

Company API v1.4

The Company API provides access to venue, user and visit data stored in the portal at the company level, and exposes an API for unsubscribing users from emails.

Contents

1. Base URL
2. Requests
3. Responses
4. Venues
5. Visitors
6. Visits
7. Presence
8. Positioning
9. Unsubscribes
10. MicroSurveys
11. MicroSurvey responses
12. Feedback & Support

1. Base URL

It is implemented as JSON via SSL. An API version is included in the base URL:

```
https://DOMAIN.COM/api/company/v1
```

Where only DOMAIN.COM is replaced by the domain of your portal.

A new API version will be generated for any future changes that are not backward-compatible, at which point endpoints under the old version will be deprecated and eventually revoked.

2. Requests

Access to any API endpoint requires authentication via signed requests, created with public and secret API keys that can be requested by emailing support@purplewifi.net - please specify the company name(s) you want an API key to be generated for.

The signature is an HMAC-SHA256 hash of a string containing the request content-type, host, URL, date timestamp (matching the request Date header) and request content (eg POST parameters) separated by a single newline, and passed with the public key in a custom header X-API-Authorization along with the public key, in the format `PUBLICKEY:SIGNATURE`. The hash should be generated using the provided private key text string.

Example request

Given the following keys:

```
Public key: f1ad72cb01218548fa7e6431b2f17aad
Private key: 1244e4317311c81834fc788877324313
```

And a request to get visitors to venue 20131 between 1st January 2014 and 31st January 2014:

```
https://purpleportal.net/api/company/v1/venue/20131/visitors?from=20140101&to=20140131
```

A request would be signed using a string in this format, separated by newlines:

```
CONTENT-TYPE
DOMAIN NAME
URI INCLUDING GET PARAMETERS
DATE IN RFC2616 FORMAT
POST DATA STRING
```

NB: newlines can be entered as Windows (CRLF) or Unix (LF), the API will accept either.

So in our example, this would be:

```
application/json purpleportal.net
/api/company/v1/venue/20131/visitors?from=20140101&to=20140131
Mon, 17 Feb 2014 11:23:34 GMT
```

Note this is a GET request so there is no POST data/request content so the final line is empty, but should still be terminated by a newline, effectively meaning our string is terminated by two newlines.

Using the private key (as a text string), the HMAC-SHA256 hash of this string is:

```
4a29639d808235edf56c90fcb83e7830193b7fcad5d4aca009f0c52f6d460365
```

So the request would be passed with the following custom header:

```
X-API-Authorization:  
f1ad72cb01218548fa7e6431b2f17aad:4a29639d808235edf56c90fcb83e7830193b7fcad5d4aca009f  
0c52f6d460365
```

The full request headers would look like:

```
Host: purpleportal.net  
Accept: */*  
Content-Type: application/json  
Content-Length: 0  
Date: Mon, 17 Feb 2014 11:23:34 GMT  
X-API-Authorization:  
f1ad72cb01218548fa7e6431b2f17aad:4a29639d808235edf56c90fcb83e7830193b7fcad5d4aca009f  
0c52f6d4603651
```

3. Responses

All endpoints will respond with an HTTP status code:

```
200: Successful request  
401: Unauthorised, the API key is invalid  
403: Forbidden, the key does not have access to this endpoint/data  
404: The endpoint does not exist or no data is found for this endpoint  
410: This endpoint has been revoked  
422: The request had semantic errors or invalid parameters  
429: Too many requests, rate limit exceeded
```

```
500: Uncaught application error
```

Error responses also include a message and (where relevant) a list of failed parameters as an array.

Example error responses including status header

```
HTTP/1.1 404 Not Found
```

```
{
  "success":false,
  "response_code":404,
  "timestamp":"2014-02-01T08:26:30+00:00",
  "message":"Venue not found"
}
```

```
HTTP/1.1 401 Unauthorized
```

```
{
  "success":false,
  "response_code":401,
  "timestamp":"2014-02-01T08:36:30+00:00",
  "message":"API key is invalid" }
}
```

Successful response examples are included below.

4. Venues

Returns an array of venues for the company attached to the API key.

Request

```
/venues
/venue/[venue_id]
```

Example result

```
{
  "success": true,
  "timestamp": "2014-02-01T10:46:06+00:00",
  "response_code": 200,
  "data": {
    "venues": [
      {
        "id": 3865,
        "name": "Grand Hotel",
        "address1": "1 Hotel Street",
        "address2": "London",
        "town": null,
        "telephone": "0200 333 4455",
        "email": "grand@hotel.com",
        "timezone": "Europe/London",
        "facebook_id": null,
        "facebook_access": false,
        "twitter_id": null,
        "twitter_access": false,
        "linkedin_id": null,
        "linkedin_access": false,
        "last_polled": "2014-02-01T09:16:02+00:00",
        "users_online_now": 12,
        "users_online_24_hours": 32,
        "hardware": [
          {
            "brand": "TP-Link",
            "name": "WDR3500",
            "mac": "AA-AA-AA-FF-FF-FF",
            "last_polled": "2014-02-01T09:16:02+00:00"
          }
        ],
        "floors": [
          {
            "id": "21312",
            "name": "Ground floor",
            "zones": [
              {
                "id": "3231",
                "name": "Menswear",
              },
              {
                "id": "3232",
                "name": "Womenswear",
              }
            ]
          }
        ]
      }
    ]
  }
}
```


