

Virtual Experience Platform Public API

The Virtual Experience Platform (VEP) exposes a Public API which allows third party vendors to view, create, update, and delete users within the VEP user base. This creates sync profiles between users in external systems and those in your VEP events. The Public API also supports Single-Sign-On from third party systems to allow users of third party systems enter the virtual environment with a single click.

The API operates using JSON over HTTPS. An API credential will be supplied on a client-by-client basis, allowing access to a specific set of users. This credential must be presented with each API request. The credential may be entered in the JSON itself (in the case of DOM-level controls like JavaScript) or in a standard HTTP-Basic Authorization header (in the case of server-side HTTP controls like cURL).

Server URL

Production: `https://<clientDomain>.6connex.com`

Staging: `https://<clientDomain>.6connexstage.com`

The `clientDomain` will be provided once the client account is created.

User APIs

The User APIs provide `create`, `read`, `update`, `delete`, and `readAll` endpoints.

We also support form-based create user request with content type as `application/x-www-form-urlencoded`. Refer to Form Based Create User API section for details.

Request Headers

Each HTTP request must provide the following headers in the request.

- `Content-Type: application/json`
- `Accept: application/json`

APIs Test Page

Public API has a web page for you to test these APIs so that you can try it before writing the code.

Production: `https://<clientDomain>.6connex.com/publicapi/test-apis`

Staging: <https://<clientDomain>.6connexstage.com/publicapi/test-apis>

API create

Path: /publicapi/users/executeAPICall

Method: POST

Request Body:

Example A: This is an example with no custom question.

```
{
  "apiUsername": "<apiUsername>",
  "apiPassword": "<apiPassword>",
  "apicallsetinput": [
    {
      "_apicall": "create",
      "firstname": "Test",
      "lastname": "User",
      "password": "qwertyuil!",
      "email": "test@3rdpartydomain.com",
      "company": "3rdparty",
      "title": "Engineer",
      "profile_image": "<url-for-image>",
      "language": "en_US",
      "event_id": <eventId>,
      "entitlement_group": "default group",
      "registration_set": "test"
    }
  ]
}
```

Note

- Parameters apiUsername, apiPassword, and eventId will be provided.
- These fields are mandatory: email, firstname, lastname, language, event_id, and entitlement_group.
- The default language is "en_US". It's dependent on the configuration of the event control panel.
- The value of parameter _apicall must be create.
- "default group" is the default entitlement group each event has. If you need a different value, please make sure it is added in Control Panel of the event.
- registration_set is optional. The value of this field is the url_slug of a registration set.

Example B: This is an example with two custom questions.

```
{
  "apiUsername": "<apiUsername>",
  "apiPassword": "<apiPassword>",
  "apicallsetinput": [
    {
      "_apicall": "create",
      "firstname": "Test",
```

```

        "lastname": "User",
        "password": "qwertyui1!",
        "email": "test@3rdpartydomain.com",
        "company": "3rdparty",
        "profile_image": "<url-for-image>",
        "title": "Engineer",
        "language": "en_US",
        "event_id": <eventId>,
        "entitlement_group": "default group",
        "registration_set": "test",
        "Custom Question1": "This is the answer",
        "Custom Question2": ["Option A", "Option B"]
    }
}
]
}

```

Note

- The custom question field name, Custom Question1 in this example, needs to be exactly the same label as added in Control Panel's registration form.
- When a question is of type "drop down" or "radio button", the answer needs to be exactly the same value as added in the registration form.
- When a question is of type "checkbox", the values need to be put in an array, like the Custom Question 2's answers in the above example.

Response

```

{
  "apicallsetoutput": [
    {
      "id": <newlyCreatedUserId>,
      "_apicall": "create",
      "_apicallresultcode": 1,
      "_apicallresultmessage": "success"
    }
  ]
}

```

Note

- The id in the response is the newly created user's ID in VEP user base.

API read

Path: /publicapi/users/executeAPICall

Method: POST

Request Body:

Example: ready by user Id

```

{
  "apiUsername": "<apiUsername>",
  "apiPassword": "<apiPassword>",
  "apicallsetinput": [
    {
      "_apicall": "read",

```

```

        "id": 426250,
        "event_id": 1234,
    }
]
}

```

Example: ready by user email

```

{
    "apiUsername": "<apiUsername>",
    "apiPassword": "<apiPassword>",
    "apicallsetinput": [
        {
            "_apicall": "read",
            "email": "john.doe@example.com",
            "event_id": 789,
        }
    ]
}

```

Note

- The `id` is the newly created user's ID received in the `create` API.
- The `event_id` is the id of the event.
- The value of field `_apicall` needs to be `read`.

