

Webhook — Event Manager

La vía **push y sin estado**: la centralita hace un HTTP a una URL de tu CRM cuando ocurre un evento de llamada (entrante, contestada, no contestada, colgada). No mantienes conexión; recibes avisos. Notifica, no controla la llamada.

Dirigido a desarrolladores e integradores

Interfaz HTTP saliente (POST a tu URL)

Dirección centralita → tu servidor

01 Qué es el Webhook

El **Webhook** es la vía por la que Conexia Infinity **notifica a tu sistema** cuando pasa algo en la telefonía. Internamente se configura como una *acción* del **Event Manager**: defines qué petición HTTP se lanza y a qué URL, y la plataforma la dispara automáticamente cuando ocurre el evento asociado.

PUSH, NO PREGUNTA-RESPUESTA

A diferencia de la *API Key* (tú preguntas, la plataforma responde), aquí es **la centralita quien inicia la petición** hacia tu URL. Y a diferencia de la *Real-time API*, **no mantienes una conexión permanente**: tu endpoint simplemente espera POSTs.

Qué cubre y qué no

Sí: avisarte de eventos de llamada (entrante, sonando, contestada, no contestada, colgada) con los datos de esa llamada, para que tu CRM registre actividad, dispare flujos o notifique a otros sistemas.

No: *controlar* la llamada (transferir, aparcar, grabar) — eso es la *Real-time API*; ni *consultar* históricos — eso es la *API Key*; ni *configurar* la centralita — eso es el panel de administración.

WEBHOOK O REAL-TIME API

Para el **Call Popup** (identificar una llamada entrante) sirven ambos. El webhook es ideal si no quieres mantener un puente vivo y te basta con «que me avisen». La Real-time API gana cuando además necesitas *controlar* la llamada o seguir su ciclo de vida completo bajo un mismo id.

02 Modelo: eventos y acciones

El Event Manager tiene dos piezas que se combinan:

Evento — el disparador: «entró una llamada», «se contestó», «se colgó»... La plataforma trae un conjunto de eventos de llamada predefinidos (ver §03).

Acción — qué hacer cuando ocurre el evento: una petición HTTP (webhook) a tu URL, con su método, cuerpo, cabeceras y autenticación (ver §05).

Asocias una o varias acciones a un evento; cuando el evento ocurre, la plataforma ejecuta esas acciones y envía los datos de la llamada a tu endpoint.

DÓNDE SE CONFIGURA

El Event Manager requiere estar habilitado en la licencia. Las acciones se definen a nivel de *tenant*. Un **Activity Monitor** registra cada disparo y si tuvo éxito o falló — úsalo para depurar.

03 Eventos de llamada

Conjunto de disparadores de llamada predefinidos. Se activan o desactivan por tenant.

Incoming Call entra una llamada

Ringing está sonando

Answered Call contestada

Unanswered Call no contestada

Hangup colgada

EVENTO	CUÁNDO SE DISPARA	USO TÍPICO EN CRM
Incoming Call	Llega una llamada al sistema	Abrir Call Popup, buscar contacto
Ringing	La llamada está sonando	Estado «entrante» en la interfaz
Answered Call	La llamada se contesta	Marcar como atendida, arrancar cronómetro
Unanswered Call	Termina sin contestarse	Registrar llamada perdida, crear tarea
Hangup	La llamada finaliza	Cerrar actividad, guardar duración

CONFIRMA EL CATÁLOGO EN TU INSTALACIÓN

La lista exacta de eventos y su comportamiento dependen de la versión y licencia. Consulta el Event Manager de *tu* panel de administración para conocer los disponibles en tu servidor.

04 Parámetros disponibles

Cada acción compone la petición con parámetros del sistema (datos de la llamada), que se referencian con **doble llave** `{{parametro}}` en la URL, el cuerpo o las cabeceras. La plataforma sustituye cada marcador por su valor real antes de enviar.

GRUPO	PARÁMETROS
Núcleo de la llamada	<code>call_id</code> , <code>caller_id</code> , <code>calling_name</code> , <code>called_number</code> , <code>direction</code> , <code>extension</code> , <code>tenant</code>
Tiempos y duración	<code>date</code> (Y-m-d), <code>time</code> (H:i:s), <code>unix_time</code> , <code>duration</code> , <code>call_date</code>
Colas / agente	<code>agent_name</code> , <code>agent_email</code> , <code>queue_id</code> , <code>queue_name</code> , <code>entry</code> , <code>exit_position</code> , <code>talk_time</code> , <code>wait_time</code> , <code>last_event</code>
Coste	<code>call_cost</code> , <code>call_rating_duration</code>

`call_id` — identificador único de la llamada. Úsalo como clave para deduplicar y para casar el evento de contestada con el de colgada.

`caller_id` / `calling_name` — quién llama (número y nombre). El número es lo que buscas en el CRM para el Call Popup.

`called_number` — la extensión / número que recibe la llamada.

`direction` — dirección de la llamada (entrante / saliente).

`duration` — duración de la llamada (útil al final, en `Hangup`).

PARÁMETROS PERSONALIZADOS

Además de los del sistema, puedes definir parámetros propios con un **Tag** (el nombre que usarás como `{{tu_tag}}`) y un valor fijo — útil para etiquetar de qué instalación o entorno viene el webhook.

