Architettura 0.01

William Simoni

08 February 2020

Contents

1 Architettura generale

2 Architettura Wizard

La maggior parte degli oggetti sono una sottoclasse di Element. Gli oggetti Element a loro volta vengono inseriti all'interno di una Categoria. Attualmente esistono la categoria VisualInterface e ConversationalInterface. La Categoria VisualInterface contiene elementi utili per la creazione di GUI mentre la Categoria ConversationalInterface contiene elementi utili per la creazione di Conversational Systems.

2.1 Classe Prop

2.1.1 Proprietà di Prop

- id: id della Prop.
- type: lista di tipi. Solo gli elementi con questi tipi possono essere memorizzati in value.
- value: Lista di Elementi o Stringhe associati alla Prop. Tutti gli oggetti di value devono avere lo stesso type.
- merge: Booleano utilizzabile nella getProp di Element.
- createFile: Booleano utilizzabile nella evalElement di Element.
- cardinality: numero di elementi che possono essere memorizzati all'interno di value.

L'unico campo modificabile di Prop è value. Tutti li altri sono immodificabili.

2.2 Classe Style

2.2.1 Proprietà di Style

- id: id di una classe di oggetti Style. Ci possono essere più oggetti Style con lo stesso id.
- type: indica a quale elemento è destinato l'oggetto Style. Non ci possono essere più oggetti Style con lo stesso id, lo stesso type e la stessa classe. Per esempio, type può assumere valore Button o Column.
- values: Mappa in cui c'è un'associazione tra proprietà stilistiche e valore associato a tali proprietà

L'unico campo modificabile di Style è values. Tutti gli altri sono immodificabili. Tutti gli oggetti Style con il medesimo id vengono memorizzati nel file con nome id+".yaml" nella cartella Styles come mostrato nell'immagine sottostante.

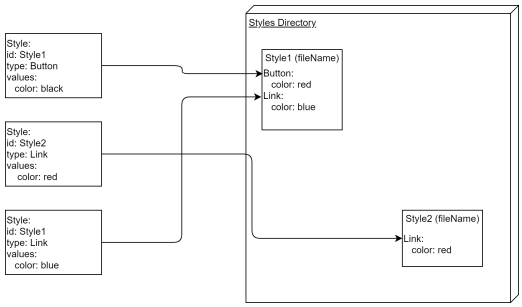


Figure 1: Associazione tra oggetto Style e file creato dal Wizard quando il progetto viene salvato

2.3 Classe Element

2.3.1 Proprietà di Element

- id: id per identificare l'elemento. All'interno di una categoria, tutti gli oggetti Elementi devono avere un id diverso.
- props: insieme delle proprietà strutturali dell'elemento. Associate a uno o più oggetto Prop.
- style: insieme delle proprietà stilistiche dell'elemento. Associate a uno o più oggetti Style. L'oggetto può contenere un solo oggetto Style di un determinato tipo (campo type di Style). Per esempio, se un oggetto è associato a un oggetto Style di tipo Button, allora non può essere associato a un ulteriore oggetto Style di tipo Button. E' però possibile sostituire l'oggetto Style di tipo Button, con un nuovo oggetto Style ti tipo Button.

