



# Connections API – Implementation Guide

---

## Introduction

Your license for the Connections API enables consuming applications to:

1. Synchronise contacts between Visit and external systems, such as CRM platforms.
2. Scan attendee QR codes using your own hardware and to submit to the API to generate a connection and return the associated contact properties.

Valid scans submitted to the API are available under the **leads** section of the **Visit Connect** user interface and are available for export to those users with the required permissions; in that respect they are no different to connections made by scanning with our own solutions.

Registered users may access the Visit Connect application at

<https://connect.visitcloud.com>

Note: This same application offers QR scanning when opening on a mobile device, subject to a scanning license being available. Without a license, QR codes can be captured, but not synchronised and decoded.

## Authentication

The Connections API uses the **OAuth2.0** workflow to authorize all incoming requests. Specifically, it employs the **Client Credentials** grant type, requiring **client ID** and **client secret** properties. In many development scenarios your consuming application will automate the process of OAuth2.0 integration for you.

To build your own authentication integration, use the following steps to request an **access token**.

Send a **POST** request to the authorization endpoint:

<https://api.visitcloud.com/connect/oauth2/token>

A successful request (http status code OK/200) will return a JSON structure including an **access token**, with a validity of 3600 seconds (1 hour). Note the access token is not refreshed on each use, therefore it will need to be renewed periodically in the case that it has expired.

Example JSON response body:

```
{
  "access_token": "adad5e3c-2808-4579-a88a-c509901449be",
  "expires_in": "3600",
  "token_type": "Bearer"
}
```

Authorize your API calls by including the following http header for all subsequent requests:

**http header**

```
Authorization: Bearer [your access token]
```

If an access token has expired or is invalid, subsequent API calls will return a status 401/unauthorized responses.

When requesting the token, for maximum compatibility with OAuth 2.0 libraries, select one of the below three methods to pass parameters.



### Method 1

Pass the **grant\_type**, **client\_id** and **client\_secret** as a JSON object in the message body. Set content type header to **application/json**:

#### http header

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

#### body

```
{  
  "grant_type": "client_credentials",  
  "client_id": "[your client ID]",  
  "client_secret": "[your client secret]"  
}
```

### Method 2

Pass the **grant\_type**, **client\_id** and **client\_secret** as a **x-www-form-urlencoded** parameters in the message body. Set content type header to **application/x-www-form-urlencoded**

#### http header

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

#### body

```
grant_type: client_credentials  
client_id: [your client ID]  
client_secret: [your client secret]
```

### Method 3

Use **basic authentication** to pass the **client\_id** as the **username**, and **client\_secret** as the **password**. This process base64 encodes the username and password and passes the result as the value of the **Authorization** request header. Set content type to **application/x-www-form-urlencoded**.

Send the **grant\_type** property as a form encoded parameter, as above

#### http header

```
Authorization: Basic [base64 encoded clientID/secret]  
Content-Type: application/x-www-form-urlencoded
```

#### body

```
grant_type: client_credentials
```

## Retrieving Connections

Send a **GET** request with the above **Authorization** header to <https://api.visitcloud.com/connect/v1/connections>



Connections are returned in a JSON **array** of up to 100 records, sorted in ascending **revision**. The revision of each record (data format is long number) can be considered as a last create/update timestamp; note that records do also expose the creation time in UTC format.

If the returned JSON array has 100 elements, the additional page(s) can be fetched by appending the **fromRevision** querystring parameter:

[https://api.visitcloud.com/connect/v1/connections?fromRevision=\[number\]](https://api.visitcloud.com/connect/v1/connections?fromRevision=[number])

For example, if the revision number of the 100<sup>th</sup> returned record is 10411084157, the subsequent **GET** request to retrieve those records with higher revision numbers would be

<https://api.visitcloud.com/connect/v1/connections?fromRevision=10411084158>

It's also possible to request a single record from the API (rather than an array) by referencing its unique alphanumeric ID in the URL. Example:

