ForecastWatch™

The ForecastWatch API

Your first hands-on session

Eric Floehr · ForecastWatch

What you'll learn today

- How API authentication works
- The handful of endpoints that answer most analytical questions
- A real customer use case, built live in about 30 lines of Python

After today you'll have a working Postman setup, an Excel workbook pulling live data, and a Colab notebook to fork and adapt.

Before we start

Three things to open:

- **Postman** desktop app, free from postman.com
- **A ForecastWatch API key** from app3.forecastwatch.com → Tools → API Portal (format: `fw_xxxxxxxxxxxx`)
- **A Google account** for Google Colab. The Python notebook runs in your browser, no install needed.

> ***Windows users:*** *I'd suggest skipping a local Python install for this session. Colab gives you the same experience in a browser tab.*

The API in 60 seconds

The API is REST and JSON. Authentication is a Bearer token in a header.

```
GET https://api.forecastwatch.com/v1/insights/monthly/?month=240
Authorization: Bearer fw_yourKeyHere
```

One key, one header, every endpoint. Your account determines which provider's data the API returns, so there's no provider ID to pass anywhere.

The endpoint catalog at `GET /v1/api-docs/` lists every endpoint with a curl example. It covers everything we don't have time for today.

Today's flow

Time	What we'll do
~15 min	**Postman**: set up auth, hit your first endpoints, chain requests
~5 min	**Excel via Power Query**: pull the API into a refreshable spreadsheet
~10 min	**Python in Colab**: build a real "chronic miscalibrated stations" report
~10 min	**Q&A**

I'll share the Postman collection, Excel spreadsheet, Jupyter notebook, and recording with you after the session.

Live demo

Switching screens. Postman first, then Excel, then Colab.

What we just built

In about 25 minutes we built a working tool to answer a real customer question:

> *"Are there individual stations where my 1-day-out forecast keeps landing in the **bold_wrong** quadrant, where my model is systematically miscalibrated?"*

Three endpoints, a pandas `groupby`, and a chart.

If this is interesting to your forecasting team, you can run it as often as you want.

Three patterns to take with you

1. Chained calls. Discover IDs from `/v1/reference/...` endpoints rather than hardcoding them. Your dashboards stay current when our internal IDs change.

2. Parameterize from your environment. Postman variables, Excel cells, Python constants. Same idea everywhere. Change one input, get a different answer.

3. Filter server-side. `?quadrant=bold_wrong` is cheaper than fetching everything and filtering in your code. Same applies to `&days_out=`, `&market=`, `&country=`, `&state=`.

Going further

We covered the headline endpoints. The catalog at `/v1/api-docs/` has the rest:

- **Forecast Risks**: divergence outcomes, station-level forecast history, watchlists
- **Exports**: bulk CSV/Excel downloads with saved presets
- **Government comparison**: head-to-head against NWS, ECMWF, and others
- **Trends**: multi-month longitudinal views, days-out trends, market position

Same auth, same query-param style, same JSON shape. If you can do today's demo, you can do these.

Resources & next steps

- **API key management:** app3.forecastwatch.com → Tools → API Portal
- **Endpoint catalog (live):** https://app3.forecastwatch.com/platform/api
- **Postman collection, Excel spreadsheet, Jupyter notebook, recording:** I'll email all four after the session
- **Support:** support@forecastwatch.com

ForecastWatch™

Questions?

I'd love to hear what you want to build next.

Eric Floehr

eric@forecastwatch.com

forecastwatch.com