Device Configurator Guida utente I.B.C.

Indice	. 2
	• -

Introduzione

Preparazione iniziale

1) <u>Menu</u>

- 1.1 Projects
- 1.2 Images
- 1.3 <u>Debug</u>
- 1.4 <u>Deploy</u>
- 1.5 <u>Settings</u>
- 1.6 <u>Help</u>
- 1.7 <u>L'emulatore</u>
- 2) <u>Toolbar, device & OS Bar</u>
- 3) <u>Configuration</u>
 - 3.1 <u>Generale</u>
 - 3.2 Languages
 - 3.3 <u>User profiles</u>
 - 3.4 Admin Profile
 - 3.5 <u>Designer params</u>
 - 3.6 <u>Item tags</u>
 - 3.7 <u>Sounds</u>
- 4) <u>App styles</u>
 - 4.1 Splash form
 - 4.2 <u>Header</u>
 - 4.3 <u>Footer</u>
 - 4.4 <u>Toolbar</u>
 - 4.5 <u>Bottombar</u>
 - 4.6 <u>Menu</u>
 - 4.7 <u>Menu page</u>
 - 4.8 Advertising bar
- 5) <u>Control's styles</u>
 - 5.1 Buttons
 - 5.2 <u>Check box</u>
 - 5.3 <u>Edit</u>

5.4 <u>Fixed text</u>

5.5 <u>Form details area</u>

5.6 <u>Form filter</u>

5.7 <u>Groups</u>

5.8 <u>Label</u>

5.9 <u>Link</u>

5.10 List item

5.11 List title, List balel e List value

5.12 <u>Memo</u>

5.13 Message

5.14 Page title

5.15 <u>Status field</u>

5.16 Custom fields

6) <u>Plugins</u>

6.1 <u>Registration/Login</u>

6.2 <u>Camera</u>

6.3 <u>Companies</u>

6.4 Documents

6.5 <u>HTML</u>

6.6 Instant Messaging

6.7 Location

6.8 Presentation

6.9 <u>Security</u>

6.10 <u>Urls</u>

6.11 <u>AppInApp</u>

6.12 <u>Google ads</u>

6.13 <u>Blog & help</u>

7) <u>Database</u>

7.1 <u>Tables and queries</u>

7.2 <u>Table options</u>

7.3 <u>Groups</u>

7.4 <u>Table actions</u>

7.5 SQL filters & join

7.6 <u>SQL</u>

7.7 <u>Field options</u>

7.7.1 <u>TStringField</u>

7.7.2 <u>TMemoField</u>

7.7.3 <u>TSmallIntField</u>, <u>TIntegerField</u>

7.7.4 <u>TFloatField</u>

7.7.5 <u>TLookupListField</u>

7.7.6 <u>TLookupEditField</u>

7.7.7 <u>TDocumentField</u>

7.7.8 <u>TDateField</u>

7.7.9 <u>TImageField</u>

7.7.10 TFixedImage

7.7.11 <u>TListField</u>

7.7.12 <u>TCheckField</u>

7.7.13 TSignatureField

7.7.14 <u>TGPSFields</u>

7.7.15 <u>TPhoneField</u>

7.7.16 <u>TUserIdField</u>

7.7.17 <u>TUserProfileField</u>

7.7.18 <u>TUserEMailField</u>

7.7.19 <u>TCompanyIdField</u>

7.7.20 TMasterField, TdetailField

7.8 <u>Query</u>

7.8.1 Query options

7.8.2 <u>SQL editor</u>

8) <u>Creazione dell'app</u>

8.1 <u>Registrazione dell'app</u>

8.2 <u>Acquisto dei plugins</u>

Appendici :

A) Come creare un proprio template applicativo

Introduzione

Il Device Configurator è lo strumento che ti permette di creare un'applicazione mobile cross platform Android/iOS a partire dal tuo sito internet. In questa guida ti illustreremo le funzionalità e le modalità d'uso del Device Configurator. Questa guida è stata strutturata in modo da descrivere i comandi a seconda della loro posizione andando dall'alto verso il basso.

Preparazione iniziale

Registrati sul sito <u>https://www.ibcappmaker.com</u> e scarica il plugin di connessione, relativo al CMS di sviluppo web utilizzato, dalla pagina <u>https://www.ibcappmaker.com/en/ibcprojects/downloads/category/5-plugins</u>.

Installa il plugin sul CMS adottato per sviluppare il sito da cui ricavare l'app.

Scarica il Device Configurator ed installalo sul tuo pc. Lo puoi trovare a questo link :

https://www.ibcappmaker.com/en/ibcprojects/downloads/category/2-device-configurator.

1) Menu

🦺 Devi	ce Config	urator [te	st-wp]			_	×
Project	Images	Debug	Deploy	Settings	Help		
The mention law del Menu and the mente and environment all'Interests of a size On exercise							

Un particolare del Menu posto nella parte superiore dell'Interfaccia Operativa.

Il menu costituisce la parte fondamentale del Device configurator perché consente l'accesso alle varie funzioni della piattaforma applicativa. E' suddiviso in 6 diverse aree che determinano le sue zone funzionali. Esclusa la parte relativa all'help ogni voce di menu serve per una data configurazione dell'app da creare.

1.1 Projects

Il menu projects consente la piena gestione dei progetti che si possono creare con il Device Configurator.

Proj	ect	Images	Debug	Deploy	Help
Ø.	Ne	w	Ctrl	+N	
Þ	Ор	en	Ctrl	+0	
2	Up	date proje	ct		
P	Sav	/e	Ctr	I+S	
	Clo	ose	Ctr	I+C	
0	Exp	port			

Immagine del menu Projects.

Si può creare un nuovo progetto cliccando sulla voce "New" o premendo la combinazione di bottoni 'Ctrl+N'.

Selezionata la voce di menu comparirà la finestra sottostante.

Create new project 🔹							
Project name : Type your project name							
Default type : English ~							
Toolbar type : Mon	ochrome icons	~					
☑ Use default system images							
	Close	Cancel					

Immagine della finestra per la creazione di un nuovo progetto.

In **Project name** inserire il nome del progetto. Gli spazi ed alcuni caratteri speciali saranno convertiti in '_'. Aprire la tendina **Default type.** Verranno visualizzate una serie di voci indicanti la lingua del progetto che si vuole creare. Il Device Configurator supporta nativamente **6 Lingue** più un progetto **Multilingua che le integra tutte**.

La voce **Toolbar Type** indica il tipo di immagini che si vogliono utilizzare nel progetto. Queste possono essere monocromatiche e seguiranno il colore del font utilizzato nella toolbar o colorate. Indipendentemente da questa scelta sarà possibile cambiarne il tipo nella gestione immagini della toolbar. Vedere <u>paragrafo 1.2</u>.

Selezionare la voce '**Use default images'** se si vuole che il progetto supporti tutte le immagini di default. Si consiglia di mantenere selezionata questa voce ed eventualmente cambiare in seguito le immagini desiderate.

Premere il testo '**Close'.** Dopo l'attesa di alcuni instanti verrà creato un nuovo progetto, sarà attivata la toolbar e sarà mostrata l'alberatura con i moduli o le sezioni da configurare. Questa parte sarà spiegata nel dettaglio nei capitoli successivi.

Le altre voci che compongono questa sezione del menu sono :

Open : per aprire un progetto creato in precedenza;

Save : per salvare il progetto corrente;

Close : per chiudere il progetto corrente ed eventualmente crearne o aprirne uno nuovo;

Export : per esportare il progetto in un file proprietario ed inviarlo ad altri sviluppatori o semplicemente per averne una copia di backup da custodire.

1.2 Images

La sezione **Images** del menu permette la completa gestione delle immagini che compongono le app create con App Maker. Il menù è diviso in due parti. La parte superiore permette la gestione delle immagini applicative mentre quella inferiore delle immagini di sistema relative a ciascun sistema operativo supportato, Android di Google e iOS di Apple.



Immagine del menu Images.

Una app creata con I.B.C. App Maker ha varie parti opzionali che lo sviluppatore può decidere di usare e mostrare al proprio utente.

Nell'immagine sottostante è possibile vedere una pagina di menu, precisamente quella creata per l'app "I.B.C. App Maker" che contiene due barre, una in alto detta **HeaderBar** ed una in basso chiamata **FooterBar**.



Immagine della pagina di menu che mostra la Headerbar e la Bottombar.

La **Headerbar** contiene il logo dell'app, un bottone per l'accesso alle notifiche ed un bottone per l'accesso al menu laterale, se attivato.

La **FooterBar** contiene una stringa, in genere quella con le informazioni sul copyright. Lo sviluppatore può personalizzare font, immagini e colori a proprio piacimento.

La **BottomBar** è una barra opzionale che contiene una serie di bottoni, definiti dallo sviluppatore, che permettono l'accesso diretto ad alcune funzionalità di menu senza doverlo usare direttamente. E' estremamente comodo per l'accesso immediato a tutte le funzionalità di maggior uso da parte dell'utente. I.B.C. App Maker ha una propria gallery di immagini chiamata **Repository** che contiene molte immagini che lo sviluppatore può utilizzare durante la creazione delle proprie soluzioni. Se queste immagini, però, non servissero allo scopo o non fossero sufficienti o di gradimento per lo sviluppatore, se ne possono importare e/o definire di aggiuntive. L'immagine sottostante mostra la **BottomBar** di I.B.C. App Maker.

Per personalizzare il titolo dell'app inserire la stringa desiderata alla voce Title, invece per personalizzare il testo della footerbar inserire, modificare la stringa FooterText nella sezione Languages \rightarrow Systems language, che tratteremo più avanti.

Gestione tabelle
Q
Elenco timbrature
Valute Lista valute
Prodotti Lista prodotti in evidenza
Utenti Lista utenti
Aziende Elenco aziende
Impiegati Lista degli Impiegati
Documenti Lista documenti
BottomBar
🔢 🔞 🕺 🗇 🐑 🔔 🐺 💐

Immagine che indica la posizione della Bottombar.

Cliccando su ciascuna delle voci del menu Images viene mostrata una finestra simile a quella sottostante. Questa finestra è chiamata **Image Editor**.



Editor che permette di associare a ciascun 'Item' presente nella combobox a sinistra una precisa immagine selezionata tramite il bottone 'Select image', sempre a sinistra.

L'**Image Editor** permette di definire ciascuna immagine che deve essere contenuta nelle varie sezioni dell'app.

Le sezioni sono :

- **BottomBar** per tutte le immagni che devono essere contenute nella BottomBar;
- ToolBar per tutte le immagini che devono essere contenute nelle varie toolbar funzionali di App Maker. Di queste parleremo più avanti;
- **Menu items** per le immagini associate alle voci del menu laterale dell'app;
- **Menu Page Items** per le immagini associate alla pagina di menu applicativo. Vedere pagina principale di App Maker.

• **Fixed images** una serie di immagini di dimensione variabile che sono associate a varie funzioni di App Maker di cui parleremo in seguito.

Quando si gestiscono le immagini della toolbar viene mostrata una combobox per il cambio d'impostazione del tipo di immagini da utilizzare nella toolbar stessa. Questa scelta sovrascrive quella fatta al momento della creazione del progetto.

Le dimensioni della immagini da inserire sono di 32*32 pixel per bottombar, toolbar e menu items, mentre di dimensione libera per Menu page items e Fixed images. Per le ultime due si consiglia l'utilizzo di immagini da 200 * 200 pixel ad un massimo di 300 * 300 pixel.

L'Image editor possiede una toolbar con 4 bottoni, le cui descrizioni sotto riportate qui sotto.

- Il bottone '**Add**' inserisce nella lista a destra una nuova immagine per l'item correntemente selezionato.
 - Il bottone 'Modify' modifica l'immagine definita per l'item correntemente selezionato nella lista a destra.
 - Il bottone '**Delete**' cancella la configurazione per l'item correntemente selezionato.
- Р

2

- Il bottone '**Save**' salva la configurazione corrente.
- Il bottone '**Refresh**' ricarica i contenuti della finestra, aggiornando gli eventuali Item types.

Per aggiungere una nuova immagine, selezionare dalla combobox '**Item**', la voce desiderata, cliccare su '...' della voce **Select image** ed infine selezionare un'immagine. Dopo premere sul bottone '+', per aggiungere l'immagine alla lista posta sulla destra.

Per modificare un'immagine già esistente, selezionarla dalla lista sulla destra del form, verrà visualizzato il preview dell'immagine stessa, selezionare una nuova immagine, premendo '…', come detto sopra e cliccare sul bottone '**Modify**' della toolbar.

Per salvare la configurazione corrente premere '**Save**', mentre per cancellare un'immagine non desiderata, premere il bottone '**Delete**'.

Nella combobox della voce **Item** sono presenti una serie di codici di default che possono essere usati per le relative immagini. Questi sono BB-Camera, BB-Chat, BB-Maps... Quando questi fossero terminati usare gli altri.

Le voci '**Android system images & Icons**' e '**iOS system images & Icons**' consentono di definire le immagini per lo splash form dell'applicazione, le icone della stessa e quelle visualizzate alla ricezione delle notifiche.

1.3 Debug

Il menu **Debug** contiene le funzioni per il test dell'app che si sta creando. Sotto l'immagine delle varie voci contenute in questa sezione.

Project Images	Debug	Deploy Help	
DI	🕨 Ri	un emulated version	F9
Device : Pho	💐 A	oply	F8
	St St	ор	Ctrl+F2
	💽 U	pload	Ctrl+U
· · · ·	•		

Immagine del menu Debug.

Cliccando sulla voce '**Run emulated version**' si avvia l'emulatore di un dispositivo mobile e viene mostrata l'app all'interno di questo. Questa funzione consente di eseguire l'app sul p.c. Windows nella sua interezza, eccetto lo splash form, proprio di ogni sistema operativo.

La funzione '**Apply**' consente di applicare all'app correntemente in esecuzione all'interno dell'emulatore le modifiche apportate alla configurazione di questa.

Questa funzione permette di non terminare l'applicazione e ripartire dall'inizio. E 'bene osservare, però, che l'app viene riportata alla schermata iniziale .

La funzione **'Stop'** termina la sessione corrente dell'emulatore.

La funzione **'Upload'** permette di caricare sul proprio server le impostazioni di progetto correntemente salvate. Questa operazione è stata creata per emulare l'app direttamente su uno smartphone. Questo è possibile scaricando ed istallando I.B.C. App Maker, disponibile gratuitamente su Google Play, sul proprio smartphone.

Avvia I.B.C. App Maker sul dispositivo e loggati con le stesse credenziali del sito www.ibcappmaker.com.

Per avviare il test della tua applicazione premi la voce 'App Test' nel menu laterale. Inserisci l'indirizzo internet del tuo sito web e in pochi secondi potrai visualizzare l'anteprima della tua app direttamente sul tuo smartphone.

E' possibile accedere a queste singole funzioni anche con la combinazione di tasti visibile a lato delle varie voci di menu.

1.4 Deploy

Il menu **Deploy** contiene tutte le funzioni necessarie per la creazione dei file di distribuzione che lo sviluppatore dovrà caricare sul proprio store per la rivendita dell'app.

L'immagine sottostante mostra la tendina aperta.



Immagine del menu Deploy.

Una volta terminata la configurazione della propria app per poterla caricare sul proprio store Google o iOS e distribuirla è necessario compilarla e creare i file di distribuzione. Queste 2 fasi non sono assolte dal **Device Configurator**, ma dai bot presenti sui server di I.B.C. App Maker. Quando lo sviluppatore decide che la configurazione della sua app è terminata deve registrare l'app sul sito www.ibcappmaker.com all'indirizzo <u>https://www.ibcappmaker.com/en/ibcprojects/android-projects</u> se si tratta di un'app Android o all'indirizzo <u>https://www.ibcappmaker.com/en/ibcprojects/ios-projects</u> per un'app iOS. Se si desidera creare un'app cross-platform, ossia si vuole sia l'app Android che quella iOS, prima si devono registrare separatamente le due app nelle relative sezioni, come descritto sopra e dopo registrare l'app alla pagina <u>https://www.ibcappmaker.com/en/ibcprojects/multi-platform-projects</u>.

Durante la fase di registrazione sul sito vengono richiesti alcuni file creati dal Device Configurator. Per ottenere questi files si devono utilizzare le funzioni presenti in questo menu.

La funzione **Deploy Android Images** genera il file di distribuzione contenente tutte le immagini di sistema appartenenti all'app Android.

La funzione **Deploy iOS Images** genera il file di distribuzione contenente tutte le immagini di sistema appartenenti all'app iOS.

La funzione **Deploy** genera il file di configurazione dell'app.

