

Actor idea: LeadGraph™ — Local Business Lead Engine (Scrape → Enrich → Score → Export)

What it does (in one run):

Takes an industry + location (or list) and returns a **deduped, enriched, CRM-ready lead list** with **contact data + intent signals + AI outreach snippets**.

This beats “single-site scrapers” because it’s an **end-to-end pipeline Actor**, not just extraction. The “Challenge picks” list is packed with great point-solutions (flights, menus, bot detection, domain inspector, etc.). Your Actor wins by being the *default engine* businesses run weekly. [Apify](#)

The killer differentiators (why it wins in the Store)

1) Multi-source collection (with dedupe + confidence)

- Sources (user can toggle): Google Maps, Yelp, BBB, Chamber sites, niche directories, SERP “near me” results, and the business website itself (contact/about pages)
- **Entity resolution:** merge duplicates across sources (name/address/phone/domain fuzzy match)
- Output includes a **confidence score** + which sources confirmed each field

2) Website contact enrichment that actually works

For each business domain:

- Crawl “Contact”, “About”, “Team”, footer, schema.org (LocalBusiness), and social links
- Extract:
 - emails (mailto + text), phones, contact form URL
 - owner/operator names if present
 - services list / specialties
 - “book online” links (Calendly/HousecallPro/Jobber/etc.)
- Optional: email validation (pluggable) + role-email detection

3) Lead scoring for sales teams (not just data)

Create a **LeadScore (0–100)** using signals like:

- Review count + rating
- Recency of reviews
- Website quality (https, speed proxy check, broken contact page)
- Ad-tech signals (Meta Pixel / Google Ads tags)
- Hiring signals (if you include an optional “Careers” check)
- Category competition intensity (how many competitors in radius)

4) “Ready-to-send” AI outreach (huge adoption boost)

For each lead, generate:

- 1 cold email draft
- 1 voicemail script
- 1 SMS opener (optional)
- Personalization hooks (“Noticed you offer X in Y...”)
This is exactly the kind of “AI workflow” Apify is spotlighting in the challenge.
[Apify+1](#)

5) Exports + integrations that reduce churn

- Outputs: JSON / CSV / Excel (standard)
 - Webhook push to: Zapier/Make, Google Sheets, HubSpot, Zoho, Salesforce (via webhook or API module)
 - “Delta mode”: only export **new/changed** leads since last run (retention gold)
-

Inputs (Apify UI-friendly)

Required

- `seedType: keyword | category | customUrls`
- `keywords: e.g., “tree service”, “laundromat”, “HVAC”`
- `locations: city/state or lat/long + radius`

Power filters

- minimum rating, minimum reviews
- “has website only”
- “exclude franchises/aggregators”
- include/exclude categories
- max results per location
- schedule-friendly mode (slow/stealth)

AI options

- tone (Direct, Friendly, Professional)
 - target offer (user enters 1–2 sentences)
 - personalization depth (low/med/high)
-

Outputs (what users get)

Each record:

- Business: name, category, address, phone
 - Online: website, socials, Google Maps/Yelp/BBB URLs
 - Contacts: emails (validated flag), contact form, key people (if found)
 - Signals: rating/reviews, last review date, claimed listing (where detectable), tech tags
 - Scores: LeadScore + reason codes
 - AI: cold email + voicemail + opener
 - Metadata: sources seen, confidence score, timestamp
-

How to build it (stack that wins on Apify)

- **Crawlee + Playwright** for dynamic sources, plus HTML parsing for websites
- **Apify Proxy + SessionPool** for reliability and anti-blocking (core Store expectation)
[Apify Blog](#)
- Concurrency controls per source (Maps/Yelp stricter than websites)
- Caching + request fingerprinting to reduce cost
- Modular “source adapters” so you can add directories fast (and market updates weekly)

Suggested repo layout:

- /src/sources/googleMaps.ts
 - /src/sources/yelp.ts
 - /src/sources/bbb.ts
 - /src/enrich/websiteCrawler.ts
 - /src/enrich/dedupe.ts
 - /src/score/leadScore.ts
 - /src/ai/generateOutreach.ts
 - /src/output/export.ts
-

Pricing that sells (simple, high margin)

- **Freemium:** 25 leads/run, no AI, no delta mode
- **Starter:** \$9–\$19 for 500 leads/mo
- **Pro:** \$49–\$99 for 10k leads/mo + AI outreach + delta mode
- **Agency:** \$199+ multi-workspace + webhook bundles

This matches how Apify Store is positioned: fast setup, structured outputs, and automation-ready. [Apify Blog+1](#)

Store listing copy (you can paste into Apify)

Title: LeadGraph™ — Local Business Lead Engine (Scrape + Enrich + Score)

One-liner: Find local businesses by industry and location, enrich contacts from their websites, score leads, and export CRM-ready lists—plus AI outreach drafts.

Use cases:

- agencies doing outbound
 - local services lead gen
 - market expansion lists
 - competitive intelligence
-

If you want, I can also write:

- the **exact Apify input schema JSON** (with UI labels/tooltips),
- the **output dataset schema**,
- and a “**Getting started**” **README** that’s optimized for Apify Store conversion.

all

Absolutely — here are **all 3**: the **Apify input schema JSON**, the **output dataset schema**, and a **Store-optimized README** you can paste straight into your Actor repo.

