

The Essbase Cloud REST API: What We Know Now



ODTUG
Kscope19 
SEATTLE, WASHINGTON • JUNE 23-27

**PLEASE FILL OUT
YOUR EVALUATIONS**

SEATTLE

 Washington State
Convention Center



Agenda

- Introductions
- REST
- Swagger
- Scripting



Vijay Kurian

- ◎ Huron Consulting Group
- ◎ 15+ years
EPM Implementation
Experience
- ◎ Certified in DRM, Planning,
Essbase and ODI
- ◎ Co-author of
theunlockedcube.com blog
- ◎ Pour over coffee enthusiast
- ◎ IPA snob

Will Andreelli

- ◎ Huron Consulting Group
- ◎ 8+ years
EPM Implementation
Experience
- ◎ Certified in Planning,
Essbase and PBCS
- ◎ Co-author of
theunlockedcube.com blog
- ◎ Football () fanatic
- ◎ Newbie [restaurant](#) owner

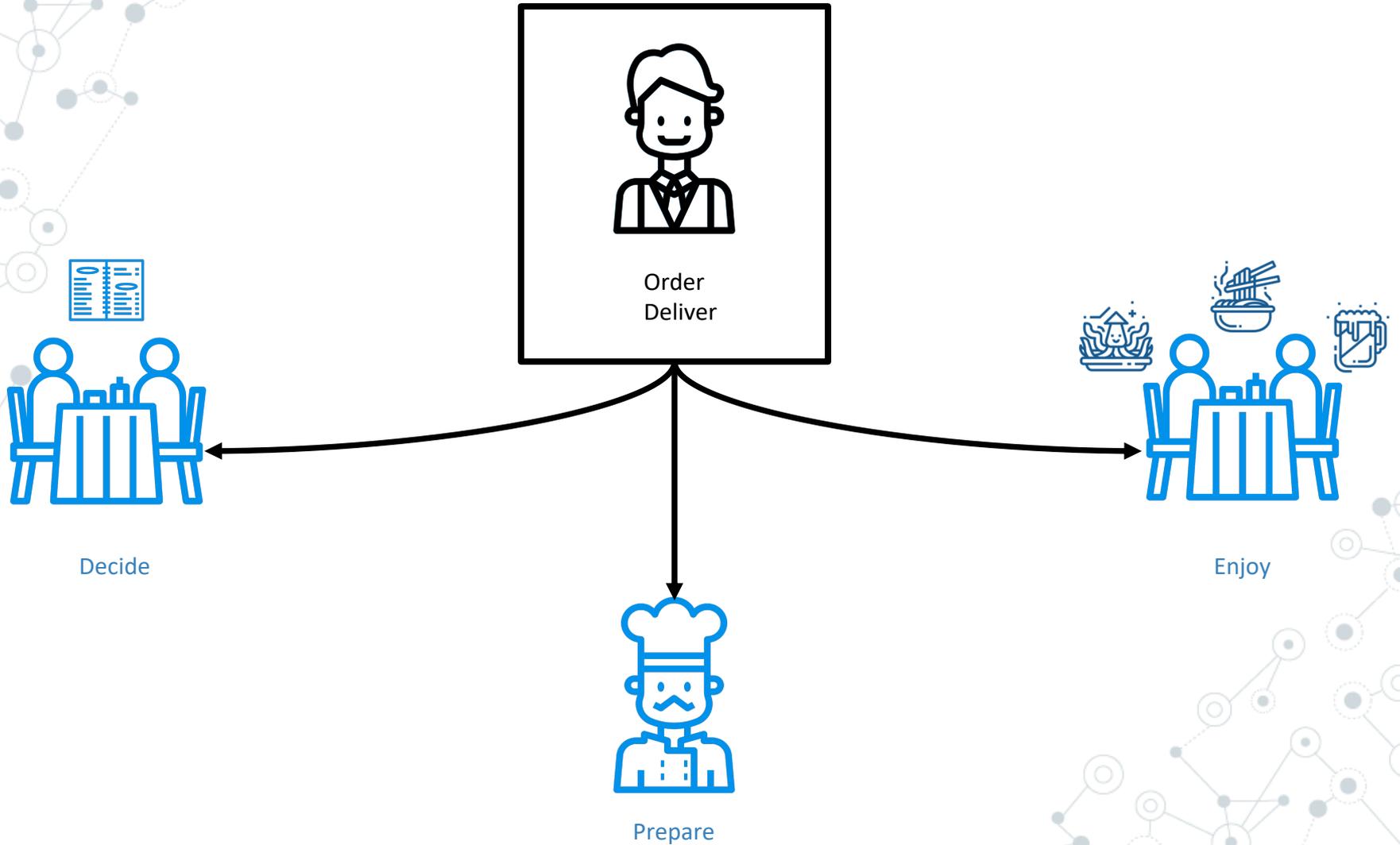


A decorative network diagram in the top-left corner, consisting of various sized circles (nodes) connected by thin lines (edges). Some nodes are solid grey, while others are hollow with a grey outline. The diagram is partially cut off by the left edge of the slide.

