**Wiley 1.0 Release Notes v1- Edited by Kojo Linder, July 6, 2015**

**Overview**

The main focus of Wiley 1.0 release is to address functionality based on storyboard designs.

• User Interface Applications: A collection of web-based search, graph and map applications integrated into several dynamic web pages.

The search window allows the user to search for recall content based on Time and Location (by country and US state). Advanced search functionality includes Keyword and Food Group, which does not exist as a selection choice in the current FDA metadata. The search results are displayed geo-spatially on a map and with index card frames. The results gain further interpretation through the use of dynamic bar, line and donut graphs.

• Custom-Built Web Service Enabled Data Services and Application Programming Interfaces (APIs): Wiley not only incorporates APIs provided by the FDA, but it also utilizes custom built APIs and data services that implements semantic search technology in order to perform data mining fact extraction.

• Cloud-Based Data Storage: Wiley utilizes a cloud-based Resource Description Framework (RDF) Sesame database that is decoupled from the data services and web application to provide expandable, quick result sets for user interface.

• Continuous monitoring: Wiley monitors user queries and provides graphical views of query counts over time as well as a tabular view of query responses.

# Data Services

Table 1 provides a complete list of the Wiley data services. Detailed descriptions of the data services are shown in Appendix A.

Table 1: Wiley Data Services

|  |  |
| --- | --- |
| **Service Name** | **Service Description** |
| /Data/writeToFile | Pulls openFDA from the Food API and writes to a flat file for processing. |
| /data/parsingJsonDataValue | Pulls from openFDA and stores into the Sesame Database for analytics to process the openFDA data |
| /data/update/coordinate | Semantically enriches the openFDA named location data with a translation services that provides the latitude and longitude from Clavin[[1]](#footnote-1) for each named location in the openFDA data. |
| /data/update/foodgroups | Semantically enriches the enforcement report with food group category based on the product description. |
| /data/searchQuery | Provides the search parameters based upon end user selection. |
| /status | Monitors the health and overall system availability. |
| /monitor/init | Initializes the monitoring database |
| /monitor/query | monitors user queries |
| /monitor/queryCount | total number of queries made to Wiley |
| /monitor/summary | providing the queries per day and average query time |
| /monitor/topten | a list of the top ten keywords that were searched for, and the associated count |
| /monitor/querytimes | Provides the count of queries listed by date |

# Detailed Data Services Description

Detailed description of data services.

|  |  |
| --- | --- |
| **URL** | **monitor/init** |
| **Method** | ***GET*** |
| **Query Params** | n/a |  |
| **Mime Type** | n/a |  |
| **Response** | **Example:** **Code:** 200  **Content:** init complete. |
| **Sample Call** | *<Just a sample call to your endpoint in a runnable format ($.ajax call or a curl request) - this makes life easier and more predictable.>* |
| **Notes** | *This is to be run as the last step of install to begin monitoring the system.* |

|  |  |
| --- | --- |
| **URL** | **monitor/query** |
| **Method** | ***Post*** |
| **Query Params** | n/a |  |
| **Mime Type** | application/json |  |
| **Response** | **Example:** **Code:** 200 |
| **Sample Call** | $.ajax({  type: 'POST',  data: JSON.stringify({"searchField":searchword,"responseTime":responseTime, "location":locationtarget}),  contentType: "application/json",  dataType: "json",  url: 'http://localhost:8080/dataIngestion/rest/monitor/query',  }) |
| **Notes** | *This records a query as having been completed*. |

|  |  |
| --- | --- |
| **URL** | **monitor/querycount** |
| **Method** | ***GET*** |
| **Query Params** | n/a |  |
| **Mime Type** | text/plain |
| **Returns** | **Example:350**  A count of the total number of queries completed. |
| **Success Response** | **Example:** **Code:** 200  **Content:** 350 |
| **Sample Call** | *<Just a sample call to your endpoint in a runnable format ($.ajax call or a curl request) - this makes life easier and more predictable.>*  var query\_stats = (function(){  var query\_stats = null;  $.ajax({  'async': false,  url: 'http://localhost:8080/dataIngestion/rest/monitor/queryCount' }  ).done(function(data) {  query\_stats = data;  }); |
| **Notes** | *<This is where all uncertainties, commentary, discussion etc. can go. I recommend timestamping and identifying oneself when leaving comments here.>* |

