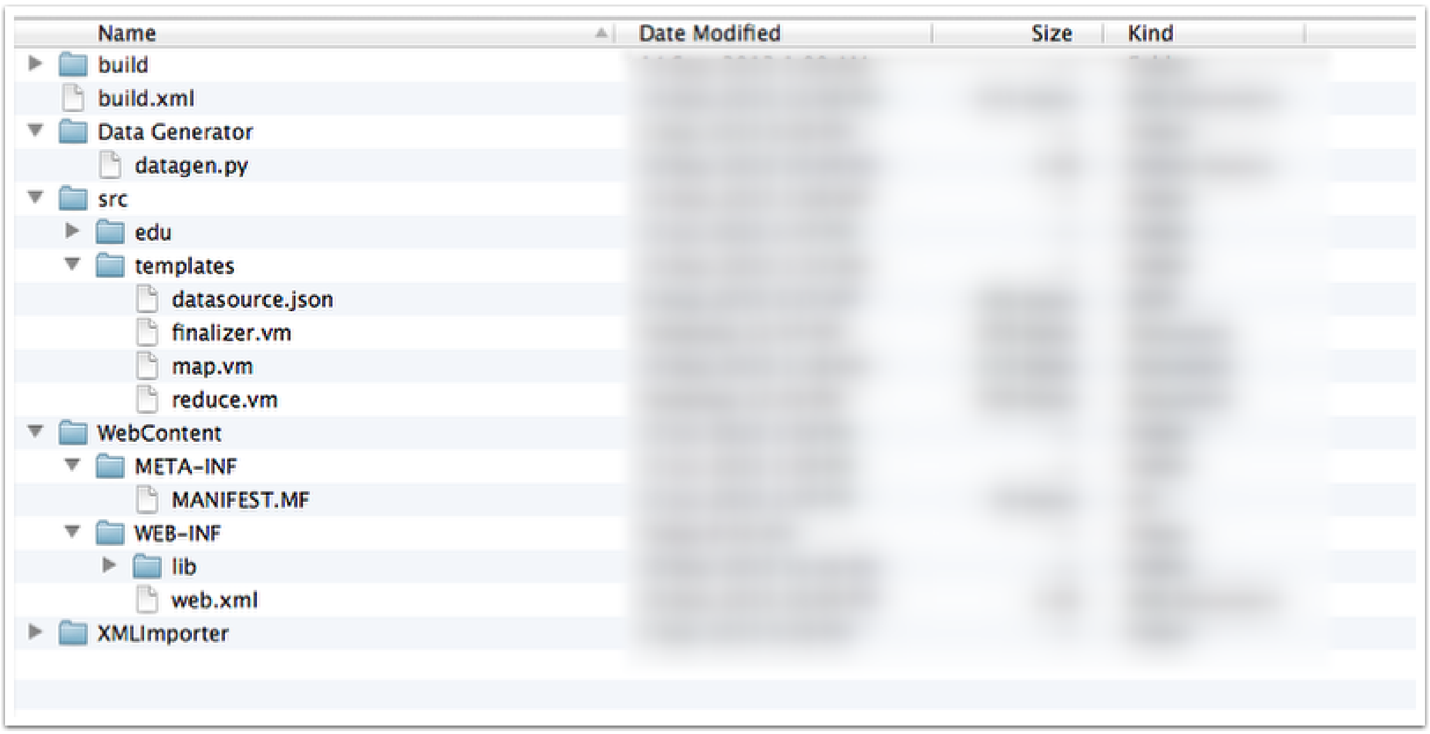
- Java SE 6 (also known as 1.6)
- Tomcat Service 6 or higher
- MongoDB
- Git (Optional)

Downloading Data Federation Engine

Currently, Data Federation Engine is hosted at <URL> and it can be downloaded either by git clone or download the whole zip package.

