HURE Developer Handoff – Exact Airtable Field Wiring (Final)

This document is a step-by-step, implementation-ready build specification for connecting the existing Airtable setup to the ESRI county map on the WordPress/Elementor website, and wiring Stripe enrollment + $100/year maintenance into Airtable so the map and workflows update automatically. All field names below match the current Airtable fields provided in screenshots.

1) Current System Snapshot (What Exists Today)

• Website: WordPress + Elementor; map embedded on the site (Relocating experience is consumer-first).

• Map: ESRI county polygons + popup UI.

• Airtable: single primary table (Counties) currently contains both county data + brokerage coverage data.

• Stripe: products exist (Enrollment tiers + Annual Maintenance Fee $100/year).

2) Airtable: Table + Exact Field Names (Do Not Rename)

Primary table (existing): Counties (single-table approach currently in use).

Confirmed fields from your Airtable screenshots:

• FIPS — Text/Number — Primary key for county; must match polygons and all lookups.

• COUNTY — Text — County name for popup display.

• Partners... — Select/Link — Assigned partner/territory owner; used for routing consumer requests.

• STATE — Select — State abbreviation for display and filtering.

• centroid_lat — Number — Popup anchor/centroid latitude.

• centroid_lng — Number — Popup anchor/centroid longitude.

• status — Select — Public logic: Open vs Assigned.

• brokerage — Text — Brokerage name (assigned counties).

• brokerage_url — URL — Consumer click-through to brokerage website (required when assigned).

• phone — Text/Phone — Brokerage phone.

• area_blurb — Long text — Area info (shown for both Open and Assigned).

• highlights — Text/Long text — Area highlights bullets (shown for both Open and Assigned).

• photo_url — URL — Local attraction photo(s) OR brokerage photo (recommend: attraction).

• stripe_enrollment_link — URL — Enrollment checkout link (tiered).

• stripe_maintenance_... — URL — Annual maintenance link ($100/year).

• coverage_json — Long text — Optional: future multi-brokerage coverage payload (not required for MVP).

Required Airtable View for website/map reads (create if not present):
• View name: Map_Public
• Filter: status is Open OR Assigned
• Sort: optional
• military relocation Specialist T/F

3) Non-Negotiable Architecture (Fix Slow Map + Protect Keys)

Do NOT call Airtable directly from client-side JavaScript. Implement a server-side cached JSON endpoint.

Data flow:
Airtable (Map_Public view) → WordPress REST endpoint (cached) → Map JS fetches one JSON payload.

4) WordPress REST Endpoint (Exact Contract)

Create a custom endpoint:

GET /wp-json/hure/v1/counties

Endpoint behavior (step-by-step):
1) On first request (or cache miss), server fetches Airtable records from the Counties table using View=Map_Public.
2) Transform Airtable records into normalized JSON schema (below).
3) Cache the full JSON response for 10 minutes (WordPress transient or object cache).
4) Return cached JSON to the browser map.
5) Optional: provide an admin-only 'flush cache' action.

4.1 JSON Response Schema (Use Exactly)

Response example:


```
{
  "generated_at": "2025-12-15T19:00:00Z",
  "records": [
    {
      "fips": "2013",
      "county": "Aleutians East",
      "state": "AK",
      "centroid_lat": 55.37,
      "centroid_lng": -161.98,
      "status": "Open",
      "partners": "Unassigned",
      "area_blurb": "...",
      "highlights": "...",
      "photo_url": "https://...jpg",
      "brokerage": {
        "name": "",
        "phone": "",
        "brokerage_url": ""
      },
      "stripe_enrollment_link": "https://...",
      "stripe_maintenance_link": "https://..."
    }
  ]
}
```


4.2 Airtable Fetch Notes (Implementation)

Use Airtable REST API with Personal Access Token (PAT) stored server-side.
Do not embed PAT in Elementor HTML or map JS.
Fetch using:

- Base ID: <your base>
- Table: Counties
- View: Map_Public

5) Map Behavior + Popup Logic (Exact Requirements)

All popup logic is driven by Airtable field: status.

5.1 When status = Open

- Show COUNTY + STATE.
- Show area_blurb (always).

• Show highlights (always).

• Show photo_url (must be local attraction imagery).

• Show Consumer Request Form (Request a Local Brokerage / Local Expert).

• Do NOT show brokerage fields unless you want a placeholder (recommended: hide).

5.2 When status = Assigned

• Show COUNTY + STATE.

• Show area_blurb + highlights + photo_url (area content remains visible).

• Show brokerage block: brokerage (name), phone, brokerage_url (click-through), and optionally a brokerage photo if you later add a brokerage_photo_url field.

