Versione 21.5 - Note di rilascio

Aggiornamento sistemi operativi

La versione 21.5 richiede Node in versione 12.18.0.

Aggiornamento package di base

In conseguenza di alcune modifiche apportate, è necessario aggiornare alla versione 21.5 i package *CoreLibraries* e *IonicUI* nei progetti in cui sono presenti. Il mancato aggiornamento potrebbe causare malfunzionamenti o differenze di comportamento rispetto alle versioni precedenti.

Compatibilità versioni precedenti

In versione 21.5 è stato aggiornato il componente socket.io. La nuova versione di socket.io non è totalmente compatibile con le versioni precedenti. In particolare la nuova versione (SERVER) è compatibile sia con la nuova che con le vecchie versioni (CLIENT). Purtroppo la vecchia versione (SERVER) non è compatibile con la nuova versione (CLIENT). Questo implica che non è possibile installare su un server di produzione in versione 21.0 o precedenti (che quindi utilizza la vecchia versione server di socket.io) un'applicazione buildata con la versione 21.5 (che utilizza la nuova versione client di socket.io).
Pertanto occorre aggiornare il server di produzione alla nuova versione prima di installare applicazioni sviluppate e compilate con la versione 21.5.

Novità della versione 21.5

IDE

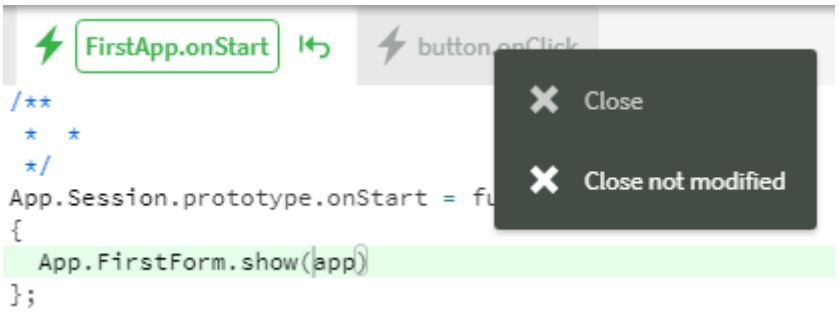
" È ora possibile copiare parti di un progetto tra progetti differenti. Per farlo è sufficiente:
- aprire il progetto sorgente
- selezionare nell'albero gli oggetti che si desidera copiare
- premere SHIFT-CTRL-C
- aprire il progetto di destinazione
- selezionare un oggetto adeguato che possa contenere gli oggetti da copiare
- premere CTRL-V

Instant Developer copierà gli oggetti selezionati e gli oggetti utilizzati dagli oggetti copiati per mantenere l'integrità degli stessi. Per esempio se si copia una videata contenente una o più query Instant Developer copierà, oltre alla videata, anche il datamodel e le relative tabelle utilizzate dalla videata qualora non fossero già presenti nel progetto di destinazione.
N.B.: Il sistema copia l'intero oggetto selezionato ma copia solo una parte degli oggetti referenziati da questo. Per esempio se viene copiata una videata che si riferisce ad una tabella del database il sistema, se non è già presente nel progetto di destinazione, non copia

l'intero database ma solo la tabella (o le tabelle) necessarie per poter copiare interamente la videata. Lo stesso avviene se si copia una datamap DO. In quel caso verrà copiato il documento (se non è già presente nel progetto di destinazione) ma ne verranno copiate solo le parti "essenziali" alla sussistenza della datamap DO. Pertanto non verranno copiati eventi e metodi del documento se questi non sono direttamente referenziati all'interno della datamap copiata. Qualora fosse necessario "completare" gli oggetti parzialmente copiati è sufficiente rieseguire la copia tra progetti di questi oggetti per copiare le parti mancanti. In alternativa è sufficiente copiare prima gli oggetti che si sa essere necessari (es: database, documenti) e poi gli oggetti che li utilizzano (videate e/o metodi).

" È stato migliorato l'algoritmo che aggiorna il codice dei metodi se modificato quando l'applicazione è in esecuzione. Sono stati gestiti diversi casi che non erano supportati. Pertanto ora è possibile, per esempio, cambiare il codice dei metodi quando l'applicazione è in esecuzione e la modifica verrà applicata immediatamente, senza che sia necessario fermare e riavviare l'applicazione. Ovviamente, se la modifica riguarda eventi non ripetibili (come per esempio onStart, onLoad delle videate) questi non saranno nuovamente notificati. Ma se le videate vengono chiuse e riaperte verrà notificato l'evento aggiornato.

