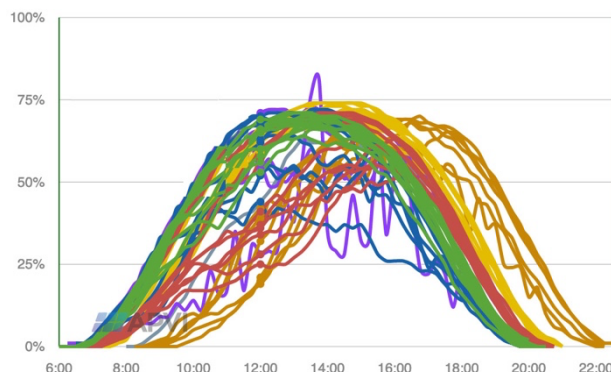


APVI Live Small-scale Solar Generation Data

The APVI PV performance API provides the average performance and output of distributed PV systems at the 2-digit postcode level throughout Australia. If you are an APVI member, please contact [Anna Bruce](mailto:a.bruce@unsw.edu.au) (a.bruce@unsw.edu.au) to get a token that you can use to access the API.



API format:

The **APVI PV performance API** has two formats for live and historical data.

To access the **live** data (the current day until the last available data) use the following API:

```
https://pv-map.apvi.org.au/api/v2/2-digit/today.json?access_token={token}
```

And for **historical** data you can specify the date and use the format below:

```
https://pv-map.apvi.org.au/api/v2/2-digit/{date}.json?access_token={token}
```

The date should be in the format of YYYY-MM-DD (e.g. 2018-07-24)

Response:

The response is a json file with the fields: capacity, performance, output, and load. A sample file can be found [here](#).

Capacity:

This is the total installed capacity of PV systems in each two digit postcode region (all postcodes beginning with the same two digits). This data is sourced from the Clean Energy Regulator and updated each quarter (when new data become available). The capacity excludes PV systems that are registered generators in the NEM as these are accounted for in AEMO's generation data. Linear interpolation is used to calculate a total capacity figure for each 2-digit postcode region. The most recent capacity figures may be an underestimate as PV system owners have up to a year to register their system with the Regulator, although these figures are updated retrospectively as better data for past months becomes available.

Performance:

This is the average performance of all PV systems contributing to the database, for each 2-digit postcode in each 15 min interval. Live output data of more than 20,000 PV systems is sourced and processed to estimate the average PV system performance in each 2-digit postcode region. The performance is reported as a percentage of the rated capacity. For example, 53 means 53% of rated capacity, or 0.53MW per 1MW installed capacity (DC).

The PV performance is kindly contributed by Solar Analytics and PVOutput.org. The data may not be representative of the average PV system output in all parts of Australia. In particular, in some '2-digit postcode regions' a small number of systems are contributing data and larger systems may dominate the aggregate. For Australian users, data is displayed in the user's timezone, so generation from PV from systems east of the users timezone will present with an earlier sunrise, and systems west of the users timezone, a later sunrise. For international users, Australian Eastern Standard Time (AEST) is used for all data, as per the convention in the National Electricity Market. The timestamp format is YYYY-MM-DD HH:mm:ss +ZZZZ, for example, 2018-08-19 07:15:00 +1000. The +ZZZZ represent time difference from GMT, for example +1000 means GMT+10 (Australia/Sydney timezone).

Output:

The estimated total output (MW) of PV systems installed in each 2-digit postcode region in each 15 minute interval. The timestamp format is similar to performance data.

Load:

State level demand data is sourced from the Australian Energy Market Operator (AEMO) for the National Electricity Market and the South West Interconnected System for each 15 min interval in MW.

Work with the API data

To access the API data you can simply type the APIs in your browser and download the json file as text. You can install a "json viewer" extension to your browser (Firefox has a built-in viewer) to view the json file in a more readable format with the ability to interact with the data (e.g. show/hide fields, regions, etc). Or you can access the API using another programming language and process the data. A sample python files has been prepared which can be used to download the data for a range of dates and filter the data by postcode region or by field and also save the data into a CSV file. This python file can be accessed [here](#).