Company API

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

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

```
https://portal.btwifi.com/api/company/v1
```

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 presence with public and secret API keys that can be requested by emailing your BT Wi-Fi account manager - please specify the company name(s) you want an API key to be generated for. The account manager will then contact purplesupport@bt.com

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 (e.g. 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://portal.btwifi.com/api/company/v1/venue/20131/visitors?from=20140101&to=201401 31
```

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
portal.btwifi.com
/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: portal.btwifi.com
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",
                            }
                        ]
                    }
                ]
            }
        ]
    }
}
```

5. Visitors

Returns an array of visitors to a venue either online now if no dates are provided, or between the optional date parameters provided as from and to as GET parameters.

Dates should be passed in the format YYYYMMDD[HHMMSS]in UTC.

NB: On request, MAC addresses for visitors can be forcibly hashed when requested by API. They will be hashed using a SHA-256 algorithm with a consistent salt, so MACs are still uniquely identifiable at the same company (for identifying repeat visitors/devices), but not identifiable against third party datasets. An example hashed MAC address may look like:

```
"mac": "fd589281c8bfe7d516431ef9d2929bc6daeccd9fdfed00ad759b6cdcdc155575    "
```

Requests

```
/venue/[venue_id]/visitors
```

Optional GET Parameters

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

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
            "terms_signed": [
               {
                  "document": "terms_cq-company_12345",
                  "dateSigned": "2017-11-06 10:17:53",
                  "locale": "en_GB",
                  "venue": 1
               }
            ]
         }
      ]
    }
}
```

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_asked": false,
            "post_agreed": false,
            "platform": "Android",
            "browser": "Android",
            "user_agent_string": "Mozilla\/5.0 (Linux; U; Android 4.0.4;"
         }
      ]
   }
}
```

7. 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"
            }
        ]
    }
}
```

8. 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"
            }
        ]
    }
}
```

9. 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
            }
        ]
    }
}
```

10. Terms accepted

Returns the exact terms a user accepted. The terms version string and locale is returned as the document and locale fields in the terms_signed array in the visitor results (see 5).

Note: this information is only available for logins from December 1st 2017.

Request

```
/terms/[terms_version]/?locale=[locale]
```

For example, in the example in section 5 the request URI would be:

```
/terms/terms_cq-company_12345?locale=en_GB
```

Example result

```
{
   "success": true,
   "timestamp": "2017-04-26T10:42:08+00:00",
   "response_code": 200,
   "data": {
      "version": "5.0",
      "content": "Terms text here"
   }
}
```

11. Feedback & Support

Any feedback or support queries can be sent to purplesupport@bt.com.