

# PropData

## API Reference & Integration Guide

All endpoints. Real examples. For every team.

**166M+**

Parcels Indexed

**20+**

API Endpoints

**16**

Data Sources

**50**

States Covered

**<50ms**

Edge Latency

### Sections covered:

Authentication & Quick Start

Market & Rent Data · Property & Parcel

Intelligence & Analytics · Lead Generation

MCP / AI Connector · Utilities

Troubleshooting & Support

# Contents

---

## Authentication & Quick Start

### Market & Rent Data

- [GET /v1/market](#)
- [GET /v1/rent](#)
- [GET /v1/listing](#)
- [GET /v1/estimate](#)

### Property & Parcel

- [GET /v1/property](#)
- [GET /v1/property/by-location](#)
- [GET /v1/property/delta](#)
- [GET /v1/property/coverage](#)
- [GET /v1/parcel-geometry](#)

### Intelligence & Analytics

- [GET /v1/neighborhood](#)
- [GET /v1/zip-intel](#)
- [GET /v1/state-intel](#)
- [GET /v1/comps](#)
- [GET /v1/geocode](#)

### Lead Generation

- [GET /v1/skip-trace](#)
- [GET /v1/preforeclosure](#)
- [GET /v1/preforeclosure/delta](#)

### MCP / AI Connector

### Utilities

- [GET /v1/changelog](#)
- [GET /v1/stats](#)
- [GET /v1/usage](#)
- [GET /v1/health](#)

### Troubleshooting & Support

# Authentication & Quick Start

Every request to PropData requires your RapidAPI key passed in two headers. All endpoints use the base URL below. Paths always start with [/v1/](#) — never [/api/v1/](#).

## Base URL

```
https://propdata-real-estate-market-intelligence-api
.p.rapidapi.com
```

## Required Headers

| Header                 | Value                                                       |
|------------------------|-------------------------------------------------------------|
| <b>x-rapidapi-key</b>  | Your RapidAPI API key                                       |
| <b>x-rapidapi-host</b> | propdata-real-estate-market-intelligence-api.p.rapidapi.com |

## Your First API Call — Market data for any ZIP

```
curl --request GET \
  --url "https://propdata-real-estate-market-
  intelligence-api.p.rapidapi.com/v1/market?zip=55104" \
  --header "x-rapidapi-key: YOUR_KEY" \
  --header "x-rapidapi-host: propdata-real-estate-
  market-intelligence-api.p.rapidapi.com"
```

## JavaScript (Fetch)

```
const BASE = "https://propdata-real-estate-market-
intelligence-api.p.rapidapi.com";

const headers = {
  "x-rapidapi-key": process.env.RAPIDAPI_KEY,
  "x-rapidapi-host": "propdata-real-estate-market-
intelligence-api.p.rapidapi.com"
};

const res = await fetch(`${BASE}/v1/market?zip=55104`, {headers});
const data = await res.json();

console.log(data.snapshot.rent.median_asking_rent); // 1534
console.log(data.snapshot.signals);                 // signals[]
```

## Property Matching — Recommended Order

Use the most specific identifier available for the most accurate result:

1. parcel + county\_fips

2. lat + lng (/by-location)

3. address + city + state + zip

4. address + zip

5. zip or state
- exact parcel match

→ coordinate point-in-polygon

→ address match

→ address match

→ area-level search

Rate Limits & Usage

| Plan       | Monthly Requests | Key Features                                        |
|------------|------------------|-----------------------------------------------------|
| Starter    | 10,000           | All market, rent, property, intelligence endpoints  |
| Builder    | 50,000+          | + skip trace, delta feeds, higher-volume enrichment |
| Scale      | 250,000+         | + bulk workflows, priority support                  |
| Enterprise | Unlimited        | + direct Supabase, white-label, SLA, custom limits  |

Call [GET /v1/usage](#) at any time to check your current tier, request count, and reset window.

## Market & Rent Data

GET /v1/market · GET /v1/rent · GET /v1/listing · GET /v1/estimate

GET

### /v1/market

Full market intelligence for any ZIP, state, or metro

The core endpoint. One call returns rent data, listing market data, demand signals, affordability metrics, FHFA home price appreciation, FRED macro context, and pre-computed investor signals — all merged from 16 live sources into a single JSON response.

