

UMA API Request and Response Guide

API Documentation - User Management (CRUD)

API Documentation - User Management (CRUD)

1.  Overview
2.  Base URL
3.  Authentication
 - 3.1 Get JWT token
4. 
 - 4.1 Static Lookup API
 -  Overview
 -  Field Descriptions
 -  Endpoint Details
 - 4.2 GLN Specific Lookup API
 -  Overview
 -  Field Descriptions
 -  Field Usage Guidelines
 -  Endpoint Details
5. 
 - 5.1  Create User
 - 5.2  Fetch User
 - 5.3  Fetch Users - Fetch all available users under provided APP_ID
 - 5.4  Update User
 - 5.5  Enable/Disable User
 - 5.6  Delete User
 - 5.7  Tracking the Request
 -  Overview
 -  Endpoint Details
 - 5.8  Unauthorized Access
6. 
 - 6.1  Overview
 - 6.2  Offering-Specific Payload Structure
 - 6.3  Payload Field → Lookup Field Mapping
 - 6.4  Key Notes
7. 
 - 7.1  Mandatory Validation
 - 7.2  Conditional Mandatory Validations
 - 7.3  Max Length Validations
 - 7.4  Invalid Data validations
 - 7.5  Other Validation messages

1. 📄 Overview

This document describes the CRUD API endpoints used for managing **Users** in the 1Worldsync system. These APIs support standard operations: **Create, Read, Update, Enable/Disable, Delete and Tracking**.

2. 🌐 Base URL

```
1 DEV: https://message-api.dev.1worldsync.com/V2/messages
2 SPECIAL: https://message-api.special.1worldsync.com/V2/messages
3 QA: https://message-api.qa.1worldsync.com/V2/messages
4 PREPROD: https://message-api.preprod.1worldsync.com/V2/messages
5 PROD: https://message-api.1worldsync.com/V2/messages
```

3. 🗝️ Authentication

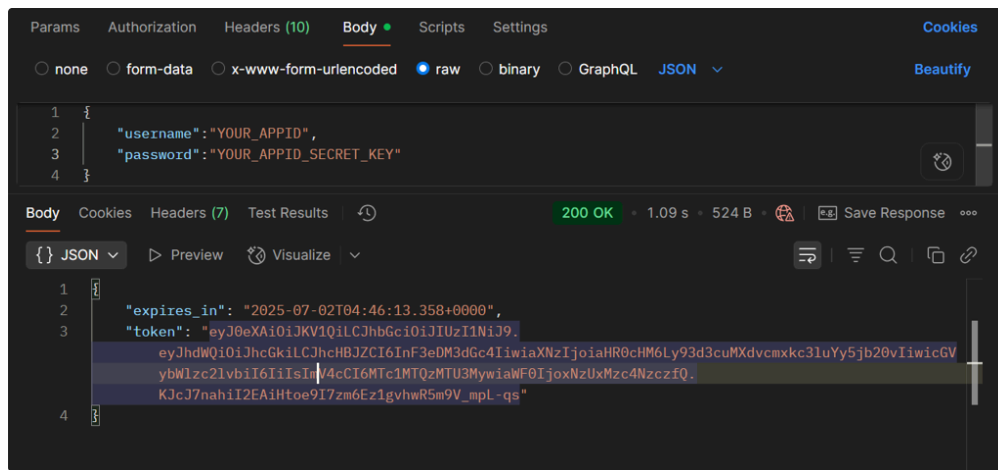
3.1 Get JWT token

- **Method:** POST
- **URL:** `/authenticate`
- **Request Body:**

```
1 {
2   "username": "YOUR_APPID",
3   "password": "YOUR_APPID_SECRET_KEY"
4 }
```

- **Response:**

```
1 {
2   "expires_in": "2025-07-02T04:46:13.358+0000",
3   "token":
"eyJ0eXAiOiJKV1shdiue797hcdQiLCJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJhcGkiLCJhcHBzI6InF3eDM3dGc4IiwiaXNzIjoiaHR0cHM6Ly93d3cuMXdvcmxkc3luYy5jb20vIiwicG9ybmVzI6ImV4cCI6MTc1MTQzMjU3MywiaWF0IjoxNzUxMzc4Nzc4fQ.KJcJ7nahlI2EAiHtoe9I7zm6Ez1gvhwR5m9V_mPL-qs"
4 }
```



Note:

- Use the JWT token generated from the authentication API to access all user-related APIs. Include it in the **Authorization** header as a Bearer token for performing user CRUD operations.
- The JWT token has an expiration time of **14 hours 40 mins**. Once it expires, you will need to generate a new token.

If you attempt to use an expired token, the system will return a **401**

Unauthorized error.

```
1 {
2   "message": "JWT token verification failed",
3   "responseCode": 401
4 }
```

- If invalid username or password, response would be like this.

```
1 {
2     "message": "Invalid username or password",
3     "responseCode": 401,
4     "userErrors": []
5 }
```

4. Lookup values Endpoints

4.1 Static Lookup API

Overview

- The following static fields are used across various API requests. These include predefined values for **prefixes**, **locales**, **roles**, **offerings**, **countries**, **currencies**, and other fields

required during **user creation or updates**.

- The API response contains **key – value** pairs.
- You must send the **key** in create and update User **APIs**.
- The **value** field is intended for **display purposes only** (e.g., for rendering UI dropdowns).
- These fields are static and subject to controlled updates. Any changes will be communicated through official channels.

Refer to the sample JSON structure below.