|  |  |
| --- | --- |
| **URL** | **monitor/summary** |
| **Method** | ***GET*** |
| **Query Params** | n/a |  |
| **Mime Type** | application/json |
| **Returns** | A Summary object with all of the summary data that is being monitored. |
| **Success Response** | **Example:** **Code:** 200  **Content:** {"QueriesSince": 15, "A\_V\_G\_QueryTime": 574292, "Y\_T\_D\_Queries": 150, "QueriesPerDay": 37, "YearlyChangeInQueries": null } |
| **Sample Call** | *<Just a sample call to your endpoint in a runnable format ($.ajax call or a curl request) - this makes life easier and more predictable.>*  $.ajax({  'async': false,  url: 'http://localhost:8080/dataIngestion/rest/monitor/summary',  }); |
| **Notes** | *<This is where all uncertainties, commentary, discussion etc. can go. I recommend timestamping and identifying oneself when leaving comments here.>* |
|  |  |

|  |  |
| --- | --- |
| **URL** | *Monitor/topten* |
| **Method** | *GET* |
| **Query Params** | n/a |  |
| **Mime Type** | application/json |
| **Returns** | A list consisting of the top queries by keyword. |
| **Success Response** | **Example:** **Code:** 200  **Content:** [{"Date":null,"Location":null,"SearchField":"Milk","Count":34,"ResponseTime":null},{"Date":null,"Location":null,"SearchField":"","Count":23,"ResponseTime":null},{"Date":null,"Location":null,"SearchField":"Chicken","Count":20,"ResponseTime":null},{"Date":null,"Location":null,"SearchField":"Pizza","Count":17,"ResponseTime":null},{"Date":null,"Location":null,"SearchField":"ice cream","Count":16,"ResponseTime":null},{"Date":null,"Location":null,"SearchField":"rice","Count":12,"ResponseTime":null},{"Date":null,"Location":null,"SearchField":"Celery","Count":12,"ResponseTime":null},{"Date":null,"Location":null,"SearchField":"orange","Count":4,"ResponseTime":null},{"Date":null,"Location":null,"SearchField":"dessert","Count":2,"ResponseTime":null},{"Date":null,"Location":null,"SearchField":"corn","Count":2,"ResponseTime":null}] |
| **Sample Call** | *<Just a sample call to your endpoint in a runnable format ($.ajax call or a curl request) - this makes life easier and more predictable.>*  $.ajax({  'async': false,  url: 'http://localhost:8080/dataIngestion/rest/monitor/topten'  }) |
| **Notes** | *<This is where all uncertainties, commentary, discussion etc. can go. I recommend timestamping and identifying oneself when leaving comments here.>* |

|  |  |
| --- | --- |
| **URL** | *monitor/querytimes* |
| **Method** | *GET* |
| **Query Params** | n/a |
| **Mime Type** | application/json |
| **Returns** | A list the count of queries done by date |
| **Success Response** | **Example:** **Code:** 200  **Content:** [{"Date":"2015-06-30","Location":null,"SearchField":null,"Count":120,"ResponseTime":null},{"Date":"2015-07-01","Location":null,"SearchField":null,"Count":15,"ResponseTime":null},{"Date":"2015-07-02","Location":null,"SearchField":null,"Count":10,"ResponseTime":null},{"Date":"2015-07-04","Location":null,"SearchField":null,"Count":5,"ResponseTime":null}] |
| **Sample Call** | *<Just a sample call to your endpoint in a runnable format ($.ajax call or a curl request) - this makes life easier and more predictable.>*  $.ajax({  'async': false,  url: 'http://localhost:8080/dataIngestion/rest/monitor/querytimes',  }) |
| **Notes** | *<This is where all uncertainties, commentary, discussion etc. can go. I recommend timestamping and identifying oneself when leaving comments here.>* |

