Calendar data

This document has the data specification and other developer information for calendar data in the POSEIDON framework. The architecture and conceptual model is described in D5.1 Development framework, chapter 5.3, as part of the technology infrastructure description. A summary of the architecture is given in the next section.

Architecture

Figure 1 illustrates the architecture. It shows that calendar events with instructions are defined in the Google Calendar system, while media referenced from event definitions are stored on the POSEIDON file server. An application, for instance a web application, gets events through the Google Calendar HTTP API[1]. It will then get media files referenced in the events from the POSEIDON file server, documented in chapter 3 of this document. When creating a new event, an application will first upload any media files to the file server, enter the returned resource IDs into the event definition, and post that to Google Calendar.

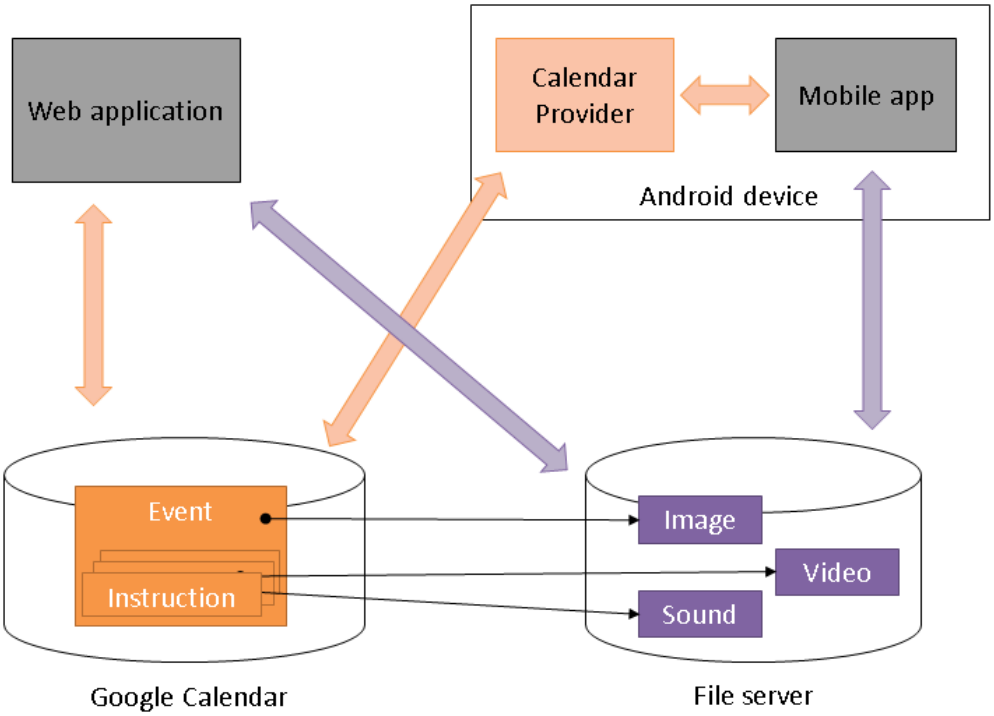

Figure 1: POSEIDON infrastructure for calendar events with instructional media.

On an Android device, applications can use the Calendar Data Provider API[2] of the Android system to interact with the calendar storage. This Android middleware provides an API which is not tied to a specific calendar service, and its implementation handles synchronization of calendar data between cloud stores and the device, simplifying the job of the application programmer. Media files are accessed through the file server API as for other applications.

[1] https://developers.google.com/google-apps/calendar/
[2] http://developer.android.com/guide/topics/providers/calendar-provider.html

Here we describe the relevant parts of the Google Calendar and Android Calendar Provider APIs, and the POSEIDON-specific data added. See the referenced online API documentation for specifications and examples of the APIs.

Data model

A consequence of the POSEIDON architecture is that the framework supports two different programming interfaces for accessing calendar data. The general principles of the underlying data models of the Google Calendar and Android Calendar Provider are roughly the same, and also similar to that of iCalendar[3]. The main entity is the event. The calendar data models are more complex than it may seem at first glance, as an event can be recurring, and there can be exceptions to the recurrence. POSEIDON uses the basic features of events with reminders. This document goes through the relevant entities, as they appear in the data of the two APIs, and how we use them.