Field Descriptions

Field Name	Description
prefixes	Prefixes to be used (e.g., Mr, Ms, Dr)
states	State code and label (e.g., AL → Alabama)
countries	Country code and label (e.g., US → UNITED STATES)
locales	Date format preferences based on locale
currencies	Currency codes and labels (e.g., USD)
umaRoles	Roles related to User Management Appliation (UMA)
offerings	System offerings/features that can be assigned to a user
communityRoles	Roles specific to Community Portal
imRoles	User Role when Item Management offering is assigned
preferredItemViews	Preferred item views available for Item Management users
preferredTargetMarkets	Target markets to be assigned when Item Management offering is given
dcRoles	User role when Digital Catalog offering is given

- **Method:** GET
- **URL:** /user/lookUpValues
- **Request Headers:** Authorization: Bearer <your-token>
- **Sample Response:**

```
1 {
2   "prefixes": [
3     {
4       "value": "Mr",
5       "key": "mr"
6     },
7     {
8       "value": "Ms",
9       "key": "ms"
10    },
11    {
12      "value": "Mrs",
13      "key": "mrs"
14    },
15    {
16      "value": "Dr",
17      "key": "dr"
18    }
19  ],
20  "locales": [
21    {
22      "value": "English (United States) DD-MON-YYYY",
23      "key": "1001"
24    },
25    {
26      "value": "English (United States) MM/DD/YYYY",
27      "key": "1004"
28    },
29    {
30      "value": "English (United States) YYYY-MM-DD",
31      "key": "1005"
32    }
33  ],
34  "communityRoles": [
35    {
36      "value": "Item Management",
37      "key": "IM"
38    },
39    {
40      "value": "Product Introduction",
41      "key": "PI"
42    },
43    {
44      "value": "Community",
45      "key": "Community"
46    },
47    {
48      "value": "Recipient",
49      "key": "Recipient"
50    }
51  ]
52 }
```

```

50     }
51 ],
52 "preferredItemViews": [
53     {
54         "value": "Basic Item",
55         "key": "quickEntry"
56     },
57     {
58         "value": "Carrefour",
59         "key": "carrefour"
60     },
61     {
62         "value": "Food Service - US",
63         "key": "foodAndserviceus"
64     },
65     {
66         "value": "Full Item",
67         "key": "onesyncview"
68     },
69     {
70         "value": "GS1 South Africa",
71         "key": "southAfrica"
72     }
73 ],
74 "dcRoles": [
75     {
76         "value": "Digital Catalog Only User",
77         "key": "DCONLY"
78     },
79     {
80         "value": "Digital Catalog Only Admin",
81         "key": "DCADMIN"
82     },
83     {
84         "value": "Digital Catalog List Only User",
85         "key": "DCLIST"
86     }
87 ],
88 "umaRoles": [
89     {
90         "value": "Org Admin",
91         "key": "ORG_ADMIN"
92     },
93     {
94         "value": "User Admin",
95         "key": "USER_ADMIN"
96     },
97     {
98         "value": "End User",
99         "key": "END_USER"
100    }
101 ],
102 "preferredTargetMarkets": [
103     {
104         "value": "PUERTO RICO",
105         "key": "PR"
106     },
107     {

```

```
108         "value": "PORTUGAL",
109         "key": "PT"
110     },
111     {
112         "value": "PALAU",
113         "key": "PW"
114     },
115     {
116         "value": "PARAGUAY",
117         "key": "PY"
118     },
119     {
120         "value": "QATAR",
121         "key": "QA"
122     },
123     {
124         "value": "ANDORRA",
125         "key": "AD"
126     }
127 ],
128 "countries": [
129     {
130         "value": "PUERTO RICO",
131         "key": "PR"
132     },
133     {
134         "value": "PORTUGAL",
135         "key": "PT"
136     },
137     {
138         "value": "PALAU",
139         "key": "PW"
140     },
141     {
142         "value": "PARAGUAY",
143         "key": "PY"
144     },
145     {
146         "value": "QATAR",
147         "key": "QA"
148     },
149     {
150         "value": "ANDORRA",
151         "key": "AD"
152     }
153 ],
154 "offerings": [
155     {
156         "value": "Community Portal",
157         "key": "community"
158     },
159     {
160         "value": "Developer API Access",
161         "key": "developerAPI"
162     },
163     {
164         "value": "Item Management (IM)",
165         "key": "I2DataCatalog"
```

```

166     },
167     {
168         "value": "Product Introduction",
169         "key": "Initiate"
170     },
171     {
172         "value": "Require MFA for Login",
173         "key": "requireMFA"
174     },
175     {
176         "value": "Digital Catalog",
177         "key": "salesKit"
178     },
179     {
180         "value": "Workflow",
181         "key": "workflow"
182     }
183 ],
184 "states": [
185     {
186         "value": "Alabama",
187         "key": "AL"
188     },
189     {
190         "value": "Alaska",
191         "key": "AK"
192     },
193     {
194         "value": "Arizona",
195         "key": "AZ"
196     },
197     {
198         "value": "Arkansas",
199         "key": "AR"
200     }
201 ],
202 "currencies": [
203     {
204         "value": "United Arab Emirates, Dirham",
205         "key": "AED"
206     },
207     {
208         "value": "Afghanistan, Afghani",
209         "key": "AFA"
210     },
211     {
212         "value": "Albania, Lek",
213         "key": "ALL"
214     },
215     {
216         "value": "Armenia, Dram",
217         "key": "AMD"
218     }
219 ],
220 "imRoles": [
221     {
222         "value": "Item Manager",
223         "key": "ItemManager"

```



```

224     },
225     {
226         "value": "Digital Catalog Only User",
227         "key": "SalesKitUserRole"
228     }
229 ]
230 }

```

 Only the **key** values must be sent in requests. **value** is for display/reference only.

