**API Specification Document - Rule Engine**

**DOCUMENT VERSION – 0.04**

**21-February-2022**

**DOCUMENT HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Document Revision Description** | **Document Author** |
| 15-Apr-2021 | 0.01 | Draft version | Indumathi.P |
| 31-Jan-2022 | 0.02 | API Related changes | Mohaneswaran P |
| 02-02-2022 | 0.03 | Incorporated internal review comments | Mohaneswaran P |
| 21-02-2022 | 0.04 | Added API details of Utility Rule and details on 2 more SIB rule details added. | Mohaneswaran P |

Table of Contents

[1. Introduction 4](#_Toc96612409)

[1.1 Purpose Of this Document 4](#_Toc96612410)

[2. Rule Engine Execute Rule API 4](#_Toc96612411)

[2.1 Single Rule Execution 6](#_Toc96612412)

[2.1.1 Single Rule Execution – Sample Input and output (Success Scenario) 7](#_Toc96612413)

[2.1.2 Single Rule Execution – Sample Input and output (Failure Scenario) 8](#_Toc96612414)

[2.2 Multiple Rule Execution 9](#_Toc96612415)

[2.2.1 Multiple Rule Execution – Sample Input and output (Success Scenario) 9](#_Toc96612416)

[2.2.2 Multiple Rule Execution – Sample Input and output (Failure Scenario) 10](#_Toc96612417)

[2.3 Group Rule Execution 13](#_Toc96612418)

[2.3.1 Group Rule Execution – Sample Input and output (Success Scenario) 13](#_Toc96612419)

[2.3.2 Group Rule Execution – Sample Input and output (Failure Scenario) 14](#_Toc96612420)

[2.4 Rule Utility 16](#_Toc96612421)

[2.4.1 Rule Util Execution – Sample Input and output (Success Scenario) 16](#_Toc96612422)

[2.4.2 Rule Util Execution – Sample Input and output (Failure Scenario) 16](#_Toc96612423)

[2.5 Get unique variable list – Sample Input and output 18](#_Toc96612424)

[2.6 Rule Details Configured for SIB 19](#_Toc96612425)

[2.6.1 Top-Up Eligibility 19](#_Toc96612426)

[2.6.2 Credit Card Eligibility 20](#_Toc96612427)

[2.6.3 Vehicle Marhaba - Salaried Employees 21](#_Toc96612428)

[2.6.4 Vehicle Marhaba -Self Employees 22](#_Toc96612429)

[2.7 Group Rule Details Configured for SIB 23](#_Toc96612430)

[2.7.1 Cross Sell 23](#_Toc96612431)

[2.8 Rule Util Details Configured for SIB 24](#_Toc96612432)

[2.8.1 Auto Finance Eligible Amount 24](#_Toc96612433)

# Introduction

This document is the reference document that details the APIs required for the executing the Rules configured in Rule Engine.

## Purpose Of this Document

This document is intended for developers or end users who consume the rule Engine APIs to their systems.

# Rule Engine Execute Rule API

The rules defined in the Rule Engine web page can be invoked by any other system by invoking the Execute Rule API.

The Execute Rule API is a RESTful API. This API use HTTP method POST and supports the JSON (JavaScript Object Notation) format for input and output.

Once the Rule is been defined in the UI, the same can be invoked by any other business module by invoking the executeRule API method.

**Sample URL**

https://<Serverhostname(or)ip>/rule/ruleEngine/executeRules

**URL**

https://cognicx.app/rule/ruleEngine/executeRules

**Method**

Post

**Success Response**

200

When execute Rules API execution is successful the response code will be 200 and this will include response body.

