

Componenti di interfaccia

Menù

I, II e III livello

Componenti applicativi di menu (hierarchical, Integration, ecc.)

Elementi del menù (LeafItem)

Tipi di LeafItem

Layout GeoExplorer - Struttura Generale XML di Progetto

Il layout grafico di un progetto si ottiene attraverso la configurazione del relativo XML la cui struttura è data dalla combinazione di una serie di tag:

`<geoexplorer>` `</geoexplorer>` configurazione di progetto
`<menuBar>` `</menuBar>` configurazione della barra dei menù
`<menuItem>` `</menuItem>` configurazione del singolo menù di I° livello
`<accordionPanelItem>` `</accordionPanelItem>` configurazione del singolo menù di II° livello
`<leafitem>` `</leafitem>` configurazione della singola voce del menù di II° livello

Per ogni tag devono in genere essere configurati i seguenti parametri:

name nome (univoco) da assegnare al tag

label etichetta da visualizzare nel layout di rappresentazione

image nome del file dell'immagine da visualizzare nel layout di rappresentazione (../WEB/images)

```
<geoexplorer>
  <menuBar>
    <menuItem name="menu name" label="menu label" image="menu
image.png">
      <accordionPanelItem name="accordion name" label="accordion label"
image="accordion image.png"
      type="leafItemContainer">
        <leafitem name="leafitem name" label="leafitem label"
image="leafitem image.png" type="gwClassList">
          <parameter name="className" value="class_name"
hideToClient="false"></parameter>
        </leafitem>
      </accordionPanelItem>
    </menuItem>
  </menuBar>
```

```
</geoexplorer>
```

Layout GeoManager - Struttura Generale XML di Progetto

Il layout grafico di un progetto si ottiene attraverso la configurazione del relativo XML la cui struttura è data dalla combinazione di una serie di tag:

```
<geoexplorer> </geoexplorer> configurazione di progetto  
<menuBar> </menuBar> configurazione della barra dei menù  
<menuItem> </menuItem> configurazione del singolo menù di I° livello  
<accordionPanelItem> </accordionPanelItem> configurazione del singolo menù di II° livello  
<leafitem> </leafitem> configurazione della singola voce del menù di II° livello
```

Per ogni tag devono in genere essere configurati i seguenti parametri:

name nome (univoco) da assegnare al tag

label etichetta da visualizzare nel layout di rappresentazione

image nome del file dell'immagine da visualizzare nel layout di rappresentazione (../WEB/images)

```
<geomanager>  
  <menuBar>  
    <menuItem name="menu name" label="menu label" image="menu  
image.png">  
      <accordionPaneItem name="accordion name" label="accordion label"  
image="accordion image.png" type="leafItemContainer">  
        <leafitem name="leafitem name" label="leafitem label"  
image="leafitem image.png" type="gwClassList">  
          <parameter name="className" value="class_name"  
hideToClient="false"></parameter>  
        </leafitem>  
      </accordionPaneItem>  
    </menuItem>  
  </menuBar>  
</geomanager>
```

Accordion e LeafItem

Accordion

L'accordion è il controllo utilizzato per 'contenere' il menù di secondo livello. Ad ogni voce di menù di primo livello possono corrispondere uno o più accordion. All'apertura solo il primo accordion verrà visualizzato aperto, mentre gli altri vengono visualizzati, chiusi in basso. Cliccando sulla intestazione dell'accordion, esso verrà aperto chiudendo a sua volta quello aperto precedentemente. L'accordion può essere un semplice contenitore, oppure può essere strutturato esso stesso come un Menu di accesso ad altri oggetti del progetto.

In particolare, in base all'attributo 'type' esso può essere:

leafItemContainer tipo semplice, ovvero aperto a contenere altri oggetti (leafItem) di diversa natura

integrationMenu visualizzazione di tutti i Temi/Classi (visibili) associati ad un progetto
layoutSwitchMenu visualizzazione strutturata di tutte le planimetrie caricate nel progetto (*)

Gli attributi dell'accordion sono: name, label, image, type.

LeafItem

Il leaf-item rappresenta la singola voce di un menù di secondo livello e può richiamare la visualizzazione di diverse tipologie di «schede»:

- lista record di una classe di Geoweb®
- gestione processo (work-flow)
- cruscotto di reportistica / report html
- cruscotto degli indicatori
- calendario
- mappa cartografica
- albero di navigazione dei layout
- albero di navigazione delle gerarchie

Ecco una lista di parametri comuni

- name: String, default null, obbligatorio. Deve essere univoco fra tutti i name dei leafItem all'interno dello stesso menu di secondo livello (AccordionPanelItem), può essere usato, insieme ai name di menu di primo e di secondo livello, per aprire il progetto settato su una specifica voce di menu.
- label: (String, default ' ' opzionale). Etichetta che compare sul menu di terzo livello
- image, (String, default null, opzionale). Rappresenta il path, comprensivo di estensione, che individua l'immagine che verrà applicata al menu del terzo livello. (a partire dalla cartella 'WEB/' dentro i contenuti statici). L'immagine sarà sempre ridimensionata a 32x32 px, e se assente un analogo spazio vuoto verrà lasciato per preservare l'allineamento delle label di più leafItem in sequenza. Ex: path/to/image.png
- type: String, default null, obbligatorio. Identifica la tipologia di scheda che verrà aperta al click
- disabled: Boolean, default false, opzionale. Quando posto a true, il leafItem viene mostrato disabilitato. Viene automaticamente posto a true, dove applicabile, per i leafItem senza permessi per l'utente corrente in quell'ambito (type: gwClassList, gwClassDetail, gwAction, etc..). Può essere esplicitamente settato a true da un utente configuratore per creare prototipi di strutture di menu con funzioni non ancora implementate.
- hidden: Boolean, default false, opzionale. Quando posto a true, il leafItem non viene mostrato. Viene automaticamente posto a true, dove applicabile, per i leafItem senza permessi per l'utente corrente in quell'ambito quando hideWhenWithoutPermissions vale true (vedi sotto) (type: gwClassList, gwClassDetail, gwAction, etc..).
- hideWhenWithoutPermissions: Boolean, default false, opzionale. Quando posto a true, il leafItem viene automaticamente nascosto in assenza di permessi, dove applicabile (type: gwClassList, gwClassDetail, gwAction, etc..). Ciò è utile per creare interfacce di progetto che cambiano dinamicamente in base al profilo utente.

Si riporta di seguito l'elenco dei leaf-item disponibili:

- [gwClassList](#) visualizzazione dei record di una classe di Geoweb in una lista «standard»
- [leafItemListFilters](#) visualizzazione dei record di una classe di Geoweb in una lista «standard» e in liste parzializzate secondo i valori assunti da uno specifico attributo

- [gwPrefilteredList](#) visualizzazione dei record di una classe di Geoweb in una lista «prefiltrata» attraverso l'applicazione di una clausola «where»
- [leafItemClassificationMenu](#) visualizzazione dei record di una classe di Geoweb «classificata» organizzati all'interno dell'albero di classificazione
- [gwClassProcesses](#) visualizzazione della pagina di gestione del processo (work-flow) associato ad una classe di Geoweb
- [gwReportsSheet](#) visualizzazione della pagina di cruscotto delle report (pdf, xls, ecc.)
- [gwHtmlReport](#) visualizzazione di una report in formato html
- [gwIndicatorsSheet](#) visualizzazione della pagina di cruscotto degli indicatori
- [gwCalendar](#) visualizzazione del «calendario»
- [gwMap](#) visualizzazione della «mappa cartografica»
- [gwAction](#) esegue il codice di un'azione (es. apertura scheda di dettaglio)
- [leafItemHierarchicalFilterMenu](#) visualizzazione albero di navigazione per una gerarchia di classi
- [leafItemLayoutChange](#) visualizzazione albero di navigazione planimetrie
- [leafItemSeparator](#) aggiunge un separatore orizzontale fra due leafItem

XML leaf-item gwClassList

Utilizzato per la visualizzazione dei record di una classe di Geoweb in una lista «standard», presenta il seguente XML di configurazione (che dovrà essere incluso all'interno di un tag di tipo «accordion».

```
<leafItem name="leafitem name" label="leafitem label"  
image="leafitem_image.png" type="gwClassList">  
  <parameter name="className" value="class_name"  
hideToClient="false"></parameter>  
</leafItem>
```

```
* leafitem name = nome del leaf item  
* label = etichetta del leaf-item visualizzata nel menù di II° livello  
* image = nome dell'immagine visualizzata nel menù di II° livello  
(../WEB/images)  
* className = nome della classe di Geoweb
```

```
<leafItem name="lf_risorse_umane" label="Risorse Umane"  
image="img_risorse_umane.png" type="gwClassList">  
  <parameter name="className" value="hr_risorse_umane"  
hideToClient="false"></parameter>  
</leafItem>
```

XML leaf-item leafItemListFilters

Utilizzato per la visualizzazione dei record di una classe di Geoweb in una lista «standard» o in liste parzializzate secondo i valori assunti da uno specifico attributo, presenta il seguente XML di configurazione (che dovrà essere incluso all'interno di un tag di tipo «accordion».

```
<leafItem name="leafitem name" label="leafitem label"  
image="leafitem_image.png" type="leafItemListFilters">
```

```

    <parameter name="className" value="class_name"
hideToClient="false"></parameter>
    <parameter name="attributeToFilter" value="field_name_attribute"
hideToClient="false"></parameter>
</leafItem>

```

- leafitem name = nome del leaf item
- label = etichetta del leaf item visualizzata nel menù di II° livello
- image = nome dell'immagine visualizzata nel menù di II° livello (../WEB/images)
- className = nome della classe di Geoweb
- attributeToFilter = nome del campo della tabella relativo all'attributo utilizzato per la parzializzazione

```

<leafItem name="lf_risorse_umane" label="Risorse Umane"
image="img_risorse_umane.png" type="leafItemListFilters">
    <parameter name="className" value="hr_risorse_umane"
hideToClient="false"></parameter>
    <parameter name="attributeToFilter" value="tipo_rapporto"
hideToClient="false"></parameter>
</leafItem>

```

XML leaf-item gwPrefilteredList

Utilizzato per la visualizzazione dei record di una classe di Geoweb pre-filtrati attraverso l'applicazione di una clausola «where», presenta il seguente XML di configurazione (che dovrà essere incluso all'interno di un tag di tipo «accordion».

```

<leafItem name= "leafitem_name" label= "leafitem_label"
image="leafitem_image.png" type="gwPrefilteredList">
    <parameter name="className" value="class_name"></parameter>
    <parameter name="MENU_TITOLO" value="title_value"></parameter>
    <parameter name="DETAIL_CLASS_NAME" value=""></parameter>
    <parameter name="QUERY" value="select field1,...,fieldN from table1_name
where fieldX='vsll1'"></parameter>
    <parameter name="OrderBy" value="field_orderby"></parameter>
    <parameter name="DETAIL_CLASS_NAME" value=""></parameter>
    <parameter name="PK_RECORD_FIELD" value="pk_class"></parameter>
    <parameter name="DATASOURCE" value="datasource_name"></parameter>
    <parameter name="PX_HIGHT" value=""></parameter>
    <parameter name="USER_GROUPS" value="group1,...,groupN"></parameter>
    <parameter name="ColumnsViews" value="field1,...,fieldN"></parameter>
    <parameter name="ColumnsLabel" value="label1,...,labelN"></parameter>
    <parameter name="ColumnsWidth" value="auto,...,auto"></parameter>
</leafItem>

```

- leafitem name = nome del leaf item
- label = etichetta del leaf item visualizzata nel menù di II° livello
- image = nome dell'immagine visualizzata nel menù di II° livello (../WEB/images)
- className = nome della classe di Geoweb
- MENU_TITOLO = valore da visualizzare come titolo della lista

- QUERY = query per la selezione dei record da visualizzare nella lista
- OrderBy = nome del campo per cui effettuare l'ordinamento della lista
- PK_RECORD_FIELD = nome del campo "chiave primaria"
- DATASOURCE = nome del datasource (schema dati)
- USER_GROUPS = elenco dei gruppi utenti che possono visualizzare la lista
- ColumnsViews = elenco dei campi da visualizzare come colonne della lista
- ColumnsLabel = elenco delle etichette da visualizzare come intestazione delle colonne della lista
- ColumnsWidth = larghezza (px) delle colonne della lista (default =auto)

(esempio)

```
<leafItem name="standard_comp" label="Standard della Compagnia"
image="doc_SC.png" type="gwPrefilteredList">
  <parameter name="className" value="DOCUMENTI"></parameter>
  <parameter name="MENU_TITOLO" value="Allegati - Documenti per Standard
della Compagnia"></parameter>
  <parameter name="DETAIL_CLASS_NAME" value=""></parameter>
  <parameter name="QUERY" value="select
id_documento,categ_documento,titolo_documento,data_redazione,url_download,st
ato_documento from view_documenti where
cod_categ_documento='SC' "></parameter>
  <parameter name="OrderBy" value=""></parameter>
  <parameter name="DETAIL_CLASS_NAME" value=""></parameter>
  <parameter name="PK_RECORD_FIELD" value="id_documento"></parameter>
  <parameter name="DATASOURCE" value="tte_dbdati_gw"></parameter>
  <parameter name="PX_HIGHT" value=""></parameter>
  <parameter name="USER_GROUPS" value="TTE,DBC"></parameter>
  <parameter name="ColumnsViews"
value="categ_documento,titolo_documento,data_redazione,stato_documento,url_d
ownload"></parameter>
  <parameter name="ColumnsLabel"
value="Categoria,Titolo,Redazione,Stato,Download"></parameter>
  <parameter name="ColumnsWidth"
value="auto,auto,auto,auto,auto"></parameter>
</leafItem>
```

XML leaf-item leafItemClassificationMenu

Utilizzato per la visualizzazione dei record di una classe di Geoweb (di tipo «classificato») organizzati in un albero di classificazione, presenta il seguente XML di configurazione (che dovrà essere incluso all'interno di un tag di tipo «accordion»).

```
<leafItem name="leafitem_name" label="leafitem_label"
image="leafitem_image.png" type="leafItemClassificationMenu">
  <parameter name="className" value="class_name"
hideToClient="false"></parameter>
  <parameter name="staticFilters"
value="{condition:'and',columnName:'attr_name',operator:'in',filterType:'str
```

```
ing',value:['val1,valn']}" hideToClient="false"></parameter>
  <parameter name="VISUALIZATION_GROUPS" value="groupV1,...,groupVn"
hideToClient="false"></parameter>
  <parameter name="MANAGER_GROUPS" value="groupM1,...,groupMn"
hideToClient="false"></parameter>
</leafItem>
```

XML leaf-item gwClassProcesses

Utilizzato per la visualizzazione della scheda di gestione di un processo associato ad una classe di Geoweb, presenta il seguente XML di configurazione (che dovrà essere incluso all'interno di un tag di tipo «accordion»).

```
<leafItem name="leafitem_name" label="leafitem_label"
image="leafitem_image.png" type="gwClassProcesses">
  <parameter name="className" value="class_name_gw"
hideToClient="false"></parameter>
  <parameter name="gwProcessName" value="process_name"
hideToClient="false"></parameter>
</leafItem>
```

XML leaf-item gwReportsSheet

Utilizzato per la visualizzazione della pagina di cruscotto delle report (formato PDF,XLS), presenta il seguente XML di configurazione (che dovrà essere incluso all'interno di un tag di tipo «accordion»).

```
<leafItem name="report_name" label="leafitem_label"
image="leafitem_image.png" type="gwReportsSheet">
  <parameter name="NAME" value="report_name"></parameter>
  <parameter name="VISUALIZATION_GROUPS"
value="groupV1,...,groupVn"></parameter>
  <parameter name="MANAGER_GROUPS"
value="groupM1,...,groupMn"></parameter>
</leafItem>
```

N.B. il valore assegnato a leafitem name deve coincidere con il valore del parametro NAME

XML leaf-item gwHtmlReport

Utilizzato per la visualizzazione di una report in formato html, presenta il seguente XML di configurazione (che dovrà essere incluso all'interno di un tag di tipo «accordion»).

```
<leafItem name="leafitem_name" label="leafitem_label"
image="leafitem_image.png" type="gwHtmlReport">
  <parameter name="reportUrl"
value="directory/nome_template_report.jasper"
hideToClient="false"></parameter>
```

```
</leafItem>
```

N.B. reportUrl: percorso a partire da «WEB/template/» dei contenuti statici

XML leaf-item gwIndicatorsSheet

Utilizzato per la visualizzazione della pagina di cruscotto degli indicatori, presenta il seguente XML di configurazione (che dovrà essere incluso all'interno di un tag di tipo «accordion»).

```
<leafItem name="leafitem_name" label="leafitem_label"
image="leafitem_image.png" type="gwIndicatorsSheet">
  <parameter name="NAME" value="Indicatori"></parameter>
  <parameter name="VISUALIZATION_GROUPS"
value="groupV1,...,groupVn"></parameter>
  <parameter name="MANAGER_GROUPS"
value="groupM1,...,groupMn"></parameter>
</leafItem>
```

