# API overview

The HSE Energy API offers flexible views to monitor devices at each stage of Sundaya’s energy management lifecycle: Harvest, Store, Enjoy (HSE).

Some example scenarios include scheduling of devices in certain modes and preferred times using power profiles to optimise energy efficiency and accommodate the user's preferences. For example, the user might want their washing to be done by 5:00 p.m. with the least electrical power costs or prefer to limit their energy consumption up to a defined limit. The API provides common schemas to implement monitoring and control for such scenarios in applications, through control preferences and policies relating to energy efficiency.

## Data visualisation

The API’s response data can be visualised in an application as a stacked bar graph, based on the following colours for each of the four datasets returned inside the data element.

|  |  |
| --- | --- |
| colour | energy-flow |
|  | harvest |
|  | store |
|  | enjoy |
|  | grid |

Table 1 Energy data colour codes

The bar graph is shown in a ‘double entry’ format (the up and down bars are the same size), as shown in the following sample.



Figure 1 Stacked bar graph format

The example shows the following behaviour:

in the 1st hour all enjoy energy came from the battery;

in the 2nd hour half came from battery (store) and the other half from grid.

in the 3rd hour all came from grid.

in the 4th hour the sun starts delivering (harvest)

in the 10th hour harvest data is more than enjoy and the energy flows into store...

The following graphs shows a month period

It shows energy usage from the grid in the bottom tier, which indicates a need for the user to get more battery capacity. In general, a graph with lot of black in it indicates that you need to do something about it.



Figure 2 Monthly usage example

## API specification

The HSE API is REST based and specified in yaml with openapi2.0/swagger, and available through the Sundaya developer portal at http://developer.sundaya.com.

Response mime types are:

application/json

application/vnd.collection+json

A single request path (hse) will return consolidated data for all four energy lifecycle scopes (Harvest. Store, Enjoy, Grid). However, the request can be parameterised (in the request Body) to filter the returned data if needed by category, subcategory, and product-type: of the harvest, store and enjoy energy asset.

In all requests the api calls will provide a mandatory customer site (siteid) and access key:

api.sundaya.com/hse/{duration}/{ending}?site={siteid}&key={accesskey}

# Request format

The data window duration (period) is a mandatory parameter.

api.sundaya.com/hse/week/{ending}

e.g. [api.sundaya.com/hse/week](http://api.sundaya.com/hse/week) - will return data for the current week.

The ending date-time of the data window (ending) is optional and will default to the current date and time if missing.

api.sundaya.com/hse/week/{ending}

e.g. api.sundaya.com/hse/week/20190210 - returns data for the week ending on 2019-02-10

Valid values for period are provided in the Table 4 below:

|  |  |
| --- | --- |
| period |  |
| hour | hourly data (1 day), broken down by *minute* and *second* The *hour* dataset produces near-real-time monitoring data for troubleshooting and device reconfiguration by onsite field technicians.  The actual interval is based on the number of seconds currently configured as the monitoring interval on the device - which is therefore the minimum window size possible. |
| timeofday | Morning, Afternoon, Evening, Night: 6-hourly blocks of time commonly used to refer to time during the day. Data for this period covers one week, so that it can be superimposed on data returned by 'week' (2 requests needed). |
| day | daily data, broken down by *hour / time-of-day* (Morning, Afternoon, Evening, Night) and *minute* |
| week | weekly data, broken down by *day-of-week* and *hour / time-of-day* |
| month | monthly data, broken down by *week* and *day* |
| quarter | quarterly data, broken down by *month* and *week* |
| year | yearly data, broken down by *quarter* and *month* |
| 5year | 5yearly data, broken down by *year* and *quarter* |

Table 2 List of periods

The ending date/time for the period (ending) must be expressed in ISO8601 date/time format.

http://api.sundaya.com/hse/{period}/{ending}

e.g. http://api.sundaya.com/hse/week/20190210

The compressed version of ISO 8601 (without semi colons) is recommended, with the time designator T preceding the time component of the representation. This can be specified in UTC or local time as follows:

UTC, expressed with a trailing Z

http://api.sundaya.com/week/YYYYMMDDThhmmssZ

e.g. http://api.sundaya.com/week/201902091830Z == 18:30 UTC

In local time with time offset

<http://api.sundaya.com/week/YYYYMMDDThhmmss±hhmm>

e.g. http://api.sundaya.com/week/201902091500-0330 == 18:30 UTC

## HTTP request and response

The following example shows a sample HTTP request and response, and a snippet of the response collection which is described in the following section.

