

# Callback Configuration

Last Updated: Tuesday, June 15, 2021

## Overview

A callback entity is a registration to receive callback events to an external URL. It is identified by a *CallbackID*, which uniquely identifies the record. Additionally, each callback is unique for a given event type and active date range.

## Topics in this Section

- [Examples](#)
  - [Person Event Types](#)
  - [Subscription Event Types](#)
  - [Acquisition Campaign Event Types](#)
  - [Subscription List Event Types](#)
  - [Unmatched Message Event Types](#)
  - [Wallet Event Types](#)
  - [Push Device Event Types](#)

## Callback Entity

The following is the JSON representation of a callback entity within the APIs.

```
{
  "callback_id": "Hzh39LP3",
  "event_type": "person_added",
  "destination": {
    "url": "http://destination.url",
    "method": "POST",
    "content_type": "application/json"
  },
  "start_date": "2017-01-01T00:00Z",
  "end_date": "2017-02-01T00:00Z",
  "url": "/companies/:company_key/config/callbacks/:callback_id",
  "created_at": "2017-04-15T14:30Z",
  "updated_at": "2017-05-18T19:30Z"
}
```

## Elements

Data Element	Type	Description	Required	Default
callback_id	String	Vibes unique identifier for each callback record.	N/A	
event_type	String	Event type to trigger this callback.	Required	
destination.url	String	External URL to submit the callback data.	Required	
destination.method	String	Method to submit the data to the destination URL. POST and PUT are valid methods.	Optional	POST
destination.content_type	String	Content type to set when submitting the data to the destination URL.  Valid content types: <ul style="list-style-type: none"><li>• application/json</li><li>• text/json</li></ul>	Optional	application/json
start_date	Timestamp	Date this callback starts. It should be in the ISO 8601 format - for example: 2017-04-05T14:30Z.	Optional	Now
end_date	Timestamp	Date this callback ends. It should be in the ISO 8601 format - for example: 2017-04-05T14:30Z.	Optional	No end date
url	String	Unique resource URL for the callback.	N/A	
created_at	Timestamp	Date this callback was created. It will be in the ISO 8601 format - for example: 2017-04-05T14:30Z.	N/A	
updated_at	Timestamp	Date this callback was last updated. It will be in the ISO 8601 format - for example: 2017-04-05T14:30Z.	N/A	

list_id	String	Vibes unique identifier for a Subscription List.	Required for Subscription Event Types	
campaign_id	String	Vibes unique identifier for an Acquisition Campaign.	Required for Acquisition Event Types	
source_code	String	The short code, long code or alpha code you want to receive unmatched message events for.	Required for Unmatched Message Event Types	
country_code	String	The country code for the source code you want to receive unmatched message events for.	Required for Unmatched Message Event Types	
campaign_token	String	The campaign token for the wallet campaign.	Required for Wallet Item Event Types	

## Examples

### Person Event Types

The *person\_added* callback event is triggered when a new Person record is added into the Mobile Database.

To register an endpoint to receive the *person\_added* callback, follow this example

```
{
  "event_type": "person_added",
  "destination": {
    "url": "https://YOUR.DESTINATION.URL",
    "method": "POST",
    "content_type": "application/json"
  }
}
```

The *person\_updated* callback event is triggered when an existing Person record has been modified in the Mobile Database.

To register an endpoint to receive the *person\_updated* callback, follow this example

```
{
  "event_type": "person_updated",
  "destination": {
    "url": "https://YOUR.DESTINATION.URL",
    "method": "POST",
    "content_type": "application/json"
  }
}
```

[More information on Person Callbacks is available here.](#)

### Subscription Event Types

The *subscription\_added* callback event is triggered when a Person has been added to a Subscription List.

The *subscription\_removed* callback event is triggered when a Person has been removed from a Subscription List.

To register an endpoint to receive the *subscription\_added* callback, follow this example:

```
{
  "event_type": "subscription_added",
  "subscription_added": {
    "list_id": "YOUR_SUBSCRIPTION_LIST_ID"
  },
  "destination": {
    "url": "https://YOUR.DESTINATION.URL",
    "method": "POST",
    "content_type": "application/json"
  }
}
```

To register an endpoint to receive the *subscription\_removed* callback, follow this example:

```

{
  "event_type": "subscription_removed",
  "subscription_removed": {
    "list_id": "YOUR_SUBSCRIPTION_LIST_ID"
  },
  "destination": {
    "url": "https://YOUR.DESTINATION.URL",
    "method": "POST",
    "content_type": "application/json"
  }
}

```

[More information on Subscription Callbacks is available here.](#)

## Acquisition Campaign Event Types

The [ack\\_participant\\_added](#) callback event is triggered when a new Participant has been added to the Acquisition Campaign.

