INTEGRATION GUIDE · THIRD-PARTY SYSTEMS

Connecting your systems to Paytia.

Three ways to connect a third-party application to Paytia — to **start a transaction** and get the **outcome** back. Pick the one that fits how much you want to build.

AUDIENCE
Product owners & developers
Evaluating how to integrate

LAST UPDATED
February 2026
Paytia Integration Services

THE THREE APPROACHES

From zero-code config to a fully embedded payment surface — the same secure Paytia capture underneath every option.

1 · Pull & Push — Paytia searches your system, autofills its form and writes the result back.
2 · API + Webhooks / Iframe — your app calls the Paytia API and embeds checkout, exchanging events via postMessage and webhooks.
3 · Push results only — Paytia posts the outcome to a URL you specify.

✓ Card data stays with Paytia
✓ PCI DSS Level 1
✓ Real-time & async

PAYTIA LIMITED
37th Floor, 1 Canada Square
Canary Wharf, London E14 5AA

API DOCS
paytia.com/api/docs

SUPPORT
support.paytia.com

EMAIL
techsupport@paytia.com

/01 The three approaches, side by side

Three ways to connect.

Every approach starts a Paytia transaction from your application and returns the outcome. They differ only in **how much you build** and **how tightly payment sits inside your own screens**. The secure capture — and your PCI scope reduction — is identical across all three.

01
Pull & Push
Paytia adds a search tool to its agent form. It securely logs in to your system, pulls a record to autofill the form, then pushes the result back once paid.

NO CODE · REAL-TIME

02
API + Webhooks / Iframe
Your app connects to the Paytia API and embeds checkout as an iframe or popup. Data flows through the browser postMessage layer and server webhooks.

BUILD · EMBEDDED UI

03
Push results only
The simplest connection. Paytia sends a webhook message with the transaction outcome to a URL you specify — one-way, asynchronous, nothing to embed.

MINIMAL · ASYNC

At a glance

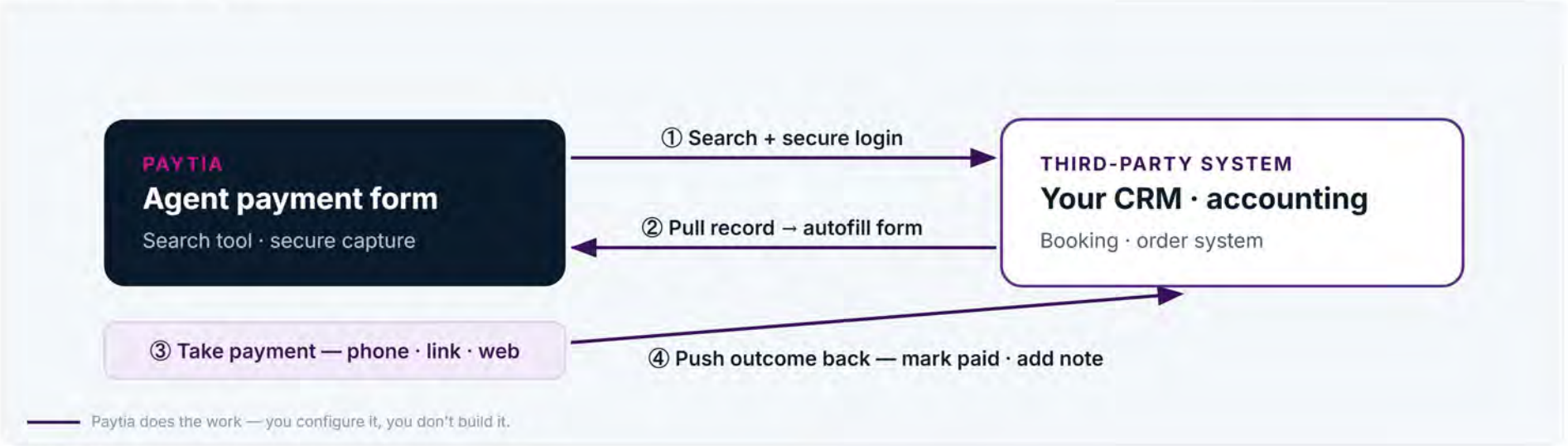
APPROACH	BUILD ON YOUR SIDE	WHERE CARD DATA FLOWS	TIMING	CHOOSE IT WHEN…
01 · Pull & Push	None — Paytia config only	Never touches your app	Real-time	Your data lives in a system Paytia can search & write to (e.g. Xero, a CRM).
02 · API + Webhooks / Iframe	Moderate — call the API, embed & listen	Inside the Paytia iframe	Real-time + async	You want payment to happen inside your own application's screens.
03 · Push results only	Minimal — expose one URL	Not applicable	Async	You only need the outcome recorded back in your system.

