



Accessing the **PI Web API** with Python

kaikecastro



Introduction

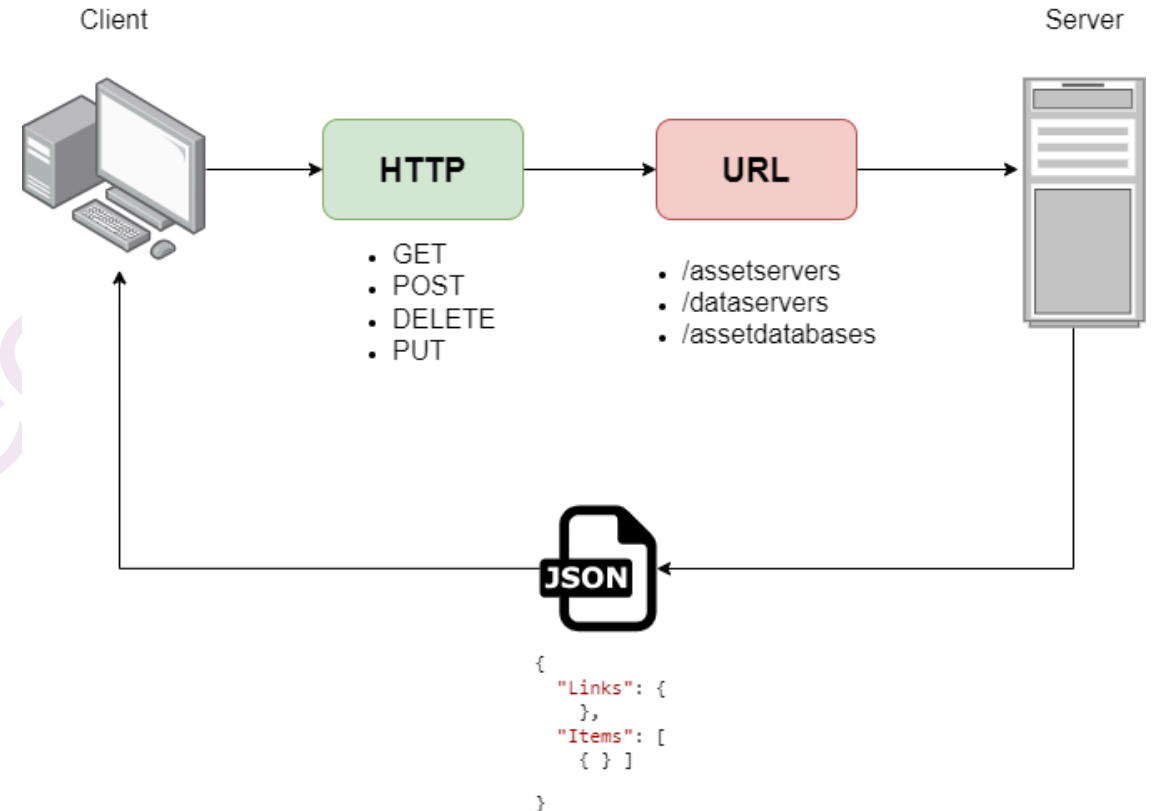
- **PI Web API** is a powerful tool for interacting with PI System programmatically.
- **Python** is a versatile language widely used for automation and data analysis.

This presentation will guide you on how to access and manipulate data using the PI Web API with Python. This process can be useful for integrating PI data with other systems and automating tasks.

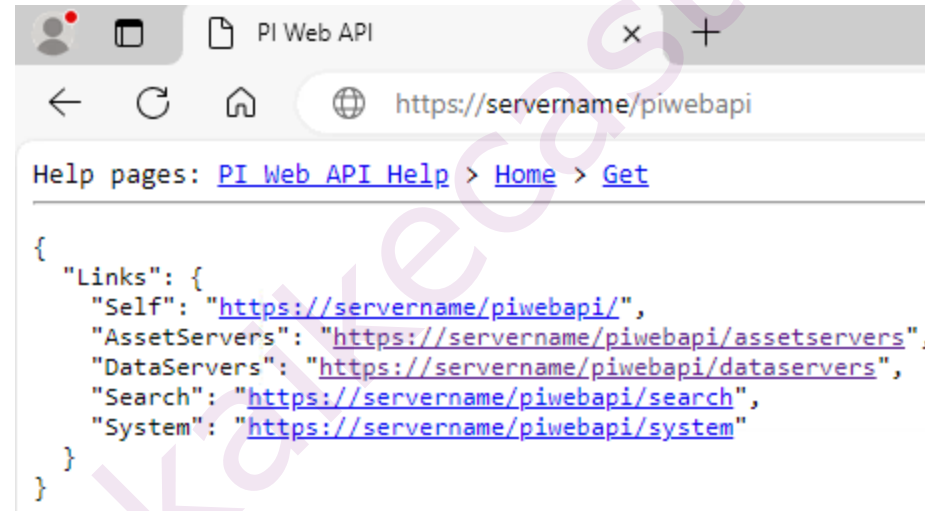


PI Web API

- The PI Web API is a RESTful interface for accessing PI System data (AF & DA) over HTTPS.
- It enables read and write access for client applications, fostering custom integrations.
- The API offers a modern, flexible approach to leverage your data for analytics, visualizations, and more.



Home Web API



Why Use the PI Web API?

- **Automate Data Retrieval:** Fetch and process PI System data programmatically.
- **Data Integration:** Integrate PI data with other applications and systems.
- **Custom Dashboards:** Build custom dashboards based on your specific needs.
- **Scheduled Tasks:** Run automated tasks like data extraction and analysis at regular intervals.



Key Principles of PI Web API

- **Stateless:** Each request is independent, simplifying client development.
- **Resource-oriented:** Interaction revolves around resources representing PI System objects (elements, attributes, points).
- **Navigable:** Links connect resources, enabling seamless navigation and discovery.
- **Uses HTTP Verbs:** GET, POST, PUT, PATCH, DELETE for standard CRUD operations.
- **JSON Format:** Data exchanged in easy-to-parse JSON, compatible with various programming languages.

Getting Started

Prerequisites:

- **PI System Access:** Ensure you have access to a PI System with the PI Web API enabled.
- **Python Installation:** Have Python installed on your system.
- **Required Libraries:** Install the `requests` library: `pip install requests`



Code Example

```
import requests

# Define the PI Web API URL and the endpoint for fetching data
pi_web_api_url = "https://your-pi-server/piwebapi"
data_endpoint = "/elements/{webId}/elements"

# Specify your PI Web API credentials
username = "your_username"
password = "your_password"

# Make the request to fetch data
response = requests.get(
    f"{pi_web_api_url}{data_endpoint}",
    auth=(username, password)
)

# Check if the request was successful
if response.status_code == 200:
    data = response.json()
    print(data)
else:
    print(f"Failed to fetch data. Status code: {response.status_code}")
```