2.3.2 Metodi di Element

- **evalElement()**, ritorna un oggetto Evaluated Element che permette di generare la configurazione YAML finale.
- **editProp(id della Prop : [String], valore da associare alla Prop : [Element/String], overWrite: Boolean)**, Se OverWrite è true: se l'oggetto ha una prop con l'id indicato,allora associa a tale prop il valore passato come parametro a patto che superi i controlli del typeChecker. Viceversa se OverWrite è false: se l'oggetto ha una prop con l'id indicato,allora associa a tale prop il valore passato come parametro a patto che superi i controlli del typeChecker e che alla prop non sia già associato un altro valore. Ritorna true in caso di inserimento avvenuto con successo, false altrimenti.
- **deleteChild(id della Prop : [String], Prop in cui si trova il figlio : id dell'elemento da eliminare dal campo value di Prop : [String])**, Elimina dal campo value della prop indicata, l'elemento con l'id indicato.
- **addStyle(StyleObj: [Style], overWrite: Boolean)**, Se overWrite è false allora StyleObj viene aggiunto alla proprietà style se l'elemento non contiene oggetti Style con lo stesso Type. Se overwrite è true, allora StyleObj viene aggiunto in ogni caso. Se è già presente un oggetto con lo stesso Type di StyleObj, allora questo viene sovrascritto dal nuovo oggetto. Ritorna true in caso di inserimento avvenuto con successo, false altrimenti.
- **getProp(id della Prop : [String], merge: Boolean)** ricerca le prop con l'id indicato. Se merge è true, la ricerca avviene anche nei sotto elementi. Ritorna una struttura dati in cui è presenta un'associazione tra id dell'elemento in cui è contenuta la Prop e id della Prop.
- **getProp(type della Prop : [String], merge: Boolean)** ricerca **tutte** le prop con il type indicato. Se merge è true, la ricerca avviene anche nei sotto elementi. Ritorna una struttura dati in cui è presenta un'associazione tra id dell'elemento in cui è contenuta la Prop e id della Prop.
- **clone()** ritorna una DEEP COPY dell'oggetto Element.

3 sottoclassi di Element generali

3.1 Data

Un oggetto Data contiene le informazioni necessarie per ottenere dei dati.

- type: Data

- Props:
  - tag: tag per interrogare l'ADM
  - device: id di un device.
  - types: (Temperature, Pression, ...)
- Style: non definito.

Abbiamo definito data supponendo che i dati vengano ricevuti nel formato Tag:{device:{type1:{},type2:{},...}}.

4 sottoclassi di Element e di Style in VisualInterface

Per ogni elemento viene indicato nella sezione Style, l'oggetto di tipo Style specifico per quel determinato elemento. Come menzionato prima, ad un oggetto di tipo Element possono essere associati più oggetti di tipo Style. In un progetto può anche essere completamente ignorata la generazione di oggetti Style. In tal caso, verranno applicate le regole di default dei generatori.

4.1 Root

L'elemento Root indica al generatore da dove iniziare la generazione delle varie pagine del progetto.

- Type: Root
- Props:
  - children: I figli della Root devono essere obbligatoriamente elementi di tipo Page.
- Style: L'elemento Root non è visibile e non ha un elemento Style specifico.

4.2 Page

- Type: Page
- Props:
  - layout: Il figlio di Page può essere un qualsiasi oggetto Element. Può avere un solo figlio. Questa prop ha le proprietà **merge = true** e **createFile = true**.
  - url: Stringa che indica l'url della pagina.
  - subpages: Elementi di tipo Page. Rappresentano sottopagine di Page.

  - pars: lista di parametri. Quando viene linkata la pagina, è necessario indicare dopo l'url effettivo della pagina, anche un ? seguito da $nomeParametro=value.

- Style: L'elemento Page non ha un elemento Style specifico.

4.3 Elementi di tipo Layout

4.3.1 Container

L'elemento Container è un elemento di forma quadrata che consente di associare al figlio, un padding, delle restrizioni, e un margine.

