Configuration wizard

Interface between front end and service layer

**Rules:**

1. Request in CGI POST format with query string name value pairs.
2. Response in JSON format.
3. action,order,direction key words are prefixed with underscore \_ to prevent conflict with field/property names
4. Send response with: response.setContentType("application/json; charset=utf-8");
5. All lower case
6. Simple types: strings and numbers. Do we need to pass True and False? The table stores 0 and 1. YES
7. Property names are the same as column names
8. Form field ID same as fully qualified field name. Example: bbb\_list\_display\_name
9. JSP page name should be the same as the table name.
10. Only the required actions need to be implemented.
11. Dataset returned as an array of objects
12. Default order by sequence\_num if available.
13. Only API requests that update data will be submitted using the ATG form handler form. These include insert, update, delete, xsequence, clone

**Keywords:**

1. \_action
   1. CRUD actions (select, insert, update, delete)
   2. Custom actions (xsequence,clone)
2. \_order - sort order column name or column index
3. \_direction – sort direction (asc=0, desc=1)

## BBB\_LIST

**SELECT list for single site :**

**REST URL :**

**/rest/model/com/admin/rest/ListOperationActor/getSiteLists?site\_id=2**

This request is needed to load the "Lists" node of the tree:

If the query value contains a %, then query using LIKE

There are two % in the site\_flag parameter one on each side of the value (contains)

**Sample SQL:** SELECT \* FROM BBB\_LIST WHERE SITE\_FLAG LIKE ‘%1%’ ORDER BY SEQUENCE\_NUM

If the request does not contain a second parameter then return all the Lists:

**bbb\_list.jsp?\_action=select**

**Sample SQL:** SELECT \* FROM BBB\_LIST ORDER BY SEQUENCE\_NUM

The default ORDER should be: ORDER BY SEQUENCE\_NUM

**JSON response:**

{

status:"ok",

data:[

{

"list\_id":1,

"type\_name":"REGISTRY",

"sub\_type\_code":"BRD",

"sub\_type\_name":"Type 1",

"display\_name":"List 1",

"sequence\_num":0,

"is\_disabled":0,

"allow\_duplicates":0,

"site\_flag":1,

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

},

{

"list\_id":10,

"type\_name":"REGISTRY",

"sub\_type\_code":"BRD",

"sub\_type\_name":"Type 1",

"display\_name":"List 10",

"sequence\_num":3,

"is\_disabled":0,

"allow\_duplicates":0,

"site\_flag":1,

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

}

]

}

## BBB\_LIST

**SELECT**

**REST URL:**

**/rest/model/com/admin/rest/ListOperationActor/getList?list\_id=2**

This request is needed to retrieve the list for edit.

The data is returned as an array even though only one row is returned. This is so that the format is consistent.

**Sample SQL:** SELECT \* FROM BBB\_LIST WHERE LIST\_ID = 1 ORDER BY SEQUENCE\_NUM

**JSON response:**

{

"status":"ok",

"data":[

{

"list\_id":1,

"type\_name":"REGISTRY",

"sub\_type\_code":"BRD",

"sub\_type\_name":"Type 1",

"display\_name":"List 1",

"sequence\_num":0,

"is\_disabled":0,

"allow\_duplicates":0,

"site\_flag":"1,2",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

}

]

}

**Sample error response:**

{

status:"error",

message:{

"id":20,

"description":"list not found"

}

}

## BBB\_LIST

**INSERT**

**REST URL:**

**/rest/model/com/admin/rest/ListOperationActor/createList?display\_name=Anniversary+list&allow\_duplicates=true&site\_flag=1,2,3&sub\_type\_code=ANN&is\_disabled=false&sequence\_num=0&lastModifiedUser=ADMINUI&user\_created=ADMINUI**

This request is needed to insert a new list

This request will be submitted using the ATG form handler for BBB\_LIST

The data is returned as an array even though only one row is returned. This is so that the format is consistent.

**JSON response:**

{

"status":"ok",

"data":[

{

"list\_id":1234,

"type\_name":"REGISTRY",

"sub\_type\_code":"BRD",

"sub\_type\_name":"Type 1",

"display\_name":"List 4",

"sequence\_num":1,

"is\_disabled":0,

"allow\_duplicates":0,

"site\_flag":"1,2",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

}

]

}

**Sample error response:**

{

status:"error",

message:{

"id":10,

"description":"invalid sub\_type\_code"

}

}

When this api is called, execute the following stored proc to set the list order.

bbb\_misc.bbb\_list\_xsequence( list\_id, sequence\_num, :rc)

The details of this stored proc can be found in the API call xsequence below.