**Error Response**

400 – Bad Request

401 – Unauthorized

403 – Forbidden

404 – Not Found

405 – Method not allowed

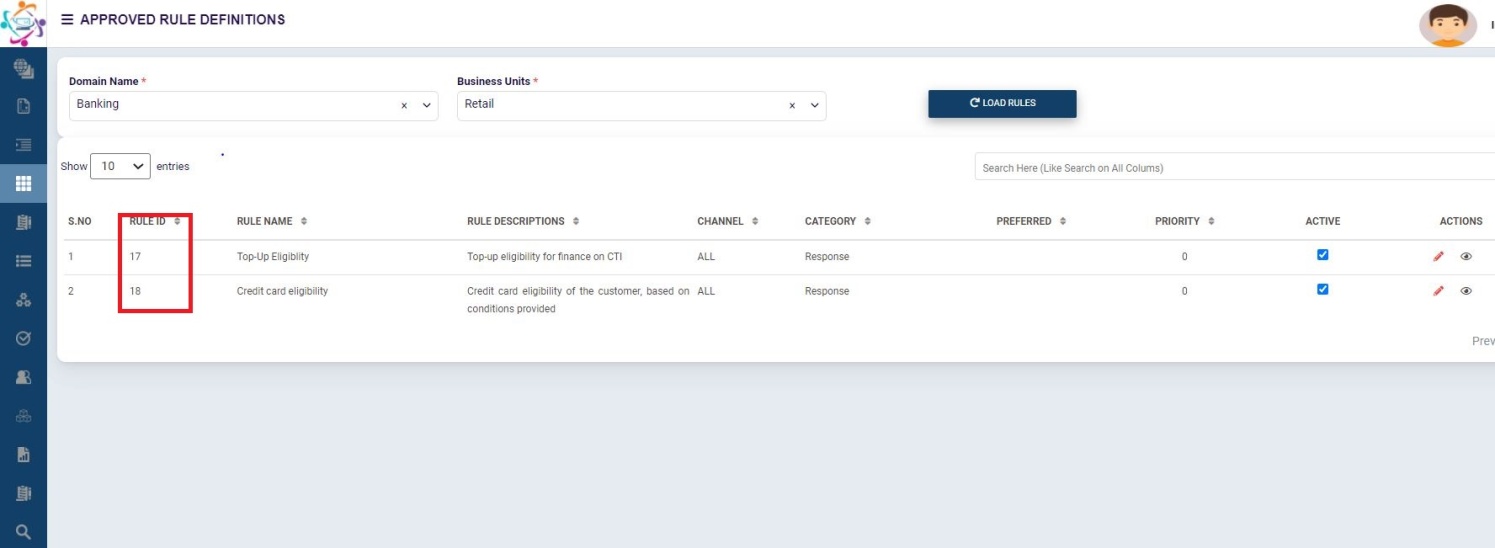
412- Preconditions failed

415 – Unsupported Media type

500 – Internal Server Error

**Approved Rule Details**

Ruleexecution can be done only for the approved rules. The list of approved rules along with details are available in Approved Rule Definition page. In Rule Engine User Interface this page can be navigated by clicking fourth icon on the left side menu bar.



The rules Ids are mandatory to execute the Rule. The Rule Id can be identified in the approved Rule Page in RuleID column as highlighted in above image.

In Rule Engine, Rules can be executed in below ways according to the number of Rules and its type.

1. Single Rule Execution
2. Multiple Rule Execution
3. Group Rule Execution.

These are detailed in below section.

**Note:**

1. All three types can be executed by calling the same URL but parameter varies.

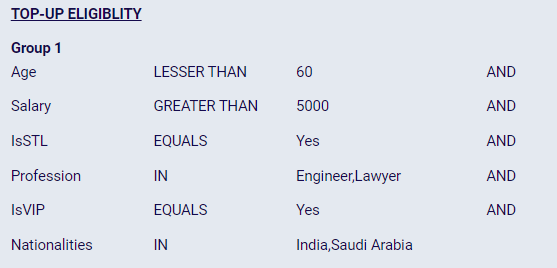
https://<Serverhostname(or)ip>/rule/ruleEngine/executeRules

2. The parameters to be passed for each rule can be fetched by getUniqueRules API which is detailed in section 2.4.[Get unique variable list – Sample Input and output](#_Get_unique_variable)

## Single Rule Execution

Whenever there is only one rule need to be executed, this type of execution will be followed. In this only one specific rule details will be provided as input and conditions configured in that rule are validated against the given input. As a response output the rule execution detail i.e. true or false is returned along with additional rule execution details.

