



# MyCalendar (version:2.0)

Prepared by FeyaSoft Inc. ([www.feyasoft.com](http://www.feyasoft.com))

## Reference Manual version:2.0

**Notice:** This is the second version and simple document of myCalendar, there is many features and functions miss in this manual, and we will complete them step by step.

## Content

- **API Reference**
  - **Class Ext.ux.calendar.CONST**
  - **Class Ext.ux.calendar.CalendarWin**
  - **Class Ext.ux.calendar.DataSource**
  - **Class Ext.ux.calendar.EventHandler**
  - **Class Ext.ux.calendar.CalendarLayout**
  - **Class Ext.ux.calendar.BackThread**
  - **Class Ext.ux.calendar.DayView**
  - **Class Ext.ux.calendar.MonthView**
  - **Class Ext.ux.calendar.CalendarContainer**
- **Note about myCalendar**
- **History Log**

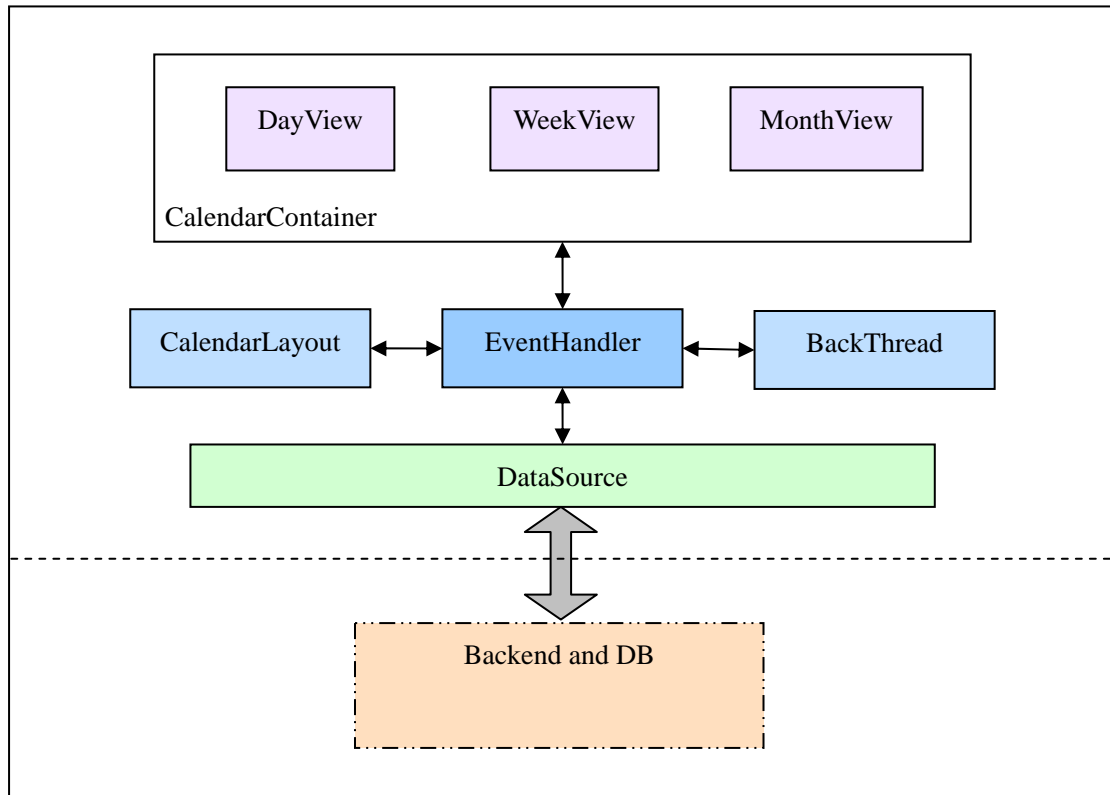
## NOTICE

This document is only licensed for myCalendar commercial licensed user who bought one of FeyaSoft myCalendar license, and contains confidential information that is proprietary to FeyaSoft Inc. No part of its contents may be used, copied, disclosed, or conveyed to any party in any manner whatsoever without prior written permission from FeyaSoft Inc.

FeyaSoft and FeyaSoft logo are trademarks of FeyaSoft Inc.

© Copyright 2006-2009 by FeyaSoft Inc. All rights reserved.

