S2BV4 UAAS

**Issue / Amendment Record**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Project ID** | |  | | | |
| **Project Name** | | S2BV4 UAAS – Interface Spécification Document | | | |
| **Document Status** | | Draft / ~~Final~~ | | | |
| **Version** | **Date** | **Author** | **Reviewer** | **Approver** | **Version Summary** |
| 1.0 | 4 May 2015 | STAR TEAM |  |  | First release |
| 1.12 | 6 Oct 2015 | DT |  |  | Modifications for consistency |
| 1.13 | 8 Oct 2015 | DT |  |  | Field modification in Transaction Auth Module, remove retired password types |
| 1.14 | 02 Nov 2015 | DT |  |  | Return Initial Pin & Unlock Code in decrypt token API,  Update status codes |
| 1.15 | 16 Nov 2015 | DT |  |  | CR related changes |
| 1.16 | 19 Nov 2015 | DT |  |  | Amendment on the new APIs |
| 1.17 | 19 Nov 2015 | DT |  |  | Update on Scope and Table of Contents |
| 1.18 | 20 Nov 2015 | DT |  |  | Update on the APIs after Ganesan’s feedback |
| 1.19 | 23 Nov 2015 | DT |  |  | Updated 7.7 and 7.8 to return public key in response |
| 1.20 | 23 Nov 2015 | DT |  |  | Remove issuerName from 7.4 |
| 1.21 | 05 Feb 2016 | DT |  |  | Updated Modify User |
| 1.22 | 19 Feb 2016 | DT |  |  | Updated Import Token from Encrypted DPX  Add API Import Token from Encrypted DPX(Asynchronous)  Add API Get Token Import Status |
| 1.23 | 04 Mar 2016 | DT |  |  | Update generate DSC ticket  Generate Login Random String (by User)  Generate Login Random String (by admin) |
| 1.24 | 29 Sep 2016 | NCS | Samineni |  | Updated 3.9, 3.10, 3.11 for passwords to be encrypted.  Updated 7.1 and 7.2 for password to be encrypted.  Added 3.15 (Change Password by SMS OTP), 7.12 (Generate Random Challenge), 7.13 (Generate SMS OTP), 7.14 (Authenticate by SMS OTP). |
| 1.25 | 30 Sep 16 | NCS | Samineni |  | Removed Obsolete Sections 4.7 (Get Encrypted Initial PIN by Admin User), 4.8 (Get Encrypted Initial PIN by User), 4.9 (Get Encrypted Unlock Code by Admin User), 4.10 (Get Encrypted Unlock Code by User), 4.13 (Decrypt Token).  Removed 7.12 (Generate Random Challenge).  Updated 3.9, 3.10, 3.11, 3.15, 7.1, and 7.2 to use random challenge from 7.7/7.8.  Added 3.16 (Change Password).  Removed keyIndex field from 3.10 and 7.13. |
| 1.26 | 5 Oct 16 | NCS | Samineni |  | Added appId in 3.7, 3.8, 3.9, 3.10, 3.11, 3.15, 3.16, 3.17, 7.2, 7.5, 7.6 and 7.8.  Updated Old Password as optional field in 3.10, 3.11 and 3.15. |
| 1.27 | 15 Nov 16 | NCS | Samineni |  | Modification on Create user,  API on random password encryption, by passing migration flag. |
| 1.28 | 03 Jan 17 | NCS | Samineni |  | COP Reset user by SMS OTP changes |
| 1.29 | 06 Feb 17 | NCS | Samineni |  | LDAP user lock status, password shouldn’t blank, COP auto unlock, OTP error codes, reset user by SMS OTP audit |
| 1.32 | 20 Feb 17 | NCS | Samineni |  | Enhancement  Reset User (by loginId)  Suspend User(by loginId)  Delete User(by loginId)  Authentication by Password (by User)  Authentication by Password (by admin) |
| 1.33 | 16 March 17 | NCS | Samineni |  | Enhancement new api(s)  1.getUserRegToken  2.verifyUserRegToken  3.getDeviceToken  4.verifyDeviceToken  5.Change password |
| 1.34 | 05 April 17 | NCS | Samineni |  | Enhancement of existed APIs  1.Get Token Challenge (by User)  2.Verify and Retrieve PIN (by User)  3.Generate Login Random String (by User)  4.Generate Login Random String (by Admin user)  5.Authentication by OTP  6.Transaction Authorization by OTP  7.Remote Transaction Authorization by OTP  8.Transaction Authorization Replay by OTP  9.Remote Transaction Authorization Replay by OTP |
| 1.35 | 20 April 17 | NCS | Samineni |  | 1.verifyUserRegToken  2.getDeviceToken  3.Change Password by Session Ticket  4.Change Password  5. Generate SMS OTP  6. Get ThreatMetrix User Data  7. ThreatMetrix Batch job |

Table of Contents

[1. Introduction 9](#_Toc480477931)

[1.1. Scope 9](#_Toc480477932)

[1.2. References 11](#_Toc480477933)

[2. System Configuration Module API 12](#_Toc480477934)

[2.1. Add Application Property 12](#_Toc480477935)

[2.2. Delete Application Property 13](#_Toc480477936)

[2.3. Modify Application Property 14](#_Toc480477937)

[2.4. Get Application Property 15](#_Toc480477938)

[2.5. Add IP Config 16](#_Toc480477939)

[2.6. Delete IP Config 17](#_Toc480477940)

[2.7. Modify IP Config 18](#_Toc480477941)

[2.8. Get IP Config 19](#_Toc480477942)

[3. User Management Module API 20](#_Toc480477943)

[3.1. Create User 20](#_Toc480477944)

[3.2. Modify User 22](#_Toc480477945)

[3.3. Reset User 23](#_Toc480477946)

[3.4. Suspend User 24](#_Toc480477947)

[3.5. On-Hold User 25](#_Toc480477948)

[3.6. Delete User (by Admin User) 26](#_Toc480477949)

[3.7. Reset User by OTP 27](#_Toc480477950)

[3.8. Reset User by DSC 29](#_Toc480477951)

[3.9. Change Password by Session Ticket 30](#_Toc480477952)

[3.10. Change Password by OTP 31](#_Toc480477953)

[3.11. Change Password by DSC 33](#_Toc480477954)

[3.12. Query User Status (by Admin user) 35](#_Toc480477955)

[3.13. Query User Status (by User) 36](#_Toc480477956)

[3.14. Query User Device Status 37](#_Toc480477957)

[3.15. Change Password by SMS OTP 39](#_Toc480477958)

[3.16. Change Password 40](#_Toc480477959)

[3.17. Reset User by SMS OTP 41](#_Toc480477960)

[4. Token Management Module API 43](#_Toc480477961)

[4.1. Import Token from Encrypted DPX 43](#_Toc480477962)

[4.2. Import Token from Encrypted DPX (Asynchronous) 44](#_Toc480477963)

[4.3. Get Token Import Status 45](#_Toc480477964)

[4.4. Modify Token Status 47](#_Toc480477965)

[4.5. Get Token Status by User 48](#_Toc480477966)

[4.6. Get Token Status by Token 49](#_Toc480477967)

[4.7. Get Token Challenge (by User) 50](#_Toc480477968)

[4.8. Verify and Retrieve PIN (by User) 50](#_Toc480477969)

[4.9. Link Token 51](#_Toc480477970)

[4.10. Unlink Token 52](#_Toc480477971)

[5. DSC Management Module API 54](#_Toc480477972)

[5.1. Preregister Certificate 54](#_Toc480477973)

[5.2. Register Certificate 55](#_Toc480477974)

[5.3. Delete Certificate 56](#_Toc480477975)

[5.4. Update Certificate Status 57](#_Toc480477976)

[5.5. Enquiry Certificate Status (by Admin User) 58](#_Toc480477977)

[5.6. Enquiry Certificate Status (by User) 60](#_Toc480477978)

[6. Central Passphrase Module API 62](#_Toc480477979)

[6.1. Generate Passphrase 62](#_Toc480477980)

[6.2. Reset Passphrase (by Admin User) 63](#_Toc480477981)

[6.3. Reset Passphrase (by User) 64](#_Toc480477982)

[6.4. Get Passphrase 64](#_Toc480477983)

[6.5. View Passphrase (by Admin User) 65](#_Toc480477984)

[6.6. View Passphrase (by User) 66](#_Toc480477985)

[7. User Authentication Module API 67](#_Toc480477986)

[7.1. Authentication by Password (by Admin user) 67](#_Toc480477987)

[7.2. Authentication by Password (by User) 68](#_Toc480477988)

[7.3. Authentication by OTP 70](#_Toc480477989)

[7.4. Authentication by DSC 71](#_Toc480477990)

[7.5. Authentication by Password and OTP 72](#_Toc480477991)

[7.6. Authentication by Password and DSC 73](#_Toc480477992)

[7.7. Generate Login Random String (by Admin user) 74](#_Toc480477993)

[7.8. Generate Login Random String (by User) 74](#_Toc480477994)

[7.9. Generate DSC Ticket String 75](#_Toc480477995)

[7.10. Logout (by Admin user) 76](#_Toc480477996)

[7.11. Logout (by User) 77](#_Toc480477997)

[7.12. Generate SMS OTP 77](#_Toc480477998)

[7.13. Authenticate by SMS OTP 78](#_Toc480477999)

[8. Transaction Authorization Module API 80](#_Toc480478000)

[8.1. Transaction Authorization by OTP 80](#_Toc480478001)

[8.2. Remote Transaction Authorization by OTP 81](#_Toc480478002)

[8.3. Transaction Authorization (with two challenge codes) by OTP 83](#_Toc480478003)

[8.4. Transaction Authorization Replay by OTP 84](#_Toc480478004)

[8.5. Transaction Authorization by DSC 85](#_Toc480478005)

[8.6. Transaction Authorization Replay by DSC 86](#_Toc480478006)

[8.7. Generate Challenge Code 87](#_Toc480478007)

[8.8. Remote Transaction Authorization Replay by OTP 87](#_Toc480478008)

[9. User Migration Module API 89](#_Toc480478009)

[9.1. User Migration 89](#_Toc480478010)

[10. Device Management 91](#_Toc480478011)

[10.1. Get User Registration Token 91](#_Toc480478012)

[10.2. Verify User Registration Token 91](#_Toc480478013)

[10.3. Get Device Token 92](#_Toc480478014)

[10.4. Verify Device Token 93](#_Toc480478015)

[10.1. Get Threat Metrix User Data 94](#_Toc480478016)

[10.2. Threat Metrix Batch job 95](#_Toc480478017)

[APPENDIX A - STATUS CODE 96](#_Toc480478018)

[APPENDIX B – PASSWORD TYPE 97](#_Toc480478019)

[APPENDIX C – USER STATUS 98](#_Toc480478020)

[APPENDIX D – APPLICATION PROPERTIES 99](#_Toc480478021)

# Introduction

This document describes the technical interface specification of the UAAS to be implemented for S2BV4.

