🔑 API Keys Setup Guide

Everything a VA needs: **which key goes where, how to get each one, and how to prove it works.**

⚡ Walkthrough — do this, in this order

For administrators and VAs wiring providers into the admin panel — the order to do it in.

1

Sign in to the Admin Panel. Open `https://zaarhub.com/admin-panel.html` and sign in as an administrator.

2

Find the card for your provider. Use **Where each control lives in the Admin Panel** to jump to the card that owns the key you have.

3

Get the key from the provider. Follow the provider's block below to the exact screen where the key is generated, and copy it.

4

Paste it into the panel — not a file. Credentials belong in the admin panel (the database), never in code and never in server-wide environment variables. Enter the key in that provider's card and save.

5

Test the capability, not just the save. Trigger the feature that uses the key (a search, an email, a payment) and confirm it works end to end.

6

Check availability flags. Providers marked REQUIRED must be configured before launch; OPTIONAL ones enrich results but degrade gracefully when absent.

7

Leave the rest unconfigured if you like. An unconfigured optional provider must log and skip — never break the page.

8

See the result. Every capability you switched on is reachable from a UI control in the admin panel, and a real request proves it works.

✅ **You're done when:** every required key is entered in the admin panel, and a real call through each configured capability succeeds.

This page mirrors the platform's live provider table (`available_providers`) — **24 providers** at the time of generation — and every provider has its own section with a stable anchor you can deep-link to (`#provider-<key>`).
Generated 2026-09-21 14:45 from `SELECT key, name, description, requires_base_url, field_label, field_help, requires_metadata, icon FROM available_providers ORDER BY name;` — 24 rows.

How keys are stored (read this once)

- All keys are pasted in **one place**: the Admin Panel (`admin-panel.html`), card 🔑 **Provider API Keys**. Every provider has its own block.
- **Every key can be named.** Type a name in the **"Name this key"** field (e.g. `Primary`, `Test account`), paste the key next to it, then press **+ Add key**. You can store **several keys for the same service** — each one appears on its own line with its name and masked value.
- Exactly one key per service is the **Default** (blue badge) — that is the one the platform uses. Press **Set as default** on another line to switch. Delete one key with **Delete** on its line; the other keys for that service stay.
- Keys are saved **per tenant**, **encrypted at rest**, and always displayed **masked** (first 4 + last 4 characters). Nobody — including you after saving — can read a key back out of the panel.
- The provider badge shows the state at a glance: NOT SET or **N keys**. The **Test** button on a key line tests the *default* key for that service and prints the real API answer at the bottom of the card.
- Where a provider needs more than the key itself, the block grows an **extra input field** and the label on that field tells you what it wants — **Base URL** for most providers, **Search Engine ID** (Google Custom Search), or **DataForSEO login (account email)** for DataForSEO, whose second credential is the account email rather than a web address. Don't skip it — the integration will not work without it.
- Stuck on where to create one? Every provider block has a **"How do I get this key? →"** link that jumps straight to that provider's section on this page.
- Never put a real key in a ticket, chat, email or screenshot. If a key leaks, rotate it in the provider's console and paste the new one.

🗺 Where each control lives in the Admin Panel

WHAT YOU WANT TO DO	WHERE IT IS
Paste / name / test / delete an API key	Admin Panel → 🔑 **Provider API Keys** card (each block has its own **Name this key** + **Paste API key** + **+ Add key**)
Choose which email service a directory sends with, verify it and send a test email	Admin Panel → ✉️ **Email / SMTP** card (details below)
Save the Google Places key and populate businesses	Admin Panel → 🔍 **Discovery / Populate** (ZaarHub Admin → Populate Businesses tab)
Review/approve what a search found before it enters the directory	ZaarHub Admin → Populate Businesses → **2 · Queue — review, then add** (details below)
Connect Stripe / PayPal so customers can pay	Admin Panel → 💳 **Payment Gateways** card (details below)
Ad slots, submission queue, B2B toggles, plan/branding settings	Admin Panel nav bar (top of the panel links every card)

The panel's nav bar reaches every admin card, so nothing is hidden on an unlinked page. The same nav links back here.

✉️ Email / SMTP card — how to configure it

Email settings are **per directory**. The card sits directly under **Provider API Keys** in `admin-panel.html`.

1. **Pick the directory** in the first dropdown.
2. **Pick the transport** in the **Transport** field. The list is read live from the email service itself, so it always matches what the server can actually send with — today `smtp`, `mailgun`, `sendgrid`, `sendiio`. The badge next to the directory dropdown shows the service status and the same list.
3. **SMTP** → fill **SMTP host**, **SMTP port** (587), **SMTP username**, **SMTP password** and **Encryption** (STARTTLS / SSL / None).
   Mailgun / SendGrid / Sendiio → leave the SMTP fields as they are; the API key those transports need is the one you saved in the **Provider API Keys** card above (Mailgun, SendGrid, Sendiio).
4. Fill **From name**, **From email** and (optional) **Reply-to**, then press **Save settings**. The card shows ✓ **Saved (transport: …)** when it lands.
5. The password is **write-only**: the card shows **write-only** on the field and never displays a saved password. Leave it blank and save to keep the stored one.
6. Press **Verify credentials (real check)** — this performs a real SMTP login / real API credential call and prints the service's own answer (`ok`, `skipped` or `error` plus detail) in the box below. A canned “success” is never shown.
7. Put your own address in **Send a real test email to…** and press **Send test email**. `{"status":"sent"}` in the box means mail actually left the server; “skipped / directory_not_configured” means the save above did not land.

💳 Payment Gateways card — Stripe & PayPal

Customers pay through **Stripe** and/or **PayPal**. The card sits under **Provider API Keys** in `admin-panel.html` and writes the `payment_providers` settings. Two values are needed per gateway: the **API key** that creates the checkout, and the **webhook verification value** that proves an incoming "payment complete" message really came from the gateway.

Why the second value is mandatory: the webhook endpoint is public (the gateway has to reach it without a login). Without a verification value the server cannot tell a real payment notice from a forged one, so it refuses every event instead of marking orders paid. That is why the card — and the API behind it — will not let you switch a gateway to **Active** until the verification value is stored.