XML leaf-item gwCalendar

Utilizzato per la visualizzazione della pagina del calendario, presenta il seguente XML di configurazione (che dovrà essere incluso all'interno di un tag di tipo «accordion»).

```
<leafItem name="leafitem_name" label="leafitem_label"
image="leafitem_image.png" type="gwCalendar">
  <parameter name="eventClassName" value="class_val1,...,class_valN"
hideToClient="false"></parameter>
  <parameter name="calendarCodes" value="val1,...,valN"
hideToClient="false"></parameter>
  <parameter name="minHours" value="0" hideToClient="false"></parameter>
  <parameter name="maxHours" value="23" hideToClient="false"></parameter>
  <parameter name="hourSize" value="60" hideToClient="false"></parameter>
  <parameter name="timeSlotDuration" value="15"
hideToClient="false"></parameter>
</leafItem>
```

XML leaf-item gwMap

Utilizzato per la visualizzazione di una mappa cartografica, presenta il seguente XML di configurazione (che dovrà essere incluso all'interno di un tag di tipo «accordion»).

```
<leafItem name="leafitem_name" label="leafitem_label"
image="leafitem_image.png" type="gwMap">
  <parameter name="mgMapName" value="mappa_name_mgos"
hideToClient="false"></parameter>
</leafItem>
```

XML leaf-item gwAction

Utilizzato per l'esecuzione del codice di una azione, presenta il seguente XML di configurazione (che dovrà essere incluso all'interno di un tag di tipo «accordion».

```
<leafItem name=" leafitem_name" label=" leafitem_label" image="
leafitem_image.png" type="gwAction">
  <parameter name="gwClassName" value="class_name"
hideToClient="false"></parameter>
  <parameter name="gwActionName" value="action_name"
hideToClient="false"></parameter>
</leafItem>
```

Attributi tag <leafItem> specifici:

- **autorun** Boolean, default, false. Quando il leafItem è il primo del primo *MenuDiSecondoLivello* del primo *MenuDiPrimoLivello*, con questo parametro a true, l'azione viene eseguita in automatico all'apertura del progetto
- **checkDynamicPermission** Boolean, default, false. (dalla **v 4.4.18**). Quando a true viene forzata la valutazione delle DACL anche al di fuori del contesto del gwRecord, come appunto è un leafItem. Da utilizzare previa specifica configurazione nel webadmin di FILTRI CUSTOM (o trasformati in CUSTOM automaticamente da Geoweb)

Esempio codice con autorun:

```
<leafItem name=" leafitem_name" label=" leafitem_label" image="
leafitem_image.png" type="gwAction" autorun="true">
  <parameter name="gwClassName" value="class_name"
hideToClient="false"></parameter>
  <parameter name="gwActionName" value="action_name"
hideToClient="false"></parameter>
</leafItem>
```

Esempio codice con checkDynamicPermission:

```
<leafItem name=" leafitem_name" label=" leafitem_label" image="
leafitem_image.png" type="gwAction" checkDynamicPermission="true">
  <parameter name="gwClassName" value="class_name"
hideToClient="false"></parameter>
  <parameter name="gwActionName" value="action_name"
hideToClient="false"></parameter>
</leafItem>
```

XML leaf-item leafItemHierarchicalFilterMenu

Permette di visualizzare un menu gerarchico che parzializza i record di una classe geoweb secondo criteri di filtro impostabili. La gerarchia non ha limiti nel numero di annidamenti, ed ogni livello può essere caratterizzato graficamente da una icona. La gerarchia è resa attraverso un widget albero dojo, ed i filtri vengono applicati in cascata man mano che si naviga l'albero partendo dal tronco ed

andando verso le foglie. I criteri di filtro, ordinati in ordine decrescente, si impostano tramite il parameter con name="filteringCriteria", e corrispondono al column_name degli attributi di geoweb costruiti sui campi della tabella che si vogliono usare come criteri di parzializzazione. In alto è presente una toolbar con un tasto refresh, che refresha l'albero ricordandosi i nodi già aperti. L'albero si refresha comunque in automatico ad ogni operazione di inserimento modifica cancellazione eseguito dalle liste della classe 'gwClassName' Il refresh non è automatico, invece, in seguito a modifiche su record delle classi effettuate tramite azioni/trigger groovy. (a meno di scrivere azioni js, che richiamano groovy, ad hoc)

Parametri :

- gwClassName: String, richiesto, nome della gwClass classificata
- filteringCriteria: String, richiesto, Imposta i criteri di filtro per ogni livello dell'albero. E' necessario almeno un criterio di filtro (piu' criteri di filtro vanno separati da virgola, e verranno applicati in ordine decrescente da sinistra verso destra)
- In filteringCriteria si puoi usare 'id_entity_classification' come criterio di filtro. Verra' quindi applicata l'intera gerarchia impostata dalla classificazione.
- iconImagePaths: String, opzionale, array parallelo a filteringCriteria dove vengono specificate le relative icone per ogni nodo generato dal criterio di filtro
- showAllAsRoot: Boolean, opzionale, default false, mostra anche un nodo root, riportante la label della classe, che se cliccato mostra la gwClassList della classe senza filtri
- rootLabel: String, opzionale, è la label che viene mostrata in caso di showAllAsRoot a true. In caso di showAllAsRoot a true, con rootLabel assente o stringa vuota viene usata la label della gwClass
- rootIconImagePath: opzionale, è l'icona relativa al nodo root. In caso di showAllAsRoot a true, con rootIconImagePath assente viene usata un'immagine di default cartellina.

Vincoli :

- filteringCriteria deve avere almeno un criterio di filtro
- iconImagePaths array parallelo a filteringCriteria
- Nel DB i valori dei campi degli attributi usati come criterio di parzializzazione non devono essere nulli, pena la non visualizzazione sull'albero gerarchico
- 'id_entity_classification' NON PUO essere il PRIMO criterio di filteringCriteria
- 'id_entity_classification' PUO non essere l'ULTIMO criterio di filteringCriteria. In tal caso ci saranno ulteriori livelli dell'albero parzializzati per gli attributi impostati

(esempio)

```
<leafItem name="menu_name" label="Menu" image="menu.png"
type="leafItemHierarchicalFilterMenu">
  <parameter name="gwClassName" value="asm_asset"
hideToClient="false"></parameter>
  <parameter name="filteringCriteria"
value="brand,model,id_entity_classification,barcode"
hideToClient="false"></parameter>
  <parameter name="iconImagePaths"
value="image_1.png,image_2.png,image_3.png"
hideToClient="false"></parameter>
  <parameter name="showAllAsRoot" value="true"
hideToClient="false"></parameter>
```

```

    <parameter name="rootLabel" value="Label elemento root"
hideToClient="false"></parameter>
    <parameter name="rootIconImagePath" value="image_1"
hideToClient="false"></parameter>
</leafItem>

```

XML leaf-item gwAction leafItemSeparator

Parametri : Vincoli: (esempio)

```

<leafItem " type="leafItemSeparator"></leafItem>

```

Azioni «standard»

Si riporta di seguito il valore assunto da «action code» per le azioni standard di Geoweb®.

- Esportazione in XLS dati di una classe (attributi della lista)

standardListSelectedExportToExcel(parameterQueryList, grid)

- Esportazione in XLS dati di una classe (tutti gli attributi)

standardListSelectedAllAttributeExportToExcel(parameterQueryList, grid)

- Esportazione in XLS dati di una classe (lista prefiltrata)

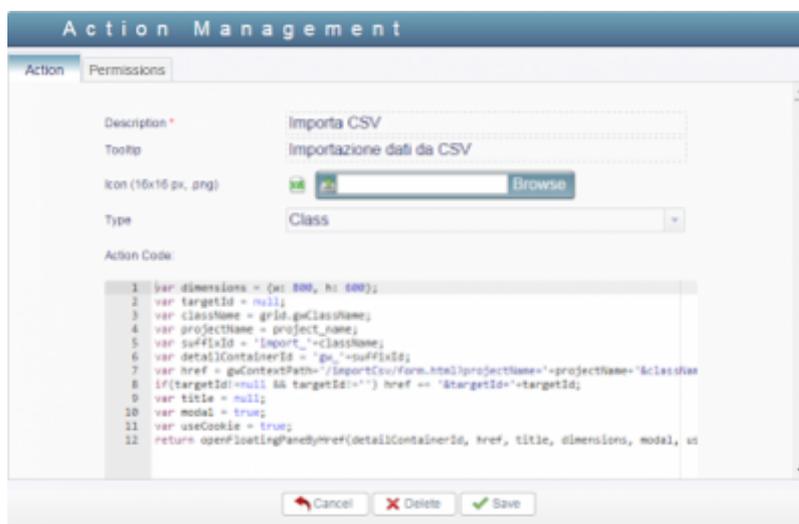
function standardPrefilterdListSelectedExportToExcel(parameterQueryList, grid)

- Esportazione in SHP dati di una classe geometrica

standardListSelectedAllAttributeExportToShp(parameterQueryList, grid)

- Aggiornamento Cache (solo per classe «gerarchizzata»)

updateCache(grid,'nome_classe');



Importazione CSV

```
var dimensions = {w: 800, h: 600};
var targetId = null;
var className = grid.gwClassName;
var projectName = project_name;
var suffixId = 'import_'+className;
var detailContainerId = 'gw_'+suffixId;
var href = gwContextPath+'/importCsv/form.html?
projectName='+projectName+'&className='+className+'&detailContainerId='+detailContainerId;
if(targetId!=null && targetId!="") href += '&targetId='+targetId;
var title = null;
var modal = true;
var useCookie = true;
return openFloatingPaneByHref(detailContainerId, href, title, dimensions, modal, useCookie);
```

Il file CSV utilizzato per l'importazione dei dati deve presentare i requisiti riportati in tabella.

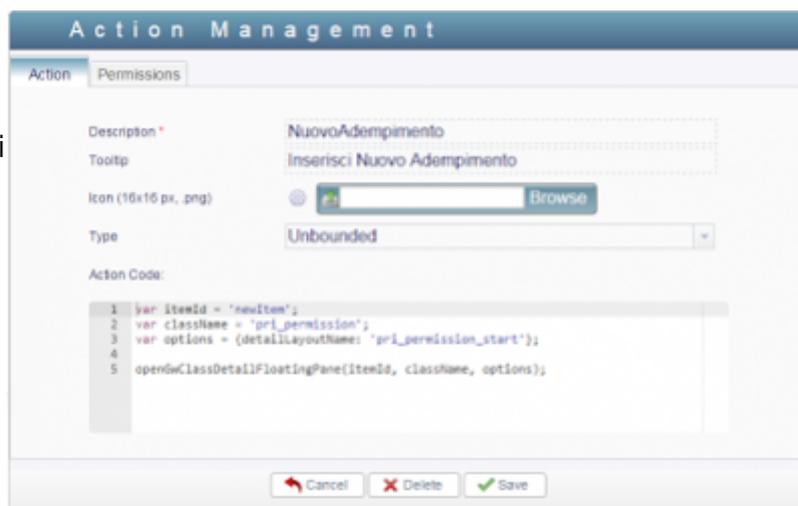
intestazione	si, nome dei campi della tabella di destinazione
separatore di campo	punto e virgola
separatore cifre decimali	punto
formato data	YYYY-MM-DD
formato data + ora	YYYY-MM-DD HH:MM:SS

- Apri Scheda Dettaglio da Leaf Item

```
var itemId = 'newItem';
var className = 'NomeClasse';
var options = {detailLayoutName: 'NomeLayoutClasse'};

openGwClassDetailFloatingPane(itemId, className, options);
```

L'azione consente di visualizzare la scheda di dettaglio di una classe in modalità «editazione» indipendentemente dai permessi attribuiti ai gruppi utente sulla classe stessa.



L'azione viene in genere configurata come voce di menù (LeafItem type = gwAction)

L'azione deve essere configurata come tipo «unbounded» (libera).

Nel codice dell'azione devono essere configurati:

il nome della classe Il nome del layout della classe da utilizzare (opzionale, nel caso in cui non venga specificato, verrà utilizzato il layout di default)

Utilità Varie

Sommario

- Funzione di ordinamento delle liste di record
- Ordinamento Liste

La configurazione dell'ordinamento dei record nelle liste avviene attraverso i seguenti tag presenti nell'XML del generico widget.

- `isDefaultOrderBy` (default: true): permette di includere il campo nelle regole di ordinamento
- `isDefaultOrderByAscending` (default: true): permette di definire se l'ordinamento deve essere effettuato in modalità "ascendente" (true) o in modalità "discendente" (false)
- `isDefaultOrderByOnFieldToShow` (default: true): permette di definire se l'ordinamento deve essere effettuato per il valore "visualizzato" (true) o per il valore "salvato" (false).

N.B. 1 - La funzione di ordinamento è resa disponibile a partire dai rilasci di ottobre 2015.

N.B. 2 - Nel caso di widget già configurati, per la visualizzazione dei tag si può:

- resettare e riconfigurare il widget (consigliato)
- copiare manualmente i tag di configurazione dell'ordinamento all'interno dell'XML del widget

Schede

In generale una componente scheda apre in Geoweb un tab specializzato nella visualizzazione/gestione di uno specifica set di dati. A seconda che il Progetto Geoweb dove ci troviamo sia di tipo 'GeoManager' o di tipo 'GeoExplorer', le schede potrebbero venire visualizzate con modalità diverse. Con il GeoManager vengono tutte aperte come tab nel tab container principale, mentre invece nel GeoExplorer, a seconda del tipo, potrebbero venire aperte in un apposito spazio, che lascia sempre in primo piano le schede del tab container principale (schede che in genere sono di tipo gwMap, ma non necessariamente). In genere tutte le schede hanno una barra header in alto, del colore del tema del framework. A seconda del tipo di scheda e del progetto di dove viene visualizzata, la barra header potrebbe essere più o meno spessa, potrebbe contenere o meno un titolo e potrebbe ospitare un apposito tasto per visualizzare la scheda a tutto schermo.

Una scheda può essere aperta in più maniere, che tipicamente sono:

- Azione JavaScript
- LeafItem

Azione JavaScript

L'azione JavaScript può essere lanciata da più parti (vedi sezione [TODO MAKE LINK]). A seconda di dove viene lanciata un'azione si avrà un diverso set di argomenti disponibili in ingresso.

LeafItem

Un leafItem è una tipologia di menu di terzo livello che si può inserire nell'interfaccia utente di un progetto Geoweb.

Per inserire questa voce di menu bisogna aggiungere nell'xml di Progetto [TODO MAKE LINK] un tag leafItem [TODO MAKE LINK].

Un tag leafItem è sempre figlio di un tag accordionPaneItem [TODO MAKE LINK] di tipo leafItemContainer.

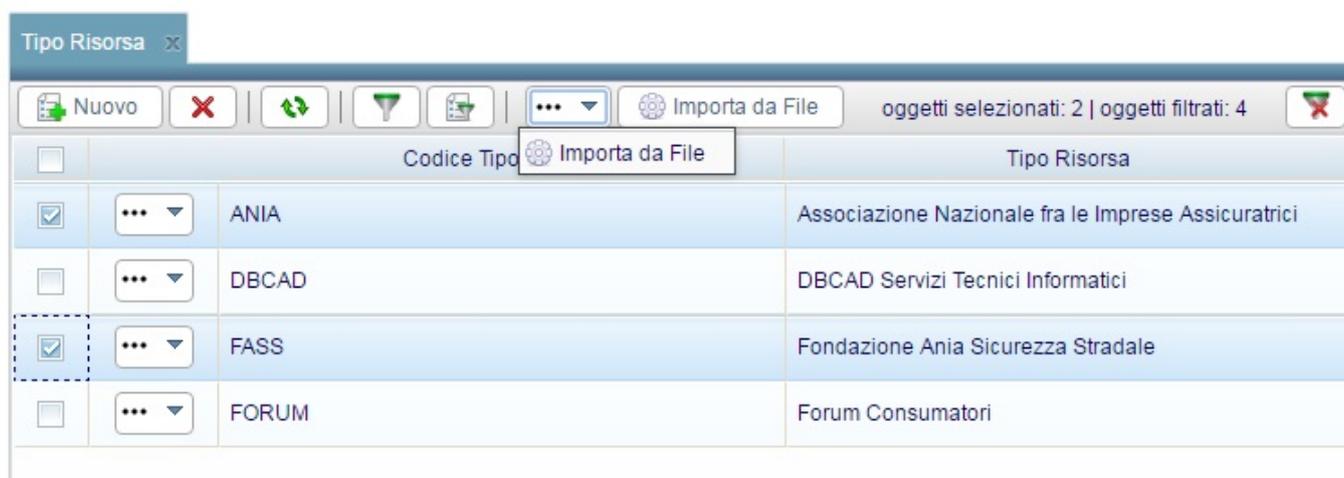
A determinare il tipo di scheda aperta sarà l'attributo type del leafItem.

