Powerwall Service

A windows service that can monitor, optionally log (to an SQL Server database), control, and optionally report (to PVOutput) data from Tesla Powerwall 2

Tested with firmware version 1.12.0 to 1.15.3

Audience

The service is designed for users who have all the following:

- A Powerwall 2 with the gateway connected to a local network
- A PV installation
- An always on windows PC (or a cloud hosted IAAS windows instance with connectivity to the Powerwall via a VPN) on the same network as the Powerwall gateway
- A tariff that includes peak and off-peak prices (that are significantly different given round trip losses) to allow for pre-charging the Powerwall to assist in load shifting the next day's peak load into the early morning (overnight) off-peak period

It may be able to work for users who do not have PV (but the author is not prepared to test by disconnecting his own PV) if it is possible to switch the Powerwall from self-consumption to backup mode.

It is of little to use to users with a flat (single priced) tariff, as the key feature is time shifting, though the secondary features (data logging, output to PVOutput) may be of interest.

Secondary features include:

- Six second logging of key Powerwall information
- Uploading Powerwall data to PVOutput
- Streaming Data to PowerBI (via the currently in beta streaming API service)

The logging data and uploading to PVOutput require an SQL Server database (local or Azure)

Overview

The service consists of three main components and relies on a minimum of two web services that require authentication (the local Powerwall web-based API, and solcast.com.au's PV forecast API). With this minimum, it can monitor the state of charge of a local Powerwall and, based on parameters set in the service's .config file and forecast power generation the next day, pre-charge the Powerwall when the expected solar generation will be insufficient to charge the Powerwall and supply the usual peak load.

The additional logging and PVOutput features require two additional web services – a sunrise-sunset service and an account on PVOutput, and require access to an SQL Server database.

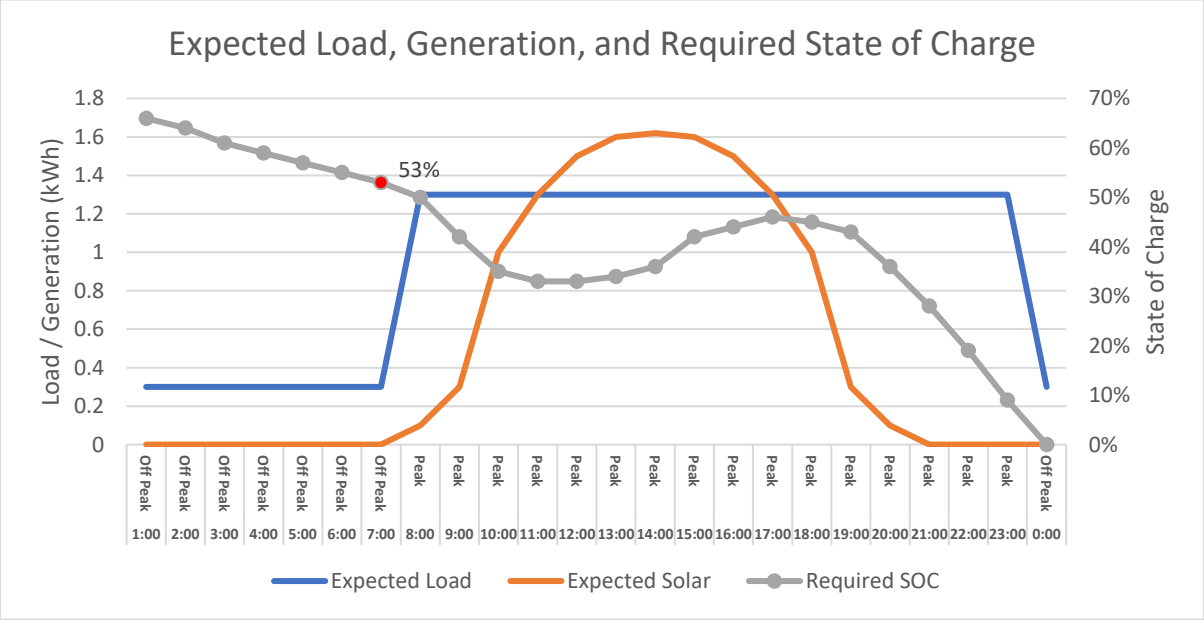
The Streaming Data feature requires you to have a Power BI login (can be free), and, as the current Streaming API is in beta and it is not possible to set up a streaming dataset by importing a pbix file, you will need to set up the streaming dataset manually.

Time Shifting (Pre-Charging) Logic

The key design objective is to predict when the next day's estimated solar generation will be insufficient to meet the next day's estimated on-peak (during peak hours according to the user's electricity tariff) consumption.

If this is predicted, then a further estimate of the expected remaining Powerwall charge at the beginning of the next day's peak period is made, and if required, the Powerwall is pre-charged.

Below is a simple chart showing the required state of charge hour by hour, given the expected consumption hour by hour, the expected solar hour by hour, with the aim being to have sufficient charge in the Powerwall to get through to off peak at the end of the day.



