

Tablehandler

Breve bigino

Tipologie

- dialogTableHandler
- borderTableHandler
- stackTableHandler
- paletteTableHandler
- plainTableHanlder

Parametri diretti

- viewResource
- formResource
- table, condition, condition_* /
relation (suggerimento: relation_name nel model)
- dialog_* /palette_*
- default_*
- (nodeld e datapath... nel caso ci siano + tablehandler afferenti alla stessa table nella stessa pagina)

Mappa

*Ogni tablehandler è da considerarsi come un macrocomponent
contenente diverse entità logiche.*

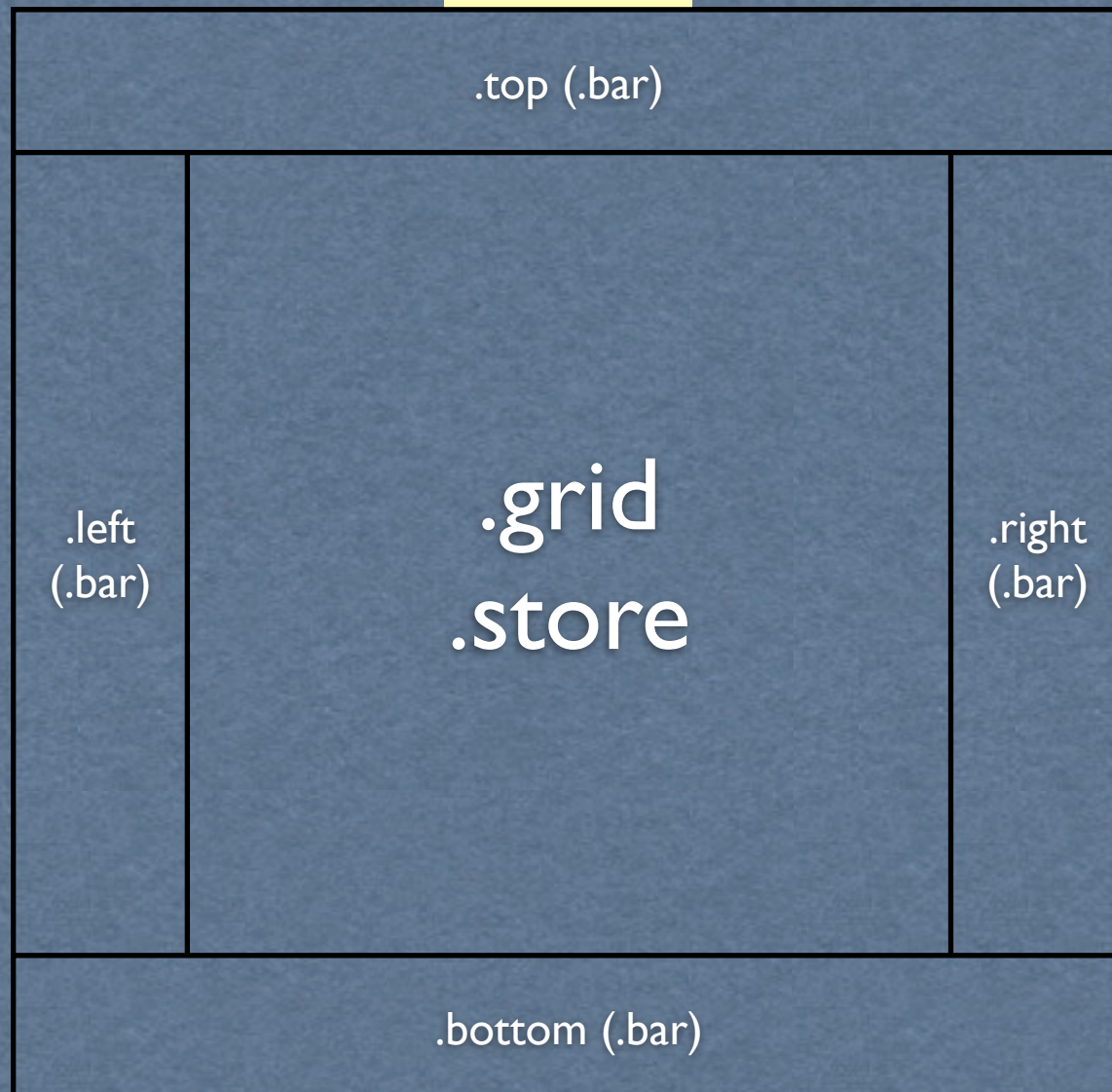
*Pertanto, in seguito alla definizione di un tablehandler è possibile
e spesso opportuno andare a customizzarei singoli sottoblocchi
accedendo ad essi tramite un percorso puntato.*