05 Configurar una acción (el webhook)

Una acción es la petición HTTP que se envía cuando salta el evento. Se define con estas secciones:

General

`Name` — nombre identificativo de la acción.

`HTTP Method` — `GET` / `POST` / `PUT` / `PATCH` / `DELETE`. `POST` es lo habitual para webhooks.

`URL` — el endpoint de tu CRM que recibirá la petición.

Cuerpo (body)

- None sin cuerpo
- Form-data pares clave-valor
- x-www-form-urlencoded cuerpo urlencoded
- Raw tal cual (p. ej. JSON)

Query params y cabeceras

Puedes añadir pares clave-valor a la URL (*query params*) y cabeceras personalizadas (por ejemplo, un token de seguridad o un `Content-Type` concreto). En cualquiera de ellos el valor puede llevar `{{parametro}}`.

06 Autenticación

La acción puede firmar la petición para que tu endpoint verifique que viene de la plataforma:

MÉTODO	CÓMO FUNCIONA
No Auth	Sin credencial (no recomendado en producción).
API Key	Se envía una clave junto con la petición para que tu endpoint la valide.
Basic Auth	Usuario y contraseña incluidos en la petición.
Bearer Token	Token propio, único entre proveedor y consumidor.
OAuth 2.0	La plataforma pide un token al <code>Token URI</code> con <code>Client ID</code> + <code>Client Secret</code> y lo añade como <code>Authorization: Bearer <access_token></code> .

RECOMENDACIÓN

Para la mayoría de integraciones, **Bearer Token** es un buen equilibrio: sencillo de validar en tu endpoint y suficiente si va sobre HTTPS. Guarda el token en un gestor de secretos, no en el repositorio.

07 Ejemplos completos

POST al colgar (form-urlencoded)

ACCIÓN → TU CRM

```
POST https://<TU_CRM>/telefonía/evento
Authorization: Bearer <TOKEN>
Content-Type: application/x-www-form-urlencoded

call_id={{call_id}}&direction={{direction}}&caller_id={{caller_id}}
&calling_name={{calling_name}}&called_number={{called_number}}
&duration={{duration}}&extension={{extension}}&unix_time={{unix_time}}
// la centralita sustituye cada {{...}} por el valor real antes de enviar
```

POST con cuerpo JSON (Raw)

ACCIÓN → TU CRM

```
POST https://<TU_CRM>/telefonía/evento
Authorization: Bearer <TOKEN>
Content-Type: application/json

{
  "call_id": "{{call_id}}",
  "direction": "{{direction}}",
  "caller_id": "{{caller_id}}",
  "calling_name": "{{calling_name}}",
  "called_number": "{{called_number}}",
  "duration": "{{duration}}",
  "extension": "{{extension}}"
}
```

08 Recibir el webhook en tu CRM

Tu endpoint es el otro extremo. Un patrón robusto:

- 1 **Verifica la credencial** (el token / Basic / API Key que configuraste) antes de procesar nada. Si no cuadra, responde `401` y descarta.
- 2 **Deduplica por** `call_id`. Un mismo evento puede llegar más de una vez; no crees dos actividades para la misma llamada.
- 3 **Encola y responde rápido**. Acepta el POST, mete el trabajo en una cola y devuelve `2xx` enseguida; no hagas trabajo pesado dentro de la petición.
- 4 **Casa contestada + colgada**. Usa el `call_id` para unir el evento de `Answered` con el de `Hangup` y cerrar la actividad con su duración.

09 Seguridad y buenas prácticas

TU ENDPOINT ES UNA PUERTA ABIERTA

Un webhook es una URL pública que recibe datos. Si no la proteges, alguien podría inyectar eventos falsos (llamadas que no ocurrieron, actividades inventadas). Trátala como un endpoint sensible.

Siempre HTTPS. Los datos de la llamada (números, nombres) viajan en cada POST.

Autentica el origen. Configura la credencial (\$06) y válidala en tu endpoint antes de procesar.

Sé idempotente. Deduplica por `call_id` para no duplicar actividad ante reintentos o eventos fuera de orden.

Responde rápido. Encola el trabajo y devuelve `2xx` enseguida.

Restringe por IP si puedes. Acepta solo peticiones desde la IP de la plataforma.

No confíes ciegamente. Valida y normaliza los campos recibidos antes de usarlos.

10 Cómo empezar

- 1 **Expón un endpoint HTTPS** en tu CRM que acepte POST y valide una credencial.
- 2 **Crea una acción** en el Event Manager apuntando a esa URL, con el método, el cuerpo y la autenticación elegidos.
- 3 **Asóciala a los eventos** que te interesen (empieza por `Incoming Call` y `Hangup`).
- 4 **Prueba y depura** con el Activity Monitor; comprueba que tu endpoint deduplica y cierra la actividad correctamente.

¿Buscas las otras vías? Consulta las referencias de **API Key y ARI** y de **Real-time API y Conector MCP** en el índice de integraciones.

CONEXIA TELECOM · REFERENCIA — WEBHOOK (EVENT MANAGER)

Documento para desarrolladores. Los valores entre `<ángulos>` son marcadores; la URL y credenciales de tu endpoint las defines tú. La disponibilidad del Event Manager depende de la versión y la licencia de la instalación.