

Versione 22.5 - Note di rilascio

Indice generale

Novità e miglioramenti della versione 22.5	3
Ottimizzazione esperienza di debug	3
Sezione console	3
Sezione lista operazioni	4
Generazione del codice	5
Indicatore modalità online/offline	6
Database Profiler	6
IDE	7
Aggiunta della voce di menu contestuale “Copia in altro progetto”	7
Ricerca nella documentazione più veloce	7
Indicazione data source degli elementi nell’editor delle videate	7
Gestione di file tramite evento app.onCommand	8
Documentazione delle proprietà di un elemento visuale	9
Cambiare tipo e dimensione del font nell’editor	9
Applicazione	9
Funzione inPreview per verificare se un’app è avviata da IDE	9
Possibilità di configurare la pagina di avvio	10
Aggiornato il componente grafico Chart.js	11
Ionic	11
Proprietà chipList dell’elemento IonSelect	11
Miglioramenti elemento IonAutoComplete	11
Proprietà readOnly in IonAutoComplete e IonSelect	11
Proprietà multiple: IonAutoComplete permette la selezione multipla	11
Proprietà disabled per una singola riga della combo-box	12
Proprietà html per una singola riga della combo-box	12
Icona mostrata anche nell’input di IonAutoComplete (comportamento modificato)	13
L’elemento IonDateTime può mostrare il nome del giorno	14
Scrollbar verticale (comportamento modificato)	14
Database	15
Metodo setConnectionOptions	15
Specificare le opzioni di connessione al server per il Cloud Connector	16
DO	16
Riassegnamento di collection figlie (comportamento modificato)	16
SYNC	17
Evento onDocUpdate notificato in più casi (comportamento modificato)	17
Correzioni della versione 22.5	18
Applicazione	18
Input date con mask: orario con “00” minuti	18

Lingua di localizzazione GoogleCharts	18
Bootstrap	18
Correzione messaggio di errore per BSInput di tipo checkbox	18
Database	18
Default Timezone “Europe/Rome” per MySQL e Oracle	18
Datamap	18
Migliorata la gestione delle date negli IonInput	19
Filtrare campi unbound delle datamap	19
Document Orientation	19
Messaggio di warning reso più esplicito	19
File system	19
Query string con caratteri codificati	19
IDE	19
Importare campi foreign key nel datamodel	19
Caso particolare di utilizzo di funzioni arrow	20
Classe CSS invalida	20
Applicazione copiata tra progetti	20
Sistema di localizzazione	20
Spazio nelle stringhe template	20
Gestione degli spazi nelle righe di codice	20
Evento definito in un nuovo plugin	20
Ionic	20
Allineamento elemento freccia IonAutoComplete e IonSelect	21
Input number con mask per numeri oltre il milione	21
Pulsante clear negli input date	21
Animazione toast su dispositivi iOS	22
SYNC	22
Metodo compareData	22
Sincronizzazione con database multi-tenant	22
Sincronizzazione differenziale in loop	22
Disconnessione durante la sincronizzazione differenziale	22
Metodo minimizeVariations	22
Launcher	22
Apertura file da browser su Android	22
Ritardo nella connessione sync su Android	23
Server	23
Errore d’installazione di un pacchetto personalizzato	23
WebAPI	23
Metodo loadByKey su classe importata da WebAPI Foundation	23
Note di rilascio di versioni precedenti	23

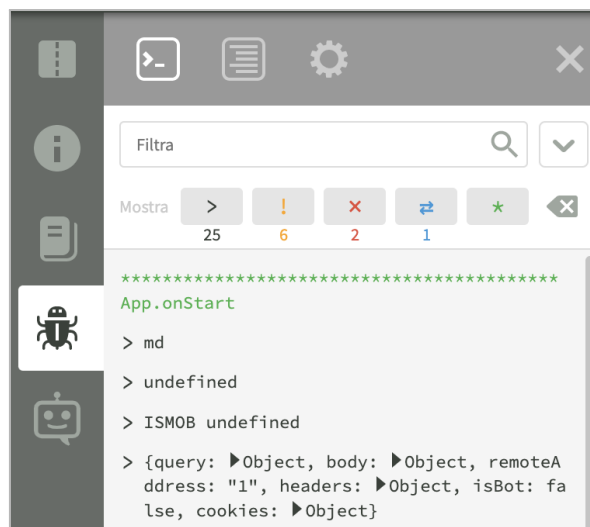
Versione 22.5 - Note di rilascio

Novità e miglioramenti della versione 22.5

Ottimizzazione esperienza di debug

Sezione console

- 1) Nella sezione della console ora viene mostrato il numero totale di messaggi di log, warning, eccezioni e query eseguite. Può essere utile per comprendere il carico complessivo sul database di una determinata sessione di lavoro.



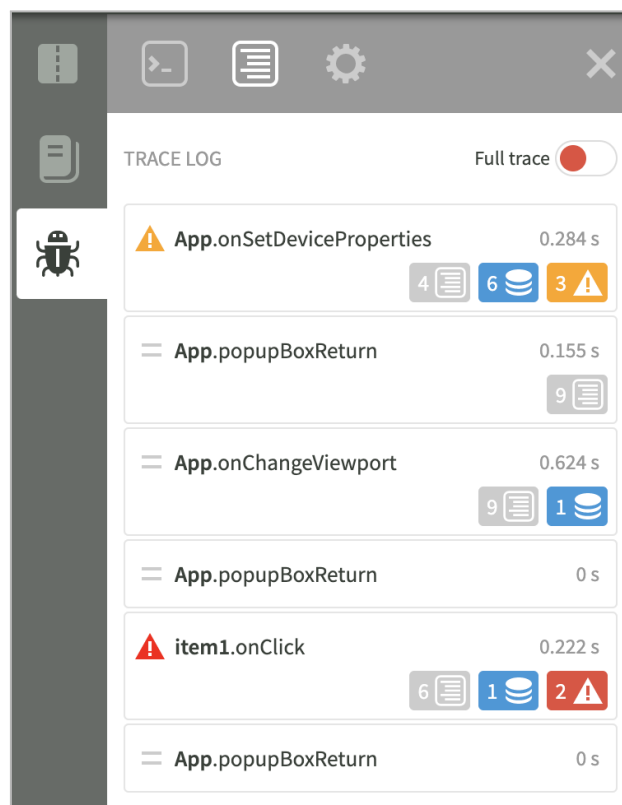
- 2) Nella testata è stato aggiunto il pulsante "Vai in fondo" per visualizzare subito il fondo della lista dei messaggi.
- 3) I messaggi della console vengono raggruppati in funzione dell'operazione che li ha generati. È possibile nascondere o mostrare i messaggi relativi ad una determinata operazione cliccando sull'intestazione della medesima. È possibile anche nascondere le intestazioni cliccando sul pulsante `<*>` nella testata.
- 4) Le query eseguite ora rimandano alla riga anche se la modalità full-trace non è attiva.
- 5) La visualizzazione dello stack delle eccezioni è stata migliorata. Cliccando sull'eccezione si viene rimandati alla riga di codice che l'ha generata; cliccando sul nome di un metodo nello stack trace, esso viene aperto nell'editor di codice.

```
x Cannot set properties of null (setting 'errore') nel NewDTT.onLoad alla riga 35
Stack trace
@ App.NewDTT.onLoad (NewDTT.js:106:71)
@ new App.NewDTT (NewDTT.js:91:10)
@ App.Session.show (app.js:2078:11)
@ App.Session.getDefaultInstance (app.js:
@ Function.App.NewDTT.show (NewDTT.js:359
@ App.Session.onStart (FirstApp.js:7:40)
@ App.Session.processStart (app.js:1648:1
@ Object.App.processAppMessage (app.js:52
```

