ZAIDYN GENERIC OUTPUT API

Description

The purpose of this document is to provide details of API and objects for ZCE (ZAIDYN Customer Engagement) Output API to avoid manual efforts for fetching result data manually through S3 or Custom flows.

Guidelines for document

- This document contains examples of URL, parameters, payload body and result used in API as well as some validation to keep in mind while fetching objects.
- It contains all the Error Codes and their corresponding Error Messages that comes as the response in case of validation failures.

Prerequisite:

1. Datalake V2

   To enable datalake v2 on instance follow the following document.

   Step-By-Step guide to deploy DatalakeV2

2. Affinity and Data Setup

   Affinity Configuration for respective datalake needs to be done along with input data setup on s3 for respective datalake client bucket locations.

3. Transition to Basic Authentication

   - JWT tokens are no longer supported from this release onwards i.e. R26.
   - Basic Authentication with service account credentials should be used instead.
   - The x-actual-user attribute has been removed and is no longer required along with basic authentication.

ZCE Generic Output APIs:

1. Metadata API

Invoke GET API to get the metadata from the given object-name

GET	[baseUrl]/api/v1/data-lakes/[data-lake-id]/ objects/[object-name]/metadata

Parameters	
Name	**Description**
Params	data-lake-id : Datalake id for respective Datalake object-name: One of these - “archived_sugg”, “truncated_archived_sugg”, “nba_history”

Response Messages

Code	Description

200	Metadata for object-name. Detailed response here.
	Example:
	{
	"Name": "archived_sugg",
	"Columns": [
	{
	"Name": "guid",
	"Type": "string"
	},
	{
	"Name": " business_unit_id",
	"Type": "string"
	},
	"PartitionKeys": [
	{
	"Name": "generated_date",
	"Type": "string"
	},
	{
	"Name": "job_id",
	"Type": "string"
	}
	]
	}

2. Export API

Invoke POST API to Export data from given object-name

POST	[baseUrl]/api/v1/data-lakes/[data-lake-id]/objects/[object-name]/export

Parameters

Name	Description
Params	data-lake-id : Datalake id for respective Datalake object-name: One of these - “archived_sugg”, “truncated_archived_sugg”, “nba_history”

Body	Columns: List of columns to be exported. All columns at once (‘*’) is not applicable.
	filter: Filters rows. JSON Expression based. Click Here for details on how to use filters.
	pageNumber: The page number to be exported.
	pageSize: The number of rows to be exported.
	
	Example:
	{
	"Columns":["job_id"],
	"filter": {},
	"PageNumber":1,
	"PageSize":1000,
	"HasNext": true
	}

Response Messages

Code	Description
200	The data is exported. With PreSigned URL as response. The HasNext is recommended for the purpose of verifying existence of data on the subsequent page.

400	{
	"ErrorCode": "",
	"Message": ""
	}

404	{
	"ErrorCode": "",
	"Message": ""
	}

Validation Messages

Code	Description
1060	Invalid Object Name

1061	Invalid Page Number
1062	Invalid Page Size
1063	Invalid Column Name
1064	Empty Column Name
1065	Invalid Datalake Version
1066	Invalid Datalake Id

How to use filters:

1. Filter Expressions:

   Logical Operators:

   a. AND
   b. OR
   c. NOT

   Comparison Operators:

   d. > - Greater Than
   e. < - Less Than
   f. >= - Greater-Than Equals
   g. <= - Less-Than Equals
   h. = - Equals
   i. <> - Not-Equals
   j. LIKE - Like Pattern

   Others:

   k. IsNull - Is Null Expression
   l. col - Column Expression
   m. Literal - Literal Expression

Examples:

Literal Expression:

```
{
    "Literal":{
        "Value":"0" //numeric, boolean, string
    }
}
```

Column Expression:

```
{
    "col":{
        "Name":"job_id"
    }
}
```

IsNull Expression:

```
{
    "IsNull":{
        //Expression
    }
}
```

Not Expression:

```
{
    "Not":{
```

```
        //Expression
    }
}
```

Binary Operators (Logical or Comparison):

```
{
    "operator":[    //Logical or Comparison operators
        {
            //Expression
        },
        {
            //Expression
        }
    ]
}
```