- Type: Container
- Props:
  - child: Il container può avere al più un figlio di tipo Element.
- Style: l'oggetto style ContainerStyle ha le seguenti proprietà:
  - backgroundColor: Colore di background.
  - borderColor: colore del bordo del container.
  - borderWidth: Spessore del bordo del container.
  - maxWidth: massima larghezza del figlio del Container.
  - minWidth: minima larghezza del figlio del Container.
  - maxHeight: massima altezza del figlio del Container.
  - minHeight: minima altezza del figlio del Container.
  - width: larghezza esatta del container. Rende inutile la definizione di max/min Width.
  - height: altezza esatta del container. Rende inutile la definizione di max/min Height.
  - paddingTop: il padding applicato al Container in alto. Può essere espresso sia in termini di pixel che in termini di percentuale.
  - paddingRight: il padding applicato al Container a destra. Può essere espresso sia in termini di pixel che in termini di percentuale.
  - paddingLeft: il padding applicato al Container a sinistra. Può essere espresso sia in termini di pixel che in termini di percentuale.
  - paddingBottom: il padding applicato al Container in basso. Può essere espresso sia in termini di pixel che in termini di percentuale.
  - marginTop: il margine applicato al Container in alto. Può essere espresso sia in termini di pixel che in termini di percentuale.
  - marginRight: il margine applicato al Container a destra. Può essere espresso sia in termini di pixel che in termini di percentuale.

- marginLeft: il margine applicato al Container a sinistra. Può essere espresso sia in termini di pixel che in termini di percentuale.
- marginBottom: il margine applicato al Container in basso. Può essere espresso sia in termini di pixel che in termini di percentuale.

Nota: E' necessario un override del metodo editProp.
Nel seguito verranno usati i termini asse principale e asse perpendicolare.
In una colonna, l'asse principale è l'asse y, mentre l'asse perpendicolare è l'asse x. Viceversa con una riga.

4.3.2 Column

Ogni elemento Column rappresenta una colonna. Ogni elemento Column deve essere "wrappabile".

- Type: Column
- Props:
  - children: i figli della Column devono essere di tipo Element.
- Style: l'oggetto Style ColumnStyle può assumere i seguenti valori
  - direction: Stringa che indica la direzione in cui mostrare gli elementi. Può assumere due valori, reverse o forward. Se il valore è forward, gli elementi vengono mostrati dall'alto verso il basso, viceversa il contrario.
  - mainAxisAlignment: Stringa che indica il posizionamento degli elementi sull'asse principale. Può assumere i valori **center** (posizionamento al centro), **start** (posizionamento in alto), **end** (posizionamento in basso), **spaceEven** (elementi equidistanti), **baseline** (allineamento lungo la baseline), **spaceBetween** e **spaceAround**.
  - crossAxisAlignment: Stringa che indica il posizionamento sull'asse perpendicolare. Può assumere li stessi valori di MainAxisAlignment.
  - mainAxisSize: dimensione dell'asse principale. Di default cerca di occupare tutto lo spazio disponibile.
  - crossAxisSize: dimensione dell'asse perpendicolare. Di default cerca di occupare il minimo spazio possibile.

4.3.3 Row

Ogni elemento Row rappresenta una riga. Ogni elemento Row deve essere "wrappabile".

- Type: Row
- Props:

  - children: i figli della Row devono essere di tipo Element.

- Style: l'oggetto Style RowStyle può assumere i seguenti valori
  - Direction: Stringa che indica la direzione con cui mostrare gli elementi. Può assumere due valori, reverse o forward. Se il valore è forward, gli elementi vengono mostrati da sinistra verso destra, viceversa il contrario.
  - MainAxisAlignment: Stringa che indica il posizionamento degli elementi sull'asse principale. Può assumere i valori **center** (posizionamento al centro), **start** (posizionamento a sinistra), **end** (posizionamento a destra), **spaceEven** (elementi equidistanti), **baseline** (allineamento lungo la baseline), **spaceBetween** e **spaceAround**.
  - CrossAxisAlignment: Stringa che indica il posizionamento sull'asse perpendicolare all'asse principale. Può assumere li stessi valori di MainAxisAlignment.
  - mainAxisSize: dimensione dell'asse principale. Di default cerca di occupare tutto lo spazio disponibile.
  - crossAxisSize: dimensione dell'asse perpendicolare. Di default cerca di occupare il minimo spazio possibile.

4.3.4 Flexible

All'interno delle Row e delle Column è possibile definire elementi Flexible.