## API Reference



### Class Ext.ux.calendar.CONST

<b>Package</b>	Ext.ux.calendar
<b>Defined in</b>	Common/CONST.js
<b>Extends</b>	Object

*This class is for all constant variables and config path used in myCalendar.*

Notice: You may need change all the paths in this file.

### Config Options

#### **SHOW\_LANGUAGE\_MENU: Boolean**

*true to show the language submenu in myCalendar, or not*

#### **BLANK\_IMAGE\_URL: Boolean**

*The blank image URL*

#### **MAIN\_PATH: String**

*The main path of your domain, for example: '/myCalendar/js/feyaSoft/home/program/calendar/'*

**CALENDAR\_LANGUAGE\_PATH: String**

*define the multi-language path of myCalendar*

**EXT\_LANGUAGE\_PATH: String**

*define the multi-language path of EXT*

*/\**

*\* Below are some URL for submitting/requesting data from backend*

*\*/*

**searchURL: String**

**showAllCalendarURL**

**showOnlyCalendarURL**

**createUpdateCalendarURL**

**deleteEventsByCalendarURL**

**deleteCalendarURL**

**loadCalendarURL**

**loadEventURL**

**loadRepeatEventURL**

**createEventURL**

**updateEventURL**

**deleteEventURL**

**deleteRepeatEventURL**

**changeDayURL**

**deleteDayURL**

**loadSettingURL**

**updateSettingURL**

**createUpdateRepeatEventURL**

*define the related URLs used in DataSource.js, see DataSource.js for more details.*

## Class Ext.ux.calendar.CalendarWin

<b>Package</b>	Ext.ux.calendar
<b>Defined in</b>	CalendarWin.js
<b>Extends</b>	Ext.Window

*This class is for the myCalendar window.*

Notice: There is one thing you need config, `userId`, and also you can config the setting of `myCalendar` here if you don't want load it from backend.

### Config Options

#### **userId: int**

*The id of the user login, if your system don't have this kind of stuff, just set it to a number.*

#### **hourFormat: string**

*The displayed hour format in myCalendar*

#### **dayFormat: string**

*The displayed day format in DayView*

#### **weekFormat: string**

*The displayed day format in WeekView*

#### **monthFormat: string**

*The displayed day format in MonthView*

#### **language: string**

*The displayed language of myCalendar*

#### **activeStartTime: string**

*The active start time of a day, for example: 09:00*

#### **activeEndTime: string**

*The active end time of a day, for example: 19:00*

#### **intervalSlot: string**

*The time interval of a row in dayview/weekview, for example: 30 for 30 minutes*

#### **startDay: string**

*The startDay of a week. 0 for Sunday; 1 for Monday*

#### **singleDay: string**

*True for disable cross-event function*

**createByDblclick: string**

*True for create event by double click. Notice, if set it true, then the drag create event function will be also disabled*

## Class Ext.ux.calendar.DataSource

<b>Package</b>	Ext.ux.calendar
<b>Defined in</b>	DataSource.js
<b>Extends</b>	Ext.util.Observable

*This class represents the primary interface of Server request.*

**Notice:** If you want to use your own backend, maybe you need rewrite this file, here we only define the input/output, once your backend can work with these input/output, it will be fine.

If you're a 1.0 user, you should know this version is a little different, so you have to re-write or change your backend, so be careful with upgrade.

### Config Options

### Functions

#### **showAllCalendar:function(sucessFn, scope)**

*For requesting backend to show all calendars*

##### **Parameters:**

{Function} **sucessFn:** The callback function when request completed successfully

*This function will add a new calendar or modify a calendar on UI*

{object } **scope:** The scope of sucessFn function

#### **showOnlyCalendar:function(calendarId, sucessFn, scope)**

*For requesting backend to hide all calendars but only show this one*

##### **Parameters:**

{int} **calendarId:** the id of a calendar

{Function} **sucessFn:** The callback function when request completed successfully

*This function will add a new calendar or modify a calendar on UI*

{object } **scope:** The scope of sucessFn function

#### **createUpdateCalendar :function(calendar, sucessFn, scope)**

*For requesting backend to create or update a calendar*

##### **Parameters:**

{object} **Calendar:** the object of a calendar, should contain all field of calendar table in db

*id: int*

*The id of calendar, primary key*

*name: string*

*The name of calendar*

*description: string*

*The description of calendar*

*color: string*

*The color of calendar, it's a string, should be one of ["blue", "red", "cyan", "orange",*

