



# Delphi Summit

June 2024 | Amsterdam

## Synchronisez vos bases de données

Patrick PREMARTIN

<https://gravatar.com/patrickpremartinfr>

<https://developpeur-pascal.fr>

## The *Global* **Delphi** Summit

*Organized by*





# Delphi Summit

June 2024 | Amsterdam

# BEGIN



# INTERFACE

- Que voulons-nous résoudre ?
- Quelles solutions existent actuellement ?
- Pourquoi ne pas les utiliser ?
- La g n se
- Ma solution
- La version open source
- Exemples d'utilisations



# IMPLEMENTATION

Nous vivons dans un monde connecté où les employés de bureau et du tertiaire en général ne veulent plus travailler au bureau.

Les utilisateurs finaux de nos logiciels, surtout les particuliers, sont nomades et veulent accéder à leurs données sur différents appareils de n'importe où.

Delphi permet de faire des logiciels desktop, des applications mobiles et des sites Internet mais ça ne résoud pas tout.



# IMPLEMENTATION

Quand on offre une liberté de mouvement à nos utilisateurs, on les oblige à rester connectés à Internet. En pratique ce n'est pas possible partout sur la planète ni au dessus. Et ça peut coûter très cher selon où on se trouve.

Les solutions techniques sont généralement la réplication de données, le travail en client/serveur ou un mélange des deux.

Je mets les solutions web dans la case client/serveur.



# IMPLEMENTATION

Certaines bases de données intègrent des systèmes de réplication par synchronisation des modifications sous forme de requêtes (en cache) ou de données (comme le fait InterBase).

L'inconvénient de ces solutions se trouve souvent au niveau des clés primaires : si plusieurs utilisateurs font des ajouts dans la même table et que ces tables sont liées à d'autres ça peut coïncider.



# IMPLEMENTATION

Les logiciels fonctionnant avec un serveur de données dépendent d'une connexion à celui-ci pour travailler.

Sur un réseau local d'entreprise ça fonctionne bien dès qu'on en sort ça peut coincer.

Une absence d'Internet interdit l'utilisation des logiciels à distance que l'on ait besoin d'une connexion directe au serveur de base de données ou que l'on passe par un programme intermédiaire comme par exemple DataSnap.



# IMPLEMENTATION

Face à ces problèmes des entreprises ont développé leurs solutions de synchronisation pour un ou plusieurs projets.

Ces solutions sont rarement publiques, partageables ou utilisables chez d'autres.



**Delphi Summit**

June 2024 | Amsterdam

[delphisummit.com](https://delphisummit.com)

13 & 14 June 2024

8



# IMPLEMENTATION

En 2017 j'ai travaillé avec Christian Hubert-Hugoud sur la refonte en Delphi de son logiciel Intelli7.



**Delphi Summit**

June 2024 | Amsterdam

[delphisummit.com](https://delphisummit.com)

13 & 14 June 2024

9

# IMPLEMENTATION

Parmi les besoins et contraintes il y avait :

- une base de données par client
- un ou plusieurs utilisateurs par base de données
- accès aux mêmes fonctionnalités depuis un ordinateur ou une tablette
- accès depuis n'importe où, avec ou sans connexion à Internet
- possibilité de modifier les bases clients depuis le serveur ou d'autres logiciels que Intelli7



# IMPLEMENTATION

Nous avons imaginé une solution mélangeant réplication et client-serveur sous forme d'API.

Le logiciel Intelli7 a été développé en Delphi sous FireMonkey et utilisait quelques composants de TMS Software pour l'affichage de graphiques et leur export en PDF.

Le logiciel utilise une base de données SQLite locale qui permet de stocker un système de fichier, des données NoSQL en JSON et des paramètres sous forme de paires clé / valeur.



# IMPLEMENTATION

Chaque instance du logiciel a une copie de la base de données et travaille dessus en local sans se préoccuper des autres.

Un processus effectue régulièrement la synchronisation des données lorsqu'une connexion Internet est disponible.

Le fait que la base de données soit synchronisée est totalement transparent pour le logiciel et les utilisateurs.





# IMPLEMENTATION

Côté serveur nous avons travaillé en PHP avec des bases de données SQLite. C'était plus simple à gérer qu'une multitude de bases MySQL afin de s'assurer qu'aucun client n'aurait jamais accès aux données des autres.

Une API a été créée pour la synchronisation.