- Type: Flexible
- Props:
  - child: Può contenere al massimo un oggetto di qualsiasi tipo.
  - flex: stringa che indica, in proporsione quanto deve occupare l'elemento all'interno della Row o della Column.
- Style: non ha stili specifici

Se gli elementi non sono flexible, allora i vari elementi si distribuiranno lo spazio equamente.

4.3.5 Scaffold

Implementa il layout di default di una pagina.

- Type: Scaffold
- Props:
  - topNavBar: Elemento di tipo TopNavigationBar. Viene posizionato in alto. Può contenere al massimo una TopNavigationBar.

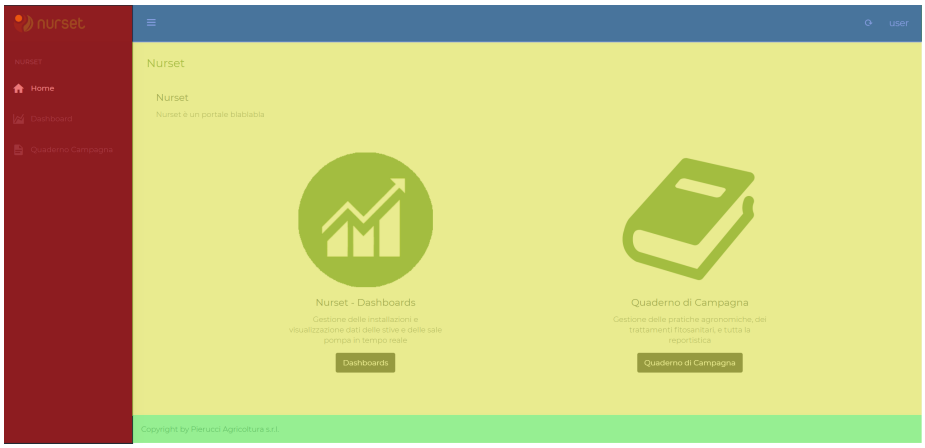


Figure 2: Nell'area rossa il drawer, nell'area blu la topNavigationBar, nell'area verde la bottomNavigationBar e nell'area gialla il body

  - bottomNavBar: Elemento di tipo BottomNavigationBar. Viene posizionato in basso. Può contenere al massimo una BottomNavigationBar.
  - drawer: Elemento di tipo Drawer. Menù a scomparsa che viene posizionato a sinistra. Può contenere al massimo un Drawer.
  - body: Body della pagina. Può contenere qualsiasi tipo di oggetto Element. Può contenere al massimo un Element.

- : Style: Scaffold non ha oggetti Style dedicati.

Quando un prop dello Scaffold viene modificato per la prima volta, l'Element Scaffold esegue a seconda del prop le seguenti operazioni:

- Se viene modificata drawer, crea un oggetto DrawerStyle, con position= left.

Nota: E' necessario un override del metodo editProp.

4.4 Elementi di Navigazione

4.4.1 Drawer

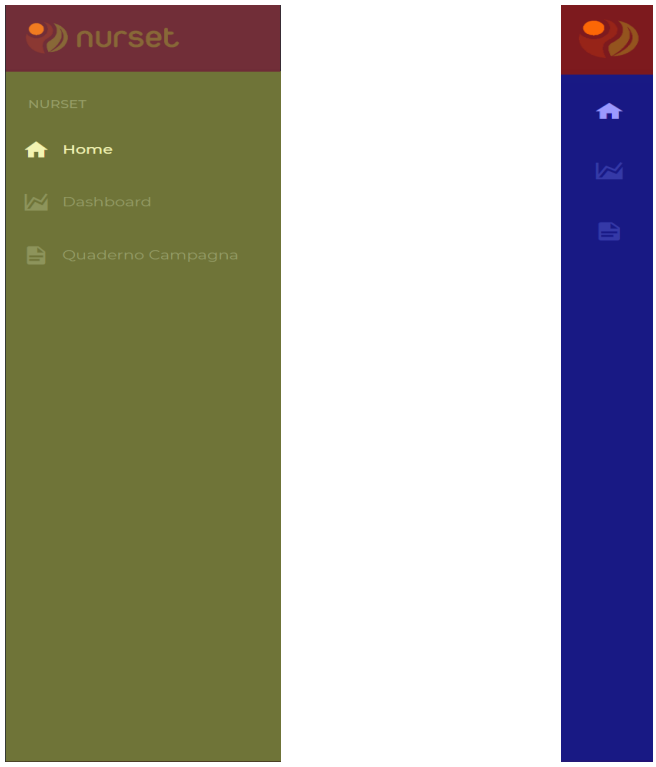
- Type: Drawer
- Props:
  - headerOpen: L'header è l'elemento che viene posizionato all'inizio del Drawer. Può essere UN qualsiasi oggetto di tipo Element. Header mostrato quando il drawer è aperto.

