

SINFONET: Event Interface Documentation:

EVENTS.....	2
Setting an Event.....	2
Retrieving Events	2
Retrieving Event RID's.....	2
Retrieving Event detailed data	3
Deleting Events	3
Retrieving Topic List.....	3
Retrieving Nation List.....	4
ORGANIZER.....	4
Setting an Organizer	4
Retrieving Organizers.....	4
Deleting Organizers	5
XML FILE STRUCTURE	5
Structure of Event.Xml	5
Event:	6
EventSummery :	7
Description :	7
EventTimeList :	7
EventDay :	8
Picture:	8
Topics:	9
Structure of DeleteEvent.XML.....	9
Event :	9
Structure of Response.XML	9
EventSaveProtocol :	10
Structure of requestEventRID.XML:	10
SearchCriteria :	10
Structure of responseEventRID.XML:	11
Event :	11
Structure of requestEventDetailed.XML.....	11
Event :	12
Structure of Organizer.XML:.....	12
Organizer :	12
OrganizerData:.....	12
Structure of requestOrganizer.XML:	13
SearchOrganizer:.....	13
Structure of deleteOrganizer.XML:	13
Organizer:	13

Structure of XmlGetTopics.XML:	13
Topics:	13
Structure of XmlGetTopicsResponse.XML:	14
Topic:.....	14
TopicData:	14

EVENTS

URL = Your personal Path

Setting/Importing an Event

To set an Event, an Xml file of type [Event.XML](#) has to be send to the following address:

[URL]/EventAPI_Event.Set

The only parameter of this function is **XmlData** that gets the [Event.XML](#) file. The response of the EventAPI_Event.Set will be of type [Response.XML](#)

Note:

If an event, after a successful import has been changed by the SMG online editorial staff, the event-description and the event-pictures can not be changed by an external institution anymore.

Retrieving Events

Retrieving Event RID's

To retrieve Event RIDs, an Xml file of type [requestEventRID.XML](#) has to be send to the following address:

[URL]/EventAPI_EventSearchRID.Get

The only parameter of this function is XmlData that gets the

[requestEventRID.XML](#) file. The response of the EventAPI_EventSearchRID.Get will be of type [responseEventRID.XML](#)

Retrieving Event detailed data

To retrieve detailed data of an event, an Xml file of type [requestEventDetailed.XML](#) has to be send to the following address:

[URL]/EventAPI_EventSearchDetail.Get

The parameters passed to this function are the **XmlData** that gets the requestEventDetailed.XML file and a optional **Language** parameter that specifies the language in which the event should be retrieved. The response of the EventAPI_EventSearchDetail.Get will be of type [EVENT.XML](#)

Deleting Events

To delete an Event, an Xml file of type [DeleteEvent.XML](#) has to be send to the following address:

[URL]/EventAPI_Event.Set

The only parameter of this function is **XmlData** that gets a [DeleteEvent.XML](#). The response of the EventAPI_Event.Set will be of type [Response.XML](#)

Retrieving Topic List

[URL]/EventTopicList.Get

according to the passed [XmlGetTopics.xml](#) file it retrieves all corresponding Topics. An optional **Language** parameter can be passed to specify in which language the topics should be returned. The response of the EventTopicList.Get will be of type [XmlGetTopicsResponse.XML](#)

Retrieving Nation List

[URL]/NationList.Get

retrieves a complete list of accepted Nations for the Nation attribute in the Event node of the [Event.xml](#) file. An optional **Language** parameter can be added to specify in which language the Nations should be returned.

ORGANIZER

Setting an Organizer

To set an Organizer, an Xml file of type [Organizer.XML](#) has to be sent to the following address:

[URL]/EventAPI_Organizer.Set

The only parameter of this function is **XmlData** that gets the [Organizer.XML](#) file. The response of the EventAPI_Organizer.Set will be of type [Response.XML](#)

Retrieving Organizers

To retrieve the organizer, an Xml file of type [requestOrganizer.XML](#) has to be sent to the following address:

[URL]/EventAPI_Organizer.Get

The parameters passed to this function are the **XmlData** that gets the [requestOrganizer.XML](#) file and an optional **Language** parameter that specifies the language in which the organizer should be retrieved. The response of the EventAPI_Organizer.Get will be of type [Organizer.XML](#), The only difference will be that there will be one root tag including all organizers.

