Table of Contents

[1. API Refresh Strategy – Large Scale 2](#_Toc119334320)

[1.1 Perquisites 2](#_Toc119334321)

[1.2 Demo Operators 2](#_Toc119334322)

[1.3 Case Type – Customer360 2](#_Toc119334323)

[1.3.1 Primary Stages 2](#_Toc119334324)

[1.3.2 Alternate Stages 2](#_Toc119334325)

[1.4 Micro Services 3](#_Toc119334326)

[1.5 Approaches 3](#_Toc119334327)

[1. Refresh Strategy Pattern 3](#_Toc119334328)

[2. Copy Page Pattern 3](#_Toc119334329)

[2. API Audit Strategy – Large Scale 4](#_Toc119334330)

[2.1 Perquisites 4](#_Toc119334331)

[2.2 Demo Operators 4](#_Toc119334332)

[2.3 Micro Services 4](#_Toc119334333)

[2.4 Approaches 5](#_Toc119334334)

[2.4.1. Audit Table - Approach 5](#_Toc119334335)

[2.4.2 Custom Table - Approach 8](#_Toc119334336)

[3. Resolve Cases – In Bulk 9](#_Toc119334337)

[3.1 Perquisites 9](#_Toc119334338)

[3.2 Demo Operators 9](#_Toc119334339)

[3.3 Approaches 9](#_Toc119334340)

[3.3.1. Audit Table - Approach 9](#_Toc119334341)

# 1. API Refresh Strategy – Large Scale

## 1.1 Perquisites

* Import RAP using LSA\_010101\_08122022.jar by logging as administrator into system
* Clone git repo git clone <https://github.com/guruprince8/api.git> into local folders then execute commands
  + npm install
  + npm start

## 1.2 Demo Operators

* User [admin@lsa.com](mailto:admin@lsa.com) / install to use developer studio
* [customer360approver1@lsa.com](mailto:customer360approver1@lsa.com) / install to validate the pattern

## 1.3 Case Type – Customer360

### 1.3.1 Primary Stages

|  |  |  |
| --- | --- | --- |
| Create | Review | Resolve |
| Create | Review | Resolve |

### 1.3.2 Alternate Stages

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

## 1.4 Micro Services

* [Get Customers](http://localhost:3000/customer-management-service/v1/customers) - Please update the endpoint URL <http://localhost:3000/customer-management-service/v1/customers> in Application Setting **BaseURL4**

## 1.5 Approaches

### 1. Refresh Strategy Pattern

|  |  |  |
| --- | --- | --- |
| Approach | Pros | Cons |
| Reference Pattern | 1. OOTB refresh pattern available at Page mode by default to load micro service data page  2. Refresh pattern applied declaratively when the data object is referred in displaying the screens either Perform / Review / Confirm / New harness | 1. Applications team needs to apply the different sources to pick during the load of micro service API data pages  2. Every future connector needs to follow the same design pattern to avoid trips to micro services  3. Completely inclined towards code change |

### 2. Copy Page Pattern

|  |  |  |
| --- | --- | --- |
| Approach | Pros | Cons |
| Copy Pattern | 1. OOTB refresh pattern available at Page mode by default to load micro service data page  2. Refresh pattern applied declaratively when the data object is referred in displaying the screens either Perform / Review / Confirm / New harness | 1. Applications team needs to apply the different sources to pick during the load of micro service API data pages  2. Every future connector needs to follow the same design pattern to avoid trips to micro services  3. Completely inclined towards code change |

# 2. API Audit Strategy – Large Scale

## 2.1 Perquisites

* 8.7 CS Framework installed PRPC instance
* Import RAP using ECRM\_010101\_09152022.jar by logging as administrator into system
* Clone git repo git clone <https://github.com/guruprince8/api.git> into local folders then execute commands
  + npm install
  + npm start

