SIMWOOD

STRAIGHT-TALKING, **FORWARD-THINKING**, TELECOMS

WebHooks

HTTP Events Beta

April 2020 v1.6

CONTENTS

Outbound WebHooks

You can configure your Simwood account to enable you to receive messages to your own HTTP endpoints for certain events (e.g. a new call, a call ending, or messages received)

BETA NOTICE

The functionality outlined in this document should be considered "BETA" and is provided as-is and without warranty and should not be relied on for any production commercial service. This beta test invites you to test new functionality before it is made generally available and your feedback during this time is very much valued. The specification of this API is subject to change without notice although we will endeavour to ensure any changes are backwards compatible.

REQUESTS

Only https URLs are supported for WebHooks.
TLSv1.0 minimum is required and TLSv1.2 is recommended.

All requests are made in **JSON** format with a Content-Type of text/json to a pre-configured URL. Each JSON message follows a standard 'wrapper' format with a number of consistent parameters;

Parameter	Description	Type
app	Application Name	String
id	Identifier for this message This is unique per message, where multiple events are sent for a single object (e.g. a call) there will be an id within the data block also.	String
timestamp	Timestamp for this message (ISO 8601 in UTC)	String
data	Array of JSON objects containing request data	Array

The application name is provided to facilitate serving multiple functions from a single endpoint, however we strongly recommend individual URLs for each type of WebHook.

AUTHENTICATING REQUESTS

A future update will introduce an
header, containing a signature that can be used to verify the authenticity of the message.

RESPONSE

Your platform at any provided URL should respond to requests with a **200 OK, 201 Created** or **202 Accepted** as appropriate. Other response codes will be treated as an error and result in the message being retried.

Unless otherwise stated (and required for functionality) content in the response will be ignored and is not processed or retained by Simwood. Consult each individual function for more information.

AVAILABILITY AND RETRY

If the URL does not respond, or responds with an error code (i.e. not one of 200, 201, or 202) then the message will be queued to be intelligently retried at a later time for a maximum of 72 hours, the retry interval increases with each attempt.

Please note that retries vary depending on the message type and the nature of the timeliness of the message. There is little point, for example, in notifying of a ringing call an hour later if your platform was unavailable at the time, but the CDR generated will still be sent. These delays and intervals are subject to change.

Configuring Events / WebHooks

Some events on the Simwood platform (for example a new call, blocked call, an answered call, low balance warning etc) can generate a notification to your own platform using standard HTTP "WebHooks" that are called when an event is triggered. These can be configured in the portal.

Each individual application type ("app") can be configured to a different URL, or you can configure a default URL for all WebHook requests. Each request includes an "app" parameter to allow one URL endpoint to be used for multiple event types if desired, although we would generally recommend discrete URLs for this purpose.

Authenticating Requests

A future update will introduce an **X-Simwood-Authenticate** header, containing a signature that can be used to verify the authenticity of the message.

Authenticating Source

All WebHook HTTP requests will originate from the following source networks. These should be allowed through your firewall on port 443 (unless a non-standard port is used for your service)

- 178.22.139.84/31
- 185.63.140.84/31
- 185.63.142.84/31
- 185.63.141.84/31
- 185.63.143.84/31

Call Events

Inbound Calls

An inbound call to a number on your account can generate a number of **call_inbound** events at various stages of the call (e.g. ringing, answered, hangup).

call_inbound			
New inbound call, or updated status on an existing inbound call.			
app	call_inbound		String
id	Unique ID for this event message		String
timestamp	The timestamp for this event		String
data	An object containing one or more of the following parameters;		Object
	call_id	ID for this call	String
	status	The current status of the call, one of initiated, ringing, answered, cancelled, busy, rejected, failed, unanswered, completed.	String
	trunk	The trunk associated with the number	String
	from	Caller ID in e164 format	String
	to	Destination number in e164 format	String
	origin	*Origin type (e.g. 'mobile', 'fixed', 'payphone')*	*String*
	route	*The route used for the call*	*String*
	reason	*If failed, the reason for the failure (where available)*	*String*
	q850	*The Q.850 Cause Code (where available)*	*Integer*

Example

```
{
  "app": "call_inbound",
  "id": "483e9bdf-b82f-4778-a83e-4722d7f45279",
  "timestamp": "2019-07-03 14:56:24",
  "data": {
     "call_id": "538c2815-4ca2-4e86-ad5a-c55ce267983e"
     "status": "ringing",
     "trunk": "930000-ACME",
     "from": "443301223000",
     "to": "447700900123",
  }
}
```

Please note that the format of IDs can change, and should not be relied upon to be consistent at this time. You should ensure that this is always treated as a string even if purely numeric.

