



Genius Webhooks

Table of contents

Introduction	
Welcome	3
Version history	4
Webhook Configuration	
Configuration and security options	5
Events	
Event structure	6
SessionEnded	7
AgentActivity	9
QueueInfo	11
CardPayment	12
Examples	
SessionEnded	14
AgentActivity	16
QueueInfo	17
CardPayment	18

Welcome

Webhooks are a simple and easy way to get a real-time feed of data from the Genius dialler platform into your own application. By using standard HTTP calls, you can get updates when [calls are made and received](#), [calls are queueing](#), or [agents change state](#), and integrate this data directly into your own application.

Webhooks are supported on all major development platforms.

Version history

V1.0 (06/07/2021) - Initial release

V1.1 (25/04/2021) - Added "queuenumber" to SessionEnded event

V1.2 (28/04/2022) - Additional "previewtime", "CLI" and "ID" fields added to SessionEnded event

V1.3 (03/03/2023) - Added CardPayment event

Configuration and security options

The Genius Support team will assist you in configuring Webhooks for your environment. There are some required details that you'll be asked for, along with a number of optional features that can be configured:

We'll ask you to supply a **Webhook URL** - this is the HTTP endpoint that data will be sent to. Genius Webhooks only support HTTP/S endpoints.

We'll ask you which kind of event updates you want to receive - we currently support

- [SessionEnded](#) - this supplies an update when any call session ends. This is regardless of whether the call connected to an agent or not.
 - You can also decide whether you want any record-level data in the event - for example, the firstname, postcode or any other non-dialler fields that have been imported into the campaign.
- [AgentActivity](#) - this event is triggered whenever any agent changes dialled state
- [QueueInfo](#) - this supplies data on inbound queue activity

In all cases, events can be filtered to specific campaigns, agents or queues - we'll ask you whether you want to receive data for all of those, or only particular ones. SessionEnded events can also be filtered to a particular set of outcomes, or specific outcome groups.

We can also automatically alert you if there are any issues sending Webhook events. To do this, we'll need a list of email addresses to notify if and when this happens.

Security

For security, we can "pin" the connection to a particular TLS certificate. This ensures that we'll only send data to a specific endpoint with a matching certificate. If you'd like to enable this, we'll need the TLS certificate thumbprint (also sometimes called a fingerprint). This can be obtained from viewing the certificate in any browser or certificate keystore - we can assist if you need help finding this.

We can also include an optional token in the HTTP header - this can be useful to secure the connection, and also classify the update if you have many different integrations using Webhooks. If you enable this feature, the token you specify will appear in the headers as "Genius-Webhook-Token".

All Webhook events from the Genius platform will come from fixed IP addresses, and we recommend you restrict incoming events to these source addresses:

UK - 84.22.187.53

Ireland - 217.114.172.100

North America - 54.39.74.26

Event structure

All events will be delivered to your Webhook endpoint as HTTP POSTs, with a JSON payload. They contain two common values in the body of the message:

action this describes the event

timestamp - the timestamp when the Webhook event was processed

The body will also contain a child object, specific to the type of event being sent: sessionended, agentactivity or queueinfo

The Genius Webhook service expects that the receiver will return an HTTP 200 (OK) response to any event sent. Any error returned will result in the Webhook service raising an alert.

SessionEnded

SessionEnded is sent when any dialler session is completed - either when the agent ends the call (including any post-call wrap) or when the dialler automatically dispositions the call.

The SessionEnded event will contain the following values:

Key	Value	Example
campaign	The campaign that the call was dialled under	Outbound_Test
agent	The agent that the call was connected to (if appropriate - this will be NULL if the call wasn't connected)	j.bloggs
URN	The Unique Reference Number of the dialler record, as specified in the source campaign	TEST123-001
telnumber	The telephone number that was dialled	01234567890
agentresult	The disposition code of the call, as determined by the agent. This will be NULL if the call didn't connect to an agent	120
diallerresult	The disposition code of the call, as determined by the dialler. Typical values are: 1 - Busy; the outbound call was to a line that was engaged 2 - No Answer; the outbound call reached the maximum specified ring time without being answered. 7 - Abandoned call - when the (outbound) call was answered by the customer but no agent was available, or where an inbound call was queued but hung up by the customer prior to being answered by an agent. 10 - Connected to agent 20-30 - Bad or unobtainable numbers	10
calltype	The call of call. Typical values are: 0 - Predictive outbound voice call 3 - Preview outbound voice call 4 - Progressive outbound voice call 5 - Inbound voice call 9 - Manually-dialled voice call 10 - Follow-on voice call 40 - Outbound Email 41 - Inbound Email 50 - Outbound SMS	5

