XBRL API Google Sheet Template: Queries Documentation

PURPOSE

This document explains the queries included in the XBRL API Google Sheet template using 2019 annual research and development data (" XBRL API Template_Research Program").

BACKGROUND

XBRL US documentation: For a detailed documentation on the XBRL Application Programming Interface (API) ("XBRL API"), please refer to the following PDF available on the XBRL US website: http://files.xbrl.us/documents/XBRL-API-V1.4.pdf. Documentation on getting started with the XBRL Google Sheet and add-on using the XBRL API can also be found on the XBRL US website: https://xbrl.us/forums/topic/api-google-sheet/. Sample queries are available at https://xbrl.us/home/use/xbrl-api-community/sample-api-queries/. Additional resources can be found on XBRL US website, such as a discussion forum with Q/A.

XBRL data: A filer may report only a report-wide or default value for an eXtensible Business Reporting Language ("XBRL") XBRL concept, only a single value for the XBRL concept tagged with an axis and member, multiple values for the XBRL concept tagged with multiple axes and members, or a combination of these values. When an XBRL concept is tagged with an axis and member, it is considered to be tagged with a dimension as shown in the following XBRL data excerpt. An axis and member are tagged as a pair, one cannot exist without the other in the XBRL data. This table shows the 2019 XBRL data for research and development expense ("R&D expense") in Axon Enterprise, Inc.'s 2/28/2020 Form 10-K filing.

entity.name	concept.local-name	fact.value	report.filing-date	dimension.local-name	member.local-name
Axon, Enterprise, Inc.	ResearchAndDevelopmentExpense	100,721,000	2/28/2020		
Axon, Enterprise, Inc.	ResearchAndDevelopmentExpense	86,252,000	2/28/2020	StatementBusinessSegmentsAxis	SoftwareandSensorsSegmentMember
Axon, Enterprise, Inc.	ResearchAndDevelopmentExpense	14,469,000	2/28/2020	StatementBusinessSegmentsAxis	TaserWeaponsSegmentMember

Three values are reported for the XBRL concept for R&D expense, ResearchAndDevelopmentExpense. Axon has total R&D expense of $100,721,000, which is the report-wide or default value for this XBRL concept. It is considered the report-wide or default value because there is no dimension associated with this value. Axon also reported values for R&D expense for its two segments as indicated by the inclusion of the dimension StatementBusinessSegmentsAxis and respective members. R&D expense of $86,252,000 is for Axon's Software and Sensors Segment. While R&D expense of $14,469,000 is for Axon's Taser Weapons Segment. This data is considered to be disaggregated by segment and because the sum of the two segment values equal the total amount for R&D expense of $100,721,000, this is a complete disaggregation of R&D expense by segment.

The report-wide or default value is often a total of the disaggregated amounts tagged with dimensions, but sometimes filers may either report the default value or report the dimensionalized value(s) without a default value for the XBRL concept.

CONTENTS

The "XBRL API Template_Research Program" has four worksheets. This document explains the contents of each sheet. Please note any cell containing a query or an input value needed for a query is highlighted in green throughout this file.

Default Data

The first sheet is called Default Data. The Default Data sheet contains the report-wide or default values for up to five XBRL concepts.

In gathering data, it is important to know how to get data by only default values. The query to retrieve XBRL data is found in cell F1 and is as follows:

=showData("https://api.xbrl.us/api/v1/fact/search?concept.local-name=ResearchAndDevelopmentExpense,ResearchAndDevelopmentExpenseExcludingAcquiredInProcessCost,ResearchAndDevelopmentExpenseSoftwareExcludingAcquiredInProcessCost,ResearchAndDevelopmentInProcess,ResearchAndDevelopmentAssetAcquiredOtherThanThroughBusinessCombinationWrittenOff&fact.has-dimensions=FALSE&report.type=10-K,10-K/A&period.fiscal-period=Y&period.fiscal-year=2019&unit=USD&fields=entity.cik,entity.name.sort(ASC),dts.id,fact.id,report.filing-date,report.type,concept.local-name,fact.value,unit,fact.ultimus-index,fact.has-dimensions,fact.offset(0)")

"=showdata" is a function that retrieves data from the XBRL US database and displays it. The query to get data is included within the parenthesis of the showdata function. There are four parts to this query. The first part of query is https://api.xbrl.us/api/v1, which is the URL for the XBRL API.

The second part of the query is /fact/search?

fact is one of the objects available for XBRL API users. An endpoint is simply a collection of objects. The XBRL API is a set of endpoints developed to help data users access structured XBRL data. Objects can be Factual, Metadata, or Assertion. For a complete list of Objects refer to the document: http://files.xbrl.us/documents/XBRL-API-V1.4.pdf.

search is a query parameter, that specifies to the API a search is required for the property of an object.