" Ora le linguette relative a Code Editor modificati vengono evidenziate e spostate all'inizio della lista delle pagine aperte. Inoltre, oltre a poter chiudere tutte le pagine, è ora possibile chiudere tutte le pagine eccetto quelle modificate.



Nuova voce di menù che permette di chiudere solo le videate non modificate.

Document Orientation

" È stato aggiunto il metodo *setSession* ai documenti per permettere di associarli ad una sessione e disassociarli da una sessione. Questo metodo può essere utile quando si vuole tenere un riferimento ad un'istanza di documento in un contesto statico (es: App) e si vuole operare con l'istanza in sessioni differenti.

" Il metodo *datamap.toCollection* ora copia nei documenti anche i valori letti nel recordset che non corrispondono a proprietà del documento.

" Sono state implementate diverse ottimizzazioni nella gestione delle collection. Pertanto in versione 21.5 le istanze di documenti e collection utilizzano meno memoria delle versioni precedenti. Questo può essere importante se vengono create molte istanze di documenti e collection nello stesso algoritmo.

Sync

❝ È stato aggiunto l'evento *app.sync.onConnectionStatusChange* che viene notificato ogni volta che cambia lo stato della proprietà *app.sync.connectionStatus*.

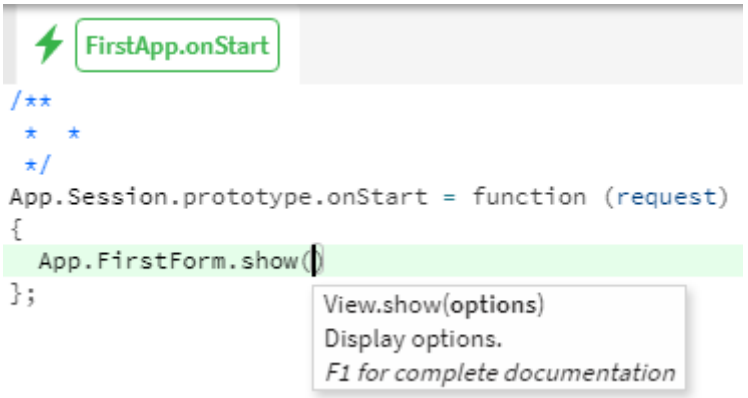
❝ Viene ora notificato l'evento *onVariationsProcessing* mentre vengono processate le variazioni ricevute dalla controparte. L'evento viene notificato periodicamente ogni secondo fino a che non ci sono più variazioni da processare. Questo evento può essere utile, per esempio, per mostrare all'utente un popup ed avvisarlo che l'app è impegnata per la sincronizzazione dei dati.

Miglioramenti della versione 21.5

IDE

❝ È stato reso più chiaro il testo del messaggio che viene mostrato quando si effettua il merge di un branch con quello attivo.

❝ È stato aggiunto un indicatore dei parametri di una funzione durante l'editing del codice. Tale tooltip mostra la documentazione relativa ai parametri evidenziando in grassetto il parametro di cui si sta editando il valore. È possibile disabilitare l'indicatore spegnendo il checkbox 'Mostra tooltip di aiuto' nella videata della configurazione dell'IDE.



Nuovo tooltip che mostra la documentazione relativa ai parametri delle chiamate

❝ Ora la ricerca nella documentazione cerca anche negli articoli relativi ai parametri delle funzioni.

❝ Ora le foreign-key con regole di cancellazione e aggiornamento impostate a NO-ACTION non vengono più create sul database. Inoltre, se si aggiorna lo schema, le foreign-key già presenti vengono eliminate poiché non necessarie.

Code Editor

“ Se si scriveva "insert into" e si portava il cursore alla fine l'intellisense del Code Editor non elencava le tabelle del datamodel.

APP

“ Il metodo *app.updateTheme* ora esegue anche il reset degli stili in modo da aggiornarli con le nuove proprietà di tema. È quindi possibile rimuovere la chiamata a *app.defineStyle()* prima o dopo aver chiamato il metodo *updateTheme*. Se si desidera impedire l'aggiornamento degli schemi è possibile chiamare *updateTheme({})* usando un oggetto vuoto come parametro.

“ Il componente Mermaid è stato aggiornato alla versione 8.10.1.