	51 - Inbound SMS 70 - Live chat	
timestamplaunched	The time the call was first initiated by the dialler, or when inbound the call was first received by the dialler	2021-06-02T12:00:59.1703823+01:00
timestampconnected	The time the call was connected to an agent - or NULL if the call didn't connect	2021-06-02T12:01:09.1103893+01:00
timestampdisconnected	The time the call was disconnected from the remote party - either because they hung up or the agent terminated the call	2021-06-02T12:01:22.0401898+01:00
timestampsessionended	The time the session was completed ended	2021-06-02T12:02:10.9903156+01:00
talktime	The time, in seconds, that the agent was talking on the call. Will be 0 for non-connected calls.	12.8912736344
wraptime	The time, in seconds, that the agent spent In Wrap after the call ended. Will be 0 for non-connected calls.	48.9320393
holdtime	The time, in seconds, that the agent spend in hold during the call. This is a sub-component of talk time, not an additional time component.	2.35
ringtime	The time, in seconds, that the call was ringing at the customer side. This should only be of use in outbound calls.	12.334
previewtime	The time, in seconds, that the call spent previewing to the agent. This is only appropriate for Outbound Preview campaigns	2.22
queuetime	The time, in seconds, that the call spent in an inbound queue before being answer by an agent or terminated by the customer.	30.232
queuenumber	The queue number the inbound call was assigned to when the session ended. This will be 0 for outbound dials.	67890
CLI	The Caller Line Identifier (CLI) that was requested to be presented to the customer when dialling.	01412800275
ID	The unique call ID of this particular session. This can be used for call recording retrieval via the Data API	665847321
recorddata	A JSON array of key-value pairs, containing the record data as returned to the database by the agent (if present)	

For an example of a SessionEnded event, see [this page](#).

AgentActivity

AgentActivity is sent when an agent changes state - for example, when they go from Talking to In Wrap, or LoggedIn to LoggedOut. Typically there will be several events per agent each minute during normal dialling activity.

The AgentActivity event will contain the following values:

Key	Value	Example																																
campaign	The campaign that the call was dialled under	Outbound_Test																																
agent	The agent that the call was connected to (if appropriate - this will be NULL if the call wasn't connected)	j.bloggs																																
extension	The number of the agent's telephone extension	748312																																
queue	The queue that the agent is currently active in	67182																																
agentstate	<p>A number representing the state the agent is currently in:</p> <table border="1"> <tbody> <tr><td>0</td><td>Not Logged In</td></tr> <tr><td>1</td><td>Connecting</td></tr> <tr><td>2</td><td>Logged in and unavailable</td></tr> <tr><td>3</td><td>Waiting for a call</td></tr> <tr><td>4</td><td>Previewing a call</td></tr> <tr><td>5</td><td>Breather between calls</td></tr> <tr><td>6</td><td>Talking</td></tr> <tr><td>7</td><td>In Wrap</td></tr> <tr><td>8</td><td>Waiting for a specific call (e.g. a callback)</td></tr> <tr><td>9</td><td>Dialling manual call</td></tr> <tr><td>10</td><td>Offering transfer</td></tr> <tr><td>11</td><td>Waiting to transfer</td></tr> <tr><td>12</td><td>Being offered transfer</td></tr> <tr><td>15</td><td>Dialling Preview</td></tr> <tr><td>16</td><td>Previewing offline message</td></tr> <tr><td>17</td><td>Processing offline message</td></tr> </tbody> </table>	0	Not Logged In	1	Connecting	2	Logged in and unavailable	3	Waiting for a call	4	Previewing a call	5	Breather between calls	6	Talking	7	In Wrap	8	Waiting for a specific call (e.g. a callback)	9	Dialling manual call	10	Offering transfer	11	Waiting to transfer	12	Being offered transfer	15	Dialling Preview	16	Previewing offline message	17	Processing offline message	6
0	Not Logged In																																	
1	Connecting																																	
2	Logged in and unavailable																																	
3	Waiting for a call																																	
4	Previewing a call																																	
5	Breather between calls																																	
6	Talking																																	
7	In Wrap																																	
8	Waiting for a specific call (e.g. a callback)																																	
9	Dialling manual call																																	
10	Offering transfer																																	
11	Waiting to transfer																																	
12	Being offered transfer																																	
15	Dialling Preview																																	
16	Previewing offline message																																	
17	Processing offline message																																	
previousagentstate	A number representing the state the agent was previously in, prior to this update.	3																																