1) `INPUT_SCHEMA.json` (Apify UI schema)

Save as `INPUT_SCHEMA.json` in your Actor root (or wherever you manage schema). This is designed for a clean UI with sensible defaults.

```
{  
  "title": "LeadGraph™ – Local Business Lead Engine",  
  "description": "Scrape + Enrich + Score local business leads by industry & location. Export deduped, CRM-ready leads with optional AI outreach drafts.",  
  "type": "object",  
  "schemaVersion": 1,  
  "properties": {  
    "seedType": {  
      "title": "Seed Type",  
      "type": "string",  
      "description": "Choose how to seed the search.",  
      "default": "keyword",  
      "editor": "select",  
      "enum": ["keyword", "category", "customUrls"],  
      "enumTitles": ["Keyword search", "Category search", "Custom URLs list"]  
    },  
    "keywords": {  
      "title": "Keywords",  
      "type": "array",  
      "items": "string"  
    }  
  }  
}
```

```
        "description": "Business keywords to search (e.g., 'tree service', 'laundromat', 'HVAC').",
        "editor": "stringList",
        "default": ["tree service"]
    },
    "categories": {
        "title": "Categories",
        "type": "array",
        "description": "Optional category list (used when Seed Type = Category search).",
        "editor": "stringList",
        "default": []
    },
    "customUrls": {
        "title": "Custom URLs",
        "type": "array",
        "description": "Provide business listing URLs (e.g., directory pages or Maps share links). Used when Seed Type = Custom URLs list.",
        "editor": "stringList",
        "default": []
    },
    "locations": {
        "title": "Locations",
        "type": "array",
        "description": "Locations to search. Use 'City, State' or 'ZIP' or 'Town, State'.",
        "editor": "stringList",
        "default": ["Pepperell, MA"]
    },
    "geoMode": {
        "title": "Geo Mode",
        "type": "string",
        "description": "Use city/town text search or precise lat/lng radius.",
        "default": "text",
        "editor": "select",
        "enum": ["text", "radius"],
        "enumTitles": ["Text locations", "Lat/Lng + radius"]
    },
    "radiusMiles": {
        "title": "Radius (miles)",
        "type": "number",
        "description": "Only used when Geo Mode = Lat/Lng + radius.",
        "default": 10,
        "minimum": 1,
        "maximum": 200
    },
    "centerLat": {
        "title": "Center Latitude",
        "type": "number",
        "description": "Only used when Geo Mode = Lat/Lng + radius.",
        "default": 42.665
    },
    "centerLng": {
```

```

    "title": "Center Longitude",
    "type": "number",
    "description": "Only used when Geo Mode = Lat/Lng + radius.",
    "default": -71.588
  },
  "sources": {
    "title": "Sources",
    "type": "array",
    "description": "Select which sources to use for discovery.",
    "editor": "checkboxGroup",
    "default": ["googleMaps", "yelp", "bbb", "serp"],
    "items": {
      "type": "string",
      "enum": ["googleMaps", "yelp", "bbb", "chambers", "serp",
      "nicheDirectories"],
      "enumTitles": [
        "Google Maps",
        "Yelp",
        "BBB",
        "Chambers of Commerce",
        "Search (SERP)",
        "Niche directories"
      ]
    }
  },
  "maxResultsPerLocation": {
    "title": "Max Results per Location",
    "type": "integer",
    "description": "Maximum businesses to collect per location (before dedupe).",
    "default": 100,
    "minimum": 1,
    "maximum": 5000
  },
  "maxTotalResults": {
    "title": "Max Total Results",
    "type": "integer",
    "description": "Hard cap across all locations (before dedupe).",
    "default": 500,
    "minimum": 1,
    "maximum": 50000
  },
  "filters": {
    "title": "Filters",
    "type": "object",
    "description": "Filtering and qualification rules.",
    "properties": {
      "minRating": {
        "title": "Min Rating",
        "type": "number",
        "description": "Only keep businesses with rating >= this value (if available).",
        "default": 0,
        "minimum": 0,
        "maximum": 5
      },
      "minReviews": {

```

```

        "title": "Min Reviews",
        "type": "integer",
        "description": "Only keep businesses with reviewCount >= this
value (if available).",
        "default": 0,
        "minimum": 0,
        "maximum": 100000
    },
    "requireWebsite": {
        "title": "Require Website",
        "type": "boolean",
        "description": "Drop leads without a website.",
        "default": false
    },
    "excludeFranchises": {
        "title": "Exclude Franchises / Chains",
        "type": "boolean",
        "description": "Best-effort filtering for chain/franchise
brands.",
        "default": false
    },
    "excludeKeywords": {
        "title": "Exclude Keywords",
        "type": "array",
        "description": "Drop leads whose name/description contains these
terms (case-insensitive).",
        "editor": "stringList",
        "default": []
    },
    "includeKeywords": {
        "title": "Include Keywords",
        "type": "array",
        "description": "Only keep leads that contain at least one of
these terms in name/description.",
        "editor": "stringList",
        "default": []
    }
}
},
{
    "enrichment": {
        "title": "Enrichment",
        "type": "object",
        "description": "Website crawl + contact extraction options.",
        "properties": {
            "crawlWebsite": {
                "title": "Crawl Website for Contacts",
                "type": "boolean",
                "description": "Visit business websites to extract
emails/phones/contact forms/social links.",
                "default": true
            },
            "maxWebsitePages": {
                "title": "Max Website Pages",
                "type": "integer",
                "description": "Limit pages per domain (contact/about/team/footer
prioritized).",
                "default": 10,
                "minimum": 1,
                "maximum": 100
            }
        }
    }
}

```

```

        "emailExtraction": {
            "title": "Email Extraction",
            "type": "string",
            "description": "How aggressively to look for emails.",
            "default": "standard",
            "editor": "select",
            "enum": ["off", "standard", "aggressive"],
            "enumTitles": ["Off", "Standard", "Aggressive"]
        },
        "phoneExtraction": {
            "title": "Phone Extraction",
            "type": "boolean",
            "description": "Extract phones from website and structured
data.",
            "default": true
        },
        "collectTechSignals": {
            "title": "Collect Tech Signals",
            "type": "boolean",
            "description": "Detect common tracking/marketing tech (best-
effort).",
            "default": true
        }
    }
),

"dedupe": {
    "title": "Dedupe & Matching",
    "type": "object",
    "description": "Controls how duplicate businesses are merged.",
    "properties": {
        "enabled": {
            "title": "Enable Dedupe",
            "type": "boolean",
            "default": true
        },
        "strategy": {
            "title": "Dedupe Strategy",
            "type": "string",
            "default": "balanced",
            "editor": "select",
            "enum": ["strict", "balanced", "aggressive"],
            "enumTitles": ["Strict (few merges)", "Balanced (recommended)",
"Aggressive (more merges)"]
        }
    }
},
"scoring": {
    "title": "Lead Scoring",
    "type": "object",
    "description": "Score leads for outreach prioritization.",
    "properties": {
        "enabled": {
            "title": "Enable Scoring",
            "type": "boolean",
            "default": true
        },
        "weightsPreset": {
            "title": "Weights Preset",
            "type": "string",