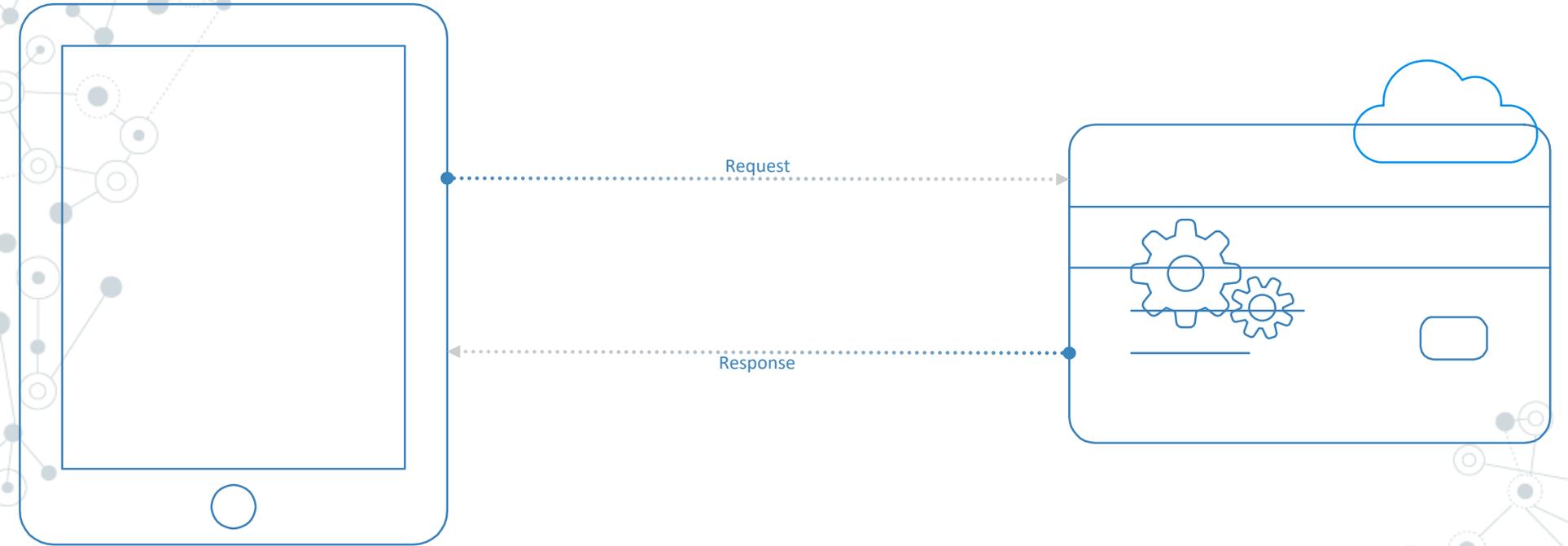
1. **REST**

Previously on REST...

RESTaurant API



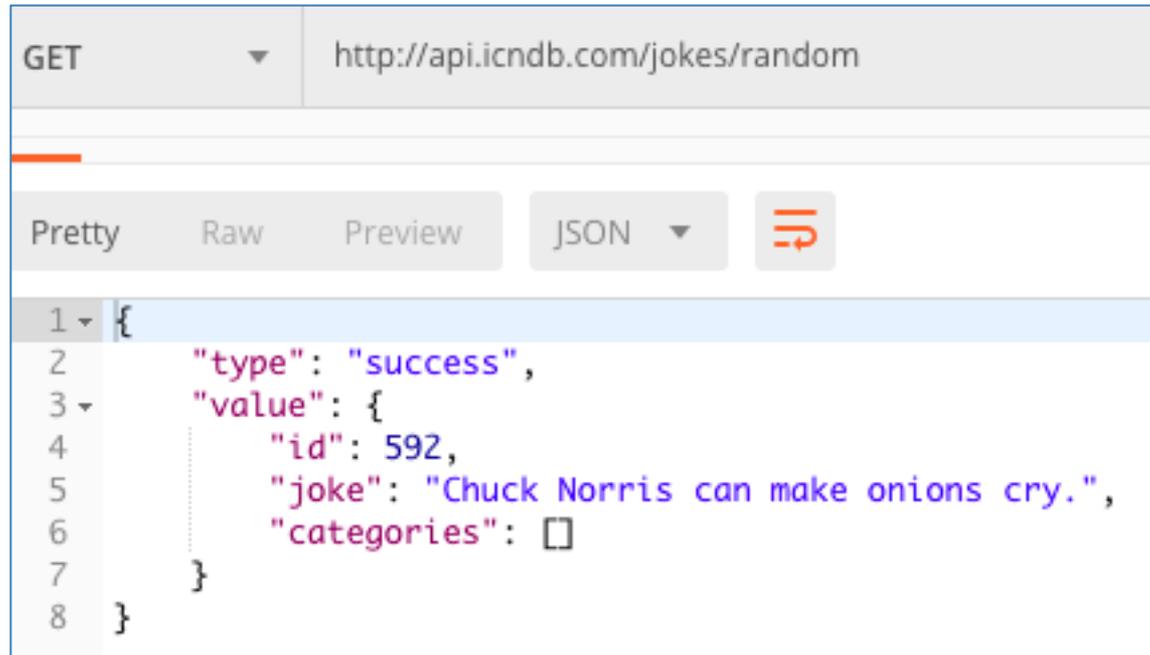
API



What is {REST}?

- ◎ *RE*presentational *St*ate *T*ransfer
- ◎ Set of constraints and properties for web applications.
- ◎ Easily understandable actions (verbs):
GET/POST/DELETE/PUT.
- ◎ Widely used, understood and accepted.

We mean widely used...



The screenshot shows a REST client interface with a GET request to `http://api.icndb.com/jokes/random`. The response is displayed in a pretty-printed JSON format. The JSON object has a `type` of `success` and a `value` object containing `id` (592), `joke` ("Chuck Norris can make onions cry."), and `categories` (an empty array).