<https://api.visitcloud.com/connect/v1/connections/364e2tkmexp9t>

## Record JSON Structure

```

{
  "contact": {
    "address": {
      "address": "Business Park",
      "city": "Elstree",
      "country": "GB",
      "houseNumber": "123",
      "houseNumberSuffix": "a",
      "postalCode": "WD2 3AB",
      "state": "Hertfordshire"
    },
    "cocNumber": "CoC1234567890123",
    "company": "Visit by GES",
    "department": "Training",
    "email": "jdoe@acme.com",
    "fax": "+44 208 123456",
    "firstInitials": "R",
    "firstName": "John",
    "gender": "male",
    "jobFunction": "Software Support",
    "lastName": "Doe",
    "lastNamePrefix": "Van de",
    "phone1": "+44 208 123456",
    "phone2": "+44 208 123456",
    "pictureUrl": "https://image.visitcloud.com/s3_visitevent_91767839-46a3-4ed9-b656-260276d0bcbd",
    "suffix": "esq",
    "title": "Mr.",
    "vatNumber": "GB999 9999 73",
    "website": "www.acme.com"
  },
  "exhibitorQuestions": [
    {
      "answers": [
        {
          "id": "1zbiwx0v7",
          "name": "Medium",
          "type": "label"
        }
      ],
      "id": "1zbiwx0v7",
      "name": "Lead rating",
      "type": "radio"
    }
  ],
  "id": "364e2tkmeyp9t",
  "notes": [
    {
      "id": "1zbixfodg",
      "scanInfo": "Ab",
      "text": "This is a text note",
      "time": "2023-11-30T13:57:31Z",
      "type": "memo"
    },
    {
      "fileURL": "https://connect.visitcloud.com/note/s3_visitcontent_70a91758-4ffa-4b49-8635-6de3f051234",
      "id": "1zbixfodp",
      "scanInfo": "Ab",
      "time": "2023-11-30T13:58:03Z",
      "type": "image"
    },
    {
      "fileURL": "https://connect.visitcloud.com/note/s3_visitcontent_fab356e3-fc66-44e8-b244-3a94f9334567",
      "id": "1zbixfodu",
      "scanInfo": "Ab",
      "time": "2023-11-30T13:58:38Z",
      "type": "audio"
    }
  ],
  "profileQuestions": [
    "profileQuestions",
    :,
    [
      {
        "answers": [
          {
            "id": "1z8s4wm0y",
            "name": "Director",
            "type": "label"
          }
        ],
        "id": "1z8s4wm0y",
        "name": "What is your seniority level?",
        "type": "radio"
      }
    ]
  ],
  "revision": 10122537397,
  "scanInfo": "John Smith",
  "source": "barcode",
  "state": "completed",
  "time": "2023-08-24T11:33:54Z"
}

```

## Interpreting the data structure

- The JSON data structure is identical same for all events and clients who offer the Connections API
- The Visit QR code raw content is not exposed in the API. To reference a specific record, use its alphanumeric ID.
- No contact fields are mandatory by design. The event organiser has control over which fields are required when registering, and by consequence the depth of data exposed in the API.
- Event Organisers have responsibility for any additional formatting and validation restrictions applied to captured data.
- The API will expose submitted text data (such as names) from any input locales and languages.
- Selected **profile questions** may be exposed in the API if enabled by the event organiser. This is a global setting in the context of the event, so the same for all API users.
- When scanning QR codes using **Visit Connect** (Progressive Web App or Native App), it's possible for users to 'qualify' the records with additional data. This information is not available to records generated through this API by POSTing scans (the arrays will be empty always)
  - **exhibitorQuestions** is an array of responses to questions defined by the exhibiting company, and are unique to the API license in use. Questions may be optionally populated when capturing visitors as leads.
  - **notes** are exposed as an array in the API, covering memo notes, images/photos and voice recordings.

JSON property	Type	Comments
contact.address.address	string	
contact.address.city	string	
contact.address.country	string	Enforced 2 letter ISO code
contact.address.houseNumber	string	
contact.address.houseNumberSuffix	string	
contact.address.postalCode	string	
contact.address.state	string	
contact.cocNumber	string	Chamber of Commerce number. Limited to 20 chars, formatting not enforced
contact.department	string	
contact.email	string	Address is checked for valid format, but deliverability is not known.
contact.fax	string	Number formatting not enforced unless enabled by organiser.
contact.firstInitials	string	
contact.firstName	string	Likely to be populated
contact.gender	enum	null / "male" / "female" / "other" / "secret"
contact.jobFunction	string	
contact.lastName	string	Likely to be populated
contact.lastNamePrefix	string	Letters appearing before <b>lastName</b> . Rarely used.

