**Yugandhar Open Master Data Management (MDM) Hub**

**API and Transaction Reference Guide**

Yugandhar Open MDM Hub Release – V2.0.0

Date – 11/06/2018

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Copyright [2017] [Yugandhar Open MDM Hub]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

Contents

[About Yugandhar Open MDM Hub Project 10](#_Toc517042229)

[About this document 11](#_Toc517042230)

[Invoking Transactions on Yugandhar MDM Hub 12](#_Toc517042231)

[ REST Webservice 12](#_Toc517042232)

[ MQ interface 12](#_Toc517042233)

[The JSON Message layout 13](#_Toc517042234)

[Common structure for invoking all transactions 13](#_Toc517042235)

[Request Message Layout 13](#_Toc517042236)

[ Header Object 13](#_Toc517042237)

[ TxnPayload Object 14](#_Toc517042238)

[ Primary Key Auto generation Vs providing in the request 14](#_Toc517042239)

[Response Message Layout 15](#_Toc517042240)

[ Header Object 15](#_Toc517042241)

[ TxnPayload Object 15](#_Toc517042242)

[ Error Response 15](#_Toc517042243)

[Sample Request and Response Message 16](#_Toc517042244)

[Master repository of sample messages 19](#_Toc517042245)

[Out of the Box Transactions 19](#_Toc517042246)

[Base entity services 19](#_Toc517042247)

[Composite Services 19](#_Toc517042248)

[Application Configuration Management services 20](#_Toc517042249)

[1. createConfigAppPropertiesBase 20](#_Toc517042250)

[2. createConfigErrorcodeRegistryBase 20](#_Toc517042251)

[3. createConfigInquiryLevelsBase 20](#_Toc517042252)

[4. createConfigLanguageCodeBase 21](#_Toc517042253)

[5. createConfigTxnRegistryBase 21](#_Toc517042254)

[6. findAllConfigErrorcodeRegistryByLanguageCodeBase 21](#_Toc517042255)

[7. findAllConfigInquiryLevelsBase 22](#_Toc517042256)

[8. findAllConfigInquiryLevelsByBusinessKeyBase 22](#_Toc517042257)

[9. findAllConfigLanguageCodesBase 22](#_Toc517042258)

[10. findConfigAppPropertiesByBusinessKeyBase 23](#_Toc517042259)

[11. findConfigErrorcodeRegistryByBusinessKeyBase 23](#_Toc517042260)

[12. findConfigLanguageCodeByBusinessKeyBase 23](#_Toc517042261)

[13. findConfigTxnRegistryByBusinessKeyBase 24](#_Toc517042262)

[14. retrieveConfigAppPropertiesBase 24](#_Toc517042263)

[15. retrieveConfigErrorcodeRegistryBase 24](#_Toc517042264)

[16. retrieveConfigInquiryLevelsBase 25](#_Toc517042265)

[17. retrieveConfigLanguageCodeBase 25](#_Toc517042266)

[18. retrieveConfigTxnRegistryBase 25](#_Toc517042267)

[19. updateConfigAppPropertiesBase 25](#_Toc517042268)

[20. updateConfigErrorcodeRegistryBase 26](#_Toc517042269)

[21. updateConfigInquiryLevelsBase 26](#_Toc517042270)

[22. updateConfigLanguageCodeBase 26](#_Toc517042271)

[23. updateConfigTxnRegistryBase 27](#_Toc517042272)

[Authorization Framework services 28](#_Toc517042273)

[1. createAuthRolesRegistryBase 28](#_Toc517042274)

[2. updateAuthRolesRegistryBase 28](#_Toc517042275)

[3. retrieveAuthRolesRegistryBase 28](#_Toc517042276)

[4. findAuthRolesRegistryByBusinessKeyBase 29](#_Toc517042277)

[5. createAuthUserRegistryBase 29](#_Toc517042278)

[6. updateAuthUserRegistryBase 29](#_Toc517042279)

[7. retrieveAuthUserRegistryBase 29](#_Toc517042280)

[8. findAuthUserRegistryByBusinessKeyBase 30](#_Toc517042281)

[9. createAuthUserroleAccesscontrolBase 30](#_Toc517042282)

[10. updateAuthUserroleAccesscontrolBase 31](#_Toc517042283)

[11. retrieveAuthUserroleAccesscontrolBase 31](#_Toc517042284)

[12. findAuthUserroleAccesscontrolByBusinessKeyBase 31](#_Toc517042285)

[13. findAuthUserroleAccesscontrolByRegistryIdpkBase 32](#_Toc517042286)

[14. createAuthUserRoleAssocBase 32](#_Toc517042287)

[15. updateAuthUserRoleAssocBase 32](#_Toc517042288)

[16. retrieveAuthUserRoleAssocBase 32](#_Toc517042289)

[17. findAuthUserRoleAssocByBusinessKeyBase 33](#_Toc517042290)

[18. findAllRecordsByAuthUserRegistryIdpkBase 33](#_Toc517042291)

[19. findAllRecordsByAuthRolesRegistryIdpkBase 33](#_Toc517042292)

[Batch Framework Management services 35](#_Toc517042293)

[1. createBatchEntityToProcessBase 35](#_Toc517042294)

[2. updateBatchEntityToProcessBase 35](#_Toc517042295)

[3. retrieveBatchEntityToProcessBase 35](#_Toc517042296)

[Match And Merge Framework Management services 36](#_Toc517042297)

[1. createInactiveLeRegistryBase 36](#_Toc517042298)

[2. createMatchCandidateLeRegistryBase 36](#_Toc517042299)

[3. createMatchMergedLeAssocBase 36](#_Toc517042300)

[1. createRefMatchScoreBase 37](#_Toc517042301)

[4. createRefMatchThresholdBase 37](#_Toc517042302)

[5. findAllRefMatchScoreByMatchEntityObjectName 37](#_Toc517042303)

[6. findAllRefMatchThresholdBase 38](#_Toc517042304)

[7. findRefMatchScoreByBusinessKeyBase 38](#_Toc517042305)

[8. findRefMatchThresholdByBusinessKeyBase 38](#_Toc517042306)

[9. retrieveInactiveLeRegistryBase 39](#_Toc517042307)

[10. retrieveMatchCandidateLeRegistryBase 39](#_Toc517042308)

[11. retrieveMatchMergedLeAssocBase 39](#_Toc517042309)

[12. retrieveRefMatchScoreBase 40](#_Toc517042310)

[13. retrieveRefMatchThresholdBase 40](#_Toc517042311)

[14. updateInactiveLeRegistryBase 40](#_Toc517042312)

[15. updateMatchCandidateLeRegistryBase 41](#_Toc517042313)

[16. updateMatchMergedLeAssocBase 41](#_Toc517042314)

[17. updateRefMatchScoreBase 41](#_Toc517042315)

[18. updateRefMatchThresholdBase 42](#_Toc517042316)

[Data Table Management Services (Base) 43](#_Toc517042317)

[1. createAccountAddressAssocBase 43](#_Toc517042318)

[2. createAccountBase 43](#_Toc517042319)

[3. createAccountPhoneAssocBase 43](#_Toc517042320)

[4. createAddressBase 43](#_Toc517042321)

[5. createCorporationnamesBase 44](#_Toc517042322)

[6. createEntityGroupAssocBase 44](#_Toc517042323)

[7. createEntityGroupBase 44](#_Toc517042324)

[8. createLeAccountAssocBase 45](#_Toc517042325)

[9. createLeAddressAssocBase 45](#_Toc517042326)

[10. createLeCorporationBase 45](#_Toc517042327)

[11. createLeIdentifierKycRegistryBase 46](#_Toc517042328)

[12. createLePersonBase 46](#_Toc517042329)

[13. createLePhoneAssocBase 46](#_Toc517042330)

[14. createLePreferencesBase 46](#_Toc517042331)

[15. createLePropertyAssocBase 47](#_Toc517042332)

[16. createLeSystemKeysRegistryBase 47](#_Toc517042333)

[17. createLeToLeRelationshipBase 47](#_Toc517042334)

[18. createLeVehicleAssocBase 48](#_Toc517042335)

[19. createLegalentityBase 48](#_Toc517042336)

[20. createMiscellaneousInfoBase 48](#_Toc517042337)

[21. createPersonnamesBase 48](#_Toc517042338)

[22. createPhoneStandardizedBase 49](#_Toc517042339)

[23. createPropertyBase 49](#_Toc517042340)

[24. createVehicleBase 49](#_Toc517042341)

[25. findAllAccountAddressAssocByAccountIdPkBase 50](#_Toc517042342)

[26. findAllAccountPhoneAssocByAccountIdPkBase 50](#_Toc517042343)

[27. findAllMiscellaneousInfoByEntityIdPkBase 50](#_Toc517042344)

[28. findCorporationnamesByLegalEntityIdBase 51](#_Toc517042345)

[29. findEntityGroupAssocByEntityGroupIdpkBase 51](#_Toc517042346)

[30. findEntityGroupAssocByEntityIdPkBase 51](#_Toc517042347)

[31. findLeAccountAssocByAccountIdPkBase 52](#_Toc517042348)

[32. findLeAccountAssocByLegalEntityIdPkBase 52](#_Toc517042349)

[33. findLeAddressByLegalEntityIdPkBase 52](#_Toc517042350)

[34. findLeIdentifierKycRegistryByLegalEntityIdPkBase 53](#_Toc517042351)

[35. findLePhoneAssocByLegalEntityIdPkBase 53](#_Toc517042352)

[36. findLePreferencesByLegalEntityIdPkBase 54](#_Toc517042353)

[37. findLePropertyAssocByLegalEntityIdPkBase 54](#_Toc517042354)

[38. findLeSystemKeysRegistryByLegalEntityIdPkBase 54](#_Toc517042355)

[39. findLeToLeRelationshipByLegalentityIdpkBase 55](#_Toc517042356)

[40. findLeVehicleAssocByLegalEntityIdPkBase 55](#_Toc517042357)

[41. findPersonnamesByLegalEntityIdBase 56](#_Toc517042358)

[42. retrieveAccountAddressAssocBase 56](#_Toc517042359)

[43. retrieveAccountBase 56](#_Toc517042360)

[44. retrieveAccountPhoneAssocBase 57](#_Toc517042361)

[45. retrieveAddressBase 57](#_Toc517042362)

[46. retrieveCorporationnamesBase 57](#_Toc517042363)

[47. retrieveEntityGroupAssocBase 57](#_Toc517042364)

[48. retrieveEntityGroupBase 58](#_Toc517042365)

[49. retrieveLeAccountAssocBase 58](#_Toc517042366)

[50. retrieveLeAddressAssocBase 58](#_Toc517042367)

[51. retrieveLeCorporationBase 58](#_Toc517042368)

[52. retrieveLeIdentifierKycRegistryBase 59](#_Toc517042369)

[53. retrieveLePersonBase 59](#_Toc517042370)

[54. retrieveLePhoneAssocBase 59](#_Toc517042371)

[55. retrieveLePreferencesBase 59](#_Toc517042372)

[56. retrieveLePropertyAssocBase 60](#_Toc517042373)

[57. retrieveLeSystemKeysRegistryBase 60](#_Toc517042374)

[58. retrieveLeToLeRelationshipBase 60](#_Toc517042375)

[59. retrieveLeVehicleAssocBase 61](#_Toc517042376)

[60. retrieveLegalentityBase 61](#_Toc517042377)

[61. retrieveMiscellaneousInfoBase 61](#_Toc517042378)

[62. retrievePersonnamesBase 62](#_Toc517042379)

[63. retrievePhoneStandardizedBase 62](#_Toc517042380)

[64. retrievePropertyBase 62](#_Toc517042381)

[65. retrieveVehicleBase 62](#_Toc517042382)

[66. updateAccountAddressAssocBase 63](#_Toc517042383)

[67. updateAccountBase 63](#_Toc517042384)

[68. updateAccountPhoneAssocBase 63](#_Toc517042385)

[69. updateAddressBase 64](#_Toc517042386)

[70. updateCorporationnamesBase 64](#_Toc517042387)

[71. updateEntityGroupAssocBase 64](#_Toc517042388)

[72. updateEntityGroupBase 64](#_Toc517042389)

[73. updateLeAccountAssocBase 65](#_Toc517042390)

[74. updateLeAddressAssocBase 65](#_Toc517042391)

[75. updateLeCorporationBase 65](#_Toc517042392)

[76. updateLeIdentifierKycRegistryBase 65](#_Toc517042393)

[77. updateLePersonBase 66](#_Toc517042394)

[78. updateLePhoneAssocBase 66](#_Toc517042395)

[79. updateLePreferencesBase 66](#_Toc517042396)

[80. updateLePropertyAssocBase 66](#_Toc517042397)

[81. updateLeSystemKeysRegistryBase 67](#_Toc517042398)

[82. updateLeToLeRelationshipBase 67](#_Toc517042399)

[83. updateLeVehicleAssocBase 67](#_Toc517042400)

[84. updateLegalentityBase 68](#_Toc517042401)

[85. updateMiscellaneousInfoBase 68](#_Toc517042402)

[86. updatePersonnamesBase 68](#_Toc517042403)

[87. updatePhoneStandardizedBase 68](#_Toc517042404)

[88. updatePropertyBase 69](#_Toc517042405)

[89. updateVehicleBase 69](#_Toc517042406)

[Data Table Management Services (Composite) 70](#_Toc517042407)

[1. searchAuthRoles 70](#_Toc517042408)

[2. searchAuthUsers 71](#_Toc517042409)

[3. searchAuthAccessControl 72](#_Toc517042410)

[4. createAccount 73](#_Toc517042411)

[5. createAccountAddressAssociation 73](#_Toc517042412)

[6. createAccountPhoneAssociation 74](#_Toc517042413)

[7. createEntityGroup 74](#_Toc517042414)

[8. createLeAccountAssociation 74](#_Toc517042415)

[9. createLeAddressAssociation 75](#_Toc517042416)

[10. createLeCorporation 75](#_Toc517042417)

[11. createLePerson 76](#_Toc517042418)

[12. createLePhoneAssociation 76](#_Toc517042419)

[13. createLePropertyAssociation 77](#_Toc517042420)

[14. createLeVehicleAssociation 77](#_Toc517042421)

[15. createLegalEntity 78](#_Toc517042422)

[16. findAllAccountAddressAssocByAccountId 78](#_Toc517042423)

[17. findAllAccountPhoneAssocByAccountId 79](#_Toc517042424)

[18. findAllLeAccountAssocByLegalEntityId 79](#_Toc517042425)

[19. findAllLeAddressAssocByLegalEntityId 80](#_Toc517042426)

[20. findAllLePhoneAssocByLegalEntityId 80](#_Toc517042427)

[21. findAllLePropertyAssocByLegalEntityId 81](#_Toc517042428)

[22. findAllLeVehicleAssocByLegalEntityId 81](#_Toc517042429)

[23. performLeMatch 82](#_Toc517042430)

[24. retrieveAccountAddressByAccountAddressAssocId 82](#_Toc517042431)

[25. retrieveAccountByAccountId 83](#_Toc517042432)

[26. retrieveAccountPhoneByAccountPhoneAssocId 83](#_Toc517042433)

[27. retrieveEntityGroupByEntityGroupId 84](#_Toc517042434)

[28. retrieveLeAccountByLeAccountAssocId 84](#_Toc517042435)

[29. retrieveLeAddressByLeAddressAssocId 85](#_Toc517042436)

[30. retrieveLeCorporationByLegalEntityId 85](#_Toc517042437)

[31. retrieveLePersonByLegalEntityId 86](#_Toc517042438)

[32. retrieveLePhoneByLePhoneAssocId 87](#_Toc517042439)

[33. retrieveLePropertyByLePropertyAssocId 87](#_Toc517042440)

[34. retrieveLeVehicleLeVehicleAssocId 88](#_Toc517042441)

[35. retrieveLegalEntityByLegalEntityId 88](#_Toc517042442)

[36. searchAccountByAccountAttributes 89](#_Toc517042443)

[37. searchAccountByLEAttributes 90](#_Toc517042444)

[38. searchInactiveLeRegistry 90](#_Toc517042445)

[39. searchLegalEntityByAccountAttributes 91](#_Toc517042446)

[40. searchLegalEntityByLEAttributes 92](#_Toc517042447)

[41. searchMatchCandidateLE 92](#_Toc517042448)

[42. searchMatchCandidateLeRegistryBase 93](#_Toc517042449)

[43. searchMatchMergedLeAssocBase 94](#_Toc517042450)

[44. updateAccount 94](#_Toc517042451)

[45. updateAccountAddressAssociation 95](#_Toc517042452)

[46. updateAccountPhoneAssociation 96](#_Toc517042453)

[47. updateEntityGroup 96](#_Toc517042454)

[48. updateLeAccountAssociation 97](#_Toc517042455)

[49. updateLeAddressAssociation 97](#_Toc517042456)

[50. updateLeCorporation 98](#_Toc517042457)

[51. updateLePerson 98](#_Toc517042458)

[52. updateLePhoneAssociation 99](#_Toc517042459)

[53. updateLePropertyAssociation 99](#_Toc517042460)

[54. updateLeVehicleAssociation 100](#_Toc517042461)

[55. updateLegalEntity 101](#_Toc517042462)

[Reference data Table Management services 102](#_Toc517042463)

[Create Request - sample JSON message 102](#_Toc517042464)

[update request - sample JSON message 102](#_Toc517042465)

[Retrieve Request - sample JSON message 103](#_Toc517042466)

[findByBusinessKey request - sample JSON message. 104](#_Toc517042467)

[findAllRecorfdsByLanguageCode request - sample JSON message. 104](#_Toc517042468)

[List of Reference data Management services following common request structure 106](#_Toc517042469)

[Inquiry levels for Legal entity and Account – Default configuration 115](#_Toc517042470)

# About Yugandhar Open MDM Hub Project

Master Data Management came a long way in last decade or so. There are currently more than 20 MDM solutions catering to various specializations of MDM like Customer Data Integration (CDI), Product Information Management (PIM), vendor and supplier management etc. However most of these solutions come with licensing costs amounting to thousands of dollar. To offer a completely free solution which would be made available through Apache 2.0 license, A Project is started in 2017 under the name ‘Yugandhar Open MDM Project’ to build Open Source MDM solutions catering to CDI, PIM and Data Governance Capabilities. Yugandhar in Sanskrit means Ever Lasting and the strongest of its time. Our vision is to build the strongest, Open Source, Multi Domain, Cross Industry and completely free MDM Solution.

