USER PROVISIONING API GUIDE

Updated June 2018

Contents

[Getting Started 3](#_Toc500488121)

[Authentication 3](#_Toc500488122)

[Request/response formats 3](#_Toc500488123)

[Rate limits 4](#_Toc500488124)

[Get Profile Fields of Program 4](#_Toc500488125)

[Get Profile Updates Made In Program 5](#_Toc500488126)

[List Users in the Program 7](#_Toc500488127)

[I. List users at the global level 7](#_Toc500488128)

[II. List users at the track level 8](#_Toc500488129)

[Retrieve User Information 10](#_Toc500488130)

[I. Retrieve user’s unique user identifier 10](#_Toc500488131)

[II. Retrieve user’s profile information 10](#_Toc500488132)

[Create New User Account 11](#_Toc500488133)

[I. Create user at the global level 11](#_Toc500488134)

[II. Create user at the track level 12](#_Toc500488135)

[Update User Information 12](#_Toc500488136)

[I. Update user status at the global level 12](#_Toc500488137)

[II. Update user status at the track level 13](#_Toc500488138)

[III. Update user’s profile 14](#_Toc500488139)

[Remove User 15](#_Toc500488140)

[I. Remove user at the global level 15](#_Toc500488141)

[II. Remove user at the track level 15](#_Toc500488142)

[Error Descriptions 16](#_Toc500488143)

Getting Started

Authentication

Chronus API Requires an API Key for authentication. Authentication is made by passing the “api\_key” parameter along with the API request. All requests must be made over SSL to https://<program\_url>. Please connect with your Customer Success Manager to get an API Key for your program.

**Sample Usage:**

https://mainprogram.chronus.com/api/v2/members?api\_key=<API\_KEY\_HERE>

Request/response formats

All API Calls support an optional return format parameter “format=json”. Note that json is the default response format. The xml is the other response format that is available.

**Sample Request:**

https://mainprogram.chronus.com/api/v2/members?email=janedoe@example.com&format=xml

**Sample Response:**

<?xml version="1.0" encoding="UTF-8"?>

<users type="array">

<user>

<first-name>Jane</first-name>

<last-name>Doe</last-name>

<email>janedoe@example.com</email>

<status type="integer">0</status>

<uuid type="integer">12345</uuid>

<login-name></login-name>

<programs>

<Test-Program>

<status type="integer">0</status>

<roles type="array">

<role>admin</role>

<role>mentor</role>

</roles>

</Test-Program>

</programs>

</user>

</users>

Rate limits

API calls are subject to rate limiting. The rate limits are:

* 1 request per second
* 30 requests per minute
* 750 requests per hour
* 10K requests per day

Get Profile Fields of Program

**Method: [HTTP GET]**

Index

**Request Parameters:**

|  |  |  |
| --- | --- | --- |
| parent\_field\_id | ID of the field for which the child profile fields needs to be fetched | Optional |
| answer\_text | Answer for the parent\_field\_id for which conditional questions are required. | Required only if parent\_field\_id is present |

When the optional parameter parent\_field\_id is passed, the answer\_text also needs to be specified. The API will return all the profile fields which are dependent on the answer text and parent\_field\_id mentioned in the request parameter.

When the optional parameters are not passed, the API will return all the profile fields in the track.

**Response:**

For each profile field: id, label, type, choices (if any), conditional profile field details (if any), description and programs that the question is applicable to (with roles it is applicable for). Returned in JSON or XML as requested by caller.

**Sample Requests:**

https://mainprogram.chronus.com/api/v2/profile\_fields

https://mainprogram.chronus.com/api/v2/profile\_fields?parent\_field\_id=123&answer\_text=engineer,doctor