Elle ne dépend ni des langages de développement ni des moteurs de bases de données utilisés.



# IMPLEMENTATION

Une démonstration de Intelli7 orientée développeurs Delphi est disponible sur <https://vimeo.com/918818000/9c4ab12439>

Un “case study” est disponible avec cette présentation sur <https://github.com/DeveloppeurPascal/DelphiSummit2024-Synchronize-Your-Databases>

Pour tout savoir sur le logiciel et la méthode Intelli7 rendez-vous sur <https://intelli7.com>



# IMPLEMENTATION

La solution développée pour Intelli7 a évolué au fil des versions. Je m'en suis inspiré pour d'autres applications jusqu'à obtenir la version que vous allez pouvoir utiliser dans vos projets.

Le principe de base est d'impacter le moins possible les logiciels existants tout en leur fournissant une possibilité de basculer en mode multi utilisateurs / multi appareils / multi lieux.



# IMPLEMENTATION

Il est nécessaire d'avoir un serveur accessible ponctuellement par les utilisateurs :

- sur Internet pour une mobilité totale
- sur un réseau d'entreprise (en local ou par un VPN) selon vos besoins de sécurité

La réplication se base sur des champs de votre base de données. Un changement de sa structure sera nécessaire.





# IMPLEMENTATION

Pour des raisons de sécurité vous devez mettre en place toutes les mesures de protection de vos bases de données et de vos logiciels.

Si vous autorisez un usage en dehors de vos locaux, ce système se basant sur une copie des données, faites le nécessaire pour les chiffrer et verrouiller l'accès à vos logiciels (pseudo, mot de passe, 2FA) !



# IMPLEMENTATION

Limitez au maximum l'impact que pourra avoir le vol d'ordinateurs, de smartphones ou de tablettes sur l'intégrité et la confidentialité de vos données.

Ceci étant tout aussi valable pour les appareils qui restent dans les locaux de vos utilisateurs, serveurs inclus.



# IMPLEMENTATION

J'ai appelé ma solution de synchronisation Table Data Sync.  
C'est une simple API client/serveur.

Une implémentation Delphi est disponible sur  
<https://github.com/DeveloppeurPascal/TableDataSync4Delphi>

D'autres implémentations pour d'autres langages seront mises  
en ligne sur  
<https://github.com/DeveloppeurPascal?tab=repositories&q=tabledatasync&type=&language=&sort=>



# IMPLEMENTATION

La synchronisation se fait en 4 étapes :

- ouverture d'une session
- réception des modifications enregistrées sur le serveur
- envoi des modifications locales vers le serveur
- fermeture de la session





# IMPLEMENTATION

La connexion permet de définir des codes d'accès et indiquer au serveur sur quelle base de données on désire travailler.

Ces données sont envoyées en http ou https.

A vous de les chiffrer pour un niveau de protection supplémentaire.



# IMPLEMENTATION

La réception des données du serveur se fait par rapport à un numéro de version stocké sur chaque enregistrement pour chacune des tables à synchroniser.



# IMPLEMENTATION

L'envoi des données locales vers le serveur sélectionne tous les enregistrements ayant été ajoutés, modifiés ou supprimés depuis la dernière synchronisation dans chaque table à synchroniser.



# IMPLEMENTATION

La clôture de la connexion libère la session côté serveur et permet de détecter la fin de la synchronisation côté client.





# IMPLEMENTATION

La détection des suppressions physiques se fait par l'intermédiaire d'une table annexe que le programme devra remplir lorsqu'un enregistrement est supprimé quelque part.

Les suppressions logiques sont gérées comme les autres modifications puisqu'elles sont généralement traitées sous forme d'un champ dans les tables concernées.



# IMPLEMENTATION

Les conflits de synchronisations sont à traiter côté client. La librairie propose des événements pour les gérer automatiquement ou manuellement.

En cas d'absence de traitement, il y a un écrasement des données en fonction de la date de dernière modification.



# IMPLEMENTATION

Pour que ça fonctionne vous devez ajouter des champs à vos tables à synchroniser :

- chaque identifiant doit être doublé pour avoir son numéro attribué du côté du serveur (en clé primaire comme en référence)
- un champ booléen (ou équivalent) doit permettre de définir si un enregistrement a été ajouté ou modifié depuis la dernière synchronisation



# IMPLEMENTATION

Pour que ça fonctionne vous devez ajouter des champs à vos tables à synchroniser :

- un champ entier servira de numéro de version de chaque enregistrement dans sa table
- un champ date/heure est à mettre à jour à chaque changement



# IMPLEMENTATION

Il est recommandé d'ajouter des index uniques en utilisant ces nouveaux champs pour accélérer les étapes de synchronisation et ne pas détériorer les temps de réponse de vos logiciels.

