

# .Net Focus on Xamarin

## Usare in modo efficiente gli Azure Azure Mobile App in Xamarin



Giampaolo TUCCI  
[www.trilogik.it](http://www.trilogik.it)

GitHub: <https://github.com/gptucci>

Blog: <https://www.informaticapressapochista.com/it/>

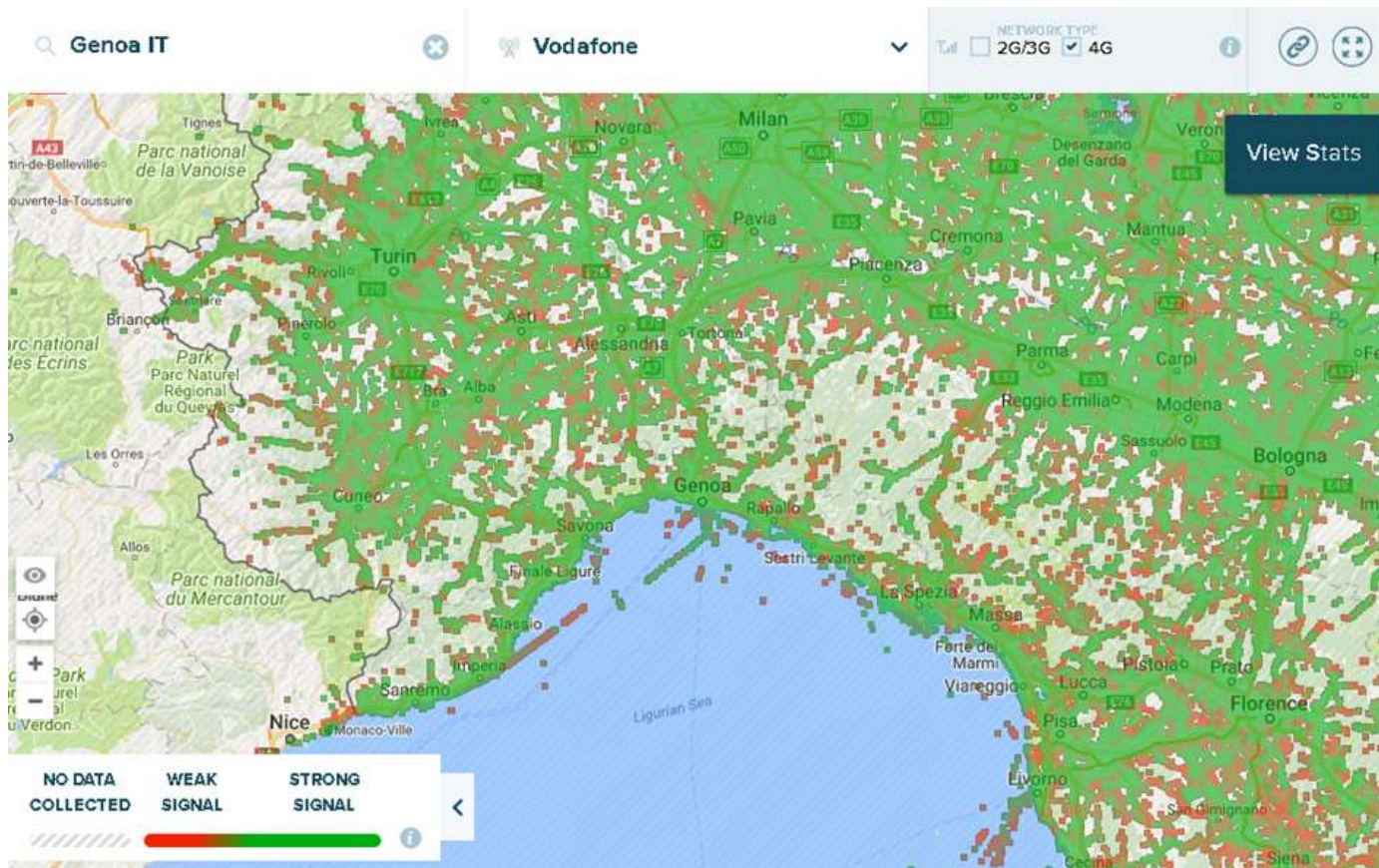
Cosa vedremo in questo intervento ??

Utilizzo dei servizi Azure Mobile App Service per creare una App dotata delle seguenti funzionalità.

- Sincronizzazione con base dati remota
- Possibilità di modificare i dati in offline

Il livello dell'intervento è Entry-Level (semplice introduzione sull'argomento).

# Perchè offline ??



Se una app enterprise-grade ha la necessità continua di interagire con un set di dati conservati remotamente (es.: Sql Server) è impensabile lavorare esclusivamente in modalità connessa.

## Problemi

- Copertura rete mobile non totale
- Situazioni geografiche sfavorevoli (palazzi, muri, etc)
- Contratto SIM che può limitare il traffico una volta superate determinate soglie

## Perchè usare Azure Mobile App Service + SDK ?

- Tutto quello che è possibile implementare con gli Azure Mobile App Service + SDK si può ottenere anche utilizzando le usuali Web API interrogate lato client con una implementazione di HttpClient.
- La differenza è che usando questo servizio e il relativo SDK lo sviluppo e il deploy della soluzione è più veloce ed efficace !