Example result

```
{
  "success": true,
  "timestamp": "2014-02-01T10:52:49+00:00",
  "response_code": 200,
  "data": {
    "visitors": [
      {
        "id": 291243,
        "first_name": "Jane",
        "last_name": "Smith",
        "gender": "F",
        "date_of_birth": "1984-01-01",
        "location": "Wakefield",
        "email": "jane.smith@email.com",
        "mobile": "+447711223344",
        "first_seen": "2013-10-11T11:01:49+00:00",
        "last_seen": "2014-01-07T08:18:02+00:00",
        "mac": "FF-FF-FF-FF-FF-FF",
        "visits": "13",
        "source": "Facebook"
      }
    ]
  }
}
```

6. Visits

Returns an array of visits made by an individual user, either a complete list of all visits if no dates are provided, or a list of visits between the optional date parameters `from` and `to`.

From and to parameters should be passed in the format `YYYYMMDD[HHMMSS]` in UTC.

Requests

```
/venue/[venue_id]/visitor/[user_id]
```

Optional GET Parameters

```
from: YYYYMMDD[HHMMSS]
to: YYYYMMDD[HHMMSS]
```

Example result

```
{
  "success": true,
  "timestamp": "2014-02-01T10:57:24+00:00",
  "response_code": 200,
  "data": {
    "visits": [
      {
        "login_datetime": "2014-01-07T03:02:06+0000",
        "post_asking": false,
        "post_agreed": false,
        "platform": "Android",
        "browser": "Android",
        "user_agent_string": "Mozilla/5.0 (Linux; U; Android 4.0.4;"
      }
    ]
  }
}
```

7. Presence

Returns an array of associated or unassociated clients seen by the venue APs where available, including enhanced visitor info for any recognised clients. Defaults to the last hour, or can accept optional date/time parameters `date`, or `from` and `to`.

The `date` parameter should be passed in the format `YYYYMMDD`.

The `from` and `to` parameters should be passed in the format `YYYYMMDDHHMMSS` in UTC.

If `date` is provided, exactly data will be returned for that date for a 24 hour period from midnight to midnight (UTC). More granular selections can be achieved by passing `from` and `to` instead, but queries are limited to a maximum of 24 hours. Any `from/to` that exceeds 24 hours will return an error.

NB: MAC addresses for anonymous devices are hashed using a SHA-256 algorithm with a salt unique to Purple, so MACs are still uniquely identifiable at the same company (for identifying repeat visitors/devices), but not identifiable against third party data sources.

Requests

```
/venue/[venue_id]/presence
```

Optional GET Parameters

```
date: YYYYMMDD
from:
YYYYMMDDHHMMSS to:
YYYYMMDDHHMMSS
```

Example result

```
{
  "success": true,
  "timestamp": "2015-08-03T10:58:12+00:00",
  "response_code": 200,
```

```
"data": {
  "presence": [
    {
      {
        "client_mac": "3c6313ba2cb72c0f4a2af1149d15...",
        "start": "2016-08-02T22:12:55+00:00",
        "end": "2016-08-02T22:12:55+00:00",
        "duration": "94",
        "rssi_min": "2",
        "rssi_max": "2",
        "vendor": "Apple"
      },
      {
        "client_mac": "329bd1e78ce7996a6eddc19f545...",
        "start": "2015-08-02T22:10:07+00:00",
        "end": "2015-08-02T22:13:30+00:00",
        "duration": "203",
        "rssi_min": "5",
        "rssi_max": "13",
        "vendor": "RIM"
      },
      {
        "client_mac": "d229d1ecd5689924e8591242eff...",
        "start": "2015-08-02T22:16:29+00:00",
        "end": "2015-08-02T22:16:30+00:00",
        "duration": "181",
        "rssi_min": "11",
        "rssi_max": "92",
        "vendor": "Samsung",
        "gender": "M",
        "age": 38,
        "user_id": 99261,
        "first_name": "Stig",
        "last_name": "Jones",
        "date_of_birth": "1976-10-01",
        "location": "Lincoln",
        "email": "stig.jones@email.com",
        "mobile": null,
        "mobile_validated": "+441234123123",
        "first_seen": "2013-10-19T17:21:23+00:00",
        "last_seen": "2015-09-11T14:10:22+00:00",
        "visits": "68",
        "source": "Facebook",
        "facebook_id": "12345678"
      }
    ]
  }
}
```

```

    ]
  }
}

```

8. Positioning

Returns an array of associated or unassociated clients seen in a venue and a list of X/Y coordinates for each client including a floor and matching zones. Displays additional data for recognised clients (see: Presence). Returns exactly one hour of data, either the previous hour or an hour from a requested start time.

