

[App Store Connect...](#) / [Power and Performance Metrics and Logs](#) / Retrieve Power and Performance Metrics and Log Insights

Sample Code

# Retrieve Power and Performance Metrics and Log Insights

Use the App Store Connect API to collect and parse diagnostic logs and metrics for your apps.

Download

## Overview

The sample code includes a number of scripts that demonstrate the following:

- Creating a JSON Web Token for accessing the API.
- Retrieving an App's aggregated power and performance metrics.
- Retrieving an App's diagnostic logs and corresponding insights.
- Parsing JSON files retrieved from the API that contain metrics or logs.

Allow a few days after releasing your app for Apple to collect and organize logs into reports. The system behind the API requires significant usage of your app before it makes metrics available, and each metric has different usage thresholds.

## Configure the Project

Install the Python libraries `pyjwt` and `docopt` to run the sample code. Use Terminal to run the following command to install the libraries locally:

```
pip3 install pyjwt[crypto] docopt --user
```

# Create a JSON Web Token to Access the App Store Connect API

To use the API, you need to generate and provide a JSON Web Token (JWT), signed using an App Store Connect API key. First, generate and retrieve a private key as described in [Creating API Keys for App Store Connect API](#). While retrieving your API key, take note of the issuer ID and the key ID, generating a JWT requires them. After retrieving your API key, use the `generate-token.py` script to generate a JWT to authenticate requests you send to the App Store Connect API.

To generate a JWT, run the `generate-token.py` script, providing the issuer ID, key ID, and the path to the private key.

```
python3 generate-token.py {issuer id} {key id} {private key path}
```

For more information about generating tokens, see [Generating Tokens for API requests](#).

## Identify Your App ID and Build ID

To retrieve power and performance metrics for your app, you need to know its App ID. Use the [List Apps](#) endpoint to get a list of your apps and metadata about them, including App IDs.

Create a JWT, then use that token with `curl` to request a list of your apps:

```
TOKEN=$(python3 generate-token.py $ISSUER_ID $KEY_ID $KEY_FILE_PATH)
curl -H "Authorization: Bearer ${TOKEN}" "https://api.appstoreconnect.apple.com/v1/apps"
```

Retrieving diagnostic logs, or build-specific metrics, requires a build ID. Use [List All Builds of an App](#) to get a list of the builds.

Use the following `curl` command to request a list of builds for an App:

```
APP_ID=YourAppId
curl -H "Authorization: Bearer ${TOKEN}" "https://api.appstoreconnect.apple.com/v1/builds/${APP_ID}"
```

## Retrieve Metric Insights for Your App

Use the `get-metrics-insights.py` script to download power and performance metrics, and to print metric hotspot data identified by insights:

```
python3 get-metrics-insights.py $APP_ID
```

# Retrieve Diagnostic Logs for a Build of Your App

Use the `get-diagnostics-logs.py` script to download disk writes, diagnostic logs, and print the callbacks resulting from the top disk write I/O exception:

```
python3 get-diagnostics-logs.py $BUILD_ID
```

## Parse JSON Files Retrieved from App Store Connect

The code in `ppparser` provides an example of how to parse the JSON performance metrics and diagnostic log data retrieved from App Store Connect API. With JSON files stored locally, use the following command to invoke the parser directly:

```
python3 ppparser -h
```

PowerPerformanceParser (PPParser)  
Parse metrics & diagnostics JSON responses.

Usage:

```
ppparser metrics --input=INPUT_FILE [--metric=METRIC_NAME] [--auto_insights]
ppparser diagnostics --input=INPUT_FILE [--limit=LIMIT] [--callstack]
ppparser -v
ppparser -h
```

metrics	Parse metrics report JSON from /v1/apps/{id}/perfPowerMetrics
diagnostics	Parse diagnostics report JSON from /v1/diagnosticSignatures/{id}/logs
-i INPUT_FILE, --input INPUT_FILE	Input JSON file
-m METRIC_NAME, --metric METRIC_NAME	Filter by metric
-d DEVICE_TYPE, --device DEVICE_TYPE	Filter by device
-p PLATFORM, --platform PLATFORM	Filter by platform
-l LIMIT, --limit LIMIT	Limit the number of diagnostics logs to print
-a, --auto_insights	Pretty-print auto-generated insights
-c, --callstack	Pretty-print callstack
-v, --version	Show version
-h, --help	Show this help screen

## See Also

## Getting Metrics and Diagnostic Logs

### Get Power and Performance Metrics for an App

Get the performance and power metrics data for the most recent version of an app.

### Get Power and Performance Metrics for a Build

Get the performance and power metrics data for a specific build.

### List All Diagnostic Signatures for a Build

List the aggregate backtrace signatures captured for a specific build.

### Download Logs for a Diagnostic Signature

Get the anonymized backtrace logs associated with a specific diagnostic signature.