Advanced Call Routing

The inbound call functionality works best with a single endpoint. Where multiple endpoints (e.g. for call hunting) are specified you may receive multiple messages (e.g. ringing / bus / rejected) for each destination leg. An upcoming update will address this and provide more information for individual legs.

Outbound Calls

An outbound call on account can generate a number of **call_outbound** events at various stages of the call (e.g. ringing, answered, hangup), together with a **cdr** event on completion of the call, or a short time afterwards, when the call is rated.

call_outbound

New outbound call, or updated status on an existing outbound call.			
app	call_outbound		String
id	Unique ID for this event message		String
timestamp	The timestamp for this event		String
data	An object containing one or more of the following parameters;		Object
	call_id	ID for this call	String
	status	The current status of the call, one of initiated, ringing, answered, cancelled, busy, rejected, failed, unanswered, completed.	String
	trunk	*The trunk associated with the call*	*String*
	tag	*The content of the X-Simwood-Tag header if present*	*String*
	from	Caller ID in e164 format	String
	to	Destination number in e164 format	String
	dest_id	*Destination ID*	*Integer*
	desc	*Destination Description*	*String*
	iso	*The destination country code (e.g. gb, de, nl)*	*String*
	reason	If status is 'failed' the reason for the failure	*String*
	q850	The Q.850 Cause Code (where available)	*Integer*

Example

```
{
  "app": "call_outbound",
  "id": "9abcf581-6951-431c-bbc2-0065630d069b",
  "timestamp": "2019-07-03 12:18:34",
  "data": {
     "call_id": "0914bd32-7ffb-48ff-8a96-55ebff912443",
     "status": "ringing",
     "trunk": "930000-ACME",
     "tag": "x403",
     "from": "443301223999",
     "to": "443301223000",
     "desc": "UK Non-Geographic 03",
     "iso": "gb"
  }
}
```

Please note that the format of IDs can change, and should not be relied upon to be consistent at this time. You should ensure that this is always treated as a string even if purely numeric.

Blocked Calls

Where a call is blocked (e.g. due to an account limit, trunk limit, or fraud prevention setting) an event is triggered as detailed below. This could be used to warn your customer, or to help inform your own monitoring.

call_blocked

Blocked call			
app	call_blocked		String
id	Unique ID for this event message		String
timestamp	The timestamp for this event		String
data	An object containing the following parameters:		Object
	direction	The direction of the call, one of **in** or **out**	String
	trunk	*The trunk associated with the call*	*String*
	tag	*The content of the X-Simwood-Tag header if present*	*String*
	from	Caller ID in e164 format	String
	to	Destination number in e164 format	String
	reason	Human readable reason this call was blocked. This is also returned in the X-Reason header on the rejection.	String

Example

```
{
  "app": "call_blocked",
  "id": "483e9bdf-b82f-4778-a83e-4722d7f45279",
  "timestamp": "2019-07-03 14:56:24",
  "data": {
     "direction": "out",
     "trunk": "930000-ACME",
     "from": "443301229999",
     "to": "447700900123",
     "reason": "Trunk rate limit of 2 calls per 30s exceeded"
  }
}
```

Please note that the format of IDs can change, and should not be relied upon to be consistent at this time. You should ensure that this is always treated as a string even if purely numeric.

Warning of Calls with Invalid Signalling

Where a call has invalid signalling (e.g. an invalid destination number, or invalid CLI) an event is triggered as detailed below. These events indicate calls that would be blocked when these are strictly enforced or, for some account types, calls that may incur a surcharge.

