Ticketmaster API

Version 2

Table of Contents

Tables

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, affiliate tracking 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 2.

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 1

- The API now additionally supports 'country' values of 'US' and 'CA' for getting details of events in America and Canada. This is dependent on being joined to the relevant affiliate programme.
- URLs are no longer returned as affiliate deeplinks.
- An alternative interface to the API is provided, which uses simpler HTTP GET requests. This is described towards the end of this document (see: Alternative API Access).

API Overview

WSDL

The current version of the Ticketmaster service WSDL can be found at:
http://ticketmaster.productserve.com/v2/soap.php?wsdl

User authentication

An API key is required with each request (see Table 1: Request 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 is one API method available: getEvents.

API Method: getEvents

The getEvents method allows you to retrieve a paginated list of event details. The Request object takes the following parameters:

Request

Table 1: Request fields

Field	Type	Nillable?	Notes
apiKey	string	No	
country	string	No	One of “UK”, “US” or “CA”.
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	Request_Sort	Yes	See Table 4: Request_Filter fields
filters	Request_Filter[]	Yes	Array of Request_Filter objects. See Table 4: Request_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.

Request_Sort

The Request_Sort object specifies the order in which the results are returned.

Table 2: Request_Sort fields

Field	Type	Nillable?	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	See Table 12: Child categories.
minPrice	The minimum price of tickets to the event
maxPrice	The maximum price of tickets to the event

Request_Filter

The Request_Filter object limits the results to certain criteria; it takes the following parameters:

Table 4: Request_Filter fields

Field	Type	Nillable?	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

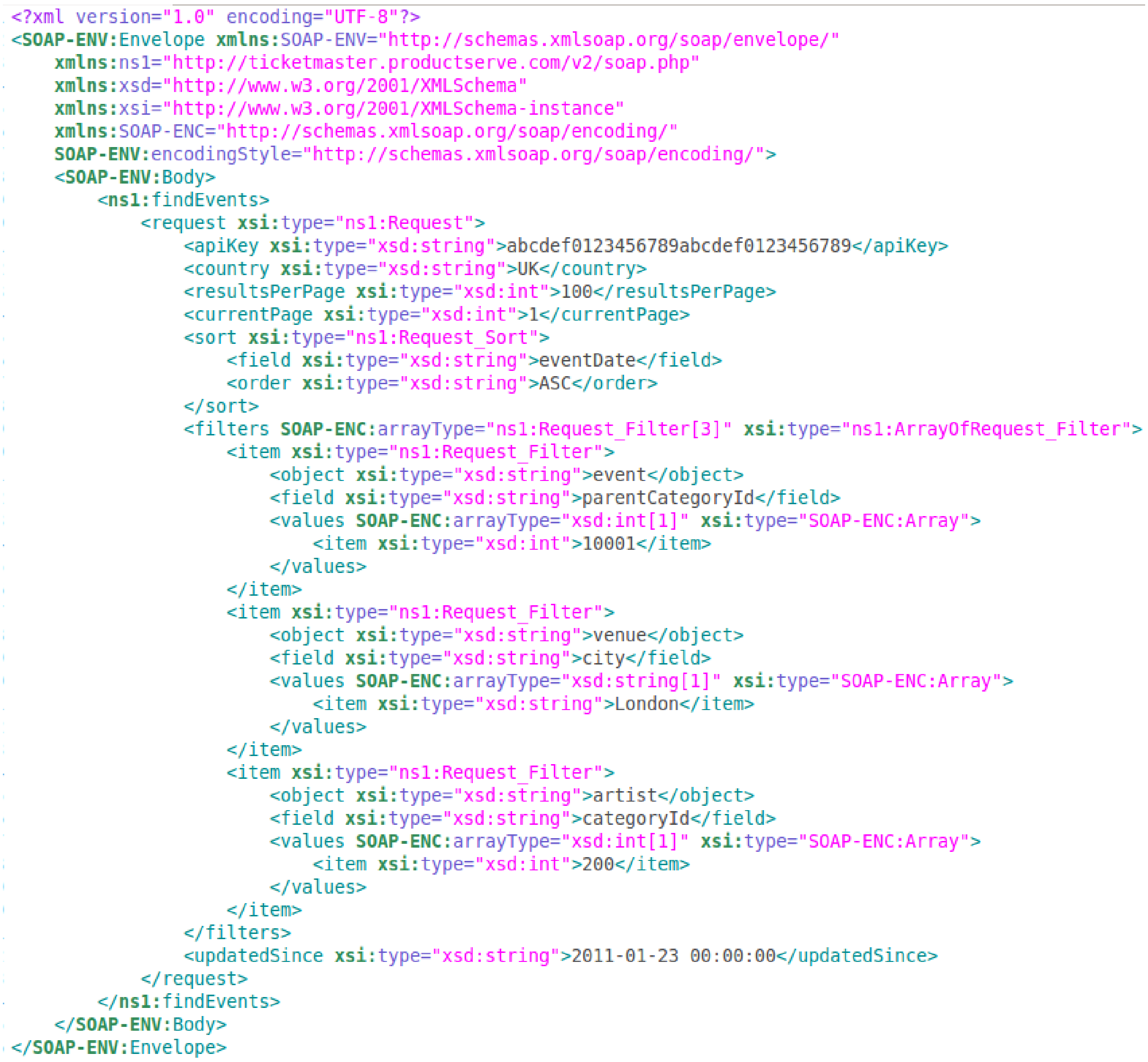
Object	Field	Notes
event	ticketmasterEventId	The event ID provided by Ticketmaster
	categoryId	See Table 12: Child categories.
	parentCategoryId	See Table 11: Parent categories.
	status	Possible values are: “onSale”, “notOnSale”, “cancelled”, and “removed”. If this filter is not given, the results are implicitly filtered to “onSale”.
artist	name	The artist name
	ticketmasterArtistId	The artist ID provided by Ticketmaster.
	categoryId	See Table 12: Child categories.
	parentCategoryId	See Table 11: Parent categories.
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”

