

Liquidity Management Rest Services User Guide  
Oracle Banking Liquidity Management  
Release 14.0.0.0.0  
[November] [2017]



---

# Table of Contents

<b>1</b>	<b>PREFACE .....</b>	<b>3</b>
1.1	INTRODUCTION .....	3
1.2	AUDIENCE.....	3
1.3	DOCUMENTATION ACCESSIBILITY .....	3
1.4	ORGANIZATION.....	3
<b>2</b>	<b>REST API SERVICES - OVERVIEW .....</b>	<b>4</b>
2.1	INTRODUCTION .....	4
<b>3</b>	<b>REST API SERVICES - DETAILS .....</b>	<b>5</b>
3.1	REST API SERVICES.....	5
3.1.1	<i>Create Bank Services.....</i>	7
3.1.2	<i>Fetch Bank Service :.....</i>	8
3.1.3	<i>Create User Service:.....</i>	9
3.1.4	<i>Fetch User Service: .....</i>	10
3.1.5	<i>Create Customer Service:.....</i>	11
3.1.6	<i>Fetch Customer Service:.....</i>	12
3.1.7	<i>Create Account Service: .....</i>	13
3.1.8	<i>Fetch Account Service: .....</i>	14
3.1.9	<i>Create User Role Service: .....</i>	15
3.1.10	<i>Fetch User Role Service:.....</i>	15
3.1.11	<i>Create Structure Service: .....</i>	16
3.1.12	<i>Fetch Structure Service: .....</i>	26
3.1.13	<i>Sweep Execution Service:.....</i>	27
3.1.14	<i>Fetch Sweep Details Service: .....</i>	27
3.1.15	<i>View of Liquidity Structures Sweep,Pool,Hybrid, Enhancement Service:.....</i>	27
3.1.16	<i>Marking of a structure to Hold status Service: .....</i>	28
3.1.17	<i>Create Country Service: .....</i>	28
3.1.18	<i>Fetch Country Service:.....</i>	29
3.1.19	<i>Update Country Service: .....</i>	30
3.1.20	<i>Delete Country Service: .....</i>	30
3.1.21	<i>Create Branch Service: .....</i>	31
3.1.22	<i>Fetch Branch Service:.....</i>	31
3.1.23	<i>Update Branch Service: .....</i>	32
3.1.24	<i>Create Customer User Linkage Service: .....</i>	33
3.1.25	<i>Fetch Customer User Linkage Service:.....</i>	33
3.1.26	<i>Update Customer User Linkage Service: .....</i>	34
3.1.27	<i>Delete Customer User Linkage Service: .....</i>	35
3.1.28	<i>Authorize Customer User Linkage Service: .....</i>	35

## 1.1 Introduction

This User Guide is intended to familiarize you with the light-weight REST based services of Oracle Banking Liquidity Management. The manual gives you an overview of the services developed, input/request parameters and the response of the each REST service.

## 1.2 Audience

The manual is intended for integrating external systems to Oracle Banking Liquidity Management using RESTFUL framework.

## 1.3 Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

## 1.4 Organization

The manual is organized into the following chapters

<u>Chapter</u>	<u>Description</u>
<b>Chapter 1</b>	<i>Preface</i> - Gives information on the intended audience. It also lists the various chapters covered in this User Manual.
<b>Chapter 2</b>	<i>Rest API - An Overview</i> provides a snapshot of the features of the entire module.
<b>Chapter 3</b>	<i>Rest API Services</i> –Provides in-depth details of each of the rest APIs.

### 2.1 Introduction

Oracle Banking Liquidity Management has adopted the industry standard of providing light-weight services through Representational State Transfer (popularly known as **Rest**) services. These lightweight services support the current trend of mobile/tablet based applications. These services help to improve the response time for tablet/mobile applications, as these applications are very chatty, involves numerous hits to the server and the expected response time is minimal for each of the requests.

The REST service requests use POST methods.

The following are the services using POST methods:

1. Create Bank.
2. Create User.
3. Create Account.
4. Fetch Customer.
5. Create Structure.

More information on the above listed services can be found in the respective sections of the User manual.

Each of the services listed above will be a URL. These URLs can be tested using any of the available browser plug-ins (E.g. Postman, RestEasy etc.) Or these services can also be integrated with an API manager depending on its use. Below is the format of a create bank service example

Http :/( IP) :( Port)/BankSetupApplicationServiceProxy/create

For the POST services we need to provide data input in JSON format. An example of JSON input is shown below. The output of REST services will also be in JSON format. Sample input/output for each service is given under the respective sections of the User manual.

```
{
  "account" : "207000124025",
  "amount" : "1",
  "branch" : "207",
  "effectiveDate" : " 2015-08-02"
}
```

The list of services will be discussed in detail in the following chapters.

### 3.1 Rest API Services

Following are the list of Rest API Services that are available.