Deleting Organizers

To retrieve the organizer, an Xml file of type [deleteOrganizer.XML](#) has to be send to the following address:

[URL]/EventAPI_Organizer.Set

The parameters passed to this function are the **XmlData** that gets the [deleteOrganizer.XML](#) file. The response of the **EventAPI_Organizer.Set** will be of type [Response.XML](#)

XML FILE STRUCTURE

Structure of Event.Xml

Example:

```
<Event (RID="") Organizer="" Email="" Phone="" WEBUrl="" AreaID="" AreaType="" ZIP=""
HourFrom="" HourTo="" HourStart="" Price="" EventRank="" ActiveType="" Nation="" FirstName=""
LastName="">
  <EventData>
    <EventSummery Title="" Street="" City="" LngID="" Location=""/>
    <Description Type="" LngID="" "><![CDATA[]]></Description>
  </EventData>
  <EventTimeList Type="" RecurrenceType="" RecurrenceValue="" Days="" Day="" Month=""
DayRecurrenceType="" From="" To="" />
  <EventDay Date=""/>
  <EventDay Date=""/>
  <Picture Path="" Type=""/>
  <Topics>
    <Topic ID="EV_Kultur">
  </Topics>
  (<Organizer TaxNumber="22" Phone="" Fax="" AreaID="ORT26" AreaType="1"
ZIP="43244" >
    <OrganizerData Name="" Street="3" City="3" LngID="DE"/>
  </Organizer>)*
```

</Event>

*** **The Organizer is passed only when retrieving an event**

- The Event Node takes all not language specific data about the event

```
<Event (RID="") Organizer="" Email="" Phone="" WEBUrl="" AreaID="" AreaType="" ZIP=""
HourFrom="" HourTo="" HourStart="" Price="" EventRank="" ActiveType="" Nation=""
FirstName="" LastName="">
```

Event:

Node Attribute name	description	Length
(RID)	RID of the event only when Event.xml is a response to requestEventDetailed.XML and used for Deleting / Modifying the Event	String(32)
Organizer	RID of the organizer	String(32)
Email	possible Email of the Event	String(80)
Phone	Information telephone of the Event	String(30)
WEbUrl	possible Web address of the Event	String(255)
AreaID	AreaID of the town in which the event takes place, can be in two formats AreaID (tblcdbG0 G0ID) or as ASTATCODE	String(20)
AreaType	Decides what type the AreaID is 0 >undefined 1 >G0ID 2 >AstatCode	Smallint
ZIP	Postal Code	String(10)
HourFrom	Start time of the event	(FORMAT HH:MM:SS)
HourTo	End time of the event	(FORMAT HH:MM:SS)
HourStart	Begin time of the event	(FORMAT HH:MM:SS)
Price	Price of the event	(Format Euro.Cents)
EventRank	Importance of the event: 0->undefined 1-> Holiday area 2-> Holiday resort 3-> Holiday Syndicate 4-> South Tyrol	Smallint
ActiveType	Determine the state of the event: 0->undefined 1->active 2->canceled 3->out of time	Smallint
Nation	The nation in which the nation is located (Gets a NatID from EventAPI_NationList.Get)	String(3)
FirstName	First name of the person responsible for this event	String(50)
LastName	Last name of the person responsible for this event	String(50)

- The Sub Nodes of the EventData node contains all language specific data:

```
<EventData>
  <EventSummery Title=" " Street="" City=" " Location="" LngID=" " />
  <Description Type="" LngID=" " "><![CDATA[***]]></Description>
</EventData>
```

EventSummery :

Node Attribute name	description	Length
Title	Title/Name of the Event	String(80)
Street	Street in which the event is located	String(80)
City	Town/City	String(50)
Location	General description of the Location	String(80)
LngID	Language	String(3)

Description :

Node Attribute name	description	Length
Type	Type of the description: 0>undefined, 1>Main description, 2>Rout description	Smallint
LngID	Language	String(3)
CDATA	the description	String(600)

There can be 1 to m (for any language) EventSummery/Descriptions

- The EventTimeList node takes all time specific data of the event

```
<EventTimeList Type="" RecurrenceType="" RecurrenceValue="" Days="" Day="" Month=""  
DayRecurrenceType="" From="" To="" />
```

EventTimeList :

Node Attribute name	description	Length
Type	Type of the EventTimeList 0>undefined 1>White list (days on which the event takes place) 2>Black list (days on which the event will not take place)	smallint
RecurrenceType	Type of recurrence: 0>undefined 1>one time (no RecurrenceValue) 2>weekly (Recurrence Type unit -> weeks) 3>monthly (Recurrence Type unit -> months) 4>yearly (Recurrence Type unit -> years)	smallint
RecurrenceValue	Value of the recurrence (is bounded to the Recurrence Type): 2->every second [Recurrence Type unit] 3-> every third [Recurrence Type unit] 4->...	smallint
Days	days that are desired as output dates format	Char(7)