The `from` parameter should be passed in the format `YYYYMMDDHHMMSS` in UTC.

NB: MAC addresses for anonymous devices are hashed using a SHA-256 algorithm with a consistent Purple salt, so MACs are still uniquely identifiable at the same company (for identifying repeat visitors/devices), but not identifiable against third party datasets.

Requests

```
/venue/[venue_id]/positioning
```

Optional GET Parameters

```
from: YYYYMMDDHHMMSS
```

Example result

```

{
  "success": true,
  "timestamp": "2016-07-28T10:54:35+00:00",
  "response_code": 200,
  "data": {
    "positioning": {
      "2b8c36e1f149d26e749ab418ba69f42f56005e87f21396f9546d050b6e05da68": {
        "zones": [

```

```
{
  {
    "id": "23932",
    "name": "Zone A",
    "start": "2016-07-28T09:54:34+00:00",
    "end": "2016-07-28T09:54:34+00:00",
    "duration": 129
  },
  {
    "id": "44302",
    "name": "Zone B",
    "start": "2016-07-28T08:54:34+00:00",
    "end": "2016-07-28T08:54:34+00:00",
    "duration": 94
  }
],
"pings": [
  {
    "x": 0.79429,
    "y": 0.79197,
    "seen": "2016-07-28T09:54:34+00:00"
  },
  {
    "x": 0.48688,
    "y": 0.6173,
    "seen": "2016-07-28T09:54:35+00:00"
  },
  {
    "x": 0.41814,
    "y": 0.56182,
    "seen": "2016-07-28T09:54:36+00:00"
  },
  {
    "x": 0.65272,
    "y": 0.7099,
    "seen": "2016-07-28T10:23:01+00:00"
  }
],
"data": {
  "mac": "2b8c36e1f149d26e749ab418ba69f4...",
  "user": "8482819",
  "name": "Jane Smith",
  "gender": "F",
  "age": "41",
  "networks": "Facebook,Form",
  "vendor": "Apple",
  "visits": 1
}
}
```

```
}
}
```

9. Unsubscribes

You can unsubscribe a user from receiving future emails from the portal if they have been unsubscribed from other CRMs the company is using.

Request

```
/venue/[venue_id]/visitor/[user_id]/unsubscribe
```

Example result

```
{
  "success": true,
  "timestamp": "2014-02-01T10:57:24+00:00",
  "response_code": 200,
  "message": "User was successfully unsubscribed",
}
```

You can also return an array of users that have unsubscribed (opted out) from the portal so you can update your own CRM. If no dates are provided, all records are returned, else a list of visits between the optional date parameters `from` and `to` are returned.

From and to parameters should be passed in the format `YYYYMMDD[HHMMSS]` in UTC.

Requests

```
/unsubscribes
```

Optional GET Parameters

```
from: YYYYMMDD[HHMMSS]
to: YYYYMMDD[HHMMSS]
```

Example result

```
{
  "success": true,
  "timestamp": "2014-02-01T10:57:24+00:00",
```

```
"response_code": 200,
"data": {
  "unsubscribes": [
    {
      "email": "jane.smith@email.com",
      "source": "website",
      "date_created": "2017-04-24T10:46:05+00:00"
    }
  ]
}
```

10. MicroSurveys

Return a complete list of available published MicroSurveys for the company.

Request

```
/microsurveys
```

Example result

```
{
  "success": true,
  "timestamp": "2017-04-26T10:36:48+00:00",
  "response_code": 200,
  "data": {
    "surveys": [
      {
        "id": "54164",
        "name": "How are we doing?",
        "created_at": "2016-12-02T11:43:01+00:00",
        "uniqid": "ms-58415e45cb940"
      }
    ]
  }
}
```

11. MicroSurvey Responses

Returns a list of MicroSurvey responses for a specific MicroSurvey, using a MicroSurvey ID retrieved from the request above.

Request

```
/microsurveys/[microsurvey_id]
```

Example result

```
{
  "success": true,
  "timestamp": "2017-04-26T10:42:08+00:00",
  "response_code": 200,
  "data": {
    "responses": [
      {
        "answers": [
          {
            "question": "How likely are you to recommend this venue?",
            "answer": "5",
            "type": "rating",
            "questionNumber": 1
          },
          {
            "question": "What is the most important reason?",
            "answer": "Great music",
            "type": "textbox",
            "questionNumber": 2
          }
        ],
        "responseDate": "2017-04-24T10:46:05+00:00",
        "venue": 992978,
        "user": 96842798
      }
    ]
  }
}
```

12. Feedback & Support

Any feedback or support queries can be sent to support@purplewifi.net