**DataMining RESTFUL WEBSERVICE BDD**

**( Build Deployment Document ]**

**Author : Gogs**

This Document describes the deployment steps for dbmining restful webservices and detailed description of Usage. Please follow the steps.

For any Queries – Contact [gog.krish@gmail.com](mailto:gog.krish@gmail.com) / 512-920-7209

This webservice will retrieve query results for **google.com** and **bing.com** and insert in to Mysql Database.

Resource Used:

For Parsing Google Results :

* Jsoup API - <http://jsoup.org/apidocs/>

For Parsing bing Results

Since we don’t have direct way to parse bing html element using jsoup,

I used Azura Api provided by bing. This is a restful webervice which will give you the bing Search Results in the form of XML or JSON.

* <http://datamarket.azure.com/browse/data>

create a database and required tables.

Steps for Creating Database and Tables:

1. Go to your Mysql database server.
2. Copy the file dbtables.sql in any location.
3. Open the sql file.

create user 'hector'@'localhost' IDENTIFIED BY 'hector123';

this line will create a username hector and password hector123 in the db.

GRANT ALL PRIVILEGES ON `datamining`.\* TO 'hector'@'%' IDENTIFIED BY 'hector123';

This line will give full privilege to the database for the username hector/hector123.

datamining - your databse Name.

hector – your database UserName.

hector123 – your databse password.

Execute the sql file as a root admin.

1. Go to mysql promt as a root. [by simple enter ‘mysql’ or mysql –h localhost –u <rootusername> -p , then enter password ]
2. Execute this sql file by entering the command – source dbtables.sql

Validation:

User Validation :

To check whether our mysql user is created. We need to execute this statement.

1. mysql –h localhost –u hector –p , password – hector123
2. Once entered into mysql successfully
3. Enter this command to check the database – use datamining.
4. Enter this command to check the tables created – show tables
5. You should see the below tables.

+------------------+

| Tables\_in\_datamining |

+------------------+

| search\_results |

| related\_results |

Table Validation.

1. Enter this command to validate all our table fields.

mysql> desc search\_results;

+-------------+--------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+-------------+--------------+------+-----+---------+----------------+

| id | int(11) | NO | PRI | NULL | auto\_increment |

| title | varchar(100) | YES | | NULL | |

| url | varchar(100) | YES | | NULL | |

| description | varchar(100) | YES | | NULL | |

| query\_text | varchar(100) | YES | | NULL | |

| source | varchar(100) | YES | | NULL | |

+-------------+--------------+------+-----+---------+----------------+

6 rows in set (0.01 sec)

mysql> desc related\_results;

+------------+--------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+------------+--------------+------+-----+---------+----------------+

| id | int(11) | NO | PRI | NULL | auto\_increment |

| title | varchar(100) | YES | | NULL | |

| url | varchar(100) | YES | | NULL | |

| query\_text | varchar(100) | YES | | NULL | |

| source | varchar(100) | YES | | NULL | |

+------------+--------------+------+-----+---------+----------------+

5 rows in set (0.00 sec)

1. Now we are good with all out table creation.
2. Now we need to do the tomcat deployment

**Tomcat War Deployment steps.**

1. Install apache-tomcat-7 in your server.
2. Go to your tomcat\_home/lib location.
3. Copy the attached db.properties in to tomcat\_home/lib location.
4. Open this file validate.

db\_server=<your DB server>

db\_name=datamining

db\_port=3306

db\_username=hector username that we created..

db\_password=hector123 – Password that we created…

1. Now copy the attached dbmining.war file in to tomcat/webapps folder.
2. Restart your tomcat server.
3. Test ps –f | grep tomcat and check tomcat server is running
4. Go to your browser and enter http://<yourServer>:<your tomcat port>
5. Check the tomcat home page is coming.

**Rest Client Validation**

1. Install Google Chrome in your machine
2. Down the plugin **Postman** and install it
3. Execute each service and validate.