1. **Stripe.** In Stripe → Developers → API keys copy the **Secret key** (`sk_live_…` or `sk_test_…`). Paste it into **API key / client credentials**.
2. **Stripe — webhook.** Go to Stripe → Developers → Webhooks → **Add endpoint**. In **Endpoint URL** paste the address the card shows (the line under the form: `<your-domain>/api/v1/webhooks/stripe`). Under **Events to send** tick `checkout.session.completed` and `checkout.session.expired`. Save, then press **Reveal** next to **Signing secret** and copy the `whsec_…` value into **Webhook signing secret**.
3. **PayPal.** In PayPal → Apps & Credentials open your app (Live or Sandbox) and put **Client ID** and **Secret** into the card joined by a colon: `client_id:secret` in **API key / client credentials**.
4. **PayPal — webhook.** Still in the app, scroll to **Webhooks** → **Add Webhook**, set the **Webhook URL** to the address the card shows (`<your-domain>/api/v1/webhooks/paypal`) and tick `PAYMENT.CAPTURE.COMPLETED` and `CHECKOUT.ORDER.APPROVED`. After saving, copy the **Webhook ID** of that webhook into **PayPal Webhook ID** (PayPal has no signing secret; it verifies each event server-side against this ID).
5. Set **Environment**: **Test / sandbox** while you are testing (sandbox keys and the sandbox webhook), **Live** for real customers. Sandbox and live are different worlds — a sandbox webhook is never accepted in Live.
6. Set **Status** to **Active** and press **Save gateway**. The row at the top of the card must show **credentials saved** and **webhook verifiable**. If you try to activate without the verification value the card refuses and tells you which value is missing.
7. Send a real test payment and check the row list: **webhook verifiable** plus the checkout session flipping to `completed` is the proof it works end to end. Rejected events are stored too — they are the fastest way to see a wrong signing secret.
8. Both fields are **write-only**: they are encrypted at rest and never shown again. Leave a field blank and save to keep the stored value.

One gateway is enough to sell; the checkout form uses whichever gateway is Active. Deleting a gateway stops new checkouts but never deletes past orders.

🔎 Discovery queue — review before adding

Google Places results are never dumped straight into the directory. They land in a **queue** you review first (ZaarHub Admin → Populate Businesses).

1. Pick the **Directory** and the **City**, then pick a **category** — the search query is filled in for you as *<category> <city>*. You can still edit the query.
2. Press **Search & add to queue**. The rows appear in **2 · Queue — review, then add**; the counts line shows *in queue · franchise · already listed*.
3. Each row shows the business name, address, rating/review count, its **auto-mapped category** (change it in the row's **Category** dropdown at any time), and its badges: **Franchise** (big chain) and **Already listed** (duplicate of a business already in that directory).
4. Franchise and already-listed rows are **unticked by default**. **Select all eligible** ticks everything else; tick a franchise row yourself if it really is a local business you want.
5. Press **+ Add selected to directory**. The status line reports *added / skipped* and then re-reads the directory to confirm the businesses are really there. **Clear queue** empties the rows that are still waiting.
6. Reloading the page is safe — the queue is stored server-side per directory and comes back exactly as you left it.

🚀 Start here — VA checklist

Work top to bottom. After every save, click **Test** on that row and confirm it reports the key as configured before moving on.

1. **Get admin access.** Log into the Admin Panel at `admin-panel.html` with the admin account (not a business/portal account).

2. **Open the key card.** Find the card titled 🔑 **Provider API Keys** (use the panel nav bar to jump to it). It lists every provider below with a live Not set / N keys badge. **Name each key** in the "Name this key" field before you press **+ Add key** — names are how you tell two accounts of the same service apart.

3. **Google Places first** — it is the one that is actually broken right now. Create a **Google Maps Platform → Places API** key and paste it (see Google Places). This unblocks Populate / business discovery.

4. **AI drafting.** Paste a **DeepSeek** key (the default AI provider) or an **OpenAI** / **Anthropic** key. Without one of these, AI blog/content drafting cannot run.

5. **System email.** If the directory sends newsletters, notifications or receipts, paste **Mailgun**, **SendGrid** or **Sendiio** — then open the ✉️ **Email / SMTP** card, pick that transport for the directory and press **Verify credentials**, then **Send test email** (steps). SMTP needs no provider key at all: just host/username/password on the same card.

6. **Card payments.** If the directory takes card payments or pays out, paste the **Stripe** *secret* key.

7. **SMS.** If the directory texts customers, paste the **Twilio** credentials (and complete carrier/A2P registration).

8. **Captured leads → CRM.** For CoreSwift, paste only the `csk_…` key — the hub URL is preconfigured. (Advanced: paste a **Base URL** too if you run a self-hosted CoreSwift.)

9. **Optionals.** Add BrightLocal / Yext / Uberall / SerpAPI / Google Custom Search / Bing / Telnyx / GrooveSell / SAM.gov / USAspending only if the client actually uses them.

10. **Verify.** Click **Test** on every configured key line (it calls `GET /api/v1/admin/provider-keys/<provider>/test` and prints the real API answer), press **Verify credentials** + **Send test email** on the Email/SMTP card, then run one real end-to-end action per feature: one Populate search (queue → add selected), one AI draft, one test email, one test payment, one test SMS.

🗂 Which key do I need for X?

IF THE GOAL IS…	YOU MUST HAVE	NICE-TO-HAVE / ALTERNATE
Populate businesses / discovery & business search	**Google Places** (required)	BrightLocal, Yext, Uberall, SerpAPI, Google Custom Search, Bing
Sending system email (newsletters, notifications, receipts)	**One email transport:** Mailgun *or* SendGrid *or* Sendiio (SMTP also works)	The other email providers, Letterman, Mintbird
Card payments / payouts (checkout, subscriptions, paid placement)	**Stripe** (required for card payments)	GrooveSell (alternate checkout platform)
AI drafting (blogs, articles, AI keyword/question generation)	**DeepSeek** (the default AI provider) *or* OpenAI *or* Anthropic	The other two AI providers, Nexweave
SMS & voice messaging, call tracking	**Twilio** (required for SMS/call tracking)	Telnyx (alternate messaging provider)
Captured leads → CRM (enquiries, bookings, claimed listings)	**CoreSwift CRM** (key)	—
Federal contracting / spending data	— (optional data feeds)	SAM.gov, USAspending

Only **one** provider is needed per row unless stated. Providers marked Optional can stay empty without breaking anything else.

🔑 Which Google key for which job — there are TWO

Google issues a key **per API**, and using the wrong one is the most common mistake we see. The two that matter here are **not** interchangeable.