Response

Example

```

{
    "apicallsetoutput": [
        {
            "id": 426250,
            "firstname": "Test",
            "lastname": "User",
            "email": "john.doe@example.com",
            "company": "3rdparty",
            "title": "Engineer",
            "contact": "chat",
            "language": "en_US",
            "events": {
                "789": {
                    "group_name": "V7QA3",
                    "entitlementgroup_name": "default group",
                    "register_date": "2014-12-20 04:39:16",
                    "event_id": 789,
                    "event_name": "V7QA3"
                }
            },
            "initially_created_by_partner": true,
            "lastmodified": "2014-12-20 04:39:16",
            "_apicall": "read",
            "_apicallresultcode": 1,
            "_apicallresultmessage": "success"
        }
    ]
}

```

```
]
}
```

API update

Path: /publicapi/users/executeAPICall

Method: POST

Request Body:

Example

```
{
  "apiUsername": "<apiUsername>",
  "apiPassword": "<apiPassword>",
  "apicallsetinput": [
    {
      "_apicall": "update",
      "id": <userId>,
      "event_id": <eventId>,
      "lastname": "Last Name Updated"
    }
  ]
}
```

Note

- The value of field `_apicall` needs to be update.
- Fields `id`, `event_id` are required.
- Email cannot be changed once created. Email will be ignored in the update request.
- By default, password won't be updated unless field `override` is set to true.

Response

Example

```
{
  "apicallsetoutput": [
    {
      "_apicall": "update",
      "_apicallresultcode": 1,
      "_apicallresultmessage": "success"
    }
  ]
}
```

API delete

Path: /publicapi/users/executeAPICall

Method: POST

Request Body:

Example

```
{
  "apiUsername": "<apiUsername>",
  "apiPassword": "<apiPassword>",
```

```

    "apicallsetinput": [
      {
        "_apicall": "delete",
        "id": <userId>
        "event_id": <eventId>
      }
    ]
  }

```

Response

Example

```

{
  "apicallsetoutput": [
    {
      "_apicall": "delete",
      "_apicallresultcode": 1,
      "_apicallresultmessage": "success"
    }
  ]
}

```

API readAll

Path: /publicapi/users/executeAPICall

Method: POST

Request Body:

Example

```

{
  "apiUsername": "<apiUsername>",
  "apiPassword": "<apiPassword>",
  "apicallsetinput": [
    {
      "_apicall": "readall",
      "timestamp": 0,
      "limit": 2,
      "offset": 0
    }
  ]
}

```

Note

- The value of field `_apicall` needs to be `readAll`.
- Parameter `offset` starts from 0. And `limit` is the size of the result to return.

Response

Example

```

{
  "apicallsetoutput": [
    {
      "id": 328298,
      "firstname": "Test",

```

```

      "lastname": "User",
      "email": "test001@testlocal.com",
      "company": "3rdparty",
      "title": "Engineer",
      "contact": "chat",
      "language": "en_US",
      "events": {
        "789": {
          "group_name": "V7QA3",
          "entitlementgroup_name": "default group",
          "register_date": "2014-08-21 03:43:19",
          "event_id": 789,
          "event_name": "V7QA3"
        }
      },
      "initially_created_by_partner": true,
      "lastmodified": "2014-08-21 03:43:19",
      "_apicall": "readall",
      "_apicallresultcode": 0,
      "_apicallresultmessage": "success"
    },
    {
      "id": 341832,
      "firstname": "Test",
      "lastname": "User",
      "email": "test0902@6connexetest.com",
      "company": "3rdparty",
      "title": "Engineer",
      "contact": "chat",
      "language": "en_US",
      "events": {
        "789": {
          "group_name": "V7QA3",
          "entitlementgroup_name": "default group",
          "register_date": "2014-09-02 05:24:52",
          "event_id": 789,
          "event_name": "V7QA3"
        }
      },
      "initially_created_by_partner": true,
      "lastmodified": "2014-09-02 05:24:52",
      "_apicall": "readall",
      "_apicallresultcode": 0,
      "_apicallresultmessage": "success"
    }
  ]
}

```

Filter results of readAll API

In order to filter the result, you need to provide the following parameters.

