

# Calendar Menu

- [Introduction](#)
- [Calendar Menu Properties](#)
  - [Configure Calendar Menu](#)
  - [Column to Data Mappings](#)
  - [Advanced](#)
    - [Google Calendar Integration](#)
    - [Events & UI](#)
  - [Performance & Offline](#)
    - [Cache Settings](#)
    - [PWA Offline Settings](#)
- [Download Demo App](#)

## Introduction

The **Calendar Menu** displays records from your preferred datalist as calendar events, in formats of day/week/month in your app's userview.

1. For this to work, you must map the **form** and **field IDs** that stores the calendar records, into the Calendar plugin properties page.
2. The Calendar Menu also supports Google API to display your preferred country's public holidays in the calendar.  
See [Google Calendar Integration](#) for more details.

To know more about Calendar Menu, do check out the [sample apps here](#).

## Calendar Menu Properties

### Configure Calendar Menu

⛶

Auto save when close? ☐ ✕

Configure Calendar Menu ?

⛶

Configure Calendar Menu > Column to Data Mappings > Advanced > Performance & Offline

Id

6E7AD0BA93DC4DA5A2C9E6F87F1094E0

Custom ID

Label \*

Calendar View

Page Title

Data Binder \*

Auto Handle Date Range? ?

☒

Figure 1: Calendar Menu Properties

Name	Description
id	Menu element unique id. Userview will use this id in the URL for the menu if the Custom ID is empty.
Custom ID	Item link slug. Optional field. <div><div><div><div>i</div><div>Ensure that value defined here is <b>unique</b> to other userview menus in the app, since the <b>first matching/conflicting ID</b> will take precedence in page loading.</div></div></div></div>
Label	The userview menu label.

Page Title	Calendar page title.
Data Binder	See list of available <a href="#">Datalist Binders</a> , to source the data for calendar entries.
Auto Handle Date Range	<p>Enable this option to automatically handle range of calendar events to be populated into the calendar, based on event dates and calendar view format.</p> <p>If this option is disabled, you need to manually handle date range via <code>#requestParam.start#</code> and <code>#requestParam.end#</code> <a href="#">request parameter hash variables</a> in your datalist binder configuration.</p> <p>Date format for date range is <code>yyyy-MM-dd</code>.</p>