> 🔒 **Places API (New) — the SERVER key.** Powers **business search, discovery and the data-enrichment cycle**. Those calls are made by the **server**, so this key must be restricted by **server IP address** in the Google Cloud console — **never** by website referrer. The console may list the API as **Places API** or **Places API (New)**: enable the Places API, and if both names appear, both.

> 🗺️ **Maps JavaScript API — the BROWSER key.** Only needed if a **map is drawn on a page in the visitor's browser**. That key travels to the browser and is visible in the page source, so it **must** be restricted by **website / HTTP referrer** to your own domains (`*.zaarhub.com` plus any custom domains). It **cannot** do the server-side search job.

🚫 **Do not use one "master" key with both APIs enabled.** It is convenient and it is the **least safe** option: a key that must work from both a browser and a server cannot be tightened to either, so one leak exposes everything. Create two keys, each restricted properly, and never a combined one.

Where each key goes

- **Places API (New) — server key:** Admin Panel → 🔑 **Provider API Keys** → **Google Places** row → **"Paste API key"** → **+ Add key** (or ZaarHub Admin → **Populate** → **"Google Places API Key"** → **Save Key** — both write the same stored key).
- **Maps JavaScript API — browser key:** the directory's **SEO settings** → **"Google Maps API Key"** field. ZaarHub's public maps use Leaflet/OpenStreetMap and need no Google key at all — set this only if a page renders a Google map.
- **Scope:** a key saved at **network level is inherited by every city in that network**, so paste the ZaarHub key **once** — a standalone directory keeps its own. Pick the scope in the row's **Scope** dropdown before adding the key.
- **Changing a key:** you do not need the old value — paste the new one. Keys are **encrypted at rest** and shown **masked (first 4 + last 4)**; an **empty field never means "no key"** — the row says when a key is stored and when it last changed.
- **Never paste a key into a chat, a ticket, a document or a code file.** Enter it in the panel. If a key has been shared, **rotate it**.

If it still fails — symptom → fix

- `The provided API key is invalid` → wrong *type* of key (most often a Maps JavaScript key in the Places field) or the key was deleted/regenerated. Create a **Places API (New)** key and paste it.

- `… not authorized to use this service` / `API not enabled` → the Places API is not enabled on that project, or the key's **API restrictions** exclude it. Enable the API and add it to the key's allowed APIs.
- `Requests from referrer … are blocked` on a server-side search → the key is restricted by **website referrer** instead of the server's IP. Switch to **IP addresses**.
- `This IP, site or mobile application is not authorized…` when a map loads → the browser key is IP-restricted or your domain is missing. Switch it to **HTTP referrers** and add `*.zaarhub.com` plus each custom domain.
- Search works but the **map is blank** → the browser key is missing or wrong; set **Google Maps API Key** in SEO settings.

Deep-link to this section with `#google-two-keys`. The same guidance, written for administrators, is in the Admin

Guide — section 19. ↑ Back to top

📋 All 24 providers at a glance

	PROVIDER	KEY (ANCHOR)	NEED	WHAT IT IS
🧠	Anthropic (Claude)	`anthropic`	Optional	Claude models for long-form AI content
🅱️	Bing Web Search	`bing`	Optional	Bing Web Search API (Azure Cognitive Services)
⭐	BrightLocal	`brightlocal`	Optional	Citation building + local rank grid
crm	CoreSwift CRM	`coreswift`	Optional	Push captured leads (directory enquiries, booking requests, claimed listings) into your CoreSwift CRM. Connect with a personal API key (csk_...) from CoreSwift Integration Center.
🔑	DataForSEO	`dataforseo`	Optional	Keyword ideas + search volume for the Blog/QA keyword tool. Needs BOTH the DataForSEO account email (login) and the API password.
🧠	DeepSeek	`deepseek`	Recommended	DeepSeek LLM API
🔍	Google Custom Search	`google_cse`	Optional	Programmable Search Engine JSON API
🗺️	Google Places	`google_places`	Required	Local business data API — listings, reviews, citations
🛒	GrooveSell	`groovesell`	Optional	Checkout, affiliate and upsell platform
newspaper	Letterman	`letterman`	Optional	Newsletter content delivery

✉️	Mailgun	`mailgun`	Optional	Transactional + campaign email
📧	Mintbird	`mintbird`	Optional	Cold-email outreach + warmup automation
video	Nexweave	`nexweave`	Optional	Personalized video/image generation
🤖	OpenAI	`openai`	Optional	GPT models for AI copy, summaries and drafts
shield	SAM.gov	`sam_gov`	Optional	Federal contracting opportunities
✉️	SendGrid	`sendgrid`	Optional	Transactional + campaign email
mail	Sendiio	`sendiio`	Optional	Email/SMS campaign delivery
🔎	SerpAPI	`serpapi`	Optional	Google/Bing SERP scraping for rank tracking
💳	Stripe	`stripe`	Required	Card payments, subscriptions, invoices and payouts
phone	Telnyx	`telnyx`	Optional	SMS/Voice API
📞	Twilio	`twilio`	Required	SMS + voice messaging
dollar-sign	USAspending	`usaspending`	Optional	Federal spending data
📍	Uberall	`uberall`	Optional	Listings management + local SEO
🗺️	Yext	`yext`	Optional	Listings sync + knowledge graph

Discovery & Listings

🅱️ Bing Web Search OPTIONAL `bing`

Bing Web Search API (Azure Cognitive Services)

1. What it powers

- Content Research → **Bing Search API** integration.
- Alternate search/discovery source.

2. How to get the key

1. Sign in to the Azure Portal.
2. Create a **Bing Search** resource under *Cognitive Services* (Bing Search v7) — this requires the Azure subscription to have a payment method attached.
3. Open the new resource and go to **Keys and Endpoint**.
4. Copy **Key 1** (either key works).

3. Where to paste it

Admin Panel → card 🔑 **Provider API Keys** → **Bing Web Search** row → **“Paste API key”** → **Save**.

4. Required or optional — and what breaks without it

Optional. Note that Microsoft has been retiring Bing Search API tiers, so confirm the resource type is still offered in the Azure region you pick before buying anything.

5. Cost & plan notes

Azure bills per transaction for the Bing Search resource; a free/low-cost tier may be offered depending on region and offer. A credit card is required to create the resource. Verify on the Azure pricing page.