\*\*\* REQUEST \*\*\*

GET /hse/week/20190209/ HTTP/1.1

Host: api.sundaya.com

Accept: application/vnd.collection+json

\*\*\* RESPONSE \*\*\*

200 OK HTTP/1.1

Content-Type: application/vnd.collection+json

Content-Length: xxx

{ "collection" : {...}, ... }

# Response format

The HSE api response contains a data array for the requested period, for each of the four energy monitoring datasets shown in Table 1 above.

harvest

store

enjoy

grid

Each dataset is in a two-dimensional data grid consisting of data for the ‘child’ and ‘grandchild’ of the requested period. For example, a request for a 'week' period will return with data for each day (child of week,) and hour (grandchild of week) as shown in the table below.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Harvest | week | 04/02/2019-10/02/2019 | |  |  |  |  |
| hour day | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday | Sunday |
| 6 | 54 | 3640 | 54 | 54 | 54 | 54 | 54 |
| 7 | 3640 | 3643 | 30 | 30 | 30 | 3520 | 30 |
| 8 | 33 | 33 | 33 | 33 | 3520 | 33 | 3520 |
| 9 | 23 | 23 | 23 | 3520 | 3640 | 2333 | 23 |
| 10 | 55 | 1233 | 2333 | 2000 | 2333 | 55 | 2000 |
| 11 | 44 | 44 | 3520 | 2000 | 44 | 44 | 44 |
| 12 | 54 | 3640 | 54 | 54 | 54 | 54 | 54 |
| 13 | 3640 | 3643 | 30 | 30 | 30 | 3520 | 30 |
| 14 | 33 | 33 | 33 | 33 | 3520 | 33 | 3520 |
| 15 | 23 | 23 | 23 | 3520 | 3640 | 2333 | 23 |
| 16 | 55 | 1233 | 2333 | 2000 | 2333 | 55 | 2000 |
| 17 | 44 | 44 | 3520 | 2000 | 44 | 44 | 44 |
| 18 | 54 | 3640 | 54 | 54 | 54 | 54 | 54 |
| 19 | 3640 | 3643 | 30 | 30 | 30 | 3520 | 30 |
| 20 | 33 | 33 | 33 | 33 | 3520 | 33 | 3520 |
| 21 | 23 | 23 | 23 | 3520 | 3640 | 2333 | 23 |
| 22 | 55 | 1233 | 2333 | 2000 | 2333 | 55 | 2000 |
| 23 | 44 | 44 | 3520 | 2000 | 44 | 44 | 44 |
| 24 | 54 | 3640 | 54 | 54 | 54 | 54 | 54 |
| 1 | 3640 | 3643 | 30 | 30 | 30 | 3520 | 30 |
| 2 | 33 | 33 | 33 | 33 | 3520 | 33 | 3520 |
| 3 | 23 | 23 | 23 | 3520 | 3640 | 2333 | 23 |
| 4 | 55 | 1233 | 2333 | 2000 | 2333 | 55 | 2000 |
| 5 | 44 | 44 | 3520 | 2000 | 44 | 44 | 44 |
| total | 15396 | 34464 | 23972 | 30548 | 38484 | 24156 | 22684 |

Table 3 Example Harvest data for a ‘week’ period

In the API response, data for each hour of each day is represented in a multivalued list as a data row, and data for each hour of the week (each sub value in the data row) represents a column.

{

"name": "harvest-data",

"value": "54 3640 33 23 55 44 54 3640 33 23 55 44 54 3640 33 23 55 44 54 3640 33 23 55 44",

"prompt": "Monday"

},

{

"name": "harvest-total",

"value": "15396"

