

Automation Tool For Customer Relationship Management Applications

Rahul Pandurang Pokale^{1*}, Nagaraja G.S.²

^{1,2}Department of Computer Science and Engineering, R.V. College of Engineering, Bangalore, India

*Corresponding Author: rahulpp.cs18@rvce.edu.in, Tel.: +91-7795184200

DOI: <https://doi.org/10.26438/ijcse/v9i6.3336> | Available online at: www.ijcseonline.org

Received: 16/Jun/2021, Accepted: 20/Jun/2021, Published: 30/Jun/2021

Abstract — Testing is one of the most important phases in Software Development Life Cycle (SDLC). To test any CRM application such as Salesforce and Siebel, Test engineer pretends as an end user of the system and checks whether the application is working as expected. Test engineer may need to test hundreds of test data combinations to ensure the correctness of the software. The same functionality might be needed to test again with the same test dataset as the changes are made to the software. This paper discusses how we can use Salesforce APIs and Siebel web services to reduce the time required for test data set up and retesting. In this paper firstly the importance of Salesforce and Siebel web services is explained. Then the detailed explanation of the tool is given under the Methodology section and finally experimental results are discussed.

Keywords — Salesforce; Siebel; Software Testing; Test Automation

I. INTRODUCTION

Test data preparation is an integral part of the Software Testing Life Cycle (STLC) [1]. Test data is data which has been specifically identified for use in tests. Test data should be prepared in such a way that it should incorporate all the application functionality but not exceeding cost and time constraint for preparing data and running tests.

In this paper, we have considered two CRM applications i.e. Salesforce [2] and Siebel. Customer relationship management (CRM) is a technology for managing all your company's relationships and interactions with customers and potential customers.

1.1 Salesforce

Salesforce provides some of the objects by default which are called as Standard Objects. Examples of standard objects are accounts, contacts, opportunities, Leads, products, campaigns, cases, users, contracts, Report, and dashboards, etc. The second type of object in Salesforce is called Custom Objects which are created by the client as per the business requirements. Each standard object has some predefined standard fields which can also be modified if required. Salesforce provides various API's for accessing the data from Salesforce and to store new data into the Salesforce. Salesforce APIs are a way for other applications to programmatically access data within your Salesforce org, in a simple and secure manner. The various API's provided by Salesforce are REST API, SOAP API, Bulk API and Streaming API. Their main purpose is to let the user manipulate Salesforce data.

To test any custom Salesforce object, the testers need to manually enter the data and check whether the data is correctly stored in correct format and at correct place. To

test application completely, the tester might need to insert hundreds of records to see whether the application is working as expected [3]. The time required for inserting the records into the Salesforce application can be reduced by making use of API's provided by the Salesforce.

1.2 Siebel

Siebel also provides predefined and custom objects. A Siebel object definition is the metadata that defines a Siebel application. Siebel object definitions define user interface elements that Siebel CRM includes in the Siebel client, business entities, and the Siebel Database. The Siebel application allows enterprises to publish any business service or business process as a Web service. This process is also known as creating an inbound Web service.

To test any customized Siebel application, the tester needs to insert records and check whether the data is stored as per the requirement. The time required to insert records can be reduced by invoking the desired web service. After the data is inserted, the tester can manually login to the Siebel UI and compare the inserted records with the expected results to check whether the application is working as expected.

II. RELATED WORK

Reference [2] talks about Salesforce architecture, MVC of Salesforce and Salesforce Object Query Language (SOQL) which is used to access Salesforce records using API and HTTP methods. [3] briefs about the Software Testing Life Cycle (STLC), Software Release Life Cycle (SRLC) and [3][5] explains about the different metrics used for testing. [4] talks about the various types of testing and the testing tool available in the market for different kinds of testing. [6] and [11] talks about the data driven automation testing framework and algorithm used for implementing the

framework. [7] explains about a testing framework along with the steps and algorithm used to develop the framework. The framework uses Selenium as the backend to perform testing. [8] talks about the testing frameworks used for different types of testing such as linear test automation framework, modular test automation framework, data driven test along with their advantages and disadvantages. [9] talks about the various manual and automation testing methods along with their advantages and disadvantages. The paper also presents a comparison between different types of testing such as static vs. dynamic testing, white box vs black box testing etc. [10] covers the concept of testing, its role in assuring quality, test cases, levels of testing, methods of testing and test planning, executing and monitoring. The papers emphasize on the use and impact of test driven environment with concept of story board based implementation. The main driving force behind [11] is to study the implications of software testing processes over the quality of software produced by the organizations at initial level.

III. METHODOLOGY

The idea is to use the scripts to invoke the Salesforce API's and Siebel Web services to store multiple records in one shot. The HTTP POST method is used to send request to the server. The request method takes three parameters i.e. URL of the Salesforce API or Siebel web service, input data and headers.