Un leafItem tipicamente può avere come tag figli un set di tag parameter [TODO MAKE LINK] che ne vanno a personalizzare la modalità grafiche ed di funzionamento della scheda.

```
<accordionPaneItem name="accordionPaneItem_name" label="Label"
image="image.png" type="leafItemContainer">
  <leafItem name="test_gw_gantt" label="Test gwGantt"
image="test_gw_gantt.png" type="gwLeafItemType">
  <!-- inizio lista dei paramteri -->
  ...
  <parameter name="parameter_name" value="parameter_value"
hideToClient="false"></parameter>
  ...
  <!-- fine lista dei paramteri -->
</leafItem>
</accordionPaneItem>
```

Tipi di schede

Scheda gwClassList



	Codice Tipo	Tipo Risorsa
<input checked="" type="checkbox"/>	ANIA	Associazione Nazionale fra le Imprese Assicuratrici
<input type="checkbox"/>	DBCAD	DBCAD Servizi Tecnici Informatici
<input checked="" type="checkbox"/>	FASS	Fondazione Ania Sicurezza Stradale
<input type="checkbox"/>	FORUM	Forum Consumatori

Con la scheda di tipo gwClassList viene aperta una griglia evoluta, che permette di visualizzare e gestire il contenuto dei record di una classe Geoweb. Termini equivalenti con i quali si fa riferimento alla griglia di Geoweb in questa guida sono: lista, grid, gwClassList.

Parametri

- **className**, String, richiesto, il nome della classe Geoweb della quale si vuole aprire la lista dei record.
- **staticFilters**, Object [], opzionale, default null, permette di imporre criteri di filtro per poter mostrare solo i record di dati aderenti. Array di oggetti JSON filtro [TODO MAKE LINK] del tipo:

```
{
  condition: 'AND',
  columnName: 'id_item',
  attributeGwid: 1,
  operator: 'IN',
  filterType: 'INTEGER',
  value: [1, 2, 3]
}
```

- **hideToolbar**, Boolean , opzionale, default false, se true nasconde la Toolbar
- **hideNewIgnorePermissions**, Boolean , opzionale, default false, se true nasconde dalla Toolbar il tasto Nuovo a prescindere dai permessi che ha il gruppo a cui appartiene l'utente su quella classe
- **hideDeleteIgnorePermissions**, Boolean , opzionale, default false, se true nasconde dalla Toolbar il tasto Cancella a prescindere dai permessi che ha il gruppo a cui appartiene l'utente su quella classe
- **hideActionsDropDownButtonColumn** Boolean, opzionale, default false. Determina se il button 'More Options' (AKA 'Tre puntini') sarà visibile o meno. Quando è valorizzato sovrascrive, localmente, l'eventuale configurazione globale per la specifica gwClass (o per tutte) che può essere impostata a livello di Progetto, nell'apposita sezione con tag <gwClassSettings>.
- **hideIndirectSelectionColumn** Boolean, opzionale, default false. Determina se la colonna della griglia adibita alla selezione delle singole righe sarà visibile o meno. Quando è valorizzato sovrascrive, localmente, l'eventuale configurazione globale per la specifica gwClass (o per tutte) che può essere impostata a livello di Progetto, nell'apposita sezione con tag <gwClassSettings>.
- **forceListedAttributes** Boolean , opzionale, default false, lavora insieme al parametro 'listedAttributesNames' se true mostra in griglia solo le colonne degli attributi il cui name è specificato nella lista listedAttributesNames
- **listedAttributesNames**, ArrayList<String>, opzionale, default null, valutato solo se 'forceListedAttributes' vale true. Lista, separata da virgola, dei nomi degli attributi le cui colonne vogliamo compaiano in griglia
- **hideAttributesNamesList**, ArrayList<String>, opzionale, default null. Lista, separata da virgola, dei nomi degli attributi le cui colonne vogliamo non compaiano in griglia: vengono rimosse dagli attributi in lista impostati sulla classe. Viene valutato solo con hideAttributesNamesList diverso da true
- **customBeforeColumns**, ArrayList<HashMap<String, String>>, opzionale, default null. Lista (Array) di oggetti, ognuno dei quali contiene tutte le informazioni necessarie alla renderizzazione di tutte le celle di una colonna. Queste colonne vengono aggiunte PRIMA delle colonne degli attributi base in lista della classe. Ogni colonna è definita da oggetti del tipo:
 - { formatter: 'column_formatter', label: 'Header Label', width: 'auto' || 50, headerStyles: 'color: red;', styles: 'color: red;' }
- dove *formatter* è il nome della funzione js che genera il contenuto della cella (invocata con questi argomenti: 'item', 'rowIndex', 'cell'), label è l'intestazione della colonna, width può essere un numero intero (px) o 'auto', headerStyles e styles sono regole css applicate rispettivamente ad intestazione della colonna e celle.
- **customAfterColumns**, ArrayList<HashMap<String, String>>, opzionale, default null. Lista

(Array) di oggetti, ognuno dei quali contiene tutte le informazioni necessarie alla renderizzazione di tutte le celle di una colonna. Queste colonne vengono aggiunte DOPO delle colonne degli attributi base in lista della classe. Ogni colonna è definita da oggetti del tipo:

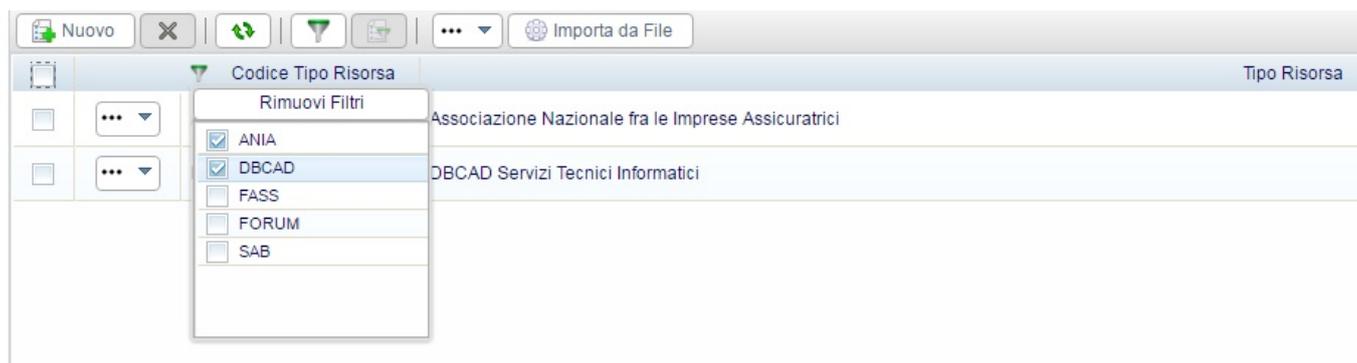
- { `formatter`: 'column_formatter', `label`: 'Header Label', `width`: 'auto' || 50, `headerStyles`: 'color: red;', `styles`: 'color: red;' }
- dove *formatter* è il nome della funzione js che genera il contenuto della cella (invocata con questi argomenti: 'item', 'rowIndex', 'cell'), `label` è l'intestazione della colonna, `width` può essere un numero intero (px) o 'auto', `headerStyles` e `styles` sono regole css applicate rispettivamente ad intestazione della colonna e celle.
- **onRowClickGwActionName**, String, opzionale, default null. Se presente indica il *name* della gwAction, esistente nella stessa gwClass, che sarà invocata con il seguente set di parametri [/*Integer*/ rowIndex, /*Object*/ grid, /*Object*/ item]. Ha priorità su *onRowClickFunctionName*
- **onRowClickFunctionName**, String, opzionale, default null. Se presente indica il nome nel definito nel name-space globale di una funzione js la quale verrà invocata (passando questi parametri [/*Integer*/ rowIndex, /*Object*/ grid]) al click della riga della griglia. Ha priorità su *onRowClickFunctionString*
- **onRowClickFunctionString**, String, opzionale, default null. Se presente indica direttamente il codice js che verrà invocato (passando questi parametri [/*Integer*/ rowIndex, /*Object*/ grid, /*Object*/ item], al click della riga della griglia

Parametri Deprecati

- **onRowClickFunction**, String, opzionale, default null. Se presente indica direttamente il codice js che verrà invocato (passando questi parametri [/*Integer*/ rowIndex, /*Object*/ grid, /*Object*/ item], al click della riga della griglia. Il parametro è deprecato: usa

onRowClickFunctionString al suo posto.

Nella parte superiore della scheda è presente una toolbar che, a seconda del set di permessi (statici e dinamici) impostati nel Geoweb Admin, mostra tutta una serie di bottoni, in ordine da sinistra a destra:



- Nuovo, visibile solo l'utente ha il permesso di inserimento, al click apre un dettaglio (su tab o floatingPane, a seconda della configurazione) con una form dove inserire i dati del nuovo record.
- Cancella, visibile solo l'utente ha il permesso di eliminazione, attivo solo con uno o più record selezionati, al click apre un dialog modale dove viene richiesta una conferma: in caso di risposta affermativa il record verrà eliminato dal DB e rimosso dalla lista
- Ricarica, sempre mostrato, al click esegue il refresh del set di dati della lista.
- Filtra, sempre mostrato, al click apre un dialogo, non modale, che permette di scegliere criteri di filtro (che si aggiungono ad eventuali filtri statici impostati col parametro staticFilters) che

andranno a filtrare i record mostrati in griglia. L'eventuale filtro impostato è rimovibile dal tasto 'Rimuovi Filtro' (vedi sotto) che compare a destra nella toolbar se un filtro è impostato.

- Raffina, sempre mostrato, attivo solo con uno o più record selezionati, al click crea un filtro a partire dalle righe selezionate (tramite l'apposito check a sinistra), mostrando solo quelle. L'eventuale filtro impostato è rimovibile dal tasto 'Rimuovi Filtro' (vedi sotto) che compare a destra nella toolbar se un filtro è impostato.
- Azioni Utente, bottone che al click apre un menu a tendina contenente tanti menulitem, i quali al click eseguono specifiche azioni, o aprono reportistiche. I menulitem report sono mostrati se esiste le relative report di classe. I menulitem di tipo azione sono mostrati se esistono le relative azioni e se sono attivi i relativi permessi di esecuzione per l'utente che sta visualizzando la griglia. In caso di nessuna azione e nessuna report configurate, il bottone non viene visualizzato.
- Azioni in Toolbar, numero variabile di bottoni, che al click eseguono specifiche azioni javascript, mostrati solo in presenza di azioni configurate e con l'utente in possesso dei relativi permessi di esecuzione.
- Rimuovi Filtro, mostrato solo in presenza di un filtro, non statico, impostato sulla lista. al click rimuove il filtro ed il tasto di conseguenza scompare.

Oltre ai bottoni relativi al filtro presenti sulla toolbar, se configurati possono esserci dei tastini detti 'filtri di colonna' [TODO MAKE LINK]. Al click viene aperto un menu a tendina, con una serie di check che attivano/disattivano i relativi filtri (messi in AND ad eventuali filtri di qualunque tipo già presenti), ed un tasto 'Rimuovi Filtri' (rimuove tutti e solo i filtri impostati dal quel particolare filtro di colonna).



Su ogni riga della griglia, nella prima colonna a sinistra, è disponibile un check, che serve per selezionare la riga, ed attivare una serie di funzioni (come affina, elimina). Nella seconda colonna è disponibile un bottone con le Azioni Utente, che permette di eseguire le azioni e lanciare le report esplicitamente configurate per operare sul singolo record. Anche qui, se non sono configurate le azioni o le report per il record singolo, l'intera colonna con i tasti Azioni Utente non viene mostrata.

Le colonne successive sono generate in automatico, a partire dagli attributi configurati nel Geoweb Admin. Si può impostare l'ordine e la larghezza della singola colonna. Lo stile degli header di colonna e delle celle possono essere anche essi ampiamente customizzati dall'xml di configurazione del tipo di controllo (widget) associato all'attributo. Nella sezione specifica sono esposti ulteriori dettagli [TODO MAKE LINK].

Al click sui record della griglia si apre il dettaglio della classe. La modalità di apertura di default è floatingPane. questa modalità può essere sovrascritta dall'xml del Progetto Geoweb, impostandola su tab [TODO MAKE LINK]

Requisiti

Per funzionare il componente ha bisogno di questi vincoli:

- esistenza della classe con name uguale al parametro 'className' nei metadati di Geoweb
- gli attributi che devono comparire in lista devono essere correttamente configurati: anche il non corretto funzionamento di uno di essi può causare, il non funzionamento della griglia (in genere sono problematici gli attributi che hanno nella loro configurazione una query, o comunque hanno il concetto di campo in visualizzazione, *fieldToShow*, e campo da salvare, *fieldToStore*)

JavaScript

Esempi con codice minimale:

```
var gwClassName = 'gwd_resource';
openGwClassListTab(gwClassName);
```

```
var gwClassName = 'gwd_resource';
var tabWidgetTitle = 'Risorse Operative'; //optional
openGwClassListTab(gwClassName, tabWidgetTitle);
```

Esempi con utilizzo filtri statici:

```
var gwClassName = 'gwd_resource';
var tabWidgetTitle = 'Risorse Operative'; //optional
var options = { //optional
  forceReplaceTab: true,
  staticFilters: [
    {condition:'AND', columnName:'type_contract',
      operator:'=', filterType:'STRING', value: ['Forfait']}
  ]
};
openGwClassListTab(gwClassName, tabWidgetTitle, options );
```

Esempi con la notazione più generale:

```
var tabWidgetType = 'gwClassList';
var identifier = 'resources_list';
var tabWidgetId = createTabId(tabWidgetType, identifier);
var tabWidgetTitle = 'Risorse Operative';

var parametersMap = {
  forceReplaceTab: true,
  className: 'gwd_resource',
  staticFilters: [
    {condition:'AND', columnName:'type_contract',
      operator:'=', filterType:'STRING', value: ['Forfait']}
  ]
};
openTab(tabWidgetId, tabWidgetType, tabWidgetTitle, parametersMap);
```

LeafItem

```
<leafItem name="resources_list" label="Resources" image="resources.png"
type="gwClassList">
  <parameter name="className" value="gwd_resource"
hideToClient="false"></parameter>
  <parameter
    name="staticFilters"
    value="{
      condition: 'AND',
      columnName: 'cod_resource',
```

```
        operator: 'IN',  
        filterType: 'INTEGER',  
        value: [1, 2, 3]  
    }]"  
    hideToClient="false">  
</parameter>  
</leafItem>
```

Scheda gwClassProcesses

In Geoweb è presente una scheda per la visualizzazione e la gestione dei processi, dove fra le altre cose, è possibile:

- avviare una nuova istanza di processo
- vedere tutte le proprie attività
- completare le proprie attività
- vedere le attività degli altri utenti coinvolti nel processo (in sola lettura)
- richiedere per se (Claim) lo svolgimento di tutti quei Task che non sono stati ancora assegnati ma prevedono una lista di utenti candidati od una lista di gruppi candidati a svolgere quella attività (fra i quale si è presenti).
- consultare la lista dei processi in corso
- consultare la lista dei processi terminati (in sola lettura)
- consultare un immagine grafica del modello .bpmn del processo
- consultare un immagine grafica del modello .bpmn per i singoli task, processi in corso, e processi terminati

Liste Attività ('Mie' e 'Tutte')

I record di tali liste hanno:

- Sfondo verde: se il task è assegnato da me.
- Sfondo Giallo: se il task è assegnato a una specifica lista di utenti, o ad una specifica lista di gruppi, nei quali è incluso l'utente corrente, od il gruppo dell'utente corrente.
- Sfondo Bianco: se il task è assegnato ad un altro utente oppure non l'utente corrente non risulta nella lista degli utenti candidati all'esecuzione.

significa che l'utente non ha ancora aperto la form task, da quando gli è stato assegnato

apre il dettaglio della classe di geoweb che contiene tutte le variabili di processo

apre lo schema del processo

Attività	Codice Ticket	Servizio	Richiesta	Richiedente	Iniziata	Scadenza
Esecuzione					04/04/2016 11:36:46	
Esecuzione					04/04/2016 11:05:24	
Esecuzione					04/04/2016 11:03:55	
Esecuzione					04/04/2016 11:01:24	
Verifica					04/04/2016 10:00:22	
Verifica					04/04/2016 09:46:36	
Verifica					04/04/2016 09:42:48	
Verifica					04/04/2016 09:20:47	
Verifica					02/04/2016 15:29:00	
Verifica					02/04/2016 15:09:33	
Verifica					02/04/2016 10:07:06	
Esecuzione					31/03/2016 18:34:57	
Esecuzione					31/03/2016 18:32:58	

Se è presente il marker nella prima colonna significa che la form del task non è mai stata aperta dall'utente dopo che gli è stata assegnata tale attività dal motore di workflow.

