

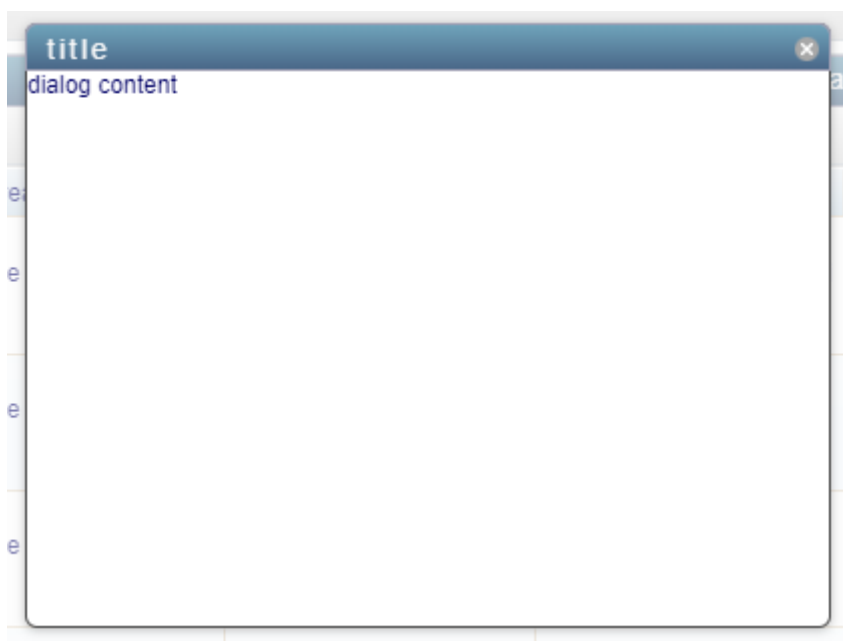
# Apertura Dialog Generico

Un Dialog è una particolare finestra di dialogo **modale** di Geoweb, basato su un componente Dojo, che ha subito un adeguamento di stile. Può essere aperto invocando la generica function `openDialog()` con il corretto set di argomenti. Essa può essere usata direttamente, ma è anche invocata internamente a tutte le function semplificate che aprono schede di tipi specifici usando un minore set di parametri o settandone alcuni default (es: `openGwClassDetailDialog()`).

Si possono aprire più di un dialog su chiamate successive, senza chiudere i precedenti.

Un dialog, in quanto modale, viene sempre aperto in cima a qualsiasi altro componente del framework, compresi eventuali `floatingPane` e dialog precedentemente aperti.

La funzione restituisce il widget dojo che di fatto rappresenta il dialog aperto, e che può esporre tutte le API generali (come `gwRefresh()`, `gwReload()`, etc.), se implementate, e API specifiche, che dipendono di volta in volta dalla tipologia di del contenuto del dialog (per esempio le API js di dettaglio di classe).



## Parametri

- **dialogId** String, required. E' un identificativo univoco per il dialog che si vuole aprire. Se usato in una successiva invocazione della function verrà usato per riutilizzare il dialog già aperto (che non verrà riportato in primo piano)
- **href** String, optional, default null. Mutualmente esclusivo con `content` (`href` ha la priorità). URL che verrà invocato e la cui risposta testuale sarà utilizzata come `content` del `floatingPane`
- **preventCache** Boolean, optional, default true. Fa sì che la richiesta all'url di `href` sia univoca e non restituita dalla cache del browser. Valutato solo se `href` è valorizzato
- **sync** Boolean, optional, default false. Fa sì che la richiesta all'url di `href` sia eseguita in modo sincrono. Valutato solo se `href` è valorizzato
- **forceRefresh** Boolean, optional, default false. Fa sì che l'eventuale `floatingPane` già aperto, in caso fosse stato aperto utilizzando `href`, effettui il refresh del suo contenuto rieseguendo la