```

```

        "default": "localService",
        "editor": "select",
        "enum": ["localService", "b2bAgency", "retail", "custom"],
        "enumTitles": ["Local service", "B2B agency", "Retail", "Custom
(advanced)"]
    },
    "customWeights": {
        "title": "Custom Weights (advanced)",
        "type": "object",
        "description": "Only used when Weights Preset = Custom.",
        "default": {},
        "properties": {
            "reviews": { "type": "number", "default": 1 },
            "recency": { "type": "number", "default": 1 },
            "hasWebsite": { "type": "number", "default": 1 },
            "techSignals": { "type": "number", "default": 1 },
            "contactability": { "type": "number", "default": 1 }
        }
    }
},
{
},
"aiOutreach": {
    "title": "AI Outreach Drafts",
    "type": "object",
    "description": "Generate email/voicemail/SMS drafts per lead
(optional).",
    "properties": {
        "enabled": {
            "title": "Enable AI Drafts",
            "type": "boolean",
            "default": false
        },
        "tone": {
            "title": "Tone",
            "type": "string",
            "default": "direct",
            "editor": "select",
            "enum": ["direct", "friendly", "professional", "highEnergy"],
            "enumTitles": ["Direct", "Friendly", "Professional", "High-
energy"]
        },
        "offer": {
            "title": "Your Offer (1-2 sentences)",
            "type": "string",
            "description": "Describe what you're offering so the drafts match
your pitch.",
            "default": "We help local businesses get more customers with
simple marketing + follow-up automation."
        },
        "personalizationDepth": {
            "title": "Personalization Depth",
            "type": "string",
            "default": "medium",
            "editor": "select",
            "enum": ["low", "medium", "high"],
            "enumTitles": ["Low (fast)", "Medium", "High (slower)"]
        },
        "includeSmsOpener": {
            "title": "Include SMS opener",
            "type": "boolean",

```

```

        "default": false
    }
}
},
"exports": {
    "title": "Exports & Integrations",
    "type": "object",
    "description": "Control output formats and automation hooks.",
    "properties": {
        "outputFormat": {
            "title": "Output Format",
            "type": "string",
            "default": "dataset",
            "editor": "select",
            "enum": ["dataset", "csv", "json", "xlsx"],
            "enumTitles": ["Apify Dataset", "CSV", "JSON", "Excel (XLSX)"]
        },
        "deltaMode": {
            "title": "Delta Mode",
            "type": "boolean",
            "description": "Only output new/changed leads compared to previous run (uses stable dedupeId).",
            "default": false
        },
        "webhookUrl": {
            "title": "Webhook URL (optional)",
            "type": "string",
            "description": "POST results or summary to a webhook (Zapier/Make/CRM endpoint).",
            "default": ""
        },
        "webhookMode": {
            "title": "Webhook Mode",
            "type": "string",
            "default": "summary",
            "editor": "select",
            "enum": ["off", "summary", "perLead", "both"],
            "enumTitles": ["Off", "Summary only", "Per-lead", "Both"]
        }
    }
},
"runMode": {
    "title": "Run Mode",
    "type": "object",
    "description": "Speed vs stealth controls.",
    "properties": {
        "stealth": {
            "title": "Stealth Mode",
            "type": "boolean",
            "description": "Lower concurrency + more human-like browsing (slower, more reliable).",
            "default": true
        },
        "maxConcurrency": {
            "title": "Max Concurrency",
            "type": "integer",
            "default": 10,
            "minimum": 1,
            "maximum": 200
        }
    }
}
}
```

```

        },
        "maxRetries": {
            "title": "Max Retries",
            "type": "integer",
            "default": 3,
            "minimum": 0,
            "maximum": 20
        },
        "proxyEnabled": {
            "title": "Use Apify Proxy",
            "type": "boolean",
            "default": true
        }
    },
    "debug": {
        "title": "Debug",
        "type": "object",
        "description": "Troubleshooting settings.",
        "properties": {
            "saveHtmlSnapshots": {
                "title": "Save HTML Snapshots",
                "type": "boolean",
                "default": false
            },
            "logLevel": {
                "title": "Log Level",
                "type": "string",
                "default": "INFO",
                "editor": "select",
                "enum": ["ERROR", "WARN", "INFO", "DEBUG"],
                "enumTitles": ["ERROR", "WARN", "INFO", "DEBUG"]
            }
        }
    }
},
"required": ["seedType", "locations"]
}

```

2) Output dataset schema (lead record)

This is the “contract” your Actor should output for each lead (Dataset item). Use it as your TypeScript interface too.

```
{
    "dedupeId": "string",
    "confidence": 0.0,
    "sources": {
        "googleMaps": { "url": "string", "placeId": "string" },
        "yelp": { "url": "string", "bizId": "string" },
        "bbb": { "url": "string", "businessId": "string" },
        "serp": { "url": "string" }
    },
    "business": {
        "name": "string",
        "category": "string",

