

OAuth & Tokens

Fire Eagle uses [OAuth](#) for application authentication and authorization. In a nutshell, the OAuth protocol defines the process through which you authenticate your application to Fire Eagle and the process for users to authorize your application to access their location information in Fire Eagle. The OAuth protocol specifies that all API calls contain token parameters which identify the application and/or the user.

Fire Eagle access tokens

Fire Eagle defines two types of access tokens which are used for different types of API methods. The API method will define what type of access token you need to use in order to generate the OAuth signature for the API request.

1. User-specific Access Token

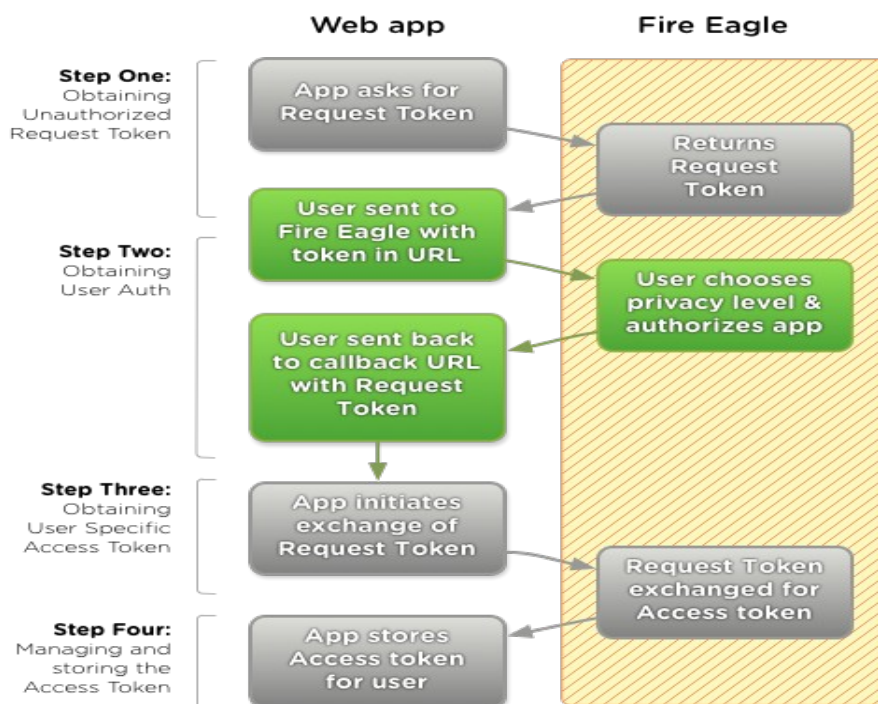
This token defines a User's relationship to a Fire Eagle application and is used for user-specific API methods. It is generated when a user authorizes your application and gives permission for your application to access his or her Fire Eagle location information.

2. General-purpose Access Token

This token defines an application's relationship to Fire Eagle. They are tied to your application and allow your application to make general-purpose API method calls (often batch-style) to Fire Eagle.

Auth for Web Apps

In web-based applications, the server is the primary connection point to Fire Eagle. For each user that wants to authorize your app, the following illustrates the authorization flow.



1. Obtaining an unauthorized request token

Request Token URL: https://fireagle.yahooapis.com/oauth/request_token

Required Parameters: `oauth_consumer_key` : application consumer key ; `oauth_callback` : the callback URL in case of web applications, or 'oob' ; `oauth_nonce`, `oauth_timestamp`, `oauth_signature_method`, `oauth_version`, `oauth_signature`

Your users are not involved in this step. First, your application makes an API call to Fire Eagle for a *request token*. Note that the `oauth_signature` you generate for this call will use only your `oauth_consumer_key` as a key. Fire Eagle will respond with a unique *request token*.

Fire Eagle's response will look something like:

```
oauth_token=jw99864fjif4&oauth_token_secret=fc3y9mdkffnb4b5j0qq&oauth_callback_confirmed=true
```

The `oauth_token` and `oauth_token_secret` are both required components of the *request token*. This *request token* is temporary for this user authorization session.

2. Obtaining user authorizations

User Authorization URL: <https://fireeagle.yahoo.net/oauth/authorize>

Required Parameters: `oauth_token` : the `request_token` that you obtained in the previous step

After obtaining the *request token*, your application constructs the authorization URL to call Fire Eagle with the *request token*. The `oauth_token` parameter is the *request token* from the response in step 1. The `oauth_token_secret` from step 1 is appended to your `oauth_consumer_secret` to create a key for generating the `oauth_signature`.