4.2 GLN Specific Lookup API

Overview

- The following values are GLN-specific and used in scenarios involving **Item Management** and **Product Introduction** offerings. These fields help managing cascade access, default playlists, and permissions applicable to a user under specific offerings.

Field Descriptions

Field Name	Description
playLists	Default playlists selectable when Item Management offering is enabled
imCascadeGlns	List of GLNs to which cascade access can be granted under Item Management
piRoles	List of roles assignable when Product Introduction offering is enabled
piCascadeGlns	List of GLNs to which cascade access can be granted under Product Introduction
piPermissions	List of permissions assignable when Product Introduction offering is enabled

Field Usage Guidelines

Structure	How to Use in API Request	Used for Display?
key – value pairs	Send key only	Yes – show value

key – value – portal combinations	Send key only	Yes – show portal + value
List of strings	Send the string value as-is	yes

Notes

- For **piRoles** and **piPermissions**, you'll see an additional **portal** field which indicates the owning portal (e.g., Kroger, Albertsons). This is **only for display**.
- For fields like **imCascadeGlns** or **piCascadeGlns**, the values are raw GLNs in a string list. Send them **as-is** in API requests.

Endpoint Details

- **Method:** GET
- **URL:** `/user/glnLookUpValues?gln=XXXXXXXXXXXXX`
 - The **gln** parameter is **optional**.
 - If you **pass a GLN**, the API will return:
 - The **playlist** associated with that GLN
 - The **IM Cascade GLNs** (Child GLNs with Item Management offering) for the given GLN
 - The **PI Cascade GLNs** (Child GLNs with Product Introduction Offerings) for the given GLN
 - The **PI Roles** configured for a given GLN
 - The **PI Permissions** configured for a given GLN
 - If you **don't pass a GLN**, it will use the **APP_ID GLN** and return:
 - The **playlist** for APP_ID GLN
 - The **IM Cascade GLNs** (Child GLNs of the APP_ID GLN with Item Management offerings)
 - The **PI Cascade GLNs** (Child GLNs of the APP_ID GLN with Product Introduction Offerings)

- The **PI Roles** configured for APP_ID GLN
- The **PI Permissions** configured for APP_ID GLN
- **Request Headers:** Authorization: Bearer <your-token>
- **Sample Response:**

```
1  {
2      "playLists": [
3          {
4              "value": "All Populated",
5              "key": "1ws_allPopulated"
6          },
7          {
8              "value": "Acosta",
9              "key": "acosta"
10         },
11         {
12             "value": "Ahold US",
13             "key": "ahold_us"
14         }
15     ],
16     "imCascadeGlns": [
17         "1242324432221",
18         "1242324432238",
19     ],
20     "piCascadeGlns": [
21         "1242324432221",
22         "1242324432238",
23     ],
24     "piPermissions": [
25         {
26             "portal": "The Kroger Company",
27             "value": "Enable SDL Hierarchy",
28             "key": "SDL_HIERARCHY"
29         }
30     ],
31     "piRoles": [
32         {
33             "portal": "Albertsons Companies",
34             "value": "Supplier Entry",
35             "key": "SW_VENDOR_ENTRY"
36         },
37         {
38             "portal": "Albertsons Companies",
39             "value": "Supplier Read-Only",
40             "key": "SW_VENDOR_READONLY"
41         },
42         {
43             "portal": "McLane",
44             "value": "Supplier Entry",
45             "key": "MCL_VENDOR_ENTRY"
46         },
47         {
48             "portal": "McLane",
49             "value": "Supplier Read-Only",
```

```

50         "key": "MCL_VENDOR_READONLY"
51     },
52     {
53         "portal": "The Kroger Company",
54         "value": "Supplier Entry",
55         "key": "KRO_VENDOR_ENTRY"
56     },
57     {
58         "portal": "The Kroger Company",
59         "value": "Supplier Read-Only",
60         "key": "KRO_VENDOR_READONLY"
61     }
62 ]
63 }

```

♦ **Note:**

If you pass a GLN which is not authorized, you may get **403 forbidden** error response.

```

1 {
2     "message": "Unauthorized access to GLN: XXXXXXXXXXXXX",
3     "responseCode": 403,
4     "userErrors": []
5 }

```

5. 🚀 API Endpoints

- ❗ ☒ All **our APIs** runs through validations and accept the request only if all validations pass through.
- ☒ If any validation fails (e.g., invalid email format, missing required fields, unauthorized action), the complete request will be rejected, and a corresponding error response will be returned. You will have to correct the data send the complete request again.
- ☒ For example, a **Create User** operation will be accepted **only when all user-level and request-level validations pass**, ensuring that the request can proceed for asynchronous processing.
- ☒ The **User Create, Update, and Delete APIs** are **asynchronous**.
This means the request is accepted and queued for background processing, and the operation may not be completed immediately.
- ☒ To track the status of these asynchronous operations, the system returns a unique `trackingId` upon request submission. You can use this ID to poll a tracking endpoint and monitor the progress or final outcome of your request.

Note:  Refer to [Section 6 - Field Mapping with Lookup APIs](#) for complete guidance on how to populate static fields (like prefix, country, roles, offerings, etc.) using lookup APIs.