## BBB\_LIST

**CLONE**

**bbb\_list.jsp?\_action=clone&list\_id&sub\_type\_code=1&display\_name=List+4+-+Copy&sequence\_num=4&is\_disabled=0&allow\_duplicates&site\_flag=12**

This request is needed to clone a list **(THIS FEATURE IS NOT NEED RIGHT NOW)**

Same as insert except the list\_id of the List that is to be cloned is included.

After the list is saved, all the categories from the source List should be copied to the new List.

This request will be submitted using the ATG form handler for BBB\_LIST

The data is returned as an array even though only one row is returned. This is so that the format is consistent.

**JSON response:**

{

"status":"ok",

"data":[

{

"list\_id":1235,

"type\_name":"REGISTRY",

"sub\_type\_code":"BRD",

"sub\_type\_name":"Type 1",

"display\_name":"List 4 - Copy",

"sequence\_num":1,

"is\_disabled":0,

"allow\_duplicates":0,

"site\_flag":"1,2",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

}

]

}

**Sample error response:**

{

status:"error",

message:{

"id":20,

"description":"list not found"

}

}

## BBB\_LIST

**UPDATE LIST**

**REST URL:**

**/rest/model/com/admin/rest/ListOperationActor/editList?list\_id=6&display\_name=Anniversary+list&allow\_duplicates=true&site\_flag=1,2,3&sub\_type\_code=ANN&is\_disabled=false&sequence\_num=1&lastModifiedUser=ADMINUI&user\_created=ADMINUI**

This request is needed to update a list

This request will be submitted using the ATG form handler for BBB\_LIST

The data is returned as an array even though only one row is returned. This is so that the format is consistent.

**JSON response:**

{

"status":"ok",

"data":[

{

"list\_id":1234,

"type\_name":"REGISTRY",

"sub\_type\_code":"BRD",

"sub\_type\_name":"Type 1",

"display\_name":"List 4",

"sequence\_num":1,

"is\_disabled":0,

"allow\_duplicates":0,

"site\_flag":"1,2",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

}

]

}

**Sample error response:**

{

status:"error",

message:{

"id":20,

"description":"list not found"

}

}

## BBB\_LIST

**DELETE LIST**

**REST URL:**

**/rest/model/com/admin/rest/ListOperationActor/removeList?list\_id=6**

This request is needed to delete a list

This request will be submitted using the ATG form handler for BBB\_LIST

**JSON response:**

{

"status":"ok"

}

**Sample error response:**

{

status:"error",

message:{

"id":20,

"description":"list not found"

}

}

## BBB\_LIST

**XSEQUENCE LIST**

**REST URL:**

**/rest/model/com/admin/rest/ListOperationActor/reorderList?list\_id=6&sequence\_num=1**

This request is needed to reorder the List

This request will be submitted using the ATG form handler for BBB\_LIST

**JSON response:**

{

"status":"ok"

}

**Sample error response:**

{

status:"error",

message:{

"id":20,

"description":"list not found"

}

}

When this api is called, execute the following stored proc to reorder the list:

CREATE OR REPLACE PROCEDURE bbb\_misc.bbb\_list\_xsequence(p\_list\_id IN VARCHAR2, p\_sequence\_num IN NUMBER, p\_cursor OUT SYS\_REFCURSOR)

AS

PRAGMA AUTONOMOUS\_TRANSACTION;

BEGIN

UPDATE bbb\_misc.bbb\_list a

SET sequence\_num = (SELECT rn FROM (

SELECT list\_id, row\_number()

OVER (ORDER BY CASE WHEN list\_id = p\_list\_id THEN p\_sequence\_num ELSE sequence\_num END,

CASE WHEN list\_id = p\_list\_id THEN (SELECT p\_sequence\_num-sequence\_num FROM bbb\_misc.bbb\_list WHERE list\_id = p\_list\_id) ELSE 0 END, list\_id) rn

FROM bbb\_misc.bbb\_list

)

WHERE list\_id = a.list\_id);

COMMIT;

OPEN p\_cursor FOR

SELECT list\_id, sequence\_num FROM bbb\_misc.bbb\_list

ORDER BY sequence\_num;

END;