These can be used to identify calls that may fail, and improve the quality of traffic you send.

call_warning			
	Call that generated a warning due to invalid signalling		
app	call_warning		String
id	Unique ID for this event message		String
timestamp	The timestamp for this event		String
data	An object containing the following parameters:		Object
	direction	The direction of the call, only **out** at present	String
	trunk	*The trunk associated with the call*	*String*
	tag	*The content of the X-Simwood-Tag header if present*	*String*
	from	Caller ID in e164 format	String
	to	Destination number in e164 format	String
	reason	Human readable description of the problem with the call (e.g. Invalid CLI, or Invalid Destination). This is also returned in the X-Warning header in the response to the INVITE	String

Example

```
{
  "app": "call_warning",
  "id": "0c72e5a2-40d2-4ab3-90dc-c9a2234c63b3",
  "timestamp": "2020-04-06 12:15:38",
  "data": {
     "direction": "out",
     "trunk": "930000-ACME",
     "from": "447700900123",
     "to": "443301229999",
     "reason": "Invalid CLI"
  }
}
```

Please note that the format of IDs can change, and should not be relied upon to be consistent at this time. You should ensure that this is always treated as a string even if purely numeric.

Calls In Progress Summary

A summary of the calls in progress (identical in structure to that provided by the API endpoint at **https://api.simwood.com/v3/voice/{ACCOUNT}/inprogress/current** is available, this is sent periodically (the time interval is variable) and can be used to produce a live summary of the activity on your Simwood account, much like that shown on the Simwood Portal.

inprogress_summary

Summary of Calls in Progress			
app	inprogress_summary		String
id	Unique ID for this event message		String
timestamp	The timestamp for this event		String
data	An object containing the following parameters:		Object
	account	Simwood account number	Integer
	answeredcount	Number of calls currently answered	Integer
	approx_seconds_remaining	Based on current calls and balance, number of seconds remaining	Integer
	balance	Current Balance	Float
	balance_available	Current Available Balance (excluding any locked balance)	Float
	balance_locked	Locked Balance	Float
	callcount	Number of calls in progress (in any state)	Integer
	calls	An object designed as an associative-array for compatibility with the APIv3 endpoint, the structure is shown below	Object
	countries	An object designed as an associative-array for compatibility with the APIv3 endpoint, the structure is shown below	Object
	currency	One of GBP, EUR, or USD	String
	datetime	Timestamp this summary was generated (this is shown in UK local time for compatibility with the APIv3 endpoint)	String
	percent_available	Percent of the available balance that is used by the calls in progress.	Float
	total	Total value of calls in progress	Float

inprogress_summary

Example

```
{
    "app": "inprogress_summary",
    "id": "b71685be-6458-46d6-b6c5-1148b79029ee",
    "timestamp": "2019-09-30T09:06:34Z",
    "data": {
        "account": "9xxxxx",
        "answeredcount": 1,
        "approx_seconds_remaining": 3766340,
        "balance": 3221.742,
        "balance_available": 369.742,
        "balance_locked": 2852,
        "callcount": 1,
        "calls": {
            "2303": {
                "answeredcount": 1,
                "callcount": 1,
                "country": "GB",
                "location": "UK - Mobile - T-Mobile",
                "total": 0.013
            }
        },
        "countries": {
            "GB": {
                "answeredcount": 1,
                "callcount": 1,
                "total": 0.013
            }
        },
        "currency": "GBP",
        "datetime": "2019-09-30 10:06:34",
        "percent_available": 0,
        "total": 0.013
    }
}
```

Please note the structure of the calls and countries objects are designed to mimic an associative array for compatibility with the similar APIv3 endpoint. These are not arrays.

Calls In Progress Detail

A list of all calls in progress is available, this is pushed periodically (the time interval is variable) - this can be used to dynamically monitor your customer activity and spend by destination, tag, or trunk and used in conjunction with the Call Control API to manage calls.

inprogress_calls

Full list of calls in progress			
app		inprogress_calls	String
id		Unique ID for this event message	String
timestamp		The timestamp for this event	String
data		An object containing the following parameters:	Object
	account	Simwood account number	Integer
	answeredcount	Number of calls currently answered	Integer
	approx_seconds_remaining	Based on current calls and balance, number of seconds remaining	Integer
	balance	Current Balance	Float
	balance_available	Current Available Balance (excluding any locked balance)	Float
	balance_locked	Locked Balance	Float
	callcount	Number of calls in progress (in any state)	Integer
	calls	Array of calls containing the following elements; calldate Start time of call call_id Call ID from Source Number/CLI to Destination Number dest_id Destination ID iso ISO Country Code desc Destination Description trunk Trunk used for call tag X-simwood-tag (if set) duration Duration (to datetime) charge Charge (to datetime)	Array
	currency	One of GBP, EUR, or USD	String
	datetime	Timestamp this report was generated (this is shown in UK local time for compatibility with the APIv3 endpoint)	String
	percent_available	Percent of the available balance that is used by the calls in progress.	Float
	total	Total value of calls in progress	Float

