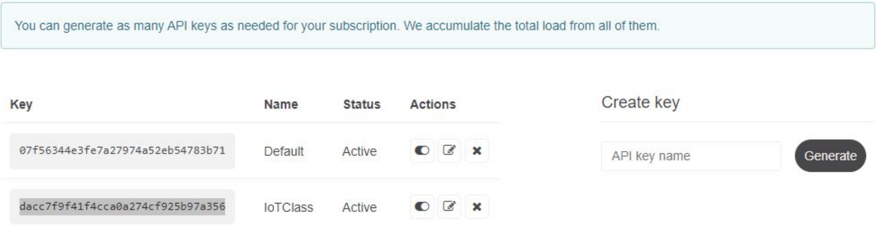
Webhooks

Step 1 - OpenWeather

- Create an account at `openweathermap.org`
- Generate an API key at: `https://home.openweathermap.org/api_keys`



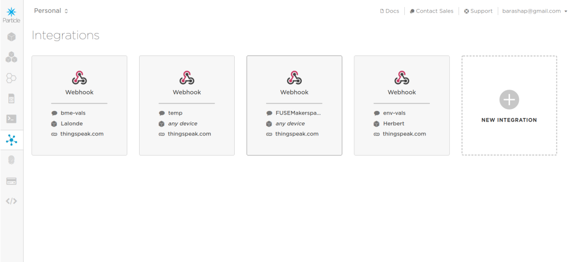
You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

Key	Name	Status	Actions
07f56344e3fe7a27974a52eb54783b71	Default	Active	
dacc7f9f41f4cca0a274cf925b97a356	IoTClass	Active	

Create key

Step 2 - Create Webhook

From `console.particle.io`:



Integrations

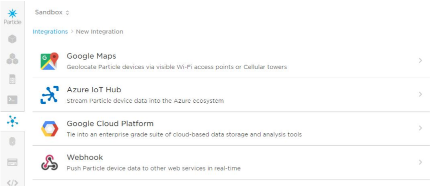
Webhook
- bme-vals
- Lalonde
- thingspeak.com

Webhook
- temp
- *any device*
- thingspeak.com

Webhook
- FUSEMakerspa...
- *any device*
- thingspeak.com

Webhook
- env-vals
- Herbert
- thingspeak.com

NEW INTEGRATION



Integrations > New Integration

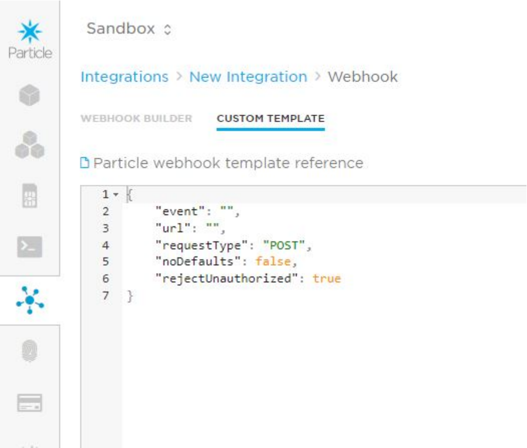
Google Maps
Geolocate Particle devices via visible Wi-Fi access points or Cellular towers

Azure IoT Hub
Stream Particle device data into the Azure ecosystem

Google Cloud Platform
Tie into an enterprise grade suite of cloud-based data storage and analysis tools

Webhook
Push Particle device data to other web services in real-time

Step 3 - Select Custom Template


Particle
Sandbox
Integrations > New Integration > Webhook
WEBHOOK BUILDER
CUSTOM TEMPLATE
Particle webhook template reference

```
{
        "event": "",
        "url": "",
        "requestType": "POST",
        "noDefaults": false,
        "rejectUnauthorized": true
}
```

Step 4 - Update Custom Template with API format

Adapted from `https://openweathermap.org/api/one-call-api` and `https://openweathermap.org/current`

```
{
    "event": "GetWeatherData",
    "responseTopic": "{{PARTICLE_DEVICE_ID}}/{{PARTICLE_EVENT_NAME}}",
    "url": "https://api.openweathermap.org/data/2.5/onecall",
    "requestType": "GET",
    "noDefaults": true,
    "rejectUnauthorized": true,
    "responseTemplate": "{\"lat\":{{lat}},\"lon\":{{lon}},\"dt\":{{current.dt}},\"temp\":{{current.temp}},\"uvi\":{{current.uvi}},\"clouds\":{{current.clouds}},\"ws\":{{current.wind_speed}},\"wd\":{{current.wind_deg}} }",
    "query": {
        "lat": "{{lat}}",
        "lon": "{{lon}}",
        "exclude": "minutely,hourly,daily,alerts",
        "units": "metric",
        "appid": "dacc7f9f41f4cca0a274cf925b97a356"
    }
}
```

Step 5 - Particle Argon Code

```
const char *EVENT_NAME = "GetWeatherData";
unsigned int lastTime;
const float lat=35.0045, lon=-106.6465;  //update to your favorite location

void setup() {
    Serial.begin(9600);
    waitFor(Serial.isConnected,15000);
    String subscriptionName = String::format("%s/%s/", System.deviceID().c_str(), EVENT_NAME);
    Particle.subscribe(subscriptionName, subscriptionHandler, MY_DEVICES);
    Serial.printf("Subscribing to %s\n", subscriptionName.c_str());
}

void loop() {
    if((millis() - lastTime) > 60000) {
        Serial.printf("\n\nTime = %i\n",millis());
        Particle.publish(EVENT_NAME, "", PRIVATE);
        Particle.publish(EVENT_NAME, String::format("{\"lat\":%0.5f,\"lon\":%0.5f}", lat, lon), PRIVATE);
        lastTime = millis();
    }
}

void subscriptionHandler(const char *event, const char *data) {
    JSONValue outerObj = JSONValue::parseCopy(data);
    JSONObjectIterator iter(outerObj);
    while(iter.next()) {
            Serial.printf("key=%s value=%s\n",(const char *) iter.name(),(const char *) iter.value().toString());
    }
}
```