

Hermes Events v1.1

Thank you for downloading Hermes Events, a ColdFusion Package developed to interact with the Eventbrite Online Event API.

For information on the latest releases, please visit <http://www.fuzzyorange.co.uk>

Hermes Events v1.1	1
License and Credits	2
Authors.....	2
Requirements	2
Installation.....	3
Getting Started	3
Instantiation.....	4
Authentication.....	4
Secure.....	4
Basic.....	5
Response Formats	5
Available Methods	6
Limitations	15
Testing.....	15

License and Credits

Copyright 2010 Fuzzy Orange Ltd (<http://www.fuzzyorange.co.uk>)

Licensed under the Apache License, Version 2.0 (the "License");

You may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

For full License details, please read the LICENSE file available with this download.

Authors

Developed by Matt Gifford AKA coldfumonkeh

Senior Developer, Fuzzy Orange Ltd

<http://www.fuzzyorange.co.uk>

<http://www.mattgifford.co.uk>

Requirements

Hermes Events requires ColdFusion MX7

Installation

Unzip the package archive to the desired location (typically in the web root). After installation, you will see the following directories:

- Installation (contains this installation guide, no more, no less)
- com (contains all of the ColdFusion Components required for the Hermes Application to interact with the Eventbrite API)

The application does not interact with any database, and as such needs no datasources.

Getting Started

To interact with the Eventbrite API, you should ideally have an account with Eventbrite.

Visit <https://www.eventbrite.com/signup> for more information and to register a free account.

Once you have an account, you need to obtain your application API Key, typically a 16 character unique key, which will let you access the API. The application key is a string assigned by Eventbrite which identifies the application making the method call.

You can request your application key here: <http://www.eventbrite.com/api/key>

You will also have been provided an API User Key, specific to you and your Eventbrite account login details.

To obtain your User Key, log in to your Eventbrite account and visit <http://www.eventbrite.com/userkeyapi>

The user key is a unique string assigned by Eventbrite which identifies the user making an API call. This replaces basic username/password authentication. A valid user key is the recommended method for user authentication with the Eventbrite API.

You might have many application keys, but you will only have one user key.

Authentication methods will be covered a little later on in the documentation.

Instantiation

Open up Application.cfc, and within the onApplicationStart() method edit the code to include your Eventbrite API key, email address, password and User Key:

```
<cffunction name="onApplicationStart" output="false">
    <cfscript>
        this.applicationKey      = '<enter your eventbrite api key>';
        this.emailAddress        = '<enter your account email address>';
        this.password            = '<enter your account password>';
        this.userKey             = '<enter your eventbrite user key>';

        Application.objEventbrite = createObject('component',
            'com.fuzzyorange.eventbrite.eventbrite')
            .init(applicationKey=this.applicationKey,
                emailAddress=this.emailAddress,
                password=this.password,
                userKey=this.userKey,
                parseResults=true);
    </cfscript>
</cffunction>
```

The Eventbrite object is then instantiated and persisted in the Application scope, sending through the user access details you have supplied.

In the above example, we are also sending through an optional argument called 'parseResults'. Set to false by default, if set to true, the response XML data from the API remote calls will be returned using the XmlParse() function. If set to false, you will receive the literal string formatted XML.

The eventbrite.cfc file is the main object, and acts as a façade or Service Layer to interact with the underlying components.

Authentication

The Eventbrite API offers two modes of authentication; Secure and Basic. All authentication requests are made over SSL.

Secure

The secure method of authentication hides your login information and instead passes a user_key into the API. For example:

```
https://www.eventbrite.com/xml/user_events_list?app_key=<APP_KEY>
    &user_key=<USER_KEY>&id=...
```

If a user_key provided, the API will validate it against the Eventbrite user database. If the user_key is valid and authenticated, the method call will continue. If the user_key is not valid or presented an incorrect password and the method requires authentication, an XML error message is returned.