In queste schede ci sono colonne fisse (nome attività, assegnatario, data inizio, scadenza, tasto apertura diagramma, tasto apertura record istanza processo) e colonne variabili (individuata in base agli attributi della classe di processo messi in lista)

Qui sotto un esempio di diagramma di processo.

Tutte le Attività

Assignatario Attività Codice Ticket Servizio Richiesta Richiedente Iniziata Scadenza

05/04/2016 12:29:32

05/04/2016 12:27:36

05/04/2016 12:06:48

04/04/2016 17:43:47

04/04/2016 17:12:41

04/04/2016 16:32:27

04/04/2016 11:36:46

04/04/2016 11:05:24

04/04/2016 11:03:55

04/04/2016 11:01:24

04/04/2016 10:00:22

04/04/2016 09:46:36

04/04/2016 09:42:48

04/04/2016 09:20:47

02/04/2016 15:29:00

```
graph LR; Start((Start by Message)) --> Verifica[Verifica]; Verifica --> Decision1{autorizzazione necessaria?}; Decision1 -- si --> Autorizzazione[Autorizzazione]; Decision1 -- no --> Esecuzione[Esecuzione]; Autorizzazione --> Decision2{autorizzazione concessa?}; Decision2 -- autorizzazione concessa --> Esecuzione; Decision2 -- autorizzazione negata --> EsecuzioneEsito[Esecuzione Esito]; Esecuzione --> VerificaEsecuzione[Verifica Esecuzione]; VerificaEsecuzione --> Decision3{non verificata?}; Decision3 -- non verificata --> Esecuzione; Decision3 -- verificata --> EsecuzioneEsito; EsecuzioneEsito --> End((End));
```

Liste Istanze di processo ('In Corso' e 'Archivate')

Questo è la parte di codice da aggiungere all'xml di Progetto Geoweb per far comparire una voce di menu che al click apra la scheda Processo:

```
<leafItem name="request_fulfilment" label="Request Fulfilment"
image="process.png" type="gwClassProcesses">
    <parameter name="className" value="rqm_wpi_request_fulfilment"
hideToClient="false"></parameter>
    <parameter name="gwProcessName" value="rqm_wpi_request_fulfilment"
hideToClient="false"></parameter>
</leafItem>
```

Scheda gwGantt

Con la scheda di tipo gwGantt Geoweb si dota di un visualizzatore/gestore di modelli gantt.

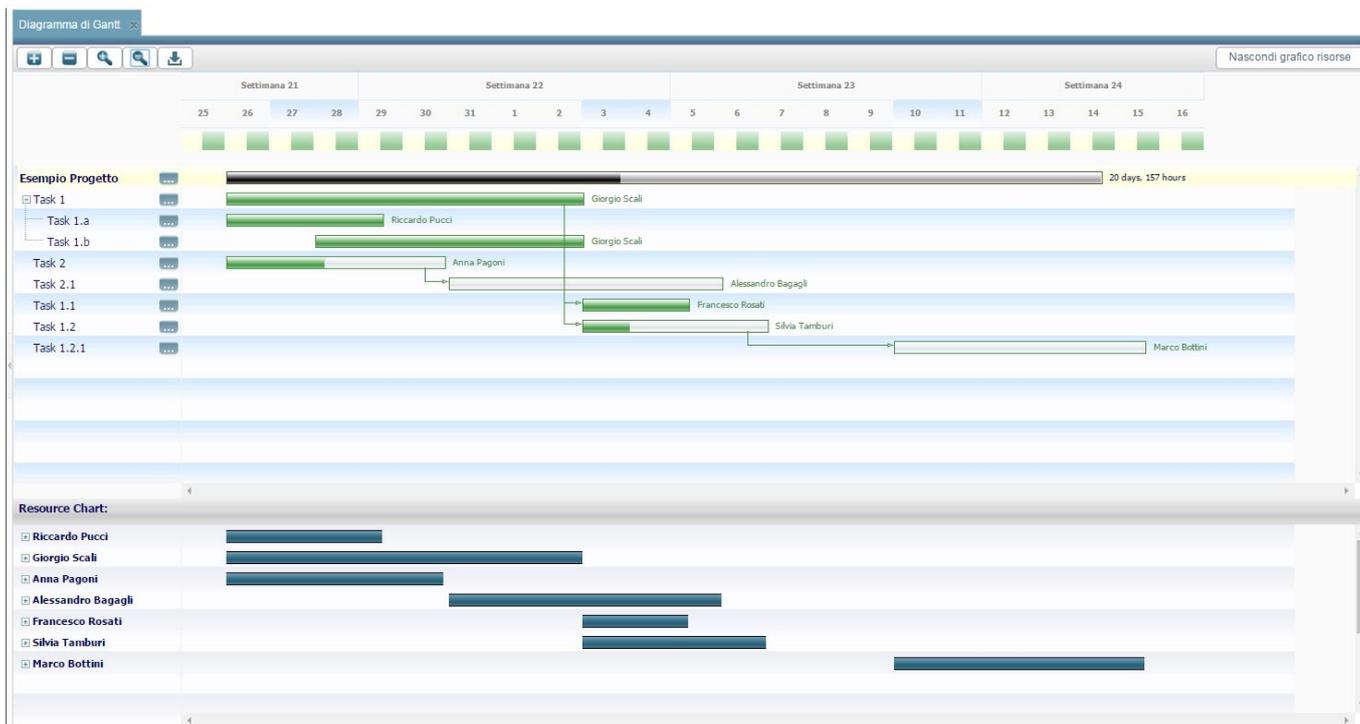
- Gantt in Generale
- Gantt in Geoweb
- Parametri
- Funzionalità
- Requisiti
- JavaScript
- LeafItem

Gantt in Generale

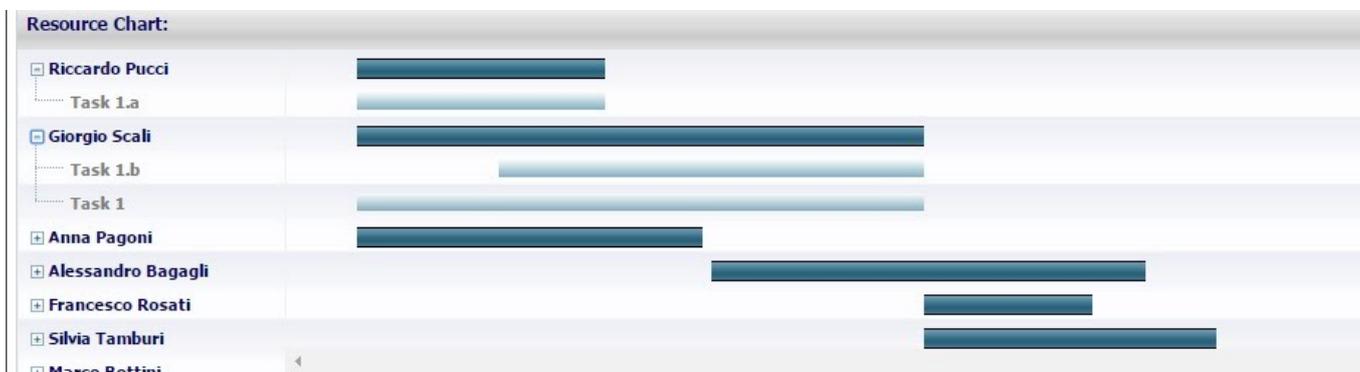
Il diagramma di Gantt è uno strumento di supporto alla gestione dei progetti, così chiamato in ricordo dell'ingegnere statunitense Henry Laurence Gantt (1861-1919), che si occupava di scienze sociali e che lo ideò nel 1917. Il diagramma di Gantt usato principalmente nelle attività di project management, è costruito partendo da un asse orizzontale - a rappresentazione dell'arco temporale totale del progetto, suddiviso in fasi incrementali (ad esempio, giorni, settimane, mesi) - e da un asse verticale - a rappresentazione delle mansioni o attività che costituiscono il progetto. Delle barre orizzontali di lunghezza variabile rappresentano le sequenze, la durata e l'arco temporale di ogni singola attività del progetto (l'insieme di tutte le attività del progetto ne costituisce la work breakdown structure). Queste barre possono sovrapporsi durante il medesimo arco temporale ad indicare la possibilità dello svolgimento in parallelo di alcune delle attività. Man mano che il progetto progredisce, delle barre secondarie, delle frecce o delle barre colorate possono essere aggiunte al diagramma, per indicare le attività sottostanti completate o una porzione completata di queste. Una linea verticale è utilizzata per indicare la data di riferimento. Un diagramma di Gantt permette dunque la rappresentazione grafica di un calendario di attività, utile al fine di pianificare, coordinare e tracciare specifiche attività in un progetto dando una chiara illustrazione dello stato d'avanzamento del progetto rappresentato; di contro, uno degli aspetti non tenuti in considerazione in questo tipo di diagrammazione è l'interdipendenza delle attività, caratteristica invece della programmazione reticolare, cioè del diagramma PERT[1]. Ad ogni attività possono essere in generale associati una serie di attributi: durata (o data di inizio e fine), predecessori, risorsa, costo.

Gantt In Geoweb

La scheda gwGantt di Geoweb si basa sul componente dojox.gantt.GwGantt



La scheda Gantt in Geoweb è costituita da due sezioni. La principale, in alto, è il diagramma vero e proprio. La seconda, in basso, denominata 'Grafico Risorse', mostra sulla linea del tempo il carico delle attività raggruppandole per risorsa. Questo grafico può essere nascosto/mostrato alla bisogna agendo sul pulsante in altro a destra della toolbar.



Essa permette, a seconda del set dei permessi, di aggiungere/rimuovere/modificare Progetti e relativi Task. Si possono aggiungere anche Task non direttamente figli di un Progetto, ma vincolati ad altri Task. In particolare abbiamo 2 tipi di vincoli: * vincolo precedente/successivo: due Task sono in questa situazione se l'esecuzione del secondo task non può avvenire prima che sia terminata l'esecuzione del primo (Es: Task 1.1 è successivo al Task 1)

* vincolo padre/figlio: un Task impostato come 'figlio' di un altro quando è in pratica una sotto attività dello stesso (Es: Task 1.a è figlio del Task 1). Deve essere eseguito nell'intervallo temporale occupato dal task padre, anche se senza un particolare ordine rispetto ad altri eventuali sub-Task 'fratelli'.

Il gwGantt preserva questi vincoli facendo validazioni quando l'utente prova a cambiare i parametri temporali dei task. In caso di validazione negativa l'operazione viene negata mostrando un messaggio di errore specifico. Si possono innestare e concatenare i vincoli fra i task a piacimento. Per esempio si può avere che un task figlio (di un altro task) abbia dei suoi task successivi. E tutte le altre possibili combinazioni.

Il gwGantt permette la modifica di tutti i parametri di sua competenza dalla propria interfaccia, ma in generale non permette la modifica contemporanea dei parametri che determinano vincoli temporali:

la modifica è permessa per un solo parametro temporale alla volta. Questa scelta viene fatta per semplificare i controlli: dipendendo alcuni parametri temporali da altri, la modifica in contemporanea di più parametri potrebbe generare dati inconsistenti.

Per esempio non posso impostare al Task 1 una durata tale per cui, sommata alla sua data di inizio, mi determini una fine prevista che vada oltre la data di inizio del task 1.1. Nel caso volessi mantenere come vincolo quella durata dovrei prima o anticipare la data di inizio del Task 1 o posticipare la data di inizio del Task 1.1.

Inoltre, per esempio, non posso far iniziare/terminare il Task 1.a, figlio del Task 1, prima/oltre dell'inizio/della fine del padre Task 1.

Proprio per preservare la consistenza dei parametri temporali di progetti e task, nelle implementazioni che usano il gwGantt è molto importante che gli attributi con name start_date (per i progetti), start_time, duration e duration_in_days (per i task), siano tutti rigorosamente posti a readonly. Questo per evitare che l'utente possa creare inconsistenze modificando tali valori dalle form in edit dei dettagli di Classe (gwClassDetail). Sarà il gwGantt stesso che internamente (e in modo trasparente a Geoweb) andrà a sovrascrivere di volta in volta le proprietà required, hidden e readonly dei vari attributi.

Parametri

- “projects”, richiesto, alternativo a projectsFilters, determina quali progetti saranno visibili nel gwGantt. Può essere un singolo valore String, multipli valori String uniti insieme e separati da virgola, o array String [] contenenti valori String.

Esempi di configurazione validi: 'project_name' 'project_name_1,project_name_2' ['project_name_1', 'project_name_2']

- “projectsFilters”, richiesto, alternativo a projects, determina quali progetti saranno visibili nel gwGantt. Array di oggetti JSON filtro [TODO MAKE LINK] del tipo:

```
[
  {
    condition: 'AND',
    columnName: 'id_project',
    attributeGwid: 1,
    operator: 'IN',
    filterType: 'INTEGER',
    value: [1, 2, 3]
  }
]
```

- “gwOuterContainerId”, richiesto, String, è l'id del contenitore dojo che racchiude il widget gwGantt. E' usato per lanciare la function di refresh del widget gwGantt.
- “projectGwClassName”, opzionale, String, default 'ppm_gantt_project', nome della gwClass dei Progetti Gantt dove lavora il gantt. Può essere utilizzato per settare il nome di gwClass eventualmente costruite su viste basate sulla tabella 'ppm_gantt_project'
- “taskGwClassName”, opzionale, String, default 'ppm_gantt_task', nome della gwClass dei Task Gantt dove lavora il gantt. Può essere utilizzato per settare il nome di gwClass eventualmente costruite su viste basate sulla tabella 'ppm_gantt_task'

- “taskDurationChooseMode”, opzionale, String, valori permessi: 'duration' (default), 'duration_in_days', 'end_time'. I valori permessi non sono altro che i nomi degli attributi (che devono tutti esistere) dei widget che permettono la modifica della durata temporale dei task. Questo parametro controlla la modalità con la quale l'utente potrà modificare nella form di dettaglio del gwGantt la durata dei singoli task. duration, valore di default, mostra un attributo che permette di impostare la durata del task in ore. duration_in_days, mostra un attributo che permette di impostare la durata del task in giorni (sul DB verrà persistito il valore della durata in ore, considerando il vincolo 8 ore/giorno). end_time, mostra un attributo che permette di impostare la data di fine del task in giorni (sul DB verrà persistito il valore della durata in ore, calcolando la differenza in giorni tra la data di fine e quella di inizio e considerando il vincolo di 8 ore/giorno). Questo parametro influisce anche su quale attributo (relativo alla durata del task) viene mostrato nelle form di dettaglio del gwGantt, anche quando si modificano altri parametri: per esempio il nome del gwGantt
- “withResourceChart”, opzionale, Boolean, default true. Flag che abilita o meno la visualizzazione del Resource Chart
- “limitEditing”, opzionale, Boolean, default false. Se true restringe il set di funzioni disponibili per l'editing di Project e Task rendendo modificabili solamente name, percentage e taskOwner. E' necessario perché attualmente il gwGantt non ha un modello dati elaborato come quello di MS Project. Accade che, modificando in GW i parametri temporali di progetto importato da file MSPDI, e successivamente riesportandolo, si generino inconsistenze. Questo perché il sistema di editing in GW non considera né i 'Calendar', né altri aspetti del modello. Limitando l'editing solo ad un set di funzioni sicure, è possibile in seguito esportare il progetto garantendone l'originale consistenza dei dati temporali.

I prossimi 7 parametri, opzionali, riguardano l'eventuale sovrascrittura dei permessi di Geoweb, rispettivamente per le classi dei Progetti Gantt e dei Task Gantt. Sono utilizzati solo se presenti. In loro mancanza valgono i permessi base (statici/dinamici) impostati in Geoweb per le classi dei Progetti e dei Task. Ciò è utile perché spesso si vuole rendere i parametri temporali di Progetti e Task editabili solo dal componente gwGantt, per prevenire possibili modifiche inconsistenti da parte degli utenti.

