Powershop API

Developer Documentation

API version: 2.0

Documentation version: 1.1 [b31920]

Authors: Will Bryant & Nahum Wild

Last updated: 31 January 2013

Table of Contents

Getting Access to the API

Before you can use the Powershop API, you must sign up for an API account for your product. Send an e-mail to api@powershop.co.nz asking for access. Please include your product name and the e-mail address you used to sign up to Powershop.

All API accounts have a controlled number of requests per hour that they may submit. If you feel that your application will be submitting many requests, please mention this to the Powershop team.

Technical Overview

The Powershop API is available over HTTPS. It returns responses in JSON or XML format.

This API uses the OAuth protocol. This allows users to authenticate themselves without entering their password into your application. OAuth libraries are available for all major development platforms, including PHP, Ruby and the iPhone.

For information about OAuth, visit:
http://oauth.net/

It is a condition of using the Powershop API that you are never able to directly capture the user's password. This includes, for example, using a web control on the iPhone inside your application. Please search for “iphone oauth” to see how others have implemented OAuth for iPhone apps.

Summary of changes since v1

- The fixed top_up API call has been replaced by a more flexible purchase API call, which allows clients to choose the products and quantities.
- The products API call has been extended to return information about pack products such as seasonal powerpacks and featured specials.
- The customer API call has been replaced by the accounts API call, which supports users who have access to more than one Powershop customer account.

The details of the changes are listed below each API call.

OAuth Configuration

Your OAuth library will require several URLs to be configured. For Powershop, these are:

Request Token URL:

https://secure.powershop.co.nz/external_api/oauth/request_token

Authorize URL:
https://secure.powershop.co.nz/external_api/oauth/authorize

Access Token URL:
https://secure.powershop.co.nz/external_api/oauth/access_token

For the test environment, replace “https://secure.powershop.co.nz” with “https://stable.test.powershop.co.nz”.

A callback URL passed to the request token URL is required, as per the OAuth 1.0a specification. Note that your OAuth library may not require this parameter; you must manually specify it in this case or update the library to the latest version.

API Documentation

URL Format

Once you have received an access token using OAuth, use your OAuth library to call the API you are interested in. The format of API URLs is:

https://secure.powershop.co.nz/external_api/v2/***apiname.format***

where ***apiname*** is one of the API calls listed in this document such as **accounts**, **products**, or **purchase**
where **format** is one of **js** or **xml**.
If you use format **js**, the response will be a JSON string. If you use format **xml**, the reponse will be in XML.
Parameters supplied to the calls should be passed URL-encoded. For example, a request to find meter readings for ICP 0123456789 between the 1st and 30th of September 2009, with a JSON response requested, would look like:
https://secure.powershop.co.nz/external_api/v2/meter_readings.js?icp_number=0123456789&
start_date=2009-09-01&end_date=2009-09-30
Your OAuth library will add OAuth token information and a signature before passing it to the Powershop server.
See below for details about each of the different API calls. Note that calls that get information require you use an HTTP GET, and calls that update information require you use an HTTP POST.

Response Codes

A successful call to the API will return an HTTP 200 status code, and either JSON or XML-encoded data.

An unsuccessful call will return HTTP status codes 400, 401 or 503, depending on the error. A single line of text describing the error will also be returned. Error messages common to all API calls are:

[E900] Invalid OAuth signature or consumer key
[E901] Unknown OAuth signature method
[E902] The application has exceeded the request rate limit for this hour
[E903] Invalid OAuth token
[E904] OAuth consumer is disabled
[E905] Expired request token
[E906] Invalid verifier
[E907] oauth_token not found
[E908] oauth_callback must be supplied

As the text of error messages may change, please use the [Exxx] error number to detect a particular error message.

accounts API call

HTTP method:

GET

API URL:

https://secure.powershop.co.nz/external_api/v2/accounts.**format**

Parameters:

none

Description:

The accounts API call returns information about the accounts and properties the currently authenticated user has access to. It does not take any parameters.