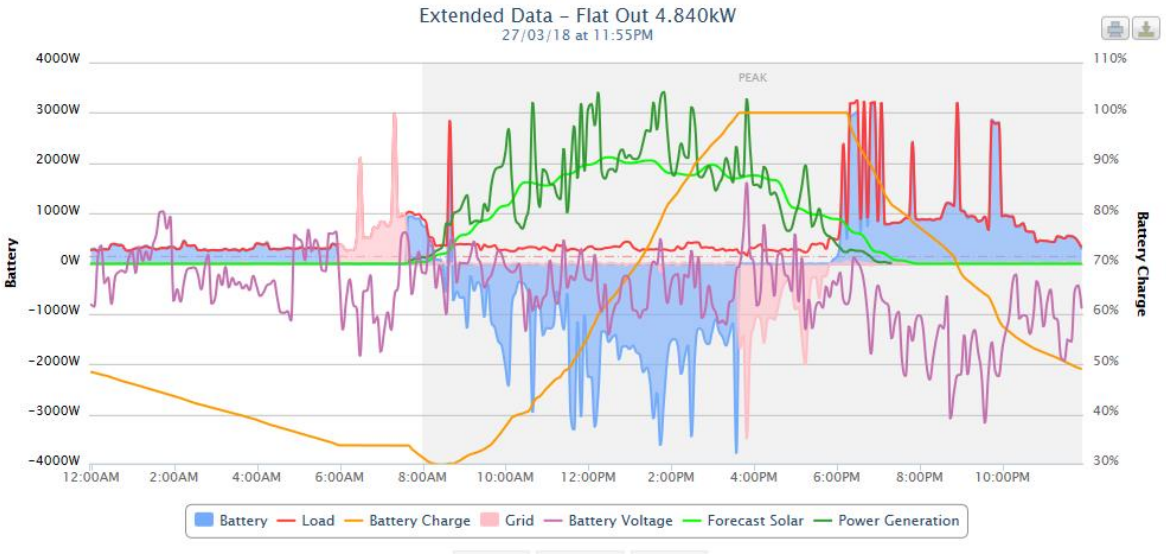
As can be seen above, at 7am, the required state of charge is 53% (red marker), so that the solar generation during the day is sufficient to charge the Powerwall to meet the load expectations through to the end of peak at 11pm.

If the Powerwall Service calculates that the expected generation is not sufficient to achieve this outcome (e.g. there is a shortfall of solar generation compared to the expected load, combined with the current state of charge), it decides to charge (or, at least, stop discharging) the Powerwall so that it has sufficient charge at the start of the peak period to see it through to the end.

Key config settings that control this process are PWOvernightLoad, PWMorningBuffer, PWMinBackupPercentage and PWPeakConsumption.

$$RequiredSOC = Morning\ Buffer + \frac{Overnight\ Charge\ Use * Remaining\ Overnight\ Hours}{Total\ Overnight\ Hours} + \frac{Shortfall}{Capacity}$$

This is best illustrated by an example based on actual data – using the graph of the activity from PVOutput, combined by the log entries from the service when it made the charge / discharge decisions late in March (mid-autumn in Melbourne):



Feel free to have a look at what my system is doing at https://pvoutput.org/intraday.jsp?sid=6943

Date and Time	Service Logging	Explanation
27/03/2018 6:00	Three Day Forecast Date: 27/03/2018 00:00:00 Total: 14673.88 Morning: 704.9293 Date: 28/03/2018 00:00:00 Total: 19954.88 Morning: 987.8926 Date: 29/03/2018 00:00:00 Total: 19234.9 Morning: 831.8743	The service regularly updates its forecasts from the solcast forecast api. The next three days are displayed, but only the next (current) day is used at the moment.
27/03/2018 6:00	In Operation Period: SOC=33.88394446422215, Required=14, Shortfall=2826.125, NewTarget=34.93426	In Operation period indicates that the service knows it is in the "off peak" period when it can charge. Its logs the current State of Charge (SOC) as 33% of capacity, required (to see through expected off-peak load) as 14% (based on the time of the day and progress through the off-peak period, the shortfall between solar and expected peak load is 2826 Wh (2.8kWh), and a new target State of Charge (SOC) as 34.9%
27/03/2018 6:00	Current SOC below required setting: SOC=33.88394446422215, Required=14, Shortfall=2826.125, NewTarget=34.93426	Reports that the current SOC is below the target from within the charge setting routine.
27/03/2018 6:00	Entered Charge Mode: SOC=33.88394446422215, Required=14, Shortfall=2826.125, NewTarget=34.93426, Mode=self_consumption, BackupPercentage=35, APIResult = 202	Reports that the service has set the Powerwall to charge (or not self-consume) by setting the backup percentage to 35%. If the backup percentage is close to the current SOC, the Powerwall stops discharging, but doesn't always re-charge to the specified backup percentage if it is close. **You can see this in the PVOutput chart above at 6am when the blue battery area on the**

		chart drops to zero as the battery stops discharging.
27/03/2018 7:40	Reached end of pre-charging period: SOC=33.82698469206123, Required=0, Shortfall=0, NewTarget=0	We've reached the end of the off-peak period - the system will not continue to charge during peak hours
27/03/2018 7:40	Exiting Charge Mode: SOC=33.82698469206123, Required=10, Shortfall=2875.614, NewTarget=31.30085	Reports that the service is about to exit charge mode.
27/03/2018 7:40	Exited Charge Mode: SOC=33.82698469206123, Required=10, Shortfall=2875.614, NewTarget=31.30085, Mode=self_consumption, BackupPercentage=5, APIResult = 202	Reports that the service has set the Powerwall to back to self-consumption by setting the backup percentage to 5%. **You can see this in the PVOutput chart above at 7:40am when the blue battery area on the chart matches load as the battery starts discharging.**
27/03/2018 8:00	Outside Operation Period: SOC=31.541473834104668	Now outside off-peak period. Just reports SOC

Pre-Charging and Standby

Without Tesla's Time Based Control (TBC) – the Powerwall will typically charge on PV, and discharge to the reserve power percentage every day.

