Technical Details

Last Updated: Wednesday, January 5, 2022

The following technical details are provided to help you better understand the Vibes APIs and Callbacks.

URL Host

The following is the standard Vibes URL host, which is the Vibes public API site.

Server Endpoints

Environment	Public URL
US	https://public-api.vibescm.com
EU	https://public-api.eu.vibes.com/ https://public-api.vibescmeurope.com/ (deprecated on 11/13/2020; may be removed in future)

Content Type

All Vibes API calls are set up to receive data in a standard JSON format. The following is the exact header to include on all API calls.

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

Authentication

Basic Authentication is the simplest way to implement access controls. Most programming languages have built-in libraries to support Basic Authentication. The following is a brief description of how the header is constructed:

1. Combine the username and password into a "username:password" string.
2. Encode the resulting string using Base64.
3. Add the Authorization HTTP header and set the value to "Basic " plus the encoded string.

Note:

- Usernames should be different in the EU and US instances.
- Include the extra space after "Basic ".

For example, if the username is 'VibesUser' and the password is "Password123" then the header is as follows:

```
Authorization: Basic VmliZXNVc2VyOlBhc3N3b3JkMTIz
```

Client Certificate Authentication

Vibes supports Client Certificate Authentication as an extra layer of security for API calls.

API HTTP Header Version

Platform APIs require you to send Vibes an HTTP `X-API-Version` with a value of `2`.

For example:

```
X-API-Version: 2
```

What's in Version 2

Error Handling and Responses

General HTTP Response Codes

APIs follow standard HTTP Response codes. See individual API calls for errors that can be returned.

Error Response Body

Any 4XX error will return a JSON body with the specific information and code. The response will be an errors Object, with an Array of all the errors that occurred on the request, as shown in the following example.

```
{"errors": [
    {"message":"The MDN field is required","code":1},
    {"message":"The zip code must be numeric","code":2}
  ]
}
```

Callbacks

Callbacks are a way for a third-party system to register and receive events from the Vibes Mobile Engagement Platform when specified data has changed. A Callback is a Vibes-initiated HTTP request to the third-party system to notify them of a data change. The customer endpoint should properly store/record the information, and then return an appropriate *HttpResponse* to acknowledge receipt of the event, keeping processing and logic as small as possible for throughput and performance.

⚠ **Note**: To ensure adequate performance and avoid rate limiting or other errors, you should keep the callback code as small as possible. To avoid connection timeouts or rate limiting from the Vibes system, any extensive back end calls or updates should be done asynchronously rather than keeping the callback HTTP call open.

One Callback can be registered per callback type, per company account. Person callbacks, because they are related to the entire MobileDB, and not to any one company, can only have one registration callback per MobileDB database.

As part of the configuration, customers can be notified of failed deliveries by setting up an automated email notification process for when a callback fails to notify.

Additional Resources

- Client Certificate Authentication for Callbacks from the Vibes Platform

Originating IP Address Ranges

The following Vibes IP address ranges are available if a customer wants to list them as allowed traffic:

US

- 35.155.139.143/32
- 52.32.61.199/32
- 35.161.244.84/32
- 18.205.120.48/32
- 52.22.43.57/32
- 18.232.9.131/32

EU

- 34.243.232.57/32
- 52.48.241.82/32
- 34.249.188.130/32
- 54.247.36.51/32
- 34.253.250.65/32
- 54.217.181.160/32

HTTP Responses

It is the receiver's responsibility to ensure processing and to return the appropriate error codes. Depending on the error code, Vibes will attempt to redeliver the events up to the maximum number of tries. It is important that the callback endpoint not return errors when no retry attempts are desired. In that case, simply accepting the message with a 2XX response is sufficient.

It is possible, in rare circumstances, that an event is delivered more than once in error scenarios, or when an improper HTTP Response code was returned or timed out. All customer end points must be able to handle this case without causing conflicts within their system, and return a response in the 2XX range.

HTTP Response Code	Description

2XX	OK - Message was accepted and processed successfully.
4XX	Permanent event specific error conditions that should not be retried. It will immediately put the event in a failure queue and notify the customer for resolution.
3XX OR 5XX	Temporary error conditions that indicate a transient failure. These events will be retried based on the configured retry scheme, and then, if still failing, it will put the event in a failure queue and notify the customer for resolution. These can also trigger downed states within a URL that may suspend additional delivery attempts for short periods of time to avoid saturating a down (or slow) service.

Callback Structure

All callback have a similar base structure to indicate the type of callback as well as specific information about each event. All the general callback information will be in the HTTP Headers for reference, parsing, and routing. The HTTP body will contain the specific data that is relevant to the callback event.

HTTP Headers

Header	required	Description
X-Event-Id	yes	Unique Event ID for the event.
X-Company-Id	yes	The Company ID that the request is for.
X-Callback-Id	yes	The Callback ID that generated this callback event.
X-Delivery-Attempts	yes	The number of delivery attempts.
X-Event-Type	yes	The free text type of the event.

HTTP Body

```
{
 <callback specific data>
}
```

Callback Codes

Code	Description	Detailed Objects
person_add	Triggered when a new Person is added to the system.	person - Person Object