## Scope

The system functionality covered by this document includes:

1. Services required for System Configuration Module:
   1. Add Application Property
   2. Delete Application Property
   3. Modify Application Property
   4. Get Application Property
   5. Add IP Config
   6. Delete IP Config
   7. Modify IP Config
   8. Get IP Config
2. Services required for User Management Module :
3. Create User
4. Modify User
5. Reset User
6. Suspend User
7. On-Hold User
8. Delete User (by Admin User)
9. Reset User by OTP
10. Reset User by DSC
11. Change User Password by Session Ticket
12. Change User Password by OTP
13. Change User Password by DSC
14. Query User Status (by Admin User)
15. Query User Status (by User)
16. Query User Device Status
17. Services required for Token Management Module:
    1. Import Token from Encrypted DPX
    2. Import Token from Encrypted DPX (Asynchronous)
    3. Get Token Import Status
    4. Modify Token Status
    5. Get Token Status by User
    6. Get Token Status by Token
    7. Get Token Challenge (by User)
    8. Verify and Retrieve PIN (by User)
    9. Link Token
    10. Unlink Token
18. Services required for DSC Management Module:
    1. Preregistered Certificate
    2. Register Certificate
    3. Delete Certificate
    4. Update Certificate Status
    5. Enquiry Certificate Status (by Admin User)
    6. Enquiry Certificate Status (by User)
19. Services required for Central Passphrase Module:
    1. Generate Passphrase
    2. Reset Passphrase (by Admin User)
    3. Reset Passphrase (by User)
    4. Get Passphrase
    5. View Passphrase (by Admin User)
    6. View Passphrase (by User)
20. Services required for User Authentication Module :
21. Authentication by Password (by Admin User)
22. Authentication by Password (by User)
23. Authentication by OTP
24. Authentication by DSC
25. Authentication by Password and OTP
26. Authentication by Password and DSC
27. Generate Login Random String (by Admin User)
28. Generate Login Random String (by User)
29. Generate DSC Ticket String
30. Logout (by Admin User)
31. Logout (by User)
32. Generate Random Challenge
33. Services required for Transaction Authorization Module:
    1. Transaction Authorization by OTP
    2. Remote Transaction Authorization by OTP
    3. Transaction Authorization (with two Challenge Codes) by OTP
    4. Transaction Authorization Replay by OTP
    5. Transaction Authorization by DSC
    6. Transaction Authorization Replay by DSC
    7. Generate Challenge Code (OTP)
    8. Remote Transaction Authorization Replay by OTP
34. User Migration Module
    1. User Migration

## References

NA

# System Configuration Module API

## Add Application Property

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| NA | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| request | JSON Object |  |
| - adminuser | JSON Object |  |
| -- userId | String | Admin user ID |
| -- password | JSON Object |  |
| --- password | String\* | Admin user password |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| - user | JSON Object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| - appProperties | JSON Object |  |
| -- propName | Integer\* |  |
| -- propValue | String Array\* |  |
| --migflg | String (optional) | The migflg by default shall be ‘N’. If they send the value shall consider the input value.  migflg =’Y’ Next group  migflg =’N Classic group |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-) |

## Delete Application Property

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| NA | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| request | JSON Object |  |
| - adminuser | JSON Object |  |
| -- userId | String\* | Admin user ID |
| -- password | JSON Object |  |
| --- password | String\* | Admin user password |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| - user | JSON Object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-) |

## Modify Application Property

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| NA | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| request | JSON Object |  |
| - adminuser | JSON Object |  |
| -- userId | String\* | Admin user ID |
| -- password | JSON Object |  |
| --- password | String\* | Admin user password |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| - user | JSON Object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| - appProperties | JSON Object |  |
| -- propName | Integer\* |  |
| -- propValue | String Array |  |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-) |

## Get Application Property

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| NA | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| request | JSON Object |  |
| - adminuser | JSON Object |  |
| -- userID | String\* | Admin user ID |
| -- password | JSON Object |  |
| --- password | String\* | Admin user password |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| - user | JSON Object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| - appProperties | JSON Object |  |
| -- propName | Integer\* |  |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1] – value of application property |

## Add IP Config

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| NA | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| request | JSON Object |  |
| - adminuser | JSON Object |  |
| -- userID | String\* | Admin user ID |
| -- password | JSON Object |  |
| --- password | String\* | Admin user password |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| - user | JSON Object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| - ipConfig | JSON Object |  |
| -- startIP | String\* | Start IP |
| -- endIP | String\* | End IP |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-) |

## Delete IP Config

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| NA | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| request | JSON Object |  |
| - adminuser | JSON Object |  |
| -- userID | String\* | Admin user ID |
| -- password | JSON Object |  |
| --- password | String\* | Admin user password |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| - user | JSON Object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| - ipConfig | JSON Object |  |
| -- startIP | String\* | Start IP |
| -- endIP | String\* | End IP |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-) |

## Modify IP Config

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| NA | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| request | JSON Object |  |
| - adminuser | JSON Object |  |
| -- userID | String\* | Admin user ID |
| -- password | JSON Object |  |
| --- password | String\* | Admin user password |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| - user | JSON Object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| - oldIPConfig | JSON Object |  |
| -- startIP | String\* | Start IP |
| -- endIP | String\* | End IP |
| - newIPConfig | JSON Object |  |
| -- startIP | String\* | Start IP |
| -- endIP | String\* | End IP |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-) |

## Get IP Config

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| NA | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - adminuser | JSON Object |  |
| -- userID | String\* | Admin user ID |
| -- password | JSON Object |  |
| --- password | String\* | Admin user password |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| - user | JSON Object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1] – List of the IP configured in the system |

# User Management Module API

## Create User

CADM will consume this API for creating new user for S2BV4 on-boarding customers.

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| NA | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - adminuser | JSON Object |  |
| -- appId | String\* | Admin User application ID |
| -- groupId | String\* | Admin User group ID |
| -- userId | String\* | Admin User ID |
| -- password | JSON Object |  |
| --- password | String\* | Admin user password |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| - user | Array of JSON Object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| -- userId | String | User ID |
| -- userType | Integer\* | (1/2) if User Type is 1 –External,2 –Internal |
| -- isLdapAuthenticated | String\* | (Y/N) If LDAP Authenticated flag is –Y Password will not be generated.  If N Password will be generated |
| -- customerName | String\* | User name |
| -- email | String\* | User email |
| -- mobileNo | String (optional) | User mobile number, used to send SMS  Field format:  “<2 Char Country Code> <Full Number with +>“  e.g.:  “SG +6512345678”  Should validate the mobile no as per field format |
| -- phoneNo | String (optional) | User fax number  Field format:  “<2 Char Country Code> <Full Number with +>“  e.g.:  “SG +6512345678” |
| -- dob | String (optional) | User date of birth  Field format:  dd/MM/yyyy |
| -- nationality | String (optional) | User nationality |
| -- address | String (optional) | User address |
| -- countryCode | String \*) | User country (two char country code) |
| -- roleId | Integer\* | User role ID |
| -- remoteTokenUser | String\* | Y/N |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of string | Array of response for each user separated with “|”  [0 … n] – <statusCode> | <appId> | <groupId> | <userId> | <uaasUserId>   |  |  |  | | --- | --- | --- | | statusCode | Integer | refer to [Appendix A](#_APPENDIX_A_-) | | uaasUserId | String | UAAS Random generated UNIQUE ID |   **Note**:  - Random generated password will be sent to customer Registered Email ID or Mobile Number.  If mobile number is not available password will be sent to Email ID via MDIS.  - If created user is an internal LDAP user, status of the user will be set as ACTIVE upon creation. For external users, user status will be FIRSTTIME. |

## Modify User

CADM will consume this API if customer user details need to be modified. This API will support modifying a multiple user in a single request.

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  Allowed user status: FIRSTTIME, ACTIVE, REACTIVATED, 2FA-CHANGED, LOCKED-OUT, ON-HOLD, UNINITIALIZED, DORMANT | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - adminuser | JSON Object |  |
| -- appId | String\* | Admin User application ID |
| -- groupId | String\* | Admin User group ID |
| -- userId | String\* | Admin User ID |
| -- password | JSON Object |  |
| --- password | String\* | Admin user password |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| - user | Array of JSON object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| -- userId | String\* | Old or existing User ID |
| -- customerName | String (optional) | User first name |
| -- email | String (optional) | User email |
| -- mobileNo | String (optional) | User mobile number  Field format:  “<2 Char Country Code> <Full Number with +>“  e.g.:  “SG +6512345678” |
| -- phoneNo | String (optional) | User phone number  Field format:  “<2 Char Country Code> <Full Number with +>“  e.g.:  “SG +6512345678” |
| -- dob | String (optional) | User date of birth  Field format:  dd/MM/yyyy |
| -- nationality | String (optional) | User nationality |
| -- address | String (optional) | User address |
| -- countryCode | String (optional) | User country (Two Char country code) |
| -- roleId | Integer (optional) | User role ID |
| -- remoteTokenUser | String (optional) | Y/N |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of String | Array of response for each user separated with “|”  [0 … n] – <statusCode> | <appId> | <groupId> | <userId> | <uaasUserId>   |  |  |  | | --- | --- | --- | | statusCode | Integer | refer to [Appendix A](#_APPENDIX_A_-) | | uaasUserId | String | UAAS Random generated UNIQUE ID |   **Note:**  - At least 1 user information field should be different from existing value; if none of the fields are modified, UAAS will return error code 154 Star\_Same\_as\_Existing. |

## Reset User

CADM will use this API to reset the user status and password. UAAS will support for multiple user reset in a single request.

As part of reset user, the user password will be reset and any other device (VASCO / DSC Cert) in locked status will be activated.

