



Obelix Group
obelixswe@gmail.com

Manuale utente

Versione	<i>v1_0_0</i>
Data creazione	2017-08-01
Redattori	Tomas Mali Emanuele Crespan Riccardo Saggese
Verificatori	Federica Schifano Nicolò Rigato
Approvazione	Silvio Meneguzzo
Stato	Approvato
Uso	Esterno
Distribuzione	Prof. Tullio Vardanega Prof. Riccardo Cardin

Sommario

Manuale utente per l'SDK Monolith e per le bolle di default e della Demo



Diario delle revisioni

Versione	Modifica	Autore e Ruolo	Data
1.0.0	Approvazione	Silvio Meneguzzo Progettista	2017-08-21
0.2.0	Verifica sezione 3	Federica Schifano Verificatore	2017-08-20
0.1.0	Verifica sezione 1-2	Nicolò Rigato Verificatore	2017-08-20
0.0.6	Stesura sezione 3	Emanuele Crespan Progettista	2017-08-19
0.0.5	Stesura sezione 2.2	Tomas Mali Progettista	2017-08-17
0.0.4	Stesura sezione 2.1	Emanuele Crespan Progettista	2017-08-15
0.0.3	Stesura Introduzione	Riccardo Saggese Progettista	2017-08-13
0.0.2	Creto Indice	Riccardo Saggese Progettista	2017-08-10
0.0.1	Creto template	Tomas Mali Progettista	2017-08-01



Indice

1	Introduzione	3
1.1	Cos'è Monolith SDK?	3
1.2	Scopo del documento	3
1.3	Glossario	3
2	Utilizzo	3
2.1	Layout	3
2.1.1	VerticalLayout	3
2.1.2	HorizontalLayout	4
2.2	Components	4
2.2.1	CheckBoxList	4
2.2.2	CheckButton	4
2.2.3	ComboBox	5
2.2.4	Image	5
2.2.5	ImageButton	5
2.2.6	LineEdit	6
2.2.7	LineEditComboBox	6
2.2.8	LineEditPushButton	6
2.2.9	PushButton	7
2.2.10	RadioButtonGroup	7
2.2.11	TextAreaButton	7
2.2.12	TextAreaComboBox	8
3	Bolle	8
3.1	CurrencyBubble	9
3.2	ListBubble	9
3.3	PollBubble	9
3.4	DiceBubble	11
4	DEMO - Lista con Checklist	12
4.1	Checklist	12
4.2	Lista	13



1 Introduzione

1.1 Cos'è Monolith SDK?

Monolith SDK è un pacchetto *Meteor*_{|G|} che consente la creazione di bolle interattive in ambiente *Rocket.chat*_{|G|}.

Lo Sviluppatore avrà la possibilità di usare i componenti della SDK per costruire le proprie bolle da integrare a quelle già esistenti sulla piattaforma di *Rocket.chat*_{|G|}.

1.2 Scopo del documento

Questo documento rappresenta il manuale utente per l'applicazione Monolith SDK nel quale vengono descritte dettagliatamente tutte le caratteristiche dell'applicativo utilizzabili dall'utente. Il manuale sarà diviso in sezioni per essere maggiormente comprensibile e spiegherà le varie componenti che l'utente potrà utilizzare per costruire la propria bolla.

1.3 Glossario

In modo tale da evitare ambiguità di linguaggio e massimizzare la comprensione, i termini tecnici, di dominio, gli acronimi e le parole che necessitano di essere chiarite, sono riportate nel documento *Glossario v3.0.0*.

Ogni occorrenza di vocaboli presenti nel Glossario è marcata da una _{|G|} maiuscola in pedice.

2 Utilizzo

2.1 Layout

2.1.1 VerticalLayout

Layout verticale che conterrà i componenti della bolla e li posizionerà ognuno sotto al proprio precedente. La props *hide* permette di nascondere la vista del layout (tramite valori *true/false*).

Utilizzo

```
<VerticalLayout hide={}>
<Children/>
<Children/>
.
.
.
</VerticalLayout>
```



2.1.2 HorizontalLayout

Layout orizzontale che conterrà i componenti della bolla e li posizionerà ognuno di fianco al proprio precedente. La props *hide* permette di nascondere la vista del layout (tramite valori *true/false*).

Utilizzo

```
<HorizontalLayout hide={}>
<Children/>
<Children/>
.
.
.
</HorizontalLayout>
```

2.2 Components

2.2.1 CheckBoxLayout

Una lista di Checkbox, i dati del CheckButton cliccato vengono ritornati tramite la funzione *getCheck*.