- “allowInsertProjects”, opzionale, Boolean, se assente valgono i permessi base (statici/dinamici)
- “allowModifyProjects”, opzionale, Boolean, se assente valgono i permessi base (statici/dinamici)
- “allowModifyProjectsPercentage”, Boolean, opzionale, default a true, valutato solo se allowModifyProjects vale true o sono presenti i permessi di modifica base (statici/dinamici) sulla classe dei progetti. Se assente o è posto a true, il gwGantt permette di settare la percentuale del progetto. Il meccanismo che, alla modifica della % di completamento di un task, ricalcola la anche la % globale del progetto (pesando le componenti delle % di completamento dei singoli task) continua a funzionare. Cioè settando la % del progetto, questa verrà poi sovrascritta alla prima modifica della % effettuata sul singolo task
- “allowDeleteProjects”, opzionale, Boolean, se assente valgono i permessi base (statici/dinamici)
- “allowInsertTasks”, opzionale, Boolean, se assente valgono i permessi base (statici/dinamici)
- “allowModifyTasks”, opzionale, Boolean, se assente valgono i permessi base (statici/dinamici)
- “allowDeleteTasks”, opzionale, Boolean, se assente valgono i permessi base (statici/dinamici)

Funzionalità

Sulla toolbar sono disponibili i seguenti comandi: Gw Gantt Toolbar Allarga sequenza temporale Contrai sequenza temporale Zoom In Zoom Out Esporta dati su file MSPDI (.xml) Importa dati da file MSPDI (.xml) In caso per qualche ragione ci siano attualmente più progetti mostrati nel grafico

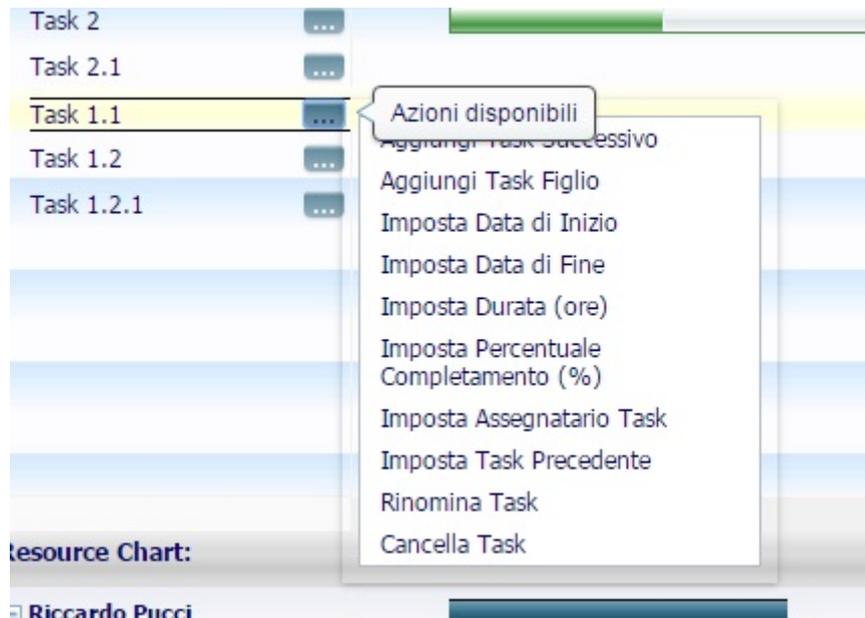
gwGantt, in fase di export ed import verrà richiesto all'utente quale progetto esportare o quale progetto sovrascrivere. In fase di esportazione, il file xml originariamente caricato (e dotato di tutti i vincoli del modello MSProject), se presente, viene usato come base, e gli vengono applicate tutte le modifiche man mano effettuate sui task. In caso di assenza del file xml, esso viene generato al volo a partire dai task presenti sul DB. Durante l'import viene sovrascritto il nome del progetto con quello del file importato. Anche qui si cerca di preservare l'univocità del name del progetto: viene aggiunta la data odierna in caso di nome già in uso, e se non dovesse bastare un ulteriore marker. Gli elementi di tipo Progetto e di tipo Task, a seconda del set di permessi impostato (derivante dai permessi base delle classi o dalle sovrascritture), espongono le seguenti azioni nel menu a tendina. Il menu a tendina compare al click sull'elemento.

Progetto:

- Aggiungi Progetto (permessi necessari: inserimento progetti)
- Rinomina Progetto (permessi necessari: modifica progetti)
- Imposta Percentuale Completamento (%) (permessi necessari: modifica progetti, permesso % progetto assente o true)
- Imposta Data di Inizio (permessi necessari: modifica progetti)
- Aggiungi Task (permessi necessari: inserimento task, modifica progetti)

Task:

- Aggiungi Task Successivo (permessi necessari: inserimento task, modifica progetti)
- Aggiungi Task Figlio (permessi necessari: inserimento task, modifica progetti)
- Imposta Data di Inizio (permessi necessari: modifica task, modifica progetti) mantiene inalterata la durata, ricalcolando (con vincolo 8 ore per giorno) la data di fine (non persistita su DB)
- Imposta Data di Fine (permessi necessari: modifica task, modifica progetti) mantiene inalterata la data di inizio, ricalcolando (con vincolo 8 ore per giorno) la durata
- Imposta Durata (ore) (permessi necessari: modifica task, modifica progetti) mantiene inalterata la data di inizio, ricalcolando (con vincolo 8 ore per giorno) la data di fine (non persistita su DB)
- Imposta Durata (giorni) (permessi necessari: modifica task, modifica progetti) mantiene inalterata la data di inizio, ricalcolando (con vincolo 8 ore per giorno) la durata in ore (persistita su DB) e la data di fine (non persistita su DB)
- Imposta Percentuale Completamento (%) (permessi necessari: modifica task) verrà ricalcolata di conseguenza anche la percentuale di completamento del progetto (ma solo in visualizzazione sul gantt, lato client: non previsto campo sul DB)
- Imposta Assegnatario Task (permessi necessari: modifica task)
- Imposta Task Precedente (permessi necessari: modifica task, modifica progetti)
- Rinomina Task (permessi necessari: modifica task)
- Cancella Task (permessi necessari: modifica task, modifica progetti)



Requisiti

Per funzionare il componente gwGantt ha bisogno di questi vincoli:

- esistenza gwClass con nome ppm_gantt_project (di default basata sull'omonima tabella), o sovrascrivendo 'projectGwClassName' con altro nome classe, magari costruita su una vista di ppm_gantt_project, ma sempre con i seguenti campi base (Il nome degli Attributi deve corrispondere al campo della tabella):

```
(  
id_project integer NOT NULL, -- chiave primaria (gantt)  
project_name character varying(250), -- nome del progetto (gantt)  
start_date timestamp without time zone, -- data di avvio del progetto  
(gantt)  
percentage integer, -- percentuale di completamento del progetto (gantt)  
expected_end_date timestamp without time zone, -- data di fine progetto  
prevista (da task collegati)  
xml character varying, usato per salvare l'xml durante un import tramite  
MSPDI .xml, in genere usato quando il gantt è settato con limitEditing a  
true  
)
```

- esistenza gwClass con nome ppm_gantt_task (di default basata sull'omonima tabella), o sovrascrivendo 'taskGwClassName' con altro nome classe, magari costruita su una vista di ppm_gantt_task, ma sempre con i seguenti campi base (Il nome degli Attributi deve corrispondere al campo della tabella):

```
(  
id_task integer NOT NULL, -- chiave primaria (gantt)  
task_name character varying(250), -- denominazione del task (gantt)  
start_time timestamp without time zone, -- data inizio del task (gantt)  
duration integer NOT NULL, -- durata (hh) del task (gantt)  
percentage integer, -- percentuale di completamento (gantt)  
previous_task_id integer, -- id del task precedente (gantt)  
task_owner character varying(100), -- assegnatario (nome e cognome) del
```

```

task (gantt)
  id_project integer NOT NULL, -- id del progetto di appartenenza (gantt)
  parent_task_id integer, -- id del task padre (gantt)
  is_milestone integer DEFAULT 0 -- flag per identificare particolari task
  unique_id integer campo di supporto usato usato durante le procedure di
import
)

```

- esistenza di un attributo di tipo Date con nome 'end_time' sulla classe dei task (anche non mappato su un campo della tabella). Infatti dall'interfaccia del gwGantt è possibile modificare la durata temporale di un Task specificandone una data di fine. A venire persistito sul DB non è però la data di fine, ma la durata, che viene calcolata a partire dalla data di fine del Task, mantenendo inalterata la data di inizio dello stesso. L'attributo serve alla scheda gwGantt solo per poterlo mostrare della scheda di dettaglio relativa al cambio data di fine del Task.
- esistenza di un attributo di tipo Integer con nome 'duration_in_days' sulla classe dei task (anche non mappato su un campo della tabella). Infatti dall'interfaccia del gwGantt è possibile modificare la durata temporale di un Task specificandone i giorni. A venire persistito (e rielto dal Gantt) sul DB non è però mai la durata in giorni, ma sempre il campo che contiene la durata in ore.
- Gli unici attributi del gwGantt che supportano widget che usano valori codificati (salvando il valueToStore, ma mostrando il valueToShow) sono il task_name, il task_owner.

JavaScript

```

var tabWidgetType = 'gwGantt';
var identifier =
'gantt_project_programmation'+ '_'+'show_project_in_gantt'+ '_' +data.itemDB.id
_project;
var gwOuterContainerId= createTabId(tabWidgetType, identifier);
var tabWidgetTitle = 'Diagramma di Gantt';
var projects = data.itemDB.project_name;

var parametersMap = {
  forceReplaceTab: true,
  //OVERRIDE CLASSES: if omitted defaults are used
  //projectGwClassName: 'ppm_gantt_project',
  //taskGwClassName: 'ppm_gantt_task',
  //withResourceChart: true,
  //OVERRIDE PERMISSIONS: if omitted defaults for related classes are used
  allowInsertProjects : false,
  allowModifyProjects : true,
  allowDeleteProjects : false,
  allowInsertTasks : true,
  allowModifyTasks : true,
  allowDeleteTasks : true,
  allowModifyProjectsPercentage: true,
  limitEditing: true,
  projects: projects,
  //availabkle values 'duration' (default), 'duration_in_days', 'end_time'
  taskDurationChooseMode: 'duration',
  gwOuterContainerId: gwOuterContainerId
}

```

```
};  
  
openTab(tabWidgetId, tabWidgetType, tabWidgetTitle, parametersMap);
```

LeafItem

E' un leafItem che al click apre la scheda di tipo gwTimeLine. Il type che caratterizza il leafItem è "gwGantt" Codice da inserire dentro un accordionPanelItem nell'xml di progetto:

```
<leafItem name="test_gw_gantt" label="Test gwGantt"  
image="test_gw_gantt.png" type="gwGantt"> <parameter  
name="projects" value="nome_progetto" hideToClient="false"></parameter>  
  <parameter name="projectGwClassName" value="ppm_gantt_project"  
hideToClient="false"></parameter>  
  <parameter name="taskGwClassName" value="ppm_gantt_task"  
hideToClient="false"></parameter>  
  
  <parameter name="withResourceChart" value="true"  
hideToClient="false"></parameter>  
  
  <parameter name="allowInsertProjects" value="true"  
hideToClient="false"></parameter>  
  <parameter name="allowModifyProjects" value="true"  
hideToClient="false"></parameter>  
  <parameter name="allowDeleteProjects" value="true"  
hideToClient="false"></parameter>  
  <parameter name="allowInsertTasks" value="true"  
hideToClient="false"></parameter>  
  <parameter name="allowModifyTasks" value="true"  
hideToClient="false"></parameter>  
  <parameter name="allowDeleteTasks" value="true"  
hideToClient="false"></parameter>  
  <parameter name="allowModifyProjectsPercentage" value="true"  
hideToClient="false"></parameter>  
  <parameter name="taskDurationChooseMode" value="duration"  
hideToClient="false"></parameter>  
  <parameter name="limitEditing" value="true"  
hideToClient="false"></parameter>  
  <parameter name="gwOuterContainerId" value="container_id"  
hideToClient="false"></parameter>  
</leafItem>
```

Scheda gwHtml

Con la scheda di tipo gwHtml è possibile mostrare in Geoweb il contenuto di un qualsiasi file html.

- Parametri
- Requisiti
- JavaScript

- LeafItem

Parametri

“htmlPath”, obbligatorio, il path del file .html, comprensivo di estensione, a partire dalla cartella WEB/html.

Requisiti

Per funzionare il componente ha bisogno di questi vincoli:

- esistenza cartella base WEB/html nei contenuti statici
- esistenza file .html sotto la cartella base

JavaScript

```
var tabWidgetType = 'gwHtml';
var identifier = 'cruscotto_svg';
var tabWidgetId = createTabId(tabWidgetType, identifier);
var tabWidgetTitle = 'Cruscotto';

var parametersMap = {
  forceReplaceTab: true,
  htmlPath: 'cruscotto_svg.html'
};
openTab(tabWidgetId, tabWidgetType, tabWidgetTitle, parametersMap);
```

LeafItem

```
<leafItem name="cruscotto" label="Cruscotto" image="cruscotto.png"
type="gwHtml">
  <parameter name="htmlPath" value="cruscotto_svg.html"
hideToClient="false"></parameter>
</leafItem>
```

Scheda gwHtmlReport

Con la scheda di tipo gwHtmlReport è possibile mostrare in Geoweb il contenuto di un qualsiasi file html.

- Parametri
- Requisiti
- JavaScript
- LeafItem

Parametri

- “reportUrl”, richiesto, path relativo, comprensivo di estensione .jasper, a partire dalla cartella WEB/template
- “className”, opzionale, è usato per applicare l'ambito corrente alla report.
- “updateOnChangeInGwClassNameList”, opzionale, va popolato con una lista di nomi di classi Geoweb, separati da virgola “,”.

Se presente la report html viene ricaricata automaticamente per tutte le operazioni di insert/update/delete effettuate sulle classi il cui nome è presente in lista.

Requisiti

Per funzionare il componente ha bisogno di questi vincoli:

- esistenza cartella base WEB/template nei contenuti statici
- esistenza file .jasper sotto la cartella base

JavaScript

Queste azioni, oltre che a aprire schede gwHtmlReport, possono a loro volta essere inglobate dentro le report stesse per creare un vero e proprio menu di navigazione

```
var reportUrl = 'path/To/file_name.jasper';
var updateOnChangeInGwClassNameList = 'class_name_1,class_name_2';
var parametersMap = { updateOnChangeInGwClassNameList:
updateOnChangeInGwClassNameList };
var title = 'Report Tilte';
openHtmlReport(reportUrl, parametersMap, title);
```

LeafItem

```
<leafItem name="cruscotto_html_report" label="HTML Report"
image="html_report.png" type="gwHtmlReport">
  <parameter name="reportUrl" value="path/To/file_name.jasper"
hideToClient="false"></parameter>
  <parameter name="className" value="class_name_1"
hideToClient="false"></parameter>
  <parameter name="updateOnChangeInGwClassNameList"
value="class_name_1,class_name_2" hideToClient="false"></parameter>
</leafItem>
```

Scheda gwTimeLine

- Parametri
- Requisiti
- JavaScript
- LeafItem

Descrizione

Con la scheda di tipo gwTimeLine è possibile gestire la pianificazione delle attività, relazionandola ad una risorsa ed una data. Il widget puo opzionalmente operare in sinergia con il piu generale modulo dei turni delle risorse, se presente.

Risorse		22/09/2016	23/09/2016	24/09/2016	25/09/2016	26/09/2016
2	0	0	0	0	0	0
2	1	0	2	0	1	0
BELLINI ALFREDO			00097/2016 - Normale 02:30 (2+0) 00104/2016 - Normale 10:30 (3+0)		00109/2016 - Normale 09:30 (2+0)	
0	0	0	0	0	1 (1)	0
COPPOLELLA GI...					00109/2016 - Normale 09:30 (2+0)	

Nomenclatura:

- per attività assegnata si intende un'attività vincolata ad una risorsa
- per attività programmata si intende un'attività vincolata ad una data

La scheda è sostanzialmente una tabella.

Essa ha delle date per intestazioni di colonna, tante quanto indicato dal parametro nDays. La prima intestazione di colonna è comunque sia sempre 'Risorse'.

Essa ha delle risorse per intestazione di riga, tante quanto indicato dal parametro gwdResources (o gwdResourcesFilters). La prima intestazione di riga è comunque sia sempre 'Senza Risorse'.

Le due tipologie di intestazioni di colonna incrociate con le due tipologie di intestazioni di riga, determinano 4 sezioni (una matrice 2x2) nella tabella, che contengono oggetti fra loro omogenei:

- La sezione con indici 0,0 nella matrice, cella all'incrocio fra la colonna 'Risorse' e la riga 'Senza Risorse', contiene le attività **ne assegnate ne programmate** (sezione formata da una sola cella, quella di indice 0,0).