**Sample Response:**

[

{

“id”: “123”,

“label”: “Gender”,

“type”: “Multichoice”,

“choices”: “M, F”,

“description”: “Gender of the user”,

“programs”: {“p1”: [“mentor”, “mentee”], “p2”: [“mentor”]}

},

{

“id”: “235”,

“label”: “Occupation”,

“type”: “Multichoice”,

“choices”: “Engineer, Doctor,Business”,

“programs”: {“p1”: [“mentor”, “mentee”], “p2”: [“mentor”]}

},

{

“id”: “678”,

“label”: “Work Location”,

“type”: “Location”,

“condition\_to\_show”: {id: “235”, answer: “Engineer,Doctor”},

“programs”: {“p1”: [“mentor”, “mentee”], “p2”: [“mentor”]}

}

]

Get Profile Updates Made In Program

**Method: [HTTP GET]**

profile\_updates

**Request Parameters:**

|  |  |  |
| --- | --- | --- |
| updated\_after | Timestamp in UTC. Lists only the profile updates that occurred after a given timestamp. |  |
| status | Comma separated list of values (0,2,3) for (Active, Suspended, Dormant) | Optional |

The API will return any profile updates in the program subject to a maximum of 10k updates.

**Response:**

For each user whose profile is updated after the given timestamp: first name, last name, uuid, email, status and updated profile values. Returned in JSON or XML as requested by caller.

**Sample Requests:**

https://mainprogram.chronus.com/api/v2/members/profile\_updates?status=2&updated\_after=2017-12-22 11:00:00

**Sample Response:**

[{

      "first\_name": "Jane",

      "last\_name": "Doe",

      "email": "janedoe@example.com",

      "status": 2,

      "uuid": 12345,

      “profile\_updates”: {

   “field\_value\_10”: {“field\_id”: 15, “field\_name”: “Gender”, “value”: “Female”, “applicable”: 1 },

         “field\_value\_11”: {“field\_id”: 16, “field\_name”: “Interests”, “value”: “Accounting,Marketing”, “applicable”: 1 }

 }

},

{

      "first\_name": "John",

      "last\_name": "Anthony",

      "email": "john@example.com",

      "status": 2,

      "uuid": 12349,

      “profile\_updates”: {

         “field\_value\_11”: {“field\_id”: 16, “field\_name”: “Interests”, “value”: “Advertising”, “applicable”: 1 },

   “field\_value\_14”: {“field\_id”: 19, “field\_name”: “City”, “value”: “Seattle”, “applicable”: 1 }

 }

}

]

List Users in the Program

I. List users at the global level

**Method: [HTTP GET]**

Index

**Request Parameters:**

|  |  |  |
| --- | --- | --- |
| status | Comma separated list of values (0,2,3) for (Active, Suspended, Dormant) | Optional |
| email | Retrieves user with the email address | Optional |
| created\_after | Timestamp in UTC. Lists the members created after the given timestamp. | Optional |

When the optional parameters are not passed, the API will return all users.

**Response:**

For each user: first name, last name, uuid, email, status and program names with status and roles (if applicable). Returned in JSON or XML as requested by caller.

**Sample Request:**

https://mainprogram.chronus.com/api/v2/members

https://mainprogram.chronus.com/api/v2/members?status=0

https://mainprogram.chronus.com/api/v2/members?email=janedoe@example.com

https://mainprogram.chronus.com/api/v2/members?created\_after=2017-12-22 11:00:00

**Sample Response:**

[{

"first\_name": "Jane",

"last\_name": "Doe",

"email": "janedoe@example.com",

"status": 2,

"uuid": 12345,

“programs”: {“p1”: {“status”: 0, “roles”: [“mentor”]}, “p2”: {“status”: 2, “roles”: [“mentor”, “mentee”]}}

},

{

"first\_name": "John",

"last\_name": "Anthony",

"email": "john@example.com",

"status": 2,

"uuid": 12349,

“programs”: {“p1”: {“status”: 1, “roles”: [“mentor”]}, “p2”: {“status”: 2, “roles”: [“mentor”]}}

}

]

II. List users at the track level

**Method: [HTTP GET]**

Index

**Request Parameters:**

|  |  |  |
| --- | --- | --- |
| roles | Comma separated list of values (mentor, mentee and/or admin) | Optional |
| email | Email address of the user | Optional |
| status | (0,1,3) Indicates (Active, Unpublished, Deactivated) | Optional |

When the optional parameters are not passed, the API will return all the users in the track.