*"green", "purple"]*, see *colorIndex* in *common/Mask.js* for detail, if you want to add more color, you need add it in *colorIndex* array and *colors* array in *Mask.js*, and also you need add related *css* in *css/calendar.css*

*hide*: boolean

*The status of calendar, true for hide, false for show*

{Function} **sucessFn**: The callback function when request completed successfully

*This function will add a new calendar or modify a calendar on UI*

{object } **scope**: The scope of sucessFn function

**deleteEventsByCalendar:function(calendarId, sucessFn, scope)**

*For delete all events belong to a calendar*

**Parameters:**

{int} **calendarId**: The id of a calendar

{function} **sucessFn**: The callback function when request completed successfully

*This function should remove all events related to this calendar from UI*

{object} **scope**: The scope of sucessFn function

**deleteCalendar:function(calendarId, sucessFn, scope)**

*For delete a calendar and all events belong to it*

**Parameters:**

{int} **calendarId**: The id of a calendar

{function} **sucessFn**: The callback function when request completed successfully

*This function will delete this calendar and all events related to it from UI*

{object} **scope**: The scope of sucessFn function

**loadCalendar:function(sucessFn, scope)**

*For load all calendars of a user*

**Parameters:**

{int} **userId**: The id of a user

{function} **sucessFn**: The callback function when request completed successfully

*This function will render all calendars on UI*

{obj} **scope**: The scope of sucessFn function

**loadEvent:function(startDate, endDate, sucessFn, scope)**

*For load all events from the start day to the end day*

**Parameters:**

{Date} **startData**: The start date

*The format should be Y-m-d*

{Date} **endData**: The end date

*The format should be Y-m-d*

{function} **sucessFn**: The callback function when request completed successfully

*This function will render all returned events on UI*

{object} **scope**: The scope of sucessFn function



**loadRepeatEvent:function(sucessFn, scope)**

*For load all repeat events from the DB*

**Parameters:**

- {function} **sucessFn**: The callback function when request completed successfully  
*This function will render all returned events on UI*
- {object} **scope**: The scope of sucessFn function

**createEvent:function(event, sucessFn, scope)**

*For create an event*

**Parameters:**

- {object} **event**: The object of event
- {function} **sucessFn**: The callback function when request completed successfully  
*This function will render a new event on UI*
- {object} **scope**: The scope of sucessFn function

**updateEvent:function(event, sucessFn, scope)**

*For update an event*

**Parameters:**

- {object} **event**: the object of event
- {function} **sucessFn**: the callback function when request completed successfully  
*This function will change the event on UI*
- {object} **scope**: the scope of sucessFn function

**deleteEvent:function(event, sucessFn, scope)**

*For delete an event*

**Parameters:**

- {object} **event**: the object of event
- {function} **sucessFn**: the callback function when request completed successfully  
*This function will remove the event from UI*
- {object} **scope**: the scope of sucessFn function

**deleteRepeatEvent:function(event, sucessFn, scope)**

*For delete a repeat event*

**Parameters:**

- {object} **event**: the object of event
- {function} **sucessFn**: the callback function when request completed successfully  
*This function will remove the event from UI*
- {object} **scope**: the scope of sucessFn function

**changeDay:function(oday, nday, sucessFn, scope, keep)**

*For change all events in a day to another day*

**Parameters:**

{string} **oday**: The old day, all events belong to this day need be changed  
 {string} **nday**: The new day, all events belong to old day will change to this day  
 {function} **sucessFn**: The callback function when request completed successfully  
     *This function will remove the events belong to old day and add them to new day on UI*  
 {object} **scope**: The scope of sucessFn function  
 {boolean} **keep**: If true will keep the events for old day, if false then delete events for old day

**deleteDay:function(day, sucessFn, scope)**

*For delete all events in a day*

**Parameters:**

{string} **day**: All events belong to this day need be deleted  
 {function} **sucessFn**: The callback function when request completed successfully  
     *This function will remove all events belong to this day from UI*  
 {object} **scope**: The scope of sucessFn function

**loadSetting:function(userId, sucessFn, scope)**

*For load setting of feyaCalendar, this function called at the very beginning of myCalendar, the myCalendar will start to init after this request completed*

**Parameters:**

{int} **userId**: the ID of current user  
 {function} **sucessFn**: The callback function when request completed successfully  
     *This function will start to init myCalendar with these settings*  
 {object} **scope**: The scope of sucessFn function