La funzione **Check plugins** indica allo sviluppatore quali plugins acquistare sullo store di I.B.C. App Maker perché l'app venga correttamente generata. Se per errore non acquistassi tutti i plugins necessari, non preoccuparti: la piattaforma di distribuzione di I.B.C. App Maker ti avviserà immediatamente durante la generazione dall'app inviandoti una email. Una volta comprati i plugins mancanti la generazione dell'app riprenderà senza problemi. Quando la generazione sarà conclusa, il file di distribuzione da caricare sul tuo store, verrà caricato sul sito di I.B.C,. App Maker, direttamente nell'area del tuo progetto e ti sarà inviata una email per avvisarti.

Durante la fase di registrazione dell'app sul sito di App Maker alle pagine sopra indicate sono richiesti vari parametri. Fondamentali sono il caricamento del file alla voce **System images** e alla voce **Deploy file**. Alla prima voce deve essere caricato il file creato tramite la funzione *Deploy Android images* o *Deploy iOS images* sopra citate. Alla seconda, il file ottenuto tramite la funzione *Deploy*.

Per la ricezione delle notifiche sia Android che iOS necessitano di file aggiuntivi che si possono creare solamente sui relativi siti degli store in questione. Rimandiamo ad una trattazione successiva questo argomento.

N.B. Ricorda! Per caricare un'app su App Store di Apple devi possedere necessariamente un computer Mac.

1.5 Settings

La voce di menu **Settings** mostra una finestra che consente alcuni impostazioni di ambiente. Sono presenti 4 categorie di impostazioni come è possibile vedere nell'immagine sotto.

Settings	
Settings Log for Debugging Disable Image warnings Project's path Sounds	Select debug options you want use Log on remote SQL statements Log on local SQL statements Save Logs to file Open automatically log window
	Close Cancel

Immagine della finestra delle impostazioni di sistema.

Log for debugging

In questa categoria sono presenti 4 funzioni:

- **Log on remote sql statement** : se abilitata consente di vedere in una finestra di log tutti gli statement sql che vengono eseguiti sul server all'interno del database mysql.
- **Log on local sql statement** : se abilitata consente di vedere in una finestra di log tutti gli statement sql che vengono eseguiti sul database locale Sqlite.
- **Save log on file** : indica il salvataggio del log su un file.
- **Open automatically log windows** : indica l'apertura automatica della finestra di log per gli statement database.

Disable warnings

Durante l'emulazione di un'app, se l'utente si fosse dimenticato di definire alcune immagini di cui alla <u>sezione 1.2</u>, l'emulatore indica allo sviluppatore quale immagine deve definire perché mancante. Per silenziare temporaneamente questa funzionalità, cliccare sulla relativa voce nella sezione '**Disable image Warnings**'. I warnings sono articolati su 4 categorie a seconda del tipo di componente che possiede l'immagine.

Projects path

In questa sezione si indica la cartella che deve contenere i progetti di AppMaker.

Sounds

In questa sezione vengono indicati i suoni che devono essere eseguiti quando un device viene attaccato ad un pc mediante cavo usb.

1.6 Help

L'**Help** è una delle sezioni più importanti per chi si sta approcciando all'uso di I.B.C. App Maker perché consente di apprenderne il funzionamento. Cliccando su ciascuna voce viene mostrato l'help in formato .chm, aperto ad una specifica pagina, contenente le informazioni dell'argomento indicato.



Immagine del menu Help.

1.7 L'emulatore

Il **Device Configurator** consente la visualizzazione dell'app che si sta sviluppando senza la necessità di un device fisico smartphone o tablet. Questa possibilità è data dalle funzioni di **Debug** viste nel <u>paragrafo 1.3</u> a cui rimandiamo per una completa panoramica del test dell'applicazione. Nella parte inferiore della finestra di emulazione sono presenti 4 bottoni che ne consentono il controllo. Questi, oltre alla possibilità offerta dai device fisici, hanno funzioni aggiuntive. Nella figura qui sotto possiamo vedere l'emulatore all'avvio.



Emulatore all'avvio dell'app.

Funzioni dei bottoni dell'emulatore:

- A) Chiude la pagina corrente e si sposta sulla precedente, comune in Android;
- **B)** Avvia l'emulazione dopo uno stop;
- **C)** Termina l'esecuzione dell'applicazione chiudendo la finestra dell'emulatore.
- **D)** Ferma l'esecuzione dell'emulatore mostrandolo come una semplice finestra. Questa funzione permette di spostare la finestra in un qualsiasi punto dello schermo. Per riprendere l'esecuzione dell'app premere il bottone **B**).

2) Toolbar, device & OS Bar

La toolbar permette l'accesso diretto alle funzioni di maggior utilizzo. Sotto una sua immagine.

Device : Phone 👻 O.S. : Android 🗸 = 💾 💋 💽 📐 🏹 🔳 🌑

Immagine della toolbar del Device Configurator.

L'utilizzo e le funzionalità sono molto intuitive. Con il parametro *Device* si imposta il device correntemente in uso *Phone* o *Tablet*, invece con il parametro *O.S.* il sistema operativo *Android* o *iOS*.

Selezionata una configurazione, tutte le modifiche apportate ai parametri del Configurator saranno relative **solamente** a questa. È possibile copiare una data configurazione su di un'altra premendo il bottone a destra della combobox O.S. Questo apre un menu di popup con 5 voci come nella figura sottostante.



Selezionando le varie voci del menu è possibile copiare la configurazione corrente su un'altra. È necessario usare con molta cautela questa funzionalità perché saranno sovrascritti tutti i parametri grafici dell'app.

È importante osservare che i parametri di configurazione sono essenzialmente di 2 tipi: *grafici* e *funzionali*. I parametri grafici sono quelli relativi all'interfaccia, ad esempio, font, colore o allineamento. Quelli funzionali invece determinano il comportamento dei vari plugins. Tutti i parametri grafici devono essere definiti su ciascuna piattaforma e sistema operativo, mentre quelli funzionali è possibile definirli una sola volta poiché vengono automaticamente condivisi.

3) Configuration

In questa sezione tratteremo i parametri di configurazione dell'app selezionabili dall'alberatura a sinistra del configurator, direttamente sotto la voce **Configuration**.

						-		-	-		-	
Device :	Phone	~ c).S. :	Android	~ =	P	2	0		•		
Configura	tion		^									
Gener	al											
Langu	ages											
User p	rofiles											
Admir	profiles											
Desig	ner params											
A App s	tyles											
Sp	lash form											
He	eader											
Fo	oter											
То	olbar											
Bo	ttombar											
M	enu											
M	enu page											
▲ Contri	ols' styles											
⊿ De	fault			0								
	Buttons											
	Checkbo	c										
	Edit											
	Fixed text	t										
	Form det	ails are	a									
	Form filte	er -										
	Groups											
	Label											
	Link											
	List title											
	List label											
	List value											
	Memo											
	Messane											

Immagine del Device configurator.

3.1 Generale

Nella sezione **General** si definiscono i parametri generali dell'applicazione.

General								
General	ManuDama							
Main form :	MenuPage	× ····						
Web site domain :	https://www.ibcappmaker.com	Test						
Presentation :	Main	~						

Immagine delle proprietà di configurazione generale.

Tutti i parametri sono univoci e condivisi tra le varie piattaforme.

La proprietà '**Main Form'** definisce il form principale dell'app. In genere è impostato come *Menu Page* ma è possibile definire qualsiasi form come principale, dopo averlo correttamente definito. Questo form viene aperto all'esecuzione dell'app.

La proprietà **Website domain** indica il sito web dove è istallato il plugin di connessione con l'app. Premendo il bottone **'Test'** si può verificare se l'istallazione di tale plugin sia corretta.

La proprietà **'Presentation'** indica se utilizzare o no una presentazione dell'app prima della registrazione utente. Il plugin presentation può essere utilizzato prima dell'utilizzo di qualisiasi plugin definito nel menù, nella menù page o nella bottombar.

CMS	
Transaction secret :	
Transaction key :	
Admin email :	Test
Admin password :	•
Immagine delle	proprietà di configurazione del CMS.

Le proprietà **'Transaction Secret'** e **'Transaction key**' indicano la parola chiave e la password con cui creare una stringa criptata per validare ciascuna al server. Questa tecnologia evita l'accesso diretto degli hacker al server tramite chiamate dirette.

Le proprietà **'CMS admin email'** e **'CMS admin password'** indicano l'account dell'amministratore del sito web che deve avere necessariamente i diritti amministrativi. Questo account è usato dal plugin **Database** per la creazione delle tabelle di database.

Rewrite Url Uvindows emulator Android 7 Android 8 Android 9 Android 10 Android 11 Android 12

Immagine delle proprietà rewrite url.

Le proprietà '**Rewrite Url**' cambiano la chiamata da https a http sui sistemi operativi indicati. Queste impostazioni devono essere utilizzate quando si utilizza un sistema operativo obsoleto rispetto agli standard attuali e le chiamate https non sono correttamente supportare.

3.2 Languages

Languages è la sezione dove è possibile gestire la lingua e le traduzioni in cui appaiono le varie stringhe dell'interfaccia. *Languages* possiede due funzionalità:

- definire/modificare le stringhe di default a seconda delle lingue presenti nel progetto, in una delle 6 già pre-impostate (Italiano, Inglese, Francese, Spagnolo, Portoghese e Tedesco) o aggiungerne una nuova lingua;
- aggiungere stringhe custom, definite dall 'utente, nelle lingue correntemente definite per la piattaforma.

Add new :			× +		
Languages :	Italiano	(it)	~ 🤊 🕅		
System strings	Custom	strings	в		
Item		Value			
Title		IBC App Maker	^		
ToolbarTitle		IBC App Maker			
FooterText		Copyright 2019			
Language		Lingua :			
Name		Nome e cogno			
Username		Nome account :			
Email		Email :			
Password		Password :			
ShowPassword		Mostra password			
GDPRConsent		Consenti l'invio			
Register		Registra			
Login		Entra			
ResetPassword		Recupera passw			
RegisterMessage		Se sei registrato			
LoginMessage	LoginMessage				
LoginMessageDeta	iled	Se non sei regis			
NamePrompt		Inserisci nome	~		
		P	🔑 🎓 💾 E F G		

Pagina di gestione delle lingue per la configurazione dell'app. Qui è possibile gestire le stringhe di sistema e le stringhe utente.

Per scegliere una delle sei lingue di default, è sufficiente selezionarla nella combobox *Languages*. Per aggiungere una nuova lingua, invece, cliccare sulla *combobox* del comando **Add New** e selezionare una delle lingue presenti, quindi cliccare sul bottone **A**.

Selezionando la linguetta '*System strings*' è possibile modificare o definire le stringhe di sistema dell'app, come ad esempio messaggi di errore, messaggi utente, label generali dell'interfaccia.

Per cancellare una lingua già definita cliccare sul bottone **C** mentre per definire la lingua di default, impostare una lingua e premere il bottone **B**.

E' importante definire una **lingua di default** perché questa viene usata quando l'app è eseguita su un telefono che non contiene una delle lingue definite per il progetto.

Il device configurator gestisce un **repository di lingue** che può essere **esportato ed importato** in un formato tabbed txt ovvero **.csv con separatore di carattere tab**. I bottoni **D** ed **E** , relativamente, esportano ed importano le stringhe che l'utente può inviare ad un traduttore o affidare a Google translator.

Il bottone **F** aggiorna e riordina alfabeticamente le stringhe mentre il bottone **G** salva le traduzioni su disco.

Selezionando la linguetta *'Custom strings'* è possibile modificare o definire le stringhe utente eventualmente presenti nell'app, come ad esempio Label, descrizioni, text prompt ... di campi di database appartenenti a tabelle create e gestite col plugin **Database** (vedi <u>capitolo 7</u>).

3.3 User profiles

In questa sezione si definiscono i profili utente supportati dall'app. In pratica si devono inserire tutti i profili definiti sul proprio CMS.

Il Device configurator di default contiene tutti i profili utente dei CMS supportati.



Immagine della sezione che definisce i profili utente.

Per aggiungere un nuovo profilo premere sul bottone '+', inserire la stringa desiderata e confermare con il tasto '**OK**'.

Per cancellare un profilo prima selezionarlo e poi premere '-'.

Volendo è possibile cambiare anche il loro ordine tramite le freccette poste sotto la lista.

Define profile to b not enabled.	e used by Databas	e plugin when R	egistration plugin is
Default profile :	Registered	~	

Immagine della sezione profilo di default.

Nei casi in cui non si voglia utilizzare il plugin '**Registration**' per la registrazione utente e comunque si decida l'utilizzo del plugin '**Database**' che necessita del profilo utente correntemente loggato è necessario impostare questa property. Infatti l'app userà come profilo utente non quello dell'utente che non viene definito ma direttamente il profilo qui definito.

3.4 Admin profiles

In questa sezione si definiscono i profili amministrativi supportati dall'app. Di default il **Device configurator** contiene *Super User* ed *Administrator*. Lo sviluppatore può aggiungere un nuovo profilo selezionandolo dalla combobox **User Profiles** e premendo, poi, il bottone **'+'**.

Per cancellarlo deve solamente selezionarlo e premere il bottone '-'.



Immagine della sezione che definisce i profili amministrativi.

Questi profili sono molto importanti perché permettono di abilitare alcune funzioni dell'app solo a chi possiede i privilegi amministrativi.

3.5 Designer params

Questa sezione contiene i parametri che indicano come devono essere disegnati i vari form che costituiscono l'app. Come molti dei parametri trattati, questi devono essere inseriti per ciascuna piattaforma. Il Device Configurator possiede già dei valori di default accettabili.

Immagine dei parametri base per la creazione dei forms.

La sezione **Database form** indica i parametri con cui sono creati tutti i form di database.

I parametri **Start X** e **Start Y** indicano in pixel le coordinate da cui deve iniziare la creazione del form stesso.

Interline indica, in pixel, lo spazio tra due campi mentre **Space** indica lo spazio tra una *Label* ed il controllo associato.

Il parametro **Space** di **Menu Page** indica, in pixel, lo spazio tra i bordi ed i vari oggetti che compongono questa pagina.

I parametri della sezione **Buttons** indicano la dimensione massima che questi devono avere all'interno dell'app ed il numero massimo di bottoni che si possono avere per singola riga ad esclusione dei bottoni delle varie toolbar dei plugins supportati.

3.6 Item tags

Questa sezione spiega come gestire gli identificatori delle immagini delle sezioni **Menu**, **Menu Page, Bottombar** e **Fixed Images,** trattati nella <u>sezione 1.2</u>. Prima di tutto selezionare un item dalla combobox **Item Type** (vedere figura sotto). Nella listbox sottostante compariranno una serie di voci. Queste sono gli identificatori della immagini attualmente in uso. Per aggiungere un nuovo item cliccare sul '+' e scrivere una stringa che identificatore non più in uso, selezionarlo e premere '-', per modificarlo, selezionare l'item, fare un doppio click col mouse e modificare la stringa proposta, premendo sul bottone '**OK**'.

Per aggiornare i codici delle delle immagini nella relativa finestra, se già aperta, è necessario chiuderla e riaprirla o premere il bottone **Refresh**.



Immagine della sezione Item tags.

3.7 Sounds

La sezione **Sounds** consente di definire e personalizzare i suoni emessi dai plugin **Instant Messaging** e **Notifications**. Selezionando, dalle

relative combobox, una voce è possibile impostare impostare il suono indicato dal file .mp3 all'item indicato.

Sono supportati i seguenti suoni per i relativi plugin :

• Notifiche

Notifications received: indica la ricezione di una nuova notifica; *Unread notifications*: emesso all'avvio dell'app, indica che ci sono notifiche già ricevute ma non ancora lette.

• Instant messaging

Message sent: indica il corretto invio di un messaggio; Message received: indica la ricezione di un nuovo messaggio; Media uploaded: indica la fine dell'invio di un file media (immagine, video, suono documento o messaggio vocale); Media downloaded: indica la fine del download di un file media ricevuto (immagine, video, suono documento o messaggio vocale);

Per ascoltare un suono premere sul bottone

Sounds						
abduction.mp3		^				
all-eyes-on-me.mp3						
another-message-has-a	arrived.mp3					
arpeggio.mp3						
arrived.mp3						
concise.mp3						
dropped.mp3						
exclamation.mp3		×				
	To play file in the above list,	select it.				
Sounds settings						
Notifications received :	concise.mp3	~ 🖸				
Unread notifications :	mission-accomplished.mp3	~ D				
Message sent :	all-eyes-on-me.mp3	~ 🖸				
Message received :	arrived.mp3	~ 🖸				
Media uploded :	hit.mp3	~ D				
Media downloaded :	proud.mp3	~ 🖸				

Immagine che mostra i parametri della sezione Sounds.