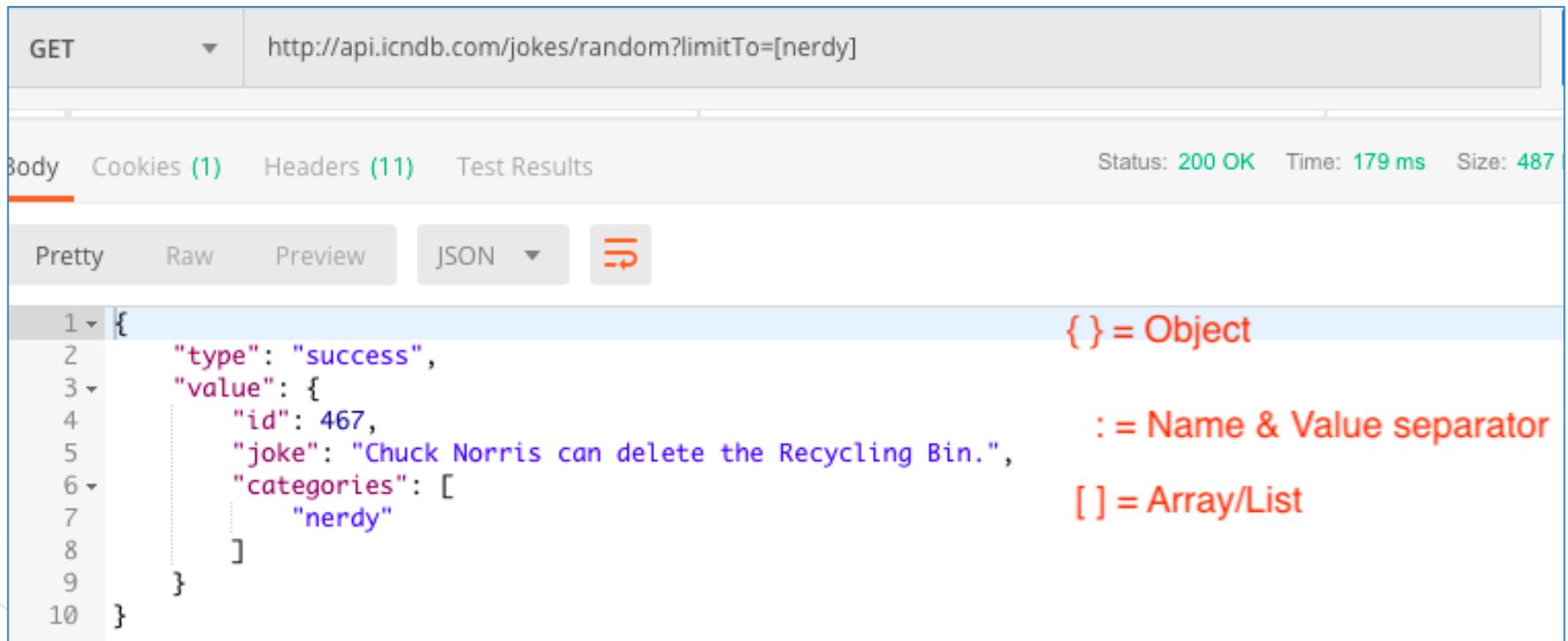
```
GET http://api.icndb.com/jokes/random

Pretty Raw Preview JSON

1 {
2   "type": "success",
3   "value": {
4     "id": 592,
5     "joke": "Chuck Norris can make onions cry.",
6     "categories": []
7   }
8 }
```

[JSON]

- ◎ *JavaScript Object Notation.*
- ◎ Lightweight & easily understandable



```

GET http://api.icndb.com/jokes/random?limitTo=[nerdy]
Status: 200 OK Time: 179 ms Size: 487
Body Cookies (1) Headers (11) Test Results
Pretty Raw Preview JSON
1 { } = Object
2   "type": "success",
3   "value": {
4     "id": 467,
5     "joke": "Chuck Norris can delete the Recycling Bin.",
6     "categories": [
7       "nerdy"
8     ]
9   }
10 }
  
```

Previously with the Essbase Cloud REST API

- ◎ We said the API was undocumented
- ◎ We showcased how to:
 - Run calcs/jobs
 - Use Postman
 - Create groovy scripts calling the API
 - Create ODI integrations with the API

Check out the presentation [here](#).

And now with the Essbase Cloud REST API

- ◎ Self-documented, with Swagger.
 - Try it out features.
- ◎ Understandable?
- ◎ Some API calls have changed over the last year.
 - Version 1 still.

Essbase Cloud REST API

- ◎ Authorization: Basic, like other EPM services.
 - username@domain.com
- ◎ Base URL:
 - http(s)://server/essbase/rest/(version)
- ◎ Current version: v1
- ◎ Responses: JSON

```
{  
  "userId": "vkurian@unlk.com",  
  "sessionId": "0",  
  "loginTimeInSeconds": "55",  
  "connectionSource": "unlk.oraclecloud.internal"  
},
```

2.

Swagger

What means Swagger?

What Means {S}wagger?

