Contents

[API overview 2](#_Toc1301490)

[Data visualisation 2](#_Toc1301491)

[API specification 3](#_Toc1301492)

[Date and time format 3](#_Toc1301493)

[Timezones 3](#_Toc1301494)

[Media Types 4](#_Toc1301495)

[Headers 4](#_Toc1301496)

[Link-relation Types 4](#_Toc1301497)

[Request paths 5](#_Toc1301498)

[Operations 5](#_Toc1301499)

[Responses 5](#_Toc1301500)

[Versions 6](#_Toc1301501)

[Response object 7](#_Toc1301502)

[Response periods 8](#_Toc1301503)

[Data representation 9](#_Toc1301504)

[Application views 11](#_Toc1301505)

[Graph format 11](#_Toc1301506)

[Data paths 12](#_Toc1301507)

Tables

[Table 1 Colour codes & energy sources 2](#_Toc1301508)

[Table 2 Link-relation Types 4](#_Toc1301509)

[Table 3 List of periods 5](#_Toc1301510)

[Table 4 API operations 5](#_Toc1301511)

[Table 5 Response codes 6](#_Toc1301512)

[Table 6 Example data structure for a ‘week’ period 7](#_Toc1301513)

[Table 7 Dataset composition for each period 8](#_Toc1301514)

[Table 8 JSON message elements 10](#_Toc1301515)

[Table 9 JSON paths for extracting message data needed by the graph 12](#_Toc1301516)

Document History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | API | Author | Description |
| 0.1 | 13/02/2019 | 0.1 | Ajit Chanmugam | First draft. Issued to Hokuapps. |
| 0.2.1 DRAFT | 21/02/2019 | 0.2 | Ajit Chanmugam | Changed all link-relations to comply with RFC8288. Added new sections: media types, link types. |

# 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 source |
|  | harvest |
|  | store |
|  | enjoy |
|  | grid |

Table 1 Colour codes & energy sources

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>.

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 if needed (in the request Body) to filter the returned data by category, subcategory, and product-type of the energy asset.

## Versions

The API endpoint host is <http://api.sundaya.com>. All requests to the API endpoint receive the latest version of the API. The client can request a specific version via the Accept header.

Accept: application/vnd.sundaya.v1+yaml

## Date and time format

Date and time parameters including the end of a period (end) must be expressed in ISO8601 date/time format and must conform with RFC3359.

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

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

The compressed version of ISO 8601 (without semi colons) is required, with the time designator T preceding the time component of the representation.

## Timezones

Timezones must be explicitly specified where API parameters allow for a timestamp to be provided.

This can be specified in UTC or local time as shown:

UTC, expressed with a trailing Z

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

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

...or local time with an offset

[http://api.sundaya.com/hse/week/YYYYMMDDThhmmss±hhmm](http://api.sundaya.com/week/YYYYMMDDThhmmss±hhmm)

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

## Media Types

Parameters in the Request Body and all Response data are currently sent and received in JSON. We encourage using Accept and Content-Type headers so that clients remain backwardly compatible as more media types and hypermedia schema are introduced in future.

These media types are currently supported:

application/json

application/vnd.collection+json

## Headers

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

## Link-relation Types

Link-relations in the Response object are based on RFC8288. The following registered link-relation types are used.

|  |  |
| --- | --- |
| rel | Description |
| self | Identifies the link's context.  In ‘collection.links’ it identifies the collection (name = week)  e.g. href=http:/api.sundaya.com/hse/week/20190210  In ‘collection.items.links’ it identifies an item in the collection (name = day)  e.g. href=http:/api.sundaya.com/hse/day/20190204 |
| item | Identifies child resources of the collection represented by the link's context.  In ‘collection.links’ it targets the item series whiich make up the collection (name = week.days).  e.g. href=http:/api.sundaya.com/hse/day/20190204/20190210  In ‘collection.items.links’ it targets subitems of the item in that context (name = day.hours).  e.g. href=http:/api.sundaya.com/hse/hour/201902050600/201902050500 |
| up | Identifies the parent the collection or item represented by the link's context (name = week.month). |
| next | Identifies the next sibling of the collection or item series represented by the link's context (name = week.next). |
| prev | Identifies the previous siblings of the collection or item series represented by the link's context (name = week.previous). |

Table 2 Link-relation Types

# Request paths

In all requests the caller must provide a mandatory customer site (siteid) and access key:

api.sundaya.com/hse/{period}/{end}?site={siteid}&key={accesskey}

period specifies the duration of the monitoring window and is a mandatory parameter.