**updateSetting:function(obj, sucessFn, scope)**

*For save setting of feyaCalendar*

**Parameters:**

{object} **obj**: The object of current setting, it should contain:  
     *hourFormat: string*  
         *The hour format in myCalendar*  
     *dayFormat: string*  
         *The day format in DayView*  
     *weekFormat: string*  
         *The day format in WeekView*  
     *monthFormat: string*  
         *The day format in monthView*  
 {function} **sucessFn**: The callback function when request completed successfully  
 {object} **scope**: The scope of sucessFn function

**createUpdateRepeatEvent:function(event, oevent, sucessFn, scope)**

*For requesting backend to create or update a calendar*

**Parameters:**

{object} **event**: the object of repeat event to create or update  
 {Object} **oevent**: the object of the old repeat event to be updated

{Function} **sucessFn:** The callback function when request completed successfully  
{object } **scope:** The scope of sucessFn function

## Class Ext.ux.calendar.EventHandler

<b>Package</b>	Ext.ux.calendar
<b>Defined in</b>	EventHandler.js
<b>Extends</b>	Ext.util.Observable

*This class represents the primary interface of render, add/edit/delete, drag/drop calendar Event function.*

### Config Options

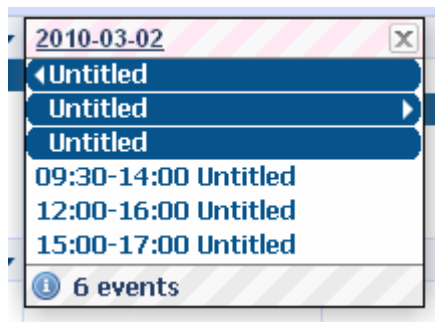
#### **baseIndex: int**

*The based z-index of events, all event Element will have a z-index  $\geq$  baseIndex.*

### Properties

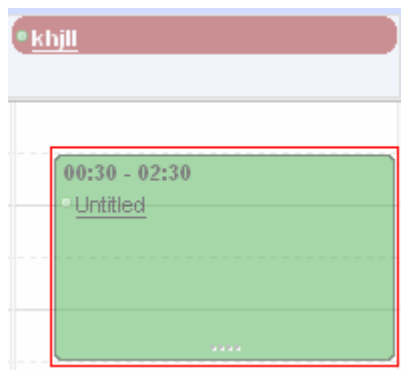
#### **detailTpl: Ext.XTemplate**

*The XTemplate for a cell in pool to show all whole events on that day.*



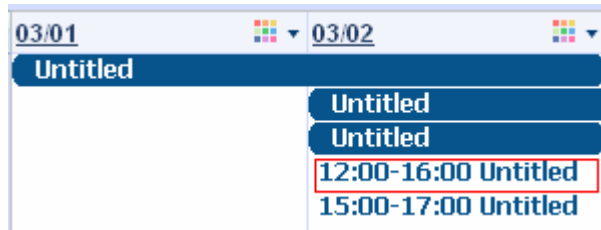
#### **eventTpl: Ext.XTemplate**

*This XTemplate is for generate event element in DayView or WeekView.*



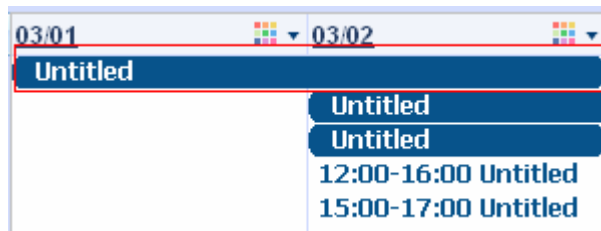
#### **legendTpl: Ext.XTemplate**

*This XTemplate is for generate event element in MonthView.*



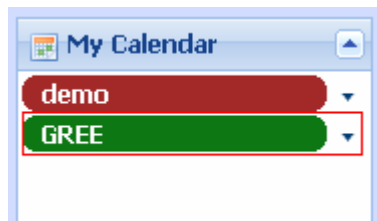
#### wholeTpl: Ext.XTemplate

This XTemplate is for generate a whole day event element in the pool of DayView, WeekView or MonthView.



#### calendarDropTpl: Ext.XTemplate

This XTemplate is for generate a calendar element with drop button.



#### calendarTpl: Ext.XTemplate

This XTemplate is for generate a calendar element without drop button.