SL No.	Service	Method	Sample URL
1	Create Bank	POST	Http ://( IP) :( Port)/BankSetupApplicationServiceProxy/create
2	Fetch Bank	POST	Http ://( IP) :( Port)/BankSetupApplicationServiceProxy/fetch
3	Create user	POST	http://(IP):(Port)/ UserProxy/createUser
4	Fetch User	POST	http://(IP):(Port)/ UserProxy /fetchUser
5	Create Customer	POST	http://(IP):(Port)/CustomerSetupApplicationServiceProxy /create
6	Fetch Customer	POST	http://(IP):(Port)/ CustomerSetupApplicationServiceProxy /fetch
7	Create Account	POST	http://(IP):(Port)/AccountSetupApplicationServiceProxy/ create
8	Fetch Account	POST	http://(IP):(Port)/ AccountSetupApplicationServiceProxy/ fetch
9	Create User Role	POST	http://(IP):(Port)/RoleMasterProxy/ createMasterRole
10	Fetch User Role	POST	http://(IP):(Port)/RoleMasterProxy / fetchSingleRole

11	Create Structure	POST	http://(IP):(Port)/StructureMaintenanceApplicationServiceProxy/createStructure
12	Fetch Structure	POST	http://(IP):(Port)/StructureMaintenanceApplicationServiceProxy/getMaxVersionStructuredetails
13	Execute Sweep	POST	http://(IP):(Port)/SweepApplicationServiceProxy/executeStructures
14	Fetch Sweep Details	POST	http://(IP):(Port)/BatchMonitorApplicationServiceProxy/fetch
15	View of Liquidity Structures Sweep,Pool,Hybrid, Enhancement Service	POST	http://(IP):(Port)/StructureMaintenanceApplicationServiceProxy/getStructureDetailsOfLatestVersion
16	Marking of a structure to Hold status	POST	http://(IP):(Port)/StructureStatusServiceProxy/holdStructure
17	Create Country	POST	http://(IP):(Port)/CountrySetupApplicationServiceProxy/create
18	Fetch Country	POST	http://(IP):(Port)/CountrySetupApplicationServiceProxy/fetch
19	Update Country	POST	http://(IP):(Port)/CountrySetupApplicationServiceProxy/update
20	Delete Country	POST	http://(IP):(Port)/CountrySetupApplicationServiceProxy/delete
21	Create Branch	POST	http://(IP):(Port)/BranchSetupApplicationServiceProxy/create
22	Fetch Branch	POST	http://(IP):(Port)/BranchSetupApplicationServiceProxy/fetch
23	Update Branch	POST	http://(IP):(Port)/BranchSetupApplicationServiceProxy/update
24	Create Customer User Linkage	POST	http://(IP):(Port)/CustomerRMLinkageApplicationServiceProxy/createCustomerRMLinkage
25	Fetch Customer User Linkage	POST	http://(IP):(Port)/CustomerRMLinkageApplicationServiceProxy/fetchCustomerRMLinkage
26	Update Customer User Linkage	POST	http://(IP):(Port)/CustomerRMLinkageApplicationServiceProxy/updateCustomerRMLinkage
27	Delete Customer User Linkage	POST	http://(IP):(Port)/CustomerRMLinkageApplicationServiceProxy/deleteCustomerRMLinkage

28	Authorize Customer User Linkage	POST	http://(IP):(Port)/CustomerRMLinkageApplicationServiceProxy/AuthorizeCustomerRMLinkage
----	---------------------------------	------	----------------------------------------------------------------------------------------

### 3.1.1 Create Bank Services

The following are the expected functionality of the services.

- These services will create new bank that are present in Oracle Banking Liquidity Management.
- On failure case i.e., if the customer searched doesn't present in the system then it will throw an error stating appropriate error message.
- On success case, record containing the bank information will be inserted in DB successfully.

Example: Given below is a sample response of a bank. In case of services which return one bank, a list of such JSON objects would be returned.

---

```
{
"bankDesc": "string",
"bankName": "string",
"bankType": "string",
"uniqueid": "string",
"crossBorderPool": "string",
"crossBorderSweep": "string",
"domesticPool": "string",
"domesticSweep": "string",
"line1": "string",
"line2": "string",
"line3": "string",
"line4": "string",
"timeZone": "string",
"countryId": "string",
"allowInterBankSweep": "string",
"isbvtallowed": "string",
"sweep": "string",
"pool": "string",
"hybrid": "string",
"crossCcySweep": "string",
"crossCcyPool": "string",
"domesticHybSweep": "string",
"domesticHybPool": "string",
"cbHybSweep": "string",
"cbHybPool": "string",
"ccHybSweep": "string",
"ccHybPool": "string",
"sweep": "string",
```

```

"bankSetupKey":{
  "bankId":"string",
},
"params":{
  "parameter": "string",
  "value":"string",
},

"bankSetupAuditDTO" :{
"revNo": "Integer",
"makerId": "string",
"makerDateStamp": "string",
"checkerId": "string",
"chekerDateStamp": "string",
"onceAuth": "string",
"authStat": "string",
"recordStat": "string",
}
}

```

### 3.1.2 Fetch Bank Service :

. The following are the expected functionality of the service.