We are happy to announce that the first release of the Yugandhar MDM Hub catering to CDI solution is built with Open source technologies like Spring and Hibernate etc, inbuilt data Model, 400+ ready to use services and having incredible Out of the Box capabilities is currently being distributed. We aim to make the current CDI offering the strongest and Planning to bring Data Stewardship and PIM solutions in upcoming years.

# About this document

This document covers the Out of the box (OOTB) API and services provided by MDM Hub which can be invoked from out side the MDM Hub or can be utilized to build a new composite service having custom logic.

# Invoking Transactions on Yugandhar MDM Hub

## REST Webservice

Testing the Yugandhar Open MDM Hub REST webservices is covered in Development Environment Setup guide section ‘TEST With SOAPUI’.

## MQ interface

MQ messages can be invoked using MQ client. You may build your own java class to send message to request queue by taking reference from YugJMSMessageSender class or use The response will be sent to default response queue. Also take help from activemq site <http://activemq.apache.org/how-to-unit-test-jms-code.html>.

We will be using SOAPUI based testing approach in this document.

# The JSON Message layout

## Common structure for invoking all transactions

All the transactions must follow a predefined format to invoke transaction on MDM hub. The request and response have Tranaction Transfer Object (TxnTransferObj). The transfer object encapsulates Header and Payload objects. The response may additionally have error response object inside transfer object.

## Request Message Layout

### Header Object

The header object has below attributes

* requesterLanguage: The language of the request. It is referred from CONFIG\_LANGUAGE\_CODE table.
* requesterLocale: The locale of the requester
* userName: The username of the requester. This attribute is used for authorization
* userRole: The role of the requester. This attribute is used for authorization.
* txnCorrelationId: The correlation id of the message. This is sent back in the response as it is.
* txnMessageId: The message id of the transaction. The value of this attribute is used to update the UPDATED\_TXN\_REF\_ID during persistent transaction. i.e. for create / update transaction all the tables getting updated will have UPDATED\_TXN\_REF\_ID updated with this this id which can be used to trace the objects changed in one transaction.
* requestOriginSource: This attribute is used to define the request origin of the message. This attribute can be used to perform any additional logic while executing transaction.
* requesterSystem: This attribute is used to define the requesting system of the message. This attribute can be used to perform any additional logic while executing transaction.
* transactionResponseCode: This is the response code of the transaction which can either be SUCCESS or FAIL.
* requestTimeStamp: The timestamp when the request is received.
* responseTimeStamp: The timestamp when the response is sent back.
* TotalExecutionTimeMillies: The total execution time of the transaction in milliseconds.
* transactionServiceName: The name of the transaction being invoked on MDM hub. E.g. createLegalEntity.

Sample header Object -

|  |
| --- |
| "txnHeader": {  "requesterLanguage": "1",  "requesterLocale": null,  "userName": "yugandhar",  "userRole": "admin",  "accessToken": null,  "txnCorrelationId": null,  "txnMessageId": "",  "requestOriginSource": null,  "requesterSystem": null,  "transactionServiceName": "createLegalEntity"  } |

### TxnPayload Object

The txnPayload is the default object encapsulating all the data objects. Also this object has the pagination related properties as well.

Refer Javadoc for the list of DOs and attributes of this object.

### Primary Key Auto generation Vs providing in the request

Snippet: Create legal entity transaction with user provided idpk

|  |
| --- |
| "transactionServiceName": "createLegalEntity"  },  "txnPayload": {  "legalentityDO": {  "primaryKeyDO": {  "idPk": "77776663666AAAA211"  },  "displayName": "JOHN MCLEAN", |

Snippet: Create legal entity transaction with auto generated idpk

|  |
| --- |
| "transactionServiceName": "createLegalEntity"  },  "txnPayload": {  "legalentityDO": {  "idPk": null  "displayName": "JOHN MCLEAN", |

Refer section ‘Pluggable primary keys’ section of ‘Development and Customization guide’ for more details.

## Response Message Layout

### Header Object

Same as request object. Additionally ‘totalExecutionTimeMillies’ is sent in the response which denotes the time in milliseconds consumed for processing the request.

### TxnPayload Object

Same as request object. Additionally the response would have responseCode as ‘SUCCESS’ or ‘FAIL’ which denotes if the transaction is successful or failed in MDM Hub.

### Error Response

Sample error response

|  |
| --- |
| {  "responseCode": "FAIL",  "txnHeader": {  "requesterLanguage": "1",  "userName": "admin",  "userRole": "admin",  "txnMessageId": "0909cd3c-b4c1-495e-8ce1-6d8f89a7c9ec",  "transactionServiceName": "createLegalEntity"  },  "txnPayload": {"errorResponseObj": {  "validationResult": false,  "errorCode": "10085",  "errorMessage": "lePersonDO.genderRefkey is needed in the request",  "additionalErrorMessage": "lePersonDO.genderRefkey is needed in the request"  }}  } |

## Sample Request and Response Message

|  |  |
| --- | --- |
| **Sample Request** | **Sample Response** |
| {  **"txnHeader"**: {  **"requesterLanguage"**: "1",  **"requesterLocale"**: **null**,  **"userName"**: "yugandhar",  **"userRole"**: "admin",  **"accessToken"**: **null**,  **"txnCorrelationId"**: **null**,  **"txnMessageId"**: "",  **"requestOriginSource"**: **null**,  **"requesterSystem"**: **null**,  **"paginationStartIndex"**: **null**,  **"paginationEndIndex"**: **null**,  **"paginationTotalResultCount"**: **null**,  **"transactionServiceName"**: "createLegalEntity"  },  **"txnPayload"**: {  **"legalentityDO"**: {  **"primaryKeyDO"**: {  **"idPk"**: "11116663666777799d1n"  },  **"displayName"**: "SACHIN",  **"entityObjectTypeRefkey"**: "1",  **"classificationCodeRefkey"**: "1",  **"importanceTypeRefkey"**: "1",  **"leRatingRefkey"**: "1",  **"statusTypeRefkey"**: "1",  **"sourceSystemRefkey"**: "1",  **"onboardingDate"**: "2015-10-23",  **"offboardingDate"**: **null**,  **"kycVerificationFlag"**: "Y",  **"description"**: "NO Issues",  **"lePersonDO"**: {  **"legalentityIdpk"**: **null**,  **"version"**: **null**,  **"personTypeRefkey"**: **null**,  **"genderRefkey"**: "1",  **"dateOfBirth"**: "1989-10-23",  **"countryOfBirthRefkey"**: "356",  **"countryCitizenshipRefkey"**: "356",  **"countryOfDomicileRefkey"**: "356",  **"preferredLanguageRefkey"**: "1",  **"maritalStatus"**: **null**,  **"highestEduQualRefkey"**: **null**,  **"isDeceasedFlag"**: "Y",  **"deceasedDate"**: **null**,  **"isHandicappedFlag"**: **null**,  **"handicappedSinceDate"**: **null**,  **"numberOfDependents"**: **null**,  **"numberOfChildren"**: **null**,  **"prefereedLanguageRefkey"**: **null**,  **"personnamesDOList"**: [{  **"idPk"**: **null**,  **"version"**: **null**,  **"legalentityIdpk"**: **null**,  **"personnameTypeRefkey"**: "1",  **"prefixNameRefkey"**: **null**,  **"prefixMisc"**: **null**,  **"nameOne"**: "SACHIN",  **"nameTwo"**: "RAMESH",  **"nameThree"**: **null**,  **"nameFour"**: **null**,  **"lastName"**: "TENDULKAR",  **"nickName"**: "Master Blaster",  **"popularName"**: "Master Blaster",  **"suffixNameRefkey"**: 1,  **"suffixMisc"**: **null**,  **"nameStandardisedFlag"**: "N",  **"sourceSystemRefkey"**: "1"  }]  },  **"leSystemKeysRegistryDOList"**: [{  **"sourceSystemRefkey"**: "1",  **"referenceId"**: "112233112",  **"legalentityIdpk"**: **null**,  **"statusInSourceRefkey"**: "1",  **"description"**: "Sample"  }]  }  }  } | {  **"responseCode"**: "SUCCESS",  **"txnHeader"**: {  **"requesterLanguage"**: "1",  **"userName"**: "yugandhar",  **"userRole"**: "admin",  **"txnMessageId"**: "326dfd40-157b-4921-b896-55679a46cfa0",  **"transactionServiceName"**: "createLegalEntity",  **"totalExecutionTimeMillies"**: "1221"  },  **"txnPayload"**: {  **"legalentityDO"**: {  **"idPk"**: "11116663666777799d1n",  **"version"**: 0,  **"createdTs"**: "2018-01-08T09.39.24.041+0000",  **"updatedTs"**: "2018-01-08T09.39.24.041+0000",  **"updatedByUser"**: "rakesh",  **"updatedByTxnId"**: "326dfd40-157b-4921-b896-55679a46cfa0",  **"displayName"**: "SACHIN",  **"entityObjectTypeRefkey"**: "1",  **"entityObjectTypeRefValue"**: "LEPERSON",  **"classificationCodeRefkey"**: "1",  **"classificationCodeRefValue"**: "CUSTOMER",  **"importanceTypeRefkey"**: "1",  **"importanceTypeRefValue"**: "HIGH",  **"leRatingRefkey"**: "1",  **"leRatingRefValue"**: "SINGLE STAR",  **"statusTypeRefkey"**: "1",  **"statusTypeRefValue"**: "ACTIVE",  **"sourceSystemRefkey"**: "1",  **"sourceSystemRefValue"**: "LE SOURCE SYSTEM 1",  **"onboardingDate"**: "2015-10-23",  **"kycVerificationFlag"**: "Y",  **"description"**: "NO Issues",  **"phoneticDisplayName"**: "SACAN",  **"lePersonDO"**: {  **"legalentityIdpk"**: "11116663666777799d1n",  **"version"**: 0,  **"createdTs"**: "2018-01-08T09.39.24.100+0000",  **"updatedTs"**: "2018-01-08T09.39.24.100+0000",  **"updatedByUser"**: "rakesh",  **"updatedByTxnId"**: "326dfd40-157b-4921-b896-55679a46cfa0",  **"genderRefkey"**: "1",  **"genderRefValue"**: "MALE",  **"dateOfBirth"**: "1989-10-23",  **"countryOfBirthRefkey"**: "356",  **"countryOfBirthRefValue"**: "IND",  **"countryCitizenshipRefkey"**: "356",  **"countryCitizenshipRefValue"**: "IND",  **"countryOfDomicileRefkey"**: "356",  **"countryOfDomicileRefValue"**: "IND",  **"isDeceasedFlag"**: "Y",  **"preferredLanguageRefkey"**: "1",  **"preferredLanguageRefValue"**: "ENGLISH",  **"personnamesDOList"**: [{  **"idPk"**: "45e1840d-50f6-4e3a-9cac-b3a0662c6ad4",  **"version"**: 0,  **"createdTs"**: "2018-01-08T09.39.24.422+0000",  **"updatedTs"**: "2018-01-08T09.39.24.422+0000",  **"updatedByUser"**: "rakesh",  **"updatedByTxnId"**: "326dfd40-157b-4921-b896-55679a46cfa0",  **"legalentityIdpk"**: "11116663666777799d1n",  **"personnameTypeRefkey"**: "1",  **"personnameTypeRefValue"**: "REGISTERED NAME",  **"nameOne"**: "SACHIN",  **"nameTwo"**: "RAMESH",  **"lastName"**: "TENDULKAR",  **"nickName"**: "Master Blaster",  **"popularName"**: "Master Blaster",  **"suffixNameRefkey"**: "1",  **"suffixNameRefValue"**: "Sr",  **"nameStandardisedFlag"**: "N",  **"sourceSystemRefkey"**: "1",  **"sourceSystemRefValue"**: "LE SOURCE SYSTEM 1",  **"phoneticNameOne"**: "SACAN",  **"phoneticNameTwo"**: "RAN",  **"phoneticLastName"**: "TANDAL"  }]  },  **"leSystemKeysRegistryDOList"**: [{  **"idPk"**: "350eb0d0-3531-4a09-843b-bbbfa61604e9",  **"version"**: 0,  **"createdTs"**: "2018-01-08T09.39.24.622+0000",  **"updatedTs"**: "2018-01-08T09.39.24.622+0000",  **"updatedByUser"**: "rakesh",  **"updatedByTxnId"**: "326dfd40-157b-4921-b896-55679a46cfa0",  **"sourceSystemRefkey"**: "1",  **"sourceSystemRefValue"**: "LE SOURCE SYSTEM 1",  **"referenceId"**: "112233112",  **"legalentityIdpk"**: "11116663666777799d1n",  **"statusInSourceRefkey"**: "1",  **"statusInSourceRefValue"**: "ACTIVE",  **"description"**: "Sample"  }]  }  }  } |

# Master repository of sample messages

Sample JSON and XML messages are available at githubrepo/resources/Testing location.

# Out of the Box Transactions

## Base entity services

Base entity services perform CRUD (create, retrieve, update and delete) operations on single database entity.

## Composite Services

Composite entity services perform CRUD (create, retrieve, update and delete) operations on multiple database entities as single transaction. Also composite services can have composition of create/ retrieve/ update or delete in single services. e.g. updateLegalEntity service update the legal entity record along with person, corporation, address and identifiers etc. also you may associate new address or identifier through this service itself.

Also, search services can also considered as composite services.

