

## List of Operations for the above screen shot with sample request structures.

### Top Volume/ Least Performance/Consumer Traffic– These sections display the top call count metric per operation/ Highest Response times per operation and top consumer’s call counts. All these sections can use the same operation.

Operation Name getMetricsData

1. Example request : For the Services page – say the Top Volume

|  |
| --- |
| firstStartTime=12670000 |
| secondStartTime=12456789 |
| duration=3600 |
| aggregationPeriod=1 |
| sortOrder=ascending |
| numRows=10 |
| metricName=CallCount |
| resourceEntityType=Service |
| resourceEntityName=FindingService |
| resourceEntityResponseType=Operation |

* More detailed comments for getMetricsData

|  |  |  |  |
| --- | --- | --- | --- |
| Parm Name | Mandatory(M)/Optional (O) | Data Type | comment |
| firstStartTime | M | long | UTC long in millsecs since 1970. |
| secondStartTime | M | long | UTC long in millsecs since 1970. |
| duration | M | long | The duration value is added to firstStartTime and secondStartTime to provide a range. |
| aggregationPeriod | M | int | The DB interval table to be queried. Value 1= 1H tables, Value 24= 1D tables. |
| sortOrder | O | enum(ascending/descending) | Indicates whether we want the call count data in ascending order or descending order. |
| numRows | O | long | The number of rows to be returned. |
| metricName | M | String – could be CallCount,ResponseTime,ErrorCount and ErrorToCallRatio. | CallCount for “Top Volume”. ResponseTime for “Least Performance” and CallCount again for Consumer Traffic sections. If “All” or missing –indicates all the possible metrics |
| resourceEntityType | M | enum(pool/machine/service/operation/consumer) | For the services page will be “services” |
| resourceEntityName | O | string | Would be the name of the entity for example “FindingService” |
| resourceEntityResponseType | M | enum(pool/machine/service/operation/consumer) | The entities response type, could be operation as in the services page. |
| resourceEntityResponseName | O | string | If this is missing indicates “ALL operations to be listed” |

### Service selection panel on the left

Operation Name getMetricsMetadata

1. Example request : For the Services page – This will be listing ALL the services

|  |
| --- |
| resourceEntityType=Service  resourceEntityTypeName=All |
| resourceEntityResponseType=Service |

We could be listing ALL the pools like such

|  |
| --- |
| resourceEntityType=Pool |
| resourceEntityTypeName=All |
| resourceEntityResponseType=Pool |

or we could be listing operations for a service like this

|  |
| --- |
| resourceEntityType=Service  resourceEntityTypeName=FindingService |
| resourceEntityResponseType=Operation |

### Trend Graphs on the right

Operation Name getMetricValue

1. Example request : For the Services page – say the Call trend

|  |
| --- |
| criteriaInfo.metricName=CallCount  criteriaInfo.serviceName=FindingService |
| criteriaInfo.roleType=Server  startTime= 123456  duration=86400  aggregationPeriod=1 |

# The following details map the various fields across 2 layers

1. SOAFwk Heartbeats to CAL and CAL Data populated to SOAAnalytics Data model
2. DB query mapping to service request.

## SOAFwk Heartbeats to CAL

This is a format of HB being logged to CAL.

H16:00:10.47 SOA\_FW\_METRIC\_client SoaFwk.Time.Total 0 July 30, 2010 4:00:10 PM MST **svc=HalfCheckoutServiceV1&op=null&case=a:Cheggea1e-0 186c-430b-83e2-7dea1eef4fa&role=0&interval=1&clientDC=PHX&serverDC=Unknown&value0=100&value1=5678**

The data fields in the HB example above are all name-value pairs

For example

**svc= HalfCheckoutServiceV1** is the name of the service, the field is called svc.

**Op=null**, here the op is the operation name.

**case=a:Cheggea1e-0 186c-430b-83e2-7dea1eef4fa** , here the case is the consumer name field

**role=0,** this indicates client side logging, role=1 would be server side logging

**interval=1**, not used by CAL

**clientDC=PHX**, not used by CAL

**serverDC=Unknown**, not used by CAL

**value0=100**

**value1=5678 Both values are stats …we do a SUM of value0 and a SUM of value1 when populating into the DB.**

The other dimension like **SoaFwk.Time.Total** is the metric name field.

Here is the Data model for the table- the data is being populated in ….Please note we typically store Ids …but that involves querying other meta data tables and would complicate matters – so for simplicity’s sake I am just using names.

* FACT table (the main metrics table)

|  |  |  |  |
| --- | --- | --- | --- |
| Name |  |  | Comment |
| SOA\_1H\_POOL\_METRIC\_ID |  |  |  |
| CAL\_CUBE\_ID |  |  | UTC long in millsecs used as startTime or endTime – this is when the data is populated into the DB |
| SOA\_SERVICE\_ID |  |  | Name of the service, HalfCheckoutServiceV1 from the example above. |
| PARTITION\_KEY |  |  |  |
| SOA\_METRIC\_ID |  |  | The metric name like “SoaFwk.Time.Total” which is composite Key with Operation and service from another metadata table. |
| SOA\_USE\_CASE\_ID |  |  | The consumer name like “**a:Cheggea1e-0 186c-430b-83e2-7dea1eef4fa”** |
| POOL\_ID |  |  | The pool name we get from the header of the log |
| ROLE\_TYPE |  |  | Role=0 is the client, role=1 is the server |
| CREATION\_DATE |  |  |  |
| METRIC\_COMPONENT\_VALUE0 |  |  | SUM of Value0 |
| METRIC\_COMPONENT\_VALUE1 |  |  | SUM of Value1 |

* Dim tables SOA-SERVICE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data Type | Default Value | Mandatory | Comment |
| SOA\_SERVICE\_ID | NUMBER(38) |  | X | Sequence generated Primary Key |
| SERVICE\_NAME | VARCHAR2(256) |  | X | Stores Service Name using SOA. |
| CREATION\_DATE | DATE |  | X | Date row was created. Standard DB server timezone (PST) |
| LAST\_MODIFIED\_DATE | DATE |  | X | Date row was last updated. Standard DB server timezone (PST) |

* Dim tables CAL\_DIM\_POOL

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data Type | Default Value | Mandatory | Comment |
| ID | NUMBER(38,0) |  | X |  |
| NAME | VARCHAR2(256) |  | X |  |
| CREATION\_DATE | DATE |  | X | Date row was created. Standard DB server timezone (PST) |
| LAST\_MODIFIED\_DATE | DATE |  | X | Date row was last updated. Standard DB server timezone (PST) |
| UTF8\_STATUS | NUMBER(1) |  |  | A Column to indicate the UTF8 Migration status of the row. Possible value is NULL or 1. Null indicated the record has not been converted to UTF-8 and 1 indicated the record is converted. |
| IS\_SECURE | NUMBER(1) |  |  | 1 = Y, 0 = N |