- **Method:** POST
- **URL:** `/users`
- **Request Headers:** `Authorization: Bearer <your-token>`
- **Request Body:**

```
1  [
2      {
3          "userName": "sample@company.com",
4          "firstName": "Sample",
5          "lastName": "Create",
6          "middleName": "V1",
7          "prefix": "mr",
8          "title": "Title",
9          "emailID": "xyz@company.com",
10         "phoneNumber": "9999999999",
11         "faxNumber": "555-123-4567",
12         "mobileNumber": "9999999999",
13         "address1": "Address1",
14         "address2": "Address2",
15         "city": "Chicago",
16         "country": "US",
17         "state": "IL",
18         "province": null,
19         "postalCode": "60606",
20         "locale": "1001",
21         "currency": "USD",
22         "umaRole": "END_USER",
23         "userGln": "8778283884823",
24         "glnName": "UMA API Test Org",
25         "userStatus": false,
26         "offering": {
27             "offerings": [
28                 "salesKit",
29                 "community",
30                 "I2DataCatalog",
31                 "Initiate"
32             ],
33             "imDetails": {
34                 "cascadeEnabled": true,
35                 "userRole": "ItemManager",
36                 "preferredItemView": "quickEntry",
37                 "preferredTargetMarket": "US",
38                 "alwaysDefaultPreferredTargetMarket": true,
39                 "cascadedGlns": [
40                     "1242324432221"
41                 ],
42                 "preferredDefaultPlayLists": [
43                     "1ws_allPopulated"
```

```

44         ]
45     },
46     "dcDetails": {
47         "userRole": "DONLY"
48     },
49     "communityDetails": {
50         "roles": [
51             "Community"
52         ]
53     },
54     "piDetails": {
55         "roles": [
56             {
57                 "role": "KRO_VENDOR_ENTRY",
58                 "cascadedGlns": [
59                     "1242324432221",
60                     "1242324432238"
61                 ]
62             },
63             {
64                 "role": "SW_VENDOR_ENTRY",
65                 "cascadedGlns": [
66                     "1242324432238"
67                 ]
68             },
69             {
70                 "role": "DEC_VENDOR_READONLY",
71                 "cascadedGlns": []
72             }
73         ],
74         "permissions": [
75             "SDL_HIERARCHY"
76         ]
77     }
78 }
79 }
80 ]

```

• Response:

Success: (202 Accepted)

```

1  {
2      "trackingId": "28905",
3      "message": "Your request has been accepted and is being processed.",
4      "responseCode": 202,
5      "userErrors": []
6  }

```

If any Validations failures: (400 Bad Request)

```

1  {
2      "trackingId": null,
3      "message": null,
4      "responseCode": 400,
5      "userErrors": [
6          {

```

```

7         "username": "sample@company.com",
8         "errors": [
9             "First Name is required"
10        ]
11    }
12 ]
13 }

```

5.2 Fetch User

- **Method:** GET
- **URL:** `/user/{userName}`
- **Request Headers:** `Authorization: Bearer <your-token>`
- **Response:**

```

1  {
2      "id": "15957",
3      "userName": "sample@company.com",
4      "firstName": "Sample",
5      "lastName": "Create",
6      "middleName": "V1",
7      "prefix": "mr",
8      "title": "Title",
9      "emailID": "xyz@company.com",
10     "phoneNumber": "9999999999",
11     "faxNumber": "555-123-4567",
12     "mobileNumber": "9999999999",
13     "address1": "Address1",
14     "address2": "Address2",
15     "city": "Chicago",
16     "country": "US",
17     "state": "IL",
18     "province": null,
19     "postalCode": "60606",
20     "locale": "1001",
21     "currency": "USD",
22     "umaRole": "END_USER",
23     "userGln": "8778283884823",
24     "glnName": "UMA API Test Org",
25     "userStatus": true,
26     "offering": {
27         "offerings": [
28             "salesKit",
29             "community",
30             "I2DataCatalog",
31             "Initiate"
32         ],
33         "imDetails": {
34             "cascadeEnabled": true,
35             "userRole": "ItemManager",
36             "preferredItemView": "quickEntry",
37             "preferredTargetMarket": "US",

```

```

38         "alwaysDefaultPreferredTargetMarket": true,
39         "cascadedGlns": [
40             "1242324432221"
41         ],
42         "preferredDefaultPlayLists": [
43             "1ws_allPopulated"
44         ]
45     },
46     "dcDetails": {
47         "userRole": "DCONLY"
48     },
49     "piDetails": {
50         "roles": [
51             {
52                 "role": "KRO_VENDOR_ENTRY",
53                 "cascadedGlns": [
54                     "1242324432221",
55                     "1242324432238"
56                 ]
57             },
58             {
59                 "role": "SW_VENDOR_ENTRY",
60                 "cascadedGlns": [
61                     "1242324432238"
62                 ]
63             },
64             {
65                 "role": "DEC_VENDOR_READONLY",
66                 "cascadedGlns": []
67             }
68         ],
69         "permissions": [
70             "SDL_HIERARCHY"
71         ]
72     },
73     "communityDetails": {
74         "roles": [
75             "Community"
76         ]
77     }
78 }
79 }

```

5.3 Fetch Users - Fetch all available users under provided APP_ID

- **Method:** GET
- **URL:** `/users?includeSubLevelUsers=true`
 - Query Parameter
 - `includeSubLevelUsers` (optional):
Determines whether to include users associated with sub-level

(child) GLNs of the `APP_ID` GLN.

- Behavior

- When `includeSubLevelUsers=true` is passed:

The API returns **all users**, including those associated with **child GLNs** of the `APP_ID` GLN.

- When `includeSubLevelUsers` is not provided:

The API returns **only users** associated with the `APP_ID` GLN.