  - headerClosed: L'header è l'elemento che viene posizionato all'inizio del Drawer. Può essere UN qualsiasi oggetto di tipo Element. Header mostrato quando il drawer è chiuso.
  - listOpen: Elemento che contiene un elemento di tipo Lista. Questa lista viene mostrata quando il drawer è aperto.
  - listClosed: Elemento che contiene un elemento di tipo Lista. Questa lista viene mostrata quando il drawer è chiuso.

- : Style: l'oggetto Style DrawerStyle può assumere i seguenti valori
  - position: Indica la posizione da cui compare il Drawer. Può assumere i valori **right,left,top,bottom**.
  - fixed: se true, quando il drawer viene chiuso vengono mostrati gli elementi puntati da headerClosed e listClosed. Se headerClosed e listClosed non sono definiti, fixed è obbligatoriamente false. In tal caso, il drawer, quando chiuso, scompare del tutto.
  - color: Indica il colore del background del Drawer.

4.4.2 TopNavigationBar

- Type: TopNavigationBar
- Props:
  - header: Elemento che viene posizionato all'estrema sinistra della NavBar. Se ha valore nullo ed è presente un Drawer, allora l'header viene sostituito automaticamente dal simbolo standard del drawer.
  - title: Elemento di tipo Text. Può essere usato per dare un titolo alla pagina.
  - children: Lista di Elementi che compaiono lungo la NavigationBar dopo il title. Gli elementi possono essere Link, Button e DropDownElement.
- Style: l'oggetto TopNavigationBarStyle può assumere i seguenti valori:
  - backgroundColor: colore di background della barra di navigazione.
  - height: Larghezza della barra.
  - alignment: Stringa che indica il posizionamento orizzontale degli elementi nella navbar. Può assumere i valori **center** (posizionamento al centro), **start** (posizionamento a sinistra), **end** (posizionamento a destra), **spaceEven** (elementi equidistanti), **baseline** (allineamento lungo la baseline), **spaceBetween** e **spaceAround**.
  - fixed: Booleano. Se false, quando avviene lo scroll verso il basso la barra scompare. Se true, la barra rimane sempre nella sua posizione.



(a) Drawer aperto (b) Drawer chiuso

Figure 3: a sinistra il drawer aperto in cui sono mostrati gli elementi headerOpen e listOpen, a destra il drawer chiuso in cui sono mostrati gli elementi headerClose e listClose

4.4.3 BottomNavigationBar

- Type: BottomNavigationBar
- Props:
  - children: Lista di Elementi che compaiono lungo la NavigationBar dopo il title. Gli elementi possono essere Link, Button e Text.
- Style: l'oggetto BottomNavigationBarStyle può assumere i seguenti valori:
  - backgroundColor: colore di background della barra di navigazione.
  - alignment: Stringa che indica il posizionamento orizzontale degli elementi nella navbar. Può assumere i valori **center** (posizionamento al centro), **start** (posizionamento a sinistra), **end** (posizionamento a destra), **spaceEven** (elementi equidistanti),