* Dim tables SOA\_SVC\_POOL

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data Type | Default Value | Mandatory | Comment |
| SVC\_POOL\_ID | NUMBER (38) |  | X | Primary key. Generated by SOA\_SVC\_POOL\_SEQ |
| SVC\_ID | NUMBER (38) |  | X | Foreign service id key , coming from SOA\_SERVICE table |
| POOL\_ID | NUMBER (38) |  | X | Foreign pool\_id key from CAL\_DIM\_POOL |
| CREATION\_DATE | DATE | SYSDATE | X | Date row was created. Standard DB server timezone (PST) |
| LAST\_MODIFIED\_DATE | DATE | SYSDATE | X | Date row was last updated. Standard DB server timezone (PST) |
| ROLE\_TYPE | NUMBER(1) |  |  | possible value will be 0/1, default is 1. Role\_type =0 means client side service is deployed on the pool, Role\_type =1 means server side service is deployed on the pool, default it should be server side service. |

* Dim tables SOA\_USECASE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data Type | Default Value | Mandatory | Comment |
| SOA\_USECASE\_ID | NUMBER(38) |  | X | Sequence generated PK for the table. |
| USECASE\_NAME | VARCHAR2(128) |  | X | The usecase name. The name is unique. |
| DESCRIPTION | VARCHAR2(512) |  |  | A detail description of the usecase. |
| CREATION\_DATE | DATE | SYSDATE | X | Date row was created. Standard DB server timezone (PST) |
| LAST\_MODIFIED\_DATE | DATE |  | X | Date row was last updated. Standard DB server timezone (PST) |

* Dim tables SOA\_METRIC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data Type | Default Value | Mandatory | Comment |
| SOA\_METRIC\_ID | NUMBER(38) |  | X | The primary key generated by sequence SOA\_METRIC\_SEQ |
| METRIC\_NAME | VARCHAR2(256) |  | X | String to identify the metric collected like “SoaFwk.Time.Total” and “SoaFwk.Err.Total” |
| SERVICE\_NAME | VARCHAR2(256) |  | X | FK to soa\_service to identify whiich service the metric is collected |
| OPERATION\_NAME | VARCHAR2(256) |  | X | Name of operation within the service that is being measured  getUser, getItem, getFee |
| CREATION\_DATE | DATE |  | X | Date row was created. Standard DB server timezone (PST) |
| LAST\_MODIFIED\_DATE | DATE |  | X | Date row was last updated. Standard DB server timezone (PST) |

* Dim tables SOA\_SVC\_USECASE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data Type | Default Value | Mandatory | Comment |
| SVC\_USECASE\_ID | NUMBER(38,0) |  | X | Generated by SOA\_SVC\_USECASE\_SEQ. |
| SVC\_ID | NUMBER(38,0) |  | X | Foreign service id key , coming from SOA\_SERVICE table |
| USECASE\_ID | NUMBER(38,0) |  | X | Foreign pool\_id key from SOA\_USE\_CASE |
| CREATION\_DATE | DATE | SYSDATE | X | Date row was created. Standard DB server timezone (PST) |
| LAST\_MODIFIED\_DATE | DATE | SYSDATE | X | Date row was last updated. Standard DB server timezone (PST) |

## Service Request – DB – Service Response Mapping

1. **getMetricMetadata**

GetMetricsMetadataRequest contains three input parameters. They are resourceEntityType, resourceEntityName, and resourceEntityResponseType. The resourceEntityType specifies the entity type of the input dimension. It can be service, pool, consumer etc. The resourceEntityName is a list of strings that contain the names of input dimension. The resourceEntityResponseType specifies the entity type of the output dimension. The output dimension can be service, pool, consumer etc. This operation is used to retrieve the names of output dimensions provided the names of input dimension as the input. This operation is also used to retrieve the list of all names of a dimension. Lets see service-pool mapping first and then will see service-consumer (use case) mapping.

Service-Pool Mapping:

The service information is available in the table SOA-SERVICE table. The pool information is available in CAL\_DIM\_POOL table. SOA\_SVC\_POOL is the mapping table for service-pool mapping. For service-pool mapping, request will be

resourceEntityType – ResourceEntity.SERVICE

resourceEntityName – list of service names like FindingService, AdGroupService etc. These service names should have an entry in SOA\_SERVICE table.

resourceEntityResponseType – ResourceEntity.POOL

For the input service names, the corresponding service ids will be fetched from SOA\_SERVICE table. For those service ids, the mapping pool ids will be fetched from SOA\_SVC\_POOL table. For those pool ids, the corresponding pool names will be fetched from CAL\_DIM\_POOL table. The list of all pool names will be sent as the response. The response GetMetricMetadataResponse has the following structure.

resourceEntityResponseNames – list of pool names.

Service-Consumer Mapping:

The service information is available in the table SOA-SERVICE table. The consumer (use case) information is available in SOA\_USECASE table. SOA\_SVC\_USECASE is the mapping table for service-consumer mapping. For service-consumer mapping, request will be

resourceEntityType – ResourceEntity.SERVICE

resourceEntityName – list of service names like FindingService, AdGroupService etc. These service names should have an entry in SOA\_SERVICE table.

resourceEntityResponseType – ResourceEntity.CONSUMER

For the input service names, the corresponding service ids will be fetched from SOA\_SERVICE table. For those service ids, the mapping consumer ids will be fetched from SOA\_SVC\_USECASE table. For those consumer ids, the corresponding consumer names will be fetched from SOA\_USECASE table. The list of all consumer names will be sent as the response. The response GetMetricMetadataResponse has the following structure.

resourceEntityResponseNames – list of consumer names.

Other than mapping, the same operation getMetricsMetadata operation can be used for retrieving the list of all names of a particular dimension. For example, to retrieve the list of service names, the request structure will be

resourceEntityType – ResourceEntity.SERVICE

resourceEntityName – to be null or empty.

resourceEntityResponseType – to be null or empty.

The list of all service names will be fetched from SOA\_SERVICE table. The response structure will be

resourceEntityResponseNames – list of service names.

Similarly, the pool names are to be fetched from CAL\_DIM\_POOL table and the consumer names are to be fetched from SOA\_USECASE table.

1. **getMetricData**

GetMetricsRequest contains two input parameters. This operation is mainly for fetching different metric data like RequestCount, ErrorCount and ResponseTime etc. The metricCriteria is of type MetricCriteria. The MetricCriteria type contains the members firstStartTime, secondStartTime, duration, aggregationPeriod, sortOrder, numRows, metricName, roleType, and autoDelay. The firstStartTime and secondStartTime are two different times so that the metric data can be compared for these times. The duration is implicit. The aggregationPeriod specifies the period over which the aggregation of data happens. For example, some users may need hourly data and some users may need daily data. For hourly data, the aggregationPeriod will be 60 (mins) and for daily data, the aggregation period will be 1440 (mins). The numRows is implicit. While asking metric data operations, if we specify the numRows as 10, then , metric data should be returned for top 10 operations. The metricName specifies the name of the metric (ErrorCount or RequestCount or ResponseTime etc.). Regarding role type, there are only two role types, they are server and client. The role type “server” will be translated to “1” and all other role types will be translated to “0”. The default role type will be “1” (server). The logic for autoDelay is given in detail in the next section. The metricResourceCriteria is of type MetricResourceCriteria. The MetricResourceCriteria type contains four members. The member resourceEntityType specifies the entity type of the input dimension. The resourceEntityResponseType specifies the entity type of the output dimension for which metric data is to be returned. The resourceEntityName specifies the names of input dimensions. The resourceEntityResponseName specifies the names of output dimensions for which metric data is to be returned. Sample request/responses are given below.