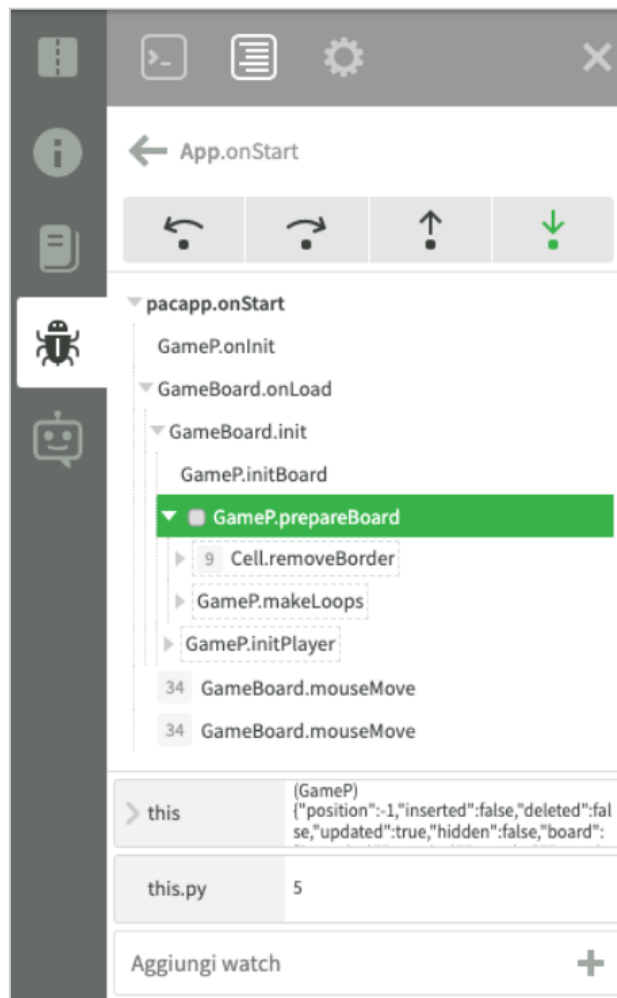
- 6) Ora viene gestito anche **console.trace**, oltre a console.log, console.warn e console.error.
- 7) Se il codice si blocca, ad esempio a causa di un ciclo infinito, il modulo di debug lo interrompe dopo 20 secondi e mostra il metodo che ha creato il loop.
- 8) I messaggi di log troppo grandi vengono troncati. È possibile modificare i limiti di log degli array modificando le costanti *App.DTT.ArrayLogLimit* (default 100) e *App.DTT.StringLogLimit* (default 5000). I messaggi di log vengono troncati anche nel log strutturato in modo da evitare di utilizzare troppo spazio nel disco del server.

Sezione lista operazioni

- 9) Se la modalità full trace non è attiva, cliccando nella lista delle operazioni viene aperta la corrispondente sezione della console.
- 10) Per ogni operazione, viene mostrato anche il numero di query effettuate. In questo modo è facile capire quali operazioni generano il maggior carico sul database.



- 11) Se la medesima operazione viene ripetuta, essa verrà mostrata come raggruppata nella lista.
- 12) La modalità full trace ora ha prestazioni adeguate in tutte le condizioni di utilizzo. Il flag full trace, quindi, non si auto-disabilita più all'avvio dell'anteprima.
- 13) È possibile aggiungere watch inserendo l'espressione nel campo di input in fondo all'elenco dei watch della richiesta attuale. Rimane sempre possibile aggiungere watch tramite il menu contestuale dell'editor di codice.
- 14) La navigazione nella traccia di codice (step-over) adesso mantiene il metodo anche in caso di operazioni asincrone. La modalità step-into invece segue tutto il codice eseguito durante l'operazione asincrona.

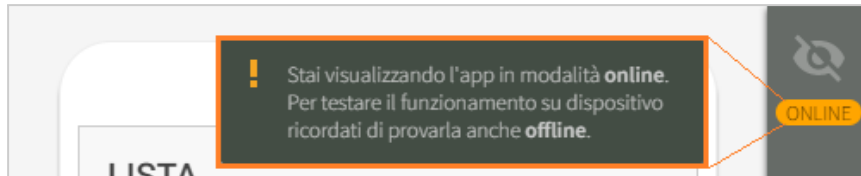


Generazione del codice

- 15) La generazione del codice per l'avvio dell'applicazione in anteprima è stata velocizzata del 40%.
- 16) È ora possibile aggiornare in real time il codice dell'applicazione mentre essa è mostrata in anteprima nell'IDE senza dover riavviare.

Indicatore modalità online/offline

- 17) Nella barra destra della finestra di anteprima viene ora visualizzata la modalità di esecuzione dell'applicazione (online/offline). L'indicatore è di colore arancione se la modalità è online e l'applicazione è mobile. In questo caso, cliccando sull'indicatore, appare un warning che ricorda di testare l'applicazione anche in modalità offline.



Database Profiler

- 18) Nella console di debug, per ogni query eseguita vengono mostrate le seguenti informazioni aggiuntive:
- Pending operations*: indica quante altre query o statement sono in attesa di essere eseguiti sull'istanza di database. Questo valore deve essere il più basso possibile e non viene mostrato se è pari a zero.
 - Total rows*: numero di record restituiti dalla query.
 - Elapsed time*: tempo totale necessario all'esecuzione della query, comprensivo di tempo di attesa delle altre query in coda sulla medesima istanza.
 - Execution time*: tempo necessario all'esecuzione della query vera e propria.
 - Wait time*: tempo di attesa in coda prima dell'esecuzione della query.

```
⚡ select id from tabl where (id = '5')
Pending operations: 5
Total rows: 0
Elapsed time (ms): 46
Execution time (ms): 0
Wait time (ms): 46
Vai alla riga
```

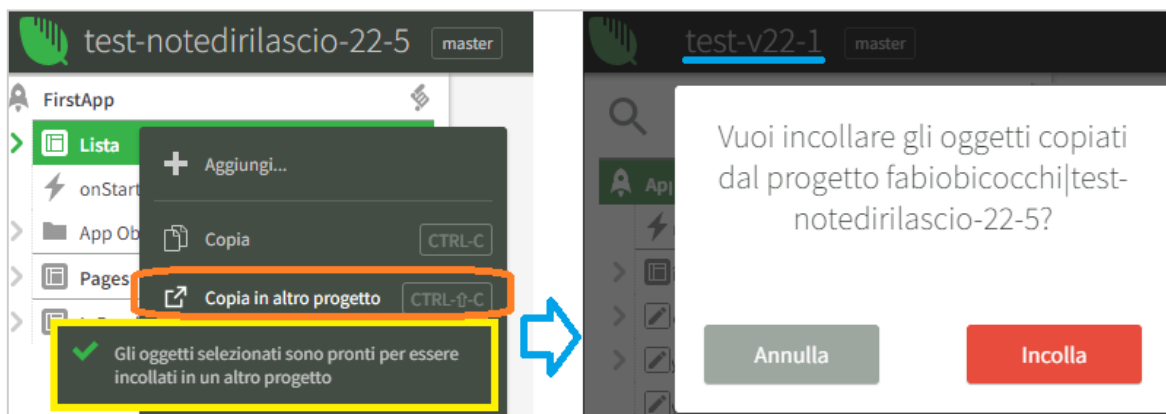
- 19) È disponibile un profiler globale di tutte le query eseguite dai database in un determinato intervallo di tempo. Per attivare il profiler, occorre impostare il parametro di applicazione *databaseProfiler* al valore *true*. Si ricorda che i parametri di applicazione si impostano tramite la console. Mentre questo parametro è *true*, per ogni query presente nel codice dell'applicazione vengono raccolte una serie di statistiche. Quando il parametro viene impostato a *false*, i dati raccolti vengono scritti in una serie di file privati nel file system dell'applicazione dal nome *dbprofiler-nnnnnn.json*. In particolare ogni worker attivo scrive il proprio file.
- 20) È stata aggiunta la possibilità di cliccare su "Vai alla riga" in modo da poter risalire alla riga di codice che ha generato la query.