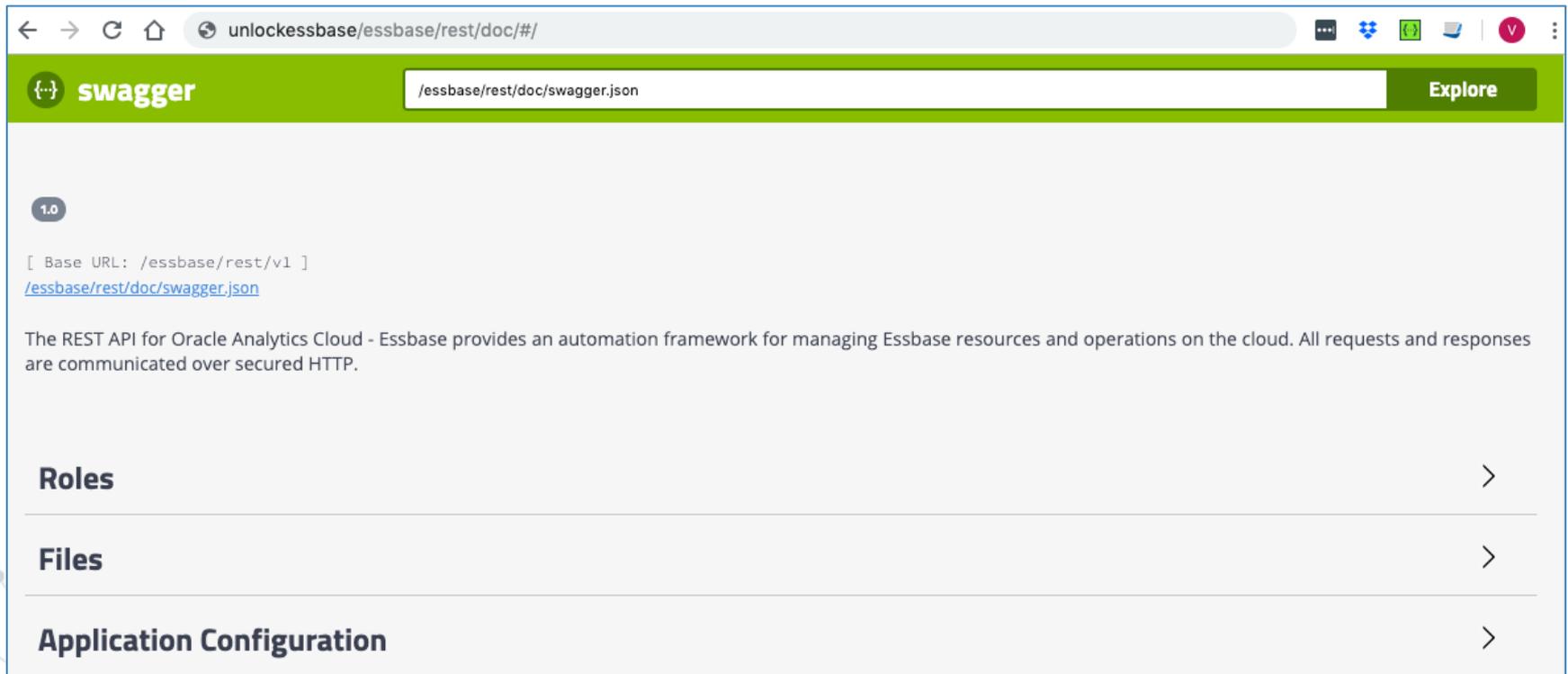


- ◎ REST API Documentation
 - Human readable
 - Machine readable
- ◎ What operations are supported?
- ◎ Parameters?
- ◎ Authorization?

Essbase REST Swagger

Available at:

- [http\(s\)://youressbaseserver/essbase/rest/doc/swagger.json](http(s)://youressbaseserver/essbase/rest/doc/swagger.json)



The screenshot shows a web browser window displaying the Swagger UI. The address bar shows the URL `unlockessbase/essbase/rest/doc/#/`. The Swagger logo is in the top left, and the path `/essbase/rest/doc/swagger.json` is in the top right, next to an `Explore` button. Below the header, the version `1.0` is displayed. The base URL is `[Base URL: /essbase/rest/v1]` with a link to `/essbase/rest/doc/swagger.json`. A description states: "The REST API for Oracle Analytics Cloud - Essbase provides an automation framework for managing Essbase resources and operations on the cloud. All requests and responses are communicated over secured HTTP." Below this, there are three expandable sections: `Roles`, `Files`, and `Application Configuration`, each with a right-pointing chevron.

Essbase REST Swagger



/essbase/rest/doc/swagger.json

Explore

1.0

[Base URL: /essbase/rest/v1]
</essbase/rest/doc/swagger.json>

The REST API for Oracle Analytics Cloud - Essbase provides an automation framework for managing Essbase resources and operations on the cloud. All requests and responses are communicated over secured HTTP.

Roles >

Files >

Application Configuration >

Application Connections >

Global Connections >

Application Data Sources >

Global Data Sources >

Drill Through Reports >

Email >

Workbook Utilities >

Filters >

Groups >

Jobs >

Essbase REST Swagger

◎ Good starting point for scripts:

Curl

```
curl -X GET "http://servername/essbase/rest/v1/sessions" -H "accept: application/json"
```

◎ Swagger Inspector plugin:

The screenshot shows the Swagger Inspector web application interface. The browser address bar displays `https://inspector.swagger.io/builder`. The application header includes the SMARTBEAR logo and the text "Swagger Inspector.", along with "SIGN UP" and "LOG IN" buttons. Below the header, there is a "DEFINITION" dropdown menu, a "GET" dropdown menu, and a text input field containing "Any HTTP resource or the URL of an OpenAPI / WSDL definition". A "SEND" button is located to the right of the input field. Below the input field, there is a note: "Supports Swagger/OAS 2, OAS 3 or WSDL file". A text input field contains the URL `http://servername/essbase/rest/v1/swagger.json`, followed by a close button (x) and a "PARSED" button. Below this, there is a "Try our Link" link and a "Swagger Operations" section with an "UNDO CLEAR" button. On the right side, there is a "History" section and a "Collections" section. The "Collections" section has a right arrow button. Below the "Collections" section, there is an "ADD COLLECTION" button and a "CREATE API DEFINITION" button with a checkmark icon.

A decorative network diagram in the top-left corner, consisting of various sized grey circles connected by thin grey lines, some solid and some dashed, creating a complex web-like structure.

3. Postman

Postman

- ◎ Must-have tool for RESTful development.
- ◎ Free
 - Postman Pro allows collaboration (Paid)
- ◎ Sync across machines.
- ◎ Quickly run REST API requests
- ◎ Can create different environments with different connection details
- ◎ Can create and use variables
- ◎ Scripting samples

=> Refer to last year's [presentation](#)

A decorative network diagram in the top-left corner, consisting of various sized circles (nodes) connected by thin lines (edges). Some nodes are solid grey, while others are hollow white with a grey border. The diagram is partially cut off by the left edge of the slide.

