INDEX

|  |
| --- |
| [**Error Codes**](#Error_Codes) |
|  |
| [**User Credential API**](#User_Credential_API)   1. [[POST] user-signup](#User_Credential_API_1) 2. [[POST] user-signin](#User_Credential_API_2) 3. [[POST] user-signout](#User_Credential_API_3) 4. [[DELETE] user-delete (delete account)](#User_Credential_API_4) 5. [[PUT] user-put (change credential data)](#User_Credential_API_5) 6. [[FETCH] user-fetch (get data for specific account)](#User_Credential_API_6) |
| [**User Profile API**](#User_Profile_API)   1. [[POST] user\_profile-add\_profile](#User_Profile_API_1) 2. [[DELETE] user\_profile-delete\_profile](#User_Profile_API_2) 3. [[PUT] user\_profile-changes\_to\_profile](#User_Profile_API_3) 4. [[FETCH] – fetch\_user\_profile\_data](#User_Profile_API_4) |
| [**Admin Credential API**](#Admin_Credential_API)   1. [[POST] – add new simple admin by prime admin](#Admin_Credential_API_1) 2. [[DELETE] – delete self admin OR delete simple admin from prime](#Admin_Credential_API_2) 3. [[PUT] - change self admin OR change simple admin from prime](#Admin_Credential_API_3) 4. [[FETCH] – get self data or simple admin data from prime](#Admin_Credential_API_4) |
| [Admin Privilege API](#Admin_Privilege_API)   1. [[POST] add new admin privilege](#Admin_Privilege_API_1) 2. [[DELETE] remove admin privilege](#Admin_Privilege_API_2) 3. [[FETCH] get admin privilege data](#Admin_Privilege_API_3) |
|  |
| [Coordinator API](#Coordinator_API)   1. [[POST] – coordinator set](#Coordinator_API_1) 2. [[DELETE] – coordinator remove](#Coordinator_API_2) 3. [[FETCH] – coordinator get self or all](#Coordinator_API_3) |
|  |
|  |
|  |
|  |
|  |

**ERROR CODES**

|  |  |
| --- | --- |
| Code | Message |
|  |  |
| 100 | Action Successful. |
|  |  |
| 101 | User already Coordinator |
| 102 | Unauthorized Hash |
| 103 | User not a Coordinator. |
| 104 | Email Id already Exists. |
| 105 | User Profile Exists. |
| 106 | User Profile Does Not Exist. |
| 107 | Subject ID invalid. |
| 108 | Subject Exists, contact admin. |
| 109 | User may not have admin privileges. |
| 110 | USER not ADMIN\_PRIME. |
| 111 | USER not ADMIN |
| 112 | USER already ADMIN |
| 113 | USER id not ADMIN |
| 114 | USER ID not exist |
| 115 | admin n privilege pair exists. |
| 116 | admin privilege does not exist. |
| 117 | admin privilege exists. |
| 118 | admin does not have necessary privileges. |
| 119 | prime coordinator already assigned to subject |
| 120 | USER PROFILE does not exist. |
| 121 | USER not logged in. |
| 150 | API AUTH PROBLEM, check message. |
|  |  |
|  |  |
|  |  |
|  |  |
| 401 | Json formatting error. |
| 402 | Key error. |
| 403 | Invalid Action/Method error. |
|  |  |
| 404 | Unexpected Error, Message accompanied. |
|  |  |
| 405 | Serialization Error. |
| 406 | Invalid api version specified. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