Returns:

version = 2.0
result =
- first_name — *the user's first name*
- last_name — *the user's last name*
- accounts **[Array]** =
  - account_number — *the Powershop customer account number*
  - account_name — *the Powershop customer account name*
  - account_balance — *the Powershop customer account balance, in dollars and cents*
  - properties **[Array]** =
    - consumer_id — *the identifier for this property to pass back in API calls*
    - connection_number — *the property's ICP number or NMI*
    - status — *see "notes" section below*
    - status_detail — *see "notes" section below*
    - available_api_calls **[Array]** = any of 'products', 'purchase', 'meter_readings', 'usage_data'
    - address = — *the address of the property*
      - property_name
      - flat_number
      - street_number
      - street_name
      - suburb
      - district
      - region
    - start_date — *yyyy-mm-dd; date property went online*
    - end_date — *yyyy-mm-dd or nil; date property offline*
    - unit_balance — *unit balance of this property*
    - daily_consumption — *number of kWh currently used per day*
    - last_account_review_at — *yyyy-mm-dd hh:mm:ss*
    - registers **[Array]** =

register_number
description *customer-supplied description*
dials *number of dials that the register has*
hidden *1 if hidden; 0 otherwise*
last_reading_at *yyyy-mm-dd hh:mm:ss*
last_reading_value
last_reading_type *actual, customer, estimated, derived*
estimated_reading_value *estimated reading as of now*

Errors:
No specific errors returned by this method.

Notes:

A customer may have several properties on the same account, and a user may also have access to several separate Powershop customer accounts. Your user interface should allow for this possibility by letting the user select which property they wish to work with, if they have more than one.

The account_name is often the name of the user themselves but not always, for example if there are joint account holders, for business accounts.

The status field is one of the following values:

active	*the property is currently being supplied by Powershop*
inbound	*the property is currently being switched to Powershop or the customer is in the process of moving in to the property*
outbound	*the customer is currently being switched away from Powershop or the customer is in the process of moving out of the property*
archived	*the property is no longer active with Powershop*

A property's lifecycle is inbound → active → outbound → archived.

The status_detail field contains a text string when the property's status is either inbound or outbound. This text details what stage the inbound or outbound process is currently in, for example, "waiting for switch from other retailer." This text string can be displayed directly to the user.

Note that you can only use the products and purchase API calls below on ICPs that have a status of active. Trying to use them on other ICPs will return error message E009.

The account_balance field gives the current dollar balance of the account. Negative values mean there is an amount owing (ie. some of the purchases and charges on the account have not yet been paid for), positive values mean the account has an unused credit.

The unit_balance field gives the current kWh balance of the property. A negative value means that the customer is that many units in arrears and will have to purchase power to make up the deficit at the next account review. A positive value means the customer has purchased more power than they have currently used.

A customer may hide a register if it is a "low use" register, for example a meter that is still

physically present but that is no longer connected or used. This means you should generally not display it on your user interface.

Changes from v1: Replaces the customer API call. Returns the user name. Lists all accounts if the user has access to more than one. Gives the account number, account name, and account balance for each account. Returns the ID to use for other API calls for this property. Returns the ICP number in the connection_number field rather than icp_number. Gives the API calls available for each property, which may be less than the full set if the user does not have full rights to a particullar property.

meter_readings GET API call

HTTP method:

GET

API URL:

https://secure.powershop.co.nz/external_api/v2/meter_readings.**format**

Parameters:

consumer_ id	*the ID number of the property as listed in the accounts results*
start_date	*yyyy-mm-dd; retrieve readings from this date*
end_date	*yyyy-mm-dd; retrieve readings up until this date*

Description:

The GET meter_readings API call returns readings for the given ICP.

They may be "actual" – retrieved from a meter reading company, "customer" – supplied by the customer, "estimated" – estimated by Powershop, normally for account review purposes, or "derived" – calculated using smart meter data.

Returns:

version = 2.0

result **[Array]** =

register_number	*reading was taken on this register number*
read_at	*yyyy-mm-dd hh:mm:ss*
reading_type	*actual, customer, estimated, derived*
reading_value	*the reading*

Errors:

[E000] The icp_number parameter must be specified for this call
[E000] The start_date parameter must be specified for this call
[E000] The end_date parameter must be specified for this call
[E001] Dates must be in YYYY-MM-DD format
[E008] icp_number specified not found for this customer
[E009] This property is not active

meter_readings POST API call

HTTP method:

POST

API URL:

https://secure.powershop.co.nz/external_api/v2/meter_readings.**format**

Parameters:

consumer_ id *the ID number of the property as listed in the accounts results*
readings[**register_number**] *the reading for* ***register_number***

Description:

The POST meter_readings API call allows users to enter a reading for the specified ICP.

Note that a full set of readings must be specified, otherwise all readings will be rejected. A full set means that all non-hidden registers must have a valid reading. The reading may not be less than the previous actual reading, and must not be exceed the daily usage thresholds that Powershop calculates for the ICP. If it does, an failure message will be returned.

