

Ticketmaster API

Version 3

Table of Contents

Tables	4
Introduction	5
Intended audience	5
The Ticketmaster service	5
Updates and versioning	5
Changes from version 2	5
API Overview	5
WSDL	5
User authentication	5
Quota	5
SOAP Request Format	6
API Method: findEvents	6
FindEventsRequest	6
Available Countries	6
FindEventsRequest_Sort	7
FindEventsRequest_Filter	7
FindEventsResponse	9
Details	9
Event	9
Artist	10
Venue	11
API Method: getCategories	13
GetCategoriesRequest	13
GetCategoriesResponse	13
ParentCategory	13
Category	13
Alternative API Access	14
URL structure	14
Required parameters (for both methods)	14
Method: /event	15
optional parameters	15

Method: /categories	16
Tracking links.....	17
Building the tracking link	17
Example.....	17
Appendix 1: Country Codes.....	19
Appendix 2: Example SOAP Request.....	20
Appendix 3: Example SOAP Response	21
Appendix 4: Example HTTP GET Request.....	24
Appendix 5: Example HTTP GET Responses.....	25

Tables

Table 1: FindEventsRequest fields	6
Table 2: FindEventsRequest_Sort fields.....	7
Table 3: Sortable fields.....	7
Table 4: FindEventsRequest_Filter fields	7
Table 5: Filter fields.....	8
Table 6: FindEventsResponse fields.....	9
Table 7: Details fields	9
Table 8: Event fields	9
Table 9: Artist fields	10
Table 10: Venue fields.....	11
Table 11: GetCategoriesRequest fields.....	13
Table 12: ParentCategory fields.....	13
Table 13: Category fields.....	13
Table 14: Available countries	19

Introduction

Intended audience

This document is intended for developers who want to programmatically access information about Ticketmaster events. It assumes the readers are familiar with web services and with the programming language they will be using.

The Ticketmaster service

The Ticketmaster service is a SOAP API, providing the necessary functionality to get information about Ticketmaster events, complete with images, URLs, venue details and artist categorization.

To use the API, you will need to build a web service client in your chosen language.

Updates and versioning

The current version of the Ticketmaster service is 3.

As the service is updated, new versions will be deployed and old versions will stop being supported. They will however be left on our servers for a limited time to allow developers a smooth transition from one version to another.

Changes from version 2

- Data for additional countries is available.
- Data for multiple countries may be queried in one call.
- A currency code is now included with each event.
- A description is included with each event (where available).
- Longitude/latitude is returned with each venue (where available).
- A new method `getCategories` returns the category tree.

API Overview

WSDL

The current version of the Ticketmaster service WSDL can be found at:

<http://ticketmaster.productserve.com/v3/soap.php?wsdl>

User authentication

An API key is required with each request (see Table 1: FindEventsRequest fields).

Quota

Each user has a daily request quota.

SOAP Request Format

SOAP requests are “envelopes” of specially formatted XML data posted to a URL. You can find out more about SOAP at www.w3.org/TR/soap/.

There are two API methods available: `findEvents` and `getCategories`.

API Method: `findEvents`

The `findEvents` method allows you to retrieve a paginated list of event details. It is called with a `FindEventsRequest` object, which takes the following parameters:

FindEventsRequest

Table 1: FindEventsRequest fields

Field	Type	Nullable?	Notes
apiKey	String	No	
country	String	No	A comma-delimited list of 2-letter country codes. See Table 14: Available countries.
resultsPerPage	Integer	Yes	If not specified, 50 results are returned per page. The maximum value is 100.
currentPage	Integer	Yes	If not specified, page 1 of the result set is returned.
Sort	FindEventsRequest_Sort	Yes	See Table 4: FindEventsRequest_Filter fields
Filters	FindEventsRequest_Filter[]	Yes	Array of FindEventsRequest_Filter objects. See Table 4: FindEventsRequest_Filter fields.
updatedSince	String	Yes	This should be a string formatted like “2012-10-28 13:42:27”. This limits the returned results to those that have been updated since the given date and time.

Available Countries

Availability of data depends on being joined to the relevant Advertiser Programme. ‘US’ and ‘CA’ data requires that you are joined to ‘Ticketmaster (US & CA)’. All other data requires that you are joined to ‘Ticketmaster UK’.

A single request can only cover a single programme.

FindEventsRequest_Sort

The `FindEventsRequest_Sort` object specifies the order in which the results are returned.