The third part of the query is concept.local-name=ResearchAndDevelopmentExpense,ResearchAndDevelopmentExpenseExcludingAcquiredInProcessCost,ResearchAndDevelopmentExpenseSoftwareExcludingAcquiredInProcessCost,ResearchAndDevelopmentInProcess,ResearchAndDevelopmentAssetAcquiredOtherThanThroughBusinessCombinationWrittenOff&fact.has-dimensions=FALSE&report.type=10-K,10-K/A&period.fiscal-period=Y&period.fiscal-year=2019&unit=USD. This part of the query can be used to filter or restrict an object's property results based on your search. So, for example, concept.local-name= is equal to a list of concepts separated by comas. The API will only look for the concepts that are listed in this part of the query. Five of the concepts included in the XBRL Taxonomy for R&D expense are being retrieved in this template. If only one XBRL concept is being queried, then we will have only one XBRL element specified in this part of query. In the second filter, fact.has-dimensions=FALSE, the data is being restricted to see only data where there is a default value. By including report.type=10-K,10-K/A, there is an additional restriction for the report type to include only 10-K or 10-K/A filings. If we wanted to see all XBRL filings, we will not have this filter, if we only wanted to see 10-Q filings, we will specify here 10-Q for report.type. Similarly,

period.fiscal-period=Y&period.fiscal-year=2019 is narrowing the search results to include only fiscal year 2019 for each XBRL concept. Lastly, unit=USD restricts the data to values reported in US dollars. If none of these restrictions apply to your data query, then remove these filters from the query. For details on additional restrictions, please see http://files.xbrl.us/documents/XBRL-API-V1.4.pdf.

The fourth and last part of the query is &fields=entity.cik,entity.name.sort(ASC),dts.id,fact.id,report.filing-date,report.type,concept.local-name,fact.value,unit,fact.ultimus-index,fact.has-dimensions,fact.offset(0). In this part, the fields to be included in the data results are specified. Some of the field names are self-explanatory like CIK, entity name. dts.id is specific to the XBRL US database. DTS stands for Discoverable Taxonomy Set, and it defines metadata related with facts in reports, taxonomies, and linkbase. Fact is a value reported for an XBRL concept and fact.id is a unique ID for a reported fact. A unit defines a fact value, it is unit of measure, for example currency. fact.ultimus-index. Ultimus index means number of times a fact has been reported. fact.ultimus-index=1 would imply the most recent fact reported.

The XBRL API access allows a query to retrieve only 2,000 records or rows at a time. To overcome this limit an offset function can be used to retrieve data beyond 2,000 records. Please refer to the query in cell F2002. This cell contains the same query that was included in cell F1, except at the end of the query, fact.offset(2000),,,false. The offset function is requesting to retrieve the next 2,000 records. The next set would include 4,000 for offset, that is, replace 2000 with 4000 within parenthesis. The very last part of the query in cell F2002 contains ,,,false. This allows the query to show data without headers, which helps to keep the continuity in the data results. Continuing with the Axon example, the query in the Default Data sheet retrieves the following data for Axon:

#NAME?	entity.name	dts.id	fact.id	report.filing-date	report.type	concept.local-name	fact.value	unit	fact.ultimus-index	fact.has-dimensions
0001069183	Axon Enterprise, Inc.	368803	225911188	2020-02-28	10-K	ResearchAndDevelopmentExpense	100721000	USD	3	FALSE
0001069183	Axon Enterprise, Inc.	368803	225911412	2020-02-28	10-K	ResearchAndDevelopmentExpense	100700000	USD	2	FALSE
0001069183	Axon Enterprise, Inc.	368803	225912102	2020-02-28	10-K	ResearchAndDevelopmentExpense	100721000	USD	1	FALSE

The data shows three records for Axon for R&D expense because these are different facts. If you notice the fact ID is unique for each of the records. In looking at Axon's Form 10-K filing, one of the facts appears in the Statement of Operations and Comprehensive Income. The record in the middle appears in a note and is rounded. And the third also appears in note. If you were to see the SEC filing, you will see each of the three facts appear in different places across the document.

Data with Dimensions

The second sheet in the R&D 2019 API file is exactly the same as the query in cell F1 in the Default Data sheet through fact,offset(0), except that fact.has-dimensions filter here is set to "TRUE" and additional fields are added to the field section of the query. The additional fields are explained in the following paragraph. The query to get the XBRL data with dimensions included in cell E1 is as follows:
=showData(("https://api.xbrl.us/api/v1/fact/search?concept.local-name=ResearchAndDevelopmentExpense,ResearchAndDevelopmentExpenseExcludingAcquiredInProcessCost,ResearchAndDevelopmentExpenseSoftwareExcludingAcquiredInProcessCost,ResearchAndDevelopmentInProcess,Researc"&"hAndDevelopmentAssetAcquiredOtherThanThroughBusinessCombinationWrittenOff&fact.has-dimensions=TRUE&report.type=10-K,10-K/A&period.fiscal-period=Y&period.fiscal-year=2019&unit=USD&fields=entity.cik,entity.name.sort(ASC),dts.id,fact.id,report.filing-date,re"&"port.type,concept.local-name,dimensions.count,dimension.local-name.sort(ASC),member.local-name,fact.value,unit,fact.ultimus-index,dimension.namespace,member.namespace,fact.offset(0)")).