- This service will query for bank details of customer that are present in Oracle Banking Liquidity Management.
- The customers should be valid Individual customer that has record stat as Open and also authorized.
- On failure case i.e., if the customer searched doesn't present in the system then it will throw an error stating "Invalid Bank Id"

```

{
"bankDesc": "string",
"bankName": "string",
"bankType": "string",
"uniqueid": "string",
"crossBorderPool": "string",
"crossBorderSweep": "string",
"domesticPool": "string",
"domesticSweep": "string",
"line1": "string",
"line2": "string",
"line3": "string",
"line4": "string",
"timeZone": "string",

```



```

"countryId": "string",
"allowInterBankSweep": "string",
"isbvtallowed": "string",
"sweep": "string",
"pool": "string",
"hybrid": "string",
"crossCcySweep": "string",
"crossCcyPool": "string",
"domesticHybSweep": "string",
"domesticHybPool": "string",
"cbHybSweep": "string",
"cbHybPool": "string",
"ccHybSweep": "string",
"ccHybPool": "string",
"sweep": "string",
"bankSetupKey":{
  "bankId":"string",
},
"params":{
  "parameter": "string",
  "value":"string",
},
"bankSetupAuditDTO" :{
"revNo": "Integer",
"makerId": "string",
"makerDateStamp": "string",
"checkerId": "string",
"chekerDateStamp": "string",
"onceAuth": "string",
"authStat": "string",
"recordStat": "string",
}
}

```

### 3.1.3 Create User Service:

The following are the expected functionality of the services.

- These services will Create new User in Oracle Banking Liquidity Management.
- On failure case i.e., if the customer searched does not present in the system then it will throw an error stating appropriate error message.
- On success case, record containing the user information will be inserted in DB successfully.

```

{
"screenSaverTimeout": "BigDecimal",
"branchUsrpwd": "string",
"ldapUser": "string",
"pwdChangedOn": "Date",
"userId": "string",
"userEmail": "string",
"userPassword": "string",
"salt": "string",
"forcePasswdChange": "BigDecimal",
"userName": "string",
"homeEntity": "string",
"mfaEnbld": "string",
"mfald": "string",
"userStatus": "string",
"statusChangedOn": "Date",
"referenceNo": "string",
"keyId": "string",
"keyValue": "string",
"userCreationAuditDTO" :{
"revNo": "Integer",
"makerId": "string",
"makerDateStamp": "string",
"checkerId": "string",
"chekerDateStamp": "string",
"onceAuth": "string",
"authStat": "string",
"recordStat": "string",
}
}

```

### 3.1.4 Fetch User Service:

The following are the expected functionality of the services.

- These services will Fetch User in Oracle Banking Liquidity Management.
- On failure case i.e., if the customer searched doesn't present in the system then it will throw an error stating appropriate error message.
- On success case, record containing the user information will be fetched from DB successfully.

```

{
"screenSaverTimeout": "BigDecimal",
"branchUsrpwd": "string",
"ldapUser": "string",

```

```

"pwdChangedOn": "Date",
"userId": "string",
"userEmail": "string",
"userPassword": "string",
"salt": "string",
"forcePasswdChange": "BigDecimal",
"userName": "string",
"homeEntity": "string",
"mfaEnblId": "string",
"mfald": "string",
"userStatus": "string",
"statusChangedOn": "Date",
"referenceNo": "string",
"keyId": "string",
"keyValue": "string",
"userCreationAuditDTO" :{
"revNo": "Integer",
"makerId": "string",
"makerDateStamp": "string",
"checkerId": "string",
"chekerDateStamp": "string",
"onceAuth": "string",
"authStat": "string",
"recordStat": "string",
}
}

```

### 3.1.5 Create Customer Service:

The following are the expected functionality of the services.

- These services will Create new Customer in Oracle Banking Liquidity Management.
- On failure case i.e., if the customer searched doesn't present in the system then it will throw an error stating appropriate error message.
- On success case, record containing the user information will be inserted in DB successfully.

```

{
"customerName": "string",
"audit" :{
"revNo": "Integer",
"makerId": "string",
"makerDateStamp": "string",
"checkerId": "string",
"chekerDateStamp": "string",
"onceAuth": "string",

```

```

"authStat": "string",
"recordStat": "string",
}
"customerID": "string",
"desc": "string",
"bankID": "string",
"branchID": "string",
"parentCustID": "string ",
"address": "string",
"externalref": "string",
}

```

### 3.1.6 Fetch Customer Service:

The following are the expected functionality of the services.

- These services will Fetch Customer in Oracle Banking Liquidity Management.
- On failure case i.e., if the customer searched doesn't present in the system then it will throw an error stating appropriate error message.
- On success case, record containing the user information will be fetched from DB successfully.

```

{
"customerName": "string",
"audit" :{
"revNo": "Integer",
"makerId": "string",
"makerDateStamp": "string",
"checkerId": "string",
"chekerDateStamp": "string",
"onceAuth": "string",
"authStat": "string",
"recordStat": "string",
}
}

```

```

"customerID": "string",
"desc": "string",
"bankID": "string",
"branchID": "string",
"parentCustID": "string",
"address": "string",

```

```
"externalref": "string",
}
```

### 3.1.7 Create Account Service:

The following are the expected functionality of the services.