chiamata XHR. Valutato solo se href è valorizzato

- **method** String, optional, default 'GET'. Valori ammessi: 'GET', 'POST'. Metodo con cui viene eseguita la richiesta di href. Valutato solo se href è valorizzato
- **postData** Object, optional, default null. Oggetto mandato con la richiesta post. Serializzato in JSON (Content-Type: 'application/json'). Valutato solo se href è valorizzato, e se method='POST'
- **content** String, Node or Object, optional, default null. Mutualmente esclusivo con href. Contenuto che verrà direttamente impostato nella proprietà 'content' del floatingPane. In caso di String, verrà interpretato come codice html. In caso la Stringa contenga codice dojo in versione dichiarativa, ne verrà contestualmente effettuato il parse al fine di instanziare i widget ivi contenuti.
- **useCookie** Boolean, optional, default false. Se posto a true un cookie verrà usato per persistere le ultime dimensioni e posizione imposte dall'utente per il floatingPane con stesso id. In caso di cookie già esistente esso verrà inoltre usato in fase di apertura per il posizionamento del floatingPane
- **title** String, optional, default null. L'intestazione che verrà mostrata sulla scheda. Alcuni tipi di scheda, come le gwClassDetail/gwClassList in caso di assenza potrebbero settarsi in automatico un titolo significativo di default.
- **dimensions** Object, optional, default {w: 800, h: 600}. Oggetto nella forma {x: 100, y: 100, w: 500, h: 300}, con x,y opzionali (se omessi vengono generati dinamicamente facendo sì che il floatingPane risulti centrato orizzontalmente e verticalmente).
- **callback** function, optional, default null. function che verrà eseguita alla fine della chiamata XHR che restituisce il content restituito da href
- **callbackParams** Object[], optional, default null. Array di parametri con cui verrà invocata l'eventuale callback configurata
- **props** Object, optional, default null. Object contenente un numero arbitrario di proprietà che verranno passate al costruttore del componente customDojo/CustomDojoDialog derivato da dojo. Quindi ne condizionano di potenzialmente il comportamento. Si possono forzare valori a tutti i parametri di configurazione o sovrascrivere il comportamento delle function interne al dialog (A tale proposito si vedano i parametri esposti dalle API dojo wui <https://dojotoolkit.org/api/?qs=1.9/dojo/main.cldr>, inserire dojox/layout/Dialog nella funzione cerca) I valori qui contenuti saranno di fatto anche ospitati e resi accessibili dalla istanza dialog restituita

## Note

Ad ogni dialog viene assegnato, in fase di prima apertura, un id univoco (dialogId) (dojo si assicura con non possa essere usato più volte). Esso è rilevante, in quanto, in caso di successive invocazioni della function openDialog(), con lo stesso dialogId, Geoweb in caso trovasse già aperto un dialog con lo stesso identificativo, lo 'riuserebbe' aggiornando semplicemente titolo, href/content. Da notare che se il dialog non è già in cima alla eventuale pila di dialog, esso non viene riportato in cima.

E' buona prassi generare dialogId utilizzando sempre le stesse metodologie per aprire tipi di contenuto simili.

## Esempi Apertura

codice minimale href GET

```
var params = {
  href: gwContextPath+'/rest/myURL?parma1=1&param2=two'
};
var dialog = openDialog(params);
```

Codice minimale href POST:

```
var params = {
  href: gwContextPath+'/rest/myURL?parma1=1&param2=two',
  method: 'POST',
  postData: {
    param3: 3,
    param4: 'four'
  },
};
var dialog = openDialog(params);
```

Codice minimale content String:

```
var params = {
  content: '<div style="color: red;">Red Content</div>'
};
var dialog = openDialog(params);
```

Codice minimale content Object widget dojo:

```
var content = new dijit.layout.ContentPane({href:
gwContextPath+'/rest/myURL?parma1=1&param2=two', dummyProp: 'dummyVlaue'});
content.startup();

var params = {
  content: content
};
var dialog = openDialog(params);
```

Codice href GET con parametri inizializzati ai default:

```
var params = {
  floatingPaneId: 'floatingPaneId',
  href: gwContextPath+'/rest/myURL?parma1=1&param2=two',
  preventCache: true,
  sync: false,
  forceRefresh: false,
  method: 'GET',
  useCookie: false,
  title: 'title',
  dimensions: {w: 800, h: 600},
  callback: null,
  callbackParams: [],
  props: {}
};
```

```
};  
var dialog = openDialog(params);
```

Codice href versione POST con parametri inizializzati ai default:

```
var params = {  
  floatingPaneId: 'floatingPaneId',  
  href: gwContextPath+'/rest/myURL?parma1=1&param2=two',  
  preventCache: true,  
  sync: false,  
  forceRefresh: false,  
  method: 'POST',  
  postData: {  
    param3: 3,  
    param4: 'four'  
  },  
  useCookie: false,  
  title: 'title',  
  dimensions: {w: 800, h: 600},  
  callback: null,  
  callbackParams: [],  
  props: {}  
};  
var dialog = openDialog(params);
```

## Chiusura Dialog

```
var dialog = openDialog();  
  
..  
  
//close the dialog  
dialog.hide();
```

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

Permanent link:  
[https://wiki.geowebframework.com/doku.php?id=custom:api\\_js\\_open\\_dialog](https://wiki.geowebframework.com/doku.php?id=custom:api_js_open_dialog)

Last update: **2019/11/06 09:42**