Figure 4: Nell'area rossa l'header, nell'area blu l'area destinata ai children. In questo caso non è presente nessun title

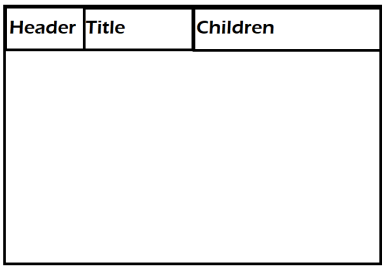


Figure 5: Suddivisione della TopNavigationBar

baseline (allineamento lungo la baseline), **spaceBetween** e **spaceAround**.

4.4.4 DropDownElement

Rappresenta un elemento che se cliccato mostra in una determinata direzione un nuovo elemento.

- Type: DropDownElement
- Props:
  - dropDownClickable: Elemento che se cliccato fa comparire il dropDownElement.
  - dropDownElement: L'elemento che compare in seguito al click.
- Style: oggetto DropDownButtonStyle può assumere i seguenti valori:
  - direction: Stringa che indica la direzione in cui compare l'elemento dropDownElement.

4.4.5 Link [Stile da definire]

Ogni elemento Link rappresenta un link.

- type: Link
- Props:

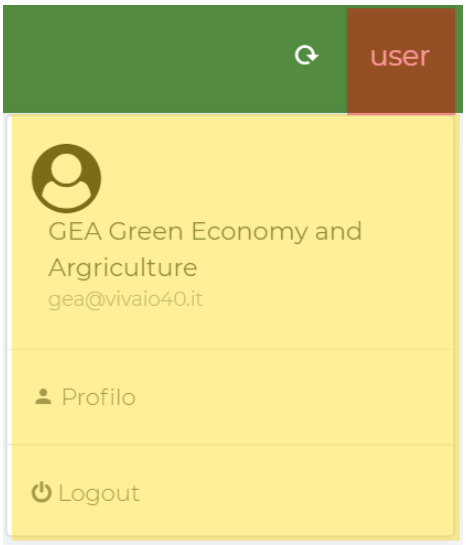


Figure 6: Nell'area rossa il dropDownClickable, nell'area gialla il dropDownElement

  - child: figlio dell'elemento Link.
  - url: l'url della pagina a cui far puntare il link.

- Style: da definire

4.5 Altri Elementi

4.5.1 Button

Ogni elemento Button rappresenta un bottone. L'elemento Button ha le seguenti proprietà:

- type: Button
  - children: elementi che sono mostrati all'interno del bottone. Al massimo possono essere due. Gli elementi devono essere di tipo Text o Icon.
  - url: url della pagina caricata se il bottone viene cliccato.
  - ??? onClick: contiene uno o più elementi di tipo Function.
- Style: l'oggetto ButtonStyle ha le seguenti proprietà:
  - size: dimensione del bottone.
  - color: colore del bottone.
  - type: tipo del bottone.(contained, text, outlined e update)

4.5.2 Text

Ogni elemento Text rappresenta un testo che può essere un paragrafo o un header.

- type: Text
- Props:
  - text: Stringa che indica il testo da mostrare. Può essere anche un elemento Data.
- Style: l'oggetto TextStyle può assumere i seguenti valori:
  - fontName: nome del font da utilizzare per la visualizzazione del testo.
  - fontSize: dimensione dei caratteri. E' possibile indicare i valori **xx-small, x-small, small, medium, large, x-large, xx-large** o un numero.
  - fontWeight: Indica lo spessore dei caratteri del testo. E' possibile indicare i valori **bold e normal**.
  - color: Colore del testo.
  - alignment: come viene allineato il testo. Può assumere i valori **center, left, right e justify**
  - textDecoration: utilizzato per nascondere le decorazioni associando il valore **none**, o per aggiungere delle descrizioni (**overline,line-through,underline)**.
  - letterSpacing: indica la distanza in pixel tra i caratteri.
  - wordSpacing: indica la distanza in pixel tra le parole.
  - lineHeight: distanza tra una linea e la successiva.
  - direction: direzione del testo. Può assumere i valori **rtl o ltr**.