4) App syles

Questa sezione contiene i parametri che definiscono i parametri grafici degli oggetti che compongono l'app come lo **Splash form**, l'**Header**, il **Footer**, la **Toolbar**, la **Bottombar** ed il **Menu**.

4.1 Splash form

Lo splash form è il form che viene mostrato durante la fase primaria del caricamento dell'app. In genere contiene l'immagine del logo dell'azienda che sviluppa l'app. Questa immagine è definita nella sezione **Android System images & Icons** o nella sezione **iOS System images & Icons**. Lo sviluppatore deve inserire immagini di diverse dimensioni a seconda che l'app venga eseguita su uno smartphone o su un tablet. La scelta della dimensione dell'immagine è lasciata al sistema operativo.



Immagine dei parametri dello splash form.

Dopo lo splash form di sistema operativo, l'app carica uno splash form proprio durante il tempo di connessione con il database, il sito web e il login utente.

Sullo Splash Form vengono visualizzate alternativamente le immagini *Phone splash o Tablet splash* definite nella sezione di menu *Images/Fixed Images*.

In questo form in basso è mostrata una stringa di cui si devono definire Font e colori. Premendo il bottone **Customize** viene mostrato il relativo designer. Premendo **Apply** si conferma la scelta.

Questo designer è utilizzato per tutte le sezioni dell'app.

E' importante osservare che sulla sinistra è mostrato l'oggetto in live ossia

cambia al variare delle proprietà inserite.

Un trucco. Se vuoi nascondere la label di caricamento, assegnale lo stesso colore del *Background color* dell'app.

4.2 Header

Header è la sezione che controlla le *properties* dell'headerbar.



Sezione per la configurazione dell'headerbar.

Premendo il bottone **Customize** si apre l'editor delle proprietà. Poiché questo è intuitivo non ci dilungheremo oltre.

Cliccando sul bottone **Show** viene mostrata una finestra che indica dove questo modulo deve essere visualizzato. Ogni singola voce dell'elenco mostra un singolo plugin o sezione dell'app, abilitando il quale si consente all'app di mostrare il modulo nella sezione indicata.

Ad es. abilitando *Menu page* lo sviluppatore vuole mostrare l'headerbar nella pagina iniziale dell'app, mentre abilitando Camera plugin questa verrà visualizzata nel plugin per la gestione della foto camera.

Per visualizzare l'effetto ottenuto all'interno dell'app si può avviare l'emulatore che seguirà nella sua configurazione correntemente salvata. Premendo Debug/Run emulated version si avvia l'emulazione. Dopo alcuni istanti sarà visualizzato un device con l'app in esecuzione. Usare

l'app come se fosse su uno smartphone o tablet, per verificare il risultato della configurazione scelta.

4.3 Footer

Footer è la sezione che controlla le *properties* della footerbar.



Sezione per la configurazione della footer bar.

Le *properties* della footerbar sono analoghe a quelle della headerbar per cui la trattazione è simile.

La stringa che viene mostrata all'interno di questa sezione è definita in *Languages/System strings/Footer text*.

4.4 Toolbar

Molti plugin disponibili con I.B.C. App Maker hanno una toolbar propria. In questa sezione è possibile definire tutte le proprietà comuni alle varie toolbar come *font, colore di sfondo, colore in primo piano, colori della progress bar,* mostrata durante il caricamento delle pagine web.

La proprietà **Show** indica se mostrare le varie toolbar all'interno dell'app. E' fortemente consigliato di mantenere abilitata questa opzione altrimenti l'app potrebbe avere comportamenti indesiderati.



Sezione per la configurazione della toolbar.

4.5 Bottombar

In questa sezione si definiscono le proprietà della bottombar. Questa può essere intesa come una toolbar posta in basso all'applicazione che contiene dei bottoni per l'accesso diretto a tutte le funzioni di maggior utilizzo all'interno dell'app stessa.

Customize	Show	Items
Bottom bar		

Immagine della sezione di configurazione della bottombar.

Le sezioni **Customize** e **Show** sono state trattate in precedenza, ad esempio al punto <u>4.2</u>, per cui non ci dilungheremo oltre.

Premendo il bottone **Items** viene mostrato l'editor degli items che compongono la bottombar. Di seguito una sua immagine.

A sinistra, questo possiede una lista con gli items creati e a destra le *properties* di ciascun item.

La combobox **Text** mostra l'elenco dei codici delle *stringhe custom* (vedi sezione <u>Languages 3.2</u>) che lo sviluppatore ha definito. Questa serve per definire la Label dell'item in questione.

Bottombar Items editor			_	×
🕒 🛹 🧕 💾 🗳				
Items :	Editor			
Empty Text				
userdata	Te	ext :	userdata	~
camera	Plug	gin :	User.viewlogged-plugin	~
	lma	ige :		~
	Presentati	ion :		~

Immagine dell'editor della bottombar.

La combobox **Plugin** mostra l'elenco dei plugins disponibili che è possibile associare all'item che si sta creando o modificando.

La combobox **Image** mostra l'elenco dei codici delle immagini disponibili che è possibile associare all'item. Queste immagini sono definite nella sezione di menu *Images/Menu images*.

La combobox **Presentation** mostra l'elenco dei codici dei moduli Presentation, definiti nel relativo plugin, che è possibile associare all'item. Per la trattazione del plugin Presentation rimandiamo al <u>paragrafo 6.7</u>.

Per alcuni plugin deve essere definito il profilo utente che abilita tale funzionalità. La combobox **Profile**, se visualizzata, indica il profilo utente per cui deve essere mostrato il relativo bottone nella barra.

I bottoni **Show on Android** e **Show on iOS** indicano in quale sistema operativo deve essere mostrata la relativa voce di menu. E' facile notare

che si possono creare bottombar diverse a seconda del sistema operativo in cui sarà eseguita l'app.

A seconda del plugin selezionato nella combobox **Plugin**, possono essere richieste *properties* accessorie.

Database.Table.newrecord-plugin

Per questo plugin è visibile la property aggiuntiva *Tables* che indica la tabella di riferimento in cui si vogliono aggiungere i nuovi record.

Database.Table.searchrecord-plugin

Anche per questo plugin, come il precedente, è presente la property *Tables* che indica su quale tabella si vuole effettuare la ricerca dei record. I campi di ricerca sono quelli che hanno impostato l'attributo *search*, definiti nel plugin **Database**, e come operatore quello definito nella *property Default search* (vedere <u>capitolo 4</u>).

Database.Table.viewrecords-plugin

Questo plugin oltre alle property *Tables*, di cui sopra, presenta la property *PreFilter*. Come indica il nome stesso il PreFilter è un filtro che viene applicato alla tabella prima di mostrarne il contenuto. La condizione SQL (Where) da applicare è definita come proprietà della tabella stessa ed è trattata nel capito relativo al plugin <u>Database</u>.

Database.Table.viewrecord-plugin

Questo plugin oltre alla property *Tables*, di cui sopra, presenta la property *SQL params*. In questa property inserisci la condizione sql che consente di identificare univocamente un record all'interno di una tabella. Es. id=1, name="myname", name="myname" and surname="mysurname", email="(ACCOUNT.EMAIL)".

Database.Query.searchrecord-plugin

Selezionando questo plugin, viene mostrata la property *Query* che indica quale query eseguire.

I campi di ricerca sono quelli che hanno impostato l'attributo *search*, definiti nel plugin **Database**, e come operatore quello definito nella *property Default search* (vedere <u>capitolo 4</u>).

Database.Query.viewrecords-plugin

Questo plugin, come il precedente, mostra la *property Query*, che ha la stessa funzione. Una maggiore trattazione è rimandata al capito relativo al plugin <u>Database</u>.

Url

Il **plugin URL**, una volta selezionato, mostra le *properties Url* e *Profile*.

Url è l'indirizzo web della pagina da mostrare. Questo deve essere copiato dalla barra degli indirizzi del browser e definito per ciascuna lingua abilitata per l'app.

Cliccando sul bottone '…' relativo all'*Url* viene mostrato un'editor che permette di definire per ciascuna lingua una singola Url. Nelle applicazione web multilingua è frequente che le pagine nelle varie lingue siano indirizzate con url diverse. In questo modo il Device configurator permette allo sviluppatore di personalizzare tutti questi indirizzi centralizzando la gestione.

Il Device Configurator sostituisce automaticamente il dominio del sito web, indicato in *Configuration/General/Web site domain* con la Tag (DOMAIN).

Profile è il profilo utente per cui deve essere mostrato questo bottone.

Se, per esempio, si definisce il profilo '*Super User*' di Joomla *o* '*Administrator*' di Wordpress, non sarà possibile mostrare questo item a un utente con profilo *Registered*, *Public o SubScriber*.
Questo editor ha una toolbar con 4 bottoni:



Prepara l'editor per l'inserimento di un nuovo bottone.



Salva l'item corrente in memoria, aggiornando la lista posta sulla sinistra.

Cancella l'item selezionato dalla lista posta sulla sinistra.

P Salva su disco la configurazione corrente.

4.6 Menu

In questa sezione si definiscono le proprietà del menu laterale visibile premendo l'apposito bottone nella **Headerbar**.



Immagine della sezione di configurazione del menu.

La trattazione delle *properties* **Customize** e **Items** compresi i loro editors è uguale a quella della bottombar a cui rimandiamo integralmente. Per aggiungere un logo al menu definire l'immagine "Menu logo" nella sezione *Images/Fixed images*. Sotto un esempio di menu.

La property **Show** indica se il menu deve essere attivato oppure no. La property **Use images** indica se si vogliano visualizzare le immagini laterali alle singole voci di menu. Vedere immagine sottostante. I bottoni descritti sotto avviano la sessione di test dell'app e sono presenti solamente in I.B.C. App Maker e non nelle app con esso create.



Avvia la sessione di Test dell'app.

Chiude la sessione di Test (Presente solo nella versione iOS).



Menu applicativo con logo senza immagini laterali alle singole voci di menu.



Menu applicativo con la presenza di immagini laterali alle singole voci di menu.

4.7 Menu page

Menu page è la sezione con cui si definiscono la grafica e il contenuto della pagina di menu dell'applicazione. Si possono vedere due esempi sotto.



Immagine della sezione che permette di configurare la Menu page.

In questa sezione sono visibili 2 bottoni: *Items* e *Customize*.

La definizione delle voci di menu è comune al menu e alla bottombar a cui rimandiamo.

L'editor della property **Customize** permette la gestione dell'item come indicato nella figura sopra. Saltiamo questa parte perché comune ad altre. Eventualmnete consultare il <u>paragrafo 4.2</u>.

La *property* **Columns** indica il numero di colonne che deve avere la pagina. Il default per lo smartphone è 2, mentre per il tablet è 3.



Menu page in un'applicazione Android su tablet.

View on Multi page abilita la visualizzazione della **Menu page** nella modalità a pagina multipla. Impostando questa opzione è possibile visualizzare il menu su più pagine, potendo scorrerle col dito. In basso nella pagina è presente un bottone, premendolo si apre la pagina associata alla relativa voce di menù.



Screenshoot di menu Multi page in un'app Android su tablet.

4.8 Advertising bar

Advertising bar è la sezione che controlla le *properties* della barra per l'advertising Google con AdMob. Dopo aver creato un account Google, indirizzare il browser su <u>https://admob.google.com</u>, effettuare il login, aggiungere la propria app e creare un unità pubblicitaria di tipo **Banner**. Inserire la chiave ottenuta in *Banner Key* del plugin *Google ads* tramite il *Device configurator*. Usa la property *Show* per scegliere i plugins su cui mostrare la barra.



Immagine della posizione del banner all'interno del form dell'app

5) Controls' styles

In questa sezione sono descritte le proprietà di tutti i controlli che possono costituire l'app. Gli stili sono di 2 tipi *Defaults* e *Customs*.

Gli stili di *Default* definiscono font, colore, allineamento, dimensioni ... per i vari controlli che sono uguali in tutta l'app mentre gli stili *Custom* definiscono le stesse proprietà applicabili, però, solo ad alcuni controlli in particolare. Questi sono i controlli che definiscono i campi del database. Lo sviluppatore, se necessario può attribuire, nella fase di data entry delle varie tabelle un particolare stile ad un campo per dargli maggiore risalto o semplicemente per distinguerlo dagli altri.

5.1 Buttons

La sezione Buttons permette di gestire le *properties* grafiche dei bottoni dell'interfaccia applicativa. Il bottone *Customize* apre l'editor per la configurazione delle *properties* del controllo.



Immagine della sezione Buttons per la gestione delle properties dei bottoni dell'app.

5.2 Checkbox

La sezione **Check box** permette di gestire le *properties* grafiche dei *check box* dell'interfaccia applicativa. Il bottone *Customize* apre l'editor per la configurazione delle *properties* del controllo.

5.3 Edit

La sezione **Edits** permette di gestire le *properties* grafiche delle linee di input dell'interfaccia applicativa. Il bottone *Customize* apre l'editor per la configurazione delle *properties* del controllo.

5.4 Fixed text

La sezione **Fixed text** permette di gestire le *properties* grafiche dei campi di tipo *TFixedField* del database all'interno dell'interfaccia applicativa. Il bottone *Customize* apre l'editor per la configurazione del font e del colore. Per vedere il funzionamento di questo tipo di controllo cliccare <u>qui</u>.

5.5 Form details area

La sezione **Form details area** definice le *properties* grafiche dei bottoni che gestiscono il master/detail nelle tabelle di database. Ad es. se lo sviluppatore volesse creare una tabella *master*, ad es. classe, ed una tabella *detail* alunni, per creare un'app che gestisca le classi di una scuola e tutti gli alunni appartenenti ad ogni singola classe, nella tabella master dovrebbe inserire dei bottoni che si occupano dell'inserimento o della modifica del record nella tabella detail. In questa sezione si definisce, appunto, lo stile di questi controlli. Per il trattamento della gestione del master/detail, che deve essere già noto allo sviluppatore, rimandiamo al plugin <u>Database</u>.

5.6 Form filter

Spesso accade che sia necessario applicare un filtro ai dati di una tabella. Il **Form filter** definisce lo stile grafico della sezione che contiene i bottoni con i vari filtri definiti nel plugin <u>Database</u>.



Immagine che mostra la sezione Form filter.

Nell'immagine sopra viene indicata l'area adibita al **Form filter** in cui sono definiti 3 diversi filtri. Premendo su ciascun bottone, individuato da una Label, il contenuto del form sarà automaticamente modificato.

5.7 Groups

La sezione **Group** permette di gestire le *properties* grafiche dei *groupbox* dell'interfaccia applicativa. Il bottone **Customize** apre l'editor per la configurazione delle *properties* del controllo.

Customize	
☑ Align frame to page	
Group label	

Sezione che permette di gestire la configurazione generale dei gruppi dell'app.

I gruppi racchiudono logicamente una serie di controlli in una data area del form. Ricorrono in varie parti dell'app, tra cui l'editing dei dati utente, dell'azienda e nell'*Instant Messaging*.

Impostando la property *Align frame to page* il gruppo automaticamente assume la larghezza del form.

5.8 Labels

La sezione **Labels** permette di gestire le *properties* grafiche delle *Labels* dell'interfaccia applicativa. Il bottone **Customize** apre l'editor per la configurazione del font e del colore.

5.9 Links

La sezione **Links** permette di gestire le *properties* grafiche dei *links* dell'interfaccia applicativa. Il bottone **Customize** apre l'editor per la configurazione delle *properties* del controllo.

5.10 List item

La sezione **List item** definisce il colore di sfondo delle liste dell'applicazione. Queste definiscono il menu laterale, i form di database ed altro.



Negli esempi sopra si può osservare l'effetto del cambiamento del colore da White a Null. Il colore Null permette la disabilitazione del pennello e quindo di visualizzare i contenuti senza un colore di sfondo.

5.11 List title, List value, List detail

La sezioni **List title**, **List value** e **List detail** permettono di gestire le *properties* grafiche delle liste di record all'interno del plugin **Database**. Il bottone **Customize** apre l'editor per la configurazione delle *properties* dei controlli.

ø

Immagine che mostra i vari componenti di una lista.

Nell'immagine sopra, è riportato un esempio di applicazione Android in cui sono elencati, in un form del plugin **Database**, i valori dei record di una tabella. In rosso sono indicate le varie parti che formano la lista, rispettivamente *List title*, *List value e List detail*.

5.12 Memo

La sezione **Memo** permette di gestire le *properties* grafiche dei campi memo negli stati Insert/Update dell'interfaccia applicativa. Il bottone **Customize** apre l'editor per la configurazione delle *properties* del controllo.

5.13 Message

La sezione **Message** permette di gestire le *properties* grafiche delle Label Message nei form di Login/Register utente e di quelle presenti nel plugin Presentation. Il bottone **Customize** apre l'editor per la configurazione delle *properties* del controllo.