Returns:

version = 2.0
result =
result *"success" or "failure"*
message *if result is "failure", a customer-readable message describing the failure*

Errors:

[E000] The icp_number parameter must be specified for this call
[E002] Readings must be in an array format
[E003] Register number not found
[E004] Only numbers may be present in the reading value
[E005] A reading for each active register must be supplied
[E008] icp_number specified not found for this customer
[E009] This property is not active

Notes:
It is recommended that you use the "dials" number, found in the results of the accounts API call against each register, to show how the user many dials the register has and allow entry of exactly that many digits. The most common is to have 8 dials.

usage_data GET API call

HTTP method:

GET

API URL:

https://secure.powershop.co.nz/external_api/v1/usage_data.**format**

Parameters:

consumer_ id	*the ID number of the property as listed in the accounts results*
start_date	*yyyy-mm-dd*
end_date	*yyyy-mm-dd; not more than a year after start_date*

Description:

The usage_data call returns pre-calculated data about the user's usage, ready to plot on a graph. It is the same data that is shown on the Powershop web site graphs. Powershop suggests you use this data when displaying graphs to give a seamless experience between the web site and your application.

If more than a year's data is requested, the end date will be truncated so that only a year's worth after the start date will be returned.

Returns:

version = 2.0

result =

start_date	*the start date of the data returned*
end_date	*the end date of the data returned*
graph_type	*optional – see below*
registers **[Array]**	*an array of register numbers*
data **[Array]** =	
day_data	*number of kWh for this day*

Errors:

[E000] The icp_number parameter must be specified for this call
[E008] icp_number specified not found for this customer
[E009] This property is not active
[E010] Invalid graph_type value

Notes:

The graph_type variable is optional. If specified, it must be either *smoothed* or *raw*. If not specified, it defaults to *smoothed*. Powershop will be changing these graph types in the future, and it is likely that this field will become unused.

An array of numbers is returned in the data variable. If you requested 60 days, the data array will have 60 elements, where the first element is the usage on the start date and the last element is the usage on the end date. Each element contains the number of kilowatt-hours (kWh) used on that day. This may be a floating point number, and under certain unusual circumstances, may be negative.

products API call

HTTP method:

GET

API URL:

https://secure.powershop.co.nz/external_api/v2/products.**format**

Parameters:

consumer_ id *the ID number of the property as listed in the accounts results*

Description:

The products API call returns the products available for purchase from the specified property, and the per unit price of the products.

Returns:

version = 2.0

result **[Array]** =

id	*the unique ID of the product*
name	*the name of the product*
description	*a long HTML-formatted description of the product*
type	*nominated, special, powerpack*
price_per_unit	*unit price in dollars, to 4d.p.*
units_per_pack	*the number of units in one pack of the product; nil if the product does not come in packs.*
price_per_pack	*price in dollars for one pack of the product, to 2dp; nil if the product does not come in packs.*
max_packs	*the maximum number of packs of this product that may be purchased at this time. nil if the product does not come in packs or has no purchase limit.*
sold_out	*true if no more units of this product are available for sale*
got_it	*true for specials if the user has already reached the purchase limit for the product.*
image_url	*URL for 130x130 or 165x130 pixel product image*
web_url	*direct link to this product on the Powershop website*
use_by	date the product must be used by

Errors:

[E000] The icp_number parameter must be specified for this call
[E008] icp_number specified not found for this customer
[E009] This property is not active

Notes:

Product IDs are persistent; a particular ID will always refer to the same product. However, all attributes of the product may change from time to time, including the unit price and pack size, and (rarely) the name. There will normally only be one product of a given name offered for a particular property at a particular time, but different consumers may be offered different products with the same name, and over time one product may be replaced by another with the same name.

Some products are offered in packs of a particular value (eg. $49.95). Since the unit price for a consumer (eg. 0.1935 c/unit) does not necessarily divide evenly into this pack price, the price_per_pack may not exactly equal price_per_unit*units_per_pack ($49.92 in this example). For

this reason you should use the supplied price_per_pack value when displaying the cost of the pack to the user, as that is what they will actually be charged if they purchase the pack. The total price of units not sold in packs should always match exactly after rounding to whole cents.

The image_url parameter will not be present if there is no image associated with the product. The image URL supports the HTTP If-Modified-Since header, so you may wish to speed up your user's experience by caching the image. Note that the image may change if the product is out of stock, in which case a different URL will be returned.

Changes from v1: id, units_per_pack, price_per_pack, max_packs, and got_it added.

purchase POST API call

HTTP method:

POST

API URL:

https://secure.powershop.co.nz/external_api/v2/purchase.**format**

Parameters:

consumer_ id	*the ID number of the property as listed in the accounts results*
product	*the ID of the product to purchase*
quantity	*the number of units of the product to purchase*
unit_price	*the expected unit price of the product*

Description:

The purchase POST API call makes a purchase of power.