6. How to verify it

Click **Test** on the row (`GET /api/v1/admin/provider-keys/bing/test`).

Deep-link to this section with `#provider-bing`. ↑ Back to top

⭐ BrightLocal OPTIONAL `brightlocal`

Citation building + local rank grid

1. What it powers

- Scraper/discovery source (`brightlocal`) in the Scraper provider list.
- Citation building and local rank-grid data.

2. How to get the key

1. Sign in to app.brightlocal.com.
2. Open the account/profile area and go to the API / integrations settings (BrightLocal exposes API access from the account settings area).
3. Generate an API key for the account and copy it to a password manager.

3. Where to paste it

Admin Panel → card 🔑 **Provider API Keys** → **BrightLocal** row → **“Paste API key”** → **Save**.

4. Required or optional — and what breaks without it

Optional. It is an alternate discovery/citation source alongside Google Places — leave it empty if you don't have a BrightLocal plan.

5. Cost & plan notes

BrightLocal is a paid subscription product with no meaningful free tier; API access comes with the plan. Check their pricing page for current plans.

6. How to verify it

Click **Test** on the row (`GET /api/v1/admin/provider-keys/brightlocal/test`) and confirm it reports the key is configured with the masked preview.

Deep-link to this section with `#provider-brightlocal`. ↑ Back to top

🔑 DataForSEO OPTIONAL `dataforseo`

Keyword ideas and search volume for the Blog/QA keyword tool

1. What it powers

- The **Blog & QA** keyword tool when the keyword source is set to **DataForSEO** (`POST /api/v1/blog-qa/fetch-keywords` with `source: "dataforseo"`).
- Nothing else — every other feature keeps working without it.

2. How to get the credentials

1. Sign in at app.dataforseo.com.
2. Open **API Access** (dashboard). It shows your **login** — which is the **email address on the account** — and your **API password**.
3. Copy both. DataForSEO uses HTTP Basic auth, so the pair is used as `login:api password`. This is a **separate password** from the one you sign in with, and it can be regenerated on that page.

3. Where to paste it

Admin Panel → card 🔑 **Provider API Keys** → **DataForSEO** row:

- **Paste API key** field → the **API password** from the API Access page (stored encrypted at rest).
- the extra field, labelled **"DataForSEO login (account email)"** → your **account email**. It is *not* a web address, even though it travels in the same column as the other providers' Base URLs.
- press **+ Add key**.

4. Required or optional — and what breaks without it

Optional. Without it, choosing the DataForSEO keyword source returns *"DataForSEO is not configured…"* and suggests the other keyword sources (AI-generated or AnswerThePublic).

5. Cost & plan notes

DataForSEO is prepaid pay-as-you-go: you buy account balance and each keyword request draws from it. There is a minimum top-up and no free tier, so confirm the client wants paid keyword data before buying balance.

6. How to verify it

The **Test** button (`GET /api/v1/admin/provider-keys/dataforseo/test`) only proves the key is *stored* — it returns the masked preview and does **not** call DataForSEO, so it cannot tell a wrong pair from a right one.

To prove the credentials really work, run a keyword fetch with `source: "dataforseo"`. A wrong login / API password (or an empty account balance) comes back as **DataForSEO refused the request (HTTP 401)** — regenerate the API password on the API Access page and paste it again. A working pair returns keyword rows instead of that error.

Deep-link to this section with `#provider-dataforseo`. ↑ Back to top

🔍 Google Custom Search OPTIONAL `google_cse`

Programmable Search Engine JSON API

1. What it powers

- Content Research → **Google Custom Search** integration (Programmable Search Engine JSON API).
- Alternate search/discovery source.

2. How to get the key

1. Sign in to Google Cloud Console and pick (or create) a project.
2. Go to **APIs & Services** → **Library**, search for **Custom Search API** and click **Enable**.
3. Go to **APIs & Services** → **Credentials** → **Create credentials** → **API key** and copy it.
4. Create the search engine itself at programmablesearchengine.google.com (**Add** / **Create** a Programmable Search Engine).
5. Open the engine's **Setup** / overview and copy its **Search engine ID** — this is the `cx` value.

3. Where to paste it

Two values go in. Admin Panel → card 🔑 **Provider API Keys** → **Google Custom Search** row: paste the API key into **"Paste API key"** and the Programmable Search Engine ID into the extra field labelled **"Search Engine ID (cx)"** → **Save**.

> **Extra field:** this provider also needs **"Search Engine ID (cx)"** — paste it into the extra input on its row, not into the key field.

4. Required or optional — and what breaks without it

Optional. This provider is one of the few that needs **two** values — the API key *and* the `cx` Search Engine ID. A key without the `cx` will not work.

5. Cost & plan notes

The Custom Search API is free up to a daily query quota, then billed per 1,000 queries; the Programmable Search Engine itself is free. Check the current quota on the Google Cloud Custom Search pricing page.

6. How to verify it

Click **Test** on the row (`GET /api/v1/admin/provider-keys/google_cse/test`) and confirm the masked key preview comes back.

Deep-link to this section with `#provider-google_cse`. ↑ Back to top

🗺️ Google Places REQUIRED `google_places`

Local business data API — listings, reviews, citations

> ⚠ **The key currently stored is INVALID.** Google replies `400 The provided API key is invalid.` for `search?query=dental&city=palm-bay`. This key **must be replaced** by a fresh Google Maps Platform → Places API key created with the steps above. Until then, business discovery is broken.

1. What it powers

- ZaarHub Admin → **Populate** tab → **Search Places** — the city + category + radius business search that finds businesses and publishes them to the directory.
- Place **autocomplete** and **place details** lookups when adding a business by hand.
- Business **enrichment** in the Data Company module (phone, website, hours, reviews).
- The scraper/discovery source list as the `google_places` source.

2. How to get the key

1. Sign in to Google Cloud Console with the Google account that owns (or will own) the billing.
2. Create a project (or pick an existing one) for this directory — e.g. *Multi-Directory Places*.
3. Enable billing on that project. Google Maps Platform keys are **free-tier-with-credit-card**: you must attach a card, you get a recurring free credit, and usage above it is billed per request.
4. Go to **APIs & Services** → **Library**, search for **Places API** and click **Enable** (enable **Places API (New)** too if it is offered — the search uses the Places endpoints).
5. Go to **APIs & Services** → **Credentials** → **Create credentials** → **API key**.
6. (Recommended) Click **Edit API key** and add an **Application restriction** of **IP addresses** — and enter the **server's IP** (this key is used server-side for the Populate search and enrichment; it must **not** be restricted by website referrer) — plus an **API restriction** limited to the **Places API**. This limits the damage if the key leaks. A **Maps JavaScript** key is a *different* key for maps drawn in the browser — see Which Google key for which job.
7. Copy the key — it starts with `AIza…`. Never paste it into a ticket, chat or email.