```

```
"categories": ["string"],
"description": "string",
"address": {
    "street": "string",
    "city": "string",
    "state": "string",
    "postalCode": "string",
    "country": "string",
    "formatted": "string"
},
"geo": { "lat": 0.0, "lng": 0.0 },
"phone": "string",
"phoneE164": "string"
},
"online": {
    "website": "string",
    "domain": "string",
    "socials": {
        "facebook": "string",
        "instagram": "string",
        "linkedin": "string",
        "youtube": "string",
        "tiktok": "string",
        "x": "string"
    }
},
"contacts": {
    "emails": [
        {
            "email": "string",
            "source": "string",
            "confidence": 0.0,
            "isRoleBased": true,
            "isValidated": false
        }
    ],
    "phones": [
        {
            "phone": "string",
            "phoneE164": "string",
            "source": "string",
            "confidence": 0.0
        }
    ],
    "contactFormUrl": "string",
    "bookingUrl": "string",
    "keyPeople": [
        {
            "name": "string",
            "title": "string",
            "source": "string",
            "confidence": 0.0
        }
    ]
},
"signals": {
    "reviews": {
        "rating": 0.0,
```

```

    "reviewCount": 0,
    "lastReviewDate": "string"
},
"hours": {
    "monday": "string",
    "tuesday": "string",
    "wednesday": "string",
    "thursday": "string",
    "friday": "string",
    "saturday": "string",
    "sunday": "string"
},
"websiteSignals": {
    "hasHttps": true,
    "pagesCrawled": 0,
    "contactPageFound": true
},
"techSignals": {
    "metaPixel": false,
    "googleAnalytics": false,
    "googleTagManager": false,
    "googleAdsRemarketing": false
}
},
"score": {
    "leadScore": 0,
    "tier": "string",
    "reasons": ["string"]
},
"ai": {
    "coldEmail": {
        "subject": "string",
        "body": "string"
    },
    "voicemail": "string",
    "smsOpener": "string",
    "personalizationHooks": ["string"]
},
"raw": {
    "collectedAt": "string",
    "runId": "string",
    "notes": "string"
}
}
}

```

Notes that help developers:

- dedupeId: stable hash like `sha1(normalizedName + normalizedAddress + domain + phoneE164)`
 - confidence: 0–1 overall confidence for “this is a real business with correct contact info”
 - ai block can be omitted entirely when AI disabled
-

3) Store-optimized `README.md`

Copy/paste as `README.md`. This is written to convert Store traffic.

```
# LeadGraph™ – Local Business Lead Engine (Scrape → Enrich → Score → Export)

LeadGraph finds local businesses by industry & location, **deduplicates them**, crawls their websites for contact info, scores leads for outreach priority, and outputs a **CRM-ready dataset** (CSV/JSON/XLSX optional). Optional: generate AI outreach drafts (cold email + voicemail + SMS opener).

## What you get
For each business, LeadGraph can return:
- Name, category, address, phone
- Website + social links
- Emails (best-effort extraction) + contact form + booking links
- Reviews (rating, count, recency where available)
- Tech signals (best-effort: GA/GTM/Meta Pixel)
- LeadScore (0-100) + reasons
- Optional AI outreach drafts

## Use cases
- Agencies building weekly lead lists
- Local service providers doing targeted outbound
- Market expansion research
- Competitive landscape snapshots

---

## Quick start (recommended)
1. Set **Seed Type = Keyword search**
2. Enter:
   - Keywords: `["tree service"]`
   - Locations: `["Pepperell, MA", "Townsend, MA"]`
3. Keep **Crawl Website for Contacts = ON**
4. Run the Actor → export results as CSV or push to your webhook.

---

## Input
Key fields:
- **keywords**: search terms (e.g. "laundromat", "HVAC", "landscaping")
- **locations**: "City, State" or ZIP
- **sources**: Google Maps / Yelp / BBB / SERP / etc.
- **filters**: min rating/reviews, require website, exclude terms
- **enrichment**: website crawl + extraction settings
- **dedupe**: strict/balanced/aggressive merging
- **scoring**: enable + weights preset
- **exports**: Dataset / CSV / JSON / XLSX + webhook + delta mode

---

## Output
Results are stored in the Actor **default dataset** as one item per lead.

### Dedupe
LeadGraph merges duplicates across sources and outputs:
- `dedupeId`: stable identifier for the business
```

- `confidence`: how reliable the record is (0-1)
- `sources`: which sources confirmed the business and URLs

Delta mode (low-churn exports)
If enabled, LeadGraph will only output **new or changed** leads compared to the previous run (uses `dedupeId`).

Webhook integration
If you provide **Webhook URL**, LeadGraph can POST:

- A summary payload (counts, top scores, errors)
- Per-lead payloads (one JSON per lead)
- Or both

Great for: Zapier, Make, custom CRMs, Google Sheets pipelines.

Tips for best results

- Use **Stealth Mode** when scraping stricter sources
- Use **Balanced** dedupe for most scenarios
- Keep **Max Website Pages** around 10-15 for fast enrichment
- Enable **Require Website** if you only want leads you can contact online

Limits & notes

- Some sources may rate-limit or change page structure; use proxy + stealth for reliability.
- Email extraction is best-effort; validation is optional (bring your own validator API if needed).
- Tech signal detection is best-effort and may not detect all tags.

Changelog

- v1.0.0: multi-source discovery + website enrichment + dedupe + scoring + exports

If you want to ship this *as a top-tier Store listing*, the next “all-in” piece is the **pricing + run-unit economics + example datasets** (3 example runs: “HVAC Boston”, “Laundromats Worcester County”, “Tree Service Nashua”) so buyers can instantly see output quality.

yes

Perfect — here’s the **final “ship-ready” package** that Apify reviewers + buyers love:

LeadGraph™

Pricing, Run-Unit Economics & Example Datasets

This section is what **separates top-earning Actors** from hobby scrapers. You can paste most of this directly into your Store listing and GitHub README.

1 Store Pricing Strategy (Battle-Tested)

Free (Discovery)

Goal: Adoption + trust

- 25 leads / run
- 1 location
- Google Maps + SERP only
- No website crawl
- No AI outreach
- No delta mode

 Perfect for first-time Apify users

Starter — \$19/month

For solopreneurs & local service owners

- 500 leads / month
- Up to 5 locations
- Google Maps + Yelp + SERP
- Website crawl (contact/about only)
- CSV / JSON export
- Basic lead scoring
- Stealth mode ON

Typical buyer: local marketing agency, cleaner, landscaper

Pro — \$79/month (sweet spot)

For agencies & sales teams

- 10,000 leads / month
- Unlimited locations
- All sources (Maps, Yelp, BBB, Chambers)
- Full website enrichment
- AI outreach drafts (email + voicemail)
- Delta mode (new leads only)

- Webhook integrations
- XLSX export

This is where most revenue lands.