1. **To get metric data (call count) for top 10 operations of a service**

GetMetricsRequest (metricCriteria, metricResourceCriteria) contains the following

firstStartTime – 1256684400

secondStartTime – 1256694400

Duration – 86401

aggregationPeriod – 60 (hourly data wanted by user)

autoDelay – false (default)

numRows – 10

metricName – RequestCount

roleType – server

resourceEntityType – ResourceE ntity.SERVICE

resourceEntityName – AdGroupService

resourceEntityResponseType – operation

resourceEntityResponseName – empty (for this scenario)

With this request, the user wants the call count metric data for top 10 operations of the AdGroupService. The service id will be fetched from SOA\_SERVICE table for the “AdGroupService”. The metric ids will be fetched from SOA\_METRIC table with service id and metric name “SoaFwk.Time.Total” (for RequestCount metric) as the inputs. Each metric id present in SOA\_METRIC table represents the unique combination of service name, operation name and metric name. For example, only one should row should be present in SOA\_METRIC table for the AdGroupService (service name), addAdGroup (operation name), and SoaFwk.Time.Total (metric) and a metric id will represent this unique combination. Once the metric ids are retrieved from SOA\_METRIC table, the final query will be made to SOA\_1H\_POOL\_METRIC table to retrieve actual metric data. The SOA\_1H\_POOL\_METRIC table will be queries twice (once for firstStartTime+duration and once for secondStartTime+duration).

While querying SOA\_1H\_POOL\_METRIC table, the following will be provided as inputs.

Metric ids (SOA\_METRIC\_ID) – the values retrieved from SOA\_METRIC table as mentioned above.

Role type (ROLE\_TYPE) – 1 (based on the same calculation explained in the “getMetricValue” section)

Cal cube id (CAL\_CUBE\_ID) – this specifies the duration.

The start cal cube id is equal to firstStartTime (provided in the request, will be recalculated if autoDelay is true). The end cal cube id is computed as start cal cube id + duration. The cal cube id condition will be given as follows in the query.

“*CAL\_CUBE\_ID >= start cal cube id AND CAL\_CUBE\_ID < end cal cube id”*

As metric data is to be aggregated for each operation of the “AdGroupService”, the query should contain the following group by clause.

*group by SOA\_METRIC\_ID*

SOA\_METRIC\_ID is the combination of service name, operation name and metric name. The rows fetched from SOA\_1H\_POOL\_METRIC will represent the same service name (AdGroupService) and same metric name (SoaFwk.Time.Total) but different operation name, hence, the call count aggregation will happen for each operation.

The output of the query will be soa\_metric\_id and sum (metric\_component\_value0). The query may start like “*select SOA\_METRIC\_ID, sum (METRIC\_COMPONENT\_VALUE0)…..”*

As mentioned above, two queries will be executed to SOA\_1H\_POOL\_METRIC table (one for firstStartTime+duration and another for secondStartTime+duration).

At the end, we will be having two sets of metric data. Each set will have a list of soa\_metric\_id and corresponding call count.

The next step will be the consolidation of these two sets of data. The first set of data corresponds to count1 (firstStartTime) and the second set of data corresponds to count2 (secondStartTime). The consolidation logic should produce the output in the following format (sample).

Soa\_metric\_id count1 count2

1111 10 20

2222 34 27

3333 67 32

As explained earlier, each soa\_metric\_id will corresponds to one operation of AdGroupService. The operation should be retrieved from soa\_metric\_id. This retrieval is implementation specific. Caching could be done during the retrieval of metric id from SOA\_METRIC table before executing query to SOA\_1H\_POOL\_METRIC table as explained above. The values are sorted based on the call count and top 10 rows (operations) are populated to MetricGroupData

The response will be the list of MetricGroupData. Each MetricGroupData object will be like

criteriaInfo – only contain operation name (let’s say, the operation addAdGroup corresponds to the soa\_metric\_id 2222 mentioned above)

count1 – 34

count2 – 27

Diff - % change between count1 and count2

1. **To get top 10 errors for a service/consumer combination and associated metrics.**

This scenario is bit different from that of one explained above. In the above scenario, the metricName is provided as “RequestCount” in the request, but, in this scenario, the metricName need not be mentioned in the request. This is because, metric defaults to ErrorCount when the response is top 10 errors. This is explained in detail below.

GetMetricsRequest(metricCriteria, metricResourceCriteria) contains the following.

firstStartTime – 1256216400

secondStartTime – 1256256000

Duration – 86401

aggregationPeriod – 60 (hourly data wanted by user)

autoDelay – false (default)

numRows – 10

metricName – (No need to provide metric name, as it defaults to error count when the response is top 10 errors)

roleType – server

resourceEntityType – ResourceE ntity.SERVICE

resourceEntityName – ShippingEngineService

resourceEntityType – ResourceEntity.CONSUMER

resourceEntityName - ShippingEngineServiceClient

resourceEntityResponseType – Error

resourceEntityResponseName – empty (for this scenario)

The flow this scenario is exactly similar to the previous one. The difference is

1. The metric name while query the SOA\_METRIC table should be taken as “SoaFwk.Err.%” , considering all the error metrics, of course, there are different kinds of error metrics so that the top 10 error list should be prepared considering all these error metrics.

The response will be the list of MetricGroupData. The response will be

CriteriaInfo – with only metricName field populated (This will represent the name of the error metric, in this case, for example “SoaFwk.Err.Severity” etc.) This is a slightly different scenario from others, as we are directly populating the actual metric name in the metricName field of criteriaInfo, rather than populating as ErrorCount, RequestCount, ResponseTime etc.

Count1 – 715.0

Count2 – 124.0

Diff - -82.66(%change between count1 and count2)

1. **getMetricValue**

GetMetricValueRequest contains five input parameters. This operation is mainly for fetching the granular data with which graphs can be plotted. The criteriaInfo specifies the combination of values of various dimensions (like service, operation, consumer, pool, role type etc.). Not all the dimensions are mandatory in criteriaInfo. The startTime and duration are implicit. The serviceConsumerType represents whether the consumer is internal or external. By default, all consumers will be considered. The aggregationPeriod specifies the period over which the aggregation of data happens. For example, some users may need hourly data and some users may need daily data. For hourly data, the aggregationPeriod will be 60 (mins) and for daily data, the aggregation period will be 1440 (mins). If the implementation does not support the aggregationPeriod that the user is requesting, the implementation may throw exception. The logic related to autoDelay is given below.

if autoDelay is true, then, delay will be calculated based on startTime, duration and max cube id.  The computation is below.

