

Table of Contents

| | |
|---|----|
| Table of Contents | 1 |
| Weezevent API | 3 |
| Fair use | 3 |
| API Base URL | 3 |
| API Methods Overview | 3 |
| POST /auth/access_token | 4 |
| Resource URL | 4 |
| Parameters | 4 |
| Example Request | 4 |
| Example Response | 4 |
| GET /events | 5 |
| Resource URL | 5 |
| Parameters | 5 |
| Example Request | 5 |
| Example Response | 5 |
| GET /dates | 6 |
| Resource URL | 6 |
| Parameters | 6 |
| Example Request | 6 |
| Example Response | 6 |
| GET /tickets | 7 |
| Resource URL | 7 |
| Parameters | 7 |
| Example Request | 7 |
| Example Response | 7 |
| GET /ticket/:id/stats | 10 |
| Resource URL | 10 |
| Parameters | 10 |
| Example Request | 10 |
| Example Response | 10 |
| GET /participant/list | 11 |
| Resource URL | 11 |
| Parameters | 11 |
| Example Request | 11 |
| (by id_event, full data) | 11 |
| (by id_event and transaction_reference) | 11 |
| (by id_ticket) | 12 |
| (by date_ticket) | 12 |
| (by date_ticket and id_ticket) | 12 |
| Example Response | 12 |
| GET /participant/id/answers | 14 |
| Resource URL | 14 |
| Parameters | 14 |
| Example Request | 14 |
| Example Response | 14 |
| GET /scan/settings | 15 |
| Resource URL | 15 |
| Parameters | 15 |
| Example Request | 15 |
| Example Response | 15 |
| POST /scan/user | 16 |
| Resource URL | 16 |
| Parameters | 16 |
| Example Request | 16 |
| Example Response | 16 |
| GET /event/:id/details/ | 17 |
| Resource URL | 17 |
| Parameters | 17 |
| Example Request | 17 |
| Example Response | 17 |
| GET /event/search/ | 19 |
| Resource URL | 19 |
| Parameters | 19 |
| Example Request | 19 |
| Example Response | 19 |
| GET/POST /event/categories/ | 21 |

| | |
|------------------|----|
| Resource URL | 21 |
| Example Request | 21 |
| Example Response | 21 |

Weezevent API

The Weezevent API gives developers of third party applications access to the database and basic functionalities of the Weezevent Ticketing solution.

Access to the API is limited to registered partners of Weezevent. All requests to the API are signed with an identifier string (API key) which is unique for each partner. Please contact Weezevent if you want to obtain an API key for your application.

The API is built on HTTP and is RESTful. By default, all data is returned in JSON format.

HTTPS is required for all requests.

Fair use

The API is intended to be called by another application which handles the data for its own purpose. Therefore, data retrieved should be stored and queries should be filtered to limit API load. Some high limits are enforced to ensure global system safety and we monitor uses. We monitor API calls and we could limit access or even ban any API access key which do not make a fair use of the API or compromise Weezevent platform.

API Base URL

<https://api.weezevent.com/>

API Methods Overview

| URL | action |
|---|--|
| /auth/access_token | Authenticate the user and return an access token |
| /events | List of events to which the user has access |
| /dates | List of dates for one or multiple events |
| /tickets | List of price categories for one or multiple events |
| /ticket/:id/stats | Return the scan statistic for one price category |
| /participant/list | Return a list of participants |
| /participant:id/answers | Return the inscription form answers of one participant |
| /scan/settings | Update the settings of the mobile control application |
| /scan/user | Provide a specific name for a control point |
| /event:id/details | Return details for specified event |
| /event/search/ | Search for specific events |
| /event/categories/ | List of event categories |

☁ POST /auth/access_token

Authenticates a Weezevent user account and returns an access token. Valid user/password combinations are either organizer or partner accounts which were created via the Weezevent Ticketing Platform.

The returned access token will be used to validate all upcoming requests. An access token is persistent. This way, the username and password needs to be transferred just once and the third party application does not need to store the username and password locally.