4. Scripting

cURL

Scripting

- ◎ Install [cURL](#), if needed.
 - “Client” URL
 - Command line tool (all OS)
 - Multiple protocol support (HTTP(S)/FTP/SOAP)
- ◎ Extremely useful tool:
 - For dad [jokes](#)

```
vkurianJGH6:~ vkurian$ curl -H "Accept: application/json" https://icanhazdadjoke.com/  
{ "id": "cxXv4Elj3wc", "joke": "Where do young cows eat lunch? In the calf-ateria.", "status": 200 }
```

- ...and more nerdy jokes

```
vkurianJGH6:~ vkurian$ curl https://geek-jokes.sameerkumar.website/api  
"SON: What are clouds made of? DAD: Linux servers, mostly"  
vkurianJGH6:~ vkurian$ █
```

A Simple Script

- ◎ Log in
- ◎ Cookies
 - Allow you to persist a connection
 - Save the cookie
- ◎ Perform the needed action(s)
- ◎ Logout
- ◎ Delete that cookie

A Simple Script

Log in + cookie storage

```
unlock>curl -X GET -u "test1:Kscope19" http://unlockessbase.oracle.com/essbase/rest/v1/session -c cookie.txt
```

Annotations: Username:password (under -u), API endpoint (under http://...), Save that cookie! (under -c cookie.txt)

Perform actions

```
unlock>curl -X GET http://unlockessbase.oracle.com/essbase/rest/v1/variables -b cookie.txt -o variables.json
```

Annotations: API endpoint (under http://...), Use that cookie! (under -b cookie.txt), Save output (under -o variables.json)

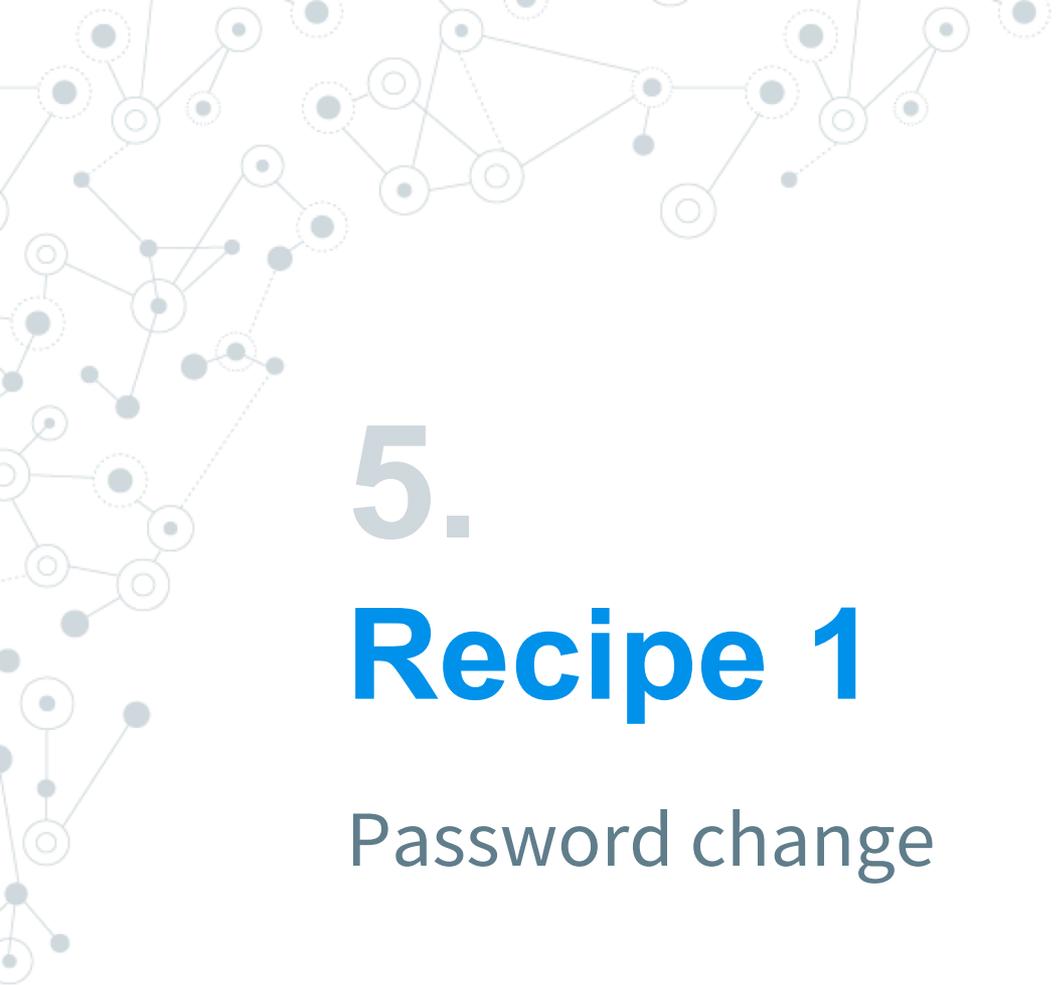
Logout

```
unlock>curl -X DELETE http://unlockessbase.oracle.com/essbase/rest/v1/session -b cookie.txt
```

Annotation: Action change (under -X DELETE)

Delete that cookie

Name
cookie.txt
variables.json

A decorative network diagram in the top-left corner, consisting of various sized grey circles connected by thin grey lines, some with dashed outlines.