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  Allowed user status: All user statuses are allowed for this API. | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - adminuser | JSON Object |  |
| -- appId | String\* | Admin User application ID |
| -- groupId | String\* | Admin User group ID |
| -- userId | String\* | Admin User ID |
| -- password | JSON Object |  |
| --- password | String\* | Admin user password |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| - user | Array of JSON object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| -- userId | String\* | User id either login d.  1. First priority : User ID  2. Second priority : login Id  The login id shall be validating the email format.  Remark: Provide either User Id or Login Id while sending the request. |
| -- loginId | String\* |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of String | Array of response for each user separated with “|”  [0 … n] – <statusCode> | <appId> | <groupId> | <userId>  **Note**:  For internal LDAP users, only user status will be set to ACTIVE upon reset.  For external users, password will be reset and sent to user via MDIS. User status will be set to REACTIVATED if user signed in before, otherwise user status will be set to FIRSTTIME. |

## Suspend User

CADM will consume this API if the user needs to be suspended.

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  Allowed user status: All user statuses are allowed except SUSPEND-ON-RQST | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - adminuser | JSON Object |  |
| -- userId | String\* | Admin user ID |
| -- password | JSON Object |  |
| --- password | String\* | Admin user password |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| - user | Array of JSON Object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| -- userId | String\* | User id either login d.  1. First priority : User ID  2. Second priority : login Id  The login id shall be validating the email format.  Remark: Provide either User Id or Login Id while sending the request. |
| -- loginId | String\* |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of String | Array of response for each user separated with “|”:  [0 … n] - <statusCode> | <appId> | <groupId> | <userId>  **Note**:  - User status alone will be updated to SUSPEND-ON-RQST remaining device status like VASCO, DSC status will remain same.  - If user status is already SUSPEND-ON-RQST, UAAS will return error code 154 Star\_Same\_as\_Existing, indicating that the user is already suspended. |

## On-Hold User

CADM will consume this API if active user needs to be ON-HOLD.UAAS will support the API for ON-HOLDING multiple users in a single request.

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  Allowed user status: ACTIVE, FIRSTTIME, REACTIVATED, 2FA-CHANGED | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - adminuser | JSON Object |  |
| -- userId | String\* | Admin user ID |
| -- password | JSON Object |  |
| --- password | String\* | Admin user password |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| - user | Array of JSON Object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| -- userId | String\* | User ID |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of string | Array of String Response separated with “|” delimiter:  [0 … n] - <statusCode> | <appId> | <groupId> | <userId>  **Note:** User Status will be changed to ON-HOLD. |

## Delete User (by Admin User)

CADM will consume this API to delete the user profile in UAAS. It will delete all the profiles tagged to user VASCO and PKI profiles. UAAS will support the API by deleting multiple user in a single request.

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  Allowed user status: SUSPEND-ON-RQST | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - adminuser | JSON Object |  |
| -- userId | String\* | Admin user ID |
| -- password | JSON Object |  |
| --- password | String\* | Admin user password |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| - user | JSON Object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| -- userId | String\* | User id either login d.  1. First priority : User ID  2. Second priority : login Id  The login id shall be validating the email format.  Remark: Provide either User Id or Login Id while sending the request. |
| -- loginId | String\* |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of string | Array of String Response separated with “|” delimiter  [0 … n] - <statusCode> | <appId> | <groupId> | <userId>  **Note:** If all transaction is failed due to common reason (e.g., Database time out, Technical Error, admin authentication failure), UAAS will provide general error codes based on the failure reason. |

## Reset User by OTP

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  User has a linked token.  Allowed user status: DORMANT, SUSPEND-ON-RQST, SUSPEND-ON-RISK, FIRSTTIME, UNINITIALIZED | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON Object |  |
| -- appId | String (optional) | User Application ID |
| -- groupId | String (optional) | User Group ID |
| -- loginId | String\* | User Login ID |
| -- token | JSON Object |  |
| --- otp | String\* | OTP (button 1) |
| --- type | String\* | Token friendly name |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1] – Mobile Number or Email.  **Note:**  User password will be reset and sent to user via MDIS. User status will be set to REACTIVATED if user signed in before, otherwise user status will be set to FIRSTTIME. |

## Reset User by DSC

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  User has a linked DSC.  Allowed user status: DORMANT, SUSPEND-ON-RQST, SUSPEND-ON-RISK, FIRSTTIME, UNINITIALIZED | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON Object |  |
| -- appId | String (optional) | User Application ID |
| -- groupId | String (optional) | User Group ID |
| -- loginId | String\* | User Login ID |
| -- dsc | JSON Object |  |
| --- ticket | String\* | PKI service ticket |
| --- digitalSignMsg | String\* | Message return from PKI applet |
| --- dscType | String\* | DSC friendly name |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1] – Mobile Number or Email.  **Note**:  - User password will be reset and sent to user via MDIS. User status will be set to REACTIVATED if user signed in before, otherwise user status will be set to FIRSTTIME. |

## Change Password by Session Ticket

Using this API customer can change the password using the session ticket passed internally to UAAS.

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  Allowed user status: ACTIVE, FIRSTTIME, REACTIVATED, 2FA-CHANGED | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON Object |  |
| -- appId | String (optional) | User Application ID |
| -- groupId | String (optional) | User Group ID |
| -- loginId | String\* | User Login ID |
| -- password | JSON Object |  |
| --- password | String\* | Session Ticket |
| -- newPassword | JSON Object |  |
| --- password | String\* | New user password  (ClearPassword\_-\_RandomChallenge, encrypted by public key obtained from Generate Login Random String) |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| -- confirmPassword | JSON Object |  |
| --- password | String\* | Confirm user password  (ClearPassword\_-\_RandomChallenge, encrypted by public key obtained from Generate Login Random String) |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  **Note**:  1) User status will be set to ACTIVE if it is not ACTIVE upon password change.  2) In GRP, It will invalidate user registration token & user device token. |

## Change Password by OTP

Using the API customer can change the password using Vasco device.

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  User has a linked token.  Allowed user status: ACTIVE, FIRSTTIME, REACTIVATED, 2FA-CHANGED | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON Object |  |
| -- appId | String (optional) | User Application ID |
| -- groupId | String (optional) | User Group ID |
| -- loginId | String\* | User Login ID |
| -- password | JSON Object |  |
| --- password | String (optional) | Old user password  (ClearPassword\_-\_RandomChallenge, encrypted by public key obtained from Generate Login Random String) |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| -- newPassword | JSON Object |  |
| --- password | String\* | New user password  (ClearPassword\_-\_RandomChallenge, encrypted by public key obtained from Generate Login Random String) |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| -- confirmPassword | JSON Object |  |
| --- password | String\* | Confirm user password  (ClearPassword\_-\_RandomChallenge, encrypted by public key obtained from Generate Login Random String) |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| -- token | JSON Object |  |
| --- otp | String\* | OTP (button 1) |
| --- type | String\* | Token friendly name |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  **Note**:  User status will be set to ACTIVE if it is not ACTIVE upon password change. |

## Change Password by DSC

Using this API customer can change the password using DSC device.

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  User has a linked DSC.  Allowed user status: ACTIVE, FIRSTTIME, REACTIVATED, 2FA-CHANGED | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON Object |  |
| -- appId | String (optional) | User Application ID |
| -- groupId | String (optional) | User Group ID |
| -- loginId | String\* | User Login ID |
| -- password | JSON Object |  |
| --- password | String (optional) | Old user password  (ClearPassword\_-\_RandomChallenge, encrypted by public key obtained from Generate Login Random String) |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| -- newPassword | JSON Object |  |
| --- password | String\* | New user password  (ClearPassword\_-\_RandomChallenge, encrypted by public key obtained from Generate Login Random String) |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| -- confirmPassword | JSON Object |  |
| --- password | String\* | Confirm user password  (ClearPassword\_-\_RandomChallenge, encrypted by public key obtained from Generate Login Random String)) |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| -- dsc | JSON Object |  |
| --- ticket | String\* | PKI service ticket |
| --- digitalSignMsg | String\* | Message return from PKI applet |
| --- dscType | String\* | DSC friendly name |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  **Note**:  User status will be set to ACTIVE if it is not ACTIVE upon password change. |

## Query User Status (by Admin user)

CADM will consume this API to query the user status from UAAS.

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system. | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | Array of JSON object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| -- userId | String\* | User id either login d.  1. First priority : User ID  2. Second priority : login Id  The login id shall be validating the email format.  Remark: Provide either User Id or Login Id while sending the request. |
| -- loginId | String\* |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | Array of JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  Following detail will be returned if user or group exists in the system:  [1 … n]- <appId> | <groupId> | <userId | <userStatus> |

**Note**:

- User status will be retrieved based on the appId, groupId combination or appId, groupId, and userId combination.

- Validation in place to query maximum of 400 Records.

## Query User Status (by User)

Interface system will consume this API to query the user status details from UAAS. It will fetch all the profiles tagged to the user like VASCO, PKI DSC.

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system. S2B NG is passing only loginId currently. | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON object |  |
| -- groupId | String (optional) | User Group ID |
| -- userId | String\* | User id either login d.  1. First priority : User ID  2. Second priority : login Id  The login id shall be validating the email format.  Remark: Provide either User Id or Login Id while sending the request. |
| -- loginId | String\* |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  Following detail will be returned if user exists in the system:  [1] – User Status  Following detail will be returned if user/device exists in the system:  If *deviceType* is “VASCO”;  [2 … n] - <deviceType> | <serialNumber> | <vascoFriendlyName> | <status> |||||||||||   |  |  |  | | --- | --- | --- | | deviceType | String | Device Type: VASCO | | serialNumber | String | VASCO Device Serial Number | | vascoFriendlyName | String | VASCO Token Friendly Name | | status | Integer | VASCO Device Status |   If *deviceType* is “DSC”;  [1 … n] - <deviceType> | <serialnumber> | <issuerName> | <status> | <issuerOrganization> | <issuerDN> | <subjectDN> | <holderName> | <holderOrganization> | <keyUsage> | <validFrom> | <validTo> | <certCRLURL> | <issuerName> | <dscStatus>   |  |  |  | | --- | --- | --- | | deviceType | String | Device Type: DSC | | serialNumber | String | DSC Serial Number | | issuerName | String | DSC Issuer Name | | status | Integer | DSC Mapping Status | | issuerOrganization | String | the O field in the issuerDN | | issuerDN | String | the distinguished name of the issuer | | subjectDN | String | the distinguished name of the certificate holder | | holderName | String | the CN field in the subjectDN | | holderOrganization | String | the O field in the subjectDN | | keyUsage | String | bit string | | validFrom | String | the validFrom timestamp in milliseconds | | validTo | String | the validTo timestamp in milliseconds | | certCRLURL | String | CRL distribution point URL | |

## Query User Device Status

Interface system will consume this API to query the user status details from UAAS.It will fetch all the profiles tagged to the user like VASCO, PKI DSC.