Resource URL

https://api.weezevent.com/auth/access_token

Parameters

| Parameter | Explanation |
|-----------|--|
| username | String - required: Weezevent organizer or partner login |
| password | String - required: Weezevent organizer or partner password |
| api_key | String - required: Your API key |

Example Request

```
POST https://api.weezevent.com/auth/access_token HTTP/1.0
content-type:application/x-www-form-urlencoded;charset=utf-8
```

```
&username=_myusername_
&password=_password_
&api_key=_monapikey_
```

Example Response

```
{
  "accessToken" : "f4b6b917cee00f13d57e3c84f56656b9"
}
```

☁ GET /events

Returns all events to which the current user (identified by the access token) has access. By default, only published events with at least one participant will be returned.

Resource URL

<https://api.weezevent.com/events>

Parameters

| Parameter | Explanation |
|-----------------------|--|
| access_token | String - required: API access token |
| api_key | String - required: Your API key |
| include_not_published | Boolean - optional : Whether or not to include events which are not published. Default is true. |
| include_closed | Boolean - optional : Whether or not to include events which are closed. Default is false. |
| include_without_sales | Boolean - optional : Whether or not to include events did not sell any tickets so far. Default is false. |

Example Request

```
GET https://api.weezevent.com/events?api_key=monapikey&access_token=f4b6b917cee00f13d57e3c84f56656b9 HTTP/1.0
```

Example Response

```
{
  "events": [
    {
      "id": 13824,
      "name": "Event 1",
      "date": {
        "start": "2011-10-10 00:00:00",
        "end": ""
      },
      "participants": 0,
      "multiple_dates": 0,
      "sales_status": {
        "id_status": 1,
        "libelle_status": "Vente en cours"
      }
    },
    {
      "id": 11435,
      "name": "Event 2",
      "date": {
        "start": "2011-11-27 00:00:00",
        "end": ""
      },
      "participants": 24,
      "multiple_dates": 1,
      "sales_status": {
        "id_status": 2,
        "libelle_status": "Bientôt en vente"
      }
    }
  ]
}
```

🌩 GET /dates

Events can have either one single or multiple dates. This is indicated in the response of `get/events` with the variable `multiple_dates`. You can obtain a list of dates for these events (`multiple_events=1`) with this function. Each date is associated with multiple tickets (price categories).

Resource URL

<https://api.weezevent.com/dates>

Parameters

| Parameter | Explanation |
|-----------------------------|--|
| <code>display_passed</code> | bool - Should we return passed dates. Default false. |
| <code>access_token</code> | String - required: API access token |
| <code>api_key</code> | String - required: Your API key |
| <code>id_event</code> | String or Array - required: Id of event |

Example Request

```
GET https://api.weezevent.com/dates?api_key=monapikey&access_token=f4b6b917cee00f13d57e3c84f56656b9&id_event[]=11435&id_event[]=10473 HTTP/1.0
```

Example Response

```
{
  "dates": [
    {
      "id": 138,
      "id_event": 11435,
      "date": "2013-01-10 14:20:00",
      "tickets": ["75632", "75633", "75634"]
    },
    {
      "id": 139,
      "id_event": 11435,
      "date": "2013-01-10 16:20:00",
      "tickets": ["75633", "75634"]
    },
    {
      "id": 116,
      "id_event": 10473,
      "date": "2013-02-25 11:55:00",
      "tickets": ["75632", "75633"]
    }
  ]
}
```

☁ GET /tickets

This function returns a list of tickets associated with one or multiple events. Tickets represent different price categories (e.g. Tarif Adult, Tarif Enfant) for an event. Tickets can be nested into categories and sub-categories. The id of a ticket is most of the time an integer value, however in the case of external tickets (distributors) the id can be a string.

Resource URL

<https://api.weezevent.com/tickets>

Parameters

| Parameter | Explanation |
|--------------|---|
| access_token | String - required: API access token |
| api_key | String - required: Your API key |
| id_event | String or Array - required: Id of event |

Example Request

```
GET https://api.weezevent.com/tickets?api_key=monapikey&access_token=f4b6b917cee00f13d57e3c84f56656b9&id_event[]=10473 HTTP/1.0
```