IDE

Aggiunta della voce di menu contestuale “Copia in altro progetto”

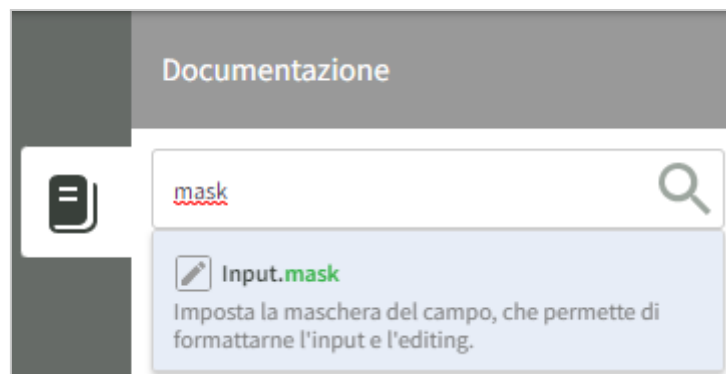
È stata aggiunta la voce di menu contestuale "Copia in altro progetto" all'albero degli oggetti. Prima di questa versione era già presente la possibilità di effettuare la copia tra progetti, ma solo mediante scorciatoia da tastiera **CTRL-SHIFT-C**.

La copia tra progetti è possibile solo tra progetti della stessa organizzazione: cliccando su questa voce gli oggetti selezionati vengono preparati per essere portati in un altro progetto. Dopo aver copiato gli oggetti utilizzando la nuova voce di menu, aprire un secondo progetto nello stesso browser, cliccare nell'albero nel punto in cui si desidera incollare e premere **CTRL-V**.



Ricerca nella documentazione più veloce

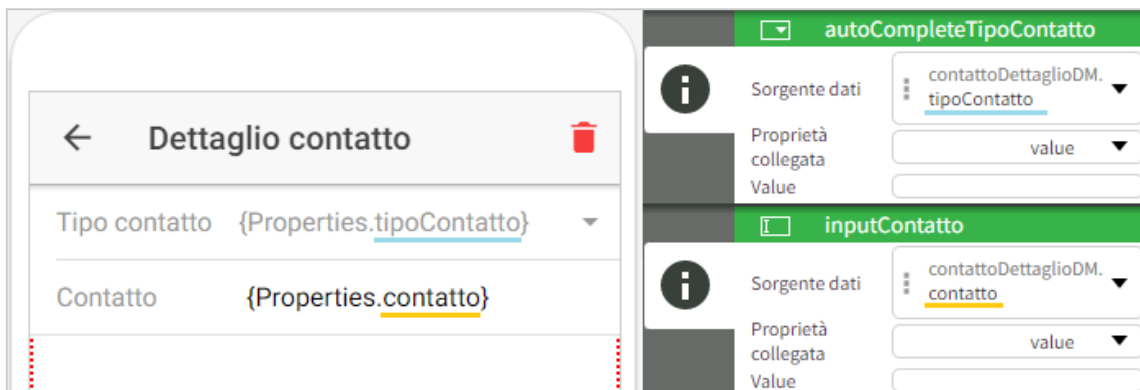
La ricerca nella documentazione delle funzioni di libreria ora fornisce i risultati immediatamente. Non ci sono più tempi di attesa.



Indicazione data source degli elementi nell'editor delle videate

Gli elementi visuali per i quali è stata impostata una data source ora vengono evidenziati nell'editor delle videate, come mostrato nell'immagine sottostante. Si consiglia quindi di non impostare la proprietà *value* a design time in modo da poter vedere la data source dell'elemento.

Se l'elemento ammette la visualizzazione di label, la data source viene mostrata come tooltip. Se la proprietà collegata o il tooltip vengono impostati a design time, viene mantenuto il comportamento delle versioni precedenti.

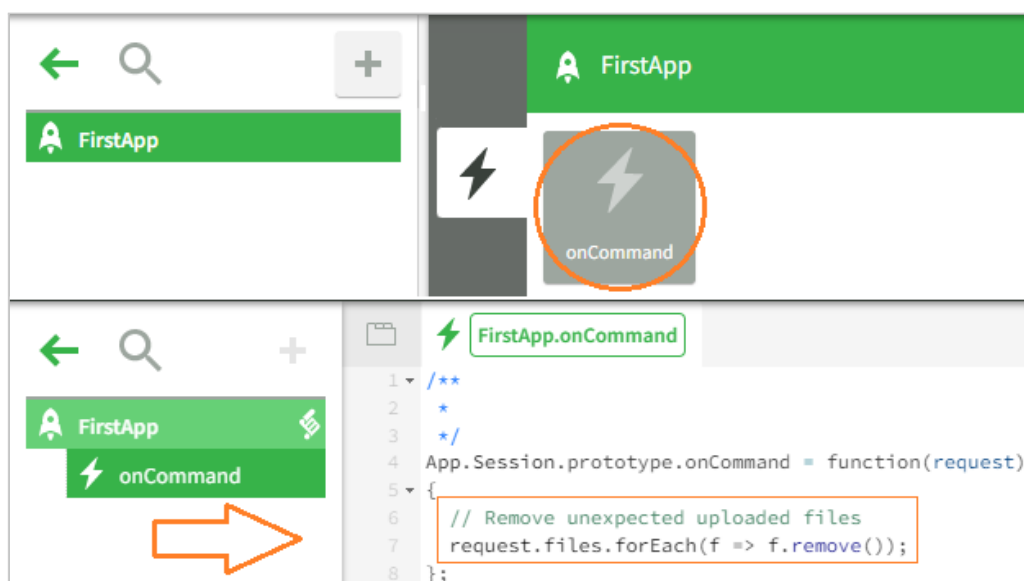


Gestione di file tramite evento `app.onCommand`

Come descritto nelle release note della [versione 22.0 \(Eliminazione di file ricevuti e non gestiti\)](#), i file ricevuti da applicazioni di terze parti vengono automaticamente cancellati se l'applicazione non gestisce l'evento `app.onCommand`.

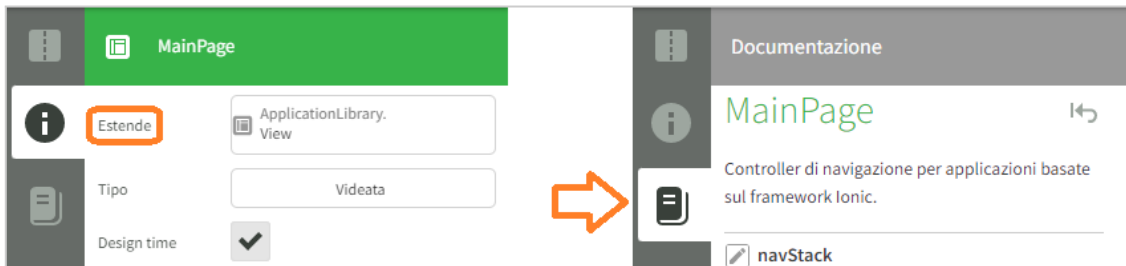
Quando si implementa l'evento `onCommand`, è necessario gestire esplicitamente i file ricevuti, perché la cancellazione automatica non avviene più. Per evitare di dimenticare questa importante funzione, ora l'IDE aggiunge esplicitamente una riga di codice all'evento `app.onCommand` quando questo viene aggiunto all'applicazione in modo da recuperare la cancellazione dei file caricati.

Si consiglia di modificare la riga aggiunta automaticamente in modo da ottenere il funzionamento desiderato.