Returns:

version = 2.0

result =

result	*"success" or "failure"*
message	*if result is "failure", a customer-readable description of the failure.*
account_balance	*if result is "success", the account balance of the customer*

Errors:

[E000] The consumer_id parameter must be specified for this call
[E009] This property is not active
[E014] consumer_id specified not found for this customer

Notes:
If the result is "failure", the message returned should be displayed to the user to inform them why the purchase failed.

Before making this API call callers need to have called the products API to find the IDs of the available products and their effective unit prices. These two attributes must be passed in this API call to ensure that the user pays the amount they expect for the product, which is important since unit prices and pack sizes can and do change regularly for individual consumers. A "failure" result with an explanatory message will be returned if either value has changed.

It is possible to purchase more than one product at a time by passing them multiple times, in which case they must be renamed to product[][product], product[][quantity], and product[][unit_price], the empty square brackets signify an array. Just repeat these params for each product to be purchased.

For pack products (those with a units_per_pack attribute in the **products** results), the quantity requested must be an exact multiple of the units_per_pack.

Changes from v1: New API call, replacing the old **top_up** POST API call. The product and quantity are now specified by the client rather than always purchasing default product to top up to 0 unit balance. Replaces the top_up API's offer_key parameter with the unit_price parameter so that the purchase can be processed if the original offer has been expired but the latest offer has the same price. Returns failures due to price changes using message rather than an error code. Supports

array parameters to purchase multiple products. Returns the unit_balance and account_balance if successful.

contact_us POST API call

HTTP method:

POST

API URL:

https://secure.powershop.co.nz/external_api/v2/contact_us.**format**

Parameters:

consumer_ id *the ID number of the property as listed in the accounts results (optional)*

comment *text of the comment to be posted to Powershop*

Description:

The contact_us POST API call posts some feedback to Powershop. The currently authenticated user's name and e-mail address are automatically attached to the feedback so that Powershop can send them a response.

The comment field may be up to 64KB in length.

Returns:

version = 2.0

result =

result *"success" or "failure"*

message *if result is "failure", a customer-readable description of the failure*

Errors:

[E000] The comment parameter must be specified for this call

Notes:

The property ID is optional but should be provided if the user was viewing a particular property at the time.

Changes from v1: id added.

promotions GET API call

HTTP method:

GET

API URL:

https://secure.powershop.co.nz/external_api/v2/promotions.**format**

Parameters:

none

Description:

The promotions call returns all active referral promotions. The user can use one of these promotions to refer Powershop to a friend.

Returns:

version = 2.0

result =

promotions **[Array]** =

promotion_id	*a numeric identifier for this promotion*
name	*the promotion's name*
description	*a plain text description*
image_url	*a URL of an image related to this promotion*
start_date	*the start date of the data returned*
end_date	*the end date of the data returned*
t_and_c_accepted	*whether the user has accepted T&C*

Errors:

none

Notes:

An empty array of promotions may be returned if there are no active referral promotions. At present, Powershop only runs at most one referral promotion at a time, but this may change in the future.

The promotion description is UTF-8 text. It may contain linefeed characters.

referral POST API call

HTTP method:

POST

API URL:

https://secure.powershop.co.nz/external_api/v2/referral.**format**

Parameters:

promotion_id	*the numeric identifier of the promotion*
name	*the name of the referee*
email	*the e-mail address of the referee*
return_url	*the URL to return to once the user has accepted the T&C*

Description:

This call allows a user to refer a friend to Powershop. The promotion ID can be found from the results of the promotions GET call. A name and valid e-mail address must be specified.

Before a referral can be made, the logged in user must accept the terms and conditions of the promotion. They only need to accept the T&C once per promotion. If the user has not yet accepted the T&C, a result of "redirect" will be returned from the API, and the application must display the specified URL in an embedded browser, or redirect the user to the URL if in a web application. When the user has accepted the T&C, the referral POST request can be resubmitted.

Returns:

version = 2.0

result =

result	*"success" or "failure" or "redirect"*
message	*if result is "failure", a customer-readable description of the failure*
url	*if result is "redirect", a URL to display to the user*

Errors:

[E000] The promotion_id parameter must be specified for this call
[E000] The name parameter must be specified for this call
[E000] The email parameter must be specified for this call
[E011] Invalid e-mail address
[E012] Invalid promotion id

Notes:

If the result is "failure", the message field should be displayed to the user to inform them why the referral failed.