- These services will Create new Account in Oracle Banking Liquidity Management.
- On failure case i.e., if the customer searched doesn't present in the system then it will throw an error stating appropriate error message.
- On success case, record containing the user information will be inserted in DB successfully.

```
{
"customerName": "string",
"customerID": "string",
"accNo": "string",
"accDesc": "string",
"acctype": "string",
"extAccNo": "string",
"isExtAcc": "string",
"bicCode": "string",
"bankID": "string",
"bankDesc": "string",
"branchID": "string",
"branchDesc": "string",
"location": "string",
"curr": "string",
"balance": "double",
"balUpdateOn": "string",
"accResidentType": "string",
"accStatus": "string",
"notionalAccFlag": "string",
"debitThreshold": "double",
"allowUnlimitedDebit": "string",
"availableBal": "double",
"IBAN": "string",
"accCategory": "string",
"audit" :{
"revNo": "Integer",
"makerId": "string",
"makerDateStamp": "string",
"checkerId": "string",
"chekerDateStamp": "string",
"onceAuth": "string",
```

```
"authStat": "string",
"recordStat": "string",
}
}
```

### 3.1.8 Fetch Account Service:

The following are the expected functionality of the services.

- These services will Fetch Account in Oracle Banking Liquidity Management.
- On failure case i.e., if the customer searched doesn't present in the system then it will throw an error stating appropriate error message.
- On success case, record containing the user information will be fetched from DB successfully.

```
{
"customerName": "string",
"customerID": "string",
"accNo": "string",
"accDesc": "string",
"acctype": "string",
"extAccNo": "string",
"isExtAcc": "string",
"bicCode": "string",
"bankID": "string",
"bankDesc": "string",
"branchID": "string",
"branchDesc": "string",
"location": "string",
"curr": "string",
"balance": "double",
"balUpdateOn": "string",
"accResidentType": "string",
"accStatus": "string",
"notionalAccFlag": "string",
"debitThreshold": "double",
"allowUnlimitedDebit": "string",
"availableBal": "double",
"IBAN": "string",
"accCategory": "string",
"audit" :{
"revNo": "Integer",
"makerId": "string",
"makerDateStamp": "string",
"checkerId": "string",
```

```
"chekerDateStamp": "string",
"onceAuth": "string",
"authStat": "string",
"recordStat": "string",
}
}
```

### 3.1.9 Create User Role Service:

The following are the expected functionality of the services.

- These services will Create new user role in Oracle Banking Liquidity Management.
- On failure case i.e., if the customer searched doesn't present in the system then it will throw an error stating appropriate error message.
- On success case, record containing the user information will be inserted in DB successfully.

```
{
"roleId": "string",
"roleDescription": "string",
"roleMasterAuditDTO" :{
"revNo": "Integer",
"makerId": "string",
"makerDateStamp": "string",
"checkerId": "string",
"chekerDateStamp": "string",
"onceAuth": "string",
"authStat": "string",
"recordStat": "string",
}
}
```

### 3.1.10 Fetch User Role Service:

The following are the expected functionality of the services.

- These services will Fetch user role\_in Oracle Banking Liquidity Management.
- On failure case i.e., if the customer searched doesn't present in the system then it will throw an error stating appropriate error message.
- On success case, record containing the user information will be fetched from DB successfully.

```

{
"roleId": "string",
"roleDescription": "string",
"roleMasterAuditDTO" :{
"revNo": "Integer",
"makerId": "string",
"makerDateStamp": "string",
"checkerId": "string",
"chekerDateStamp": "string",
"onceAuth": "string",
"authStat": "string",
"recordStat": "string",
}
}
}

```

### 3.1.11 Create Structure Service:

The following are the expected functionality of the services.

- These services will Create new structure in Oracle Banking Liquidity Management.
- On failure case i.e., if the customer searched doesn't present in the system then it will throw an error stating appropriate error message.
- On success case, record containing the user information will be inserted in DB successfully.

```

{
"structureKey":{
"structureId":"string",
"versionNo":1
},

"customerId":"string",
"desc":"string",
"priority":integer,
"effDate":"date",
"endDate":"date",
"multibankChk":boolean,
"crossBorderChk":boolean,
"crossCcyChk":boolean,
"accountlst":[
{
"accountDetails":{
"accountKey":{
"accountNo":"string",
"branchCodeId":"string",
"branchCode":"string",

```



```

        "ccylId":"string",
        "ccy":"string"
    }
},
"parentAccountKey":{
},
"instructionPriority":integer,
"thirdPartAccountChk":boolean,
"baseAmt":double,
"bookingAccount":{
    "accountNo":"string",
    "branchCodeId":"string",
    "branchCode":"string",
    "ccylId":"string",
    "ccy":"string"
},
"cashCCMethod":"string",
"paymentInstructionList":{
    "paymentAcclInstructList":[

    ]
},
"instructionList":{
    "instructiondetailList":[

    ]
},

"allowRevSweep":boolean,
"hold":boolean,
"eodexecution":boolean,
"percentageShare":0
},
{
"accountDetails":{
    "accountKey":{
        "accountNo":"string",
        "branchCodeId":"string",
        "branchCode":"string",
        "ccylId":"string",
        "ccy":"string"
    }
},
"parentAccountKey":{
    "accountNo":"string",
    "branchCodeId":"string",
    "branchCode":"string",
    "ccylId":"string",
    "ccy":"string"
}
}