Common thread → In every approach the card details are captured by Paytia, so they never enter your application — keeping you at **SAQ A**. The rest of this guide takes one page per approach.

/02 Approach 01 · No development required

Pull & Push.

Paytia adds a dedicated **search tool** to its agent form. It logs in to your third-party system on your behalf, **pulls** the matching record to autofill the payment form, and after payment **pushes** the outcome straight back — marking the record paid and adding a note.



How it works

01 Your agent opens the Paytia form and uses the **built-in search** — by invoice number, name, or any field you choose.

02 Paytia signs in to your system over its API and **pulls the record**, dropping the details into the right fields — no re-keying.

03 The agent takes payment however suits — phone keypad, payment link, or web checkout — all captured securely by Paytia.

04 On completion Paytia **writes the result back**: records the payment, marks the item paid and adds a note for the audit trail.

Best for

WHEN TO PICK IT

Your records already live in a system Paytia can connect to and search — accounting platforms, CRMs and booking tools. The live example is the **Xero** integration, where invoices are searched, paid, and reconciled automatically.

What you build

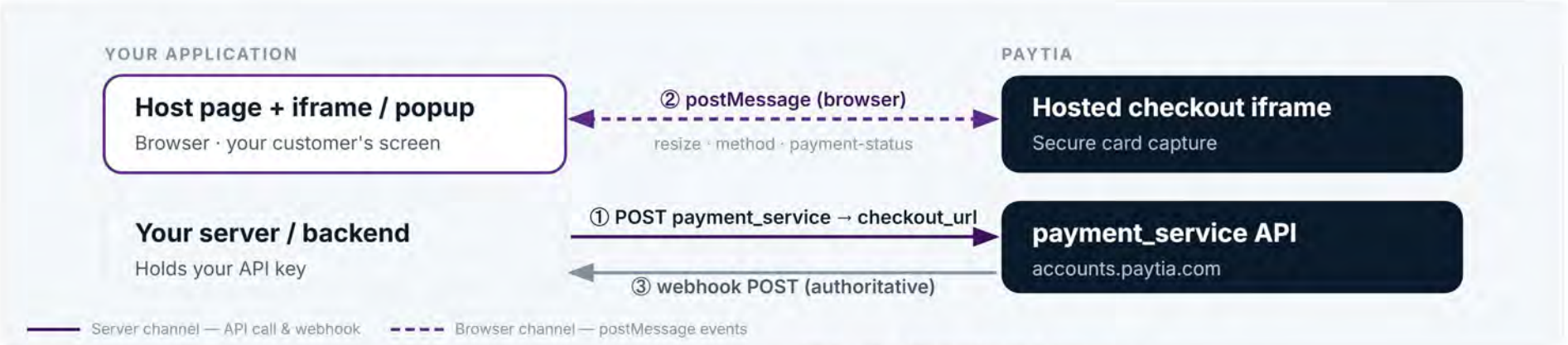
EFFORT

Nothing. This is a switch-on integration — a licence and a one-time authorisation of Paytia against your system. Your team never logs in to the third-party system to take a payment.

/03 Approach 02 · Payment inside your own app

API + Webhooks / Iframe.