## Application Configuration Management services

### createConfigAppPropertiesBase

#### Description: create a record in CONFIG\_APP\_PROPERTIES table

#### Transaction Service Name: createConfigAppPropertiesBase

#### Request DO: ConfigAppPropertiesDO

#### Response DO: ConfigAppPropertiesDO

#### Mandatory Input: key, value

### createConfigErrorcodeRegistryBase

#### Description:create a record in CONFIG\_ERRORCODE\_REGISTRY table

#### Transaction Service Name: createConfigErrorcodeRegistryBase

#### Request DO: ConfigErrorcodeRegistryDO

#### Response DO: ConfigErrorcodeRegistryDO

#### Mandatory Input: configLanguageCodeKey, errorCode, errorMessage

### createConfigInquiryLevelsBase

#### Description: create a record in CONFIG\_INQUIRY\_LEVELS table.

#### Transaction Service Name: createConfigInquiryLevelsBase

#### Request DO: ConfigInquiryLevelsDO

#### Response DO: ConfigInquiryLevelsDO

#### Mandatory Input: inquiryLevel, applicableDobj, childDobj

### createConfigLanguageCodeBase

#### Description:create a record in CONFIG\_LANGUAGE\_CODE table.

#### Transaction Service Name: createConfigLanguageCodeBase

#### Request DO: ConfigLanguageCodeDO

#### Response DO: ConfigLanguageCodeDO

#### Mandatory Input: key, value

### createConfigTxnRegistryBase

#### Description: create a record in CONFIG\_TXN\_REGISTRY table

#### Transaction Service Name: createConfigTxnRegistryBase

#### Request DO: ConfigTxnRegistryDO

#### Response DO: ConfigTxnRegistryDO

#### Mandatory Input: txnserviceName, txnserviceClass,txnserviceClassmethod

### findAllConfigErrorcodeRegistryByLanguageCodeBase

#### Description:find all records in CONFIG\_ERRORCODE\_REGISTRY table for given language code

#### Transaction Service Name: findAllConfigErrorcodeRegistryByLanguageCodeBase

#### Request DO: ConfigErrorcodeRegistryDO

#### Response DO: ConfigErrorcodeRegistryDO

#### Mandatory Input: configLanguageCodeKey

#### inquiryFilter: ACTIVE/INACTIVE/ALL

### findAllConfigInquiryLevelsBase

#### Description: find all records in CONFIG\_INQUIRY\_LEVELS table

#### Transaction Service Name: findAllConfigInquiryLevelsBase

#### Request DO: ConfigInquiryLevelsDO

#### Response DO: ConfigInquiryLevelsDO

#### Mandatory Input: N/A

#### inquiryFilter: ACTIVE/INACTIVE/ALL

### findAllConfigInquiryLevelsByBusinessKeyBase

#### Description: find all records in CONFIG\_INQUIRY\_LEVELS table based on inquiryLevel and applicableDobj

#### Transaction Service Name: findAllConfigInquiryLevelsByBusinessKeyBase

#### Request DO: ConfigInquiryLevelsDO

#### Response DO: ConfigInquiryLevelsDO

#### Mandatory Input: inquiryLevel and applicableDobj

#### inquiryFilter: ACTIVE/INACTIVE/ALL

### findAllConfigLanguageCodesBase

#### Description:find all records in CONFIG\_LANGUAGE\_CODE table based on idpk

#### Transaction Service Name: findAllConfigLanguageCodesBase

#### Request DO: ConfigLanguageCodeDO

#### Response DO: ConfigLanguageCodeDO

#### Mandatory Input: N/A

#### inquiryFilter: ACTIVE/INACTIVE/ALL

### findConfigAppPropertiesByBusinessKeyBase

#### Description: find a record in CONFIG\_APP\_PROPERTIES table based on key

#### Transaction Service Name: findConfigAppPropertiesByBusinessKeyBase

#### Request DO: ConfigAppPropertiesDO

#### Response DO: ConfigAppPropertiesDO

#### Mandatory Input: key

#### inquiryFilter: ACTIVE/INACTIVE/ALL

### findConfigErrorcodeRegistryByBusinessKeyBase

#### Description:find a record in CONFIG\_ERRORCODE\_REGISTRY table based on business key

#### Transaction Service Name: findConfigErrorcodeRegistryByBusinessKeyBase

#### Request DO: ConfigErrorcodeRegistryDO

#### Response DO: ConfigErrorcodeRegistryDO

#### Mandatory Input: configLanguageCodeKey, errorCode

### findConfigLanguageCodeByBusinessKeyBase

#### Description:find a record in CONFIG\_LANGUAGE\_CODE table based on business key

#### Transaction Service Name: findConfigLanguageCodeByBusinessKeyBase

#### Request DO: ConfigLanguageCodeDO

#### Response DO: ConfigLanguageCodeDO

#### Mandatory Input: key

### findConfigTxnRegistryByBusinessKeyBase

#### Description: find a record in CONFIG\_TXN\_REGISTRY table based on business key

#### Transaction Service Name: findConfigTxnRegistryByBusinessKeyBase

#### Request DO: ConfigTxnRegistryDO

#### Response DO: ConfigTxnRegistryDO

#### Mandatory Input: txnserviceName

### retrieveConfigAppPropertiesBase

#### Description: retrieve a record in CONFIG\_APP\_PROPERTIES table based on idpk

#### Transaction Service Name: retrieveConfigAppPropertiesBase

#### Request DO: ConfigAppPropertiesDO

#### Response DO: ConfigAppPropertiesDO

#### Mandatory Input: idPk

### retrieveConfigErrorcodeRegistryBase

#### Description:retrieve a record in CONFIG\_ERRORCODE\_REGISTRY table based on idpk

#### Transaction Service Name: retrieveConfigErrorcodeRegistryBase

#### Request DO: ConfigErrorcodeRegistryDO

#### Response DO: ConfigErrorcodeRegistryDO

#### Mandatory Input: idPk

### retrieveConfigInquiryLevelsBase

#### Description: retrievea record in CONFIG\_INQUIRY\_LEVELS table based on idpk

#### Transaction Service Name: retrieveConfigInquiryLevelsBase

#### Request DO: ConfigInquiryLevelsDO

#### Response DO: ConfigInquiryLevelsDO

#### Mandatory Input: idPk

### retrieveConfigLanguageCodeBase

#### Description: retrieve a record in CONFIG\_LANGUAGE\_CODE table based on idpk

#### Transaction Service Name: retrieveConfigLanguageCodeBase

#### Request DO: ConfigLanguageCodeDO

#### Response DO: ConfigLanguageCodeDO

#### Mandatory Input: idPk

### retrieveConfigTxnRegistryBase

#### Description: retrieve a record in CONFIG\_TXN\_REGISTRY table based on idpk

#### Transaction Service Name: retrieveConfigTxnRegistryBase

#### Request DO: ConfigTxnRegistryDO

#### Response DO: ConfigTxnRegistryDO

#### Mandatory Input: idPk

### updateConfigAppPropertiesBase

#### Description: update a record in CONFIG\_APP\_PROPERTIES table based on idpk

#### Transaction Service Name: updateConfigAppPropertiesBase

#### Request DO: ConfigAppPropertiesDO

#### Response DO: ConfigAppPropertiesDO

#### Mandatory Input: idPk and version

### updateConfigErrorcodeRegistryBase

#### Description: update a record in CONFIG\_ERRORCODE\_REGISTRY table based on idpk

#### Transaction Service Name: updateConfigErrorcodeRegistryBase

#### Request DO: ConfigErrorcodeRegistryDO

#### Response DO: ConfigErrorcodeRegistryDO

#### Mandatory Input: idPk and version

### updateConfigInquiryLevelsBase

#### Description: update a record in CONFIG\_INQUIRY\_LEVELS table based on idpk

#### Transaction Service Name: updateConfigInquiryLevelsBase

#### Request DO: ConfigInquiryLevelsDO

#### Response DO: ConfigInquiryLevelsDO

#### Mandatory Input: idPk, version

### updateConfigLanguageCodeBase

#### Description: update a record in CONFIG\_LANGUAGE\_CODE table based on idpk

#### Transaction Service Name: updateConfigLanguageCodeBase

#### Request DO: ConfigLanguageCodeDO

#### Response DO: ConfigLanguageCodeDO

#### Mandatory Input: idPk, version

### updateConfigTxnRegistryBase

#### Description: update a record in CONFIG\_TXN\_REGISTRY table based on idpk

#### Transaction Service Name: updateConfigTxnRegistryBase

#### Request DO: ConfigTxnRegistryDO

#### Response DO: ConfigTxnRegistryDO

#### Mandatory Input: idPk, version

## Authorization Framework services

### createAuthRolesRegistryBase

#### Description: create a record in AUTH\_ROLES\_REGISTRY table

#### Transaction Service Name: createAuthRolesRegistryBase

#### Request DO: AuthRolesRegistryDO

#### Response DO: AuthRolesRegistryDO

#### Mandatory Input: roleName

### updateAuthRolesRegistryBase

#### Description: update a record in AUTH\_ROLES\_REGISTRY based on idPk

#### Transaction Service Name: updateAuthRolesRegistryBase

#### Request DO: AuthRolesRegistryDO

#### Response DO: AuthRolesRegistryDO

#### Mandatory Input: idPk, version, roleName

### retrieveAuthRolesRegistryBase

#### Description: retrieve a record fromAUTH\_ROLES\_REGISTRY based on idPk

#### Transaction Service Name: retrieveAuthRolesRegistryBase

#### Request DO: AuthRolesRegistryDO

#### Response DO: AuthRolesRegistryDO

#### Mandatory Input: idPk

### findAuthRolesRegistryByBusinessKeyBase

#### Description: find a record from AUTH\_ROLES\_REGISTRY based on business key

#### Transaction Service Name: findAuthRolesRegistryByBusinessKeyBase

#### Request DO: authRolesRegistryDO

#### Response DO: authRolesRegistryDO

#### Mandatory Input: roleName

### createAuthUserRegistryBase

#### Description: create a record in AUTH\_USER\_REGISTRY table

#### Transaction Service Name: createAuthUserRegistryBase

#### Request DO: authUserRegistryDO

#### Response DO: authUserRegistryDO

#### Mandatory Input: userName

### updateAuthUserRegistryBase

#### Description: update a record in AUTH\_USER\_REGISTRY based on idPk

#### Transaction Service Name: updateAuthUserRegistryBase

#### Request DO: authUserRegistryDO

#### Response DO: authUserRegistryDO

#### Mandatory Input: idPk, version, userName

### retrieveAuthUserRegistryBase

#### Description: retrieve a record from AUTH\_USER\_REGISTRY based on idPk

#### Transaction Service Name: retrieveAuthUserRegistryBase

#### Request DO: authUserRegistryDO

#### Response DO: authUserRegistryDO

#### Mandatory Input: idPk

### findAuthUserRegistryByBusinessKeyBase

#### Description: find a record from AUTH\_USER\_REGISTRY based business key

#### Transaction Service Name: findAuthUserRegistryByBusinessKeyBase

#### Request DO: authUserRegistryDO

#### Response DO: authUserRegistryDO

#### Mandatory Input: userName

### createAuthUserroleAccesscontrolBase

#### Description: create a record in AUTH\_USERROLE\_ACCESSCONTROL table

#### Transaction Service Name: createAuthUserroleAccesscontrolBase

#### Request DO: authUserroleAccesscontrolDO

#### Response DO: authUserroleAccesscontrolDO

#### Mandatory Input: profileType, authUserRoleRegistryIdpk, configTxnRegistryIdpk

### updateAuthUserroleAccesscontrolBase

#### Description: update a record in AUTH\_USERROLE\_ACCESSCONTROL table based on idPk

#### Transaction Service Name: updateAuthUserroleAccesscontrolBase

#### Request DO: authUserroleAccesscontrolDO

#### Response DO: authUserroleAccesscontrolDO

#### Mandatory Input: idPk, version, profileType, authUserRoleRegistryIdpk, configTxnRegistryIdpk

### retrieveAuthUserroleAccesscontrolBase

#### Description: retrieve a record from AUTH\_USERROLE\_ACCESSCONTROL based on idPk

#### Transaction Service Name: retrieveAuthUserroleAccesscontrolBase

#### Request DO: authUserroleAccesscontrolDO

#### Response DO: authUserroleAccesscontrolDO

#### Mandatory Input: idPk

### findAuthUserroleAccesscontrolByBusinessKeyBase

#### Description: find a record from AUTH\_USERROLE\_ACCESSCONTROL table based on business key

#### Transaction Service Name: findAuthUserroleAccesscontrolByBusinessKeyBase

#### Request DO: authUserroleAccesscontrolDO

#### Response DO: authUserroleAccesscontrolDO

#### Mandatory Input: profileType, authUserRoleRegistryIdpk, configTxnRegistryIdpk

### findAuthUserroleAccesscontrolByRegistryIdpkBase

#### Description: find a record from AUTH\_USERROLE\_ACCESSCONTROL based on business key

#### Transaction Service Name: findAuthUserroleAccesscontrolByRegistryIdpkBase

#### Request DO: authUserroleAccesscontrolDO

#### Response DO: authUserroleAccesscontrolDO

#### Mandatory Input: profileType, authUserRoleRegistryIdpk

### createAuthUserRoleAssocBase

#### Description: create a record in AUTH\_USER\_ROLE\_ASSOC table

#### Transaction Service Name: createAuthUserRoleAssocBase

#### Request DO: authUserRoleAssocDO

#### Response DO: authUserRoleAssocDO

#### Mandatory Input: authRolesRegistryIdpk, authUserRegistryIdpk

### updateAuthUserRoleAssocBase

**Description**: update a record in AUTH\_USER\_ROLE\_ASSOC based on idPk

***Transaction Service Name:*** *updateAuthUserRoleAssocBase*

***Request DO:*** *authUserRoleAssocDO*

***Response DO:*** *authUserRoleAssocDO*

***Mandatory Input:*** *idPk, version , authRolesRegistryIdpk, authUserRegistryIdpk*

### retrieveAuthUserRoleAssocBase

**Description:** retrieve a record from AUTH\_USER\_ROLE\_ASSOC based on idPk

**Transaction Service Name:** retrieveAuthUserRoleAssocBase

**Request DO:** authUserRoleAssocDO

**Response DO:** authUserRoleAssocDO

**Mandatory Input:** idPk

### findAuthUserRoleAssocByBusinessKeyBase

**Description:** find a record from AUTH\_USER\_ROLE\_ASSOC based on business key

**Transaction Service Name:** findAuthUserRoleAssocByBusinessKeyBase

**Request DO:** authUserRoleAssocDO

**Response DO:** authUserRoleAssocDO

**Mandatory Input:** authRolesRegistryIdpk and authUserRegistryIdpk

### findAllRecordsByAuthUserRegistryIdpkBase

**Description:** find AUTH\_USER\_ROLE\_ASSOC based on authUserRegistryIdpk

**Transaction Service Name:** findAllRecordsByAuthUserRegistryIdpkBase

**Request DO:** authUserRoleAssocDO

**Response DO:** authUserRoleAssocDO

**Mandatory Input:** authUserRegistryIdpk

### findAllRecordsByAuthRolesRegistryIdpkBase

**Description:** find all records from AUTH\_USER\_ROLE\_ASSOC based on authRolesRegistryIdpk

**Transaction Service Name:** findAllRecordsByAuthRolesRegistryIdpkBase

**Request DO:** authUserRoleAssocDO

**Response DO:** authUserRoleAssocDO

**Mandatory Input:** authRolesRegistryIdpk

## Batch Framework Management services

### createBatchEntityToProcessBase

**Description:** create a record in BATCH\_ENTITY\_TO\_PROCESS table

**Transaction Service Name:** createBatchEntityToProcessBase

**Request DO:** batchEntityToProcessDO

**Response DO:** batchEntityToProcessDO

**Mandatory Input:** entityObjectTypeRefkey, entityIdpk, batchProposedActionRefkey, batchActionStatusRefkey, entryMadeBySubsystemName

### updateBatchEntityToProcessBase

**Description:** update a record in BATCH\_ENTITY\_TO\_PROCESS table based on idPk

**Transaction Service Name:** updateBatchEntityToProcessBase

**Request DO:** batchEntityToProcessDO

**Response DO:** batchEntityToProcessDO

**Mandatory Input:** idPk, version

### retrieveBatchEntityToProcessBase

**Description:** retrieve a record from BATCH\_ENTITY\_TO\_PROCESS table based on idPk

**Transaction Service Name:** retrieveBatchEntityToProcessBase

**Request DO:** batchEntityToProcessDO

**Response DO:** batchEntityToProcessDO

**Mandatory Input:** idPk

## Match And Merge Framework Management services

### createInactiveLeRegistryBase

#### Description: create a record in INACTIVE\_LE\_REGISTRY table

#### Transaction Service Name: createInactiveLeRegistryBase

#### Request DO: inactiveLeRegistryDO

#### Response DO: inactiveLeRegistryDO

#### Mandatory Input: legalentityIdpk, inactivatedTs, inactivationReasonRefkey

### createMatchCandidateLeRegistryBase

#### Description: create a record in MATCH\_CANDIDATE\_LE\_REGISTRY table

#### Transaction Service Name: createMatchCandidateLeRegistryBase

#### Request DO: matchCandidateLeRegistryDO

#### Response DO: matchCandidateLeRegistryDO

#### Mandatory Input: legalentityIdpk, candidateLegalentityidpk, matchPattern, matchProposedActionRefkey, matchActionstatusRefkey