Some more Examples:

```
{
    "AND":[
        {
            ">":[
                {
                    "col":{
                        "Name":"job_id"
                    }
                },
                {
                    "Literal":{
                        "Value":"1234"
                    }
                }
            ]
        },
        {
            "Literal":{
                "Value":false
            }
        }
    ]
}
```

```
{
    "Not":{
        "IsNull":{
            "col":{
                "Name":"guid"
            }
        }
    }
}
```

```
{
        "LIKE":[
            {
                "col":{
                    "Name" : "guid"
                }
            },
            {
                "Literal":
                {
                    "Value": "%fffebfaf%"
                }
            }
        ]
    }
```

2. Column Datatypes:
    a. Map
    b. Timestamp
    c. String
    d. Integer
    e. Boolean
    f. Struct

Examples:

Struct Type:

```
{
    "Columns":["guid","insights"],
    "filter":
    {
        "=":[
            {
                "col":{
                    "Name":"insights.insight_text"
                }
            },
            {
                "Literal":{
                    "Value":"some insight text"
                }
            }
        ]
    },
    "PageNumber":1,
    "PageSize":1000
}
```

Timestamp:

```
{
    "Columns":["action_id","generated_on"],
```

```
    "filter":
    {
        ">":[
            {
                "col":{
                    "Name":"generated_on"
                }
            },
            {
                "Literal":{
                    "Value":"CAST('2023-12-14' AS timestamp)"
                }
            }
        ]
    },
    "PageNumber":1,
    "PageSize":1000
}
```

Map Type:

To filter by key: Use keyword “key” after column name with map datatype

```
{
    "=":[
        {
            "col":{
                "Name":"suppressionconditionvalues.key"
            }
        },
        {
            "Literal":{
                "Value":"some_key"
            }
        }
    ]
}
```

To filter by value: Use keyword “value” after column name with map datatype

```
{
    "=":[
        {
            "col":{
                "Name":"suppressionconditionvalues.value"
            }
        },
        {
            "Literal":{
                "Value":"some_value"
            }
        }
    ]
}
```

Note: Filter by both key-value is not supported currently. Only filter by either key or by value is supported.

Supported Object-Names and their Columns:

1. archived_sugg


```
{
    "Name": "archived_sugg",
    "Columns": [
        {
            "Name": "guid",
            "Type": "string"
        },
        {
            "Name": "business_unit_id",
            "Type": "string"
        },
        {
            "Name": "pseudo_salesforce_id",
            "Type": "string"
        },
        {
            "Name": "entities",
            "Type":
"struct<key:array<string>,values:map<string,struct<id:string,type:string,external_id:string,name:string>>>"
        },
        {
            "Name": "business_category_sk",
            "Type": "string"
        },
        {
            "Name": "metric_group_sk",
            "Type": "string"
        },
        {
            "Name": "metric_group_type",
            "Type": "string"
        },
        {
            "Name": "rep_id",
            "Type": "string"
        },
        {
            "Name": "action_id",
            "Type": "string"
        },
        {
            "Name": "action_text",
            "Type": "string"
        },
        {
            "Name": "action_integration_key",
            "Type": "string"
        },
        {
            "Name": "cycle_time_id",
            "Type": "string"
        },
        {
            "Name": "expiry_date",
            "Type": "string"
        },
        {
            "Name": "sugg_posted_date",
            "Type": "string"
        },
        {
            "Name": "insights",
            "Type": "array<struct<insight_id:string,insight_text:string>>"
```

```
        },
        {
            "Name": "priority_score",
            "Type": "double"
        },
        {
            "Name": "survey_id",
            "Type": "string"
        },
        {
            "Name": "marketing_mapping",
            "Type": "struct<enabled:boolean,channel:string,contentType:string,contentCode:string>"
        },
        {
            "Name": "priority_vod",
            "Type": "string"
        },
        {
            "Name": "email_template",
            "Type":
"struct<Email_Template_ID_vod__c:string,Email_Template_Vault_ID_vod__c:string,Email_Template_vod__c:boolean>"
        },
        {
            "Name": "cml_result",
            "Type":
"map<string,struct<rawPrediction:array<double>,probability:array<double>,prediction:double,seed_model_accurac
y:double,ml_model_accuracy:double,seed_model_predicted_message:string,ml_model_predicted_message:string,selec
ted_msg_source:int>>"
        },
        {
            "Name": "prioritization_details",
            "Type":
"struct<normalized_rep_score:double,normalized_field_score:double,tie_breaker_value:double,priorityScoreBefor
eTieBreaker:double>"
        },
        {
            "Name": "parent_id",
            "Type": "string"
        },
        {
            "Name": "suppressionconditionvalues",
            "Type": "map<string,int>"
        },
        {
            "Name": "rationale",
            "Type": "array<string>"
        },
        {
            "Name": "suppressed",
            "Type": "boolean"
        },
        {
            "Name": "business_sub_category_sk",
            "Type": "string"
        },
        {
            "Name": "veevadonotdisplayonhomepage",
            "Type": "string"
        },
        {
            "Name": "veevadisplaydismiss",
            "Type": "string"
        },
        {
            "Name": "veevadisplaymarkascomplete",
            "Type": "string"
        },
        {
```

```
            "Name": "packageid",
            "Type": "string"
        },
        {
            "Name": "packageversionid",
            "Type": "string"
        },
        {
            "Name": "valuesetid",
            "Type": "string"
        }
    ],
    "PartitionKeys": [
        {
            "Name": "generated_date",
            "Type": "string"
        },
        {
            "Name": "job_id",
            "Type": "string"
        }
    ]
}
```