/

## BBB\_LIST\_CHLDCAT

**SELECT**

**/rest/model/com/admin/rest/ListOperationActor/getCategories?list\_id=4**

This request is needed to return the categories assigned to a list

**JSON response:**

{

"status":"ok",

"data":[

{

"list\_id":1,

"list\_cat\_id":1,

"category\_url":"",

"image\_url":"",

"suggested\_qty":0,

"primary\_parent\_cat\_id":0,

"display\_name":"Category 1",

"threshold\_qty":0,

"threshold\_amt":0,

"service\_type\_cd":"",

"is\_disabled":0,

"baby\_category\_url":"",

"baby\_image\_url":"",

"ca\_category\_url":"",

"ca\_image\_url":"",

"tbs\_category\_url":"",

"tbs\_image\_url":"",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

},

{

"list\_id":1,

"list\_cat\_id":2,

"category\_url":"",

"image\_url":"",

"suggested\_qty":0,

"primary\_parent\_cat\_id":0,

"display\_name":"Category 2",

"threshold\_qty":0,

"threshold\_amt":0,

"service\_type\_cd":"",

"is\_disabled":0,

"baby\_category\_url":"",

"baby\_image\_url":"",

"ca\_category\_url":"",

"ca\_image\_url":"",

"tbs\_category\_url":"",

"tbs\_image\_url":"",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

},

{

"list\_id":1,

"list\_cat\_id":3,

"category\_url":"",

"image\_url":"",

"suggested\_qty":0,

"primary\_parent\_cat\_id":0,

"display\_name":"Category 3",

"threshold\_qty":0,

"threshold\_amt":0,

"service\_type\_cd":"",

"is\_disabled":1,

"baby\_category\_url":"",

"baby\_image\_url":"",

"ca\_category\_url":"",

"ca\_image\_url":"",

"tbs\_category\_url":"",

"tbs\_image\_url":"",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

}

]

}

**Sample error response:**

{

status:"error",

message:{

"id":20,

"description":"list not found"

}

}

## BBB\_LIST\_CHLDCAT

**INSERT**

**Endpoint url for adding category to list:**

**/rest/model/com/admin/rest/ListOperationActor/addCategory?list\_id=4&list\_cat\_id=1&sequence\_num=0**

This request is needed to add a category to the list

This request will be submitted using the ATG form handler for BBB\_LIST\_CHLD\_CAT

**JSON response:**

{

"status":"ok"

}

**Sample error response:**

{

status:"error",

message:{

"id":30,

"description":"list not found"

}

}

When this api is called, execute the following stored proc to set the list order.

bbb\_misc.bbb\_list\_chldcat\_xsequence( list\_id, sequence\_num, :rc)

The details of this stored proc can be found in the API call xsequence below.

## BBB\_LIST\_CHLDCAT

**DELETE**

**End point url for removing category :**

**/rest/model/com/admin/rest/ListOperationActor/removeCategory?list\_id=4&list\_cat\_id=1**

This request is needed to remove a category to the list

This request will be submitted using the ATG form handler for BBB\_LIST\_CHLDCAT

**JSON response:**

{

"status":"ok"

}

**Sample error response:**

{

status:"error",

message:{

"id":30,

"description":"category not found"

}

}

## BBB\_LIST\_CHLDCAT

**XSEQUENCE**

**Endpoint url :**

**/rest/model/com/admin/rest/ListOperationActor/reorderCategory?list\_id=4&list\_cat\_id=1&sequence\_num=1**

This request is needed to reorder a category in the list

This request will be submitted using the ATG form handler for BBB\_LIST\_CHLD\_CAT

**JSON response:**

{

"status":"ok"

}

**Sample error response:**

{

status:"error",

message:{

"id":30,

"description":"category not found"

}

}

When this api is called, execute the following stored proc to reorder the list:

CREATE OR REPLACE PROCEDURE bbb\_misc.bbb\_list\_chldcat\_xsequence(p\_list\_id IN VARCHAR2, p\_list\_cat\_id VARCHAR2, p\_sequence\_num IN NUMBER, p\_cursor OUT SYS\_REFCURSOR)

AS

PRAGMA AUTONOMOUS\_TRANSACTION;

BEGIN

UPDATE bbb\_misc.bbb\_list\_chldcat a

SET sequence\_num = (SELECT rn FROM (

SELECT list\_id, list\_cat\_id, row\_number()

