

WattzOn Snapshot API v1 Documentation

- WattzOn Snapshot API v1 Documentation
 - Definitions
 - Service Introduction
 - Authentication
 - Rate Limit
 - API Return Codes
 - Job Service
 - Job Statuses
 - Create Job
 - Retrieve a List of Recent Jobs
 - Retrieve Results for a Specific Job
 - Retrieving a Bill Image
 - Utility Providers
 - Revision

WattzOn Snapshot is a software-as-a-service (SaaS) platform for extracting usage data from images of bills from residential utility providers. This document describes the individual Snapshot API calls and their features.

The Snapshot API uses JSON as a serialization format. As such, it requires no specific software to use. Authentication and communication proceed over TLS; clients must present a certificate.

Definitions

- *Client* The software interacting with the WattzOn Snapshot API.

Service Introduction

The API provides for:

- Creating a Job (uploading images of a utility bill)
- Retrieving a list of recent Jobs
- Retrieving the results of a specific Job
- Retrieving bill images associated with a Job

Results are returned as JSON objects or HTTP error codes.

Authentication

All client identification and authorization is made through a client-side TLS certificate. You do not need to purchase a certificate; WattzOn will act as a certificate authority for its own application. The procedure is as follows:

1. Generate a CSR (Certificate Signing Request). Here's how to do it with openssl:

```
openssl req -new -newkey rsa:2048 -nodes -keyout client.key -out client.csr
```

Don't enter a challenge password. This creates two files: `client.csr` (the CSR) and `client.key` (the private key).

1. Send the CSR to your WattzOn technical or sales contact. Do NOT send the private key; keep it secret.
2. WattzOn will sign the CSR and send you a certificate named `client.crt`. You can now use this in conjunction with your private key (`client.key`) to connect to the Snapshot API. However, depending on your client software, you may need to perform another step to create a combined certificate-key file.
 - Some HTTPS clients may require a `.pfx/.pf12` file, a DER file, or another format. OpenSSL can convert to these formats.

NOTE: TLS certificates expire after one year. To avoid an interruption in service, arrange to have a new certificate signed before your current one expires. Your existing signed certificate will continue to work until it expires, giving you time to deploy the new certificate.

Rate Limit

Job creation is limited to a maximum of 6 requests per 3 seconds per client. Job creation requests exceeding this rate limit are rejected with a 429 "Too Many Requests" response.

API Return Codes

Successful calls will return an HTTP 2xx code (typically HTTP 200 (Successful), HTTP 201 (Created), or HTTP 202 (Accepted)). Unsuccessful calls will return one of the following HTTP codes:

Code	Meaning
401	Unauthorized. The certificates are not present or are not correct for this client
403	Forbidden. The client does not have access to the requested resource

404	Not found. The requested resource does not exist
429	Too Many Requests. Rate Limit has been exceeded
500	Internal Server Error

Job Service

Job Statuses

A Job can be in one of several states, as reflected by the `status` attribute. Newly created jobs are either `PENDING` or, if the provider is 'Other', `IMAGE_ONLY`. (The 'Other' provider indicates that images are to be held without processing.)

Once a Job has entered the processing pipeline, its state becomes `IN_PROGRESS`.

After processing, the job state will be either `COMPLETE`, indicating that results are available, or, in very rare circumstances, `ERROR`, indicating that something has gone unrecoverably wrong with processing.

`IMAGE_ONLY`, `COMPLETE`, and `ERROR` are 'final' states, meaning that no further processing will happen.

Snapshot may, over time, add additional Job statuses. As a convenience, the `final` attribute indicates whether the job status is a 'final' one (`final` will be 0 if the job is still being processed, or 1 if the job is complete).

Create Job

path: `/api/v1/jobs/`

method: POST

input parameters:

- `customer_email`: A valid email address for the customer of the Job
- `provider`: A utility provider identifier from the table below
- `tags`: A list of tags to associate with this Job.
- `attachment`: A list of files to process. Each attachment element contains
 - `name`: The name of the file, with extension
 - `content`: The base64-encoded contents of the file

Tags are strings. (Numbers will be converted to strings). Each is limited to 255 characters, and should not contain commas. Limit yourself to alphanumeric, colons, and dashes, and you'll be fine.

The attachment list must contain either one or two elements. If one element, the file must be a PDF of a bill. If two elements, the files must be JPEG images of the summary and detail pages of the bill, in that order.

output:

- **HTTP 201 (Created)**
- **id:** The tracking ID of the Job
- **created_at:** The time (in UTC) that the Job was created
- **customer_email:** The email address provided when the Job was created
- **provider:** The ID of the bill's provider, from the Provider table below
- **status:** Either **PENDING** or, if the provider is 'Other', **IMAGE_ONLY**
- **tags:** The list of tags to associate with this Job

Example:

```
page1content=$(base64 pge-bill-page1.jpg)
page3content=$(base64 pge-bill-page3.jpg)
curl --key <CLIENT_KEY_FILE> --cert <CLIENT_CERT_FILE> \
-H "Content-Type: application/json" \
-X POST -d '{"customer_email":"user@example.com","tags":["sfid:12345","demo"],"attachments":
[{"name":"pge-bill-page1.jpg","content":"$page1content"},{"name":"pge-bill-
page3.jpg","content":"$page4contents"}],"provider":4}' \
https://snapshot.wattzon.com/api/v1/jobs/
```

which returns

```
{
  "id": "579b83a40640fd4cb0837ad5",
  "created_at": "2016-07-25T16:26:12.320000",
  "customer_email": "user@example.com",
  "status": "PENDING",
  "final": 0,
  "tags": ["sfid:12345", "demo"]
}
```

Note that it typically takes at least 30 seconds to process a Job.