This means that it will discharge during off-peak tariff periods. There are two scenarios where you don't want this to happen:

1. You don't have enough stored power (and expected PV generation) to see you through the next peak period, or
2. You actually get paid more to feed in solar than you pay for off-peak from the grid.

In both of these scenarios, you don't want to discharge the Powerwall into off peak load, only to then either have to charge it from off peak (to ensure you can not draw from the grid during peak – as round trip losses mean you're paying around 15% more than you need to), or, instead of feeding into the grid at a higher rate, you're recharging the Powerwall energy that was used to offset low cost off-peak grid usage.

The various standby settings are designed to avoid these losses (either round-trip losses or losses where feed in tariffs pay you more than you pay for off-peak grid paower).

The second scenario (you get paid more for feed in, than you pay for off peak grid power) is not currently supported by Telsa's TBC (as there is no capability to enter rates into the Tesla app).

Note: *although there are rate settings in the Powerwall Service's .config file, these are not yet used to make intelligent decisions – charge / discharge / standby logic is controlled simply from the settings that enable standby mode and set peak and off-peak periods.*

Prerequisites

A Windows always on machine

The Powerwall Service is implemented as a windows service and developed on the Microsoft .Net Framework. It requires version 4.6 or above. You can download version 4.6 of the .Net Framework from https://www.microsoft.com/net/download/dotnet-framework-runtime/net46 or the latest version of the .Net Framework from https://www.microsoft.com/net/download/Windows/run.

A Powerwall 2

I guess that is sort of obvious, but if you've got a Powerwall 1, sorry, I can't help you.

A local wired LAN or WiFi connection to the Powerwall

The Powerwall must be on a fixed IP address or resolve using a DNS name (i.e. you have a DNS/DHCP server in your router that can assign a resolvable name and serve an IP address for the name)

Powerwall Settings in PowerwallService.config

Key settings are the IP address (or DNS name if you've set one up) and the password which is the gateway serial number prefixed with the letter "S" – no quotes.

An API account at solcast.com.au

You can sign up (for free – while in beta) at https://www.solcast.com.au/api/. You will need the API key you are provided once you've signed up to allow the Powerwall Service to get forecast solar generation for your solar installation.

Solcast Settings in PowerwallService.config

There are several settings that control the solcast API – the API key, as well as the PV system's latitude, longitude, capacity, install date, tilt and azimuth. These are the same settings you've probably already set up in PVOutput.

Prerequisites for Optional Features

SQL Server

The logging features (and the PVOutput features – that depend on logging) require an SQL Server database available to the service. This can either be local (for example, SQL Server express edition – available for free from https://www.microsoft.com/en-au/sql-server/sql-server-downloads) or an azure SQL database.

PVOutput Donation Account

If you want to upload data about the Powerwall's operation, such as flow into and out of the battery state of charge, etc., you will need to enable donation mode on PVOutput by donating.

PVOutput settings in PowerwallService.config

There are lots of PVOutput settings in the config file – key are the API key and System ID (SID), and flags to control if the service should upload generation, load, forecasts, and Powerwall extended data. If you already have an uploader uploading data, you might just want to start with PVSendPowerwall set to true, but have PVSendLoad, PVSendPV and PVSendVoltage set to false. Note that PVSendPowerwall also sends forecast data.

Setup instructions

1) Unzip into a directory (e.g. C:\Tools\PowerwallService)
2) Unblock exes and DLLs if they show the unblock button in properties
3) Open an administrator command prompt and navigate to that director
4) Run the following command in the directory where you unzipped to install the service

```
installutil PowerwallService.exe
```

5) In the same directory, edit PowerwallService.exe.config in your favourite XML or text editor (an XML editor is good, because it will highlight errors if you accidently corrupt the XML)
6) Review the comments on each parameter
7) Ignore the parameters that are NOT CURRENTLY USED
8) Make sure you find and replace (or delete, after reviewing) every TODO in the file
9) Note that there are TODO comments in comments, and TODO tags in values - you must replace or remove these
10) Don't attempt to set up a database for logging yet - let's just get the service running.
11) From the administrator command prompt you opened earlier, once you've edited the config file

```
NET START PowerwallService
```

With any luck, the service will start successfully.

Don’t forget to set the service to start automatically in service manager and set the recovery options to restart the service on failure for all three options. This will ensure the service keeps running.

```
The Powerwall Logging and Control Service service is starting.
The Powerwall Logging and Control Service service was started successfully.
```

If so, hop into event viewer, windows logs, application log, and look for at least three entries from PowerwallService as it starts:

```
EventID: 100      Powerwall Service Starting
EventID: 101      Powerwall Service Started
EventID: 0  Service started successfully
```

You'll then see a "running" event (Source=PowerwallService, EventID = 200) every minute.

You will also see:

```
EventID 108:      Observation Timer Started
EventID 109:      One Minute Timer Started
EventID 110:      PVOutput Timer Started (only if you've enabled sending to PVOutput)
EventID 111:      Solar Forecast and Charge Monitoring Timer Started
```

You should also see EventID 1000 which will show you've got a solar forecast back from the API.