Table 2: FindEventsRequest_Sort fields

Field	Type	Nullable?	Notes
Field	String	No	See Table 3: Sortable fields.
Order	String	No	Possible values are "ASC" or "DESC"

Table 3: Sortable fields

Field	Notes
Name	The event name
eventDate	The date the event is due to occur
onSaleDate	The date that tickets for the event go/went on sale.
preSaleDate	The date that tickets for the event go/went on pre-sale
categoryId	A child category ID. See the response from <code>getCategories</code> .
minPrice	The minimum price of tickets to the event
maxPrice	The maximum price of tickets to the event

FindEventsRequest_Filter

The `FindEventsRequest_Filter` object limits the results to certain criteria.

Table 4: FindEventsRequest_Filter fields

Field	Type	Nullable?	Notes
Object	string	No	One of "event", "artist" or "venue"
Field	string	No	Which field to filter on. See Table 5: Filter fields.

values	string[]	No	An array of possible values for the given field. See Table 5: Filter fields
--------	----------	----	--

Table 5: Filter fields

Object	Field	Notes
event	ticketmasterEventId	The event ID provided by Ticketmaster
	categoryId	A child category ID. See the response from getCategories.
	parentCategoryId	A parent category ID. See the response from getCategories.
	status	Possible values are: "onSale", "notOnSale", "cancelled", and "removed". If this filter is not given, the results are implicitly filtered to "onSale".
artist	name	The artist name
	ticketmasterArtistId	The artist ID provided by Ticketmaster.
	categoryId	A child category ID. See the response from getCategories.
	parentCategoryId	A parent category ID. See the response from getCategories.
venue	name	The venue name.
	ticketmasterVenueId	The venue ID provided by Ticketmaster.
	postcode	Postcode/zipcode, E.g. "E1 8EU"
	city	City. E.g. "London"
	state	State or province (US/CA only). E.g. "TX"

FindEventsResponse

Table 6: FindEventsResponse fields

Field	Type	Notes
Details	Details	See Table 7: Details fields.
Results	Event[]	Array of Event objects. See Table 8: Event fields.

Details

Table 7: Details fields

Field	Type	Notes
totalResults	Integer	The total number of events in the result set.
totalPages	Integer	The total number of pages for the result set.
currentPage	Integer	The page returned in this response.
resultsPerPage	Integer	How many events are included in each page. Note that the final page in the set may have fewer events than this number.

Event

Table 8: Event fields

Field	Type	Notes
eventId	Integer	The ID used internally to identify the event.
ticketmasterEventId	String	The ID used by Ticketmaster to identify the event.
status	String	One of "onSale", "notOnSale", "cancelled", or "removed".
name	String	The name of the event.
url	String	URL to the Ticketmaster page for the event.
eventDate	String	When the event is scheduled to occur. Format is like this: "2011-11-28 19:30:00".

onSaleDate	String	When tickets for the event go/went on sale. Format is like this: "2011-11-28 19:30:00".
preSaleDate	String	When tickets for the event go/went on pre-sale. Format is like this: "2011-11-28 19:30:00".
category	String	Human-readable description of the category given in categoryId.
categoryId	Integer	See the response from getCategories.
parentCategory	String	Human-readable description of the category given in categoryId.
parentCategoryId	Integer	See the response from getCategories.
minPrice	Decimal	The minimum price of tickets.
maxPrice	Decimal	The maximum price of tickets.
currency	String	3-letter currency code
artists	Artist[]	An array of Artist objects. See Table 9: Artist fields.
venue	Venue	A Venue object. See Table 10: Venue fields.
description	String	Further information about the event, where available.

Artist

Table 9: Artist fields

Field	Type	Notes
artistId	Integer	Internal ID for this artist.
ticketmasterArtistId	Integer	ID used by Ticketmaster for this artist.
name	String	The name of the artist.
url	String	URL to a Ticketmaster page about the artist.

imageUrl	String	The URL of an image of the artist, or a generic image.
category	String	Human-readable description of the category given in categoryId. See Error! Reference source not found..
categoryId	Integer	See Error! Reference source not found..
parentCategory	String	Human-readable description of the category given in categoryId.
parentCategoryId	Integer	See the response from getCategories.