**Perfect for:**

- Rental pricing tools — feed median rent and HUD FMR into pricing logic
- Investment screeners — use gross yield, vacancy, and DOM signals
- Market report apps — 12-month history built in (extend with months=24)
- AI real estate agents — one call gives LLM full market context

**Parameters**

| Parameter | Type    | Required  | Description                        |
|-----------|---------|-----------|------------------------------------|
| zip       | string  | Optional* | 5-digit ZIP code — e.g. 55104      |
| state     | string  | Optional* | 2-letter state code — e.g. MN, TX  |
| metro     | string  | Optional* | Metro area name — e.g. Minneapolis |
| months    | integer | Optional  | History depth. Default 12, max 24  |

\* At least one location parameter is required.

**Example Request**

```
GET /v1/market?zip=55104
GET /v1/market?state=MN
GET /v1/market?metro=Minneapolis&months=24
```

**Example Response (abbreviated)**

```
{
  "location": { "zip": "55104", "state": "MN" },
  "snapshot": {
    "rent": {
      "median_asking_rent": 1534,
      "rent_yoy_pct": 3.6,
      "fmr_1br": 1275, "fmr_2br": 1510, "fmr_3br": 1980
    },
    "market": {
      "median_listing_price": 319950,
      "median_days_on_market": 33,
      "active_listings": 112, "price_per_sqft": 213
    },
    "signals": [
      { "key": "gross_yield", "value": "5.8%", "sentiment": "neutral" },
      { "key": "rent_growth", "value": "+3.6%", "sentiment": "neutral" },
      { "key": "vacancy", "value": "4.8%", "sentiment": "neutral" }
    ]
  },
  "macro": { "mortgage_rate_30yr": 6.82 },
  "sources": { "rent_index": "Zillow ZORI", "fmr": "HUD FMR" }
}
```

■ The *"signals"* array gives pre-computed investor context — gross yield, DOM category, vacancy label, rent growth, and sale-to-list ratio. Use these directly in dashboards without further calculation.

GET

### /v1/rent

Rent benchmarks — Zillow ZORI + HUD FMR + Census median gross rent

A lighter, faster rent-only call. Returns Zillow ZORI observed rent index with YoY and MoM change, HUD Fair Market Rents by bedroom for every county, Census median gross rent, and source metadata. Use this instead of /v1/market when you only need rent data.

Perfect for:

- Landlord pricing tools — pull FMR by bedroom for Section 8 compliance
- Rent estimation widgets — low-latency rent lookup for property cards
- Subsidy calculators — HUD FMR is the exact figure used for vouchers

Parameters

| Parameter | Type   | Required  | Description         |
|-----------|--------|-----------|---------------------|
| zip       | string | Optional* | 5-digit ZIP code    |
| state     | string | Optional* | 2-letter state code |
| metro     | string | Optional* | Metro area name     |

\* At least one location parameter is required.

Example Request

```
GET /v1/rent?zip=55104
GET /v1/rent?state=TX
```

### Example Response (abbreviated)

```
{
  "location": { "zip": "55104" },
  "rent": {
    "median_asking_rent": 1534,
    "rent_yoy_pct": 3.6,
    "rent_mom_pct": 0.4,
    "fmr_1br": 1275,
    "fmr_2br": 1510,
    "fmr_3br": 1980,
    "fmr_year": 2025
  },
  "sources": {
    "rent_index": "Zillow ZORI",
    "fmr": "HUD Fair Market Rents"
  }
}
```

**GET**

## /v1/listing

Listing market data — price, DOM, inventory, price reductions

Realtor.com Research listing data for a ZIP, state, or metro. Returns median listing price, days on market, active inventory, new listings, price reduction count, price per sqft, and monthly history.

### Perfect for:

- Market velocity dashboards — track DOM trends over time
- Investment screeners — filter markets by inventory pressure
- Buyer/seller tools — price reduction count signals negotiating power

### Parameters

| Parameter             | Type   | Required  | Description         |
|-----------------------|--------|-----------|---------------------|
| <a href="#">zip</a>   | string | Optional* | 5-digit ZIP code    |
| <a href="#">state</a> | string | Optional* | 2-letter state code |
| <a href="#">metro</a> | string | Optional* | Metro area name     |

\* At least one required.

### Example Request

```
GET /v1/listing?zip=55104
GET /v1/listing?state=FL
```

### Example Response (abbreviated)

```
{
  "listing": {
    "median_listing_price": 319950,
    "price_yoy_pct": 3.0,
    "median_days_on_market": 33,
    "active_listings": 112,
    "new_listings": 28,
    "price_reduced_count": 9,
    "price_per_sqft": 213,
    "median_sqft": 1603
  }
}
```

**GET**

### /v1/estimate

Proprietary rent AVM — low/mid/high range with confidence score

PropData's weighted rent estimate engine. Blends four independent sources — Zillow ZORI (40%), Census ACS (25%), HUD FMR (20%), Redfin cap rate (15%) — into a low/mid/high range with a confidence score and market trend. This is the only rent estimate that shows you exactly how it was calculated.