Basic

The Basic level of authentication involves providing the parameters "user" and "password" to each method call along with other parameters. For example:

```
https://www.eventbrite.com/xml/user_events_list?app_key=<APP_KEY>
    &user=<USER_EMAIL>&password=<USER_PASSWORD>&id=...
```

If a user and password are provided, the API will validate them against the Eventbrite user database. If the user is valid and authenticated, the method call will continue. If the user is not valid or presented an incorrect password and the method requires authentication, an XML error message is returned.

The Secure option is the recommended method, and Hermes has been developed with this authentication process as default, as the Basic method sends user account information in the parameters.

You can change the authentication method in the eventbrite.cfc by passing a separate argument through the init() method:

```
Application.objEventbrite = createObject('component',
    'com.fuzzyorange.eventbrite.eventbrite')
    .init(applicationKey=this.applicationKey,
        emailAddress=this.emailAddress,
        password=this.password,
        userKey=this.userKey,
        parseResults=true,
        authLevel='Basic');
```

Response Formats

All responses from the Eventbrite API are returned in XML format.

All errors are handled by the Eventbrite API, which will return something similar to the example below:

```
<error>
    <error_type>Authentication Error</error_type>
    <error_message>Invalid user_key.</error_message>
</error>
```

Available Methods

Here is a list of all available methods within the Hermes Events package to interact with the Eventbrite API.

copyEvent

This method returns the ID of the new event copied.

Parameters:

id: numeric, required, id - The ID of the event to update.

event_name: string, required, event_name - The event title.

eventSearch

I look for events by keyword. Only public events are listed.

Parameters:

keywords: string, optional, keywords - The search keywords.

category: string, optional, category - Event categories (comma separated): conference, conventions, entertainment, fundraisers, meetings, other, performances, reunions, sales, seminars, social, sports, tradeshow, travel, religion, fairs, food, music, recreation.

address: string, optional, address - The venue address.

city: string, optional, city - The venue city.

region: string, optional, region - The venue state/province/county/territory depending on the country. 2-letter state code is required for US addresses.

postal_code: string, optional, postal_code - The postal/zip code of the venue.

country: string, optional, country - 2-letter country code, according to the ISO 3166 format.

within: string, optional, within - If 'within' is set and the 'city' or 'zipcode' resolve to a specific geolocation, the search will be restricted to the specified within radius. The sorting default will be set to 'distance'.

within_unit: string, optional, within_unit - If within is set, you can specify the unit to use: 'M' for miles, or 'K' for kilometers. By default, the API will use miles.

latitude: string, optional, latitude - If 'within' is set you can limit your search to wgs84 coordinates (latitude, longitude).

longitude: string, optional, longitude - If 'within' is set you can limit your search to wgs84 coordinates (latitude, longitude).

date: string, optional, date - The event start date. Limit the list of results to a date range, specified by a label or by exact dates. Currently supported labels include: 'All', 'Future', 'Past', 'Today', 'Yesterday', 'Last Week', 'This Week', 'Next week', 'This Month', 'Next Month' and months by name, e.g. 'October'. Exact date ranges take the form 'YYYY-MM-DD YYYY-MM-DD', e.g. '2008-04-25 2008-04-27'.

date_created: string, optional, date_created - The date range the event was created, specified by a label or by exact dates. Currently supported labels include: 'Today', 'Yesterday', 'Last Week', 'This Week', 'This Month'. Exact date ranges take the form 'YYYY-MM-DD YYYY-MM-DD', e.g. '2008-04-25 2008-04-27'.

date_modified: string, optional, date_modified - The date the event was last modified, specified by a label or by exact dates. Currently supported labels include: 'Today', 'Yesterday', 'Last Week', 'This Week', 'This Month'. Exact date ranges take

the form 'YYYY-MM-DD YYYY-MM-DD', e.g. '2008-04-25 2008-04-27'.

organizer: string, optional, organizer - The organizer name.

max: numeric, optional, max - Limit the number of events returned. Maximum limit is 100 events per page. Default is 10.