Today Search				
		Time	Calendar	Subject
		2010-03-04 09:30 to 14:00	demo	(No Subject)
		2010-03-02 to 2010-03-03 Whole day	demo	(No Subject)

## Functions

#### createEvent:function(event, cview, col)

For create an event in DB and then render it on UI

#### Parameters:

{object} **event**: the event config to create

The event config should contain:

*eventId*: string

**Notice:** Use a fake Id here, for example 'prepare', because this function will call CreateEvent function in DataSource.js to create a record in backend, and

then replace the fake Id with the new Id in db.

*calendarId: int*

*The calendar id this event belong to*

*color: string*

*The color of this event, it should be one of {"blue", "red", "cyan", "orange", "green", "purple"}*

*startRow: int*

*The start time of event, startRow is in [0, 47], for 00:00-23:30*

*endRow: int*

*The end time of event, endRow is in [1, 48], for 00:30-24:00*

*day: string*

*The start day of this event, should be 'Y-m-d' format*

*eday: string*

*The end day of this event, should be 'Y-m-d' format*

*subject: string*

*The subject of this event*

*content: string*

*The content of this event*

*alertFlag: Boolean*

*true means myCalendar will alert user when this event is beginning*

*repeatType: string or object*

*Can be "no", "exception" or an object of repeat info*

**{class} cview:**

*an instance of DayView or MonthView; the container where these events to render*

**{int} col:** the index of column in cview to render the events,

*For DayView, it's always 0,*

*for WeekView, it's from 0 to 6,*

*for MonthView, it's from 0 to 41*

**updateEvent:function(event, cview, col)**

*For update an event in DB and then modify it on UI*

**Parameters:**

**{object} event:** the event config to create

*The event config should contain:*

*eventId: int*

*The id of this event*

*calendarId: int*

*The calendar id this event belong to*

*color: string*

*The color of this event, it should be one of {"blue", "red", "cyan", "orange", "green", "purple"}*

*startRow: int*

*The start time of event, startRow is in [0, 47], for 00:00-23:30*

*endRow: int*

*The end time of event, endRow is in [1, 48], for 00:30-24:00*

*day: string*

*The start day of this event, should be 'Y-m-d' format*

*eday: string*

*The end day of this event, should be 'Y-m-d' format*

*subject: string*

*The subject of this event*

*content: string*

*The content of this event*

*alertFlag: Boolean*

*true means myCalendar will alert user when this event is beginning*

*repeatType: string or object*

*Can be "no", "exception" or an object of repeat info*

**{class} cview:**

*an instance of DayView or MonthView; the container where these events to render*

**{int} col:** *the index of column in cview to render the events,*

*For DayView, it's always 0,*

*for WeekView, it's from 0 to 6,*

*for MonthView, it's from 0 to 41*

**deleteEvent:function(event, cview, col)**

*For delete an event from db and remove it from UI*

**Parameters:**

**{object} event:** *the event config to create*

*The event config should contain:*

*eventId: int*

*The id of this event*

*calendarId:int*

*The calendar id this event belong to*

*color: string*

*The color of this event, it should be one of {"blue", "red", "cyan", "orange", "green", "purple"}*

*startRow: int*

*The start time of event, startRow is in [0, 47], for 00:00-23:30*

*endRow: int*

*The end time of event, endRow is in [1, 48], for 00:30-24:00*

*day: string*

*The start day of this event, should be 'Y-m-d' format*

*eday: string*

*The end day of this event, should be 'Y-m-d' format*

*subject: string*

*The subject of this event*

*content: string*

*The content of this event*

*alertFlag: Boolean*

*true means myCalendar will alert user when this event is beginning*

*repeatType: string or object*

*Can be "no", "exception" or an object of repeat info*

**{class} cview:**

*an instance of DayView or MonthView; the container where these events to render*

**{int} col:** *the index of column in cview to render the events,*

*For DayView, it's always 0,*

*for WeekView, it's from 0 to 6,*

*for MonthView, it's from 0 to 41*

**deleteRepeatEvent:function(event, cview, makeException)**

*For delete a repeat event from db and remove it from UI, or just delete current instance of this repeat event*

**Parameters:**

**{object} event:** *the event config to create*

*The event config should contain:*

*eventId: int*

*The id of this event*

*calendarId:int*

*The calendar id this event belong to*

*color: string*

*The color of this event, it should be one of {"blue", "red", "cyan", "orange", "green", "purple"}*