Risorse		22/09/2016	23/09/2016	24/09/2016	25/09/2016	26/09/2016
2	0	0	0	0	0	0
2	1	0	2	0	1	0
BELLINI ALFREDO			00097/2016 - Normale 02:30 (2+0) 00104/2016 - Normale 10:30 (3+0)		00109/2016 - Normale 09:30 (2+0)	
0	0	0	0	0	1 (1)	0
COPPOLELLA GI...					00109/2016 - Normale 09:30 (2+0)	

- La sezione con indici 0,1 nella matrice, contenente celle all'incrocio fra una colonna data e la riga 'Senza Risorse', contiene le attività **programmate ma non assegnate** (sezione formata dalla prima colonna di celle, esclusa la prima cella).

Risorse	22/09/2016	23/09/2016	24/09/2016	25/09/2016	26/09/2016
2 0 Senza risorsa	0 0	0 0	0 0	0 0	0 0
2 1 BELLINI ALFREDO	0 0 0	2 5 0	0 0 0	1 2 0	0 0 0
		00097/2016 - Normale 02:30 (2+0) 00104/2016 - Normale 10:30 (3+0)		00109/2016 - Normale 09:30 (2+0)	
0 0 COPPOLELLA GI...	0 0 0	0 0 0	0 0 0	1 (1) 2 (2) 0	0 0 0
				00109/2016 - Normale 09:30 (2+0)	

- La sezione con indici 1,0 nella matrice, contenente celle all'incrocio fra la colonna 'Risorse' e una riga risorsa, contiene le attività **assegnate ma non programmate** (sezione formata dalla prima riga di celle, esclusa la prima cella).

Risorse	22/09/2016	23/09/2016	24/09/2016	25/09/2016	26/09/2016
2 0 Senza risorsa	0 0	0 0	0 0	0 0	0 0
2 1 BELLINI ALFREDO	0 0 0	2 5 0	0 0 0	1 2 0	0 0 0
		00097/2016 - Normale 02:30 (2+0) 00104/2016 - Normale 10:30 (3+0)		00109/2016 - Normale 09:30 (2+0)	
0 0 COPPOLELLA GI...	0 0 0	0 0 0	0 0 0	1 (1) 2 (2) 0	0 0 0
				00109/2016 - Normale 09:30 (2+0)	

- La sezione con indici 1,1 nella matrice, contenente celle all'incrocio fra una colonna data e una riga risorsa, contiene le attività **assegnate e programmate** (sezione formata da tutte le celle ad esclusione di quelle appartenenti alla prima riga e alla prima colonna).

Risorse	22/09/2016	23/09/2016	24/09/2016	25/09/2016	26/09/2016
Senza risorsa	0 0	0 0	0 0	0 0	0 0
BELLINI ALFREDO	0 0 0	2 5 0	0 0 0	1 2 0	0 0 0
COPPOLELLA GI...	0 0 0	0 0 0	0 0 0	1 (1) 2 (2) 0	0 0 0

Sulle celle di ogni sezione è presente una toolbar sommario che può indicare dove applicabile:

- numero delle attività della risorsa per quel giorno (tra parentesi le eventuali a supporto).
- numero delle ore assegnate alla risorsa per quel giorno (tra parentesi le eventuali a supporto) (diventa con sfondo rosso se le ore assegnate superano quelle lavorative).
- numero delle ore lavorative della risorsa per quel giorno.
- disponibilità della risorsa per un dato giorno (compare in verde solo in caso di disponibilità, con info sugli orari di disponibilità).

Sulla sezione delle attività assegnate e programmate compare inoltre una preview delle attività. Tale preview presenta per ogni attività informazioni come codice, criticità, ora di inizio e durata. Essa si limita a mostrare le prime due attività del giorno, eventualmente evidenziando la presenza di altre (tramite '...').

In corrispondenza della prima colonna, sulle celle di ogni risorsa, è presente un tastino che permette di espandere/collapsare l'intera riga relativa alla risorsa, così da mostrare in preview un numero maggiore di attività.

Dove applicabile sono presenti numerosi tooltip contenenti tutte le informazioni necessarie, molto utili in contesti dove la mole delle attività potrebbe nascondere le info delle stesse (es: scheda dettaglio attività programmate ed assegnate)

Scheda di dettaglio gruppo attività

Al click su una cella una qualsiasi sezione viene aperto un **dettaglio del gruppo di attività** ivi presenti.

Per la sezione delle attività assegnate e programmate il dettaglio è rappresentato tramite un Calendario (dojo),

Stampa Programmazione Settimanale

Risorse	22/09/2016	23/09/2016	24/09/2016	25/09/2016
Senza risorsa	0 0	0 0	0 0	0 0

Attività assegnate e programmate

BELLINI ALFREDO - 23/09/2016

2016 23/set/2016

02:00

02:30

03:00 **00097/2016 - Normale (2+0)**

03:30 Inizio previsto: 23/09/16 02:30

04:00 Fine prevista: 23/09/16 04:30

04:30 Risorsa: BELLINI ALFREDO asa

05:00

05:30

06:00

Precedente risorsa | Precedente data | Chiudi | Prossima data | Prossima risorsa

Mentre per le altre tre sezioni viene usata una lista.

Stampa Programmazione Settimanale

Risorse	22/09/2016	23/09/2016	24/09/2016	25/09/2016
Senza risorsa	0 0	0 0	0 0	0 0

Attività ne assegnate ne programmate

Senza risorsa

oggetti totali: 2

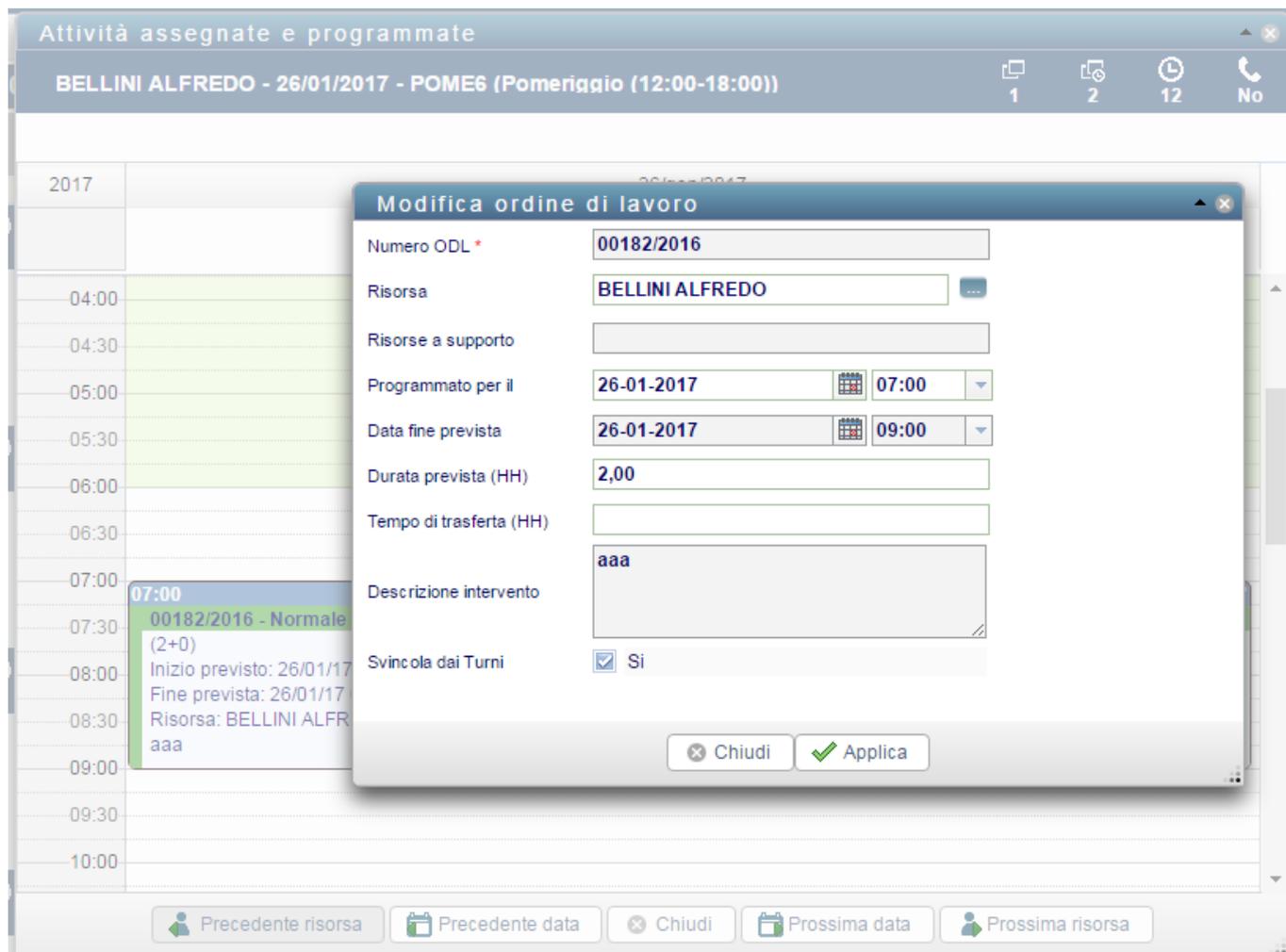
	Numero ODL	Priorità	Programmato per i	Data fine prevista	Descrizione intervento	Descr Resources Supp
<input type="checkbox"/>	00102/2016	!			hhh	...
<input type="checkbox"/>	00096/2016	!			sisisis	...

Chiudi

Scheda di dettaglio della singola attività

All'interno dei dettagli di gruppi di attività è possibile gestire la pianificazione delle singole attività, riprogrammando e/o riassegnando le stesse.

Ciò viene fatto dentro il **dettaglio della singola attività**. Questo è apribile con singolo click sulle righe, per i dettagli (di gruppo attività) delle celle che usano liste come forma di rappresentazione delle attività, o doppio click per i dettagli (di gruppo attività) delle celle contenenti attività programmate e assegnate.



La scheda di dettaglio della singola attività permette di cambiare a piacimento sia la risorsa assegnataria che la data di programmazione (e durata intervento). A seconda dei valori assunti dalla coppia risorsa/data, al click su 'Applica' l'attività potrà andare in una specifica sezione delle quattro viste precedentemente.

Nel caso in cui si scelga sia la risorsa che la data di programmazione, il sistema provvederà a calcolare in automatico la data di fine. In presenza del modulo dei turni, di default questa operazione viene fatta distribuendo sequenzialmente (fino ad esaurimento) le ore della durata dell'attività sulle fasce orarie dei turni programmati per una data risorsa (da una certa data in poi).

Questo viene fatto fino a quando non vengono esaurite le ore 'residue' dell'attività o non esistono turni programmati, o comunque non coprono l'entità della durata.

Se la durata è grande in misura per cui non è possibile distribuirla totalmente sulle fasce orarie dei turni già programmati l'utente viene notificato con un messaggio. L'utente viene anche notificato quando sceglie una data di inizio che sta fuori da tutte le fasce orarie dei turni relative alla risorsa scelta.

Per escludere questo comportamento è disponibile il check 'Svincola dai turni'. Quando attivo esso escluderà il meccanismo di calcolo della data di fine sopra descritto, e la data di fine verrà calcolata

semplicemente sommando la durata alla data di inizio. Inoltre ogni data di inizio sarà considerata valida, anche se fuori dai turni.

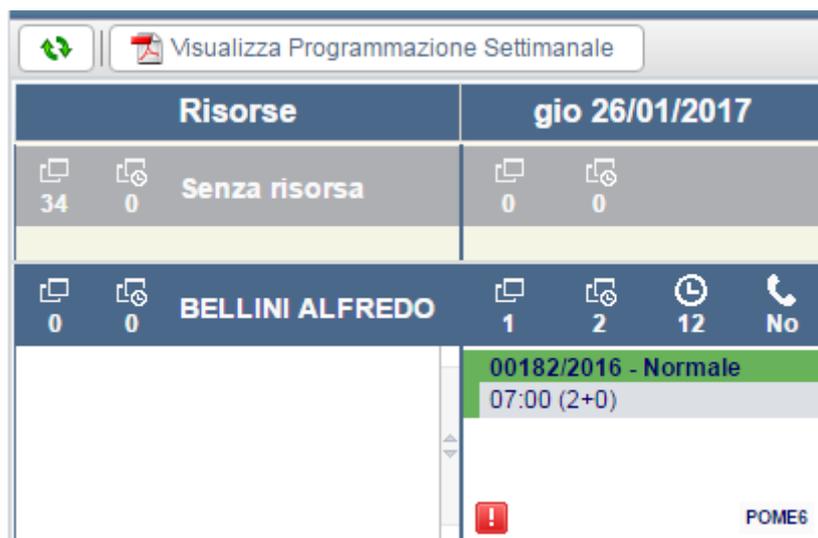
Da notare che a prescindere da come sia stata calcolata la data di fine, in caso di emerse incompatibilità con la programmazione sottostante essa viene 'preservata', anche eventualmente a danno della consistenza del valore della durata. Per esempio, se si effettua una programmazione di un'attività, non svincolandola dai turni, basandosi su una programmazione dei turni relativi ad una certa risorsa, indirettamente si sta fissando una data di fine (secondo le modalità descritte sopra). Se ora, tramite la gestione della programmazione dei turni, si va a modificare i turni della stessa risorsa, nella stessa settimana in cui si era programmato l'attività, si può generare un'inconsistenza. Tale inconsistenza sarà segnalata all'utente o tramite il meccanismo di segnalazione delle ore fuori dai turni (vedi sotto) e/o tramite l'ammontare di ore assegnate per la giornata, che sarà in genere più alto del giusto (ed eventualmente segnalato in rosso, se eccede il monte ore lavorative). Infatti preservando le variabili data inizio/data fine, la durata dell'attività sarà la mera differenza delle due, anche se originariamente la programmazione era stata fatta non svincolandola dai turni. Ne consegue che se, per esempio, l'attività aveva una durata per cui era stata originariamente spalmata sui turni di più giorni adesso, dopo che la programmazione sottostante è stata modificata, essa verrà mostrata nella scala temporale nel dettaglio dell'attività come un unicum che va dalla sua data di inizio e termina sulla sua data di fine, mostrando una durata potenzialmente superiore a quella presente sul DB.

Ore assegnate fuori dai turni

Nelle celle contenenti attività assegnate e programmate, il widget segnala eventualmente la presenza di ore (appartenenti ad una qualche attività) al di fuori delle fasce orarie individuate dai turni della risorsa.

Ciò viene ovviamente fatto solo in presenza del modulo dei turni.

Tale segnalazione è effettuata applicando un marker (punto esclamativo su sfondo rosso) nell'angolo in basso a sinistra della cella. Basta anche solo la presenza di una singola attività con ore fuori dai turni per quel giorno per far comparire il segnale. Se l'attività coinvolge più giorni, e presenta ore fuori dai turni in altri giorni, cioè verrà segnalato solo nelle celle di competenza dei giorni interessati.



Risorse	gio 26/01/2017
34 0 Senza risorsa	0 0
0 0 BELLINI ALFREDO	1 2 12 No
	00182/2016 - Normale 07:00 (2+0)

Questa segnalazione in genere si manifesta:

- Quando l'attività è segnata come 'svincolata dai turni' e le sue ore ricadono fuori dalle fasce orarie dei turni di quella risorsa
- In seguito ad una riprogrammazione dei turni per una data risorsa in una data settimana. Ciò

puo aver invalidato la programmazione delle attività (non svincolate dai turni) che era stata fatta interpolando i dati dell'attività (data di inizio, durata e data di fine risultante) con quelli dei turni programmati (fasce orarie ora inizio/ora fine)

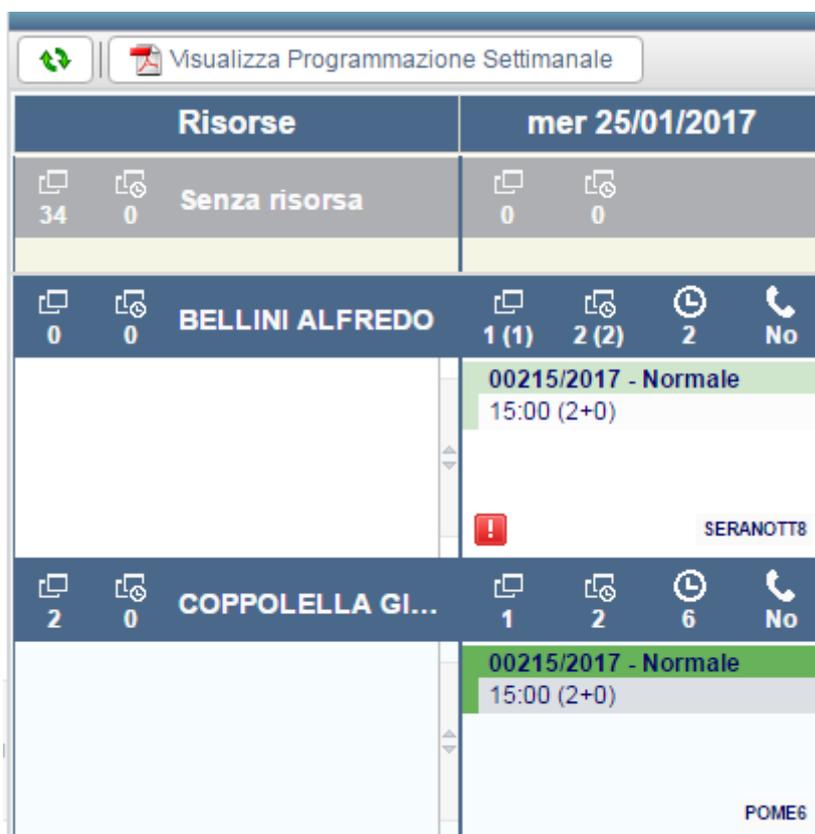
Attività in cui la risorsa è a supporto

Geoweb permette, esternamente al widget, di definire risorse a supporto per una data attività assegnata ad una risorsa principale.