#### Perfect for:

- Landlord software — bedroom-level rent range for any ZIP
- Lease pricing tools — confidence score tells you how much to trust the estimate
- Investor underwriting — mid estimate with methodology audit trail

#### Parameters

| Parameter             | Type    | Required  | Description                      |
|-----------------------|---------|-----------|----------------------------------|
| <a href="#">zip</a>   | string  | Optional* | 5-digit ZIP code                 |
| <a href="#">state</a> | string  | Optional* | 2-letter state code              |
| <a href="#">beds</a>  | integer | Optional  | 0=studio, 1, 2, 3, or 4 bedrooms |

\* At least one location parameter required.

#### Example Request

```
GET /v1/estimate?zip=55104&beds=2
GET /v1/estimate?state=MN&beds=3
```

### Example Response (abbreviated)

```
{
  "location": { "zip": "55104", "beds": 2 },
  "estimate": {
    "monthly_low": 1288,
    "monthly_mid": 1400,
    "monthly_high": 1512,
    "confidence_pct": 82,
    "market_trend": "stable",
    "yoy_rent_change": "+3.6%"
  },
  "model": "PropData Weighted Multi-Source",
  "sources_used": 4
}
```

■ Omitting *"beds"* returns an overall market estimate. A *"confidence\_pct"* below 65 indicates sparse source coverage for that ZIP — treat as directional rather than precise.

# Property & Parcel

GET /v1/property · /by-location · /delta · /coverage · /parcel-geometry

GET

## /v1/property

Parcel lookup — owner, value, sale history, enrichment pipeline

The primary property lookup endpoint. Search by address, parcel ID, county FIPS, or ZIP. Returns owner name, mailing address, assessed value, market value, improvement value, last sale price and date, bedrooms, bathrooms, sqft, year built, and enrichment status. Add `enrich=full` for tax context, rent context, comps, risk flags, confidence scoring, equity estimate, and flood zone.

When a property is not yet production-indexed, PropData fires a source-backed enrichment pipeline instead of returning null — the Miss Flywheel. Owner names are never fabricated.

### Perfect for:

- Acquisition tools — pull owner, value, and equity before making an offer
- Portfolio analysis — batch property lookup by parcel ID list
- Due diligence — `enrich=full` returns flood zone, tax delinquency flags, comps
- Lead qualification — absentee owner flag and equity estimate in one call

### Parameters

| Parameter                     | Type    | Required  | Description                                                 |
|-------------------------------|---------|-----------|-------------------------------------------------------------|
| <code>address</code>          | string  | Optional* | Street address — e.g. 123 Main St                           |
| <code>city</code>             | string  | Optional  | City name                                                   |
| <code>state</code>            | string  | Optional  | 2-letter state code                                         |
| <code>zip</code>              | string  | Optional* | 5-digit ZIP code                                            |
| <code>parcel</code>           | string  | Optional* | Parcel ID — also: <code>parcel_id</code> , <code>apn</code> |
| <code>county_fips</code>      | string  | Optional  | County FIPS — speeds exact match                            |
| <code>enrich</code>           | string  | Optional  | Set to full for deep enrichment                             |
| <code>include_geometry</code> | integer | Optional  | Set to 1 for GeoJSON boundary                               |

\* At least one of `address+zip`, `parcel`, or `county_fips` is required.

### Example Requests

```
GET /v1/property?parcel=12086-0101100501060&enrich=full
GET /v1/property?address=10548+Kentucky+Ave+S&zip=55438
GET /v1/property?parcel=0821220320820&county_fips=12086
```

### Example Response — exact parcel match (abbreviated)

```
{
  "ok": true, "count": 1,
  "results": [{
    "parcel_id": "0101100501060",
    "county_fips": "12086",
    "full_address": "1 NE 1 ST, Miami, FL 33132",
    "owner_name": "MIAMI WORLDWIDE EXCHANGE LLC",
    "market_value": 27500000,
    "last_sale_price": 27500000,
    "last_sale_date": "2024-01-16",
    "property": { "sqft": 225054, "year_built": 1926 },
    "flags": { "is_absentee_owner": true },
    "match_level": "parcel_county_fips_exact",
    "enrichment_status": "partial",
    "missing_fields": ["assessed_value", "annual_tax"]
  }]
}
```

■ **"enrichment\_status": "partial"** means PropData matched the parcel and returned all available source-backed fields. Some fields vary by county source. **"missing\_fields"** lists what wasn't available from that source.

