Microservices List C&S :(Success flow)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.No | User Story/Function | Service Name | *Service Description* | Resource Endpoint |  |
| 1. | Managing segments accessible to different users | UserService | Post the data in segementBundleMapping table | /v1/segmentBundleMapping | HTTP METHOD : POST  Request:  {    "segmentRequestList":[    {      "userId":"UserMapping10",      "segmentName":"Segment Sprint 4",      "isOperationFlag" :true      },{      "userId":"UserMapping10",      "segmentName":"Segment Sprint 2",      "isOperationFlag" :true      }    ]  }  Response :  {      "response": {          "code": 200,          "responseMessage": "USER\_SEGMENT\_MAPPING table updated successfully"      },      "segmentResponses": [          {              "userId": "UserMapping10",              "segmentId": "Segment1",              "responseMessage": "UserId created successfully in USER\_SEGMENT\_MAPPING table"          },          {              "userId": "UserMapping10",              "segmentId": "Segment3",              "responseMessage": "UserId created successfully in USER\_SEGMENT\_MAPPING table"          }      ]  } |
| 2 | Managing segments accessible to different users | UserService | Delete the data in segementBundleMapping table | /v1/segmentBundleMapping | HTTP METHOD : DELTE  Request:  {    "segmentRequestList":[    {      "userId":"UserMapping10",      "segmentName":"Segment Sprint 4",      "isOperationFlag" :false      },{      "userId":"UserMapping10",      "segmentName":"Segment Sprint 2",      "isOperationFlag" :false      }    ]  }    Response:  {      "response": {          "code": 200,          "responseMessage": "USER\_SEGMENT\_MAPPING table updated successfully"      },      "segmentResponses": [          {              "userId": "UserMapping10",              "segmentId": "Segment1",              "responseMessage": "UserId deleted successfully in USER\_SEGMENT\_MAPPING table"          },          {              "userId": "UserMapping10",              "segmentId": "Segment3",              "responseMessage": "UserId deleted successfully in USER\_SEGMENT\_MAPPING table"          }      ]  } |
| 3 | As an NCL admin/super-admin user, I should be able to create new user profiles | UserService | Get member details from member table based on memberid | /v1/features/{ memberId } | HTTP METHOD : GET  {  "memberId": "Member1",  "memberUserType": "NCL Admin",  "memberType": "Admin",  "segmentId": "Segment1",  "segmentName": "Segment Sprint 4",  "userType": [  {  "userType": "NCL Admin",  "bundleTypeList": [  {  "bundleName": "BundleSN1",  "bundleDescription": "Description 1"  },  {  "bundleName": "BundleSN2",  "bundleDescription": "Description 2"  }  ]  },  {  "userType": "NCL",  "bundleTypeList": null  }]} |
| 4 | As an NCL admin/super-admin user, I should be able to create new user profiles | UserService | Validate user details on user table | /v1/validateUser | HTTP METHOD :POST  Request:  {  "userid" : "User1",  "memid" : "Member1",  "emailid" : "juhi.shrm@xyz.com",  "phoneNo" : 978462007  }  Response:  {  "statuscode": 404,  "responsemsg": "User Doesn't Exists in Table!",  "filedname": "UserId"  },  {  "statuscode": 404,  "responsemsg": "Email ID Doesn't Exists in Table!",  "filedname": "Email"  },  {  "statuscode": 201,  "responsemsg": "Member ID Already Exists in Table!",  "filedname": "Member ID"  },  {  "statuscode": 404,  "responsemsg": "Mobile Number Doesn't Exists in Table!",  "filedname": "Mobile Number"  }  ]  Success Response :  If user present -  [  {  "statuscode": 201,  "responsemsg": "User Already Exists in Table!",  "filedname": "UserId"  },  {  "statuscode": 404,  "responsemsg": "Email ID Doesn't Exists in Table!",  "filedname": "Email"  },  {  "statuscode": 201,  "responsemsg": "Member ID Already Exists in Table!",  "filedname": "Member ID"  },  {  "statuscode": 404,  "responsemsg": "Mobile Number Doesn't Exists in Table!",  "filedname": "Mobile Number"  }  ] |