Agency — \$199/month+

For scale & automation

- 50,000+ leads / month
- Priority concurrency
- Per-lead webhooks
- Multi-workspace support
- Custom scoring presets
- White-label outputs (optional)

2 Run-Unit Economics (Why This Is Profitable)

This matters internally **and** impresses Apify staff.

Average cost per 1,000 leads

Component	Approx Cost
Apify compute	\$0.40 – \$0.80
Proxy usage	\$0.30 – \$0.60
AI generation (optional)	\$0.20 – \$0.50
Total cost / 1k leads	\$0.90 – \$1.90

Example margin (Pro plan)

- User pays: **\$79**
- Uses: 10,000 leads
- Your cost: ~\$15
- **Gross margin:** ~80%

This is **Apify gold**.

3 Example Dataset #1

HVAC Companies — Boston, MA

Input

```
{  
  "seedType": "keyword",  
  "keywords": ["HVAC contractor"],  
  "locations": ["Boston, MA"],  
  "maxTotalResults": 50,  
  "enrichment": { "crawlWebsite": true },  
  "scoring": { "enabled": true }  
}
```

Sample Output (1 of 50)

```
{  
  "dedupeId": "b9f3a8c1",  
  "confidence": 0.93,  
  "business": {  
    "name": "Boston Climate Control",  
    "category": "HVAC contractor",  
    "address": {  
      "city": "Boston",  
      "state": "MA",  
      "formatted": "Boston, MA"  
    },  
    "phone": "(617) 555-0182"  
  },  
  "online": {  
    "website": "https://bostonclimatecontrol.com",  
    "domain": "bostonclimatecontrol.com"  
  },  
  "contacts": {  
    "emails": [  
      {  
        "email": "service@bostonclimatecontrol.com",  
        "confidence": 0.82,  
        "isRoleBased": true  
      }  
    ],  
    "contactFormUrl": "https://bostonclimatecontrol.com/contact"  
  },  
  "signals": {  
    "reviews": { "rating": 4.7, "reviewCount": 128 },  
    "websiteSignals": { "hasHttps": true }  
  },  
  "score": {  
    "leadScore": 88,  
    "tier": "A",  
    "reasons": [  
      "High review volume",  
      "Website with contact form",  
      "Active marketing signals detected"  
    ]  
  }  
}
```



Example Dataset #2

Laundromats — Worcester County, MA

Why this example converts:

Niche + underserved = 💰

Sample Output

```
{  
  "business": {  
    "name": "CleanSpin Laundromat",  
    "category": "Laundromat",  
    "phone": "(508) 555-0144"  
  },  
  "contacts": {  
    "emails": [],  
    "contactFormUrl": "https://cleanspinlaundromat.com/contact"  
  },  
  "signals": {  
    "reviews": { "rating": 4.2, "reviewCount": 54 }  
  },  
  "score": {  
    "leadScore": 71,  
    "tier": "B",  
    "reasons": [  
      "Moderate reviews",  
      "Website present",  
      "No booking system detected"  
    ]  
  }  
}
```

👉 Great for upselling AI outreach + enrichment

5 Example Dataset #3

Tree Services — Nashua, NH

Why this example converts:

High-ticket services + owner-operator = fast close

Sample Output

```
{  
  "business": {  
    "name": "Granite State Tree Care",  
    "category": "Tree service",  
    "phone": "(603) 555-0199"  
  },  
  "contacts": {  
    "emails": [  
      {  
        "email": "info@granitestatetreecare.com",  
        "confidence": 0.91  
      }  
    ]  
  }  
}
```

```

        }
    ],
    "bookingUrl": "https://granitestatetreecare.com/request-estimate"
},
"ai": {
    "coldEmail": {
        "subject": "Quick question about tree jobs in Nashua",
        "body": "Noticed you're getting great reviews for tree work in
Nashua..."
    },
    "voicemail": "Hi, this is Erica – quick question about helping tree
companies book more estimates..."
},
"score": {
    "leadScore": 92,
    "tier": "A"
}
}

```

6 Why This Actor Wins the Challenge Picks

Feature	Typical Actor LeadGraph™
Single-site scraping	✓ ✗
Multi-source merge	✗ ✓
Website enrichment	✗ ✓
Lead scoring	✗ ✓
AI outreach	✗ ✓
Delta mode	✗ ✓
CRM-ready output	⚠ ✓
Weekly re-run value	✗ ✓

This isn't a scraper.
It's a **revenue engine**.

7 Next (optional but 🔥)

If you want, I can:

1. Write the **Apify Store marketing description** (headline + hero bullets)
2. Generate **3 downloadable example CSVs** you can attach
3. Design the **Actor icon + banner**
4. Map this directly into your **TradeHive / AdSuite CRM**

Just say “**next**” and tell me which one.

If you built it already then map it directly for TradeHive / AdSuite CRM, if not build this Actor first and then map it

I haven't built the Actor in your Apify account (I can't deploy code into Apify from here), but I *can* do the next best thing right now:

1. give you the **build spec that a dev can implement exactly**, and
2. provide the **direct mapping + webhook payloads** so LeadGraph™ pipes cleanly into **TradeHive CRM / AdSuite CRM** the moment it runs.

Below is the full "build + map" package.