3. Where to paste it

Admin Panel (`admin-panel.html`) → card 🔑 **Provider API Keys** → the **Google Places** row → paste into the **“Paste API key”** field → **Save**. The same key can also be pasted from **ZaarHub Admin** (`zaarhub-admin.html`) → **Populate** tab → **“Google Places API Key”** field → **Save Key**. Both write the same stored key.

4. Required or optional — and what breaks without it

Required for Populate / discovery. Without a valid key the Populate search and place details return nothing and the Data Company enrichment cannot resolve a provider.

5. Cost & plan notes

Google Maps Platform requires a **billing account with a credit card**. A recurring monthly free credit is applied, then requests are billed pay-as-you-go. See the current credit and per-request rates on Google's *Maps Platform pricing* page — do not rely on remembered numbers.

6. How to verify it

On the Google Places row click **Test** → `GET /api/v1/admin/provider-keys/<provider>/test` returns the **masked** key preview, never the raw key. For a real end-to-end check open **ZaarHub Admin** → **Populate** → **Test Key**, which actually calls Google Autocomplete (`POST /api/v1/zaarhub/admin/provider-keys/google-places/test`) and fails loudly if the key is invalid.

Deep-link to this section with `#provider-google_places`. ↑ Back to top

🔎 SerpAPI OPTIONAL `serpapi`

Google/Bing SERP scraping for rank tracking

1. What it powers

- Content Research → search integration (**SerpAPI** — the recommended SERP option in `content-research.html`).
- SERP scraping for rank tracking / discovery.

2. How to get the key

1. Sign up / sign in at serpapi.com.
2. Open **Dashboard** → your **API Key** panel.
3. Copy the key shown there.

3. Where to paste it

Admin Panel → card 🔑 **Provider API Keys** → **SerpAPI** row → **“Paste API key”** → **Save**. It can also be added from the Content Research page's integration form.

4. Required or optional — and what breaks without it

Optional — only needed if you want SERP/rank data in Content Research.

5. Cost & plan notes

SerpAPI offers a free plan with a limited monthly search allowance, then paid plans. See their pricing page for current limits.

6. How to verify it

Click **Test** on the row (`GET /api/v1/admin/provider-keys/serpapi/test`).

Deep-link to this section with `#provider-serpapi`. ↑ Back to top

📍 Uberall OPTIONAL `uberall`

Listings management + local SEO

1. What it powers

- Listings management and local SEO data.
- Scraper/discovery source (`uberall`) in the Scraper provider list.

2. How to get the key

1. Sign in to your Uberall account (uberall.com → login).
2. In the dashboard, open the API / integrations settings and create an API key (Uberall issues keys per account/location group).
3. Copy the key.

3. Where to paste it

Admin Panel → card 🔑 **Provider API Keys** → **Uberall** row → **“Paste API key”** → **Save**.

4. Required or optional — and what breaks without it

Optional. Alternate listings/SEO source; not needed for the core Populate flow.

5. Cost & plan notes

Uberall is a paid listings-management platform; API access requires an Uberall plan. Check with the client whether one exists.

6. How to verify it

Click **Test** on the row (`GET /api/v1/admin/provider-keys/uberall/test`).

Deep-link to this section with `#provider-uberall`. ↑ Back to top

🗺️ Yext OPTIONAL `yext`

Listings sync + knowledge graph

1. What it powers

- Listings sync and knowledge-graph data.
- Scraper/discovery source (`yext`) in the Scraper provider list.

2. How to get the key

1. Sign in to your Yext account (yext.com → sign in).
2. Open the account settings / developer area and create an API key (Yext issues API keys tied to the account).
3. Copy the key.

3. Where to paste it

Admin Panel → card 🔑 **Provider API Keys** → **Yext** row → **“Paste API key”** → **Save**.

4. Required or optional — and what breaks without it

Optional. Only useful if the client already runs listings through Yext.

5. Cost & plan notes

Yext is an enterprise subscription platform — API access requires a Yext contract. Confirm with the client whether an account exists before pursuing this.

6. How to verify it

Click **Test** on the row (`GET /api/v1/admin/provider-keys/yext/test`) and confirm the configured/masked response.

Deep-link to this section with `#provider-yext`. ↑ Back to top

AI & Content

🧠 Anthropic (Claude) OPTIONAL `anthropic`

Claude models for long-form AI content

1. What it powers

- AI content drafting — Claude is selectable as the drafting model (`blog_generator.rs` provider switch).
- Content Research → content-drafting integration option.

2. How to get the key

1. Sign in at dashboard.anthropic.com.
2. Open **API Keys** and click **Create Key**.
3. Copy the key (`sk-ant-…`) — it is shown once.
4. Add credit under **Billing** before use.

3. Where to paste it

Admin Panel → card 🔑 **Provider API Keys** → **Anthropic (Claude)** row → **“Paste API key”** → **Save**.

4. Required or optional — and what breaks without it

Optional — a third AI drafting option alongside DeepSeek and OpenAI.

5. Cost & plan notes

Pay-as-you-go with prepaid credits; a credit card is required. See Anthropic's pricing page for current model rates.

6. How to verify it

Click **Test** on the row (`GET /api/v1/admin/provider-keys/anthropic/test`).

Deep-link to this section with `#provider-anthropic`. ↑ Back to top

🧠 DeepSeek RECOMMENDED `deepseek`

DeepSeek LLM API

1. What it powers

- **The default AI drafting provider.** The blog generator defaults `llm_provider` to `deepseek`, so blog drafts, article copy and AI keyword/question generation run through DeepSeek unless a different provider is selected.
- AI blog + content drafting via `blog_generator.rs` (`call_llm_json`).

2. How to get the key

1. Sign up / sign in at platform.deepseek.com.
2. Open **API Keys** in the left nav and click **Create new API key**.
3. Copy the key immediately (`sk-…`) — some consoles only show it once.
4. Top up the account balance: the DeepSeek API is prepaid, so a funded balance is required before any call succeeds.