API

User Credential API: **server\_url/api/user/user/**

|  |  |  |  |
| --- | --- | --- | --- |
| [POST] user-signup | | | |
| Variable | Datatype | Mandatory | Description |
| api.auth | string | yes | API auth key provided by admin |
| api.version | string | yes | api version currently 1.0 |
| data.action | string | yes | place “signup” |
| data.data.data.user\_f\_name | string | yes | user’s first name, capitalize |
| data.data.data.user\_m\_name | string | no | user’s middle name, capitalize, if any. |
| data.data.data.user\_l\_name | string | yes | user’s last name, capitalize |
| data.data.data.user\_email | string | yes | proper formatted email, place check on frontend. |
| data.data.data.user\_password | string | yes | min 8 character,  characters, numbers. |
| data.data.data.user\_security\_question | string | no | user specified security question, if any |
| data.data.data.user\_security\_answer | string | no | user specified security answer, if any. |
|  |  |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| [POST] user-signin | | | |
| Variable | Datatype | Mandatory | Description |
| api.auth | string | yes | API auth key provided by admin |
| api.version | string | yes | api version currently 1.0 |
| data.action | string | yes | put "signin/login” as parameter. |
| data.data.hash | string | yes | api version currently 1.0 |
| data.data.data.user\_email | string | yes | proper formatted email id, place check on frontend. |
| data.data.data.user\_password | string | yes | min 8 character,  [a-zA-Z0-9~!@#$%^&\*()\_+] |
|  |  |  |  |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [POST] user-signout | | | | |
| Variable | Datatype | Mandatory | | Description |
| api.auth | string | yes | API auth key provided by admin | |
| api.version | string | yes | api version currently 1.0 | |
| data.action | string | yes | | put "signout/logout” as parameter. |
| data.data.hash | string | yes | | user validation hash from stored in frontend or client cookie. |
|  |  |  | |  |



|  |  |  |  |
| --- | --- | --- | --- |
| [DELETE] user-delete (delete account) | | | |
| Variable | Datatype | Mandatory | Description |
| api.auth | string | yes | API auth key provided by admin |
| api.version | string | yes | api version currently 1.0 |
| data.data.hash | string | yes | user hash in case of self delete  admin hash in case of deleting other accounts |
| data.data.user\_id | array(integers) | no | only if admin is trying to delete other accounts  0 -> delete all(-1)  [1,2,3,4] -> delete specific accounts |
|  |  |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| [PUT] user-put (change credential data) | | | |
| Variable | Datatype | Mandatory | Description |
| api.auth | string | yes | API auth key provided by admin |
| api.version | string | yes | api version currently 1.0 |
| data.data.hash | string | yes | user hash for authorization. |
| data.data.user\_f\_name | string | yes | user’s first name, capitalize |
| data.data.data.user\_m\_name | string | no | user’s middle name, capitalize |
| data.data.data.user\_l\_name | string | yes | user’s last name, capitalize |
| data.data.data.user\_email | string | yes | user’s original email or changed email. |
|  |  |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| [FETCH] user-fetch (get data for specific account) | | | |
| Variable | Datatype | Mandatory | Description |
| api.auth | string | yes | API auth key provided by admin |
| api.version | string | yes | api version currently 1.0 |
| data.data.hash | string | yes | user hash in case of self delete  admin hash in case of deleting other accounts |
| data.data.user\_id | array(integers) | no | only if admin is trying to delete other accounts  0 -> delete all(-1)  [1,2,3,4] -> delete specific accounts |
|  |  |  |  |



User Profile API: **server\_url/api/user/profile**

|  |  |  |  |
| --- | --- | --- | --- |
| [POST] user\_profile-add\_profile | | | |
| Variable | Datatype | Mandatory | Description |
| api.auth | string | yes | API auth key provided by admin |
| api.version | string | yes | api version currently 1.0 |
| data.data.hash | string | yes | user hash in case of self delete  prime\_admin hash in case of deleting other accounts |
| data.data.data.user\_profile\_headline | string | yes | profile headline |
| data.data.data,user\_bio | string | no | user full descriptive bio. |
| data.data.data.user\_english\_efficiency | number | yes | 1 -> **Beginner**, default  2 -> Intermediate  3 -> Advanced |
| data.data.data.user\_git\_profile | url | no | user’s github profile link, if any |
| data.data.data.user\_likedin\_profile | url | no | user’s linkedin profile link, if any |
| data.data.data.user\_profile\_pic\_url | 404 | 404 | 404, **under construction** |
| data.data.data.user\_roll\_number | number | yes | 12 digit university roll |
|  |  |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| [DELETE] user\_profile-delete\_profile | | | |
| Variable | Datatype | Mandatory | Description |
| api.auth | string | yes | API auth key provided by admin |
| api.version | string | yes | api version currently 1.0 |
| data.data.hash | string | yes | user hash in case of self delete  admin hash in case of deleting other accounts |
| data.data.user\_id | array(integers) | no | only if admin is trying to delete other accounts  0 -> delete all(-1)  [1,2,3,4] -> delete specific accounts |
|  |  |  |  |

