API Documentation for as3googlecalendarlib

http://code.google.com/p/as3googlecalendarlib/

Table of Contents

GoogleCalendarAuthenticator

Use this class to authenticate with Google Calendar API.

authenticateUser(userName:String,userPassword:String, applicationId:String = "sample-application")

This function authenticates a user. AUTHENTICATION_RESPONSE event will be dispatched when the server responds. In the AUTHENTICATION_RESPONSE event, you can access variable called *authenticatedUser* which is of the type GoogleCalendarUserVO. Use the *authenticated* variable in GoogleCalendarUserVO to check if the user's authentication is successful.

Events to listen for:

GoogleCalendarAuthenticatorEvent.AUTHENTICATION_RESPONSE

GoogleCalendarAuthenticatorEvent.AUTHENTICATION_FAULT

Arugments:

userName: your email id

userPassword: your password

GoogleCalendarService

Use this class for operations on Calendar. This will allow you to add, edit and delete your calendars.

getAllCalendars(user:GoogleCalendarUserVO)

This function retrieves all calendars in your Google account. GET_ALL_CALENDARS_RESPONSE events will be dispatched when the server responds with calendar details. In the GET_ALL_CALENDARS_RESPONSE event object, access *allCalendars* variable which is a ArrayCollection. *allCalendars* will contain objects of type GoogleCalendarVO.

Events to listen for:

GoogleCalendarServiceEvent.GET_ALL_CALENDARS_RESPONSE

GoogleCalendarServiceEvent.GET_ALL_CALENDARS_FAULT

Arguments:

user: this should be the object which you got from the GoogleCalendarAuthenticator

getOwnedCalendars(user:GoogleCalendarUserVO)

This function retrieves calendars the logged in user owns. GET_OWNED_CALENDARS_RESPONSE event is dispatched when the server responds. GET_OWNED_CALENDARS_RESPONSE events object will contain variable called *ownedCalendars* which is a ArrayCollection. *ownedCalendars* will contain objects of type GoogleCalendarVO.

Events to listen for:

GoogleCalendarServiceEvent.GET_OWNED_CALENDARS_RESPONSE

GoogleCalendarServiceEvent.GET_OWNED_CALENDARS_FAULT

Arguments:

user: this is the object which you got from GoogleCalendarAuthenticator

addCalendar(calendar:GoogleCalendarVO,user:GoogleCalendarUserVO)

This function adds the calendar passed as argument to your Google account. Create a calendar object and pass it to this function to add. ADD_CALENDAR_RESPONSE event will be dispatched if the calendar is added. You can access the added calendar object returned from Google server using *returnedCalendar* variable in ADD_CALENDAR_RESPONSE event object.

Events to listen for:

GoogleCalendarServiceEvent.ADD_CALENDAR_RESPONSE

GoogleCalendarServiceEvent.ADD_CALENDAR_FAULT

Arguments:

calendar: calendar to add

user: authenticated use object

updateCalendar(calendar:GoogleCalendarVO,user:GoogleCalendarUserVO)

This function updates the calendar object passed on the server. UPDATE_CALENDAR_RESPONSE event will be dispatched when the calendar is updated. You can access the updated calendar object from the server using the *returnedCalendar* in UPDATE_CALENDAR_RESPONSE event object.

Events to listen for:

GoogleCalendarServiceEvent.UPDATE_CALENDAR_RESPONSE

GoogleCalendarServiceEvent.UPDATE_CALENDAR_FAULT

Arguments:

calendar: updated calendar object

user: authenticated user object

deleteCalendar(calendar:GoogleCalendarVO, user:GoogleCalendarUserVO)

This function deletes the calendar passed as argument from your Google calendars list on the server. DELETE_CALENDAR_RESPONSE event is dispatched if the calendar is deleted from the server.

Note: This service exposed by Google is not working at times.

Events to listen for:

GoogleCalendarServiceEvent.DELETE_CALENDAR_RESPONSE

GoogleCalendarServiceEvent.DELETE_CALENDAR_FAULT

Arguments:

calendar: calendar object to delete

user: authenticated user object

GoogleCalendarEventsService

Use this class for operations on events in a calendar. This class will allow you to retrieve events, add events, update event and delete event.

getEventsForDateRange(calendar:GoogleCalendarVO, user:GoogleCalendarUserVO, startDate:Date, endDate:Date)

This function will retrieve events in a calendar for the date range specified. GET_EVENTS_FOR_DATE_RANGE_RESPONSE event will dispatched when the server responds. Access *calendarEvents* object in GET_EVENTS_FOR_DATE_RANGE_RESPONSE event object for events returned. *calendarEvents* is ArrayCollection which contains object of type GoogleCalendarEventVO.