“ Il componente interno utilizzato dall’HTMLEditor è stato aggiornato all'ultima versione così come i plugin base64 ed emoji.

“ Il componente Swiper è stato aggiornato alla versione 6.7.0 ed è stata aggiunta la proprietà *breakpoints* che permette di cambiare la configurazione del componente in base alla dimensione dello schermo del device.

“ È stata aggiunta la proprietà *indeterminate* alla libreria *IonCheckbox*. Se abilitata il checkbox mostra il valore Null come indeterminato.

“ Al metodo *file.getPublicUrl* è stato aggiunto un parametro di opzioni per poter specificare se restituire un url relativo o assoluto.

“ È ora disponibile la proprietà *material* dei grafici realizzati con *GoogleCharts* che permette di mostrare la versione in stile material per i grafici che la supportano.

“ Nel metodo *app.popup*, è stata aggiunta la proprietà booleana *html* alle opzioni del tipo "toast" e agli item del tipo "menu" per poter inserire codice HTML nel popup.

“ È stato aggiunto l'evento *OnDatesRender* della libreria *Calendar* che viene notificato quando il calendario mostra un nuovo set di date, per esempio quando si premono i bottoni presenti nell'header del calendario o si usano le funzioni di navigazione.

Server

“ È stata resa più veloce la modifica delle configurazioni dei server tramite la Console. In particolare se erano presenti pacchetti personalizzati e si modificava un qualunque parametro del server il sistema riverificava i pacchetti personalizzati. Tale operazione non viene più effettuata se la lista dei pacchetti personalizzati non è cambiata quando vengono salvate le modifiche ai parametri del server tramite la console.

❝ Sono stati aggiornati tutti i pacchetti node utilizzati dal framework.

Team Works

❝ È stata migliorata la gestione delle risorse di tipo testuale (file caricati tramite l'IDE e modificati tramite il Code Editor). In particolare ora vengono segnalati conflitti se il progetto master ed il progetto forked modificano la stessa risorsa.

❝ In alcuni casi particolari la merge di una PR o l'esecuzione di una FETCH potevano generare righe di codice incomplete. Il problema si manifestava in questo caso:
- sul master si modificava l'espressione di inizializzazione di una variabile (es: da "var i = 0" a "var i = 1");
- sul progetto fork si inseriva nel metodo l'istruzione "console.log(i)".
- il progetto fork inviava una PR al master
- il master mergiava la PR

In questo caso l'istruzione "console.log(i)", dopo la merge, risultava "console.log()" poiché il riferimento alla variabile i non era più corretto (era cambiato a causa della modifica del valore iniziale eseguita sul master). Ora, il sistema cerca di gestire tale caso ricollegando il riferimento cambiato alla nuova variabile i.

❝ Ora, quando ci si muove tra i commit durante la visualizzazione delle differenze, l'IDE segnala l'inizio e la fine dell'operazione. Nelle versioni precedenti l'IDE non segnalava nulla pertanto non era chiaro se l'operazione di calcolo delle differenze del commit era in corso.

Correzioni della versione 21.5

IDE

❝ In alcuni casi il Code Editor rimuoveva o non inseriva spazi se si utilizzavano operatori di tipo "bitwise". Il malfunzionamento si manifestava nei casi seguenti:

```
a = 10 & 8;   // gli spazi prima e dopo il simbolo '&' venivano rimossi al salvataggio
c &= 4;       // gli spazi prima e dopo il simbolo '&=' venivano rimossi al salvataggio
c |= 3;       // ogni volta che si salvava veniva inserito un nuovo spazio dopo il simbolo '|'
a = c ^ b;    // gli spazi prima e dopo il simbolo '^' venivano rimossi al salvataggio
a ^= b;       // gli spazi prima e dopo il simbolo '^=' venivano rimossi al salvataggio
```

Malfunzionamento relativo alla richiesta di assistenza 000246-2021.

❝ In alcuni casi molto particolari le applicazioni installate dentro InstaLauncher non si comportavano nello stesso modo in cui si comportavano quando avviate nell'IDE o lanciate da un server di produzione. Il difetto era molto raro ed era dovuto ad un modulo Node che tentava di "adattare" il codice dell'applicazione affinché potesse funzionare anche su dispositivi datati. Ora tale "adattamento" è stato rimosso ed il codice delle applicazioni è lo

stesso che viene eseguito quando l'applicazione viene avviata dall'IDE o lanciata da un server di produzione.