Genius Webhooks

timestampcurrentstate	The time the agent entered their current state	2021-06-02T12:00:59.1703823+01:00
timestamppreviousstate	The time the agent entered their previous state	2021-06-02T12:00:39.1101238+01:00
breakreason	The last reason code specified by the agent for leaving the campaign.	1

See [this page](#) for an example of an AgentActivity event.

QueueInfo

QueueInfo reflects the current state of each dialler queue, and will be sent when anything changes in a queue (for example, a call starts queueing). A separate event is sent for each queue.

The QueueInfo event will contain the following values:

Key	Value	Example
campaign	The campaign that the call was dialled under	Inbound_Test
queuenumber	The unique number of the dialler queue	67182
queuename	The display name of the dialler queue	Inbound Test Queue #1
callsinqueue	The number of unanswered calls currently in the queue	2
oldestcall	The timestamp of when the oldest call in the queue was received	2021-06-02T12:00:59.1703823+01:00
averageanswer	The average time to answer calls in this queue	15.123764583
averageanswer5mins	The average time to answer calls in this queue during the last 5 minutes	18.1238291
totalcalls	The total number of calls received in this queue, including any that were abandoned	192
totalabandoned	The total number of calls that were abandoned (or dropped) before they were answered by an agent	6

For an example of a QueueInfo event, see [here](#).

CardPayment

CardPayment is sent whenever a card payment is attempted. This is sent regardless of whether or not the payment is successful, and regardless of the interface used (for example, DTMF-based card payments via an agent script, Genius SafePay etc.)

The CardPayment event will contain the following values:

Key	Value	Example
campaign	The campaign that the call was dialled under	Inbound_Test
agent	The agent, if any, who took the card payment	Agent1234
timestamp	The time the card payment was attempted	2023-03-03T10:04:21.95
URN	The Unique Reference Number, if available, of the account that the payment is being taken for	REF12345678
cardtype	The Card Type being used. May not be available for all payment interfaces. Typically will be one of the following: <ul style="list-style-type: none"> • MASTERCARD or MC • VISA • Amex 	VISA
cardholder	The cardholder name. May not be available for all payment interfaces	Mr A Customer
last4digits	The last 4 digits of the card number used for payment. May not be available for all payment interfaces	4321
amount	The payment amount that was attempted. This is in the smallest currency unit - i.e. pennies when collecting GBP, cents when collecting USD or EUR.	12345 <i>(which equals £/\$/€123.45)</i>
currency	The currency used for the payment. Will be one of the following: <ul style="list-style-type: none"> • GBP • EUR • USD 	GBP
result	The result of the payment attempt. Will typically contain "Success", or a descriptive error message if the attempt failed.	Success

For an example of a CardPayment event, see [here](#).

Examples

These are some simple HTTP captures of events sent to a test Webhook receiver, and should give an illustration of the kind of traffic your service can expect to receive.