Language :	
English	~
Email :	
Insert your email add	ress
Password :	
•••••	Q
All	
 Allow sending of letters and notific 	emails, new ations
□ Allow sending of letters and notific Register	emails, new ations Login
 Allow sending of eletters and notific Register If you are not regis account press residues 	emails, new ations Login Message tred to create an gister button.
Allow sending of eletters and notific Register If you are not regis account press re	emails, new ations Login Message tred to create an gister button. ess Company. All righrs a ed.

Immagine che mostra la posizione delle Label Message

5.14 Page title

La sezione **Page title** permette di gestire le *properties* grafiche dei titoli delle pagine nei vari tabsheet. Il bottone *Customize* apre l'editor per la configurazione delle *properties* del controllo.

5.15 Status field

La sezione **Status field** permette di gestire le *properties* grafiche delle labels impiegate per comunicare gli errori e di quelle usate nei campi TStatusField del plugin Database. I controlli di tipo TstatusField sono presenti nel plugin *Instant Messaging & Users* per gestire il blocco dell'utente o la sua segnalazione. Il bottone **Customize** apre l'editor per la configurazione delle *properties* del controllo.

5.16 Custom fields

I **Custom fields** sono controlli per cui lo sviluppatore può definire degli stili personalizzati ed assegnarli a singoli componenti che, in questo caso, non seguono gli stili di deafult.



Immagine del menu di popup per la gestione dei Custom fields

Facendo click con il tasto destro del mouse su Customs viene visualizzato il menu nell'immagine sopra. Cliccando su ciascuna voce *Create* si crea uno stile a cui deve essere assegnato un nome. Selezionando lo stile viene visualizzato la sua pagina attributi, dove con il relativo editor, mostrato con la pressione del bottone **Customize**, è possibile definirne i parametri.

Questi stili vengono usati all'interno del plugin **Database**, dove possono essere associati a ciascun campo di una tabella. Lo sviluppatore può definire diversi stili per il singolo campo a seconda che questo si trovi nello stato *View* o *Edit*. Da notare che, per l'applicazione degli stili, lo stato di *Edit* vale sia quando si inserisce un nuovo record sia quando il record è in modifica.

Gli stili di tipo checkbox, edit, label e memo possono essere applicati al form che mostra il singolo record di una tabella negli stati di visualizzazione, inserimento o modifica. Mentre gli stili di tipo form filter e list item si applicano al form che mostra i record di una tabella. In questo modo la lista dei record presenti in una determinata tabella può essere personalizzata e resa differente da tutte le altre. Se lo sviluppatore vuole creare tabelle con una interfaccia differente l'una dall'altra deve seguire questo schema.

6) Plugins

6.1 Registration/Login

La sezione **Registration/Login** permette di configurare il plugin di registrazione e/o login utente.

🗹 Enable	
Enabled GDPR consent Force GDPR consent	
Show registration	
☑ Show labels	
☑ Show prompt	
Detailed registration (Recommended for Wordpress)	
○ Single button	
OTP registration - Smtp parameters	
ibc	~

Sezione che permette di gestire la configurazione del plugin per la registrazione e login utente.

Descrizione delle *properties*:

- **Enable:** la *property Enable* abilita il plugin per la registrazione ed il login utente;
- Enable GDPR consent: la property Enable GDPR consent visualizza un check box che chiede all'utente conferma dell'accettazione in merito alla ricezione di news letter, email o messaggio a scopo pubblicitario. Il checkbox utilizza la stringa Languages\System language\GDPR consent.
- Force GDPR consent: la *property Force GDPR consent* non consente la registrazione dell'utente se non acconsente all'accettazione delle norme GDPR.
- Show registration: mostra la pagina di registrazione all'utente, se questo non risulta registrato. Questa pagina viene mostrata anche qualora un utente precedentemente registrato disinstallasse l'applicazione.

- **Show Labels**: mostra/nasconde le *Labels* della pagina di registrazione/login utente;
- Show Prompt: mostra/nasconde i prompt dei campi della pagina;
- **Detailed Registration**: mostra la pagina di registrazione/login avanzata gestendo, oltre a '*Email*' e '*Password*', anche lo '*Username*';
- **Single Button** o **Buttons**: queste due *property* configurano il plugin per due diverse modalità di funzionamento. La prima abilita un solo bottone per pagina e la possibilità di cambiarla da quella di registrazione a quella di login, cliccando sulla *Label di messaggio;* la seconda ne abilita due: uno per la registrazione ed uno per il login utente.
- **OTP registration Smtp parameters** : mostra la finestra con cui definire tutti i parametri per l'utilizzo di un server smtp per l'invio di email.

	>	+ +	_	\checkmark	X	0
Name :						
SMTP host :						
User name :						
Password :						
Port :		🗹 Us	e SSL			
M Test S	MTP		Cancel		Close	2

Immagine dell'Smtp editor.

L'editor mostra le seguenti proprietà:

- **Name** : nome del smtp per la selezione nella relativa combobox;
- **Smtp host** : host della casella email da utilizzare;
- **User name** : utente della casella email ovvero indirizzo email per l'invio delle email;
- **Password** : password dell'utente per l'accesso alla casella email;
- **Port** : porta della casella email;

• **Use SSL** : indica se usare il protocollo ssl.

Premendo il bottone *Test* vine inviata una email di prova all'indirizzo indicato per verificare la correttezza delle impostazioni. Se le impostazioni fossero errate viene mostrato un messaggio.

La funzione OTP è stata inserita per validare l'utente che vuole registrarsi sulla app.

(
	TP S	
Email : Insert your email address		
Password : Insert your password		
Show password		
If you are not reg	istred to create an a	account click here.
	Register	

Esempio di pagina di registrazione per tablet in un'app Android.

Email : Insert your email addres:	8
Password : Insert your password	
Show password	
If you are alre	eady registered for login click here.

Finestra di login/registrazione dell'app su un tablet.

Earlyuage .	
English	
Email :	
Insert your email ad	dress
Password :	
Insert your passwor	d
Register	Login
If you are not regis	stred to create a

Finestra di login/registrazione dell'app su uno smartphone.

IMPORTANTE ! Wordpress necessita dell'attivazione della Detailed Registration, per un corretto funzionamento.

6.2 Camera

Il plugin **Camera** permette di gestire le immagini della gallery del dispositivo e le foto scattate. È possibile attivare questo servizio per una determinata categoria di utenti tramite la *property* **Profile**.

🗹 Delete images	☑ Share images	
🗹 Enable Galley	🗹 Enable Photo	
✓ Enable effects		
Enable for profile :	All	~

Sezione che permette di configurare il plugin 'Camera'.

Descrizione delle *properties*:

- **Delete images:** mostra/nasconde il bottone che permette la cancellazione delle immagini;
- **Share images**: mostra/nasconde il bottone per la condivisione dell'immagine;
- **Enable Gallery:** mostra/nasconde il bottone che permette l'importazione delle immagini dalla gallery;
- Enable Photo: mostra/nasconde il bottone che permette l'accesso alla photo camera;
- **Enable effects:** mostra/nasconde il bottone che mostra il menù con i gli oltre 60 filtri grafici disponibili;
- **Enable for profile**: indica i profili utente abilitati a utilizzare la funzionalità **Camera**.

Per usare un filtro grafico su un'immagine (vedi figura sotto), seleziona la categoria filtro (*filter category*), dall'elenco a destra, quindi il filtro desiderato (*filter type*). Successivamente imposta le opzioni del filtro tramite le *filter options* sulla sinistra. Ogni tipologia di filtro possiede un numero diverso di controlli che ne gestiscono le proprietà. Ogni volta che si seleziona un filtro questo viene applicato all'immagine direttamente. Nel caso in cui più filtri vengano selezionati in serie, ciascuno di questi sarà applicato all'immagine ottenuta applicando il filtro precedente. Se desideri

annullare l'effetto di un filtro, clicca su *None*. Ogni elenco di *filter types* possiede come prima voce un elemento *None*.



Plugin camera ed i filtri disponibili per modificare le immagini

Elenco funzioni dei bottoni:

Bottone Funzione

- Cancella l'immagine corrente
- Condivide l'immagine corrente
- Apre la galleria per importare un'immagine
- Accede alla foto camera, scatta una foto e la salva

Elenco funzioni della toolbar degli effetti:

S	Ruota l'immagine di 90 gradi
•	Salva l'immagine corrente
\bigcirc	Ricarica l'immagine

6.3 Companies

Il plugin **Companies** è legato alla gestione *Instant Messaging*. Con questo plugin è possibile gestire i dati aziendali di un'impresa o ente pubblico che voglia usare la chat di I.B.C. App Maker.

Corporate data	🎽 🔞 🗹 (
COMP	
Company :	
Insert your company name	
Web site :	
Insert your web site U.R.L.	
Country name :	
Insert your country name	
Locality :	
Insert your locality	
Postal code :	
Insert your postal code	
Address :	
Insert your address	
Share your address	

Pagina di inserimento dati di un'azienda in un'applicazione Android.

Per personalizzare i campi di questa tabella utilizzare il plugin Database che contiene 2 tabelle di sistema proprie di I.B.C. App Maker. Queste sono *ibc_companies_data* e *ibc_users_data*. Le tabelle vengono create sul database del CMS all'istallazione del plugin con una propria struttura di

default che può essere modificata dallo sviluppatore utilizzando il plugin **Database**.

Le funzioni della toolbar sono comuni a tutte le tabelle database gestite dal relativo plugin:

Bottone	Funzione Apre la galleria per importare un'immagine
0	Accede alla foto camera, scatta una foto e la salva
H.	Esegue il salvataggio dei dati
	Abilita il GPS e carica i dati ottenuti nei vari campi di tipo GPS

Per ogni sezione dati è possibile specificare se questa debba essere condivisa con altri utenti oppure no. Se i dati di una sezione non sono condivisi, nessuno può vederli e restano riservati.

6.4 Documents

Poter pubblicare contenuti senza dover accedere al web è una prestazione sempre più richiesta dagli sviluppatori. Per questo è stato creato il plugin **Documents** che consente la gestione di contenuti a partire da file .doc o .docx. Questi files devono essere creati come se fossero testi web, con un titolo, un testo ed eventuali immagini. Pre tradotti nella lingua di destinazione, devono essere inseriti nel Documents editor (visibile sotto). Il Device configurator provvederà alla loro conversione automatica, tramite l'utilizzo di Open Office, Libre Office o Microsoft Office, preinstallati sul pc, in file .html. La gestione di questi files è del tutto trasparente per lo sviluppatore che si deve limitare ad inserire un codice nel campo code, e il nome del file con relativo path nell'item della lingua interessata.

Documents					_	
🕀 🧕 💾	🝺 📂					
Code	en (English)	it (Italiano)	es (Español)	fr (Français)	pt (Português)	de (Deutsch)
aboutus	C:\AppMaker Pr					
createapp	C:\AppMaker Pr					
joomla	C:\AppMaker Pr					
magento2	C:\AppMaker Pr					
wordpress	C:\AppMaker Pr					
prestashop	C:\AppMaker Pr					
php	C:\AppMaker Pr					
					Close	Cancel

Immagine del Documents editor.

6.5 Html

Un altro modo per poter pubblicare contenuti ipertestuali è quello di creare delle pagine .html da inserire direttamente nell'app. Il codice non risiederà su un server ma direttamente sul device per cui risulterà particolarmente veloce. Questi file possono essere creati con qualsiasi strumento come editor di testo o editor wysiwygh. Sul nostro sito web potete scaricarne uno che riteniamo semplice e intuitivo. E' necessario creare un contenuto per ciascuna lingua, come i per documenti, se si sta creando un app multilingua. La gestione di questi file è del tutto trasparente per lo sviluppatore che si deve limitare ad inserire un codice nel campo code e il nome del file con relativo path nell'item della lingua interessata.

6.6 Instant messaging & Users

Il plugin **Instant messaging & Users** gestisce la chat di I.B.C. App Maker. Questo servizio è molto simile a Whatsapp o Telegram. Questa sezione consente di configurare le proprietà del plugin.

Instant messaging



Immagine della sezione che permette di configurare il plugin 'Instant messaging & Users'.

Descrizione delle *properties*:

- Chant font: permette la modifica del font delle sezioni *Send messages*, *Received messages* e *Blocked messages* che indicano relativamente i messagi inviati, i messaggi ricevuti ed i messaggi bloccati.
- **Info Font**: permette la modifica del font dei messaggi informativi della chat come ad esempio '**Nuovi messaggi**'.
- **Media font**: permette la modifica del font dei messaggi relativi ai media come suoni, video, documenti o messaggi vocali.
- **Stickers** : mostra ed aggiunge gli stickers all'app. N.B. Per caricare gli stickers sul server copiare i file .zip dei pacchetti da utilizzare nella cartella https://www.mydomain/device/stickers.
- **Show companies** : indica se abilitare il plugin **'Companies'** e mostrate le aziende definite all'interno dell'app.
- **Auto load companies** : indica se caricare le aziende alla selezione della relativa pagina oppure attendere un filtraggio.

- **Show users** : indica se mostrare l'elenco degli utenti registrati sulla app automaticamente alla selezione della relativa pagina oppure attendere un filtraggio.
- **Auto load users** : indica se caricare automaticamente gli utenti alla selezione della relativa pagina oppure attendere un filtraggio.
- **Encrypt chat**: indica se criptare i contenuti delle chat oppure no. Impostando questo attributo saranno criptati l'invio e la ricezione dei messaggi, il database dei messaggi sul server o sul device e tutti i file immagine, video, documenti, audio o messaggi vocali utilizzati dalle varie chat. In questo modo le **chat risulteranno inaccessibili**.
- Use contact for private chat : indica se utilizzare i contatti selezionati presenti sul proprio device. In questo caso i contatti indicati veranno inviati al server e se questi avranno installata l'app saranno abilitati per l'invio dei messaggi.

La *property* **Show users** consente l'utilizzo dell'app come un social, infatti sarà possibile inviare messaggi a tutti gli utenti presenti sul database.

La *property* **Use contact for private chat** consente l'utilizzo dell'app in modo privato (come whatsapp), infatti sarà possibile inviare messaggi SOLO agli utenti di cui possediamo il numero di telefono.



Immagine della toolbar del plugin Instant messaging.

Per filtrare gli utenti presenti nella chat usare i bottoni sotto.



Mostra un elenco con il nome della proprietà con cui si vuole effettuare la ricerca degli utenti. Ogni proprietà indica un campo database definito nel relativo plugin. Per abilitare la ricerca di un campo impostare la proprietà **'Find in form'** nelle impostazioni del field della tabella.



Mostra una riga di input su cui scrivere il testo da ricercare nella proprietà relativa al campo definito sopra.

Aggiunge, se abilitato l'anagrafica di una nuova azienda. Questo bottone viene visualizzato se abilitata la *property* **Show companies**

Per impostare il colore del font e dello sfondo di ciascuna sezione, fare un doppio click sulla sezione desiderata e si aprirà un editor con cui effettuare l'operazione.



Pagina relativa al plugin Instant Messaging & Users in un'app Android.

Cliccando su *Send messages*, *Receive messages*, *Date messages*, *Info messages e Hour & data font color* viene mostrato il property editor per personalizzare font e colore dei messaggi all'interno della chat.

Il plugin *Instant Messaging & Users* è stato pensato per inviare messaggi di testo, immagini, audio, video e documenti real time agli utenti della rubrica aziendale.

La pagina si presenta come nell'immagine sopra. Il tabsheet *Companies* viene mostrato solo se il profilo dell'utente loggato è compatibile con il profilo *Manage profile* del plugin **Companies**.

Nella pagina *chat* viene mostrato l'elenco di tutte le chat attive. Cliccando sul nome utente si apre la chat, mentre cliccando sulla sua immagine vengono visualizzati i dati pubblici del suo profilo.

Le pagine *Users* e *Companies* permettono di cercare utenti ed aziende presenti nella rubrica secondo criteri impostati dall'utente stesso. È possibile aggiungere un utente o un'azienda alla pagina *Favorites* cliccando sulla stella in alto a destra nella pagina dati utente. Le chat possono essere avviate solo con gli utenti e non con le aziende di cui si visualizzano solo i dati pubblici.

Elenco funzioni della toolbar:



Permette la cancellazione delle chat e degli utenti/aziende dalla pagina *Favorites*



Avvia la ricerca utenti ed aziende nelle relative pagine



Se abilitato permette di inserire i dati aziendali da parte dell'utente abilitato.



Imposta i criteri di ricerca utenti/azienda nelle relative pagine

Abilita/disabilita la presenza di un utente/azienda nell'elenco dei *Favourites*.

Premendo il bottone per l'impostazione dei criteri di ricerca viene mostrato l'elenco visibile nell'immagine sotto.