Utilizzo

```
<CheckBoxLayout
classes= // CSS classes
options={opt}
getCheck={this.fun}
/>
```

Esempio opzioni

```
var opt=[{id: 1, value: 'Hello World'}, {id: 2, value: 'value'}];
```

Funzione di ritorno

```
function fun(m) {...}

m={id:'', value:'', check:''};
```

2.2.2 CheckButton

Componente Checkbox, i suoi dati vengono ritornati (ad ogni click) tramite la funzione *getCheck*.

Utilizzo

```
<CheckButton
id= ""
classes= // CSS classes
getCheck={this.fun}
value="" // Valore CheckButton
/>
```

*Funzione di ritorno*

```
function fun(m) {...}  
  
m={id:'', value:'', check:''};
```

2.2.3 ComboBox

Semplice elemento ComboBox. La funzione di ritorno riceve l'elemento selezionato.

Utilizzo

```
<ComboBox  
id= "  
classes= // CSS classes  
options={["a","b","c"]} //array of options  
getSelection={this.fun}  
>
```

2.2.4 Image

Semplice elemento Immagine senza funzione di ritorno.

Utilizzo

```
<Image  
id= "  
classes= // CSS classes  
src= // like HTML "src" attribute  
alt= // like HTML "alt" attribute  
width= // like HTML "width" attribute  
height= // like HTML "height" attribute  
>
```

2.2.5 ImageButton

Elemento Immagine cliccabile come pulsante. La funzione di ritorno viene avviata quando l'immagine viene cliccata.

Utilizzo

```
<ImageButton  
id= "  
src= // like HTML "src" attribute  
alt= // like HTML "alt" attribute  
width= // like HTML "width" attribute  
height= // like HTML "height" attribute  
handleClick={this.fun}  
>
```



2.2.6 LineEdit

Casella di testo editabile. La funzione "fun" salva il testo digitato e viene chiamata ad ogni modifica.

Utilizzo

```
<LineEdit
id= ""
classes= // CSS classes
updateState={this.fun}
value="default value"
/>
```

2.2.7 LineEditComboBox

Elemento formato da una casella di testo editabile e una ComboBox. La funzione "fun1" riceve il testo dalla LineEdit mentre "fun2" l'opzione selezionata dal ComboBox.

Utilizzo

```
<LineEditComboBox
idle= // lineEdit id
idcb= // comboBox id
classesle= // lineEdit CSS classes
classescb= // comboBox CSS classes
textUpdate={this.fun1}
options=[["a","b","c"]] //array of option
comboUpdate={this.fun2}
/>
```

2.2.8 LineEditPushButton

Elemento formato da una casella di testo editabile e un pulsante cliccabile. La funzione "fun" riceve il testo dalla LineEdit quando il PushButton viene cliccato.

Utilizzo

```
<LineEditPushButton
idle= // lineEdit id
idpb= // pushButton id
classesle= // lineEdit CSS classes
classesp= // pushButton CSS classes
getText={this.fun}
buttonName="button name"
/>
```



2.2.9 PushButton

Pulsante cliccabile. La funzione "fun" viene eseguita al click sul pulsante.

Utilizzo

```
<PushButton
id= ""
classes= // CSS classes
handleClick={this.fun}
buttonName="button name"
/>
```

2.2.10 RadioButtonGroup

Elemento formato da un gruppo di RadioButton. La funzione "fun" restituisce l'opzione selezionata ad ogni cambiamento.

Utilizzo

```
<RadioButtonGroup
classes= // CSS classes
options={["a","b","c"]} //array of options
getValue={this.fun}
/>
```

2.2.11 TextAreaButton

Elemento area di testo editabile ed un PushButton.

La funzione "fun" riceve il testo dalla TextArea quando il PushButton viene cliccato.

Utilizzo

```
<TextAreaButton
idta= // textArea id
classesta= // textArea CSS classes
idpb= // PushButton id
classesp= // CSS classes
getText={this.fun}
width= // textarea width
height= // textarea height
buttonName= // button name
/>
```




2.2.12 TextAreaComboBox

Elemento area di testo editabile ed un ComboBox.