count_only: boolean, optional, count_only - Only return the total number of events ('true' or 'false'). Default is 'false'.

sort_by: string, optional, sort_by - Sort the list of events by 'id', 'date', 'name', 'city'. The default is 'date'.

page: numeric, optional, page - Allows for paging through the results of a query. Default is 1.

since_id: numeric, optional, since_id - Returns events with id greater than 'since_id' value. Default is 1.

tracking_link: string, optional, tracking_link - The tracking link code to add to the event URLs.

getEvent

This method returns the data for a given event. Only public events are viewable if no authentication is passed.

Parameters:

id: numeric, required, id - The ID of the requested event.

getUser

I return the data for a given user. Only User or sub user which match login credentials are viewable.

Parameters:

user_id: string, required, user_id - The ID of the subuser account.

email: string, required, email - The email address of the subuser account.

listEventAttendees

This method returns a list of attendees for a given event. If no authentication is passed, only publicly available attendee records will be returned.

Parameters:

id: numeric, required, id - The ID of the event.

count: numeric, optional, count - Limit the number of attendees returned. Default limit is 50 attendees per page if paginate selected.

page: numeric, optional, page - Allows for paging through the results of a query. Default is 1. Page size will be setted with 'count' parameter.

listEventDiscounts

This method returns the list of discount codes created for a given event.

Parameters:

id: numeric, required, id - The ID of the event.

listOrganizerEvents

I list all events created by the supplied organizer. Only public events are returned if no authentication is passed.

Parameters:

id: string, required, id - The organizer ID

listUserEvents

I create a new user. Only the fields passed as arguments will be modified. It returns the ID of the updated user.

Parameters:

user: string, optional, user - The user email. Defaults to the authenticated user if not provided.

listUserOrganizers

This method lists the organizers created by this user.

Parameters:

user: string, optional, user - The user email.

password: string, required, password - The user password.

listUserTickets

This method lists the tickets purchased by the authenticated user.

Parameters:

user: string, optional, user - The user email.

password: string, required, password - The user password.

listUserVenues

This method lists the venues created by this user.

Parameters:

user: string, optional, user - The user email.

password: string, required, password - The user password.

newDiscount

This method creates a new discount code. It returns the ID of the newly created discount code.

Parameters:

event_id: numeric, required, event_id - The event ID. The event must have been previously created using newEvent.

code: string, required, code - The discount code. Spaces, apostrophes and non-alphanumeric characters are not allowed, except for dashes and underscores.

Examples: 'earlybirdspecial_08', 'membersonly', 'dc121232'.

amount_off: string, required, amount_off - The fixed amount off the ticket price. Each discount code can have a fixed discount amount or a variable (percentage) discount amount, but not both.

percent_off: string, optional, percent_off - The percentage off the ticket price. Each discount code can have a fixed discount amount or a variable (percentage) discount amount, but not both.

tickets: string, optional, tickets - Comma-separated list of ticket IDs for which the discount applies. If not provided, the discount will apply to all ticket types.

quantity_available: string, optional, quantity_available - Maximum number of times this discount can be used. If not provided, no maximum is set.

start_date: string, optional, start_date - The discount start date and time, in ISO 8601 format (e.g., '2007-12-31 23:59:59').

end_date: string, optional, end_date - The discount end date and time, in ISO 8601 format (e.g., '2007-12-31 23:59:59').

newEvent

This method creates a new event. It returns the ID of the newly created event.