Elenco con i criteri di ricerca utente/azienda in un'app Android.

Il criterio di default è *Name*, ma possono essere selezionati dall'utente anche gli altri. Una volta scelto il criterio di ricerca, è sufficiente cominciare a digitare il testo relativo alla ricerca desiderata: questa verrà avviata alla pressione del bottone di ricerca effettuando una *ricerca per testo contenuto* nei campi del database contenenti le informazioni desiderate (se pubbliche per il singolo utente). Se un utente non rendesse pubblico il proprio indirizzo, effettuando una ricerca per *Locality*, risulterebbe impossibile trovarlo.

È evidente che questa gestione della chat non è semplice messaggistica, ma permette il contatto tra un utente di un'azienda/una comunità professionale anche con sconosciuti, come in un *social media*.

Il singolo utente non deve conoscere il numero di telefono o la email di un soggetto per comunicare con lui, ma è sufficiente che tale utente sia registrato e pubblico all'interno dell'app che ospita il plugin *Instant Messagging & Users*.

Per personalizzare i campi della tabella utenti, ossia la tabella che contiene i dati degli utenti della chat è necessario utilizzare il plugin Database. La tabella di riferimento è ibc_users_data. Ogni modifica a questa tabella si riflette automaticamente nella gestione del plugin, **pertanto si consiglia di stare molto attenti alle modifiche da effettuare**.

6.7 Location

Il plugin **Location** consente la gestione di mappe e la ricerca di attività presenti su Google Maps in base ad elementi preconfigurati.

Questo plugin è utile in quelle app dove si trattano categorie merceologiche specifiche e si richiede che l'utente possa cercarle nella sua zona.

Per esempio, se volessi creare un'app per articoli sanitari e la distribuissi ad agenti di commercio, questi vorrebbero cercare nella loro zona i vari ospedali o i medici per raggiungerli col navigatore e/o contattarli telefonicamente.

Se invece creassi un'app per ristoranti, alberghi ... potresti aiutare gli utenti nella prenotazione e nella scelta, poiché il plugin mostra all'utente il rating assegnato all'attività e i commenti lasciati dagli utenti stessi.

Search items		
Show topics		
Enable for profile :	Registered	~

Sezione che permette di configurare il plugin 'Location'.

Descrizione delle properties:

- **Show topics:** mostra/nasconde la pagina *Near me* con cui è possibile ricercare attività commerciali secondo i parametri impostati tramite la *property* **Search items** e selezionarle come in figura sotto;
- **Search items**: definisce i parametri di ricerca delle attività commerciali presenti su Google Maps;
- **Enable Location for profile:** indica il profilo utente abilitato all'utilizzo di questo plugin;

Nell'immagine sotto è rappresentata la pagina *Near me*, così come appare sull'interfaccia utente. Puoi vedere dove sono indicati i *Topics* con cui effettuare la ricerca e la sezione dedicata ai risultati della ricerca stessa. Cliccando sul singolo item nella lista dei risultati della ricerca, viene

mostrata una pagina con i dati dell'attività selezionata: nome, indirizzo, numero di telefono e sito web (se pubblici). Inoltre, se il dispositivo lo consente, è possibile telefonare direttamente, accedere al sito web o, tramite il navigatore, essere guidati presso l'indirizzo dell'attività in questione.



Risultato di una ricerca per topics nella zona dell'utente in un'app Android.

Cliccando sul bottone **Search items** si apre il form visibile sotto. A sinistra viene mostrata una lista con l'elenco di tutti i topics creati, a destra un editor con 3 *properties* per definirli.

• **Text:** mostra un elenco di codici di stringhe custom che possono essere associate come labels al topic in questione. Un esempio sono Bars, Restaurant... visibili nell'immagine. Le labels nell'app sono visualizzate nella lingua corrente.

- **Category:** è un elenco di categorie definite da Google che raggruppa le varie attività registrate su Maps.
- Search for: come text mostra un elenco di codici di stringhe custom che però sono usate per effettuare la ricerca su *Maps*.



Editor dei topics del plugin Locations.

Se volessi visualizzare tutti i ristoranti presenti nella zona dell'utente, secondo il raggio impostato nella pagina sensor del plugin, dovrei prima definire un text per es. restaurant (testo definito nelle stringhe customs), poi una categoria: restaurant ed infine potrei omettere il campo *search for*. In questo modo, usando la categoria *Restaurant di Google*, il risultato sarebbe costituito da tutti i ristoranti nella zona dell'utente secondo il raggio nella pagina *Sensor*.

Se invece volessi cercare tutte le pizzerie dovrei impostare 'text = pizzeria', 'category = restaurant' e 'search for = pizza' o 'pizzeria'. Ricorda che prima dovresti definire tra le *custom strings* 'pizza' o 'pizzeria' nelle varie lingue. Al momento della ricerca su Maps, oltre ad effettuare una ricerca per categoria merceologica, verrebbe effettuata anche una ricerca per testo, inserendo la traduzione del codice definito per 'pizza' o 'pizzeria' nella lingua del device.

L'editor ha un funzionamento simile al *Menu items editor* o al *Menu page items editor* a cui rimandiamo.

6.8 Presentation

Il plugin **Presentation**, ultimo arrivato sulla piattaforma I.B.C. App Maker consente allo sviluppatore di creare app dove prima dell'utilizzo di un qualsiasi plugin o della app stessa, l'utente possa essere informato sulle funzionalità che sta per utilizzare, avere un eventuale help documentale e solamente se disposto a procedere può continuare con l'utilizzo della funzionalità selezionata.

Presentation editor	- 0	Х
💐 🛹 🧕 🔂 🛷	🔟 📔 🔁 😂	
Presentations :	Component	
Main ^		
AboutUs	Name : Main	
Joomla Wordpress	Title : whats_appmaker	~
Magento2 Prestachon	Description : whats_message	~
	Document help :	~
Pages :	Html help :	~
Demo debug	Url Help :	~
languages	Button text : continue	~
no_coding	Show "Do not show again" check box	¢
	Check style : ListDetail	~
\sim \checkmark		

Immagine del Presentation editor. Parametri generali.

La configurazione è molto semplice e richiede l'inserimento dei dati elencati sotto che identificano i parametri generali della presentazione.

- **Name** : nome/codice della presentazione elencato nel form di configurazione del menu o della Menu Page;
- **Title** : titolo della presentazione, selezionato tra le Custom Strings e visibile sulla toolbar della stessa;
- **Document help** : eventuale documento, scelto tra l'elenco dei disponibili, adibito ad una funzione esplicativa delle funzioni del plugin che si sta utilizzando;
- **Html help** : eventuale file html, scelto tra l'elenco dei disponibili, adibito ad una funzione esplicativa delle funzioni del plugin che si sta utilizzando;
- **Url help** : eventuale url, scelta tra l'elenco delle disponibili, adibita ad una funzione esplicativa delle funzioni del plugin che si sta utilizzando;
- **Button text** : testo del bottone presente nel form che consente di procedere con la funzionalità scelta;
- **Show** : indica se mostrare all'utente o no il checkbox '*Dont'show again*' che, se premuto, consente all'utente di non visualizzare più la funzionalità selezionata;
- **Check style** : stile del checkbox scelto tra i Custom Styles definiti nell'apposita sezione.

Presentation editor		_	×
💐 🛹 🧕 🔂 🛷	🗕 💾 😂		
Presentations :	Component Pages		
Camera			
Main	Page name : page		
	Image : Menu logo		\sim
	Title : login		~
\sim \sim			
0	Title style : Presentation_Title		\sim
rages :	1		
page	Message : login		
	Message style : Presentation		~

Presentation editor. Parametri per la definizione delle singole pagine.

Dopo aver configurato questi parametri, l'utente, per ciascuna pagina, deve definire i parametri seguenti :

Page name : nome della pagina che consente di distinguerla nella lista sulla destra;

Image : immagine, scelta tra le Fixed Images, mostrata al centro del form; **Title** : label mostrata sotto l'immagine, selezionata tra le varie Custom Strings;

Title style : stile della label di cui sopra;

Message : stringa, selezionata tra le Custom Strings, identificante il messaggio utente;

Message style : stile del messaggio sopra citato.



Immagine del plugin **Presentation** all'interno di un app emulata.

Il bottone **A**, nell'immagine sopra, presente solamente se viene impostata la property *Help*, mostra il documento identificato dalla stessa proprietà, quando viene premuto dall'utente.

Il bottone **B** viene mostrato solamente se impostata la property *Show "Do not show again" checkbox.*

Tale bottone indica se mostrare il form della Presentation ogni volta oppure interromperne la visualizzazione dopo la richiesta dell'utente.

6.9 Security

Il plugin **Security** è stato creato per dare agli utenti il massimo della sicurezza possibile nell'utilizzo della propria app.

Security		
✓ Enabled		
SIM checker	🗹 Trojan checker	
☑ User Security block	App block	

Immagine delle impostazioni del plugin Security.

Il plugin permette le seguenti impostazioni:

- **Enabled**: abilita il plugin permettendo tutte le funzionalità ad esso connesso ed attivate.
- **SIM Checker**: se abilitato, consente l'uso dell'app solo se il device non possiede una sim card installata. Questo plugin è essenziale se si desiderasse creare un'app per evitare il rilevamento della posizione da parte di terzi, l'intercettazione delle telefonate e l'istallazione sul device di trojan mediante sms inviati da soggetti terzi. Le app come Whatsapp e Telegram funzinano con un accint basato sul numero di telefono, I.B.C. Maker basa il proprio account su una email, permettendo all'utente di essere anonimo e che il proprio device non venga associato a nessuna persona fisica.
- **Trojan checker**: se abilitato permette la rilevazione di eventuali app installate dopo l'ultimo utilizzo. Queste app possono essere state installate volontariamente o no come i trojan. Per un corretto utilizzo, inserire a menu i plugin *Security.Info.SystemApp-plugin* e *Security.Info.Packages-plugin*. Questi consentono all'utente di riconoscere le app di sistema o da lui installate come valide e quindi da non segnalare quando necessario. Per selezionare le app da riconoscere, tenere premuto il dito su una voce in modo prolungato e si abiliterà la modalità di selezione. Al termine premere il bottone pe salvare l'operazione. Se l'utente non seleziona nessuna app come riconosciuta e quindi valida, il *Trojan checker* non viene attivato.
- Apps block: blocca la funzionalità di tutte le app correntemente in uso per non consentire fughe di dati da parte di eventuali spyware.
 I.B.C. App Maker blocca questa funzione e le rende disponibile, per motivi di sicurezza, SOLO NELLE APP CREATE DAI PROPRI UENTI.
- User Security block: se abilitato permette l'attivazione del gruppo di plugin *Security.Block.***-plugin*. Questi plugin devono essere inseriti nella **MenuPage** o sul **Menu** laterale per essere utilizzati. Lo scopo di questi plugin è quello di bloccare l'attività degli utenti se qualcuno tra questi ravvisasse la necessità di blocco di tutte le comunicazioni per un periodo limitato o in modo permanente.

I plugin in oggetto sono 4 :

- 1. *Security.Block.1hour-plugin*: blocca tutti gli utenti per 1 ora;
- 2. *Security.Block.12hour-plugin*: blocca tutti gli utenti per 12 ore;
- 3. *Security.Block.24hour-plugin*: blocca tutti gli utenti per 24 ore;
- 4. *Security.Block.Permanent-plugin*: blocca tutti gli utenti in modo permanente. Questi potranno riprendere la loro attività dopo che l'app sarà riabilitata dallo sviluppatore.

L'utilizzo congiunto di tutte queste funzioni permette di creare un'app sicura e **libera da trojan, spyware e intercettazioni non desiderate**.

6.10 Urls

Il plugin **Urls** consente la rapida conversione di un sito web in un'app nativa. E' sufficiente elencare le varie urls nell'apposito editor nell'item relativa alla lingua interessata. Queste url saranno automaticamente gestite da I.B.C. App Maker quando deve accedere a contenuti web utilizzando la lingua indicata dal device.

Urls for diction	ary	- 0	×
🕀 🗟 💾			
Code	en (English)	it (Italiano)	
AboutUs	C:\App Maker\Documents	C:\App Maker\Documents\	^
HowCreateApp	(DOMAIN)/en/how-to-mo	(DOMAIN)/en/how-to-mo	
Features	(DOMAIN)/en/platforms-c	(DOMAIN)/en/platforms-c	
Joomla	(DOMAIN)/en/platforms-c	(DOMAIN)/en/platforms-c	
Magento2	(DOMAIN)/en/platforms-c	(DOMAIN)/en/platforms-c	
Prestashop	(DOMAIN)/en/platforms-c	(DOMAIN)/en/platforms-c	
Wordpress	(DOMAIN)/en/platforms-c	(DOMAIN)/en/platforms-c	~
		Close Ca	ancel

Immagine dell'Urls editor

Il funzionamento dell'editor è simile a quello del plugin Documents paragrafo 6.4.

6.11 AppInApp

Il plugin **AppInApp** è costituto da due funzionalità *AppInApp.Runnerplugin* e *AppInApp.Container-plugin*. Il primo individua un'app da lanciare direttamente mentre il secondo apre una finestra con un elenco di app che possono essere eseguite dall'utente a sua richiesta. Il vantaggio di questa tecnologia è che permette l'utilizzo di più app sul device senza che queste vengano installate sullo stesso e che consumino spazio sul disco. L'istallazione di molte app sul device crea spesso confusione e consente lo spreco di spazio che potremmo usare per altro. In questo modo lo sviluppatore può creare app contenitore di altre app. Questo plugin risulta molto comodo per e-commerce di piccole aziende che possono condividere gli utenti tra le varie app e le politiche commerciali tramite le notifiche. Un altro utilizzo è quello di creare app contenitore per enti pubblici. In questo modo l'utente non deve istallare sul proprio device molte app ma una soltanto.

Menu Page items editor [Android-Phone] —		
🔁 🛹 🧕 💾 🗢		
Items :	Editor	
aboutus		
joomla	Text : webuc	~
wordpress		~
magento2	Plugin : AppinApp.runner-plugin	•
prestashop	Image : Webuc	~
javaphp		
camera	Profile : All	~
Maps	Presentation AppInAppRunner	~
notifications		
chat	Store code :	
userdata		
database		
system_applications		
installed_applications		
webuc	1	
appcontainer		
^		

Editor di menu con i parametri di AppInApp.Runner-plugin

E' rilevante osservare la presenza della *property* **Store Code**. Questa viene assegnata automaticamente dal sito web di I.BC. App Maker quando l'app viene registrata. Questo parametro individua in modo univoco la nostra app quando l'acquistiamo sullo store oppure in questo caso.

Creare un'app conteiner è molto semplice. Basta usare l'editor del plugin. Questo è simile a quello dei menu o della BottomBar, i parametri sono identici per cui non ci dilungheremo oltre.

6.12 Google interstitial advertising

La pubblicità sul web e all'interno delle app offre notevoli possiblità di sviluppo agli e-commerce e di visibiltà per le aziende. Le app che contengono pubblicità sono il modo più semplice e veloce per comunicare eventi, offerte e lanciare nuovi prodotti sul mercato. Se sei un'azienda consolidata preserva anche il tuo brand e/o la tua immagine con una presenza costante sui media. Questo plugin permette di mostrare uno spot pubblicitario Google con *Google admob* prima dell'esecuzione di un comando di menu o della Bottombar. Per abilitare questa possibilità, dopo aver creato un account Google, indirizzare il browser su https://admob.google.com, effettuare il login, aggiungere la propria app e creare un unità pubblicitaria di tipo **Interstitial**.

Inserire la chiave ottenuta in *Interstitial Key* del plugin *Google ads* tramite il Device configurator. Per abilitare lo spot seleziona la *property* **Show advertising** della voce di menu o dell'item della bottombar interessata.

Quando l'utente selezionerà una voce di menu o premerà il bottone interessato, prima verrà mostrato lo spot di Google e dopo procederà mostrando il plugin selezionato con la relativa *presentation* se definita.



Immagine della posizione di Google ads Interstitial all'interno dell'app

6.13 Blog & help

La necessità di mostrare un elenco di contenuti come documenti, pagine htlm o url ha portato alla creazione di questi 2 plugin. Il plugin help contiene due diverse versioni **Help.presentations.list-plugin** e **Help.documents.list-plugin**. Il plugin Blog invece ha una sola istanza di tipo **Blog.documents.list-plugin**.

Il plugin *Help.presentations.list-plugin* mostra una pagina con un elenco, definito dalla sviluppatore, di contenuti a cui far accedere l'utente. Questi contenuti vengono visualizzati a seconda del profilo utente. Sotto l'immagine dell'editor del plugin.