You get the user's browser pointed at the authorization URL where he or she can choose whether to authorize your application or not. If the user authorizes your application, Fire Eagle will invoke your application callback URL with the *request token* appended as parameters. An `oauth_verifier` parameter shall also be appended to the callback.

3. Obtaining user-specific access token

Access Token URL: https://fireeagle.yahooapis.com/oauth/access_token

Required Parameters: `oauth_consumer_key` : application consumer key ; `oauth_verifier` : received through the callback in step 2 ; `oauth_token` : the `request_token` passed to your application callback URL by FE. This should be the same as the `request_token` you obtained in step one ; `oauth_nonce`, `oauth_timestamp`, `oauth_signature_method`, `oauth_version`, `oauth_signature`

After the user authorizes your application, your application needs to exchange the *request token* for a permanent *user-specific access token*, *access token* for short. The *access token* uniquely identifies the user to your application, represents the permissions the user has authorized to your application and allows your application to update or query Fire Eagle for the user's location information on behalf of the user.

To get the *access token*, the callback URL registered with your application will be called by Fire Eagle with the *request token* after the user authorizes your application. At this point, your application needs to make an API call to Fire Eagle with the *request token*. Similar to step 2, the `oauth_token` parameter is the *request token* token, while the *request token's* `oauth_token_secret` is appended to your `oauth_consumer_secret` to create a key for generating the `oauth_signature`.

If the user has properly authorized your application, Fire Eagle will respond with a unique *access token* for the user. Fire Eagle will respond with something like:

```
oauth_token=1q3kfvcmey74&oauth_token_secret=68xedxj4cbwov5agufgea3v1z80p16s3
```

The `oauth_token` and `oauth_token_secret` passed back in this step are the *access token* for this user. You will no longer need to worry about the *request token* (for this user).

4. Managing and storing the access token

You will receive an access token and access secret for the user which you need to store together securely. The *access token* ties the user to your application and is your pass to update and query for the the user's location within Fire Eagle. You need to figure out how your application associates the *access token* with your application's representation of the user. The *access secret* is used to sign your application's query and update requests for the user.

For server-based applications, *access tokens* and *access secrets* should be treated as private data on your web server. Protect this data from the public as the corresponding user's location information may be inadvertently exposed if the user's *access token* and *access secret* are compromised. **User-specific access tokens should be considered as the property of your users.**

Calling the API

API method types

Fire Eagle has two classes of API methods: *user-specific* and *general-purpose*. The user-specific methods are called on behalf of an authenticated user while the general-purpose methods are called on behalf of the application.

User-specific

API Methods for updating and querying a user's own location. Calls to this API should use a *user-specific access token*. The primary User-specific calls are user, update, and lookup.

General-purpose

API Methods for getting information about all users of the application, e.g. recently updated locations, and users who are within a location. Calls to this API should use the *general-purpose access token*. The primary General-purpose calls are recent, within, and lookup.

Only web-based applications are allowed to call general-purpose methods. When you create a web-based application, you will be assigned a general-purpose token and general-purpose token secret.

URL & Response Formats

Fire Eagle requests are RESTful. The URL also controls the response format. The proper HTTP method changes depending on the API call.

URL Format

Information (including OAuth authentication) is sent with the URL as query string parameters in the form of:

`https://[URL]/api/[version]/[method].[response_format]?[query-string-parameters]`

Response Format

Fire Eagle has two response formats: XML or JSON. Response formats are specified by appending the requested format to the URL. If no response format is specified, then by default, the response format is XML.

For example, to request a user location in XML format:

https://fireeagle.yahooapis.com/api/0.1/user.xml?oauth_consumer_ke...

For JSON, change the url to end with .json. If you need the resulting data wrapped in a function, include a callback parameter with its name.

https://fireeagle.yahooapis.com/api/0.1/user.json?oauth_consumer_ke...

HTTP Method

The HTTP method (GET and POST) varies on the request type. In RESTful fashion, queries (such as /user, /recent, and /lookup) are GETs. Updates (currently only /update) are POSTs. The method *must* match the operation (or you will get a 405 Unsupported Method error).

Request Parameters

Parameters fall into three categories: Required, Default, and Invalid.

Required

Required parameters must be specified in the request. They have no default values. Passing in a null (for example foo has a null value: `?city=anytown&foo=&state=anystate`) or invalid argument will return an error. Some parameters are only required in conditional cases. For example, when specifying a location, some valid set of values is required, but only one such set.