**GET**

## /v1/property/by-location

Coordinate-to-parcel via point-in-polygon — no address string matching

Drop a lat/lng, get the legal parcel. PropData resolves the county FIPS from the coordinates, queries the verified county ArcGIS parcel GIS layer, and returns the parcel whose polygon contains that point. No address string matching is involved — this avoids the common problem where a map provider's geocoded address differs from the legal situs address.

If the point lands in a road, right-of-way, water, or unparceled area, PropData returns a clean no-match response instead of forcing a wrong parcel.

### Perfect for:

- Map-based property tools — user clicks a map, you get the legal parcel
- GIS workflows — coordinate from GPS or geocoder → assessor record
- AVM tools — resolve any lat/lng to parcel ID, then call /v1/property

### Parameters

| Parameter               | Type    | Required | Description                      |
|-------------------------|---------|----------|----------------------------------|
| <b>lat</b>              | float   | Required | Latitude — also: latitude        |
| <b>lng</b>              | float   | Required | Longitude — also: lon, longitude |
| <b>include_geometry</b> | integer | Optional | Set to 1 for GeoJSON boundary    |
| <b>county_fips</b>      | string  | Optional | Hint to speed resolution         |

|       |        |          |                          |
|-------|--------|----------|--------------------------|
| state | string | Optional | Hint to speed resolution |
|-------|--------|----------|--------------------------|

### Example Request

```
GET /v1/property/by-location?lat=25.8990225371643&lng=-80.237689454084
GET /v1/property/by-location?lat=25.899&lng=-80.237&include_geometry=1
```

### Example Response — successful match (Miami-Dade)

```
{
  "ok": true,
  "match_level": "point_in_polygon",
  "coverage_status": "matched",
  "county": {
    "county_fips": "12086",
    "county_name": "Miami-Dade County",
    "state": "FL"
  },
  "source_name": "Miami-Dade MD_LandInformation Parcels",
  "parcel": {
    "parcel_id": "0821220320820",
    "situs_address": "2296 NW 136 ST",
    "situs_city": "Opa-locka",
    "owner_name": "PRAXIS INVESTMENTS LLC",
    "county_fips": "12086"
  },
  "geometry": {
    "available": true,
    "centroid": { "lat": 25.899, "lon": -80.237 }
  },
  "next_step": "/v1/property?parcel=0821220320820"
}
```

- Use ["next\\_step"](#) to chain immediately to `/v1/property` for full field enrichment on the matched parcel.
- If ["coverage\\_status"](#) is `"source_not_configured_for_county"`, that county's GIS source isn't yet configured. Fall back to `/v1/property` with address + zip.

GET

## /v1/property/delta

Changed property records since a timestamp — cursor paginated

Builder+

Returns all property records that changed since a given ISO 8601 timestamp. Supports cursor pagination for large result sets. Filter by state or ZIP. Use this to keep your own database in sync with PropData's nightly refresh cycle without re-pulling the full dataset.

### Perfect for:

- Database sync jobs — pull only records changed since your last sync
- Change monitoring — detect new sales, value updates, owner changes
- Pipeline triggers — fire downstream enrichment only on changed records

## Parameters

| Parameter              | Type    | Required | Description                           |
|------------------------|---------|----------|---------------------------------------|
| <a href="#">since</a>  | string  | Required | ISO 8601 — e.g. 2026-06-01T00:00:00Z  |
| <a href="#">zip</a>    | string  | Optional | Filter to a single ZIP                |
| <a href="#">state</a>  | string  | Optional | Filter to a state                     |
| <a href="#">cursor</a> | string  | Optional | Pagination cursor from prior response |
| <a href="#">limit</a>  | integer | Optional | Records per page                      |

## Example Request

```
GET /v1/property/delta?since=2026-06-01T00:00:00Z
GET /v1/property/delta?since=2026-06-01T00:00:00Z&state=MN
```

## Example Response (abbreviated)

```
{
  "ok": true, "count": 142,
  "cursor": "eyJzYXN0X2lkIjoimTIzNDU...",
  "since": "2026-06-01T00:00:00Z",
  "records": [{
    "parcel_id": "27123-456789",
    "changed_fields": ["market_value", "annual_tax"],
    "market_value": 412000,
    "updated_at": "2026-06-15T08:23:11Z"
  }]
}
```

**GET**

### /v1/property/coverage

Active assessor coverage by county — no parameters required

Returns county-level assessor coverage metadata — which counties are live, which sources are configured, and availability status. Use this to understand PropData's coverage in your target markets before building a workflow.

