# Spektrix API v1.0 Documentation

*Revisions:*

| | | |
|---|---|---|
| MPS | 04-07-08 | Initial draft |
| MPS | 04-06-09 | Revised to include most recent additions |
| MPS | 01-07-09 | Corrected urls. |
| MPS | 30-09-09 | More corrections. |
| MAN | 08-10-09 | Added various elements to Event. |
| MPS | 11-11-09 | Added new fields to EventTime. |
| MPS | 10-12-09 | Added WebEventId and WebInstanceId field documentation. |
| RB | 05-11-10 | Added documentation for all RESTful methods that return instance attributes. |
| MPS | 28-01-11 | Added info about polling. |
| AJ | 05-10-11 | Added info about OnSaleOnWeb. |

## Contents

# 1. Introduction

The Spektrix Api v1.0 is designed to allow 3rd party web applications to retrieve an events feed from the Spektrix system. The feed provides basic event information and the dates and times of the performances (or occurrences) of each event. The information is thus quite ‘high level’ and does not include a lot of the intricate details that the Spektrix system itself deals with - the aim at v1.0 is to allow 3rd party websites to be able to filter and list events to provide event calendar and event list functionality. To proceed to selling the event the 3rd party website should link to the pages hosted on the Spektrix site. The details of this are outside the scope of this document but more information can be found in the Spektrix Web Integration Guide.

The Spektrix API v1.0 supports two protocols, SOAP and REST. The following sections give an overview of the interfaces available on each protocol.

# 2. System Concepts

*Note that the terms event and event instance used in this section refer to internal Spektrix system data, not the data available through the API.*

The Spektrix system uses two concepts to deal with shows and performances: *Events* and *Event Instances*. An *Event* represents things such as a show or production or recital series and an *Event Instance* represents the specific times or performances when that event will take place. Each *Event* therefore has one or more *Event Instances*.

An *Event* contains details of the *name*, *description*, and *duration* of a show along with a collection of *Event Instances* and custom attributes (amongst other things). The attributes are controlled by the client and can be used to store any additional information that they need or that the 3rd party website needs. They can also be used as criteria for reporting.

Each *Event Instance* contains details of the date and time of an occurrence of an *Event* and a collection of custom attributes for this particular occurrence. Along with this a seating plan (amongst other things) is stored. The plan has a name which is typically be the name of the venue in which the performance will be held. This is set on an instance basis to allow shows to be performed over a number of different venues at different times. In fact, most of the information about specific instances of events can be customised on an *Event Instance* basis.

# 3. Polling

To improve the performance of your website it can be advantageous to poll the Spektrix API and cache the data locally. While this can be an effective way to increase performance polling too frequently can put unnecessary load on the Spektrix servers. To discourage abuse, the Spektrix API caches the data itself. If polling occurs at too high a frequency than the API will simply return the cached record, not the fresh version. If you do poll for data, *please do not poll more than once every 30 minutes for data.*