**Response:**

For each user: first name, last name, uuid, email, status and role(s). Returned in JSON or XML as requested by caller.

**Sample Request:**

https://mainprogram.chronus.com/p/p1/api/v2/users

https://mainprogram.chronus.com/p/p1/api/v2/users?roles=mentor

https://mainprogram.chronus.com/p/p1/api/v2/users?roles=mentor&status=0

**Sample Response:**

[{

"first\_name": "Jane",

"last\_name": "Doe",

"email": "janedoe@example.com",

"status": 0,

"uuid": 12345,

“roles”: [“mentor”,”mentee”]

},

{

"first\_name": "John",

"last\_name": "Anthony",

"email": "john@example.com",

"status": 1,

"uuid": 12349,

“roles”: [“mentor”,”mentee”]

}

Retrieve User Information

I. Retrieve user’s unique user identifier

**Method: [HTTP GET]**

get\_uuid

**Request Parameters:**

|  |  |
| --- | --- |
| email | Email of the user |

**Response:**

uuid at the global level. Returned in JSON or XML as requested by caller.

**Sample Request:**

https://mainprogram.chronus.com/api/v2/members/get\_uuid?email=janedoe@example.com

**Sample Response:**

{

"uuid": "12345"

}

II. Retrieve user’s profile information

**Method: [HTTP GET]**

show

**Request Parameters:**

|  |  |  |
| --- | --- | --- |
| uuid | Unique identifier of user |  |
| profile | (0, 1). Defaults to 1 when not specified indicating that user’s profile would be retrieved along with other information | Optional |

**Response:**

first name, last name, uuid, email, status, program names with roles (if applicable) and profile data if requested. Returned in JSON or XML as requested by caller.

**Sample Request:**

https://mainprogram.chronus.com/api/v2/members/12345

**Sample Response:**

{

"first\_name": "Jane",

"last\_name": "Doe",

"email": "janedoe@example.com",

"status": 0,

"uuid": 12345,

“programs”: {

“p1”: {“status”: 0, “roles”: [“mentor”]},

“p2”: {“status”: 2, “roles”: [“mentor”, “mentee”]}

},

“profile”: {

“field\_value\_10”: {“field\_id”: 15, “field\_name”: “Gender”, “value”: “Female”, “applicable”: 1 },

“field\_value\_11”: {“field\_id”: 16, “field\_name”: “Interests”, “value”: “Accounting,Marketing”, “applicable”: 1 }

}

}

Create New User Account

I. Create user at the global level

**Method: [HTTPS POST]**

create

**Request Parameters:**

|  |  |  |
| --- | --- | --- |
| email | Email address of user |  |
| first\_name | First name of user |  |
| last\_name | Last name of user |  |
| login\_name | Unique identifier of user, used for SSO | Optional |

**Response:**

uuid of newly created user

**Sample Request:**

https://mainprogram.chronus.com/api/v2/members?email=janedoe@example.com&first\_name=Jane&last\_name=Doe&login\_name=jdoe

**Sample Response:**

{

“uuid”: “12345”

}

II. Create user at the track level

**Method: [HTTPS POST]**

create

**Request Parameters:**

|  |  |  |
| --- | --- | --- |
| uuid | Unique identifier of user |  |
| roles | Comma separated list of values (mentor, mentee and/or admin) |  |
| send\_invite | (1, 0). Indicates if user should receive invite email from Chronus Mentor or not. Defaults to 0 when not specified | Optional |

**Response:**

uuid of newly created user

**Sample Request:**

https://mainprogram.chronus.com/p/p1/api/v2/users?uuid=12345&roles=mentor&first\_name=Jane&last\_name=Doe

**Sample Response:**

{

“uuid”: “12345”

}

Update User Information

I. Update user status at the global level

**Method: [HTTPS PUT]**

update\_status

**Request Parameters:**

|  |  |
| --- | --- |
| uuid | Unique identifier of user [cannot be changed] |
| status | (0, 2). Indicates (Active, Suspended) |

**Response:**

Updated user information: first name, last name, uuid, email and status. Returned in JSON or XML as requested by caller.