## Example Request & Response

```
GET /v1/property/coverage

{
  "ok": true,
  "counties_configured": 847,
  "states_with_coverage": 50,
  "coverage": [{
    "county_fips": "27123",
    "county": "Ramsey", "state": "MN",
    "status": "live",
    "source": "METC Regional Parcel Service"
  }]
}
```

GET

### /v1/parcel-geometry

GeoJSON parcel boundary, centroid, and bbox from county GIS — Beta

Returns a GeoJSON parcel boundary polygon, centroid coordinates, and bounding box for a parcel identified by county FIPS and parcel ID. Coverage depends on the county GIS source being configured. If geometry is not available, returns HTTP 200 with `"geometry_available": false` and a coverage status message.

Parameters

| Parameter   | Type   | Required | Description                      |
|-------------|--------|----------|----------------------------------|
| county_fips | string | Required | County FIPS — also: fips         |
| parcel      | string | Required | Parcel ID — also: parcel_id, apn |

Example Request & Response (abbreviated)

```
GET /v1/parcel-geometry?county_fips=27053&parcel=27-117-21-14-0085

{
  "ok": true,
  "geometry_available": true,
  "parcel_id": "27-117-21-14-0085",
  "county_fips": "27053",
  "geometry": {
    "type": "Feature",
    "geometry": { "type": "Polygon", "coordinates": [[[...]]] },
    "properties": {
      "centroid": { "lat": 44.978, "lon": -93.265 },
      "bbox": [-93.267, 44.976, -93.263, 44.980]
    }
  }
}
```

## Intelligence & Analytics

GET /v1/neighborhood · /v1/zip-intel · /v1/state-intel · /v1/comps · /v1/geocode

GET

### /v1/neighborhood

Neighborhood profile — demographics, safety, schools, FEMA hazard

Returns a full neighborhood intelligence profile for a ZIP code: Census ACS demographics (population, income, vacancy, renter %), safety score, school quality score, walkability label, FEMA hazard context, and affordability metrics. An A–F safety grade and neighborhood score are derived from composite signals.

**Perfect for:**

- Home buyer tools — show school scores and safety grade alongside listings
- Rental platforms — neighborhood context for property pages
- Institutional due diligence — FEMA hazard and social vulnerability signals

**Parameters**

| Parameter | Type    | Required  | Description                                       |
|-----------|---------|-----------|---------------------------------------------------|
| zip       | string  | Optional* | 5-digit ZIP code                                  |
| city      | string  | Optional  | City name (use with state for city-level queries) |
| state     | string  | Optional  | 2-letter state code                               |
| limit     | integer | Optional  | Max results for city+state queries                |

\* zip or city+state required.

**Example Request**

```
GET /v1/neighborhood?zip=55104
GET /v1/neighborhood?state=NY&city=Albany
```

**Example Response (abbreviated)**

```
{
  "zip": "55104",
  "location": { "city": "Saint Paul", "state": "MN" },
  "demographics": {
    "population": 18420,
    "median_hh_income": 52800,
    "vacancy_rate_pct": 8.2,
    "renter_pct": 61.4,
    "rent_to_income_ratio": 31.8
  },
  "neighborhood_score": 68,
  "walkability_label": "walkable",
  "safety_grade": "C",
  "census_enrichment": { "source": "Census ACS 5-Year" }
}
```

GET

**/v1/zip-intel**  
ZIP investment intelligence + AI-generated 3-sentence narrative

Returns market scores, gross yield estimate, vacancy, unemployment, year-over-year appreciation, and an AI-generated 3-sentence investment summary that opens with a BUY / HOLD / PASS signal based solely on the available market evidence. No opinions — only data-backed analysis.

Perfect for:

- Investment screeners — BUY/HOLD/PASS signal for any ZIP in seconds
- AI real estate agents — drop the narrative directly into LLM context
- Market summaries — automated investment brief for any ZIP code

Parameters

| Parameter | Type   | Required | Description      |
|-----------|--------|----------|------------------|
| zip       | string | Required | 5-digit ZIP code |

Example Request

```
GET /v1/zip-intel?zip=78701
```

Example Response (abbreviated)

```
{
  "zip": "78701",
  "market": {
    "avm_value": 485000,
    "gross_yield": 6.4,
    "yoy_change_pct": 3.1,
    "unemployment": 3.8
  },
  "ai_analysis": "BUY. ZIP 78701 offers compelling 6.4% yield...",
  "data_quality": { "confidence": "high" }
}
```

**GET**

### /v1/state-intel

State-level signal — BUY/HOLD/AVOID, friction score, top metros

Returns a state-level market signal with BUY/HOLD/AVOID recommendation, friction score (landlord-friendliness, regulatory complexity), grade, median price, DOM, months of supply, sale-to-list ratio, top metro areas, and risk index.