All entries (and errors) will be logged under source Powerwall Service. Key log entries to look out for are Events 500 to 520 - this is where the service is logging its thoughts about pre-charging and actions when it takes them – and 1000 – where it logs the next three day's forecast (including today) at least every hour. You should see EventID 500 or 503 every 10 minutes where it reports your powerwall SOC and whether it is in the operation period (e.g. off peak) or not. There are two custom views for the event viewer that show Key (charge / forecast) and All events (right click on Custom Views in event viewer and select import Custom View to import them).

If you stay up until the start of off peak (or set the clock forward on your PC / server) you will see it report when it is in the operation period.

SQL Server Setup

There are two setup scripts provided – one for an Azure database, and one for SQL Server 2016 or above.

If you are creating a database in Azure, you'll likely have to create the database either through the portal or using PowerShell, rather than being able to execute the create portion of the script. Look for the TODO in the script to locate the section to be executed in the database once created.

You will need to define a login and user and password to access the database – it is recommended that you use and SQL login rather than a windows login (unless you're running in a domain and are able to set up the service to run as a domain account or are comfortable setting up machine accounts as logins in SQL Server).

For example, for a local SQL Server, you can execute the sample script (please change the random password, though) to create a login, a user, and grant rights to the logging database.

```
USE [master]
GO

/* For security reasons the login is created with a random password. */
/****** Object:  Login [PWMonitor]    Script Date: 1/04/2018 6:37:43 PM ******/
CREATE LOGIN [PWMonitor] WITH
PASSWORD=N'LkXiEaDFQuIIQYFDC0GYweH3+orYxqI6qc5Ax2kCgJ0=',
DEFAULT_DATABASE=[master], DEFAULT_LANGUAGE=[us_english], CHECK_EXPIRATION=OFF,
CHECK_POLICY=ON
GO

USE [PWHistory]
GO

/****** Object:  User [PWMonitor]    Script Date: 1/04/2018 6:37:57 PM ******/
CREATE USER [PWMonitor] FOR LOGIN [PWMonitor] WITH DEFAULT_SCHEMA=[dbo]
GO
```

You can then set up this information in the connection string setting in the PowerwallService.config file and enable logging by setting the LogData setting to True.

Updating SQL Server if you have already set up a server.

At this stage, alter scripts are not provided to account for changes from the originally provided script, nor is there an automated update process. Release notes will point out any key SQL Server changes.

Config Settings

There are comments in the config file for all the settings – these (together with the pre-filled values) should give you enough information to know how to set up the service. The config file is a simple XML file – provided you don't accidently change or delete the tags and structure, it should be straightforward to edit values to your required settings.

Each config setting is in the following form:

```
<setting name="PVSendPowerwall" serializeAs="String">
  <!--Used if you want to send logged powerwall data to PVOutput-->
  <value>True</value>
</setting>
```

An empty value – for example, to not use a particular PVOutput extended parameter – can be specified as either `<value></value>` or `<value />` – for example:

```
<setting name="PVv12" serializeAs="String">
  <!--The data to send to extended parameter 12 -->
  <value />
</setting>
```

Comments in XML start with `<!--` and end with `-->` (and can span multiple lines – everything in between is ignored.

PowerBI Streaming Set Up

Note: *PowerBI Streaming is currently in beta – it is expected that there will be changes at a later time as this service moves out of beta. It is possible that it won't be available for PowerBI free subscriptions, but is currently available.*

Note: *It is not possible to export definitions of a streaming dataset nor a dashboard in a convenient way for another user to import. Whilst it is probably possible to programmatically define both streaming data sets and dashboards, given that streaming data sets and dashboards are both in beta, documentation on how to set these up programmatically is not yet available. As a result, setting up both streaming data sets and dashboards is a manual process (but both are quick and easy to do).*

The Powerwall Service can write to two PowerBI streaming datasets. If the API endpoint is non-blank in the config file, then it will write to that dataset.

Live Logger Dataset

The live logger dataset will receive updated details read from the Powerwall every six seconds, allowing you to track load, solar generation, battery and grid flow, state of charge, etc.

