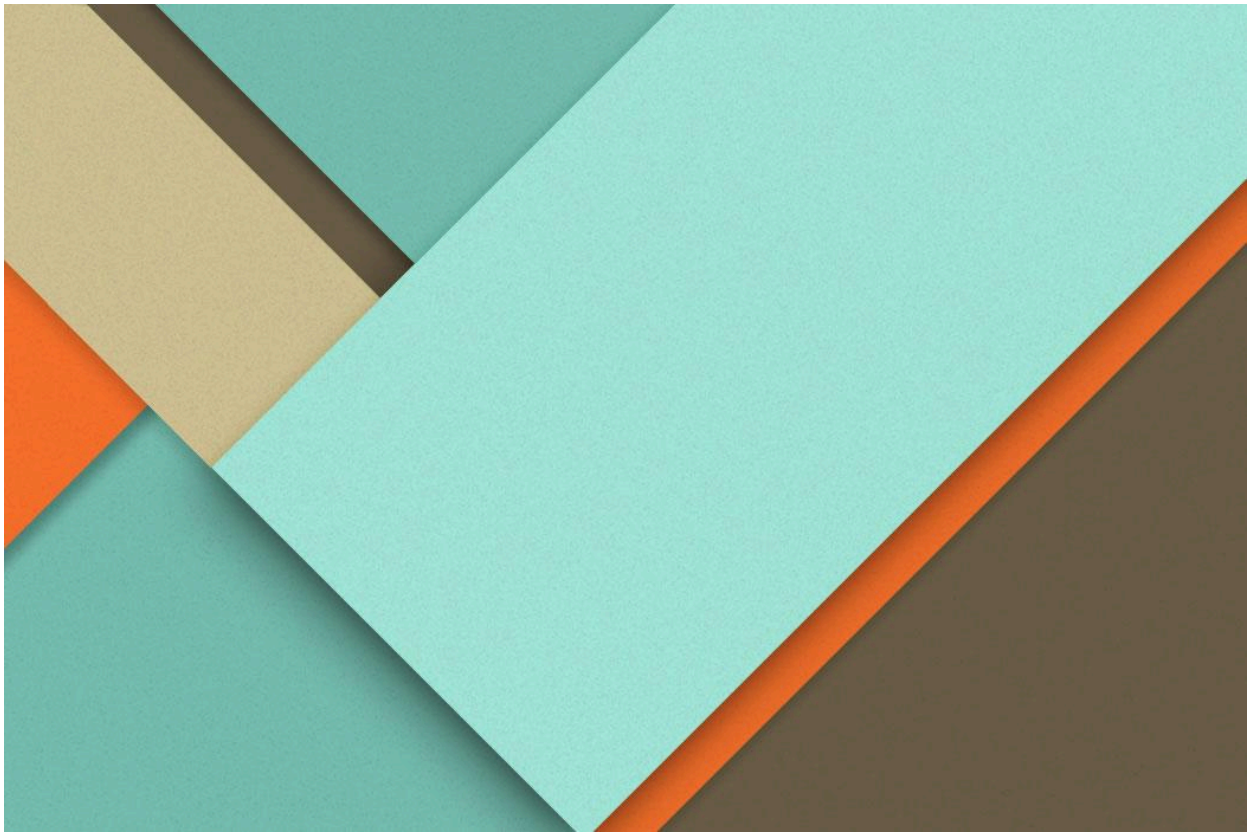


Hasura

Report

Name: Jetashree Ravi

Table of Contents

Overview: Charging Just Got Easier!

Picture this: You're driving through a scenic route, your electric vehicle (EV) humming softly along when suddenly, that ominous *"Low Battery"* light blinks on. *Panic!* You're in an unfamiliar neighborhood, and you have no idea where the nearest charging station is. Will you make it? Or will you have to find yourself stranded, waiting for a tow? Unfortunately, this scenario isn't uncommon, especially with the rise of electric vehicles.

Finding a charging station shouldn't be like finding a needle in a haystack. It's a modern dilemma that many drivers face. But what if we could make this problem go away, and instantly guide drivers to the nearest charging point—no matter where they are?

The Solution: Introducing PromptQL—Your Personal EV Assistant

Enter PromptQL.

What if you could just ask, *"Where's the nearest EV charging station?"* and, *bam*, a real-time, accurate answer appears right before you? No apps to open, no websites to search. It's as easy as sending a text.

PromptQL is a groundbreaking solution that empowers drivers with **real-time information** by tapping into the **NREL API** (National Renewable Energy Laboratory). The NREL API houses a treasure trove of real-time data on alternative fueling stations, including EV charging stations. With **PromptQL**, drivers can query this data through a **chat interface**, effortlessly finding charging stations, their locations, and availability.

This isn't just another app. It's a **revolution in driving** that transforms how we interact with the infrastructure around us—simply and instantly.