```

```

},
"instructionPriority":1,
"thirdPartAccountChk":boolean,
"baseAmt":double,
"bookingAccount":{
  "accountNo":"string",
  "branchCodeId":"string",
  "branchCode":"string",
  "ccylId":"string",
  "ccy":"string"
},
"rateType":"string",
"cashCCMethod":"string",
"paymentInstructionList":{
  "paymentAcclInstructList":[
    {
      "strKey":{
        "structureId":"string",
        "versionNo":1
      },
      "fromAccKey":{
        "accountNo":"string",
        "branchCodeId":"string",
        "branchCode":"string",
        "ccylId":"string",
        "ccy":"string"
      }
    },
    "beneficiaryAccKey":{
      "accountNo":"string",
      "branchCodeId":"string",
      "branchCode":"string",
      "ccylId":"string",
      "ccy":"string"
    }
  ],
  "swpDirection":"string",
  "networkId":integer,
  "paymentInstructionParameter":[
    {
      "instructionParamKey":{
        "paramName":"string"
      },
      "msgType":"string",
      "serialNum":integer
    },
    {
      "instructionParamKey":{
        "paramName":"string"
      },
      "msgType":"string",

```

```

        "serialNum":integer
    }
]
}
]
},
"instructionList":{
    "instructiondetailList":[
        {
            "instructionDetailKey":{
                "instructionDetailId":"string"
            },
            "instructionParamLst":[
                {
                    "instructionParamKey":{
                        "paramName":"string"
                    },
                    "paramvalue":"string"
                }
            ],
            "frequencyList":{
                "frequencyList":[
                    {
                        "frequencyKeyDTO":{
                            "frequencyId":"string"
                        }
                    }
                ]
            },
            "instructionInstructPriority":integer,
            "isCallFromSweep":boolean
        },
        {
            "instructionDetailKey":{
                "instructionDetailId":"string"
            },
            "instructionParamLst":[
                {
                    "instructionParamKey":{
                        "paramName":"string"
                    },
                    "paramvalue":"string"
                }
            ],
            "frequencyList":{
                "frequencyList":[
                    {
                        "frequencyKeyDTO":{
                            "frequencyId":"string"
                        }
                    }
                ]
            }
        }
    ]
}

```

```

    }
  }
]
},
"instructionInstructPriority":integer,
"isCallFromSweep":boolean
},
{
  "instructionDetailKey":{
    "instructionDetailId":"string"
  },
  "instructionParamLst":[
    {
      "instructionParamKey":{
        "paramName":"string"
      },
      "paramvalue":"string"
    }
  ],
  "frequencyList":{
    "frequencyList":[
      {
        "frequencyKeyDTO":{
          "frequencyId":"string"
        }
      }
    ]
  },
  "instructionInstructPriority":integer,
  "isCallFromSweep":boolean
},
{
  "instructionDetailKey":{
    "instructionDetailId":"string"
  },
  "instructionParamLst":[
    {
      "instructionParamKey":{
        "paramName":"string"
      },
      "paramvalue":"string"
    }
  ],
  "frequencyList":{
    "frequencyList":[
      {
        "frequencyKeyDTO":{
          "frequencyId":"string"
        }
      }
    ]
  }
}

```

```

    }
  ]
},
"instructionInstructPriority":integer,
"isCallFromSweep":boolean
},
{
  "instructionDetailKey":{
    "instructionDetailId":"string"
  },
  "instructionParamLst":[
    {
      "instructionParamKey":{
        "paramName":"string"
      },
      "paramvalue":"string"
    }
  ],
  "frequencyList":{
    "frequencyList":[
      "frequencyKeyDTO":{
        "frequencyId":"string"
      }
    ]
  },
  "instructionInstructPriority":integer,
  "isCallFromSweep":boolean
}
]
},
"sweepDirection":"string",
"allowRevSweep":boolean,
"revFreq":"string",
"hold":boolean,
"eodexecution":boolean,
"percentageShare":integer
},
{
  "accountDetails":{
    "accountKey":{
      "accountNo":"string",
      "branchCodeId":"string",
      "branchCode":"string",
      "ccyId":"string",
      "ccy":"string"
    }
  },
  "parentAccountKey":{
    "accountNo":"string",

```

```

    "branchCodeId":"string",
    "branchCode":"string",
    "ccylId":"string",
    "ccy":"string"
  },
  "instructionPriority":integer,
  "thirdPartAccountChk":boolean,
  "baseAmt":double,
  "bookingAccount":{
    "accountNo":"string",
    "branchCodeId":"string",
    "branchCode":"string",
    "ccylId":"string",
    "ccy":"string"
  },
  "rateType":"string",
  "cashCCMethod":"string",
  "paymentInstructionList":{
    "paymentAcclInstructList":[
      {
        "strKey":{
          "structureId":"string",
          "versionNo":integer
        },
        "fromAccKey":{
          "accountNo":"string",
          "branchCodeId":"string",
          "branchCode":"string",
          "ccylId":"string",
          "ccy":"string"
        },
        "beneficiaryAccKey":{
          "accountNo":"string",
          "branchCodeId":"string",
          "branchCode":"string",
          "ccylId":"string",
          "ccy":"string"
        },
        "swpDirection":"string",
        "networkId":integer,
        "paymentInstructionParameter":[
          {
            "instructionParamKey":{
              "paramName":"string"
            },
            "msgType":"string",
            "serialNum":integer
          },
          {
            "instructionParamKey":{