- **filterBy**: String
 - Supported values:
 - **lastModifiedDate**

- This is the last modified date of this user via Public API's create, update API.
- **registeredDate**
 - This is the date the user is registered.
- **lastLoginDate**
 - This is the date the user last login to the environment.
- **startDate**: String
 - The items in the result will have date greater than or equal to this value.
 - Only date and time pattern `YYYY-MM-DDThh:mm:ssZ` is supported. For example `2017-01-23T12:00:00Z`. The Z at the end means it is UTC time zone.
- **endDate**: String
 - The items in the result will have date less than this value.
 - Only date and time pattern `YYYY-MM-DDThh:mm:ssZ` is supported. For example `2017-01-23T12:00:00Z`. The Z at the end means it is UTC time zone.

Parameter `filterBy` is required. You can choose to pass only `startDate` or `endDate`, or you can pass both of them.

Note: The date and time pattern `YYYY-MM-DDThh:mm:ssZ` is W3C standard ISO 8601 (<https://www.w3.org/TR/NOTE-datetime>). It might not match the one used in the programming language you're using. For example, in ISO 8601, hh is 0-23, while in Java, HH is 0-23.

Example:

```
{
  "apiUsername": <Username>,
  "apiPassword": <Password>,
  "apicallsetinput": [{
    "_apicall": "readall",
    "limit": 2,
    "offset": 0,
    "filterBy": "lastLoginDate",
    "startDate": "2017-12-22T00:00:00Z",
    "endDate": "2017-12-23T00:00:00Z"
  }]
}
```

Multiplicity

All of these API calls can be nested into a single "apicallsetinput" array and sent to the API processor. The order of action processing is as such; first to last: delete, create, update, and read. While available, it's not advisable to add "readall" to a set of other API calls, due to problems of parsing the result.

Example

```
{
  "apiUsername": "3rdPartyName",
  "apiPassword": "password",
  "apicallsetinput": [
    {
      "_apicall": "create",
      "firstname": "First",
      "lastname": "Last",
      "email": "email@domain.com"
    },
    {
      "_apicall": "delete",
      "id": 3
    },
    {
      "_apicall": "update",
      "id": 7828,
      "firstname": "newFirst",
      "lastname": "newLast",
      "email": "email@domain.com"
    },
    {
      "_apicall": "read",
      "id": 20
    },
    {
      "_apicall": "read",
      "id": 92
    }
  ]
}
```

About Result Codes, Errors, and ACLs

The output of each API call is, at a minimum, the input JSON plus a `"_apicallresultcode"` field. This field contains a response code of a positive number indicating success and 0 or a negative number indicating failure. Additionally a `"_apicallresultmessage"` field gives the text description of the response or error if any.

Failure of an API call, as indicated above, is generally due to failure of the API credential to satisfy the strictures of an Access Control List.

By default, the following restrictions apply:

- All API credentials can execute the `create` API call without structure.
- API credentials can execute the `update` and `delete` API calls only on users created by the `create` API call.
- API credentials can execute the `read` API call, but will be denied access to any user not attached to an event owned by the API credential's owner (VEP client user).

- The `readall` API call will only retrieve users attached to events owned by the API credential's owner (VEP client user).

Codes and Fields

Response codes:

- 0: API call failure
- 1: API call success

Default fields:

- `id` (Integer, read-only)
- `firstname` (String 64)
- `lastname` (String 64)
- `email` (String 64)
- `password` (String 30)
- `title` (String 64)
- `company` (String 64)
- `profile_image` (String 512)
- `language` (String 32)
 - The value of the language can be find at the Appendix.
- `event_id` (Integer)
- `entitlement_group` (String 128)
 - "default_group" by default. The value should match the entitlement groups added in Control Panel's Access & Entitlement.
- `registration_set` (String 80)
- `lastmodified` (String, read-only)
- `address1` (String 512)
- `address2` (String 512)
- `zipcode` (String 16)
- `city` (String 64)
- `state_province` (String 64)
- `country` (String 128)
- `country_code` (String 20)
 - This is the phone number's country code, not the ISO code of country field.
- `area_code` (String 20)
- `phone_no` (String 20)
- `extension` (String 20)
- `promo_code` (String 255)

Example of custom question fields.

- Twitter Id (String)
- Facebook Id (String)

- What's your favorite color? (String)

Valid "_apicall" values:

- read
- readall
- create
- update
- delete

Single Sign-on with Token

To login a user from third party system, the third party system needs to create a token. The token is then passed into the URL to VEP's Public API via a POST or GET method.

