DATA THISTLE

Data Thistle Default Feed Specification

v1.1

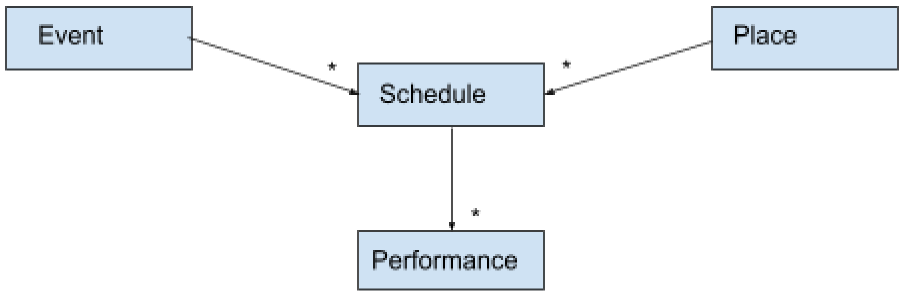
This document describes Data Thistle's default json format for event information. Some details are subject to the agreement with Data Thistle, so some parts may not apply and there may be other features not described below.

Main Structure

Main object types:

- **Event**: the thing happening, with a name, a description and some categorization
- **Place**: a physical location, typically with a name, an address and a latitude/longitude
- **Schedule**: a link between an event and a place, effectively a group of performances
- **Performance**: an actual time when the event is occurring at a place

Various extra fields can be attached to each of these entities as appropriate, these are detailed below, but a rough ER diagram looks something like this:



So informally an event, such as a band, might have a concert at a place, such as an arena or stadium, on a Friday and Saturday. The band could be touring so they might have several performances at other stadiums on dates over several weeks.

The actual json file structure used to represent these entities can be of two types:

- **Default**: The file has two top level fields "events" and "places". Events is an array of event objects each of which contains a "schedules" array, and each of these contains a "place_id" which is a reference to a place object. Places is a list of place objects each of which has a place_id. Schematically, the json file has the following structure:

```
{events: [..
          {..
           schedules: [..
                        {place_id,
                         performances: [.. {}]
                        }
                       ]
          }
         ],
 places: [.. {}]
}
```

- **Inline Places**: The file consists of an array of event objects each of which contains a "schedules" array, but here each schedule contains a place object directly. The structure is as follows:

```
[..
 {..
  schedules: [..
              {place: {}
               performances: [.. {}]
              }
             ]
 }
]
```

The two file structures are equivalent, but the inline places version duplicates the place information on each schedule. This is sometimes more convenient for importing, despite the extra size overhead, and so is available as an option.

As this is json, the order of fields within any object is not guaranteed within the file.

Event

An event object represents a band, show, film or other kind of artistic work being performed, or an activity, meeting or a social gathering.

name	string	required	Name of event to display
sort_name	string	required	Name used for sorting
event_id	number	required	Unique id
id	number		An id used by www.datathistle.com
created_ts	ISO8601 Z format string	required	when the record corresponding to the object was created
modified_ts	ISO8601 Z format string	required	when the record corresponding to the object was last modified
status	string	required	in all but bespoke feeds this will always be the value "live"
description	string		
descriptions	Array of description objects		
tags	Array of strings		
category	string		The main tag
website	string		
phone_numbers	Phone numbers object		
image	Image object		Described below. Used when at most a single image per record are part of the feed service
images	Array of image objects		Described below. Used when multiple images are part of the feed service

An example event object might look like:

```
{
 "event_id": 17949,
 "name": "Andy Irvine",
 "sort_name": "Andy Irvine",
 "descriptions": [ . . . ],
    "website": "http://www.andyirvine.com",
    "tags": ["Folk","Music"],
    "category": "Music",
  }
```

Place

A place object represents a physical place, typically a building.

name	string	required	Name of place to display
sort_name	string	required	Name used for sorting
place_id	number	required	
created_ts	ISO8601 Z format string	required	when the record corresponding to the object was created
modified_ts	ISO8601 Z format string	required	when the record corresponding to the object was last modified
status	string	required	in all but bespoke feeds this will always be the value "live"
address	string		
town	string		
postal_code	string		
country_code	string		Always “GB”
loc	Loc object		
description	string		
descriptions	Array of description objects		
tags	Array of strings		

website	string		
phone_numbers	Phone numbers object		
email	string		
image	Image object		Described below. Used when at most a single image per record are part of the feed service
images	Array of image objects		Described below. Used when multiple images are part of the feed service

An example place object might look like:

```
{
 "name": "The Star Hall",
 "sort_name": "Star Hall",
 "modified_ts": "2021-06-15T11:53:21Z",
 "created_ts": "2020-03-14T10:15:23Z",
 "address": "9 Laws Lane, Finedon",
 "postal_code": "NN9 5LU",
 "town": "Wellingborough",
 "country_code": "GB",
 "place_id": 127036,
 "loc": {
   "latitude": "52.3375024",
   "longitude": "-0.6560897"
 },
 "tags": [
   "Theatres"
 ],
 "status": "live"
}
```

Schedule

A schedule acts a join between an event and a place

start_ts	ISO8601 + format string	required	
end_ts	ISO8601 + format string	required	
place_id	number	required	
place	Place object		
performances	Array of performance objects	required	

For the inline places version of the feed file the place field contains a full copy of the place object, for the default version the place object, if present at all, may contain only minimal information such as the place name and town.