```

```

        "paramName":"string"
    },
    "msgType":"string",
    "serialNum":integer
}
]
}
],
"instructionList":{
"instructiondetailList":[
{
"instructionDetailKey":{
"instructionDetailId":"string"
},
}
],
"instructionParamLst":[
{
"instructionParamKey":{
"paramName":"string"
},
"paramvalue":"string"
}
],
"frequencyList":{
"frequencyList":[
{
"frequencyKeyDTO":{
"frequencyId":"string"
}
}
]
},
"instructionInstructPriority":integer,
"isCallFromSweep":boolean
},
{
"instructionDetailKey":{
"instructionDetailId":"string"
},
"instructionParamLst":[
{
"instructionParamKey":{
"paramName":"string"
},
"paramvalue":"string"
}
]
},
]

```

```

"frequencyList":{
  "frequencyList":[
    {
      "frequencyKeyDTO":{
        "frequencyId":"string"
      }
    }
  ]
},
"instructionInstructPriority":integer,
"isCallFromSweep":boolean
},
{
  "instructionDetailKey":{
    "instructionDetailId":"string"
  },
  "instructionParamLst":[
    {
      "instructionParamKey":{
        "paramName":"string"
      },
      "paramvalue":"string"
    }
  ],
  "frequencyList":{
    "frequencyList":[
      {
        "frequencyKeyDTO":{
          "frequencyId":"string"
        }
      }
    ]
  },
  "instructionInstructPriority":integer,
  "isCallFromSweep":boolean
},
{
  "instructionDetailKey":{
    "instructionDetailId":"string"
  },
  "instructionParamLst":[
    {
      "instructionParamKey":{
        "paramName":"string"
      },
      "paramvalue":"string"
    }
  ],
  "frequencyList":{

```



```

    "frequencyList":[
      {
        "frequencyKeyDTO":{
          "frequencyId":"string"
        }
      }
    ]
  },

  "instructionInstructPriority":integer,
  "isCallFromSweep":boolean
},
{
  "instructionDetailKey":{
    "instructionDetailId":"string"
  },
  "instructionParamLst":[
    {
      "instructionParamKey":{
        "paramName":"string"
      },
      "paramvalue":"string"
    }
  ],
  "frequencyList":{
    "frequencyList":[
      {
        "frequencyKeyDTO":{
          "frequencyId":"string"
        }
      }
    ]
  },
  "instructionInstructPriority":integer,
  "isCallFromSweep":boolean
}
]
},
"sweepDirection":"string",
"allowRevSweep":boolean,
"revFreq":"F10",
"hold":boolean,
"eodexecution":boolean,
"percentageShare":0
}
],
"balType":"VD",
"headerAccount":{

```

```

    "accountNo":"string",
    "branchCodeId":"string",
    "branchCode":"string",
    "ccylId":"string",
    "ccy":"string"
  },
  "allowSweepOnCcyHol":boolean,
  "interestMethod":"string",
  "eodexecution":boolean
"audit":{
  "revNo":1,
  "makerId":"string",
  "makerDateStamp":"string",
  "onceAuth":boolean,
  "authStat":"string",
  "recordStat":"string"
},

  "holidayTreatment":"string",
  "maxBackwardDays":string,
  "reallocationMethod":"string",
  "structureType":"string",
  "structureStatus":"string"
}

```

### 3.1.12 Fetch Structure Service:

The following are the expected functionality of the services.

- These services will Fetch Structure details\_in Oracle Banking Liquidity Management.
- On failure case i.e., if the customer searched does not present in the system then it will throw an error stating appropriate error message.
- On success case, record containing the user information will be fetched from DB successfully.

```

{"structureKey":
{"structureId":"string"
},
"customerId":"string",
"accountlst":[]
,"eodexecution":boolean
}

```

### 3.1.13 Sweep Execution Service:

The following are the expected functionality of the services.

- These services will execute sweep\_in Oracle Banking Liquidity Management.
- On failure case i.e., if the customer searched doesn't present in the system then it will throw an error stating appropriate error message.

```
{"strListKeyDTO":  
[  
  {"structureId":"string",  
    "versionNo":integer  
  },  
  {"overRidePending":boolean}]
```

### 3.1.14 Fetch Sweep Details Service:

The following are the expected functionality of the services.

- These services will Fetch Sweep details\_in Oracle Banking Liquidity Management.
- On failure case i.e., if the customer searched doesn't present in the system then it will throw an error stating appropriate error message.
- On success case, record containing the user information will be fetched from DB successfully.

```
{  
"fromDate":"string",  
"toDate":"string",  
"structureId":"string",  
"customerId":"string",  
"filter":"string"  
}
```

### 3.1.15 View of Liquidity Structures Sweep,Pool,Hybrid, Enhancement Service:

The following are the expected functionality of the services.