- **Request Headers:** `Authorization: Bearer <your-token>`

- **Response:**

```
1  [
2    {
3      "id": "15957",
4      "userName": "sample@company.com",
5      "firstName": "Sample",
6      "lastName": "Create",
7      "middleName": "V1",
8      "prefix": "mr",
9      "title": "Title",
10     "emailID": "xyz@company.com",
11     "phoneNumber": "9999999999",
12     "faxNumber": "555-123-4567",
13     "mobileNumber": "9999999999",
14     "address1": "Address1",
15     "address2": "Address2",
16     "city": "Chicago",
17     "country": "US",
18     "state": "IL",
19     "province": null,
20     "postalCode": "60606",
21     "locale": "1001",
22     "currency": "USD",
23     "umaRole": "END_USER",
24     "userGln": "8778283884823",
25     "glnName": "UMA API Test Org",
26     "userStatus": true,
27     "offering": {
28       "offerings": [
29         "salesKit",
30         "community",
31         "I2DataCatalog",
32         "Initiate"
33       ],
34       "imDetails": {
35         "cascadeEnabled": true,
36         "userRole": "ItemManager",
37         "preferredItemView": "quickEntry",
```

```

38         "preferredTargetMarket": "US",
39         "alwaysDefaultPreferredTargetMarket": true,
40         "cascadedGlns": [
41             "1242324432221"
42         ],
43         "preferredDefaultPlayLists": [
44             "1ws_allPopulated"
45         ]
46     },
47     "dcDetails": {
48         "userRole": "DCONLY"
49     },
50     "piDetails": {
51         "roles": [
52             {
53                 "role": "KRO_VENDOR_ENTRY",
54                 "cascadedGlns": [
55                     "1242324432221",
56                     "1242324432238"
57                 ]
58             },
59             {
60                 "role": "SW_VENDOR_ENTRY",
61                 "cascadedGlns": [
62                     "1242324432238"
63                 ]
64             },
65             {
66                 "role": "DEC_VENDOR_READONLY",
67                 "cascadedGlns": []
68             }
69         ],
70         "permissions": [
71             "SDL_HIERARCHY"
72         ]
73     },
74     "communityDetails": {
75         "roles": [
76             "Community"
77         ]
78     }
79 }
80 }
81 ]

```

5.4 Update User

Note:  Refer to [Section 6 - Field Mapping with Lookup APIs](#) for complete guidance on how to populate static fields (like prefix, country, roles, offerings, etc.) using lookup APIs.

- **Method:** PUT

- URL: `/users`
- Request Headers: `Authorization: Bearer <your-token>`
- Request Body:

```
1  [
2    {
3      "id": "15957",
4      "userName": "sample@company.com",
5      "firstName": "Sample",
6      "lastName": "Create",
7      "middleName": "V2",
8      "prefix": "mr",
9      "title": "Title",
10     "emailID": "xyz@company.com",
11     "phoneNumber": "9999999999",
12     "faxNumber": "555-123-4567",
13     "mobileNumber": "9999999999",
14     "address1": "Address1",
15     "address2": "Address2",
16     "city": "Chicago",
17     "country": "US",
18     "state": "IL",
19     "province": null,
20     "postalCode": "60606",
21     "locale": "1001",
22     "currency": "USD",
23     "umaRole": "END_USER",
24     "userGln": "8778283884823",
25     "glnName": "UMA API Test Org",
26     "userStatus": true,
27     "offering": {
28       "offerings": [
29         "salesKit",
30         "community",
31         "I2DataCatalog",
32         "Initiate"
33       ],
34       "imDetails": {
35         "cascadeEnabled": true,
36         "userRole": "ItemManager",
37         "preferredItemView": "quickEntry",
38         "preferredTargetMarket": "US",
39         "alwaysDefaultPreferredTargetMarket": true,
40         "cascadedGlns": [
41           "1242324432221"
42         ],
43         "preferredDefaultPlayLists": [
44           "1ws_allPopulated"
45         ]
46       },
47       "dcDetails": {
48         "userRole": "DCONLY"
49       },
50       "piDetails": {
51         "roles": [
52           {
```

```

53         "role": "KRO_VENDOR_ENTRY",
54         "cascadedGlns": [
55             "1242324432221",
56             "1242324432238"
57         ]
58     },
59     {
60         "role": "SW_VENDOR_ENTRY",
61         "cascadedGlns": [
62             "1242324432238"
63         ]
64     },
65     {
66         "role": "DEC_VENDOR_READONLY",
67         "cascadedGlns": []
68     }
69 ],
70 "permissions": [
71     "SDL_HIERARCHY"
72 ]
73 },
74 "communityDetails": {
75     "roles": [
76         "Community"
77     ]
78 }
79 }
80 }
81 ]

```

- **Response:**

Success: (202 Accepted)

```

1 {
2     "trackingId": "28910",
3     "message": "Your request has been accepted and is being processed.",
4     "responseCode": 202,
5     "userErrors": []
6 }

```

If any Validations failures: (400 Bad Request)

```

1 {
2     "trackingId": null,
3     "message": null,
4     "responseCode": 400,
5     "userErrors": [
6         {
7             "username": "sample@company.com",
8             "errors": [
9                 "First Name is required"
10            ]
11        }
12    ]
13 }

```

5.5 Enable/Disable User

- **Method:** PUT
- **URL:** `/usersStatus`
- **Request Headers:** `Authorization: Bearer <your-token>`
- **Request Body:**

To Enable the User(s)

```
1 {  
2   "users": [  
3     "sample@company.com",  
4     "xyz@company.com"  
5   ],  
6   "enable": true  
7 }
```

