

Real Time Integration of Cloud Services with On-Premise Application

¹Manjunath S Hosalli, ²Monica R Mundada*, ³Meeradevi*, ⁴Anuraga Deepika*

¹Dept. of CSE, M S Ramaiah Institute of Technology, Bangalore, India

²Dept. of CSE, M S Ramaiah Institute of Technology, Bangalore, India

³Dept. of CSE, M S Ramaiah Institute of Technology, Bangalore, India

⁴IDC UIT Reporting & Analytics, UNISYS India Pvt Ltd, Bangalore, India

Authors Mail Id's: hosallimanjunath046@gmail.com

Corresponding Authors Mail Id's: monica@msrit.edu; meera_ak@msrit.edu;
Anuraga.deepika@in.unisys.com

ABSTRACT:

The paper presents design and development of REST API for integrating on- premise application to cloud services. REST API's are design and developed in mulesoft tool since this tool will provide platform to integrate almost all on- premise applications (data base) to cloud services. Demand for API increasing day by day as most of the industry applications moving from on- premise to cloud. Organization's focusing on (Response First Process Later) RFPL rule so availability, security and flexibility of internal information are very important for organization. Oracle CPQ Cloud is a complete solution that streamlines sales, from inquiry to order with flexible quoting software capabilities. At present, the Oracle CPQ is communicating with on premise database for products/service pricing whenever a quote is submitted from Oracle CPQ cloud. The requests from oracle CPQ will trigger an API in Informatica cloud through the agent server running on-premise. This is a real-time or just in time communication, light weight data transfer to get a response from on-premise database to Oracle CPQ. This mode of communication will be re-designed with Mulesoft integration to eliminate the license cost which is spent for connectors used in Informatica Cloud. Also, the proposed design will get rid of the open issues like time out error while processing process objects through JDBC connection, cache issue with MMTP cost.

Keywords: Cloud, JDBC, MMTP COST, Mulesoft, Database

1. INTRODUCTION:

Integrating On-Premise applications such as SQL server, Oracle SQL Developer and DB with the SAAS (Software as a Service) Cloud service such as Salesforce Cloud, Microsoft SQL Azure

and Google Analytics. While integrating any on-premise applications encounter a question of why and how, purpose of integration is to provide the data ease, to overcome time error, cache error which exist with integration tool. How to achieve this as definition said source and destination need to configure the path by using third party tool. To integrate application JDBC driver will be used since JDBC drive is the wrapper that uses underlying web service API which makes connection and consumption of data between on-premise (Oracle SQL Developer) applications and Cloud services (salesforce) [1]. Now question about what is the third party tool needed to perform the integration. Mulesoft is eclipse based tool based on Anypoint studio it provides facilities in both mode offline and online. This tool used as an integration tool it stores the credentials required to access the on-premise applications and it provides many components where developer can configure them based on requirements like want to insert data, update data, delete some data. To perform this action mulesoft also uses the JDBC driver as a wrapper. This gives option to generate REST (Represtantional State Transfer) API and SOAP (Simple Object Access Protocol) API [2]. In proposed model REST API is preferred since it has capability to provide on-time service to the user and overcome time error. Demonstration purpose we used mulesoft tool trail version. In way tried to perform all DML and DDL commands from Cloud service after generating REST API. From cloud service accessed these on-premise data through the mulesoft. By giving input parameters which are created in RAML (RESTful API Modeling Language) as the front end Language given to the Cloud service user. And in Mulesoft created a property file in YAML (yaml ain't markup language) where all the on-premise access credentials will be stored. Key role here is ability to integrate them quickly or through API. In our paper in Module I we have discussed about Introduction, in Module II Related Work, in Module III Proposed Model, in Module IV Real Time Integration, in Module V Testing demo with on-premise application and in Module VI what we have achieved.

2. RELATED WORK:

When we start analysis page for designing API for making communication from on-Premise Applications like Oracle SQL developer, Microsoft SQL developer etc. to Cloud software Application such as Sales etc. As software evolves Developers went on making use of created stored Procedure which does the specified set of functions now we call it as API (Application Programming Interface) There are mainly two type of APIs SOAP API and REST API SOAP (Simple Object Access Protocol) It Only uses the POST method to send request and has only one URL for service entrance[3]. REST (Representational State Transfer) that defines set of constraints to be used for creating web services. Based on our requirement we selected REST API REST API's make applications communicate with each other based on its design and function defined in it. One of the crucial requirements for API is reliability, with Request coming from clients (cloud services) every time to perform particular task, API designed and implemented if it is not configured properly chances of it getting affect to the business is more for both the on-premise and cloud service. To retain integrate of the task need to take care of its performance carefully[4]. There are various API testing tools available like SOAP Ui, Postman[5]. The cloud service uses REST or SOAP API request through HTTP (Hyper Text Markup Language) to communicate with the on-premise application.