This query is for data without report-wide or default values. In the fields section of this query, there are additional fields as follows: dimension.local-name,member.local-name,fact.ultimus-index,dimension.namespace,member.namespace. The first two fields are the names of axis and member. Fact.ultimus-index. Ultimus index means the number of times a fact has been reported. Fact.ultimus-index=1 would imply the most recent fact reported. If data user is interested in only the most recent fact reported, a filter can be added as part of the query. . namespace is a reference to the taxonomy in which the axis/member is originating. For example, if the value for dimension.namespace or member.namespace starts with "http://fasb.org/us-gaap/," the axis element is from the US GAAP Financial Reporting Taxonomy, if the value starts with "http://fasb.org/srt/," it is from the SEC Reporting Taxonomy, or if it starts with http://www., it is an entity-specific extension element. Interpretations of these values are included in additional columns with the same name as "us-gaap," "srt," or "extension," respectively. This column is derived using regular expression, extract function.

Continuing with the Axon example, the query in the Data with Dimensions sheet retrieves the following data for Axon.

entity.name	dts.id	fact.id	report.filing-date	report.type	concept.local-name	dimensions.count	dimension.local-name	member.local-name	fact.value	unit	fact.ultimus-index	dimension.namespace	member.namespace	dimension.namespace	member.namespace
Axon Enterprise, Inc.	368803	225912257	2020-02-28	10-K	ResearchAndDevelopmentExpense	1	StatementBusinessSegmentsAxis	SoftwareandSensorsSegmentMember	86252000	USD	1	http://fasb.org/us-gaap/2019-01-31	http://taser.com/20191231	us-gaap	extension
Axon Enterprise, Inc.	368803	225911201	2020-02-28	10-K	ResearchAndDevelop	1	StatementBusinessSe	TaserWeaponsSeg	14469000	USD	1	http://fasb.org/us-	http://taser.com/2	us-gaap	extension

All Data

This sheet combines the data from the Default Data and Data with Dimensions sheets. To view the combination of the default values and the values with dimensions, the following query is used:

=showData(("https://api.xbrl.us/api/v1/fact/search?&concept.local-name=ResearchAndDevelopmentExpense,ResearchAndDevelopmentExpenseExcludingAcquiredInProcessCost,ResearchAndDevelopmentExpenseSoftwareExcludingAcquiredInProcessCost,ResearchAndDevelopmentInProcess,ResearchAndDevelopmentAssetAcquiredOtherThanThroughBusinessCombinationWrittenOff&report.type=10-K,10-K/A&period.fiscal-period=Y&period.fiscal-year=2019&unit=USD&fields=entity.cik,entity.name.sort(ASC),dts.id,fact.id,report.filing-date,report.type,concept.local-name,fact.value,unit,fact.ultimus-index,fact.has-dimensions,dimensions.count,dimensions,fact.offset(0)"))

There is a limitation to the details that can be retrieved. In this data, note you do not see the name of the axis or member. The dimension data is retrieved in a string, a user would have to parse the data in the dimension column or revert to the query format in Data with Dimensions. Therefore, the data user could adjust the query depending on their need.

Continuing with the Axon example, the query in the All Data sheet retrieves the following data for Axon.

entity.name	dts.id	fact.id	report.filing-date	report.type	concept.local-name	fact.value	unit	fact.ultimus-index	fact.has-dimensions	dimensions.count	dimensions
Axon Enterprise,	368803	225911201	2020-02-28	10-K	ResearchAndDevelopmentExpense	14469000	USD	1	TRUE	1	(0 : (dimension_namespace : http://fasb.org/us-gaap/2019-01-31; dimension_local_name : StatementBusinessSegmentsAxis; member_namespace : http://taser.com/20191231; member_local_name : TaserWeaponsSegmentMember;))
Axon Enterprise,	368803	225911188	2020-02-28	10-K	ResearchAndDevelopmentExpense	100721000	USD	3	FALSE	0	null
Axon Enterprise,	368803	225911412	2020-02-28	10-K	ResearchAndDevelopmentExpense	100700000	USD	2	FALSE	0	null
Axon Enterprise,	368803	225912102	2020-02-28	10-K	ResearchAndDevelopmentExpense	100721000	USD	1	FALSE	0	null
Axon Enterprise,	368803	225912257	2020-02-28	10-K	ResearchAndDevelopmentExpense	86252000	USD	1	TRUE	1	(0 : (dimension_namespace : http://fasb.org/us-gaap/2019-01-31; dimension_local_name : StatementBusinessSegmentsAxis; member_namespace : http://taser.com/20191231; member_local_name : SoftwareandSensorsSegmentMember;))

