Visit Create API

This new JSON based version of the Visit API supersedes the existing JSON API, which will be maintained for compatibility with existing integrations (as will the XML API).

Terminology

Visit Create – the administrative interface for configuring events.
https://create.visitcloud.com

Visit Connect – the online portal / mobile app for exhibiting companies (partners)
https://connect.visitcloud.com

Visit Discover – the online portal / mobile app for event participants (visitors)
https://discover.visitcloud.com

Benefits of the new API include:

- Support for powerful webhooks, enabling software notifications
- Increased efficiency of use by separating pre-event registration and event generated data
- Improvement and standardisation of field names and structures, in line with recommended API practices
- Resolution of known issues when manging visitors and partners using 3rd party IDs (references)

The create API is located at:

https://api.visitcloud.com/create/v2/

Authorization

API keys are managed by event organisers, subject to having the API module available under their contract with Visit, and the required permission on their account. Event organisers are also responsible for the management of keys with any 3rd parties undertaking integrations.

API keys may be configured as one of:

- Access to **all events** under the event organizer company
- Access to **specified events only**. This approach is recommended for increased security and traceability.

When configuring keys, they may optionally be assigned:

- **Write permission -** enables API consuming applications to create, update and delete data in Visit. The write option is NOT enabled by default, we recommend leaving disabled unless there is a known need to modify Visit data.
- **Leads permission -** enables API consuming applications to use the **connections** endpoint, which details visitor and partner (exhibitor) interactions.

API requests are authorized using **Basic Authorization**. Enter the API key as the username, password is not required.

API Endpoints

Endpoint	Description	Querystring
expos	Returns details of the events available to the API key. To see full detail of an individual event, append its ID to the request URL **Read only.**	limit=[number] fromRevision=[number] reference=[string]
visitors	Returns up to 100 registrations for the selected event. To request a specified visitor, append its ID to the URL. Visitor responses contain the full registration data including the contact details and any additional profiling questions captured. Where visitors are associated with exhibiting companies, the **partner** property is populated. Visitors: • Are captured from online forms, imports and managed using the API • Related properties are the registration type, language and registration form. • Have access to the Visit Discover mobile app. • Have badges (passes) with a QR code. The QR code value is not the same as their API ID. • May be managed via the API (with write permission)	limit=[number] fromRevision webhookId=[string] contactRefference=[string] showDeleted=[Boolean] registrationStates=[list]
partners	Returns up to 100 participating companies for the selected event. To request a specified partner, append its ID to the URL. Note that partner contacts are administrative rather than being event participants. In the case that a partner admin is participating in the event, there will be a linked record available from the viitors endpoint. Partners: • Have a partner registration type • May add associated visitors as team members, or as guests • Have an optional booth number and size • May be managed via the API (with write permission)	limit=[number] fromRevision=[number] webhookId=[string] contactRefference=[string] showDeleted=[Boolean]
participants	Returns up to 100 Visitors who took part in the event (have associated 'onsite' data) have participant records. The participants endpoint exposes the attendance timings, also details of any activities the participant has taken part in. The show/noShow property is populated according to settings for the event, as defined in Visit Create; it's possible for a participant in noShow state to have related participation data. **Read only**	limit=[number] fromRevision=[number] webhookId=[string]
contents	Returns up to 100 content records. Content may relate to the event (event information), or to partners. A partner has a **profile** content item, and optionally additional **products** dependant on the event configuration. Contents may be managed via the API with write permission.	limit=[number] fromRevision=[number] webhookId=[string]
licenses	Returns up to 100 licence items. Licences are used to enable visit hardware/software solutions in the context of both the event and exhibiting companies. Some license types may be managed via the API with write permission.	limit=[number] fromRevision=[number]
payments	For events with an eCommerce (shop) component, payments returns pages of up to 100 records. To get details of a specific payment, include its ID in the URL. For events using invoice payments, the payment state me be set using the API, subject to having write permission.	limit=[number] fromRevision=[number] webhookId=[string]
touchpoints	Touchpoints are NFC devices which event visitors can interact with using their badge. Touchpoints are linked to a content item, so are used to identify an event information item, a specific exhibiting company (partner) or a product or service that company offers.	limit=[number] fromRevision=[number]
actions	Can be considered as the raw data for visitors taking part at event (either in person and / or online) Returns a list of up to 100 interactions between visitors and any of the Visit hardware/software functions. **Read-only.**	llmit=[number] fromRevision=[number] webhookId=[string]
connections	Returns up to 100 connections made between visitors and exhibitors (partners). The data may be exhibitor-led (when using Visit Connect for lead capture), or visitor led when using Visit Discover or NFC badges to generate connections. **Read-only, required 'leads' permission on API key.**	llmit=[number] fromRevision=[number] webhookId=[string]
activities	Details activities scheduled for the event, as configured in Visit Create. Activities have a date/time, and a location. This endpoint also exposes reports of participants for each activity, as calculated based on data in the **actions** endpoint. To get full details of an activity, include its ID in the request URL. **Read-only.**	
upload	Function for uploading identity images for a visitor record. **Write only.**	
sendemail	Function for sending confirmation email associated with a registration. The sendemail function would typically be used when adding registrations through the API, as if the person had used an online form. **Requires write permission.**	
webhooks	For managing webhooks in the context of the current event. Webhooks can be created, modified, and deleted. It's not necessary to have an API key with write permissions – webhooks themselves don't modify any aspects of registrant / exhibitor / event data.	