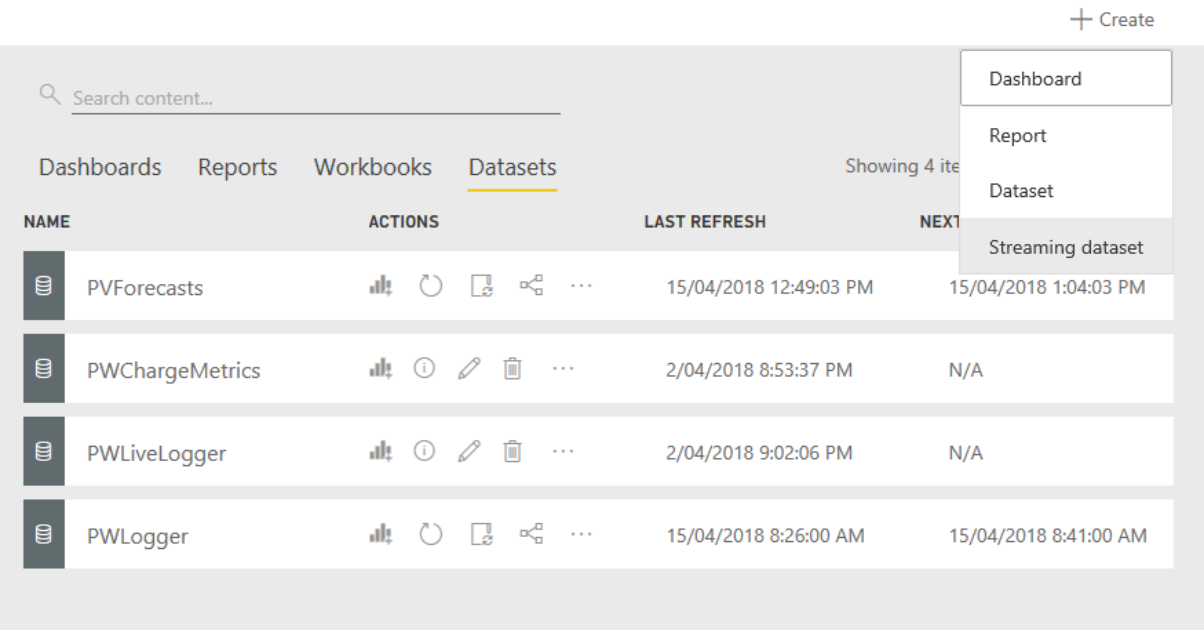
Charge Intent Dataset

The charge intent dataset will receive updated details every ten minutes, providing information about the charge / standby plans / actions of the Powerwall Service, including required charge level needed at end of Off Peak, next day PV forecast, PV generation shortfall, remaining generation.

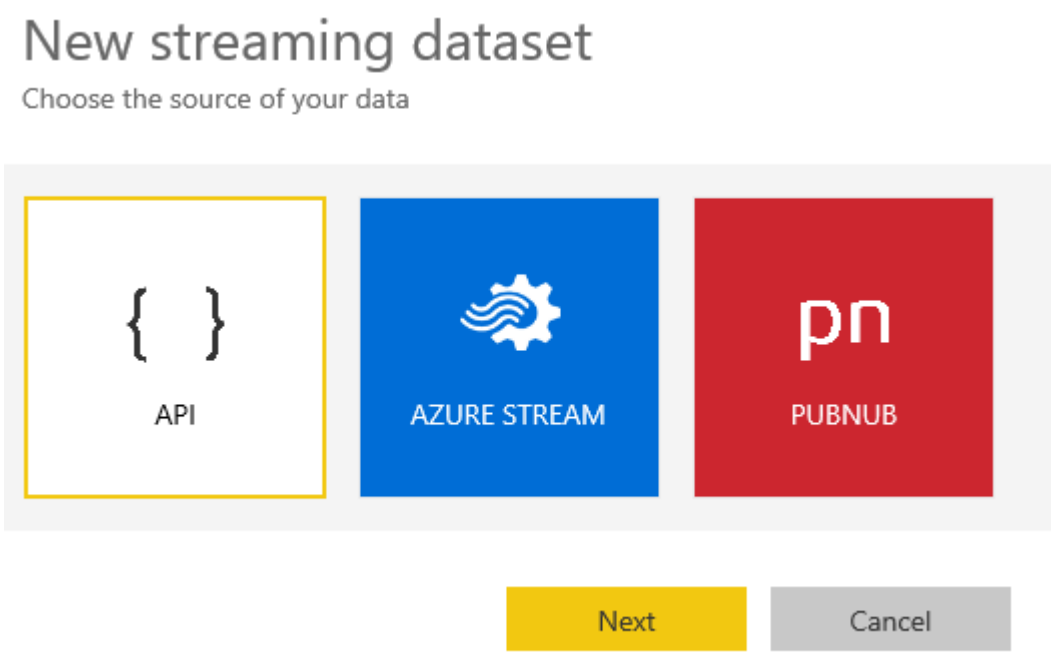
Setting up each Dataset

Log into Power BI on the web, and go to your dashboard, datasets (https://app.powerbi.com/groups/me/contentlist/datasets?onlySharedWithMe=false should work).

Click on the + Create button near the top right, and choose Streaming Dataset



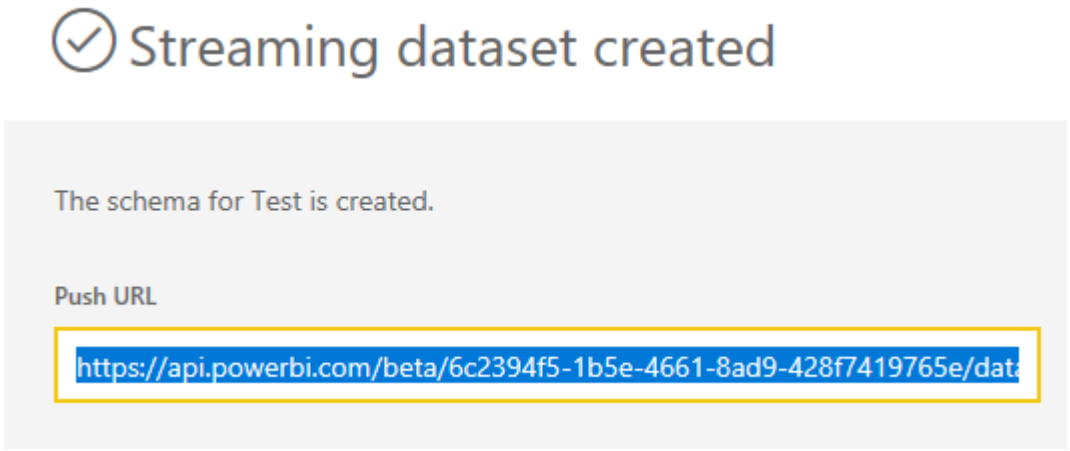
Choose API, and click on Next



Fill in the details as show on the next two pages – the name of the dataset is not critical (but I'd suggest you use LiveLogging and ChargeIntent so you can match them to the .config entries), but the Values from stream section must match the name (including case sensitivity) and type (Text, Number or DateTime) exactly.