3. PROPOSED MODEL:

Real time integration cloud services with on-premise applications using REST APIs. Proposed model designing and developing an REST API used various other integration tool most of the tools have given time error and cache error while extracting respective quote, to fetch the data they took more than required timing and while once extract the data same data should not extract twice it will create cache error hence we prefer latest integration tool Mulesoft, is the best to perform integration. It didn't fetch data too late and didn't give data twice so resolved the time and cache error. Where actual organization data stored in on premises like SQL developer applications these applications data want to access by the Cloud services like CPQ for quoting the product. Mulesoft as an integration tool as shown in Fig1. Has an access to the on-premise applications through user credentials and generates API. Whenever Cloud Applications CPQ want to access data from on-Premise they need to execute the API by providing input parameters. This communication happens in real time execution.

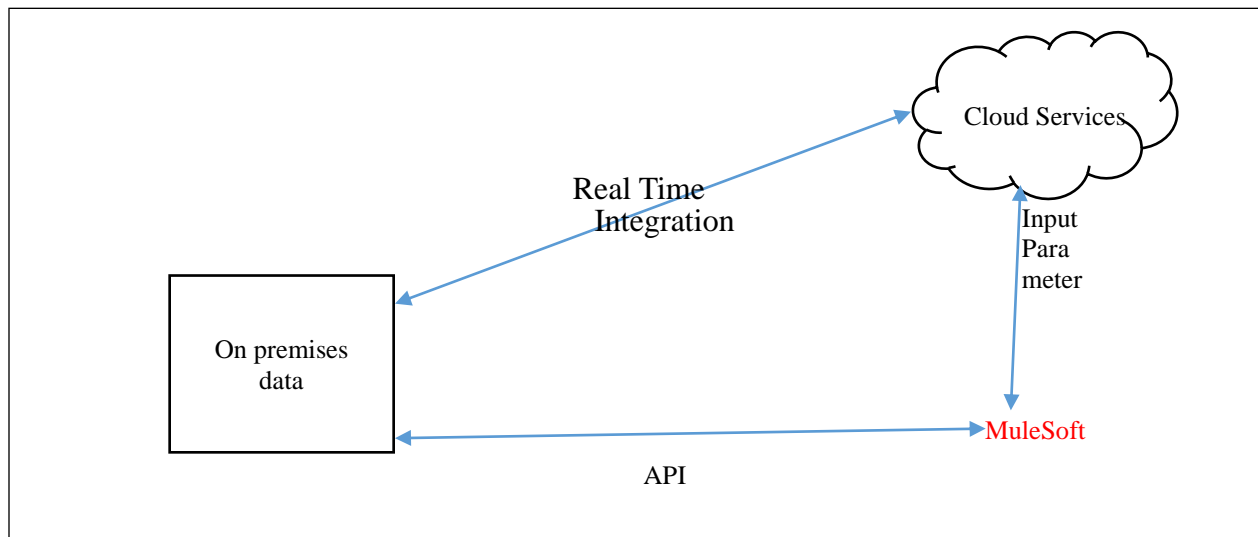


Fig 1: Integrate on- premise application and cloud services

4. REAL TIME INTEGRATION:

In Our demo model we have used Oracle SQL developer as an on-Premise data base tool and MuleSoft as an integration tool this tool provides various components and features advantage of this tool it accept input in flat files (CSV, JSON) file format and convert them into java object file for that purpose it uses JDBC driver. Datawave as an expression language we can use it when writing of code in one format may later require to have it in different format. Datawave is manly used in to pass the input in any format and get the data in any format as required by the user. In our demo we preferred XML format file as an input, JSON format as an output[6]. Sales as a Cloud Application. Through this model we are overcoming Time Out Error and Cache Error. Here consuming large data is easy since we are accessing data in flat file format and when we access data hugely. When request come for the particular end point it will give result accurately on time then there is no cache Error[7].