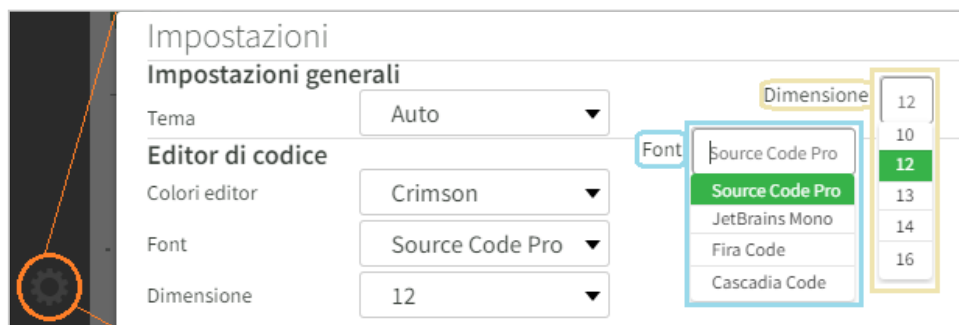
Documentazione delle proprietà di un elemento visuale

Cliccando sull'intestazione di una proprietà di un elemento nell'editor delle proprietà, adesso viene mostrata la documentazione relativa.



Cambiare tipo e dimensione del font nell'editor

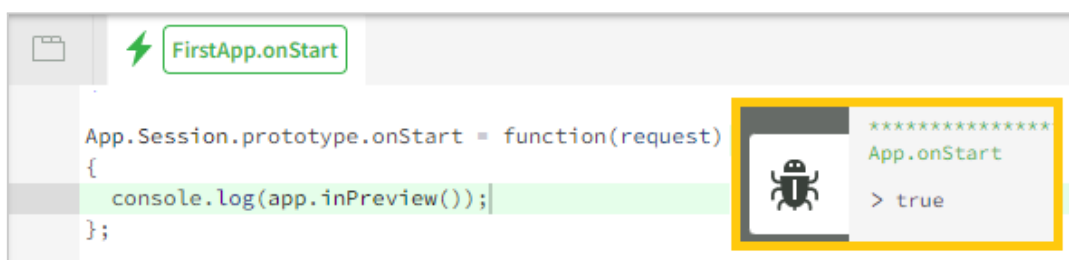
Nelle impostazioni dell'IDE è stata aggiunta la possibilità di cambiare tipo e dimensione del font per l'editor di codice del progetto. I font a disposizione sono *Source Code Pro* (default), *JetBrains Mono*, *Fira Code* e *Cascadia Code*, mentre le dimensioni selezionabili sono 10, 12 (default), 13, 14 e 16.



Applicazione

Funzione inPreview per verificare se un'app è avviata da IDE

È stata aggiunta la funzione `inPreview()` all'oggetto `app` che restituisce un valore booleano per indicare se l'applicazione è stata avviata da IDE (`true`) o installata in un server o in un launcher (`false`).



Possibilità di configurare la pagina di avvio

Quando un'applicazione viene installata su un server di produzione, il link di avvio è `nomeServer/nomeApplicazione` e di default viene servita la pagina `index.html`. Da questa versione è possibile configurare la pagina di avvio delle applicazioni installate su un server di produzione.

Per farlo è sufficiente valorizzare il parametro `startPage` nella videata di configurazione dell'applicazione nella console di Instant Developer Cloud. Tale parametro deve contenere un url relativo alla directory client dell'applicazione.

INSTALLAZIONE ARRESTATA

@installationName @buildName
@server session abilitata - @modalità offline

Installato il @Date da @MemberName
Build @buildname creata il @buildDate dal progetto @projectName di @username.

Stato e configurazione

Server session
Se abilitato mantiene sempre attiva una sessione server senza interfaccia utente. Utile per eseguire procedure periodicamente.

Modalità offline
Permette di avviare l'applicazione a partire da un URL del server e di farla eseguire localmente all'interno del browser.

Pagina di avvio

Url di avvio personalizzata dell'applicazione per sovrascrivere il valore default `/client/index.html`. Questo parametro può essere valorizzato con il percorso relativo di un file contenuto nell'applicazione (es. `offline.html` per far partire sempre la versione offline dell'app).

Spazio usato Non disponibile [SFOGLIA](#)

Database Non disponibile [QUERY](#)

Per esempio l'applicazione `ToBuyApp`, installata sul server `prod3-pro-gamma`, ha il seguente url di avvio: <https://prod3-pro-gamma.instantdevelopercloud.com/ToBuyApp> che serve la pagina `/client/index.html`. Per avviare sempre l'applicazione come una PWA occorre aggiungere i seguenti parametri di runtime nell'installazione sulla console:

- `allowOffline = true`
- `startPage = offline.html`

Il primo parametro permette l'utilizzo offline dell'applicazione e il secondo forza la partenza in modalità offline.

Aggiornato il componente grafico Chart.js

Il componente grafico *Chart.js* è aggiornato all'ultima versione disponibile compatibile con i progetti esistenti, la versione 2.9.4.

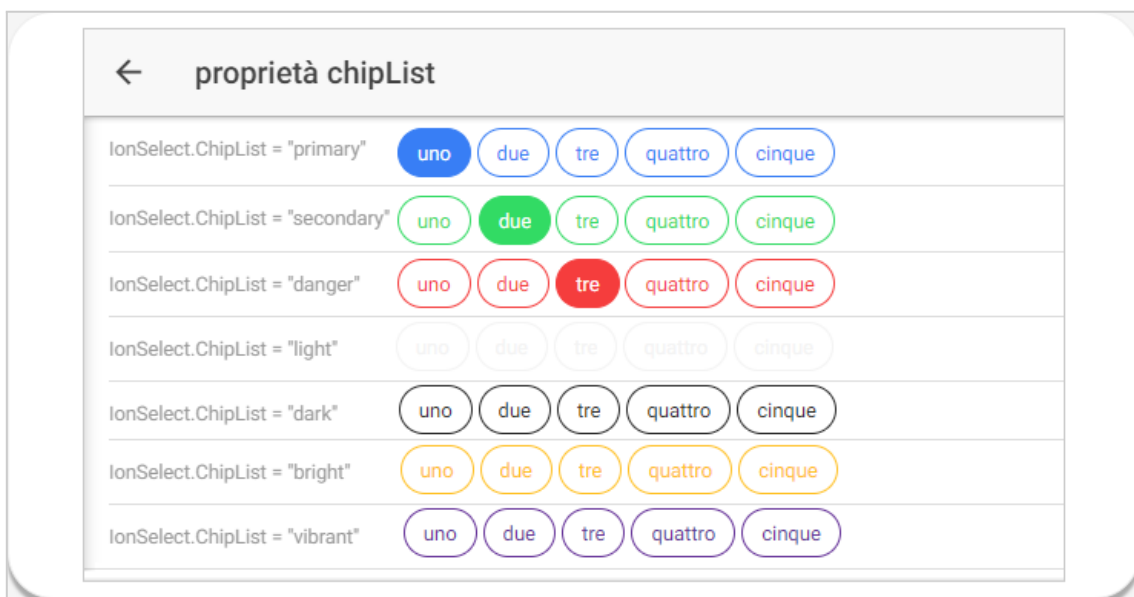
Attenzione: anche se il produttore dichiara compatibilità con la versione precedente, si consiglia di testare nuovamente le proprie applicazioni che fanno uso di *Chart.js*.

Chart.js è l'elemento che visualizza grafici tramite l'omonima libreria: per ulteriori informazioni è possibile visitare il [sito ufficiale](#).

Ionic

Proprietà chipList dell'elemento IonSelect

È stata aggiunta la proprietà booleana *chipList* all'elemento *IonSelect*. Attivando questa proprietà gli elementi della lista verranno visualizzati sotto forma di pulsanti selezionabili. Si consiglia di attivare questa opzione quando il numero di elementi è basso (minore di 20), come ad esempio avviene in un sondaggio in cui l'utente può scegliere fra diverse opzioni di punteggio.