# 4. SOAP Events Interface

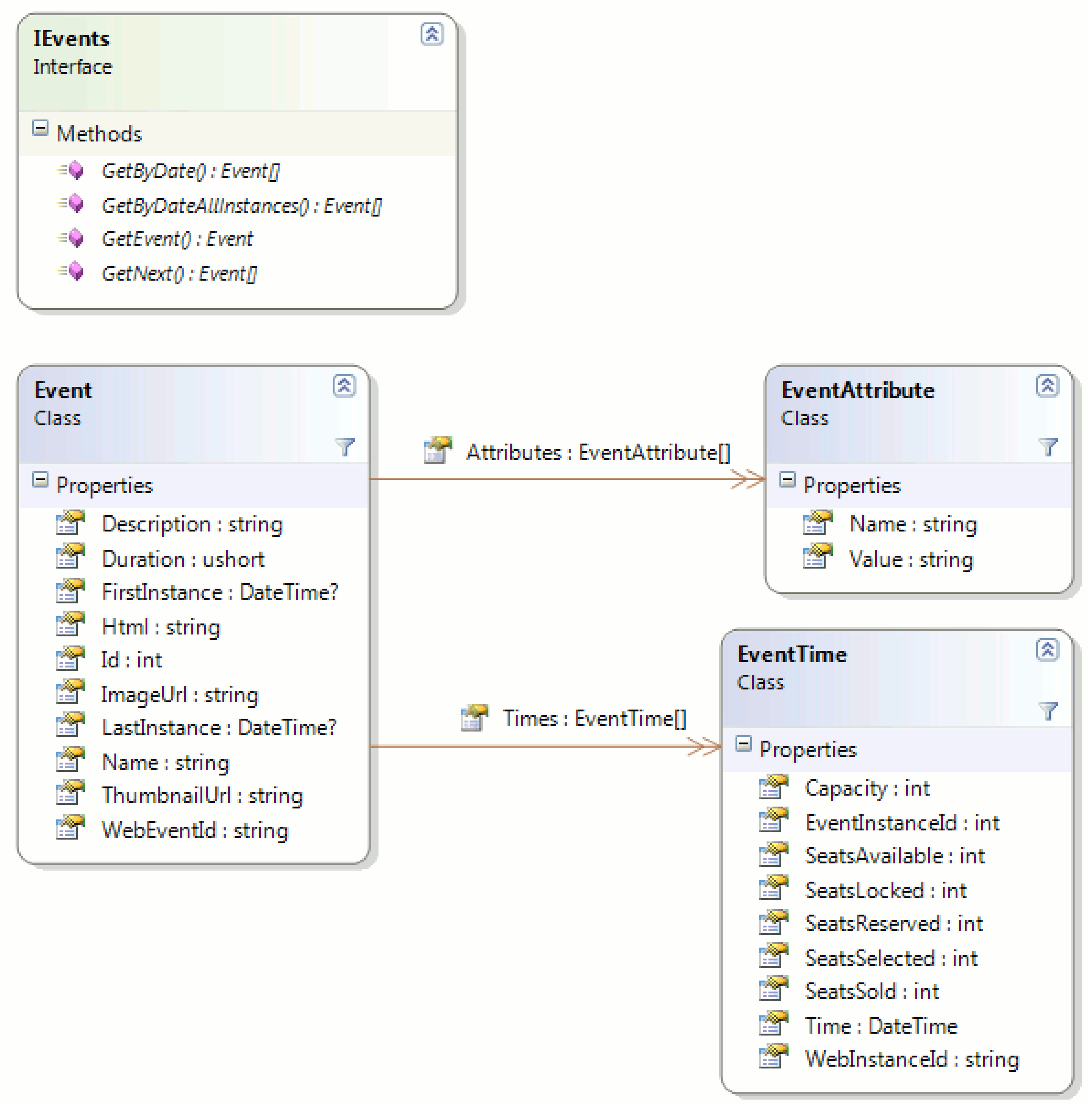

**Figure 1 - UML Style Overview**

The SOAP Events web service has a schema that maps onto the conceptual UML diagram shown in Figure 1. Data are represented as properties of dumb classes and the API functions are represented as methods of interfaces. Please refer to Section 6.1 for a detailed description of the data obtained from the interface.

The service can be accessed at

`https://system.spektrix.com/<client>/api/v1/events.svc`

and it's WSDL can be obtained through

`https://system.spektrix.com/<client>/api/v1/events.svc?wsdl`

where `<client>` should be replaced with the appropriate client name, to be obtained from Spektrix.

## 4.1. GetByDate

Returns an array of events that have instances between the given dates, or all events if no dates are given. The array of EventTime elements in the events returned will only include instances that occur between the dates given.

*Parameters:*

| | | |
|---|---|---|
| **dateFrom** | *Optional* | Inclusive date/time. Only events with instances occurring on or after the value will be returned. |
| **dateTo** | *Optional* | Inclusive date/time. Only events with instances occurring on or before the value will be returned. |

## 4.2. GetByDateAllInstances

Returns an array of events that have instances between the given dates, or all events if no dates are given. This differs from GetByDate in that it returns all the instances associated with each event, rather than just those that meet the date requirement.

Parameters:

| | | |
|---|---|---|
| **dateFrom** | *Optional* | Inclusive date/time. Only events with instances occurring on or after the value will be returned. |
| **dateTo** | *Optional* | Inclusive date/time. Only events with instances occurring on or before the value will be returned. |

## 4.3. GetNext

Returns an array of the next `n` events that have instances that start on or after the current date.