Retrieve a List of Recent Jobs

path: /api/v1/jobs/

method: GET

input: (optional) tag

output:

- HTTP 200 on success.
- A list of Job summaries. Each element contains:
 - id: The tracking ID of the Job
 - customer_email: The email address provided when the Job was created
 - created_at: The data and time (in UTC) that the Job was created
 - status: The job status, as described above
 - final: 0 if the job is still being processed or 1 if the job is complete
 - tags: The list of tags associated with a Job

The most recent 20 Jobs are returned, ordered by submission time, with most recently submitted Job first. (At times more than 20 Job summaries may be returned, but this version of the API guarantees only the most recent 20 summaries.)

A future version of the API specification will describe a way to page through Job summaries.

Example:

```
curl --key <CLIENT_KEY_FILE> --cert <CLIENT_CERT_FILE> -X GET  
https://snapshot.wattzon.com/api/v1/jobs/
```

will return something like:

```
[  
  {  
    "created_at": "2016-06-12T04:25:40.746000",  
    "customer_email": "user@example.com",  
    "id": "57846eec0640fd577c800c93",  
    "tags": [],  
    "status": "IN_PROGRESS",  
    "final": 0  
  },  
  {  
    "created_at": "2016-06-12T04:15:40.746000",  
    "customer_email": "user@example.com",  
    "id": "57846eec0640fd577c700fb1",  
    "tags": ["sfid:12345"],  
    "status": "COMPLETE",  
    "final": 1  
  }  
]
```

Filtering the Job List by tag

Adding `?tag=value` to the URL will restrict the search to Jobs that have "value" in their tag lists.

Retrieve Results for a Specific Job

path: `/api/v1/jobs/<id>/`

method: GET

input: id: The tracking ID of the Job

output:

- HTTP 200 on success. HTTP 404 on error.
- id: The tracking ID of the Job
- created_at: The time (in UTC) that the Job was created
- customer_email: The email address provided when the Job was created
- status: A Job status, as described above
- tags: The list of tags associated with the Job
- attachments: A list describing the files representing pages of the bill

An attachment contains the following fields:

- id: The id that can be used to retrieve the file
- name: The name of the file
- Image validation fields (see below)

If the Job is complete, the output will additionally contain

- provider: The provider code supplied when the Job was created
- read_data: Information extracted from the bill images
 - account_number: The account number extracted from the bill
 - meter_number: The meter number extracted from the bill
 - service_address: The service address extracted from the bill
 - period_start: The period start extracted from the bill
 - period_end: The period end extracted from the bill
 - days_in_period: The number of days in the billing period
 - usage_units: The kilowatt hours for this billing period, extracted from the bill
 - amount_due: The amount due, in dollars, for this billing period, extracted from the bill
- chart: A list of data extracted from the usage bar chart, one element per bar
 - month: A month (1-12) corresponding to the bar
 - year: The year (e.g., 2016) corresponding to the bar
 - value: The 'value' of the electric component of the bar. The interpretation of this number depends on the utility provider; depend instead on `monthly_kWh_avg` in `computed_data` below.

Note: If Snapshot was unable to read the chart, the `chart` list will be empty.

- `computed_data`: Data computed from information extracted from the bill
- `monthly_kWh_avg`: Average monthly kilowatt hours
- `annual_kWh`: Annual kilowatt hours
- `monthly_kWh_avg`: A list corresponding to `chart`, but with `value` converted in to kilowatt hours (kWh)

Additional data may appear in `read_data`, depending on the utility provider.

Image Validation (new in 1.30)

Information extraction depends on the quality of the bill images. To provide feedback guidance for improving pictures, and thus results, Snapshot adds several boolean attributes to each JPEG attachment to signal certain conditions.

- `blur` is true if Snapshot detects a blurry, out-of-focus image.
- `dark` is true if Snapshot detects a dark image. This is most often caused by poor lighting.
- `low_contrast` is true if Snapshot detects low contrast. This is often caused by over-lighting or wash-out due to flash.
- `tilt` is true if Snapshot determines that the picture is sufficiently off-center as to cause OCR problems. This happens if the camera isn't centered over the image, causing the image to 'tilt'.