To Disable the User(s):

```
1 {  
2   "users": [  
3     "sample@company.com",  
4     "xyz@company.com"  
5   ],  
6   "enable": false  
7 }  
8
```

- **Response:**

Success: (200 OK)

For Enable:

```
1 {  
2   "trackingId": null,  
3   "message": "2 user(s) activated",  
4   "responseCode": 200,  
5   "userErrors": []  
6 }
```

For Disable:

```
1 {  
2   "trackingId": null,  
3   "message": "2 user(s) deactivated",  
4   "responseCode": 200,  
5   "userErrors": []  
6 }  
7
```

If any validation failures: if incase no userNames supplied in 'users' field in payload (400 Bad Request)

```
1 {
2   "message": "Users are not Provided",
3   "responseCode": 400,
4   "userErrors": []
5 }
```

5.6 Delete User

- **Method:** DELETE
- **URL:** /users
- **Request Headers:** Authorization: Bearer <your-token>
- **Request Body:**

```
1 [
2   "sample@company.com",
3   "xyz@company.com"
4 ]
```

- **Response:**

Success: (202 Accepted)

```
1 {
2   "trackingId": "28915",
3   "message": "Your request has been accepted and is being processed.",
4   "responseCode": 202,
5   "userErrors": []
6 }
```

If any validation failures: if incase no userNames supplied in 'users' field in payload (400 Bad Request)

```
1 {
2   "message": "Users are not Provided",
3   "responseCode": 400,
4   "userErrors": []
5 }
```

5.7 📌 Tracking the Request

📄 Overview

- The **User Create**, **Update**, and **Delete** APIs are **asynchronous**.

This means the request is accepted and queued for background processing, and the operation may not be completed immediately.

- To track the status of these asynchronous operations, the system returns a unique `trackingId` upon request submission. You can use this ID to poll a tracking endpoint and monitor the progress or final outcome of your request.

- 📌 **Tracking ID**

- A `trackingId` is returned in the response when you submit a Create, Update, or Delete request.
- This ID serves as a **reference** for tracking the processing status of the operation.

- 🔍 **How to Track the Status**

- To check the status and progress of a submitted request, use the following tracking endpoint by passing the `trackingId` as a **path parameter**.

🔗 Endpoint Details

- **Method:** GET
- **URL:** `/user/requestStatus/{trackingId}`
- **Request Headers:** `Authorization: Bearer <your-token>`
- **Response Scenarios:**
 - ✅ 1. Status: `COMPLETED` – All Users Processed Successfully
 - HTTP Status Code: `200 OK`
 - `data.success` : contains a list of all successfully processed users.
 - `data.failures` : will be empty.
 - `message` : summarizes number of users processed successfully.

```

1 {
2   "data": {
3     "failures": [],
4     "success": [
5       "sample@company.com"
6     ]
7   },
8   "message": "Successfully processed 1 user(s).",
9   "responseCode": 200,
10  "status": "COMPLETED"
11 }

```

◦ 2. Status: **COMPLETED** – Partial or Complete Failures

- HTTP Status Code: **200 OK**
- **data.failures** : contains failed users with detailed error messages.
- **data.success** : contains users processed successfully (if any).
- **message** : indicates how many users failed and succeeded.

```

1 {
2   "data": {
3     "failures": [
4       {
5         "userName": "sample@company.com",
6         "errors": [
7           "error1",
8           "error2"
9         ]
10      }
11    ],
12    "success": []
13  },
14  "message": "Failed to process 1 user(s).",
15  "responseCode": 200,
16  "status": "COMPLETED"
17 }

```

◦ 3. Status: **SUBMITTED** – Request Queued but Not Yet Processed

- HTTP Status Code: **202 Accepted**
- **data** : **{}** (empty)

- **message** : indicates the request is submitted but not yet started

```
1 {
2   "data": {},
3   "message": "Your request is submitted and yet to be processed. Please try again later.",
4   "status": "SUBMITTED",
5   "responseCode": 202
6 }
```

-  4. Status: **IN_PROGRESS** – Request Currently Being Processed

- HTTP Status Code: **202 Accepted**
- **data** : **{}** (empty)
- **message** : indicates that processing is underway

```
1 {
2   "data": {},
3   "message": "Your request is still being processed. Please try again later.",
4   "status": "IN_PROGRESS",
5   "responseCode": 202
6 }
```

-  5. Status: **FAILED** – Unexpected Error Encountered

- HTTP Status Code: **200 OK**

Even though the original operation failed, the Tracking API will **not return a 500** status. This ensures the client receives a consistent response structure and can determine the final status (**FAILED**) from the payload itself. The original request failed during processing due to an unexpected internal error.

- **data** : **{}** (empty)
- **status** : **"FAILED"**

- **responseCode** : **500** (indicating the failure of the **original request**)
- **message** : a generic message indicating an internal error

```

1 {
2   "data": {},
3   "message": "An error occurred while processing your request. Please contact system administrator.",
4   "status": "FAILED",
5   "responseCode": 500
6 }
7

```

◦ ? 6. Status: **NOT_FOUND** – Invalid or Unknown Tracking ID

- HTTP Status Code: **404 Not Found**
- **data** : **{}** (empty)
- **message** : indicates that the provided **trackingId** is invalid or no longer available

```

1 {
2   "data": {},
3   "message": "Tracking ID not found.",
4   "status": "NOT_FOUND",
5   "responseCode": 404
6 }

```

✓ Updated Summary Table

Status	Http Status Code	Data Field	Response Code (Body)	Message Field
COMPLETED	200 OK	success and/or failure s	200	Summary of processed users