Example Response

```
{
  "events": [
    {
      "id": 11435,
      "name": "Event 1",
      "tickets": [
        {
          "id": "75632",
          "name": "my default price",
          "price": 10,
          "participants": 110,
          "quota": 400,
          "start_sale": null,
          "end_sale": "2013-01-29T13:00:00+0000"
        },
        {
          "id": "ABCDE125632",
          "name": "Tarif 2",
          "price": 12,
          "participants": 800,
          "quota": 1200,
          "start_sale": "2013-01-23T11:00:00+0000",
          "end_sale": "2013-01-29T13:00:00+0000"
        }
      ]
    },
    {
      "id": "11745",
      "name": "Categorie 1",
      "tickets": [
        {
          "id": "44272",
          "name": "b2",
          "price": 9.99,
          "participants": 99
        }
      ]
    }
  ]
}
```

```

    participants": 0,
    "quota": 0,
    "start_sale": "2013-01-23T11:00:00+0000",
    "end_sale": "2013-01-29T13:00:00+0000"
  }
],
"categories": [
  {
    "id": "11746",
    "name": "Sous Categorie 1",
    "tickets": [
      {
        "id": "65632",
        "name": "b3",
        "price": 0,
        "participants": 210,
        "quota": 0,
        "start_sale": "2013-01-23T11:00:00+0000",
        "end_sale": "2013-01-29T13:00:00+0000"
      },
      {
        "id": "65638",
        "name": "b4",
        "price": 10,
        "participants": 190,
        "quota": 0,
        "start_sale": "2013-01-23T11:00:00+0000",
        "end_sale": "2013-01-29T13:00:00+0000"
      }
    ]
  }
]
}
]
},
{
  "id": 10473,
  "name": "Event 2",
  "tickets": [
    {
      "id": "40413",
      "name": "Billet 1",
      "price": 10,
      "participants": 0,
      "quota": 100,
      "start_sale": "2013-01-23T11:00:00+0000",
      "end_sale": "2013-01-29T13:00:00+0000"
    }
  ]
},
"categories": [
  {
    "id": 4654,
    "name": "Categorie 1",
    "tickets": [
      {
        "id": 42637,
        "name": "Billet 2",
        "price": 0,
        "start_sale": "2013-01-23T11:00:00+0000",
        "end_sale": "2013-01-29T13:00:00+0000"
      }
    ]
  },
  {
    "id": "4655",
    "name": "Sous categorie 1",
    "tickets": [
      {
        "id": "42638",
        "name": "Billet 3",
        "price": 10,
        "participants": 110,

```



```
        "quota": 110,  
        "start_sale": "2013-01-23T11:00:00+0000",  
        "end_sale": "2013-01-29T13:00:00+0000"  
    }  
  ]  
}  
],  
{  
  "id": "5998",  
  "name": "Categorie 2",  
  "tickets": [  
    {  
      "id": "42639",  
      "name": "Billet 4",  
      "price": 10,  
      "participants": 1011,  
      "quota": 1100,  
      "start_sale": "2013-01-23T11:00:00+0000",  
      "end_sale": "2013-01-29T13:00:00+0000"  
    }  
  ]  
}  
]  
}  
]  
}
```

GET /ticket/:id/stats

This function is solely used for access control scenarios. It returns the scan statistic for one price category.

Resource URL

<https://api.weezevent.com/ticket/:id/stats>

Parameters

| Parameter | Explanation |
|--------------|-------------------------------------|
| access_token | String - required: API access token |
| api_key | String - required: Your API key |
| id_date | String - optional: id of date |

Example Request

```
GET https://api.weezevent.com/ticket/44272/stats?api_key=monapikey&access_token=f4b6b917cee00f13d57e3c84f56656b9&id_date=138 HTTP/1.0
```

Example Response

```
{
  "stats": {
    "total": 23,
    "scanned": 18,
    "in": 18,
    "out": 0
  }
}
```

🔒 GET /participant/list

This function returns a list of participants for one or multiple events, one or multiple tickets or ticket/date combinations. Depending on the type of inscription form used during the order process, either the buyer or the participant himself is indicated as "owner".