- These services will able to view liquidity structures of sweep, pool, hybrid in Oracle Banking Liquidity Management.

- On failure case i.e., if the customer searched doesn't present in the system then it will throw an error stating appropriate error message.
- On success case, record containing the user information will be fetched from DB successfully.

```
{
  "structureKey":
  {"structureId":"string"
  },
  "customerId":"string",
  "accountlst":[]
  ,"eodexecution":boolean
}
```

### **3.1.16 Marking of a structure to Hold status Service:**

The following are the expected functionality of the services.

- These services will able to hold status of structure in Oracle Banking Liquidity Management.
- On failure case i.e., if the customer searched doesn't present in the system then it will throw an error stating appropriate error message.

```
{
  "structurestatusdetkey":
  {
  "acc_structureId":"string",
  "versionNo":integer
  },
  "user":"string",
  "event":"string",
  "branchID":"string",
  "strStatus":"string",
  "channel":"string"
}
```

### **3.1.17 Create Country Service:**

The following are the expected functionality of the services.

- These services will able used to create country in Oracle Banking Liquidity Management.
- On failure case i.e., if the country already created in the system then it will throw an error stating appropriate error message.

```

{"crcSetupKey":
{
"countryCode":"string"
},
"countryName":"string",
"baseCurrency":"string",
"domesticSweep":"string",
"crossBorderSweep":"string",
"domesticNotionalPool":"string",

"crossBorderNotionalPool":"string",
"crossBorderAllowed":"string",
"hybridStructureAllowed":"string",
"allowedAccountType":"string"
}

```

### 3.1.18 Fetch Country Service:

The following are the expected functionality of the services.

- These services will be able to fetch country in Oracle Banking Liquidity Management.
- On failure case i.e., if the customer searched doesn't present in the system then it will throw an error stating appropriate error message.

```

{"crcSetupKey":
{
"countryCode":"string"
},
"countryName":"string",
"baseCurrency":"string",
"domesticSweep":"string",
"crossBorderSweep":"string",
"domesticNotionalPool":"string",
"crossBorderNotionalPool":"string",
"crossBorderAllowed":"string",
"hybridStructureAllowed":"string",
"allowedAccountType":"string"
}

```

### **3.1.19 Update Country Service:**

The following are the expected functionality of the services.

- These services will be able to update country details in Oracle Banking Liquidity Management.
- On failure case i.e., if the customer searched country doesn't present in the system then it will throw an error stating appropriate error message.

```
{"crcSetupKey":
{
"countryCode":"string"
},
"countryName":"string",
"baseCurrency":"string",
"domesticSweep":"string",
"crossBorderSweep":"string",
"domesticNotionalPool":"string",
"crossBorderNotionalPool":"string",
"crossBorderAllowed":"string",
"hybridStructureAllowed":"string",
"allowedAccountType":"string"
}
```

### **3.1.20 Delete Country Service:**

The following are the expected functionality of the services.

- These services will be able to delete country in Oracle Banking Liquidity Management.
- On failure case i.e., if the customer searched doesn't present in the system then it will throw an error stating appropriate error message.

```
{"crcSetupKey":
{
"countryCode":"string"
},
"countryName":"string",
"baseCurrency":"string",
"domesticSweep":"string",
"crossBorderSweep":"string",
"domesticNotionalPool":"string",
"crossBorderNotionalPool":"string",
"crossBorderAllowed":"string",
"hybridStructureAllowed":"string",
```

```
"allowedAccountType":"string"
}
```

### 3.1.21 Create Branch Service:

The following are the expected functionality of the services.

- These services will be able to create branch in Oracle Banking Liquidity Management.
- On failure case i.e., if the branch already created in the system then it will throw an error stating appropriate error message.

```
{"branchSetupKey":
{
"branchId":"string"
},
"bankId":"string",
"branchName":"string",
"timeZn":"string",
"domesticSweepAllowed":"string",
"domesticPoolAllowed":"string",
"localCurrency ":"string",
"crossBorderSweepAllowed ":"string",
"crossBorderPoolAllowed ":"string",
"line1 ":"string",
"line2":"string",
"line3":"string",
"line4":"string",
"country":"string",
"zip":"string",
"city":"string",
"region":"string",
}
```

### 3.1.22 Fetch Branch Service:

The following are the expected functionality of the services.

- These services will be able to fetch branch in Oracle Banking Liquidity Management.
- On failure case i.e., if the customer searched doesn't present in the system then it will throw an error stating appropriate error message.

```
{"branchSetupKey":
{
"branchId":"string"
},
"bankId":"string",
```

```

"branchName":"string",
"timeZn":"string",
"domesticSweepAllowed":"string",
"domesticPoolAllowed":"string",
"localCurrency ":"string",
"crossBorderSweepAllowed ":"string",
"crossBorderPoolAllowed ":"string",
"line1 ":"string",
"line2":"string",
"line3":"string",
"line4":"string",
"country":"string",
"zip":"string",
"city":"string",
"region":"string",
}

```

### 3.1.23 Update Branch Service:

The following are the expected functionality of the services.