*Var maxcubeid = maximum cube id available in the fact table 1H, or 1D or 1MIN (data is available only upto this cube id)*

*If startTime+duration > maxcubeid*

*delay = startTime+duration-maxcubeid*

*else*

*delay = 0*

Based on this calculation, startTime (request parameters that the user passes) will be recalculated.

*startTime = startTime – delay*

The GetMetricValueResponse contains timeSlot and corresponding count, which may be used for plotting graphs on the client side UI.

Let’s see some sample requests given below.

1. Request with metric name “RequestCount” (Simple metric)

criteriaInfo - p:v3apicscore (consumerName), **RequestCount** (metricName), v3soa1core (poolName), server (roleType), AnonymizerService (serviceName)

startTime – 1256684400

duration – 86401

aggregationPeriod – 60 (hourly data wanted by user)

autoDelay – false (default)

serviceConsumerType – null (

With this request, the user wants RequestCount metric and is the simple metric (can be retrieved directly from database and no additional computation required). The metric ids (SOA\_METRIC\_ID) will be retrieved from SOA\_METRIC table by providing service name, operation name and metric name as inputs (SERVICE\_NAME, OPERATION\_NAME, METRIC\_NAME). The metric name should be given as SoaFwk.Time.Total (RequestCount) while querying SOA\_METRIC table. The pool id will be retrieved from CAL\_DIM\_POOL by providing pool name as input. The consumer id (also called use case id) will be retrieved from SOA\_USECASE table by providing consumer name as input. Regarding role type, there are only two role types, they are server and client. The role type “server” will be translated to “1” and all other role types will be translated to “0”. The default role type will be “1” (server). In this example, role type will be translated into “1”. As aggregation period is 60, the SOA\_1H\_POOL\_METRIC table will be queried for the results in this example. If aggregationPeriod is 1440, then, SOA\_1D\_POOL\_METRIC table will be queried. These decisions depend on the data model design.

While querying SOA\_1H\_POOL\_METRIC table, the following will be provided as inputs.

Metric ids (SOA\_METRIC\_ID) – the values retrieved from SOA\_METRIC table as mentioned above.

Pool id (POOL\_ID) – the value retrieved from CAL\_DIM\_POOL table as mentioned above.

Usecase (or consumer) id (SOA\_USE\_CASE\_ID) – the value retrieved from SOA\_USECASE table as mentioned above.

Role type (ROLE\_TYPE) – calculation explained above

Cal cube id (CAL\_CUBE\_ID) – this specifies the duration.

The start cal cube id is equal to startTime (provided in the request, will be recalculated if autoDelay is true). The end cal cube id is computed as start cal cube id + duration. The cal cube id condition will be given as follows in the query.

“*CAL\_CUBE\_ID >= start cal cube id AND CAL\_CUBE\_ID < end cal cube id”*

The output of the query (to SOA\_1H\_POOL\_METRIC) will be cal\_cube\_id, sum (metric\_component\_value0) as follows.

The query may start like “*select CAL\_CUBE\_ID, sum(METRIC\_COMPONENT\_VALUE0)…..”*

The response object mapping will be as follows.

timeSlot – cal\_cube\_id

count – sum(METRIC\_COMPONENT\_VALUE0)

criteria – a string representing the combination of criteria (entity combination).

Note: As the user requests RequestCount metric (a simple metric), sum (METRIC\_COMPONENT\_VALUE0) will be mapped directly to “count” in the response object.

1. Request with metric name “ResponseTime” (Composite metric)

criteriaInfo - p:v3apicscore (consumerName), **ResponseTime** (metricName), v3soa1core (poolName), server (roleType), AnonymizerService (serviceName)

startTime – 1256684400

duration – 86401

aggregationPeriod – 60 (hourly data wanted by user)

autoDelay – false (default)

The only difference in this sample request is “ResponseTime” instead of “RequestCount”. The ResponseTime is a complex metric the computation formula is given below.

SoaFwk.Time.Total:TotalTime/(SoaFwk.Time.Total:Count \* 1000000)

This example is similar to the previous example till the point of executing the final query to SOA\_1H\_POOL\_METRIC table. In the query, sum (METRIC\_COMPONENT\_VALUE1) is also needed along with sum (METRIC\_COMPONENT\_VALUE0). The query may start like

“*select CAL\_CUBE\_ID, sum(METRIC\_COMPONENT\_VALUE0),sum(METRIC\_COMPONENT\_VALUE1)…..”*

Once the query is executed, the response object mapping will be as follows.

timeSlot – cal\_cube\_id

count – sum(METRIC\_COMPONENT\_VALUE1)/(sum(METRIC\_COMPONENT\_VALUE0) \* 1000000)

criteria – a string representing the combination of criteria (entity combination).

Note: ResponseTime is computed, and is not a direct mapping from the select query.

1. When the user is requesting for “ErrorCount” instead of “RequestCount” or “ResponseTime”, it is exactly similar to example 1 with the following change.

“The metric ids (SOA\_METRIC\_ID) will be retrieved from SOA\_METRIC table by providing service name, operation name and metric name as inputs (SERVICE\_NAME, OPERATION\_NAME, METRIC\_NAME). The metric name should be given as **SoaFwk.Err.Total** (ErrorCount) while querying SOA\_METRIC table.

1. **getErrorMetricsMetadata**

This operation is used to retrieve the error details of an error using the unique identifier. This unique identifier may be the error id or error name based on the storage provider. The request and response structure is below.

GetErrorMetricsMetadataRequest contains three parameters. The first one is “errorId”. The “errorId” refers to the unique identifier of each error. The second parameter is “errorName”. Either errorid or error name should be provided while calling this operation but not both.

<<The service name parameter to be updated>>

GetErrorMetricsMetadataResponse contains an “ErrorInfos” object. The fields of “ErrorInfos” are id (error id, and not required if error name itself is the unique identifier), name (error name), category (category of the error like system error, application error, or system error), severity (severity of the error whether it is critical, warning etc.), domain (domain in which the error normally occurs), and sub domain (if applicable). The dimension used in this operation is SOA\_DIM\_ERROR (structure is given below).

1. **getErrorGraph**

This operation is used to retrieve the error metric data at the granular level with which graphs can be plotted.

GetErrorGraphRequest contains 7 fields. The fields serviceName, operationName, consumerName are self explanatory. The field “errorId” refers to the unique identifier of the error, and may be error id, or error name based on the storage provider. The “errorCategory” refers to the category of the error like system error, application error, etc. The field “errorSeverity” can be critical, warning etc. The field “metricCriteria” has already been explained in the previous operations. Again, as this operation is specifically meant for error related metrics, there is no need to specify the “metricName” field in metricCriteria. The graph data will always be returned for singled duration of time so that the field “secondStartTime” of MetricCriteria need not be provided in the request. The graph data will be the split up of data based on various timeslot (response described below) for the error id passed in the request.