SSO Test Page

Production: <https://<clientDomain>.6connex.com/publicapi/test-sso>

Staging: <https://<clientDomain>.6connexstage.com/publicapi/test-sso>

Data required

The third party uses the following information to generate the token.

- API User Name: <apiUsername>
- API User Password: <apiPassword>
- Login User Email: <email>
- Target Event ID: <eventId>
- DeepLink (Optional): <deepLink>

DeepLink can be a location key of the following formats

- Room location key: r{roomId}. For example, r123.
- Space location key: s{spaceId}. For example, s123.
- Unique location key for a room or space. For example, auditorium.
- Location key with content window id (nodeId): {locationKey}/n{nodeId}. For example, auditorium/n3456.

Process

The above information needs to be encrypted using MD5 and Base64 using the following process.

Get current timestamp

```
now = (new Date()).getTime() = 1455971882468
```

Append the data

```
data = email:eventId:now:apiUsername:apiUserPsw = james.ye@hotmail.com:789:1455971882468:v7qa:test123!
```

Calculate MD5 hash

```
hash = calcMD5(data) = ef209a1fceebe0e1db5a6283def7f1a05
```

Encode Deep Link

```
deepLink = encode64(deepLink) = YXVkaXRvcml1bQ==
```

Append the token

```
token = email:eventId:now:apiUsername:hash:deepLink =
james.ye@hotmail.com:
789:1455971882468:v7qa:ef209a1fcee0e1db5a6283def7f1a05:YXVkaXRvcml1bQ=
=
```

Base64 encode the token

```
encode64(token) =  
amFtZXMueWVAaG90bWFpbC5jb206Nzg5OjE0NTU5NzE4ODI0Njg6djdxYTplZjIwOWExZmNlZWlwZTFkYyVhNjI4M2RlZjdmMWEwNTpZWfZrYVhScmNtbnRlZmF1UT09
```

Call API with POST

Path: /publicapi/users/signon2

Method: POST

Request Headers

```
- Content-Type: application/x-www-form-urlencoded
```

Request Body:

APIResponse=dGVzdDAwMUB0ZXN0bG9jYWwuyY29tOjc4OToxNDE5MDU1NTU2ODM4OnY3cWE6NGMxNDJiM2RlZDUyMGMxNWE0MjYzYWZyYTY0OTcwZDk%3D

Call API with GET

Path: /publicapi/users/signon2

Method: GET

Query Parameter:

APIResponse=dGVzdDAwMUB0ZXN0bG9jYWwuyY29tOjc4OToxNDE5MDU1NTU2ODM4OnY3cWE6NGMxNDJiM2RlZDUyMGMxNWE0MjYzYWZyYTY0OTcwZDk%3D

Complete URL:

https://<clientDomain>.6connex(stage).com/publicapi/users/signon2?

APIResponse=dGVzdDAwMUB0ZXN0bG9jYWwuyY29tOjc4OToxNDE5MDU1NTU2ODM4OnY3cWE6NGMxNDJiM2RlZDUyMGMxNWE0MjYzYWZyYTY0OTcwZDk%3D

Form Based Create User API

Path: /publicapi/users/create?eventId=<eventId>

Method: POST

Content-Type: application/x-www-form-urlencoded

Request Body:

email=test001@6connexetest.com&firstname=Test&lastname=QA&password=qwert
tyuil!&company=6Connex
Test&title=QA&language=en_US&event_id=2&entitlement_group=default_group

Response:

- 1 – Success
- 0 - Failure

Note

- URL parameter `eventId` is required.
- Field `email` is mandatory. Whether or not other fields, like `firstname`, `lastname`, `password`, `company`, `title`, are required depends on the setting of your registration form.
- The default language is "en_US". It's dependent on the configuration of the event control panel.
- "Default group" is the default entitlement group each event has. If you need a different value, please make sure it is added in Control Panel of the event.

Appendix

Supported Languages

The followings are all the languages supported in the system. Use the value in the **locale** column to send in the API request, such as ``en_US``, or ``zh_CN``, etc.

locale	name
zh_CN	Chinese (Simplified) - 简体字
en_US	English
fr_FR	French - Français
fr_CA	French (Canada) - Français
de_DE	German - Deutsch
it_IT	Italiano
ja_JP	Japanese - 日本語
ko_KR	Korean - 한국어
pl_PL	Język polski
pt_BR	Brazil - Português
ru_RU	русский язык
es_ES	Spanish - Español
es_NS	Spanish (Latin America) - Español

th_TH	Thai
-------	------