Resource URL

<https://api.weezevent.com/participant/list>

Parameters

| Parameter | Explanation |
|-----------------------|---|
| access_token | String - required: API access token |
| api_key | String - required: Your API key |
| id_event | Array - optional: id of event |
| id_ticket | Array - optional: id of price category |
| date_ticket | Array - optional: associative array of id_date/id_ticket tuples |
| last_update | Date (YYYY-MM-DD HH:MM:SS) - Optional: If this parameter is given, the method will return only participants modified after this date. This parameter is mainly used in an access control scenario. |
| last_update_before | Date (YYYY-MM-DD HH:MM:SS) - Optional: If this parameter is given, the method will return only participants modified before this date. This parameter is mainly used in an access control scenario. |
| include_deleted | Boolean - optional: by default set to false. Indicates whether or not deleted participants are included in the results. |
| moderation | Boolean - optional: by default set to false. Indicates whether or not the moderation status of a participant is included in the results (Beta). |
| include_unpaid | Boolean - optional: by default set to false. Indicates whether or not unpaid participants are included in the results. |
| full | Boolean - optional : If set to true, all inscription form answers will be included in the response. Default is false. |
| minimized | Boolean - optional : Reduces the amount of transferred data by shortening the labels of the response. Default is false. |
| return_count | Boolean - optional : return number of active and deleted participants |
| return_count_total | Boolean - optional : return total number of participants |
| return_removed | Boolean - optional : return only deleted participants |
| max | int - optional : max result by page |
| page | int - optional : current page of the response |
| transaction_reference | Array - optional: reference of the transaction made by the participant |

Example Request

(by id_event, full data)

```
GET https://api.weezevent.com/participant/list?api_key=monapikey&access_token=58708763b57ee89003fe86e6a2ade958&id_event[]=11122&id_event[]=112331&full=1 HTTP/1.0
```

(by id_event and transaction_reference)

```
GET https://api.weezevent.com/participant/list?api_key=monapikey&access_token=58708763b57ee89003fe86e6a2ade958&id_event[]=1122&id_event[]=112331&transaction_reference[]=C15E265226O63029&transaction_reference[]=C4E265226O63029 HTTP/1.0
```

#

(by id_ticket)

```
GET https://api.weezevent.com/participant/list?api_key=monapikey&access_token=58708763b57ee89003fe86e6a2ade958&id_ticket[]=63049045c39e21cc8d8e358def382cbb&id_ticket[]=1121223 HTTP/1.0
```

(by date_ticket)

```
GET https://api.weezevent.com/participant/list?api_key=monapikey&access_token=58708763b57ee89003fe86e6a2ade958&date_ticket[138][]=44272&date_ticket[138][]=44274 HTTP/1.0
```

(by date_ticket and id_ticket)

```
GET https://api.weezevent.com/participant/list?api_key=monapikey&access_token=58708763b57ee89003fe86e6a2ade958&date_ticket[138][]=44272&date_ticket[138][]=44274&id_ticket[]=63049045c39e21cc8d8e358def382cbb&id_ticket[]=1121212 HTTP/1.0
```

Example Response

```
{
  "participants": [
    {
      "id_participant": 7759645,
      "id_event": 49015,
      "id_date": 0,
      "id_ticket": "198251",
      "barcode": "7448370",
      "id_weez_ticket": "T6E49015O5822",
      "deleted": "0",
      "refund": "",
      "comment": "",
      "create_date": "2014-01-23 12:41:55",
      "origin": "saisir",
      "id_transaction": 2029446,
      "paid": true,
      "code_member": "CODEMEMBER",
      "transaction_reference": "C15E265226O63029",
      "control_status": {
        "status": "0",
        "scan_date": "0000-00-00 00:00:00",
        "scan_user": "0",
        "scan_user_name": ""
      },
      "owner": {
        "first_name": "",
        "last_name": "",
        "email": "",
        "comment": "",
        "custom_field": "",
        "origin": "saisir"
      },
      "upselling": [
        {
          "barcode": "33212392186194",
          "category": {
            "id": 1,
            "name": "CASHLESS"
          },
          "product": {
            "id": 1131,
            "name": "Préchargement cashless",
            "reference": "PC0001"
          },
          "sub_product": {
```

```

        "id": 135131,
        "name": "Préchargement de 10.50€",
        "reference": "PCP001",
        "price": 10.50
    },
    "quantity": 1,
    "paid_price": 10.50,
    "deleted": false
},
{
    "barcode": "33212392186195",
    "category": {
        "id": 1,
        "name": "CASHLESS"
    },
    "product": {
        "id": 1131,
        "name": "Préchargement cashless",
        "reference": "PC0001"
    },
    "sub_product": {
        "id": 135132,
        "name": "Préchargement de 15.00€",
        "reference": "PCP002",
        "price": 15
    },
    "quantity": 1,
    "paid_price": 15,
    "deleted": false
}
]
},
{
    "id_participant": 7759646,
    "id_event": 49015,
    "id_date": 0,
    "id_ticket": "198251",
    "barcode": "7448371",
    "id_weez_ticket": "T7E49015O5822",
    "deleted": "0",
    "refund": "1",
    "comment": "",
    "create_date": "2014-01-23 12:41:55",
    "origin": "saisir",
    "id_transaction": 2029446,
    "paid": true,
    "code_member": "CODEMEMBER",
    "transaction_reference": "C4E265226O63029",
    "control_status": {
        "status": "0",
        "scan_date": "0000-00-00 00:00:00",
        "scan_user": "0",
        "scan_user_name": ""
    },
    "owner": {
        "first_name": "",
        "last_name": "",
        "email": "",
        "comment": "",
        "custom_field": "",
        "origin": "saisir"
    }
}
],
"server_time": "2014-01-23 12:52:19"
}

