

Utilizzo delle API Javascript

Introduzione

Apertura dettaglio classe (gwClassDetail)

Di seguito una lista di funzioni javascript usabili per aprire le varie tipologie di dettaglio della classe:

- su **floatingPane** (dimensione modificabile dall'utente, non modale se non diversamente specificato)
- su **dialog** (dimensioni fisse, modale)
- su **tab**

Ogni funzione ha i primi due parametri, *itemId* e *gwClassName*, fissi ed obbligatori, mentre il terzo, *options*, é opzionale e serve per poter customizzare l'apertura del dettaglio della classe.

Parametri Comuni

Opzioni disponibili, valide per tutte le function `openGwClassDetail*()`:

- **detailContainerId** String, if present override autogenerated detailContainerId
- **detailLayoutName** String, if present gwClassDetail is opened with the specified detailLayout
- **editIgnorePermissions** Boolean, default false (**already @Deprecated, @Removed from version 4.7.1**)
- **visualizeIgnorePermissions** Boolean, default false
- **useCookie** Boolean, default true. Usually set false, in combination with a specific dimensions, in aim to avoid that dimensions will be saved on cookie and thereafter applied to all gwClassDetail of the same GwClass opened by user in a project
- **dimensions** Object, default { w: 800, h: 600 }. Parameters w, h, respectively width and height. Are available x, y parameters too, that design the offset with origin the top*left corner. If x, y are omitted are automatically computed in aim to center dialog in browser window. Ex: { w: 800, h: 600, x: 200, y: 200 }
- **autoTitle** Boolean, default true. When false, the parameter title is evaluated
- **title** String, is evaluated only if autoTitle == false
- **hideTopToolbar** Boolean, default false. When true the top toolbar is not shown
- **hideBottomToolbar** Boolean, default false. When true, the bottom toolbar is not shown
- **hideDefaultButtons** Boolean, default false. When true, the bottom toolbar default buttons are not shown
- **createLabel** String, if present override default one
- **createIconClass** String, if present override default one
- **createAndCloseLabel** String, if present override default one
- **createAndCloseIconClass** String, if present override default one
- **saveLabel** String, if present override default one
- **saveIconClass** String, if present override default one
- **saveAndCloseLabel** String, if present override default one
- **saveAndCloseIconClass** String, if present override default one

- **extraBottomToolBarButtons** Object[], default null. List of buttons definitions. Buttons are added in the bottom toolbar. A single button definition has this format { label: 'button_label', iconClass: 'button_icon_class', functionCode: function(evt, itemId, gwClassName, detailContainerId, targetId){ doSomething()* } }
- **attributesOverrisionsMap** Object, default null. Object done like that { attribute_name_1: { readonly: true, required: false, hidden: false }, attribute_name_2: {...} }
- **initialsValuesMap** Object, default null. Object done like that { attribute_name_1: value1, attribute_name_2: value2, ..}

Deprecati:

- **targetId** String, @Deprecated, id of a grid that needs to be notified about gwClass Item modifications

FLOATING PANE

```
function openGwClassDetailFloatingPane(
/*String*/ itemId,
/*String*/ gwClassName,
/*object*/ options
){
    ...
}
```

Opzioni specifiche per la function openGwClassDetailFloatingPane():

- **modal** Boolean, default false

Esempi:

```
//Apertura dettaglio del record 12345 della classe 'gw_class_name' usando
uno specifico
//layout, di nome custom_detail_layout, forzando il readonly e nascondendo
la top toolbar
var itemId = 12345;
var gwClassName = 'gw_class_name';
var options = {
    detailLayoutName: 'custom_detail_layout',
    visualizeIgnorePermissions: true,
    hideTopToolBar: true
};
openGwClassDetailFloatingPane(itemId, gwClassName, options);
```

```
//Apertura dettaglio del record 12345 della classe 'gw_class_name', usando
gli attributi
//specificati in forcedAttributesNamesList, sovrascrivendone le proprietà
come imposto in
//attributesOverrisionsMap, imponendo dimensioni ad hoc, usando useCookie a
false per far
//si che le dimensions non siano salvate su cookie ed applicate a tutti i
```

```

gwClassDetail
//aperti dall'utente per quella classe nel progetto, nascondendo la top
toolbar,
//nascondendo i bottoni di default e fornendo dei bottoni custom come
specificato
//nell'array extraBottomToolbarButtons (qui sono ricreati il tasto 'chiudi'
e
//'salva e chiudi', con una callback per aggiungere codice - al posto di
TODO - al
//comportamento base)
var itemId = 12345;
var gwClassName = 'gw_class_name';
var options = {
  attributesOverrissionsMap: {
    attribute_name_1: { readonly: true, required: false, hidden:
false},
    attribute_name_2: { readonly: false, required: true, hidden:
false},
    attribute_name_3: { readonly: false, required: true, hidden: false}
  },
  forcedAttributesNamesList:
['attribute_name_1', 'attribute_name_2', 'attribute_name_3'],
  dimensions: { w: 400, h: 200 },
  useCookie: false,
  hideTopToolbar: true,
  //editIgnorePermissions: true, //(already @Deprecated, @Removed from
version 4.7.1)
  hideDefaultButtons: true,
  extraBottomToolbarButtons: [
    {
      label: closeLabel,
      iconClass: 'iconClose',
      functionCode: function(
        evt, itemId, gwClassName, detailContainerId, targetId
      ){
        onDetailCloseButtonClick(detailContainerId, this);
      }
    },
    {
      label: saveAndCloseLabel,
      iconClass: 'iconSaveAndClose',
      functionCode: function(
        evt, itemId, gwClassName, detailContainerId, targetId
      ){
        var detailContainerWidget =
dijit.registry.byId(detailContainerId);
        detailContainerWidget._hadToBeClosed = true;
        var callback = function(
          /*Object*/ params
          /*className, itemId, sqlStatement, detailContainerId,
sendFormDomNode,

```

```

        formsValuesGwidKeyed, formsValuesNameKeyed,
formsValuesToShowGwidKeyed,
        formsValuesToShowNameKeyed, allFormsValuesGwidKeyed,
        allFormsValuesNameKeyed, allFormsValuesToShowGwidKeyed,
        allFormsValuesToShowNameKeyed*/
    ){
        //TODO aggiungi qui codice js da eseguire al termine
del salvataggio
    };
    var params = {
        className: gwClassName,
        sqlStatement: 'update',
        detailContainerId: detailContainerId,
        buttonWidget: this,
        callback: callback
    };
    sendGwClassDetailForm(params);
    }
}
]
}
}
openGwClassDetailFloatingPane(itemId, gwClassName, options);

```

DIALOG

```

function openGwClassDetailDialog(
    /*String*/ itemId,
    /*String*/ gwClassName,
    /*object*/ options
){
    ...
}

```

TAB (GEOMANAGER)

```

function openGwClassDetailTab(
    /*String*/ itemId,
    /*String*/ gwClassName,
    /*object*/ options
){
    ...
}

```

Opzioni specifiche per openGwClassDetailTab() in GeoManager:

- **insertIndex** Integer, default null (tab added in last position), if present force the tab insert index - forceRefreshTab Boolean, default false

- **forceReplaceTab** Boolean, default false}

TAB (GEOEXPLORER)

```
function openGwClassDetailTab(  
    /*String*/ itemId,  
    /*String*/ gwClassName,  
    /*object*/ options  
) {  
    ...  
}
```

From:

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

Permanent link:

<https://wiki.geowebframework.com/doku.php?id=custom:apiclient>

Last update: **2024/07/25 18:19**