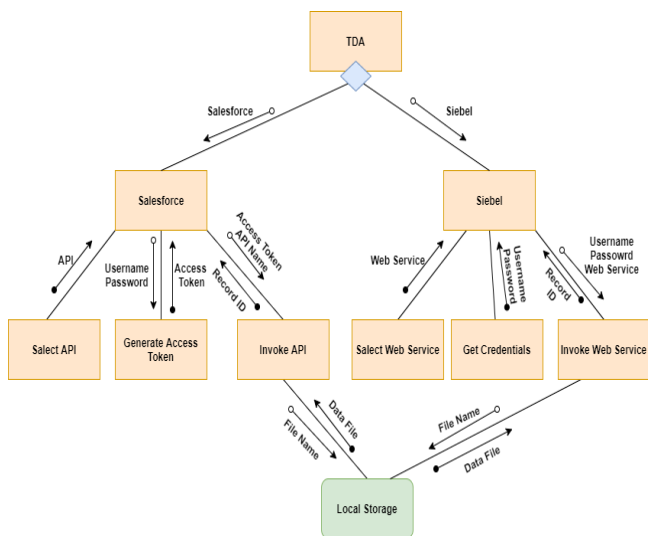


Figure 1. Structure Chart

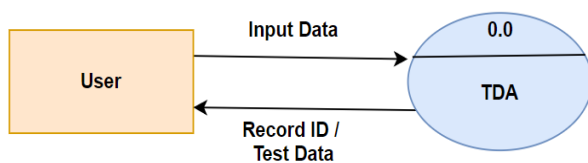


Figure 2. High Level Overview of the tool

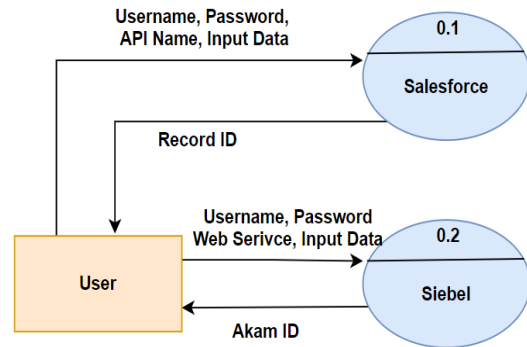


Figure 3. Overview of the modules

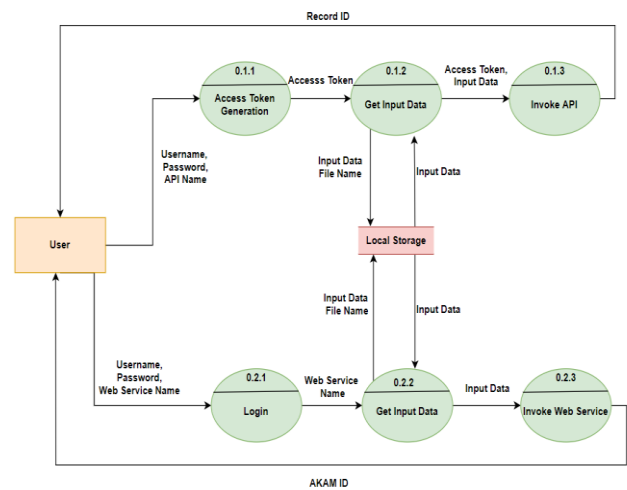


Figure 4. Detailed Data Flow of the tool

3.1 Salesforce

The input test data is stored in the JSON format. The test data should be stored in a key:value format where key should correspond the field of the Salesforce object.

```
{
  "AccountId" : "001A0000004uqhGIAQ",
  "RecordTypeId" : "012G000000z10uIAA",
  "Request_Sub_Type_c" : "N/A - Not Applicable",
  "Request_Type_c" : "Manual Invoice Review",
  "Service_c" : "Trial / PoC",
  "Description" : "Case Description",
  "Do_Not_Show_In_Portal_Picklist_c" : "Customer",
  "Subject" : "Case Subject",
  "Origin" : "Akamai"
},
```

Figure 5. Sample Salesforce Input Test Data

To invoke any Salesforce API, we need an access token. Access token can be generated by invoking the POST request with the parameters client id, client secret, username and password. The server responds with the access token if the passed credentials are valid. This access token is then used to invoke the API. This API inserts the records specified in the input JSON file into the Salesforce object specified by the API. The URL format of Salesforce API is :

“https://<<organization-name>>.salesforce.com/services/data/v20.0/objects/<<API Name>>”.

The server responds with the response message which includes various fields such as record id, error and status. The record id can then be used to identify each inserted record uniquely. By storing the test data in the JSON file, the time required for test data setup during retesting can be eliminated by invoking the API with the existing test data file.

3.2 Siebel

The input data is stored in the XML file. The XML tag corresponds to the field and XML value corresponds to the data to be inserted. Username and Password is provided as part of the XML input file to invoke the web service. Secured connection with the Siebel application is established by making use of certificates and keys. The Siebel object into which a record must be inserted is determined by the SOAPAction field in the header part of the POST request. The parameters provided with the POST request include URL of the server, input data in the form of XML file, header, and certificates.

The SOAPAction to invoke the Account web service is

```
'SOAPAction': "document/http://www.<<domain-name>>.com/uds/UIDataContact:InsertUIContact"
```

The server responds with the XML file which includes Siebel record id if the record is successfully inserted or the error code and error description if the execution fails to insert a record.