**Rule Details**

**Sample screen of Rule Definition from Rule Engine Simulation Page**

### Single Rule Execution – Sample Input and output (Success Scenario)

**Input and Output JSON format** (Success)

 ****

### Single Rule Execution – Sample Input and output (Failure Scenario)

**Input and Output JSON format** (Failure)

** **

**Sample JSON file - Single Rule Execution:**

****

**For easy reference, the sample input and output parameters are embedded here.**

## Multiple Rule Execution

Whenever there is requirement to execute more than one rule this Multiple Rule Execution can be used. In this more than one rule details can be provided as input and conditions configured in that rule are validated against the given input. As a response output the rule execution detail i.e. true or false is returned for each rule along with additional rule execution details.

**Sample screen of Rule Definition from Rule Engine Simulation Page**

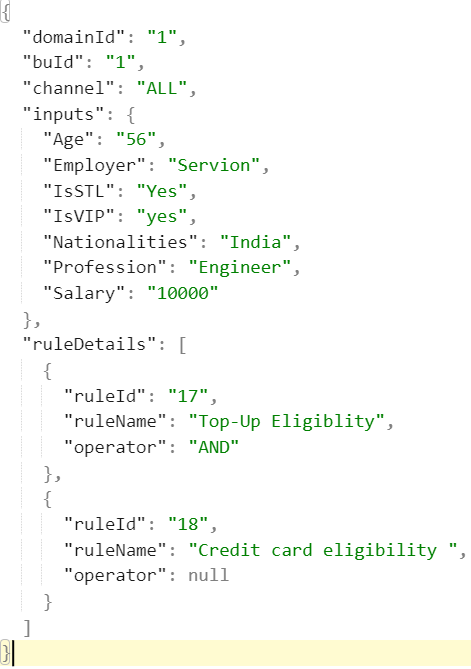
**Rule Details**

****

In Multiple Rule execution, the rules can be executed by providing AND or OR condition between them.

### Multiple Rule Execution – Sample Input and output (Success Scenario)

**Input and Output JSON format for two rule execution** (Success)

** **

### Multiple Rule Execution – Sample Input and output (Failure Scenario)

**Input and Output JSON format for two rule execution** (Failure)

** **

**Sample JSON file – Multiple Rule Execution:**

****

**Rule Execution Request Field Description:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Field** | **Description** | **Sample Input** |
| 1 | domainId | Particular domain id for which the rule is mapped in Rule engine application. For SIB, the created rules are mapped under banking domain. Hence the ID is 1. | **1** |
| 2 | buId | Particular Business unit id for which the rule is mapped in Rule engine application. For SIB the created rules are mapped under banking domain and Retail business unit and the ID is 1 | **1** |
| 3 | channel | This is to identify from which channel the rule is getting executed | CTI |
| 4 | inputs | * List of variables for which the data need to be mapped and sent as input for rule Execution. * Input details which will be validated against the conditions configured in the rule. * This list of input fields can be dynamically fetched using a API, those details are mentioned in [section 2.4](#_Get_unique_variable) | "inputs": {  "Age": "56",  "Employer": "Servion",  "IsSTL": "yes",  "IsVIP": "yes",  "Nationalities": "India",  "Profession": "Engineer",  "Salary": "10000"  }, |
| 5 | ruleDetails | This is parent element for specifying Rule Details. |  |
| 6 | ruleId | Here the rule name and Rule ID to be executed will be passed. The rule id can be seen from the Approved Rule definition page of Rule Engine User Interface application. | 17 |
| 7 | ruleName | It is a non mandatory field, this ruleName is displayed in application’s Approved Rule definition page, so along with ruleId the name is also passed as input in this field. | Top-Up Eligibility |
| 8 | operator | This parameter is used to specify the logical operator AND or OR in case of Multiple rules.  And this will be null in case of single Rule. In case of multiple Rules operator value will be “OR” or “AND”. | AND |