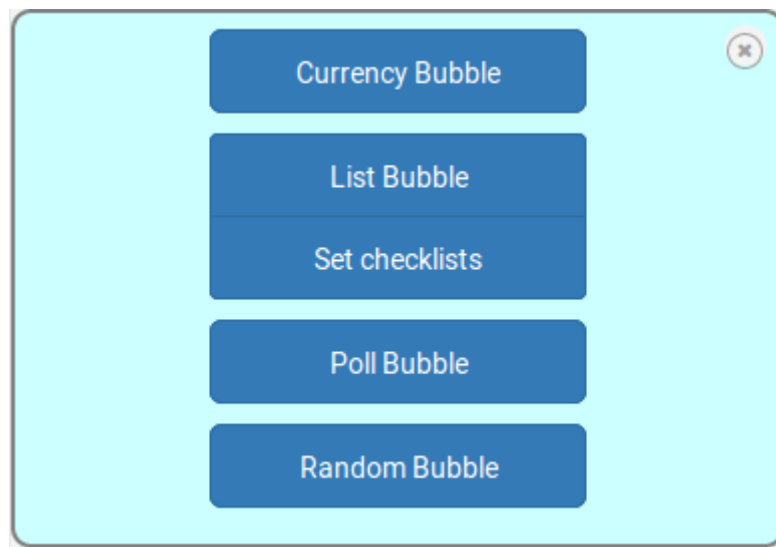
La funzione "fun1" riceve il testo dalla TextArea mentre "fun2" l'opzione selezionata dal ComboBox.

Utilizzo

```
<TextAreaComboBox  
  idtx= //textArea id  
  classestx= // textArea CSS classes  
  idcb= //combobox id  
  classescb= // combobox CSS classes  
  width= // textarea width  
  height= // textarea height  
  textUpdate={this.fun1}  
  options={["a","b","c"]} // array of options  
  comboUpdate={this.fun2}  
>
```

3 Bolle

Vengono fornite alcune bolle tipo le quali vengono create tramite il menù presente nella SideArea1.





3.1 CurrencyBubble

La *CurrencyBubble* è una bolla che funge da convertitore di valuta.

Una volta selezionata dal menù la si può configurare impostando la valuta di base, di uscita ed il valore da convertire.

Scegli la valuta base

EUR

Scegli la valuta finale

GBP

Inserisci il valore:

2

Send

Una volta inviata cliccando su **Send** il risultato sarà:

From:
EUR: 2

To:
GBP: 1.83

3.2 ListBubble

La *ListBubble* è una bolla che permette di creare una propria lista di elementi *CheckBox*.

Può essere usata, ad esempio, per una classica lista della spesa o per una "ToDoList".

3.3 PollBubble

La *PollBubble* è una bolla che permette di creare un sondaggio da lanciare ai membri della chat in questione.

Una volta arrivati al menù di configurazione si impostano la domanda e le varie opzioni (che si possono aggiungere con il pulsante **Add**).



Domanda:

Cosa mangiamo domani sera?

Option 1:

Pizza

Option 2:

Pasta

Option 3:

Pesce

Add

Send

Una volta inviata cliccando su **Send** il risultato sarà:

Cosa mangiamo domani sera?

Pizza Votes: 1

Pasta Votes: 0

Pesce Votes: 0

Ora che la bolla è stata creata sarà possibile votare cliccando sull'opzione desiderata.



3.4 DiceBubble

La *DiceBubble* è una bolla che lancia un certo tipo di dado (a scelta in base al numero di facce) e ne ritorna un risultato casuale.

Dal menù di configurazione bisognerà scegliere il numero di facce del dado.

4
6
8
12
20

Send

Una volta inviata cliccando su **Send** il risultato sarà:

Rolled number: 2

Re-roll

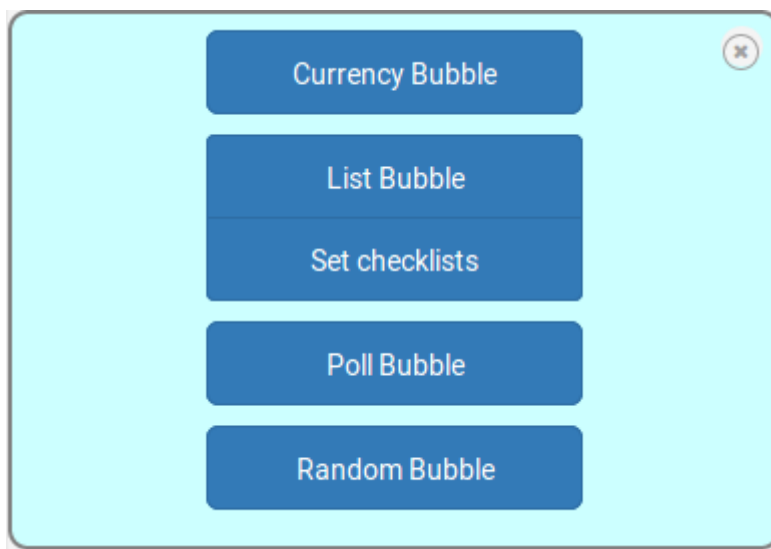
Attraverso il pulsante **Re-roll** si potrà ri-lanciare il dado.