Default Values

If a parameter is not required, it may have a default value. The default will only be used if the parameter is

not supplied in the method call. If you supply a null or invalid value for a parameter, the default will not be assumed in its place.

Invalid Values

Required or not, each parameter may have invalid parameters. For example, longitude, and latitude values must be between -180.0 and 180.0 and -90.0 and 90.0, respectively. If you supply an invalid value for a parameter, an error will be returned even if the parameter is optional, is conditionally optional (as in location queries & updates) or has a default value.

Response Structure

Fire Eagle returns two response formats: XML or JSON. Set xml or json as your [response_format] in the API request.

XML Response Format

```
<?xml version="1.0" encoding="UTF-8"?>
<rsp stat="ok" xmlns:georss="http://www.georss.org/georss">
  [api response]
</rsp>
```

JSON Response Format

```
{"user": {api response}, "stat": "ok"}
```

API Calls

update

Sets a user's current location using using a set of location parameters. If the user provides a location unconfirmed with lookup method then Fire Eagle makes a best guess as to the user's location and updates.

Called with the user-specific access token: <https://fireeagle.yahooapis.com/api/0.1/update>

POST Data: oauth_consumer_key=00000000000000&oauth_signature_method=HMAC-SHA1&oauth_version=1.0&oauth_token=mEhrJ6ZI8RHI&postal=94107&oauth_timestamp=1204592174&oauth_nonce=Jc0B3D&oauth_signature=w5JkljU3lk3MvAxINnyPJcGuBAQ%3D

XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<rsp stat="ok">
  <user token="mEhrJ6ZI8RHI" located-at="2008-08-06T16:04:46-08:00"/>
</rsp>
```

JSON Response (using /update.json)

```
{"user": {
  "token": "mEhrJ6ZI8RHI",
  "located_at": "2008-08-06T16:04:46-08:00"},
"stat": "ok"}
```

lookup

Disambiguates potential values for update. Results from lookup can be passed to update to ensure that Fire Eagle will understand how to parse the location parameter.

Called with the user-specific or general purpose access token:

https://fireeagle.yahooapis.com/api/0.1/lookup.xml?oauth_consumer_key=00000000000000&oauth_signature_method=HMAC-SHA1&oauth_version=1.0&oauth_token=UZuTj74EUF70&address=Pensacola&oauth_timestamp=1204590774&oauth_nonce=voOR9B&oauth_signature=m66dL4XQ4PzhX4CkXaGiOE76lig%3D

XML Response

```
<rsp stat="ok">
  <query>address=Pensacola</query>
  <locations start="0" total="3" count="3">
    <location>
      <name>Pensacola, FL</name>
      <place-id>qQ7Vig2bBZsZCy82</place-id>
      <woeid>2470377</woeid>
    </location>
    <location>
      ...
    </location>
    ...
  </locations>
</rsp>
```

JSON Response (using /lookup.json)

```
{"start":0,
"stat":"ok",
"locations":[{"name":"Pensacola, FL",
  "place_id":"qQ7Vig2bBZsZCy82",
  "woeid":2470377},
{"name":"Pensacola, NC",
  "place_id":"4.CTKJibBZv.G1c2",
  "woeid":2470405},
{"name":"Pensacola, OK",
  "place_id":"bRDKoQibBZtXopXZ",
  "woeid":2470375}],
"count":3,
"total":3,
"query":"address=Pensacola"}
```

user

Returns the location of a specific user in a location hierarchy format.

Called with the user-specific access token:

https://fireagle.yahooapis.com/api/0.1/user?oauth_consumer_key=0000000000000&oauth_signature_method=HMAC-SHA1&oauth_version=1.0&oauth_token=mEhrJ6ZI8RHI&oauth_timestamp=1204588542&oauth_nonce=WXeVhx&oauth_signature=OPIgcbKC2WQVO3kL6oATdiUdf4g%3D

XML Response

```
<rsp stat="ok">
  <user token="mEhrJ6ZI8RHI" located-at="2008-08-06T16:05:48-08:00">
    <location-hierarchy timezone="America/Los_Angeles">
      <location best-guess="true">
        <id>114031</id>
        <georss:point>37.7812461853 -122.3957595825</georss:point>
        <level>0</level>
        <level-name>exact</level-name>
        <located-at>2008-03-03T10:58:55-08:00</located-at>
        <name>500 3rd St, San Francisco, CA</name>
      </location>
      <location best-guess="false">
        <id>114041</id>
        <georss:box>
          37.7494697571 -122.40650177 37.7862281799 -122.3790893555
        </georss:box>
        <level>1</level>
```