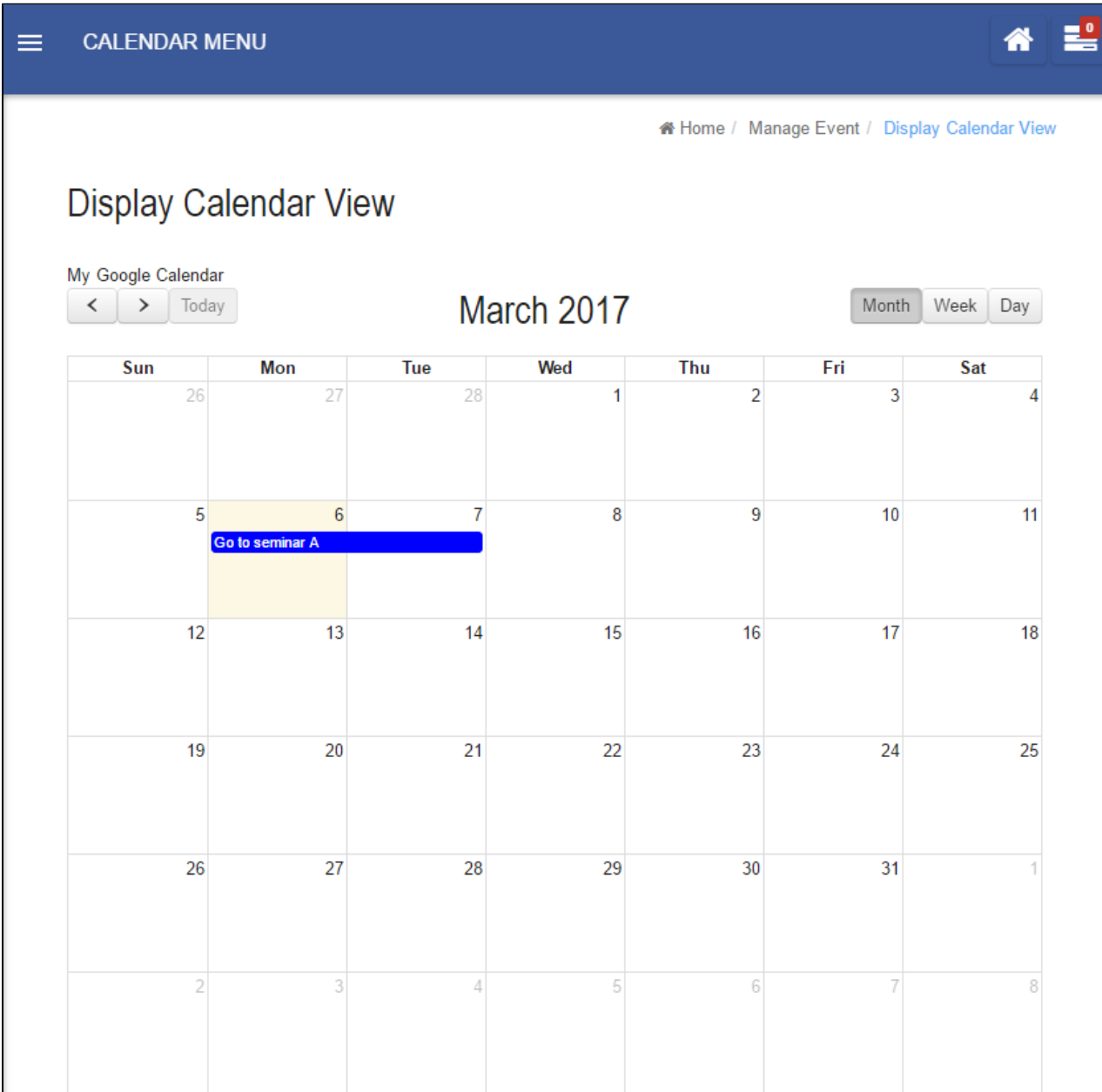


Figure 2: Sample Calendar Menu in runtime

Column to Data Mappings



To Date (column ID)	Calendar entry end date value. <div><div><div></div><div>You can also point to <a href="#">Date Picker</a> form field with time support.</div></div></div>
To Time (column ID)	Calendar entry end time value. <div><div><div></div><div>If the <b>To Date</b> value already contains a time value, leave this field empty.</div></div></div>
Date Format	You must specify the actual date format used by the <b>From/To Date</b> values.  Date Format in Java is expected here, e.g.: <b>yyyy-MM-dd</b> . <div><div><div></div><div>If the date values has a time value, just define the date format only and <b>not</b> the time format.  Because the time format is already handled by the <b>Time Format</b> property.</div></div></div>
Time Format	Available options: <ul style="list-style-type: none"><li>• 24 Hour System (e.g: 18:00)</li><li>• 12 Hour System (e.g: 6:00 PM)</li></ul>
Event Url (column ID)	Calendar entry URL.
Color (column ID)	Calendar entry CSS color code to style the calendar event.
CSS Class (column ID)	Calendar entry CSS class to style the calendar event.
Additional Data	Any additional data to be added to the calendar entry.

Advanced

Google Calendar Integration

Auto save when close? ☐ ✕

Advanced

Configure Calendar Menu > Column to Data Mappings > **Advanced** > Performance & Offline

Google Calendar Integration

Additional Data

GOOGLE CALENDAR API KEY

CALENDAR ID

CSS CLASSES

COLOR

Figure 4: Google Calendar Integration Properties

Name	Description
------	-------------

Additional Data	<p>Configurations to load calendar entries from Google Calendar.</p> <p><b>Notes if you want the Calendar Menu plugin to connect to Google Calendar:</b></p> <ol style="list-style-type: none"> <li>Every request your application sends to the Google Calendar API must include an authorization token. The token also identifies your application to Google. Here are the steps to obtain the Google Calendar API Key: <ul style="list-style-type: none"> <li>Go to the Google Developer Console and create a new project (it might take a second).</li> <li>Once in the project, go to APIs &amp; auth &gt; APIs on the sidebar.</li> <li>Find "Calendar API" in the list and turn it ON.</li> <li>On the sidebar, click APIs &amp; auth &gt; Credentials.</li> <li>In the "Public API access" section, click "Create new Key".</li> <li>Choose "Browser key".</li> <li>If you know what domains will host your calendar, enter them into the box. Otherwise, leave it blank. You can always change it later.</li> <li>Your new API key will appear. It might take second or two before it starts working.</li> <li>Copy and paste your API key into the Calendar menu properties "Configure Calendar Menu&gt;Data Binder&gt;Advanced" page.</li> </ul> </li> <li>Steps to enable your Google Calendar to be public: <ul style="list-style-type: none"> <li>In the Google Calendar interface, locate the "My calendars" area on the left.</li> <li>Hover over the calendar you need and click the downward arrow.</li> <li>A menu will appear. Click "Calendar Settings".</li> <li>Check "Make this calendar public".</li> <li>Make sure "Share only my free/busy information" is unchecked.</li> <li>Click "Save".</li> <li>Obtain your Google Calendar's ID: <ul style="list-style-type: none"> <li>In the Google Calendar interface, locate the "My calendars" area on the left.</li> <li>Hover over the calendar you need and click the downward arrow.</li> <li>A menu will appear. Click "Calendar settings".</li> <li>In the "Calendar Address" section of the screen, you will see your Calendar ID. It will look something like "abcd1234@group.calendar.google.com".</li> <li>Copy and paste your API key into the Calendar menu properties "Configure Calendar Menu&gt;Data Binder&gt;Advanced" page.</li> <li>You can read more at <a href="#">Google Calendar API</a> site.</li> </ul> </li> </ul> </li> </ol>
-----------------	--