Blog Items editor [Android-Phone] -					Х	
🕀 🛹 🧕 💾 🤣		1				
Blog list :		Editor				
joomla			[
wordpress		Title :	joomla			~
magento2 prestashop		Text :	help_blog_features			~
		Group name :				~
		Group description :				~
		lmage :	joomla			~
	0	Profile :	All			~
		Document :				~
		Html :	joomla			~
		Url :				~
\sim \checkmark						

Immagine dell'editor del plugin Help

L'editor visualizzato nella figura sopra, mostra come creare la lista di contenuti. E' importante notare che che documenti, file .html o url sono alternativi.

Il plugin **Help.presentations.list-plugin** mostra la lista di tutti i moduli *Presentations* definiti all'interno dell'app. Questo può risultare utile quando l'utente decida di non visualizzare più le informazioni fornite dal plugin *Presentation* ogni volta che l'utente seleziona una data voce di menu o della bottom bar e si sia dimenticato di alcune sue funzionalità o decida di accedere all'help della funzione definito direttamente nel modulo Presentation. In alternativa si può utilizzare il plugin precedentemente trattato.

Il plugin **Blog** si configura come il plugin **Help** ed ha la stessa funzione del plugin **Help.documents.list-plugin** consentendo però un'ulteriore istanza con contenuti diversi.

Descrizione delle *properties* :

- **Title**: indica il titolo del contenuto all'interno della lista. Assume lo stile List title.
- **Text**: indica la descrizione del contenuto all'interno della lista. Assume lo stile List value.
- **Group name** : indica il nome dell'eventuale gruppo a cui l'item deve appartenere. Definendo questa proprietà si può ottenere una lista di contenuti gerarchizzata per argomento.
- **Group description** : indica la descrizione dell'eventuale gruppo a cui l'item deve appartenere.
- **Image**: immagine da associare all'item nella lista. Questa deve essere definita tra le *Fixed-images*.
- **Profile**: profilo utente a cui deve essere associato l'item. Solo gli utenti con questo profilo saranno abilitati ad accedere al contenuto.
- **Document**: documento, definito nel plugin documents, da mostrare nella lista.
- **Html**: file .html, definito nel plugin Html, da mostrare nella lista.
- **Url**:url, definito nel plugin Url, da mostrare nella lista.

Per le funzionalità dei bottoni dell'editor si rimanda agli editor del <u>Menu</u>, della <u>Menu Page</u> o della <u>Bottombar</u>.



Immagine della lista di contenuti all'intero dell'app

7) Database

Il plugin "*Database*" permette di definire tabelle e query di database e la loro struttura record, mediante la definizione di specifici campi con opportune proprietà. È possibile, inoltre, creare in automatico gli statement SQL da eseguire sul server, in genere MySQL, per la creazione delle tabelle stesse. Premendo il bottone '**Tables manager**', si apre l'editor per la definizione delle tabelle e dei loro campi. Premendo il bottone '**Query manager**', si apre l'editor per la definizione delle query.

7.1 Table and queries



Sezione che permette di configurare il plugin 'Database'.

Il plugin database ha due editor uno per gestire le tabelle '**Table manager**' e l'altro per gestire le query '**Queries manager**'. Il *Tables manager* è un editor che consente di definire tabelle, campi, notifiche, azioni php, filtraggi e master/detail. Le tabelle definite possono essere create direttamente sul database remoto Mysql o localmente usando Sqlite. La struttura delle tabelle sul database remoto può essere può essere direttamente importata o aggiornata al cambiamento della definizione della struttura.

Il *Queries manager* ha le stesse funzionalità relative alla definizione delle query. Queste sono create a partire da uno statement sql gestito da un editor con syntax highlight.



Editor del plugin per la gestione del database.

Funzioni per la gestione delle tabelle:

	New table	Prepara la finestra per l'inserimento dei dati per la creazione di una nuova tabella.				
4	Save table	Salva le impostazioni della tabella correntemente creata o modificata.				
G	Delete table	Elimina la definizione della tabella corrente.				

Funzioni per la gestione dei campi:

Ð	New field	Aggiunge un nuovo campo dopo avere selezionato il tipo.
8	Save field	Salva le impostazioni inserite per il campo corrente.
×	Delete field	Elimina il campo correntemente selezionato.
	Add strings	Aggiunge, direttamente dal Tables Manager, le custom strings al dizionario dell'applicativo.
Ø	Add Urls to dictionary	Definisce nuove url multi lingua da aggiungere al dizionario.

Funzioni per la gestione del database:

2	Users' profiles	Esporta sul database MySQL i profili utenti definiti per ciascuna tabella.
Sal	Make sql statement for table	Visualizza lo statement Sql relativo alla creazione della tabella selezionata. Copiando sul server questo statement si crea la tabella desiderata all'interno di un database.
Mysqu.	Create table on MySql db	Crea o aggiorna la tabella selezionata direttamente sul database del CMS. Se la tabella esiste già e contiene dei dati, questi vengono cancellati.
I	Save to disk	Salva tutte le impostazioni sul disco.

Dopo aver premuto il bottone 'New table', nel form, appariranno vari gruppi di comandi (combobox, label e checkbox) distinti per affinità in tre tabsheet: *Table options, Groups* e *SQL*.

7.2 Table options

Nel tabsheet *Table options*, i comandi sono divisi in quattro gruppi di parametri: *Database action's profiles*, *Table's attributes*, *Insert/Update attributes* e *List mode attributes*.

Action's profiles: in questo gruppo di parametri sono contenuti i profili utente abilitati alle relative funzioni sul database.

- Insert: indica il profilo utente abilitato all'inserimento dei dati,
- Update: indica il profilo utente abilitato alla modifica dei dati,
- **View**: indica il profilo utente abilitato alla visualizzazione dei dati.
- **Delete**: indica il profilo utente abilitato alla cancellazione dei dati,
- Filter: indica il profilo utente abilitato al filtraggio dei dati,
- **Lookup**: indica il profilo utente abilitato alla navigazione sui campi di lookup,
- View in DB Manager: indica il profilo utente abilitato a visualizzare la tabella nel plugin *Tables-management-plugin*.

Action's profiles	List styles
All V	Update ∕ Registered ✓
View	Delete
All N	✓ Super Users ✓
Filter	Lookup
Super Users 🔷 💊	All 🗸
View in DB Mana	ger
Super Users	~

Immagine che mostra la sezione in cui definire i profili utente di una tabella.

Tutti i profili in elenco sono quelli dei CMS supportati. Per maggiori dettagli vedere le sezioni <u>3.3</u> e <u>3.4</u>.

Styles: come visto nel capitolo <u>5.11</u> è possibile definire degli stili per le varie parti che compongono il form della lista dei record di una tabella.

Usando gli stili di default non sarebbe possibile creare form con diversi aspetti grafici. Il Device Configurator a questo proposito permette di associare gli stili customs alle varie parti che compongono la lista dei record di una tabella, cambiandone le impostazione grafiche.

Action's profiles	List styles	
List title		
		~
List value		
		~
List detail		
		~
Form filter		
		~

Immagine che mostra i parametri per personalizzare la lista dei campi.

Selezionando da ciascuna combobox uno stile custom creato opportunamente è possibile ottenere una tabella con un'interfaccia diversa da quella ottenuta applicando gli stili di default.

Table's attributes: in questo gruppo di parametri sono presenti le opzioni relative agli attributi delle tabelle.

- **SQL Table name**: indica il nome SQL della tabella nel database. La tabella viene creata sul database applicando il suffisso previsto dal CMS.
- **Tabel name**: indica nome della tabella visibile all'utente, tradotto nelle varie lingue, mediante l'utilizzo delle custom strings.
- **Description**: definisce una stringa utente con la descrizione del contenuto della tabella, tradotta nelle varie lingue, mediante l'utilizzo delle custom strings.

Lacal tables: questa sezione definisce i parametri per la gestione delle tabelle locali.

• **Use as local**: indica se la tabella sia locale al device o remota sul sito web di riferimento;

- **Empty table before download**: indica se la tabella debba essere svuotata con uno statement SQL *truncate* prima di essere popolata di record con l'operazione di download records.
- **Download by userid**: indica se il download dei record debba essere effettuato solo sui record relativi all'utente correntemente loggato. Questa funzione necessita che sulle tabelle remota e locale sia presente un campo di tipo *TUserIdField*.

Il plugin *Tables-management-plugin* consente agli utenti abilitati di effettuare il download dei record dal database remoto al database locale e, viceversa, l'upload dei record dal database locale al database remoto. Questa operazione è possibile solamente se le tabelle remote e locali hanno lo stesso nome e struttura.

Insert/Update attributes: in questo gruppo di parametri, vengono collocate le impostazioni relative alla modifica della tabella.

- **Grouped**: consente di raggruppare graficamente i record secondo i gruppi definiti nel tabsheet *Groups* con le opzioni grafiche definite nella sezione *Groups* della pagina *Data;*
- **Close page after inserting/editing record**: impostando questa opzione la pagina viene chiusa dopo l'inserimento o la modifica di un record;
- **Reload data after Insert/Edit**: indica se devono essere caricati automaticamente i dati della tabella dopo un operazione di inserimento o modifica dei record essere mostrato un messaggio all'utente che tenta di salvare un record senza avere inserito campi obbligatori;
- Show labels: mostra le *labels* della tabella;
- **Hide empty fields on view:** consente di nascondere i campi vuoti nello stato di visualizzazione. Durante l'inserimento e la modifica questi campi sono presenti.

List mode attributes: in questo gruppo di parametri, sono presenti le impostazioni che regolano la visualizzazione dei record nelle liste (plugin <u>Database.Table.viewrecords-plugin</u>).



Immagine che mostra la struttura della lista dei record di una tabella.

- **Table has CMS prefix:** indica se la tabella sul database remoto ha un prefisso nel nome dato dal prefisso utilizzato dal CMS al momento dell'istallazione; ad es. wp_users o sgh_users ...
- **Mark record**: indica se il record corrente deve essere taggato, ossia mostrato all'apertura della tabella;
- **Show first item as 'list title'**: visualizza il primo campo del record con lo stile List title;
- **Show labels**: mostra le labels nelle liste dei record della tabella;
- All records searchable for admin: l'amministratore può visualizzare tutti i record inseriti da qualsiasi utente gestendo i relativi filtri;
- Has favourites records: impostando questo parametro si abilita la gestione dei favoriti. Ciascuna tabella locale o remota può avere localmente una tabella in cui vengono inseriti i record preferiti, come ad es. gli utenti del plugin relativo all'Instant messaging. Se lo sviluppatore dovesse creare una tabella prodotti, eventi o altro potrebbe gestire i record preferiti dall'utente con questa semplice gestione.
- Auto filter: questo parametro indica come e se filtrare automaticamente una tabella quando questa visualizza i propri dati. Questo parametro ha 3 valori : *None, Auto filter by userid* e *Auto filter by company.*

None: indica che non deve essere applicato nessun filtro;

Auto filter by userid: indica che deve essere applicato un filtro sull'utente, pertanto saranno visualizzati solo i record inseriti da quest'ultimo;

Auto filter by company: indica che deve essere applicato un filtro sull'azienda, pertanto saranno visualizzati solo i record inseriti da quell'impresa.

• **Click action:** definisce il comportamento che deve avere il form del plugin <u>Database.Table.viewrecords-plugin</u> quando un utente seleziona un determinato record. I parametri sono 3 : *View, Change* e *Document*.

View: indica che il record selezionato deve essere mostrato in visualizzazione.

Change: indica che il record selezionato deve essere mostrato in modifica.

Open document: indica che per il record selezionato deve essere mostrato il documento allegato al record stesso definito nella *property Document field*.

- **Document field:** indica il campo da utilizzare come document nella *property* Click action.
- **Direction:** indica la disposizione delle *labels* all'interno del record. Questo parametro ha 2 valori: Horizontal e Vertical.

Horizontal: dispone label e valore su un'unica riga;

Vertical: dispone label e valore su righe separate;







Record dove non sono visualizzate le labels.

• Alignment: indica l'allineamento dei valori e delle label all'interno del record. Questo parametro ha 3 valori: *Left, Right* e *Center*.

Left: allinea i dati a sinistra; *Right:* allinea i dati a destra;

Center: centra i dati;



Suggeriamo di fare alcune prove e vedere quale impostazione risulti preferibile.

7.3 Groups

La sezione *Groups* definisce i gruppi logici su cui devono essere suddivisi i campi di un record. Se non si definiscono gruppi, i dati vengono mostrati in modo piatto, senza nessun raggruppamento.

Table options	Groups	SQL	Field options	
Values :				
			-	+
Photo				
badge data				
GPS				

Immagine che mostra la sezione Groups.

Per definire un gruppo selezionare dalla combobox *Values*, contenente le Custom strings, un valore e premere il bottone '+'. Questo sarà visualizzato nella lista sottostante. Premendo il bottone '-' è possibile cancellare la voce selezionata. Selezionando un item della lista e premendo i bottoni in fondo ad essa, sarà possibile cambiare il loro ordine all'interno della lista stessa. Di seguito sono mostrate alcune immagini che esemplificano il contenuto di alcune tabelle per cui sono state definiti dei gruppi.

	-
oducts	10 🖸 🗶
the second	
A Construction of the second se	
a the second	
Code *	()
Boat	
Description *	
Low price boat	
Low price boat	
Low price boat Currency * Dollar	
Low price boat Currency * Dollar Price *	
Low price boat Currency * Dollar Price * 15.000	
Low price boat Currency * Dollar Price * 15.000	

Record definito su 2 gruppi, uno per l'immagine ed uno per i dati.

Solo Ch. SOS • • • •	ta		9.B1009	/ 11:05
Locate vares	sino			
Country				
Italia				
Phone				
Fax				
Mobile				
EMail	-			
Facebook	. Incellered.		el apatara	4
16 🕅	Ð	0	2	Q
	0	⊲	Ŧ	

Record definito su più gruppi.

7.4 Table actions

Php table actions

Le azioni database non si limitano solamente alle classiche Insert, Update o Delete di un record. Chi sa ben gestire i database conosce l'utilizzo dei trigger. In alcuni casi gli sviluppatori non hanno sufficiente dimestichezza con questi costrutti, invece hanno familiarità con il PHP. Per questo motivo è stato introdotto il meccanismo delle **Table actions,** script PHP da eseguire, "come dei trigger", prima o dopo una azione di database.

🦺 Table event actio	n	×	<
Name			
Insert action name			
Event type		~]
HTML action type	● Get	O Post	
PHP file			
Insert php url file			
PHP params			
Insert php params			
	Close	Cancel	

Immagine del Table actions editor.

Le properties per definire una action sono :

- **Name**: nome della action che ci permette di elencarla in una lista;
- **Event type**: tipo di evento che decide quando deve essere eseguita la action. Gli eventi sono :

Before insert, After insert indicano rispettivamente che la action deve essere eseguita prima o dopo l'inserimento di un nuovo record;

Before update, After update indicano rispettivamente che la action deve essere eseguita prima o dopo la modifica di un record;

Before delete, After delete indicano rispettivamente che la action deve essere eseguita prima o dopo la cancellazione di un record.

- **HTML action type**: il tipo di trasferimento dati utilizzato dal file PHP, GET o POST;
- **PHP file**: il path del file PHP che deve essere eseguito;
- **PHP params**: i parametri della url che deve essere eseguita.

La definizione della url php sarà costituita ad es. :

PHP file: (DOMAIN)/device/BlockUser.php;

PHP params: (ACCOUNT.EMAIL),id=(ID),op=(BLOCKED).

In maiuscolo tra parentesi tonde vengono inserite le Tag che si riferiscono ai nomi dei campi della tabella. Al momento dell'esecuzione della action le Tag saranno sostituite con i valori reali dei campi. Device Configurator ha alcune tag di sistema : (ACCOUNT.EMAIL): email dell'utente correntemente loggato; (ACCOUNT.USERNAME): username dell'utente correntemente loggato; (ACCOUNT.NAME): nome dell'utente correntemente loggato; (USER.ID): id dell'utente correntemente loggato; (USER.PROFILE): profilo dell'utente correntemente loggato.

(VERSION): utilizzato nella url indica la versione del plugin di connessione del dispositivo utilizzato.

Email notifications

Ad ogni azione database come inserimento, modifica, cancellazione o visualizzazione di un record è possibile inviare una email di notifica. Per far ciò definire un Smtp tramite l'apposito editor (vedere <u>capitolo 6.1</u>).

Emails editor			×
Name		SMTP	~
Receiver 🗌 Logged	user 🗌 App admir	n 🗌 Record us	erid
Record state 🗌 Insert	Update	View	Delete
Subject type	Field name	Text	
By field \sim		~	~
Message type	Field name	Text	
By field \sim		~	~
Attach 1 :	Attach 2 :	Attach 3 :	
~		~	~
MTP editor		Cancel	Close

Immagine dell'editor per l'invio di email di notifica

La *property* **Receiver** indica chi deve ricevre l'email:

- Logged user : utente correntemente loggato;
- App Admin : utente amministratore dell'app;

• **Record userid** : l'utente indicato dallo userid nel record correntemente selezionato.