*startRow: int*

*The start time of event, startRow is in [0, 47], for 00:00-23:30*

*endRow: int*

*The end time of event, endRow is in [1, 48], for 00:30-24:00*

*day: string*

*The start day of this event, should be 'Y-m-d' format*

*eday: string*

*The end day of this event, should be 'Y-m-d' format*

*subject: string*

*The subject of this event*

*content: string*

*The content of this event*

*alertFlag: Boolean*

*true means myCalendar will alert user when this event is beginning*

*repeatType: string or object*

*Can be "no", "exception" or an object of repeat info*

**{class} cview:**

*an instance of DayView or MonthView; the container where these events to render*

**{boolean} makeException:** *true for make an exception, or delete the whole repeat event*



**hideCalendar:function(calendar, flag)**

*This function is for hide/show calendar and all events belong to it on UI, it will also save the status in db.*

**Parameters:**

{object} **calendar:** The object of calendar  
 {boolean} **flag:** true for hide, false for show

**changeColor:function(calendar, index)**

*For change color of a calendar and all events belong to it*

**Parameters:**

{object} **calendar:** the object of calendar  
 {string} **index:** the index of the color, should be one of {"blue", "red", "cyan", "orange", "green", "purple"}

**renderEvent:function(cview, rs, col)**

*This function is for render and init an event in DayView or WeekView.*

**Parameters:**

{object} **event:** The config of event  
 {class} **cview:**  
     an instance of DayView or MonthView; the container where these events to render  
 {int} **col:** the index of column in cview to render the events  
     *For DayView, it's always 0,*  
     *for WeekView, it's from 0 to 6,*  
     *for MonthView, it's from 0 to 34*

**generateLegend:function(cview, e)**

*For generate an event Element in monthView or in the pool of DayView and WeekView*

**Parameters:**

{Class} **cview:**  
     an instance of DayView or MonthView; the container where these events to render  
 {Object} **e:** an object of event

**generateLegend:function(cview, e)**

*For generate an event Element in monthView or in the pool of DayView and WeekView*

**Parameters:**

{Class} **cview:**  
     an instance of DayView or MonthView; the container where these events to render  
 {Object} **e:** an object of event

**updateRepeatEvent:function(event, cview, oevent)**

*For update a repeat event*

**Parameters:**

{Object} **event:** an object of the new event

{Class} **cview:**

an instance of DayView or MonthView; the container where these events to render

{Object} **oevent:** an object of old event

Notice: at least one of event or oevent's repeatType is not "no" or "exception"

**reloadCalendar:function(container, cflag)**

*For reload all Calendar and events*

**Parameters:**

{Element} **container:** the element to contain the calendar element

{boolean} **cflag:** true for load calendars, or only load events

Notice: after call this function, all cache will be clean

## Class Ext.ux.calendar.CalendarLayout

<b>Package</b>	Ext.ux.calendar
<b>Defined in</b>	Layout/CalendarLayout.js
<b>Extends</b>	Ext.util.Observable

*This class handle the layout of events in dayview and events in monthview, it will contain a cache for all the events load from server end.*

### Config Options

### Function

#### **getWholeList:function(cview, day, all, single)**

*For get all the whole events on this day*

##### **Parameters:**

- {Class} **cview**:  
an instance of DayView or MonthView;
- {string} **day**: the day in "Y-m-d" format
- {boolean} **all**: true for contain both shown and hidden events, or only contain shown events
- {boolean} **single**: true means get list for a single day, false for get list for a week. It's different, because there may have some events cross a week

#### **updateWholeList: function(arr, action)**

*For add/update/delete some events in whole events list*

##### **Parameters:**

- {Array} **arr**: a list of events;
- {string} **action**: "add", "update" or "delete"

#### **getLayout:function(day, cview, eventList, newFlag, refresh)**

*Get the grid layout object for a day, and layout the eventlist passed*

##### **Parameters:**

- {String} **day**: the day string in "Y-m-d" format
- {Class} **cview**: an instance of DayView or MonthView
- {Array} **eventList**: an array of events need be layouted
- {boolean} **newFlag**: true for create a new layout if it's null
- {boolean} **refresh**: true for create a new layout no matter it's exist or not

### Class Ext.ux.calendar.BackThread

<b>Package</b>	Ext.ux.calendar
<b>Defined in</b>	BackThread.js
<b>Extends</b>	Ext.util.Observable