# Perchè usare Azure Mobile App Service + SDK ?

Azure Mobile App Service permette di implementare un backend ad alta resilienza e basso costo di setup e gestione.

Inoltre il servizio dispone di un SDK dedicato, *Azure Mobile SDK*, che facilita l'utilizzo dei suoi servizi.

L'accoppiata dei due permette di implementare, in modo facile, una app con le seguenti caratteristiche.

- Accesso ai dati via REST/Json in modo trasparente
- Offline sync (sincronia di una database Sqlite locale con un database Sql Server, o sezionamento di questo)
- Authentication via OAuth2 e l'ausilio di diversi IdP (Facebook, Google, Twitter, MSA Microsoft Account Identity)
- Push notification
- .....

## Linkografia per partire !

### Come creare e configurare il servizio Azure + Documentazione

<https://docs.microsoft.com/en-us/azure/app-service-mobile/app-service-mobile-value-prop>

### Dove scaricare i progetti di esempio

<https://github.com/Azure/azure-mobile-apps-quickstarts>

# Struttura tabella ToDo

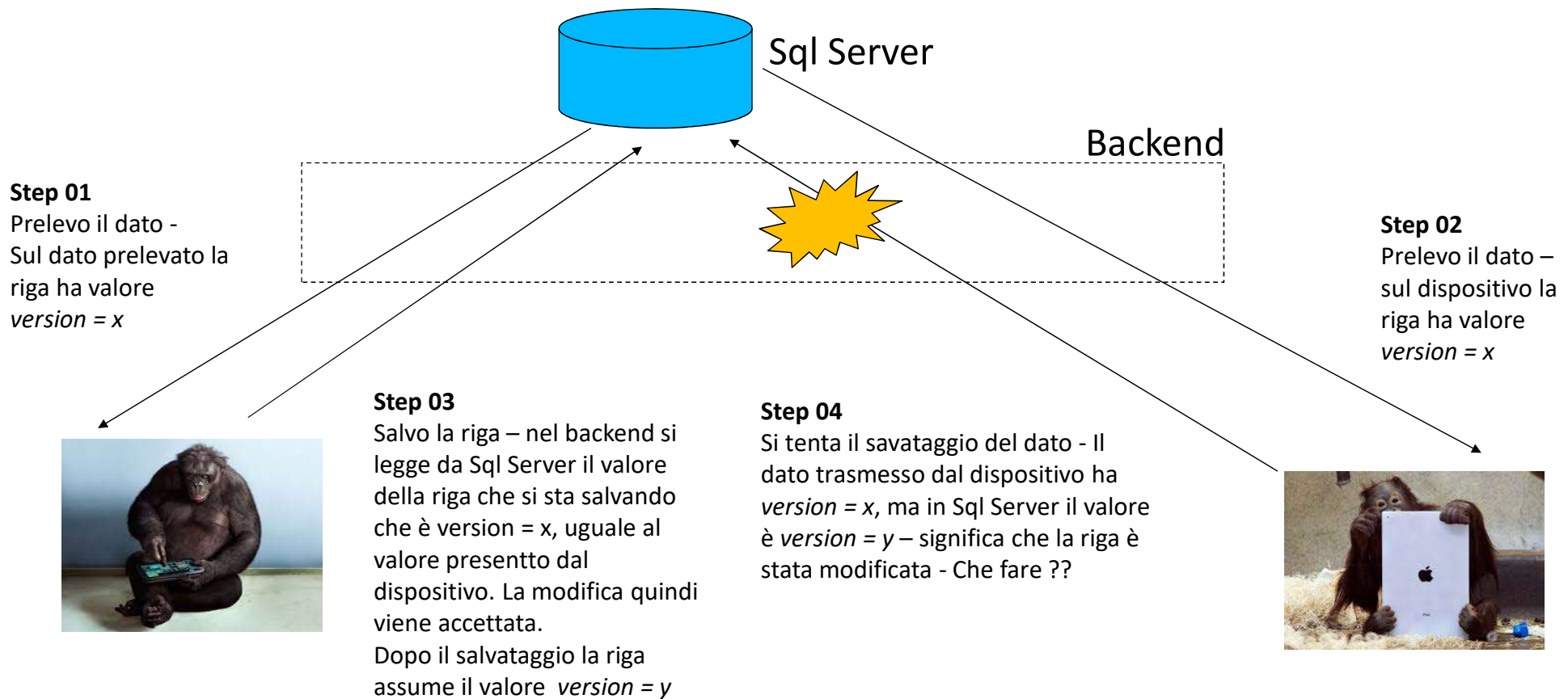
Nome Campo	Note
<u>Id</u>	Chiave Guid Chiave della tabella
Version	Timestamp, cioè revisione riga (Azure) Per gestione concorrenza
CreatedAt	Data creazione riga (Azure) Per ottimizzare performance accesso alla tabella (partizionamento orizzontale contenuti)
UpdatedAt	Ultima data modifica della riga (Azure) Ottimizzazione performance sync offline
Deleted	Indica se il record è stato cancellato Utilizzato nell'offline sync
Text	Testo ToDo
Complete	Per marcare se l'attività è completata

Perchè non possiamo essere amici ?  
Gestione della concorrenza





# Gestione della concorrenza (Optimistic Concurrency)



# ...and the winner is..... la gestione concorrenza lato client



La gestione della concorrenza può essere facilmente implementata lato server, ma spesso è necessario presentare all'utilizzatore la scelta di quale versione debba vincere, cioè quale sia la modifica vincente.