The Problem: EV Charging Stations Are Hard to Find

As EV adoption skyrockets, one glaring issue still remains: **Where do I charge next?** The infrastructure just hasn't caught up with the growing number of electric vehicles on the road. Charging stations are still a rare commodity, especially outside of major cities.

While traditional gas stations are easily accessible, EV charging stations aren't quite as ubiquitous—yet. Many drivers often rely on outdated maps, apps, or websites to locate

nearby stations, but these sources are unreliable at best. Information can be inaccurate, out of date, or simply non-existent.

The burning question: *"Where's the closest charging station, and can I actually use it?"*

The Solution: Real-Time Data at Your Fingertips

Enter **PromptQL**—a solution that pulls data from the **NREL API** and turns it into a real-time, user-friendly experience. Now, no more endless searching or unreliable apps.

With **PromptQL**, you can query live, up-to-date data about EV stations instantly. Need to know the closest station? Ask. Wondering about the station's availability? Just ask. The data is pulled in real-time from NREL's extensive database, ensuring that drivers always have the most accurate and reliable information when they need it most.

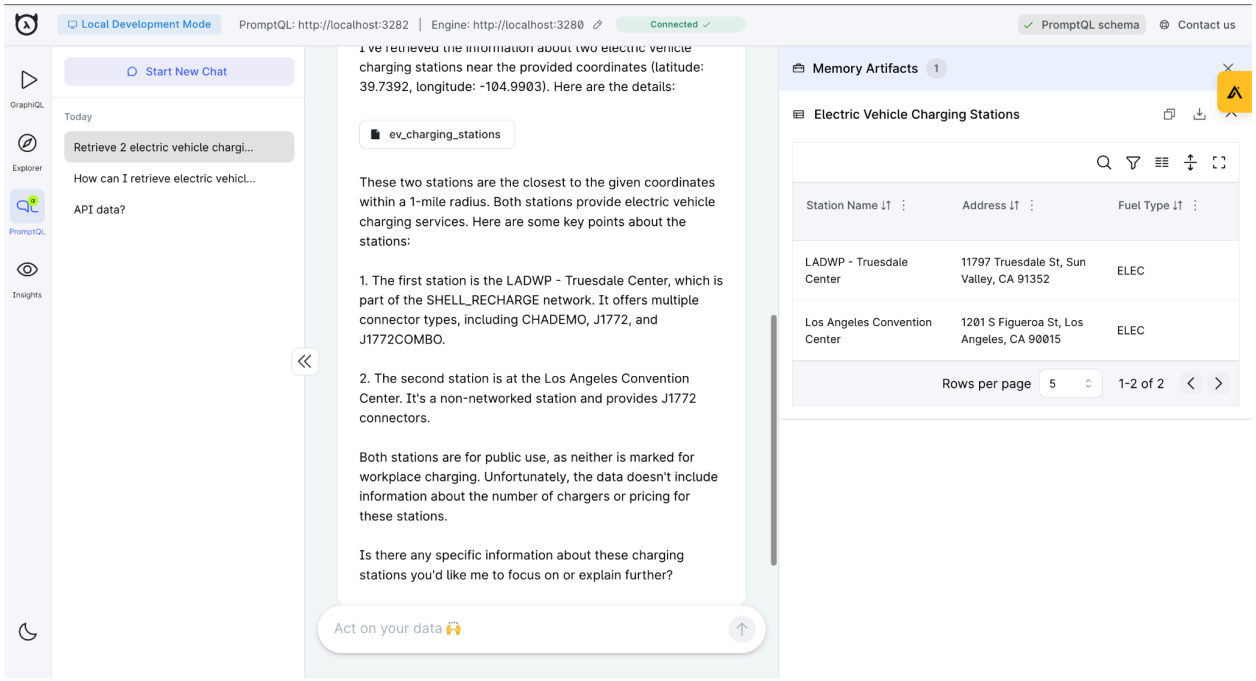
Here's how it works:

1. **Real-time queries**: Ask where the nearest EV stations are.
2. **Instant results**: Receive live updates on station availability, type of connectors, and location.
3. **Easy interaction**: It's just like chatting with a friend—except you're getting information that could save you from running out of charge on the road.

Demo: See the Magic in Action

Watch the demo

The demo showcases how users can interact with the chat interface and get precise, up-to-date information about nearby EV stations.



Challenges with Implementation: The Roadblocks We Overcame

Building a seamless real-time chat interface isn't as simple as waving a magic wand. Here are the challenges we faced along the way:

1. Real-Time Data Accuracy:

Keeping the data fresh and accurate was essential. Charging stations are constantly going offline, undergoing maintenance, or having their availability change. Ensuring the data is **up-to-date** meant we had to implement **robust error handling** and **data validation** techniques.

Further Improvements: The Road Ahead

While this solution is a giant leap forward, there's always room to grow. Here's where we plan to take it next:

1. Personalized Recommendations:

Imagine the app remembering your charging preferences—whether it's your preferred network, connector type, or charging speed. By tracking your history, the system could offer **tailored recommendations**, making every charge more efficient.