}

## Response periods

The response includes (href) links to the parent dataset and next-previous datasets, which allows developers and users to graph and visualise data through a single request, and to navigate hyperlinks to the next and previous dataset (next week or previous week); or to zoom out to the parent dataset (this month's data). The following table depicts the response data structures for each of the eight periods shown in Table 4 List of periods.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| <period> |  | <-href=*next/prev*-> |  | hour |  | <-href=next/prev(*hour*)-> |  | timeofday |  | <-href=next/prev(*timeofday*)-> |
|  | <y dataset> [#] | href[]=<*child>* |  |  | second [60] | href[]=*minute* |  |  | timeofday [4] | href[]=*day* |
|  |  | <x dataset> [# elements] |  |  |  | minute [60] |  |  |  | day [7] |
| href=<*parent>* |  |  |  | href=*day* |  |  |  | href= |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| day |  | <-href=next/prev(*day*)-> |  | week |  | <-href=next/prev(*week*)-> |  | month |  | <-href=next/prev(*month*)-> |
|  | minute [60] | href[]=*hour* |  |  | hour [24] | href[]=*day* |  |  | day [7] | href[]=*week* |
|  |  | hour [24] |  |  |  | day [7] |  |  |  | week [4] |
| href=*week* |  |  |  | href=*month* |  |  |  | href=*quarter* |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| quarter |  | <-href=next/prev(*quarter*)-> |  | year |  | <-href=next/prev(*year*)-> |  | 5year |  | <-href=next/prev(*5year*)-> |
|  | week [4] | href[]=*month* |  |  | week [4] | href[]=*month* |  |  | month [12] | href[]=*year* |
|  |  | month [4] |  |  |  | month [12] |  |  |  | year [5] |
| href=*year* |  |  |  | href=*5year* |  |  |  | href= |  |  |

Table 4 Dataset composition for each period

## Response message elements

The response data is formatted as anonymous objects, so that client apps can process and navigate the response using generic hypermedia controls based on the Collection+JSON standard. The following describes the collection response data structure.

|  |  |  |  |
| --- | --- | --- | --- |
| version |  |  | The api version. Previous and deprecated endpoints will be hosted with a version as a base path prepended to the api path. For example, http://v1.0/api.sundaya.com/hse/week/20190209 |
| href |  |  | Permanent link to the item (a self-reference). For example:  http://api.sundaya.com/hse/week/20190209 represents hse data for the week ending 09-Feb-2019. |
| links[] |  |  | Array of links to resources related to the requested collection. Each links has the following properties to describe its function: |
|  | name |  | Contains the key name which is generally intended for presentation |
|  | rel |  | The rel property contains the link-to-collection relationship descriptor, which can be one of the following values:  period - the period classifier for this collection as indicated by the name property. The name property contains a classifier corresponding to the period specified in the original request. The name property should be presented to the user. The href property does not typically need to be presented for this link.  interval - the starting and ending time points for the period. The name property should be presented to the user. The href property does not typically need to be presented for this link.  parent - a link to the parent of the requested collection. For example: <http://api.sundaya.com/hse/week/20190202> links to the previous week.  next, previous - a link to the next or previous sibling of this collection. For example: <http://api.sundaya.com/hse/week/20190202> links to the previous week.  itemdata-period - the period classifier for each collection. The name property contains the period classifier of the data and totals in each data object.  itemdata-headings - the column headings for data and totals in the items collection. The name property contains column headings in ssv format for the data and totals in each data object.  itemdata-sub value-period - the period classifier for each sub value in the item data object. The name property contains the period classifier of the (ssv) subitems in each data object.  itemdata-sub value-headings - the row headings for data sub values in the items collection. The name property contains row headings in ssv format for the (ssv) subitems in each data object. |
|  | prompt |  | Documentation for this link, which may be used as tooltips in the presentation. |
|  | href |  | The URI of the related resource, and the link identifier. This property may be used to search and identify a link through JSONpath |
|  | render |  | 'image' or 'text' if the link should be retrieved and embedded; or 'link' to display as-is. If the property is missing it indicates that the link does not need to be presented to the end-user. |
| items[] |  |  | Collection members, represented by an object with 3 predefined properties. |
|  | href |  | Permanent URI link to the item. |
|  | data[] |  | Array of key value pairs with 3 predefined properties, |
|  |  | name | The key name, which can be one of the following.  harvest-data - the value property contains row data and a total for each row.  harvest-totals - the value property contains the total for the harvest-data value.  The data element will contain a child element for each of the four energy-flows (Harvest, Store, Energy, Grid) like the two examples shown above for harvest. |
|  |  | value | The values for this item. Multiple values are provided in ssv format (as space separated values). |
|  |  | prompt | A description of the data item. |
|  | links[] |  | Array of links to resources related to this single item. Each links has the same properties as the collection links described above. This property is optional. |

Table 5 JSON message elements

## Data Representation

The API response message contains data representations needed for rendering the graph, including display labels and hyperlinks to navigate to related datasets. Entities which need to be rendered onto the graph are shown below in Figure 3.