Miglioramenti elemento IonAutoComplete

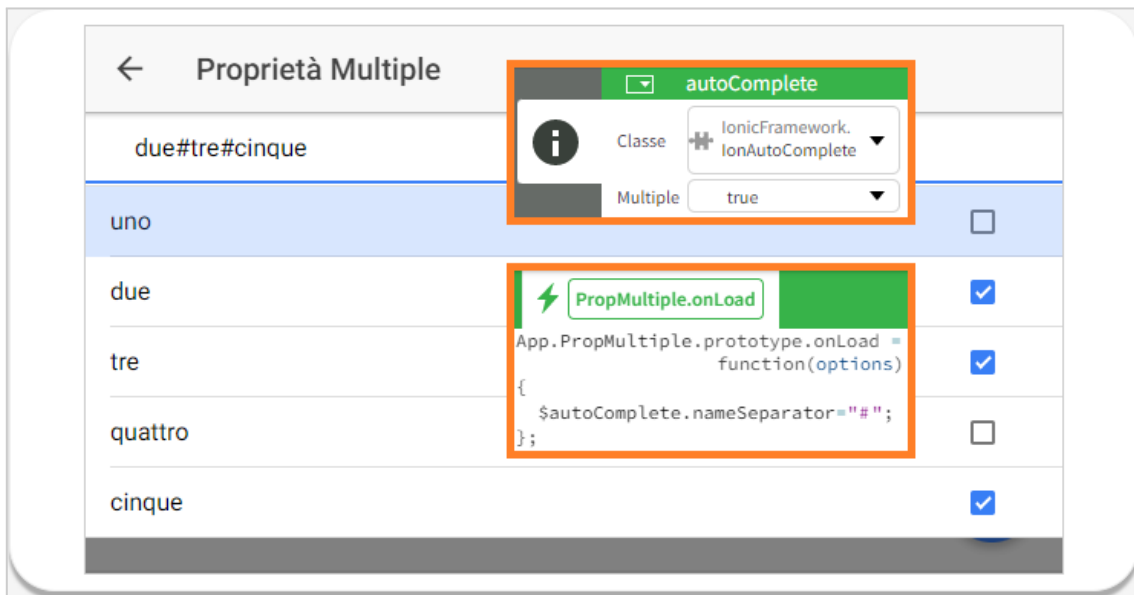
Proprietà *readOnly* in *IonAutoComplete* e *IonSelect*

È stata aggiunta la proprietà *readOnly* agli elementi *IonAutoComplete* e *IonSelect*. Questa proprietà permette di rendere in sola lettura i due elementi, senza che risultino disabilitati.

Proprietà *multiple*: *IonAutoComplete* permette la selezione multipla

È stata aggiunta all'elemento *IonAutoComplete* la proprietà booleana *multiple*, che permette di selezionare più valori, come già avviene per l'elemento *IonSelect*.

Sono state aggiunte anche le proprietà *valueSeparator* e *nameSeparator* per permettere di modificare i separatori di valori e nomi, di default impostati a "," (virgola).



Proprietà disabled per una singola riga della combo-box

È stata aggiunta la possibilità di specificare la proprietà booleana *disabled* per ogni elemento della lista, così da poter disabilitare solo alcune righe della combo-box. Se ad esempio si imposta la proprietà *list* come segue, la seconda riga risulterà disabilitata:

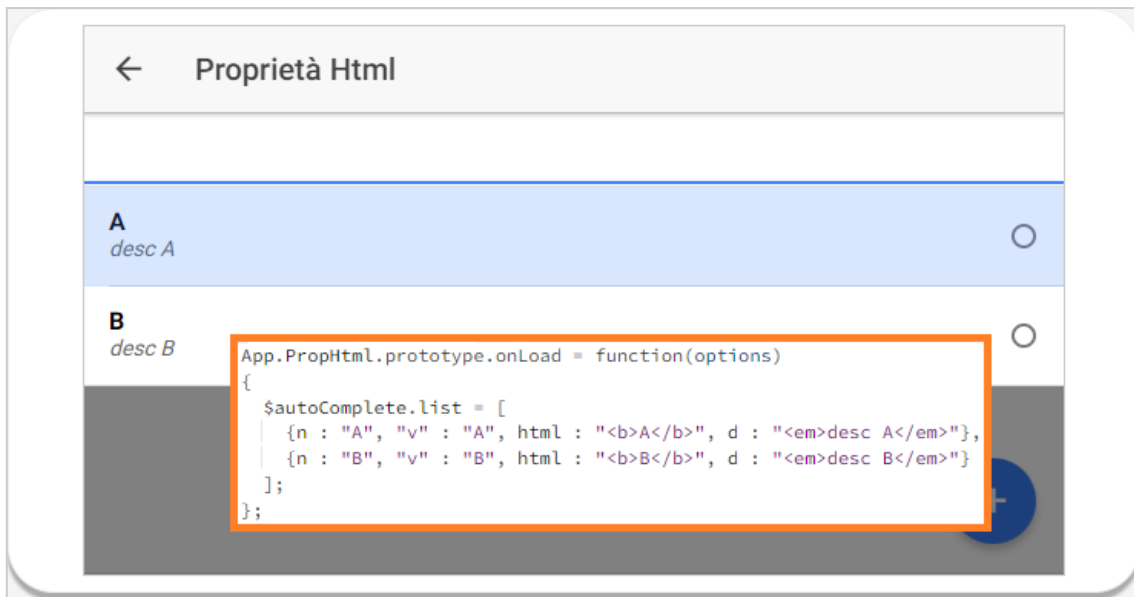
```
$autocomplete.list = [  
  {n:"Riga 1", v:"r1"},  
  {n:"Riga 2", v:"r2", disabled:true}  
];
```

Proprietà html per una singola riga della combo-box

Durante la creazione della lista dell'elemento *IonAutoComplete* è possibile specificare anche la proprietà booleana *html* per ogni item della lista. Questa nuova proprietà permette di rendere il testo della riga come stringa html invece che come valore testuale.

Attivando la proprietà *html*, occorrerà comunque indicare anche la corrispondente descrizione testuale come proprietà *n* della riga: questa infatti verrà usata per comporre la descrizione dei valori quando la lista è chiusa. Utilizzando la proprietà *html* anche la proprietà *d* (dettaglio) verrà resa come stringa html.

Si ricorda infine che le stringhe visualizzate come html devono essere opportunamente sanificate per evitare problemi di sicurezza lato client. A tal fine è presente il metodo *App.Utils.htmlEscape()*.

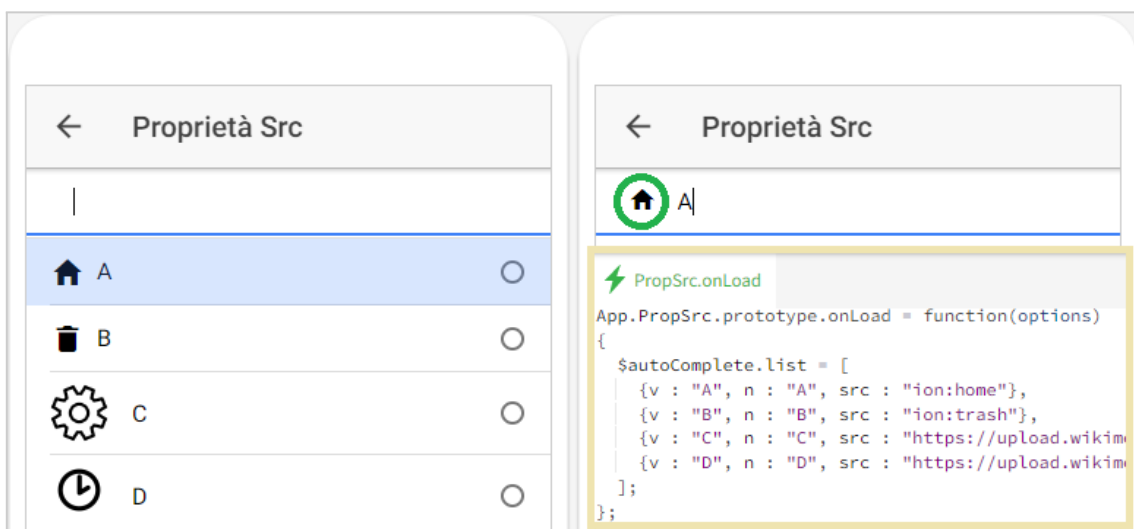


Icona mostrata anche nell'input di IonAutoComplete (**comportamento modificato**)

Se gli elementi della lista hanno delle immagini o icone, ora queste vengono mostrate anche nell'input dell'elemento *IonAutoComplete*, accanto al valore selezionato. Precedentemente era possibile impostare le immagini, ma queste venivano visualizzate solo nella lista di selezione quando aperta.