L'utilizzo dell'SDK Mobile App Service permette di ottenere, lato client, la versione del server che provoca l'incompatibilità tramite l'utilizzo di un errore *412-Precondition Failed*.

Lato client è possibile ad entrambe le versioni (quella in Sql Server e quella in salvataggio) ed è possibile decidere quale debba vincere.

# ..l'annoso problema dei record cancellati..

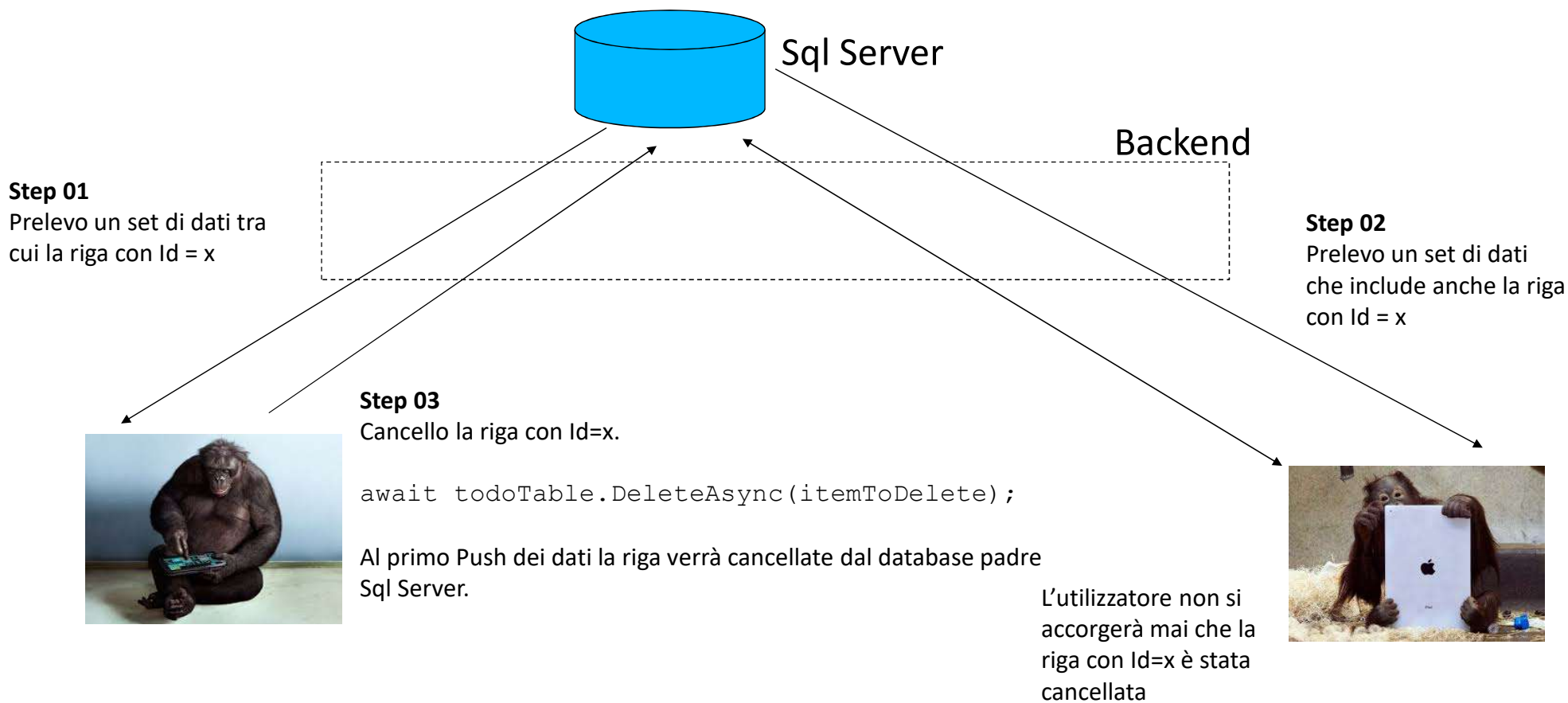
**SORRY THIS  
ITEM IS NO  
LONGER  
AVAILABLE**

Ovvero: Come è possibile cancellare un item in una serie arbitraria di dispositivi se tutti questi operano in offline sync ??

Infatti se un record viene cancellato per mano di un client questo non sarà più presente lato backend. Gli altri dispositivi come fanno a sapere che è stato cancellato ?

Semplice: Soft Delete !!

# Senza soft-delete



Grazie !