Venue

Table 10: Venue fields

Field	Type	Notes
venueId	Integer	Internal ID for this venue.
ticketmasterVenueId	Integer	ID used by Ticketmaster for this venue.
name	String	The name of the venue.
street	String	The street address of the venue - e.g. "268-269 Tottenham Court Road".
city	String	The city where the venue is located – e.g. "London".
country	String	The country where the venue is located – e.g. "United Kingdom".
postcode	String	The postcode of the venue – e.g. "W1T 7AQ".
url	String	URL to a Ticketmaster page about the venue.
imageUrl	String	The URL of an image of the venue, or a generic image.
state	String	State or province (if applicable).
longitude	Float (or null)	Longitude, where available.

latitude

Float (or null)

Latitude, where available.

API Method: getCategoryes

The getCategoryes method returns the current category tree which is in use. Using this information, you can establish the category IDs you can query for. It is called with a GetCategoriesRequest object, which takes the following parameters:

GetCategoriesRequest

Table 11: GetCategoriesRequest fields

Field	Type	Nullable?	Notes
apiKey	String	No	
Country	String	No	A comma-delimited list of 2-letter country codes. See Table 14: Available countries.

GetCategoriesResponse

The getCategoryesResponse object contains a single field 'parentCategory', which is an array of ParentCategory objects.

ParentCategory

Table 12: ParentCategory fields

Field	Type	Notes
categoryId	Integer	
name	String	The name of the parent category (given in English).
categories	Category[]	An array of child categories.

Category

Table 13: Category fields

Field	Type	Notes
categoryId	Integer	
name	String	The name of the child category (given in English)

Alternative API Access

As an alternative to the SOAP API, requests can be made using a simple HTTP GET request.

Results are returned as JSON by default, or optionally as XML depending on the `Accept` header sent with the request.

URL structure

The endpoint for requests is:

`http://ticketmaster.productserve.com/v3/<method>`

Where `<method>` is one of `event` or `categories`.

To this URL, a variety of required and optional parameters are added as a query string, as detailed below.

These parameters correspond with those detailed for the SOAP API.

Required parameters (for both methods)

`apiKey` – Your API key

`country` – Countries the request is for. Comma-separated list of country codes.

Method: /event

This method is used to retrieve details of events. It corresponds to the `findEvents` SOAP method.

optional parameters

pagination

`resultsPerPage`

`currentPage`

These parameters follow the guidelines given in Table 1: FindEventsRequest fields

sorting

`sort.field`

`sort.order`

Valid values for these parameters are as given in Table 2: FindEventsRequest_Sort fields

Updated since

`updatedSince`

If passed, this should have a date value in the following format: 2012-11-26 16:23:37

filtering

`filter.<object>.<field>`

Valid values for <object> and <filter> are as described in Table 4: FindEventsRequest_Filter fields

For example, requesting cancelled events:

```
filter.event.status=cancelled
```

where multiple values are desired for a given filter, the parameter name is followed by [].

In this example, we are looking for events where the venue's city is either Nottingham or London:

```
filter.venue.city[]=Nottingham&filter.venue.city[]=London
```

Method: /categories

This method is used to retrieve the category tree. It corresponds to the `getCategories` SOAP method.

No additional parameters are supported for this call.

Tracking links

You will need to prefix any URLs from the API that you wish to deep link to with an Affiliate window tracking redirect.

This ensures traffic is redirected through our domain in order to log the click and recognize any referrals made against your publisher id.

Building the tracking link

1. Choose the event landing page URL you wish to link to. This might be the URL for an event, artist or venue obtained from the API.
2. Create the tracking URL.

US and CA links:

All US and CA links should be prefixed with:

```
http://www.awin1.com/awclick.php?awinmid=4103&awinaffid=<publisherId>&platform=tm&clickref=<clickRef>&p=
```

Other Links (UK and other European links):

Other links should be prefixed with:

```
http://www.awin1.com/awclick.php?awinmid=3589&awinaffid=<publisherId>&platform=tm&clickref=<clickRef>&p=
```

Note: where <publisherId> is the ID of your Affiliate Window account, and <clickRef> is an optional Click reference.