OVER (ORDER BY CASE WHEN list\_id = p\_list\_id AND list\_cat\_id = p\_list\_cat\_id THEN p\_sequence\_num ELSE sequence\_num END,

CASE WHEN list\_id = p\_list\_id AND list\_cat\_id = p\_list\_cat\_id THEN (SELECT p\_sequence\_num-sequence\_num FROM bbb\_misc.bbb\_list\_chldcat WHERE list\_id = p\_list\_id AND list\_cat\_id = p\_list\_cat\_id) ELSE 0 END, list\_id, list\_cat\_id) rn

FROM bbb\_misc.bbb\_list\_chldcat

WHERE list\_id = p\_list\_id

)

WHERE list\_id = a.list\_id

AND list\_cat\_id = a.list\_cat\_id)

WHERE list\_id = p\_list\_id;

COMMIT;

OPEN p\_cursor FOR

SELECT list\_id, list\_cat\_id, sequence\_num FROM bbb\_misc.bbb\_list\_chldcat

WHERE list\_id = p\_list\_id

ORDER BY sequence\_num;

END;

/

## BBB\_LIST\_TYPE

**SELECT LISTTYPES**

**REST URL:**

**/rest/model/com/admin/rest/ListOperationActor/getListType**

Returns the list types

**Sample SQL:** SELECT \* FROM BBB\_LIST\_TYPE ORDER BY TYPE\_NAME, SUB\_TYPE\_NAME

**JSON response:**

{

status:"ok",

data:[

{

"type\_id":1,

"type\_name":"REGISTRY",

"sub\_type\_code":"BRD",

"sub\_type\_name":"Wedding",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

},

{

"type\_id":2,

"type\_name":"REGISTRY",

"sub\_type\_code":"ANN",

"sub\_type\_name":"Anniversary",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

},

{

"type\_id":3,

"type\_name":"REGISTRY",

"sub\_type\_code":"HSW",

"sub\_type\_name":"Housewarming",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

},

{

"type\_id":4,

"type\_name":"REGISTRY",

"sub\_type\_code":"COL",

"sub\_type\_name":"College",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

},

{

"type\_id":5,

"type\_name":"REGISTRY",

"sub\_type\_code":"BIR",

"sub\_type\_name":"Birthday",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

},

{

"type\_id":6,

"type\_name":"REGISTRY",

"sub\_type\_code":"RET",

"sub\_type\_name":"Retirement",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

},

{

"type\_id":7,

"type\_name":"REGISTRY",

"sub\_type\_code":"PNH",

"sub\_type\_name":"Pack and Hold",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

},

{

"type\_id":8,

"type\_name":"REGISTRY",

"sub\_type\_code":"OTH",

"sub\_type\_name":"Pack and Hold",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

},

{

"type\_id":9,

"type\_name":"REGISTRY",

"sub\_type\_code":"BA1",

"sub\_type\_name":"Baby",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

},

{

"type\_id":10,

"type\_name":"SERVICES",

"sub\_type\_code":"AAA",

"sub\_type\_name":"Roadside Service",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

},

{

"type\_id":11,

"type\_name":"PROMOTIONS",

"sub\_type\_code":"BB",

"sub\_type\_name":"Big Blue",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

}

]

}

## BBB\_LIST\_CATEGORY

**/rest/model/com/admin/rest/CategoryOperationActor/searchByName?display\_name=B%**

This request is needed for category search.

If the query value contains a %, then query using LIKE

There is only one % at the end of the display\_name parameter in this case (starts with)