Your app calls the Paytia API to start a payment and embeds secure checkout as an **iframe or popup**. Two channels carry the result back — **browser postMessage** events for live UI, and server-to-server **webhooks** for the authoritative record.



The two channels

Start the transaction from your server, embed the returned URL, then listen on both channels:

`POST` `accounts.paytia.com/api/payment_service`

CHANNEL	CARRIES
postMessage (`browser`)	Live UI — auto-resize, method selected, success / failure for your on-page confirmation.
webhook_url (`server`)	The authoritative outcome — reconcile your records against this, not the browser event.

Reconcile on webhooks → Use browser events for a responsive UI; treat the webhook as the source of truth.

Listen for the outcome (browser)

```
window.addEventListener("message", (event) => {
  // verify event.origin is a Paytia host
  const d = event.data;
  if (d.type === "payment-status" && d.status === "success")
    showThankYou(d);  // your on-page confirmation
});
```

Best for

WHEN TO PICK IT

You want the customer to pay **without leaving your application** — an in-page checkout with your own confirmation screen. See the *Iframe Controls API* and *Payment Service API* docs for full detail.

/04 Approach 03 · The simplest connection

Push results only.

The lightest-touch option. A payment is taken in Paytia as normal, and when it completes Paytia sends a **webhook message** — the transaction outcome — to a single **URL you specify**. One-way, asynchronous, nothing to embed.



How it works

01 You tell Paytia the **webhook URL** to notify — set once in your account or passed on the payment request as `webhook_url`.

02 A payment is taken through any Paytia channel — agent, IVR, payment link or checkout.

03 On completion Paytia **POSTs the outcome** to your URL as JSON. Your endpoint records it against the right order or account.

Best for

WHEN TO PICK IT

"Just tell me when it's paid." You don't need to control the payment UI — you only need the result flowing back so your system stays up to date automatically.

Example webhook payload

```
POST /your-webhook  — from Paytia
{
  "status": "success",
  "payment_status": "paid",
  "reference_id": "INV-12345",
  "amount": "127.50",
  "currency_code": "GBP",
  "card_brand": "Visa",
  "card_last_four_digit": "4242",
  "gateway_response": "Approved",
  "transaction_id": "PTX-1719408123456"
}
```

Match on your reference → Send a `reference_id` on the payment request and Paytia echoes it back, so you can tie the outcome to the right record. Webhooks are server-to-server, so the URL must be publicly reachable (not `localhost`).

/05 Which one is right for you?

Choosing your approach.

Work down these questions in order — the first one that fits is usually your answer. You can also combine them: many customers start with **Push results only** and add deeper integration later.

01 **Do your records live in a system Paytia can connect to?**
If they sit in a supported platform (e.g. Xero) or a CRM Paytia can search and write to, choose **Pull & Push** — no development, Paytia does it all.

02 **Do you want customers to pay inside your own app?**
If payment should happen on your screens with your confirmation UI, choose **API + Webhooks / Iframe** — embed checkout and listen on both channels.

03 **Do you just need the outcome recorded?**
If you only need to know when something's paid, choose **Push results only** — expose one URL and let Paytia notify it.

04 **Not sure, or need something bespoke?**
Talk to us. Paytia's team can advise on the right mix, and Bespoke Workflow Solutions can tailor an integration to your process.

Decision summary

IF YOU WANT TO...	USE
Take payment with zero development	**01 · Pull & Push**
Keep the customer inside your app UI	**02 · API + Iframe**
Simply be told the outcome	**03 · Push results only**
Reconcile against a guaranteed record	Webhooks (02 & 03)

Where to go next

DOCUMENTATION HUB
API docs, OpenAPI & Postman
`paytia.com/api/docs`

SUPPORT
Raise a ticket
`support.paytia.com/portal/en/newticket`

ONE PLATFORM · EVERY CHANNEL

Not sure which to pick?

Talk to your Paytia account manager or write to techsupport@paytia.com — we'll help you choose the simplest approach that fits your systems.

© 2026 Paytia Limited