Inside Data Federation Engine directory

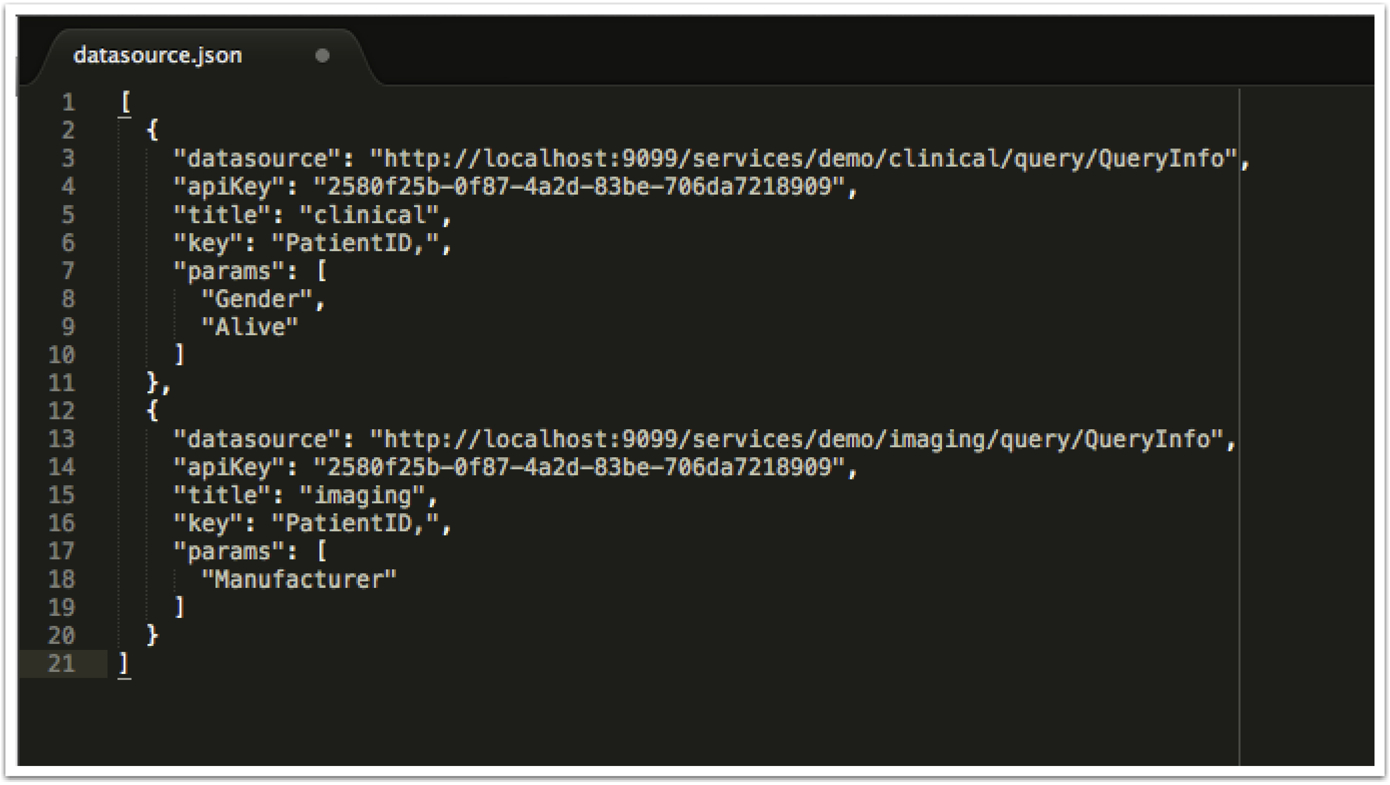
After downloading, you will be able to see the following directory of Data Dederation. All the source code in under src and dependanacy libraries Jar files can be found in WEB-INF/lib folders. Ant build script is named as build.xml where it will compile the project into a War file after configuration.
You can either use Ecplise to open the project or using any text editor to edit the files in the templates directory for configuration purpose.



```
Name                          Date Modified     Size     Kind
▶ build
  build.xml
▼ Data Generator
    datagen.py
▼ src
  ▶ edu
  ▼ templates
      datasource.json
      finalizer.vm
      map.vm
      reduce.vm
▼ WebContent
  ▼ META-INF
      MANIFEST.MF
  ▼ WEB-INF
    ▶ lib
      web.xml
▶ XMLImporter
```