REST API for Communication:

Designing APIs are entirely different thing. There are best practices to ensure APIs are easy to consume. REST API provides easy way to communicate with the applications there are no hurdles between the cloud services and the applications the proximity of providing accurate results when we went for real time communication this API given the best result with the mulesoft tool. Able to perform all the methods which are provided by REST API. Let's list of five of them below. Using REST we have created Endpoints one Endpoint bsp we have taken as a demo [8][9].

TABLE 1: REST METHODS

RESOURCES	GET READ	POST CREATE	PUT UPDATE	DELETE
/BSP	RETURNS A LIST OF BSP POSSIBLY ALL THE BSP YOU HAVE	CREATE A NEW BSP	BULK UPDATE OF BSP	DELETE ALL BSP
/BSP/1004	RETURN A SPECIFIC BSP	METHOD NOT ALLOWED	UPDATES A SPECIFIC BSP	DELETES A SPECIFIC BSP

Design AND Develop:

All new apps/services have some kind of API and almost every app uses at least one API. What is an API? In layman terms, an API provides an interface between two applications so that they can talk to each other. RAML was introduced in 2013 and received great support from MuleSoft. RAML (RESTful API Modeling Language) is used as the front end language for an API to get input from the Cloud service, RAML is based on YAML language it will describing RESTful APIs. RAML will have the information which required to describing the RESTful APIs. Provides simpler way to design APIs[10]. Mulesoft API designer provides both web and offline mode for designing, documenting and testing APIs. If designing done in any mode can be imported it into any other mode and perform required configuration on it. It provides easy way to engage with the API consumer. The API Designer, you can add a new API by just clicking on "Add new API." It will ask for the API name and version name. Start the design of the API by providing API Name and version name[11]. Once above details are filled, Anypoint Platform will show API administration page, which includes the API definition, portal, and status of the API. Where designing for API is started (writing RAML) under "API definition. Go to Edit in API Designer and start writing the API[12].

TABLE 2: RAML CODE

RAML CODE
<pre> #%RAML 1.0 BASEURI: HTTPS://ANYPOINT.MULESOFT.COM/MOCKING/API/V1/LINKS/2A10ACE5- A2EF-41AF-913D-8A9CBDED2A88/ # TITLE: BSP & CER API VERSION: 0.1 /BSP: GET: SECUREDBY: -BASIC QUERYPARAMETERS: P_STYLE: TYPE: STRING REQUIRED: TRUE P_DESCRIPTION: TYPE: STRING REQUIRED: TRUE P_BOL_ORG: TYPE: STRING REQUIRED: TRUE RESPONSES: 200: BODY: </pre>

```
APPLICATION/JSON:

EXAMPLE: |

    {
      "DATA": {PAYLOAD}
    },
    "SUCCESS": TRUE,
"STATUS": 200
  }

400:

BODY:

APPLICATION/JSON:

DESCRIPTION: BAD REQUEST
```

Once you have finished with the Design and develop of RAML file, import same file into Anypoint studio. After importing the file into Anypoint studio design the end points for inserting, updating based on the requirements.

5. TEST API WITHON-PREMISE:

After implementing deploy the application into cloud. While testing click on specific end point and enter all the input parameters which are made as a mandatory in RAML. After passing parameters click submit if you get designed data than your succeeded otherwise no then look back into the design and correct it. In our demo we have executed it on local machine using <http://localhost/endpointname>.