inprogress_calls

Example

```
{
    "app": "inprogress_calls",
    "id": "12359e47-28be-4862-b4b0-af2cb7c86230",
    "timestamp": "2019-10-01 15:48:25",
    "data": {
       "answeredcount": 63,
       "approx_seconds_remaining": 21600,
       "balance": 260.17275,
       "balance_locked": 0,
       "balance_available": 260.17275,
       "callcount": 72,
       "calls": [
          {
             "calldate": "2019-10-01 15:45:14",
             "call_id": "12359e47-28be-4862-b4b0-af2cb7c86230",
             "from": "443301229999",
             "to": "447700900123",
             "dest_id": 2303,
             "iso": "GB",
             "desc": "UK - Mobile",
             "trunk": "9xxxxx-ACME",
             "tag": "ACME-TEST",
             "duration": 190,
             "charge": 0.019
          }, {
             ...
          }
       ],
      "currency": "GBP",
      "datetime": "2019-10-01 15:48:25",
      "percent_available": 0.25,
      "total": 0.663,
   }
}
```

Call Control

The **call_id** provided in the above events can be used to inspect or end a call via the Simwood API, for example to end a call in progress send a DELETE request to the following API endpoint;

https://api.simwood.com/v3/voice/{ACCOUNT}/inprogress/{CALL_ID}

DELETE End an in-progress call.

Response

```
{
    "success": true
}
```

For more information see the full Simwood API Documentation at
https://simwood.com/docs/simwood_apiv3.pdf

CDR

On completion of a call, or a short time afterwards, a CDR event is generated.

cdr				
	Completed call CDR			
	app	cdr		String
	id	Unique ID for this event message		String
	timestamp	The timestamp for this event		String
	data	An object containing one or more of the following parameters;		Object
		call_id	Unique ID for this call This will match call_inbound / call_outbound events	String
		cdr_id	The ID for this CDR record This will match your downloadable CDR	
		calldate	Date of call in form YYYY-MM-DD HH:mm:ss	String
		direction	The direction of the call, one of **in** or **out**	String
		nts_source	*For a forwarded call (e.g. PSTN route) the original number dialled that generated this call*	*String*
		trunk	*The trunk associated with the call*	*String*
		tag	*The content of the X-Simwood-Tag header if present*	*String*
		from	Caller ID in e164 format	String
		to	Destination number in e164 format	String
		dest_id	Destination ID	Integer
		todesc	Destination Description	String
		iso	*The destination country code (e.g. gb, de, nl)*	*String*
		secs_call	Duration of the call (in seconds)	Integer
		secs_billed	Billable duration of the call (in seconds)	Integer
		chg_total	The cost of call conveyance	Float
		currency	Reference currency for above (GBP, EUR or USD)	String

IMPORTANT NOTICE

CDRs from the beta WebHooks service are not, at this time, suitable as the sole means of billing.

Whilst we endeavour to ensure pushed CDRs are accurate and complete there is the possibility, during the beta period, that this service will not be available 100% of the time. In the event of any period of unavailability, CDRs generated during this time will not be retransmitted.

We therefore encourage all customers to use the API to download CDRs, at least on a daily basis, and reconcile against their own records.

Messaging Events

At this time, you can configure inbound SMS and Fax individually on a per-number basis, these features will be exposed via the standardised WebHook service in a future update.

For more information please see the main APIv3 Documentation at https:///simwood.com/docs/simwood_apiv3.pdf

Account Management Events

Updated Ratecard

Your Simwood rate card will change from time to time, we can send a notification of this to your platform to facilitate automatic buy rate updates ensuring you always have the most up-to-date rates for destinations over Simwood.

rate_update			
	New rate cards available		
app	rate_update		String
id	Unique ID for this message		String
timestamp	The timestamp for this message		String
data	An object containing the following parameters:		Object
	rate_type	Currently 'voice'	String
	date_effective	Date of fax in form YYYY-MM-DD HH:mm:ss when the new rates take effect	String
	url	The URL where the new rate card can be retrieved (CSV format)	String

Example

```
{
   "app": "rate_update",
   "id": "78506da4-c587-4a23-ad8d-c9564e232021",
   "data": {
      "rate_type": "voice",
      "date_effective": "2019-09-01 00:00:00",
      "url": "..."
   }
}
```

Trunk and Configuration Events

When a new trunk is added or removed from your Simwood account, or a configuration change is made, an event is generated. In a future update WebHooks will be provided for these to allow you to create an audit trail of these actions, or respond to unexpected changes.