Example 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 2011.
- We would like the results in ascending date order

The following has been formatted for clarity:



```xml
<?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/v2/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:Request">
                <apiKey xsi:type="xsd:string">abcdef0123456789abcdef0123456789</apiKey>
                <country xsi:type="xsd:string">UK</country>
                <resultsPerPage xsi:type="xsd:int">100</resultsPerPage>
                <currentPage xsi:type="xsd:int">1</currentPage>
                <sort xsi:type="ns1:Request_Sort">
                    <field xsi:type="xsd:string">eventDate</field>
                    <order xsi:type="xsd:string">ASC</order>
                </sort>
                <filters SOAP-ENC:arrayType="ns1:Request_Filter[3]" xsi:type="ns1:ArrayOfRequest_Filter">
                    <item xsi:type="ns1:Request_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:Request_Filter">
                        <object xsi:type="xsd:string">venue</object>
                        <field xsi:type="xsd:string">city</field>
                        <values SOAP-ENC:arrayType="xsd:string[1]" xsi:type="SOAP-ENC:Array">
                            <item xsi:type="xsd:string">London</item>
                        </values>
                    </item>
                    <item xsi:type="ns1:Request_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">2011-01-23 00:00:00</updatedSince>
            </request>
        </ns1:findEvents>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

The Response object contains the following fields:

Table 6: Response fields

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

Details

The details object contains pagination information. It contains the following fields:

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

Event objects contain the following fields:

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. See Table 12: Child categories.
categoryId	Integer	See Table 12: Child categories.
parentCategory	String	Human-readable description of the category given in categoryId. See Table 11: Parent categories.
parentCategoryId	Integer	See Table 11: Parent categories.
minPrice	Decimal	The minimum price of tickets (currency depending on the country).
maxPrice	Decimal	The maximum price of tickets.
artists	Artist[]	An array of Artist objects. See Table 9: Artist fields.
venue	Venue	A Venue object. See Table 10: Venue fields.

Artist

Artist objects contain the following fields:

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 Table 12: Child categories.
categoryId	Integer	See Table 12: Child categories.
parentCategory	String	Human-readable description of the category given in categoryId. See Table 11: Parent categories.
parentCategoryId	Integer	See Table 11: Parent categories.

Venue

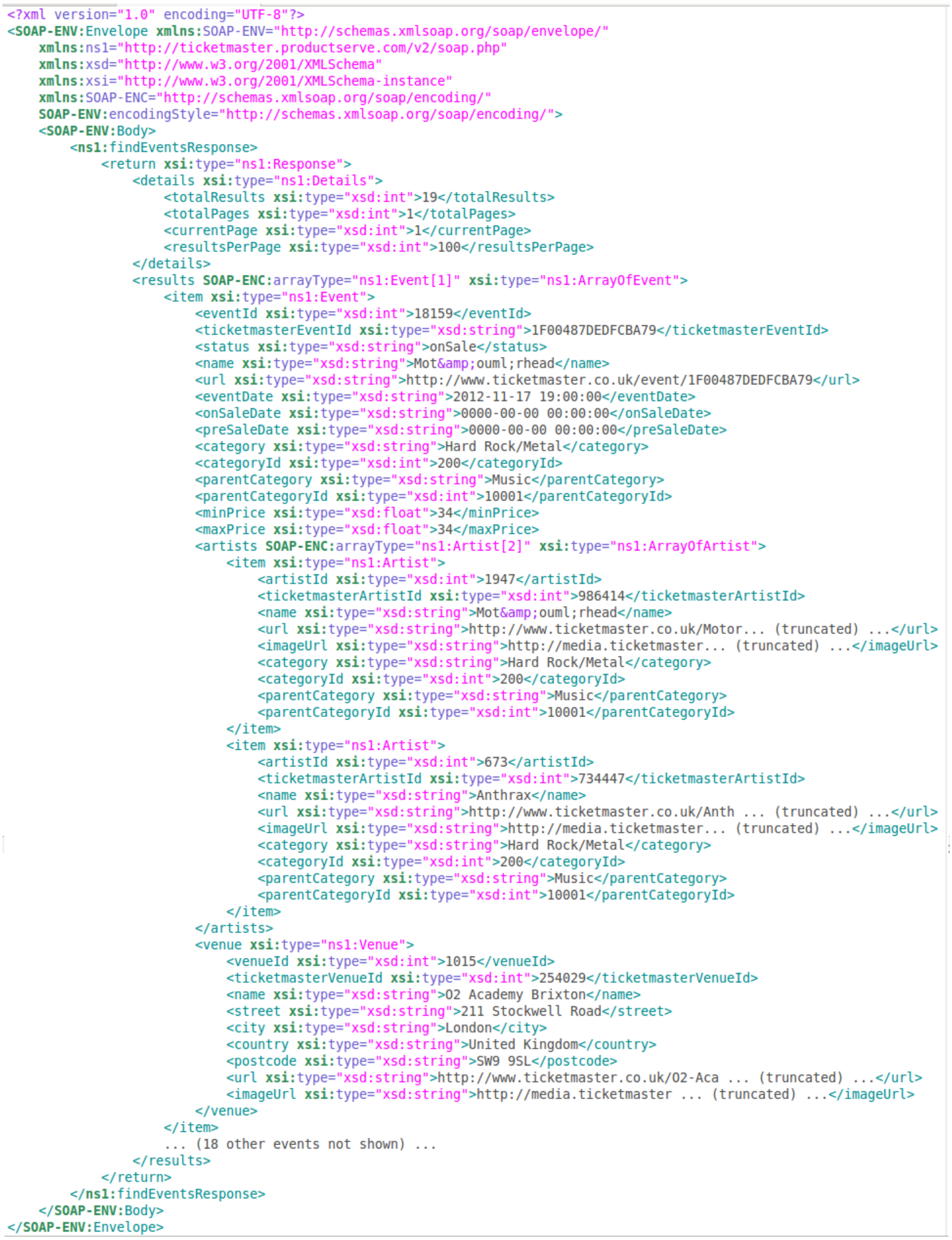
Venue objects contain the following fields:

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 (US and CA only).

Example Response

The following has been formatted for clarity:



```
<?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/v2/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:Response">
                <details xsi:type="ns1:Details">
                    <totalResults xsi:type="xsd:int">19</totalResults>
                    <totalPages xsi:type="xsd:int">1</totalPages>
                    <currentPage xsi:type="xsd:int">1</currentPage>
                    <resultsPerPage xsi:type="xsd:int">100</resultsPerPage>
                </details>
                <results SOAP-ENC:arrayType="ns1:Event[1]" xsi:type="ns1:ArrayOfEvent">
                    <item xsi:type="ns1:Event">
                        <eventId xsi:type="xsd:int">18159</eventId>
                        <ticketmasterEventId xsi:type="xsd:string">1F00487DEDFCBA79</ticketmasterEventId>
                        <status xsi:type="xsd:string">onSale</status>
                        <name xsi:type="xsd:string">Mot&amp;ouml;rhead</name>
                        <url xsi:type="xsd:string">http://www.ticketmaster.co.uk/event/1F00487DEDFCBA79</url>
                        <eventDate xsi:type="xsd:string">2012-11-17 19:00:00</eventDate>
                        <onSaleDate xsi:type="xsd:string">0000-00-00 00:00:00</onSaleDate>
                        <preSaleDate xsi:type="xsd:string">0000-00-00 00:00:00</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">34</minPrice>
                        <maxPrice xsi:type="xsd:float">34</maxPrice>
                        <artists SOAP-ENC:arrayType="ns1:Artist[2]" xsi:type="ns1:ArrayOfArtist">
                            <item xsi:type="ns1:Artist">
                                <artistId xsi:type="xsd:int">1947</artistId>
                                <ticketmasterArtistId xsi:type="xsd:int">986414</ticketmasterArtistId>
                                <name xsi:type="xsd:string">Mot&amp;ouml;rhead</name>
                                <url xsi:type="xsd:string">http://www.ticketmaster.co.uk/Motor... (truncated) ...</url>
                                <imageUrl xsi:type="xsd:string">http://media.ticketmaster... (truncated) ...</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>
                            <item xsi:type="ns1:Artist">
                                <artistId xsi:type="xsd:int">673</artistId>
                                <ticketmasterArtistId xsi:type="xsd:int">734447</ticketmasterArtistId>
                                <name xsi:type="xsd:string">Anthrax</name>
                                <url xsi:type="xsd:string">http://www.ticketmaster.co.uk/Anth ... (truncated) ...</url>
                                <imageUrl xsi:type="xsd:string">http://media.ticketmaster... (truncated) ...</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">1015</venueId>
                            <ticketmasterVenueId xsi:type="xsd:int">254029</ticketmasterVenueId>
                            <name xsi:type="xsd:string">02 Academy Brixton</name>
                            <street xsi:type="xsd:string">211 Stockwell Road</street>
                            <city xsi:type="xsd:string">London</city>
                            <country xsi:type="xsd:string">United Kingdom</country>
                            <postcode xsi:type="xsd:string">SW9 9SL</postcode>
                            <url xsi:type="xsd:string">http://www.ticketmaster.co.uk/02-Aca ... (truncated) ...</url>
                            <imageUrl xsi:type="xsd:string">http://media.ticketmaster ... (truncated) ...</imageUrl>
                        </venue>
                    </item>
                    ... (18 other events not shown) ...
                </results>
            </return>
        </ns1:findEventsResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Alternative API Access