Se fosse necessario ripristinare il comportamento precedente, è necessario inserire nel foglio CSS dell'applicazione la regola:

```
.ion-autocomplete .value-icon {
  display:none;
}
```



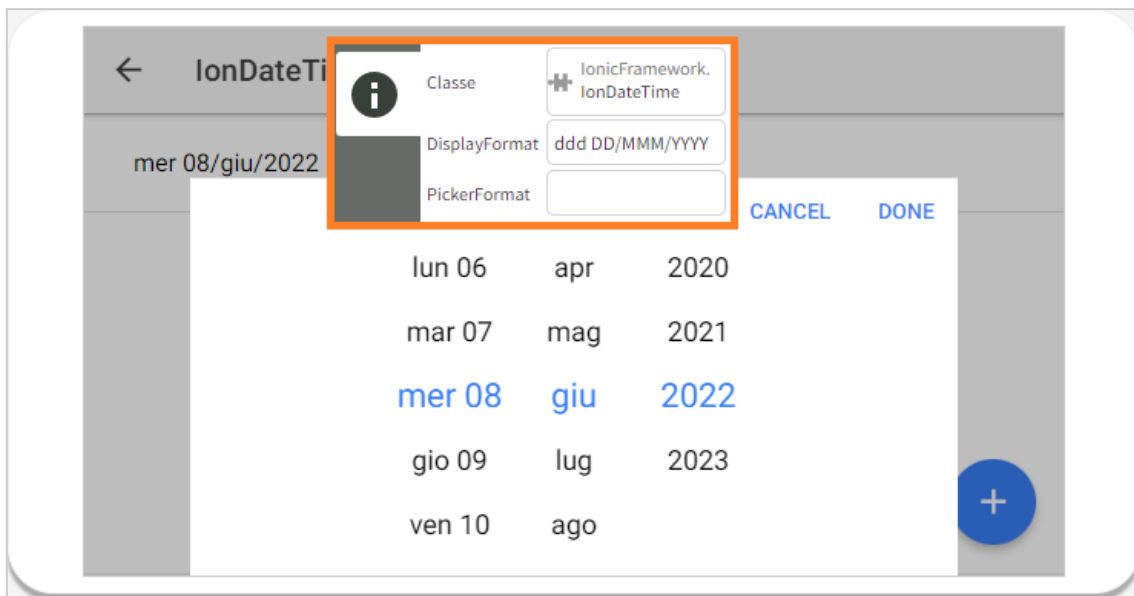
Si ricorda che per mostrare un'immagine occorre valorizzare la proprietà *src* con l'url relativo. Per visualizzare le icone, invece, *src* deve essere impostata a "ion:nomeicona".

L'elemento `IonDateTime` può mostrare il nome del giorno

L'elemento `IonDateTime` è ora in grado di mostrare il nome del giorno, anche nella lista di selezione.

Il nome del giorno è associato al token `DDD` nella proprietà `pickerFormat`. Se, ad esempio, si valorizza questa proprietà con `"DDD/MMM/YYYY"`, si ottiene `"mer 08 giu 2022"`.

Un modo alternativo per ottenere questa formattazione è quello di valorizzare la proprietà `displayFormat` a `"ddd DD/MMM/YYYY"`, lasciando vuota la proprietà `pickerFormat`. Il token `ddd DD` in `displayFormat` viene infatti interpretato come `DDD` a livello di `pickerFormat`.



Scrollbar verticale (comportamento modificato)

Se il contenuto della pagina è più alto della pagina stessa, ora la pagina acquisisce una scrollbar verticale se viene visualizzata in un browser `"desktop"` di tipo webkit (Chrome, Safari, Opera, Edge, ecc...).

Se si desidera ripristinare il comportamento precedente, è necessario aggiungere al foglio CSS dell'applicazione la seguente regola:

```
ion-nav.has-scrollbar scroll-content: hover: : -webkit-scrollbar {
  display: none;
}
```



Database

Metodo setConnectionOptions

Quando l'applicazione viene installata su un server My Cloud o Self Managed è possibile impostare le opzioni di connessione delle varie istanze di database tramite il metodo *db.setConnectionOptions*.

A seconda del tipo di database (Postgres, Oracle, SQLServer, MySQL) la forma dei parametri di connessione cambia, in funzione del driver node.js che viene utilizzato. L'esempio seguente è relativo ad un database SQL Server.

```
{
  "server" : "your_server",
  "port" : 1433,
  "database" : "your_database"
  "authentication" : {
    "type" : "default",
    "options" : {
      "userName" : "your_username",
      "password" : "your_password"
    }
  }
}
```

Specificare le opzioni di connessione al server per il Cloud Connector

Nel caso di utilizzo del Cloud Connector collegato ad un server di produzione di tipo My Cloud corredato da un certificato HTTPS autogenerato, potrebbero verificarsi errori di connessione legati all'impossibilità di verificare il certificato da parte del Cloud Connector.

Sebbene questa pratica sia sconsigliata per motivi di sicurezza, ci sono casi in cui è stata richiesta la possibilità di impostare le opzioni dei parametri di connessione della socket tra i due server.

È quindi ora possibile specificare nel file *config.json* di configurazione del Cloud Connector le opzioni da usare in fase di connessione ed in particolare il parametro *rejectUnauthorized* impostandolo a *false*. Di seguito un esempio di impostazione del parametro:

```
{
  "name": "{connectorName}",
  "remoteServers": [{"applicationUrl1"}, {"applicationUrl2"}],
  "remoteUserNames": [{"ideUsername"}, {"ideOrganization/IdeUsername"},
    {"ideUrl@ideUsername"}],
  "remoteConfigurationKey": "00000000-0000-0000-0000-000000000000",
  "connectionOptions": {
    "rejectUnauthorized": false
  }
}
```

Attenzione: esporre su internet un server di produzione con certificati HTTPS non sicuri è fortemente sconsigliato. Utilizzare questo parametro solo nella fase iniziale di test del server fino a quando non si sia installato un certificato valido.

Per chiarimenti sui tipi di server My Cloud consultare il manuale [I server di produzione](#) al capitolo *I server My Cloud*.

DO

Riassegnamento di collection figlie (comportamento modificato)

L'assegnamento da codice di una collection figlia di un documento genera un warning a runtime e l'operazione viene annullata.

Non è corretto riassegnare una collection figlia caricata dal documento sostituendola con un'altra in quanto una collection non può avere più padri.

Supponiamo, per esempio, di avere un documento Category con una collection figlia Prodotti di documenti di tipo Product.