Through the Google Calendar API, data is exchanged in a JSON structure, for instance with an event element containing reminder elements. The Android Data Provider API is basically a database interface, making queries and updates to tables. Figure 2 shows the database schema exposed through this API. It serves as an illustration for the following presentation of data objects.

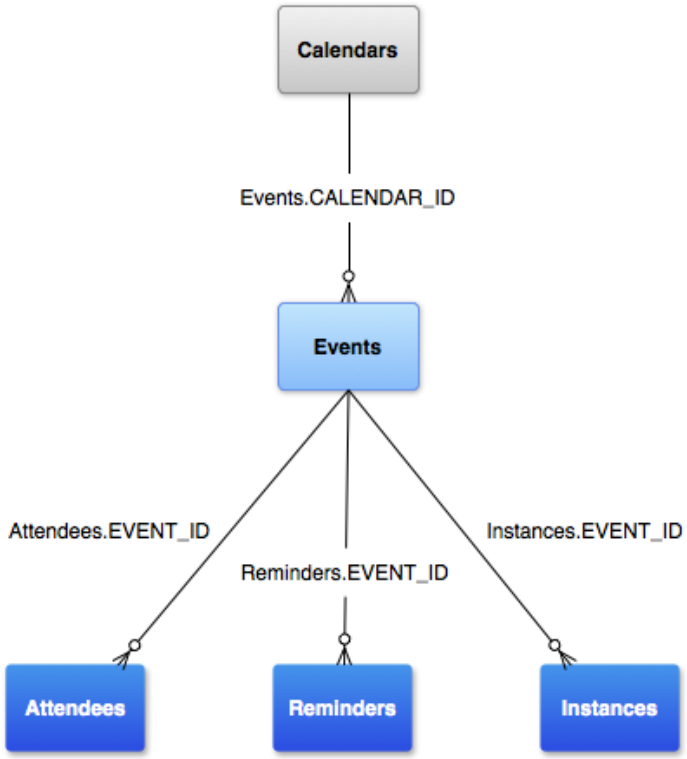

Figure 2: Android Calendar Provider data model.

Data objects

These are the data objects available through the APIs. We list them with JSON property names for the Google Calendar API and Java constants for the Android calendar provider API.

[3] iCalendar is the closest thing to a standard for exchange of calendar data, see http://en.wikipedia.org/wiki/ICalendar

Calendar

A calendar in this data model is an entity which events belong to – all events belongs to a calendar. A Google account can have any number of calendars. It can also be linked with calendars of other Google accounts. In the Android Calendar Provider API, a device may have multiple accounts with calendar data, and each account may have multiple calendars. Accounts may be of different types, one of which is Google. So the situation can get quite complex. For the prototype system, each user is restricted to a Google account with a single calendar. The Android device is set up with this Google account, and no other calendar accounts. This is to ensure there is no ambiguity as to where the data is stored, and that the same data is available on the web and on the phone.

Event

The event is the key entity in both the Google and Android calendar data models. The complicating factor is that an event may either be for a single occurrence on a specific date, or it may be recurring, which means that it represents a potentially open-ended amount of occurrences based on some pattern. Being able to represent recurring events is very useful, as it would be very tedious to have to enter the same event every week manually. Therefore we support recurring events in POSEIDON.

An event may also be an "all day" event. This means it doesn't have specific start and end times, but rather cover one or more whole days. In order to keep the primary user interfaces simple, listing events sorted on time, we do not support "all day" events in POSEIDON. For the same reason, we don't want events spanning multiple dates.

The Android calendar API has an instance entity in addition to event. We can query for instances in a given timeframe, and this will retrieve one instance for each occurrence of an event in this timeframe. So the API handles creating instances with specific start and end times for the repeating events. However, it is not possible to insert, update or delete instances, as they are derived from events.

The following table lists the relevant fields for event data, with corresponding Google Calendar JSON names and constants of the Android calendar API. For the Android API, we read data as instances, but insert, update and delete as events.

Google Calendar API	Android Calendar API	Description
	Events.CALENDAR_ID Instances.CALENDAR_ID	Identifier for the calendar the event belongs to.
id	Event._ID Instances.EVENT_ID	For the Android API, a database ID is used to refer to a specific event (needed to update or delete an event).
summary	Events.TITLE Instances.TITLE	Name of the event.
start.dateTime	Events.DTSTART Instances.BEGIN	Datetime for the start of an event/instance.
end.dateTime	Events.DTEND Instances.END	Datetime for the end of an event/instance. A recurring event does not specify this in the Android API.
start.date **end.date**	Events.ALL_DAY Instances.ALL_DAY	An "all day" event specifies start/end as date without time in the Google data. In the Android model there is an allday flag. We do not currently support this in the POSEIDON system.