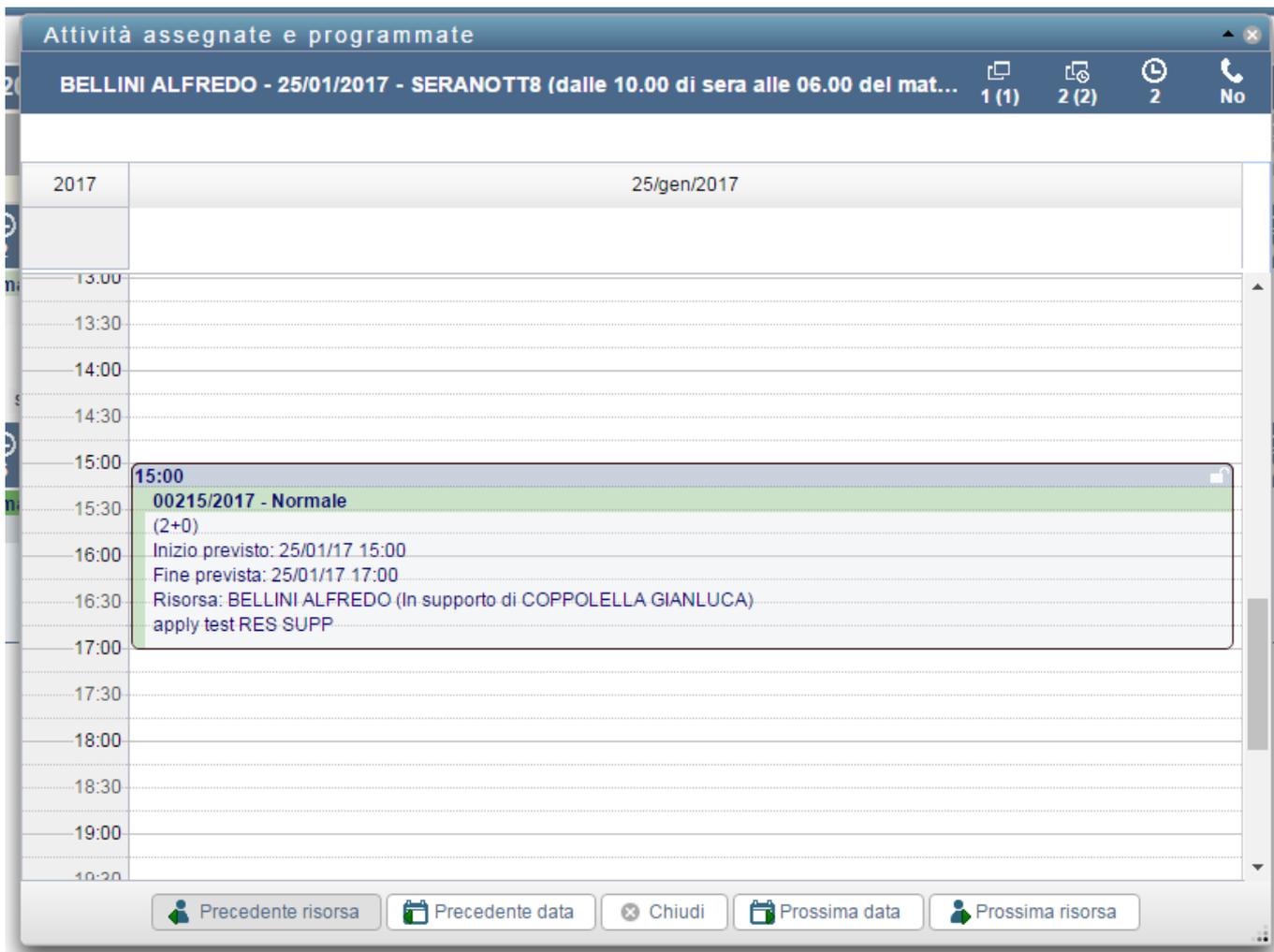
Nel widget vengono visualizzate e gestite anche le attività programmate per cui una risorsa è a supporto di un'altra. Queste vengono considerate nel computo del numero delle attività e delle ore assegnate ad una data risorsa in una data giornata.

In generale nel widget le attività in cui la risorsa è a supporto vengono visualizzate con una minore opacità dei colori di sfondo.

Per esempio, nell'immagine qui sotto, l'attività '00215/2017' risulta assegnata alla risorsa 'COPPOLELLA GIANLUCA', in data '25/01/2017'. Per questa attività la risorsa 'BELLINI ALFREDO' risulta a supporto. Infatti la stessa attività '00215/2017' compare anche nella sua cella per lo stesso giorno. Ciò viene denotato grazie alla differente gradazione dei colori, e al fatto che nel riepilogo vengono segnalate, tra parentesi, le attività e le ore a supporto (sul totale) di quella risorsa in quella giornata.



Qui un esempio di come viene renderizzata un'attività programmata in un dato giorno ed assegnata ad una certa risorsa, quando la si visualizza dal dettaglio calendario (risorsa/giorno) di un'altra risorsa, che è a supporto per quell'attività.



Parametri

- "className", obbligatorio nome della gwClass dove lavora la timeline
- "listedAttributesNames", opzionale. Lista di nomi di attribute di geoweb, separati da ','. Permette di controllare le colonne che vengono mostrate nelle liste della timeline. Se omesso verranno usati gli attributi di lista base della classe
- "gwdResources", opzionale, alternativo a gwdResourcesFilters, è una stringa con i valori gwd_resource, separati da virgola, dei record della tabella gwd_resource. Fra i valori può essere inserita la notazione #{gw_activeUser} che sarà sostituita con il valore dell'utente corrente in sessione
- "gwdResourcesFilters", opzionale, alternativo a gwdResources, è un array di oggetti filtro, che impostano criteri per la selezione di gwd_resource.
- "nDays", opzionale, valore di default 7, è un intero che rappresenta l'orizzonte temporale di programmazione attività che verrà mostrato all'utente quando aprirà la scheda gwTimeLine (parte sempre dal giorno risultante dall'impostazione del parametro fromTodayOffset).
- "fromTodayOffset", opzionale, valore di default 1, è un intero che rappresenta l'offset (in giorni) della prima data mostrata nel widget, rispetto ad oggi. Può essere anche negativo. Es: se vale 0 la prima data mostrata sarà oggi; se vale 1, domani.
- "useWorkshift", opzionale, valore di default false, flag che consente di far interagire la gwTimeLine con il modulo dei turni. Ciò comporta che alla modifica dei parametri temporali (inizio e durata o inizio e fine) di un'attività assegnata (ad una risorsa) comporti un' interpolazione con i turni della risorsa per quel giorno. In caso di incongruenze la modifica verrà negata segnalando la cosa con un messaggio di errore. In sostanza l'effetto con useWorkshift posto a true è che non sarà possibile assegnare un'attività ad una risorsa in un certo giorno

fuori dal suo turno di lavoro. Inoltre se posto a true, farà comparire un controllo di tipo 'check' nella scheda di modifica dell'attività (con label 'Svincola da turni'), che permetterà di escludere quell'attività dall'incrocio con i moduli dei turni, rendendola di fatto programmabile per qualsiasi giorno ed ora.

- “readonly”, opzionale, valore di default false, flag che consente di aprire la scheda in modalità di sola lettura. Ciò comporta che le schede di modifica delle attività saranno aperte in modalità visualizzazione, indipendentemente da tutti i permessi statici e dinamici impostati per la classe con nome className.
- “reportPath”, opzionale, String, default null, se presente sarà disponibile sulla toolbar un bottone che al click apre la report come da path, passando come parametro 'gwd_resources', String contenente una clausola fra parentesi tonde di valori (valori gwd_resource in ingresso al widget) String separati dal carattere ','. Esempio: gwd_resources: ('001','002').
- “openReportButtonLabel”, opzionale, String, default null, valutato solo se reportPath è presente. E' la label (e tooltip) applicata al bottone che apre la report. Se omessa viene mostrato un bottone senza label e con un generico 'Apri' localizzato come tooltip.
- “dayColumnWidth ”, opzionale, valore di default 120, è un intero che rappresenta la dimensione in px della larghezza della colonna giorno della griglia della gwTimeLine. Può essere impostato anche con il valore 'auto', in tal caso lo spazio a disposizione viene equamente diviso fra tutte le colonne giorno.
- I parametri “hourSize”, “timeSlotDuration”, “minHours”, “maxHours” sono ripresi direttamente fra quelli esposti da dojo, ed influiscono sulla grafica della scheda dettaglio delle attività assegnate e programmate in un dato giorno. Sono opzionali, e in caso di omissione sono utilizzati dei valori di default. Essi agiscono come descritto nella documentazione ufficiale visualizzabile qui.

Requisiti

- gwClass
 - jor_workshift_resource_wsinst (solo con modulo dei turni presente)
 - jor_odl
- groovy
 - jor_odl_trigger.groovy

JavaScript

```
var tabWidgetType = 'gwTimeLine';
var tabWidgetId = createTabId(tabWidgetType, 'activities_programmation');
var tabWidgetTitle = 'Programmazione Attività';

//ALTERNATIVA 1
var gwdResources = '001,002,003,004,005';
//ALTERNATIVA 2
//se lanciata da un azione di tipo 'List', gwdResourcesFilters è reperibile
dai filtri attualmente impostati sulla gwClassList, così
//var gwdResourcesFilters = parameterQueryList.filters;
//altrimenti si possono passare filtri così
//var gwdResourcesFilters = [
// {condition: 'AND', columnName: 'cod_resource', operator: '=', filterType:
'STRING', value: ['002']}
// {condition: 'AND', columnName: 'cod_resource', operator: 'IS NOT NULL',
filterType: 'STRING'}
```

```
//};

var parametersMap = {
  forceReplaceTab: true,
  className: 'jor_odl',
  detailLayoutName: 'who_n_when',
  listedAttributesNames:
'job_order_code,pr_type,pr_expected_start_date,pr_expected_end_date,pr_expec
ted_timspan,job_order_description',
  //ALTERNATIVA 1
  gwdResources: gwdResources,
  //ALTERNATIVA 2
  //gwdResourcesFilters: gwdResourcesFilters,
  nDays: '15',
  dayColumnWidth: '180',
  hourSize: '60',
  timeSlotDuration: '15',
  minHours: '8',
  maxHours: '18',
  fromTodayOffset: '1',
  useWorkshift: true, //workshift module is used to compute timeslots,
  readonly: false,
  reportPath: 'path/to/report.jasper',
  openReportButtonLabel: 'Stampa Programmazione Settimanale'
};

openTab(tabWidgetId, tabWidgetType, tabWidgetTitle, parametersMap);
```

LeafItem

E' un leafItem che al click apre la scheda di tipo gwTimeLine. Il type che caratterizza il leafItem è "gwTimeLine"

Codice da inserire dentro un accordionPanelItem nell'xml di progetto:

```
<leafItem name="test_time_line" label="Test gwTimeLine"
image="evt_avvisi_scadenza.png" type="gwTimeLine">
  <parameter name="className" value="jor_job_to_manage"
hideToClient="false"></parameter>
  <parameter name="detailLayoutName" value="who_n_when"
hideToClient="false"></parameter>
  <parameter name="listedAttributesNames"
value="job_order_code,pr_type,pr_expected_start_date,pr_expected_end_date,pr
_expected_timespan,job_order_description,address" hideToClient="false"
></parameter>
  <parameter name="gwdResources" value="001,002"
hideToClient="false"></parameter>
  <parameter name="nDays" value="15" hideToClient="false"></parameter>
  <parameter name="dayColumnWidth" value="120"
hideToClient="false"></parameter>
  <!--DAY DETAIL GHRAPHIC PARAMS STARTS-->
  <parameter name="hourSize" value="60" hideToClient="false"></parameter>
```

```

    <!-- The desired size in pixels of one hour, 100 default -->
    <parameter name="timeSlotDuration" value="15"
hideToClient="false"></parameter>
    <!-- The duration of minutes of the time slot (must be a divisor of 60),
15 default -->
    <parameter name="minHours" value="8" hideToClient="false"></parameter>
    <!-- in range [0, 23], 8 default-->
    <parameter name="maxHours" value="18" hideToClient="false"></parameter>
    <!-- in range [1, 36], 16 default-->
    <!--DAY DETAIL GHRAPHIC PARAMS ENDS-->
    <parameter name="fromTodayOffset" value="1"
hideToClient="false"></parameter>
    <parameter name="useWorkshift" value="true"
hideToClient="false"></parameter>
    <parameter name="readonly" value="false"
hideToClient="false"></parameter>
    <parameter name="reportPath" value="path/to/report.jasper"
hideToClient="false"></parameter>
    <parameter name="openReportButtonLabel" value="Stampa Programmazione
Settimanale" hideToClient="false"></parameter>
</leafItem>

```

Widget

Tipi di widget standard

Nelle maschere utente vengono utilizzati dei controlli (tecnicamente widget) che sono associati agli attributi delle classi. Ognuno di essi è configurabile attraverso una serie di parametri, che vengono poi salvati e mantenuti in formato xml. Le interfacce di amministrazioni degli attributi in alcuni casi guidano alla configurazione, ma nella maggior parte dei casi presentano il contenuto dell'XML direttamente modificabile in forma testuale. In questo paragrafo saranno elencati tutti i widget esistenti nella versione corrente, e descritti i parametri di configurazione attesi. Esistono dei parametri comuni a tutti i tipi di controllo, che saranno trattati in un'unica soluzione all'inizio del paragrafo.

tipologia	Nome	
	ACTION_WIDGET	
	ACTIVE_USER_WIDGET	
	ACTIVE_GROUP_WIDGET	
	SCOPE_VALUE_WIDGET	
	ATTACHMENTS	Controllo per gestire i documenti salvati in alfresco a partire da un dato folder; per utilizzare il widget è necessario che alfresco sia installato nel server, sia stata dedicata una cartella per l'applicazione e siano configurati i parametri di alfresco nel file di configurazione dentro i contenuti statici
	EXTERNALLINK	Controllo per inserire un link che punta ad un url esterno al sistema

tipologia	Nome	
	CHECKBOX	Casella con segno di spunta.
		Può assumere 3 valori: True, False, Null
	CMIS_DOCUMENT_WIDGET	
	COLORPICKER	
Data/Ora	DATE	Inserimento di una data attraverso uso del calendario.
		Inserimento opzionale dell'ora
	TIME	Inserimento dell'ora con widget tipo Spinner
Input guidato	COMBOBOX	Lista di scelta mono-selezione in cui i valori sono definiti internamente al widget
	DBCOMBOBOX	Lista di scelta mono-selezione in cui i valori sono recuperati dal database tramite query da configurare
	DBIMAGECOMBOBOX	
	DBSUGGESTBOX	
	DBWINDOWLIST	Lista di scelta mono-selezione in cui i valori sono recuperati dal database tramite query da configurare. Questo controllo permette di mostrare all'utente più colonne dei record da selezionare
		Non richiede che sia definita una classe sulla tabella o vista utilizzata nella query di selezione Viene mostrata con una finestra modale che si apre in pop-up
	DOCUMENT	Controllo per gestire l'inserimento un singolo documento, che viene salvato nel DB. Il documento viene selezionato tramite una finestra standard windows
	IMAGE	Controllo per gestire l'inserimento un file immagine che viene salvata nel DB Il controllo mostrerà direttamente l'immagine in fase di consultazione o modifica del record corrente nel form. L'immagine viene selezionata tramite una finestra standard windows
	DOMAIN	Controllo per la gestione di tabelle di dominio condivise
Etichette	LABEL	Inserisce un testo descrittivo non collegato ad alcun campo del database
Relazioni tra classi	LINKLIST	Presentazione di un elenco di record relazionati 1 a N rispetto alla classe corrente Il click sull'elenco darà accesso al record visualizzato nel form di dettaglio La navigazione si sposta alla classe collegata Il record di dettaglio si aprirà come scheda separata, in pop-up o docked, secondo la configurazione già definita per la classe collegata
Relazioni tra classi	LINKLISTNAM	Presentazione di un elenco di record relazionati N a M rispetto alla classe corrente Il click sull'elenco darà accesso al record visualizzato nel form di dettaglio La navigazione si sposta alla classe collegata Il record di dettaglio si aprirà come scheda separata, in pop-up o docked, secondo la configurazione già definita per la classe collegata

tipologia	Nome	
Relazioni tra classi	EXTERNALTABLE	Lista di scelta mono-selezione che permette di selezionare un record presente in una classe definita in GW Viene mostrata con una finestra modale che si apre in pop-up Permette di aggiungere un nuovo record nella classe utilizzata dalla lista
Relazioni tra classi	CHILDLIST	Presentazione di una anagrafica relazionata 1 a molti rispetto a quella corrente, inserita nel form come elenco di valori direttamente editabile riga per riga (inserimento, modifica cancellazione)
Data input	NORMAL	Casella di testo semplice
Data input	NUMBERBOX	Casella di testo con controlli dell'input specifici per campi numerici
Data input	TEXTAREA	Casella di testo estesa per gestione di campi descrittivi disposti su più righe di testo (xml o note)
Geometrie	POINT	Inserimento, visualizzazione, zoom su mappa di una geometria di tipo puntuale
	POLYGON	Inserimento, visualizzazione, zoom su mappa di una geometria di tipo polilinea
	POLYLINE	Inserimento, visualizzazione, zoom su mappa di una geometria di tipo poligonale
Posizione nella mappa?	POSITION_WIDGET	
	POSITION2_WIDGET	

I widget sotto elencati sono disponibili nella versione Enterprise Edition, e sono compresi in appositi plugin, eventualmente forniti con tale versione.