SUBMITTED	202 Accepted	{}	202	Request submitted and queued
IN_PROGRESS	202 Accepted	{}	202	Request is being processed
FAILED	200 OK	{}	500	Unexpected error occurred during request processing
NOT_FOUND	404 Not Found	{}	404	Tracking ID not found

5.8 Unauthorized Access

If the requester is **not authorized** to perform a specific action, all APIs will return a **403 Forbidden** response with the following structure.

```

1 {
2   "trackingId": null,
3   "message": null,
4   "responseCode": 403,
5   "userErrors": [
6     {
7       "username": "sample@company.com",
8       "errors": [
9         "You are not allowed to <action performed> user(s)"
10      ]
11    }
12  ]
13 }
```

6. Field Mapping with Lookup APIs

6.1 Overview

When creating or updating a user, certain fields in the request payload must use predefined values retrieved from the **Static Lookup API** and **GLN-Specific Lookup API**. These APIs provide the list of acceptable **key** values for various dropdowns or selection fields.

- The **key** must be sent in the API request.

- The `value` is meant for display purposes only (e.g., UI labels/dropdowns).
- For offerings/services such as **Product Introduction**, **Item Management**, **Digital Catalog** and **Community Portal**, specific nested objects (`piDetails` , `imDetails` , `dcDetails` and `communityDetails` respectively) must be included based on the selected offerings.

6.2 🌱 Offering-Specific Payload Structure

When a particular offering is selected in the `offering.offerings[]` array, the corresponding detail object must be passed with required subfields:

Offering (key)	Offering(va lue)	Payload Object	Fields
Initiate	Product Introduction	piDetails	Mandatory: • <code>piDetails.roles[] .role</code> Optional: • <code>piDetails.roles[] .cascadedGlns</code> • <code>piDetails.permissions[]</code>
I2DataCatalog	Item Management	imDetails	Mandatory: • <code>imDetails.userRole</code> • <code>imDetails.preferredItemView</code> Optional: • <code>imDetails.preferredTargetMarke</code>

			• <code>imDetails.alwaysDefaultPreferredTargetMarket</code> • <code>imDetails.cascadeEnabled</code> • <code>imDetails.cascadedGlns</code> • <code>imDetails.preferredDefaultPlayLists[]</code> Conditional Mandatory: • If <code>cascadeEnabled = true</code>, then <code>cascadedGlns</code> is required
salesKit	Digital Catalog	dcDetails	Mandatory: • <code>dcDetails.userRole</code>
community	Community Portal	communityDetails	Optional: • <code>communityDetails.roles</code>

6.3 Payload Field → Lookup Field Mapping

- ♦ From Static Lookup API (`/user/lookupValues`)

Payload Field	Lookup Field
prefix	prefixes
state	states
country	countries
locale	locales
currency	currencies
umaRole	umaRoles
offering.offerings[]	offerings
offering.imDetails.userRole	imRoles
offering.imDetails.preferredItemView	preferredItemViews
offering.imDetails.preferredTargetMarket	preferredTargetMarkets
offering.dcDetails.userRole	dcRoles
offering.communityDetails.roles[]	communityRoles

- ♦ From GLN-Specific Lookup API (`/user/glnLookUpValues?gln=XXXXXXXXXXXX`)

Payload Field	GLN Lookup Field
offering.imDetails.cascadedGlns[]	imCascadeGlns
offering.imDetails.preferredDefaultPlaylists[]	piLists
offering.piDetails.roles[].role	piRoles
offering.piDetails.roles[].cascadedGlns[]	piCascadeGlns
offering.piDetails.permissions[]	piPermissions

6.4 📌 Key Notes

- Always use the **key** values when making API requests. The **value** and **portal** (if present) are used only for display in UI.

- If no `key-value` structure is present (e.g., just a list of Strings), directly pass the string value (e.g., in `cascadedGlns`).
- The `portal` field in `piRoles` and `piPermissions` helps identify which organization the role/permission belongs to—used for Display purpose.

 Example Reference

If `offering.offerings` includes:

```
1 "offerings": ["Initiate", "I2DataCatalog", "salesKit", "community"]
```

Then, include:

- `piDetails`
- `imDetails`
- `dcDetails`
- `communityDetails`

7. Validations

7.1 Mandatory Validation

	Use Case	Validation Message
1	User Name is blank	User Name is required
2	User GLN is blank	User GLN is required
3	First Name is blank	First Name is required
4	Last Name is blank	Last Name is required
5	Email ID is blank	Business Email Address is required
6	Address is blank	Business Address is required
7	City is Blank	Business City is required
8	Country is blank	Business Country is required
9	Postal Code is blank	Business Postal Code is required
10	Locale is blank	Locale is required

11	UMA Role is blank	UMA Role is required
----	-------------------	----------------------

7.2 Conditional Mandatory Validations

	Use Case	Validation Message
1	Giving Item Management offering without Role	User Role is a mandatory field if Item Management Offering is selected
2	Giving Digital Catalog offering without Role	Digital Catalog Role is a mandatory field if Digital Catalog Offering is selected
3	Giving Production Introduction offering without Role	Product Introduction role is mandatory field if Product Introduction Offering is selected
4	Country = US but no State	Business State is a mandatory field if the Business Country selected is United States
5	Country $\neq$ US and Province is blank	Business Province is a mandatory field if the Business Country selected is not United States