## 2.2 Demo Operators

* User ECRMAuthor / Sep@2022 to use developer studio

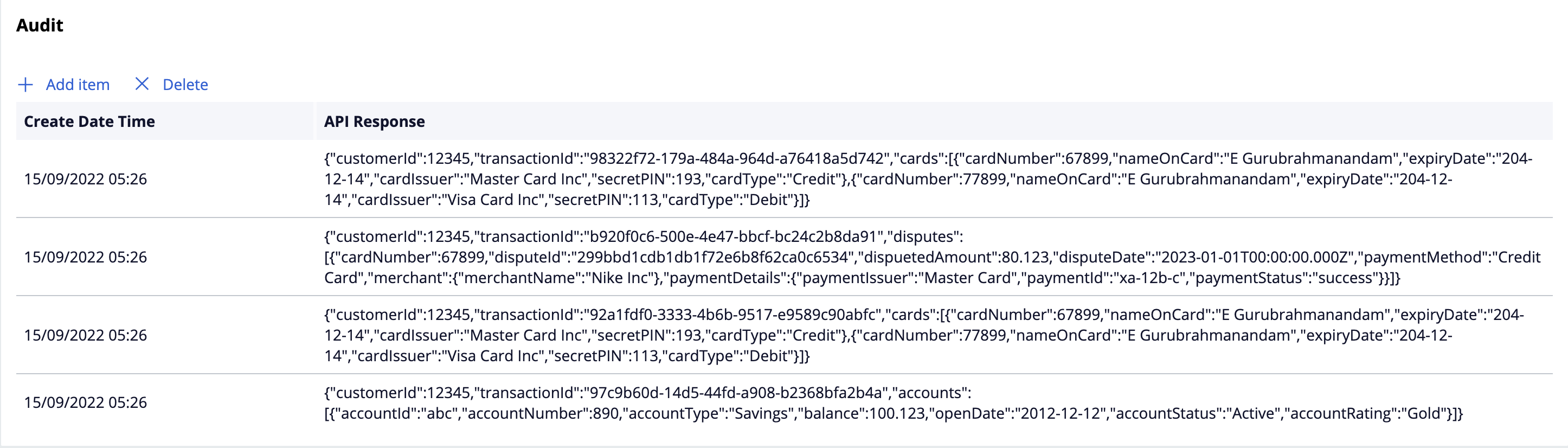
## 2.3 Micro Services

|  |  |
| --- | --- |
| API Name | API URL |
| Accounts | http://localhost:3000/account-management-service/v1/accounts/summary |
| Cards | http://localhost:3000/account-management-service/v1/cards/summary |
| Disputes | http://localhost:3000/account-management-service/v1/disputes/summary |
| Loans | <http://localhost:3000/account-management-service/v1/loans/summary> |
| Playground | http://localhost:3000/customer-management-service/v1/customers/playground |

## 2.4 Approaches

### 2.4.1. Audit Table - Approach

|  |  |  |
| --- | --- | --- |
| Approach | Pros | Cons |
| OOTB Audit Table | 1. OOTB history properties associated with every API call | 1. API execution gets stores into multiple table as history table is defined for each work pool |



Graphical user interface, text, application

Description automatically generated

OOTB App Studio Data Source Consumption View – WIP – Same can be adopted in pretty print

Graphical user interface, application

Description automatically generated

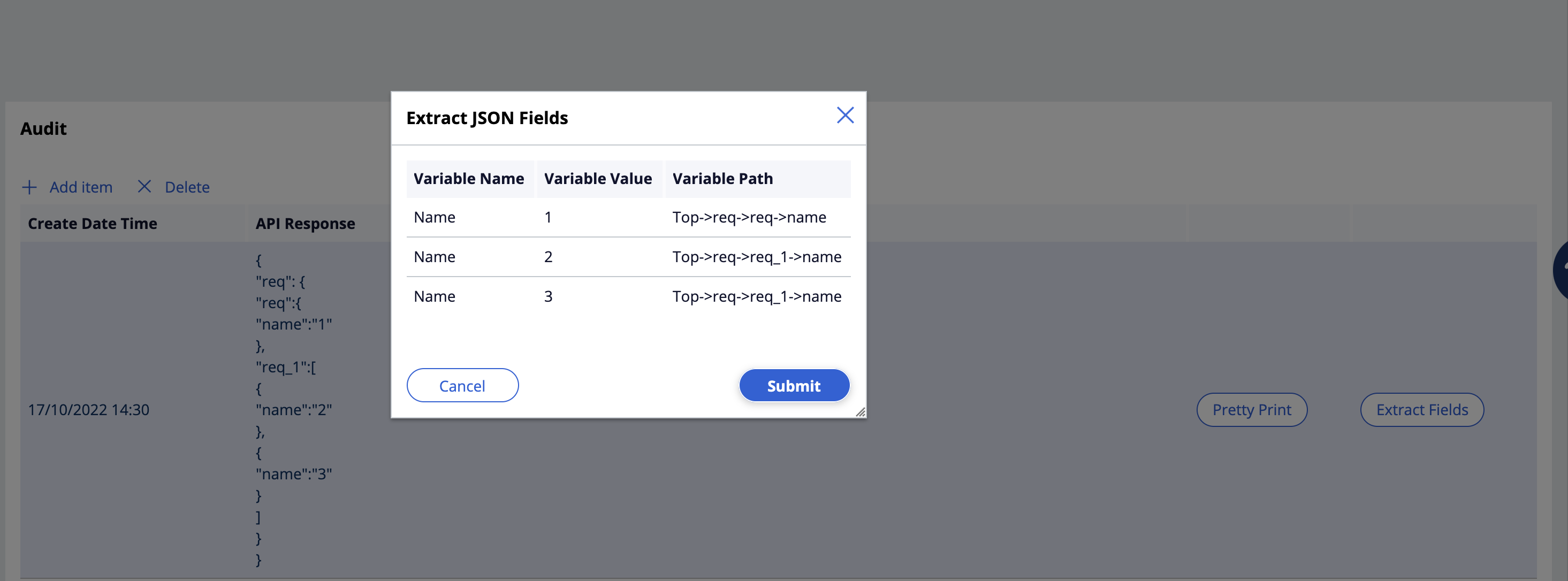
JSON Extract Fields



Graphical user interface, application, Teams

Description automatically generated

JSON Extract along variable path



Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

### 2.4.2 Custom Table - Approach

|  |  |  |
| --- | --- | --- |
| Approach | Pros | Cons |
| Dedicated Table | 1. Dedicated API table can be customized fully to support all sort of requirements  2. Dedicated table can be effectively used to create API execution summary reports and alert notifications  3. Single place holder to store all the API call’s  4. Performance issues can be addressed easily as there is no impact to OOTB functionality | 1. There will be new table introduce in the DATA schema  2. Maintenance lies with application team |

# 3. Resolve Cases – In Bulk

## 3.1 Perquisites

* 8.7 installed PRPC instance
* Import RAP using LSA\_87\_010101\_11142022.jar by logging as administrator into system

## 3.2 Demo Operators

* Enable [lsa@abc.com](mailto:lsa@abc.com) operator id once RAP is imported
* User lsa@abc.com / December@2022 to use developer studio

## 3.3 Approaches

### 3.3.1. Audit Table - Approach

|  |  |  |
| --- | --- | --- |
| Approach | Pros | Cons |
| Activity | 1. OOTB pxForceCaseClose takes care of auto commit / rollback policies |  |

Graphical user interface, application, website

Description automatically generated