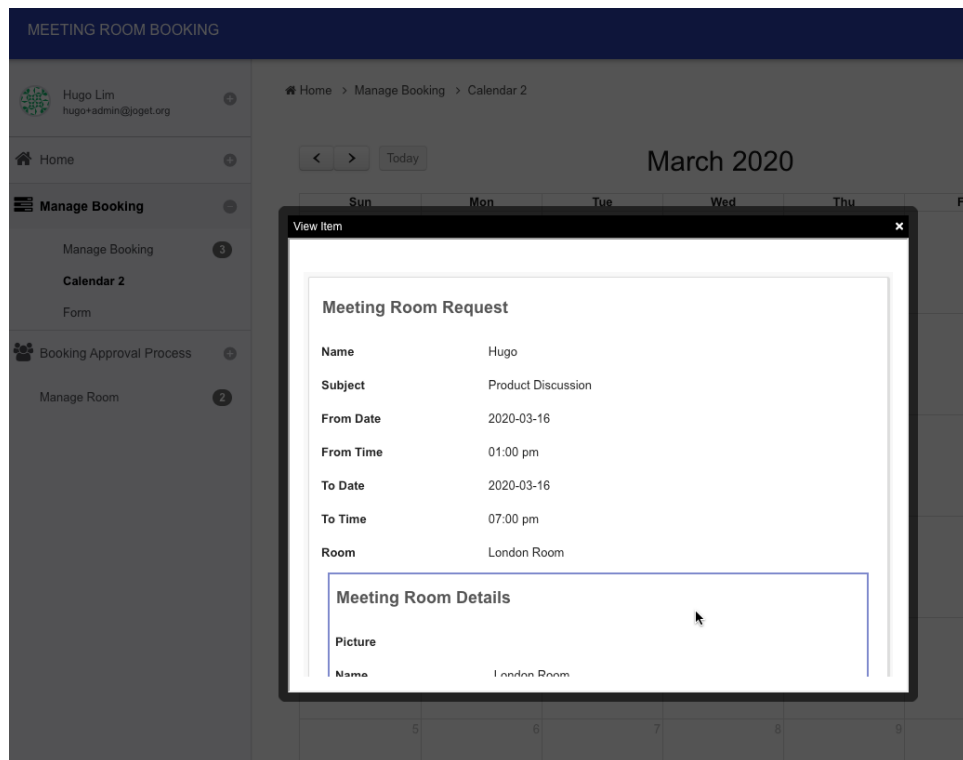
## Events & UI

Events & UI		
Event Rendering Callback (Javascript)	1	//console.log(event); console.log(element); console.log(view);
Event After Rendering Callback (Javascript)	1	//console.log(event); console.log(element); console.log(view);
Event After All Rendering Callback (Javascript)	1	//console.log(view);
Event Click Callback (Javascript)	1	//console.log(event); console.log(jsEvent); console.log(view);
Time Display Format ?		
Custom Header (HTML)	1	
Custom Footer (HTML)	1	