To register an endpoint to receive the [ack\\_participant\\_added](#) callback, follow this example:

```

{
  "event_type": "ack_participant_added",
  "ack_participant_changed": {
    "campaign_id": "YOUR_ACQUISITION_CAMPAIGN_KEY"
  },
  "destination": {
    "url": "https://YOUR.DESTINATION.URL",
    "method": "POST",
    "content_type": "application/json"
  }
}

```

The [ack\\_participant\\_changed](#) callback event is triggered when a Participant has confirmed or declined the opt-in.

To register an endpoint to receive the [ack\\_participant\\_changed](#) callback, follow this example:

```

{
  "event_type": "ack_participant_changed",
  "ack_participant_changed": {
    "campaign_id": "YOUR_ACQUISITION_CAMPAIGN_KEY"
  },
  "destination": {
    "url": "https://YOUR.DESTINATION.URL",
    "method": "POST",
    "content_type": "application/json"
  }
}

```

[More information on Acquisition Callbacks is available here.](#)

## Subscription List Event Types

The [subscription\\_list\\_added](#) callback event is triggered when a new Subscription List has been created in the Mobile Database.

To register an endpoint to receive the [subscription\\_list\\_added](#) callback, follow this example:

```

{
  "event_type": "subscription_list_added",
  "destination": {
    "url": "https://YOUR.DESTINATION.URL",
    "method": "POST",
    "content_type": "application/json"
  }
}

```

The [subscription\\_list\\_updated](#) callback event is triggered whenever a Subscription List has been modified in the Mobile Database.

To register an endpoint to receive the [subscription\\_list\\_updated](#) callback, follow this example:

```
{
  "event_type": "subscription_list_updated",
  "destination": {
    "url": "https://YOUR.DESTINATION.URL",
    "method": "POST",
    "content_type": "application/json"
  }
}
```

[More information on Subscription List Callbacks is available here.](#)

## Unmatched Message Event Types

The [unmatched\\_message\\_received](#) callback event is triggered when a message originating from a consumer does not match any campaign or system keyword.

To register an endpoint to receive the [unmatched\\_message\\_received](#) callback, follow this example:

```
{
  "event_type": "unmatched_message_received",
  "unmatched_message_received": {
    "source_code": "YOUR_SHORT_CODE",
    "country_code": "YOUR_COUNTRY_CODE"
  },
  "destination": {
    "url": "https://YOUR.DESTINATION.URL",
    "method": "POST",
    "content_type": "application/json"
  }
}
```

[More information on Unmatched Message Callbacks is available here.](#)

## Wallet Event Types

The [wallet\\_item\\_install](#) callback event is triggered when a new wallet item is installed by an end user.

To register an endpoint to receive the [wallet\\_item\\_install](#) callback, follow this example:

```
{
  "event_type": "wallet_item_install",
  "wallet_item_install": {
    "campaign_token": "YOUR_WALLET_CAMPAIGN_TOKEN"
  },
  "destination": {
    "url": "https://YOUR.DESTINATION.URL",
    "method": "POST",
    "content_type": "application/json"
  }
}
```

The [wallet\\_item\\_remove](#) callback event is triggered when a wallet item is removed by an end user.

To register an endpoint to receive the [wallet\\_item\\_remove](#) callback, follow this example:

```
{
  "event_type": "wallet_item_remove",
  "wallet_item_install": {
    "campaign_token": "YOUR_WALLET_CAMPAIGN_TOKEN"
  },
  "destination": {
    "url": "https://YOUR.DESTINATION.URL",
    "method": "POST",
    "content_type": "application/json"
  }
}
```

[More information on Wallet Callbacks is available here.](#)

## Push Device Event Types

The [device\\_added](#) callback event is triggered when a new app has been installed on a device or when a device is moved from another person to this person.

To register an endpoint to receive the *device\_added* callback, follow this example:

```
{
  "event_type": "device_added",
  "device_added": {
    "app_id": "YOUR_APPLICATION_ID"
  },
  "destination": {
    "url": "https://YOUR.DESTINATION.URL",
    "method": "POST",
    "content_type": "application/json"
  }
}
```

The [device\\_removed](#) callback event is triggered when an app has been uninstalled from a device, otherwise no longer available, or when a device is moved from this person to another person.

To register an endpoint to receive the *device\_removed* callback, follow this example:

```
{
  "event_type": "device_removed",
  "device_removed": {
    "app_id": "YOUR_APPLICATION_ID"
  },
  "destination": {
    "url": "https://YOUR.DESTINATION.URL",
    "method": "POST",
    "content_type": "application/json"
  }
}
```

The [device\\_updated](#) callback event is triggered when an Apple or Google push\_token is added or removed. It will not be triggered for other types of updates.

To register an endpoint to receive the *device\_updated* callback, follow this example:

```
{
  "event_type": "device_updated",
  "device_added": {
    "app_id": "YOUR_APPLICATION_ID"
  },
  "destination": {
    "url": "https://YOUR.DESTINATION.URL",
    "method": "POST",
    "content_type": "application/json"
  }
}
```

[More information on Push Device Callbacks is available here.](#)