The start_ts and end_ts give some boundaries on the set of actual performance times grouped together by the schedule, so might be a good basis for an index to rapidly find possible schedules lying within a given time period. The start_ts time is usually the midnight of the day of the first performance, and end_ts is the midnight of the day of the last performance, though note for this purpose we treat a performance starting in the early hours of the day as 'belonging' to the previous day. For a schedule containing only one performance, the start_ts and end_ts will be the same.

An example schedule object in the default format might look like:

```
{
 "start_ts": "2021-10-02T19:30:00+01:00",
 "end_ts": "2021-10-02T19:30:00+01:00",
 "place_id": 59772,
 "performances": [ … ]
}
```

Where the place_id 59772 refers to a place object given elsewhere in the feed file.

There is no guarantee that there will not be more than one schedule present in the file for any given event/place combination.

Performance

Captures information about a particular performance

ts	ISO8601 + format string or YYYY-MM-DD string	required	Timestamp of the start of the performance
time_unknown	string		

duration	number		Expect duration of the event in minutes
links	Array of link objects		Described below
tickets	Array of ticket objects		Described below
properties	Properties object		a json object from strings to strings, booleans or numbers which is used to store arbitrary extra fields appropriate to the parent object

A regular performance will have a ts field as a timestamp in a full ISO8601 format string, and the time_unknown field absent. If for some reason we are not certain about the start time for a performance the ts field will contain a date in a YYYY-MM-DD format string indicating the day of the performance, and the time_unknown field is used to elaborate. It can contain the following values:

- "Times tbc" times have not yet been confirmed
- "Times vary" times vary through the day
- "By appointment only" times will be agreed on booking

An example regular performance object might look like:

```
{
 "ts": "2021-10-05T19:00:00+01:00",
 "duration": 120,
 "links": [
   {
     "type": "booking",
     "url": "https://www.buytickets.com/h2s04"
   }
 ],
 "tickets": [
   {
     "type": "Standard",
     "currency": "GBP",
     "min_price": 22
   }
 ]
}
```

And an example performance using times_unknown might appear as:

```
{
 "ts": "2021-08-06",
 "time_unknown": "Times tbc",
 . . .
}
```

The main properties for performances are:

event.festival	string	The name of any wider festival this event is part of
event.session	string	The name of a session or series this event is part of
performance.cancelled	boolean	True if the performance has been cancelled
performance.sold-out	boolean	True if the performance is sold out
descriptions	Array of description objects	Extra descriptions applicable only to this performance
event.minimum-age	string	Indication of any age restrictions on this performance if known

Other object types

Details of other types of objects mentioned above.

Description

type	string	required	Either "description.official" or "description.list.default"
description	string	required	

A description object pairs a description text with a type which indicates the description's source - official is used if the description came originally from a third party, otherwise the description was sourced internally.

Ticket

type	string	required	
currency	string	required	Usually “GBP“
min_price	number		
max_price	number		
description	string		

Type is a rough text description for the type of ticket the is, typical values are: “Standard“, “Concession“, “Family“, “Children“, although they can be any text value. A min_price of 0 denotes a free ticket. Description gives further information about the ticketing and should be displayed alongside the price. It is possible for both min_price and max_price to be absent, in which case the description is the only information available.

Link

type	string	required	Always “booking”
url	string	required	

The only link type currently supported is booking links for purchasing tickets.

Image

url	string	required	Url to download image, may be in any format
width	number	required	
height	number	required	
alt_text	string	required	Suitable for populating img element's alt_text attribute

picture_credits	string		

If the field "picture_credits" is present and non-empty this must be shown to any end user of the image. If this is not possible, please let us know so we can remove these images from your feed.

The url given should be used to download the image and not used directly within end user web pages or applications.

Loc

latitude	string	required	
longitude	string	required	

Should really be numbers

Phones

info	string		
box_office	string		

The info field holds a phone number for information and general enquiries, and the box_office field is used specifically for booking. Where these numbers are the same, the info field is used.

Tags

Tags are arbitrary strings used to classify a given event.

Feed Usage

The following notes about feed importing and displaying the contained data are worth highlighting:

- The feed is a snapshot and the data can be modified and removed with each new update. Accordingly care must be taken to not cache the data for a long period of time, and otherwise clear out data that is no longer present in the feed.

- If a time_unknown field is present for a performance, care should be taken not to display the ts field as a regular time without clarification, otherwise the event would appear to always start at midnight.
- In a schedule, the list of performances can be quite long so avoid simply listing all performance times without considering the page space available in your application.
- Some events occur at multiple places, especially those that are an aggregation of other events, such as a festival. In these cases, we usually adopt the convention that the place has a name ‘Various venues…’ with little further information other than the loc field giving a latitude and longitude.