La *property* **Record state** indica quando inviare l'email di notifica se durante l'inserimento, la modifica, la cancellazione o la visualizzazione di un record.

Subject type definisce l'oggetto della email. Se impostato come **by field** l'oggetto è indicato dal contenuto della property **Field name** altrimenti dalla stringa indicata dalla property **Text**.

La *property* **Field name** contiene tutti i campi della tabella mentre la *property* Text il codice di una stringa utente.

Con lo stesso criterio si definisce il messaggio della email utilizzando la property **Message Type.**

Le properties **Attach 1**, **Attach 2** e **Attach 3** indicano 3 campi da cui prelevare eventuali immagini da inviare come allegati alla email.

7.5 SQL filters & join

In questa sezione vengono definiti una serie di costrutti che permettono allo sviluppatore di creare filtraggi su tabella di database o il master/detail.

SQL prefilters

I prefilter sono filtri, ovvero predicati SQL (WHERE) che vengono applicati al momento del caricamento dei dati di una tabella. Nelle sezioni relative alla *bottombar*, *menu* e *menu page* abbiamo visto che per la creazione di un item con il plugin **Database.Table.viewrecords-plugin** è possbile definire anche un PreFilter. Questa proprietà consente di non caricare tutti i dati della tabella nell'ordinamento naturale dovuto al suo indice primario ma permette di caricare solo i record rispondenti ad una query ordinati secondo una specifica condizione. Per definire un prefilter selezionare un codice dalla combobox **Prefilter codes** necessario ad identificare il filtro all'interno di una lista. I codici presenti in questa combobox sono caricati dalle *Custom strings*. Dopo premere il bottone '+', sarà così mostrato l'editor relativo.

■ SQL code editor	_		×
Select :			
id,code,description,price,currency			
Where : (do not insert 'WHERE' claus	ie)		
code = 'boat'			
Order by : (do not insert 'ORDER BY'	clause)		
code asc			
		-	
	Close	Can	ncel

Immagine del PreFilter editor

Nella sezione **Where** inserire la parte dello statement SQL relativo al predicato in questione. Questa sezione identifica il filtro. Nella sezione **Order by** inserire la condizione SQL per l'ordinamento dei dati. La parte dello statement relativa ai campi della **Select** è automatica. I campi sono quelli della tabella che hanno impostato l'attributo **Visible in list**.

Per una più completa trattazione di questa parte si rimanda lo sviluppatore ad un manuale SQL.

Form SQL filters

Spesso lo sviluppatore si trova a dover effettuare filtraggi sui dati di una tabella non solo al suo caricamento ma anche durante la sua gestione, come detto nel <u>paragrafo 5.6</u>. A questo proposito è stato introdotto il costrutto **Form SQL filters** che consente all'utente con un semplice click di filtrare i dati secondo filtri impostati dallo sviluppatore. La definizione di questi filtri è analoga alla precedente a cui rimandiamo.

Join

Il master/detail è uno dei costrutti relazionali più utilizzati nei database.

Per definire una relazione master/detail, ossia 1 a N, il Device Configurator necessita di 2 fasi.

Nella prima, tramite le *properties* del **Master/Detail relation definitions**, si definisce la relazione che lega 2 tabelle, usando l'editor visibile nell'immagine sotto. Per prima cosa si seleziona la tabella di detail rispetto a quella correntemente selezionata nel **Table manager**. Poi si seleziona un campo dalla lista di sinistra, tabella master, un campo dalla lista di destra, tabella detail, e si preme il bottone '+'. La relazione tra i 2 campi viene aggiunta nella lista in fondo al form. Se la relazione tra le 2 tabelle è costituita da più campi, ripetere l'operazione. Al temine premere il bottone 'Close'.

Per modificare una relazione esistente fare doppio click sulla relativa riga della lista proposta, mentre per cancellarne una, premere '-' dopo avere selezionato la relazione da eliminare.

🦫 Master/detail definition			;	×
Master table :			Detail table :	
products			· · · · · · · · · · · · · · · · · · ·	/
id photo code description currency price Market address	~	+		
			Close Cancel	

Immagine del master/detail editor

Definita la relazione tra le 2 tabelle è necessario definire una action che l'utente deve eseguire durante un'operazione di data entry.

Premere '+' dalla sezione Master/detail action e sarà mostrato il relativo editor.

Un'immagine del Join editor è visibile più sotto.

Le *properties* dell'editor sono descritte di seguito.

- **Name**: nome della action;
- **Master table state**: stato della tabella master in cui il bottone relativo alla action deve essere mostrato. Gli stati sono 3 : *on Insert* durante l'inserimento di un record nella tabella master; *on Edit* durante la modifica del record nella tabella master e *on View* durante la visualizzazione dello stesso record.
- Join relation: relazione da applicare al join tra quelle definite precedentemente;
- Action type on db: tipo di action o stato del form di detail;
- **Button description**: label del bottone da inserire nel form master. Questo codice è definito tra le *Custom strings*.
- **Image code**: codice immagine da visualizzare sul bottone. Queste immagini sono definite in *Images/bottombar*.

🦺 Join action		×
Name		
Insert action nar	ne	
Master table sta	ite	
 On insert 	🔘 On update	🔘 On view
Join relation		
		~
Action type on	db	
Insert	 Update 	🔘 View
Button descript	ion	
		~
lmage code		
		~
Action position		
Show button	s on bottom	
○ Show select	area on top	
O Show record	s on form	
	Close	Cancel

Immagine del master/detail action editor.

• Action position: posizione dei bottoni relativi alle funzioni master/detail. Le possibili poszioni sono 3 :

Show buttons on bottom: i bottoni vengono mostrati in fondo al form dopo tutti campi;

Show select area on top: i bottoni sono mostrati nell'area relativa al master/detail sotto la toolbar del form. Qui si applicano font e colori definiti in <u>Form details area</u>.

Show record on form: i campi del record in questione sono mostrati direttamente nel form master.

7.6 SQL

La pagina SQL gestisce la creazione delle tabelle del database.

E' possibile visualizzare lo statement di creazione della tabella ed eseguirlo manualmente sul server, oppure creare direttamente la tabella.

Funzioni per la gestione del database

	Make sql statement for table	Visualizza lo statement Sql relativo alla creazione della tabella selezionata. Copiando sul server questo statement si crea la tabella desiderata all'interno di un database.
⊜ SQL	Import table structure	Importa la struttura di una tabella presente sul database remoto MySQL.
N ISSQL	Create table on MySql db	Crea o aggiorna la struttura della tabella selezionata direttamente sul database del CMS. Se la tabella esiste già e contiene dei dati, questi saranno cancellati.
SQL	Create preferites or local tables	Crea o aggiorna la struttura della tabella selezionata direttamente sul database locale del device. Se la tabella esiste già e contiene dei dati, questi vengono cancellati.

7.7 Field options

Una volta creata una tabella e definite le sue proprietà si possono aggiungere i suoi campi. Questi, definiti all'interno del *Device Configurator*, sono campi logici, mentre quelli fisici sono presenti sul database.

۵ 🔊	<u>.</u>		•
SOL		TStringField ·	
JQL		TIntegerField	
	T (1)	TFloatField	
files	Table's att	TListField	
	SQL Table na	TLookupListField	
•	badges	TCheckField	
	Table name	TDateField	
•	badges	TTimeField ·	•

Creare il campo di una tabella.

Per creare un campo, selezionare il tipo dalla combobox presente sulla toolbar (figura sopra) e premere il bottone *New field* della toolbar. Immediatamente sarà visualizzato il tabsheet visibile nella figura sottostante.

🐌 Tables manager	X
💐 🛹 🧕 🔂 🛷 🖉	l 🔮 🖉 🚨 📴 🚉 🚉 💾 🗭 TStringField 🗸
Tables : ibc_users_data	Table options Groups Table actions SQL filters & Joins SQL Field options
ibc_companies_data	Crud Lookup Style TStringField Options Insert profile SQL Field name Mandatory Required Readonly Visible Visible Visible in list Filter Search Find in form
o Fields : id fixedtext image	Style Concat field on list view Separator Group name V V
name aboutme status companyid companyname companyrole gpslabel locality postalcode province countryname	Field's options Text prompt Default Simple string Url string Social url string Url prefix ✓
rharaaddrarr Y	

Parametri di un field.

I dati presenti in questa sezione sono divisi in varie aree : *Profiles, Field's attributes, Attributes, Group name, Concat field on list view* ed altri peculiari del tipo di campo stesso.

Profiles

La sezione profiles indica i profili utente abilitati ad eseguire le varie operazioni sui campi della tabella.

I *Crud profiles* indicano i profili utente abilitati alle funzioni di Insert, Update e View nella modalità di inserimento/modifica/visualizzazione dati.

I *Lookup profiles* indicano i profili utente abilitati alle funzioni di Update e View nella modalità di lookup.

Style

Un campo database oltre a seguire gli stili di default trattati nel <u>capitolo 5</u>, può avere stili personalizzati. I <u>Custom styles</u> hanno questa funzione.

Per ogni singolo campo si possono definire 3 diversi stili :

- Label: definisce lo stile della label del campo;
- **On view**: definisce lo stile del controllo nella stato di View del record;
- **On Edit**: definisce lo stile del controllo nella stato di Edit del record;

Ogni tabella può essere gestita direttamente o come lookup all'interno di un campo di tipo TLookupField. Questo non è altro che un reference ad uno specifico record di una data tabella.

Per quanto riguarda la gestione dei profili rimandiamo a quanto detto precedentemente nelle sezioni <u>3.3 User profiles</u> e <u>3.4 Admin profiles</u>. Un parametro importante è il checkbox **Owner**. Se impostato questo parametro, la funzione indicata nel relativo profilo, oltre che agli utenti di quel dato profilo, viene estesa all'utente che ha creato/modificato il record in questione. Per esempio:

Crud	Lookup	Style	
Insert	profile		
All			~
Update	e profile		
None		\sim	Owner
View p	rofile		
All		~	Owner

Immagine dei parametri CRUD e di stile dei campi database.

secondo l'immagine sopra l'inserimento e la visualizzazione del campo sono consentite a tutti gli utenti mentre la modifica a nessun utente. Se fosse stato impostato il checkbox **Owner** per la *property Update profile*, la gestione del campo sarebbe estesa all'utente che ha inserito o modificato il dato.

Field's attributes

In questa sezione vengono definiti, come per la tabella, il nome fisico del campo sul database tramite la *property* **SQL field name**, un suo nome tradotto, visualizzato all'utente, usando le *Custom strings* presenti nel parametro **Field name**, e se richiesta, la dimensione del campo stesso (numero di caratteri per stringa).

TImageField	
SQL Field name	
image	
Caption	
image	~
Size	Default search
0	~
Downl	oad on local table

Immagine relativa ad un field attribute di tipo TImageField.

La property **Default search** indica l'operatore da applicare al campo se definita la property **Search** tra gli attributi **Options**.

La property **Download on local table** indica se il campo deve essere inserito tra quelli di download nel caso in cui si effettui un'operazione di download su una tabella locale.

Options

Le **Options** sono *properties* molto importanti che determinano il comportamento del campo stesso :

- **Mandatory**: indica se l'utente è obbligato ad inserire dei valori;
- **Required**: indica che il campo appartiene a un gruppo di cui almeno uno deve essere valorizzato obbligatoriamente dall'utente. Per esempio, se una tabella contiene i campi A, B, C e D, con A mandatory e sia B sia C required, il record viene inserito solo se per il campo A è stato definito un valore e per i campi B e C almeno uno è stato valorizzato;
- **Readonly**: indica se il campo deve essere modificabile o no. Per esempio un campo derivante dalle informazioni del GPS non deve essere modificato dall'utente;
- **Visible**: indica se il campo deve essere visibile o no (userid in genere non è visibile);
- **Visible in list**: indica se il campo deve essere visualizzato quando viene mostrata la lista dei record;
- **Filter**: indica se il campo deve essere aggiunto ai campi soggetti a filtraggio avanzato (stile query);
- **Search:** indica se il campo deve essere aggiunto alla lista dei campi presenti nel form di ricerca (vedi <u>Database.Table.Searchrecord-plugin</u>);
- **Find in form**: indica se il campo deve essere aggiunto ai campi per cui può essere effettuata una ricerca nel form.



Immagine di un filtraggio record utilizzando la funzione 'Find in form'.

Group Name indica il gruppo logico a cui deve essere associato il campo. I gruppi sono creati nel tabsheet <u>Groups</u>.

Non è obbligatorio definire questa *property* a meno che non sia impostata la *property Grouped* della tabella nella sezione *Insert/Update attributes* della stessa.

Il parametro **Concat field on list view** ha effetto solamente quando vengono visualizzati i record di una tabella (<u>Database.Table.Viewrecords-plugin</u>). Spesso per motivi di rappresentazione, è necessario mostrare 2 campi insieme. Per esempio, in un'anagrafica che contiene 'Cognome' e 'Nome', è importante unire questi due campi come se fossero uno solamente. Per fare ciò, quando si definisce il campo 'Surname', si deve inserire il campo 'Name' nella *property Concat field on list view* e i dati saranno mostrati, come nella figura sotto.



La property **Separator** indica i caratteri che devono essere interposti tra i due campi concatenati da visualizzare nella lista dei record di una tabella.

Group name	Concat field on list view		Separator
personaldata	▼ name	•	
Text prompt			
sumameprompt	•		
Default			
Simple string O Url string (Social url string		
Impostare la prop	perty concat field	on list vie	ew.

Ciascun campo di database ha una parte di opzioni comuni, come quelle sopra, e una parte che sono peculiari della tipologia del campo stesso.

Le tipologie di campi attualmente supportate sono :

Tipo campo	Funzione
<u>TStringField</u>	Gestione stringhe
<u>TMemoField</u>	Gestione campo memo
<u>TSmallIntField</u>	Gestione numeri smallint
<u>TIntegerField</u>	Gestione numeri interi
<u>TFloatField</u>	Gestione numeri con virgola
TListField	Gestione combobox
<u>TLookupListField</u>	Gestione combobox con reference ad un field di un'altra tabella
<u>TLookupEditField</u>	Gestione combobox editabile. Il testo editato indica la sottostringa da ricercare nella tabella.
<u>TCheckField</u>	Gestione check box
<u>TDateField</u>	Gestione data
<u>TTimeField</u>	Gestione ora
<u>TDateTimeField</u>	Gestione data ed ora
<u>TImageField</u>	Gestione immagine o foto
<u>TFixedImage</u>	Gestione immagine fissa allo scopo di

	rendere l'interfaccia più gradevole
TFixedText	Gestione testo statico, per notificare messaggi o spiegazioni all'utente. Questo è simile ad una label.
<u>TSignatureField</u>	Gestione firma utente fatta col dito o pennino
TBarCodeField	Gestione/lettura codice a barre
TEmailField	Gestione email utente
<u>TPhoneField</u>	Gestione numero di telefono
<u>TDocumentField</u>	Gestione documento (pdf, doc,)
TStatusField	Campo che indica uno stato simile al checkbox
TGPSLatitudeField	Gestione latitudine letta da GPS
TGPSLongitudeField	Gestione longitudine letta da GPS
TGPSCountryNameField	Gestione nazione derivata da lettura GPS
TGPSLocalityField	Gestione località derivata da lettura GPS
<u>TGPSPostalCodeField</u>	Gestione codice postale derivata da lettura GPS
TGPSAddressField	Gestione indirizzo derivata da lettura GPS
<u>TGPSSubAdminAreaField</u>	Gestione sub località derivata da lettura GPS
<u>TUseridField</u>	Gestione id utente
<u>TUserProfileField</u>	Gestione profilo utente
<u>TUserEmailField</u>	Gestione email utente
<u>TCompanyIdField</u>	Gestione id azienda
TMasterField, TDetailField	Campi per gestione di tabelle master/detail

Ogni campo può avere delle *property* proprie che ne determinano il comportamento.

7.7.1 TStringField

TStringField: campo che gestisce le stringhe di dati.

Field's options			
Text prompt		Defa	ault
surnameprompt		~	
 Simple string 	🔘 Url string	○ Social url str	tring
Url prefix			
			~

Properties dedicate di un campo di tipo stringa.