GetErrorGraphResponse contains a list of ErrorGraphData objects. The ErrorGraphData contains only two fields, and they are timeSlot and count.

This operation is simply the error metrics version of “getMetricValue”, and is similar to that. The metric table used for this operation is SOA\_1H\_ERROR\_METRIC, and the structure is given below.

1. **getErrorMetricsData**

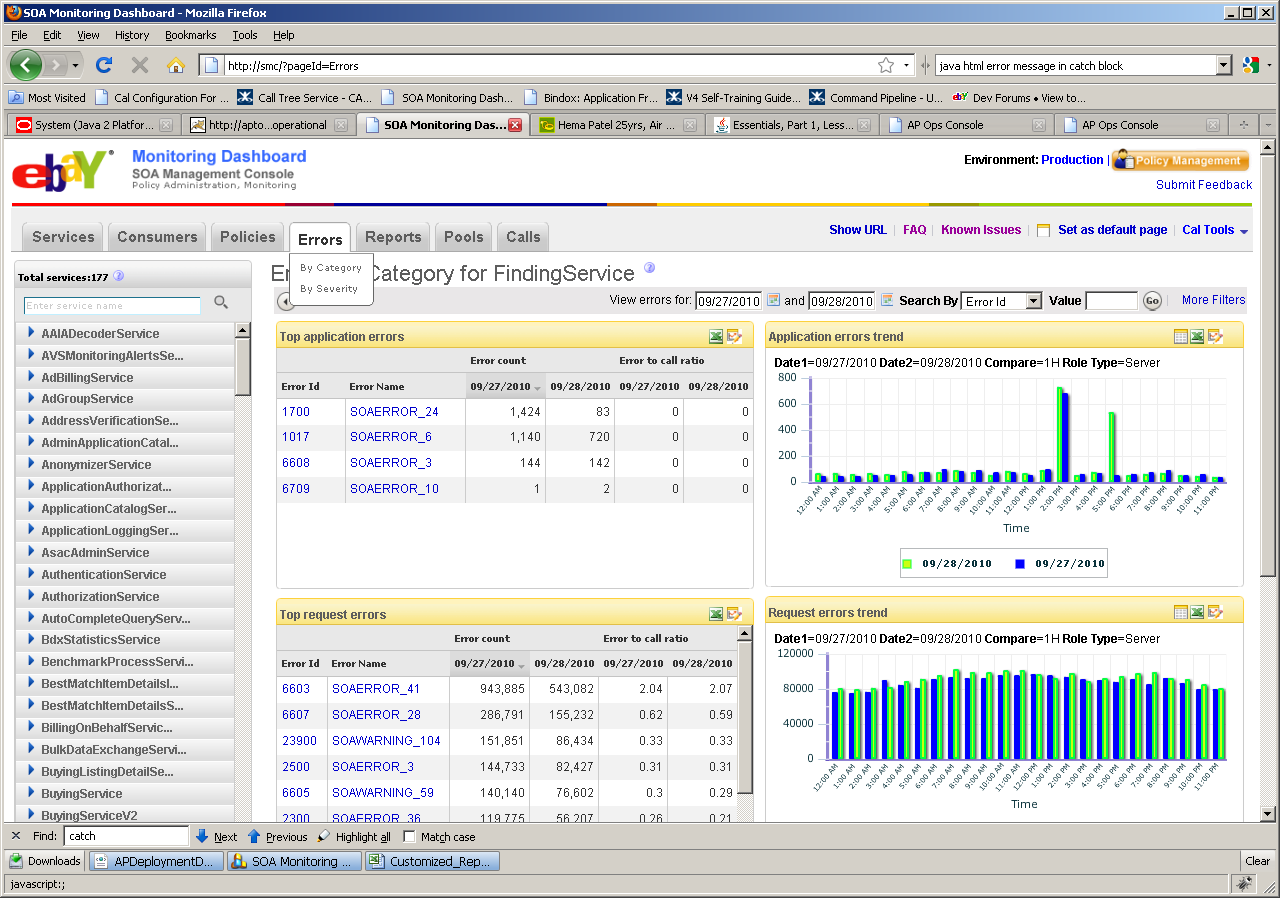
This operation is used to retrieve the error metric data. The three main use cases of this operation are given below.

* 1. To retrieve the top N errors by category (refer “SOA Error Page by category” section given below)
  2. To retrieve the top N errors by severity (refer “SOA Error Page by Severity” section given below)
  3. To retrieve the consumers for an error id or error name and associated error metrics (refer “Drill Down 2” section given below)

GetErrorMetricsDataRequest contains errorType, serviceName, operationName, consumerName, errorId, errorName, errorCategory, errorSeverity, and responseCriteria. The responseCriteria may contain the values “consumer” or “error”. If the responseCriteria is “consumer”, then, errorId or errorName field should be populated in the request, but not both. The operation will return the list of consumer names for that particular error id or error name with associated error metrics. If the responseCriteria is “error”, then, errorId or errorName should not be provided in the request. The operation will return the list of errors with associated error metric data. The responseCriteria will be “consumer” for the scenario 3 mentioned above, and will be “error” for the scenarios 1 and 2 mentioned above. The “errorCategory” parameter has to be provided in the request for the scenario 1 mentioned above and “errorSeverity” has to be provided for the scenario 2.

GetErrorMetricsDataResponse contains the list of ErrorViewData. The ErrorViewData contains criteriaName, errorCount1, errorCount2, errorCallRatio1, errorCallRatio2, errorDiff, and ratioDiff. For the scenarios 1 and 2 mentioned above, the criteriaName will be error id or error name. For the scenario 3, the criteriaName will be the name of the consumer. The metric table used for this operation is SOA\_1H\_ERROR\_METRIC, and the structure is given below.

