Requirements for data source

integration with Reserva

# Event Data

The following describes the fields of event data from a data source required for integration with Reserva.

## Mandatory fields

These fields must be provided for all events

|  |  |
| --- | --- |
| **uid** | Unique identifier for the event  In the case of recurring events, it must uniquely identify the individual occurrence. Can be any string and composed of a mix of other identifiers. |
| **subject** | Display name of the event |
| **start\_time** | UTC\* ISO 8601 formatted start date/time |
| **end\_time** | UTC\* ISO 8601 formatted end date/time |

\* - preferably UTC formatted to avoid time zone confusions. ISO 8601 such as  
**yyyy-MM-ddTHH:mm:ssZ**

## Optional fields

These fields can be provided if available, but are not required.

|  |  |
| --- | --- |
| **location** | Location name of the event |
| **description** | An additional description field for the event. Reserva can be configured to display this together with Subject in some circumstances. |
| **organiser** | Display name of organiser |
| **organiser\_email** | Email address of organiser |
| **all\_day** | Flag indicating an all-day event (optionally can be inferred from start/end times). |
| **private** | Flag indicating if the event is private (subject name will be hidden etc). |
| **confirmed** | Flag indicating if the event is confirmed (for systems that support tentative & confirmed events). This can also be provided in **Meeting Status**. |
| **cancelled** | Flag indicating if the event has been cancelled (some data sources retain events in the output feed but flag as cancelled instead of deleting the event) |
| **meeting\_status** | A field indicating event status, such as scheduled, cancelled or confirmed. Some data sources provide this to allow filtering. By default values of Scheduled, Cancelled or Confirmed will be processed. All other values will be ignored. This can be adjusted on integration or this field can be otherwise ignored. |

## Events XML specification

The following is an example of an XML data source. For XML sources for use with Reserva this format should be adopted.

<?xml version="1.0" encoding="utf-8" ?>

<events xmlns="http://www.reserva.co.uk/2016/XMLSchema"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.reserva.co.uk/2016/XMLSchema events.xsd">

<event>

<uid>123456789ABCDEFG</uid>

<subject>Event Subject</subject>

<start\_time>2016-09-28T13:00:00Z</start\_time>

<end\_time>2016-09-28T14:00:00Z</end\_time>

<location>Room 1</location>

<description>Details about event</description>

<organiser>Organiser Name</organiser>

<organiser\_email>organiser@somewhere</organiser\_email>

<all\_day>false</all\_day>

<private>false</private>

<confirmed>true</confirmed>

<cancelled>false</cancelled>

<meeting\_status>scheduled</meeting\_status>

</event>

<event>

<uid>ABCDEFG123456789</uid>

<subject>Another Event Subject</subject>

<start\_time>2016-09-28T14:00:00Z</start\_time>

<end\_time>2016-09-28T15:00:00Z</end\_time>

<location>Room 1</location>

<description>Details about event</description>

<organiser>Organiser Name</organiser>

<organiser\_email>organiser@somewhere</organiser\_email>

<all\_day>false</all\_day>

<private>false</private>

<confirmed>true</confirmed>

<cancelled>false</cancelled>

<meeting\_status>scheduled</meeting\_status>

</event>

</events>

## Events XML Schema (XSD)

The following is an XML Schema which defines the requirements for the XML format and may be useful for validation.

<?xml version="1.0" encoding="utf-8"?>

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema"

targetNamespace="http://www.reserva.co.uk/2016/XMLSchema" xmlns="http://www.reserva.co.uk/2016/XMLSchema events.xsd">

<xs:element name="events">

<xs:complexType>

<xs:sequence>

<xs:element name="event" minOccurs="0" maxOccurs="unbounded">

<xs:complexType>

<xs:all>

<xs:element name="uid" type="xs:string" minOccurs="1" maxOccurs="1">

<xs:annotation>

<xs:documentation>Unique identifer for the event. In the case of recurring events, it must uniquely identify the individual occurrence.</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="subject" type="xs:string" minOccurs="1" maxOccurs="1">

<xs:annotation>

<xs:documentation>Display name of the event</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="start\_time" minOccurs="1" maxOccurs="1" >

<xs:annotation>

<xs:documentation>Start date/time. ISO 8601 formatted, preferably UTC</xs:documentation>

</xs:annotation>

<xs:simpleType>

<xs:restriction base="xs:dateTime">

<xs:pattern value="\d{4}-\d\d-\d\dT\d\d:\d\d:\d\d([zZ]|[+\-]\d\d:\d\d)"/>

</xs:restriction>

</xs:simpleType>

</xs:element>

<xs:element name="end\_time" minOccurs="1" maxOccurs="1" >

<xs:annotation>

<xs:documentation>End date/time. ISO 8601 formatted, preferably UTC</xs:documentation>