7.3 Max Length Validations

	Use Case	Validation Message
1	User Name	User Name cannot exceed more than 50 characters
2	First Name	First Name cannot exceed more than 128 characters
3	Middle Name	Middle Name cannot exceed more than 10 characters
4	Last Name	Last Name cannot exceed more than 128 characters
5	Title	Title cannot exceed more than 64 characters

6	Email ID	Email ID cannot exceed more than 128 characters
7	Phone Number	Telephone number cannot exceed more than 30 characters
8	Mobile Number	Mobile number cannot exceed more than 30 characters
9	Fax Number	Fax number cannot exceed more than 20 characters
10	Address1	Business Address cannot exceed more than 128 characters
11	Address2	Business Address2 cannot exceed more than 128 characters
12	City	Business City cannot exceed more than 128 characters
13	Province	Business Province cannot exceed more than 128 characters
14	Postal Code	Business Postal Code cannot exceed more than 50 characters

7.4 Invalid Data validations

	Use Case	Validation Message
1	Invalid GLN (not 13 digits)	GLN is a mandatory field. It should be a valid 13 digit number
2	Invalid Prefix	Invalid Prefix provided: mr_1
3	Invalid Country	Invalid Country provided: US_Test
4	Invalid State	Invalid State provided: IL_Test
5	Invalid Locale	Invalid Locale provided: 1001_1
6	Invalid Currency	Invalid Currency provided: USD_Test
7	Invalid UMA Role	Invalid UMA Role provided: ORG_ADMIN_Test

8	Invalid Email Format	Business Email Address is not a valid email
9	Invalid Login ID (username not in email format)	The Login ID is not valid. The Login ID should be a valid email
10	Invalid offerings	Invalid offering(s) provided: workflow_test, community_test
11	Invalid Item Management Role	Invalid Item Management Role provided: ItemManager_Test
12	invalid Catalog Attribute View	Invalid Preferred Item Attribute View provided: quickEntry_Test
13	Invalid Preferred Target Market	Invalid Preferred Target Market provided: US_Test
14	Invalid Default Playlists	Invalid Playlist(s) provided: playlist_test
15	Invalid Digital Catalog Role	Invalid Digital Catalog Role provided: DCLIST_Test
16	Digital Catalog Offering without Item Management Offering	Digital Catalog Offering cannot be given without Item Management offering
17	Item Management Role as 'Digital Catalog Only User' but org doesn't have Digital Catalog Offering	User cannot be assigned this role. The organization must first have the Digital Catalog offering
18	Invalid community Role	Invalid Community Role(s) provided: Community_Test
19	More than 3 Community Roles selected	You can select a maximum of 3 community roles
20	Invalid Product Introduction Role	Invalid Product Introduction Role(s) provided: KRO_VENDOR_ENTRY_Test, DEC_VENDOR_READONLY_Test

21	Invalid Product Introduction Permissions	Invalid Product Introduction Permission(s) provided: SDL_HIERARCHY_Test
22	Country ≠ US but State is selected	Business State should not be populated if the Business Country is not United States
23	First Name with invalid characters (< , >)	First Name should not have > or < character
24	Last Name with invalid characters (< , >)	Last Name should not have > or < characters
25	More than 10 default playlists selected	You can select a maximum of 10 default playlists

7.5 Other Validation messages

	Use Case	Validation Message
1	Username duplication	User already exists
2	Offering not present for org	You cannot assign the following service offering(s) because the organization 3442322222222 does not have them: workflow
3	Changing GLN in Edit	User GLN cannot be modified. This user is already registered with: 3442322222222
4	Changing UserName in Edit	User doesn't exist or User Name cannot be modified
5	Batch user already exists	Batch user already exists for this organization
6	Cascade enabled but no Child GLNs	Item Management Cascade access requires at least one GLN to be specified when cascade is enabled

7	Cascade GLNs not in APP_ID GLN hierarchy	Item Management Cascade access cannot be granted to the following GLNs, as they do not belong to the child hierarchy of the user's GLN(344232222222): XXXXXXXXXXXXXX, XXXXXXXXXXXXXX
8	Product Introduction Cascade GLNs not in APP_ID GLN hierarchy	Product Introduction cascade access cannot be granted to the following GLNs, as they do not belong to the child hierarchy of the user's GLN (8778283884823): XXXXXXXXXXXXXX, XXXXXXXXXXXXXX
9	Creating/Updating user under a GLN that has no service offerings	The GLN 2332425352357 under which you are creating the user does not have any service offerings configured
10	User GLN is Solution Partner or Transora HQ, but IM Role is not Super User/Super User Read-only	The Item Management role is invalid
11	Viewing user details where User GLN is not same as APP ID GLN or its child GLNs	You are not allowed to view user details
12	Creating user where User GLN is not same as APP ID GLN or its child GLNs	You are not allowed to create user(s)
13	Updating user where User GLN is not same as APP ID GLN or its child GLNs	You are not allowed to update user(s)
14	Changing status of user where User GLN is not same as APP ID GLN or its child GLNs	You are not allowed to change the status of user(s)

15	Deleting user where User GLN is not same as APP ID GLN or its child GLNs	You are not allowed to delete user(s)
16	No UserNames provided for status update	Users are not provided
17	No UserNames provided for delete	Users are not provided

8. 🚨 Error Codes & Responses

Status Code	Reason	Description
200	OK	Operation successful
202	Accepted	Request Accepted
204	No Content	No Data available in our System
400	Bad Request	Validation or input error
401	Unauthorized	Invalid or missing token
403	Forbidden	Not authorized to perform operation/action
404	Not Found	URL doesn't exist
500	Internal Server Error	Unexpected server-side error

9. Postman Collection

This postman collection is for aforementioned APIs.