****

|  |  |  |  |
| --- | --- | --- | --- |
| [PUT] user\_profile-changes\_to\_profile | | | |
| Variable | Datatype | Mandatory | Description |
| api.auth | string | yes | API auth key provided by admin |
| api.version | string | yes | api version currently 1.0 |
| data.data.hash | string | yes | user hash in case of self delete  prime\_admin hash in case of deleting other accounts |
| data.data.data.user\_profile\_headline | string | yes | profile headline |
| data.data.data,user\_bio | string | no | user full descriptive bio. |
| data.data.data.user\_english\_efficiency | number | yes | 1 -> **Beginner**, default  2 -> Intermediate  3 -> Advanced |
| data.data.data.user\_git\_profile | url | no | user’s github profile link, if any |
| data.data.data.user\_likedin\_profile | url | no | user’s linkedin profile link, if any |
| data.data.data.user\_profile\_pic\_url | 404 | 404 | 404, **under construction** |
| data.data.data.user\_roll\_number | number | yes | 12 digit university roll |
|  |  |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| [FETCH] – fetch\_user\_profile\_data | | | |
| Variable | Datatype | Mandatory | Description |
| api.auth | string | yes | API auth key provided by admin |
| api.version | string | yes | api version currently 1.0 |
| data.data.hash | string | yes | user hash in case of self delete  admin hash in case of deleting other accounts |
| data.data.user\_id | array(integers) | no | only if admin is trying to delete other accounts  0 -> delete all(-1)  [1,2,3,4] -> delete specific accounts |
|  |  |  |  |



Admin Credential API: **server\_url/api/admin/admin/**

|  |  |  |  |
| --- | --- | --- | --- |
| [POST] – add new simple admin by prime admin | | | |
| Variable | Datatype | Mandatory | Description |
| data.hash | string | yes | user validation hash from stored in frontend or client cookie. |
| data.data.user\_credential\_id | number | yes | user id to get admin access |
| api\_v | string | yes | api version currently 1.0 |
|  |  |  |  |
|  |  |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| [DELETE] – delete self admin OR delete simple admin from prime | | | |
| Variable | Datatype | Mandatory | Description |
| data.hash | string | yes | user validation hash from stored in frontend or client cookie. |
| data.data.user\_credential\_id | number | no | user id to revoke admin access, or don’t include attribute if self revoke. |
| api\_v | string | yes | api version currently 1.0 |
|  |  |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| [PUT] - change self admin OR change simple admin from prime | | | |
| Variable | Datatype | Mandatory | Description |
| api\_v | string | yes | api version currently 1.0 |
| data.hash | string | yes | user validation hash from stored in frontend or client cookie. |
| data.data.user\_credential\_id | number | no | only if adding privileges to others, no need if action on self |
| data.data.privileges | array(int) | yes | privilege id can be tracked from admin privilege api  positive : enforce privilege  negative : retract privilege. |
|  |  |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| [FETCH] get admin privilege data | | | |
| Variable | Datatype | Mandatory | Description |
| api\_v | string | yes | api version currently 1.0 |
| data.hash | string | yes | user validation hash from stored in frontend or client cookie. |
| data.data.user\_credential\_id | number | no | only if hash owner is prime admin, then can view other admin status |
|  |  |  |  |



