

# Swarm Administration API

Version 2 - 18/06/2014

The administration API is a part of the Swarm-Gateway API but only accessible by Swarm users. There are two roles

- **ROLE\_USER**, these users can integrate new POS systems into the gateway
- **ROLE\_ADMIN**, administrative privileges give access to the status page and its functionality

## User Service

The Swarm Administration API requires Basic Authentication, you can try whether user's session is already authenticated using the **GET /user** service. Like so:

```
~$ curl http://pos-gateway.swarm-mobile.com/swarm/api/user -i -u  
sonrisa:sonrisa2013
```

This may give the following responses:

- HTTP Unauthorized (401), wrong username or password
- HTTP Forbidden (403), access denied for user
- HTTP OK (200), user has at least **ROLE\_USER**.

Example:

```
{  
  "userName": "admin",  
  "screenName": "Sonrisa Admin",  
  "role": "ROLE_ADMIN"  
}
```

## Store Service

Get detailed status of stores, like the current pos\_status script implementation would.

### GET /admin/stores

Available parameters (GET parameters)

- (int, optional) **skip**: Result will be the query's skip, skip+1, ..., skip+take-1 items
- (int, optional) **take**: Result will be the query's skip, skip+1, ..., skip+take-1 items (maximum 10,000)
- (string, optional) **order\_by**: Values can be *store\_id*, *name*, *created* (default)
- (int, optional) **order\_dir**: Values can be *asc* or *desc* (default), specifies the order direction.
- (string, optional) **active**: Filter for stores with matching active value, possible values are *true* or

*false*

- (string, optional) **api**: Filter for stores with matching api name (shopify, lspro, etc.)
- (string, optional) **status**: Only stores matching status will be returned, either *OK*, *WARNING* or *ERROR*
- (string[], optional) **status**: Same as the one above, but with values separated by comma, and either matching is returned

Example:

```
{
  "stores": [
    {
      "store_id": 905,
      "name": "Taylor Stitch",
      "api": "shopify",
      "created": "2014-01-28 14:14:30",
      "active": true,
      "notes": "",
      "status": "OK",
      "details": {
        "last_invoice": "2014-01-28 14:14:30",
        "last_extract": "2014-01-28 16:14:30",
        "invoice_count": 48482
      }
    }
  ]
}
```

### PUT /admin/stores/{id}

Replaces store fields with content. Changing store\_id, api or created is not supported.

Example:

```
{
  "name": "Artsy Abode - Daytona Beach (StoreNo: DAY, SBS: 0)",
  "active": true,
  "notes": ""
}
```

### GET /admin/retailpro/stores

Available parameters (GET parameters):

- (string, optional) **swarm\_id**: Only stores with matching swarm\_id will be returned
- (int, optional) **skip**: Result will be the query's skip, skip+1, ..., skip+take-1 items
- (int, optional) **take**: Result will be the query's skip, skip+1, ..., skip+take-1 items
- (string, optional) **order\_by**: Values can be *swarm\_id*, *store\_id*, *name*, *created* (default)
- (int, optional) **order\_dir**: Values can be *asc* or *desc* (default), specifies the order direction.

- (string, optional) **api**: Filter for stores with matching api name (retailpro8, retailpro9, etc.)
- (string, optional) **status**: Only stores matching status will be returned, either *OK*, *WARNING* or *ERROR*
- (string[], optional) **status**: Same as the one above, but with values separated by comma, and either matching is returned

Example:

```
{
  "stores": [
    {
      "store_id": 1017,
      "name": "Artsy Abode - Daytona Beach (StoreNo: DAY, SBS: 0)",
      "api": "retailpro8",
      "created": "2014-02-28 14:14:30",
      "swarm_id": "artsydayton",
      "timezone": "US/East",
      "notes": "",
      "status": "OK",
      "details": {
        "last_invoice": "2014-01-28 14:14:30",
        "last_heartbeat": "2014-01-28 14:14:30",
        "invoice_count": 801,
        "client_version": "1.6.2.0"
      }
    },
    {
      "store_id": 931,
      "name": "Frye Chicago",
      "api": "retailpro9",
      "created": "2014-02-28 14:14:30",
      "swarm_id": "Frye-Production",
      "notes": "",
      "status": "WARNING",
      "reason": [
        "No invoice since 2014-06-03 09:19:20",
        "Timezone missing."
      ],
      "details": {
        "last_invoice": "2014-06-03 09:19:20",
        "last_heartbeat": "2014-06-18 12:01:52",
        "invoice_count": 2639,
        "client_version": "1.1.3.1"
      }
    }
  ]
}
```

**PUT /admin/retailpro/stores/{id}**

Replaces store fields with content. Changing store id, swarm id, api id or created is not supported.

Example:

```
{
  "name": "Artsy Abode - Daytona Beach (StoreNo: DAY, SBS: 0)",
  "timezone": "US/East",
  "notes": ""
}
```

## Client log download

This service enables Swarm administrators to download client log with severity level ERROR or higher.

**GET /admin/retailpro/clients**

```
{
  "clients": [
    {
      "swarm_id": "artsyabode",
      "log_level": "Warn",
      "most_severe": {
        "severity": "ERROR",
        "date": "2014-03-07 05:38:25.9773"
      }
    }
  ]
}
```

Or administrators may choose to download log files manually.

**GET /admin/retailpro/log/{swarmid}**

Available parameters (GET parameters):

- (string, optional) **date**: Get log file for specific date, date format: **YYYY-mm-dd**, if date is specified as today, unspecified or is any future date then the last available log file is returned

Possible results (401 and 403 responsible are possible as well with improper authentication):

- HTTP 404 Not found: If log file is not found for the specific date.
- HTTP 200 OK: With content type "text/txt" and content of the log file.

## Error responses

If the administrator user's credentials are accepted, but the request still can not be completed the server will send an error response like this.

**GET /admin/retailpro/stores?take=-123**

```
{  
  "errorType": "Invalid request",  
  "errorMessage": "Illegal value for take: -123"  
}
```