	'0000000' the first character is Monday the last Sunday: 0->no 1->yes	
Day	Specific day value (valid values between 1 and 31)	smallint
Month	Specific month value (valid values between 1 and 12)	smallint
DayRecurrenceType	Day recurrence type (used only for monthly/yearly Recurrence): 0>undefined 1>first day 3>second day 4>fourth day 5>last day	smallint
From	Start date of the event	(AMERICAN DATE FORMAT MM/DD/YYYY)
To	End date of the event	(AMERICAN DATE FORMAT MM/DD/YYYY)

- The EventDay node gives the possibility to enter a date without using the EventTimeList

<EventDay Date="" />

EventDay :

Node Attribute name	description	Length
Date	Date of the Event	(AMERICAN DATE FORMAT MM/DD/YYYY)

- The Picture node defines the pictures

<Picture Path="" Type="" />

Picture:

Node Attribute name	description	Length
Path	Full path to the picture	String(255)
Type	Type of the picture: 0> undefined 1> Small size (98X75 pixel) 2> Big size (170X128 pixel)	smallint

- Topic of the Event(maximum 3)


```
<Topics>
  <Topic ID="***">
  <Topic ID="***">
  <Topic ID="***">
</Topics>
```

Topics:

Node Attribute name	description	Length
ID	ID's of event: EV_Kultur, EV_Gastronomie, EV_KinderFamilie, EV_Tradizion, EV_Messe, EV_Musik, EV_Kunst, EV_Volksfest, EV_Unterhaltung, EV_Theater, EV_Wandern, EV_Treffen , EV_Sport, EV_Handwerk, EV_GefuehrteTour	String(30)

Structure of DeleteEvent.XML

Example:

```
<Event RID="" >
```

Event :

Node Attribute name	description	Length
RID	RID of the event that should be deleted	String(32)

Structure of Response.XML

Example:

Successfully stored in DB:

```
<EventSaveProtocol RID="C8DDD2DFA1734E3C8113E663E792493B" tatus="true"/>
```

Not stored in DB:

```
<EventSaveProtocol Status="false" ErrorID="1"/>
```

EventSaveProtocol :

Node Attribute name	description	Length
RID	RID of the stored event (only passed when Status == true)	String(32)
Status	Status of the storing procedure (true/false)	bool
ErrorID	Error in: 1-> Event <Tag> 2-> Picture <Tag> 3-> Description <Tag> 4-> EventSummery <Tag> 5-> EventDay <Tag> 6-> EventTimeList <Tag> 7-> Topics <Tag>	smallint

Structure of requestEventRID.XML:

Example:

```
<Events>  
  <SearchCriteria From="12/12/2004" To="12/31/2004" AreaID="ORT26"  
    (ParentAreaID="ORT11") AreaType="1" Ranking="" >  
    <Topic ID=""/>  
  </SearchCriteria>  
</Events>
```

SearchCriteria :

Node Attribute name	description	Length
From	Start date of the event	(AMERICAN DATE FORMAT MM/DD/YYYY)
To	End date of the event	(AMERICAN DATE FORMAT MM/DD/YYYY)
AreaID	AreaID of the town in which the event takes place, can be in two formats AreaID(tblcdbG0G0ID) or as ASTATCOD	String(20)
ParentAreaID	ParentAreaID is the AreaID of the town/region/... which is hirarchically above the AreaID specified in AreaID. When set, not the Ranking of the AreaID will be applaied, but the Ranking of the ParentAreaID.	String(20)
AreaType	Decides what type the AreaID/ParentAreaID is 0 >undefined 1 >tblcdbG0..G0ID	Smallint

	2 >AstatCode	
Ranking	Importance of the event: 0->undefined 1->Holiday Area 2->Holiday Resort 3->Holiday Syndicate 4->South Tyrol	smallint

Topic:

Node Attribute name	description	Length
ID	ID's of event(Centraldb..tbcdbS9): EV_Kultur, EV_Gastronomie, EV_KinderFamilie, EV_Tradizion, EV_Messe, EV_Musik, EV_Kunst, EV_Volksfest, EV_Unterhaltung, EV_Theater, EV_Wandern, EV_Treffen , EV_Sport, EV_Handwerk, EV_GefuehrteTour	String(30)

Structure of responseEventRID.XML:

Example:

```
<Events>
  <Event RID="1234567890113456789ddd34563892xx"/>
  <Event RID="1234567890113456789ddd34563892x1"/>
  <Event RID="1234567890113456789ddd34563892x2"/>
  <Event RID="1234567890113456789ddd34563892x3"/>
</Events>
```

Event :

Node Attribute name	Description	Length
RID	RID of Event	String(32)

Structure of requestEventDetailed.XML (maximum one Event)

Example:

```
<Events>
  <Event RID="1234567890113456789ddd34563892xx"/>
</Events>
```

Event :

Node Attribute name	Description	Length
RID	RID of Event	String(32)

Structure of Organizer.XML:

Example:

```
<Organizer TaxNumber="22" Phone="" Fax="" AreaID="ORT26" AreaType="1" ZIP="43244" >  
  <OrganizerData Name="" Street="3" City="3" LngID="DE"/>  
</Organizer>
```

- The Organizer node contains all not language specific data for an Object of type Organizer

```
<Organizer (RID="") TaxNumber="22" AreaID="ORT26" AreaType="1" ZIP="43244" >
```

Organizer :

Node Attribute name	Description	Length
(RID)	RID of Organizer used only for deletion and modification	String(32)
TaxNumber	TaxNumber of the Organizer	String(50)
Phone	Phone Number of the Organizer	String(30)
Fax	Fax Number of the Organizer	String(30)
AreaID	AreaID of the town in which the event takes place, can be in two formats AreaID(tblcdbG0G0ID) or as ASTATCOD	String(20)
AreaType	Decides what type the AreaID is 0 >undefined 1 >tblcdbG0..G0ID 2 >AstatCode	smallint
ZIP	Postal code (address part)	String(10)

- The OrganizerData node contains all language specific data for an Object of type Organizer

```
<OrganizerData Name="" Street="3" City="3" LngID="DE"/>
```

OrganizerData:

Node Attribute name	Description	Length
Name	Name of the Organizer	String(100)
Street	Street (address part)	String(80)
City	City/Town (address part)	String(50)

LngID	Language	String(3)
-------	----------	-----------

Structure of requestOrganizer.XML:

```
<SearchOrganizer AreaID="ORT26" AreaType="1" />
```

SearchOrganizer:

Node Attribute name	description	Length
AreaID	AreaID of the town in which the event takes place, can be in two formats AreaID(tblcdbG0G0ID) or as ASTATCOD	String(20)
AreaType	Decides what type the AreaID is 0 >undefined 1 >tblcdbG0..G0ID 2 >AstatCode	smallint

Structure of deleteOrganizer.XML:

```
<Organizer RID="">
```

Organizer:

Node Attribute name	description	Length
RID	RID of organizer	String(32)

Structure of XmlGetTopics.XML:

```
<?xml version="1.0" encoding="ISO-8859-15" ?>  
<Event>  
  <Topics ID="" Group="Events" />  
</Event>
```

Topics:

Node Attribute name	description	Length
ID	ID's of event(Centralsdb..tblcdbS9): EV_Kultur, EV_Gastronomie, EV_KinderFamilie, EV_Tradizion, EV_Messe, EV_Musik, EV_Kunst, EV_Volksfest, EV_Unterhaltung, EV_Theater, EV_Wandern, EV_Treffen ,	String(30)

	EV_Sport, EV_Handwerk, EV_GefuehrteTour	
Group	Topic group: ->Events	String(20)

Structure of XmlGetTopicsResponse.XML:

```
<EventTopics>
<Event>
  <Topic ID="EV_Gastronomie" Group="Events" >
    <TopicDate Description="Gastronomie" LngID="DE" />
    <TopicDate Description="Gastronomy" LngID="EN" />
    <TopicDate Description="Gastronomy" LngID="FL" />
    <TopicDate Description="Eno-gastronomia" LngID="IT" />
  </Topic>
</Event>
</EventTopics>
```

Topic:

Node	Attribute name	description	Length
ID		ID's of event(Centraldb..tbcdbS9): EV_Kultur, EV_Gastronomie, EV_KinderFamilie, EV_Tradizion, EV_Messe, EV_Musik, EV_Kunst, EV_Volksfest, EV_Unterhaltung, EV_Theater, EV_Wandern, EV_Treffen , EV_Sport, EV_Handwerk, EV_GefuehrteTour, EV_Sport, EV_Handwerk, EV_GefuehrteTour	String(30)
Group		Topic Group ->Events	String(20)

TopicData:

Node	Attribute name	description	Length
Description		Description of the Topic	String(50)
LngID		In the corisponding Language	String(3)