*Parameters:*

| | | |
|---|---|---|
| **n** | *Required* | An integer specifying the number of events to return. |

## 4.4. GetEvent

Returns a single event record given an identifier. Calls to this method will populate the `Event.Html` field.

*Parameters:*

| | | |
|---|---|---|
| **eventId** | *Required* | An integer specifying the event identifier of the event to return. |

# 5. SOAP Authentication Interface

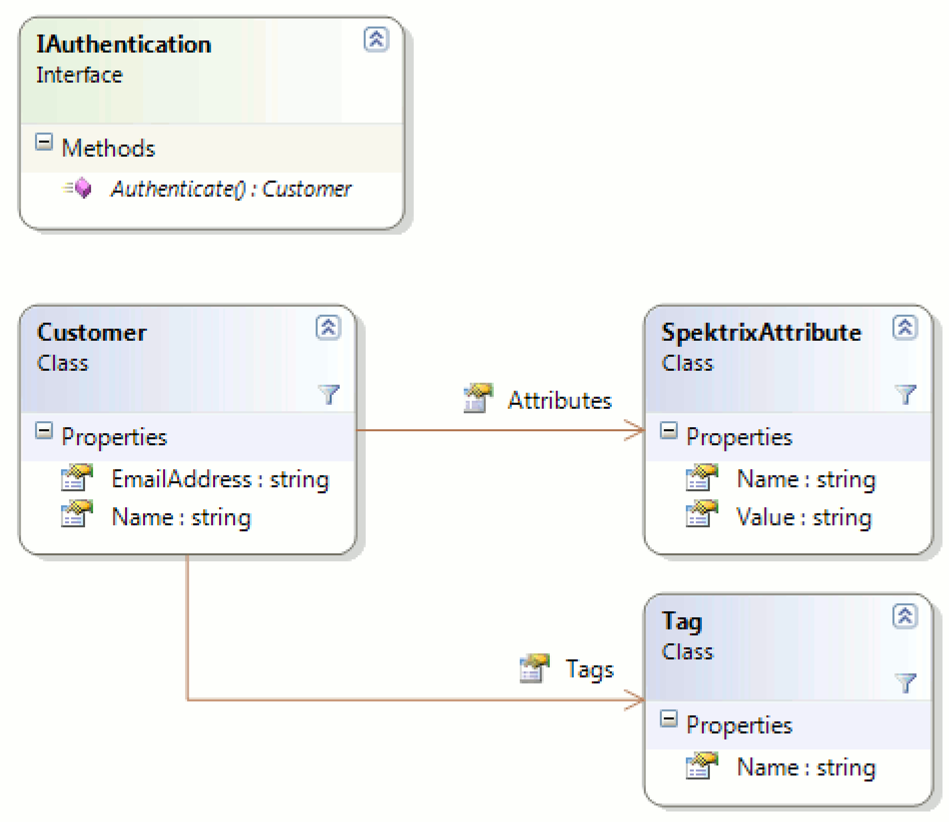

**Figure 2 - UML Overview**

The SOAP Authentication web service has a schema that maps onto the conceptual UML diagram shown in Figure 2Figure 2 - UML Overview. Data are represented as properties of dumb classes and the API functions are represented as methods of interfaces. Please refer to Section 6.1 for a detailed description of the data obtained from the interface.

The service can be accessed at

`https://system.spektrix.com/<client>/api/v1/authentication.svc`

and it's WSDL can be obtained through

`https://system.spektrix.com/<client>/api/v1/authentication.svc?wsdl`

where `<client>` should be replaced with the appropriate client name, to be obtained from Spektrix.

## 5.1. Authenticate

Authenticates a customer, returning their basic details if authentication is successful. Nothing is returned if authentication is not successful.