API will support for more than one user and provide list of all the security devices linked with user and corresponding status / additional information.

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system. | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | Array of JSON object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| -- userId | String | User ID |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](file:///C:\Users\aerkol\Desktop\SCB-UAAS-ENHANCE\Documents\S2BV4_UAAS_CADM_InterfaceSpecs%20v0%207.docx#_APPENDIX_A_-)  Following detail will be returned if user/device exists in the system:  If *deviceType* is “VASCO”;  [1 … n] - <appId> | <groupId> | <userId> | <deviceType> | <serialNumber> | <vascoFriendlyName> | <status> |||||||||||   |  |  |  | | --- | --- | --- | | appId | String | User App ID | | groupId | String | User Group ID | | userId | String | User ID | | deviceType | String | Device Type: VASCO | | serialNumber | String | VASCO Device Serial Number | | vascoFriendlyName | String | VASCO Token Friendly Name | | status | Integer | VASCO Device Status |   If *deviceType* is “DSC”;  [1 … n] - <appId> | <groupId> | <userId> | <deviceType> | <serialnumber> | <issuerName> | <status> |  <issuerOrganization> | <issuerDN> | <subjectDN> | <holderName> |  <holderOrganization> | <keyUsage> | <validFrom> | <validTo> | <certCRLURL> | <issuerName> | <dscStatus>   |  |  |  | | --- | --- | --- | | appId | String | User App ID | | groupId | String | User Group ID | | userId | String | User ID | | deviceType | String | Device Type: DSC | | serialNumber | String | DSC Serial Number | | issuerName | String | DSC Issuer Name | | status | Integer | DSC Mapping Status | | issuerOrganization | String | the O field in the issuerDN | | issuerDN | String | the distinguished name of the issuer | | subjectDN | String | the distinguished name of the certificate holder | | holderName | String | the CN field in the subjectDN | | holderOrganization | String | the O field in the subjectDN | | keyUsage | String | bit string | | validFrom | String | the validFrom timestamp in milliseconds | | validTo | String | the validTo timestamp in milliseconds | | certCRLURL | String | CRL distribution point URL | |

**Note:**

**-** User Device details will be retrieved based on the userAppId, userGroupId combination or userAppId, userGroupId, userId combination.

- Validation in place to query maximum of 400 Records.

## Change Password by SMS OTP

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  Allowed user status: ACTIVE, FIRSTTIME, REACTIVATED, 2FA-CHANGED | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON Object |  |
| -- appId | String (optional) | User Application ID |
| -- groupId | String (optional) | User Group ID |
| -- loginId | String\* | User Login ID |
| -- password | JSON Object |  |
| --- password | String (optional) | Old user password  (ClearPassword\_-\_RandomChallenge, encrypted by public key obtained from Generate Login Random String) |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| -- newPassword | JSON Object |  |
| --- password | String\* | New user password  (ClearPassword\_-\_RandomChallenge, encrypted by public key obtained from Generate Login Random String) |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| -- confirmPassword | JSON Object |  |
| --- password | String\* | Confirm user password  (ClearPassword\_-\_RandomChallenge, encrypted by public key obtained from Generate Login Random String) |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| -- otpInfo | JSON Object |  |
| --- otpSn | String\* | OTP serial number |
| --- encOtp | String\* | OTP value  (ClearOTP\_-\_RandomChallenge, encrypted by public key obtained from Generate Login Random String) |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  **Note**:  User status will be set to ACTIVE if it is not ACTIVE upon password change. |

## Change Password

Using this API customer can change the password upon their first login or whenever the user wants.

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  Allowed user status: ACTIVE, FIRSTTIME, REACTIVATED, 2FA-CHANGED | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON Object |  |
| -- appId | String (optional) | User Application ID |
| -- groupId | String (optional) | User Group ID |
| -- loginId | String\* | User Login ID |
| -- password | JSON Object |  |
| --- password | String\* | Old user password  (ClearPassword\_-\_RandomChallenge, encrypted by public key obtained from Generate Login Random String) |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --newPassword | JSON Object |  |
| --- password | String\* | New user password  (ClearPassword\_-\_RandomChallenge, encrypted by public key obtained from Generate Login Random String) |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| -- confirmPassword | JSON Object |  |
| --- password | String\* | Confirm user password  (ClearPassword\_-\_RandomChallenge, encrypted by public key obtained from Generate Login Random String) |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  **Note**:  User status will be set to ACTIVE if it is not ACTIVE upon password change. |

## Reset User by SMS OTP

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  Allowed user status: DORMANT, SUSPEND-ON-RQST, SUSPEND-ON-RISK, FIRSTTIME, UNINITIALIZED, LOCKED-OUT The user status shall applicable to other then COP users.  Only for COP Allowed user status: DORMANT, SUSPEND-ON-RQST, SUSPEND-ON-RISK, FIRSTTIME, UNINITIALIZED,ACTIVE, LOCKED-OUT | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON Object |  |
| -- appId | String\* | User Application ID (Ex:’IDC’, ‘COP’) as per configuration |
| -- groupId | String\* | User Group ID |
| -- loginId | String\* | User Login ID (must be external user only) should be email format. |
| - otpInfo | JSON Object |  |
| -- otpSn | String\* | OTP serial number |
| -- encOtp | String\* | OTP value  (ClearOTP\_-\_RandomChallenge, encrypted by public key obtained from Generate Login Random String) |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1] – Mobile Number or Email.  **Note:**  User password will be reset and sent to user via MDIS. User status will be set to REACTIVATED if user signed in before, otherwise user status will be set to FIRSTTIME. |

# Token Management Module API

## Import Token from Encrypted DPX

CADM will consume this API to import the DPX token and EPV pin. UAAS will provide the number of token file processed and details of the token like friendly name of the token. This API only accepts data from multipart form.

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| NA | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| dpxFile | FilePart | contentType = "application/octet-stream" |
| evpFile | FilePart | contentType = "application/octet-stream" |
| jsonPart | StringPart | content type = “application/json” |
| -request | JSON Object |  |
| -- adminuser | JSON Object |  |
| --- appId | String\* | Admin User application ID |
| --- groupId | String\* | Admin User group ID |
| --- userId | String\* | Admin User ID |
| --- password | JSON Object |  |
| ---- password | String\* | Admin user password |
| ---- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| ---- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| ---- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| -- tokenFile | JSON Object |  |
| --- dpxFile | String\* | DPX token file path (Base64 encoded) |
| --- dpxEncTsptKey | String\* | DPX Transport Key encrypted with KEK |
| --- dpxTsptKeyKCV | String\* | Key check valued of DPX Transport Key in hexadecimal |
| --- epvFile | String\* | EPV initial pin file path (Base64 encoded) |
| --- epvEncTsptKey | String\* | EPV Transport Key encrypted with KEK |
| --- epvTsptKeyKCV | String\* | Key check value of EPV Transport Key in hexadecimal |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1] – Number of token profiles processed  [2 … n] – List of the token detail and status for each token:  <tokenSerialNo> | <tokenAppName> | <status> | <tokenFriendlyName>   |  |  |  | | --- | --- | --- | | tokenSerialNo | String | VASCO Device Serial Number | | tokenAppName | String | VASCO Device Application Name | | status | Boolean | Import operation status (true/false) | | tokenFriendlyName | String | VASCO Device Friendly Name |   **Note**: Token Friendly name should be passed during Link Token, Unlink Token API. And it should be maintained in CADM with respective tokens.  ex: 2900593712 | I\_APPLI1 | true | GLOBAL\_TOKEN |

## Import Token from Encrypted DPX (Asynchronous)

CADM will consume this API to import the DPX token and EPV pin. UAAS will provide the import token session ID for query the import status. This API only accept data from multipart form.

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| NA | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| dpxFile | FilePart | contentType = "application/octet-stream" |
| evpFile | FilePart | contentType = "application/octet-stream" |
| jsonPart | StringPart | content type = “application/json” |
| -request | JSON Object |  |
| -- adminuser | JSON Object |  |
| --- appId | String\* | Admin User application ID |
| --- groupId | String\* | Admin User group ID |
| --- userId | String\* | Admin User ID |
| --- password | JSON Object |  |
| ---- password | String\* | Admin user password |
| ---- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| ---- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| ---- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| -- tokenFile | JSON Object |  |
| --- dpxFile | String\* | DPX token file path (Base64 encoded) |
| --- dpxEncTsptKey | String\* | DPX Transport Key encrypted with KEK |
| --- dpxTsptKeyKCV | String\* | Key check valued of DPX Transport Key in hexadecimal |
| --- epvFile | String\* | EPV initial pin file path (Base64 encoded) |
| --- epvEncTsptKey | String\* | EPV Transport Key encrypted with KEK |
| --- epvTsptKeyKCV | String\* | Key check value of EPV Transport Key in hexadecimal |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1] – Token Import Session ID |

## Get Token Import Status

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| Token is imported to the system. | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - adminuser | JSON Object |  |
| -- appId | String\* | Admin User application ID |
| -- groupId | String\* | Admin User group ID |
| -- userId | String\* | Admin User ID |
| -- password | JSON Object |  |
| --- password | String\* | Admin user password |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| - tokenImportSessionId | String | Get from API 4.2 Import Token from Encrypted DPX (Asyncronous) |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1] – Token Import Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [2] – Number of token profiles processed  [3 … n] – List of the token detail and status for each token:  <tokenSerialNo> | <tokenAppName> | <status> | <tokenFriendlyName>   |  |  |  | | --- | --- | --- | | tokenSerialNo | String | VASCO Device Serial Number | | tokenAppName | String | VASCO Device Application Name | | status | Boolean | Import operation status (true/false) | | tokenFriendlyName | String | VASCO Device Friendly Name |   **Note**: Token Friendly name should be passed during Link Token, Unlink Token API. And it should be maintained in CADM with respective tokens.  ex: 2900593712 | I\_APPLI1 | true | GLOBAL\_TOKEN |