3. Where to paste it

Admin Panel → card 🔑 **Provider API Keys** → **DeepSeek** row → **“Paste API key”** → **Save**.

4. Required or optional — and what breaks without it

Recommended — set this first if you use AI drafting, because it is the default provider. Without *any* AI provider key, blog generation and AI content drafting fail.

5. Cost & plan notes

Pay-as-you-go, prepaid balance; a credit card is needed to top up. Per-token pricing is set by DeepSeek — check their pricing page for current rates.

6. How to verify it

Click **Test** on the row (`GET /api/v1/admin/provider-keys/deepseek/test`), then run a short draft in the blog generator to confirm real output.

Deep-link to this section with `#provider-deepseek`. ↑ Back to top

newspaper Letterman OPTIONAL `letterman`

Newsletter content delivery

1. What it powers

- Newsletter content delivery.

2. How to get the key

1. In the provider's dashboard, open **Settings** → **API Keys** (or **Integrations**) and create a key.
2. Copy the key to a password manager.

3. Where to paste it

Admin Panel → card 🔑 **Provider API Keys** → **Letterman** row → **“Paste API key”** → **Save**.

4. Required or optional — and what breaks without it

Optional. Only configure if the client's newsletter flow actually uses Letterman.

5. Cost & plan notes

Vendor-run pricing — confirm with the client whether a Letterman account exists and what it costs.

6. How to verify it

Click **Test** on the row (`GET /api/v1/admin/provider-keys/letterman/test`).

Deep-link to this section with `#provider-letterman`. ↑ Back to top

📧 Mintbird OPTIONAL `mintbird`

Cold-email outreach + warmup automation

1. What it powers

- Cold-email outreach and warmup automation.

2. How to get the key

1. In the provider's dashboard, open **Settings** → **API Keys / Integrations** and generate a key.
2. Copy it.

3. Where to paste it

Admin Panel → card 🔑 **Provider API Keys** → **Mintbird** row → **“Paste API key”** → **Save**.

4. Required or optional — and what breaks without it

Optional — outreach automation only.

5. Cost & plan notes

Vendor-run pricing; check the client's account before configuring.

6. How to verify it

Click **Test** on the row (`GET /api/v1/admin/provider-keys/mintbird/test`).

Deep-link to this section with `#provider-mintbird`. ↑ Back to top

video Nexweave OPTIONAL nexweave

Personalized video/image generation

1. What it powers

- Personalized video / image generation.

2. How to get the key

1. In the provider's dashboard, open **Settings** → **API / Integrations** and create an API key.
2. Copy it.

3. Where to paste it

Admin Panel → card 🔑 **Provider API Keys** → **Nexweave** row → **“Paste API key”** → **Save**.

4. Required or optional — and what breaks without it

Optional — personalized media only.

5. Cost & plan notes

Vendor-run pricing; confirm an account exists before configuring.

6. How to verify it

Click **Test** on the row (`GET /api/v1/admin/provider-keys/nexweave/test`).

Deep-link to this section with `#provider-nexweave`. ↑ Back to top

🤖 OpenAI OPTIONAL `openai`

GPT models for AI copy, summaries and drafts

1. What it powers

- AI blog + content drafting (`blog_generator.rs` → `call_openai` / `call_llm_json`) when OpenAI is the selected provider.
- AI copy, summaries and drafts elsewhere in the app.
- Fallback AI keyword/question generation for keyword research.

2. How to get the key

1. Sign up / sign in at platform.openai.com.
2. Open **API keys** (under your account / dashboard) and click **Create new secret key**.
3. Name it (e.g. *multi-directory export*) and copy the `sk-…` value — it is shown only once.
4. Add a payment method and credit under **Billing** so requests are not rejected.

3. Where to paste it

Admin Panel → card 🔑 **Provider API Keys** → **OpenAI** row → **“Paste API key”** → **Save**.

4. Required or optional — and what breaks without it

Optional — an alternative to (or addition to) DeepSeek. Only one AI provider key is required for AI drafting to work.

5. Cost & plan notes

Pay-as-you-go with prepaid credits; a credit card is required. Rate/price details live on OpenAI's pricing page.

6. How to verify it

Click **Test** on the row (`GET /api/v1/admin/provider-keys/openai/test`), then generate a short draft with OpenAI selected.

Deep-link to this section with `#provider-openai`. ↑ Back to top

Email

✉️ Mailgun OPTIONAL `mailgun`

Transactional + campaign email

1. What it powers

- Transactional + campaign email delivery.
- One of the selectable per-directory email **transports** for the in-house email service on `127.0.0.1:3456` (the directory's transport can be SMTP, Mailgun, SendGrid or Sendiio — the choice is stored per directory and read at send time).
- System mail: newsletter sends, notifications and receipts routed through the email service.

2. How to get the key

1. Sign in at app.mailgun.com (create an account if the client has none).
2. Add and verify the **sending domain**, then add the DNS records Mailgun gives you — delivery will not work until the domain shows as verified.
3. Open **Settings** → **API Keys** and create (or copy) an API key.
4. Copy the key and keep it in a password manager.

3. Where to paste it

Admin Panel → card 🔑 **Provider API Keys** → **Mailgun** row → **“Paste API key”** → **Save**. Then, **per directory**, set the email transport to **Mailgun** in the directory's email settings so the email service actually uses it.

4. Required or optional — and what breaks without it

Optional as a provider, required as a transport choice: if a directory's email transport is set to Mailgun, this key must be present or that directory's system email will not send. (SMTP is the default transport, so nothing breaks until you deliberately switch.)

5. Cost & plan notes

Mailgun has a free tier for low monthly volume, then paid plans; a card is required for higher tiers and some verification. Check app.mailgun.com pricing for current limits.

6. How to verify it

Click **Test** on the row (`GET /api/v1/admin/provider-keys/mailgun/test`) for the masked key preview, then send a real test email through the directory (newsletter test send) to confirm the transport works end to end.

Deep-link to this section with `#provider-mailgun`. ↑ Back to top

✉️ SendGrid OPTIONAL `sendgrid`

Transactional + campaign email

1. What it powers

- Transactional + campaign email delivery.
- One of the selectable per-directory email **transports** for the in-house email service on `127.0.0.1:3456`.
- System mail: newsletter sends, notifications and receipts routed through the email service.