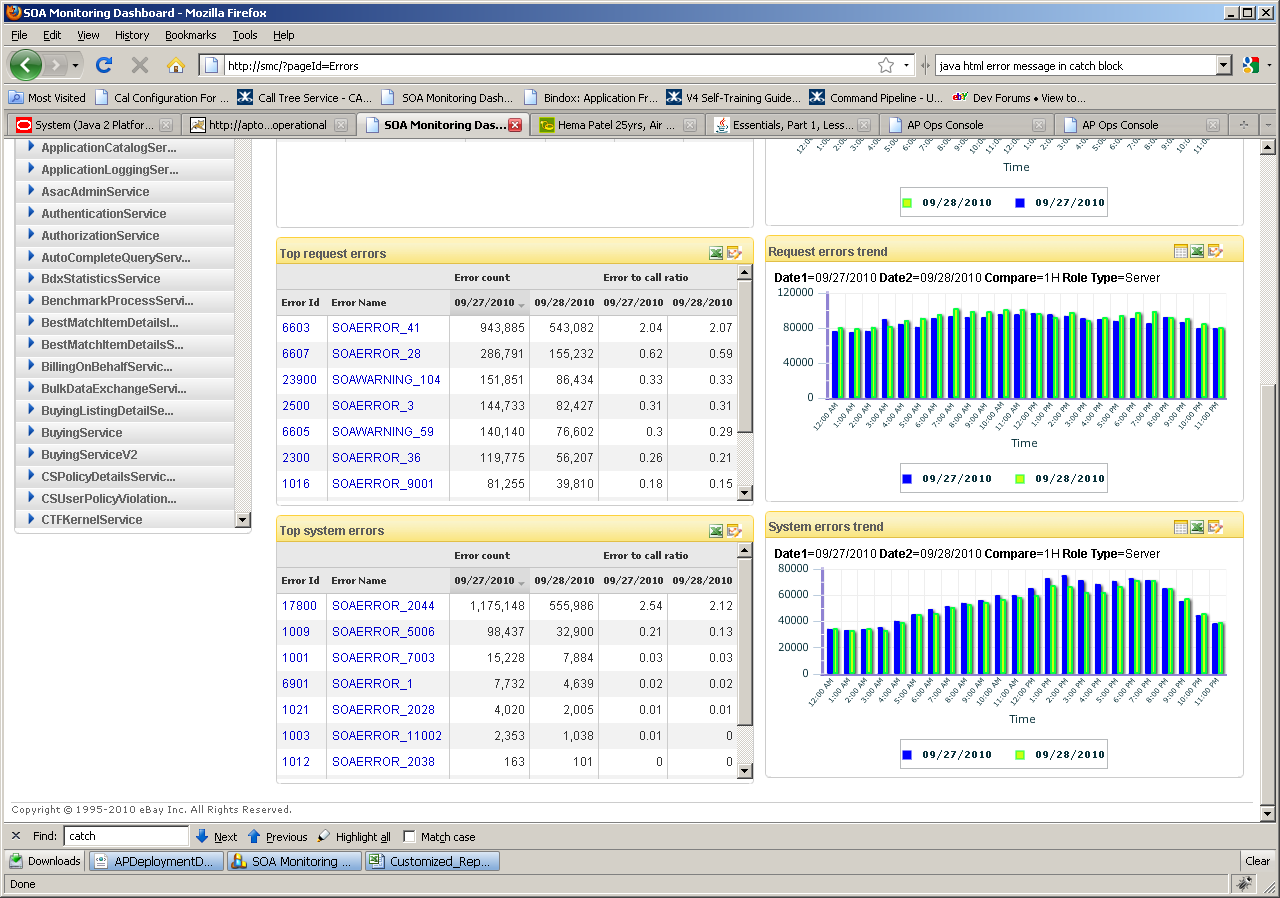
#### TOP LANDING PAGE -> SOA Errors Page - ByCategory



Page can be selected “ByCategory” and “BySeverity”. Currently displayed page is “ByCategory”

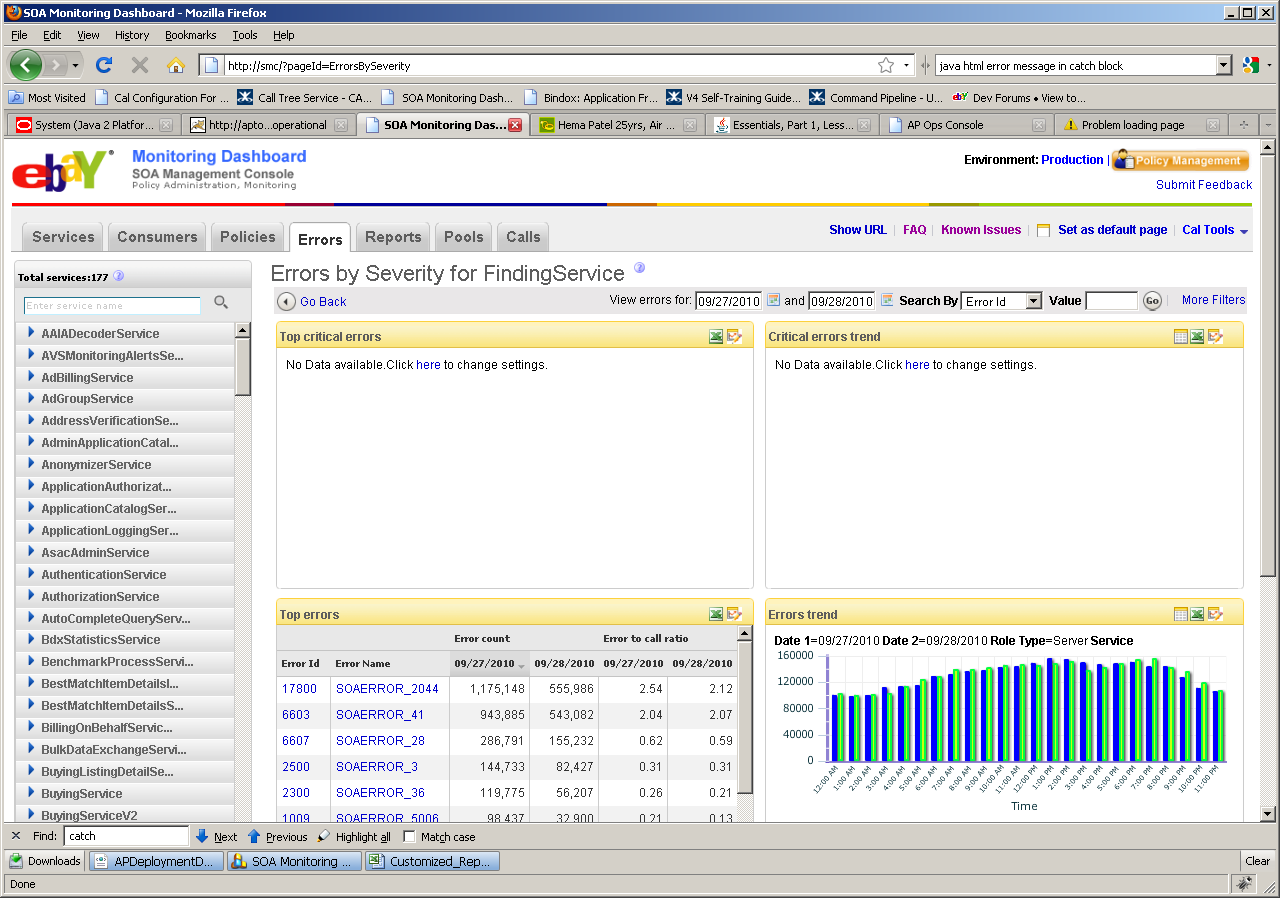
From page above Operations used –

1. Left selection panel -> getMetricsMetadata to get list of services
2. Top Application Errors -> getErrorMetricsDataRequest
3. Top Request Errors -> getErrorMetricsDataRequest
4. Request Error Trend ->getErrorGraphRequest
5. Application Error Trend ->getErrorGraphRequest



