Guest Vacation & Itinerary API

This documentation covers all apis and their schema model for the Itinerary Vacation Engine. The terminology used here is specific to the IVE domain. It may require domain mapper for other business terminology such as E2K or commerce.

Version 1.0.0

Paths

`/my_itinerary`

GET /my_itinerary

My Itineraries

Summary

This end point returns all personal itineraries of the guest.

Description

Every personal itineries can contain multiple `ItineraryEventGroup`. The ship schedule event group contains the seaport and their arrival and departure time. Guest can manage personal event group for their own activities. The product booking event group and the cruise compass event group are managed by the commerce engine. `eventGroupType` is used to identify the type of the itinerary.

Parameters

Name	Located in	Description	Required	Schema
sailingID	query	Unique identifier of the sailing (Ship + SailDate (YYYYmmDD). E.g OA20220328	Yes	⇄ string
cruiseReservationID	query	Unique cruise reservation id	Yes	⇄ number (int)
guestID	query	guest ID within the cruise reservation	Yes	⇄ number (int)

Responses

Code	Description	Schema
200	MyItinerary object, which contains list of calendars for the given guest	⇄ ▾ **MyItinerary {** *Personal Itinerary of the given Cruise Reservation and guest. It lists all events scheduled for guests. Events could be based on the activities purchased/scheduled by the shopping transactions or personal appointments created by the guest themselves.* cruiseReservationID: ▸ number guestID: ▸ number sailingID: ▸ string firstName: ▸ string lastName: ▸ string itineraryEventGroups: ▸ **[]** **}**

Code	Description	Schema
default	Unexpected error	⇄ ▾ [▸ Error { }]

Try this operation

Models

LocationTypeENUM

⇄ ▾ **LocationTypeENUM** string
Type of the location in which a specific activity or product is offered
Enum:
▸ Array[4]

MyItinerary

⇄ ▾ **MyItinerary {**
Personal Itinerary of the given Cruise Reservation and guest. It lists all events scheduled for guests. Events could be based on the activities purchased/scheduled by the shopping transactions or personal appointments created by the guest themselves.

```
cruiseReservationID:   ▸ number
guestID:               ▸ number
sailingID:             ▸ string
firstName:             ▸ string
lastName:              ▸ string
itineraryEventGroups:  ▸[]
}
```

ItineraryEventGroup

▾ **ItineraryEventGroup {**

Grouping of the same type of itinerary event

eventGroupType: ▸ string

⇄

eventGroupDescription: ▸ string

itineraryEvents: ▸**[]**

}

ItineraryEvent

▾ **ItineraryEvent {**

Personal Itinerary of the given Cruise Reservation and guest. It lists all events scheduled for guests. Events could be based on the activities purchased/scheduled by the shopping transactions or personal appointments created by the guest themselves.

eventID: ▸ string

eventType: ▸ string

eventName: ▸ string

⇄

eventLocale: ▸ string

eventLocation: ▸**EventLocation { }**

eventStartTime: ▸**EventTime { }**

eventEndTime: ▸**EventTime { }**

sourceDetail: ▸**EventSourceDetail { }**

attendees: ▸**[]**

}

EventSourceDetail

▾ **EventSourceDetail {**

Source of the specific event. It covers the details like which system generated the event and its meta details

sourceSystemApplicationID: ▸ string
sourceSystemReservationID: ▸ string
sourceSystemOrderID: ▸ string
sourceSystemExternalOrderID: ▸ string
sourceSystemProductID: ▸ string
sourceSystemOfferingID: ▸ string
sourceSystemProductName: ▸ string

}

EventAttendee

▾ **EventAttendee {**

Details of the attendoes of a specific event

cruiseReservationID: ▸ number
guestID: ▸ number
sailingID: ▸ string
firstName: ▸ string
lastName: ▸ string

}

EventTime

▾ **EventTime {**

Start or End time of the actual Event

eventDate: ▸ string
eventTime: ▸ string
eventTimeZone: ▸ string

}

EventLocation

▾ **EventLocation {**

location in which a specific activity or product is offered / happenning

locationCode: ▸ string

⇄ locationType: ▸**LocationTypeENUM** string

locationName: ▸ string

}

Error

▾ **Error {**

code: integer

⇄ message: string

fields: string

}