Parameters:

title: string, required, title - The event title.

description: string, optional, description - The event description.

start_date: string, optional, start_date - The event start date and time, in ISO 8601 format (e.g., '2007-12-31 23:59:59').

end_date: string, optional, end_date - The event end date and time, in ISO 8601 format (e.g., '2007-12-31 23:59:59').

timezone: string, optional, timezone - The event time zone in relation to GMT (e.g., 'GMT+01', 'GMT+02', 'GMT-01').

privacy: numeric, optional, privacy - 0 for a private event, 1 for a public event. If not provided, will default to 1.

personalized_url: string, optional, personalized_url - The event registration URL. If you pass 'testevent', the event will be accessible at 'http://testevent.eventbrite.com'.

venue_id: numeric, optional, venue_id - The event venue ID.

organizer_id: numeric, optional, organizer_id - The event organizer ID.

capacity: numeric, optional, capacity - The maximum number of people who can attend the event.

currency: string, optional, currency - The event currency in ISO 4217 format (e.g., 'USD', 'EUR').

status: string, optional, status - The event status. Allowed values are 'draft', 'live' for new events. If not provided, status will be 'draft', meaning that the event registration

page will not be available publicly.

custom_header: string, optional, custom_header - Custom HTML header for your registration page.

custom_footer: string, optional, custom_footer - Custom HTML footer for your registration page.

background_color: string, optional, background_color - Custom hexadecimal color for your registration page. Format: FFFFFFFF without the pound.

text_color: string, optional, text_color - Custom hexadecimal color for your registration page. Format: FFFFFFFF without the pound.

link_color: string, optional, link_color - Custom hexadecimal color for your registration page. Format: FFFFFFFF without the pound.

title_text_color: string, optional, title_text_color - Custom hexadecimal color for your registration page. Format: FFFFFFFF without the pound.

box_background_color: string, optional, box_background_color - Custom hexadecimal color for your registration page. Format: FFFFFFFF without the pound.

box_text_color: string, optional, box_text_color - Custom hexadecimal color for your registration page. Format: FFFFFFFF without the pound.

box_border_color: string, optional, box_border_color - Custom hexadecimal color for your registration page. Format: FFFFFFFF without the pound.

box_header_background_color: string, optional, box_header_background_color - Custom hexadecimal color for your registration page. Format: FFFFFFFF without the pound.

box_header_text_color: string, optional, box_header_text_color - Custom hexadecimal color for your registration page. Format: FFFFFFFF without the pound.

newOrganizer

I create a new organizer, and I return the ID of the newly created organizer.

Parameters:

name: string, required, name - The organizer name.

description: string, optional, description - The organizer description.

newTicket

I create a new fixed-price ticket, and I return the ID of the newly created ticket.

Parameters:

event_id: numeric, required, event_id - The event ID. The event must have been previously created using newEvent.

is_donation: numeric, optional, is_donation - 0 for fixed-price tickets, 1 for donations. 0 will be used by default if not provided.

name: string, required, name - The ticket name.

description: string, optional, description - The ticket description.

price: string, optional, price - The ticket price. Enter 0.00 for free tickets. Leave blank for a donation.

quantity: string, optional, quantity - The number of tickets available. Not required for donations.

start_sales: string, optional, start_sales - The date and time when ticket sales start, in ISO 8601 format (e.g., '2007-12-31 23:59:59').

end_sales: string, optional, end_sales - The date and time when ticket sales stop, in ISO 8601 format (e.g., '2007-12-31 23:59:59').

include_fee: numeric, optional, include_fee - 0 to add the Eventbrite service fee on top of ticket price, or 1 to include it in the ticket price. 0 will be used by default if not provided.

min: string, optional, min - The minimum number of tickets per order.

max: string, optional, max - The maximum number of tickets per order.

newUser

I create a new user, and I return the ID of the newly created user.

Parameters:

email: string, required, email - The user email address.

passwd: string, required, passwd - The user password (needs to be at least 4 characters long).

newVenue

I create a new venue, and I return the ID of the newly created venue.

Parameters:

organizer_id: numeric, required, organizer_id - The organizer ID

venue: string, required, venue - The venue name.

address: string, optional, address - The venue address (line 1).

address_2: string, optional, address_2 - The venue address (line 2).

city: string, optional, city - The venue city.

region: string, required, region - The venue state/province/county/territory depending on the country. 2-letter state code is required for US addresses.

postal_code: string, optional, postal_code - The postal code of the venue.

country_code: string, required, country_code - 2-letter country code, according to the ISO 3166 format.

updateDiscount

This method update an existing discount code. Only the fields passed as arguments will be modified. This method returns the ID of the modified discount code.