```
<level-name>postal</level-name>
<located-at>2008-03-03T10:58:55-08:00</located-at>
<name>San Francisco, CA 94107</name>
<normal-name>94107</normal-name>
<place-id>8Xq01wWYA5u_OEMhyQ</place-id>
<woeid>12797158</woeid>
</location>
...
</location-hierarchy>
</user>
</rsp>
```

JSON Response (using /user.json)

```
{ "stat" : "ok",
  "user" : {
    "token" : "RzpwVb7fmznJ",
    "timezone" : "America/Los_Angeles",
    "location_hierarchy" : [{
      "label" : null,
      "located_at" : "2008/08/07 17:50:15 -0700",
      "level" : 3,
      "normal_name" : "San Francisco",
      "name" : "San Francisco, CA",
      "geometry" : {
        "type" : "Polygon",
        "coordinates" : [[[-122.5154571533, 37.7037811279],
                          [-122.32472229, 37.7037811279],
                          [-122.32472229, 37.8545417786],
                          [-122.5154571533, 37.8545417786],
                          [-122.5154571533, 37.7037811279]]],
        "bbox" : [[-122.5154571533, 37.7037811279],
                  [-122.32472229, 37.8545417786]]
      },
      "level_name" : "city",
      "woeid" : 2487956,
      "best_guess" : true,
      "id" : 17711351,
      "place_id" : "kH8dLOubBZRvX_YZ"
    }], {
      "label" : null,
      "located_at" : "2008/08/07 17:50:15 -0700",
      "level" : 4,
      "normal_name" : "San Francisco",
      "name" : "San Francisco County, California",
      "geometry" : {
        "type" : "Polygon",
        "coordinates" : [[[-123.1080169678, 37.6930503845],
                          [-122.3567581177, 37.6930503845],
                          [-122.3567581177, 37.832359314],
                          [-123.1080169678, 37.832359314],
                          [-123.1080169678, 37.6930503845]]],
        "bbox" : [[-123.1080169678, 37.6930503845],
                  [-122.3567581177, 37.832359314]]
      },
      "level_name" : "region",
      "woeid" : 12587707,
      "best_guess" : false,
      "id" : 17711341,
      "place_id" : "hCca8XSYA5nn0X1Sfw"
```

```
  },
  ...
]
}
```

recent

Returns a list of users of the application who have updated their location within the given amount of time.

Called with the general-purpose access token:

https://fireagle.yahooapis.com/api/0.1/recent.xml?oauth_consumer_key=000000000000&oauth_signature_method=HMAC-SHA1&oauth_version=1.0&oauth_token=UZuTj74EUF70&count=2&time=&oauth_timestamp=1204605774&oauth_nonce=NPvHhi&oauth_signature=LexXe6PhQX814McdNifQ%2Bp6CiHQ%3D

XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<rsp stat="ok" xmlns:georss="http://www.georss.org/georss">
  <users>
    <user token="8zuzkn86wh5x" located-at="2008-08-01T10:59:09-07:00"/>
  </users>
</rsp>
```

JSON Response (using /recent.json)

```
{"users": [
  {
    "located_at": "2008-08-01T10:59:09-07:00",
    "token": "8zuzkn86wh5x"
  }
],
"stat": "ok"
}
```

within

Takes a Place ID or a WoE ID and returns a list of users using your application who are within the bounding box of that location.

Called with the general-purpose access token:

https://fireagle.yahooapis.com/api/0.1/within.xml?oauth_consumer_key=000000000000&oauth_signature_method=HMAC-SHA1&oauth_version=1.0&oauth_token=UZuTj74EUF70&place_id=&woeid=12796255&oauth_timestamp=1204596221&oauth_nonce=9IZU3E&oauth_signature=nYhHrF6uhGBgFkqPVqSNBbHg5NI%3D

XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<rsp stat="ok" xmlns:georss="http://www.georss.org/georss">
  <users>
    <user token="8zuzkn86wh5x" located-at="2008-08-01T10:59:09-07:00"/>
  </users>
</rsp>
```

JSON Response (using /within.json)

```
{"users": [
  {
    "located_at": "2008-08-01T10:59:09-07:00",
    "token": "8zuzkn86wh5x"
  }
],
"stat": "ok"
}
```

}