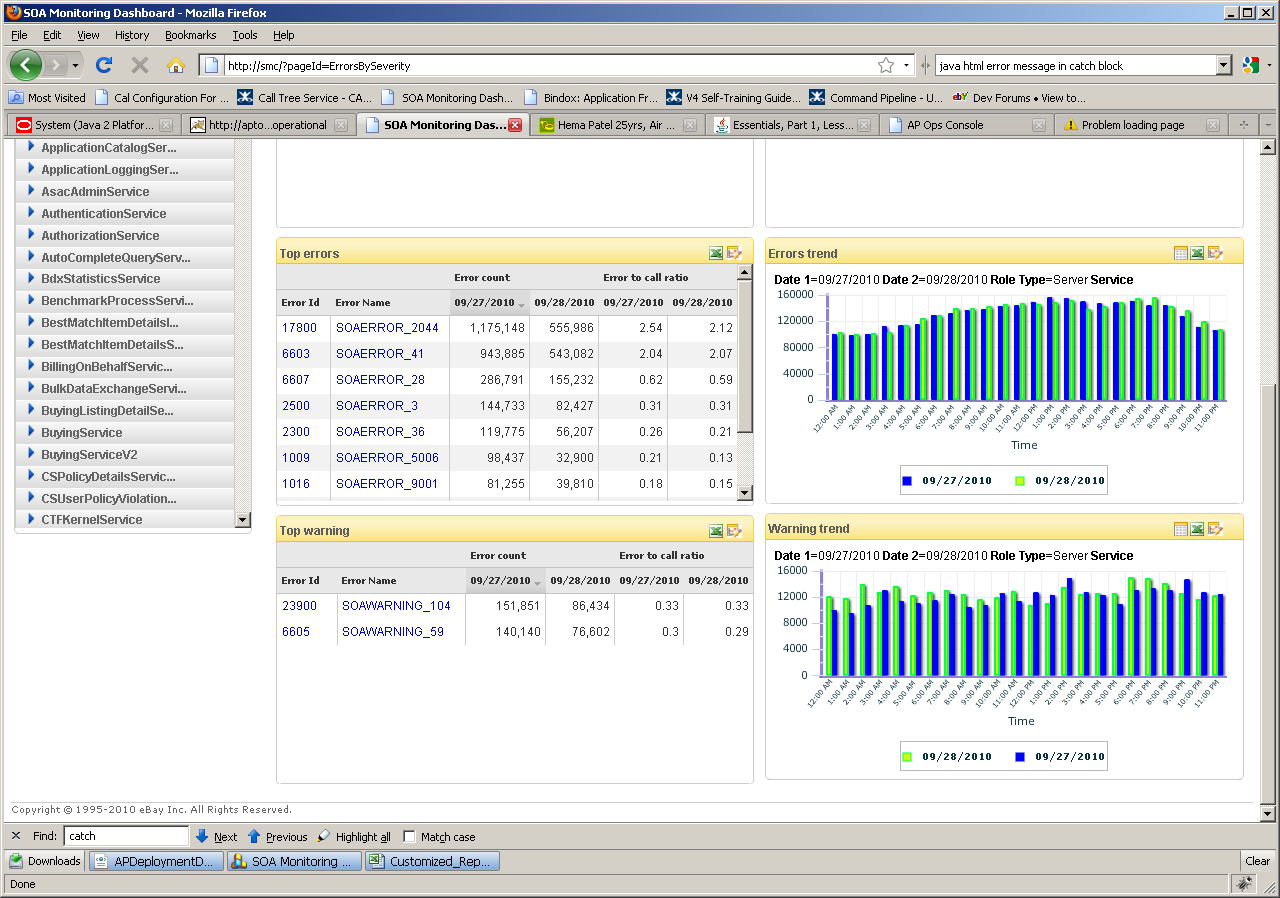
1. Top System Errors -> getErrorMetricsDataRequest
2. System Error Trend ->getErrorGraphRequest

#### SOA Errors Page – BySeverity



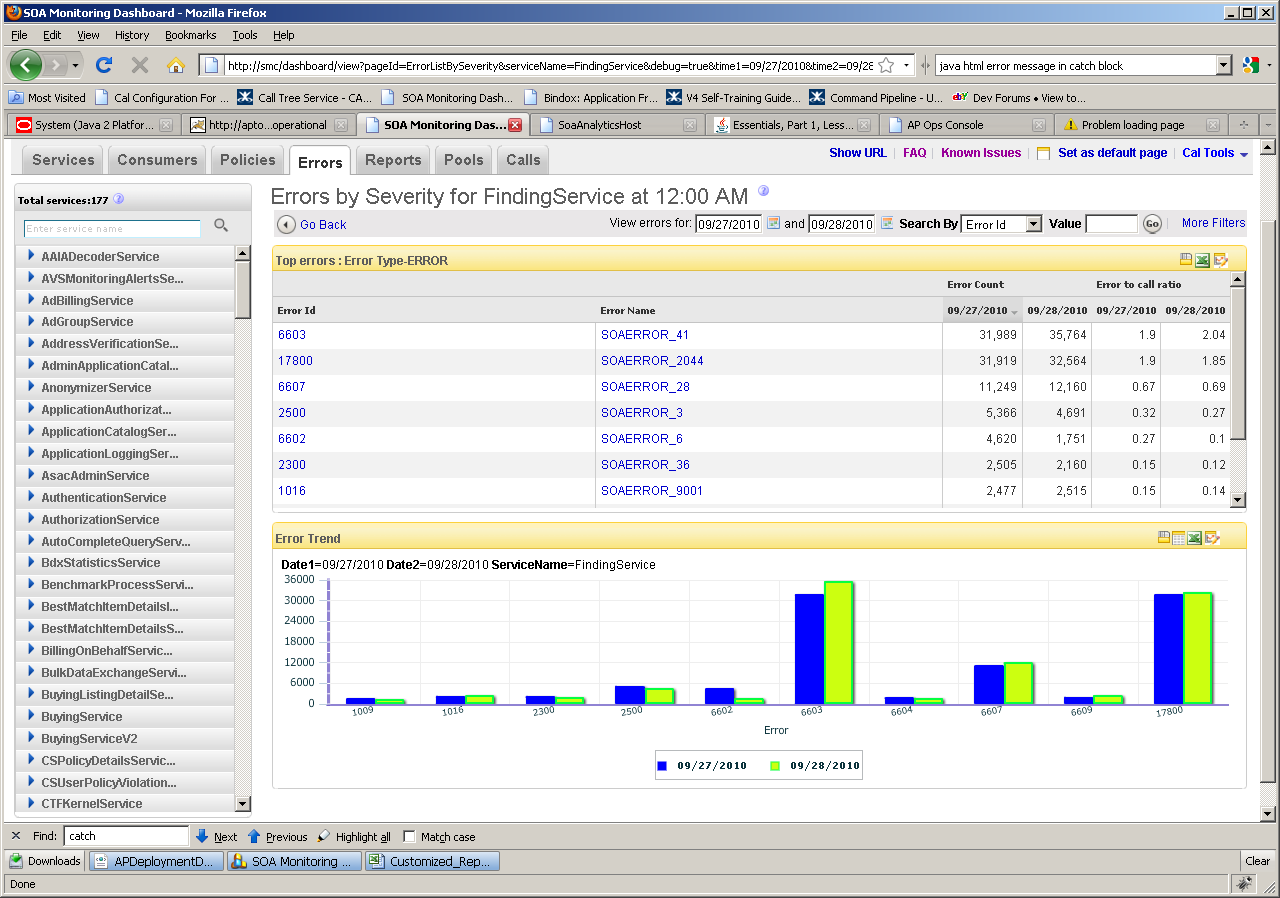
From page above Operations used –

1. Left selection panel -> getMetricsMetadata to get list of services
2. Top Critical Errors -> getErrorMetricsDataRequest
3. Top Errors -> getErrorMetricsDataRequest
4. Critical Error Trend ->getErrorGraphRequest
5. Error Trend ->getErrorGraphRequest



