

Apertura scheda gwCalendar

La gwCalendar è una particolare tipologia di scheda descritta [qui](#).

openGwCalendarTab()

Per aprire questa tipologia di tab è esposta l'apposita function `openGwCalendarTab()`.

Parametri

- **title** String, optional
- **params** Object, required
 - **calendarCodes** String, array: comma separated values
 - **eventName** String, array: comma separated values. parallel array to **calendarCodes** (each n position refers to nth calendarCodes's one)
 - **filters** Object or Object[], optional *removable* filters. Inside filters `columnName` property are available some predefined placeholders: '#id_event#', '#cod_event#', '#summary#', '#place#', '#start_date#', '#end_date#', '#notes#', '#notes#', '#cod_calendar#', '#cod_event_status#', '#cod_priority#', '#cod_event_type#'. This can be used to make filters shared with different gwClasses implementing the event interface (each one that may overrides some field through a specific mapping). Simple filter based on a specific (=gwClass related) `columnName`, for example `evt_id_event` instead of the default `id_event` still will works, but gwClass cross compatibility may not be granted. Ex:

```
{condition: 'AND', columnName: '#cod_event_status#', operator: '=', filterType: 'STRING', value: 'CLO'}
```

- **staticFilters** Object or Object[], optional *fixed* filters. Inside filters `columnName` property are available some predefined placeholders: '#id_event#', '#cod_event#', '#summary#', '#place#', '#start_date#', '#end_date#', '#notes#', '#notes#', '#cod_calendar#', '#cod_event_status#', '#cod_priority#', '#cod_event_type#'. This can be used to make filters shared with different gwClasses implementing the event interface (each one that may overrides some field through a specific mapping). Simple filter based on a specific (=gwClass related) `columnName`, for example `evt_id_event` instead of the default `id_event` still will works, but gwClass cross compatibility may not be granted. Ex:

```
{condition: 'AND', columnName: '#cod_event_type#', operator: '=', filterType: 'STRING', value: 'ETP3'}
```

- **minHours** Integer, optional, in range [0, 23], 8 default. Min hours shown is day view
- **maxHours** Integer, optional, in range [1, 36], 16 default. Max hours shown is day view
- **date** Date, optional: ex: value="new Date(2015, 26, 3)". If not set today is used. new Date() takes 3 parameters, in order: year, month, day
- **dateInterval** String, optional, default 'week'. possible options: 'day', 'week', 'month'
- **dateIntervalSteps** Integer, optional, default 5. n days from the reference date

Gli ultimi 5 parametri sono ripresi direttamente fra quelli esposti da dojo, ed influiscono sulla grafica

della scheda Calendario. Sono opzionali, e in caso di omissione sono utilizzati dei valori di default. Essi agiscono come descritto nella documentazione ufficiale visualizzabile [qui](#).

- **insertIndex** Integer, opzionale. Determina in quale posizione verrà aperta la scheda. Se omesso la scheda verrà aperta a destra, in ultima posizione

Esempi

Codice minimale.

[example.js](#)

```
var title = 'Calendar Events';
var params = {
  calendarCodes: 'CAL1,CAL2', //array: comma separated values
  eventClassName: 'class_name_1,class_name_2' //array: comma
separated values. parallel array to calendarCodes
};
var gwCalendarTab = openGwCalendarTab(title, params);
```

Esempio con utilizzo di tutti i parametri, anche opzionali

[example.js](#)

```
var title = 'Title';
var tabWidgetSuffixId = 'cal_name'; //Optional, Used to eventually
reuse an already opened tab. if omitted a random one is generated, and
a new tab is opened
var title = 'Calendar Events';
var params = {
  calendarCodes: 'CAL1,CAL2', //array: comma separated values
  eventClassName: 'class_name_1,class_name_2', //array: comma
separated values. parallel array to calendarCodes
  filters: {condition: 'AND', columnName: '#cod_event_status#',
operator: '=', filterType: 'STRING', value: 'CLO'},
  staticFilters: {condition: 'AND', columnName: '#cod_event_type#',
operator: '=', filterType: 'STRING', value: 'ETP3'},
  minHours: 0, //Integer, optional, default 0. min hours shown is day
view
  maxHours: 23, //Integer, optional, default 23. Max hours shown is
day view
  date: null, // Optional: ex: value="new Date(2015, 26, 3)". If not
set today is used. new Date() takes 3 parameters, in order: year,
month, day
  dateInterval: 'week', //String, optional, default 'week'. possible
options: 'day', 'week', 'month'
  dateIntervalSteps: 5 //Integer, optional, default 5. n days from
```

```

the reference date
};
var gwCalendarTab = openGwCalendarTab(title, params);

```

Esempio apertura calendario con filtri presi da lista classe filtrata, aprendolo sulla prima data utile:
<code javascript example.js>

Available incoming parameters: In the linkList grid parameter has been added: start class id and name, start class key column and key column type, *target class name*, *target class attribute id* and *target class key column*. /*String*/ gwClassName, /*Object*/ queryParameter, /*Object*/ grid, /*Object*/ detailContainerId debugger; var filters = [] if(grid.query && grid.query.query){ filters = grid.query.query.filters || []; if(grid.query.query.staticFilters) filters = [filters, ...grid.query.query.staticFilters]; TODO MERGE STATICFILTERS BETTER } var sort = null; var callback = function(items){

```

    var openingDate = items.length!=null ? new Date(items[0].evt_start_date) :
new Date();
    var itemIdList = [];
    items.forEach(function(item){
        itemIdList.push(item[grid.store.objectStore.idProperty]);
        if(new Date(item.evt_start_date) < openingDate) openingDate = new
Date(item.evt_start_date);
    });
    var filters = [{condition: 'AND', columnName: '#id_event#', operator:
'IN', filterType: 'INTEGER', value: itemIdList}];
var title = 'Calendario Eventi';
    var tabWidgetSuffixId = gwClassName+'_filtered'; //Optional, Used to
eventually reuse an already opened tab. if omitted a random one is
generated, and a new tab is opened
    var params = {
        calendarCodes: 'PEM_ASS,PEM_BUI,PEM_IMP,PEM_SPA', //array: comma
separated values
        eventClassName:
gwClassName+', '+gwClassName+', '+gwClassName+', '+gwClassName, //array: comma
separated values. parallel array to calendarCodes
        filters: filters,
        staticFilters: [],
        minHours: 0, //Integer, optional, default 0. min hours shown is day
view
        maxHours: 23, //Integer, optional, default 23. Max hours shown is day
view
        date: openingDate, // Optional: ex: value="new Date(2015, 26, 3)". If
not set today is used. new Date() takes 3 parameters, in order: year, month,
day
        dateInterval: 'month', //String, optional, default 'week'. possible
options: 'day', 'week', 'month'
        dateIntervalSteps: null //Integer, optional, default 5. n days from
the reference date
    };

```

```
var gwCalendarTab = openGwCalendarTab(title, params);
```

```
}; selectGwClassRecords(gwClassName, filters, sort, callback); 4.5.* and 4.4.*  
selectGwClassRecordsRaw(gwClassName, filters, sort, callback); from 4.6.8
```

From: <https://wiki.geowebframework.com/> - **GeowebFramework**

Permanent link: https://wiki.geowebframework.com/doku.php?id=custom:api_js_opengwcalendartab&rev=1730734282

Last update: **2024/11/04 16:31**