#### Parameters

| Parameter    | Type   | Required | Description         |
|--------------|--------|----------|---------------------|
| <b>state</b> | string | Required | 2-letter state code |

#### Example Request & Response (abbreviated)

```
GET /v1/state-intel?state=MN
```

```
{
  "state": "MN", "signal": "BUY", "grade": "B+",
  "friction_score": 42,
  "median_listing_price": 345000,
  "top_metros": ["Minneapolis", "Saint Paul", "Rochester"],
  "market_trend": "stable"
}
```

**GET**

### /v1/comps

Comparable sales ranked by price, sqft, recency, and bedroom count

Returns comparable property records for a ZIP or parcel, ranked by price similarity, square footage match, recency, bedroom count, and location proximity. Use as a comps engine for AVM tools, offer analysis, or investment underwriting.

#### Parameters

| Parameter  | Type   | Required  | Description      |
|------------|--------|-----------|------------------|
| <b>zip</b> | string | Optional* | 5-digit ZIP code |

|        |         |           |                                       |
|--------|---------|-----------|---------------------------------------|
| beds   | integer | Optional  | Filter by bedroom count               |
| parcel | string  | Optional* | Parcel ID for property-specific comps |

\* zip or parcel required.

Example Request & Response (abbreviated)

```
GET /v1/comps?zip=55104&beds=3

{
  "zip": "55104", "count": 8,
  "comps": [{
    "address": "789 Oak Ave",
    "sale_price": 312000,
    "sale_date": "2026-04-12",
    "sqft": 1640, "beds": 3,
    "distance_miles": 0.4
  }]
}
```

GET

### /v1/geocode

Address or ZIP lookup from PropData's own parcel database

Resolves an address or ZIP to lat/lng, county FIPS, county name, and normalized location context — using PropData's own county assessor parcel database rather than a third-party geocoder.

Parameters

| Parameter | Type   | Required  | Description      |
|-----------|--------|-----------|------------------|
| address   | string | Optional* | Street address   |
| zip       | string | Optional* | 5-digit ZIP code |

\* address or zip required.

Example Request & Response (abbreviated)

```
GET /v1/geocode?address=123+Main+St+Saint+Paul+MN

{
  "ok": true,
  "address": "123 Main St, Saint Paul, MN 55104",
  "zip": "55104", "county": "Ramsey",
  "state": "MN", "county_fips": "27123",
  "lat": 44.954, "lng": -93.093
}
```

## Lead Generation

GET /v1/skip-trace · GET /v1/preforeclosure · GET /v1/preforeclosure/delta

**GET**

### /v1/skip-trace

Owner phones, email, mailing address, equity estimate

**Builder+**

Returns owner contact information — phone numbers, email, mailing address — plus an equity estimate and lead status flag. Contact fields are source-backed when available and are never fabricated. Available on Builder tier and above.

#### Perfect for:

- Acquisition outreach — direct mail and phone lists from parcel data
- Lead scoring — equity estimate + absentee owner flag = motivated seller signal
- Wholesaling workflows — skip trace on pre-foreclosure leads

#### Parameters

| Parameter      | Type   | Required  | Description         |
|----------------|--------|-----------|---------------------|
| <b>address</b> | string | Optional* | Street address      |
| <b>zip</b>     | string | Optional* | 5-digit ZIP code    |
| <b>state</b>   | string | Optional* | 2-letter state code |

\* At least one required. address+zip gives the most targeted result.

#### Example Request & Response (abbreviated)

```
GET /v1/skip-trace?address=123+Main+St&zip=55104
```

```
{
  "ok": true,
  "owner_name": "Jane Smith",
  "phones": ["+16515550101", "+16515550202"],
  "email": "jsmith@email.com",
  "mailing_address": "123 Main St, Saint Paul MN 55104",
  "equity_estimate": 148000,
  "lead_status": "absentee_owner"
}
```

**GET**

### /v1/preforeclosure

Pre-foreclosure leads — notice date, auction date, lender, default amount

Returns active pre-foreclosure records for a ZIP or state. Each lead includes notice date, scheduled auction date, case number, lender, trustee, default amount, loan amount, and assessed value. Combine with /v1/property and /v1/skip-trace for a complete lead profile.

Perfect for:

- Distressed property acquisition — find deals before they hit the market
- Note investors — lender, loan amount, default status in one call
- Wholesale lead generation — pre-screened by distress signal

Parameters

| Parameter | Type   | Required  | Description         |
|-----------|--------|-----------|---------------------|
| zip       | string | Optional* | 5-digit ZIP code    |
| state     | string | Optional* | 2-letter state code |

\* zip or state required.