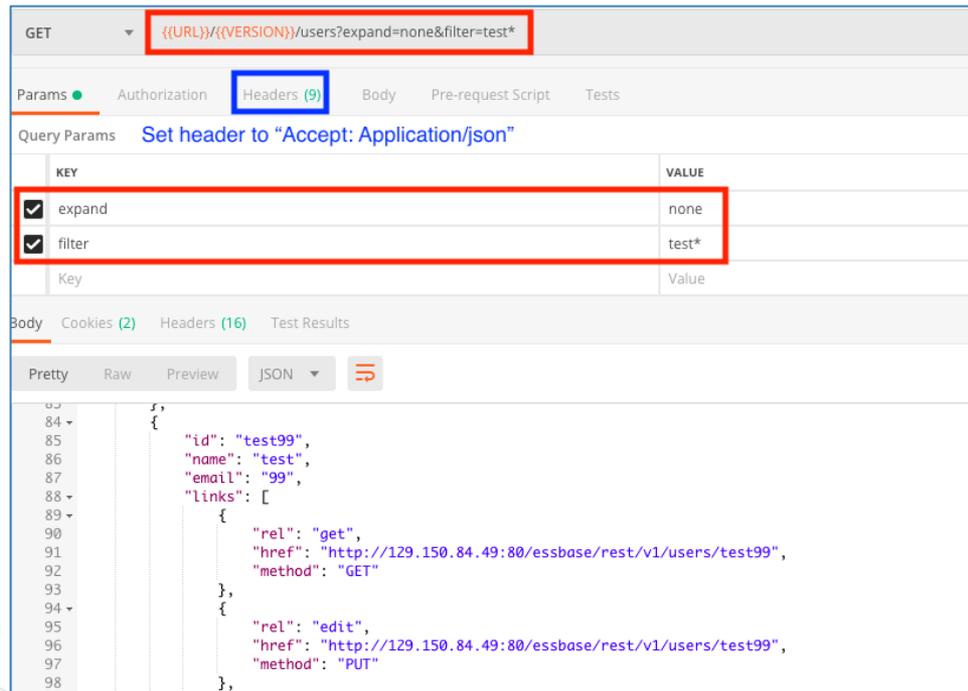
5.

Recipe 1

Password change

Password Change

- ◎ Forcing password changes
 - Native security?
 - OAC Classic users?
- ◎ How to GET users with a filter?
 - `http(s)://BaseURL/users`



The screenshot shows a REST client interface with the following configuration and response:

Request Configuration:

- Method: GET
- URL: `{{URL}}/{{VERSION}}/users?expand=none&filter=test*`
- Headers (9): Set header to "Accept: Application/json"
- Query Params:

KEY	VALUE
<input checked="" type="checkbox"/> expand	none
<input checked="" type="checkbox"/> filter	test*
Key	Value

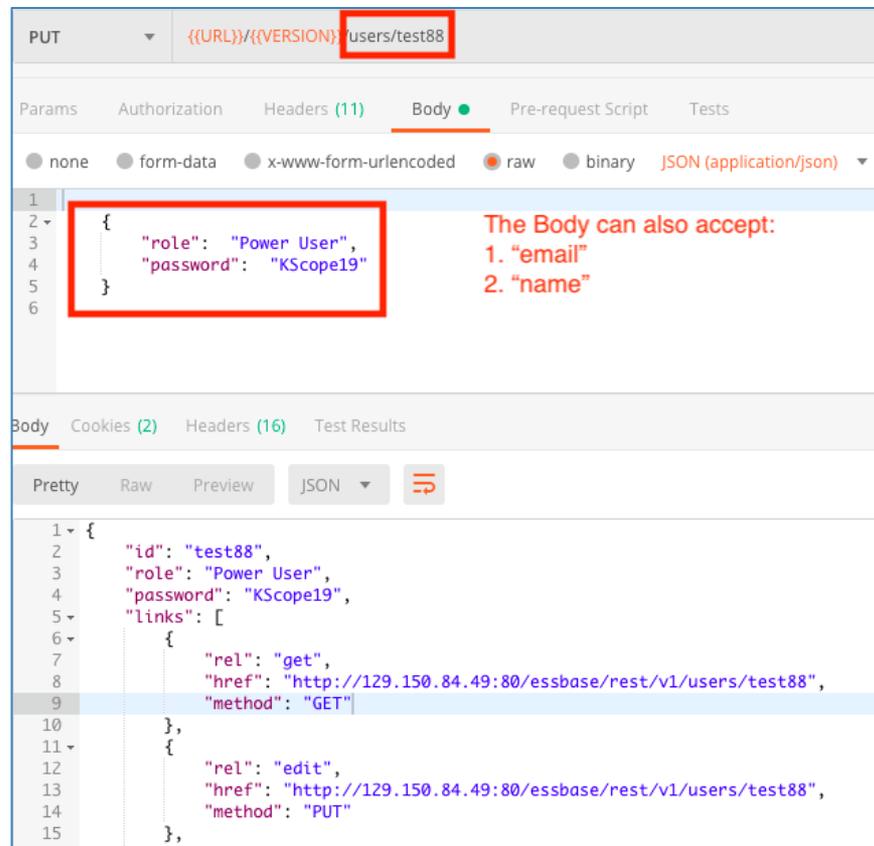
Response (JSON):

```

84 {
85   "id": "test99",
86   "name": "test",
87   "email": "99",
88   "links": [
89     {
90       "rel": "get",
91       "href": "http://129.150.84.49:80/essbase/rest/v1/users/test99",
92       "method": "GET"
93     },
94     {
95       "rel": "edit",
96       "href": "http://129.150.84.49:80/essbase/rest/v1/users/test99",
97       "method": "PUT"
98     }
99   ]
100 }
  
```

Password Change

- ◎ Simple PUT to change passwords from generated csv file
 - http(s)://BaseURL/users/userID (i.e. “vkurian”)



The screenshot shows a REST client interface with the following details:

- Method:** PUT
- URL:** `{{URL}}/{{VERSION}}/users/test88`
- Body Type:** raw
- Body Content:**

```
{
  "role": "Power User",
  "password": "KScope19"
}
```
- Response:**

```
{
  "id": "test88",
  "role": "Power User",
  "password": "KScope19",
  "links": [
    {
      "rel": "get",
      "href": "http://129.150.84.49:80/essbase/rest/v1/users/test88",
      "method": "GET"
    },
    {
      "rel": "edit",
      "href": "http://129.150.84.49:80/essbase/rest/v1/users/test88",
      "method": "PUT"
    }
  ]
}
```

The interface also includes tabs for Params, Authorization, Headers (11), Body, Pre-request Script, and Tests. The response is displayed in a Pretty view with a JSON dropdown menu.