IV. RESULTS AND DISCUSSION

The tool has been tested for three applications i.e. PS Case (Professional Service) and Financial Force which are the applications of Salesforce and Contract is an object in Siebel. The time required to create the test data for PS Case application is 30 hours and the application needs to be tested once per two months with the same dataset as it undergoes changes in the implementation. Hence by using the automation tool, we need to create the test data just once and can use it for any number of times. Hence, $5 * 30 = 150$ hours can be saved per year.

In the second application i.e. Financial Force, the specific functionality needs to be tested once in every three months and approximately 18 hours are required to create and set the test data and therefore we can save 54 hours per year in setting up the test data for retesting. In the same way, Contract object of Siebel needs to be tested once per three months and it takes 15 hours to set up the test data for Contract object and hence by making use of TDA tool, we can save 45 hours per year.

Table 1. Experimental Results

Application	No. of times application needs to be tested per year	No. of hours required to create test data	No. of hours saved
PS Case	6	30	150
Financial Force	4	18	54
Contract	4	15	45

V. CONCLUSION AND FUTURE SCOPE

Test Automation provides a user-friendly interface that not only saves the time and manual effort required for test data creation but also provides reusability of test data for defect retesting. The no. of hours saved by using this tool increases with the no. of times the application is retested periodically. For instance, more time will be saved if we use the same test data for multiple cycles of testing. The total time saved by using the tool is not concrete and depends on the no. of times the application needs to be tested and on the complexity of the test data set up. We saved 150 hours for PS case, 54 hours in case of financial force and 45 hours in case of Contract. Following features are to be added to the Test Automation framework.

- Reporting of Test Case Execution
- Adding email functionality
- Integrating TDA framework with other testing tools
- Making the framework more secure

ACKNOWLEDGMENT

I would like to express my deep gratitude towards my guide Dr. Nagaraja G. S. and Head of Computer Science Department Dr. Ramakanth Kumar P, for providing their guidance, enthusiastic encouragement, valuable and constructive suggestions during planning and development of this case study paper. I also thank our beloved Principal Dr. K. N. Subramanya for their continuous support to carry out this project.

REFERENCES

- [1] Arun Kumar Arumugam, "Software Testing Techniques and New Trends", *International Journal of Engineering Research & Technology (IJERT)*, **Volume 8, Issue 12, December 2019**
- [2] Rakesh Kumar, Yogeshwari Sharma, Sonu Agarwal, Pragya, Bhanu Bhushan Parashar, "Extremely effective CRM Solution Using Salesforce", *Journal of Emerging Technologies and Innovative Research (JETIR)*, **Volume 1 Issue 5, October 2014**
- [3] Hitesh Tahabildar, Bichitra Kalita, "Automated Software Test Data Generation", *International Journal of Computer Science & Engineering Survey (IJCSES)*, **Vol.2, No.1, Feb 2011**
- [4] Ms. Karuturi Sneha, Mr. Malle Gowda M, "Research on Software Testing Techniques and Software Automation Testing Tools", *International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS-2017)*, **12 August 2017**
- [5] Tarik Sheth, Dr. Santosh Kumar Singh, "Software Test Automation- Approach on evaluating test automation tool", *International Journal of Scientific and Research Publications*, **Volume 5, Issue 8, August 2015**
- [6] Prof Vina M Lomte, Rishikesh Chandra, Ayush Gondhali, "Data Driven Automation Testing Framework", *International Journal of Emerging Engineering Research and Technology*, **Volume 2, Issue 7, October 2014**
- [7] Sonu Lamba, Vinay Rishiwal, Arjun Rana, "An Automated Data Driven Continuous Testing Framework", *International Journal of Advanced Technology In Engineering And Science*, **Volume No 03, Special Issue No. 01, February 2015**
- [8] Khalid Eldrandaly, Mahmoud Abd ElLatif, Nora Zaki, "Comparative Study of Software Test Automation Frameworks",

International Journal of Engineering Trends and Technology (IJETT), Volume 67, Issue 11, Nov 2019

- [9] Shruti Malve and Pradeep Sharma, "Investigation of Manual and Automation Testing using Assorted Approaches", *International Journal of Scientific Research in Computer Science and Engineering*, Volume 5, Issue 2, 2017
- [10]Nirali Honest, "Role of Testing in Software Development Life Cycle", *International Journal of Computer Sciences and Engineering*, Vol.7, Issue.5, pp.886-889, 2019
- [11]R. Sharma, R. Dadhich, "Implications of Software Testing Strategies at Initial Level of CMMI: An Analysis," *International Journal of Computer Sciences and Engineering*, Vol.6, Issue.5, pp.1055-1061, 2018.

AUTHORS PROFILE

Rahul Pandurang Pokale An undergraduate student studying Computer Science of Engineering in R.V. College of Engineering, Bangalore, India. His areas of interest include Web Development, Android Application Development and Cloud Computing



Dr. Nagaraja G. S. is a professor and associate dean at Department of Computer Science and Engineering in R. V. College of Engineering, Bangalore, India. He has 25 years of teaching experience. His areas of interest include Computer Networks & Management, Multimedia Communications, Computer Architecture, Protocol Design