```

📄 GET /participant/:id/answers

This function returns an array of question/answer data for one participant. If no data was collected for this participant, the answers of the buyer are returned.

Resource URL

<https://api.weezevent.com/participant/:id/answers>

Parameters

| Parameter | Explanation |
|--------------|-------------------------------------|
| access_token | String - required: API access token |
| api_key | String - required: Your API key |
| id | String - required: Participant id |

Example Request

```
GET https://api.weezevent.com/participant/2291148/answers?api_key=monapikey&access_token=58708763b57ee89003fe86e6a2ade958 HTTP/1.0
```

Example Response

```
{
  "answers": [
    {
      "label": "Etes-vous accompagn\u00e9 par un enfant de - de 12 ans sur votre v\u00e9lo (si\u00e9ge v\u00e9lo enfant ou charrette)?",
      "value": "Oui"
    },
    {
      "label": "Si oui, merci de renseigner les coordonn\u00e9es de l'enfant de - de 12 ans",
      "value": "Nom"
    }
  ]
}
```

GET /scan/settings

This function is solely used for access control applications. For each user account, the settings of the client application can be stored on the Weezevent server.

Resource URL

<https://api.weezevent.com/scan/settings>

Parameters

| Parameter | Explanation |
|--------------|-------------------------------------|
| access_token | String - required: API access token |
| api_key | String - required: Your API key |

Example Request

```
GET https://api.weezevent.com/api/v2/scan/settings?api_key=monapikey&access_token=58708763b57ee89003fe86e6a2ade958 HTTP/1.0
```

Example Response

```
{
  "settings": {
    "test": "test",
    "sync_interval": "120",
    "scan_device": "",
    "scan_focus": "0",
    "scan_mode": "0",
    "scan_vibration": "0",
    "scan_sound": "0"
  }
}
```

☁ POST /scan/user

This function is solely used for access control applications. Each client application (identified by `access_token`) can give itself a custom name which will be used to identify the control station.

Resource URL

<https://api.weezevent.com/scan/user>

Parameters

| Parameter | Explanation |
|---------------------------|-------------------------------------|
| <code>access_token</code> | String - required: API access token |
| <code>api_key</code> | String - required: Your API key |
| <code>user</code> | String - required: Name of the user |

Example Request

```
POST https://api.weezevent.com/scan/user HTTP/1.0
content-type:application/x-www-form-urlencoded;charset=utf-8

&api_key=toto
&access_token=58708763b57ee89003fe86e6a2ade958
&user=foo
```

Example Response

```
{
  "id": "1789"
}
```


☁ GET /event/:id/details/

This resource is used for retrieving details for specific event. It returns an array with all details.

Resource URL

<https://api.weezevent.com/event/:id/details/>

Parameters

| Parameter | Explanation |
|--------------|-------------------------------------|
| access_token | String - required: API access token |
| api_key | String - required: Your API key |
| :id | Integer - required: Event ID |

Example Request

```
GET https://api.weezevent.com/event/36799/details?api_key=monapikey&access_token=58708763b57ee89003fe86e6a2ade958 HTTP/1.0
```