### createMatchMergedLeAssocBase

#### Description: create a record in MATCH\_MERGED\_LE\_ASSOC table

#### Transaction Service Name: createMatchMergedLeAssocBase

#### Request DO: matchMergedLeAssocDO

#### Response DO: matchMergedLeAssocDO

#### Mandatory Input: survivorLegalentityIdpk, mergedLegalentityIdpk, mergeReasonRefkey

### createRefMatchScoreBase

#### Description: create a record in REF\_MATCH\_SCORE reference table.

#### Transaction Service Name: createRefMatchScoreBase

#### Request DO: refMatchScoreDO

#### Response DO:refMatchScoreDO

#### Mandatory Input: matchEntityObjectName, matchAttrPattern, matchResultRefkey, matchProposedActionRefkey

### createRefMatchThresholdBase

#### Description: create a record in REF\_MATCH\_THRESHOLD table.

#### Transaction Service Name: createRefMatchThresholdBase

#### Request DO: refMatchThresholdDO

#### Response DO: refMatchThresholdDO

#### Mandatory Input: attrBlockName, matchThreshold, decayThresholdInDays, decayPercentage, maxDecayPercentage

### findAllRefMatchScoreByMatchEntityObjectName

#### Description: retrieve all records in REF\_MATCH\_SCORE reference table based on entity object name.

#### Transaction Service Name: findAllRefMatchScoreByMatchEntityObjectName

#### Request DO: refMatchScoreDO

#### Response DO: refMatchScoreDO

#### Mandatory Input: matchEntityObjectName

### findAllRefMatchThresholdBase

#### Description: find all records from REF\_MATCH\_THRESHOLD table.

#### Transaction Service Name: createRefMatchThresholdBase

#### Request DO: refMatchThresholdDO

#### Response DO: refMatchThresholdDOList

#### Mandatory Input: NA

### findRefMatchScoreByBusinessKeyBase

#### Description: find a record in REF\_MATCH\_SCORE reference table based on business key

#### Transaction Service Name: findRefMatchScoreByBusinessKeyBase

#### Request DO: refMatchScoreDO

#### Response DO: refMatchScoreDO

#### Mandatory Input: matchEntityObjectName, matchAttrPattern

### findRefMatchThresholdByBusinessKeyBase

#### Description: find a record from REF\_MATCH\_THRESHOLD table based on business key.

#### Transaction Service Name: findRefMatchThresholdByBusinessKeyBase

#### Request DO: refMatchThresholdDO

#### Response DO: refMatchThresholdDO

#### Mandatory Input: attrBlockName

### retrieveInactiveLeRegistryBase

#### Description: retrieve a record in INACTIVE\_LE\_REGISTRY table based on idPk

#### Transaction Service Name: retrieveInactiveLeRegistryBase

#### Request DO: inactiveLeRegistryDO

#### Response DO: inactiveLeRegistryDO

#### Mandatory Input: idPk

### retrieveMatchCandidateLeRegistryBase

#### Description: retrieve a record in MATCH\_CANDIDATE\_LE\_REGISTRY table based on idPk

#### Transaction Service Name: retrieveMatchCandidateLeRegistryBase

#### Request DO: matchCandidateLeRegistryDO

#### Response DO: matchCandidateLeRegistryDO

#### Mandatory Input: idPk

### retrieveMatchMergedLeAssocBase

#### Description: retrieve a record in MATCH\_MERGED\_LE\_ASSOC table based on idPk

#### Transaction Service Name: retrieveMatchMergedLeAssocBase

#### Request DO: matchMergedLeAssocDO

#### Response DO: matchMergedLeAssocDO

#### Mandatory Input: idPk

### retrieveRefMatchScoreBase

#### Description: retrieve a record in REF\_MATCH\_SCORE reference table based on idpk

#### Transaction Service Name: retrieveRefMatchScoreBase

#### Request DO: refMatchScoreDO

#### Response DO: refMatchScoreDO

#### Mandatory Input: idPk

### retrieveRefMatchThresholdBase

#### Description: retrieve a record in REF\_MATCH\_THRESHOLD table based on idPk

#### Transaction Service Name: retrieveRefMatchThresholdBase

#### Request DO: refMatchThresholdDO

#### Response DO: refMatchThresholdDO

#### Mandatory Input: idPk

### updateInactiveLeRegistryBase

#### Description: update a record in INACTIVE\_LE\_REGISTRY table based on idPk

#### Transaction Service Name: updateInactiveLeRegistryBase

#### Request DO: inactiveLeRegistryDO

#### Response DO: inactiveLeRegistryDO

#### Mandatory Input: idPk, version

### updateMatchCandidateLeRegistryBase

#### Description: update a record in MATCH\_CANDIDATE\_LE\_REGISTRY table based on idPk

#### Transaction Service Name: updateMatchCandidateLeRegistryBase

#### Request DO: matchCandidateLeRegistryDO

#### Response DO: matchCandidateLeRegistryDO

#### Mandatory Input: idPk

### updateMatchMergedLeAssocBase

#### Description: update a record in MATCH\_MERGED\_LE\_ASSOC table based on idPk

#### Transaction Service Name: updateMatchMergedLeAssocBase

#### Request DO: matchMergedLeAssocDO

#### Response DO: matchMergedLeAssocDO

#### Mandatory Input: idPk, version

### updateRefMatchScoreBase

#### Description: update a record in REF\_MATCH\_SCORE reference table based on idpk

#### Transaction Service Name: updateRefMatchScoreBase

#### Request DO: refMatchScoreDO

#### Response DO: refMatchScoreDO

#### Mandatory Input: idPk

### updateRefMatchThresholdBase

#### Description: retrieve a record in REF\_MATCH\_THRESHOLD table based on idPk

#### Transaction Service Name: updateRefMatchThresholdBase

#### Request DO: refMatchThresholdDO

#### Response DO: refMatchThresholdDO

#### Mandatory Input: idPk

## Data Table Management Services (Base)

### createAccountAddressAssocBase

**Description**: create a record in ACCOUNT\_ADDRESS\_ASSOC table

**Transaction Service Name**: createAccountAddressAssocBase

**Request DO**: accountAddressAssocDO

**Response DO**: accountAddressAssocDO

**Mandatory Input:** accountIdpk, addressIdpk, addressTypeRefkey, addressSubtypeRefkey

### createAccountBase

**Description:**  create a record in ACCOUNT table

**Transaction Service Name:** createAccountBase

**Request DO:** accountDO

**Response DO:** accountDO

**Mandatory Input:** accountName

### createAccountPhoneAssocBase

**Description**: create a record in ACCOUNT\_PHONE\_ASSOC table

**Transaction Service Name:** createAccountPhoneAssocBase

**Request DO:** AccountPhoneAssocDO

**Response DO:** AccountPhoneAssocDO

**Mandatory Input:** accountIdpk, phoneTypeRefkey, phoneNumber

### createAddressBase

**Description:** create a record in ADDRESS table

**Transaction Service Name:** createAddressBase

**Request DO:** AddressDO

**Response DO:** AddressDO

**Mandatory Input:** addressLineOne, countryRefkey, city

### createCorporationnamesBase

**Description:** create a record in CORPORATIONNAMES table

**Transaction Service Name:** createCorporationnamesBase

**Request DO:** corporationnamesDO

**Response DO:** corporationnamesDO

**Mandatory Input:** legalEntityIdPk, corporationNameTypeRefkey, corporationName

### createEntityGroupAssocBase

**Description:**  create a record in ENTITY\_GROUP\_ASSOC table

**Transaction Service Name:** createEntityGroupAssocBase

**Request DO:** entityGroupAssocDO

**Response DO:** entityGroupAssocDO

**Mandatory Input:** entityObjectTypeRefkey, entityIdpk, entityGroupIdpk, assocTypeRefkey

### createEntityGroupBase

**Description:** create a record in ENTITY\_GROUP table

**Transaction Service Name:** createEntityGroupBase

**Request DO:** entityGroupDO

**Response DO:** entityGroupDO

**Mandatory Input:** groupTypeRefkey, groupName

### createLeAccountAssocBase

**Description:** create a record in LE\_ACCOUNT\_ASSOC table.

**Transaction Service Name:** createLeAccountAssocBase

**Request DO:** leAccountAssocDO

**Response DO:** leAccountAssocDO

**Mandatory Input:** legalEntityIdPk, roletypeRefkey, accountIdpk

### createLeAddressAssocBase

**Description:** create a record in LE\_ADDRESS\_ASSOC table

**Transaction Service Name:** createLeAddressAssocBase

**Request DO:** leAddressAssocDO

**Response DO:** leAddressAssocDO

**Mandatory Input:** legalEntityIdPk, addressIdpk, addressTypeRefkey, addressSubtypeRefkey

### createLeCorporationBase

**Description:** create a record in LE\_CORPORATION table

**Transaction Service Name:** createLeCorporationBase

**Request DO:** LeCorporationDO

**Response DO:** LeCorporationDO

**Mandatory Input:** legalentityIdpk

### createLeIdentifierKycRegistryBase

**Description:** create a record in LE\_IDENTIFIER\_KYC\_REGISTRY table

**Transaction Service Name:** createLeIdentifierKycRegistryBase

**Request DO:** leIdentifierKycRegistryDO

**Response DO:** leIdentifierKycRegistryDO

**Mandatory Input:** legalEntityIdPk, identificationTypeRefkey, identificationNumber

### createLePersonBase

**Description:** create a record in LE\_PERSON table

**Transaction Service Name:** createLePersonBase

**Request DO:** lePersonDO

**Response DO:** lePersonDO

**Mandatory Input:** genderRefkey

### createLePhoneAssocBase

**Description:** create a record in LE\_PHONE\_ASSOC table

**Transaction Service Name:** createLePhoneAssocBase

**Request DO:** lePhoneAssocDO

**Response DO:** lePhoneAssocDO

**Mandatory Input:** LegalEntityIdPk, phoneTypeRefkey, phoneNumber

### createLePreferencesBase

**Description:** create a record in LE\_PREFERENCES table

**Transaction Service Name:** createLePreferencesBase

**Request DO:** lePreferencesDO

**Response DO:** lePreferencesDO

**Mandatory Input:** legalEntityIdPk, preferenceTypeRefkey

### createLePropertyAssocBase

**Description:** create a record in LE\_PROPERTY\_ASSOC table

**Transaction Service Name:** createLePropertyAssocBase

**Request DO:** lePropertyAssocDO

**Response DO:** lePropertyAssocDO

**Mandatory Input:** legalEntityIdPk, propertyIdpk, propertyLeReltypeRefkey

### createLeSystemKeysRegistryBase

**Description:** create a record in LE\_SYSTEM\_KEYS\_REGISTRY table

**Transaction Service Name:** createLeSystemKeysRegistryBase

**Request DO:** leSystemKeysRegistryDO

**Response DO:** leSystemKeysRegistryDO

**Mandatory Input:** LegalEntityIdPk, sourceSystemRefkey, referenceId

### createLeToLeRelationshipBase

**Description:** create a record in LE\_TO\_LE\_RELATIONSHIP table

**Transaction Service Name:** createLeToLeRelationshipBase

**Request DO:** leToLeRelationship

**Response DO:** leToLeRelationship

**Mandatory Input:** fromLegalentityIdpk, toLegalentityIdpk, leRelationshipTypeRefkey

### createLeVehicleAssocBase

**Description:** create a record in LE\_VEHICLE\_ASSOC table

**Transaction Service Name:** createLeVehicleAssocBase

**Request DO:** leVehicleAssocDO

**Response DO:** leVehicleAssocDO

**Mandatory Input:** legalEntityIdPk, leRoletypeRefkey

### createLegalentityBase

**Description:** create a record in LEGALENTITY table

**Transaction Service Name:** createLegalentityBase

**Request DO:** legalentityDO

**Response DO:** legalentityDO

**Mandatory Input:** displayName, entityObjectTypeRefkey

### createMiscellaneousInfoBase

**Description:** create a record in MISCELLANEOUS\_INFO table

**Transaction Service Name:** createMiscellaneousInfoBase

**Request DO:** miscellaneousInfoDO

**Response DO:** miscellaneousInfoDO

**Mandatory Input:** entityObjectTypeRefkey, entityIdpk, name1, value1

### createPersonnamesBase

**Description:** create a record in PERSONNAMES table

**Transaction Service Name:** createPersonnamesBase

**Request DO:** personnamesDO

**Response DO:** personnamesDO

**Mandatory Input:** LegalEntityIdPk, personnameTypeRefkey, nameOne

### createPhoneStandardizedBase

**Description:** create a record in PHONE\_STANDARDIZED table

**Transaction Service Name:** createPhoneStandardizedBase

**Request DO:** phoneStandardizedDO

**Response DO:** phoneStandardizedDO

**Mandatory Input:** N/A

### createPropertyBase

**Description:** create a record in PROPERTY table

**Transaction Service Name:** createPropertyBase

**Request DO:** propertyDO

**Response DO:** propertyDO

**Mandatory Input:** propertyName

### createVehicleBase

**Description:** create a record inVEHICLE table

**Transaction Service Name:** createVehicleBase

**Request DO:** vehicleDO

**Response DO:** vehicleDO

**Mandatory Input:** vinNumber

### findAllAccountAddressAssocByAccountIdPkBase

**Description:** find all records in ACCOUNT\_ADDRESS\_ASSOC table based on account idPk

**Transaction Service Name:** findAllAccountAddressAssocByAccountIdPkBase

**Request DO:** accountAddressAssocDO

**Response DO:** accountAddressAssocDOList

**Mandatory Input:** accountIdpk

**Pagination Supported:** Yes

**Filter:** ACTIVE / INACTIVE / ALL

### findAllAccountPhoneAssocByAccountIdPkBase

**Description:** create a record in ACCOUNT\_PHONE\_ASSOC table

**Transaction Service Name:** findAllAccountPhoneAssocByAccountIdPkBase

**Request DO:** accountPhoneAssocDO

**Response DO:** accountPhoneAssocDOList

**Mandatory Input:** accountIdpk

**Pagination Supported:** Yes

**Filter:** ACTIVE / INACTIVE / ALL

### findAllMiscellaneousInfoByEntityIdPkBase

**Description:** find all records in MISCELLANEOUS\_INFO table based on entity id

**Transaction Service Name:** findAllMiscellaneousInfoByEntityIdPkBase

**Request DO:** miscellaneousInfoDO

**Response DO:** miscellaneousInfoDOList

**Mandatory Input:** entityIdpk, entityObjectTypeRefkey

**Pagination Supported:** Yes

**Filter:** ACTIVE / INACTIVE / ALL

### findCorporationnamesByLegalEntityIdBase

**Description:** find all records in CORPORATIONNAMES table based on legal entity Idpk

**Transaction Service Name:** findCorporationnamesByLegalEntityIdBase

**Request DO:** corporationnamesDO

**Response DO:** corporationnamesDOList

**Mandatory Input:** legalentityIdpk

**Pagination Supported:** Yes

**Filter:** ACTIVE / INACTIVE / ALL

### findEntityGroupAssocByEntityGroupIdpkBase

**Description:**  find all records in ENTITY\_GROUP\_ASSOC table based on entity group idPk

**Transaction Service Name:** findEntityGroupAssocByEntityGroupIdpkBase

**Request DO:** entityGroupAssocDO

**Response DO:** entityGroupAssocDOList

**Mandatory Input:** entityGroupIdpk

**Pagination Supported:** Yes

**Filter:** ACTIVE / INACTIVE / ALL

### findEntityGroupAssocByEntityIdPkBase

**Description:**  find all records in ENTITY\_GROUP\_ASSOC table based on entity idPk

**Transaction Service Name:** findEntityGroupAssocByEntityIdPkBase

**Request DO:** entityGroupAssocDO

**Response DO:** entityGroupAssocDOList

**Mandatory Input:** entityIdpk

**Pagination Supported:** Yes

**Filter:** ACTIVE / INACTIVE / ALL

### findLeAccountAssocByAccountIdPkBase

**Description:** find all records in LE\_ACCOUNT\_ASSOC table based on account idPk

**Transaction Service Name:** findLeAccountAssocByAccountIdPkBase

**Request DO:** leAccountAssocDO

**Response DO:** leAccountAssocDOList

**Mandatory Input:** accountIdPk

**Pagination Supported:** Yes

**Filter:** ACTIVE / INACTIVE / ALL

### findLeAccountAssocByLegalEntityIdPkBase

**Description:** find all records in LE\_ACCOUNT\_ASSOC table based on legal entity idPk

**Transaction Service Name:** findLeAccountAssocByAccountIdPkBase

**Request DO:** leAccountAssocDO

**Response DO:** leAccountAssocDOList

**Mandatory Input:** legalEntityIdPk

**Pagination Supported:** Yes

**Filter:** ACTIVE / INACTIVE / ALL

### findLeAddressByLegalEntityIdPkBase

**Description:** find all records in LE\_ADDRESS\_ASSOC table based on legal entity idPk

**Transaction Service Name:** findLeAddressByLegalEntityIdPkBase

**Request DO:** leAddressAssocDO

**Response DO:** leAddressAssocDOList

**Mandatory Input:** legalentityIdpk

**Pagination Supported:** Yes

**Filter:** ACTIVE / INACTIVE / ALL

### findLeIdentifierKycRegistryByLegalEntityIdPkBase

**Description:** find all records in LE\_IDENTIFIER\_KYC\_REGISTRY table based on legal entity idPk

**Transaction Service Name:** findLeIdentifierKycRegistryByLegalEntityIdPkBase

**Request DO:** leIdentifierKycRegistryDO

**Response DO:** leIdentifierKycRegistryDOList

**Mandatory Input:** legalentityIdpk

**Pagination Supported:** Yes

**Filter:** ACTIVE / INACTIVE / ALL

### findLePhoneAssocByLegalEntityIdPkBase

**Description:** find all records in LE\_PHONE\_ASSOC table based on legal entity idPk

**Transaction Service Name:** findLePhoneAssocByLegalEntityIdPkBase

**Request DO:** lePhoneAssocDO

**Response DO:** lePhoneAssocDOList

**Mandatory Input:** idPk

**Pagination Supported:** Yes

**Filter:** ACTIVE / INACTIVE / ALL

### findLePreferencesByLegalEntityIdPkBase

**Description:** find all records from LE\_PREFERENCES table based on legal entity idPk

**Transaction Service Name:** findLePreferencesByLegalEntityIdPkBase

**Request DO:** lePreferencesDO

**Response DO:** lePreferencesDOList

**Mandatory Input:** legalEntityIdPk

**Pagination Supported:** Yes

**Filter:** ACTIVE / INACTIVE / ALL

### findLePropertyAssocByLegalEntityIdPkBase

**Description:** find all records in LE\_PROPERTY\_ASSOC table based legal entity idPk

**Transaction Service Name:** indLePropertyAssocByLegalEntityIdPkBase

**Request DO:** lePropertyAssocDO

**Response DO:** lePropertyAssocDOList

**Mandatory Input:** legalEntityIdPk

**Pagination Supported:** Yes

**Filter:** ACTIVE / INACTIVE / ALL

### findLeSystemKeysRegistryByLegalEntityIdPkBase

**Description: :** find all records in LE\_SYSTEM\_KEYS\_REGISTRY table based legal entity idPk

**Transaction Service Name:** findLeSystemKeysRegistryByLegalEntityIdPkBase

**Request DO:** leSystemKeysRegistryDO

**Response DO:** leSystemKeysRegistryDOList

**Mandatory Input:** legalEntityIdPk

**Pagination Supported:** Yes

**Filter:** ACTIVE / INACTIVE / ALL

### findLeToLeRelationshipByLegalentityIdpkBase

**Description: :** find all records in LE\_TO\_LE\_RELATIONSHIP table based legal entity idPk. The legal entity needs to be provided in reqLeToLeRelationshipDO.fromLegalentityIdpk attribute. However the search happens on both the attributes fromLegalentityIdpk or toLegalentityIdpk in database.