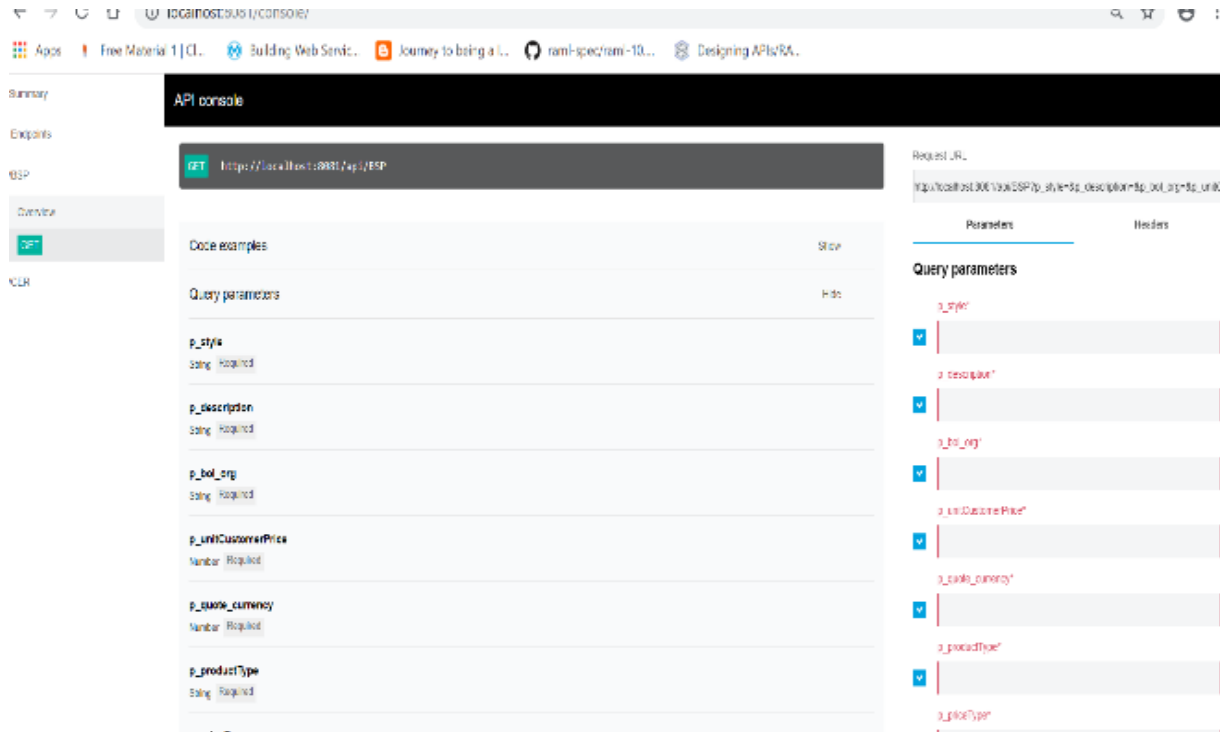


Fig 2: Endpoint with parameter tab

While accessing data from cloud need to pass all the parameters after passing if it match with the data type of the parameters you will get result.

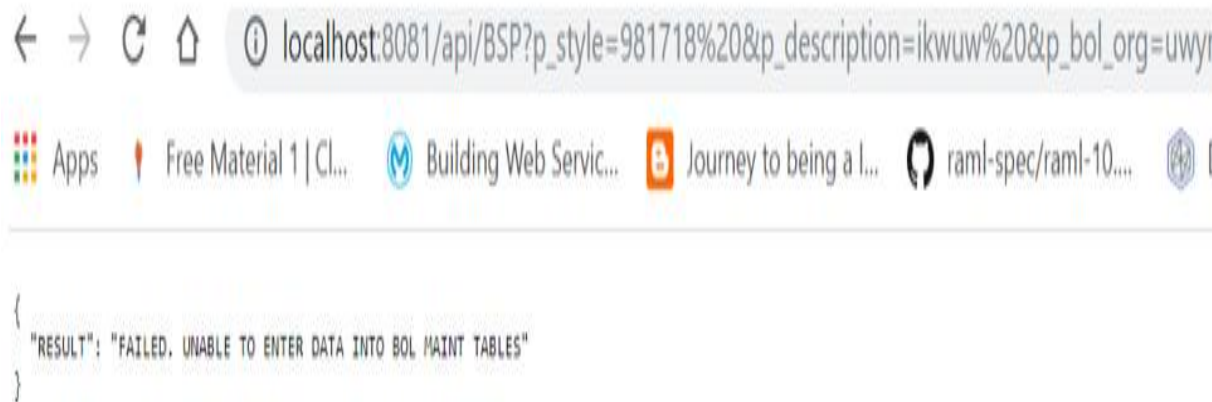


Fig 3: Endpoint Result

As per our design, bsp end point has data “Result”: “Failed, unable to enter into bol maint tables”.