The default sort order for this request should be ORDER BY DISPLAY\_NAME

**Sample SQL:** SELECT \* FROM BBB\_LIST\_CATEGORY WHERE DISPLAY\_NAME LIKE ‘B%’ ORDER BY DISPLAY\_NAME

**JSON response:**

{

"status":"ok",

"data":[

{

"list\_cat\_id":10,

"category\_url":"",

"image\_url":"",

"suggested\_qty":0,

"primary\_parent\_cat\_id":1,

"primary\_parent\_cat\_display\_name":"Primary Category 1",

"display\_name":"Bedding",

"threshold\_qty":0,

"threshold\_amt":0,

"service\_type\_cd":"",

"is\_disabled":0,

"baby\_category\_url":"",

"baby\_image\_url":"",

"ca\_category\_url":"",

"ca\_image\_url":"",

"tbs\_category\_url":"",

"tbs\_image\_url":"",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

},

{

"list\_cat\_id":20,

"category\_url":"",

"image\_url":"",

"suggested\_qty":0,

"primary\_parent\_cat\_id":1,

"primary\_parent\_cat\_display\_name":"Primary Category 1",

"display\_name":"Bath",

"threshold\_qty":0,

"threshold\_amt":0,

"service\_type\_cd":"",

"is\_disabled":0,

"baby\_category\_url":"",

"baby\_image\_url":"",

"ca\_category\_url":"",

"ca\_image\_url":"",

"tbs\_category\_url":"",

"tbs\_image\_url":"",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

},

{

"list\_cat\_id":30,

"category\_url":"",

"image\_url":"",

"suggested\_qty":0,

"primary\_parent\_cat\_id":1,

"primary\_parent\_cat\_display\_name":"Primary Category 1",

"display\_name":"Kitchen",

"threshold\_qty":0,

"threshold\_amt":0,

"service\_type\_cd":"",

"is\_disabled":0,

"baby\_category\_url":"",

"baby\_image\_url":"",

"ca\_category\_url":"",

"ca\_image\_url":"",

"tbs\_category\_url":"",

"tbs\_image\_url":"",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

}

]

}

Alternate version

**bbb\_list\_category.jsp?\_action=select&list\_cat\_id=1%25**

**Sample SQL:** SELECT \* FROM BBB\_LIST\_CATEGORY WHERE LIST\_CAT\_ID LIKE ‘1%’ ORDER BY LIST\_CAT\_ID

## BBB\_LIST\_CATEGORY

Getting single category

**REST URL:**

**/rest/model/com/admin/rest/CategoryOperationActor/getCategory?category\_id=1**

This request is needed to retrieve the category for edit.

The data is returned as an array even though only one row is returned. This is so that the format is consistent.

**Sample SQL:** SELECT \* FROM BBB\_LIST\_CATEGORY WHERE LIST\_CAT\_ID = 10

**JSON response:**

{

"status":"ok",

"data":[

{

"list\_cat\_id":10,

"category\_url":"",

"image\_url":"",

"suggested\_qty":0,

"primary\_parent\_cat\_id":1,

"primary\_parent\_cat\_display\_name":"Primary Category 1",

"display\_name":"Bedding",

"threshold\_qty":0,

"threshold\_amt":0,

"service\_type\_cd":"",

"is\_disabled":0,

"baby\_category\_url":"",

"baby\_image\_url":"",

"ca\_category\_url":"",

"ca\_image\_url":"",

"tbs\_category\_url":"",

"tbs\_image\_url":"",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

}

]

}

## BBB\_LIST\_CATEGORY

**INSERT**

**REST URL:**

**/rest/model/com/admin/rest/CategoryOperationActor/createCategory?&category\_url=example.com&image\_url=example.com&suggested\_qty=0&primary\_parent\_cat\_id=0&display\_name=New+Category&threshold\_qty=0&threshold\_amt=0&service\_type\_cd=1&is\_disabled=0&baby\_category\_url=example.com&baby\_image\_url=example.com&ca\_category\_url=example.com&ca\_image\_url=example.com&tbs\_category\_url=example.com&tbs\_image\_url=example.com**

This request is needed to insert a new category

This request will be submitted using the ATG form handler for BBB\_LIST\_CATEGORY

The data is returned as an array even though only one row is returned. This is so that the format is consistent.

If list\_id is not blank, then attach the new category to the list. (BBB\_LIST\_CHLDCAT)

If parent\_cat\_id is not blank then attach the new category to the parent category. (BBB\_LIST\_CAT\_CHLDCAT)

**JSON response:**

{

"status":"ok",

"data":[

{

"list\_cat\_id":1000,

"category\_url":"example.com",

"image\_url":"example.com",

"suggested\_qty":0,

"primary\_parent\_cat\_id":1,

"primary\_parent\_cat\_display\_name":"Primary Category 1",

"display\_name":"New Category",

"threshold\_qty":0,

"threshold\_amt":0,

"service\_type\_cd":"1",

"is\_disabled":0,

"baby\_category\_url":"example.com",

"baby\_image\_url":"example.com",

"ca\_category\_url":"example.com",

"ca\_image\_url":" example.com",

"tbs\_category\_url":"example.com",

"tbs\_image\_url":"example.com",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

}

]

}

**Sample error response:**

{

status:"error",

message:{

"id":50,

"description":"primary\_parent\_cat\_id not found"