**Rule Execution Response Field Description:**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **Field** | **Description** | **Sample Output** |
| 1 | status | Rest API http response status. | **200** |
| 2 | message | Success or failure message for API request. | **Success** |
| 3 | value | Rule execution related details are provided in this field only. |  |
| 4 | resultDetail | Execution related details of each rule comes under this field. |  |
| 5 | ruleResult | Boolean field which returns **True** if the rule execution is success for provided data, otherwise **False.** |  |
| 6 | reason | If **ruleResult** field is **False** then reason for that is given here, if it is **True** then success. |  |

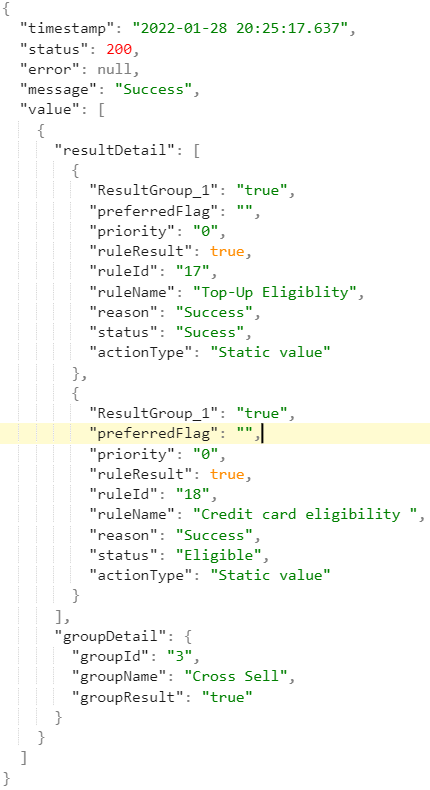
## Group Rule Execution

When multiple rules to be executed as a group, this will be used. The advantage is, if there are more rules are added with same parameters. Need not pass the individual Rule ids in case of multiple rules. Group rule are multiple rules which are mapped in a single group, inputs are provided for the rules mapped under a group and based on the configured conditions inputs are validated. As a response output, the rule execution detail i.e. true or false is returned for each rule along with additional rule execution details.

**Group Details**

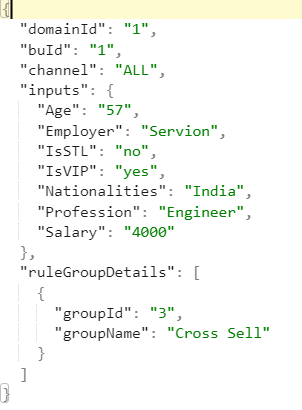
### Group Rule Execution – Sample Input and output (Success Scenario)

**Input and Output JSON for Group Execution** (Success)

** **

### Group Rule Execution – Sample Input and output (Failure Scenario)

**Input and Output JSON for Group Execution**

**Sample JSON file – Group Rule Execution:**

**Rule Execution Request Field Description:**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **Field** | **Description** | **Sample Input** |
| 1 | domainId | ID for domain. Particular domain id for which the rule is mapped in Rule engine application. For Banking domain ID is 1.. | **1** |
| 2 | buId | Particular Business unit id for which the rule is mapped in Rule engine application. The created rules are mapped under banking domain and Retail business unit. | **1** |
| 3 | channel | This is to identify from which channel the rule is getting executed. This can be CTI or IVR as per the channel given during Rule Creation. | CTI |
| 4 | inputs | * List of variables for which the data need to be mapped and sent as input for rule Execution. * Input details which will be validated against the conditions configured in the rule. * This list of input fields can be dynamically fetched using a API, those details are mentioned in [section 2.4](#_Get_unique_variable) | "inputs": {  "Age": "56",  "Employer": "Servion",  "IsSTL": "yes",  "IsVIP": "yes",  "Nationalities": "India",  "Profession": "Engineer",  "Salary": "10000"  }, |
| 5 | ruleGroupDetails | Group detail object, where respective group id and group name are provided for which the execution happens |  |
| 6 | groupId | This group Id will be displayed in Group rule module. |  |
| 7 | groupName | It is a non mandatory field, this groupName is displayed in application’s Group Rule module. | Cross sell |