Se all'interno di Text è contenuto un elemento Data, viene mostrato il valore ricevuto.

4.5.3 Icon

Un oggetto Icon rappresenta un'icona. Ogni icona è identificata da un nome. L'insieme dei nomi accettati dai generatori sarà redatto prima o poi.

- type: Icon
- Props:
  - name: nome con cui viene identificata una icona. Per esempio name può essere Home.

- Style: l'oggetto IconStyle ha le seguenti proprietà:
  - color: colore dell'icona.
  - size: dimensione dell'icona.

4.5.4 List

Ogni elemento List rappresenta una lista di elementi. Gli elementi di una lista sono rappresenti dagli elementi ListTile. Se la lista non entra nello schermo, deve essere possibile scrollare la lista.

- type: List
- Props:
  - Children: Gli elementi contenuti all'interno della lista. Ogni elemento è di tipo ListTile.
- Style: l'elemento ListStyle ha le seguenti proprietà:
  - length: indica la lunghezza della lista. Se è definito la lista è detta fixedList, altrimenti la lista ha una lunghezza indefinita.
  - spacing: distanza tra i vari ListTile.

4.5.5 Divider

Linea che divide gli elementi di una lista.

- type: Divider
- Props:
  - nessuna
- Style: Non ha oggetti Style specifici

4.5.6 Card

Una Card ha bordi arrotondati e un'ombra. Una Card è utilizzata per contenere informazioni correlate fra loro come i dettagli di un profilo o una posizione geografica.

- type: Card
- Props:
  - Child: può contenere un elemento di ogni tipo.
- Style: l'oggetto CardStyle ha le seguenti proprietà:
  - color: colore della Card.
  - zCoordinate: elevazione della Card.
  - margin: espressa in pixel, indica il margine esterno della Card.

4.5.7 Image

type: Image

Props:

- src: path relativo o assoluto dove è presente l'immagine

Style: l'oggetto ImageStyle ha le seguenti proprietà:

- width: larghezza dell'immagine.
- height: altezza dell'immagine.

4.5.8 Chart

Ogni elemento Chart rappresenta un grafico. L'elemento Chart ha le seguenti proprietà:

- type: Chart
- Props:
  - chartType: Stringa che indica il tipo del grafico. Se non definito, viene calcolato automaticamente dai generatori.
  - data: questa proprietà deve essere associato a un elemento Data.
  - legend: se true, viene mostrata una legenda. Non viene mostrata se false.
- Style: l'oggetto ChartStyle ha le seguenti proprietà:
  - theme: tema usato per colorare il grafico. Lista dei temi da scrivere prima o poi.

4.5.9 Function [IGNORATELA]

Rappresenta una funzione implementata in un determinato linguaggio di programmazione.

- type: Function
- Props:
  - language: Stringa che indica il linguaggio di programmazione usato.
  - name: nome della funzione.
  - description: descrizione di cosa fa la funzione.
  - src: path dove è memorizzata la funzione.
- Style: non definito.

4.5.10 Table [Stile da definire]

Ogni elemento Table rappresenta una tabella. Nota che non è un elemento di layout, in base al generatore utilizzato, verranno sempre mostrati degli elementi su schermo. L'elemento tabella ha le seguenti proprietà:

- Type: Table
- Props:
  - header: la prima riga della tabella. Può contenere elementi di tipo testo.
  - columns: Indica il numero di colonne della tabella.
  - children: gli elementi che verranno inseriti all'interno della tabella.
- Style: da definire.

5 sottoclassi di Element e di Style in ConversationalInterface

5.1 Root

- type: Root
- Props:
  - botName: nome del bot
  - outputVoice: voce output di Alexa.
  - sessionTimeout: tempo di timeout di una sessione.
  - utterances: elemento di tipo Utterances. Ha createFile = true.
  - slots: elemento di tipo Slots. Ha createFile = true.
- Style: non definito.