2. truncated_archived_sugg

```
{
    "Name": "truncated_archived_sugg",
    "Columns": [
        {
            "Name": "guid",
            "Type": "string"
        },
        {
            "Name": "business_unit_id",
            "Type": "string"
        },
        {
            "Name": "pseudo_salesforce_id",
            "Type": "string"
        },
        {
            "Name": "entities",
            "Type":
"struct<key:array<string>,values:map<string,struct<id:string,type:string,external_id:string,name:string>>>"
        },
        {
            "Name": "business_category_sk",
            "Type": "string"
        },
        {
            "Name": "metric_group_sk",
            "Type": "string"
        },
        {
```

```
            "Name": "metric_group_type",
            "Type": "string"
        },
        {
            "Name": "rep_id",
            "Type": "string"
        },
        {
            "Name": "action_id",
            "Type": "string"
        },
        {
            "Name": "action_text",
            "Type": "string"
        },
        {
            "Name": "action_integration_key",
            "Type": "string"
        },
        {
            "Name": "cycle_time_id",
            "Type": "string"
        },
        {
            "Name": "expiry_date",
            "Type": "string"
        },
        {
            "Name": "sugg_posted_date",
            "Type": "string"
        },
        {
            "Name": "insights",
            "Type": "array<struct<insight_id:string,insight_text:string>>"
        },
        {
            "Name": "priority_score",
            "Type": "double"
        },
        {
            "Name": "survey_id",
            "Type": "string"
        },
        {
            "Name": "marketing_mapping",
            "Type": "struct<enabled:boolean,channel:string,contentType:string,contentCode:string>"
        },
        {
            "Name": "priority_vod",
            "Type": "string"
```

```
        },
        {
            "Name": "email_template",
            "Type":
"struct<Email_Template_ID_vod__c:string,Email_Template_Vault_ID_vod__c:string,Email_Template_vod__c:boolean>"
        },
        {
            "Name": "cml_result",
            "Type":
"map<string,struct<rawPrediction:array<double>,probability:array<double>,prediction:double,seed_model_accurac
y:double,ml_model_accuracy:double,seed_model_predicted_message:string,ml_model_predicted_message:string,selec
ted_msg_source:int>>"
        },
        {
            "Name": "prioritization_details",
            "Type":
"struct<normalized_rep_score:double,normalized_field_score:double,tie_breaker_value:double,priorityScoreBefor
eTieBreaker:double>"
        },
        {
            "Name": "parent_id",
            "Type": "string"
        },
        {
            "Name": "suppressionconditionvalues",
            "Type": "map<string,int>"
        },
        {
            "Name": "rationale",
            "Type": "array<string>"
        },
        {
            "Name": "suppressed",
            "Type": "boolean"
        },
        {
            "Name": "business_sub_category_sk",
            "Type": "string"
        },
        {
            "Name": "veevadonotdisplayonhomepage",
            "Type": "string"
        },
        {
            "Name": "veevadisplaydismiss",
            "Type": "string"
        },
        {
            "Name": "veevadisplaymarkascomplete",
            "Type": "string"
```

```
        },
        {
            "Name": "packageid",
            "Type": "string"
        },
        {
            "Name": "packageversionid",
            "Type": "string"
        },
        {
            "Name": "valuesetid",
            "Type": "string"
        }
    ],
    "PartitionKeys": [
        {
            "Name": "generated_date",
            "Type": "string"
        },
        {
            "Name": "job_id",
            "Type": "string"
        }
    ]
}
```

