## Overview

The Calendar component rule, **calendarDisplayField**, is defined with the following inputs:

| **Param** | **Type** | **Default Value** | **Options** |
| --- | --- | --- | --- |
| title | String / Input ***- optional*** | Month Year (e.g. January 2021) |  |
| showTitle | Boolean / Input ***- optional*** | true |  |
| viewType | String / Input ***- required*** | week view | DAY, WEEK, MONTH |
| dateContext | Text / Input ***- required*** | Today |  |
| detailDateFormat | Text / Input ***- optional*** | MM/DD/YYYY |  |
| visibleWeeks | Integer / Input ***- optional*** | All weeks for a month |  |
| weekendDisplay | String / Input ***- optional*** | Show weekends | FULL, NARROW, NONE |
| useDetailPopup | Boolean / Input ***- optional*** | True |  |
| readOnly | Boolean / Input ***- optional*** | False |  |
| events | Dictionary?list / Input ***- optional*** | n/a |  |
| selectedEvent | Text / Output ***- optional*** | n/a |  |
| theme | Dictionary ***- optional*** | Basic theme using site’s Accent color |  |
| timezone | Text - ***optional*** | Browser timezone | TZ timezone name or code (<https://en.wikipedia.org/wiki/List_of_tz_database_time_zones>) |

### Timezones

This component expects all times are provided as GMT since the component inherits the browser/system timezone. Your events may appear at incorrect times if you do not follow this guidance.

### Provided Utilities

* Rule - ***GCG\_RULE\_Util\_mergeChangedEvent***
	+ If working with an editable calendar, this rule helps with merging a changed event back into a local variable (typically your *events* input
* Rule - ***GCG\_CAL\_Util\_convertDatetimetoUTC***
	+ Shows an example of taking a datetime string and transforming it to UTC format.
* Site - ***GCG Calendar Demo***
	+ Showcases a few working implementations of the calendar

### Event Structure

The events dictionary has the following support elements. Note that ToastUI supports more elements but they are not officially supported by this component in the current release:

| **Param** | **Required** | **Type** | **Description** |
| --- | --- | --- | --- |
| ***id*** | No | Integer | A unique integer value for an event, such as a database Primary Key. Not required, but strongly recommended. |
| ***calendarId*** | Yes | Integer | Must use **1** to inherit styling characteristic.. |
| ***category*** | Yes | Text | Must be **time** at this time. |
| ***start*** | No | UTC formatted timestamp | Start datetime of an event. If left blank, the event time will be marked as the current time. Must be GMT to support browser timezone offset. |
| ***end*** | No | UTC formatted timestamp | End datetime of an event. If left blank, the event will end 30 minutes after the start time. Must be GMT to support browser timezone offset. |
| ***location*** | No | Text | Location of event. |
| ***attendees*** | No | Text | Text?list | Attendees of event. Can be a single string or a Dictionary array. Strings are presented as-is; Arrays are joined with a comma delimiter |
| ***body*** | No | Text | Body of event. |
| ***color*** | No | Text | Text color of title in event card. Supports HTML color, HEX code, or rgba() string. |
| ***bgColor*** | No | Text | Background color of event card. Supports HTML color, HEX code, or rgba() string. |
| ***borderColor*** | No | Text | Color of left bar of event card. Supports HTML color, HEX code, or rgba() string. |

## Example Events and Behaviors

The following example events showcase how an event may render based on different inputs:

| **Scenario** | **Definition** | **Reference Screenshot** |
| --- | --- | --- |
| **Typical event** |  { id: "5", calendarId: "1", title: "Swim Practice", category: "time", start: "2020-01-30T09:30:00.000Z", end: "2020-01-30T10:30:00.000Z", location: "Swimming pool", attendees:"Mary, John, and Sue", body: "Need to do 10 laps" }  |  |
| **No or invalid calendar Id** - A*ccent color is lost* |  { id: "5", title: "Swim Practice", category: "time", start: "2020-01-30T09:30:00.000Z", end: "2020-01-30T10:30:00.000Z", location: "Swimming pool", attendees:"Mary, John, and Sue", body: "Need to do 10 laps" }  |  |
| **Title and Time only** |  { id: "5", calendarId: "1", title: "Swim Practice", category: "time", start: "2020-01-30T09:30:00.000Z", end: "2020-01-30T10:30:00.000Z" }  |  |
| **Multi-day event** - Appears in top section. S*hows date based on date parameter (MM/DD/YYYY in example 1, YYYY-MM-DD in example 2)* |  { id: "4", calendarId: "1", title: "Study Group", category: "time", start: "2020-01-27T01:00:00.000Z", end: "2020-01-28T02:00:00.000Z", location: "Jim's House", body: "Cram session for the exam." } |  |
| **Missing Endtime** - *Event ends 30 minutes after Start* |  { id: "5", calendarId: "1", title: "Swim Practice", category: "time", start: "2020-01-30T09:30:00.000Z } |  |
| **Missing Start and End** - *Event will load for a 30 minute window based on current time* |  { id: "5", calendarId: "1", title: "Swim Practice", category: "time" } |  |
| **Missing Start Only** - *Event will load with incorrect timing* |  { id: "5", calendarId: "1", title: "Swim Practice", category: "time", end: "2020-01-30T10:30:00.000Z" } |  |
| **Missing Category** |  { id: "4", calendarId: "1", title: "Study Group", start: "2020-01-27T01:00:00.000Z", end: "2020-01-28T02:00:00.000Z", location: "Jim's House", body: "Cram session for the exam." } | Event will NOT appear. |
| **Individual event colors** | { id:”2”, calendarId:”1”, title:”My Title”, category:”time”, location:”My Location”, start:”2023-02-21T14:30:00.000Z”, end:”2023-02-21T19:00:00.000Z,” attendees:”My attendees”, body:”My Body”, bgColor:”GreenYellow”, color:”#FF0000} |  |

### Styling

The calendar inherits the accent color of your Site or Tempo environment. Additional styling can be set using the **theme** parameter. Some examples are available in the sample interface ***GCG\_CAL\_CMPT\_exampleFive***, and the complete list of overridable attributes is available in the ToastUI documentation @ <https://github.com/nhn/tui.calendar/blob/main/docs/en/apis/theme.md>

To adjust styling, set the ***theme*** parameter to a dictionary containing a list of attributes and style values (typically either a color or a small css snippet). Colors can generally be set by a common html color (e.g. “red”, “blue”), a valid hex code, or an RGBa string.

Some styles only apply to specific calendar views – generally speaking, the first word will indicate if it’s a common, monthly, weekly, or other style. Styles related to unsupported calendar views are not supported.

**NOTE**: If you set the theme and want to change it, at this time you will need to re-render the entire calendar component to ensure all previously set styles are reset. As such, it is generally not recommended to allow user interactions to dynamically change styles.

## Example Styling

The following example events showcase how an event may render based on different inputs to the ***theme*** input:

|  |  |  |
| --- | --- | --- |
| **Scenario** | **Definition** | **Reference Screenshot** |
| **Color in Background** | { "common.backgroundColor":  “rgba(0, 255, 0, 0.5)”} |  |
| **Thick Borders** | { "common.border":  "4px groove #e5e5e5", "month.dayname.borderLeft":  "4px groove #e5e5e5", "week.daygridLeft.borderRight": "4px groove #e5e5e5" }  |  |
| **Darker Weekend and Out of Month Dates** | { "common.saturday.color":  "blue", "common.holiday.color":  "green", "month.holidayExceptThisMonth.color": "#FF00AB", "month.dayExceptThisMonth.color":  "rgba(10, 10, 10, 1)"} |  |
| **Altering the current time indicators(past, present, future)** | { "week.daygrid.backgroundColor":  "green", "week.currentTimeLineToday.border":  "4px solid green", "week.currentTimeLineFuture.border":  "1px solid red"} |  |
| **Background Colors** |  {  "month.dayname.backgroundColor":  "rgba(80, 80, 80, 0.8)", "common.backgroundColor": "rgba(100, 100, 100, 0.2)", "common.dayname.color": "#000", "common.holiday.color": "#000", "common.saturday.color":  "#000", "month.weekend.backgroundColor":  "rgba(80, 80, 80, 0.8)", "month.weekday.backgroundColor":  "rgba(80, 80, 80, 0.8)", "month.holidayExceptThisMonth.color":  "#000", "month.dayExceptThisMonth.color":  "#000" } |  |

All theme options with default values below:

 'common.border': '1px solid #e5e5e5',

 'common.backgroundColor': 'white',

 'common.holiday.color': '#ff4040',

 'common.saturday.color': '#333',

 'common.dayname.color': '#333',

 'common.today.color': '#333',

 // creation guide style

 'common.creationGuide.backgroundColor': 'rgba(81, 92, 230, 0.05)',

 'common.creationGuide.border': '1px solid #515ce6',

 // month header 'dayname'