**Rest Web Service Validation.**

1. Go to your postman plugin in Google chrome.
2. Create collections for each URL and Execute.

**Dbmining Search query for Google and Bing and Insert in to DB**

Method : post

Header : Accept - application/xml

**Post Query :**

<http://localhost:8080/dbmining/rest/results/SearchAndInsert>

**Post Data:**

<record>

<query\_text>test automation</query\_text>

</record>

**Response:**

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<SearchResponse>

<responseErrorCode>DATA\_INSERTED</responseErrorCode>

<responseErrorMessage>Information Searched and Inserted Successfully.</responseErrorMessage>

<responseStatus>SUCCESS</responseStatus>

</SearchResponse>

**Search Results by Attribute ( title, url , description , query\_text )**

Method : post

Header : Accept - application/xml

**Post Query :**

<http://localhost:8080/dbmining/rest/results/searchresults/getResultsByAttr>

**Post Data:**

<record>

<attr>title</attr>

<resource>google.com</resource>

</record>

This will give all the title for google.com .

You can use ( title, url, description )

You can use ( google.com / bing.com )

**Response:**

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<SearchResponse>

<responseErrorCode>DATA\_RETRIVED</responseErrorCode>

<responseErrorMessage>Information Retrived Successfully.</responseErrorMessage>

<responseStatus>SUCCESS</responseStatus>

<searchDetails>

<query\_text>test automation</query\_text>

<resource>google.com</resource>

<title>Test automation - Wikipedia, the free encyclopedia</title>

</searchDetails>

<searchDetails>

<query\_text>test automation</query\_text>

<resource>google.com</resource>

<title>Images for test automation</title>

</searchDetails>

<searchDetails>

<query\_text>test automation</query\_text>

<resource>google.com</resource>

<title>Google Test Automation Conference ? Google Developers</title>

</searchDetails>

<searchDetails>

<query\_text>test automation</query\_text>

<resource>google.com</resource>

<title>Why Automated Testing? - TestComplete Support | SmartBear</title>

</searchDetails>

<searchDetails>

<query\_text>test automation</query\_text>

<resource>google.com</resource>

<title>Automated Testing, Automated Testing Tools - TestComplete ...</title>

</searchDetails>

</SearchResponse>

**Search All Results ( google.com / bing.com / ALL )**

Method : post

Header : Accept - application/xml

**Post Query :**

<http://localhost:8080/dbmining/rest/results/searchresults/getAllResultsSource>

**Post Data:**

<record>

<resource>ALL</resource>

</record>

This will give all results for google.com and bing.com

You can use ( google.com ) to get only google.com search results.

You can use ( bing.com ) to get only bing.com search results

**Response:**

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<SearchResponse>

<responseErrorCode>DATA\_RETRIVED</responseErrorCode>

<responseErrorMessage>Information Retrived Successfully.</responseErrorMessage>

<responseStatus>SUCCESS</responseStatus>

<searchDetails>

<desc>Compare with Manual testing. In software testing, test automation is the use of special software (se</desc>

<query\_text>test automation</query\_text>

<resource>google.com</resource>

<title>Test automation - Wikipedia, the free encyclopedia</title>

<url>en.wikipedia.org/wiki/Test\_automation</url>

</searchDetails>

<searchDetails>

<desc>NO\_DATA</desc>

<query\_text>test automation</query\_text>

<resource>google.com</resource>

<title>Images for test automation</title>

<url>NO\_DATA</url>

</searchDetails>

<searchDetails>

<desc>The MacBook Pro is a line of Macintosh portable computers introduced in January 2006 by Apple Inc., </desc>

<query\_text>mac book pro</query\_text>

<resource>bing.com</resource>

<title>MacBook Pro - Wikipedia, the free encyclopedia</title>

<url>http://en.wikipedia.org/wiki/Mac\_Book\_Pro</url>

</searchDetails>

<searchDetails>

<desc>Get fast, free shipping on Mac Pro from the Apple Online Store. Choose between Quad-Core, 12-Core, o</desc>

<query\_text>mac book pro</query\_text>

<resource>bing.com</resource>

<title>Mac Pro - Buy Mac Pro Computers - Apple Store (U.S.)</title>

<url>http://store.apple.com/us/buy-mac/mac-pro</url>

</searchDetails>

</SearchResponse>

**Compare All Results ( For Google and Bing.com )**

Method : post

Header : Accept - application/xml

**Post Query :**

<http://localhost:8080/dbmining/rest/results/searchresults/getCompareResultsByAttr>