**Rule Execution Response Field Description:**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **Field** | **Description** | **Sample output** |
| 1 | status | Rest API http response status. | **200** |
| 2 | message | Success or failure message for API request. | **Success** |
| 3 | value | Rule execution related details are provided in this field only. |  |
| 4 | resultDetail | Execution related details of each rule comes under this field. |  |
| 5 | ruleResult | Boolean field which returns **True** if the rule execution is success for provided data, otherwise **False.** |  |
| 6 | reason | If **ruleResult** field is **False** then reason for that is given here, if it is **True** then success. |  |
| 7 | groupDetail | This field has the details for which the group execution happened. |  |

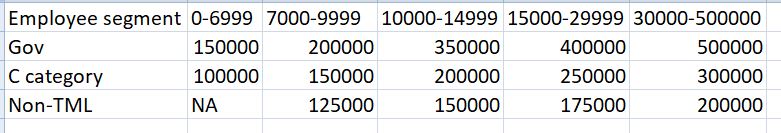
## Rule Utility

Rule Utility is the support functionality to configure rules based on 2-dimensional matrix.

By using this utility, if any value be returned based on two conditions, it can be achieved.

**Sample Rule Util Details (Configured for SIB)**

The details of excel uploaded in Rule Utility for SIB need is as below.



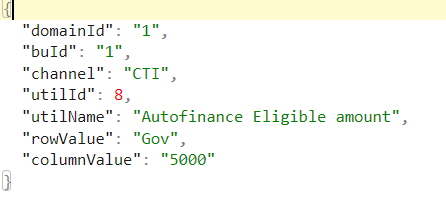
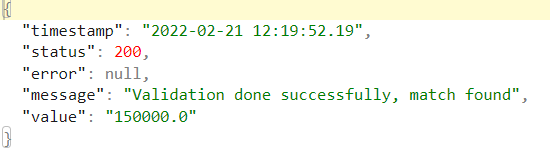
It has Employee segment mapped against salary range.

The Employee segments configured are Gov, C Category and Non-TML

The Salary ranges are 0-6999, 7000-9999,1000-14999, 15000-29999 and 30000-50000

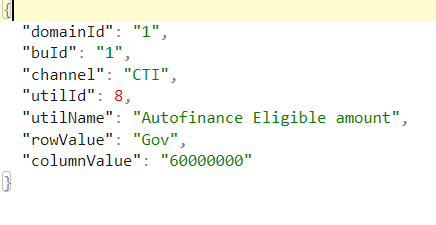
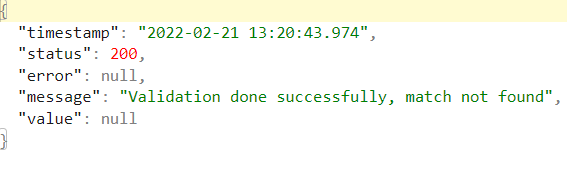
### Rule Util Execution – Sample Input and output (Success Scenario)

**Input and Output JSON for Rule Util Execution** (Success)

### Rule Util Execution – Sample Input and output (Failure Scenario)

**Input and Output JSON for Rule Util Execution** (Failure)

**Sample JSON file – Rule Util Execution:**

**Rule Util Execution Request Field Description:**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **Field** | **Description** | **Sample Input** |
| 1 | domainId | ID for domain. Particular domain id for which the rule is mapped in Rule engine application. For Banking domain ID is 1.. | **1** |
| 2 | buId | Particular Business unit id for which the rule is mapped in Rule engine application. The created rules are mapped under banking domain and Retail business unit. | **1** |
| 3 | channel | This is to identify from which channel the rule is getting executed. This can be CTI or IVR as per the channel given during Rule Creation. | CTI |
| 4 | utilId | Rule Utility Id for which the execution needs to be done. This utilId can be obtained from Approved Rule Util tab in Rule Utility module | **8** |
| 5 | utilName | Name of the Utility, which is a optional field | **Autofinance Eligible amount** |
| 6 | rowValue | Any of the row value in the uploaded excel | Gov |
| 7 | columnValue | Any of the column value in uploaded excel, if Util type is normal then direct value in the column will be validated. If util type is Range check then any value in the column range can be validated. | **5000** |