“ L'operazione di copia/incolla di elementi multipli nell'albero creava nuovi oggetti in ordine opposto a quelli di origine.

“ La creazione di documenti direttamente dalla videata del DataModel non creava tutti i documenti selezionati.

“ Se si apriva il grafico del diagramma di un datamodel con una foreign key non correttamente definita si otteneva un errore javascript.

“ Se si apriva, con la versione 21.0 di Instant Developer Cloud, un progetto contenente un branch creato in versioni precedenti alla versione 19.5 si otteneva un errore.

“ Se si copiava una proprietà da una classe ad un'altra con una collection senza contentType specificato si otteneva un errore.

“ Se si implementava un evento in una classe estesa e l'evento era definito come asincrono nella classe base, la chiamata all'evento della classe base, inserita automaticamente dall'IDE, non era asincrona (quindi non iniziava con "yield"). Il malfunzionamento si manifestava se l'evento definito nella classe base non era parte delle librerie standard di Instant Developer Cloud.

“ Se si specificava un valore iniziale di una proprietà senza tipo si otteneva un errore quando si avviava l'applicazione. Malfunzionamento relativo alla richiesta di assistenza 00072-2021.

“ L'aggiornamento dello schema falliva se veniva specificato un default per i campi date, datetime e time.

“ Non era possibile creare istruzioni di tipo "insert into" tramite il Code Editor se le tabelle possedevano uno schema. Malfunzionamento relativo alla richiesta di assistenza 000654-2021.

APP

“ Effettuando il refresh di un'app offline l'app si chiudeva anziché riavviarsi.

“ Il file explorer che si apre in seguito al click sulla dropzone non filtrava subito i file in base al valore della proprietà acceptedFiles. I file venivano filtrati solo dopo la scelta di un primo file.

“ Se veniva implementato l'evento *OnInput* dell’oggetto ColorPicker era impossibile selezionare un colore.

“ Se da un'applicazione installata su iOS si usava il metodo *app.open* per aprire un'applicazione installata su un server, la proprietà *app.device.isPWA* di quest'ultima applicazione veniva erroneamente valorizzata a true, causando un comportamento anomalo.

“ Se si impostava la proprietà *btns* dell’oggetto HTMLEditor da codice gli eventi non venivano più notificati.

“ Se si impostava la chiave di *GoogleMaps* come parametro di server essa non veniva applicata. Nelle versioni precedenti è sufficiente scrivere nell'evento onStart dell’applicazione:

```
app.theme.gmapKey = "key=" + this.getParameter("gmapKey").
```

“ Se si utilizzava SQLServer non era possibile eseguire più query contemporaneamente sulla stessa connessione/transazione.

“ È stata migliorata la gestione dell'aggiornamento del componente swiper al resize dell'applicazione.

Document orientation

“ Se si usava il metodo *datamap.filterBy* su una datamap DO le query di caricamento successive della classe DO riportavano erroneamente il filtro impostato sulla datamap.

“ Se si eseguiva il metodo *load o reload* di una datamap DO la proprietà loaded rimaneva false. Il malfunzionamento si manifestava solo se la datamap aveva almeno una proprietà unbound (non collegata a campi del datamodel) oppure se si impostava una collection ad codice subito dopo aver eseguito il metodo *load o reload*.

DataMap

“ I filtri QBE su campi datetime (ad esempio "now") non teneva conto del timezone della sessione utente nelle applicazioni online.

“ Il metodo *find* su una datamap senza primary key restituiva sempre una sola riga.

“ Se si caricava una datamap utilizzando il metodo *add*, si chiamava il metodo *clear* e si ricaricava la datamap non veniva più notificato l’evento *onNextPage* se era attiva la paginazione delle righe.

“ Se si impostava un filtro invalido su un campo di tipo datetime di una datamap il caricamento della stessa non terminava.

“ Se si modificava una riga di una datamap figlia di una riga appena modificata gli elementi visuali potevano essere visualizzati correttamente.

Ionic

“ In casi particolari i messaggi di errore rimanevano visibili per i controlli di tipo ION-SELECT e ION-AUTOCOMPLETE.

“ Se un oggetto di tipo *HTMLEditor* veniva inserito in un *IonItem* non venivano notificati gli eventi.