## Modify Token Status

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| Token is imported to the system. | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - adminuser | JSON Object |  |
| -- appId | String\* | Admin User application ID |
| -- groupId | String\* | Admin User group ID |
| -- userId | String\* | Admin User ID |
| -- password | JSON Object |  |
| --- password | String\* | Admin user password |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| - token | JSON Object |  |
| -- serialNo | String\* | Token Serial Number |
| -- status | String\* | New token status |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-) |

## Get Token Status by User

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system. | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| request | JSON Object |  |
| - user | JSON Object |  |
| -- groupId | String (optional) | User Group ID |
| -- loginId | String\* | User Login ID |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1 … n] – <vascoFriendlyName> | <status> | <lastUsedDate>   |  |  |  | | --- | --- | --- | | vascoFriendlyName | String | VASCO Token Friendly Name | | status | Integer | VASCO Device Status | | lastUsedDate | Long | Last used date in timestamp format (miliseconds) | |

## Get Token Status by Token

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| NA. | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - token | JSON Object |  |
| -- serialNo | String\* | Token Serial No |
| -- appName | String\* | Token application name |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1] – Token status  [2] – Last used date in timestamp format  [3] – User count  [4] – List of users’ loginId linked to this token separated with “|”:  <user1’s loginId> | <user2’s loginId> | … | <user n’s loginId> |

## Get Token Challenge (by User)

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system. | | |
| Input Parameters | | |
| API Name /GetTokenChallenge | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON Object |  |
| -- groupId | String (optional) | User Group Id |
| -- loginId | String\* | User id either login d.  Remark: Provide either User Id or Login Id while sending the request. |
| --userid | String\* |
| - token | JSON Object |  |
| -- serialNo | String\* | Token serial number |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1] – VASCO Token Challenge |

## Verify and Retrieve PIN (by User)

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system. | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON Object |  |
| -- groupId | String (optional) | User Group Id |
| -- loginId | String\* | User id either login d.  Remark: Provide either User Id or Login Id while sending the request. |
| --user id | String\* |
| - token | JSON Object |  |
| -- unlockChallenge | String (optional) | Challenge code to unlock the token displayed on the token.  00000 or NULL is used to get initial pin. |
| -- encryptedString | String\* | VASCO Token challenge |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1] – Initial PIN or unlock PIN |

## Link Token

CADM will consume this API to link the token to the userId. CADM should pass the token friendly name.

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system. | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - adminuser | JSON Object |  |
| -- appId | String\* | Admin User application ID |
| -- groupId | String\* | Admin User group ID |
| -- userId | String\* | Admin User ID |
| -- password | JSON Object |  |
| --- password | String\* | Admin user password |
| --- type | Integer | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| - user | Array of JSON Object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| -- userId | String\* | User ID |
| - token | Array of JSON Object |  |
| -- serialNo | String\* | Token serial number |
| -- type | String\* | Token friendly name |
| -- encryptedString | String\* | Encyrpted String (Y/N)  **Note:** If *encryptedString* is “Y”, UAAS will send the Encrypted String to customer’s e-mail via MDIS.  If flag is “N”, UAAS will not generate encrypted String. |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of string | Array of String response delimited with “|”  [0 … n] - <statusCode> | <appId> | <groupId> | <userId> | <tokenSerialNo>   |  |  |  | | --- | --- | --- | | statusCode | String | Status Code, refer to [Appendix A](#_APPENDIX_A_-) | | appId | String | User application ID | | groupId | String | User group ID | | userId | String | User ID | | tokenSerialNo | String | Token serial number |   **Note**:  - Token Status will be changed to ACTIVATE.  **-** If *Encrypted String* is “Y”, UAAS will send the *Encrypted String* to customer’s e-mail via MDIS.  If flag is “N”, UAAS will not generate encrypted String.  - User |

## Unlink Token

CADM will consume the API to unlink the token to the user. CADM should pass token friendly name in the input parameter. This service should support multiple user Unlinking the token in a single request.

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  User has token assigned to them. | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - adminuser | JSON Object |  |
| -- appId | String\* | Admin User application ID |
| -- groupId | String\* | Admin User group ID |
| -- userId | String\* | Admin User ID |
| -- password | JSON Object |  |
| --- password | String\* | Admin user password |
| --- type | Integer | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| - user | Array of JSON Object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| -- userId | String\* | User ID |
| - token | Array of JSON Object |  |
| -- serialNo | String\* | Token serial number |
| -- type | String\* | Token Friendly Name |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of string | Array of String response delimited with “|”  [0 … n] - <statusCode> | <appId> | <groupId> | <userId> | <tokenSerialNo>     |  |  |  | | --- | --- | --- | | statusCode | String | Status Code, refer to [Appendix A](#_APPENDIX_A_-) | | appId | String | User application ID | | groupId | String | User group ID | | userId | String | User ID | | tokenSerialNo | String | Token serial number |   **Note:** Multiple users can be unlinked in a single request. *user* and *token* JSON arrays should be in same order. UAAS should return the status code of each user with details with “|” delimiter. |

# DSC Management Module API

## Preregister Certificate

CADM will consume this API will doing the pre-registration for the user DSC device. UAAS will respond with DSC friendly name and CADM should maintain this value from their end.

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  CA must be authorized by CCA. The serial number must be unique within the CA. | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - adminuser | JSON Object |  |
| -- appid | String\* | Admin User application ID |
| -- groupid | String\* | Admin User group ID |
| -- userId | String\* | Admin User ID |
| -- password | JSON Object |  |
| --- password | String\* | Admin user password |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| - user | JSON Object |  |
| -- appid | String\* | User application ID |
| -- groupid | String\* | User group ID |
| -- userId | String\* | User ID |
| -- dscArray | Array of JSON Object |  |
| --- issuerName | String\* | issuer Name – pre-defined CA name  i.e. “cca\_india” , “e-mudhra” , “ncodesolutions” , “safescrypt” , “idrbt” , “tcs” |
| --- publicCertSN | String\* | Serial number of the X509 certificate in upper case Hex string, without 0x in the front. If the first digit is 0, remove it as well, e.g., use CAB59 instead of 0CAB59 |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of string | The result for each pre-registered certificate separated with “|”:  [0 … n] - <statusCode> | <publicCertSN> | <issuerName>  statusCode: refer to [Appendix A](file:///C:\Users\aerkol\Desktop\SCB-UAAS-ENHANCE\Documents\S2BV4_UAAS_CADM_InterfaceSpecs%20v0%208.docx#_APPENDIX_A_-) |

## Register Certificate

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  CA must be authorized by CCA. The serial number must be unique within the CA.  Allowed user status: FIRSTTIME, ACTIVE, REACTIVATED, 2FA-CHANGED | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON object |  |
| -- groupId | String (optional) | User Group ID |
| -- loginId | String\* | User Login ID |
| -- dsc | JSON object |  |
| --- ticket | String\* | PKI service ticket |
| --- digitalSignMsg | String\* | Client message returned from PKI Applet |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of string | [0] - Status code (refer to [Appendix A](#_APPENDIX_A_-))  [1] – Date of expiry of certificate in milliseconds  [2] – Certificate status  [3]- Issuer Name |

## Delete Certificate

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  CA must be authorized by CCA. The serial number must be unique within the CA. | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| request | JSON Object |  |
| - adminuser | JSON Object |  |
| -- appId | String\* | Admin User application ID |
| -- groupId | String\* | Admin User group ID |
| -- userId | String\* | Admin User ID |
| -- password | JSON Object |  |
| --- password | String\* | Admin user password |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| - user | JSON Object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| -- userId | String\* | User ID |
| -- dscArray | Array of JSON Object |  |
| --- issuerName | String\* | issuer Name – pre-defined CA name  i.e. “cca\_india” , “e-mudhra” , “ncodesolutions” , “safescrypt” , “idrbt” , “tcs” |
| --- publicCertSN | String\* | Serial number of the X509 certificate in upper case Hex string, without 0x in the front. If the first digit is 0, remove it as well, e.g., use CAB59 instead of 0CAB59 |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| response | JSON Object |  |
| - response | Array of string | The result for each certificate separated with “|”:  [0 … n] - <statusCode> | <publicCertSN> | <issuerName>  statusCode: refer to [Appendix A](file:///C:\Users\aerkol\Desktop\SCB-UAAS-ENHANCE\Documents\S2BV4_UAAS_CADM_InterfaceSpecs%20v0%208.docx#_APPENDIX_A_-) |

## Update Certificate Status

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system  CA must be authorized by CCA. The serial number must be unique within the CA. | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| request | JSON Object |  |
| - adminuser | JSON Object |  |
| -- appId | String\* | Admin User application ID |
| -- groupId | String\* | Admin User group ID |
| -- userId | String\* | Admin User ID |
| -- password | JSON Object |  |
| --- password | String\* | Admin user password |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| - user | JSON Object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| -- userId | String\* | User ID |
| -- dscArray | Array of JSON Object |  |
| --- issuerName | String\* | issuer Name – pre-defined CA name  i.e. “cca\_india” , “e-mudhra” , “ncodesolutions” , “safescrypt” , “idrbt” , “tcs” |
| --- publicCertSN | String\* | Serial number of the X509 certificate in upper case Hex string, without 0x in the front. If the first digit is 0, remove it as well, e.g., use CAB59 instead of 0CAB59 |
| --- status | Integer\* | 1: ACTIVE  2: SUSPENDED  3: DELETED |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| response | JSON Object |  |
| - response | Array of string | The result for each certificate with pipe delimiter:  [0 … n] - <statusCode> | <certserialno> | <issuerName>  statusCode: refer to [Appendix A](file:///C:\Users\aerkol\Desktop\SCB-UAAS-ENHANCE\Documents\S2BV4_UAAS_CADM_InterfaceSpecs%20v0%208.docx#_APPENDIX_A_-) |