Supported Methods

Endpoint	GET	GET (item)	POST	PUT	DELETE
expos	✔	✔			
visitors	✔	✔	✔	✔	✔
partners	✔	✔	✔	✔	✔
participants	✔	✔			
contents	✔	✔	✔	✔	✔
licenses	✔	✔	✔	✔	✔
payments	✔	✔		✔	
touchpoints	✔				
actions	✔				
connections	✔	✔			
activities	✔	✔			
upload			✔		
sendemail			✔		
webhooks	✔	✔	✔	✔	✔

	Write API permission required
	Leads API permission required

Working with the Visit API

- All **id** fields are 13 alphanumeric strings
- Most endpoints require the **expoId** to be included in the URL since requests are always in the context of an event. An exception is the **expos** endpoint, which can be called without an expoId to return a list of events available to the API key.
- For endpoints which expose arrays, append the **id** field to the URL to fetch a specific item.
- An API key with write permission is required to modify any data. This is configured by the event organizer. Note that write permission is not required to manage **webhooks**
- In the case of **PUT** and **DELETE** requests, **id** is mandatory; it's not possible to perform bulk operations from a single request.
- Most endpoints have a **revision** property, which is an incremental figure on every database edit. When obtaining a list of items, records are returned in increasing revision number order i.e. the most recent updates are towards the end.

Operation	**Verb**	**URL format**
Fetch list	GET	https://api.visitcloud.com/create/v2/<endpoint>/<expoId>
Fetch item	GET	https://api.visitcloud.com/create/v2/<endpoint>/<expoId>/<id>
Create item	POST	https://api.visitcloud.com/create/v2/<endpoint>/<expoId> Include relevant JSON body with required field(s)
Update item	PUT	https://api.visitcloud.com/create/v2/<endpoint>/<expoId>/<id> Include relevant JSON body with required field(s)
Delete item	DELETE	https://api.visitcloud.com/create/v2/<endpoint>/<expoId>/<id>

.

