

HYDRA Energy Intelligence API - Quick Start Guide

Get Started in 3 Steps

Step 1: Install .NET 9 SDK (if not installed)

Linux/macOS:

```
wget https://dot.net/v1/dotnet-install.sh
chmod +x dotnet-install.sh
./dotnet-install.sh --channel 9.0
export PATH="$PATH:$HOME/.dotnet"
```

Windows:

Download from: <https://dotnet.microsoft.com/download/dotnet/9.0>

Step 2: Start the API

```
cd Backend
chmod +x start-api.sh
./start-api.sh
```

Or manually:

```
cd Backend
dotnet restore
dotnet build
dotnet run
```

Step 3: Test the API

Open your browser to: **<http://localhost:5000/swagger>**

Or run automated tests:

```
./test-api.sh
```

Available Endpoints

Health Check

```
GET http://localhost:5000/health
```

Authentication

```
POST http://localhost:5000/api/auth/token
GET http://localhost:5000/api/auth/validate
```

Energy Data

```
GET http://localhost:5000/api/energy/data?fromDate=2025-03-01&toDate=2025-03-31
GET http://localhost:5000/api/energy/total?fromDate=2025-03-01&toDate=2025-03-31
GET http://localhost:5000/api/energy/average?fromDate=2025-03-01&toDate=2025-03-31
```

Weather Data

```
GET http://localhost:5000/api/weather/data?fromDate=2025-03-01&toDate=2025-03-31
GET http://localhost:5000/api/weather/date/2025-03-01
```

Analytics

```
GET http://localhost:5000/api/analytics/summary?fromDate=2025-03-01&toDate=2025-03-31
GET http://localhost:5000/api/analytics/anomalies?fromDate=2025-03-01&toDate=2025-03-31
```

Forecasting

```
GET http://localhost:5000/api/forecast?days=3
GET http://localhost:5000/api/forecast/predictions?days=3
```

Insights

```
GET http://localhost:5000/api/insights?fromDate=2025-03-01&toDate=2025-03-31
GET http://localhost:5000/api/insights/type/WeatherImpact?fromDate=2025-03-01&toDate=2025-03-31
GET http://localhost:5000/api/insights/severity/warning?fromDate=2025-03-01&toDate=2025-03-31
```



Configuration

OpenWeatherMap API Key (Optional)

Edit `appsettings.json` :

```
{
  "WeatherSettings": {
    "ApiKey": "YOUR_API_KEY_HERE"
  }
}
```

Note: Without an API key, the system uses simulated weather data.

CORS Settings

To add more allowed origins, edit `appsettings.json` :

```
{
  "Cors": {
    "AllowedOrigins": [
      "http://localhost:3000",
      "http://localhost:5173",
      "http://localhost:8080",
      "http://your-frontend-url.com"
    ]
  }
}
```



Example Usage with curl

Get Energy Data

```
curl -X GET "http://localhost:5000/api/energy/data?fromDate=2025-03-01&toDate=2025-03-31" -H "accept: application/json"
```

Get Analytics Summary

```
curl -X GET "http://localhost:5000/api/analytics/summary?fromDate=2025-03-01&toDate=2025-03-31" -H "accept: application/json"
```

Get Forecast

```
curl -X GET "http://localhost:5000/api/forecast?days=3" -H "accept: application/json"
```

Get Insights

```
curl -X GET "http://localhost:5000/api/insights?fromDate=2025-03-01&toDate=2025-03-31" -H "accept: application/json"
```



Troubleshooting

Port 5000 already in use

Edit `Properties/launchSettings.json` and change the port:

```
"applicationUrl": "http://localhost:5005"
```

Build errors

```
dotnet clean
dotnet restore
dotnet build
```

Cannot connect to HYDRA API

- Check your internet connection
- Verify HYDRA API is accessible
- Credentials are pre-configured in `appsettings.json`



Documentation

- **Full README:** [README.md](#) (README.md)
- **Project Summary:** [PROJECT_SUMMARY.md](#) (PROJECT_SUMMARY.md)
- **Interactive API Docs:** <http://localhost:5000/swagger>



What's Pre-Configured

- ✓ HYDRA OAuth2 credentials
- ✓ HYDRA API endpoints
- ✓ Device and Sensor IDs
- ✓ CORS for common frontend ports
- ✓ Comprehensive error handling
- ✓ Logging configuration
- ✓ Swagger/OpenAPI documentation



Tips

1. **Swagger UI** is the best way to explore and test the API
2. All endpoints return **JSON responses** in a consistent format
3. **Date parameters** use format: `yyyy-MM-dd` (e.g., 2025-03-01)
4. **Weather data** will use simulated values if OpenWeatherMap API key is not set
5. Use **Health Check** endpoint to verify API is running



Next Steps

1. Start the API
2. Test endpoints via Swagger UI
3. Integrate with your frontend (Vue 3/React/Angular)
4. Deploy to Azure/AWS/Docker



Support

For questions or issues:

- Check the full [README.md](#) (README.md)
 - Review [PROJECT_SUMMARY.md](#) (PROJECT_SUMMARY.md)
 - Contact: priaash@awarenesscompany.co.za
-

Happy Coding! 🚀