**Dataset Details**

* **Source:** [data.fingrid.fi](https://data.fingrid.fi/en/datasets/191?utm_source=chatgpt.com)
* **Dataset ID:** 191
* **Name:** *Hydropower production in Finland (real-time)*
* **Description:** Total electricity production from all hydropower plants connected to the Finnish grid, measured every 3 minutes.
* **Unit:** **MW (megawatts)**
* **Aggregation level:** **National total** — i.e., *sum of all hydropower plants in Finland*.

**Why I’m Ingesting This**

* It’s your **real-time operational dataset** in the pipeline.
* I’ll use it to:
  + Monitor **current hydropower generation**.
  + Compute **observed capacity factor** by comparing with total installed capacity (from metadata).
  + Compare against **modeled capacity factors** (from Zenodo) as a baseline.

**How It Fits in the Architecture**

| **Layer** | **Dataset** | **Purpose** |
| --- | --- | --- |
| **Bronze** | Fingrid Dataset #191 (raw JSON via API) | Raw ingestion of live hydropower data (3-min frequency). |
| **Silver** | Aggregated hourly averages | Cleansed and aligned for comparison with Zenodo data. |
| **Gold** | Joined with metadata + Zenodo | Compute efficiency, capacity factor, and deviations. |

So the Fingrid data that I’m pulling in the Spark notebook is **dataset 191: Hydropower production in Finland (real-time)** — the live, national-level measurement of hydropower generation in megawatts.

**🎯 The Goal of Project**

**Not** trying to build a massive time-series database of 40+ years of hydropower data.  
Instead, building a **proof-of-concept energy data platform** in **Microsoft Fabric** that demonstrates:

✅ Cloud-native data engineering (Bronze → Silver → Gold)  
✅ Integration of real-time, historical, and metadata sources  
✅ Computation of key hydropower KPIs (like capacity factor & efficiency)  
✅ Visualization and automation (CI/CD, Power BI)

**So, Why These Specific Datasets?**

**1️⃣ Fingrid API – 10 Days of Real-Time Data**

* This is the **“operational data feed.”**
* Purpose:
  + To **simulate real-time ingestion** into your Fabric Lakehouse (Bronze → Silver).
  + To **demonstrate streaming / incremental updates**.
  + To **calculate current capacity factor** (actual generation ÷ installed capacity).
* No need years of Fingrid data — 10 days is enough to show:
  + ETL ingestion pipeline
  + Spark transformations
  + Real-time dashboarding

👉 *It’s the “real, changing data stream.”*

**2️⃣ Zenodo Dataset – Historical Modeled Capacity Factors (1981–2010)**

* This is the **“historical climate baseline.”**
* Purpose:
  + To **provide context**: what’s “normal” hydropower performance for Finland?
  + To **train or compute seasonal averages** (baseline by month/hour).
  + To **compare current performance vs. historical norms**.
* Even though it ends in 2010, it gives us 30 years of hourly data — plenty to build monthly or seasonal averages.

👉 *It’s our “climate potential reference.”*

**3️⃣ Hydropower Metadata – Static Plant Information**

* This is our **“structural data.”**
* Purpose:
  + To get total installed capacity of Finnish hydropower plants (MW).
  + To classify by type (run-of-river, storage, pumped).
  + To compute *observed capacity factor* for Fingrid data.

👉 *It’s our “dimension table / lookup table.”*

**What I Compute / Analyze (The Final Output)**

| **Analysis** | **Formula / Logic** | **Data Source(s)** |
| --- | --- | --- |
| **Observed Capacity Factor** | Fingrid generation / total installed capacity | Fingrid + Metadata |
| **Historical Capacity Factor (baseline)** | Average Zenodo CF (1981–2010) by month/hour | Zenodo |
| **Deviation / Anomaly** | Observed CF – Historical CF | All three |
| **Type-based Efficiency** | Compare storage vs. run-of-river trends | Metadata + Zenodo |
| **Seasonal Insights** | Monthly average performance vs. baseline | All three |

Then visualize all of this in **Power BI dashboards** (Gold layer).

**What I’m Proving**

By doing this, I’m demonstrating that I can:

1. **Design a modern data platform** — multi-source ingestion, transformation, storage, analytics.
2. **Handle real-time + historical data integration** (a core use case in energy companies).
3. **Work cloud-natively in Microsoft Fabric with Spark, Delta, CI/CD.**
4. **Deliver business insight** — “Are we producing as efficiently as our long-term climate potential allows?”

That’s exactly what an energy-sector data engineering clients want to see.

✅ **In summary:**  
Not building a production-scale forecast system.  
I’m building a *realistic, cloud-native data platform prototype* that:

*Combines live hydropower output, long-term climate-based potential, and plant metadata to analyze operational efficiency and climatic deviations.*

A diagram of a company

AI-generated content may be incorrect.