“ Per il componente ION-AUTOCOMPLETE sono stati corretti le seguenti anomalie:
- se si cliccava su un altro elemento della videata e la lista era aperta non veniva chiusa;
- se si cliccava su un’altra autocomplete quando la lista era aperta il cursore veniva riportato nella prima autocomplete;
- se il componente aveva il flag *noFilter* attivo, su smartphone non si apriva la tastiera;
- se il flag *allowNull* era attivo, aprendo la combo senza filtri, la riga vuota appariva sempre.

“ Se si scrollava una lista con un campo fuocato questa veniva riportata all'oggetto attivo dopo qualche secondo.

Server

“ L'aggiornamento automatico dei certificati let's encrypt non funzionava correttamente. In particolare il certificato su disco veniva aggiornato ma il sistema conservava in memoria il vecchio certificato continuando ad utilizzarlo anche se scaduto.

“ Se si apriva l'URL di un server di produzione, era stata specificata un'applicazione predefinita e l'url di avvio conteneva una query string questa si perdeva durante la redirect all'applicazione predefinita. Per esempio se si apriva un browser e si specificava l'url *https://server/?command=test* il browser veniva reindirizzato all'url *https://server/defaultApp* ma la query string veniva persa.

“ Se si caricava un certificato custom il sistema non rispondeva più se il bundle del certificato custom era vuoto.

“ Se si incorporava un'applicazione web prodotta con Instant Developer Cloud all'interno di un IFRAME si potevano ottenere comportamenti non previsti su Safari. In particolare se si apriva la pagina contenente l'IFRAME, si cambiava pagina e si premeva il tasto BACK del browser, Safari mostrava un IFRAME vuoto che, dopo qualche secondo, veniva reindirizzato a www.instantdeveloper.com.

“ Se, tramite la console, si creava un parametro o si modificava il valore di un parametro di un server di produzione le sessioni delle applicazioni già in esecuzione sul server non venivano informate. Il malfunzionamento non si manifestava se il parametro era definito a livello di applicazione installata sul server.

Sync

“ Ora le variazioni vengono cancellate dall'app client solo dopo che esse siano state gestite con successo lato server. In precedenza venivano cancellate subito dopo essere state ricevute dal server senza attendere l'esito dell'elaborazione.

“ Nell'evento *app.sync.DO.onError*, notificato in caso di errore durante la gestione di una variazione, è stata aggiunta la proprietà *variationUid* tra le opzioni per sapere l'uid della variazione che ha generato errore lato server. È stato aggiunto anche il metodo *app.sync.DO.deleteStoredMessage(uid)* per permettere di cancellare la variazione lato client qualora necessario.

“ Se arrivava una variazione di un documento non presente nell'applicazione offline non veniva notificato l'evento *onSync* con parametri *full=false* e *status=completed*.

“ Se in un'app offline si ricevevano tante variazioni mentre se ne stavano generando altrettante lato client le variazioni ricevute dal server venivano processate solo dopo aver terminato di generare le variazioni lato client.

“ Si potevano ottenere errori durante la sincronizzazione qualora il datastore fosse di tipo MySQL con il parametro lower_case_table_names (attivo di default su linux).

“ Se si verificava un'eccezione durante la sincronizzazione verso un'app Foundation o durante una query remota e l'app era compilata senza debug si otteneva un errore.

WebAPI

“ Se caricava una risorsa di una classe avente come chiave primaria un UUID(36) si verificava un'eccezione.

Team Works

“ In un caso molto particolare se si eseguiva il comando REVERT di uno o più commit sul master si potevano ottenere errori quando veniva eseguita la FETCH su progetto fork. Malfunzionamento relativo alla richiesta di assistenza 001208-2021.

“ In un caso particolare le modifiche ad una risorsa server (file caricato all'interno del progetto e modificato tramite Code Editor) non venivano mostrate correttamente. Malfunzionamento relativo alla richiesta di assistenza 000231-202.

“ Se si creava un nuovo progetto e, senza fare modifiche, si creava un nuovo branch si potevano ottenere errori quando il branch veniva mergiato nel branch master. Il difetto si manifestava solo se il branch veniva creato dopo la prima apertura del progetto, subito dopo la creazione. Se il branch veniva creato dopo un qualunque commit eseguito sul branch master il difetto non si manfiestava.

Modifiche non documentate

Sono state infine apportate ulteriori **150** miglioramenti e correzioni che non vengono inserite in questo documento perché non richiedono modifiche all’utilizzo normale del sistema.