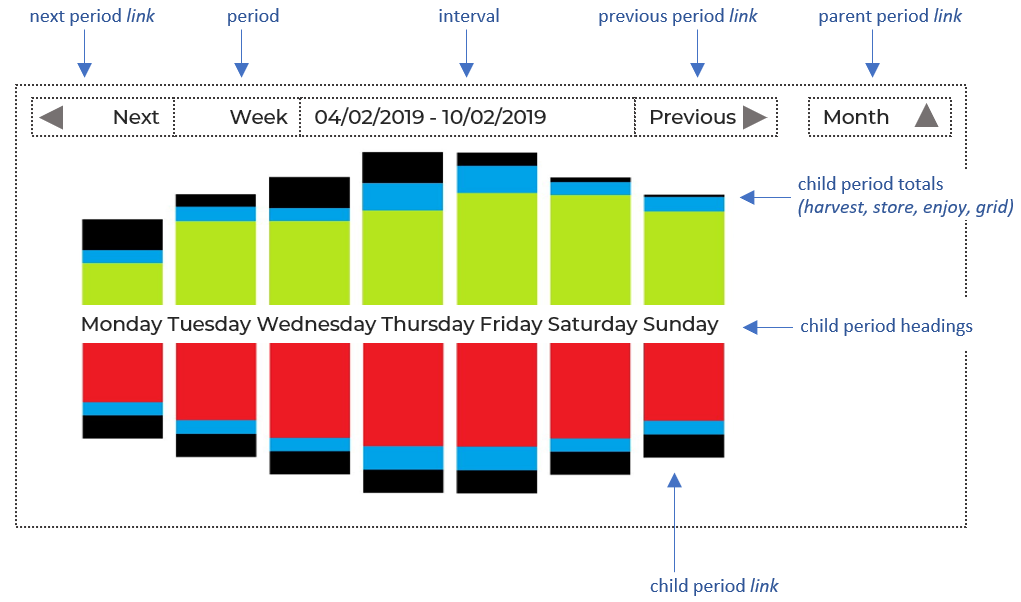


Figure 3 Data entities needed for rendering the graph

Table 6 shows the JSON paths for extracting data from the response message. These paths can be tested with the sample data file (hse-response.json) at <http://jsonpath.com/>. As shown totals, headings, and links contain a similar number of items (seven for week period).

|  |  |  |
| --- | --- | --- |
| entity | JSONPath | sample data |
| period | $.collection.links[?(@.rel == 'period')].name | "Week" |
| interval | $.collection.links[?(@.rel == 'interval')].name  $.collection.href | "04/02/2019 - 10/02/2019"  "http://api.sundaya.com/hse/week/20190210" |
| next period *link* | $.collection.links[?(@.rel == 'next')].name  $.collection.links[?(@.rel == 'next')].href | "Next"  "http://api.sundaya.com/hse/week/20190203" |
| previous period *link* | $.collection.links[?(@.rel == 'previous')].name  $.collection.links[?(@.rel == 'previous')].href | "Previous"  "http://api.sundaya.com/hse/week/20190217" |
| parent period *link* | $.collection.links[?(@.rel == 'parent')].name  $.collection.links[?(@.rel == 'parent')].href | "Month"  "http://api.sundaya.com/hse/month/201902" |
| child period totals | $.collection.items.[data].[?(@.name == 'harvest-total')]  $.collection.items.[data].[?(@.name == 'store-total')]  $.collection.items.[data].[?(@.name == 'enjoy-total')]  $.collection.items.[data].[?(@.name == 'grid-total')] | "15396","34464","23972","30548","38484","24156","22684"  "15396","34464","23972","30548","38484","24156","22684"  "15396","34464","23972","30548","38484","24156","22684"  "15396","34464","23972","30548","38484","24156","22684" |
| child period headings | $.collection.links[?(@.rel == 'itemdata-headings')].name | "Monday Tuesday Wednesday Thursday Friday Saturday Sunday" |
| child period *link* | $.collection.items.[href] | "http://api.sundaya.com/hse/day/20190204",  "http://api.sundaya.com/hse/day/20190205",  "http://api.sundaya.com/hse/day/20190206",  "http://api.sundaya.com/hse/day/20190207",  "http://api.sundaya.com/hse/day/20190208",  "http://api.sundaya.com/hse/day/20190209",  "http://api.sundaya.com/hse/day/20190210" |

Table 6 JSON paths for extracting message data needed by the graph

As shown by the paths and sample data above:

totals - paths shown in child period totals are used to retrieve the data needed to graph the bar chart.

child data - each bar has a hyperlink to its data; these links can be retrieved with the paths in child period link.

related data - hyperlinks for related datasets may be retrieved with the paths shown for next, previous, parent, and child.

labels - display labels can be retrieved with paths shown for period, interval and child period headings.