**Transaction Service Name:** findLeToLeRelationshipByLegalentityIdpkBase

**Request DO:** leToLeRelationship

**Response DO:** leToLeRelationshipList

**Mandatory Input:** fromLegalentityIdpk

**Pagination Supported:** Yes

**Filter:** ACTIVE / INACTIVE / ALL

### findLeVehicleAssocByLegalEntityIdPkBase

**Description:** find all records in LE\_VEHICLE\_ASSOC table based legal entity idPk

**Transaction Service Name:** findLeVehicleAssocByLegalEntityIdPkBase

**Request DO:** leVehicleAssocDO

**Response DO:** leVehicleAssocDOList

**Mandatory Input:** legalEntityIdPk

**Pagination Supported:** Yes

**Filter:** ACTIVE / INACTIVE / ALL

### findPersonnamesByLegalEntityIdBase

**Description:** find all records in PERSONNAMES table based legal entity idPk

**Transaction Service Name:** findPersonnamesByLegalEntityIdBase

**Request DO:** personnamesDO

**Response DO:** personnamesDOList

**Mandatory Input:** legalEntityIdPk

**Pagination Supported:** Yes

**Filter:** ACTIVE / INACTIVE / ALL

### retrieveAccountAddressAssocBase

**Description:** retrieve a record from ACCOUNT\_ADDRESS\_ASSOC table based idPk

**Transaction Service Name:** retrieveAccountAddressAssocBase

**Request DO:** accountAddressAssocDO

**Response DO:** accountAddressAssocDO

**Mandatory Input:** Idpk

### retrieveAccountBase

**Description:** *retrieve* a record in ACCOUNT table *based on idPk*

**Transaction Service Name:** *retrieve*AccountBase

**Request DO:** *accountDO*

**Response DO:** *accountDO*

**Mandatory Input:** *idPk*

### retrieveAccountPhoneAssocBase

**Description**: create a record in ACCOUNT\_PHONE\_ASSOC table based on idPk

**Transaction Service Name:** retrieveAccountPhoneAssocBase

**Request DO:** AccountPhoneAssocDO

**Response DO:** AccountPhoneAssocDO

**Mandatory Input:** idPk

### retrieveAddressBase

**Description:** retrieve a record from ADDRESS table based on idPk

**Transaction Service Name:** retrieveAddressBase

**Request DO:** AddressDO

**Response DO:** AddressDO

**Mandatory Input:** idPk

### retrieveCorporationnamesBase

**Description:** retrieve a record in CORPORATIONNAMES table based on idPk

**Transaction Service Name:** retrieveCorporationnamesBase

**Request DO:** corporationnamesDO

**Response DO:** corporationnamesDO

**Mandatory Input:** idPk

### retrieveEntityGroupAssocBase

**Description:**  retrieve a record in ENTITY\_GROUP\_ASSOC table based on idPk

**Transaction Service Name:** retrieveEntityGroupAssocBase

**Request DO:** entityGroupAssocDO

**Response DO:** entityGroupAssocDO

**Mandatory Input:** idPk

### retrieveEntityGroupBase

**Description:** retrieve a record in ENTITY\_GROUP table based on idPk

**Transaction Service Name:** retrieveEntityGroupBase

**Request DO:** entityGroupDO

**Response DO:** entityGroupDO

**Mandatory Input:** idPk

### retrieveLeAccountAssocBase

**Description:** retrieve a record from LE\_ACCOUNT\_ASSOC table based on idPk

**Transaction Service Name:** retrieveLeAccountAssocBase

**Request DO:** leAccountAssocDO

**Response DO:** leAccountAssocDO

**Mandatory Input:** idPk

### retrieveLeAddressAssocBase

**Description:** retrieve a record from LE\_ADDRESS\_ASSOC table based on idPk

**Transaction Service Name:** retrieveLeAddressAssocBase

**Request DO:** leAddressAssocDO

**Response DO:** leAddressAssocDO

**Mandatory Input:** idPk

### retrieveLeCorporationBase

**Description:** retrieve a record from LE\_CORPORATION table based on idPk

**Transaction Service Name:** retrieveLeCorporationBase

**Request DO:** LeCorporationDO

**Response DO:** LeCorporationDO

**Mandatory Input:** idPk

### retrieveLeIdentifierKycRegistryBase

**Description:** retrieve a record from LE\_IDENTIFIER\_KYC\_REGISTRY table based on idPk

**Transaction Service Name:** retrieveLeIdentifierKycRegistryBase

**Request DO:** leIdentifierKycRegistryDO

**Response DO:** leIdentifierKycRegistryDO

**Mandatory Input:** idPk

### retrieveLePersonBase

**Description:** retrieve a record from LE\_PERSON table based on idPk

**Transaction Service Name:** retrieveLePersonBase

**Request DO:** lePersonDO

**Response DO:** lePersonDO

**Mandatory Input:** idPk

### retrieveLePhoneAssocBase

**Description:** retrieve a record in LE\_PHONE\_ASSOC table based on idPk

**Transaction Service Name:** retrieveLePhoneAssocBase

**Request DO:** lePhoneAssocDO

**Response DO:** lePhoneAssocDO

**Mandatory Input:** idPk

### retrieveLePreferencesBase

**Description:** retrieve a record in LE\_PREFERENCES table based on idPk

**Transaction Service Name:** retrieveLePreferencesBase

**Request DO:** lePreferencesDO

**Response DO:** lePreferencesDO

**Mandatory Input:** idPk

### retrieveLePropertyAssocBase

**Description:** retrieve a record from LE\_PROPERTY\_ASSOC table based idPk

**Transaction Service Name:** retrieveLePropertyAssocBase

**Request DO:** lePropertyAssocDO

**Response DO:** lePropertyAssocDO

**Mandatory Input:** idPk

### retrieveLeSystemKeysRegistryBase

**Description:** retrieve a record from LE\_SYSTEM\_KEYS\_REGISTRY table based on idPk

**Transaction Service Name:** retrieveLeSystemKeysRegistryBase

**Request DO:** leSystemKeysRegistryDO

**Response DO:** leSystemKeysRegistryDO

**Mandatory Input:** idPk

### retrieveLeToLeRelationshipBase

**Description:** retrieve a record from LE\_TO\_LE\_RELATIONSHIP table based on idPk

**Transaction Service Name:** retrieveLeToLeRelationshipBase

**Request DO:** leToLeRelationship

**Response DO:** leToLeRelationship

**Mandatory Input:** idPk

### retrieveLeVehicleAssocBase

**Description:** retrieve a record from LE\_VEHICLE\_ASSOC table based on idPk

**Transaction Service Name:** retrieveLeVehicleAssocBase

**Request DO:** leVehicleAssocDO

**Response DO:** leVehicleAssocDO

**Mandatory Input:** idPk

### retrieveLegalentityBase

**Description:** retrieve a record from LEGALENTITY table based on idPk

**Transaction Service Name:** retrieveLegalentityBase

**Request DO:** legalentityDO

**Response DO:** legalentityDO

**Mandatory Input:** idPk

### retrieveMiscellaneousInfoBase

**Description:** retrieve a record from MISCELLANEOUS\_INFO table based on idPk

**Transaction Service Name:** retrieveMiscellaneousInfoBase

**Request DO:** miscellaneousInfoDO

**Response DO:** miscellaneousInfoDO

**Mandatory Input:** idPk

### retrievePersonnamesBase

**Description:** retrieve a record from PERSONNAMES table based on idPk

**Transaction Service Name:** retrievePersonnamesBase

**Request DO:** personnamesDO

**Response DO:** personnamesDO

**Mandatory Input:** idPk

### retrievePhoneStandardizedBase

**Description:** retrieve a record from PHONE\_STANDARDIZED table based on idPk

**Transaction Service Name:** retrievePhoneStandardizedBase

**Request DO:** phoneStandardizedDO

**Response DO:** phoneStandardizedDO

**Mandatory Input:** idPk

### retrievePropertyBase

**Description:** retrieve a record from PROPERTY table based on idPk

**Transaction Service Name:** retrievePropertyBase

**Request DO:** propertyDO

**Response DO:** propertyDO

**Mandatory Input:** idPk

### retrieveVehicleBase

**Description:** retrieve a record fromVEHICLE table based on idPk

**Transaction Service Name:** retrieveVehicleBase

**Request DO:** vehicleDO

**Response DO:** vehicleDO

**Mandatory Input:** idPk

### updateAccountAddressAssocBase

**Description**: update a record from ACCOUNT\_ADDRESS\_ASSOC table based idPk

**Transaction Service Name:** updateAccountAddressAssocBase

**Request DO:** accountAddressAssocDO

**Response DO:** accountAddressAssocDO

**Mandatory Input:** Idpk, version

### updateAccountBase

**Description:** *update* a record in ACCOUNT table *based on idPk*

**Transaction Service Name:** *update*AccountBase

**Request DO:** *accountDO*

**Response DO:** *accountDO*

**Mandatory Input:** *idPk, version*

### updateAccountPhoneAssocBase

**Description**: update a record in ACCOUNT\_PHONE\_ASSOC table based on idPk

**Transaction Service Name:** updateAccountPhoneAssocBase

**Request DO:** AccountPhoneAssocDO

**Response DO:** AccountPhoneAssocDO

**Mandatory Input:** idPk, version

### updateAddressBase

**Description:** update a record from ADDRESS table based on idPk

**Transaction Service Name:** updateAddressBase

**Request DO:** AddressDO

**Response DO:** AddressDO

**Mandatory Input:** idPk, version

### updateCorporationnamesBase

**Description:** update a record in CORPORATIONNAMES table based on idPk

**Transaction Service Name:** updateCorporationnamesBase

**Request DO:** corporationnamesDO

**Response DO:** corporationnamesDO

**Mandatory Input:** idPk, version

### updateEntityGroupAssocBase

**Description:**  update a record in ENTITY\_GROUP\_ASSOC table based on idPk

**Transaction Service Name:** updateEntityGroupAssocBase

**Request DO:** entityGroupAssocDO

**Response DO:** entityGroupAssocDO

**Mandatory Input:** idPk, version

### updateEntityGroupBase

**Description:** update a record in ENTITY\_GROUP table based on idPk

**Transaction Service Name:** updateEntityGroupBase

**Request DO:** entityGroupDO

**Response DO:** entityGroupDO

**Mandatory Input:** idPk, version

### updateLeAccountAssocBase

**Description:** update a record from LE\_ACCOUNT\_ASSOC table based on idPk

**Transaction Service Name:** updateLeAccountAssocBase

**Request DO:** leAccountAssocDO

**Response DO:** leAccountAssocDO

**Mandatory Input:** idPk, version

### updateLeAddressAssocBase

**Description:** update a record from LE\_ADDRESS\_ASSOC table based on idPk

**Transaction Service Name:** updateLeAddressAssocBase

**Request DO:** leAddressAssocDO

**Response DO:** leAddressAssocDO

**Mandatory Input:** idPk, version

### updateLeCorporationBase

**Description:** update a record in LE\_CORPORATION table based on idPk

**Transaction Service Name:** updateLeCorporationBase

**Request DO:** LeCorporationDO

**Response DO:** LeCorporationDO

**Mandatory Input:** idPk, version,legalentityIdpk

### updateLeIdentifierKycRegistryBase

**Description:** update a record from LE\_IDENTIFIER\_KYC\_REGISTRY table based on idPk

**Transaction Service Name:** updateLeIdentifierKycRegistryBase

**Request DO:** leIdentifierKycRegistryDO

**Response DO:** leIdentifierKycRegistryDO

**Mandatory Input:** idPk, version

### updateLePersonBase

**Description:** update a record from LE\_PERSON table based on idPk

**Transaction Service Name:** updateLePersonBase

**Request DO:** lePersonDO

**Response DO:** lePersonDO

**Mandatory Input:** idPk, version, legalentityIdpk

### updateLePhoneAssocBase

**Description:** update a record in LE\_PHONE\_ASSOC table based on idPk

**Transaction Service Name:** updateLePhoneAssocBase

**Request DO:** lePhoneAssocDO

**Response DO:** lePhoneAssocDO

**Mandatory Input:** idPk, version

### updateLePreferencesBase

**Description:** update a record in LE\_PREFERENCES table based on idPk

**Transaction Service Name:** updateLePreferencesBase

**Request DO:** lePreferencesDO

**Response DO:** lePreferencesDO

**Mandatory Input:** idPk, version

### updateLePropertyAssocBase

**Description:** update a record from LE\_PROPERTY\_ASSOC table based idPk

**Transaction Service Name:** updateLePropertyAssocBase

**Request DO:** lePropertyAssocDO

**Response DO:** lePropertyAssocDO

**Mandatory Input:** idPk, version

### updateLeSystemKeysRegistryBase

**Description:** update a record in LE\_SYSTEM\_KEYS\_REGISTRY table based on idPk

**Transaction Service Name:** updateLeSystemKeysRegistryBase

**Request DO:** leSystemKeysRegistryDO

**Response DO:** leSystemKeysRegistryDO

**Mandatory Input:** idPk, version

### updateLeToLeRelationshipBase

**Description:** update a record in LE\_TO\_LE\_RELATIONSHIP table based on idPk

**Transaction Service Name:** updateLeToLeRelationshipBase

**Request DO:** leToLeRelationship

**Response DO:** leToLeRelationship

**Mandatory Input:** idPk, version

### updateLeVehicleAssocBase

**Description:** update a record in LE\_VEHICLE\_ASSOC table based on idPk

**Transaction Service Name:** updateLeVehicleAssocBase

**Request DO:** leVehicleAssocDO

**Response DO:** leVehicleAssocDO

**Mandatory Input:** idPk

### updateLegalentityBase

**Description:** update a record in LEGALENTITY table based on idPk

**Transaction Service Name:** updateLegalentityBase

**Request DO:** legalentityDO

**Response DO:** legalentityDO

**Mandatory Input:** idPk, version

### updateMiscellaneousInfoBase

**Description:** update a record in MISCELLANEOUS\_INFO table based on idPk

**Transaction Service Name:** updateMiscellaneousInfoBase

**Request DO:** miscellaneousInfoDO

**Response DO:** miscellaneousInfoDO

**Mandatory Input:** idPk, version

### updatePersonnamesBase

**Description:** update a record in PERSONNAMES table based on idPk

**Transaction Service Name:** updatePersonnamesBase

**Request DO:** personnamesDO

**Response DO:** personnamesDO

**Mandatory Input:** idPk, version

### updatePhoneStandardizedBase

**Description:** update a record in PHONE\_STANDARDIZED table based on idPk

**Transaction Service Name:** updatePhoneStandardizedBase

**Request DO:** phoneStandardizedDO

**Response DO:** phoneStandardizedDO

**Mandatory Input:** idPk, version

### updatePropertyBase

**Description:** update a record in PROPERTY table based on idPk

**Transaction Service Name:** updatePropertyBase

**Request DO:** propertyDO

**Response DO:** propertyDO

**Mandatory Input:** idPk, version

### updateVehicleBase

**Description:** update a record inVEHICLE table based on idPk

**Transaction Service Name:** updateVehicleBase

**Request DO:** vehicleDO

**Response DO:** vehicleDO

**Mandatory Input:** idPk, version

## Data Table Management Services (Composite)

### searchAuthRoles

**Description:** This service returns the roles based on username and/or role name. This transaction search the database as per below condition

1. If username and userRole are present in the request then this transaction will return the result only if below conditions are satisfied
   * The given user must be present in AUTH\_USER\_REGISTRY table.
   * The given role must be present in AUTH\_ROLES\_REGISTRY table.
   * The user must be associated with the given role in AUTH\_USER\_ROLE\_ASSOC table.
   * The result will have the rows from AUTH\_ROLES\_REGISTRY table for given user name and role name combination.
2. If userRole is present and username is null in the request then this transaction will return the result only if below conditions are satisfied
   * The given role must be present in AUTH\_ROLES\_REGISTRY table.
   * The result will have the rows from AUTH\_ROLES\_REGISTRY table for given role.
3. If username is present and userRole is null in the request then this transaction will return the result only if below conditions are satisfied
   * The given user must be present in AUTH\_USER\_REGISTRY table.
   * The result will have the rows from AUTH\_ROLES\_REGISTRY table to which the user is associated in AUTH\_USER\_ROLE\_ASSOC table. i.e. all the roles to which a user is associated with will be returned.

**Transaction Service Name:** searchAuthRoles

**Request DO:** searchAuthAccessControlDO

**Response DO:** authRolesRegistryDOList (empty list if no records found)

**Mandatory Input:** rolename or username

**Pagination Supported:** Yes

**Filter:** ACTIVE / INACTIVE / ALL

**Wildcard Support:** Yes. % can be used for wildcard searches

**Use Case –**

1. To check if given userName and roleName combination is valid
2. To check if roleName is valid
3. To get all the roles associated with a valid user

**Special Notes –** This service is used to in authorization framework.

### searchAuthUsers

**Description:** This service returns the roles based on username and/or role name. This transaction search the database as per below condition

1. If username and userRole are present in the request then this transaction will return the result only if below conditions are satisfied
   * The given user must be present in AUTH\_USER\_REGISTRY table.
   * The given role must be present in AUTH\_ROLES\_REGISTRY table.
   * The user must be associated with the given role in AUTH\_USER\_ROLE\_ASSOC table.
   * The result will have the rows from AUTH\_USER\_REGISTRY table for given user name and role name combination.
2. If userRole is present and username is null in the request then this transaction will return the result only if below conditions are satisfied
   * The given role must be present in AUTH\_ROLES\_REGISTRY table.
   * The result will have the rows from AUTH\_USER\_REGISTRY table for given role.
3. If username is present and userRole is null in the request then this transaction will return the result only if below conditions are satisfied
   * The given user must be present in AUTH\_USER\_REGISTRY table.
   * The result will have the rows from AUTH\_USER\_REGISTRY table to which the user is associated in AUTH\_USER\_ROLE\_ASSOC table. i.e. all the users to which a role is associated with will be returned.

**Transaction Service Name:** searchAuthRoles

**Request DO:** searchAuthAccessControlDO

**Response DO:** authUserRegistryDOList (empty list if no records found)

**Mandatory Input:** rolename or username

**Pagination Supported:** Yes

**Filter:** ACTIVE / INACTIVE / ALL

**Wildcard Support:** Yes. % can be used for wildcard searches

**Use Case –**

1. To check if given userName and roleName combination is valid
2. To check if userName is valid
3. To get all the users associated with a valid role

**Special Notes –** N/A

### searchAuthAccessControl

**Description:** This service is used to get the transaction service name based on userName or roleName

The response is sent back based on below rule

1. If username and userRole are present in the request then authorization if all of the below conditions are satisfied
   * The given user must be present in AUTH\_USER\_REGISTRY table.
   * The given role must be present in AUTH\_ROLES\_REGISTRY table.
   * The user must be associated with the given role in AUTH\_USER\_ROLE\_ASSOC table
   * The response will be sent back having all the rows if txnserviceName have entry in Access control table for given role i.e. AUTH\_USERROLE\_ACCESSCONTROL have entry for given role and transaction service name.
2. If userRole is present and username is null in the request then role is authorized to perform the given transaction if all the below conditions are satisfied
   * The given role must be present in AUTH\_ROLES\_REGISTRY table.
   * The response will be sent back having all the rows if txnserviceName have entry in Access control table for given role i.e. AUTH\_USERROLE\_ACCESSCONTROL have entry for given role and transaction service name.
3. If username is present and userRole is null in the request then user is authorized to perform the given transaction if all the below conditions are satisfied
   * The given user must be present in AUTH\_USER\_REGISTRY table.
   * The response will be sent back having all the rows if txnserviceName have entry in Access control table for given user i.e. AUTH\_USERROLE\_ACCESSCONTROL have entry for given user and transaction service name.