3. URL-encode the URL you have from step 1 and append it to the tracking URL from step 2.

Example

If your Publisher ID is 45628 and you wish to link to the UK event 'London Fields' with a Click Reference of 'test':

[http://www.awin1.com/awclick.php?awinmid=3589&awinaffid=45628&platform=tm&clickref=test&p=http%3A%2F%2Fwww.ticketmaster.co.uk%2Fevent%2F37004B6CEA41C362%3Fcamefrom%3DCFC UK BUYAT%26brand%3D%5B%3DBRAND%3D%5D](http://www.awin1.com/awclick.php?awinmid=3589&awinaffid=45628&platform=tm&clickref=test&p=http%3A%2F%2Fwww.ticketmaster.co.uk%2Fevent%2F37004B6CEA41C362%3Fcamefrom%3DCFC%20UK%20BUYAT%26brand%3D%5B%3DBRAND%3D%5D)

Further information on creating links can be found in the Wiki:

http://wiki.affiliatewindow.com/index.php/FAQ_Publisher_Technical#Linking

Appendix 1: Country Codes

Table 14: Available countries

Country Code	Country Name
UK	United Kingdom
US	United States of America
CA	Canada
SE	Sweden
NO	Norway
DE	Germany
NL	Netherlands
FI	Finland
DK	Denmark
AT	Austria

Appendix 2: Example SOAP Request

In this example, we request events with the following criteria:

- The event should have a parent category of 10001 (Music)
- The venue is in London
- Artists have category 200 (Metal/Hard Rock)
- We only want events which have been updated since midnight on the 23rd of January 2014.
- We would like the results in ascending date order

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns1="http://ticketmaster.productserve.com/v3/soap.php"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
>
<SOAP-ENV:Body>
  <ns1:findEvents>
    <request xsi:type="ns1:FindEventsRequest">
      <resultsPerPage xsi:type="xsd:int">1</resultsPerPage>
      <currentPage xsi:type="xsd:int">1</currentPage>
      <sort xsi:type="ns1:FindEventsRequest_Sort">
        <field xsi:type="xsd:string">eventDate</field>
        <order xsi:type="xsd:string">ASC</order>
      </sort>
      <filters SOAP-ENC:arrayType="ns1:FindEventsRequest_Filter[2]"
xsi:type="ns1:ArrayOfFindEventsRequest_Filter">
        <item xsi:type="ns1:FindEventsRequest_Filter">
          <object xsi:type="xsd:string">event</object>
          <field xsi:type="xsd:string">parentCategoryId</field>
          <values SOAP-ENC:arrayType="xsd:int[1]" xsi:type="SOAP-
ENC:Array">
            <item xsi:type="xsd:int">10001</item>
          </values>
        </item>
        <item xsi:type="ns1:FindEventsRequest_Filter">
          <object xsi:type="xsd:string">artist</object>
          <field xsi:type="xsd:string">categoryId</field>
          <values SOAP-ENC:arrayType="xsd:int[1]" xsi:type="SOAP-
ENC:Array">
            <item xsi:type="xsd:int">200</item>
          </values>
        </item>
      </filters>
      <updatedSince xsi:type="xsd:string">2014-01-23
00:00:00</updatedSince>
      <apiKey xsi:type="xsd:string">abdef0123456789</apiKey>
      <country xsi:type="xsd:string">UK</country>
    </request>
  </ns1:findEvents>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Appendix 3: Example SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns1="http://ticketmaster.productserve.com/v3/soap.php"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
>

<SOAP-ENV:Body>
  <ns1:findEventsResponse>
    <return xsi:type="ns1:FindEventsResponse">
      <details xsi:type="ns1:Details">
        <totalResults xsi:type="xsd:int">121</totalResults>
        <totalPages xsi:type="xsd:int">121</totalPages>
        <currentPage xsi:type="xsd:int">1</currentPage>
        <resultsPerPage xsi:type="xsd:int">1</resultsPerPage>
      </details>
      <results SOAP-ENC:arrayType="ns1:Event[5]"
xsi:type="ns1:ArrayOfEvent">
        <item xsi:type="ns1:Event">
          <eventId xsi:type="xsd:int">1035032</eventId>
          <ticketmasterEventId
xsi:type="xsd:string">36004AD5DBBD979C</ticketmasterEventId>
          <status xsi:type="xsd:string">onSale</status>
          <name xsi:type="xsd:string">Dream Theater</name>
          <url
xsi:type="xsd:string">http://www.ticketmaster.co.uk/event/36004AD5
DBBD979C?camefrom=CFC_UK_BUYAT&brand=[=BRAND=]</url>
          <eventDate xsi:type="xsd:string">2014-02-13
18:30:00</eventDate>
          <onSaleDate xsi:type="xsd:string">0000-00-00
00:00:00</onSaleDate>
          <preSaleDate xsi:type="xsd:string">2013-06-26
10:00:00</preSaleDate>
          <category xsi:type="xsd:string">Hard
Rock/Metal</category>
          <categoryId xsi:type="xsd:int">200</categoryId>
          <parentCategory
xsi:type="xsd:string">Music</parentCategory>
          <parentCategoryId
xsi:type="xsd:int">10001</parentCategoryId>
          <minPrice xsi:type="xsd:float">35</minPrice>
          <maxPrice xsi:type="xsd:float">55</maxPrice>
          <description xsi:type="xsd:string">...</description>
          <currency xsi:type="xsd:string">GBP</currency>
```