Configurating datasource.json

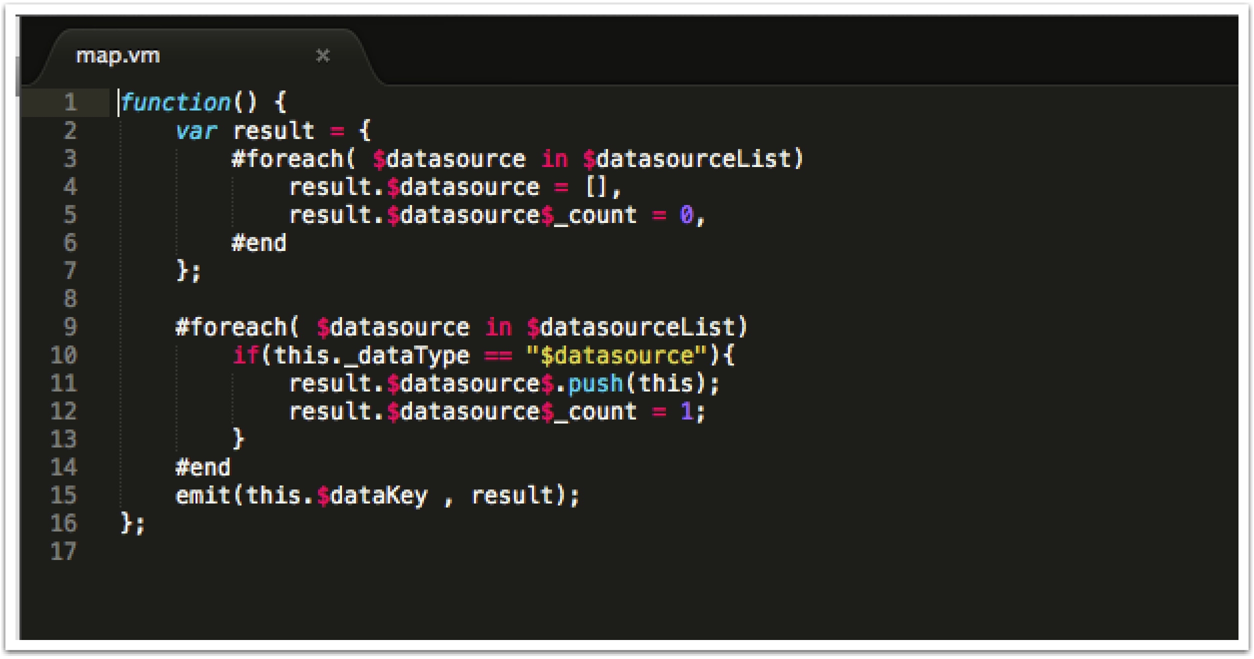
In datasource.json, you will have to add in the the url for each datasource, the apikey, the title of the datasource, the "Primary key" for the API and lastly, the type of parameters that users can call onto the datsource.


datasource.json

```
[
  {
    "datasource": "http://localhost:9099/services/demo/clinical/query/QueryInfo",
    "apiKey": "2580f25b-0f87-4a2d-83be-706da7218909",
    "title": "clinical",
    "key": "PatientID,",
    "params": [
      "Gender",
      "Alive"
    ]
  },
  {
    "datasource": "http://localhost:9099/services/demo/imaging/query/QueryInfo",
    "apiKey": "2580f25b-0f87-4a2d-83be-706da7218909",
    "title": "imaging",
    "key": "PatientID,",
    "params": [
      "Manufacturer"
    ]
  }
]
```

Configurating map.vm

For map.vm, reduce.vm and finalizer.vm, will be loaded for doing MapReduce processing in MongoDB. For each of the template, you will have to either follow Java Velocity templating language or you can write each fuctions and it will not be replace by templating engine.

