# Pokémon MCP Data Resource Documentation

1. Overview

The Pokémon MCP (Model Context Protocol) data resource has been designed to provide a hybrid approach to accessing Pokémon battle-related data. Instead of relying solely on the PokéAPI or Kaggle datasets, this resource strategically combines both. This **reduces unnecessary API load, ensures faster responses, and provides reliable offline availability** of core data.

1. Why Hybrid Approach?

The hybrid approach **combines the static Pokémon dataset from Kaggle with dynamic battle-relevant data from the PokéAPI**. This design ensures efficiency and scalability:

* **Kaggle** dataset provides **bulk Pokémon** **metadata** (base stats, typing, moves, etc.)
* **PokéAPI** provides **additional up-to-date details** (abilities, latest movesets, type relations).
* Combining both **reduces API requests**, preventing rate-limit issues.
* Core data remains available locally, ensuring **offline functionality**.

1. Data Sources

The data resource integrates **multiple sources**, with specific responsibilities:

* **Kaggle Dataset** – Used for Pokémon base stats (HP, Attack, Defense, Speed, etc.), typings, and legacy moves.
* **PokéAPI** – Queried for real-time updates like ability descriptions, new movesets, and type-effectiveness charts.
* Normalization Script – A custom-built **Python script cleans** and unifies both datasets into a consistent **JSON schema**.

1. Data Normalization & JSON Conversion

To ensure **compatibility and standardization**, a normalization pipeline was developed. The script performs:

* **Cleaning redundant fields** from Kaggle dataset.
* **Mapping PokéAPI attributes** into corresponding Kaggle fields.
* **Merging datasets** into a single Pokémon schema.
* Converting each Pokémon entry into a **standardized JSON format**.

1. MCP Data Resource Integration

Once normalized, the **JSON data is hosted within the MCP server**. This allows LLMs or external systems to query Pokémon data efficiently. The MCP server **exposes endpoints** that the LLM can query during battle simulations, such as fetching stats, type matchups, or move effects.

1. Example LLM Queries

Below are examples of how an LLM might interact with the MCP data resource:

* Query: 'Fetch Pikachu’s base stats.' → Response: JSON with HP, Attack, Défense, Speed, etc.
* Query: 'Get ability details for Squirtle.' → Response: JSON with ability name and description.

1. Conclusion

By combining the **Kaggle dataset with PokéAPI**, normalizing the data, and **serving it via MCP**, this resource offers an efficient, scalable, and reliable solution for Pokémon battle simulations. The design **balances offline stability** with real-time accuracy, ensuring a robust foundation for both research and gameplay.