This will replace the current HTTP notifications for these events.

Compliance Events

EHA Data Notice

Where a call is made to the Emergency Services (999) without valid address data these are reported to us by the EHA, and we ask that you endeavour to obtain the relevant information from your customer and update the number using the API. This WebHook allows you to automate this process.

eha_missing_data

Report of a 999 call made without valid address data

app	eha_missing_data		String
id	Unique ID for this message		String
timestamp	The timestamp for this message		String
data	An object containing the following parameters:		Object
	number	The telephone number that made the call to 999 in E.164 format	String
	calldate	The date and time of the call	String
	charge	If present, the charge applicable for processing this notice as outlined in our current Rates	Float
	charge_date	The date the charge will take effect.	String

Example

```
{
  "app": "eha_missing_data",
  "id": "e2adc8dd-9087-419f-bf90-340442287408",
  "timestamp": "2019-11-26T14:09:12Z",
  "data": {
    "number": "443301223999",
    "calldate": "2019-11-26 10:00:00",
    "charge": 150,
    "charge_date": "2019-12-16"
  }
}
```

Number Portability Events

Porting Status Update

Currently for UK GNP and NGNP only, events are generated when the status of a pending port changes, these coincide with updates to the ticket.

porting

Change of status of an outstanding GNP or NGNP porting request			
app	porting		String
id	Unique ID for this message		String
timestamp	The timestamp for this message		String
data	An object containing the following parameters:		Object
	orderid	The Order ID for this porting request	Integer
	status	The current status of the porting request (one of the below)	String
	port_date	If status is **accepted** the date of port	String
	reason	Human-readable rejection message where status is one of **rejected_lcp, rejected_rh,** or **rejected**	String
	comments	Human-readable status message	String

Example

```
{
  "app": "porting",
  "id": "46f6cac0-4e4a-4895-874a-fb1ad62f8a9d",
  "timestamp": "2019-10-21T16:45:32Z",
  "data": {
     "orderid": 72763,
     "status": "submitted_lcp",
     "comments": "Approved by Simwood, Submitted to LCP"
  }
}
```

The sequence of events you receive will differ depending on the type of port (for example a subsequent port will have a submission to the LCP and Rangeholder separately)

The possible porting statuses are shown below. At the time of writing, this is an exhaustive list although subject to additional information or additions without notice.

Status	Description
received	Porting request has been received
rejected	Port was rejected by Simwood (e.g. due to error in submission)
rejected_lcp	Port was rejected by the LCP. **reason** will contain the reason for rejection
rejected_rh	Port was rejected by the Rangeholder. **reason** will contain the reason for rejection
submitted_lcp	Port has been submitted to the LCP.
submitted_rh	Port has been submitted to the Rangeholder.
accepted_lcp	Port has been accepted by the LCP.
accepted_rh	Port has been accepted by the Rangeholder.
accepted	Port has been accepted, **port_date** will contain the agreed date of the port.
complete	Port is complete
cancel_own	Port has been cancelled (at your request)
cancel_other	Port has been cancelled (at the other party's request)
raised_error	Port was raised in error
exception	Another error occurred, see **comments** for more information

Contact

If you have any questions regarding this document, please contact us;

Simwood House
Cube M4 Business Park
Old Gloucester Road
BRISTOL
BS16 1FX

+44 (0)330 122 3000
team@simwood.com

https://www.simwood.com/

Document History

Version	Date	Author	Notes
1.6	2020-04-06	RM	Addition of call_warning specification
1.5	2019-11-26	RM	Addition of EHA Data Notice
1.4	2019-10-21	RM	Add Porting Status
1.3	2019-10-01	RM	Add Calls in Progress (Detailed)
1.2	2019-09-30	RM	Add Calls in Progress Summary
1.1	2019-09-23	RM	Add reference to Call Control functionality in APIv3
1.0	2019-08-30	RM	Initial public beta release
0.9	2019-08-30	RM	Update call_inbound specification. Remove future features.
0.8	2019-08-23	RM	Add CDRs
0.7	2019-08-21	RM	Fix ID References
0.6	2019-08-16	RM	BETA release feature freeze
0.5	2019-08-12	RM	Additional event types added