## Enquiry Certificate Status (by Admin User)

|  |  |  |  |
| --- | --- | --- | --- |
| Prerequisite | | | |
| User is created in the system.  CA must be authorized by CCA. The serial number must be unique within the CA. | | | |
| Input Parameters | | | |
| Variable | Data Type | | Explanation |
| request | | JSON Object |  |
| - user | | JSON Object |  |
| -- appId | | String\* | User application ID |
| -- groupId | | String\* | User group ID |
| -- userId | | String\* | User ID |
| -- dsc | | JSON Object |  |
| --- issuerName | | String\* | issuer Name – pre-defined CA name  i.e. “cca\_india” , “e-mudhra” , “ncodesolutions” , “safescrypt” , “idrbt” , “tcs” |
| --- publicCertSN | | String\* | Serial number of the X509 certificate in upper case Hex string, without 0x in the front. If the first digit is 0, remove it as well, e.g., use CAB59 instead of 0CAB59 |
| Output Parameters | | | |
| Variable | Data Type | | Explanation |
| response | | JSON Object |  |
| - response | | Array of string | [0] - Status code (refer to [Appendix A](#_APPENDIX_A_-))     1. Enquiry API using APPID, GROUPID, USERID   [1] – Issuer Name  [2] – Certificate Serial Number  [3] – Issuer Organisation  [4] – Issuer DN  [5] – Subject DN  [6] – Holder Name  [7] – Holder Organisation  [8] – Key Usage  [9] – Valid From in millisecond format  [10] – Valid To in millisecond format  [11] – Certificate CRL URL  [12] – Status   1. Enquiry API using APPID, GROUPID   [1] – User ID  [2] – Issuer Name  [3] – Certificate Serial Number  [4] – Issuer Organisation  [5] – Issuer DN  [6] – Subject DN  [7] – Holder Name  [8] – Holder Organisation  [9] – Key Usage  [10] – Valid From in millisecond format  [11] – Valid To in millisecond format  [12] – Certificate CRL URL  [13] – Status   1. Enquiry API using APPID, GROUPID, USERID, Issuer Name and certSN   [1] – Issuer Name  [2] – Certificate Serial Number  [3] – Issuer Organisation  [4] – Issuer DN  [5] – Subject DN  [6] – Holder Name  [7] – Holder Organisation  [8] – Key Usage  [9] – Valid From in millisecond format  [10] – Valid To in millisecond format  [11] – Certificate CRL URL  [12] – Status |

## Enquiry Certificate Status (by User)

|  |  |  |  |
| --- | --- | --- | --- |
| Prerequisite | | | |
| User is created in the system.  CA must be authorized by CCA. The serial number must be unique within the CA. | | | |
| Input Parameters | | | |
| Variable | Data Type | | Explanation |
| Request | | JSON Object |  |
| - user | | JSON Object |  |
| -- groupId | | String (optional) | User Group ID |
| -- loginId | | String\* | User Login ID |
| -- dsc | | JSON Object |  |
| --- issuerName | | String\* | issuer Name – pre-defined CA name  i.e. “cca\_india” , “e-mudhra” , “ncodesolutions” , “safescrypt” , “idrbt” , “tcs” |
| --- publicCertSN | | String\* | Serial number of the X509 certificate in upper case Hex string, without 0x in the front. If the first digit is 0, remove it as well, e.g., use CAB59 instead of 0CAB59 |
| Output Parameters | | | |
| Variable | Data Type | | Explanation |
| response | | JSON Object |  |
| - response | | Array of string | [0] - Status code (refer to [Appendix A](#_APPENDIX_A_-))     1. Enquiry API using APPID, GROUPID, USERID   [1] – Issuer Name  [2] – Certificate Serial Number  [3] – Issuer Organisation  [4] – Issuer DN  [5] – Subject DN  [6] – Holder Name  [7] – Holder Organisation  [8] – Key Usage  [9] – Valid From in millisecond format  [10] – Valid To in millisecond format  [11] – Certificate CRL URL  [12] – Status   1. Enquiry API using APPID, GROUPID   [1] – User ID  [2] – Issuer Name  [3] – Certificate Serial Number  [4] – Issuer Organisation  [5] – Issuer DN  [6] – Subject DN  [7] – Holder Name  [8] – Holder Organisation  [9] – Key Usage  [10] – Valid From in millisecond format  [11] – Valid To in millisecond format  [12] – Certificate CRL URL  [13] – Status   1. Enquiry API using APPID, GROUPID, USERID, Issuer Name and certSN   [1] – Issuer Name  [2] – Certificate Serial Number  [3] – Issuer Organisation  [4] – Issuer DN  [5] – Subject DN  [6] – Holder Name  [7] – Holder Organisation  [8] – Key Usage  [9] – Valid From in millisecond format  [10] – Valid To in millisecond format  [11] – Certificate CRL URL  [12] – Status |

# Central Passphrase Module API

## Generate Passphrase

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| NA | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| request | JSON Object |  |
| - adminuser | JSON Object |  |
| -- appId | String\* | Admin User application ID |
| -- groupId | String\* | Admin User group ID |
| -- userId | String\* | Admin User ID |
| -- password | JSON Object |  |
| --- password | String\* | Admin user password |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| - user | JSON Object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1] – User group passphrase |

## Reset Passphrase (by Admin User)

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| NA | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| request | JSON Object |  |
| - adminuser | JSON Object |  |
| -- appId | String\* | Admin User application ID |
| -- groupId | String\* | Admin User group ID |
| -- userId | String\* | Admin User ID |
| -- password | JSON Object |  |
| --- password | String\* | Admin user password |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| - user | JSON Object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1] – User group passphrase  [2] – Effective date in timestamp format |

## Reset Passphrase (by User)

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| NA | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON Object |  |
| -- groupId | String (optional) | User Group ID |
| -- loginId | String\* | User Login ID |
| -- password | JSON Object |  |
| --- password | String\* | Session Ticket |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of string | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1] – User group passphrase  [2] – Effective date in timestamp format |

## Get Passphrase

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| NA | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON Object |  |
| -- appId | String\* | User Application ID |
| -- groupId | String\* | User Group ID |
| -- appCode | String\* | App Code |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1] – User group passphrase |

## View Passphrase (by Admin User)

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| NA | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - adminuser | JSON Object |  |
| -- appId | String\* | Admin User application ID |
| -- groupId | String\* | Admin User group ID |
| -- userId | String\* | Admin User ID |
| -- password | JSON Object |  |
| --- password | String\* | Admin user password |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| - user | JSON Object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1] – List of passphrase histories and effective date  List of Passphrase histories will be listed.  As per BAU. |

## View Passphrase (by User)

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| NA | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| request | JSON Object |  |
| - user | JSON Object |  |
| -- groupId | String (optional) | User Group ID |
| -- loginId | String\* | User Login ID |
| -- password | JSON Object |  |
| --- password | String\* | Session Ticket |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1] – List of passphrase histories and effective date  List of Passphrase histories will be listed.  As per BAU. |

# User Authentication Module API

## Authentication by Password (by Admin user)

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  Allowed user status: FIRSTTIME, ACTIVE, REACTIVATED, 2FA-CHANGED | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| request | JSON Object |  |
| - user | JSON object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| -- userid | String\* | User id either login d.  1. First priority : User ID  2. Second priority : login Id  The login id shall be validating the email format.  Remark: Provide either User Id or Login Id while sending the request. |
| -- loginId | String\* |
| -- password | JSON Object |  |
| --- password | String\* | Admin user password  (ClearPassword\_-\_RandomChallenge, encrypted by public key obtained from Generate Login Random String) |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  Following detail will be returned if user is successfully authenticated:  [1] – Last login date and time in millisecond  [2] – Last failed login in millisecond  [3] – Session ticket |

## Authentication by Password (by User)

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  Allowed user status: FIRSTTIME, ACTIVE, REACTIVATED, 2FA-CHANGED | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| request | JSON Object |  |
| - user | JSON object |  |
| -- appId | String | User Application ID (Eg. ‘IDC’ | ‘COP) |
| -- groupId | String (optional) | User Group ID |
| -- userid | String\* | User id either login d.  1. First priority : User ID  2. Second priority : login Id  The login id shall be validating the email format.  Remark: Provide either User Id or Login Id while sending the request. |
| -- loginId | String\* |
| -- password | JSON Object |  |
| --- password | String\* | User password  (ClearPassword\_-\_RandomChallenge, encrypted by public key obtained from Generate Login Random String) |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  Following detail will be returned if user is successfully authenticated:  [1] – Last login date and time in millisecond  [2] – Last failed login in millisecond  [3] – Session ticket  **Note:**  If user status is FIRSTTIME or REACTIVATED, it will be changed to ACTIVE upon successful authentication. |

## Authentication by OTP

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  User has a linked token.  Allowed user status: FIRSTTIME, REACTIVATED, ACTIVE, 2FA-CHANGED | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON object |  |
| -- appId | String | User Application ID (Eg. ‘IDC’ ) |
| -- groupId | String (optional) | User Group ID |
| -- loginId | String\* | User id either login d.  Remark: Provide either User Id or Login Id while sending the request. |
| --userid | String\* |
| -- password | JSON Object |  |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| -- token | JSON Object |  |
| --- otp | String\* | OTP (button 1) |
| --- type | String\* | Token friendly name |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  Following detail will be returned if user is successfully authenticated:  [1] – Last login date and time in millisecond  [2] – Last failed login in millisecond  [3] – Session ticket  **Note:** If user status is FIRSTTIME or REACTIVATED, it will be changed to ACTIVE upon successful authentication. |

## Authentication by DSC

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  User has a linked DSC.  Allowed user status: FIRSTTIME, REACTIVATED, ACTIVE, 2FA-CHANGED | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON object |  |
| -- appId | String | User Application ID (Eg. ‘IDC’ ) |
| -- groupId | String (optional) | User Group ID |
| -- loginId | String\* | User Login ID |
| -- password | JSON Object |  |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| -- dsc | JSON Object |  |
| --- ticket | String\* | PKI service ticket |
| --- digitalSignMsg | String\* | Message return from PKI applet |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  Following detail will be returned if user is successfully authenticated:  [1] – Last login date and time in millisecond  [2] – Last failed login in millisecond  [3] – Session ticket  **Note:** If user status is FIRSTTIME or REACTIVATED, it will be changed to ACTIVE upon successful authentication. |

## Authentication by Password and OTP

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  User has a linked token.  Allowed user status: FIRSTTIME, ACTIVE, REACTIVATED, 2FA-CHANGED | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON object |  |
| -- appId | String (optional) | User Application ID |
| -- groupId | String (optional) | User Group ID |
| -- loginId | String\* | User Login ID |
| -- password | JSON Object |  |
| --- password | String\* | User password  (ClearPassword\_-\_RandomChallenge, encrypted by public key obtained from Generate Login Random String) |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| -- token | JSON Object |  |
| --- otp | String\* | OTP (button 1) |
| --- type | String\* | Token friendly name |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  Following detail will be returned if user is successfully authenticated:  [1] – Last login date and time in millisecond  [2] – Last failed login in millisecond  [3] – Session ticket |