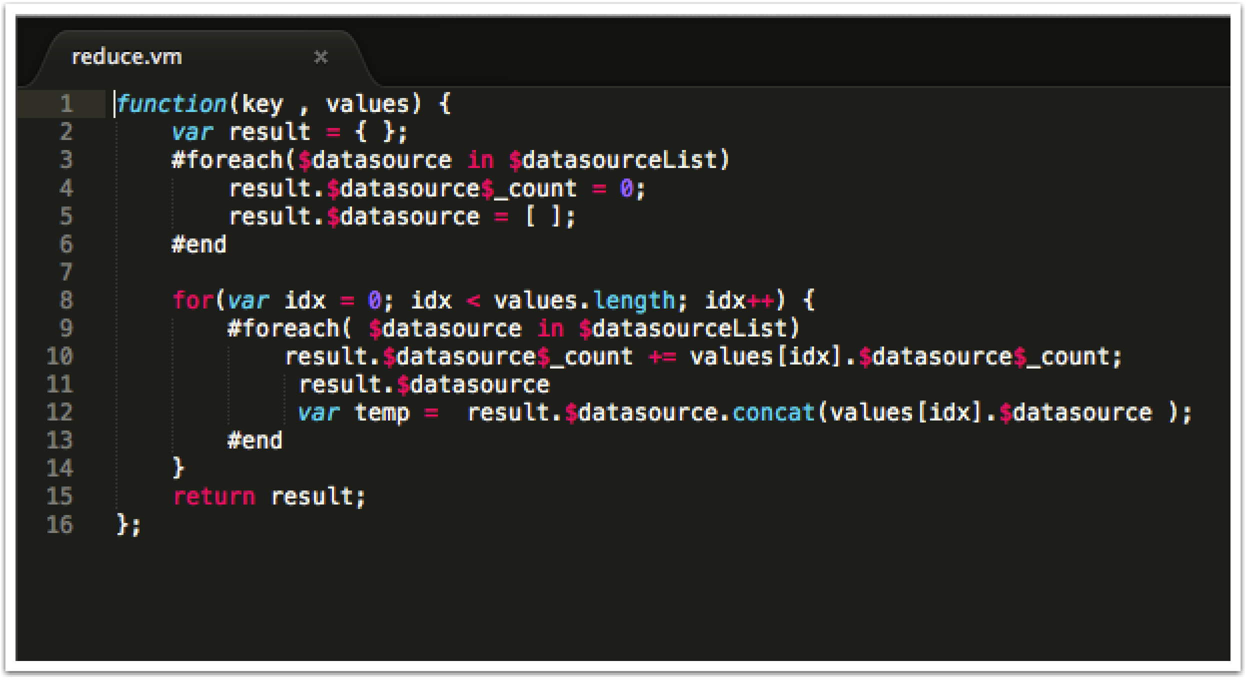


map.vm

```
function() {
    var result = {
        #foreach( $datasource in $datasourceList)
            result.$datasource = [],
            result.$datasource$_count = 0,
        #end
    };

    #foreach( $datasource in $datasourceList)
        if(this._dataType == "$datasource"){
            result.$datasource$.push(this);
            result.$datasource$_count = 1;
        }
    #end
    emit(this.$dataKey , result);
};
```

Configurating reduce.vm

Example of reduce function in reduce.vm.

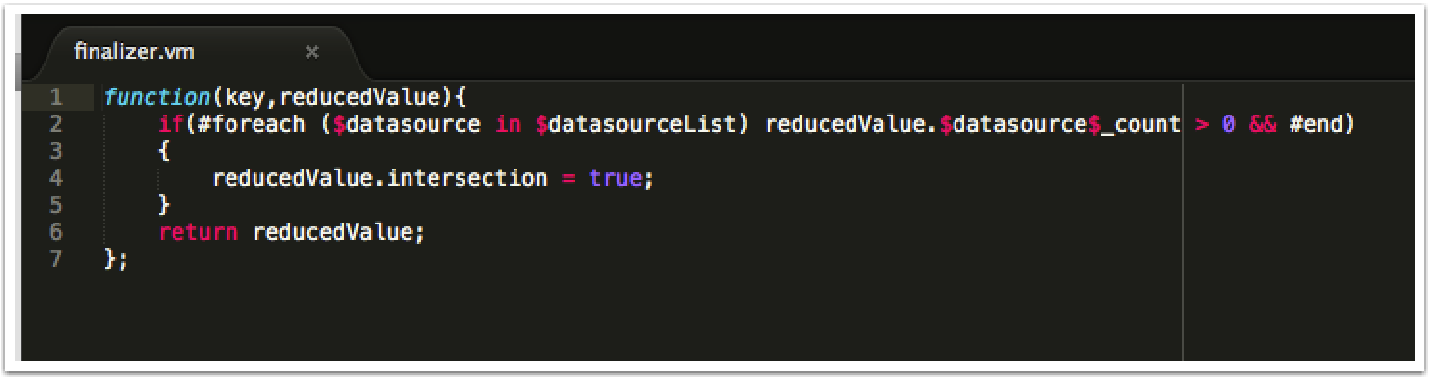


reduce.vm

```
function(key , values) {
    var result = { };
    #foreach($datasource in $datasourceList)
        result.$datasource$_count = 0;
        result.$datasource = [ ];
    #end

    for(var idx = 0; idx < values.length; idx++) {
        #foreach( $datasource in $datasourceList)
            result.$datasource$_count += values[idx].$datasource$_count;
             result.$datasource
             var temp =  result.$datasource.concat(values[idx].$datasource );
        #end
    }
    return result;
};
```