Se si esegue un'operazione come da codice seguente:

```
let c1 = yield App.BE.Category.loadByKey(app, {CategoryID : 1}, {childLevel : 1});
let c2 = yield App.BE.Category.loadByKey(app, {CategoryID : 2}, {childLevel : 1});
c1.Prodotti = c2.Prodotti
```


viene mostrato il seguente warning in console di debug:

It isn't possible to reassign the collection of a document: the statement was ignored nel `CategoryManagement1.onLoad` alla riga 10

Il modo corretto di operare è spostare i documenti da una collection all'altra come nell'esempio seguente:

```
c1.Prodotti.clear();
let numProd = c2.prodotti.count;
for (let i = 0; i < numProd; i++) {
  let prod = c2.Prodotti.rows[i];
  c1.Prodotti.add(prod);
}
```

SYNC

Evento `onDocUpdate` notificato in più casi (**comportamento modificato**)

L'evento `onDocUpdate` viene ora notificato nelle applicazioni offline anche alla sessione che ha generato la variazione, come avveniva già nel caso online. Il comportamento tra applicazione online ed offline è stato quindi ulteriormente uniformato.

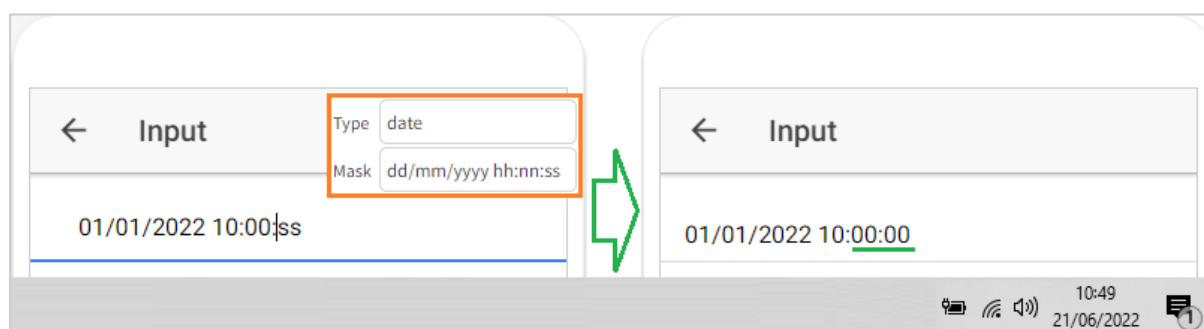
Questo cambiamento di comportamento in generale non richiede alcuna modifica al codice applicativo perché le sessioni online presentavano già questo comportamento. Si consiglia comunque di controllare il codice degli eventi `onDocUpdate` gestiti dalla propria applicazione.

Correzioni della versione 22.5

Applicazione

Input date con mask: orario con “00” minuti

È stato corretto un caso particolare degli elementi di input di tipo “date” che si verificava quando la proprietà mask conteneva anche i token relativi alla parte ora. Correzione relativa alla segnalazione 000765-2022.



Lingua di localizzazione GoogleCharts

I grafici generati usando il componente GoogleCharts vengono ora localizzati automaticamente usando la lingua del browser. Segnalazione relativa al ticket 000413-2022.

Bootstrap

Correzione messaggio di errore per BSInput di tipo checkbox

Impostando un messaggio di errore tramite la proprietà *errorText* su un elemento di tipo *BSInput checkbox*, ora il checkbox rimane visibile. Nelle versioni precedenti risultava nascosto. Segnalazione relativa al ticket 000785-2022.

Database

Default Timezone “Europe/Rome” per MySQL e Oracle

Per i database MySQL e Oracle è stato impostato il valore di default di *timezone* a “Europe/Rome”. Precedentemente, le query che contenevano valori *datetime* generavano un errore se nella definizione dei corrispondenti campi del database non era stata specificata esplicitamente la *timezone*.

Datamap

Migliorata la gestione delle date negli IonInput

La gestione delle date da parte dell'elemento *IonInput* di tipo *date*, *time* e *datetime-local* è stata uniformata a quella degli elementi *IonDateTime*. Segnalazione relativa al ticket 000283-2022.

Si ricorda che è comunque consigliabile usare elementi *IonDateTime* per trattare l'input di un valore di tipo data.

Filtrare campi unbound delle datamap

È stata aggiunta la possibilità di filtrare anche i campi *unbound* delle datamap. Precedentemente, impostando filtri sulle proprietà *unbound* si otteneva un errore in fase di caricamento. Esempio di codice:

```
$dmProdotti.clearFilters();  
$dmProdotti.addFilter("CampoUnboundDM", "ch");  
$dmProdotti.reload();
```

Document Orientation

Messaggio di warning reso più esplicito

Quando vengono trovati più risultati in seguito al caricamento di un documento, viene ora specificata nel messaggio di malfunzionamento anche la sua classe: *"The document <ClassName> was not loaded because it has been found more than one result"*. Il messaggio può essere di errore o warning, a seconda del metodo che lo ha generato.

File system

Query string con caratteri codificati

La classe *app.fs.url* ora applica correttamente la decodifica di parametri query string contenenti caratteri codificati. Ad esempio, la query string contenente *?par=a%2Bb* viene correttamente interpretata come *a+b*.

Segnalazione relativa al ticket 000193-2022.

IDE

Importare campi foreign key nel datamodel

Da questa versione 22.5 è di nuovo possibile creare foreign key verso campi che non sono primary key. Nella versione 21.5 era stato aggiunto un controllo in fase di validazione per garantire che le FK puntassero solo a campi PK: questa restrizione è stata ora rimossa.

Segnalazione relativa al ticket 000358-2022.

Caso particolare di utilizzo di funzioni arrow

È stato corretto un malfunzionamento relativo all'utilizzo di funzioni arrow, che si poteva evidenziare aggiungendo una riga e salvando in un metodo contenente il seguente codice:

```
let a = [].map(function (x) {return x});  
a.reduce((x, y)=>x + y, 0);
```

Segnalazione relativa alla richiesta di assistenza 000282-2022.

Classe CSS invalida

È stato protetto il caso in cui in una risorsa CSS viene specificata una classe invalida, come ad esempio “*..elenco*“. Segnalazione relativa alla richiesta di assistenza 000114-2022.

Applicazione copiata tra progetti

Copiando un'applicazione da un progetto ad un altro, per la nuova applicazione viene ora generato un differente ID applicazione.

Malfunzionamento relativo alla richiesta 000576-2022.

Sistema di localizzazione

È stata corretta la visualizzazione delle liste che contengono molte stringhe da localizzare. Correzione relativa al ticket 000887-2022.

Spazio nelle stringhe template

Precedentemente, dopo aver definito una stringa template, in alcuni casi l'editor di codice inseriva spazi non voluti tra i simboli delle espressioni. Ora, invece, la stringa tra gli accenti gravi `` viene mantenuta fedelmente, senza modificarne la spaziatura interna.

Si ricorda che l'utilizzo di stringhe template è sconsigliato in quanto non sono compatibili con il modulo di gestione della localizzazione.

Malfunzionamento relativo alle richieste di assistenza 000119-2021 e 000170-2021.

Gestione degli spazi nelle righe di codice

Sono stati corretti alcuni casi di inserimento di spazi all'interno di righe di codice, che non generavano errori applicativi, ma rendevano il codice meno leggibile.

Evento definito in un nuovo plugin

Ora è possibile implementare eventi definiti nei propri plugin nativi. Precedentemente gli eventi dei plugin non apparivano nella lista di quelli implementabili e quindi era più difficile poterli definire. Segnalazione relativa al ticket di malfunzionamento 000228-2022.