**Rule Util Execution Response Field Description:**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **Field** | **Description** | **Sample output** |
| 1 | status | Rest API http response status. | **200** |
| 2 | message | Success or failure message for API request. | **Validation done successfully, match found** |
| 3 | error | If there is any internal server error, the message will be returned in this field | **null** |
| 4 | value | When the given input is validated and the match found means then that particular value is returned. If the match is not found means null is returned. | **150000** |

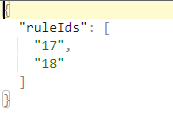
## Get unique variable list – Sample Input and output

This API is a supporting API to get the list of input parameters required to passed for executing the rule. If the input parameters are identified by consuming application once, this method is not mandatory.

If any parameters for the Rule changes, this would help to get the updated parameters.

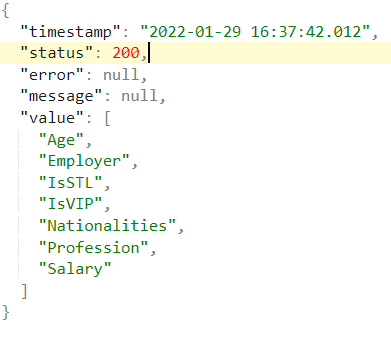
Sample URL: https://<Serverhostname(or)ip>/rule/ruleEngine/getUniqueVariables

**Input Request:**

****

**Output Response:**

The **value** field here provides the list of variables for which the data need to be mapped and sent as input for rule Execution.

****

**Sample JSON file – Variable List:**

****

## Rule Details Configured for SIB

For SIB couple of rules are configured in UAT server, those rule details along input request are mentioned here

### Top-Up Eligibility



**Rule Execution Request Input:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SNO** | **Field** | **Type** | **Mandatory** | **Description** | **Input** |
| 1 | domainId | Static | yes | When rule is created a particular Domain will be mapped | **1** |
| 2 | buId | Static | yes | When rule is created a particular Business unit will be mapped | **1** |
| 3 | channel | Dynamic | yes | This is to identify from which channel the rule is getting executed | CTI |
| 4 | inputs | Dynamic | yes | This list of input fields can be dynamically fetched using a API, those details are mentioned in [section 2.4.](#_Get_unique_variable) Input data is mapped with these fields and passed as input. | Refer attached input JSON file |
| 5 | ruleDetails | Static | yes | List of rule detail object, for single rule it has only one rules Id and name where the operator value is null. If it’s multiple rule execution means then multiple rule details along with operator value as logical AND OR | Refer attached input JSON file |
| 6 | ruleId | Static | yes | This ruleId is displayed in application’s Approved Rule definition page, so for executing a particular rule respective ruleId is passed as input in this field. | 17 |
| 7 | ruleName | Static | Optional | It is a non-mandatory field, this ruleName is displayed in application’s Approved Rule definition page, so along with ruleId the name is also passed as input in this field. | Top-Up Eligibility |
| 8 | operator | Dynamic | Optional | Mandatory for multiple rule execution, to mention a logical operation between different rules this field is used. Where in case of single rule execution it is null | null |