Querystring parameters

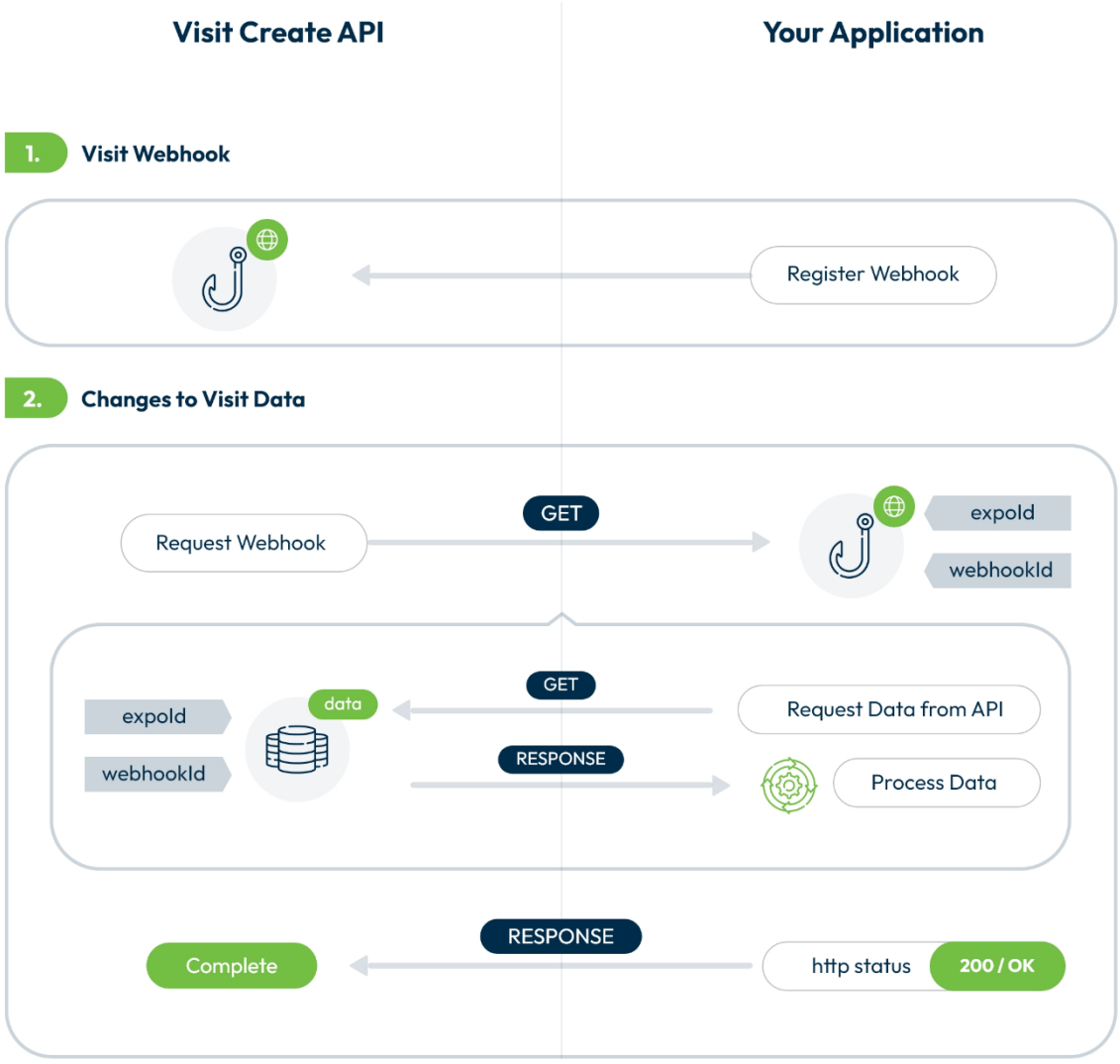
?limit=<integer>	Reduces the number of items when requesting a list of records from an endpoint. Endpoints have upper limits on the size of the arrays retuned.
?fromRevision=<revision>	Returns all records where the revision is greater to or equal to n. When working with arrays, use the **fromRevision** filter to get the additional page(s) of data. For example if the a request returns 100 records we know there may be additional page(s). If the initial request returns **9552787992** as the 100th revision, the next page can be obtained using **?fromRevision=9552787993**
?contactId=<id>	For the **visitors** or **partners** endpoints, this can be used as a lookup function to find the visitor/partner record (if exists) with the referenced contact based on Visit's contact ID. The request will return zero or one result.
?contactReference=<reference>	For the **visitors** or **partners** endpoints, this can be used as a lookup function to find the visitor/partner record (if exists) with the referenced contact based on a 'foreign key'. The request will return zero or one result.
?webhookId=<id>	Use when working with webhooks to return the relevant dataset. This function applies to endpoints that support webhooks, so: • visitors • partners • actions • participants • connections • contents • payments
?reference=<reference>	Use with the **expos** endpoint to return any events with reference field matching the supplied string. The reference field isn't enforced to be unique, so multiple records may be returned.
?registrationStates=[list]	When using the **visitors** endpoint, by default only records in **registered** state are returned (registered visitors are complete/confirmed registrations) Possible states are: • invited • registering • registered • denied • pending-confirmation • stopped • hold To query multiple states, comma separate them in a list, e.g. **?registrationStates=invited,registered,denied**

?showDeleted=<Boolean>	When using the **visitors** and **partners** endpoints to retrieve an array of records, information about deleted records (which were previously 'registered') is exposed by default. Use the **showDeleted** parameter to hide these.

Introducing Webhooks

Webhooks provide a mechanism to notify consuming applications when new/modified data is available from the Visit API. After registering webhooks with the Visit API, the registered webhooks are called automatically when changes to data at the associated endpoints occur. Webhooks are created, managed and removed through the API.

Webhooks are associated with specified events, and exist in the context of an API key. Since API keys may be limited to access certain events only, requests using a given key may only manage webhooks within the scope of allocated events.



The following 7 webhook notifications are available:

visitor	Relates to **visitors** endpoint Properties of visitor records are new/updated. Visitor data is captured and updated via online registration forms, imports, using Visit Create and via the API. Visitor data includes exhibitor (partner) team members.
partner	Relates to **partners** endpoint Properties of partner records are new/updated. Partner data is managed through imports, using Visit Create and via the API.
action	Relates to **actions** endpoint (read-only) Times and details of interactions between event participants (visitors) and any Visit systems.
participant	Relates to **participants** endpoint (read-only) Participant data is generated for every visitor that has interactions at the event. This endpoint also details timings of any activities attended
connection	Relates to **connections** endpoint (read-only) Details of connections made between visitors and exhibitors, through interactions with Visit Connect, Visit Discover and TouchPoints. This includes leads captured.
content	Relates to **contents** endpoint Changes have been made to event digital content (managed by the event organiser) or partner profile/product information (managed by partner admin accounts of the event organiser)
payment	Relates to **payments** endpoint A payment has been created or updated. Payments are associated with registrations making use of Visit's eCommerce capabilities.