*Parameters:*

| | | |
|---|---|---|
| **email** | *Required* | String containing the email address of the Customer being authenticated. |
| **password** | *Required* | String containing the password of the Customer being authenticated. |

# 6. RESTful Events Interface

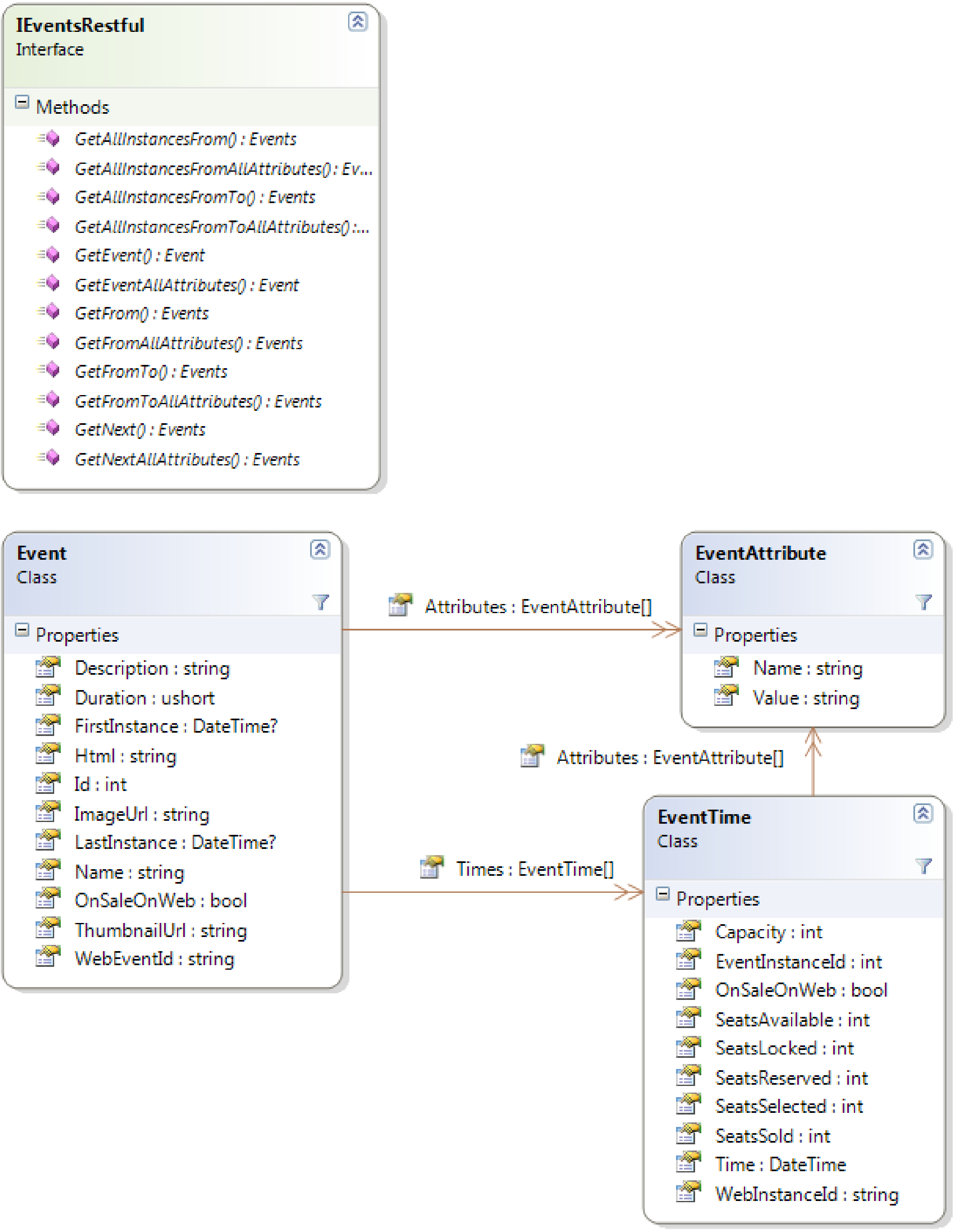

**Figure 3 - UML Overview**

The RESTful Events interface can be conceptually viewed as the UML diagram shown in Figure 3. XML Data is represented by the dumb classes and the RESTful API resources are represented as methods of interfaces. Please refer to Section 6.1 for a detailed description of the data obtained from the interface.

The base address of the Events interface is

`https://system.spektrix.com/<client>/api/v1/eventsrestful.svc`

where `<client>` should be replaced with the appropriate client name, to be obtained from Spektrix.