SessionEnded

This is a sample of a SessionEnded event that would be received after a call ends. This sample shows a Preview call (calltype 3) being made by agent **agent1001**, and ended with outcome 205. The option to include **recorddata** has been enabled.

```
POST https://test.net/ HTTP/1.1
Content-Type: application/json
Genius-Webhook-Token: ASampleToken
Content-Type: text/plain; charset=utf-8
Host: test.net
Content-Length: 1240
Expect: 100-continue
```

```
{
  "action": "SessionEnded",
  "timestamp": "2021-07-06T10:09:19.9036161+01:00",
  "session": {
    "campaign": "Test2",
    "agent": "agent1001",
    "URN": "test-513649415",
    "telnumber": "07976916443",
    "diallerresult": 10,
    "agentresult": 205,
    "calltype": 3,
    "timestamplaunched": "2021-07-06T10:09:07.2924254+01:00",
    "timestampconnected": "2021-07-06T10:09:14.0229213+01:00",
    "timestampdisconnected": "2021-07-06T10:09:17.5331582+01:00",
    "timestampsessionended": "2021-07-06T10:09:19.7173036+01:00",
    "talktime": 3.5102368999999998,
    "wrapptime": 2.1841454,
    "holdtime": 0,
    "ringtime": 3.1241451,
    "queuetime": 0,
    "queuenumber": 0,
    "CLI": "01412800275",
    "ID": 123456789,
    "recorddata": {
      "URN": "test-513649415",
      "Address1": null,
      "Postcode": "",
      "Tel1": "07976916443",
      "Tel2": "",
      "Tel3": "",
      "Agent": "katest",
      "Retry_Number": null,
      "SourceInfo": "",
      "filename": "ovtest",
      "callback": "",
      "Round": null,
      "Name": "8600013",
      "Bal_O_S": null,
      "Result": null,
      "Arrears": null,
      "Dues": null,
      "Collected": null,
      "Pct_Collected": null,
      "Last_Paid": null,
      "Agreement": null,
    }
  }
}
```

```
"DOB": null,  
"Carrier": null,  
"Abandon": null,  
"Skill": null,  
"SkillGroup": "43",  
"First_Name": null,  
"full_address": null,  
"Surname": "Anderson",  
"Forename": "Kenny",  
"email": "kanderson@geniusppt.com",  
"TimeZone": null,  
"MaxAttempts": 100,  
"PERSON_ID": null,  
"TestBalance": 1,  
"AMDMessage": "Abandon"  
}  
}  
}
```

AgentActivity

This is an example of agent **agent1001** going from state 2 (Logged In but unavailable) to state 0 (not logged in).

```
POST https://test.net/ HTTP/1.1
Content-Type: application/json
Genius-Webhook-Token: ASampleToken
Content-Type: text/plain; charset=utf-8
Host: test.net
Content-Length: 336
Expect: 100-continue
{
  "action": "AgentActivity",
  "timestamp": "2021-07-06T10:00:16.0958645+01:00",
  "agentactivity": {
    "campaign": "Test2",
    "agent": "agent1001",
    "extension": "7002",
    "queue": null,
    "agentstate": 0,
    "previousstate": 2,
    "timestampcurrentstate": "2021-07-06T10:00:15.860834+01:00",
    "timestamppreviousstate": "2021-07-06T09:58:15.8553223+01:00",
    "breakreason": 0
  }
}
```


QueueInfo

This QueueInfo event shows a call being queued in the queue **67890**.

```
POST https://test.net/ HTTP/1.1
Content-Type: application/json
Genius-Webhook-Token: ASampleToken
Content-Type: text/plain; charset=utf-8
Host: test.net
Content-Length: 299
Expect: 100-continue
Connection: Keep-Alive
```

```
{
  "action": "QueueInfo",
  "timestamp": "2021-07-06T10:06:58.6135464+01:00",
  "queueinfo": {
    "campaign": "In_Test",
    "queuenumber": 67890,
    "queuename": "Test queue 1",
    "callsinqueue": 1,
    "oldestcall": "2021-07-06T10:06:58.443302+01:00",
    "averageanswer": 0,
    "totalcalls": 1,
    "totalabandoned": 0,
    "averageanswer5mins": 0
  }
}
```

CardPayment

This CardPayment event shows a successful payment being taken for £123.45

```
POST https://test.net/ HTTP/1.1
ContentType: application/json
Genius-Webhook-Token: ASampleToken
Content-Type: text/plain; charset=utf-8
Host: test.net
Content-Length: 299
Expect: 100-continue
Connection: Keep-Alive
```

```
{
  "action": "CardPayment",
  "timestamp": "2023-03-03T10:04:30.8857259+00:00",
  "CardPayment": {
    "campaign": "In_Test",
    "agent": "Agent.Name",
    "timestamp": "2023-03-03T10:04:21.95",
    "URN": "123456",
    "cardtype": "Visa",
    "cardholder": "Mr A CardHolder",
    "last4digits": "9030",
    "amount": 12345,
    "currency": "GBP",
    "result": "Success"
  }
}
```