*This class is a thread which will check the activated event for alert by every minute and update the timeline in DayView and WeekView.*

#### Config Options

#### Function

## Class Ext.ux.calendar.DayView

<b>Package</b>	Ext.ux.calendar
<b>Defined in</b>	DayView.js
<b>Extends</b>	Ext.Panel

*This class is for render DayView and WeekView.*

### Config Options

#### **dayNum: int**

define the day number in this view, 1 for DayView, 7 for WeekView;

### Function

#### **updateTimeline: function()**

*For update the timeline*

#### **cleanup:function(col, pool)**

*For update the timeline*

{int}**col**: the index of the column you want to clean

{boolean}**pool**:true for also clean the column of pool, or only clean the view

#### **checkLayout:function(force, refresh)**

*For add/update/delete some events in whole events list*

#### **Parameters:**

{boolean} **force**: true for force relayouting events of all days in this view, or only relayout the day changed;

{boolean} **refresh**: true for requesting server to reload events

## Class Ext.ux.calendar.MonthView

<b>Package</b>	Ext.ux.calendar
<b>Defined in</b>	MonthView.js
<b>Extends</b>	Ext.Panel

*This class is for render MonthView.*

### Config Options

### Function

#### **cleanup:function(w)**

*For update the timeline*

{int}**w**: the index of the row you want to clean

#### **checkLayout:function(force, refresh)**

*For add/update/delete some events in whole events list*

##### **Parameters:**

{boolean} **force**: true for force relayouting events of all days in this view, or only relayout the day changed;

{boolean} **refresh**: true for requesting server to reload events

## Class Ext.ux.calendar.CalendarContainer

<b>Package</b>	Ext.ux.calendar
<b>Defined in</b>	CalendarContainer.js
<b>Extends</b>	Ext.Panel

*This class is a container for DayView, WeekView and MonthView.*

### Config Options

### Function

## Note about myCalendar

- myCalendar is an extjs-based calendar widget for visualizing time-based events. It is capable of rendering dynamic, calendar-style views of event data from JSON data. MyCalendar does not tight to any specific server for its data backend. You can use Grails, PHP, .NET to create JSON data and feed to MyCalendar. In here, we provide two backend version (Grails and PHP) as demo for your better understanding.
- myCalendar is tested in ext3.0.
- myCalendar is tested in IE7, FF3.0 and Chrome browsers.



# History Log

1. **Version 1.0.0, released at August 22, 2009**
  - First release
2. **Version 1.0.1, released at August 29, 2009**
  - Fixed: error when click on the date picker: p.fireFn is undefined
3. **Version 1.0.2, released at September 29, 2009**
  - Set default appear hour start at 9 am (scroll bar down a little bit).
  - Change code to make it do not need request backend everytime when change days. The events of visited days will save in js side, so this will decrease the pressure of server.
  - Single click and drag to create new event - like google calendar did.
4. **Version 1.1.0, released at October 17, 2009**
  - Changed some CSS issues and Fixed several bugs;
  - Make myCalendar work with Extjs2.2
  - Change the hide/show calendar function to make it faster when there is a lot of events in month view;
  - Add more options in the setting window;
  - Add lock/unlock functions in the context menu of events, which change the db table fields too;
  - Add Multi-language setting in the tab More menu.
5. **Version 2.0, released at November 28, 2009**
  - Changed some CSS issues and fixed several bugs;
  - Add cross-day event function;
  - Add repeat event function;
  - Add 7 more settings as below,
    - activeStartTime:define the active start time of a day, example: '09:00'
    - activeEndTime:define the active end time of a day, example: '19:00'
    - hideInactiveRow:true for hide the inactive time, or show it at different css
    - intervalSlot:the interval of a row in dayview/weekview, for example: '30' means 30 minutes
    - startDay:define the week start at monday or sunday, 1 for monday, 0 for sunday
    - singleDay:true for disable cross-day event function
    - createByDbclick:true for creat evnet by double click, notice when you set it true, the drag create function won't work
  - For more details about these fields, you can see [README/php/myCalendar2.0-tables](#) for detail;
  - Change the old event editor, and add DetailEditor;
  - Add week num in monthview;
  - Add refresh function
    - now if you want to clean and reload all calendar and events from DB, you just call 'CalendarContainer.fireEvent("refresh")';

- Change the DatePicker component to show the current date range.