Figure 5: Events & UI Properties

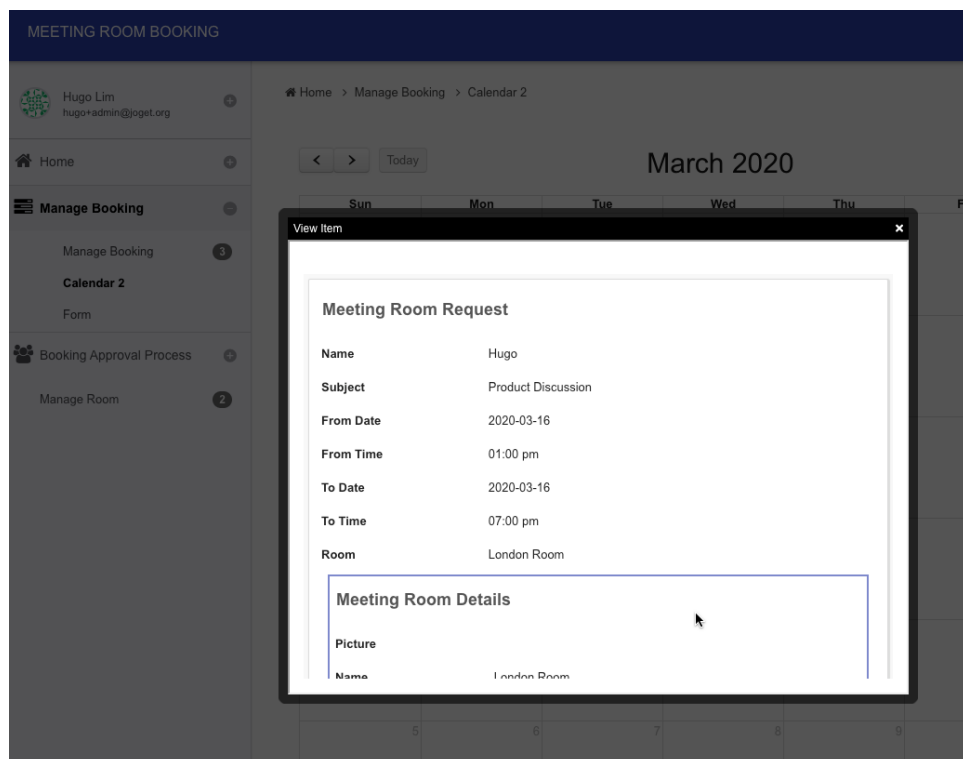
Name	Description
------	-------------

Event Rendering Callback (Javascript)	Custom Javascript code to execute upon rendering each calendar event.
Event After Rendering Callback (Javascript)	<p>Custom Javascript code to execute for each calendar event, after it has been rendered.</p> <div> <p><b>Sample Code to change the event time indicator to a more meaningful value</b></p> <pre> var fromTime = event.start.format("H:mm").toString(); var toTime = event.end.format("H:mm").toString();  if (fromTime.indexOf("10:00") &gt; -1 &amp;&amp; toTime.indexOf("16:20") &gt; -1) {     \$(element).find('span.fc-time').text("First Shift --&gt; "); } </pre> </div> <p>See sample app here for more info: <a href="#">APP_calendarTest-sample.jwa</a></p>
Event After All Rendering Callback (Javascript)	<p>Custom Javascript code to execute after all rendering of the calendar menu has been completed.</p> <p>In the sample below, upon clicking on a calendar date, it will redirect to an event form and prepopulate a date field with the clicked date.</p> <div> <p><b>Sample Code</b></p> <pre> \$("div.calendar_menu_body td.fc-day").click(function() {     var preferredDefaultTime = "09:00";     var dateFromFieldId = "date_from";     window.location = "calendarViewEditable?" + dateFromFieldId + "="     + \$(this).attr("data-date") + "%20" + preferredDefaultTime; }); </pre> </div>
Event Click Callback (Javascript)	<p>Custom Javascript code to execute when a calendar event is clicked on.</p> <div> <p><b>Sample Code to open up form view in a popup dialog</b></p> <pre> var popupActionDialog = new PopupDialog("EventFormMenuId?id="+event.id+"&amp;embed=true"); popupActionDialog.init(); </pre> </div>



Show in JPopup Dialog

```
var address = "EventFormMenuId?embed=true&id=" + event.id;  
JPopup.show("testPopup", address, {}, "View Item", "50%", "50%");
```





Time Display Format	<div>Refer to <a href="https://fullcalendar.io/docs/date-formatting-string">https://fullcalendar.io/docs/date-formatting-string</a> for formats.</div> <div><div>Example</div><div><pre>h(:mm)t //2:12p, 1p HH:mm //14:12, 13:00 hh(:mm)t //02:12p, 01p</pre></div></div>
Custom Header (HTML)	Custom header in HTML to show on the top of the calendar.
Custom Footer (HTML)	Custom footer in HTML to show on the bottom of the calendar.

## Performance & Offline

### Cache Settings

Name	Description
Scope	<p>Caching options are:</p> <ul style="list-style-type: none"> <li>• <b>None:</b> No caching.</li> <li>• <b>Application:</b> Cache content by application where all users will see the same content.</li> <li>• <b>User:</b> Cache by username.</li> </ul>
Duration (s)	Duration in seconds to cache content, defaults to 20 seconds if applicable.

Read more at [Performance Improvement with Userview Caching](#).

### PWA Offline Settings

**Note:** PWA Offline support is dependent on the theme used.

Name	Description
Enable cache for offline support	Enable caching for offline support, usually set for use on mobile browsers.

Read [Progressive Web Application \(PWA\)](#) for more information.

## Download Demo App

- [Calendar Menu App](#) in Joget Marketplace (Note: For Joget Workflow v5 only)
- [APP\\_calendarTest-sample.jwa](#) - Sample use of **Event After Rendering Callback** field