Ensure that you have turned on "Historic data analysis" if you want to report on this data as well.

Click on Create and copy the Push URL shown – you will copy this into the .config item for each data set. You simply paste the entire URL in between the `<value></value>` tags.



e.g.

```
<setting name="PBILiveLoggingEndpoint" serializeAs="String">
      <value>https://api.powerbi.com/beta/6c2394f5-1b5e-4661-8ad9-
      428f7419765e/datasets/0e4debd8-4c53-4151-9454-
      bd036c19fe13/rows?key=asdf</value>
</setting>
```

Edit streaming dataset

Create a streaming dataset and integrate our API into your device or application to send data. Learn more about the API.

* Required

Dataset name *

PWChargeMetrics

Values from stream *

CurrentSOC	Number
MorningBuffer	Number
RemainingOffPeak	Number
ForecastGeneration	Number
Shortfall	Number
RequiredSOC	Number
AsAt	DateTime
OperatingIntent	Text
RemainingInsolation	Number
Enter a new value name	Text

```
[
  {
    "CurrentSOC" : 98.6,
    "MorningBuffer" : 98.6,
    "RemainingOffPeak" : 98.6,
    "ForecastGeneration" : 98.6,
    "Shortfall" : 98.6,
    "RequiredSOC" : 98.6,
    "AsAt" : "2018-04-03T10:30:35.513Z",
    "OperatingIntent" : "AAAAA555555",
    "RemainingInsolation" : 98.6
  }
]
```