|  |  |
| --- | --- |
| **URL** | *data/searchQuery* |
| **Method** | *GET* |
| **Query Params** | bngDateRng  endDateRng  loc  keywordSearch  foodGroup |  |
| **Mime Type** | application/json |
| **Returns** | A list of all food recalls matching the query. |
| **Success Response** | **Example:**  **Content:** {  "head" : {  "vars" : [  "recallNumber",  "reportDate",  "eventId",  "recallingFirm",  "status",  "location",  "latitude",  "longitude",  "foodGroup",  "classification",  "recallInitiationDate",  "productDescription",  "productQty",  "codeInfo",  "distPattern",  "recallReason",  "voluntaryMandated",  "notification"  ]  },  "results" : {  "bindings" : [  {  "recallNumber" : {  "type" : "literal",  "value" : "F-2317-2012"  },  "productQty" : {  "type" : "literal",  "value" : "xx"  },  "location" : {  "type" : "literal",  "value" : "Fremont,CA"  },  "status" : {  "type" : "literal",  "value" : "Ongoing"  },  "productDescription" : {  "type" : "literal",  "value" : "Licorice Root, WHL;\nDistributed by San Francisco Herb and Natural Food Co.\nFremont, Ca."  },  "foodGroup" : {  "type" : "literal",  "value" : ""  },  "codeInfo" : {  "type" : "literal",  "value" : "Lot 120019."  },  "recallingFirm" : {  "type" : "literal",  "value" : "San Francisco Herb & Natural Food Company"  },  "distPattern" : {  "type" : "literal",  "value" : "Nationwide and Canada."  },  "voluntaryMandated" : {  "type" : "literal",  "value" : "Voluntary: Firm Initiated"  },  "eventId" : {  "type" : "literal",  "value" : "62737"  },  "recallInitiationDate" : {  "datatype" : "http://www.w3.org/2001/XMLSchema#dateTime",  "type" : "literal",  "value" : "2012-07-26T00:00:00.000Z"  },  "classification" : {  "type" : "literal",  "value" : "Class II"  },  "notification" : {  "type" : "literal",  "value" : "Press Release"  },  "recallReason" : {  "type" : "literal",  "value" : "CA Department of Public Health inspectors found a pervasive infestation of mice in the facility."  },  "longitude" : {  "datatype" : "http://www.w3.org/2001/XMLSchema#double",  "type" : "literal",  "value" : "0.0"  },  "reportDate" : {  "datatype" : "http://www.w3.org/2001/XMLSchema#dateTime",  "type" : "literal",  "value" : "2012-09-26T00:00:00.000Z"  },  "latitude" : {  "datatype" : "http://www.w3.org/2001/XMLSchema#double",  "type" : "literal",  "value" : "0.0"  }  }  ]  }  } |
| **Sample Call** | *<Just a sample call to your endpoint in a runnable format ($.ajax call or a curl request) - this makes life easier and more predictable.>*  TODO |
| **Notes** | *<This is where all uncertainties, commentary, discussion etc. can go. I recommend timestamping and identifying oneself when leaving comments here.>* |

|  |  |
| --- | --- |
| **URL** | *data/writeToFile* |
| **Method** | *GET* |
| **Query Params** | n/a |
| **Mime Type** | text/html |
| **Returns** | A success message. |
| **Success Response** | **Example:** **Code:** 200  **Content:** {"sucess": true} |
| **Sample Call** | *<Visiting the url in a web browser is sufficient.>* |
| **Notes** | *<This is where all uncertainties, commentary, discussion etc. can go. I recommend timestamping and identifying oneself when leaving comments here.>* |

|  |  |
| --- | --- |
| **URL** | *data/* *parsingJsonDataValue* |
| **Method** | *GET* |
| **Query Params** | n/a |
| **Mime Type** | text/html |
| **Returns** | A success message. |
| **Success Response** | **Example:** **Code:** 200  **Content:** {"sucess": true} |
| **Sample Call** | *<Visiting the url in a web browser is sufficient.>* |
| **Notes** | *<This is where all uncertainties, commentary, discussion etc. can go. I recommend timestamping and identifying oneself when leaving comments here.>* |

|  |  |
| --- | --- |
| **URL** | *data/update/coordinates* |
| **Method** | *GET* |
| **Query Params** | n/a |
| **Mime Type** | text/html |
| **Returns** | A success message. |
| **Success Response** | **Example:** **Code:** 200  **Content:** {"sucess": true} |
| **Sample Call** | *<Visiting the url in a web browser is sufficient.>* |
| **Notes** | *Obtains the locations available in the Sesame store and looks up their coordinates in Clavin. Once coordinates have been obtained, they are added to Sesame.* |

|  |  |
| --- | --- |
| **URL** | *data/update/foodgroups* |
| **Method** | *GET* |
| **Query Params** | n/a |
| **Mime Type** | text/html |
| **Returns** | A success message. |
| **Success Response** | **Example:** **Code:** 200  **Content:** {"sucess": true} |
| **Sample Call** | *<Visiting the url in a web browser is sufficient.>* |
| **Notes** | *Loads the product description of each report loaded into Sesame and add a relationship with appropriate Food Group based on string matching with the food groups list.* |

1. https://github.com/Berico-Technologies/CLAVIN [↑](#footnote-ref-1)