Enter PowerShell

- ◎ Ingest csv file

```
$users = Import-Csv users.csv
```

- ◎ Store credentials in Base64 form

```
$Credentials = "admin:Hyperion123"
$Bytes = [System.Text.Encoding]::ASCII.GetBytes($Credentials)
$EncodedText = [Convert]::ToBase64String($Bytes)
$headers = @{ Authorization = "Basic $EncodedText" }
$headers
```

- Clear text "username:password"
- Get Bytes
- Encode to Base64
- Header with "Basic"

- ◎ Loop through each record in csv

- ◎ ForEach
- ◎ Convert csv record to JSON

```
$body = $user | Select-Object -Property role,@{Name='password';Expression={'KScope19'}} | ConvertTo-Json
```

JSON conversion

Hardcoded password

- ◎ Invoke-RestMethod performs the PUT action

```
$url = "http://unlockessbase.oracle.com/essbase/rest/v1/users/" + $user.id
Invoke-RestMethod -Uri $url -Method PUT -ContentType "application/json" -Headers $headers -UseBasicParsing -Body $body
```

A decorative network diagram in the top-left corner, consisting of various sized grey circles connected by thin grey lines, some with dashed outlines.

6.

Recipe 2

Streaming Data Loads

Streaming Data Loads

◎ Why do this?

- Move EPBCS data to Essbase Cloud?
- Orchestrate an integration using OIC/ODI?
- Other applications?

◎ Steps involved:

- {POST}: Generate a data load session and retrieve a stream ID.
- {POST}: Start the data load with the stream ID.
- {DELETE}: Terminate the load session.

◎ DELETE step is important, as data will not be committed otherwise.

Ingredient 1: Generate a session

⦿ POST action

- http(s)://BaseURL/applications/{appname}/databases/{dbname}/dataload/
- Add the load rule you want to use.
- Gather the stream ID.

▶ 1_POST_Load_2

POST ▼ http://1[redacted]essbase/rest/v1/applications/xtest/databases/xtest/dataload/

Params Authorization Headers (1) **Body** ● Pre-request Script Tests ●

● none ● form-data ● x-www-form-urlencoded ● **raw** ● binary JSON (application/json) ▼

```

1 {
2   "ruleFileName": "test3"
3 }
```

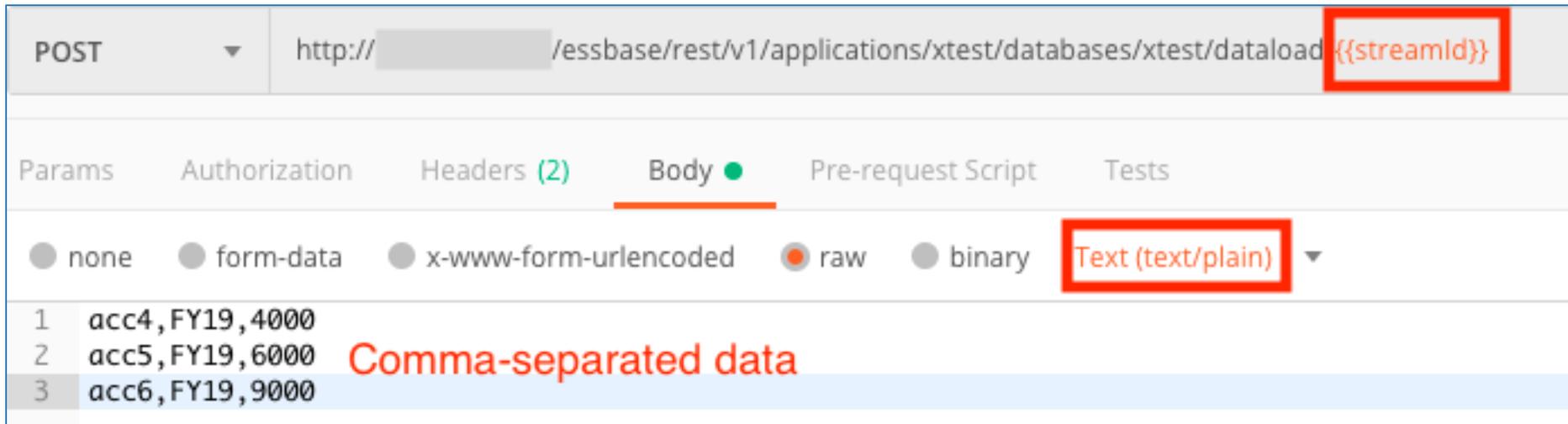
↓

```
{
  "streamId": "1980397126",
  "links": [
```

Ingredient 2: Generate a session

⦿ POST action

- http(s)://BaseURL/applications/{appname}/databases/{dbname}/dataload/
- Start the load to the generated streamID.
- Content-Type should be “text/plain”.



POST ▼ http://[redacted]/essbase/rest/v1/applications/xtest/databases/xtest/dataload/{{streamId}}

Params Authorization Headers (2) **Body** ● Pre-request Script Tests

● none ● form-data ● x-www-form-urlencoded ● **raw** ● binary **Text (text/plain)** ▼