Events to listen for:

GoogleCalendarEventsServiceEvent.GET_EVENTS_FOR_DATE_RANGE_RESPONSE

GoogleCalendarEventsServiceEvent.GET_EVENTS_FOR_DATE_RANGE_FAULT

Arguments:

calendar: calendar from which the events should be retrieved

User: authenticated user object

startDate: date range start

endDate: date range end

addEventToCalendar(calendarEvent:GoogleCalendarEventVO, calendar:GoogleCalendarVO,user:GoogleCalendarUserVO)

This function will add calendar event passed as argument to the calendar which is also passed argument. ADD_EVENT_RESPONSE event is dispatched when the server responds. *returnedCalendarEvent* variable in the ADD_EVENT_RESPONSE event object contains the event added on the server.

Events to listen for:

GoogleCalendarEventsServiceEvent.ADD_EVENT_RESPONSE

GoogleCalendarEventsServiceEvent.ADD_EVENT_FAULT

Arguments:

calendarEvent: calendar event to add

calendar: calendar to which the event should be added

user: authenticated user object

updateEventInCalendar(calendarEvent:GoogleCalendarEventVO, user:GoogleCalendarUserVO, forceUpdate:Boolean = false)

This function updates the calendar event passed as argument on the server. UPDATE_EVENT_RESPONSE will be dispatched when the server responds. *returnedCalendarEvent* variable in UPDATE_EVENT_RESPONSE event object will contain the updated calendar event returned from server.

Events to listen for:

GoogleCalendarEventsServiceEvent.UPDATE_EVENT_RESPONSE

GoogleCalendarEventsServiceEvent.UPDATE_EVENT_FAULT

Arguments:

calendarEvent: calendar event to be updated

user: authenticated user object

forceUpdate: if this is true event is forced to update with your version. If this is false and the event has been updated by other user, you will receive an fault from the server.

deleteEventInCalendar(calendarEvent:GoogleCalendarEventVO, user:GoogleCalendarUserVO, forceDelete:Boolean = false)

This function deletes calendar event passed as argument. DELETE_EVENT_RESPONSE is dispatched when the server responds after successfully deleting.

Events to listen for:

GoogleCalendarEventsServiceEvent.DELETE_EVENT_RESPONSE

GoogleCalendarEventsServiceEvent.DELETE_EVENT_FAULT

Arguments:

calendarEvent: calendar event to delete

user: authenticated user object

forceDelete: if this is false and the calendar event object version you have is not the latest you will receive a fault. If this is true, event is forced to delete.

GoogleCalendarVO

This class represents a Calendar.

setTitle(text:String,type:String=GoogleCalendarEventUtil.TEXT_CONTENT_TYPE)

This function will set the title for a calendar. Title for a calendar is a object by itself, this is added to set a title easily.

setSummary(text:String, type:String=GoogleCalendarEventUtil.TEXT_CONTENT_TYPE)

This function will set summary for a calendar.

addWhere(valueString:String, rel:String="",label:String="Calendar location")

This function will add a location to a calendar. Location itself is a complex object for a Google Calendar, this function will let you add a location easily.

Returns: WhereVO

GoogleCalendarEventVO

This class represents a Google Calendar Event. Schema details: http://code.google.com/apis/gdata/elements.html#gdEventKind

setTitle(text:String, type:String=GoogleCalendarEventUtil.TEXT_CONTENT_TYPE)

This function sets title for the calendar event.

Arguments:

text: title of calendar event

type: text type.

setContent(text:String,type:String=GoogleCalendarEventUtil.TEXT_CONTENT_TYPE)

This function sets content for calendar event.

addWhere(valueString:String,rel:String=null,label:String="Event location")

This function adds a location for a calendar event.

Arguments:

valueString: A simple string value that can be used as a representation of this location

rel: Specifies the relationship between the containing entity and the contained location.

label: Specifies a user-readable label to distinguish this location from other locations.

Returns: WhereVO

addWhen(startDateTime:Date,endDateTime:Date,isFullDay:Boolean)

This function is used to set calendar event period.

Arguments:

startDateTime: start date and time of the calendar event

endDateTime: end date and time of the calendar event

isFullDay: indicates if this event is full day

Returns: WhenVO. Use this to add Reminders to the calendar event.

isFullDay()

This function returns true if a calendar event is full day.

Returns: Boolean

WhenVO

addReminder(days:int,hours:int,minutes:int,method:String = "alert")

This function adds a reminder to a calendar event. Time for a reminder is with respect to calendar event time.

Arguments:

days: number of days before calendar event

hours: number of hours before calendar event

minutes: number of minutes before calendar event

method: alert method. Supported are SMS, Popup and email. Refer ReminderVO for constants.

Returns: ReminderVO