2. How to get the key

1. Sign in at app.sendgrid.com (create an account if needed).
2. Complete **Sender Authentication** (verify the sending domain and add the DNS records SendGrid provides).
3. Open **Settings** → **API Keys** → **Create API Key**.
4. Give it a name and grant at least the **Mail Send** permission (a restricted key is safer than full access).
5. Copy the key (`SG.…`) immediately — SendGrid shows it once.

3. Where to paste it

Admin Panel → card 🔑 **Provider API Keys** → **SendGrid** row → **“Paste API key”** → **Save**. Then set the directory's email transport to **SendGrid** in its email settings.

4. Required or optional — and what breaks without it

Optional as a provider, required as a transport choice: if a directory's transport is SendGrid, this key must exist or that directory's system email will not send.

5. Cost & plan notes

SendGrid has a free tier for low monthly volume, then paid plans; paid tiers need a card. Check app.sendgrid.com pricing for current limits.

6. How to verify it

Click **Test** on the row (`GET /api/v1/admin/provider-keys/sendgrid/test`), then send a test email through the directory to confirm the transport.

Deep-link to this section with `#provider-sendgrid`. ↑ Back to top

mail Sendiio OPTIONAL `sendiio`

Email/SMS campaign delivery

1. What it powers

- Email / SMS campaign delivery.
- One of the selectable per-directory email **transports** for the in-house email service on `127.0.0.1:3456`.

2. How to get the key

1. Sign in to the Sendiio dashboard.
2. Open **Settings** → **API / Integrations** and generate an API key (Sendiio exposes API access from the account settings).
3. Copy the key.

3. Where to paste it

Admin Panel → card 🔑 **Provider API Keys** → **Sendiio** row → **“Paste API key”** → **Save**. Then set the directory's email transport to **Sendiio** in its email settings.

4. Required or optional — and what breaks without it

Optional as a provider, required as a transport choice: if a directory's transport is Sendiio, this key must exist or that directory's system email will not send.

5. Cost & plan notes

Sendiio is a paid subscription product. Confirm the plan/pricing in the Sendiio dashboard before choosing it as a transport.

6. How to verify it

Click **Test** on the row (`GET /api/v1/admin/provider-keys/sendiio/test`).

Deep-link to this section with `#provider-sendiio`. ↑ Back to top

Payments

🛒 GrooveSell OPTIONAL `groovesell`

Checkout, affiliate and upsell platform

1. What it powers

- Checkout, affiliate and upsell platform — an alternate checkout provider.

2. How to get the key

1. Sign in to the GrooveSell dashboard.
2. Open **Settings** → **API / Integrations** and copy (or generate) the API key for the account.

3. Where to paste it

Admin Panel → card 🔑 **Provider API Keys** → **GrooveSell** row → **“Paste API key”** → **Save**.

4. Required or optional — and what breaks without it

Optional — only if the client sells through GrooveSell instead of Stripe.

5. Cost & plan notes

Vendor-run pricing (checkout platforms usually charge per transaction rather than a flat API fee). Confirm with the client.

6. How to verify it

Click **Test** on the row (`GET /api/v1/admin/provider-keys/groovesell/test`).

Deep-link to this section with `#provider-groovesell`. ↑ Back to top

💳 Stripe REQUIRED `stripe`

Card payments, subscriptions, invoices and payouts

1. What it powers

- **Card payments at checkout** (`checkout_handler.rs` resolves the active provider `stripe` and stores its credentials as the active payment provider).
- Subscriptions, invoices and payouts.
- Marketplace / listing upgrades and any paid placement flow that goes through checkout.

2. How to get the key

1. Create or sign in to an account at dashboard.stripe.com.
2. Complete the business/identity verification so payouts can be enabled (Stripe needs business details and a bank account — this is more than just a credit card).
3. Open **Developers** → **API keys** (in Test mode first if you want to dry-run).
4. Copy the **Secret key** (`sk_live_…` for live, `sk_test_…` for testing). The **publishable** key (`pk_…`) is *not* the one you paste here.
5. (Recommended) Use a **restricted key** scoped to the payment endpoints instead of the full secret key.

3. Where to paste it

Admin Panel → card 🔑 **Provider API Keys** → **Stripe** row → **“Paste API key”** → **Save** (paste the **secret** key).

4. Required or optional — and what breaks without it

Required if the directory takes card payments or pays out. Without it, checkout fails and paid placements cannot be charged. Payments are simply unavailable if you never enable them.

5. Cost & plan notes

No monthly fee; Stripe charges a per-transaction fee and holds the funds. Live keys require a verified Stripe account with bank details. Check Stripe's pricing page for current rates.

6. How to verify it

Click **Test** on the row (`GET /api/v1/admin/provider-keys/stripe/test`) to confirm the masked key, then run a real test-mode checkout end to end before switching to live keys.

Deep-link to this section with `#provider-stripe`. ↑ Back to top

SMS & Voice

phone Telnyx OPTIONAL `telnyx`

SMS/Voice API

1. What it powers

- SMS / voice API — an alternate messaging provider to Twilio.

2. How to get the key

1. Sign in at portal.telnyx.com.
2. Open **API Keys** and click **Create API Key**.
3. Copy the key (shown once) and buy the number you will send from.

3. Where to paste it

Admin Panel → card 🔑 **Provider API Keys** → **Telnyx** row → **“Paste API key”** → **Save**.

4. Required or optional — and what breaks without it

Optional — use instead of Twilio if the client already runs Telnyx. Do not configure both unless you intend to.

5. Cost & plan notes

Pay-as-you-go with a card on file; check portal.telnyx.com pricing for current rates.

6. How to verify it

Click **Test** on the row (`GET /api/v1/admin/provider-keys/telnyx/test`).

Deep-link to this section with `#provider-telnyx`. ↑ Back to top

📞 Twilio REQUIRED `twilio`

SMS + voice messaging

1. What it powers

- **SMS and voice messaging.**
- Call-tracking numbers (the `twilio_numbers` records power call tracking).
- Phone-number intelligence / lookups (the Contact Intelligence module calls the Twilio Lookups API).

2. How to get the key

1. Create or sign in to an account at console.twilio.com.
2. Complete phone/identity verification — carriers require a verified sender before you can send SMS.
3. Open **Account → API keys & tokens** and create a **Standard** API key.
4. Copy the **SID** and the **Secret** (the secret is shown once) — and note your **Account SID** from the console home.
5. Buy (or port) the phone number you will send from, and if you are messaging US numbers, complete the required A2P 10DLC brand/campaign registration or your messages will be blocked.