A) Build LeadGraph™ Actor (implementation-ready spec)

Actor outputs (2 datasets + 1 key-value file)

Dataset: LEADS (1 item = 1 deduped business lead)

Dataset: CHANGES (optional delta-mode: new/changed only)

Key-Value Store file: RUN_SUMMARY.json (counts, errors, timings, source coverage)

Stability keys (critical for delta + CRM sync)

- `dedupeId` (stable hash):
`sha1(normalize(name) + normalize(address.formatted) + domain + phoneE164)`
- `sourceIds` (if available): `placeId / yelpBizId / bbbId`

“CRM-first” guarantee

Every record must include these minimum fields even if blank:

- `business.name`
 - `business.address.formatted`
 - `business.phone` OR `online.website` OR `contacts.contactFormUrl`
 - `online.domain` (derived when website present)
 - `raw.collectedAt, raw.runId`
-

B) Direct mapping to TradeHive / AdSuite CRM

Because both are *your* CRMs, the cleanest integration is **one webhook endpoint per CRM** that accepts:

- **Run summary** (once)
- **Per-lead upsert** (per lead) or **batch upsert** (recommended)

Recommended flow

1. LeadGraph Actor runs
2. Actor posts **batch payloads** of 25–200 leads to:
 - o POST `https://<your-crm-domain>/api/v1/integrations/apify/leadgraph/upsert`
3. CRM replies with:
 - o `created, updated, skipped, errors[]`

Why batch > per-lead

- faster
- fewer timeouts
- easier retry logic
- better rate-limit control

B1) TradeHive CRM field mapping (Lead → Company/Account + Contact)

Upsert rule:

- Primary key: `dedupeId`
- Secondary match: `phoneE164` or `domain` (if `dedupeId` missing)

TradeHive “Company/Account” object

LeadGraph field	TradeHive field (suggested)
<code>business.name</code>	<code>AccountName</code>
<code>business.category</code>	<code>IndustryCategory</code>
<code>business.address.formatted</code>	<code>BillingAddress</code>
<code>business.address.city</code>	<code>City</code>
<code>business.address.state</code>	<code>State</code>
<code>business.address.postalCode</code>	<code>Zip</code>
<code>business.phoneE164 / business.phone</code>	<code>MainPhone</code>
<code>online.website</code>	<code>Website</code>
<code>online.domain</code>	<code>Domain</code>
<code>sources.googleMaps.url</code>	<code>GoogleListingUrl</code>
<code>signals.reviews.rating</code>	<code>ReviewRating</code>
<code>signals.reviews.reviewCount</code>	<code>ReviewCount</code>

LeadGraph field	TradeHive field (suggested)
signals.reviews.lastReviewDate	LastReviewDate
score.leadScore	LeadScore
score.tier	LeadTier
score.reasons	LeadScoreReasons (text/JSON)
confidence	DataConfidence
raw.collectedAt	LeadCollectedAt
raw.runId	LeadSourceRunId
"LeadGraph"	LeadSource

TradeHive “Primary Contact” object (optional)

If you find a person name/title (`contacts.keyPeople[0]`):

LeadGraph field	TradeHive field
<code>contacts.keyPeople[0].name</code>	ContactName
<code>contacts.keyPeople[0].title</code>	ContactTitle
best email from <code>contacts.emails[]</code>	Email
best phone	Phone

If no person found, create a “General Contact”:

- `ContactName`: “Office / General”
- `Email`: best role email (info@, service@, etc.)

TradeHive “Next action” fields (for your callers)

LeadGraph field	TradeHive task field
<code>ai.coldEmail.subject/body</code>	SuggestedEmail
<code>ai.voicemail</code>	SuggestedVoicemail
<code>ai.smsOpener</code>	SuggestedSmsOpener
<code>contacts.contactFormUrl</code>	BestContactPath
<code>contacts.bookingUrl</code>	BookingLink

B2) AdSuite CRM field mapping (Lead → Agency Prospect + Sales Pipeline)

AdSuite is agency-centric, so map into a pipeline stage automatically.

Pipeline suggestion

- Stage 1: New Lead (LeadGraph)
- Stage 2: Qualified
- Stage 3: Contacted
- Stage 4: Booked

- Stage 5: Closed Won/Lost

AdSuite “Lead/Prospect” object

LeadGraph field	AdSuite field
business.name	ProspectName
business.category	Vertical
business.phoneE164	Phone
online.website	Website
online.domain	Domain
locations[] used in run	Territory
score.leadScore	PriorityScore
score.tier	PriorityTier
signals.websiteSignals.hasHttps	HasHttps
signals.techSignals	MarketingTechSignals (JSON/text)
sources.*.url	SourceLinks (JSON/text)
raw.collectedAt	ImportedAt
“LeadGraph”	LeadSource
set stage	PipelineStage = New Lead (LeadGraph)

Automation triggers (AdSuite)

When PipelineStage = New Lead (LeadGraph):

- Auto-assign to a caller (round robin)
- Create task: “Call within 24 hours”
- If AI drafts exist, attach to notes + email template suggestion

C) Webhook payloads you can implement immediately

C1) Batch upsert payload (recommended)

Actor sends this:

```
{
  "source": "apify-leadgraph",
  "runId": "apify_run_123",
  "collectedAt": "2025-12-18T14:22:11.000Z",
  "mode": "full",
  "crmTarget": "tradehive",
  "leads": [
    {
      "dedupeId": "b9f3a8c1...",
      "confidence": 0.93,
      "name": "John Doe",
      "email": "john.doe@example.com",
      "phone": "+1234567890",
      "vertical": "Technology"
    }
  ]
}
```

```

"business": {
    "name": "Example Tree Service",
    "category": "Tree service",
    "address": { "formatted": "Pepperell, MA 01463" },
    "phone": "(978) 555-0101",
    "phoneE164": "+19785550101"
},
"online": { "website": "https://example.com", "domain": "example.com" },
"contacts": {
    "emails": [{ "email": "info@example.com", "confidence": 0.8 }],
    "contactFormUrl": "https://example.com/contact"
},
"signals": { "reviews": { "rating": 4.7, "reviewCount": 128 } },
"score": { "leadScore": 88, "tier": "A", "reasons": ["High reviews", "Website present"] },
"ai": {
    "coldEmail": { "subject": "Quick question", "body": "..." },
    "voicemail": "..."
},
"sources": { "googleMaps": { "url": "..." } }
}
]
}

```

CRM responds:

```
{
  "runId": "apify_run_123",
  "created": 18,
  "updated": 62,
  "skipped": 5,
  "errors": [
    { "dedupeId": "xxx", "message": "Missing ProspectName" }
  ]
}
```

C2) Delta mode payload (new/changed only)

Same payload, just "mode": "delta".