Example Request & Response (abbreviated)

```
GET /v1/preforeclosure?state=TX

{
  "ok": true, "count": 14,
  "leads": [{
    "address": "456 Elm St, Dallas TX 75201",
    "notice_date": "2026-05-01",
    "auction_date": "2026-08-15",
    "lender": "First National Bank",
    "default_amount": 18400,
    "assessed_value": 287000
  }]
}
```

GET

**/v1/preforeclosure/delta**  
Foreclosure records updated since a timestamp — cursor paginated

Builder+

Returns foreclosure records that changed status since a given timestamp. Use to detect new filings, auction date changes, or status updates without re-pulling the entire dataset. Cursor pagination for large result sets.

Parameters

| Parameter | Type   | Required | Description                               |
|-----------|--------|----------|-------------------------------------------|
| since     | string | Required | ISO 8601 — e.g. 2026-06-01T00:00:00Z      |
| zip       | string | Optional | Filter to a ZIP                           |
| state     | string | Optional | Filter to a state                         |
| status    | string | Optional | Filter by status (e.g. auction_scheduled) |
| cursor    | string | Optional | Pagination cursor from prior response     |

|              |         |          |                  |
|--------------|---------|----------|------------------|
| <b>limit</b> | integer | Optional | Records per page |
|--------------|---------|----------|------------------|

### Example Request & Response (abbreviated)

```
GET /v1/preforeclosure/delta?since=2026-06-01T00:00:00Z&state=TX
```

```
{
  "ok": true, "since": "2026-06-01T00:00:00Z",
  "count": 23,
  "cursor": "eyJzYXN0X2lk...",
  "records": [{
    "parcel_id": "27053-789012",
    "status_change": "auction_scheduled",
    "auction_date": "2026-09-10",
    "updated_at": "2026-07-01T11:45:00Z"
  }]
}
```

## MCP / AI Connector

Use PropData inside ChatGPT and Claude — no code required

### What is the MCP Connector?

PropData ships a dedicated OAuth-protected MCP (Model Context Protocol) gateway — a separate worker from the REST API, built to let supported AI assistants call live PropData tools directly. Instead of writing API calls yourself, you connect PropData to your AI assistant once. From that point, you can ask the assistant natural-language questions and it calls the live PropData tools automatically.

### MCP Connector URL

```
https://propdata-mcp.sales-fd3.workers.dev/mcp
```

### How to Connect

- 1 Open PropData MCP connector in your supported AI assistant (Claude or ChatGPT)
- 2 Enter your email and your direct PropData API key when prompted
- 3 PropData validates your key against your active subscription
- 4 The AI assistant can now call PropData tools using your subscriber key

**RapidAPI subscribers:** RapidAPI keys work for REST calls only. The MCP connector requires a direct PropData API key. Contact PropData support — we'll issue one tied to your account at no extra cost.

### Available MCP Tools

| Tool                                | Calls          | Returns                                                 |
|-------------------------------------|----------------|---------------------------------------------------------|
| <a href="#">propdata_market</a>     | /v1/market     | Full market intelligence for a ZIP                      |
| <a href="#">propdata_rent</a>       | /v1/rent       | Rent, HUD FMR, Census enrichment, affordability metrics |
| <a href="#">propdata_coverage</a>   | /v1/coverage   | Coverage score, available and missing layers for a ZIP  |
| <a href="#">propdata_source_map</a> | /v1/source-map | Field-level source provenance and confidence per ZIP    |

Additional tool categories available through the connector include comps, neighborhood intelligence, parcel geometry, and due diligence snapshots. Contact PropData support for the full tool list after connecting.

### Example Prompts Inside Claude or ChatGPT

Use PropData to get market data for ZIP 55104.

Use PropData to look up the property at:  
lat 25.8990225371643, lng -80.237689454084.

Use PropData to find rent estimates and comps for ZIP 78701.

Use PropData to analyze this address for acquisition due diligence.

Use PropData to show me the coverage score for ZIP 37203.

## Utilities

GET /v1/changelog · GET /v1/stats · GET /v1/usage · GET /v1/health

GET

### /v1/changelog

Versioned log of all endpoint additions, fixes, and pipeline updates. ...

Versioned log of all endpoint additions, fixes, and pipeline updates. No authentication required. Subscribe to this endpoint to detect new endpoints, source changes, and MCP connector updates before they affect your integration.