Eg: th.view.top

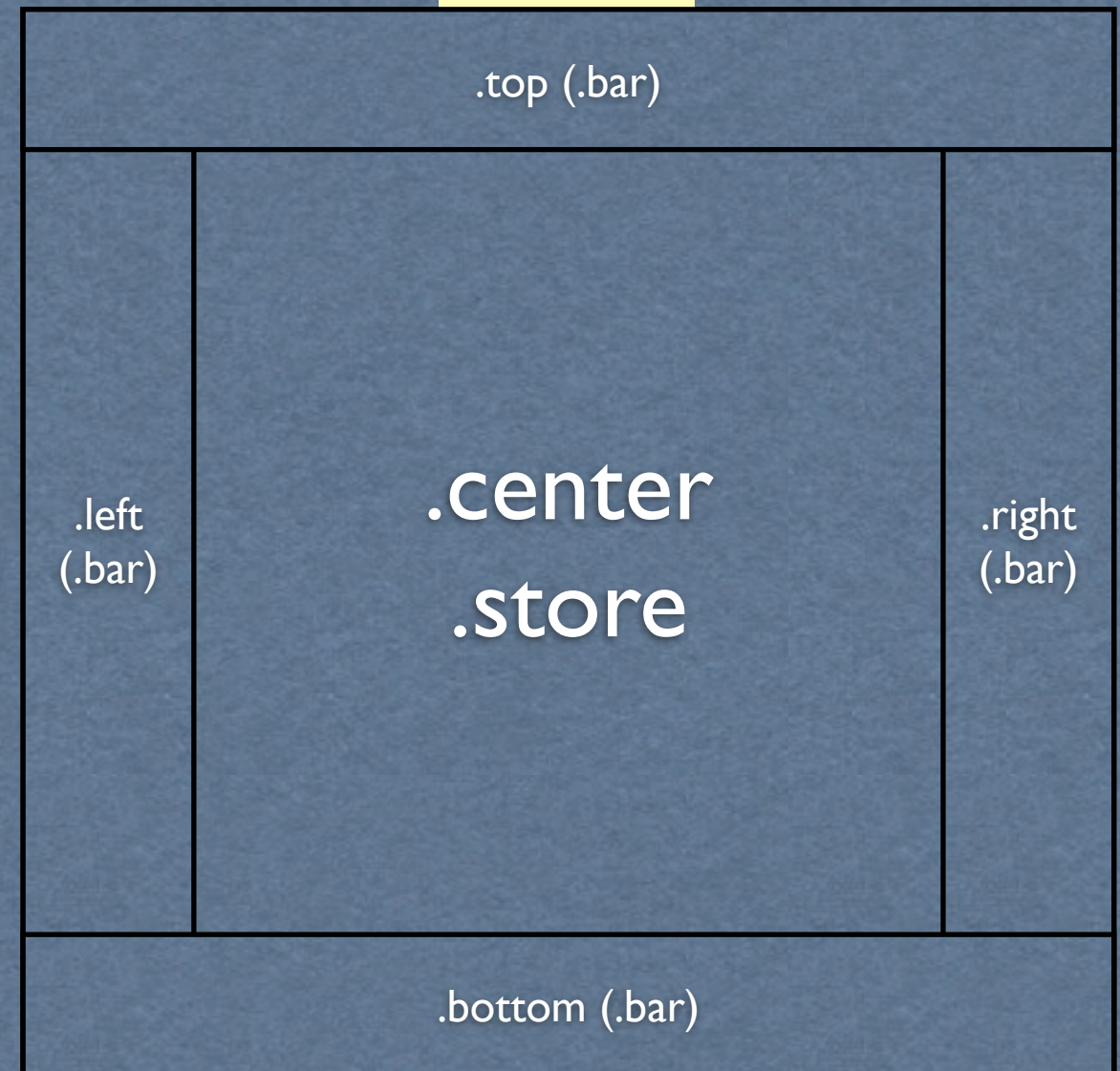
Mappa childname

TableHandler

.view



.form



Nb: Nel caso sia presente una toolbar o slotbar netop,bottom,left,right questo automaticamente aggiunge un livello .bar