**Post Data:**

<record>

<attr>title</attr>

</record>

which attribute you want to compare. Here the input is title. What are all the common titles appears for all the source ( source is = google.com, bing.com…… etc )

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<SearchResponse>

<responseErrorCode>DATA\_RETRIVED</responseErrorCode>

<responseErrorMessage>Information Retrived Successfully.</responseErrorMessage>

<responseStatus>SUCCESS</responseStatus>

<searchDetails>

<attr>Test automation - Wikipedia, the free encyclopedia</attr>

<query\_text>test automation</query\_text>

<resource>google.com</resource>

</searchDetails>

<searchDetails>

<attr>Automated Testing, Automated Testing Tools - TestComplete ...</attr>

<query\_text>test automation</query\_text>

<resource>google.com</resource>

</searchDetails>

<searchDetails>

<attr>Test automation - Wikipedia, the free encyclopedia</attr>

<query\_text>test automation</query\_text>

<resource>bing.com</resource>

</searchDetails>

<searchDetails>

<attr>Automated Testing, Automated Testing Tools - TestComplete ...</attr>

<query\_text>test automation</query\_text>

<resource>bing.com</resource>

</searchDetails>

</SearchResponse>

Related Results APIs

Same functionality for Related results. But different API calls.

In related results, we have only title and urls.

**Get Results by arribute ( title , url )**

Method : post

Header : Accept - application/xml

**Post Query :**

<http://localhost:8080/dbmining/rest/results/relatedresults/getResultsByAttr>

**Post Data:**

<record>

<attr>url</attr>

<resource>bing.com</resource>

</record>

This will give all the urls for bing.com.

You can use : attr = url, title

You can use : resource = bing.com, google.com

Response:

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<SearchResponse>

<responseErrorCode>DATA\_RETRIVED</responseErrorCode>

<responseErrorMessage>Information Retrived Successfully.</responseErrorMessage>

<responseStatus>SUCCESS</responseStatus>

<searchDetails>

<query\_text>test automation</query\_text>

<resource>bing.com</resource>

<url>http://www.bing.com/search?q=Automated+Software+Testing+Tools</url>

</searchDetails>

<searchDetails>

<query\_text>test automation</query\_text>

<resource>bing.com</resource>

<url>http://www.bing.com/search?q=Cucumber+Test+Automation</url>

</searchDetails>

<searchDetails>

<query\_text>test automation</query\_text>

<resource>bing.com</resource>

<url>http://www.bing.com/search?q=Web+Test+Automation</url>

</searchDetails>

</SearchResponse>

**Get All Results ( title , url )**

Method : post

Header : Accept - application/xml

**Post Query :**

<http://localhost:8080/dbmining/rest/results/relatedresults/getAllResultsSource>

**Post Data:**

<record>

<resource>ALL</resource>

</record>

This will give all related results for bing and google.

You can give : resource = google.com to get only google results or bing.com to get only bing results.

Response:

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<SearchResponse>

<responseErrorCode>DATA\_RETRIVED</responseErrorCode>

<responseErrorMessage>Information Retrived Successfully.</responseErrorMessage>

<responseStatus>SUCCESS</responseStatus>

<searchDetails>

<query\_text>test automation</query\_text>

<resource>google.com</resource>

<title>open source test automation tools</title>

<url>http://google.com/search?q=open+source+test+automation+tools</url>

</searchDetails>

</SearchResponse>

**Get compared Results ( title , url )**

Method : post

Header : Accept - application/xml

**Post Query :**

http://localhost:8080/dbmining/rest/results/relatedresults/getCompareResultsByAttr

**Post Data:**

<record>

<attr>description</attr>

</record>

This will compare the attributes for google and bing.

Here the attribute is description. It will get all the same description for google and bing.

Response:

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<SearchResponse>

<responseErrorCode>NO\_DATA\_FOUND</responseErrorCode>

<responseErrorMessage>No data matching query results.</responseErrorMessage>

<responseStatus>FAILURE</responseStatus>

</SearchResponse>