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

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

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

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

e.g. api.sundaya.com/hse/week/20190210 - 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* |
| timeofday | 6-hourly blocks of time referred to as Morning, Afternoon, Evening, Night. |
| day | daily data, broken down by *hour* and *minute.* |
| week | weekly data, broken down by *dayofweek* and *hour*, from 0600-2400 hrs. |
| 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 3 List of periods

## Operations

The API supports basic CRUD operations (create, read, update, and delete) using standard HTTP method requests, as summarized in the following table.

|  |  |
| --- | --- |
| Operation | Definition |
| GET | Retrieve information about the resource. |
| POST | Create, backup, or restore the resource. |
| PUT | Update the resource. |
| DELETE | Delete the resource. |

Table 4 API operations

## Responses

The API supports a limited set of responses for each API path, based on the following.

|  |  |  |
| --- | --- | --- |
| Code | Status | Definition |
| 200 | OK | 200 is returned for a successful GET or PUT, if the request was successfully completed. |
| 201 | Created | 201 is returned for a successful POST, if a resource was successfully created. The Location response header provides the URL of the created resource. |
| 400 | Bad Request | If the request was missing information (such as a required value), malformed or invalid and could not be processed. |
| 401 | Unauthorized | If the request is not authorized or is missing or contains an invalid access key. |
| 403 | Forbidden | If the request was trying to do something that is not allowed. |
| 404 | Not Found | If the resource never existed at this Location. |
| 410 | Gone | If the resource once existed here but is no longer available at this Location. |
| 405 | Method Not Allowed | If the requested operation (DELETE, GET, POST, PUT) is not supported by this API path. |
| 415 | Not Acceptable | If the requested Content-Type  is not supported, for example if XML is requested for an API path which accepts only JSON. |
| 500 | Internal Server Error | If the server encountered an unexpected condition and we don’t know what happened (and did not fulfil the request). |

Table 5 Response codes

# Response object

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 provided 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 totals for each ‘day’ (child of week,) and data for each ‘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 6 Example data structure 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 ",

"value": "15396"

},

{

"name": "harvest.hours",

"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",

}

## 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*)-> |
|  | <grandchild> [#] | href[]=<*child>* |  |  | second [60] | href[]=*minute* |  |  | timeofday [4] | href[]=*day* |
|  |  | <child> [# 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 7 Dataset composition for each period

## Data representation

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:  self - the period classifier for the collection. The prompt attribute contains a generic label (e.g. ‘Week’) The title attribute contains the starting and ending time points for the interval which this period spans.  item – this link refers to the items in the collection, each of which contains data for the child period. The prompt attribute describes the child period interval (e.g. ‘Day’). The title attribute contains row headings in ssv format for the totals (for harvest, store, enjoy, grid) corresponding to the titles of all items in the collection.  up - the link to the parent of the collection. The prompt attribute contains a generic label (e.g. ‘Month’); The title attribute contains the period title (e.g. ‘February’). The href property contains a URL link to the parent resource of the requested collection. For example: [http://api.sundaya.com/hse/month/201902](http://api.sundaya.com/hse/period/month/20190202).  next,prev - the link to the next or previous period in the series. For example: [http://api.sundaya.com/hse/week/20190203](http://api.sundaya.com/hse/period/week/20190203) is a link to the previous week. | |
|  | prompt | Documentation for this link, which may be used as tooltips in the presentation. | |
|  | href | The URI or URL of the resource. | |
|  | render | 'image' or 'text' if the link should be retrieved and embedded; or 'link' to display as-is. If the property is missing the href link does not need to be presented. | |
| 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.day - the value property contains the total (column data) for the item.  harvest.day.hours - the value property contains multivalues (row data) for each item.  The data element will contain a child element for each of the four energy sources (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 link has the same five properties as the collection links described above. This property is optional. | |
|  |  | rel | The rel property contains the link-to-dataitem relationship descriptor, which can be one of the following values:  self - the title for this period is in the title attribute.  item - The ‘grandchild’ of the collection, corresponding to the (ssv) sub values in the value field of data objects in the array. The title property contains the row headings in ssv format for the (ssv) subitems in each data object (e.g. harvest.hours). |
|  |  | name | Contains the key name, for presentation. |
|  |  | prompt | Documentation for this link, which may be used as tooltips in the presentation. |
|  |  | title | The title of the link as described above in rel. |
|  |  | href | The URI of the related resource, and the link identifier. |
|  |  | 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. |

Table 8 JSON message elements

# Application views

The API response message data can be rendered graphically or as a table of composite data.

## Graph format

The graph format including display labels and hyperlinks to navigate to related datasets, is shown below in Figure 3.