**Transaction Service Name:** searchAuthAccessControl

**Request DO:** searchAuthAccessControlDO

**Response DO:** authUserRegistryDOList (empty list if no records found)

**Mandatory Input:** txnserviceName and either rolename or userName

**Pagination Supported:** Yes

**Filter:** ACTIVE / INACTIVE / ALL

**Wildcard Support:** NO

**Use Case –** To check if given user or role is authorized to invoke a given transaction.

**Special Notes –** This is the primary service used by MDM Hub to perform authorication of the request based on userName/roleName and txnserviceName

### createAccount

**Description:** create an account with its address and phone associations

**Transaction Service Name:** createAccount

**Request DO:** accountDO

**Response DO:** accountDO

**Mandatory Input:** accountDO

**Use Case –** create a standalone account along with its address and phone associations.

**Special Notes –** accountDO.accountAddressAssocDOList and accountDO.acountPhoneAssocDOList will be processed if present.

### createAccountAddressAssociation

**Description:** Associate existing account with existing or new address. If address Id in the request (txnPayload.accountAddressAssocDO.addressIdpk) is null then new address will be created in ADDRESS table and will be associated with account. If address Id (txnPayload.accountAddressAssocDO.addressIdpk) is present in the request then account and address association will be created.

**Transaction Service Name:** createAccountAddressAssociation

**Request DO:** accountAddressAssocDO

**Response DO:** accountAddressAssocDO

**Mandatory Input:** accountAddressAssocDO, accountAddressAssocDO .accountIdpk

**Use Case –**

1. Associate existing account with existing address
2. Associate existing account with new address

**Special Notes –** N/A

### createAccountPhoneAssociation

**Description:** associate a phone number with existing account

**Transaction Service Name:** createAccountPhoneAssociation

**Request DO:** accountPhoneAssocDO

**Response DO:** accountPhoneAssocDO

**Mandatory Input:** accountPhoneAssocDO.accountIdPk

**Use Case –** associate a phone number with existing account

**Special Notes –** The standardized phone number phoneStandardizedDO can also be through this service. If existing idPk of phoneStandardizedDO is provided in the accountPhoneAssocDO. phoneStandardizedIdpk then the same will be used.

### createEntityGroup

**Description:** create an entry in ENTITY\_GROUP and ENTITY\_GROUP\_ASSOC table

**Transaction Service Name:** createEntityGroup

**Request DO:** entityGroupDO

**Response DO:** entityGroupDO

**Mandatory Input:** As per createEntityGroupBase and createEntityGroupAssocbase service

**Use Case –** create grouping of entities

**Special Notes –** N/A

### createLeAccountAssociation

**Description:** This transaction associates an account to existing legal entity. If Account is already present then a new association is added between LE and account. It’s also possible to create a new account through this service.

**Transaction Service Name:** createLeAccountAssociation

**Request DO:** leAccountAssocDO

**Response DO:** leAccountAssocDO

**Mandatory Input:** leAccountAssocDO.legalEntityIdPk, txnPayload.leAccountAssocDO.accountIdpk or txnPayload.leAccountAssocDO.accountDO

**Use Case –**

1. create legal entity to account association for existing LE and existing account
2. create a new account and associate it with legal entity

**Special Notes –** N/A

### createLeAddressAssociation

**Description:** create an association between le and address. Create association based on AddressDO or AddressIdPk of existing address

**Transaction Service Name:** createLeAddressAssociation

**Request DO:** leAddressAssocDO

**Response DO:** leAddressAssocDO

**Mandatory Input:** leAddressAssocDO.legalEntityIdPk, leAddressAssocDO.addressIdpk or leAddressAssocDO .addressDO

**Use Case –**

1. create legal entity to address association for existing LE and existing address
2. create a new address and associate it with legal entity

**Special Notes –** N/A

### createLeCorporation

**Description:** create a legal entity of type Corporation along with corporation names.

**Transaction Service Name:** createLeCorporation

**Request DO:** leCorporationDO

**Response DO:** leCorporationDO

**Mandatory Input:** leCorporationDO.legalentityIdpk

**Use Case –** N/A

**Special Notes –** This method does not create a record in LEGALENTITY table so this service should be used only if LEGALENTITY is already created through different process and there is a need to create records in LE\_CORPORATION and CORPORATIONNAMES. Its recommended to use createLegalEntity service to create a legal entity.

### createLePerson

**Description:** create a legal entity of type Person along with person names.

**Transaction Service Name:** createLePerson

**Request DO:** lePersonDO

**Response DO:** lePersonDO

**Mandatory Input:** lePersonDO.legalentityIdpk

**Use Case –** N/A

**Special Notes –** This method does not create a record in LEGALENTITY table so this service should be used only if LEGALENTITY is already created through different process and there is a need to create records in LE\_PERSON and PERSONNAMES. It’s recommended to use createLegalEntity service to create a legal entity.

### createLePhoneAssociation

**Description:** create an association between legal entity and phone. It’s also possible to create a standardized phone number record through this service.

**Transaction Service Name:** createLePhoneAssociation

**Request DO:** lePhoneAssocDO

**Response DO:** lePhoneAssocDO

**Mandatory Input:** lePhoneAssocDO.legalEntityIdPk

**Use Case –**

1. Associate legal entity with phone number
2. Associate a legal entity with phone number and also store the phone number in standardized format.

**Special Notes –** N/A

### createLePropertyAssociation

**Description:** create an association between existing legal entity and existing property. It’s also possible to create a new property record and associate it with legal entity.

**Transaction Service Name:** createLePropertyAssociation

**Request DO:** lePropertyAssocDO

**Response DO:** lePropertyAssocDO

**Mandatory Input:** lePropertyAssocDO.legalEntityIdPk, lePropertyAssocDO.propertyIdpk or lePropertyAssocDO. propertyDO

**Use Case –**

1. create legal entity to property association for existing LE and existing property
2. create a new property and associate it with legal entity

**Special Notes –** N/A

### createLeVehicleAssociation

**Description:** create an association between existing legal entity and existing Vehicle. It’s also possible to create a new vehicle record and associate it with legal entity.

**Transaction Service Name:** createLeVehicleAssociation

**Request DO:** leVehicleAssocDO

**Response DO:** leVehicleAssocDO

**Mandatory Input:** leVehicleAssocDO.legalEntityIdPk, leVehicleAssocDO.vehicleDO or leVehicleAssocDO.vehicleIdPk

**Use Case –**

1. create legal entity to property association for existing LE and existing property
2. create a new property and associate it with legal entity

**Special Notes –** N/A

### createLegalEntity

**Description:** create a legal entity along with person or corporation, system registry keys, preferences, identifiers, address, phone, account, property, vehicle and miscellaneous info

**Transaction Service Name:** createLegalEntity

**Request DO:** legalentityDO

**Response DO:** legalentityDO

**Mandatory Input:** entityObjectTypeRefkey, lePersonDO or leCorporationDO

**Use Case –**

1. create new legal entity along with all its objects like person or corporation, system registry keys, preferences, identifiers, address, phone, account, property, vehicle and miscellaneous info

**Special Notes –** N/A

### findAllAccountAddressAssocByAccountId

**Description:** This service returns all the address associated with an account.

**Transaction Service Name:** findAllAccountAddressAssocByAccountId

**Request DO:** accountAddressAssocDO

**Response DO:** accountAddressAssocDOList

**Mandatory Input:**  accountIdPk

**Pagination Supported:** YES

**Filter:** ACTIVE/INACTIVE/ALL

**Wildcard Support:** N/A

**Use Case –** get all the addresses associated with an account based on account idPk

**Special Notes –** This service does not return any account details.

### findAllAccountPhoneAssocByAccountId

**Description:** This service returns all the phone numbers associated with an account.

**Transaction Service Name:** findAllAccountPhoneAssocByAccountId

**Request DO:** accountPhoneAssocDO

**Response DO:** accountPhoneAssocDOList

**Mandatory Input:** accountIdPk

**Pagination Supported:** YES

**Filter:** ACTIVE/INACTIVE/ALL

**Wildcard Support:** N/A

**Use Case –** get all the phone numbers associated with an account based on account idPk

**Special Notes –** This service does not return any account details.

### findAllLeAccountAssocByLegalEntityId

**Description:** This service returns all the accounts associated with legal entity

**Transaction Service Name:** findAllLeAccountAssocByLegalEntityId

**Request DO:** LeAccountAssocDO

**Response DO:** LeAccountAssocDOList

**Mandatory Input:** leAccountAssocDO. legalEntityIdPk

**Pagination Supported:** YES

**Filter:** ACTIVE/INACTIVE/ALL

**Wildcard Support:** N/A

**Use Case –**  get all the accounts associated with a legal entity.

**Special Notes –** Supports account level inquiry levels.Set the leAccountAssocDO.accountInquiryLevel to retrieve as per inquiry level else the default inquiry level as defined in configuration properties would be used. Refer ‘Default Inquiry levels for Legal entity and Account – Default configuration’ section to know more about inquiry levels available for account level. This service does not return any Legal entity details.

### findAllLeAddressAssocByLegalEntityId

**Description:** This service returns all the addresses associated with a legal entity.

**Transaction Service Name:** findAllLeAddressAssocByLegalEntityId

**Request DO:** leAddressAssocDO

**Response DO:** leAddressAssocDOList

**Mandatory Input:** leAddressAssocDO.legalEntityIdPk

**Pagination Supported:** YES

**Filter:** ACTIVE/INACTIVE/ALL

**Wildcard Support:** N/A

**Use Case –** get all the addresses associated with a legal entity.

**Special Notes –** This service does not return legal entity details.

### findAllLePhoneAssocByLegalEntityId

**Description:** This service returns all the phone numbers associated with a legal entity.

**Transaction Service Name:** findAllLePhoneAssocByLegalEntityId

**Request DO:** lePhoneAssocDO

**Response DO:** lePhoneAssocDOList

**Mandatory Input:** lePhoneAssocDO.legalEntityIdPk

**Pagination Supported:** YES

**Filter:** ACTIVE/INACTIVE/ALL

**Wildcard Support:** N/A

**Use Case –** get all the phone numbers associated with a legal entity.

**Special Notes –** This service does not return legal entity details.

### findAllLePropertyAssocByLegalEntityId

**Description:** This service returns all the property details associated with a legal entity.

**Transaction Service Name:** findAllLePropertyAssocByLegalEntityId

**Request DO:** lePropertyAssocDO

**Response DO:** lePropertyAssocDO

**Mandatory Input:** lePropertyAssocDO. legalentityIdpk

**Pagination Supported:** YES

**Filter:** ACTIVE/INACTIVE/ALL

**Wildcard Support:** N/A

**Use Case –** get all the property details associated with a legal entity.

**Special Notes –** This service does not return legal entity details.

### findAllLeVehicleAssocByLegalEntityId

**Description:** This service returns all the vehicles associated with a legal entity. This service retrieves the legal entity form the database based on input legalentityIdpk and invokes matching rule as defined in the configuration properties.

**Transaction Service Name:** findAllLeVehicleAssocByLegalEntityId

**Request DO:** leVehicleAssocDO

**Response DO:** leVehicleAssocDO

**Mandatory Input:** leVehicleAssocDO. legalentityIdpk

**Pagination Supported:** YES

**Filter:** ACTIVE/INACTIVE/ALL

**Wildcard Support:** N/A

**Use Case –** get all the vehicles associated with a legal entity.

**Special Notes –** This service does not return legal entity details.

### performLeMatch

**Description:** This service is used to perform the match of given legal entity with the LEs in database.

**Transaction Service Name:** performLeMatch

**Request DO:** performLeMatchRequestDO

**Response DO:** performLeMatchRequestDO

**Mandatory Input:**

* performLeMatchRequestDO.legalentityDO.idpk
* performLeMatchRequestDO.matchEngineType

**Use Case –** perform the match of given legal entity with the LE present in database.

**Special Notes –** This service is used by MDM Hub for performing matching. The users who want to customize the logic they must create a new service. The code of this service must not be changed else the default matching may not work properly.

### retrieveAccountAddressByAccountAddressAssocId

**Description:** retrieves account to address association based on accountAddressAssocDO.idPk

**Transaction Service Name:** retrieveAccountAddressByAccountAddressAssocId

**Request DO:** accountAddressAssocDO

**Response DO:** accountAddressAssocDO

**Mandatory Input:** accountAddressAssocDO.idPk

**Pagination Supported:** N/A

**Filter:** N/A

**Wildcard Support:** N/A

**Use Case –** retrieve the address of the account based on accountAddressAssocDO.idPk

**Special Notes –** N/A

### retrieveAccountByAccountId

**Description:** This service retrieves account details along with Addresses, phones and le-account associations.

**Transaction Service Name:** retrieveAccountByAccountId

**Request DO:** accountDO

**Response DO:** accountDO

**Mandatory Input:** accountDO.IdPk

**Pagination Supported:** N/A

**Filter:** N/A

**Wildcard Support:** N/A

**Use Case –**

**Special Notes –** This transaction supports inquiry levels, set the accountDO.inquiryLevel to a valid value as mentioned in ‘Default Inquiry levels for Legal entity and Account – Default configuration’ section.

### retrieveAccountPhoneByAccountPhoneAssocId

**Description:** retrieves an account to phone association based on accountPhoneAssocDO.idPk

**Transaction Service Name:** retrieveAccountPhoneByAccountPhoneAssocId

**Request DO:** accountPhoneAssocDO

**Response DO:** accountPhoneAssocDO

**Mandatory Input:** accountPhoneAssocDO.idPk

**Use Case –** retrieve the phone numbers of the account based on accountAddressAssocDO.idPk

**Special Notes –** N/A

### retrieveEntityGroupByEntityGroupId

**Description:** This service retrieves the entire grouping information containing the associated entities. The data from ENTITY\_GROUP along with all associated records from ENTITY\_GROUP\_ASSOC table are returned in this service.

**Transaction Service Name:** retrieveEntityGroupByEntityGroupId

**Request DO:** entityGroupDO

**Response DO:** entityGroupDO

**Mandatory Input:** entityGroupDO.idPk

**Pagination Supported:** YES

**Filter:** ACTIVE/INACTIVE/ALL

**Wildcard Support:** N/A

**Use Case –** retrieve the entire grouping information containing the associated entities.

**Special Notes –** N/A

### retrieveLeAccountByLeAccountAssocId

**Description:** This service retrieves account details based on LeAccountAssocId

**Transaction Service Name:** retrieveLeAccountByLeAccountAssocId

**Request DO:** leAccountAssocDO

**Response DO:** leAccountAssocDO

**Mandatory Input:** leAccountAssocDO.idPk

**Pagination Supported:** N/A

**Filter:** N/A

**Wildcard Support:** N/A

**Use Case –** retrieve account details based on LeAccountAssocId

**Special Notes –** This service supports inquiry level for retrieving account details. Provide valid inquiry level in leAccountAssocDO .accountInquiryLevel attribute as mentioned in ‘Default Inquiry levels for Legal entity and Account – Default configuration’ section.

### retrieveLeAddressByLeAddressAssocId

**Description:** retrieves legal entity to address association based on leAddressAssocDO.idPk

**Transaction Service Name:** retrieveLeAddressByLeAddressAssocId

**Request DO:** LeAddressAssocDO

**Response DO:** LeAddressAssocDO

**Mandatory Input:** leAddressAssocDO.idPk

**Use Case –** retrieves legal entity to address association based on leAddressAssocDO.idPk

**Special Notes –** N/A

### retrieveLeCorporationByLegalEntityId

**Description:** retrieve legal entity of corporation type along with its names

**Transaction Service Name:** retrieveLeCorporationByLegalEntityId

**Request DO:** leCorporationDO

**Response DO:** leCorporationDO

**Mandatory Input:** leCorporationDO.legalentityIdpk

**Pagination Supported:** YES (pagination supported to retrieve names)

**Filter:** N/A

**Wildcard Support:** N/A

**Use Case –** retrieve legal entity of corporation type along with its names

**Special Notes –** This service does not retrieve a record from LEGALENTITY table so this service should be used only if LEGALENTITY data is needed. It’s recommended to use retrieveLegalEntityByLegalEntityId service to retrieve a legal entity.

### retrieveLePersonByLegalEntityId

**Description:** retrieve legal entity of person type along with its names

**Transaction Service Name:** retrieveLePersonByLegalEntityId

**Request DO:** lePersonDO

**Response DO:** lePersonDO

**Mandatory Input:** lePersonDO. legalentityIdpk

**Pagination Supported:** YES (pagination supported to retrieve names)

**Filter:** N/A

**Wildcard Support:** N/A

**Use Case –** retrieve legal entity of person type along with its names

**Special Notes –** This service does not retrieve a record from LEGALENTITY table so this service should be used only if LEGALENTITY data is needed. It’s recommended to use retrieveLegalEntityByLegalEntityId service to retrieve a legal entity.