Configurating finalizer.vm

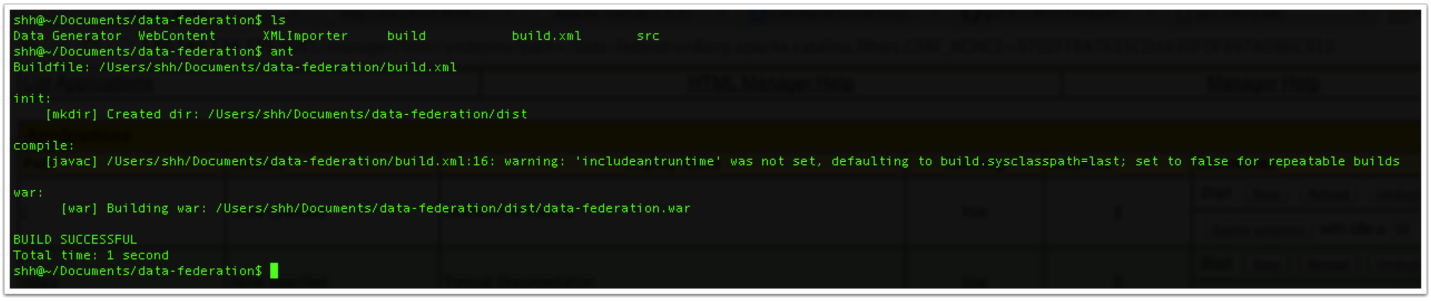
Example of finalizer function in finalizer.vm.



```
function(key,reducedValue){
    if(#foreach ($datasource in $datasourceList) reducedValue.$datasource$_count > 0 && #end)
    {
        reducedValue.intersection = true;
    }
    return reducedValue;
};
```

Runing Ant Script

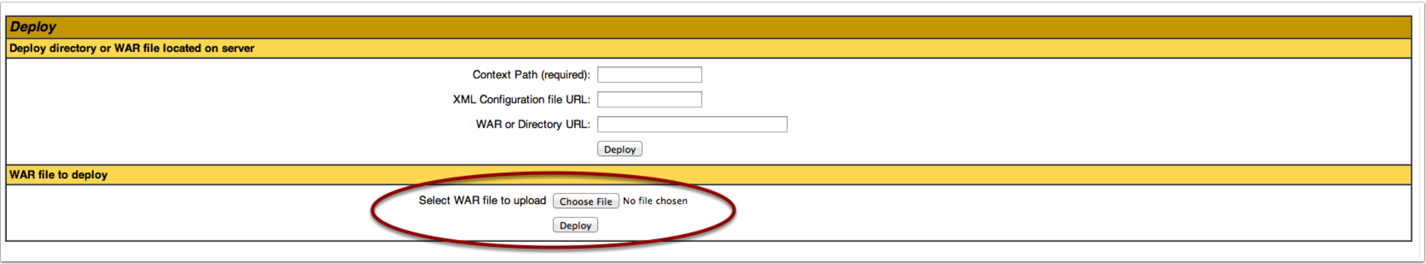
After configure the four files, you will have open naviagte to the directiory of the project and run ant build to create a war file for deployment.



```
shh@~/Documents/data-federation$ ls
Data Generator  WebContent      XMLImporter     build           build.xml       src
shh@~/Documents/data-federation$ ant
Buildfile: /Users/shh/Documents/data-federation/build.xml

init:
    [mkdir] Created dir: /Users/shh/Documents/data-federation/dist

compile:
    [javac] /Users/shh/Documents/data-federation/build.xml:16: warning: 'includeantruntime' was not set, defaulting to build.sysclasspath=last; set to false for repeatable builds

war:
      [war] Building war: /Users/shh/Documents/data-federation/dist/data-federation.war

BUILD SUCCESSFUL
Total time: 1 second
shh@~/Documents/data-federation$
```

Deplying to Tomcat

Lastly, login to Tomcat's App Manager page and select the WAR file to upload for deployment.



Appendix: Bindaas

For this project, Bindaas is being used to create a Restful API to simulate different data source. Bindaas is a project created and developed by Yusuf Nadir Saghar from Emory University. Bindaas is a quick out-of-the-box Restful Services Web Application, which support from various database, such as MySQL, MongoDB etc. It also allows users to have various plugins.

Bindaas can be downloaded from http://imaging.cci.emory.edu/wiki/display/BDS/Downloads and it requires Java SE to be installed.

Appendix: Links

Data Federation Engine is currently hosted at https://github.com/pshken/data-federation.

If there is any problem or bugs, can file a bug at https://github.com/pshken/data-federation/issues and please feel free to fork the project!