## Authentication by Password and DSC

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  User has a linked DSC.  Allowed user status: FIRSTTIME, ACTIVE, REACTIVATED, 2FA-CHANGED | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON object |  |
| -- appId | String (optional) | User Application ID |
| -- groupId | String (optional) | User Group ID |
| -- loginId | String\* | User Login ID |
| -- password | JSON Object |  |
| --- password | String\* | User password  (ClearPassword\_-\_RandomChallenge, encrypted by public key obtained from Generate Login Random String) |
| --- type | Integer\* | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| -- dsc | JSON Object |  |
| --- ticket | String\* | PKI service ticket |
| --- digitalSignMsg | String\* | Message return from PKI applet |
| --- issuerName | String\* | DSC Issuer Name |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  Following detail will be returned if user is successfully authenticated:  [1] – Last login date and time in millisecond  [2] – Last failed login in millisecond  [3] – Session ticket |

## Generate Login Random String (by Admin user)

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system. | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| -- userId | String\* | User id either login d.  Remark: Provide either User Id or Login Id while sending the request. |
| --login id | String\* |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1] – Random string  [2] – Public key in <Modulus>|<PublicExponent> format (In HEX String) |

## Generate Login Random String (by User)

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system. | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON object |  |
| -- appId | String | User Application ID |
| -- groupId | String (optional) | User Group ID |
| -- loginId | String\* | User id either login d.  Remark: Provide either User Id or Login Id while sending the request. |
| --userid | String\* |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  Following detail will be returned if user exists:  [1] – Random string  [2] – Public key in <Modulus>|<PublicExponent> format (In HEX String) |

## Generate DSC Ticket String

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  Allowed user status: FIRSTTIME, ACTIVE, REACTIVATED, 2FA-CHANGED | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON object |  |
| -- appId | String | User Application ID (Eg. ‘IDC’ ) |
| -- groupId | String (optional) | User Group ID |
| -- loginId | String\* | User Login ID |
| -- dsc | JSON Object |  |
| --- eventType | Integer\* | 1: Certification Registration  2: User Authentication  3: Transaction Authorization |
| --- txRefNo | String\* | Transaction Ref No |
| --- txnData | String\* | Transaction Data |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1] – Ticket string |

## Logout (by Admin user)

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  User Status is FIRSTTIME or ACTIVE. | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| -- userId | String\* | User ID |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-) |

## Logout (by User)

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  User Status is FIRSTTIME or ACTIVE. | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON object |  |
| -- appId | String | User Application ID (Eg. ‘IDC’ ) |
| -- groupId | String (optional) | User Group ID |
| -- loginId | String\* | User Login ID |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-) |

## Generate SMS OTP

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system. | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| -- loginId | String\* | Login ID (should be email format since it is external user) |
| - smsInfo | JSON Object |  |
| -- msgTemplate | String\* | SMS message template containing placeholder **{otp}** for the OTP value |
| -- sendViaSms | String\* | Flag to indicate whether OTP should be sent via SMS. Possible values are:  - “Y”: OTP will be sent via SMS.  - “N”: OTP will be returned in the response. |
| --isEncoded | String (option) | The value set as ‘Y’ for multi-language support and msgTemplate data to be encode using base64. |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1] – OTP serial number  [2] – OTP prefix  [3] – OTP value |

## Authenticate by SMS OTP

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system. | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User Group ID |
| -- loginId | String\* | User Login ID (must be external user only) should be email format. |
| - otpInfo | JSON Object |  |
| -- otpSn | String\* | OTP serial number |
| -- encOtp | String\* | OTP value  (ClearOTP\_-\_RandomChallenge, encrypted by public key obtained from Generate Login Random String) |
| - ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  Following detail will be returned if user is successfully authenticated:  [1] – Last login date and time in millisecond  [2] – Last failed login in millisecond  [3] – Session ticket  **Note:** If user status is FIRSTTIME or REACTIVATED, it will be changed to ACTIVE upon successful authentication. |

# Transaction Authorization Module API

## Transaction Authorization by OTP

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  User has token assigned to them.  Allowed user status: ACTIVE, 2FA-CHANGED | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON object |  |
| -- appId | String | User Application ID (Eg. ‘IDC’) |
| -- groupId | String (optional) | User Group ID |
| -- loginId | String\* | User id either login d.  Remark: Provide either User Id or Login Id while sending the request. |
| --userId | String\* |
| - transaction | JSON object |  |
| -- txRefNo | String\* | Transaction Ref No |
| -- randomNo | String\* | Random number. Refer to [Get Challenge API](#_Generate_Challenge_Code) |
| -- challengeSize | Integer\* | Challenge Size |
| -- response | String\* | Response from token based on the challenge entered (button 2) |
| -- tokenType | String\* | Token friendly name |
| -- challengeTxData | String\* | Transaction data from the application used to produce the challenge |
| -- MACTxData | String\* | Transaction data from the application used to produce the MAC |
| -- clientMAC | String\* | MAC produced on the application side |
| -- timestamp | Long\* | Date and time in seconds |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1] – The number of transaction authorization failed attempts left. |

## Remote Transaction Authorization by OTP

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| Proxy and Token User is created in the system.  Proxy and Token User Status is ACTIVE.  Token User has token assigned to them. | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - proxyUser | JSON object |  |
| -- appId | String | User Application ID (Eg. ‘IDC’) |
| -- groupId | String (optional) | User Group ID |
| -- loginId | String\* | User id either login d.  Remark: Provide either User Id or Login Id while sending the request. |
| --user id | String\* |
| - tokenUser | JSON object |  |
| -- appId | String | User Application ID (Eg. ‘IDC’ ) |
| -- groupId | String (optional) | User Group ID |
| -- loginId | String\* | User id either login d.  Remark: Provide either User Id or Login Id while sending the request. |
| --user id | String\* |
| - transaction | JSON object |  |
| -- txRefNo | String\* | Transaction Ref No |
| -- randomNo | String\* | Random number. Refer to [Get Challenge API](#_Generate_Challenge_Code) |
| -- challengeSize | Integer\* | Challenge Size |
| -- response | String\* | Response from token based on the challenge entered (button 3) |
| -- tokenType | String\* | Token friendly name |
| -- challengeTxData | String\* | Transaction data from the application used to produced the challenge |
| -- MACTxData | String\* | Transaction data from the application used to produce the MAC |
| -- clientMAC | String\* | MAC produced on the application side |
| -- timestamp | Long\* | Date and time in seconds |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1] – The number of transaction authorization failed attempts left. |

## Transaction Authorization (with two challenge codes) by OTP

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  User has token assigned to them.  Allowed user status: ACTIVE, 2FA-CHANGED | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON object |  |
| -- appId | String | User Application ID (Eg. ‘IDC’) |
| -- groupId | String (optional) | User Group ID |
| -- loginId | String | User Login ID |
| - transaction | JSON object |  |
| -- txRefNo | String | Transaction Ref No |
| -- challenge1 | String | First challenge |
| -- challenge2 | String | Second challenge |
| -- response | String | response from token (button 4) |
| -- tokenType | String | Token friendly name |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1] – The number of transaction authorization failed attempts left. |

## Transaction Authorization Replay by OTP

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  User has token assigned to them.  Allowed user status: ACTIVE, 2FA-CHANGED | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| request | JSON Object |  |
| - user | JSON object |  |
| -- appId | String | User Application ID (Eg. ‘IDC’) |
| -- groupId | String (optional) | User Group ID |
| -- loginId | String\* | User id either login d.  Remark: Provide either User Id or Login Id while sending the request. |
| --userid | String\* |
| - transaction | JSON object |  |
| -- txRefNo | String\* | Transaction Ref No |
| -- randomNo | String\* | Random number. Refer to [Get Challenge API](#_Generate_Challenge_Code) |
| -- challengeSize | Integer\* | Challenge Size |
| -- response | String\* | Response from token based on the challenge entered (button 2) |
| -- tokenType | String\* | Token friendly name |
| -- challengeTxData | String\* | Transaction data from the application used to produce the challenge |
| -- MACTxData | String\* | Transaction data from the application used to produce the MAC |
| -- clientMAC | String\* | MAC produced on the application side |
| -- timestamp | Long\* | Date and time in seconds |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-) |

## Transaction Authorization by DSC

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  User has DSC assigned to them.  Allowed user status: ACTIVE, 2FA-CHANGED | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| request | JSON Object |  |
| - user | JSON object |  |
| -- appId | String | User Application ID (Eg. ‘IDC’) |
| -- groupId | String (optional) | User Group ID |
| -- loginId | String\* | User Login ID |
| -- dsc | JSON object |  |
| --- txRefNo | String\* | Transaction Ref No, the value should be same as the one during get ticket |
| --- txnData | String\* | Transaction data, the value should be same as the one during get ticket |
| --- ticket | String\* | PKI service ticket |
| --- digitalSignMsg | String\* | Message returned from PKI Applet |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1] – Expiry date of the certificate in timestamp format |

## Transaction Authorization Replay by DSC

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  User has DSC assigned to them.  Allowed user status: ACTIVE, 2FA-CHANGED | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| request | JSON Object |  |
| - user | Array of JSON object |  |
| -- appId | String | User Application ID (Eg. ‘IDC’) |
| -- groupId | String (optional) | User Group ID |
| -- loginId | String\* | User Login ID |
| -- dsc | JSON object |  |
| --- txRefNo | String\* | Transaction Ref No, the value should be same as the one during get ticket |
| --- txnData | String\* | Transaction data, the value should be same as the one during get ticket |
| --- ticket | String\* | PKI service ticket |
| --- digitalSignMsg | String\* | Message returned from PKI Applet |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| response | JSON Object |  |
| - response | Array of String | Result for each user, details delimited with “|”:  [0] – <statusCode> | <txRefNo>  statusCode: refer to [Appendix A](#_APPENDIX_A_-) |

## Generate Challenge Code

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  User has token assigned to them.  Allowed user status: ACTIVE, 2FA-CHANGED | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| request | JSON Object |  |
| - transaction | JSON object |  |
| -- txnData | String\* | Transaction data to be hashed |
| -- challengeSize | String\* | Number of digits of the challenge to be returned |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  [1] – Challenge Code  [2] – Random No |