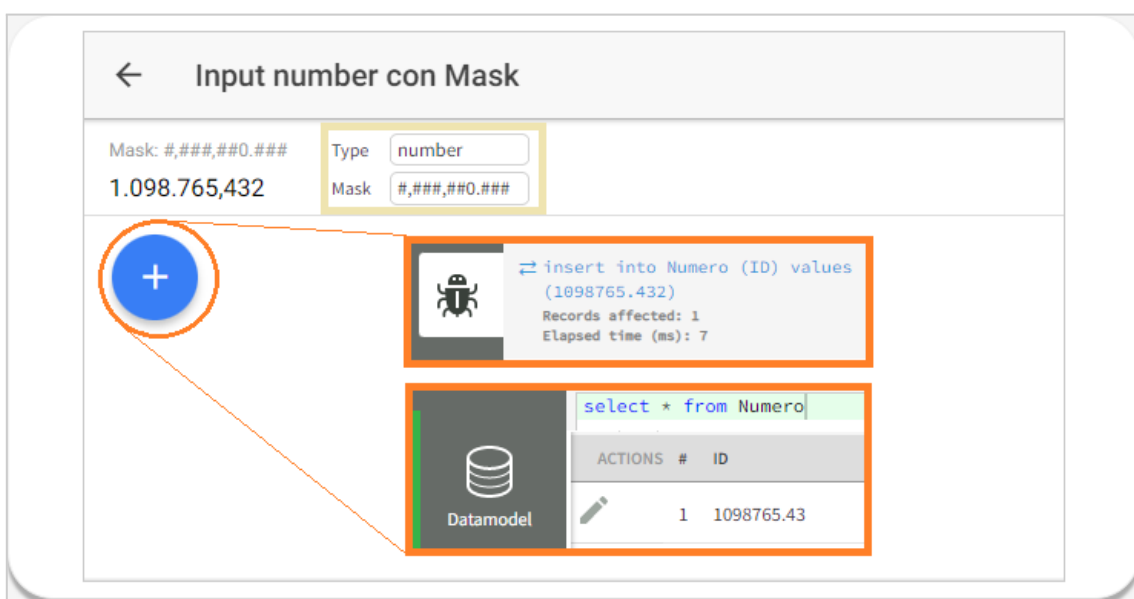
Ionic

Allineamento elemento freccia IonAutoComplete e IonSelect

La posizione dell'elemento grafico "freccia" visualizzato sulla destra del componente *IonAutoComplete* è stata allineata con quella di *IonSelect*. In questo modo, posizionando questi elementi uno sotto l'altro, le relative frecce risultano correttamente posizionate.

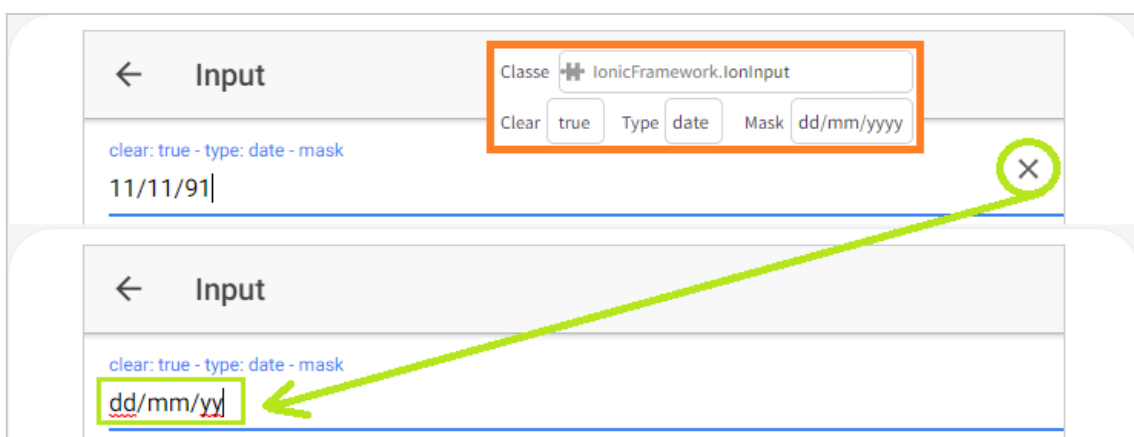
Input number con mask per numeri oltre il milione

Negli Input di tipo "number" con la proprietà *mask* configurata per prevedere l'inserimento di valori superiori al milione, le cifre inserite vengono ora comunicate correttamente al server. Malfunzionamento relativo al ticket 000731-2022.



Pulsante clear negli input date

Il pulsante *clear* in un elemento *IonInput* di tipo *date* ora ripristina correttamente la visualizzazione della maschera da compilare quando viene attivato.



Animazione toast su dispositivi iOS

È stata corretta l'animazione dei popup di tipo *toast* sui dispositivi iOS con il notch/safe area. Correzione relativa al ticket 000022-2022.

SYNC

Metodo `compareData`

È stato corretto il metodo `compareData` utilizzato per confrontare i dati attesi in un dispositivo con quelli effettivamente presenti e scoprire eventuali problemi di sincronizzazione.

Attenzione: la correzione richiede la ripubblicazione sia della parte server che di quella client.

Correzione relativa al ticket 000965-2022.

Sincronizzazione con database multi-tenant

È stata corretta la sincronizzazione in real-time per i database multi-tenant per cui ora non vengono notificati messaggi a sessioni con tenant differenti.

Precedentemente poteva succedere che avendo topics "ambigui" per più tenant, alcuni messaggi venissero consegnati a sessioni con tenant differenti.

Sincronizzazione differenziale in loop

È stato corretto un caso particolare in cui durante la sincronizzazione differenziale si poteva verificare un loop se lato server veniva caricato un intero blocco di variazioni che non corrispondeva ai topics del client.

Disconnessione durante la sincronizzazione differenziale

È stato protetto un caso particolare in cui, durante la sincronizzazione differenziale, si verifica una disconnessione temporanea e i messaggi di variazione potevano essere comunicati al server più volte, generando così errori di inserimento dati.

Metodo `minimizeVariations`

È stato corretto il metodo `minimizeVariations` che poteva generare l'errore "*msg is not defined*". Segnalazione relativa a questo [thread](#) del forum.

Launcher

Apertura file da browser su Android

A causa di una modifica alla politica di accesso ai file da parte di Android, sulle app con target Platform SDK 30 non era più possibile aprire nel browser (tramite `app.open()`) i file presenti sul dispositivo in una cartella esterna a quella dell'app. Questo problema aveva un impatto sui launcher compilati con la versione 21.5.0 o successiva del template launcher.

Ritardo nella connessione sync su Android

A causa di modifiche al servizio safe-browsing presente in Android, la connessione al server di sincronizzazione era più lenta del previsto. Ripubblicando l'applicazione con il template launcher della versione 22.5 questo problema viene risolto.

Segnalazione relativa al ticket 000825-2022.

Server

Errore d'installazione di un pacchetto personalizzato

Se durante l'installazione di un pacchetto personalizzato in un server IDE o di produzione si otteneva un errore di installazione, il sistema non ripristinava correttamente il file di configurazione *package.json*. Questo poteva impedire successive installazioni di pacchetti personalizzati.

Segnalazione relativa al ticket 000758-2022.

WebAPI

Metodo loadByKey su classe importata da WebAPI Foundation

Il metodo *loadByKey* applicato ad una classe importata da WebApi Foundation è stato corretto in alcuni casi particolari. Malfunzionamento relativo al ticket 000473-2022.

Note di rilascio di versioni precedenti

Di seguito la lista delle note di rilascio delle ultime tre versioni di Instant Developer Cloud.

VERSIONE 22.0	VERSIONE 21.5	VERSIONE 21.0
[01/02/22] Versione 22.0 [15/02/22] Versione 22.0.1 [28/02/22] Versione 22.0.2 [08/04/22] Versione 22.1 [07/06/22] Versione 22.1.1	[19/07/21] Versione 21.5 [01/09/21] Versione 21.5.1 [29/09/21] Versione 21.5.2 [29/11/21] Versione 21.5.3	[08/02/21] Versione 21.0 [16/02/21] Versione 21.0.1 [02/03/21] Versione 21.0.2 [23/03/21] Versione 21.0.3 [07/04/21] Versione 21.0.4 [24/06/21] Versione 21.0.5