<b>contact.phone1</b>	string	Organiser can confirm if relates to landline or mobile. Number formatting not enforced unless enabled by organiser.
<b>contact.phone2</b>	string	Organiser can confirm if relates to landline or mobile. Number formatting not enforced unless enabled by organiser.
<b>contact.pictureUrl</b>	url	Hyperlink to profile photo image, if captured on registration.
<b>contact.suffix</b>	string	Letters appearing after <b>lastName</b> . Rarely used.
<b>contact.title</b>	string	Organiser can optionally provide a selection list, causing standardised responses across records
<b>contact.vatNumber</b>	string	Limited to 20 chars, formatting not enforced
<b>contact.website</b>	string	
<b>exhibitorQuestions</b>	array	List of responses to exhibitor questions (if configured). The context of questions and answers can be derived from their <b>name</b> property. Question and answer <b>id</b> values are unique to the API license.
<b>id</b>	string	13 character alphanumeric unique id for the connection record. Always populated.
<b>notes</b>	array	Array of optional properties captured through the Visit Connect scanning app. <ul style="list-style-type: none"> <li>● <b>id</b> = unique ID for the note</li> <li>● <b>scanInfo</b> = username of Visit Connect app operative</li> <li>● <b>file</b> = URL if type is <b>image</b> or <b>audio</b></li> <li>● <b>text</b> = text entered by operative</li> <li>● <b>time</b> = UTC time of note</li> <li>● <b>type</b> = <b>memo</b> / <b>image</b> / <b>audio</b></li> </ul>
<b>profileQuestions</b>	array	Array of profile questions and answers captured for the registrant. Questions are only exposed if both completed by the registrant and enabled by the Event Organiser. Supported question types are single select, multi select and free text. For selection types, text values may be optionally associated with them. Typical shared questions may include displaying a registrant's area of business interest(s), as captured when registering.
<b>revision</b>	long	Refer to documentation on <b>paging</b> and <b>change tracking</b>
<b>scanInfo</b>	string	Username of Visit Connect app operative. Will be empty for scans submitted via the API.
<b>source</b>	enum	<b>Barcode</b> always (read as QR code in this context)
<b>state</b>	enum	See connection state definitions below
<b>time</b>	time	UTC time scan captured



## Connection States Explained

Visit Connect is licensed in the context of an event. As such, the app can only capture connections (leads) where the scanned badge available to the license.

Lead state	Meaning
Completed	Record was captured and synchronised correctly, either from Visit Connect or submitted via the API
NoAccess	The scanned QR code is recognised by the API, but as Valid from a different Visit Event. The lead is not valid in this context of this event and the associated licenses. The full data cannot be viewed in the API.
NoLicense	Lead has been scanned (using Visit Connect) without an active license. The full data cannot be viewed in the API.

## Synchronizing connections (leads)

A common use case is the replication of your company's contact connections to another system. To monitor the Connections API for changes (new or updated records), persist the revision number between each of your API calls and append this using **?fromRevision=[highest seen revision + 1]** querystring when requesting. That is to say, the mechanism for getting pages of connections is the same as for tracking changes. You may call the API at an interval to suit your refresh cycle, for example every minute.

**Under our fair use policy, we request the API is used responsibly – there is no added value in repeatedly requesting the entire connections dataset, requesting through multiple threads, or sending requests with unnecessary frequency. Please ensure you use a single instance of the revision tracking described above to retrieve only the new/updated data you require. API requests are monitored; abuse of the Connections API may result in your access token being restricted.**

## Submitting connection scans (leads)

Scan the visit badge QR code with your own hardware/software solution and submit to the Connections API as a POST request. Connections will be generated successfully in the case:

1. a valid OAuth 2.0 access token is in use
2. the submitted QR code value exists within the context of the licensed event

It may be possible to scan badges from concurrent related events (same event dates) if the organiser has enabled this feature.

Send a **POST** request with authorization header to

<https://api.visitcloud.com/connect/v1/connections>

http header

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

example body

```
{  
  "code": "y12abcdefghijklmnopqrsTUVwxyz" }  
}
```

Connections are generated at the time of submitting. Optionally a custom UTC scan time may be included, in the following format:

```
{  
  "code": "y12abcdefghijklmnopq1GBSmithJ",  
  "time": "2023-10-01T13:20:07Z"  
}
```

A successful POST (http status code OK/200) will return the details of the created record in the same JSON format as those returned by the GET request. If unsuccessful, the reasons are detailed in the response. **The IDs contained in the badge QR codes are not returned in the JSON data.**

### Troubleshooting

The Connections API only supports

- POST requests (for generating an OAuth2.0 access token or submitting a QR scan)
- GET requests (for retrieving an array of records or an individual record)

In all cases a 200 (OK) http response will be returned for a successful operation.

In the case of failed operations (other http statuses, normally 4xx), the response body may contain detail about the underlying cause where possible.

### Feedback / Technical Support

For technical enquiries or to report any bugs/issues. please raise a support ticket to our team of experts at [visit.support@ges.com](mailto:visit.support@ges.com)