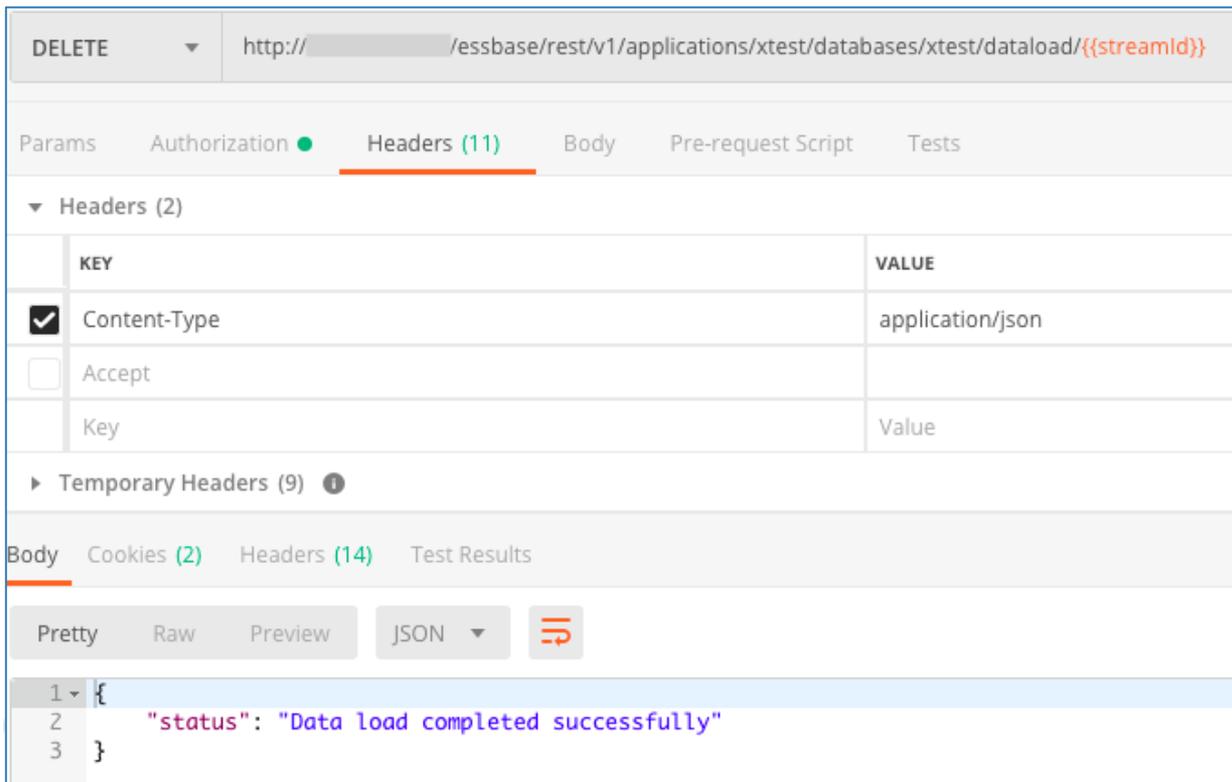
1	acc4, FY19, 4000
2	acc5, FY19, 6000
3	acc6, FY19, 9000

Comma-separated data

Ingredient 3: Terminate the session

⦿ DELETE action

- http(s)://BaseURL/applications/{appname}/databases/{dbname}/dataload/
- Terminate the session.



The screenshot shows a REST client interface with the following details:

- Method:** DELETE
- URL:** http://.../essbase/rest/v1/applications/xtest/databases/xtest/dataload/{{streamId}}
- Headers (2):**

KEY	VALUE
<input checked="" type="checkbox"/> Content-Type	application/json
<input type="checkbox"/> Accept	
Key	Value
- Temporary Headers (9)**
- Body:**

```

1 {
2   "status": "Data load completed successfully"
3 }
```

Demo Time

Recipe 2



Demo: Streaming Load

Video can be found on the unlocked cube Youtube channel

Video Name: Stream data to Essbase Cloud using the REST API

Video Link: <https://www.youtube.com/watch?v=TI1g0mwZyNA&feature=youtu.be>

A decorative network diagram in the top-left corner, consisting of various sized circles (nodes) connected by thin lines (edges). Some nodes are solid grey, while others are hollow with a grey outline. The diagram is partially cut off by the left edge of the slide.

7.

Recipe 3

Create and/or update dimension members

Create and/or update dimension members

Why do this?

- Orchestrate an integration using OIC/ODI
- Other applications?

Ingredients:

- Type of Request: POST
- Request URL:
`http(s)://BaseURL/applications/{AppName}/databases/{dbName}/boe`
(Boe stands for batch outline edit)
- JSON body

Create and/or update dimension members

```

POST http://theunlockedcube/essbase/rest/v1/applications/xtest/databases/xtest/boe

Params Authorization Headers (11) Body ● Pre-request Script Tests
● none ● form-data ● x-www-form-urlencoded ● raw ● binary JSON (application/json) ▼
1  {
2    "otlVersion": "0",
3    "validate": "true",
4    "validateFormulas": "true",
5    "keepTransaction": "true",
6    "restructOption": "allData",
7    "jsonformatLog": "false",
8    "editActions" : [
9      {
10     "mbrAdd" : {
11       "mbrName" : "Acct_UNLK1",
12       "parent" : "Account"
13     }
14   },
15   {
16     "mbrUpdate" : {
17       "mbrInfo" : {
18         "dataStorage" : "STORE_DATA",
19         "consolidation" : "+",
20         "alias" : [ {
21           "aliasTable" : "default",
22           "alias" : "UNLK1 - LYON"
23         } ]
24       },
25       "thisMbr" : "Acct_UNLK1"
26     }
27   }
28 ]
29 }
  
```

Main Element

otlVersion: Required

validate: true|false. Optional. Default = true

restructOption: allData|noData|lowData|inData. Default = allData

editActions

Sub Elements

mbrAdd: mbrName & Parent (additional options available like preSibling)

mbrUpdate: used to update properties like aliases, data Storage consolidation sign, UDAs, Time balance, formula, mbrID, etc.

Each mbrUpdate requires thisMbr to be defined

Demo Time

Recipe 3





Demo: Create and/or update dimension members

Video can be found on the unlocked cube Youtube channel

Video Name: Update dimension in Essbase Cloud using the REST API

Video Link: https://www.youtube.com/watch?v=umiLSckM_CA&feature=youtu.be

Summary

- ◎ REST
- ◎ Swagger
- ◎ Postman
- ◎ Scripts

Thanks!

Any questions?

You can find us at:

[@theunlockedcube](#)

www.theunlockedcube.com

vkurian@huronconsultinggroup.com

wandreelli@huronconsultinggroup.com



ODTUG
Kscope19
SEATTLE, WASHINGTON • JUNE 23-27



**PLEASE FILL OUT
YOUR EVALUATIONS**

SEATTLE

 Washington State
Convention Center