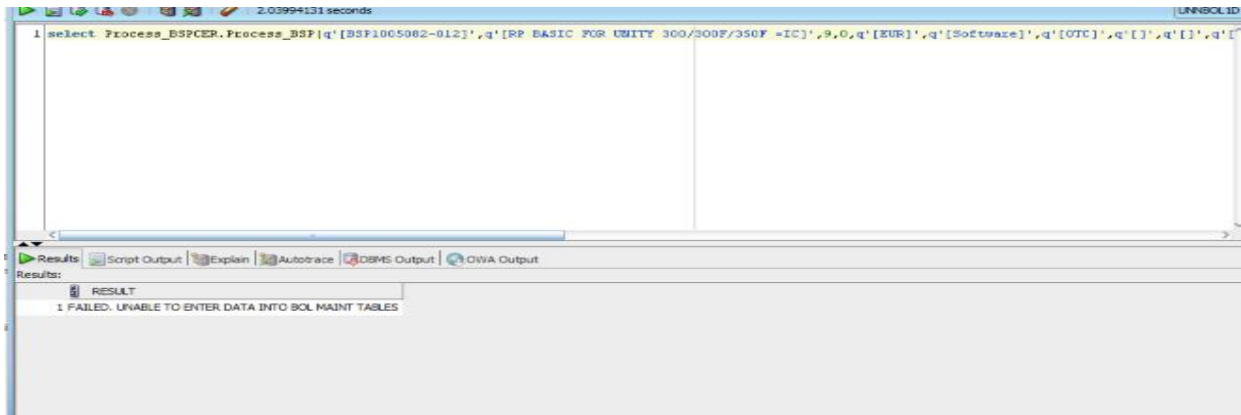


Fig. 4: Acknowledgement for respective Endpoint

To verify whether the endpoint result is accurate or not we have checked On-Premises database. Get exact data. Within time and one time.

6. ACKNOWLEDGE:

Thanks to Unisys India Pvt Ltd, for providing me the opportunity to design and develop REST API with logic – by taking mulesoft as the integration tool.

7. CONCLUSION:

Proposed model interface with on premise application to cloud services through integration tools. It helps to build API and thus communications with the Cloud services becomes effective and provide security. Mulesoft also allows specific API's calls those can be exposed to build interfaces with other applications. Our work consists to build API to communicate in between Cloud services and On-premises system so that the transfer of data happens real time from this demo we are able to achieve Time Error and Cache Error We got result for mulesoft.

REFERENCES

- [1] K. Boonchuay, Y. Intasorn and K. Rattanaopas, "Design and implementation a REST API for association rule mining," 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Phuket, 2017, pp. 668-671.
- [2] W. Zhou, L. Li and W. Chou, "SDN Northbound REST API with Efficient Caches," 2014 IEEE International Conference on Web Services, Anchorage, AK, 2014, pp. 257-264.
- [3] S. Liu, G. Bai, J. Sun and J. S. Dong, "Towards Using Concurrent Java API Correctly," 2016 21st International Conference on Engineering of Complex Computer Systems (ICECCS), Dubai, 2016, pp. 219-222.
- [4] C. Zhao, M. Estep and F. Moinian, "Teaching Undergraduate Computer Graphics: Game Design and Implementation Utilizing the OpenGL Library and the Windows API,"

- 2017 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, 2017, pp. 1195-1198.
- [5] R. R. Hariadi, W. N. Khotimah and E. A. Wiyono, "Design and implementation of food nutrition information system using SURF and FatSecret API," 2015 International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA), Surabaya, 2015, pp. 181-183.
- [6] D. Murayama, S. Mizuno and A. Fujii, "Distributed Synchronization over RESTful Web API," 2015 18th International Conference on Network-Based Information Systems, Taipei, 2015, pp. 195-198.
- [7] M. V. Lavrentyeva and P. Y. Chimitov, "Implementation of recognition algorithm with NXOpen API in siemens NX," 2017 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), St. Petersburg, 2017, pp. 1-4.
- [8] M. Liu, J. Wang, J. Wang, G. Tang and M. Sun, "Design and Implementation of Emergency and Evacuation System Based on Baidu Map API," 2018 26th International Conference on Geoinformatics, Kunming, 2018, pp. 1-5.
- [9] J. Zhang and S. Cao, "Design and implementation of a third-party developers auditing system based on open API," 2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, 2015, pp. 803-806.
- [10] S. Jonnada and J. K. Joy, "Measure your API Complexity and Reliability," 2019 IEEE 17th International Conference on Software Engineering Research, Management and Applications (SERA), Honolulu, HI, USA, 2019, pp. 104-109.
- [11] C. Xu, B. Min, X. Sun, J. Hu, B. Li and Y. Duan, "MULAPI: A Tool for API Method and Usage Location Recommendation," 2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), Montreal, QC, Canada, 2019, pp. 119-122.
- [12] F. Thing, R. J. Oentaryo, D. Lo and Y. Tian, "WebAPIRec: Recommending Web APIs to Software Projects via Personalized Ranking," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 1, no. 3, pp. 145-156, June 201.