1. Top warning -> getErrorMetricsDataRequest
2. Warning Trend ->getErrorGraphRequest

#### DRILL DOWN – ERROR PAGE 1

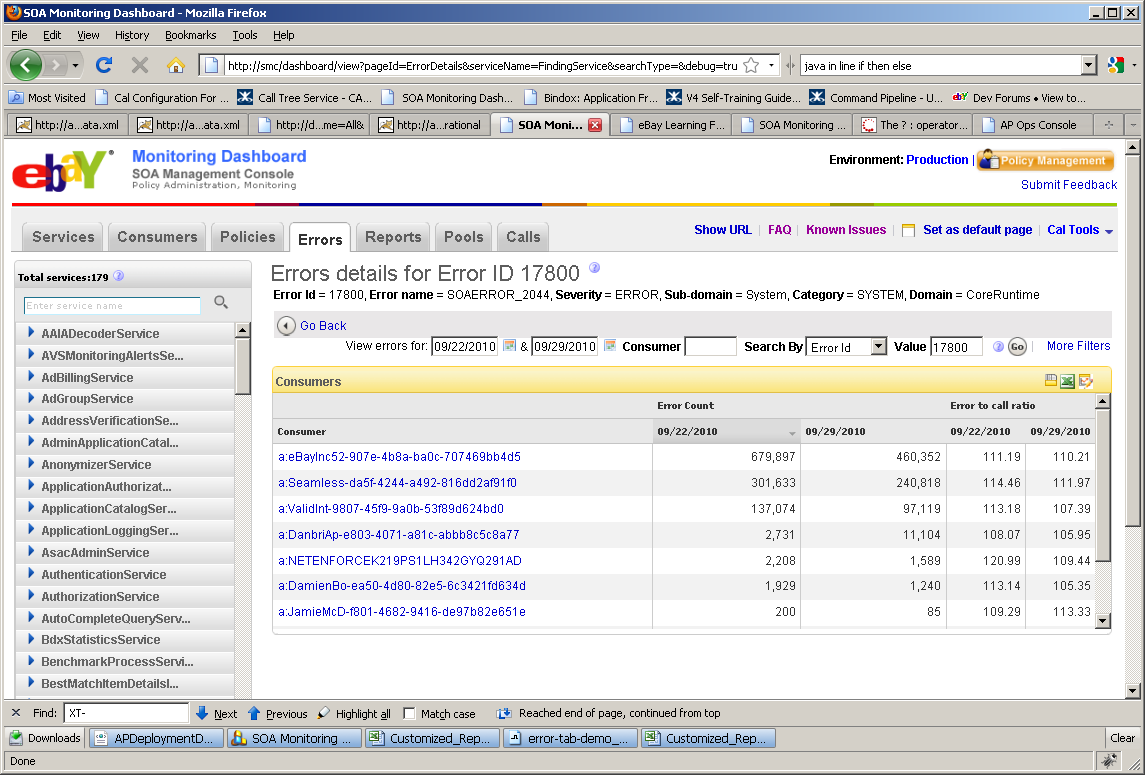


From page above Operations used –

1. Left selection panel -> getMetricsMetadata to get list of services
2. Top Errors -> getErrorMetricsDataRequest
3. Error Trend ->getErrorMetricsDataRequest

#### DRILL DOWN – ERROR PAGE 2

Clicking on the data point on the graph or the error name/id on the table will take u to the final level drill down, which shows the consumer level counts.



From page above Operations used –

1. Left selection panel -> getMetricsMetadata to get list of services
2. Header ->getErrorMetricsMetaDataRequest
3. Table -> getErrorMetricsDataRequest

#### DataModel for Errors

**SOA\_1H\_ERROR\_METRIC**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data Type | Default Value | Mandatory | Comment |
| CAL\_CUBE\_ID | NUMBER(38) |  | X | CAL Cube ID, the number(long) represents the time the Cube was generated. |
| SOA\_DIM\_SERVICE\_ID | NUMBER(38) |  | X | service id, refer to SOA\_SERVICE table |
| POOL\_ID | NUMBER(38) |  | X | pool id, refer to CAL\_DIM\_POOL table |
| SOA\_DIM\_OPERATION\_ID | NUMBER(38) |  | X | operation id, refer to SOA\_DIM\_OPERATION table |
| SOA\_DIM\_CONSUMER\_ID | NUMBER(38) |  | X | consumer id, refer to SOA\_USECASE table |
| PARTITION\_KEY | NUMBER(4) | mod(floor((trunc(sysdate) - to\_date('03-JAN-2005', 'DD-MON-YYYY'))/7), 6) | X | Day table - 6 month retention 8 monthly partitions mod 8 of months going forward, values 0 - 7 Oracle Syntax:  mod( TO\_NUMBER(TO\_CHAR(sysdate,'YYYY'))\*12 + to\_number(to\_char(sysdate, 'MM')), 8) |
| SOA\_DIM\_ERROR\_ID | NUMBER(38) |  | X | error id refer to SOA\_DIM\_ERROR |
| ERROR\_COUNT | NUMBER(38) |  | X | error value |
| ROLE\_TYPE | NUMBER(1) |  |  | 0 mean client, 1 mean server |
| CREATION\_DATE | DATE |  | X | Date row was created. Standard DB server timezone (PST) |
| SEVERITY | VARCHAR2(256) |  | X | error or warning |
| CATEGORY | VARCHAR2(256) |  | X | possible category are system, request, application |

**SOA\_DIM\_ERROR**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data Type | Default Value | Mandatory | Comment |
| SOA\_DIM\_ERROR\_ID | NUMBER(38) |  | X | Primary key generated by SOA\_DIM\_ERROR\_SEQ. |
| NAME | VARCHAR2(256) |  |  | error name |
| CATEGORY | VARCHAR2(256) |  |  | possible catergory are system, request, application |
| DOMAIN | VARCHAR2(256) |  |  | domain name for this error |
| SUB\_DOMAIN | VARCHAR2(256) |  |  | sub domain name for this error |
| SEVERITY | VARCHAR2(256) |  |  | error or warning |
| CREATION\_DATE | DATE |  | X | Date row was created. Standard DB server timezone (PST) |