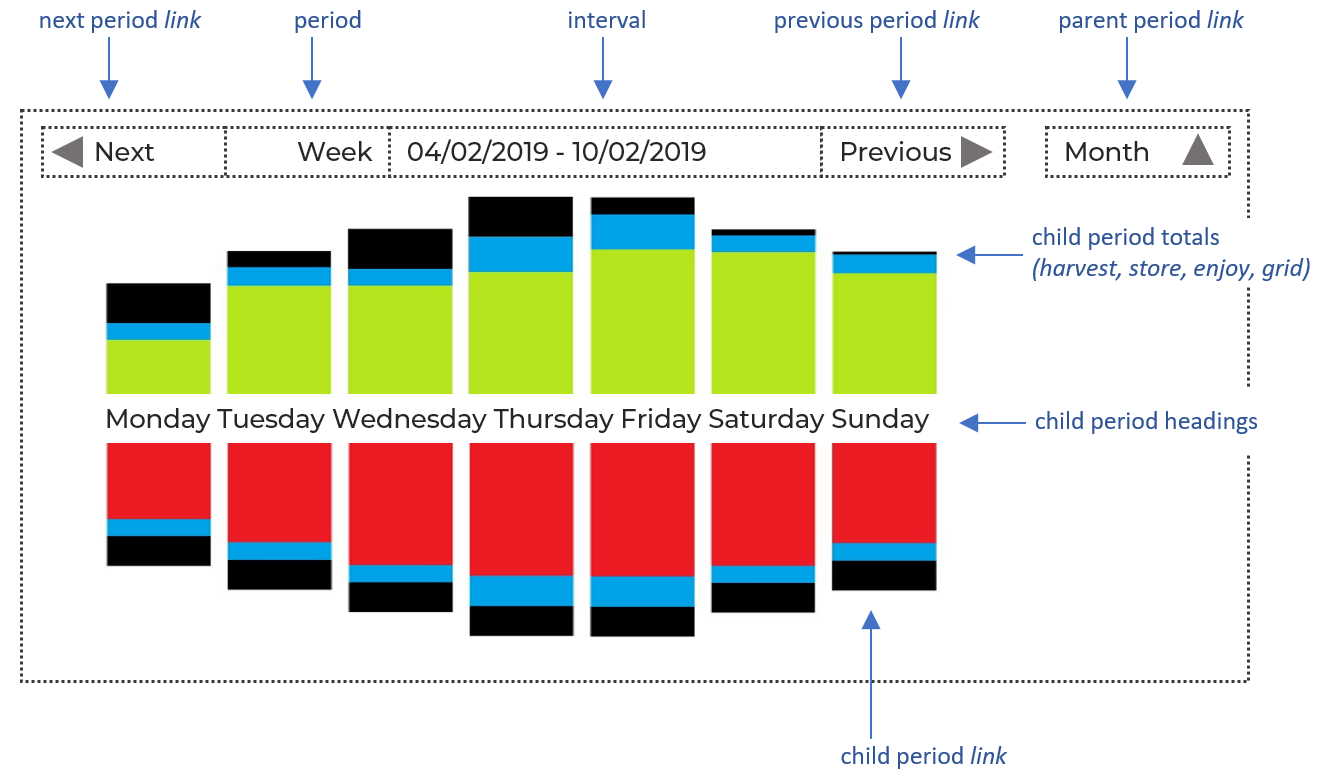


Figure 3 Required data entities for rendering a graph

## Data paths

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 correlated items (seven in the case of a ‘week’).

|  |  |  |
| --- | --- | --- |
| entity | JSONPath | data sample |
| period | $.collection.links[?(@.rel == 'self')].prompt | "Week" |
| interval | $.collection.links[?(@.rel == 'self')].title | "04/02/2019 - 10/02/2019" |
| next period link | $.collection.links[?(@.rel == 'next')].prompt  $.collection.links[?(@.rel == 'next')].href | "Next"  "http://api.sundaya.com/hse/week/20190217" |
| previous period link | $.collection.links[?(@.rel == 'prev')].prompt  $.collection.links[?(@.rel == 'prev')].href | "Previous"  "http://api.sundaya.com/hse/week/20190203" |
| parent period link | $.collection.links[?(@.rel == 'up')].prompt  $.collection.links[?(@.rel == 'up')].href | "Month"  "http://api.sundaya.com/hse/month/201902" |
| child period totals | $.collection.items.[data].[?(@.name == 'harvest.day')].value  $.collection.items.[data].[?(@.name == 'store.day.in')].value  $.collection.items.[data].[?(@.name == 'store.day.out')].value  $.collection.items.[data].[?(@.name == 'enjoy.day')].value  $.collection.items.[data].[?(@.name == 'grid.day.in')].value  $.collection.items.[data].[?(@.name == 'grid.day.out')].value | "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 titles | $.collection.items.[links].[?(@.rel == 'self')].title | "Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday" |
| child period link | $.collection.items.[links].[?(@.rel == 'self')].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 9 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.