Example:

```
curl -X GET --key client.key --cert client.crt
https://snapshot.wattzon.com/api/v1/jobs/57846eec0640fd577c700fb1/
```

will return something like:

```
{
  "id": "57846eec0640fd577c700fb1",
  "customer_email": "user@example.com",
  "provider": 4,
  "created_at": "2016-06-12T04:15:40.746000",
  "tags": ["sfid:99999"],
  "attachments": [{
    "id": "579b96566dee5606dd2ed8d5",
    "name": "pge10047-1.jpg",
    "blur": false, "dark": false, "low_contrast": false, "tilt": false,
  }, {
    "id": "579b96566dee5606dd2ed8d6",
    "name": "pge10047-3.jpg",
    "blur": false, "dark": false, "low_contrast": false, "tilt": false,
  }],
  "status": "COMPLETE",
  "final": 1,
```

```
"read_data": {
  "account_number": "9999999999-9",
  "amount_due": 120.62,
  "days_in_period": 30,
  "meter_number": "9999999999",
  "name": "DAVE SMITH",
  "period_end": "2015-12-24",
  "period_start": "2015-11-25",
  "rate_schedule": "E1 X Residential Service",
  "service_address": "123 ANY STREET BENICIA, CA 94510",
  "service_number": "9999999999",
  "usage_units": "1,740.000000"
},
"chart": {
  "columns": [
    {"month": 1, "ordinal": 0, "value": 154.54545454545453, "year": 2015},
    {"month": 2, "ordinal": 1, "value": 131.8181818181818, "year": 2015},
    {"month": 3, "ordinal": 2, "value": 95.45454545454545, "year": 2015},
    {"month": 4, "ordinal": 3, "value": 92.42424242424242, "year": 2015},
    {"month": 5, "ordinal": 4, "value": 93.93939393939394, "year": 2015},
    {"month": 6, "ordinal": 5, "value": 87.87878787878788, "year": 2015},
    {"month": 7, "ordinal": 6, "value": 139.3939393939394, "year": 2015},
    {"month": 8, "ordinal": 7, "value": 174.24242424242425, "year": 2015},
    {"month": 9, "ordinal": 8, "value": 189.3939393939394, "year": 2015},
    {"month": 10, "ordinal": 9, "value": 198.4848484848485, "year": 2015},
    {"month": 11, "ordinal": 10, "value": 177.27272727272728, "year": 2015},
    {"month": 12, "ordinal": 11, "value": 183.33333333333334, "year": 2015},
    {"month": 1, "ordinal": 12, "value": 269.6969696969697, "year": 2016}
  ]
},
"computed_data": {
  "annual_kWh": 8415.536812949367,
  "month": null,
  "monthly_kWh_avg": [
    {"average_kWh": 725.4307882043531, "month": 1, "year": 2015},
    {"average_kWh": 631.9060963272968, "month": 2, "year": 2015},
    {"average_kWh": 516.6126314519541, "month": 3, "year": 2015},
    {"average_kWh": 500.1254123918064, "month": 4, "year": 2015},
    {"average_kWh": 498.65933769413147, "month": 5, "year": 2015},
    {"average_kWh": 471.55587143247476, "month": 6, "year": 2015},
    {"average_kWh": 657.1687252087493, "month": 7, "year": 2015},
```

```
{
  "average_kWh": 761.1816336629186, "month": 8, "year": 2015},
  {"average_kWh": 815.8096590436903, "month": 9, "year": 2015},
  {"average_kWh": 851.6600638395927, "month": 10, "year": 2015},
  {"average_kWh": 795.9022166315358, "month": 11, "year": 2015},
  {"average_kWh": 822.9044768553144, "month": 12, "year": 2015},
  {"average_kWh": 1092.0506884099027, "month": 1, "year": 2016}
],
},
}
```

Retrieving a Bill Image

path: /api/v1/jobs/<job_id>/attachments/<file_id>/

method: GET

input:

- job_id: The tracking ID of the Job
- file_id: The id of an attached file

output:

- HTTP 200 on success.
- HTTP 404 if Job or File does not exist.
- id: The ID of the file
- name: The name of the file
- content: The base64-encoded contents of the file

Example:

```
curl -X GET --key client.key --cert client.crt
```

```
https://snapshot.wattzon.com/api/v1/jobs/57846eec0640fd577c700fb1/attachments/579b96566dee5606dd2ed8d5/
```

will return something like:

```
{
  "id": "579b96566dee5606dd2ed8d5",
  "name": "pge10047-1.jpg",
  "content": "..."
}
```

where ... is a very long 64-encoded string representing the contents of the image.

Utility Providers

WattzOn Snapshot currently supports to following utility providers.

ID	Provider
0	'Other' (images only)
4	Pacific Gas & Electric
1994	Xcel Energy
2118	San Diego Gas and Electric
2251	Southern California Edison

Revision

- 7/11/2016: Initial documentation of the Snapshot v1 API
- 7/12/2016: Formatting and cosmetic fixes to doc
- 7/24/2016: Updated to describe tags and attachments, and the endpoint to retrieve bill images
- 8/27/2016: Noted that on failure to read a chart, an empty list would be returned in the `chart` field
- 1/13/2017: Add description image validation attributes. Add additional description of statuses and the `final` attribute. Explain the 'Other' provider.
- 1/19/2017: Clarified meaning of 'tilt' in image validation