GET /v1/changelog

```
{
  "version": "2026.07.03",
  "entries": [{
    "date": "2026-07-03",
    "type": "endpoint_addition",
    "description": "coordinate-parcel validation improvements"
  }]
}
```

GET

### /v1/stats

Live parcel count and API stats. No authentication required. Use this ...

Live parcel count and API stats. No authentication required. Use this to verify the current indexed parcel count and confirm the API is live.

GET /v1/stats

```
{
  "parcels_indexed": 166482391,
  "states_covered": 50,
  "counties_indexed": 2814,
  "last_updated": "2026-07-09T06:00:00Z"
}
```

GET

### /v1/usage

Returns your current key tier, request count this window, plan limit, ...

Returns your current key tier, request count this window, plan limit, and reset timestamp. Call this whenever you hit a 429 to understand your current window before retrying.

```
GET /v1/usage
```

```
{
  "tier": "builder", "plan": "Builder",
  "status": "active",
  "requests_this_hour": 42,
  "limit_per_hour": 500,
  "reset_at": "2026-07-09T15:00:00Z"
}
```

**GET**

## /v1/health

API health check. Returns status, version, source health, plan mapping...

API health check. Returns status, version, source health, plan mapping, and freshness metadata. Use in monitoring and uptime checks.

```
GET /v1/health
```

```
{
  "status": "ok", "version": "v1",
  "parcels": "166M+",
  "sources": {
    "zori": "live", "hud_fmr": "live",
    "realtor": "live"
  },
  "generated_at": "2026-07-09T14:32:11Z"
}
```

# Troubleshooting & Support

---

## 400 — Bad Request

A required location or property identifier is missing.

- Add zip, state, or metro to market/rent/listing/estimate calls
- Add address+zip, parcel, or county\_fips to property calls
- Add lat and lng to /v1/property/by-location
- Add since (ISO 8601) to delta endpoints
- Path must start with /v1/ — never /api/v1/

## 401 — Unauthorized

API key is invalid, missing, or your RapidAPI subscription is inactive.

- Check x-rapidapi-key header is present and correct
- Check x-rapidapi-host is exactly: propdata-real-estate-market-intelligence-api.p.rapidapi.com
- Confirm your RapidAPI subscription is active
- For MCP access, you need a direct PropData key — contact support

## 403 — Forbidden

The endpoint requires a higher plan tier.

- Skip trace requires Builder+ — check your current tier at /v1/usage
- Nationwide delta queries may require Builder+ depending on plan
- Contact PropData to discuss tier upgrades

## 404 — Not Found

Endpoint path is wrong.

- Correct: /v1/market — Incorrect: /api/v1/market
- All paths use /v1/ prefix — double-check spelling

## 429 — Rate Limited

You've exceeded your plan's request limit for the current window.

- Call GET /v1/usage immediately to see reset time
- Stagger requests 500ms–1000ms apart
- Cache market/rent lookups for 1–24 hours client-side
- Reduce duplicate calls — same ZIP + same endpoint can share a cached result
- Upgrade your plan or contact PropData for production integration limits

## 500 — Server Error

Unexpected server-side error.

- Retry after a 5–10 second delay
- If persistent, contact PropData support with the full request + response

---

## Common Behavior Notes

- Most valid no-match property searches return HTTP 200 with structured no-match or coverage metadata — not a hard 404.
- Partial enrichment is normal. A parcel can match exactly while some fields are unavailable from that county source. Look for `enrichment_status` and `missing_fields` in the response.
- First uncached property lookups may take longer due to live source queries. Repeat lookups are faster due to PropData's Cloudflare KV edge cache.
- If `/v1/property/by-location` returns `source_not_configured_for_county`, the county GIS source isn't configured yet — fall back to `/v1/property` with address + zip.
- Owner names and contact fields are never fabricated. They remain null until source-verified from a permitted government record.

---

## Contact & Support

For RapidAPI integration support, use the RapidAPI Discussion tab.

For MCP access, enterprise terms, custom limits, direct API keys, production integration support, or white-label inquiries — contact PropData directly.

|          |                                                                          |
|----------|--------------------------------------------------------------------------|
| Email    | <a href="mailto:sales@proptechusa.ai">sales@proptechusa.ai</a>           |
| Phone    | 1-888-784-3881                                                           |
| Website  | <a href="https://propdata.proptechusa.ai">propdata.proptechusa.ai</a>    |
| RapidAPI | <code>propdata-real-estate-market-intelligence-api.p.rapidapi.com</code> |

When contacting support, please include: endpoint used, full request URL (without your API key), response status code, response body, timestamp, and your RapidAPI username or PropData account email.