Managing webhooks

Webhooks are configured and managed using the **webhooks** endpoint

GET https://api.visitcloud.com/create/v2/webhooks/<expoId>

Returns an array of webhooks created under the API key

To get full detail of a specific webhook. Including usage history:

GET https://api.visitcloud.com/create/v2/webhooks/<expoId]>/<webhook_id>

To add a new webhook:

POST https://api.visitcloud.com/create/v2/webhooks/<expoId>

Body

```
{
    "enabled": true,
    "type": "visitor",
    "url": "[target_URL]"
}
```

Type can be one of **visitor, partner, action, participant, connection, content, payment**

To modify a webhook (for example, to disable):

PUT https://api.visitcloud.com/create/v2/webhooks/<expoId]>/<webhookId>

Body

```
{
    "enabled": false
}
```

To delete a webhook

DELETE https://api.visitcloud.com/create/v2/webhooks/<expoId>/<webhookId>

Webhook Behaviour

For each webhook request sent when new/updated data is available, the target URL is called with expo and webhook querystring parameters. For example, if a webhook is configured as https://mywebhook.com

The associated call will be structured:
https://mywebhook.com?expoId=<expoId>&webhookId=<webhookId>
The consuming application will then typically read from the associated endpoint, passing the webhookId and expoId in the URL.

Important: For successful operation, the consuming application must return http **status 200** within 30 seconds. In the case of other http responses, or if the consuming application is inaccessible, the webhook request is assumed to have failed.

Failed webhook requests are re-attempted according to the following schedule:

0 minutes	Initial call after webhook created
10 seconds	Retry #1
1 minute	Retry #2
5 minutes	Retry #3
15 minutes	Retry #4
1 hour	Retry #5
1 day	Retry #6. On failure, webhook **state** property is set to **error.** Webhook requests will not be re-attempted until state is reset to **wait** (use a PUT request)

Webhook worked example

Consider it's required to set up a Visitor notification for event with ID **0rwwipz7fufs1**

The target system for consuming visitor data has an example endpoint configured at:
https://visitors.mywebhook.com

Assuming the application has already been built (which will be using the /visitors endpoint), define and activate the webhook:

POST https://api.visitcloud.com/create/v2/webhooks/0rwwipz7fufs1

BODY

```
{
    "enabled": true,
    "type": "visitor",
    "url": "https://visitors.mywebhook.com"
}
```

The POST response returns the id of the created webhook and the associated detail. Note that `"log": []` is initially an empty array, as the webhook hasn't been used yet:

RESPONSE


```
{
    "currentRevision": 10525205227,
    "enabled": true,
    "errorCount": 0,
    "id": "34bg2hx3fla87",
    "lastRevision": 0,
    "log": [],
    "sentTime": null,
    "state": "wait",
    "type": "visitor",
    "url": "https://visitors.mywebhook.com"
}
```


Note: The **enabled** property is optional., but if omitted the webhook's enabled state will be set to **false** on creation. To modify the enabled status of an existing webhook, use a PUT request as follows:

PUT https://api.visitcloud.com/create/v2/webhooks/0rwwipz7fufs1/34bg2hx3fla87

BODY

```
{
    "enabled": true/false,
}
```

When changes are made to visitor data (i.e. changes to data available at the visitors endpoint), the configured webhook is called from the Visit API:

GET https://visitors.mywebhook.com?expoId=0rwwipz7fufs1&webhookId=34bg2hx3fla87

When processing the webhook, the consuming application would typically request data from the Visit API's **visitors** endpoint, passing the **webhookId** as a parameter.

GET https://api.visitcloud.com/create/v2/visitors/0rwwipz7fufs1?webhookId=34bg2hx3fla87

On completion of processing the webhook request, the consuming application will return http response 200 (OK). On the assumption there are additional visitor records (remembering that the maximum size of the dataset is 100 records) the webhook will be re-triggered to the consuming application.

GET https://visitors.mywebhook.com?expoId=0rwwipz7fufs1&webhookId=34bg2hx3fla87

Requesting visitor data again with the same webhookId returns the next set of (up to 100) records

GET https://api.visitcloud.com/create/v2/visitors/0rwwipz7fufs1?webhookId=34bg2hx3fla87

The maximum number of items may be reduced from 100 by including a **limit** parameter in the requesting URL. Considering the 30 second limit on a request/response, for a slow process you may want to reduce the number of records returned.

GET
https://api.visitcloud.com/create/v2/visitors/0rwwipz7fufs1?webhookId=34bg2hx3fla87&limit=20

Will return up to 20 visitor records.