5.2 Slots

Lista di Slot

- type: Slots
- Props:
  - Slots: lista di elementi di tipo Slot.
- Style: non definito.

5.3 Slot

Elemento che rappresenta uno Slot.

- type: Slot
- Props:
  - name: nome dello Slot.
  - type: tipo dello Slot. Può essere custom oppure il nome di uno Slot type predefinito di Amazon.
  - values: lista di stringhe che indica i valori che lo Slot può assumere. Questa proprietà è necessaria solo se type è custom.
- Style: non definito.

In particolare sono definiti di default 4 Slots. Lo Slot tag di tipo Custom a cui sono associati i valori dei tag. Quando viene creato un Chart, il tag assegnato al Chart viene aggiunto ai valori di questo Slot. L'eliminazione del Chart va gestita bene :¿.

5.4 Utterances

Lista di Utterances.

- type: Utterances
- Props:
  - utterances: lista di elementi di tipo Intent.
- Style: non definito.

5.5 Intent

Ogni elemento Link rappresenta un link.

- type: Utterance
- Props:
  - intent: nome dell'intento.
  - utterances: lista di utterances per il determinato intento.
- Style: non definito.

6 Creazione della pagina Home di Nurset

Partiamo col creare una pagina a cui associamo url **https://vivaio40.doriansrl.it/dashboard/portal/home** e senza subpages. Successivamente creiamo un oggetto Scaffold che poniamo come layout della pagina. Poi creiamo oggetto TopNavigationBar. Non tocchiamo header e title. Aggiungiamo però un figlio, ossia un DropDownElement a cui diamo id Profile. All'interno del DropDownElement eseguaiamo le seguenti operazioni:

- Creaiamo un bottone. Al bottone associamo alla prop child, un elemento Icon con name=User. Inoltre creiamo un oggetto Style con id TopNavigationBar e type Button in cui associamo alla proprietà type il valore text. Inseriamo il bottone come ClickableElement del DropDownElement.
- Creiamo una Card. Successivamente creiamo una List e cinque ListTile che comporanno la lista. Il primo ListTile è un Container con all'interno una seconda Lista in cui inseriamo altri tre ListTile, il primo è un'icona con name fa fa-user-circle, i secondi sono semplici elementi Text che contengono informazioni di sessione. Ai Text associamo due TextStyle diversi, quindi con id TextStyle1 e TextStyle2, e con parametri diversi. Al Container associamo uno oggetto Style con id TopNavigationBar, type=Container e padding: 10 10 10 10. Il secondo elemento della lista principale è un elemento Divider. Il terzo è un link che punta alla pagina del profilo. Il quarto è un Divider. Il quinto è un link che punta alla pagina di logout.

Aggiungiamo la TopNavigationBar al campo topNavBar dello Scaffold. L'aspetto della pagina dovrebbe essere il seguente:

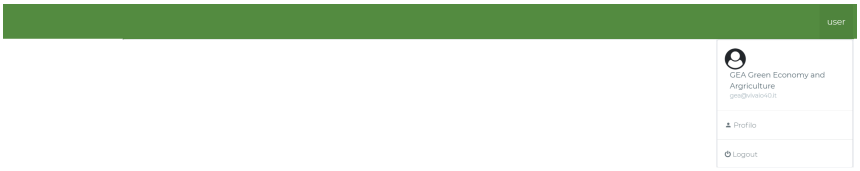


Figure 7: Creazione della topnavBar

A questo punto creiamo un Drawer. Per il Drawer pensiamo inizialmente all'header. Vogliiam

7 Elementi pre Configurati

Il Wizard offrirà delle configurazioni Yaml e dei layout di pagina pre fatti. A tal fine definiremo delle classi "Factory" che avranno il compito di creare strutture complesse già riempite.