}

}

## BBB\_LIST\_CATEGORY

**UPDATE Category**

**REST URL**

**/rest/model/com/admin/rest/CategoryOperationActor/editCategory?&list\_cat\_id=cat4011&category\_url=mandar.com&image\_url=example.com&suggested\_qty=0&primary\_parent\_cat\_id=0&display\_name=New+Category&threshold\_qty=0&threshold\_amt=0&service\_type\_cd=1&is\_disabled=0&baby\_category\_url=example.com&baby\_image\_url=example.com&ca\_category\_url=example.com&ca\_image\_url=example.com&tbs\_category\_url=example.com&tbs\_image\_url=example.com&lastModifiedUser=ADMINUI&user\_created=ADMINUI&category\_name=hello**

This request is needed to update a new category

This request will be submitted using the ATG form handler for BBB\_LIST\_CATEGORY

The data is returned as an array even though only one row is returned. This is so that the format is consistent.

**JSON response:**

{

"status":"ok",

"data":[

{

"list\_cat\_id":1000,

"category\_url":"example.com",

"image\_url":"example.com",

"suggested\_qty":0,

"primary\_parent\_cat\_id":1,

"primary\_parent\_cat\_display\_name":"Primary Category 1",

"display\_name":"New Category",

"threshold\_qty":0,

"threshold\_amt":0,

"service\_type\_cd":"1",

"is\_disabled":0,

"baby\_category\_url":"example.com",

"baby\_image\_url":"example.com",

"ca\_category\_url":"example.com",

"ca\_image\_url":" example.com",

"tbs\_category\_url":"example.com",

"tbs\_image\_url":"example.com",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

}

]

}

**Sample error response:**

{

status:"error",

message:{

"id":50,

"description":"category not found"

}

}

## BBB\_LIST\_CATEGORY

**DELETE CATGEORY**

**REST URL**

**/rest/model/com/admin/rest/CategoryOperationActor/removeCategory?&list\_cat\_id=cat4011**

This request is needed to delete a category

This request will be submitted as a POST using the ATG form handler for BBB\_LIST\_CATEGORY

For deleting a category, we will use the bbb\_list\_category.jsp and submit the form in it by just passing value of input field ‘bbb\_list\_category\_list\_cat\_id’.

**JSON response:**

{

"status":"ok"

}

**Sample error response:**

{

status:"error",

message:{

"id":50,

"description":"category not found"

}

}

## BBB\_LIST\_CAT\_CHLDCAT

## BBB\_LIST\_CAT\_CHLDCAT

**SELECT Child categories of a Category**

**REST URL:**

**/rest/model/com/admin/rest/CategoryOperationActor/getChildCategorieslist\_cat\_id=4**

This request is needed to return the categories assigned to a parent category

**JSON response:**

{

"status":"ok",

"data":[

{

"list\_cat\_id":1,

"child\_list\_cat\_id":1,

"category\_url":"",

"image\_url":"",

"suggested\_qty":0,

"primary\_parent\_cat\_id":0,

"display\_name":"Category 1",

"threshold\_qty":0,

"threshold\_amt":0,

"service\_type\_cd":"",

"is\_disabled":0,

"baby\_category\_url":"",

"baby\_image\_url":"",

"ca\_category\_url":"",

"ca\_image\_url":"",

"tbs\_category\_url":"",

"tbs\_image\_url":"",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

},

{

"list\_cat\_id":1,

"child\_list\_cat\_id":2,

"category\_url":"",

"image\_url":"",

"suggested\_qty":0,

"primary\_parent\_cat\_id":0,

"display\_name":"Category 2",

"threshold\_qty":0,

"threshold\_amt":0,

"service\_type\_cd":"",

"is\_disabled":0,

"baby\_category\_url":"",

"baby\_image\_url":"",

"ca\_category\_url":"",

"ca\_image\_url":"",

"tbs\_category\_url":"",

"tbs\_image\_url":"",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

},

{

"list\_cat\_id":1,

"child\_list\_cat\_id":3,

"category\_url":"",

"image\_url":"",

"suggested\_qty":0,

"primary\_parent\_cat\_id":0,

"display\_name":"Category 3",

"threshold\_qty":0,

"threshold\_amt":0,

"service\_type\_cd":"",

"is\_disabled":1,

"baby\_category\_url":"",

"baby\_image\_url":"",

"ca\_category\_url":"",

"ca\_image\_url":"",

"tbs\_category\_url":"",

"tbs\_image\_url":"",

"create\_user":"user1",

"create\_date":"01/01/2016",

"last\_mod\_user":"user1",

"last\_mod\_date":"01/01/2016"