 'month.dayname.height': '31px',

 'month.dayname.borderLeft': 'none',

 'month.dayname.paddingLeft': '10px',

 'month.dayname.paddingRight': '0',

 'month.dayname.backgroundColor': 'inherit',

 'month.dayname.fontSize': '12px',

 'month.dayname.fontWeight': 'normal',

 'month.dayname.textAlign': 'left',

 // month day grid cell 'day'

 'month.holidayExceptThisMonth.color': 'rgba(255, 64, 64, 0.4)',

 'month.dayExceptThisMonth.color': 'rgba(51, 51, 51, 0.4)',

 'month.weekend.backgroundColor': 'inherit',

 'month.day.fontSize': '14px',

 // month schedule style

 'month.schedule.borderRadius': '2px',

 'month.schedule.height': '24px',

 'month.schedule.marginTop': '2px',

 'month.schedule.marginLeft': '8px',

 'month.schedule.marginRight': '8px',

 // month more view

 'month.moreView.border': '1px solid #d5d5d5',

 'month.moreView.boxShadow': '0 2px 6px 0 rgba(0, 0, 0, 0.1)',

 'month.moreView.backgroundColor': 'white',

 'month.moreView.paddingBottom': '17px',

 'month.moreViewTitle.height': '44px',

 'month.moreViewTitle.marginBottom': '12px',

 'month.moreViewTitle.borderBottom': 'none',

 'month.moreViewTitle.padding': '12px 17px 0 17px',

 'month.moreViewList.padding': '0 17px',

 // week header 'dayname'

 'week.dayname.height': '42px',

 'week.dayname.borderTop': '1px solid #e5e5e5',

 'week.dayname.borderBottom': '1px solid #e5e5e5',

 'week.dayname.borderLeft': 'none',

 'week.dayname.paddingLeft': '0',

 'week.dayname.backgroundColor': 'inherit',

 'week.dayname.textAlign': 'left',

 'week.today.color': 'inherit',

 'week.pastDay.color': '#bbb',

 // week vertical panel 'vpanel'

 'week.vpanelSplitter.border': '1px solid #e5e5e5',

 'week.vpanelSplitter.height': '3px',

 // week daygrid 'daygrid'

 'week.daygrid.borderRight': '1px solid #e5e5e5',

 'week.daygrid.backgroundColor': 'inherit',

 'week.daygridLeft.width': '72px',

 'week.daygridLeft.backgroundColor': 'inherit',

 'week.daygridLeft.paddingRight': '8px',

 'week.daygridLeft.borderRight': '1px solid #e5e5e5',

 'week.today.backgroundColor': 'rgba(81, 92, 230, 0.05)',

 'week.weekend.backgroundColor': 'inherit',

 // week timegrid 'timegrid'

 'week.timegridLeft.width': '72px',

 'week.timegridLeft.backgroundColor': 'inherit',

 'week.timegridLeft.borderRight': '1px solid #e5e5e5',

 'week.timegridLeft.fontSize': '11px',

 'week.timegridOneHour.height': '52px',

 'week.timegridHalfHour.height': '26px',

 'week.timegridHalfHour.borderBottom': 'none',

 'week.timegridHorizontalLine.borderBottom': '1px solid #e5e5e5',

 'week.timegrid.paddingRight': '8px',

 'week.timegrid.borderRight': '1px solid #e5e5e5',

 'week.timegridSchedule.borderRadius': '2px',

 'week.timegridSchedule.paddingLeft': '2px',

 'week.currentTime.color': '#515ce6',

 'week.currentTime.fontSize': '11px',

 'week.currentTime.fontWeight': 'normal',

 'week.currentTimeLinePast.border': '1px dashed #515ce6',

 'week.currentTimeLineBullet.backgroundColor': '#515ce6',

 'week.currentTimeLineToday.border': '1px solid #515ce6',

 'week.currentTimeLineFuture.border': 'none',

 // week creation guide style

 'week.creationGuide.color': '#515ce6',

 'week.creationGuide.fontSize': '11px',

 'week.creationGuide.fontWeight': 'bold',

 // week daygrid schedule style

 'week.dayGridSchedule.borderRadius': '2px',

 'week.dayGridSchedule.height': '24px',

 'week.dayGridSchedule.marginTop': '2px',

 'week.dayGridSchedule.marginLeft': '8px',

 'week.dayGridSchedule.marginRight': '8px'