```

        <artists SOAP-ENC:arrayType="ns1:Artist[1]"
xsi:type="ns1:ArrayOfArtist">
            <item xsi:type="ns1:Artist">
                <artistId xsi:type="xsd:int">73471</artistId>
                <ticketmasterArtistId
xsi:type="xsd:int">744581</ticketmasterArtistId>
                <name xsi:type="xsd:string">Dream Theater</name>
                <url
xsi:type="xsd:string">http://www.ticketmaster.co.uk/Dream-Theater-
tickets/artist/744581?camefrom=CFC_UK_BUYAT&brand=[=BRAND=]</u
rl>
                <imageUrl
xsi:type="xsd:string">http://media.ticketmaster.com/tm/en-
us/tmimages/TM_GenCatImgs_Music_BW.jpg</imageUrl>
                <category xsi:type="xsd:string">Hard
Rock/Metal</category>
                <categoryId xsi:type="xsd:int">200</categoryId>
                <parentCategory
xsi:type="xsd:string">Music</parentCategory>
                <parentCategoryId
xsi:type="xsd:int">10001</parentCategoryId>
            </item>
        </artists>
        <venue xsi:type="ns1:Venue">
            <venueId xsi:type="xsd:int">47501</venueId>
            <ticketmasterVenueId
xsi:type="xsd:int">444077</ticketmasterVenueId>
            <name xsi:type="xsd:string">O2 Apollo
Manchester</name>
            <street xsi:type="xsd:string">Stockport Road</street>
            <city xsi:type="xsd:string">Manchester</city>
            <country xsi:type="xsd:string">United
Kingdom</country>
            <postcode xsi:type="xsd:string">M12 6AP</postcode>
            <url
xsi:type="xsd:string">http://www.ticketmaster.co.uk/O2-Apollo-
Manchester-tickets-
Manchester/venue/444077?camefrom=CFC_UK_BUYAT&brand=</url>
            <imageUrl
xsi:type="xsd:string">http://media.ticketmaster.co.uk/dbimages/214
8v.jpg</imageUrl>
            <state xsi:type="xsd:string"></state>
            <latitude />
            <longitude />
            <state xsi:type="xsd:string"></state>

        </venue>
    </item>
</results>
</return>
</ns1:findEventsResponse>
</SOAP-ENV:Body>

```

</SOAP-ENV:Envelope>

Appendix 4: Example HTTP GET Request

The following URL mirrors the example given in Appendix 2:

- The event should have a parent category of 10001 (Music)
- The venue is in London
- Artists have category 200 (Metal/Hard Rock)
- We only want events which have been updated since midnight on the 23rd of January 2014.
- We would like the results in ascending date order

```
http://ticketmaster.productserve.com/v3/event?apiKey=abcdef0123456789abcdef0123456789&country=UK&filter.event.parentCategoryId=10001&filter.venue.city=London&filter.artist.categoryId=200&updatedSince=2014-01-23%2000:00:00&sort.field=eventDate&sort.order=ASC&resultsPerPage=1&currentPage=1
```


Appendix 5: Example HTTP GET Responses

Depending on the Accept header, the data is returned in either JSON or XML formats.

With an Accept header of application/json (and by default), the following JSON is returned:

```
{
  "details":{
    "totalResults":17,
    "totalPages":17,
    "currentPage":1,
    "resultsPerPage":1
  },
  "results":[
    {
      "eventId":18744,
      "ticketmasterEventId":"1F004944F97B7D5E",
      "status":"onSale",
      "name":"Cauldron",
      "url":
"http://www.ticketmaster.co.uk/event/1F004944F97B7D5E?camefrom=CFC_UK_BUYAT&#38;brand=[=BRAND=]",
      "eventDate":"2014-11-28 19:30:00",
      "onSaleDate":"0000-00-00 00:00:00",
      "preSaleDate":"0000-00-00 00:00:00",
      "category":"Hard Rock\\Metal",
      "categoryId":200,
      "parentCategory":"Music",
      "parentCategoryId":10001,
      "minPrice":"10.00",
      "maxPrice":"10.00",
      "currency": "GBP",
      "description": "...",
      "artists":[
        {
          "artistId":3480,
          "ticketmasterArtistId":1499847,
          "name":"Cauldron",
          "url":
"http://www.ticketmaster.co.uk/Cauldron-tickets/artist/1499847?camefrom=CFC_UK_BUYAT&#38;brand=[=BRAND=]",
          "imageUrl":
"http://media.ticketmaster.co.uk/tmimages/TM_GenCatImgs_Music_BW.jpg",
          "category":"Hard Rock\\Metal",
          "categoryId":200,
          "parentCategory":"Music",
          "parentCategoryId":10001
        }
      ]
    }
  ]
}
```

```
    }  
  ],  
  "venue":{  
    "venueId":1447,  
    "ticketmasterVenueId":255546,  
    "name":"Barfly Camden",  
    "street":"49 Chalk Farm Road",  
    "city":"London",  
    "country":"United Kingdom",  
    "postcode":"NW1 8AN",  
    "url":"http:\\\\www.ticketmaster.co.uk\\Barfly-Camden-  
tickets-London\\venue\\255546?camefrom=CFC_UK_BUYAT&brand=",  
    "imageUrl":  
"http:\\\\media.ticketmaster.co.uk\\dbimages\\386v.gif",  
    "state":"",  
    "longitude":null,  
    "latitude":null,  
  }  
}  
]  
}
```

With an Accept header of application/xml, the response looks like this:

```
<?xml version="1.0"?>  
<data>  
  <details>  
    <totalResults>17</totalResults>  
    <totalPages>17</totalPages>  
    <currentPage>1</currentPage>  
    <resultsPerPage>1</resultsPerPage>  
  </details>  
  <results>  
    <node>  
      <eventId>18744</eventId>  
      <ticketmasterEventId>1F004944F97B7D5E</ticketmasterEventId>  
      <status>onSale</status>  
      <name>Cauldron</name>  
      <url>  
http://www.ticketmaster.co.uk/event/1F004944F97B7D5E?camefrom=CFC_  
UK_BUYAT&brand=[=BRAND=]</url>  
      <eventDate>2012-11-28 19:30:00</eventDate>  
      <onSaleDate>0000-00-00 00:00:00</onSaleDate>  
      <preSaleDate>0000-00-00 00:00:00</preSaleDate>  
      <category>Hard Rock/Metal</category>  
      <categoryId>200</categoryId>  
      <parentCategory>Music</parentCategory>  
      <parentCategoryId>10001</parentCategoryId>  
      <minPrice>10.00</minPrice>  
      <maxPrice>10.00</maxPrice>
```

```
<currency>GBP</currency>
<description>...</description>
<artists>
  <node>
    <artistId>3480</artistId>
    <ticketmasterArtistId>1499847</ticketmasterArtistId>
    <name>Cauldron</name>
    <url>http://www.ticketmaster.co.uk/Cauldron-
tickets/artist/1499847?camefrom=CFC_UK_BUYAT&#38;brand=[=BRAND
=]</url>
    <imageUrl>
http://media.ticketmaster.co.uk/tmimages/TM_GenCatImgs_Music_BW.jp
g
    </imageUrl>
    <category>Hard Rock/Metal</category>
    <categoryId>200</categoryId>
    <parentCategory>Music</parentCategory>
    <parentCategoryId>10001</parentCategoryId>
  </node>
</artists>
<venue>
  <venueId>1447</venueId>
  <ticketmasterVenueId>255546</ticketmasterVenueId>
  <name>Barfly Camden</name>
  <street>49 Chalk Farm Road</street>
  <city>London</city>
  <country>United Kingdom</country>
  <postcode>NW1 8AN</postcode>
  <url>http://www.ticketmaster.co.uk/Barfly-Camden-tickets-
London/venue/255546?camefrom=CFC_UK_BUYAT&#38;brand=</url>
  <imageUrl>
http://media.ticketmaster.co.uk/dbimages/386v.gif</imageUrl>
  <state></state>
  <longitude></longitude>
  <latitude></latitude>
</venue>
</node>
</results>
</data>
```