2. Real-Time Availability Updates:

With the growing adoption of EVs, charging stations will inevitably become crowded. By integrating real-time **occupancy data**, users will know in advance whether a charging port is available or occupied—much like checking a parking space before pulling in.

3. Expanded Coverage:

While the NREL API is excellent, there's always more to explore. By integrating additional data sources, such as regional networks or utility companies, we can provide **even more comprehensive coverage**—making this a one-stop-shop for all EV charging needs.

Getting Started: Plug Into the Future of EV Charging

Ready to get started? Here's how you can set up the **PromptQL** chat interface with the **NREL API**:

> **Note**: Ensure you have the latest version of the DDN CLI installed. Find installation instructions in the CLI documentation.

Step 1: Update the CLI to the Alpha Release

```
ddn update-cli --version v2.12.0-alpha.2
```

Step 2: Clone the Repo

Use HTTP:

```
git clone https://github.com/Hasura-JetashreeRavi-EVStations
```

Or SSH:

```
git clone git@github.com:Hasura-JetashreeRavi-EVStations
```

Step 3: Set Up Your .env File

```
cd Hasura-JetashreeRavi-EVStations
cp .env.example .env
```

Add your **Anthropic API key**:

```
echo "ANTHROPIC_API_KEY=your-anthropic-api-key" >> .env
```

> **Note**: You can generate an API key here.

Step 4: Initialize the DDN Project

```
ddn project init
```

Step 5: Build and Run PromptQL

First, create the build:

```
ddn supergraph build local
```

Then, start the local services:

```
ddn run docker-start
```

Step 6: Open the Console

In a new terminal, run:

```
ddn console --local
```

Step 7: Start Chatting with NREL API!

From the console, try asking:

```
> Where are the EV charging stations near Mountain View?
```

Appendix

In the journey to create innovative solutions, we explored several promising ideas that, while fascinating, encountered significant challenges during implementation. These ideas, though not fully realized, provided valuable insights into the complexities of real-time data handling and integration. Here are a few ideas that we experimented with, but ultimately, couldn't implement successfully due to various roadblocks.

Using Hallucination Data on Hugging Face to Identify Patterns in AI Hallucinations

The idea behind using **Hallucination Data** from **Hugging Face** was to dive deep into the quirks of AI-generated responses. Imagine a system that could spot when an AI confidently spits out fabricated facts or invents details. By analyzing **hallucination datasets**, we sought to identify patterns—did certain types of questions trigger more hallucinations? Could question complexity or ambiguity influence the frequency of errors? Using **PromptQL**, we aimed to dynamically query the dataset to understand the conditions under which hallucinations occurred most often. This could have led to a new level of **AI error detection**, but we quickly realized that identifying nuanced hallucinations in real-time responses required more than just simple querying—it needed advanced **semantic analysis** and contextual understanding, which was beyond our reach.

Comparing Different Insurance Policies Using HealthSherpa API

Navigating the labyrinth of health insurance plans can be overwhelming—so we imagined a tool that would instantly compare policies using the **HealthSherpa API**. By tapping into this powerful resource, we envisioned helping users find the best plans for their needs, whether they cared about premiums, coverage, or specific networks. The idea was simple: users could ask a question like, "Which plan is right for me?" and instantly see a side-by-side comparison of the best available options. While the concept was compelling, the complexity of insurance data, with its numerous variables and ever-changing plans, proved to be a tough nut to crack. Integrating the **HealthSherpa API** into a real-time, dynamic comparison tool required more than just data retrieval—it needed a seamless blend of **user-friendly interfaces** and **real-time compliance** with privacy regulations, which posed significant challenges. Still, the idea opened new doors to rethinking how **health insurance decisions** can be made easier and more transparent.

Real-Time Traffic Data Integration for Optimal EV Route Planning

Imagine you're driving your electric vehicle (EV) on a road trip, and you need to find the quickest route with available charging stations along the way. The idea was to create a

real-time route planner that would not only take into account the fastest routes but also seamlessly integrate **live traffic data** and **charging station availability**. By connecting to APIs like **Google Maps** and **NREL for charging stations**, the system would dynamically update the best route as you drive, factoring in traffic jams, road closures, and charging station availability in real-time. This would ensure that EV drivers could always find the most efficient and feasible route with charging points along the way.

While the concept was exciting, we ran into several challenges. **Real-time traffic data** changes constantly, making it hard to maintain accurate predictions. Integrating **charging stations in real-time** added another layer of complexity, as station status could change with little notice. The **integration of multiple APIs**, each with its own structure and rate limits, became a technical bottleneck. Despite these hurdles, this idea remains a game-changer for optimizing EV travel, providing a clearer pathway to improving **long-distance electric vehicle journeys**.