## 6.1. GetAllInstancesFrom

`.../alltimes/from?date={date}`

Returns a set of events that have instances that start between the given dates, inclusive. This differs from GetFrom in that it returns all the instances associated with each event, rather than just those that meet the date requirement. Custom attributes are included in the returned data for events but not for instances.

*Parameters:*

| | |
|---|---|
| **{date}** | Date and time value. Formatted as `YYYY-MM-DDTHH:mm:ss`. |

## 6.2. GetAllInstancesFromAllAttributes

`...alltimes/allattributes/from?date={from}`

Returns a set of events that have instances that start between the given dates, inclusive. This differs from GetFrom in that it returns all the instances associated with each event, rather than just those that meet the date requirement. Custom attributes are included in the returned data for both events and instances.

*Parameters:*

| | |
|---|---|
| **{date}** | Date and time value. Formatted as `YYYY-MM-DDTHH:mm:ss`. |

## 6.3. GetAllInstancesFromTo

`.../alltimes/between?dateFrom={dateFrom}&dateTo={dateTo}`

Returns a set of events that have instances that start between the given dates, inclusive. This differs from GetFrom in that it returns all the instances associated with each event, rather than just those that meet the date requirement. Custom attributes are included in the returned data for events but not for instances.

*Parameters:*

| | |
|---|---|
| **{dateFrom}** | Date and time value. Formatted as `YYYY-MM-DDTHH:mm:ss`. |
| **{dateTo}** | Date and time value. Formatted as `YYYY-MM-DDTHH:mm:ss`. |

## 6.4. GetAllInstancesFromToAllAttributes

`.../alltimes/allattributes/between?dateFrom={from}&dateTo={to}`

Returns a set of events that have instances that start between the given dates, inclusive. This differs from GetFrom in that it returns all the instances associated with each event, rather than just those that meet the date requirement. Custom attributes are included in the returned data for both events and instances.

*Parameters:*

| | |
|---|---|
| **{dateFrom}** | Date and time value. Formatted as `YYYY-MM-DDTHH:mm:ss`. |
| **{dateTo}** | Date and time value. Formatted as `YYYY-MM-DDTHH:mm:ss`. |

## 6.5. GetEvent

`.../details/{eventId}`

Returns the details of a specific event, including all it’s instances, attributes and wiki-text (rendered as HTML). Custom attributes are included in the returned data for events but not for instances.

*Parameters:*

| | |
|---|---|
| **{eventId}** | Integer value specifying the identifier of the Event. |

## 6.6. GetEventAllAttributes

`.../details/allattributes/{eventId}`

Returns the details of a specific event, including all it’s instances, attributes and wiki-text (rendered as HTML). Custom attributes are included in the returned data for both events and instances.

*Parameters:*

| | |
|---|---|
| **{eventId}** | Integer value specifying the identifier of the Event. |

## 6.7. GetFrom

`.../from?date={date}`

Returns a set of events that have instances that start between the given dates, inclusive. This differs from GetAllInstancesFrom in that the instances associated with each event are filtered by the date parameter. Custom attributes are included in the returned data for events but not for instances.

*Parameters:*

| | |
|---|---|
| **{date}** | Date and time value. Formatted as `YYYY-MM-DDTHH:mm:ss`. |

## 6.8. GetFromAllAttributes

```
...allattributes/from?date={from}
```

Returns a set of events that have instances that start between the given dates, inclusive. This differs from GetAllInstancesFrom in that the instances associated with each event are filtered by the date parameter. Custom attributes are included in the returned data for both events and instances.

*Parameters:*

| | |
|---|---|
| **{date}** | Date and time value. Formatted as `YYYY-MM-DDTHH:mm:ss`. |

## 6.9. GetFromTo

```
.../between?dateFrom={dateFrom}&dateTo={dateTo}
```

Returns a set of events that have instances that start between the given dates, inclusive. This differs from GetAllInstancesFrom in that the instances associated with each event are filtered by the date parameters. Custom attributes are included in the returned data for events but not for instances.