Historic data analysis

On

Done Cancel

* Required

Dataset name *

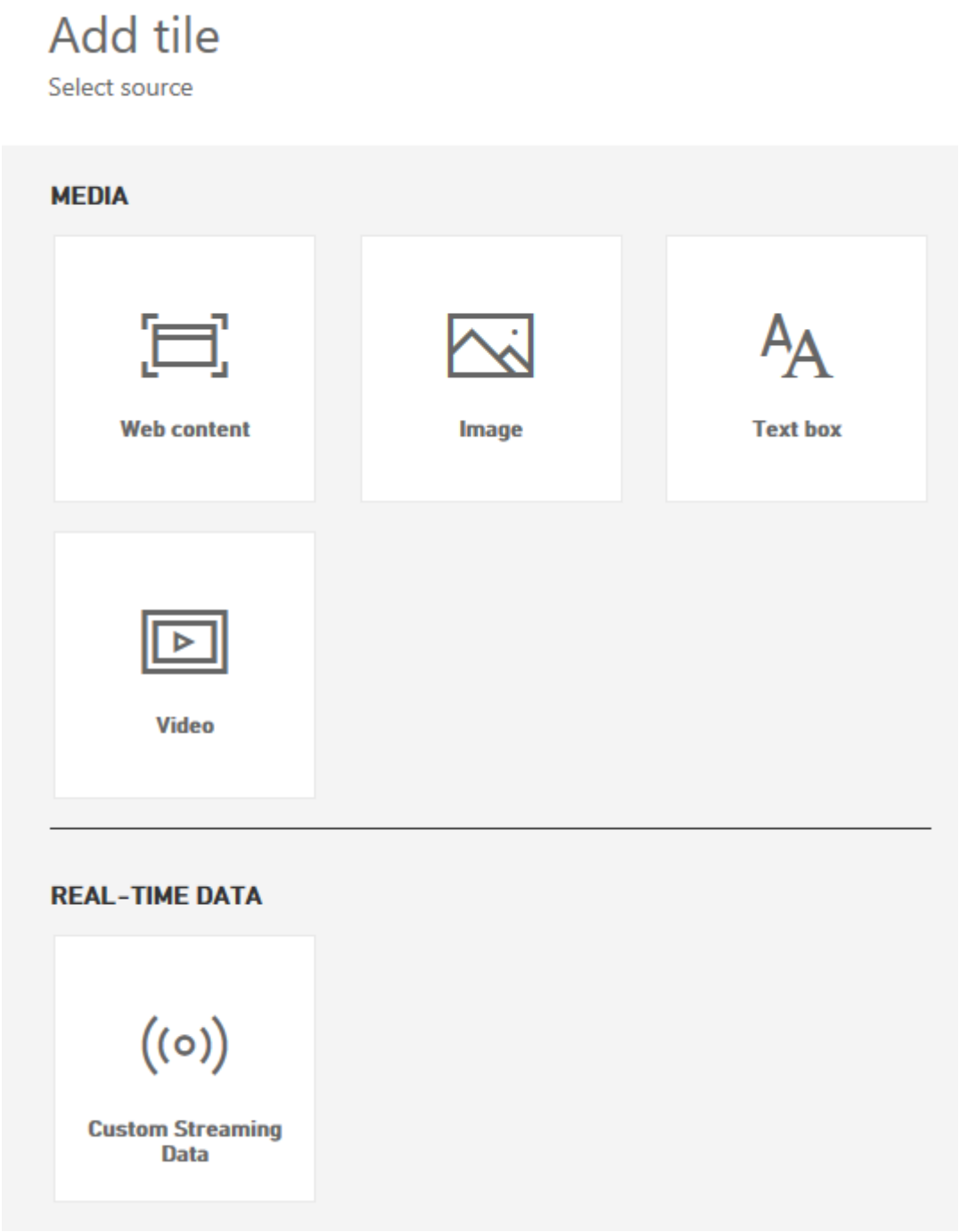
PWLiveLogger

Values from stream *

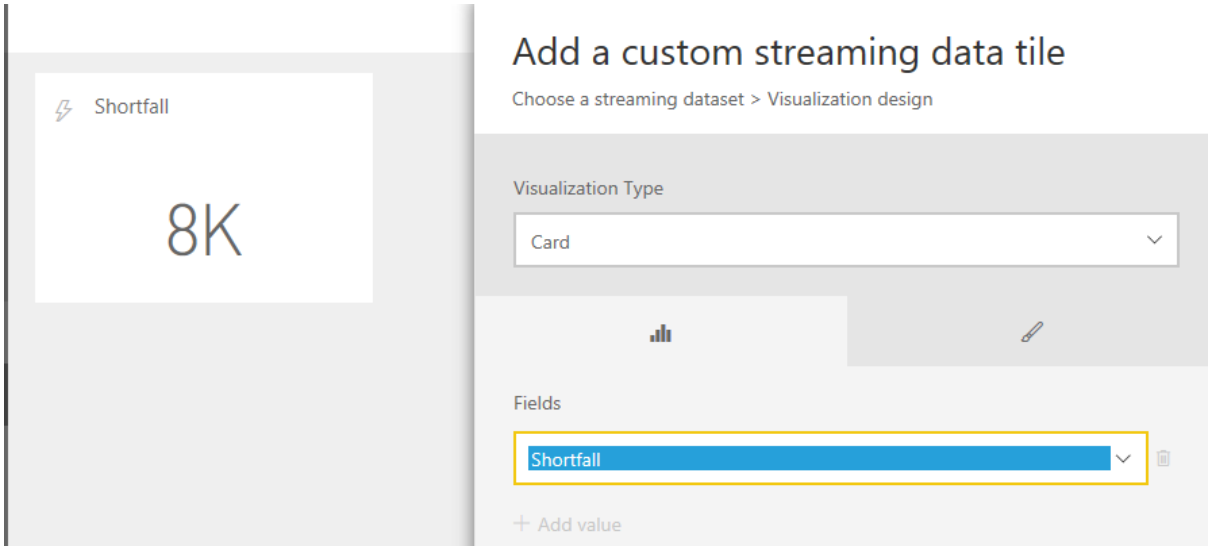
Load	Number
Solar	Number
Grid	Number
Battery	Number
Voltage	Number
SOC	Number
AsAt	DateTime
kWMax	Number
kWMin	Number
SOCMax	Number
SOCMin	Number
VoltageMax	Number
VoltageMin	Number
VoltageTarget	Number
SOCTarget	Number
Enter a new value name	Text

Creating a Dashboard

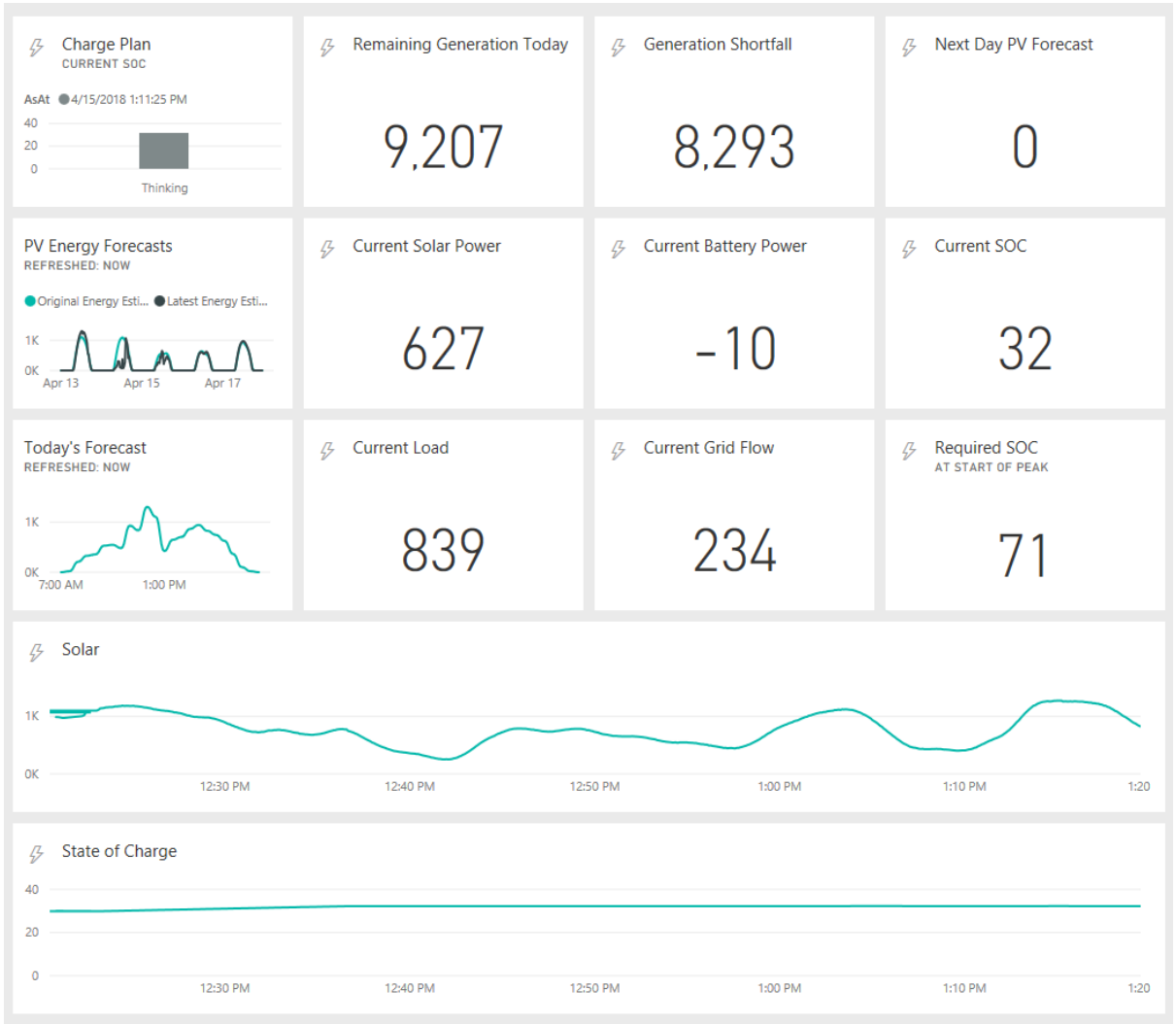
Create a dashboard within PowerBI, and add a tile (click on the plus symbol near the set of icons near the top right) choosing custom streaming data as the source.