3. Where to paste it

Admin Panel → card 🔑 **Provider API Keys** → **Twilio** row → **“Paste API key”** → **Save**. Paste the credential the app expects for Twilio (API key secret / auth token as documented by the integration).

4. Required or optional — and what breaks without it

Required for SMS/voice and call tracking. Without it, no text messages send and call-tracking number provisioning fails. Everything else works fine without it.

5. Cost & plan notes

Pay-as-you-go: a trial credit is included and a credit card is required to upgrade to sending at volume. A2P registration may have its own fees. Check console.twilio.com pricing for current rates.

6. How to verify it

Click **Test** on the row (`GET /api/v1/admin/provider-keys/twilio/test`), then send one real SMS to a phone you control.

Deep-link to this section with `#provider-twilio`. ↑ Back to top

CRM & Federal Data

crm CoreSwift CRM OPTIONAL coreswift

Push captured leads (directory enquiries, booking requests, claimed listings, loyalty members) into your CoreSwift CRM. Connect with a personal API key (csk_...) from CoreSwift Integration Center.

1. What it powers

- Every lead this directory captures is pushed into your **CoreSwift CRM** as a contact: directory enquiries/submissions, visitor booking requests, claimed listings and loyalty members (native integration — `coreswift::push_lead_to_coreswift`).
- The **Integrations** panel (view id `integrations`) → CoreSwift card → **Connect**.
- Capture never breaks when CoreSwift is not connected: the lead is kept locally and nothing is pushed.

2. How to get the key

1. Sign in to CoreSwift and open the **Integration Center**.
2. Generate a **personal API key** — CoreSwift keys start with `csk_`.
3. Nothing else: the hub URL is resolved for you (`provider_keys.base_url` → preset → default). Only for a **self-hosted** CoreSwift do you also paste a base URL.

3. Where to paste it

Business Portal → 🔌 **Integrations** (view id `integrations`) → **CoreSwift CRM** card → paste the `csk_…` key and press **Connect**. The same key can be pasted in the Admin Panel → card 🔑 **Provider API Keys** → **CoreSwift CRM** row. Either way the credential is stored per tenant (BYOK) and is never echoed back in full — reads show a masked preview.

> **No extra field required:** the key alone is enough. A **Base URL** input is shown only while the key is not stored, for self-hosted CoreSwift instances — leave it empty for the standard hub.

4. Required or optional — and what breaks without it

Optional — but this is the CRM the whole directory flows into. With no key, captures still succeed locally; the leads simply never reach CoreSwift. With a key, every captured lead lands in your CoreSwift account as a contact tagged with `source:multidirectory` (source app `multidirectory`).

5. Cost & plan notes

Internal/first-party product — there is no per-call API fee. Licensing is handled separately.

6. How to verify it

Press **Test connection** on the card (`POST /api/v1/provider-keys/coreswift/test`) — it calls the hub live and reports how many lists it can see; `GET /api/v1/integrations/coreswift/status` tells you whether a key is stored, and `GET /api/v1/integrations/coreswift/lists` proxies the hub's list picker. Then capture one real lead (submit a business enquiry) and confirm the contact appears in CoreSwift.

Deep-link to this section with `#provider-coreswift`. ↑ Back to top

shield SAM.gov OPTIONAL `sam_gov`

Federal contracting opportunities

1. What it powers

- Federal contracting opportunities data (SAM.gov listings).

2. How to get the key

1. Sign in at sam.gov with a registered account (registration is free but requires identity verification, which can take time).
2. Open your **Account Details** and use the **API key** section (the “Get API Key” action) to request/rotate your key.
3. Copy the key from the account page.

3. Where to paste it

Admin Panel → card 🔑 **Provider API Keys** → **SAM.gov** row → **“Paste API key”** → **Save**.

4. Required or optional — and what breaks without it

Optional — only relevant if the directory surfaces federal contracting opportunities.

5. Cost & plan notes

Free — SAM.gov API keys are issued to registered users at no charge.

6. How to verify it

Click **Test** on the row (`GET /api/v1/admin/provider-keys/sam_gov/test`).

Deep-link to this section with `#provider-sam_gov`. ↑ Back to top

dollar-sign USAspending OPTIONAL `usaspending`

Federal spending data

1. What it powers

- Federal spending data (USAspending).

2. How to get the key

1. The USAspending data API (api.usaspending.gov) is **public and does not require a key** for normal use.
2. If you want a registered key for higher rate limits, request one from the USAspending API documentation/registration page.
3. Paste the issued key if you were given one.

3. Where to paste it

Admin Panel → card 🔑 **Provider API Keys** → **USAspending** row → **“Paste API key”** → **Save**. Leave it empty if you are using the public API unauthenticated.

4. Required or optional — and what breaks without it

Optional / usually unnecessary — the public API works without a key, so this row can stay empty.

5. Cost & plan notes

Free / public data.

6. How to verify it

Click **Test** on the row (`GET /api/v1/admin/provider-keys/usaspending/test`) if a key was entered.

Deep-link to this section with `#provider-usaspending`. ↑ Back to top

⚠ Credentials that are *not* in this provider table

Some parts of the app read credentials that are **not** listed in the 24-row provider catalogue above, so they do not appear as rows in the Provider API Keys card. Don't hunt for them in the panel:

- **Yelp** — business data. Read from the provider-keys store / `YELP_API_KEY` environment variable by the Data Company module, but Yelp is not a row in `available_providers`.
- **AnswerThePublic** — keyword research (the blog Q&A / keyword fetcher). It reads a `provider_keys` row directly and is **not** listed in the availability catalogue, so it is not pasteable from this card. (Its cousin **DataForSEO** *is* now a row — see its section above.)
- **Gemini** — selectable inside the blog generator's model routing, but likewise not a row in this table.
- **SMTP** — host/port/username/password live in each directory's own email settings, not in the provider keys card.

If you need one of these, tell the developer — they are configured outside the VA-facing panel on purpose.

Section anchors are stable and safe to deep-link: `#provider-<provider_key>` for a provider, `#start-here`, `#quick-map`, `#all-providers`. This page is generated from the live `available_providers` table — if a provider is added there, it must be given a section here (the generator refuses to build otherwise).

← Back to all guides