### Credit Card Eligibility



**Rule Execution Request Input:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SNO** | **Field** | **Type** | **Mandatory** | **Description** | **Input** |
| 1 | domainId | Static | yes | When rule is created a particular Domain will be mapped | **1** |
| 2 | buId | Static | yes | When rule is created a particular Business unit will be mapped | **1** |
| 3 | channel | Dynamic | yes | This is to identify from which channel the rule is getting executed | CTI |
| 4 | inputs | Dynamic | yes | This list of input fields can be dynamically fetched using a API, those details are mentioned in section 2.4. Input data is mapped with these fields and passed as input. | Refer attached input JSON file |
| 5 | ruleDetails | Static | yes | List of rule detail object, for single rule it has only one rules Id and name where the operator value is null. If it’s multiple rule execution means then multiple rule details along with operator value as logical AND OR | Refer attached input JSON file |
| 6 | ruleId | Static | yes | This ruleId is displayed in application’s Approved Rule definition page, so for executing a particular rule respective ruleId is passed as input in this field. | 18 |
| 7 | ruleName | Static | Optional | It is a non-mandatory field, this ruleName is displayed in application’s Approved Rule definition page, so along with ruleId the name is also passed as input in this field. | Credit card Eligibility |
| 8 | operator | Dynamic | Optional | Mandatory for multiple rule execution, to mention a logical operation between different rules this field is used. Where in case of single rule execution it is null | null |

### Vehicle Marhaba - Salaried Employees



**Vehicle Marhaba - Salaried Employees Rule Execution Request Input:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SNO** | **Field** | **Type** | **Mandatory** | **Description** | **Input** |
| 1 | domainId | Static | yes | When rule is created a particular Domain will be mapped | **1** |
| 2 | buId | Static | yes | When rule is created a particular Business unit will be mapped | **1** |
| 3 | channel | Dynamic | yes | This is to identify from which channel the rule is getting executed | CTI |
| 4 | inputs | Dynamic | yes | This list of input fields can be dynamically fetched using a API, those details are mentioned in section 2.4. Input data is mapped with these fields and passed as input. | Refer attached input JSON file |
| 5 | ruleDetails | Static | yes | List of rule detail object, for single rule it has only one rules Id and name where the operator value is null. If it’s multiple rule execution means then multiple rule details along with operator value as logical AND OR | Refer attached input JSON file |
| 6 | ruleId | Static | yes | This ruleId is displayed in application’s Approved Rule definition page, so for executing a particular rule respective ruleId is passed as input in this field. | 19 |
| 7 | ruleName | Static | Optional | It is a non-mandatory field, this ruleName is displayed in application’s Approved Rule definition page, so along with ruleId the name is also passed as input in this field. | Vehicle Marhaba - Salaried Employees |
| 8 | operator | Dynamic | Optional | Mandatory for multiple rule execution, to mention a logical operation between different rules this field is used. Where in case of single rule execution it is null | null |

### Vehicle Marhaba -Self Employees



**Vehicle Marhaba - Salaried Employees Rule Execution Request Input:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SNO** | **Field** | **Type** | **Mandatory** | **Description** | **Input** |
| 1 | domainId | Static | yes | When rule is created a particular Domain will be mapped | **1** |
| 2 | buId | Static | yes | When rule is created a particular Business unit will be mapped | **1** |
| 3 | channel | Dynamic | yes | This is to identify from which channel the rule is getting executed | CTI |
| 4 | inputs | Dynamic | yes | This list of input fields can be dynamically fetched using a API, those details are mentioned in section 2.4. Input data is mapped with these fields and passed as input. | Refer attached input JSON file |
| 5 | ruleDetails | Static | yes | List of rule detail object, for single rule it has only one rules Id and name where the operator value is null. If it’s multiple rule execution means then multiple rule details along with operator value as logical AND OR | Refer attached input JSON file |
| 6 | ruleId | Static | yes | This ruleId is displayed in application’s Approved Rule definition page, so for executing a particular rule respective ruleId is passed as input in this field. | 20 |
| 7 | ruleName | Static | Optional | It is a non-mandatory field, this ruleName is displayed in application’s Approved Rule definition page, so along with ruleId the name is also passed as input in this field. | Vehicle Marhaba -Self Employees |
| 8 | operator | Dynamic | Optional | Mandatory for multiple rule execution, to mention a logical operation between different rules this field is used. Where in case of single rule execution it is null | null |

**Note:**

For the Rules “**Vehicle Marhaba – Salaried Employees**” and “**Vehicle Marhaba – Self Employees**”

“Eligible Amount” as one of the Input parameters.