</xs:annotation>

<xs:simpleType>

<xs:restriction base="xs:dateTime">

<xs:pattern value="\d{4}-\d\d-\d\dT\d\d:\d\d:\d\d([zZ]|[+\-]\d\d:\d\d)"/>

</xs:restriction>

</xs:simpleType>

</xs:element>

<xs:element name="location" type="xs:string" minOccurs="0" maxOccurs="1">

<xs:annotation>

<xs:documentation>Location name of the event (optional)</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="description" type="xs:string" minOccurs="0" maxOccurs="1">

<xs:annotation>

<xs:documentation>Description for the event (optional)</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="organiser" type="xs:string" minOccurs="0" maxOccurs="1">

<xs:annotation>

<xs:documentation>Display name of the organiser (optional)</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="organiser\_email" type="xs:string" minOccurs="0" maxOccurs="1">

<xs:annotation>

<xs:documentation>Email address of the organiser (optional)</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="all\_day" type="xs:boolean" minOccurs="0" maxOccurs="1">

<xs:annotation>

<xs:documentation>Flag indicating an all-day event (optional)</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="private" type="xs:boolean" minOccurs="0" maxOccurs="1">

<xs:annotation>

<xs:documentation>Flag indicating the event is private (optional)</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="confirmed" type="xs:boolean" minOccurs="0" maxOccurs="1">

<xs:annotation>

<xs:documentation>Flag indicating the event has been confirmed (optional)</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="cancelled" type="xs:boolean" minOccurs="0" maxOccurs="1">

<xs:annotation>

<xs:documentation>Flag indicating the event has been cancelled</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="meeting\_status" type="xs:string" minOccurs="0" maxOccurs="1">

<xs:annotation>

<xs:documentation>A field indicating the status of the meeting, used for filtering. (optional)

Values of Scheduled, Cancelled or Confirmed will be processed. All other values will be ignored</xs:documentation>

</xs:annotation>

</xs:element>

</xs:all>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

# Room Resource List

To provide **Room Finder** functionality a list of room resources should also be provided.

## Mandatory fields

These fields must be provided for all room resources

|  |  |
| --- | --- |
| **uid** | Unique identifier for the room resources. This should be an identifier that is used to filter queries for room events (e.g. through an XML feed parameter). |
| **name** | Name of the room resource, typically used for display |
| **description** | Description if different (e.g. to provide more detailed description in Room Finder) |

## Room Resource List XML Specification

The following is an example of an XML room resource list.

<?xml version="1.0" encoding="utf-8" ?>

<rooms xmlns="http://www.reserva.co.uk/2016/XMLSchema"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.reserva.co.uk/2018/XMLSchema rooms.xsd">

<room>

<uid>ROOM\_ID\_1</uid>

<name>Room 1</name>

<description>Room 1 Description</description>

</room>

<room>

<uid>ROOM\_ID\_2</uid>

<name>Room 2</name>

<description>Room 2 Description</description>

</room>

<room>

<uid>ROOM\_ID\_3</uid>

<name>Room 3</name>

<description>Room 3 Description</description>

</room>

<room>

<uid>ROOM\_ID\_4</uid>

<name>Room 4</name>

<description>Room 4 Description</description>

</room>

</rooms>

## Events XML Schema (XSD)

The following is an XML Schema which defines the requirements for the XML format and may be useful for validation.

<?xml version="1.0" encoding="utf-8"?>

<xs:schema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" attributeFormDefault="unqualified" elementFormDefault="qualified"

targetNamespace="http://www.reserva.co.uk/2018/XMLSchema" xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="rooms">

<xs:complexType>

<xs:sequence>

<xs:element minOccurs="0" maxOccurs="unbounded" name="room">

<xs:complexType>

<xs:sequence>

<xs:element name="uid" type="xs:string" minOccurs="1" maxOccurs="1" >

<xs:annotation>

<xs:documentation>Unique identifer for the location</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="name" type="xs:string" minOccurs="1" maxOccurs="1" >

<xs:annotation>

<xs:documentation>Name of the location</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="description" type="xs:string" minOccurs="1" maxOccurs="1" >

<xs:annotation>

<xs:documentation>Description for the location</xs:documentation>

</xs:annotation>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

# XML Feed

If using an XML feed via HTTP or HTTPS, the above XML should be returned from two URLs, one returning a Room Resource List, and the other returning events for a specific room resource. Ideally the event feed should also allow filtering by date.

Room Resource URL spec:

**http|https://[server]/[rooms]**

Events URL spec:

**http|https://[server]/[events]?room=[room\_uid]&start\_date=yyyy-MM-dd&end\_date=yyyy-MM-dd**

**Note,** the URL and parameter names can be customised to fit the data source feed, but it’s expected basic parameters to request events for a room by ID and filter on date should be available, and that the ID should match the IDs returned from the Room Resource List.

For example, a typical URL for Room Resource List would be:

<http://server/rooms.asp>

and for Events:

<http://server/events.asp?room=ROOM_1_ID&start_date=2018-02-01&end_date=2018_02_30>

Preferably the date formats should be ISO 8601 format yyyy-MM-dd to avoid confusion with locales.