| 5 | As an NCL admin/super-admin user, I should be able to create new user profiles | UserService | Create a new user on User table | /v1/createUser | HTTP METHOD : POST  Request:  {  "emailId": "email@1235",  "firstName": "Debajyoti1",  "lastName": "Ghatak1",  "memberId": "Member1",  "mobileNo": 1234565,  "userBundlesList": [  {  "bundleName": "BundleSN1"  },  {  "bundleName": "BundleSN2"  }  ],  "userId": "userId6",  "userSegmentList": [  {  "segmentName": "Segment Sprint 2"  },  {  "segmentName": "Segment Sprint 3"  }  ],  "userType": "NCL\_Admin"  }  Response :  {  "response": {  "code": 200,  "responseMessage": "UserId has been created successfully"  },  "createUserResponse": {  "userId": "userId6",  "memberId": "Member1"  }  } |
| 6 | As an NCL admin/super-admin user, I should be able to create new user profiles | UserService | Post the data in BundleMapping table | /v1/bundleMapping | RequestBody:{    "bundleRequestList":[    {      "userId":"UserMapping1",      "bundleName":"BundleSN1",      "isOperationFlag" :true      },{      "userId":"UserMapping1",      "bundleName":"BundleSN2",      "isOperationFlag" :true      }    ]  }  Response:{      "response": {          "code": 200,          "responseMessage": "USER\_BUNDLE\_MAPPING table updated successfully"      },      "bundleResponse": [          {              "userId": "UserMapping1",              "bundleId": "Bundle1",              "responseMessage": "UserId created successfully in USER\_BUNDLE\_MAPPING table"          },          {              "userId": "UserMapping1",              "bundleId": "Bundle2",              "responseMessage": "UserId created successfully in USER\_BUNDLE\_MAPPING table"          }      ]  } |
| 7 | As an NCL admin/super-admin user, I should be able to create new user profiles | UserService | Delete the data in BundleMapping table | /v1/bundleMapping | RequestBody{    "bundleRequestList":[    {      "userId":"UserMapping1",      "bundleName":"BundleSN1",      "isOperationFlag" :false      },{      "userId":"UserMapping1",      "bundleName":"BundleSN2",      "isOperationFlag" :false      }    ]  }  Response{      "response": {          "code": 200,          "responseMessage": "USER\_BUNDLE\_MAPPING table updated successfully"      },      "bundleResponse": [          {              "userId": "UserMapping1",              "bundleId": "Bundle1",              "responseMessage": "UserId deleted successfully in USER\_BUNDLE\_MAPPING table"          },          {              "userId": "UserMapping1",              "bundleId": "Bundle2",              "responseMessage": "UserId deleted successfully in USER\_BUNDLE\_MAPPING table"          }      ]  } |

Failure scenarios:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | API | Status Code | Description | Sample Response |
| 1 | /v1/validateUser | 200 | Success | {  code: 200  message: " The user does not exist in the DB”} |
| 2 | /v1/validateUser | 400 | Failed! User already existed | {  errorCode: 400  errorMsg: "Bad Request",  errorResponse: " User already existed "  } |
| 3 | /v1/validateUser | 401 | Unauthorized to create a user | errorCode: 401  errorMsg: "Unauthorized",  errorResponse: "The Authorization header was of an unrecognized format or uses an unsupported credential type" |
| 4 | /v1/validateUser | 500 | Internal Server Error | {  errorCode: 500  errorMsg: "InternalError",  errorResponse: "Internal server Error. Try again!!!"  } |
| 5 | /v1/validateUser | 503 | Service Unavailable | {  errorCode: 500  errorMsg: "Service Unavailable",  errorResponse: " Service Unavailable. Try again!!!"  } |