Selected Company

The data set here is for only for the top 5 companies from the S&P 500. Cell B18 – B22 contains the tickers which are then used to get data in the query in cell F1. The query can be mapped to look at input fields as seen in the query in F1 of this sheet. The query in cell F1 is same as one from the Default Value sheet. There are three additional queries in this worksheet that were not covered in the previous sections. Cell R1, U1, and F17 contains the additional query as discussed below.

The query in R1 is searching on the object, Label. Facts and Label are separate objects that cannot be nested together. But they can be joined through dts.id., which is specific to the XBRL US database. DTS stands for Discoverable Taxonomy Set, and it defines metadata related with facts in reports, taxonomies, and linkbases.

The Query in R1 is as follows:

=showData(CONCATENATE("https://api.xbrl.us/api/v1/label/search?concept.local-name=ResearchAndDevelopmentExpense,ResearchAndDevelopmentExpenseExcludingAcquiredInProcessCost,ResearchAndDevelopmentExpenseSoftwareExcludingAcquiredInProcessCost,ResearchAndDevelopmentInProcess,ResearchAndDevelopmentAssetAcquiredOtherThanThroughBusinessCombinationWrittenOff&dts.id=", TEXTJOIN(",", TRUE, H2:H6),"&label.role-short=label&fields=dts.id.sort(DESC),label.text,label.role-short.sort(ASC)")). This query is searching on object label, /label/search? and the TEXTJOIN function is used to include the range of dts.id that are results from the query in F1. TEXTJOIN includes a comma in between the range of dts.id defined by cell H2:H6. TRUE ignores any empty cells. Since we want to connect query R1 and U1 to query F1 based on dts.id we are sorting all the three queries on dts.id. Query parameter sort(DESC) can be added in any field followed by a period. CONCATENATE is used since we are including dts.id from the range of cells. In the US GAAP Taxonomy the label is used to describe an XBRL element and generally represents the human-readable version of the element name. A filer can tailor this label to their own disclosure and such labels are generally referred to their preferred labels.. For a detailed list of labels you can refer to the document: http://www.xbrl.org/Specification/XBRL-2.1/REC-2003-12-31/XBRL-2.1-REC-2003-12-31+corrected-errata-2013-02-20.html#Standard-label-role-attribute-values. In this part of the query, label.role-short=label, the API is retrieving labels when they are the standard label (label). The field label.text is the preferred label as defined by a filer. Labels are useful particularly when there are entity-specific disclosures or extensions.

The query in U1 is searching on the object, Label. Facts and Label are separate objects that cannot be nested together. Query in Q1 can be seen below:

=showdata(CONCATENATE("https://api.xbrl.us/api/v1/relationship/search?dts.id=",TEXTJOIN(",", TRUE, H2:H6),"&relationship.target-name=",TEXTJOIN(",", TRUE, C2:C6),"&network.role-description-like=-

Statement&network.link-name=presentationLink&fields=dts.id.sort(DESC),relationship.target-name,network.role-description.sort(ASC)")) . Here the query is searching on the object relationship. Relationship.target-name is same as concept.local-name from earlier queries. Therange of concepts selected that can be found in cell c2:c6, instead of hardcoding the concepts:ResearchAndDevelopmentExpense,ResearchAndDevelopmentExpenseExcludingAcquiredInProcessCost,ResearchAndDevelopmentExpenseSoftwareExcludingAcquiredInProcessCost,ResearchAndDevelopmentInProcess,ResearchAndDevelopmentAssetAcquiredOtherThanThroughBusinessCombinationWrittenOff. This query is similar to other queries covered earlier. There is one function here which is deployed by XBRL exclusively on network.role-description field only. The -like=- Statement appended to network.role-description allows the user of data to search where the network.role-description contains the words "like" Statement. The field network.role-description is location where a fact appears on a Statement or Disclosure, and the specific name of a Statement or a Disclosure.

The query in cell F17 is similar to query in cell F1. In F1 we are getting data for the 5 specified R&D elements from the US GAAP Taxonomy. In this query, we are searching for concepts where there are entity-specific disclosures or extensions. When concept.is-base is set to false (concept.is-base=false), only the concepts that are extension concepts will be retrieved in the data set. The data results in F1 did not include research and development expense for Amazon. Looking down at the query results from F17 we can see that Amazon is disclosing technology and content expense. This filer specific extension contains research and development expense. Filers who use extensions for research and development typically use terms such as product development expense, engineering and development expense, or technology related expense.