*Parameters:*

| | |
|---|---|
| **{dateFrom}** | Date and time value. Formatted as `YYYY-MM-DDTHH:mm:ss`. |
| **{dateTo}** | Date and time value. Formatted as `YYYY-MM-DDTHH:mm:ss`. |

## 6.10. GetFromToAllAttributes

```
...allattributes/between?dateFrom={from}&dateTo={to}
```

Returns a set of events that have instances that start between the given dates, inclusive. This differs from GetAllInstancesFrom in that the instances associated with each event are filtered by the date parameters. Custom attributes are included in the returned data for both events and instances.

*Parameters:*

| | |
|---|---|
| **{dateFrom}** | Date and time value. Formatted as `YYYY-MM-DDTHH:mm:ss`. |
| **{dateTo}** | Date and time value. Formatted as `YYYY-MM-DDTHH:mm:ss`. |

## 6.11. GetNext

```
.../next?n={n}
```

Returns a list of the next `n` events to occur. Custom attributes are included in the returned data for events but not for instances.

*Parameters:*

| | |
|---|---|
| **{n}** | Integer specifying the number of events to retrieve. |

## 6.12. GetNextAllAttributes

`...allattributes/next?n={n}`

Returns a list of the next `n` events to occur. Custom attributes are included in the returned data for both events and instances.

*Parameters:*

| | |
|---|---|
| **{n}** | Integer specifying the number of events to retrieve. |

# 7. API Data Description

*Note that the event data provided by the API is based mainly on the information stored in the internal system* event *data only. Detailed event instance information, including seating area and venue information are therefore not provided. This data should be displayed using an iframe pointing to the EventDetails.aspx or ChooseSeats.aspx page. More details on the iframe integration can be found in the Spektrix Web Integration Guide.*

The following section details the classes of data that the API provides.

***NB: We do not guarantee the order in which XML elements will appear.***

## 7.1. Event

*Elements:*

| | |
|---|---|
| **Attributes** | An array of EventAttribute elements. These are custom attributes that the client has full control over. These are entered by the client when creating an Event in the Spektrix Admin Interface. |
| **Description** | A 'teaser summary' of the Event. This is unlimited in length. |
| **Duration** | The duration of the Event, in minutes. |
| **FirstInstance** | The date and time of the first instance of this event. |
| **Html** | The html code generated from the wiki-text entered in the admin interface. This is unlimited in length and is not be returned unless specified in the service call documentation. |
| **ImageUrl** | The fully qualified URL for the image associated with the event, if one has been uploaded. |
| **LastInstance** | The date and time of the last instance of this event. |
| **Name** | The name/title of the Event. This is limited to 100 characters. |
| **OnSaleOnWeb** | Indicates whether the event is on sale on the website values are 'true' or 'false'. |
| **ThumbnailUrl** | The fully qualified URL for the thumbnail image associated with the event, if one has been uploaded. |
| **Times** | An array of EventTime elements. These represent the Event Instances. |
| **WebEventId** | A configurable, unique identifier entered by a system admin user in the Admin Iterface. This allows linking with third part CMS systems. |

## 7.2. EventAttribute

*Elements:*

| | |
|---|---|
| **Name** | The name of the attribute, as entered by the client when creating the event in the Spektrix Admin Interface. This is limited to 50 characters. |
| **Value** | The value of the attribute, as entered by the client when creating the event in the Spektrix Admin Interface. This is limited to 200 characters. |

## 7.3. EventTime

*Elements:*

| | |
|---|---|
| **Attributes** | An array of EventAttribute elements. These are custom attributes that the client has full control over. These are entered by the client when creating an Event in the Spektrix Admin Interface. |
| **Capacity** | The total capacity of the event instance. |
| **EventInstanceId** | The id of the Spektrix event instance referred to by this item. Event instances are an internal system item, this is just a reference identifier to be used when linking to pages hosted on the Spektrix system (via iframes). |
| **SeatsAvailable** | The number of seats available. |
| **SeatsLocked** | The number of seats that are locked. |
| **SeatsReserved** | The number of seats that have been reserved. |
| **SeatsSelected** | The number of seats that are selected by users (i.e. in their basket). |
| **SeatsSold** | The number of seats that have been sold. |
| **Time** | The start date/time of this event instance. |
| **WebInstanceId** | A configurable, unique identifier entered by a system admin user in the Admin Iterface. This allows linking with third part CMS systems. |