D) What your dev needs to do inside TradeHive/AdSuite (small list)

1. **Create fields** (or match existing ones) listed in the mapping tables
2. Add **unique index** on dedupeId (or enforce uniqueness in upsert logic)
3. Build 1 endpoint per CRM:
 - o /integrations/apify/leadgraph/upsert
4. Add optional endpoint for run summaries:
 - o /integrations/apify/leadgraph/summary
5. Turn on automations:
 - o create tasks, assign owners, set pipeline stage

E) If you want “plug-and-play” for your team tomorrow

Tell your dev to implement **exactly two things** first:

1. `dedupeId` upsert + field mapping into the CRM lead object
2. auto-create a **call task** + set **PipelineStage = New Lead (LeadGraph)**

That alone makes it immediately usable for your callers.

If you paste your **TradeHive** and **AdSuite** object names (what you call Leads/Accounts/Contacts internally) and whether you already have an **API key auth pattern**, I'll rewrite the webhook + mapping in your exact naming so your dev can copy/paste it with almost no interpretation.

Give me what I need to give to my developer for all that I need to build and integrate into the CRMs

Below is a **developer handoff pack** you can copy/paste into a ticket or Google Doc. It includes **everything needed** to build the LeadGraph™ Apify Actor **and** integrate it into **TradeHive CRM + AdSuite CRM** with clean upserts, dedupe, delta mode, and caller-ready tasks.

LeadGraph™ (Apify Actor) + TradeHive/AdSuite Integration

Developer Handoff Spec (Build + Integrate)

0) Goal

Build an Apify Actor named **LeadGraph™** that:

1. Collects local business leads from selected sources
2. Dedupes and merges records into one “best” lead per business
3. Crawls the business website to extract contact info + signals
4. Scores leads (0–100) and optionally generates AI outreach drafts
5. Exports results and **pushes them into TradeHive CRM and/or AdSuite CRM via webhook upserts**

-
- 6. Supports **delta mode** (only new/changed leads since last run) so repeated runs don't spam CRMs

1) Deliverables (what you must build)

A) Apify Actor (LeadGraph)

- Tech: **Node.js + TypeScript + Crawlee + Playwright**
- Must include:
 - Multi-source discovery (Google Maps + Yelp + BBB + SERP at minimum)
 - Website enrichment crawler (contact/about/team/footer + schema)
 - Dedupe/merge engine
 - Lead scoring engine
 - Dataset outputs
 - Webhook push to CRM endpoints (batch preferred)
 - Delta mode support

B) CRM Integration (TradeHive + AdSuite)

- Add fields (or map to existing ones)
- Implement **2 API endpoints per CRM**:
 1. POST /api/v1/integrations/apify/leadgraph/upsert
 2. POST /api/v1/integrations/apify/leadgraph/summary (optional but recommended)
- Implement upsert logic keyed by `dedupeId`
- Create caller tasks + pipeline stage assignment automations

2) Apify Actor Input (UI schema already provided)

Use the `INPUT_SCHEMA.json` I supplied earlier. Actor should accept:

- `keywords[]`, `locations[]`, `sources[]`
- `filters` object
- `enrichment` object
- `dedupe settings`
- `scoring settings`
- `aiOutreach settings`
- `exports` including `webhookUrl`, `webhookMode`, `deltaMode`
- `runMode` settings (`proxy`, `stealth`, `concurrency`)

Required behavior:

- If `exports.webhookUrl` is set:
 - send summary payload once

- send leads in batches (25–200 per request) OR per-lead if selected
-

3) Actor Outputs (Data Contract)

Dataset `LEADS`

One item per **deduped** business lead, using this JSON schema (must match):

- `dedupeId` (string, stable)
- `confidence` (0–1)
- `sources` (links/ids)
- `business` (name, category, address, geo, phones)
- `online` (website, domain, socials)
- `contacts` (emails, phones, contact form, booking link, key people)
- `signals` (reviews, hours, website signals, tech signals)
- `score` (leadScore 0–100, tier A/B/C/D, reasons[])
- `ai` (optional) cold email subject/body + voicemail + smsOpener
- `raw` (collectedAt ISO, runId, notes)

Dataset `CHANGES` (only if `deltaMode = true`)

Only leads that are **new or changed** since previous run.

Key-Value Store `RUN_SUMMARY.json`

Counts + timings + errors + top stats:

- `totalFound`, `totalAfterDedupe`, `enrichedCount`, `aiCount`
 - created/updated/skipped (from CRM response, if pushed)
 - `errors[]`, `sourceCoverage{}, runTimeMs`
-

4) Critical: Stable ID + Delta Mode

`dedupeId` generation (must be stable)

Compute as:

- Normalize business name (lowercase, strip punctuation, collapse whitespace)
- Normalize address (formatted string if possible)
- Use domain (if present) and phoneE164 (if present)

Algorithm:

```
dedupeId = sha1(normName + "|" + normAddress + "|" + domain + "|" +  
phoneE164)
```

Delta mode behavior

When `exports.deltaMode = true`:

- Store prior run's lead hash in Key-Value Store:
 - `LEAD_STATE.json map { dedupeId: contentHash }`
- Compute `contentHash = sha1(JSON.stringify(coreFields))`
- Only output/push leads where:
 - `dedupeId` not in prior map OR `contentHash` changed

Core fields for hash (do not include transient timestamps):

- business name, address, phoneE164
- website/domain
- emails list
- score.leadScore/tier
- key signals (rating/reviewCount, techSignals booleans)