• Brokerage_url must open the brokerage's website directly.

• Optional: still show Consumer Request Form as secondary CTA (recommended).

There is no public 'Inactive' state. Public states: Open and Assigned only.

6) Consumer Request Form (Popup) – End-to-End

Requirement: Even Open counties must show area info and allow consumers to request a local brokerage.
On submit, the request must route to the assigned person for the county (Partner/territory owner).

6.1 Form Fields (Recommended)

• name

• email

• phone

• moving_from

• moving_to

• notes

• hidden: FIPS (county identifier)

6.2 Where the lead is stored

Create an Airtable table named Leads (recommended) OR a separate intake base. At minimum store: name, email, phone, requested FIPS, timestamp, assigned recipient.

6.3 Routing Logic (Exact)

Routing must use the existing field: Partners... (on the Counties row). If Partners... = Unassigned, route to Founder/Admin.

Implementation steps:
1) Form submits to a WordPress REST endpoint (POST /wp-json/hure/v1/lead).
2) Server looks up Counties record by FIPS to read Partners... and/or territory email (if you add one).
3) Server creates a Leads record in Airtable.
4) Server sends email notification to the assigned Partner + Admin.
5) Respond success to browser and show confirmation message.

7) For Brokers Page – Intake Form (Must Function)

Current issue: broker signup form does not route anywhere. Replace with working intake pipeline.
Implementation steps:
1) Elementor form posts to WP REST endpoint (POST /wp-json/hure/v1/broker-intake).
2) Server writes submission to Airtable table Brokerage_Intake (recommended).
3) Email admin/founder with submission details.
Required fields: brokerage name, phone, email, website, picture.
Questions: franchise affiliation, VA loans, military relocation specialist.
Note: Military specialist will later drive a badge/flag on the county map.

8) Stripe Integration – Enrollment + Annual Maintenance $100

Stripe products already exist in sandbox: Enrollment tiers + Annual Maintenance Fee $100/year.
Airtable contains two links per county:
• stripe_enrollment_link
• stripe_maintenance_...

Critical requirement: Stripe must update Airtable automatically via webhooks.

8.1 Enrollment Flow (One-Time)

Preferred implementation (fast): Use the existing stripe_enrollment_link in Airtable.
When brokerage enrolls, they pay via the link.
Webhook must update the matching Airtable county record to status=Assigned and populate brokerage fields.

8.2 Webhooks (Required)

• Create webhook endpoint (WP REST recommended): POST /wp-json/hure/v1/stripe-webhook

• Verify signature using Stripe webhook signing secret.

• Listen to: checkout.session.completed, invoice.payment_succeeded, invoice.payment_failed, customer.subscription.deleted

8.3 Metadata (Required for reliable mapping)

When generating or sending a user to Stripe, ensure the Checkout Session contains metadata:
• metadata.fips = <FIPS>
If you cannot control metadata using static links, the developer must implement dynamic Checkout Sessions.
Without metadata, webhooks cannot reliably update the correct county row.

8.4 Annual Maintenance – Delinquency Policy (Exact)

Goal: Notify immediately so the agent can reach out, keep county assigned for a grace period, then revert to Open.

Day 0 (invoice.payment_failed):
• Send email to assigned Partner (Partners...) + Admin.
• Mark internal delinquency (recommend adding fields annual_payment_status, past_due_date).
• Keep status as Assigned (consumer sees no disruption).

Days 1–30:
• Keep Assigned.
• Optional reminders day 7/14/21.

Day 31:
• If still unpaid, set status=Open (public).
• Clear brokerage fields from public map display if desired.

Implementation: daily cron job checks past_due_date <= today-30 and reverts.

9) Email Delivery + Logging (Recommendation)

Do not rely on default WordPress mail. Implement one of:
• SendGrid (recommended)
• Mailgun
• SMTP with authenticated domain

Enable logging so admin can confirm notifications were sent.

10) Test Checklist (Developer Must Execute)

- Map loads quickly; no persistent 'loading' state.
- Map fetches only /wp-json/hure/v1/counties (single request).
- Open county popup shows area_blurb/highlights/photo_url + consumer request form.
- Assigned county popup shows area_blurb/highlights/photo_url + brokerage + phone + brokerage_url link works.
- Consumer request submission creates lead record + routes email to correct Partner/Admin.
- Broker intake submission creates record + emails admin.
- Stripe enrollment webhook updates correct county row to status=Assigned.
- Stripe annual payment failure sends email and starts 30-day clock.
- Day 31 cron reverts county to Open if still unpaid.