3. nba_history

```
{
    "Name": "nba_history",
    "Columns": [
        {
            "Name": "4cinputs",
            "Type":
"array<struct<channel_id:string,content_id:string,content_type:string,content_type_id:string,content_type_nam
e:string,cadence:string,range:int,trigger_name:string,rule_id:string>>"
        },
        {
            "Name": "action_id",
            "Type": "string"
        },
        {
            "Name": "business_unit_id",
            "Type": "string"
        },
        {
            "Name": "channel",
            "Type": "string"
```

```
        },
        {
            "Name": "channellastdate",
            "Type": "map<string,timestamp>"
        },
        {
            "Name": "content",
            "Type": "string"
        },
        {
            "Name": "contenttype",
            "Type": "struct<id:string,name:string>"
        },
        {
            "Name": "content_recommendation_array",
            "Type":
"array<struct<channelId:string,date:timestamp,contentId:string,content_type_recommended:struct<id:string,name
:string>,rationale:string>>"
        },
        {
            "Name": "date_format",
            "Type": "string"
        },
        {
            "Name": "fixed_event_name",
            "Type": "string"
        },
        {
            "Name": "customer_id",
            "Type": "string"
        },
        {
            "Name": "mloutput",
            "Type":
"struct<recommendation:array<struct<DATE:string,RATIONALE:string,CHANNEL:string,CONTENT:string>>,score:double
>"
        },
        {
            "Name": "ml_flag",
            "Type": "string"
        },
        {
            "Name": "ml_rationale",
            "Type": "struct<rationaleKey:string,triggerName:string,fixedEventName:string>"
        },
        {
            "Name": "product_id",
            "Type": "string"
        },
        {
```

```
            "Name": "recommendation_period",
            "Type": "int"
        },
        {
            "Name": "rule_id",
            "Type": "string"
        },
        {
            "Name": "rule_name",
            "Type": "string"
        },
        {
            "Name": "sequence_length",
            "Type": "int"
        },
        {
            "Name": "rationale",
            "Type": "struct<key:string,text:string>"
        },
        {
            "Name": "generated_on",
            "Type": "timestamp"
        },
        {
            "Name": "channeltouchpoints",
            "Type": "array<struct<CHANNEL:string,CONTENT:string,MAX_ALLOWED:string,GRANULARITY:string>>"
        },
        {
            "Name": "min_max_gap_config",
            "Type":
"array<struct<FROM_CHANNEL:string,FROM_CONTENT:string,TO_CHANNEL:string,TO_CONTENT:string,MIN_GAP:string,MAX_
GAP:string>>"
        },
        {
            "Name": "past_events",
            "Type": "array<struct<CHANNEL:string,DATE:string,CONTENT:string>>"
        },
        {
            "Name": "marketing_mapping_inputs",
            "Type":
"array<struct<CHANNEL:string,CONTENT:string,CONTENT_TYPE:string,DATE:string,RANGE:int,FIXED_EVENT_NAME:string
,IS_REQUIRED_SEQ:boolean,ACTION:string,REQ_SEQ_CHANNEL:string,SOURCE:string>>"
        },
        {
            "Name": "dimension1_values",
            "Type": "array<string>"
        },
        {
            "Name": "dimension2_values",
            "Type": "array<string>"
```

```
        },
        {
            "Name": "dimensions",