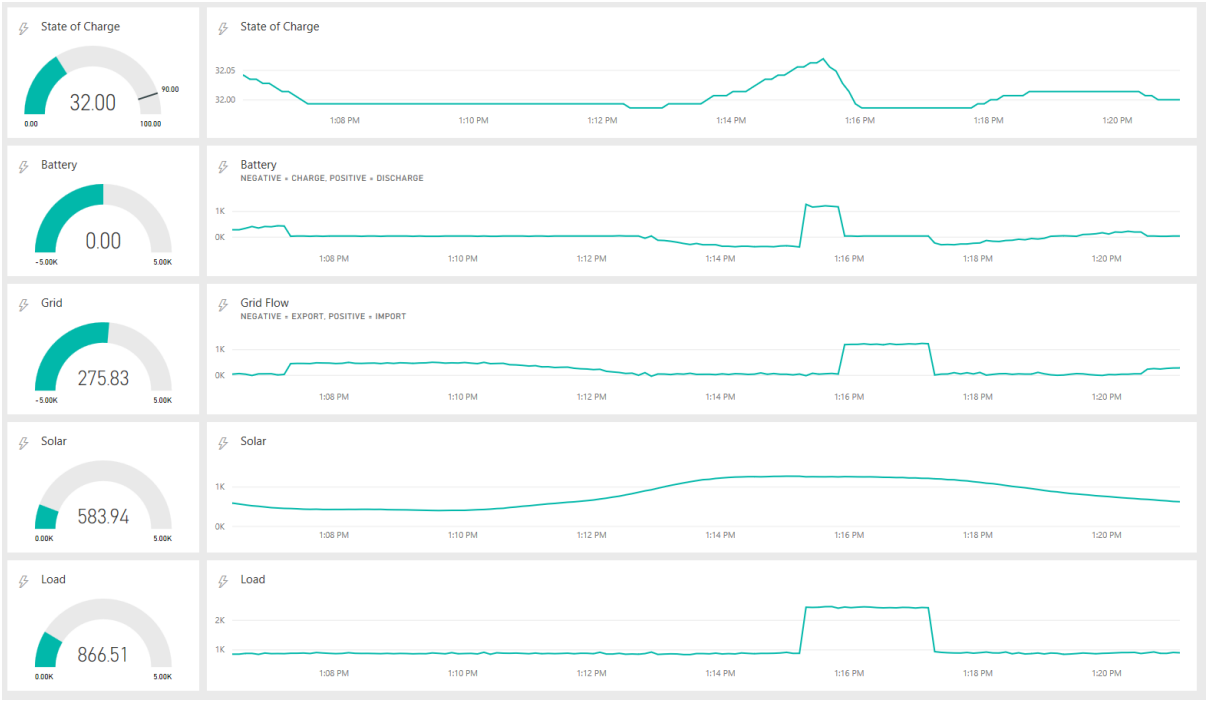


Choose one of the datasets you've created, and choose the visualisation type you'd like, and then add a value from the streaming dataset.



Some sample dashboards are shown below (note that one of them makes use of a report that reports forecast PV data that requires additional set up – beyond the scope of this document.




State of Charge
32.00
90.00
0.00
100.00
State of Charge
32.05
32.00
1:08 PM
1:10 PM
1:12 PM
1:14 PM
1:16 PM
1:18 PM
1:20 PM
Battery
0.00
-5.00K
5.00K
Battery
NEGATIVE = CHARGE, POSITIVE = DISCHARGE
1K
0K
1:08 PM
1:10 PM
1:12 PM
1:14 PM
1:16 PM
1:18 PM
1:20 PM
Grid
275.83
-5.00K
5.00K
Grid Flow
NEGATIVE = EXPORT, POSITIVE = IMPORT
1K
0K
1:08 PM
1:10 PM
1:12 PM
1:14 PM
1:16 PM
1:18 PM
1:20 PM
Solar
583.94
0.00K
5.00K
Solar
1K
0K
1:08 PM
1:10 PM
1:12 PM
1:14 PM
1:16 PM
1:18 PM
1:20 PM
Load
866.51
0.00K
5.00K
Load
2K
1K
1:08 PM
1:10 PM
1:12 PM
1:14 PM
1:16 PM
1:18 PM
1:20 PM