- These services will be able to update branch details in Oracle Banking Liquidity Management.
- On failure case i.e., if the customer searched branch doesn't present in the system then it will throw an error stating appropriate error message.

```

{"branchSetupKey":
{
"branchId":"string"
},
"bankId":"string",
"branchName":"string",
"timeZn":"string",
"domesticSweepAllowed":"string",
"domesticPoolAllowed":"string",
"localCurrency ":"string",
"crossBorderSweepAllowed ":"string",
"crossBorderPoolAllowed ":"string",
"line1 ":"string",
"line2":"string",
"line3":"string",
"line4":"string",
"country":"string",
"zip":"string",
"city":"string",
"region":"string",
}

```



### 3.1.24 Create Customer User Linkage Service:

The following are the expected functionality of the services.

- These services will be able to create customer user linkage in Oracle Banking Liquidity Management.
- On failure case i.e., if the customer-searched branch does not present in the system then it will throw an error stating appropriate error message.

```
{"customerRMLinkageDTOList":  
[  
  {  
    "seqNo ":"string",  
    "customerId ":"string",  
    "customerName ":"string",  
    "userId ":"string",  
    "userName ":"string",  
    "tabname ":"string",  
    "audit ":"  
    {  
      "revNo":,  
      "makerId":"string",  
      "makerDateStamp":"string",  
      "checkerId":"string",  
      "chekerDateStamp":"string",  
      "onceAuth":"string",  
      "authStat":"string",  
      "recordStat":"string"  
    }  
  }  
]  
}
```

### 3.1.25 Fetch Customer User Linkage Service:

The following are the expected functionality of the services.

- These services will be able to fetch customer user linkage in Oracle Banking Liquidity Management.
- On failure case i.e., if the customer-searched branch does not present in the system then it will throw an error stating appropriate error message.

```
{"customerRMLinkageDTOList":  
[  
  {  
    "seqNo ":"string",  
    "customerId ":"string",  
    "customerName ":"string",
```

```

"userId ":"string",
"userName ":"string",
"tabname ":"string",
"audit ":
{
"revNo":,
"makerId":"string",
"makerDateStamp":"string",
"checkerId":"string",
"chekerDateStamp":"string",
"onceAuth":"string",
"authStat":"string",
"recordStat":"string"
}
}
]
}

```

### **3.1.26 Update Customer User Linkage Service:**

The following are the expected functionality of the services.

- These services will be able to update customer user linkage in Oracle Banking Liquidity Management.
- On failure case i.e., if the customer-searched branch does not present in the system then it will throw an error stating appropriate error message.

```

{"customerRMLinkageDTOList":
[
{
"seqNo ":"string",
"customerId ":"string",
"customerName ":"string",
"userId ":"string",
"userName ":"string",
"tabname ":"string",
"audit ":
{
"revNo":,
"makerId":"string",
"makerDateStamp":"string",
"checkerId":"string",
"chekerDateStamp":"string",
"onceAuth":"string",
"authStat":"string",
"recordStat":"string"
}
}
]
}

```

### 3.1.27 Delete Customer User Linkage Service:

The following are the expected functionality of the services.

- These services will be able to delete customer user linkage in Oracle Banking Liquidity Management.
- On failure case i.e., if the customer-searched branch does not present in the system then it will throw an error stating appropriate error message.

```
{"customerRMLinkageDTOList":  
[  
{  
  "seqNo ":"string",  
  "customerId ":"string",  
  "customerName ":"string",  
  "userId ":"string",  
  "userName ":"string",  
  "tabname ":"string",  
  "audit " :  
  
  {  
    "revNo":,  
    "makerId":"string",  
  
    "makerDateStamp":"string",  
    "checkerId":"string",  
    "chekerDateStamp":"string",  
    "onceAuth":"string",  
    "authStat":"string",  
    "recordStat":"string"  
  }  
}  
]  
}
```

### 3.1.28 Authorize Customer User Linkage Service:

The following are the expected functionality of the services.

- These services will be able to authorize customer user linkage in Oracle Banking Liquidity Management.
- On failure case i.e., if the customer-searched branch does not present in the system then it will throw an error stating appropriate error message.

```
{"customerRMLinkageDTOList":
```

```
[
{
"seqNo ":"string",
"customerId ":"string",
"customerName ":"string",
"userId ":"string",
"userName ":"string",
"tablename ":"string",
"audit ":
{
"revNo":,
"makerId":"string",
"makerDateStamp":"string",
"checkerId":"string",
"chekerDateStamp":"string",
"onceAuth":"string",
"authStat":"string",
"recordStat":"string"
}
}
]
}
```



Liquidity Management Rest Service Guide  
Oracle Banking Liquidity Management  
Version 14.0.0.0.0  
[November] [2017]

Oracle Financial Services Software Limited  
Oracle Park  
Off Western Express Highway  
Goregaon (East)  
Mumbai, Maharashtra 400 063  
India

Worldwide Inquiries:  
Phone: +91 22 6718 3000  
Fax: +91 22 6718 3001  
[www.oracle.com/financialservices/](http://www.oracle.com/financialservices/)

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

**U.S. GOVERNMENT END USERS:** Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.