Parameters:

id: numeric, required, id - The discount ID to update.

code: string, required, code - The discount code. Spaces, apostrophes and non-alphanumeric characters are not allowed, except for dashes and underscores.

Examples: 'earlybirdspecial_08', 'membersonly', 'dc121232'.

amount_off: string, required, amount_off - The fixed amount off the ticket price. Each discount code can have a fixed discount amount or a variable (percentage) discount amount, but not both.

percent_off: string, optional, percent_off - The percentage off the ticket price. Each discount code can have a fixed discount amount or a variable (percentage) discount amount, but not both.

tickets: string, optional, tickets - Comma-separated list of ticket IDs for which the

discount applies. If not provided, the discount will apply to all ticket types.

quantity_available: string, optional, quantity_available - Maximum number of times this discount can be used. If not provided, no maximum is set.

start_date: string, optional, start_date - The discount start date and time, in ISO 8601 format (e.g., '2007-12-31 23:59:59').

end_date: string, optional, end_date - The discount end date and time, in ISO 8601 format (e.g., '2007-12-31 23:59:59').

updateEvent

This method updates an existing event. Only the fields passed as arguments will be modified. This method returns the ID of the modified event.

Parameters:

event_id: numeric, required, event_id - The ID of the event to update.

title: string, optional, title - The event title.

description: string, optional, description - The event description.

start_date: string, optional, start_date - The event start date and time, in ISO 8601 format (e.g., '2007-12-31 23:59:59').

end_date: string, optional, end_date - The event end date and time, in ISO 8601 format (e.g., '2007-12-31 23:59:59').

timezone: string, optional, timezone - The event time zone in relation to GMT (e.g., 'GMT+01', 'GMT+02', 'GMT-01').

privacy: numeric, optional, privacy - 0 for a private event, 1 for a public event. If not provided, will default to 1.

personalized_url: string, optional, personalized_url - The event registration URL. If you pass 'testevent', the event will be accessible at 'http://testevent.eventbrite.com'.

venue_id: numeric, optional, venue_id - The event venue ID.

organizer_id: numeric, optional, organizer_id - The event organizer ID.

capacity: numeric, optional, capacity - The maximum number of people who can attend the event.

currency: string, optional, currency - The event currency in ISO 4217 format (e.g., 'USD', 'EUR').

status: string, optional, status - The event status. Allowed values are 'draft', 'live' for new events. If not provided, status will be 'draft', meaning that the event registration page will not be available publicly.

custom_header: string, optional, custom_header - Custom HTML header for your registration page.

custom_footer: string, optional, custom_footer - Custom HTML footer for your registration page.

background_color: string, optional, background_color - Custom hexadecimal color for your registration page. Format: FFFFFFFF without the pound.

text_color: string, optional, text_color - Custom hexadecimal color for your registration page. Format: FFFFFFFF without the pound.

link_color: string, optional, link_color - Custom hexadecimal color for your registration page. Format: FFFFFFFF without the pound.

title_text_color: string, optional, title_text_color - Custom hexadecimal color for your registration page. Format: FFFFFFFF without the pound.

box_background_color: string, optional, box_background_color - Custom hexadecimal color for your registration page. Format: FFFFFFFF without the pound.

box_text_color: string, optional, box_text_color - Custom hexadecimal color for your registration page. Format: FFFFFFFF without the pound.

box_border_color: string, optional, box_border_color - Custom hexadecimal color for your registration page. Format: FFFFFFFF without the pound.

box_header_background_color: string, optional, box_header_background_color - Custom hexadecimal color for your registration page. Format: FFFFFFFF without the pound.

box_header_text_color: string, optional, box_header_text_color - Custom hexadecimal color for your registration page. Format: FFFFFFFF without the pound.

updateOrganizer

I update an existing organizer. Only the fields passed as arguments will be modified. It returns the ID of the updated organizer.