Plugin	Nome	Descrizione breve	Note
CALENDAR	FINAL_BALANCE_JOBS_WIDGET		
	SELECTION_LIST_WIDGET		
	LABEL_VALUES_XML_WIDGET		
	EVENT_OBJECT_LIST		
	SELECTION_LIST2_WIDGET		
WORKFLOW	ACTION_WORKFLOW_WIDGET		
CLASSIFICATION	CLASSIFICATION		
	VARIABLEATTRIBUTES		
	RELATION_WIDGET		

Altri controlli presenti

Tipo di controllo	Descrizione	Note
DATERANGE	Casella di testo per inserire un intervallo di dati	
LINK	Controllo per inserire un link ad un'altra anagrafica	
GEOCODING	Controllo che interagisce con il sistema di geocoding di Google visualizzando l'indirizzo più prossimo alla posizione dell'oggetto in mappa	

Matrice applicabilità tipo widget per tipo di attributo

	Nome widget	Non Definito	STRING	INTEGER	NUMBER	BOOLEAN	DATE	DOCUMENT	IMAGE	POINT	POLYGON	POLYLINE
base	ACTION_WIDGET	x										
	ACTIVE_USER_WIDGET		x									
	ACTIVE_GROUP_WIDGET		x									
	SCOPE_VALUE_WIDGET		x									
	ATTACHMENTS	x										
	CHECKBOX		x	x	x	x						
	CHILDLIST	x										
	CMIS_DOCUMENT_WIDGET	x										
	COMBOBOX		x	x	x	x						
	COLORPICKER		x	x	x							
	DATE							x				
	TIME							x				
	DBCOMBOBOX		x	x	x	x						
	DBIMAGECOMBOBOX		x	x	x	x						
	DBSUGGESTBOX		x	x	x	x						
	DBWINDOWLIST		x	x	x	x						
	DOCUMENT								x			
	IMAGE									x		
	DOMAIN	x										
	EXTERNALLINK	x										
	EXTERNALTABLE		x	x	x							
	LABEL	x										
	LINKLIST	x										
	LINKLISTNAM	x										
	NORMAL		x	x	x	x						
NUMBERBOX		x	x	x								
POINT									x			
POLYGON										x		
POLYLINE											x	
POSITION_WIDGET		x	x	x								
POSITION2_WIDGET	x											
TEXTAREA												
enterprise	FINAL_BALANCE_JOBS_WIDGET	x										
	SELECTION_LIST_WIDGET		x									
	LABEL_VALUES_XML_WIDGET	x										
	EVENT_OBJECT_LIST	x										
	SELECTION_LIST2_WIDGET		x									
	ACTION_WORKFLOW_WIDGET	x										
	CLASSIFICATION			x	x							
	VARIABLEATTRIBUTES	x										
RELATION_WIDGET	x											

Elenco dei Parametri comuni alla maggior parte degli widget

Nome	Descrizione breve
width	Larghezza in pixel del controllo visualizzato nel form
height	Altezza in pixel del controllo visualizzato nel form
defaultValue	Valore di default che compare solo in fase di inserimento
required	Impostabile a True o False per definire se l'attributo sia Obbligatorio nel form corrente. Nel caso sia impostato a True, non è possibile chiudere la maschera finché l'attributo non viene valorizzato
readonly	Impostabile a True o False per definire se l'attributo debba essere visualizzato in modalità di sola lettura

Nome	Descrizione breve
isDefaultOrderBy	Impostabile a True o False per definire se l'attributo debba essere utilizzato per ordinare la lista dei record presenti nella classe Per default è impostato a True, che significa che viene utilizzato il criterio standard di GW, applicato a tutti gli attributi. Occorre quindi impostare il parametro a False per 'sganciarlo' da tale criterio (1)
IsDefaultOrderByAscending	Impostabile a True o False per definire se l'attributo debba essere utilizzato per ordinare la lista dei record in ordine crescente (Ascending) Per default è impostato a True, che significa che viene utilizzato il criterio standard di GW, applicato a tutti gli attributi. Occorre quindi impostare il parametro a False per 'sganciarlo' da tale criterio (1)
isDefaultOrderByOnFieldToShow	Impostabile a True o False per definire se l'attributo debba essere utilizzato per ordinare la lista dei record in base al parametro FieldToShow definito. Per default è impostato a True, che significa che viene utilizzato il criterio standard di GW, applicato a tutti gli attributi. Occorre quindi impostare il parametro a False per 'sganciarlo' da tale criterio (1) Applicabile a controlli di tipo Elenco (DBCombobox, DbWindowList, ecc.)
listCellTextAlign	Imposta in modo rapido l'allineamento del valore del widget dentro la cella. Sovrascrive le impostazioni di base che avrebbe il widget (es: 'center' è il default degli widget 'Date' in lista) (2) Valori ammessi: 'center', 'left', 'right'
listCellStyleRules	Le regole css vengono applicate alle celle che ospitano il widget (2) Valori ammessi: qualsiasi regola css
listCellHeaderStyleRules	Le regole di stile vengono applicate all'header della colonna del widget (2) Valori ammessi: qualsiasi regola css
listCellClass	Le regole css definite dalla classe vengono applicate alle celle. La regola css deve essere già esistente fra quelle base (o essere stata in qualche modo aggiunta tramite i meccanismi dei plugin) (2) Valori ammessi: qualsiasi nome di classe css

(1) Nella modalità di visualizzazione Lista della classe, GW ordina normalmente in modo crescente (Ascending) utilizzando i primi 3 attributi presenti in lista. Usando i parametri isDefaultOrderBy... è possibile agire su questo criterio, definendo in autonomia per quali attributi debba essere ordinabile la lista, se crescente o decrescente, ecc.

(2) Gli attributi vengono presentati in Lista secondo dei criteri di stilizzazione standard GW. In particolari esigenze può essere utile renderizzare una colonna in maniera diversa dalle altre. Usando i parametri listCell... è possibile agire in tale senso. Ulteriori spiegazioni sono disponibili al paragrafo [Personalizzazione stile dei controlli in Lista](#)

Dettaglio singoli widget

Normal

Questo tipo di controllo è una semplice casella di testo dove viene editata e mostrata l'informazione. I parametri specifici sono i seguenti:

Parametro	Descrizione	Parametro xml
Width (px)	Larghezza in pixel della casella di testo	width

Parametro	Descrizione	Parametro xml
Height (px)	Altezza in pixel della casella di testo	height
Max Length	Lunghezza massima della stringa da inserire/modificare	maxLength
Default Value	Valore di default che compare solo in fase di inserimento	defaultValue
Scope Name	Nome dell'ambito. Il sistema visualizzerà nella scheda il valore dell'ambito corrente	scopeName
Required	Va indicato Si (il valore nella stringa xml è true) se si vuole che la valorizzazione dell'attributo sia obbligatoria altrimenti va selezionato No (false)	required possibili valori: true false
Read Only	Va indicato Si (il valore nella stringa xml è true) se si vuole che il controllo sia in sola lettura altrimenti va selezionato No (false)	readonly possibili valori: true false

Widget Speciali

CmisDocumentWidget

A partire dalla revision 5868 (2017-01-11) è stato corretto un problema che riguarda il mancato salvataggio dell'estensione dei file caricati nel documentale (es. Alfresco) con il widget CmisDocument. Inoltre è stata sviluppata una procedura per rinominare i file del documentale che mancano dell'estensione. GUIDA PER AGGIUNGERE L'ESTENSIONE AI NOMI DEI FILE DEL DOCUMENTALE: - In webadmin:

1. creare un'azione (tipologia: 'classe') per la classe su cui è definito l'attributo CmisDocument.
2. il corpo dell'azione deve contenere il seguente codice javascript:

```
Azione: renameCmisDocumentAddExtension var params = {}; params.projectName = project_name;
params.className = 'class_xxx'; params.attributeName = 'attribute_xxx';
renameCmisDocumentAddExtension(params);
```

dove i parametri da variare sono: - class_xxx -> nome della classe (obbligatorio) - attribute_xxx -> nome dell'attributo CmisDocument (opzionale). - project_name -> nome del progetto corrente (opzionale). La variabile project_name contiene già il nome del progetto corrente e non c'è bisogno di modificarlo La riga: params.attributeName = 'attribute_xxx' è opzionale. Se omessa il sistema cercherà tra tutti gli attributi della classe class_xxx un attributo di tipo CmisDocument. es.
renameCmisDocumentAddExtension({projectName:gw_project_name,className:'class_xxx',attributeName:'attribute_xxx'});

Position2Widget

Il widget Position2 permette di memorizzare l'informazione della posizione di un record di una classe. La posizione è rappresentata dal codice di un site o di un building o di una room o di una workstation o di una generic position. Poiché esiste una gerarchia tra questi oggetti, scegliendo uno di questi elementi verranno automaticamente popolati i campi gerarchicamente collegati. Nella tabella della classe devono essere presenti 5 campi (nomi fissi): **(CODE_COLUMN_LIST_POSITION) "cod_site" , "cod_building" , "cod_room" , "cod_workstation" , "cod_generic_position"**

Nel widget è possibile specificare i nomi delle 5 classi per la posizione (DEFAULT_CLASS_NAME_LIST) (di default sono: **“gwd_site”** ,**“gwd_building”** ,**“gwd_room”** ,**“gwd_workstation”** ,**“gwd_generic_position”**) e il nome del 'campo codice' per ognuna delle 5 classi (CODE_COLUMN_LIST) (di default sono: **“site_code”** ,**“cod_building”** ,**“cod_room”** ,**“cod_workstation”** ,**“cod_position”**).

Per il corretto funzionamento del widget sono fissi i seguenti campi:

DEFAULT_COLUMN_FILTER_NAME (attributi utilizzati per i filtri) = {“site_code” ,“type_building” ,“room_number” ,“type_workstation” ,“cod_position”};

DESCRIPTION_COLUMN_LIST = {“description” ,“descr_building” ,“description” ,“name” ,“cod_position”};

NAME_COLUMN_LIST = {“name_site” ,“name_building” ,“description” ,“name” ,“cod_position”};

TABELLA RIASSUNTIVA:

CLASSE DEL WIDGET					
“colonne codice” (fisse)	cod_site	cod_building	cod_room	cod_workstation	cod_generic_position
CLASSI COLLEGATE					
nomi delle classi (parametro del widget)	gwd_site	gwd_building	gwd_room	gwd_workstation	gwd_generic_position
“colonne codice”(parametro del widget)	site_code	cod_building	cod_room	cod_workstation	cod_position
“colonne nome” (fisse)	name_site	name_building	description	name	cod_position
“colonne descrizione” (fisse)	description	descr_building	description	name	cod_position

é possibile nascondere uno o più di questi campi attraverso la proprietà 'classToHide' del widget (specificando i nomi delle classi da nascondere).

La proprietà 'showZoomButtonList' permette di visualizzare/nascondere il bottone 'zoom' per ognuna delle classi.

```
<Position2Widget>
  <width>341</width>
  <height>18</height>
  <maxLength>300</maxLength>
  <defaultValue></defaultValue>
  <required>>false</required>
  <readonly>>false</readonly>
  <isDefaultOrderBy>>true</isDefaultOrderBy>
  <isDefaultOrderByAscending>>true</isDefaultOrderByAscending>
  <isDefaultOrderByOnFieldToShow>>true</isDefaultOrderByOnFieldToShow>
  <listCellTextAlign></listCellTextAlign>
  <listCellStyleRules></listCellStyleRules>
```

```
<listCellClass></listCellClass>
<listCellHeaderStyleRules></listCellHeaderStyleRules>
<importCSVWithoutDecoding>>false</importCSVWithoutDecoding>
<classToHide>aec_workstation,gwd_position</classToHide>
<classList>
  <string>gwd_site</string>
  <string>gwd_building</string>
  <string>aec_room</string>
  <string>aec_workstation</string>
  <string>gwd_position</string>
</classList>
<codeColumnList>
  <string>site_code</string>
  <string>cod_building</string>
  <string>room_code</string>
  <string>cod_workstation</string>
  <string>code_position</string>
</codeColumnList>
<showZoomButtonList>
  <boolean>>false</boolean>
  <boolean>>false</boolean>
  <boolean>>true</boolean>
  <boolean>>true</boolean>
  <boolean>>true</boolean>
</showZoomButtonList>
</Position2Widget>
```

SelectionListWidget

```
HashMap<String,Object> selectionMap = new HashMap<>();
```

```
HashMap<String,Object> map0 = new HashMap<>(); map0.put("tableName", "gwd_site");
map0.put("columnType", "type_site"); map0.put("columnCod", "site_code");
map0.put("columnDescr", "description"); map0.put("columnName", "name_site");
map0.put("parent", ""); map0.put("parentCod", ""); map0.put("gwRadioClassIndex", 0);
```

```
HashMap<String,Object> map1 = new HashMap<>(); map1.put("tableName", "view_gwd_building");
map1.put("columnType", "type_building"); map1.put("columnCod", "cod_building");
map1.put("columnDescr", "descr_building"); map1.put("columnName", "name_building");
map1.put("parent", "gwd_site"); map1.put("parentCod", "cod_site"); map1.put("gwRadioClassIndex",
1);
```

```
HashMap<String,Object> map2 = new HashMap<>(); map2.put("tableName", "view_gwd_system");
map2.put("columnType", "type_system"); map2.put("columnCod", "cod_system");
map2.put("columnDescr", "descr_system"); map2.put("columnName", "name_system");
map2.put("parent", "gwd_building"); map2.put("parentCod", "cod_building");
map2.put("gwRadioClassIndex", 2);
```

```
HashMap<String,Object> map3 = new HashMap<>(); map3.put("tableName", "view_gwd_item");
map3.put("columnType", "class_fullpath"); map3.put("columnCod", "cod_item");
```

```
map3.put("columnDescr", "descr_item"); map3.put("columnName", "name_item");  
map3.put("parent", "gwd_system"); map3.put("parentCod", "cod_system");  
map3.put("gwRadioClassIndex", 3);
```

```
HashMap<String,Object> map4 = new HashMap<>(); map4.put("tableName", "view_gwd_assets");  
map4.put("columnType", "class_asset"); map4.put("columnCod", "cod_asset");  
map4.put("columnDescr", "descr_asset"); map4.put("columnName", "name_asset");  
map4.put("parent", "gwd_assets"); map4.put("parentCod", "cod_system");  
map4.put("gwRadioClassIndex", 4);
```

```
this.selectionMap = new HashMap<>(); selectionMap.put("gwd_site", map0);  
selectionMap.put("gwd_building", map1); selectionMap.put("gwd_system", map2);  
selectionMap.put("gwd_item", map3); selectionMap.put("gwd_assets", map4);
```

From:

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

Permanent link:

<https://wiki.geowebframework.com/doku.php?id=gwusermanual:interface&rev=1652199284>

Last update: **2022/05/10 18:14**