5) CRM Integration: Endpoints + Auth

Authentication (choose one and standardize)

- Preferred: `Authorization: Bearer <API_KEY>`
- Also acceptable: `x-api-key: <API_KEY>`

Apify Actor will send header:

- `Authorization: Bearer ${CRM_API_KEY}`

You will provide two keys:

- `TRADEHIVE_LEADGRAPH_API_KEY`
- `ADSUITE_LEADGRAPH_API_KEY`

6) CRM Upsert Endpoint Spec (both CRMs)

Endpoint

`POST /api/v1/integrations/apify/leadgraph/upsert`

Request payload (batch)

```
{  
  "source": "apify-leadgraph",  
  "crmTarget": "tradehive",  
  ...}
```

```

"runId": "apify_run_123",
"collectedAt": "2025-12-18T14:22:11.000Z",
"mode": "full",
"leads": [ /* lead schema item */ ]
}

```

Response payload

```

{
  "runId": "apify_run_123",
  "created": 0,
  "updated": 0,
  "skipped": 0,
  "errors": [
    { "dedupeId": "abc", "message": "Validation error..." }
  ]
}

```

Rules

- Must **upsert by dedupeId**
 - If dedupeId missing (shouldn't happen), fallback match by:
 - domain OR phoneE164
 - If lead already exists and unchanged, increment `skipped`
-

7) TradeHive CRM: Field Mapping (what to create / map)

TradeHive Object 1: Account/Company (primary)

Create/map these fields:

Identifiers

- `LeadSource` (string) = “LeadGraph”
- `LeadSourceRunId` (string)
- `DedupeId` (string, unique index)
- `DataConfidence` (number)

Business

- `AccountName` (string)
- `IndustryCategory` (string)
- `Website` (string)
- `Domain` (string)
- `MainPhone` (string)
- `BillingAddress` (string)
- `City` (string)
- `State` (string)
- `Zip` (string)

Source links

- GoogleListingUrl (string)
- YelpUrl (string)
- BBBUrl (string)

Signals

- ReviewRating (number)
- ReviewCount (number)
- LastReviewDate (date)

Score

- LeadScore (number)
- LeadTier (string)
- LeadScoreReasons (long text / JSON)

Contactability

- BestContactPath (string) (contact form url)
- BookingLink (string)

AI Suggestions (optional)

- SuggestedEmailSubject (string)
- SuggestedEmailBody (long text)
- SuggestedVoicemail (long text)
- SuggestedSmsOpener (string)

Meta

- LeadCollectedAt (datetime)

TradeHive Object 2: Contact (optional but recommended)

If CRM supports linked Contacts:

- Create/update Primary Contact for the account using best found email/phone
- If no person, create “Office / General”

Fields

- ContactName
- Email
- Phone
- Title
- ContactSource = “LeadGraph”

TradeHive Task Automation (required)

On create (new lead):

- Create Task:
 - Title: Call new LeadGraph lead
 - Due: within 24 hours
 - Notes: include best phone, best email, contact form url, AI scripts
 - Set Stage/Pipeline if applicable: New Lead (LeadGraph)
-

8) AdSuite CRM: Field Mapping + Pipeline

AdSuite Object: Prospect/Lead

Create/map fields:

Identifiers

- LeadSource = “LeadGraph”
- LeadSourceRunId
- DedupeId (unique)
- ImportedAt

Business

- ProspectName
- Vertical
- Phone
- Website
- Domain
- Territory (location used)

Score

- PriorityScore
- PriorityTier
- ScoreReasons (text/JSON)

Signals

- ReviewRating, ReviewCount
- HasHttps
- MarketingTechSignals (JSON/text)
- SourceLinks (JSON/text)

AI (optional)

- SuggestedEmailSubject, SuggestedEmailBody, SuggestedVoicemail, SuggestedSmsOpener

AdSuite Pipeline Behavior (required)

When new lead created:

- Set PipelineStage = New Lead (LeadGraph)
 - Assign owner via round-robin (or default owner)
 - Create Task: Call within 24 hours
-

9) Webhook Summary Endpoint (optional but recommended)

Endpoint

POST /api/v1/integrations/apify/leadgraph/summary

Payload

```
{  
  "source": "apify-leadgraph",  
  "crmTarget": "tradhive",  
  "runId": "apify_run_123",  
  "mode": "delta",  
  "stats": {  
    "totalFound": 420,  
    "totalAfterDedupe": 280,  
    "enrichedCount": 260,  
    "aiCount": 120  
  },  
  "errors": [],  
  "timingMs": {  
    "discovery": 120000,  
    "enrichment": 240000,  
    "pushToCrm": 45000  
  }  
}
```

Use this to show run history in CRM.

10) Error Handling + Retries (must have)

Actor:

- Retries failed pages up to runMode.maxRetries
- If CRM push fails:

- retry each batch with exponential backoff
- store failed batch payloads in Key-Value Store:
FAILED_BATCHES/<timestamp>.json

CRM endpoint:

- Validate payload sizes
 - Return useful error messages per lead
-

11) Acceptance Criteria (definition of done)

Actor

- Can run: keyword + location → produces LEADS dataset
- Dedupe merges duplicates correctly across sources
- Website crawl extracts at least one of: email/phone/contact form/socials
- LeadScore populated
- Delta mode works (2nd run produces minimal CHANGES dataset)
- Webhook pushes in batches successfully

CRM

- Upsert endpoint creates/updates correctly by dedupId
 - Required fields mapped into correct objects
 - Tasks created automatically on new leads
 - Pipeline stage set to New Lead (LeadGraph)
-

12) Configs I (Erica) will provide to you

- TradeHive API base URL + API key
 - AdSuite API base URL + API key
 - Which CRM objects/field names exist already (if different than above)
 - Default owner(s) + pipeline stage IDs
-

If you want, paste back **your actual existing field names** in TradeHive/AdSuite (even screenshots are fine) and I'll rewrite the mapping section to match your CRM exactly—so your dev doesn't have to interpret anything.