Parameters:

id: string, required, id - The organizer ID

name: string, required, name - The organizer name.

description: string, optional, description - The organizer description.

updatePayment

This method creates or updates the payment options for this event. Only the fields passed as arguments will be modified. It returns the list of active method of payment.

Parameters:

event_id: numeric, required, event_id - The event ID.

accept_paypal: numeric, optional, accept_paypal - Accept PayPal payments (1 or 0).

paypal_email: string, optional, paypal_email - Your PayPal email. Defaults to the user email address if not provided.

accept_google: numeric, optional, accept_google - Accept Google Checkout payments (1 or 0).

google_merchant_id: string, optional, google_merchant_id - Google Checkout Merchant ID. Required if accept_google is 1.

google_merchant_key: string, optional, google_merchant_key - Google Checkout Merchant Key. Required if accept_google is 1.

accept_check: numeric, optional, accept_check - Accept 'Pay by Check' payments (1 or 0).

instructions_check: string, optional, instructions_check - Instructions to attendees who want to pay by check.

accept_cash: numeric, optional, accept_cash - Accept 'Pay with Cash' payments (1 or 0).

instructions_cash: string, optional, instructions_cash - Instructions to attendees who want to pay with cash.

accept_invoice: numeric, optional, accept_invoice - Accept 'Send an Invoice' payments (1 or 0).

instructions_invoice: string, optional, instructions_invoice - Instructions to attendees who need to be sent an invoice.

updateTicket

This method updates an existing ticket type. Only the fields passed as arguments will be modified. It returns the ID of the updated ticket.

Parameters:

- id:** numeric, required, id - The ticket ID.
- is_donation:** numeric, optional, is_donation - 0 for fixed-price tickets, 1 for donations. 0 will be used by default if not provided.
- name:** string, required, name - The ticket name.
- description:** string, optional, description - The ticket description.
- price:** string, optional, price - The ticket price. Enter 0.00 for free tickets. Leave blank for a donation.
- quantity:** string, optional, quantity - The number of tickets available. Not required for donations.
- start_sales:** string, optional, start_sales - The date and time when ticket sales start, in ISO 8601 format (e.g., '2007-12-31 23:59:59').
- end_sales:** string, optional, end_sales - The date and time when ticket sales stop, in ISO 8601 format (e.g., '2007-12-31 23:59:59').
- include_fee:** numeric, optional, include_fee - 0 to add the Eventbrite service fee on top of ticket price, or 1 to include it in the ticket price. 0 will be used by default if not provided.
- min:** string, optional, min - The minimum number of tickets per order.
- max:** string, optional, max - The maximum number of tickets per order.
- hide:** string, optional, hide - Show or hide the ticket type. Valid options: (y or n).

updateUser

I update an existing user. Only the fields passed as arguments will be modified. It returns the ID of the updated user.

Parameters:

- new_email:** string, required, new_email - New user email address.
- new_password:** string, required, new_password - New user password (needs to be at least 4 characters long).

updateVenue

This method updates an existing venue. Only the fields passed as arguments will be modified. It returns the ID of the updated venue.

Parameters:

- id:** numeric, required, id - The venue ID
- venue:** string, required, venue - The venue name.
- adress:** string, optional, adress - The venue address (line 1).
- adress_2:** string, optional, adress_2 - The venue address (line 2).
- city:** string, optional, city - The venue city.
- region:** string, optional, region - The venue state/province/county/territory depending on the country. 2-letter state code is required for US addresses.
- postal_code:** string, optional, postal_code - The postal code of the venue.
- country_code:** string, optional, country_code - 2-letter country code, according to the ISO 3166 format.

Limitations

Testing

Hermes Events has been tested on the Adobe ColdFusion platform.
We have yet to test fully on Railo or OpenBlueDragon.