}

]

}

**Sample error response:**

{

status:"error",

message:{

"id":20,

"description":"Categories not found"

}

}

## BBB\_LIST\_CAT\_CHLDCAT

**INSERT**

**/rest/model/com/admin/rest/CategoryOperationActor/addCategory?list\_id=4&list\_cat\_id=1&sequence\_num=0**

This request is needed to add a category to a parent category

This request will be submitted using the ATG form handler for BBB\_LIST\_CAT\_CHLD\_CAT

**JSON response:**

{

"status":"ok"

}

**Sample error response:**

{

status:"error",

message:{

"id":30,

"description":"list not found"

}

}

When this api is called, execute the following stored proc to set the list order.

bbb\_misc.bbb\_list\_cat\_chldcat\_xsequence( list\_id, sequence\_num, :rc)

The details of this stored proc can be found in the API call xsequence below.

## BBB\_LIST\_CAT\_CHLDCAT

**DELETE**

**End point url :**

**/rest/model/com/admin/rest/CategoryOperationActor/deleteCategory?list\_cat\_id=4&list\_chldcat\_id=1**

This request is needed to remove a category from a parent category

This request will be submitted using the ATG form handler for BBB\_LIST\_CAT\_CHLDCAT

**JSON response:**

{

"status":"ok"

}

**Sample error response:**

{

status:"error",

message:{

"id":30,

"description":"category not found"

}

}

## BBB\_LIST\_CAT\_CHLDCAT

**xsequence**

**/rest/model/com/admin/rest/CategoryOperationActor/reOrder?list\_cat\_id=4&list\_chldcat\_id=1&sequence\_num=1**

This request is needed to reorder a category under a parent category

This request will be submitted using the ATG form handler for BBB\_LIST\_CAT\_CHLD\_CAT

**JSON response:**

{

"status":"ok"

}

**Sample error response:**

{

status:"error",

message:{

"id":30,

"description":"category not found"

}

}

When this api is called, execute the following stored proc to reorder the list:

CREATE OR REPLACE PROCEDURE bbb\_misc.bbb\_list\_cat\_chldcat\_xsequence(p\_list\_cat\_id IN VARCHAR2, p\_child\_list\_cat\_id VARCHAR2, p\_sequence\_num IN NUMBER, p\_cursor OUT SYS\_REFCURSOR)

AS

PRAGMA AUTONOMOUS\_TRANSACTION;

BEGIN

IF (p\_child\_list\_cat\_id = 0) THEN

UPDATE bbb\_misc.bbb\_list\_cat\_chldcat SET sequence\_num = 0

WHERE list\_cat\_id = p\_list\_cat\_id;

END IF;

UPDATE bbb\_misc.bbb\_list\_cat\_chldcat a

SET sequence\_num = (SELECT rn FROM (

SELECT list\_cat\_id, child\_list\_cat\_id, row\_number()

OVER (ORDER BY CASE WHEN list\_cat\_id = p\_list\_cat\_id AND child\_list\_cat\_id = p\_child\_list\_cat\_id THEN p\_sequence\_num ELSE sequence\_num END,

CASE WHEN list\_cat\_id = p\_list\_cat\_id AND child\_list\_cat\_id = p\_child\_list\_cat\_id THEN (SELECT p\_sequence\_num-sequence\_num FROM bbb\_misc.bbb\_list\_cat\_chldcat WHERE list\_cat\_id = p\_list\_cat\_id AND child\_list\_cat\_id = p\_child\_list\_cat\_id) ELSE 0 END, list\_cat\_id, child\_list\_cat\_id) rn

FROM bbb\_misc.bbb\_list\_cat\_chldcat

WHERE list\_cat\_id = p\_list\_cat\_id

)

WHERE list\_cat\_id = a.list\_cat\_id

AND child\_list\_cat\_id = a.child\_list\_cat\_id)

WHERE list\_cat\_id = p\_list\_cat\_id;

COMMIT;

OPEN p\_cursor FOR

SELECT list\_cat\_id, child\_list\_cat\_id, sequence\_num FROM bbb\_misc.bbb\_list\_cat\_chldcat

WHERE list\_cat\_id = p\_list\_cat\_id

ORDER BY sequence\_num;

END;

/