## Remote Transaction Authorization Replay by OTP

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system.  User has a linked token.  User status of the token user should be ACTIVE. | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| request | JSON Object |  |
| - user | JSON object |  |
| -- appId | String | User Application ID (Eg. ‘IDC’) |
| -- groupId | String (optional) | Token user Group ID |
| -- loginId | String\* | User id either login id.  Remark: Provide either User Id or Login Id while sending the request. |
| --userid | String\* |
| - transaction | JSON object |  |
| -- txRefNo | String\* | Transaction Ref No |
| -- randomNo | String\* | Random number. Refer to [Get Challenge API](#_Generate_Challenge_Code) |
| -- challengeSize | Integer\* | Challenge Size |
| -- response | String\* | Response from token based on the challenge entered (button 2) |
| -- tokenType | String\* | Token friendly name |
| -- challengeTxData | String\* | Transaction data from the application used to produce the challenge |
| -- MACTxData | String\* | Transaction data from the application used to produce the MAC |
| -- clientMAC | String\* | MAC produced on the application side |
| -- timestamp | Long\* | Date and time in seconds |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-) |

# User Migration Module API

## User Migration

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| NA | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| request | JSON Object |  |
| - adminuser | JSON Object |  |
| -- appId | String | Admin User application ID |
| -- groupId | String | Admin User group ID |
| -- userId | String | Admin User ID |
| -- password | JSON Object |  |
| --- password | String | Admin user password |
| --- type | Integer | Refer to [Appendix B](#_APPENDIX_B_–) |
| --- ipAddr | String (optional) | If Remote IP address has value then UAAS will validate the IP address |
| --- sessionId | String (optional) | If session ID has value then password type must be either 17 or 18 |
| - user | JSON Object |  |
| -- appId | String | User application ID |
| -- groupId | String | User group ID |
| -- userId | String | User ID |
| -- loginId | String | User Login ID |
| -- customerName | String (optional) | User first name |
| -- email | String (optional) | User email / Login ID |
| -- mobileNo | String (optional) | User mobile number |
| -- phoneNo | String (optional) | User fax number |
| -- dob | String (optional) | User date of birth, dd/MM/yyyy |
| -- nationality | String (optional) | User nationality |
| -- address | String (optional) | User address |
| -- countryCode | String (optional) | User country (two char country code) |
| -- roleId | Integer (optional) | User role ID |
| ~~-- userType~~ | ~~Integer (optional)~~ | ~~User type:~~  ~~1: external user~~  ~~2: internal user~~ |
| -- isLdapAuthenticated | String | Y/N, to indicate if user is authenticated via ldap. It is required because the value needs to be stored and used to determine if sending password via MDS. |
| -- remoteTokenUser | Integer | Y/N |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| response | JSON Object |  |
| - response | Array of string | Array of String response separated with “|” delimiter:  [0 … n] - <statusCode> | <appId> | <groupId> | <userId> | <loginId> | <uaasUserId> |

# Device Management

## Get User Registration Token

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| API Name : getUserRegToken | | |
| User is created in the system.  Allowed user status: All user statuses are allowed except SUSPEND-ON-RQST | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON object |  |
| -- appId | String\* | User Application ID |
| -- groupId | String\* | User Group ID |
| -- userId | String\* | User id either login d.  1. First priority : User ID  2. Second priority : login Id  The login id shall be validating the email format.  Remark: Provide either User Id or Login Id while sending the request. |
| -- loginId | String\* |
| --expDate | String\* | Expire date in the format (dd/MM/yyyy HH24:MI:SS) |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  Following detail will be returned if user exists:  [1] – Unique random token  (appid\_-\_groupid\_-\_userid/loginid\_-\_ expDate\_-\_ random token\_-\_milliseconds(salt)) encrypted string  [2] – token generated date (dd/MM/yyyy HH24:MI:SS) |

## Verify User Registration Token

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| API Name : verifyUserRegToken | | |
| User is created in the system.  Allowed user status: All user statuses are allowed except SUSPEND-ON-RQST | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - device | JSON object |  |
| --encToken | String\* | User Token encryption format  (appid\_-\_groupid\_-\_userid/loginid\_-\_ expDate\_-\_ random token\_-\_milliseconds(salt)) encrypted string |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  Following detail will be returned if user exists:  [1]- appid,groupid,userid/loginid,session ticket |

## Get Device Token

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| API Name : getDeviceToken | | |
| User is created in the system.  Allowed user status: All user statuses are allowed except SUSPEND-ON-RQST | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON object |  |
| -- appId | String\* | User Application ID |
| -- groupId | String\* | User Group ID |
| -- userId | String\* | User id either login d.  1. First priority : User ID  2. Second priority : login Id  The login id shall be validating the email format.  Remark: Provide either User Id or Login Id while sending the request. |
| -- loginId | String\* |
| --expDate | String\* | Expire date in the format (dd/MM/yyyy HH24:MI:SS) |
| --sessionId | String\* | Session data string |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  Following detail will be returned if user exists:  [1] – Unique random token  (appid\_-\_groupid\_-\_userid/loginid\_-\_ expDate\_-\_ random token\_-\_milliseconds(salt)) encrypted string  **Note**: For GRP, user registration token will be invalidated. |

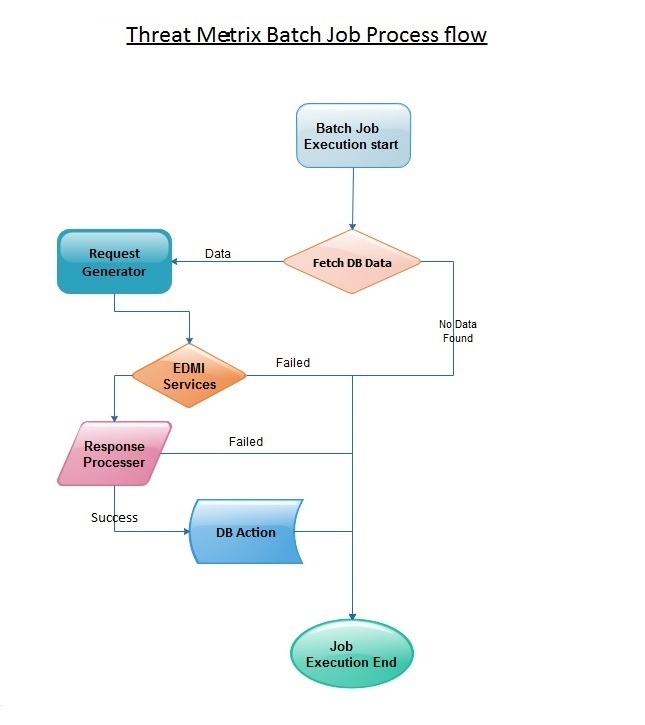
## Verify Device Token

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| API Name : verifyDeviceToken | | |
| User is created in the system.  Allowed user status: All user statuses are allowed except SUSPEND-ON-RQST | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - device | JSON object |  |
| --encToken | String\* | User Token encryption format  Unique random token  (appid\_-\_groupid\_-\_userid/loginid\_-\_ expDate\_-\_ device random token\_-\_milliseconds(salt)) encrypted string |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  Following detail will be returned if user exists:  [1]- appid,groupid,userid/loginid |

## Get Threat Metrix User Data

|  |  |  |
| --- | --- | --- |
| Prerequisite | | |
| User is created in the system. | | |
| Input Parameters | | |
| Variable | Data Type | Explanation |
| Request | JSON Object |  |
| - user | JSON object |  |
| -- appId | String\* | User application ID |
| -- groupId | String\* | User group ID |
| -- userid | String\* | User id either login d.  1. First priority : User ID  2. Second priority : login Id  The login id shall be validating the email format.  Remark: Provide either User Id or Login Id while sending the request. |
| -- loginId | String\* |
| - tmInfo | JSON Object |  |
| -- akamaiipaddress | String (Optional) | IP address of akamai |
| -- tmsessionid | String (Optional) | Unique session id |
| --eventtype | String \* | Event type one of the below.  Login  PreAuthorization  Authorization |
| --txrefnumber | String \* | Unique Transaction Reference number |
| Output Parameters | | |
| Variable | Data Type | Explanation |
| Response | JSON Object |  |
| - response | Array of String | [0] – Status Code, refer to [Appendix A](#_APPENDIX_A_-)  Following detail will be returned if user exists:  [1]- appid,groupid,userid/loginid |

## Threat Metrix Batch job



# APPENDIX A - STATUS CODE



# APPENDIX B – PASSWORD TYPE

|  |  |
| --- | --- |
| **Password Type** | **Remark** |
| 10 | LDAP |
| 16 | NEXTGEN ASYMM ENC RANDOMIZED |
| 11 | NEXTGEN ASYMM ENC RANDOMIZED LDAP |
| 17 | NEXTGEN ASSYM ENC SESSION RANDOMIZED |
| 18 | NEXTGEN ASSYM ENC RANDOMIZED SESSION LDAP |

# APPENDIX C – USER STATUS

|  |  |
| --- | --- |
| **Status** | **Remark** |
| 0 | PENDING |
| 1 | FIRSTTIME |
| 2 | ACTIVE |
| 3 | LOCKED-OUT |
| 4 | SUSPEND-ON-RQST |
| 5 | SUSPEND-ON-RISK |
| 6 | ON-HOLD |
| 7 | REACTIVATED |
| 8 | DORMANT |
| 9 | UNINITIALIZED |
| 10 | 2FA-CHANGED |

# APPENDIX D – APPLICATION PROPERTIES

|  |
| --- |
| **Properties** |
| I\_APPPROP\_DESC |
| I\_APPPROP\_MAXLOGONFAILS |
| I\_APPPROP\_UIDLEN |
| I\_APPPROP\_AUTHTYPE |
| I\_APPPROP\_CERTMAP |
| I\_APPPROP\_AUTH |
| I\_APPPROP\_PASSWDVAL |
| I\_APPPROP\_ROLECHK |
| I\_APPPROP\_BATCHSUSPEND |
| I\_APPPROP\_SUSPENDDORMANTDAYS |
| I\_APPPROP\_BATCHDEL |
| I\_APPPROP\_DELETEDORMANTDAYS |
| I\_APPPROP\_PASSWDHIST |
| I\_APPPROP\_PASSWDEXPDAYS |
| I\_APPPROP\_DUALCTRL |
| I\_APPPROP\_PASSWDLEN |
| I\_APPPROP\_PASSWDTYPE |
| I\_APPPROP\_PASSPHRASELEN |
| I\_APPPROP\_PASSPHRASEHIST |
| I\_APPPROP\_MAXTXAUTHFAILS |