The procedure of arriving this amount is described below.

**Steps:**

1.CTI Should pass the parameters to Rule Engine Utility (as described [here](#_Rule_Util_Details)) to Get the Maximum Eligible Amount based on Employee segment and employee salary.

2. CTI will have the information of availed amount by customer by accessing other SIB service and Calculate the Eligible amount specific to customer (Eligible Amount = Maximum Eligible Amount – Amount Availed)

3. This final amount will be passed to the above rules.

## Group Rule Details Configured for SIB

For SIB one Group Rules is configured in UAT server, this group details along input request are mentioned here

### Cross Sell

Two rules which are mentioned in previous section are grouped into this **Cross Sell** group, in future new rules can be added to this group so that the dynamic portion of the input request may vary.



**Group Rule Execution Request Input:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SNO** | **Field** | **Type** | **Mandatory** | **Description** | **Sample Input** |
| 1 | domainId | Static | yes | Particular domain id for which the rule is mapped in Rule engine application. For now the created rules are mapped under banking domain. | **1** |
| 2 | buId | Static | yes | Particular Business unit id for which the rule is mapped in Rule engine application. For now the created rules are mapped under banking domain and Retail business unit. | **1** |
| 3 | channel | Dynamic | yes | This is to identify from which channel the rule is getting executed. | CTI |
| 4 | Inputs | Dynamic | yes | This list of input fields can be dynamically fetched using a API, those details are mentioned in [section 2.4](#_Get_unique_variable). Input data is mapped with these fields and passed as input. | Refer attached Input JSON file |
| 5 | ruleGroupDetails | Static | yes | Group detail object, where respective group id and group name are provided for which the execution happens. | Refer attached Input json file above. |
| 6 | groupId | Static | yes | This group Id will be displayed in Group rule module. | 3 |
| 7 | groupName | Static | Optional | It is a non-mandatory field, this groupName is displayed in application’s Group Rule module. | Cross Sell |

## Rule Util Details Configured for SIB

For SIB one Rule Utility named “**Autofinance Eligible amount**” is configured. In that the eligible amount as per SIB requirement based on various Employee segment and Salary. The details for invoking the API is detailed below.

### Auto Finance Eligible Amount

To get the eligible amount based on the employee segment and the salary this rule util is configured. Attached the sample input parameters to invoke the util API.



**AutoFinance Eligible Amount Util Request Input:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SNO** | **Field** | **Type** | **Mandatory** | **Description** | **Sample Input** |
| 1 | domainId | Static | yes | Particular domain id for which the rule util is mapped in Rule engine application. For now the created rules are mapped under banking domain. | **1** |
| 2 | buId | Static | yes | Particular Business unit id for which the rule util is mapped in Rule engine application. For now the created rules are mapped under banking domain and Retail business unit. | **1** |
| 3 | channel | Static | yes | This is to identify from which channel the rule is getting executed. | CTI |
| 4 | utilId | Static | yes | Rule Utility Id for which the execution needs to be done. This utilId can be obtained from Approved Rule Util tab in Rule Utility module | 1 |
| 5 | utilName | Static | No | Name of the Utility, which is a optional field | AutoFinance Eligible Amount |
| 6 | rowValue | Dynamic | Yes | It is the row header name of the utility excel uploaded through Rule Engine UI. For SIB requirement of **Autofinance Eligible amount** util one of the employee segments for whom we want to retrieve the eligible amount is passed. | Non-TML |
| 7 | columnValue | Dynamic | Yes | It is the column header name of the utility excel uploaded through Rule Engine UI. For SIB, Autofinance **Eligible amount** the employee salary will be passed  **Note:**  When the given input is validated and the match found then that particular value configured is returned. If the match is not found null is returned in the response | 8000 |

The output details are explained [here](#_Rule_Util_Execution)