4 DEMO - Lista con Checklist

4.1 Checklist

Le Checklist sono delle liste di elementi che possono essere selezionate e inserite in altre liste. Per creare una nuova checklist selezionare il pulsante **Set Checklist**.



Una volta selezionato il pulsante **Set Checklist** dal menù di creazione delle bolle si aprirà il menù di configurazione della checklist.

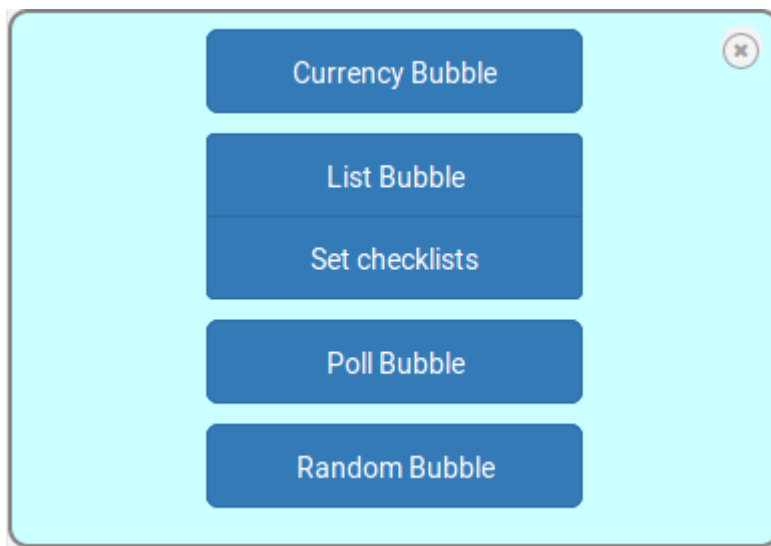
A screenshot of a light green rectangular configuration menu with rounded corners. At the top, it says 'Checklist Name:'. Below this are three input fields. The first field contains the text 'Spesa'. Below it, the label 'Option n:1:' is followed by a field containing 'Pane'. Then, the label 'Option n:2:' is followed by a field containing 'Formaggio'. Finally, the label 'Option n:3:' is followed by a field containing 'Latte'. At the bottom left, there are two dark blue buttons with white text: 'Add' and 'Send'.



Nel menù di configurazione bisogna inserire il nome della checklist e una o più opzioni. Per aggiungere una opzione bisogna selezionare il pulsante **Add**. Una volta completata la compilazione premendo il pulsante **Send** la checklist verrà inviata al server e sarà disponibile nella creazione delle liste.

4.2 Lista

La *Lista* è una bolla che permette di creare una lista nella quale è possibile aggiungere degli elementi dalle checklist precedentemente definite. Per creare una nuova lista selezionare il pulsante **Lista** dal menù di creazione delle bolle.



Una volta selezionato il pulsante **Lista** si aprirà il menù di configurazione della lista.

A screenshot of a web application interface for configuring a checklist. The interface has a light green background and a rounded rectangular border. At the top, the text "List Name:" is displayed in a bold, dark font. Below this, there are three input fields. The first field contains the text "Festa". The second field is preceded by the label "Option n:1:" and contains the text "Patatine". The third field is preceded by the label "Option n:2:" and contains the text "Fanta". Below the third field, there is a label "Option n:3:" followed by the text "Costicine". At the bottom left of the form, there are two blue buttons with white text: "Add" and "Send". A blue button with the text "CheckLists" is positioned between the first and second input fields.

Nel menù di configurazione bisogna inserire il nome della lista e una o più opzioni. Per aggiungere una opzione bisogna selezionare il pulsante **Add** oppure si può selezionare degli elementi da inserire nella lista delle checklist. Per aprire il menù di selezione delle checklist bisogna premere il pulsante **Checklists** nel menù di configurazione della lista.



List Name:

Festa

CheckLists

Spesa

☒ Pane

☒ Formaggio

☐ Latte

Frutta

☐ Mela

☐ Mirtillo

☐ Fragola

Add Checklist

Option n:1:

Patatine

Option n:2:

Fanta

Option n:3:

Costicine

Premendolo nuovamente il menù delle checklist verrà chiuso. Una volta selezionate le checklist bisogna premere il pulsante **Add checklist** per aggiungerle alla lista.



List Name:

Festa

CheckLists

Option n:1:

Patatine

Option n:2:

Fanta

Option n:3:

Costicine

Opzione 4:

Pane

Opzione 5:

Formaggio

Add

Send

Una volta completata la compilazione premendo il pulsante **Send** la lista verrà inviata al server e sarà visualizza nell'area contenente le bolle inviate.