            "Type": "array<array<string>>"
        },
        {
            "Name": "mloutput_3d",
            "Type":
"struct<recommendation:array<struct<DATE:string,DIMENSION1:string,DIMENSION2:string,RATIONALE:string,SOURCE:s
tring>>,score:double>"
        },
        {
            "Name": "content_recommendation_array_3d",
            "Type":
"array<struct<date:timestamp,rationale:string,dimension1:string,dimension2:string,source:string,channelId:str
ing,contentType:string,product:string,indication:string,contentId:string,content_type_recommended:struct<id:s
tring,name:string>>>"
        },
        {
            "Name": "marketing_mapping_inputs_3d",
            "Type":
"array<struct<DIMENSION1:string,DIMENSION2:string,CONTENT:string,DATE:string,SOURCE:string,RANGE:int,FIXED_EV
ENT_NAME:string,IS_REQUIRED_SEQ:boolean,REQ_SEQ_CHANNEL:string,ACTION:string>>"
        },
        {
            "Name": "fixed_events_3d",
            "Type":
"array<struct<DIMENSION1:string,DIMENSION2:string,CONTENT:string,DATE:string,SOURCE:string,RANGE:int,FIXED_EV
ENT_NAME:string,IS_REQUIRED_SEQ:boolean,REQ_SEQ_CHANNEL:string,ACTION:string>>"
        },
        {
            "Name": "flexi_events_3d",
            "Type":
"array<struct<DIMENSION1:string,DIMENSION2:string,CONTENT:string,DATE:string,SOURCE:string,RANGE:int,FIXED_EV
ENT_NAME:string,IS_REQUIRED_SEQ:boolean,REQ_SEQ_CHANNEL:string,ACTION:string>>"
        },
        {
            "Name": "blackout_dates_3d",
            "Type": "array<struct<DIMENSION1:string,DIMENSION2:string,DATE:string,RANGE:int>>"
        },
        {
            "Name": "max_allowed_touchpoints",
            "Type":
"array<struct<DIMENSION_VALUE:string,MIN_ALLOWED:int,MAX_ALLOWED:int,GRANULARITY:string,DIMENSION_TYPE:string
>>"
        },
        {
            "Name": "hybrid_max_allowed_touchpoints",
            "Type": "array<struct<DIMENSION1:string,DIMENSION2:string,MIN_ALLOWED:int,GRANULARITY:string>>"
        },
```

```
        {
            "Name": "min_max_gap_3d",
            "Type":
"array<struct<FROM_DIMENSION1:string,FROM_DIMENSION2:string,TO_DIMENSION1:string,TO_DIMENSION2:string,MIN_GAP
:int,MAX_GAP:int>>"
        },
        {
            "Name": "fixed_events",
            "Type":
"array<struct<CHANNEL:string,CONTENT:string,CONTENT_TYPE:string,DATE:string,RANGE:int,FIXED_EVENT_NAME:string
,IS_REQUIRED_SEQ:boolean,ACTION:string,REQ_SEQ_CHANNEL:string,SOURCE:string>>"
        },
        {
            "Name": "blackout_dates",
            "Type": "array<struct<CHANNEL:string,DATE:string,CONTENT:string,RANGE:int>>"
        },
        {
            "Name": "deltaoldnewflag",
            "Type": "int"
        },
        {
            "Name": "indication",
            "Type": "string"
        },
        {
            "Name": "packageid",
            "Type": "string"
        },
        {
            "Name": "packageversionid",
            "Type": "string"
        },
        {
            "Name": "datasourcekey",
            "Type": "string"
        },
        {
            "Name": "status",
            "Type": "int"
        },
        {
            "Name": "cadence",
            "Type": "timestamp"
        },
        {
            "Name": "refresh_date",
            "Type": "string"
        },
        {
            "Name": "start_date",
```

```
            "Type": "string"
        }
    ],
    "PartitionKeys": [
        {
            "Name": "job_id",
            "Type": "string"
        }
    ]
}
```