Admin Privilege API: **server\_url/api/admin/privilege/**

|  |  |  |  |
| --- | --- | --- | --- |
| [POST] add new admin privilege | | | |
| Variable | Datatype | Mandatory | Description |
| api\_v | string | yes | api version currently 1.0 |
| data.hash | string | yes | prime admin validation hash from stored in frontend or client cookie. |
| data.data.admin\_privilege\_name | string | yes | code name of privilege containing no more than 10 alphabetic character |
| data.data.admin\_privilege\_description | string | yes | brief description of admin privilege being initialized |
|  |  |  |  |

****

|  |  |  |  |
| --- | --- | --- | --- |
| [DELETE] remove admin privilege | | | |
| Variable | Datatype | Mandatory | Description |
| api\_v | string | yes | api version currently 1.0 |
| data.hash | string | yes | prime admin validation hash from stored in frontend or client cookie. |
| data.admin\_privilege\_id | number | yes | privilege id |
|  |  |  |  |

****

|  |  |  |  |
| --- | --- | --- | --- |
| [FETCH] get admin privilege data | | | |
| Variable | Datatype | Mandatory | Description |
| api\_v | string | yes | api version currently 1.0 |
| data.hash | string | yes | prime admin validation hash from stored in frontend or client cookie. |
| data.admin\_privilege\_id | array(integer) | yes | [0]-> all  or an array of ids |
|  |  |  |  |

****

Coordinator API: **server\_url/api/content/coordinator/**

|  |  |  |  |
| --- | --- | --- | --- |
| [POST] – coordinator set | | | |
| Variable | Datatype | Mandatory | Description |
| data.hash | string | yes | admin validation hash from stored in frontend or client cookie. |
| api\_v | string | yes | api version currently 1.0 |
| data.user\_credential\_id | array(integer) | yes | user to be granted coordinator role by special admins. **COPG** |
|  |  |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| [DELETE] – coordinator remove | | | |
| Variable | Datatype | Mandatory | Description |
| data.hash | string | yes | admin validation hash from stored in frontend or client cookie. |
| api\_v | string | yes | api version currently 1.0 |
| data.user\_credential\_id | array(integer) | no | don’t include if self-revoke, or else for **COPG** admin revoking others inclusion necessary. |
|  |  |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| [FETCH] – coordinator get self or all | | | |
| Variable | Datatype | Mandatory | Description |
| data.hash | string | yes | admin validation hash from stored in frontend or client cookie. |
| api\_v | string | yes | api version currently 1.0 |
| data.user\_credential\_id | array(integer) | yes | [0]-> all  or an array of ids |
|  |  |  |  |



Subject API: **server\_url/api/content/subject/**

|  |  |  |  |
| --- | --- | --- | --- |
| [POST] – Add new subject | | | |
| Variable | Datatype | Mandatory | Description |
| data.hash | string | yes | admin validation hash from stored in frontend or client cookie. |
| api\_v | string | yes | api version currently 1.0 |
| data.data.subject\_name | string | yes | Name of subject (be precise) |
| data.data.subject\_description | stirng | yes | a brief but precise description |
|  |  |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| [Delete] – delete subjects in bunch | | | |
| Variable | Datatype | Mandatory | Description |
| data.hash | string | yes | admin validation hash from stored in frontend or client cookie. |
| api\_v | string | yes | api version currently 1.0 |
| data.subject\_id | array(integer) | yes | an array of subject ids |
|  |  |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| [FETCH] – get subject data in bunch | | | |
| Variable | Datatype | Mandatory | Description |
| data.hash | string | yes | admin validation hash from stored in frontend or client cookie. |
| api\_v | string | yes | api version currently 1.0 |
| data.subject\_id | array(integer) | yes | [0] -> all  [1,2,3…] -> specific subject ids |
|  |  |  |  |