### retrieveLePhoneByLePhoneAssocId

**Description:** Retrieves an account to phone association based on lePhoneAssocDO.idPk

**Transaction Service Name:** retrieveLePhoneByLePhoneAssocId

**Request DO:** lePhoneAssocDO

**Response DO:** lePhoneAssocDO

**Mandatory Input:** lePhoneAssocDO.idPk

**Pagination Supported:** N/A

**Filter:** N/A

**Wildcard Support:** N/A

**Use Case –** retrieves an account to phone association based on lePhoneAssocDO.idPk

**Special Notes –** N/A

### retrieveLePropertyByLePropertyAssocId

**Description:** Retrieve the property details associated with legal entity based on LePropertyAssocId

**Transaction Service Name:** retrieveLePropertyByLePropertyAssocId

**Request DO:** lePropertyAssocDO

**Response DO:** lePropertyAssocDO

**Mandatory Input:** lePropertyAssocDO.idPk

**Pagination Supported:** N/A

**Filter:** N/A

**Wildcard Support:** N/A

**Use Case –** Retrieve le property details based on LePropertyAssocId

**Special Notes –** N/A

### retrieveLeVehicleLeVehicleAssocId

**Description:** Retrieve the vehicle details associated with legal entity based on LeVehicleAssocId

**Transaction Service Name:** retrieveLeVehicleLeVehicleAssocId

**Request DO:** leVehicleAssocDO

**Response DO:** leVehicleAssocDO

**Mandatory Input:** leVehicleAssocDO.idPk

**Pagination Supported:** N/A

**Filter:** N/A

**Wildcard Support:** N/A

**Use Case –** Retrieve the vehicle details associated with legal entity based on LeVehicleAssocId

**Special Notes –** N/A

### retrieveLegalEntityByLegalEntityId

**Description:** Retrieve legal entity and all its associated person or corporation, system registry keys, preferences, identifiers, address, phone, account, property, vehicle and miscellaneous info

**Transaction Service Name:** retrieveLegalEntityByLegalEntityId

**Request DO:** legalentityDO

**Response DO:** legalentityDO

**Mandatory Input:** legalentityDO.idpk

**Pagination Supported:** YES

**Filter:** N/A

**Wildcard Support:** N/A

**Use Case –** retrieve legal entity and all its associated person or corporation, system registry keys, preferences, identifiers, address, phone, account, property, vehicle and miscellaneous info

**Special Notes –** LE and account inquiry levels are supported for this service. Provide valid value in legalentityDO .inquiryLevel for LE inquiry and legalentityDO.accountInquiryLevel for account inquiry. Account inquiry level will only be used if LE inquiry level retrieves an account else this attribute will be ignored. If LE inquiry level is not provided then le will be retrieved based on default inquiry level as set in configuration properties.

### searchAccountByAccountAttributes

**Description:** search the account in database based on account attributes

**Transaction Service Name:** searchAccountByAccountAttributes

**Request DO:** searchAccountRequestDO

**Response DO:** accountDOList

**Mandatory Input:** at least one attribute from the below

* searchAccountRequestDO.accountidPk
* searchAccountRequestDO.sourceSystemRefkey
* searchAccountRequestDO.sourceAccountId
* searchAccountRequestDO.accountName
* searchAccountRequestDO.accountName2
* searchAccountRequestDO.accountDescription
* searchAccountRequestDO.accountSourceStatusRefkey
* searchAccountRequestDO.accountMdmStatusRefkey

**Pagination Supported:** YES.

**Filter:** ACTIVE/INACTIVE/ALL

**Wildcard Support:** YES

**Use Case –** search and account based on account attributes and retrieve the detailed information of the account along with its address, phones in search result.

**Special Notes –**

* Account will be retrieved as per searchAccountRequestDO.inquiryLevel. If inquiry level is not provided then only the search result with details for AccountDO (ACCOUNT table) will be returned.
* if searchAccountRequestDO.inquiryFilter is null then only ACTIVE accounts will be searched

### searchAccountByLEAttributes

**Description:** search the account in database based on associated legal entity attributes. The database will be searched for legal entities matching the search criteria and the accounts associated with these Legal entities will be returned in response.

**Transaction Service Name:** searchAccountByLEAttributes

**Request DO:** searchAccountRequestDO

**Response DO:** searchAccountRequestDO

**Mandatory Input:** One of the attributes SearchAccountRequestDO.displayName, nameOne, lastName, corporationName is required if systemKeysRegistryReferenceId or identificationNumber or phoneNumber is not provided.

**Pagination Supported:** YES

**Filter:** ACTIVE/INACTIVE/ALL

**Wildcard Support:** YES

**Use Case –** get the accounts associated with searched legal entities

**Special Notes –**

* Account will be retrieved as per searchAccountRequestDO.inquiryLevel. If inquiry level is not provided then only the search result with details for AccountDO (ACCOUNT table) will be returned.
* if searchAccountRequestDO.inquiryFilter is null then only ACTIVE legal entities and accounts will be searched

### searchInactiveLeRegistry

**Description:** search the INACTIVE\_LE\_REGISTRY table based on its attributes

**Transaction Service Name:** searchInactiveLeRegistry

**Request DO:** inactiveLeRegistryDO

**Response DO:** inactiveLeRegistryDO

**Mandatory Input:** inactiveLeRegistryDO.legalentityIdpk or inactivationReasonRefkey

**Pagination Supported:** YES

**Filter:** ACTIVE/INACTIVE/ALL

**Wildcard Support:** YES

**Use Case –** search the INACTIVE\_LE\_REGISTRY table based on its attributes

**Special Notes – N/A**

### searchLegalEntityByAccountAttributes

**Description:**  This service search legal entity based on account attributes. All the legal entities associated with the accounts searched will be returned in the response

**Transaction Service Name:** searchLegalEntityByAccountAttributes

**Request DO:** searchLegalEntityRequestDO

**Response DO:** legalentityDOList

**Mandatory Input:** at least one of the following

* searchLegalEntityRequestDO.displayName
* searchLegalEntityRequestDO.accountidPk
* searchLegalEntityRequestDO.sourceSystemRefkey
* searchLegalEntityRequestDO.sourceAccountId
* searchLegalEntityRequestDO.accountName
* searchLegalEntityRequestDO.accountName2
* searchLegalEntityRequestDO.accountDescription
* searchLegalEntityRequestDO.accountSourceStatusRefkey
* searchLegalEntityRequestDO.accountMdmStatusRefkey

**Pagination Supported:** YES

**Filter:** ACTIVE/INACTIVE/ALL

**Wildcard Support:** YES

**Use Case –** This service search legal entity based on account attributes

**Special Notes –** The legal entity can be retrieved based on inquiry level. Provide valid value in searchLegalEntityRequestDO.inquiryLevel attribute to retrieve legal entity based on inquiry level. if inquiry level is not provided then the search response will result only the LegalentityDOs without any child objects of respective LegalentityDO.

### searchLegalEntityByLEAttributes

**Description:** This service search legal entity based on LE attributes.

**Transaction Service Name:** searchLegalEntityByLEAttributes

**Request DO:** searchLegalEntityRequestDO

**Response DO:** legalentityDOList

**Mandatory Input:** One of the attributes SearchLegalEntityRequestDO.displayName, nameOne, lastName, corporationName is required if systemKeysRegistryReferenceId or identificationNumber or phoneNumber is not provided

**Pagination Supported:** YES

**Filter:** ACTIVE/INACTIVE/ALL

**Wildcard Support:** YES

**Use Case –** search legal entity based on LE attributes.

**Special Notes –** The legal entity can be retrieved based on inquiry level. Provide valid value in searchLegalEntityRequestDO.inquiryLevel attribute to retrieve legal entity based on inquiry level. if inquiry level is not provided then the search response will result only the LegalentityDOs without any child objects of respective LegalentityDO.

### searchMatchCandidateLE

**Description:** This service search the candidates for given LE. If the

**Transaction Service Name:** searchMatchCandidateLE

**Request DO:** searchMatchCandidateLERequestDO

**Response DO:** legalentityDOList

**Mandatory Input:**

* searchMatchCandidateLERequestDO.IdentificationTypeRefkeyList
* searchMatchCandidateLERequestDO.identificationNumberList
* One of the below
  + searchMatchCandidateLERequestDO.nameOneList
  + searchMatchCandidateLERequestDO.lastNameList

**OR**

* + searchMatchCandidateLERequestDO.corporationNameList

**Pagination Supported:** YES

**Filter:** ACTIVE/INACTIVE/ALL

**Wildcard Support:** YES

**Use Case –**

* search the match candidates for person or corporation type of LE based on exact match
* search the match candidates for person or corporation type of LE based on phonetic match

**Special Notes –** if phonetic search then set the isPhoneticSearch = true.

### searchMatchCandidateLeRegistryBase

**Description:** search the match candidates from the MATCH\_CANDIDATE\_LE\_REGISTRY table. Its considered that the matching is already performed and candidates list is created in the MATCH\_CANDIDATE\_LE\_REGISTRY table.

**Transaction Service Name:** searchMatchCandidateLeRegistryBase

**Request DO:** matchCandidateLeRegistryDO

**Response DO:** matchCandidateLeRegistryDOList

**Mandatory Input:** one of the attributematchCandidateLeRegistryDO.legalentityidpk or candidateLegalentityidpk or matchPattern or matchProposedActionRefkey or matchActionstatusRefkey is needed.

**Pagination Supported:** YES

**Filter:** ACTIVE/INACTIVE/ALL

**Wildcard Support:** YES

**Use Case:**

* search the match candidates of particular LE
* search the match candidates of particular match pattern
* search the match candidates of specific proposed action or action status

**Special Notes:** N/A

### searchMatchMergedLeAssocBase

**Description:** This service search the relation of the legal entities merged.

**Transaction Service Name:** searchMatchMergedLeAssocBase

**Request DO:** matchMergedLeAssocDO

**Response DO:** matchMergedLeAssocDOList

**Mandatory Input:** MatchMergedLeAssocDO.survivorLegalentityIdpk or mergedLegalentityIdpk or mergeReasonRefkey

**Pagination Supported:** YES

**Filter:** ACTIVE/INACTIVE/ALL

**Wildcard Support:** YES

**Use Case:**

* search the legal entities merged based on survivor legal entity
* search the legal entities merged based on non-survivor (merged) legal entity
* search the legal entities based on merged reason

**Special Notes:** N/A

### updateAccount

**Description:** update an account with its address and phone associations.

**Transaction Service Name:** updateAccount

**Request DO:** accountDO

**Response DO:** accountDO

**Mandatory Input:** accountDO.idPk, version

**Use Case:**

* Update existing account attributes and update account to address and/or account to phone associations.
* Update existing account and create new address/phone association.

**Special Notes:**

* The idPk of each child object alongwith version needs to be present in the request if the same needs to be updated. E.g. if the address of the account needs to be updated then idPk of the accountDO.accountAddressAssocDOList(0).idPk and accountDO.accountAddressAssocDOList(0).addressDO.idPk must be present with respective version attribute.
* If the idPk at the child object level is null then its assumed to be a new associated and a new record will be created in the database. So update transaction can be used to create new address or phone number association with existing account.

### updateAccountAddressAssociation

**Description:** update the existing account – address association. If address Id in the request (txnPayload.accountAddressAssocDO.addressIdpk) is null then new address will be created in ADDRESS table and will be associated with account. If address Id (txnPayload.accountAddressAssocDO.addressIdpk) is present in the request then account and address association updated.

**Transaction Service Name:** updateAccountAddressAssociation

**Request DO:** accountAddressAssocDO

**Response DO:** accountAddressAssocDO

**Mandatory Input:** accountAddressAssocDO.accountgalEntityIdPk

**Use Case:**

* update existing account – address association record and update the assoc attributes as well as address attributes
* update existing account – address association record and update the assoc attributes and create a new address record in database.

**Special Notes –** This service does not update any account attributes.

### updateAccountPhoneAssociation

**Description:** update the phone number associated with an account

**Transaction Service Name:** updateAccountPhoneAssociation

**Request DO:** accountPhoneAssocDO

**Response DO:** accountPhoneAssocDO

**Mandatory Input:** accountPhoneAssocDO.accountIdPk

**Use Case:**

* update existing account – phone number association record and update the phoneStandardizedDO of the phone number.
* update existing account – address association record and update the assoc attributes and create a new phoneStandardizedDO record in database.

**Special Notes:** N/A

### updateEntityGroup

**Description:** This service is used to update existing entity group as well as new entity grouping association.

**Transaction Service Name:** updateEntityGroup

**Request DO:** entityGroupDO

**Response DO:** entityGroupDO

**Mandatory Input:** entityGroupDO.idPk

**Use Case –**

* update existing entity group
* update existing entity group association
* add a new association to entity entity group

**Special Notes –** entityGroupDO contains entityGroupAssocDOList object which can be used to add associations. To update the existing association, idPk and version needs to be provided.

### updateLeAccountAssociation

**Description:** Update existing LE Account association

**Transaction Service Name:** updateLeAccountAssociation

**Request DO:** leAccountAssocDO

**Response DO:** leAccountAssocDO

**Mandatory Input:**

* LeAccountAssocDO.idPk
* LeAccountAssocDO.LegalEntityIdPk
* leAccountAssocDO.AccountIdpk or AccountDO to create new account

**Use Case:**

* update existing Le account association
* update existing Le account association and update account attributes
* update existing Le account association to replace the existing account by creating new account

**Special Notes:** This service internally uses update Account so all the account details including address and phone association can be updated through the same.

### updateLeAddressAssociation

**Description:** update existing address association along with address details

**Transaction Service Name:** updateLeAddressAssociation

**Request DO:** leAddressAssocDO

**Response DO:** leAddressAssocDO

**Mandatory Input:**

* LeAddressAssocDO.idPk
* LeAddressAssocDO.LegalEntityIdPk
* leAddressAssocDO.addressIdpk or AddressDO to create new address

**Use Case:**

* update existing Le address association
* update existing Le address association and update address attributes
* update existing Le address association to replace the existing address by creating new address

**Special Notes:** N/A

### updateLeCorporation

**Description:**  update a legal entity of type Corporation along with corporation names.

**Transaction Service Name:** updateLeCorporation

**Request DO:** leCorporationDO

**Response DO:** leCorporationDO

**Mandatory Input:** leCorporationDO.legalentityIdpk

**Use Case –**

* update a corporation and update existing name
* update a corporation and add new name

**Special Notes –** This method does not update a record in LEGALENTITY table.

### updateLePerson

**Description:** update a legal entity of type Person along with person names.

**Transaction Service Name:** updateLePerson

**Request DO:** lePersonDO

**Response DO:** lePersonDO

**Mandatory Input:** lePersonDO.legalentityIdpk

**Use Case –**

* update a person and update existing name
* update a person and add new name

**Special Notes –** This method does not create a record in LEGALENTITY.

### updateLePhoneAssociation

**Description:** update an existing association between legal entity and phone. It’s also possible to create or update a standardized phone number record through this service.

**Transaction Service Name:** updateLePhoneAssociation

**Request DO:** lePhoneAssocDO

**Response DO:** lePhoneAssocDO

**Mandatory Input:**

* LePhoneAssocDO.idpk
* LePhoneAssocDO.version
* LePhoneAssocDO.LegalEntityIdPk

**Use Case –**

1. Update existing legal entity to phone association
2. Update existing legal entity to phone association and add standardized phone number.

**Special Notes –** N/A

### updateLePropertyAssociation

**Description:** update an existing association between existing legal entity and existing property. It’s also possible to create a new property record and associate it with legal entity.

**Transaction Service Name:** updateLePropertyAssociation

**Request DO:** lePropertyAssocDO

**Response DO:** lePropertyAssocDO

**Mandatory Input:**

* lePropertyAssocDO.legalEntityIdPk
* lePropertyAssocDO.propertyIdpk or lePropertyAssocDO. propertyDO
* lePropertyAssocDO.version

**Use Case –**

1. update existing legal entity to property association
2. update existing legal entity to property association by creating new property record.

**Special Notes –** N/A

### updateLeVehicleAssociation

**Description:** update an existing association between existing legal entity and existing Vehicle. It’s also possible to create a new vehicle record and associate it with legal entity.

**Transaction Service Name:** updateLeVehicleAssociation

**Request DO:** leVehicleAssocDO

**Response DO:** leVehicleAssocDO

**Mandatory Input:**

* leVehicleAssocDO.legalEntityIdPk
* leVehicleAssocDO.vehicleDO or leVehicleAssocDO.vehicleIdPk
* leVehicleAssocDO.version

**Use Case –**

1. update existing legal entity to property association and update existing property record as well
2. update existing legal entity to property association and create new property record

**Special Notes –** N/A

### updateLegalEntity

**Description:** update a legal entity along with person or corporation, system registry keys, preferences, identifiers, address, phone, account, property, vehicle and miscellaneous info

**Transaction Service Name:** updateLegalEntity

**Request DO:** legalentityDO

**Response DO:** legalentityDO

**Mandatory Input:**

* legalentityDO.IdPk
* legalentityDO.version
* entityObjectTypeRefkey
* lePersonDO or leCorporationDO

**Use Case –**

1. update existing legal entity along with all its objects like person or corporation, system registry keys, preferences, identifiers, address, phone, account, property, vehicle and miscellaneous info
2. update existing legal entity and create all its objects like system registry keys, preferences, identifiers, address, phone, account, property, vehicle and miscellaneous info

**Special Notes –** N/A

## Reference data Table Management services

All the reference data services follow same structure as follows. The transactionServiceName and txnPayload.xxxxxxxDO will change as per the reference list name.