Example Response

```
{
  "last_update": "2013-05-27T18:02:16+02:00",
  "events": {
    "id": "36799",
    "title": "API API API",
    "site_url": "",
    "description": "",
    "image": "http://weezevent.dev/uploads/evenement_affiche/affiche_36799.png?1368811166",
    "category": {
      "id": "14",
      "name": "Soir\u00e9e clubbing"
    },
    "period": {
      "start": "2013-06-27 09:00:00",
      "end": "2013-06-30 09:00:59"
    },
    "price": {
      "min": "142.00",
      "max": "142.00"
    },
    "venue": {
      "name": "Bureau",
      "address": "110 rue des Poissonniers",
      "zip_code": "75018",
      "city": "Paris",
      "state": "",
      "country": {
        "code": "FR",
        "name": "FRANCE"
      },
      "coordinates": {
        "long": "0.000000",
        "lat": "0.000000"
      }
    },
    "extras": {
      "minisite_url": "http://weezevent.dev/api-api-api"
    }
  }
}
```

```
    "sale_period": {
      "start": "2013-06-27 09:00:00",
      "end": "2013-06-30 09:00:00"
    },
    "dates": [
      {
        "start": "2013-06-27 09:00:00",
        "end": ""
      },
      {
        "start": "2013-06-30 09:00:00",
        "end": ""
      }
    ],
    "organizer": {
      "organization_name": "",
      "first_name": "Savic",
      "last_name": "Yasmina",
      "location": {
        "address": "110 rue des Poissonniers",
        "zip_code": "H1Y 2G5",
        "city": "Montr\u00e9al",
        "state": "",
        "country": {
          "code": "CA",
          "name": "CANADA"
        }
      },
      "phone": "0141414141",
      "cell_phone": "",
      "fax": "",
      "email": "y.savic@weezevent.com",
      "website": "",
      "blog": ""
    }
  }
}
```

GET /event/search/

This resource is used for retrieving events depending on some criteria. It returns an array with all matching events.

Resource URL

[https://api.weezevent.com/event/search/\(full/\)](https://api.weezevent.com/event/search/(full/))

Parameters

| Parameter | Explanation |
|--------------|---|
| access_token | String - required: API access token |
| api_key | String - required: Your API key |
| date | Date - optional: Retrieve events for unique date |
| date_start | Date - optional : Retrieve events in a period |
| date_end | Date - optional : Retrieve events in a period |
| category | Integer - optional : Search by category ID |
| city | String - optional: Search by city name |
| zip_code | String - optional: Search by postal code |
| country | String - optional: Search by country (ISO code) |
| province | String - optional: Search by province |
| organizer | String - optional: Search by an organizer name |
| max_result | Integer - optional: Number of maximal event to find |

Example Request

```
GET https://api.weezevent.com/event/search/?api_key=monapikey&access_token=58708763b57ee89003fe86e6a2ade958&date_start=2013-06-01&date_end=2013-06-30&category=13&max_result=1 HTTP/1.0
```

Example Response

```

{
  "last_update": "2013-05-27T17:56:50+02:00",
  "count": 1,
  "events": [
    {
      "id": "19988",
      "title": "Inscription NODE 2012/2013",
      "site_url": "http://bxno.de",
      "description": "\u003Cp\u003E\r\n\t\u003Cstrong\u003E Cette inscription donne acc\u00e8s au NODE jusqu'\u0026#39au 31 d\u0000e9cembre 2013.\u003C/strong\u003E\u003Cp\u003E\r\n\t\u003Cp\u003E\r\n\t\u003Cp\u003E\r\n\t\u003Cem\u003E (les appels \u00e0 cotisation pour 2014 seront lanc\u00e9 le 1er juillet 2013.)\u003C/em\u003E\u003Cp\u003E\r\n",
      "image": "",
      "category": {
        "id": "38",
        "name": "Dons en ligne"
      },
      "period": {
        "start": "2012-07-20 00:00:00",
        "end": "2013-06-30 23:55:59"
      },
      "price": {
        "min": "0.00",
        "max": "1000.00"
      },
      "venue": {
        "name": "",
        "address": "",
        "zip_code": "",
        "city": "",
        "state": "",
        "country": {
          "code": null,
          "name": null
        },
        "coordinates": {
          "long": "0.000000",
          "lat": "0.000000"
        }
      },
      "extras": {
        "minisite_url": "http://weezevent.dev/inscription-node-2012-2013"
      }
    }
  ]
}

```

GET/POST /event/categories/

This resource is used for retrieving our event categories.

Access for this list is public. So no need for API key and access token.

Resource URL

<https://api.weezevent.com/event/categories/>

Example Request

```
GET https://api.weezevent.com/event/categories/ HTTP/1.0
```

Example Response

```
{
  "last_update": "2013-05-24T16:02:27+02:00",
  "event_categories": [
    {
      "id": "14",
      "name": "Soirée clubbing",
      "name_other": "Soirée clubbing"
    },
    {
      "etc": "..."
    }
  ]
}
```

[Back to Top](#)