As an alternative to the SOAP API, details of events can be retrieved 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/v2/<country>/event
```

Where <country> is one of UK, US or CA.

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 parameter

`apiKey`

optional parameters

pagination

`resultsPerPage`

`currentPage`

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

sorting

`sort.field`

sort.order

Valid values for these parameters are as given in Table 2: Request_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: Request_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
```

s

Example Request

The following URL mirrors the Example Request given for the SOAP API:

```
http://ticketmaster.productserve.com/v2/UK/event?apiKey=abcdef012345
6789abcdef0123456789&filter.event.parentCategoryId=10001&filter.venu
e.city=London&filter.artist.categoryId=200&updatedSince=2011-01-
23%2000:00:00&sort.field=eventDate&sort.order=ASC
```

Example Response

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

```
{
  "details":{
    "totalResults":17,
    "totalPages":1,
    "currentPage":1,
    "resultsPerPage":50
  },
  "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":"2012-11-28 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",
      "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&amp;brand=",
        "imageUrl":
"http:\/\/media.ticketmaster.co.uk\/dbimages\/386v.gif",
        "state":""
      }
    }, ( … results omitted …)
  ]
}
```

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

```
<?xml version="1.0"?>
<data>
  <details>
    <totalResults>17</totalResults>
    <totalPages>1</totalPages>
    <currentPage>1</currentPage>
    <resultsPerPage>50</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&amp;#38;
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>
      <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&amp;#38;brand=[=BRAND=]</url>
          <imageUrl>
http://media.ticketmaster.co.uk/tmimages/TM_GenCatImgs_Music_BW.jpg
```

```
          </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&amp;amp;brand=</url>
        <imageUrl>
http://media.ticketmaster.co.uk/dbimages/386v.gif</imageUrl>
        <state></state>
      </venue>
    </node>
    ( … results omitted …)
  </results>
</data>
```

Appendix 1: Categories

Table 11: Parent categories

parentCategoryId	Category Name
10001	Music
10002	Arts, Theatre & Comedy
10003	Family & Attractions
10004	Sport
10005	Miscellaneous

Table 12: Child categories

categoryId	Category Name	parentCategoryId
1	Rock/Pop	10001
2	Country/Folk	10001
3	Rap and Hip-Hop	10001
4	Jazz/Blues	10001
5	World	10001
52	More Music	10001
60	Alternative and Indie	10001
200	Hard Rock/Metal	10001
201	Clubs and Dance	10001
202	R&B/Urban Soul	10001
542	Tribute Bands	10001
12	Ballet & Dance	10002
13	Opera	10002
14	Museums	10002

51	Comedy	10002
53	More Arts, Theatre & Comedy	10002
203	Classical	10002
207	Musicals	10002
509	Drama	10002
19	Family Attractions	10003
22	Ice Shows	10003
23	Family Shows	10003
54	Festivals	10003
55	More Family Attractions	10003
218	Flower Shows	10003
232	Ice Rinks	10003
514	Fashion Shows	10003
558	Pantomime	10003
754	Exhibitions	10003
7	Basketball	10004
9	Ice Hockey	10004
11	Football	10004
25	Motorsports	10004
27	Wrestling	10004
30	Golf	10004
31	Tennis/Squash	10004
33	Boxing	10004
102	More Sport	10004

205	Athletics	10004
225	Rugby Union	10004
582	Gymnastics	10004
676	Cricket	10004
677	Equestrian	10004
693	Hockey	10004
695	Rugby League	10004
716	Horse Racing	10004
718	Cycling	10004
729	GAA	10004
742	Martial Arts	10004
0	Miscellaneous	10005
106	Cruise and Sightseeing	10005