- **Text prompt**: indica il prompt della linea di input;
- **Default**: indica il valore di default del controllo;
- **Simple string**: gestisce il controllo come una semplice stringa;
- **Url string**: il controllo viene mostrato come l'indirizzo di un sito internet sensibile al click. Cliccando su questo si apre la relativa pagina web;
- **Social url string**: analoga a Url string, ma specifica per i socials. Potrai utilizzare questo componente solo se disponi del plugin Social nella tua app.
- In caso contrario sarà impossibile per gli utenti aprire i siti indicati.
- Url prefix: indica la url associata al campo nel caso sia impostato l'attributo Url string.

7.7.2 TMemoField

TMemoField: campo che gestisce i memo. Un campo memo è un campo di testo disposto su più linee. Ha solamente properties generiche.

7.7.3 TSmallIntField, TIntegerField

TSmallIntField e TIntegerField: campo che gestiscono numeri interi.

La loro differenza consiste nel range di numeri che possono contenere. TSmallIntField fa riferimento al tipo smallint SQL mentre TIntegerField crea e gestisce un campo di tipo integer.

Field's options				
Text prompt				
vote_prompt		~		
	Min	Max	Default	Currency/measures
Auto Increment	1	10	6	

Immagine che mostra le properties dedicate di un campo di tipo Intero.

- **Text prompt**: indica il prompt della linea di input;
- **Auto Increment**: indica se il campo ha la proprietà di *auto_increment* a livello di database; generalmente un campo con questo parametro attivo non è visibile;
- **Min, Max, Default:** impostano relativamente il minimo, il massimo ed il valore di default del controllo;
- **Currency/measures**: indica se il campo ha un 'a unità di misura che visualizza nella lista di records.

7.7.4 TFloatField

TFloatField: campo che gestisce numeri con virgola. Per le opzioni vedere <u>TIntegerField</u>.

7.7.5 TLookupListField

TLookupListField: campo che gestisce una combobox con funzione di lookup.
	Field Id	
~	id	~
	Copy value in field	
~	companystr	~
	~	Field Id v id Copy value in field v companystr

Properties dedicate di un campo di lookup.

- **Table name**: indica la tabella a cui fare riferimento per il lookup;
- **Field Id**: indica il campo della tabella sopra citata a cui fare riferimento per il lookup, gestendo questo campo di tipo *Integer* come reference;
- **Field description:** definisce il campo della tabella sopra citata da motrare all'utente nella tendina della combobox.
- **Copy value in field**: indica su quale campo della tabella copiare il valore del campo **Field description**. Questa funzione può essere utilizzata quando lo sviluppatore vuole effettuare una ricerca (Find in form), su un campo di lookup. Poiché nella tabella viene salvato solo in valore intero del lookup, la ricerca può essere effettuata su questo campo.

7.7.6 TLookupEditField

TLookupEditField: campo che gestisce una combobox editabile con funzione di lookup.

Questa campo è simile al precedente differendo in un solo aspetto.

Mentre la combobox del campo *TLookupListField* carica tutti i valori dalla tabella di lookup ed è generalmente utilizzato in tabelle dove i record per i valori di lookup sono relativamente pochi. Il campo *TLookupEditField* è utilizzato quando i record che costituiscono i valori di lookup sono molti e il tempo di caricamento dei dati sarebbe eccessivo.

Questo campo presenta una combobox editabile vuota ,ma quando l'utente comincia a scrivervi dentro dopo il terzo carattere viene avviata una ricerca sul database su tutti i valori che contengono i caratteri digitati dall'utente. Questo restringe le ricerca ed il tempo di caricamento dei dati.

7.7.7 TDocumentField

TDocumentField: campo che gestisce contenuti documentali, ad esempio pdf, doc, immagini, video ...

Field's options			
Text prompt		Document type	
NoPrompt	~	Document	~

Immagine che mostra le properties dedicate di un campo di tipo document.

- **Text prompt:** indica il prompt della linea di input;
- **Document type:** se la property <u>Click action</u> ha valore 'Document', potrai qui indicare il tipo di contenuto del campo, tra *Audio*, *Video e Document*, che sarà possibile aprire con un semplice click quando la stato del record è View.

7.7.8 TDateField, TTimeField, TDateTimeField

TDateField, TTimeField, TDateTimeField: campi che gestiscono data, ora o entrambi.

Field's option	5
🗹 Auto field	Format ddd dd mmm yyyy hh:nn:ss

Properties dedicate di un campo di tipo data/ora.

- Auto field: indica se il campo deve impostare data e/o ora automaticamente;
- **Format:** indica il formato che deve assumere il campo quando visualizza i dati.

7.7.9 TImageField

TImageField: campo che gestisce un 'immagine catturata tramite la fotocamera o selezionata dalla gallery del dispositivo.

Field's op	otions				
Width	Height	List default image		Form default image	
			~		\sim
Url prefix					
					\sim
		Inpu	t		
Show la	ibel	Can	nera		\sim

Properties dedicate di un campo grafico.

- **Width**: indica la larghezza desiderata per l'anteprima dell'immagine. Se *Width è vuoto* l'immagine occupa la larghezza utile del form.
- **Height**: indica la larghezza desiderata dell'anteprima dell'immagine. Se *Height è vuoto* l'altezza dell'immagine è uguale alla sua larghezza.
- List default image: indica l'immagine dei default che deve essere mostrata nella lista record di una tabella quando il campo non ha nessun valore. Questa è selezionata tra le Fixed Images.
- **Form default image**: indica l'immagine dei default che deve essere mostrata in un form quando il campo non ha nessuna immagine associata. Questa è selezionata tra le Fixed Images.
- Url prefix: indica la parte iniziale dell'url dell'immagine che deve essere aperta. A questa viene sommato il valore del campo presente nel database. Utilizzare questa property quando si accede a tabelle gestite da siti web che contengono immagini i cui files non vengono inseriti nel path (DOMAIN)/images.
- **ShowLabel**: indica se si vuole mostrare una Label per l'immagine stessa.
- **Input** : indica il metodo di acquisizione dell'immagine tra gallery, fotocamera o entrambe.

7.7.10 TFixedImage

TFixedImage: campo che gestisce un'immagine fissa, selezionabile tra quelle presenti nelle *Fixed images*. Il contenuto di questo campo non è inserito nel database. L'immagine ha solo una funzione puramente estetica.

Width	Height
Fixed image	

Properties dedicate di un campo FixedImage.

- **Width**: indica la larghezza desiderata dell'anteprima dell'immagine. Se *Width è vuoto*, l'immagine occupa la larghezza utile del form.
- **Height**: indica la larghezza desiderata dell'anteprima dell'immagine. Se *Height è vuoto*, l'altezza dell'immagine è uguale alla sua larghezza.
- **Fixed image**: combobox per la scelta dell'immagine per questo controllo. Le immagini tra cui scegliere sono definite nella sezione *Images/Fixed Images* di cui abbiamo parlato nel <u>paragrafo 1.2</u>.

7.7.11 TListField

TListField: campo che gestisce una combobox con items predefiniti.

Value	- +	
Male		
female		

Properties dedicate di un campo lista (combobox).

L'editor in figura mostra come creare gli items della combobox a partire dalle <u>custom strings</u> definite dall'utente. Nel database viene salvato un'intero che indica la posizione del valore all'interno della combobox.

7.7.12 TCheckField

TCheckField: campo che gestisce un checkbox con valori alfanumerici.

Field's options		
Checked by default Hide group	Hide group when checked	Show group for owner
Checked value Unchecked value		
1 0		

Properties dedicate di un campo di tipo checkbox.

- **Checked by default**: indica se lo stato iniziale del controllo deve essere checked;
- **Hide group**: indica se il checkbox deve essere responsabile dello stato di visibilità del gruppo che che lo contiene. In questo caso quando il record sarà nello stato *View* se il checkbox sarà nello stato checked il gruppo sarà visualizzato altrimenti verrà nascosto.
- **Hide group when checked**: indica che lo stato del bottone in cui deve essere nascosto il gruppo in cui è contenuto è quello *unchecked*.
- Show group for owner: forza la visualizzazione del gruppo di controlli in cui è presente il *checkbox* in visualizzazione o modifica quando è loggato l'utente che ha inserito il record.
- **Checked value** : stringa che identifica il valore da salvare nel database quando il checkbox è selezionato.
- **Unchecked value** : stringa che identifica il valore da salvare nel database quando il checkbox NON è selezionato.

7.7.13 TSignatureField

TSignatureField: campo che gestisce la firma utente fatta con un pennino o col dito.





Firma fatta con un pennino in un'app Android.

7.7.14 GPS fields

GPS fields: questa tipologia di campo gestisce dati derivanti dalla lettura del GPS.

I campi in questione sono : *TTGPSLatitudeField*, *TGPSLongitudeField*, *TGPSLocalityField*, *TGPSCountryNameField*, *TGPSPostalCodeField*, *TGPSAddressField*, *TGPSSubAdminAreaField*. **7.7.15 TPhoneField**

TPhoneField: campo che gestisce numeri telefonici, fax e cellulare. Il campo viene mostrato all'utente con lo stile del <u>Link</u> e quando l'utente vi clicca sopra inizia una chiamata telefonica.

7.7.16 TUserIdField

TUserIdField: campo che gestisce l'id utente inserendolo in un campo della tabella database.

L'utilizzo di questo campo è necessario in due circostanze :

- la tabella deve gestire la *property Auto filter* impostata al valore Auto filter by Userid. In questo modo l'utente visualizza solo dati inseriti o modificati da lui stesso;
- La tabella deve gestire i dati in base al profilo utente. Vedere paragrafo 7.1.

7.7.17 TUserProfileField

TUserProfileField: campo che gestisce il profilo utente inserendolo in un campo della tabella database.

7.7.18 TUserEmailField

TUserEmailField: campo che gestisce l'email utente inserendola in un campo della tabella database. Il campo viene mostrato all'utente con lo stile del <u>Link</u> e quando l'utente vi clicca sopra viene avviato il servizio per l'invio di una nuova email.

7.7.19 TCompanyIdField

TCompanyIdField: campo che gestisce l'id di un'azienda e lo inserisce in un campo della tabella database. Questo campo è necessario se la tabella ha la necessità di gestire/impostare la *property Auto filter* al valore **Auto filter by CompanyId** perché un utente possa gestire solamente dati riconducibili alla propria azienda. E' necessario se l'app gestisce dati di più aziende.

7.7.20 TMasterField, TDetailField

I campi **TMasterField** e **TDetailField** consentono la gestione di tabelle master/detail.

Se lo sviluppatore volesse creare tabelle master/detail deve inserire un campo di tipo TMasterField all'interno della tabella master ed un campo TDetailField nella tabella detail. I campi, gestiti internamente con valori univoci da I.B.C. App Maker, vengono utilizzato nella creazione della relazione di master/detail. Lo sviluppatore si dovrà preoccupare di mappare questi campi usando il Join editor.

7.8 Query

Parte essenziale dei database sono proprio le **Query**. I.B.C. App Maker ha affrontato la tematica permettendo allo sviluppatore di creare query parametriche in modo semplice e veloce. Le conoscenze di SQL necessarie sono basilari. E' possibile, anche, creare statement molto complessi, tutto dipende dalle esigenze e capacità dello sviluppatore stesso.

Nell'immagine seguente si può vedere il *Query Manager*. Sostanzialmente questo è molto simile al *Table Manager* e presenta le stesse funzionalità ma in modo ridotto. La sezione *Profiles*, ovviamente ridotta, presenta solo 3 profili poiché con le query on si inseriscono, modificano o cancellano record.

Per le altre sezioni presenti rimandiamo al *Table Manager*.

7.8.1 Query options

 Query manager Query manager	1 🔮 🖋 🚉 🖻 🕏	- C ×
Queries : users users_data	Query options Groups Form filters Profiles View Filter	A Z SQL's attributes SQL statement name
	All View in DB Manager	users_data Label userdata V
	Styles	Description userdata ~
Fields : id image	List value	List mode attributes ✓ Show first item as 'list title' ✓ Show Labels All records searchable for admin
name aboutme companyname companyrole	List detail V	Use as local Auto filter None
locality postalcode province	✓ Detail attributes	Document field Grouping by
companyid countryname shareaddress phone	 Grouped Show Labels Hide empty fields on view 	Direction Alignment Horizontal V
mobile email		

Immagine del query manager

La differenza sostanziale tra il *Table Manager* ed il *Query Manager* è individuata dal metodo di creazione dell'oggetto database.

Per creare una query dopo aver definito le sue opzioni nel *Query options* premere il bottone **A** visibile nella figura sopra. Verrà visualizzato il SQL editor, visibile sotto. Questo è un editor di testo, dotato di syntax highlight che prevede alcune tag per parametrizzare lo statement SQL.

7.8.2 SQL editor

Le tag previste sono le seguenti:

%filter% o **Where %filter%**: viene sostituita dalle condizioni WHERE create da I.B.C. App Maker, riguardanti le funzionalità *filter, search, Find in form* del singolo campo della tabella o *Form filters* della query stessa;

%and filter%: ha la stessa funzionalità della tag *%filter%* con la differenza che viene utilizzata per congiungere due o più predicati. Ad es. WHERE (name like '%J%')%and filter%;

%or filter%: ha la stessa funzionalità della tag *%filter%* con la differenza che viene utilizzata per disgiungere due o più predicati. Ad es. WHERE (name like '%John%')%or filter%;

%limit%: viene sostituito con i valori corretti di limit previsti da I.B.C. App Maker. Di default questo valore è 25;

%offset%: viene sostituito con i valori corretti di offset previsti da I.B.C. App Maker, per lo scorrimento dei record.



Immagine dell'SQL editor

Dopo la pressione del bottone '*Close*' viene mostrato un field mapper che consente di definire per ogni campo della query, in base al tipo SQL, il tipo di campo App Maker da associare. Questa finestra si chiama *DataSet structure manager* perché permette di definire la struttura della query.

Cliccando su *field type* si apre una tendina che permette la selezione del tipo di campo supportato per il campo SQL in esame. Al termine dell'operazione premere il bottone '*Close*'. Tutti i campi della query saranno importati e visualizzati nella lista posta a sinistra della finestra. Lo sviluppatore deve completare la configurazione aggiungendo per ogni singolo campo la *Label* i *Field options, prifili, stili*...



Come per le tabelle è possibile aggiungere dei singoli campi di tipo *TFixedImage* o *TFixedText*. L'aggiunta manuale di qualsiasi altro tipo di campo è proibita. I campi di tipo Fixed sono previsti per rendere gradevole o maggiormente esplicativo il form di dettaglio della query stessa. Quando l'utente vuole visualizzare una query, prima viene visualizzata una lista di campi che hanno l'attributo '*Visible in list*' abilitato, come per la tabella, poi premendo sul record viene visualizzato un form con il dettaglio del record che prevede tutti i campi della query. Anche per la query sono

previste le funzioni di <u>Click Action</u> e *'List attributes'* presenti nel *Table Manager*.

8) Creazione dell'app

8.1 Registrazione dell'app

Terminata la configurazione della tua app, per procedere al suo acquisto, è necessario registrarla sul sito <u>https://www.ibcappmaker.com</u>. Per effettuare la registrazione, crea un account sul sito, se non lo possiedi già, ed effettua il login. Quindi apri la pagina *Projects/Android projects* per registrare un'app Android o la pagina *Projects/iOS projects* per registrare un'app iOS. Riempi tutti i campi con i dati richiesti e inserisci come *Deploy file* il file .deploy, creato dal Device configurator tramite la relativa funzione **Deploy,** e come campo **System images** il file generato dalle funzioni *Images => Android system images and icons* o *Images => iOS system images and icons*.

8.2 Acquisto dei plugins

Dopo la registrazione dell'app, è necessario acquistarla. Vai sulla pagina *Store* del sito e clicca sulla piattaforma desiderata.



Per decidere cosa acquistare si può far riferimento alla funzione Check plugins del Device configurator. Questa funzione indica quali plugins acquistare e il numero esatto di licenze per ottenere l'app correntemente configurata. A questo punto, cerca questi plugins sullo store, aggiungili al carrello e procedi all'acquisto. Durante la fase dell'acquisto, è necessario indicare nel campo *Store code* il codice ottenuto durante la registrazione dell'app, nell'omonimo campo. Paga e dopo un massimo di 48 ore, riceverai una email con la notifica del termine della creazione. Scarica il binario ottenuto e caricalo sul tuo Store Android o iOS.

Se non vuoi gestire la tua app su Google Play Store o Apple Store, puoi acquistare il servizio di pubblicazione e il tuo store sarà interamente gestito dai nostri tecnici.

Appendice A.

Come creare un proprio progetto template

Per creare un proprio template per prima cosa creare un nuovo progetto completamente funzionante o solo in parte. Al termine del progetto che si intende comune a tutti i futuri progetti, copiare la sua cartella nella sotto directory *Device Projects\Defaults*. Alla successiva creazione di un nuovo progetto tra i valori di default verrà proposto anche il progetto sviluppato dall'utente.