	Events.DURATION	A recurring event specifies duration for each instance, rather than an end time, in the Android API.
description	Events.DESCRIPTION Instances. DESCRIPTION	This field is meant for a text description of the event. As the POSEIDON framework require a number of additional fields, we instead use the description of the Google and Android data as the place to hold POSEIDON-specific data. This is a data structure in JSON format, described in section 0.
recurrence[]	Events.RDATE	A recurring event can specify recurring events with this field.
	Events.RRULE	A recurring event can specify rules for recurrence with this field.
location	Instances.EVENT_LOCATION Events.EVENT_LOCATION	Location of the event. The POSEIDON system uses this to connect an event to a route or destination for travel. Details given below.
start.timeZone **end.timeZone**	Events.EVENT_TIMEZONE	A time zone is specified when creating an event in the Android API. Google data has separate timeZone values for start and end times.

An event can be connected to a location or planned route stored on the POSEIDON file server. Each location configured for the user has a name. If navigation is added to an event, the location or route name is stored in the location field of the event. An event with location is interpreted as the planned time to start travelling to this location. A calendar application capable of starting navigation functionality should look for the location in the list of configured locations or routes of the user, and invoke navigation if a route is found.

Reminder

An event can have one or more reminder objects. This object specifies that the user should be notified of an upcoming event a specific number of minutes before event start. Calendar applications for the primary user must notify the user at the specified time.

The reminder entity is very simple, without any own message. Google Calendar can include default reminders which are applied to all events, unless the event data explicitly specifies not to use defaults. The default reminders are configured in Google Calendar, independently of POSEIDON. If used, such reminders will show up in the Android data.

In the Google Calendar API, reminders are included in the data structure for an event. Any number of event-specific reminders can be added to an array in this data. In the Android API, the reminder is a separate entity. Each reminder entity has a reference to the event it belongs to.

Google Calendar API	Android Calendar API	Description
	Reminders.EVENT_ID	For the Android API, a reminder entity is retrieved with a reference to the event it belongs to.

useDefault		Google Calendar has a configuration for default reminders, applied to all events unless this flag is false.
overrides[].method	Reminders.METHOD	Google Calendar supports several reminder methods. Email and SMS methods are supported in both APIs, and handled by Google. We do not recommend using these reminder types for our user group. The type “popup” in the Google data, corresponding to Reminders.METHOD_ALERT in Android, is the type to handle with notifications in an application.
overrides[].minutes	Reminders.MINUTES	Number of minutes before event start.

Attendee

For completeness we also include the attendee object. It represents a person or other entity which should be present at an event, mainly identified via email address. The POSEIDON framework does not make any special use of this object. It is useful for adding a social aspect to the calendar, specifying people to meet.

POSEIDON data extension

The POSEIDON framework adds additional data to events, in the form of event media and a list of instructions. To fit this into the data model of the Google and Android calendars, we put this data in the event’s description field. It is stored as a JSON structure, with the following possible content:

```
{
	description:<event description>,
	eventIcon:<resourceID or URL>,
	eventVideo:<resourceID or URL>,
	instructions:[
		{
			index:<place in sequence>,
			title:<instruction text>,
			image:<resourceID or URL>,
			audio:<resourceID or URL>,
			video:<resourceID or URL>
		}
	]
}
```

An event can have a list of instructions, where each should have at least one of title, image, audio or video. These are shown to the user in sequence. All media can be specified either as a URL to any online resource, or just as a resource ID for a file on the POSEIDON file server. Each element is also described in the table below.

JSON identifier	Datatype	Description
description	string	A text description for the event.
eventIcon	string	An image to represent the event, specified by URL or resource ID.
eventVideo	string	A video, as an audio-visual alternative to the description (URL or resource ID).

index	integer	Each instruction can explicitly specify its place in the sequence of instructions (low to high number), otherwise the sequence is given by the order in the JSON array.
title	string	Text of the instruction.
image	string	Image of the instruction; URL or resource ID.
audio	string	Audio of the instruction; URL or resource ID.
video	string	Video of the instruction; URL or resource ID.