Alcuni esempi

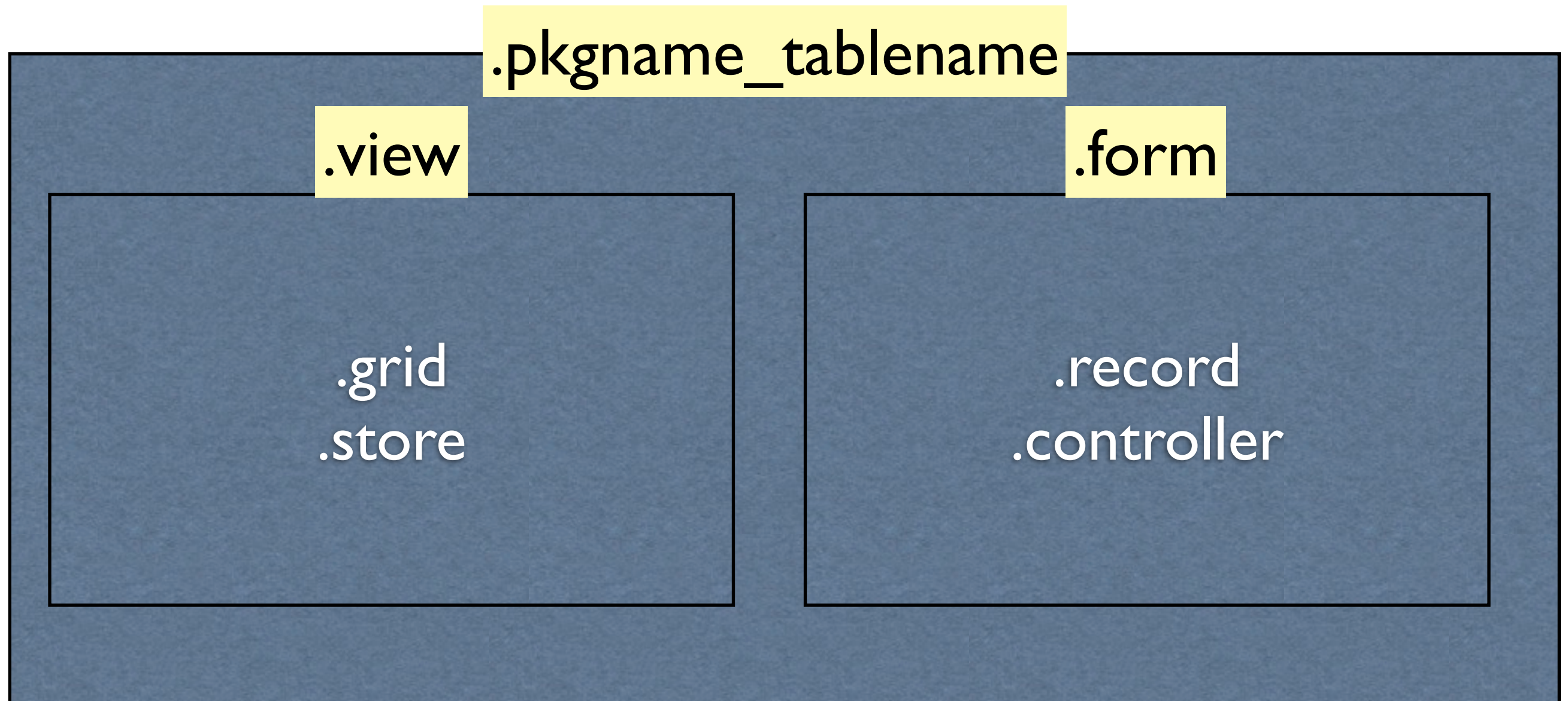
Recuperare dei sottoblocchi

- **il SelectionStore:** *th.view.store*
- **Griglia:** *th.view.grid*
- **Toolbar della form:** *th.form.top.bar*

Nb:

Ogni sottoblocco è un elemento complesso con il suo insieme di regole. Pertanto verranno spiegate le possibili configurazioni in seguito

Mappa dati



Consiglio aprire l'explorer dati per vedere in dettaglio

Risorse: tipo View: hook

- `th_struct(struct)`
- `th_query`
- `th_condition`
- `th_order`
- `th_top_*,th_bottom_*,th_left_*,th_right_*`
- `th_options`
 - *`excludeLogicalDeleted(bool)`*
 - *`excludeDraft(bool)`*
 - *`tableRecordCount(bool)`*

Risorse: tipo Form: hook

- `th_form(form)`
- `th_top_*,th_bottom_*,th_left_*,th_right_*`
- `th_options`
 - *`showtoolbar(bool)`*
 - *`showfooter(bool)`*
 - *`navigation(bool)`*
 - *`readOnly(bool)`*
 - *`linker(bool)`*
 - *`selector(bool)`*
 - *`lockable(bool)`*
 - *`navigation(bool)`*
 - *`modal(bool)`*

Customizzazioni

Forse non tutti sanno che...quando usi la property attributes in un nodo della struttura descrittiva di una pagina, si ha un dizionario. Pertanto per customizzare agevolmente le sottoparti del nostro tablehandler è utile avvalersi di questa proprietà.

Un po' di esempi...

- `th.view.store.attributes.update(_onStart=True,...)`
- `th.view.grid.attributes.update(onDrag= '...')`
- `th.form.attributes.update(rounded=6)`

Customizzazioni: Toolbar

`th.view.top.bar.replaceSlots('addrow','picker')`

NB se si usa # si intende tutti gli slot

eg: `th.view.top.bar.replaceSlots('#','foo')`

consiglio quando non sapete cosa contiene una slotbar

`print x`. verificate gli attributes e vedrete gli slot presenti

E' possibile customizzare una toolbar anche in risorsa sfruttando

gli hook `th_top_`, `th_bottom_*` etc...*

E' conveniente fare così quando si vuole raccogliere a fattore comune la customizzazione

Lo store della form

Lo store della form ha da possibilità di aggiungere degli handler per interagire sulle fasi di load,save e delete.

- `th.form.store.handler('load',default_foo='=..')`
ricordarsi che se ci si riferisce ad un dato della form padre (tipo la form è di riga_fattura e il default è un dato di fattura ad conviene usare usare il path #FORM/parent/#FORM.record per arrivare al record della form padre(fattura)
- `th.form.store.handler('save',onSaving='function(kw){}')`

Nb:

ogni chiamata di handler, allo stesso modo di una dataRpc ritorna un oggetto al quale è possibile appendere un addCallback