### Create Request - sample JSON message

|  |
| --- |
| {  "txnHeader": {  "requesterLanguage": "1",  "requesterLocale": null,  "userName": "admin",  "userRole": "admin",  "accessToken": null,  "txnCorrelationId": null,  "txnMessageId": "1234567890123",  "requestOriginSource": null,  "requesterSystem": null,  "transactionServiceName": "create<referenceTableName>Base"  },  "txnPayload": {  "<referenceTableName>DO": {  "idPk": null,  "version": null,  "createdTs": null,  "deletedTs": null,  "updatedTs": null,  "updatedByUser": null,  "updatedByTxnId": null,  "configLanguageCodeKey": null,  "key": null,  "value": null,  "description": null,  }}    } |

### update request - sample JSON message

|  |
| --- |
| {  "txnHeader": {  "requesterLanguage": "1",  "requesterLocale": null,  "userName": "admin",  "userRole": "admin",  "accessToken": null,  "txnCorrelationId": null,  "txnMessageId": "1234567890123",  "requestOriginSource": null,  "requesterSystem": null,  "transactionServiceName": "update<referenceTableName>Base"  },  "txnPayload": {  "<referenceTableName>DO": {  "idPk": null,  "version": null,  "createdTs": null,  "deletedTs": null,  "updatedTs": null,  "updatedByUser": null,  "updatedByTxnId": null,  "configLanguageCodeKey": null,  "key": null,  "value": null,  "description": null,  }    }    } |

### Retrieve Request - sample JSON message

|  |
| --- |
| {  "txnHeader": {  "requesterLanguage": "1",  "requesterLocale": null,  "userName": "admin",  "userRole": "admin",  "accessToken": null,  "txnCorrelationId": null,  "txnMessageId": "1234567890123",  "requestOriginSource": null,  "requesterSystem": null,  "transactionServiceName": "retrieve<referenceTableName>Base"  },  "txnPayload": {  "<referenceTableName>DO": {  "idPk": "1"    }    }    } |

### findByBusinessKey request - sample JSON message.

|  |
| --- |
| {  "txnHeader": {  "requesterLanguage": "1",  "requesterLocale": null,  "userName": "admin",  "userRole": "admin",  "accessToken": null,  "txnCorrelationId": null,  "txnMessageId": "1234567890123",  "requestOriginSource": null,  "requesterSystem": null,  "transactionServiceName": "find<referenceTableName>ByBusinessKeyBase"  },  "txnPayload": {  "<referenceTableName>DO": {  "configLanguageCodeKey": "1",  "key": "ACTIVE"  }    }    } |

### findAllRecorfdsByLanguageCode request - sample JSON message.

The response of this message is the list object of request DO. E.g. if the request do is refCountryIsoDO then the response object (in the txnPayload) would be refCountryIsoDOList.

|  |
| --- |
| {  "txnHeader": {  "requesterLanguage": "1",  "userName": "Rakesh9",  "userRole": "admin9",  "txnMessageId": "12312311115999",  "transactionServiceName": "findAll<referenceTableName>ByLanguageCodeBase"  },  "txnPayload": {  "paginationIndexOfCurrentSlice": 0,  "paginationPageSize": 25,  "<referenceTableName>DO": {  "configLanguageCodeKey": "1"  }    }    } |

All the below mentioned services would have same structure as mentioned above so those are not covered in detail independently.

### List of Reference data Management services following common request structure

#### createRefAccountMdmStatusBase

#### createRefAccountSourceStatusBase

#### createRefAddressSubtypeBase

#### createRefAddressTypeBase

#### createRefAgreementTypeBase

#### createRefAssocTypeBase

#### createRefBillingModeTypeBase

#### createRefBranchCodeBase

#### createRefClassificationCodeBase

#### createRefCorporationNameTypeBase

#### createRefCorporationTypeBase

#### createRefCountryIsoBase

#### createRefCurrencyBase

#### createRefDeactivationReasonBase

#### createRefEntityObjectTypeBase

#### createRefGenderBase

#### createRefGroupSubtypeBase

#### createRefGroupTypeBase

#### createRefHighestEduQualBase

#### createRefIdentificationTypeBase

#### createRefImportanceTypeBase

#### createRefIndustryCodeBase

#### createRefLanguageCodeBase

#### createRefLeRatingBase

#### createRefLeRelationshipTypeBase

#### createRefLeRoletypeBase

#### createRefLobtypeBase

#### createRefPersonTypeBase

#### createRefPersonnameTypeBase

#### createRefPhoneSubtypeBase

#### createRefPhoneTypeBase

#### createRefPreferenceTypeBase

#### createRefPrefixNameBase

#### createRefPropertyLeReltypeBase

#### createRefRelationshipStatusBase

#### createRefSourceSystemBase

#### createRefStateProvinceBase

#### createRefStatusInSourceBase

#### createRefStatusTypeBase

#### createRefSuffixNameBase

#### createRefTerminationReasonBase

#### findAllRefAccountMdmStatusByLanguageCodeBase

#### findAllRefAccountSourceStatusByLanguageCodeBase

#### findAllRefAddressSubtypeByLanguageCodeBase

#### findAllRefAddressTypeByLanguageCodeBase

#### findAllRefAgreementTypeByLanguageCodeBase

#### findAllRefAssocTypeByLanguageCodeBase

#### findAllRefBillingModeTypeByLanguageCodeBase

#### findAllRefBranchCodeByLanguageCodeBase

#### findAllRefClassificationCodeByLanguageCodeBase

#### findAllRefCorporationNameTypeByLanguageCodeBase

#### findAllRefCorporationTypeByLanguageCodeBase

#### findAllRefCountryIsoByLanguageCodeBase

#### findAllRefCurrencyByLanguageCodeBase

#### findAllRefDeactivationReasonByLanguageCodeBase

#### findAllRefEntityObjectTypeByLanguageCodeBase

#### findAllRefGenderByLanguageCodeBase

#### findAllRefGroupSubtypeByLanguageCodeBase

#### findAllRefGroupTypeByLanguageCodeBase

#### findAllRefHighestEduQualByLanguageCodeBase

#### findAllRefIdentificationTypeByLanguageCodeBase

#### findAllRefImportanceTypeByLanguageCodeBase

#### findAllRefIndustryCodeByLanguageCodeBase

#### findAllRefLanguageCodeByLanguageCodeBase

#### findAllRefLeRatingByLanguageCodeBase

#### findAllRefLeRelationshipTypeByLanguageCodeBase

#### findAllRefLeRoletypeByLanguageCodeBase

#### findAllRefLobtypeByLanguageCodeBase

#### findAllRefPersonTypeByLanguageCodeBase

#### findAllRefPersonnameTypeByLanguageCodeBase

#### findAllRefPhoneSubtypeByLanguageCodeBase

#### findAllRefPhoneTypeByLanguageCodeBase

#### findAllRefPreferenceTypeByLanguageCodeBase

#### findAllRefPrefixNameByLanguageCodeBase

#### findAllRefPropertyLeReltypeByLanguageCodeBase

#### findAllRefRelationshipStatusByLanguageCodeBase

#### findAllRefSourceSystemByLanguageCodeBase

#### findAllRefStateProvinceByLanguageCodeBase

#### findAllRefStatusInSourceByLanguageCodeBase

#### findAllRefStatusTypeByLanguageCodeBase

#### findAllRefSuffixNameByLanguageCodeBase

#### findAllRefTerminationReasonByLanguageCodeBase

#### findRefAccountMdmStatusByBusinessKeyBase

#### findRefAccountSourceStatusByBusinessKeyBase

#### findRefAddressSubtypeByBusinessKeyBase

#### findRefAddressTypeByBusinessKeyBase

#### findRefAgreementTypeByBusinessKeyBase

#### findRefAssocTypeByBusinessKeyBase

#### findRefBillingModeTypeByBusinessKeyBase

#### findRefBranchCodeByBusinessKeyBase

#### findRefClassificationCodeByBusinessKeyBase

#### findRefCorporationNameTypeByBusinessKeyBase

#### findRefCorporationTypeByBusinessKeyBase

#### findRefCountryIsoByBusinessKeyBase

#### findRefCurrencyByBusinessKeyBase

#### findRefDeactivationReasonByBusinessKeyBase

#### findRefEntityObjectTypeByBusinessKeyBase

#### findRefGenderByBusinessKeyBase

#### findRefGroupSubtypeByBusinessKeyBase

#### findRefGroupTypeByBusinessKeyBase

#### findRefHighestEduQualByBusinessKeyBase

#### findRefIdentificationTypeByBusinessKeyBase

#### findRefImportanceTypeByBusinessKeyBase

#### findRefIndustryCodeByBusinessKeyBase

#### findRefLanguageCodeByBusinessKeyBase

#### findRefLeRatingByBusinessKeyBase

#### findRefLeRelationshipTypeByBusinessKeyBase

#### findRefLeRoletypeByBusinessKeyBase

#### findRefLobtypeByBusinessKeyBase

#### findRefPersonTypeByBusinessKeyBase

#### findRefPersonnameTypeByBusinessKeyBase

#### findRefPhoneSubtypeByBusinessKeyBase

#### findRefPhoneTypeByBusinessKeyBase

#### findRefPreferenceTypeByBusinessKeyBase

#### findRefPrefixNameByBusinessKeyBase

#### findRefPropertyLeReltypeByBusinessKeyBase

#### findRefRelationshipStatusByBusinessKeyBase

#### findRefSourceSystemByBusinessKeyBase

#### findRefStateProvinceByBusinessKeyBase

#### findRefStatusInSourceByBusinessKeyBase

#### findRefStatusTypeByBusinessKeyBase

#### findRefSuffixNameByBusinessKeyBase

#### findRefTerminationReasonByBusinessKeyBase

#### retrieveRefAccountMdmStatusBase

#### retrieveRefAccountSourceStatusBase

#### retrieveRefAddressSubtypeBase

#### retrieveRefAddressTypeBase

#### retrieveRefAgreementTypeBase

#### retrieveRefAssocTypeBase

#### retrieveRefBillingModeTypeBase

#### retrieveRefBranchCodeBase

#### retrieveRefClassificationCodeBase

#### retrieveRefCorporationNameTypeBase

#### retrieveRefCorporationTypeBase

#### retrieveRefCountryIsoBase

#### retrieveRefCurrencyBase

#### retrieveRefDeactivationReasonBase

#### retrieveRefEntityObjectTypeBase

#### retrieveRefGenderBase

#### retrieveRefGroupSubtypeBase

#### retrieveRefGroupTypeBase

#### retrieveRefHighestEduQualBase

#### retrieveRefIdentificationTypeBase

#### retrieveRefImportanceTypeBase

#### retrieveRefIndustryCodeBase

#### retrieveRefLanguageCodeBase

#### retrieveRefLeRatingBase

#### retrieveRefLeRelationshipTypeBase

#### retrieveRefLeRoletypeBase

#### retrieveRefLobtypeBase

#### retrieveRefPersonTypeBase

#### retrieveRefPersonnameTypeBase

#### retrieveRefPhoneSubtypeBase

#### retrieveRefPhoneTypeBase

#### retrieveRefPreferenceTypeBase

#### retrieveRefPrefixNameBase

#### retrieveRefPropertyLeReltypeBase

#### retrieveRefRelationshipStatusBase

#### retrieveRefSourceSystemBase

#### retrieveRefStateProvinceBase

#### retrieveRefStatusInSourceBase

#### retrieveRefStatusTypeBase

#### retrieveRefSuffixNameBase

#### retrieveRefTerminationReasonBase

#### updateRefAccountMdmStatusBase

#### updateRefAccountSourceStatusBase

#### updateRefAddressSubtypeBase

#### updateRefAddressTypeBase

#### updateRefAgreementTypeBase

#### updateRefAssocTypeBase

#### updateRefBillingModeTypeBase

#### updateRefBranchCodeBase

#### updateRefClassificationCodeBase

#### updateRefCorporationNameTypeBase

#### updateRefCorporationTypeBase

#### updateRefCountryIsoBase

#### updateRefCurrencyBase

#### updateRefDeactivationReasonBase

#### updateRefEntityObjectTypeBase

#### updateRefGenderBase

#### updateRefGroupSubtypeBase

#### updateRefGroupTypeBase

#### updateRefHighestEduQualBase

#### updateRefIdentificationTypeBase

#### updateRefImportanceTypeBase

#### updateRefIndustryCodeBase

#### updateRefLanguageCodeBase

#### updateRefLeRatingBase

#### updateRefLeRelationshipTypeBase

#### updateRefLeRoletypeBase

#### updateRefLobtypeBase

#### updateRefPersonTypeBase

#### updateRefPersonnameTypeBase

#### updateRefPhoneSubtypeBase

#### updateRefPhoneTypeBase

#### updateRefPreferenceTypeBase

#### updateRefPrefixNameBase

#### updateRefPropertyLeReltypeBase

#### updateRefRelationshipStatusBase

#### updateRefSourceSystemBase

#### updateRefStateProvinceBase

#### updateRefStatusInSourceBase

#### updateRefStatusTypeBase

#### updateRefSuffixNameBase

#### updateRefTerminationReasonBase

#### createRefBatchActionStatusBase

#### updateRefBatchActionStatusBase

#### retrieveRefBatchActionStatusBase

#### findRefBatchActionStatusByBusinessKeyBase

#### findAllRefBatchActionStatusByLanguageCodeBase

#### createRefBatchProposedActionBase

#### updateRefBatchProposedActionBase

#### retrieveRefBatchProposedActionBase

#### findRefBatchProposedActionByBusinessKeyBase

#### findAllRefBatchProposedActionByLanguageCodeBase

#### createRefInactivationReasonBase

#### createRefMatchActionstatusBase

#### createRefMatchProposedActionBase

#### createRefMatchResultBase

#### findAllRefInactivationReasonByLanguageCodeBase

#### findAllRefMatchActionstatusByLanguageCodeBase

#### findAllRefMatchProposedActionByLanguageCodeBase

#### findAllRefMatchResultByLanguageCodeBase

#### findRefMatchResultByBusinessKeyBase

#### findRefInactivationReasonByBusinessKeyBase

#### findRefMatchActionstatusByBusinessKeyBase

#### findRefMatchProposedActionByBusinessKeyBase

#### findAllRefMergeReasonByLanguageCodeBase

#### retrieveRefInactivationReasonBase

#### retrieveRefMatchActionstatusBase

#### retrieveRefMatchProposedActionBase

#### retrieveRefMatchResultBase

#### updateRefInactivationReasonBase

#### updateRefMatchActionstatusBase

#### updateRefMatchProposedActionBase

#### updateRefMatchResultBase

#### createRefMergeReasonBase

#### findRefMergeReasonByBusinessKeyBase

#### retrieveRefMergeReasonBase

#### updateRefMergeReasonBase

# Inquiry levels for Legal entity and Account – Default configuration

Below is the list of inquiry levels configured by default in the CONFIG\_INQUIRY\_LEVELS table. Currently Account and Legal entity level inquiry levels are configured which can also be customized. Please refer development guide to understand more about defining new inquiry level.

Note – By default, the inquiry level is numeric in nature; Higher the number higher the inquiry level.

|  |  |  |  |
| --- | --- | --- | --- |
| **Inquiry Level** | **Applicable DO** | **Child DO** | **Comments if any** |
| 101 | LegalentityDO | LeSystemKeysRegistryDO |  |
| 102 | LegalentityDO | LeCorporationDO |  |
| 102 | LegalentityDO | LePersonDO |  |
| 102 | LegalentityDO | LeSystemKeysRegistryDO |  |
| 103 | LegalentityDO | LeAddressAssocDO |  |
| 103 | LegalentityDO | LeCorporationDO |  |
| 103 | LegalentityDO | LePersonDO |  |
| 103 | LegalentityDO | LePhoneAssocDO |  |
| 103 | LegalentityDO | LePreferencesDO |  |
| 103 | LegalentityDO | LeSystemKeysRegistryDO |  |
| 104 | LegalentityDO | LeAddressAssocDO |  |
| 104 | LegalentityDO | LeCorporationDO |  |
| 104 | LegalentityDO | LeIdentifierKycRegistryDO |  |
| 104 | LegalentityDO | LePersonDO |  |
| 104 | LegalentityDO | LePhoneAssocDO |  |
| 104 | LegalentityDO | LePreferencesDO |  |
| 104 | LegalentityDO | LeSystemKeysRegistryDO |  |
| 104 | LegalentityDO | MiscellaneousInfoDO |  |
| 105 | LegalentityDO | LeAccountAssocDO |  |
| 105 | LegalentityDO | LeAddressAssocDO |  |
| 105 | LegalentityDO | LeCorporationDO |  |
| 105 | LegalentityDO | LeIdentifierKycRegistryDO |  |
| 105 | LegalentityDO | LePersonDO |  |
| 105 | LegalentityDO | LePhoneAssocDO |  |
| 105 | LegalentityDO | LePreferencesDO |  |
| 105 | LegalentityDO | LeSystemKeysRegistryDO |  |
| 105 | LegalentityDO | MiscellaneousInfoDO |  |
| 106 | LegalentityDO | LeAccountAssocDO |  |
| 106 | LegalentityDO | LeAddressAssocDO |  |
| 106 | LegalentityDO | LeCorporationDO |  |
| 106 | LegalentityDO | LeIdentifierKycRegistryDO |  |
| 106 | LegalentityDO | LePersonDO |  |
| 106 | LegalentityDO | LePhoneAssocDO |  |
| 106 | LegalentityDO | LePreferencesDO |  |
| 106 | LegalentityDO | LePropertyAssocDO |  |
| 106 | LegalentityDO | LeSystemKeysRegistryDO |  |
| 106 | LegalentityDO | LeToLeRelationshipDO |  |
| 106 | LegalentityDO | LeVehicleAssocDO |  |
| 106 | LegalentityDO | MiscellaneousInfoDO |  |
| 107 | LegalentityDO | LeAccountAssocDO |  |
| 107 | LegalentityDO | LeCorporationDO |  |
| 107 | LegalentityDO | LePersonDO |  |
| 107 | LegalentityDO | LeSystemKeysRegistryDO |  |
| 108 | LegalentityDO | LeAddressAssocDO | This inquiry level is used by default Match and Merge framework so must not be customized |
| 108 | LegalentityDO | LeCorporationDO | This inquiry level is used by default Match and Merge framework so must not be customized |
| 108 | LegalentityDO | LeIdentifierKycRegistryDO | This inquiry level is used by default Match and Merge framework so must not be customized |
| 108 | LegalentityDO | LePersonDO | This inquiry level is used by default Match and Merge framework so must not be customized |
| 108 | LegalentityDO | LePhoneAssocDO | This inquiry level is used by default Match and Merge framework so must not be customized |