**Sample Request:**

https://mainprogram.chronus.com/api/v2/members/update\_status?uuid=12345&status=0

**Sample Response:**

{

"first\_name": "Jane",

"last\_name": "Doe",

"email": "janedoe@example.com",

"status": 0,

"uuid": 12345

}

II. Update user status at the track level

**Method: [HTTPS PUT]**

update\_status

**Request Parameters:**

|  |  |
| --- | --- |
| uuid | Unique identifier of user [cannot be changed] |
| status | (0, 3). Indicates (Active, Deactivated) |

**Response:**

Updated user information: first name, last name, uuid, email and status. Returned in JSON or XML as requested by caller.

**Sample Request:**

https://mainprogram.chronus.com/p/p1/api/v2/users/update\_status?uuid=12345&status=3

**Sample Response:**

{

"first\_name": "Jane",

"last\_name": "Doe",

"email": "janedoe@example.com",

"status": 0,

"uuid": 12345,

“roles”: [“mentor”,“mentee”]

}

III. Update user’s profile

**Method: [HTTPS PUT]**

update

**Request Parameters:**

|  |  |  |
| --- | --- | --- |
| uuid | Unique identifier of user [cannot be changed] | Required |
| email | Updated email address of user | Optional |
| first\_name | Updated first name of user | Optional |
| last\_name | Updated last name of user | Optional |
| profile[id] | Profile question to be updated | Optional |
| login\_name | Updated login identifier of user (used for SSO login) | Optional |

**Response:**

Updated user information: first name, last name, uuid, email and profile fields. Returned in JSON or XML as requested by caller.

**Sample Request:**

https://mainprogram.chronus.com/api/v2/members/12345?email=janedoe@example.com&profile[10]=female&profile[11]=Accounting,Finance[&l](https://mainprogram.chronus.com/api/v2/members/12345?email=janedoe@example.com&profile%5B10%5D=female&profile%5B11%5D=Accounting,Finance&l)ogin\_name=jane

Where 10 and 11 are profile field ids.

**Sample Response:**

{

"first\_name": "Jane",

"last\_name": "Doe",

"email": "janedoe@example.com",

"status": 0,

"uuid": 12345,

"login\_name": "jane",

“profile”: {

“field\_value\_15”: {“field\_id”: 10, “field\_name”: “Gender”, “value”: “Female”, “applicable”: 1 },

“field\_value\_16”: {“field\_id”: 11, “field\_name”: “Interests”, “value”: “Accounting,Finance”, “applicable”: 1 }

}

}

Remove User

I. Remove user at the global level

**Method: [HTTPS DELETE]**

destroy

**Request Parameters:**

|  |  |
| --- | --- |
| uuid | Unique identifier of the user |

**Response:**

uuid of removed user at the global level

**Sample Request:**

https://mainprogram.chronus.com/api/v2/members/12345

**Sample Response:**

{

"uuid": "12345"

}

II. Remove user at the track level

**Method: [HTTPS DELETE]**

destroy

**Request Parameters:**

|  |  |
| --- | --- |
| uuid | Unique identifier of the user |

**Response:**

uuid of removed user at the track level

**Sample Request:**

https://mainprogram.chronus.com/p/p1/api/v2/users/12345

**Sample Response:**

{

"uuid": "12345"

}

Error Descriptions

In case the Web Service encounters an error with the parameters passed in appropriate HTTP status, a code with error message (in XML, JSON or HTML specified by caller) will be returned. The most common errors returned by the interfaces are listed below.

|  |  |  |
| --- | --- | --- |
| **Error** | **Description** | **HTTP Status Code** |
| User not found | uuid is not present in the Chronus database, this error is returned. | 404 |
| Invalid Role Names | Role names should be mentor, mentee or admin | 404 |
| Invalid Status Update | The following status updates are invalid:  At the global level:   * Active/suspended to dormant * Dormant to active * Suspended to active when no user records exist, the user will be moved to dormant state   At the track level:   * Active/Deactivated to Unpublished | 404 |
| Authorization failure |  | 403 |