Si vous faites des suppressions physiques dans vos tables vous devrez ajouter une table destinée à en conserver l'historique et les répercuter sur les autres installations de votre logiciel.





# IMPLEMENTATION

Vous devez ensuite ajouter des déclencheurs sur vos tables ou traiter ces opérations par code (mais attention à ne pas en oublier) :

- une insertion ou une mise à jour doivent positionner le champ de synchro et changer la date de mise à jour
- une suppression physique doit ajouter la clé unique dans la table des suppressions



# IMPLEMENTATION

Dans la version actuelle la librairie Delphi ne prend en charge que les clés primaires entières ou les combinaisons de clés entières.

Chaque table à synchroniser doit être accessible avec un index unique sur lequel peut se baser la synchronisation.



# IMPLEMENTATION

La base de données côté clients doit avoir la même structure que sur le serveur. Ca implique de la mettre à jour en même temps des deux côtés ou gérer des versions différentes selon vos changements dans sa structure.

Le fonctionnement actuel n'est pas destiné à de gros volumes de modifications par table. Il serait trop consommateur en bande passante.



# IMPLEMENTATION

Le fonctionnement de Table Data Sync est plutôt adapté à des utilisateurs “nomades” travaillant sur leurs propres données et devant se synchroniser avec d’autres utilisateurs depuis un ou plusieurs appareils.

Par exemple des commerciaux, des livreurs, des personnes en télétravail, des prestataires ou assistants “virtuels” et bien entendu des particuliers utilisant un logiciel pour lequel vous fournissez l’hébergement des bases de données synchronisées.



# IMPLEMENTATION

Table Data Sync est avant tout une API fonctionnant sur le protocole HTTP/s pour passer partout.

Mon implémentation Delphi ou PHP de la librairie clientes et du serveur peuvent ne pas vous convenir. Rien ne vous empêche de les modifier et les adapter à vos besoins tout en conservant “l’autre côté”.





# IMPLEMENTATION

Rien ne vous empêche de ne transférer qu'une partie des données sur chaque poste en fonction des identifiants utilisés lors de l'ouverture de session.

Vous pouvez aussi avoir plusieurs logiciels travaillant avec le même serveur de synchronisation mais n'utilisant qu'une partie des données (ce qui est d'ailleurs recommandé pour éviter les duplications d'informations inutiles qui augmenteraient le risque de fuites).



# IMPLEMENTATION

Il est possible d'avoir un serveur indépendant de la structure de la base de données si vous n'avez pas besoin de traiter les données côté serveur.



# IMPLEMENTATION

La version de Table Data Sync disponible sur <https://github.com/DeveloppeurPascal/TableDataSync4Delphi> est distribuée sous licence AGPL v3.

Vous pouvez l'utiliser sur vos projets personnels comme professionnels.

Si la licence AGPL ne vous convient pas je vends également des "licences développeurs", "licences sites" et "licences entreprises" classiques.



# IMPLEMENTATION

S'agissant de manipulation de données sans aucune visibilité pour l'utilisateur final, faites des tests avant de mettre vos projets en production.

Faites des backups des données de vos utilisateurs dans vos logiciels et sur les serveurs afin de pouvoir revenir en arrière si vous détectez une anomalie un jour.

L'utilisation de mes implémentations de Table Data Sync se fait sous votre contrôle. Aucune garantie n'est fournie.



# IMPLEMENTATION

Je me sers de cette version de la librairie dans de petits projets mais n'ai pas eu l'occasion de la déployer largement. tout fonctionne bien pour mon usage.

Il y a sans doute des cas problématiques que je n'ai pas encore détectés. Restez méfiants et remontez moi toute anomalie ou bizarrerie.





# IMPLEMENTATION

Entre deux streams Twitch ou développements open source “pour la communauté” (et pour moi) je fais aussi de la prestation de service.

Si vous avez besoin de conseils pour vérifier la pertinence de ma solution ou pour la mettre en oeuvre dans vos projets n'hésitez pas à m'en parler. Je verrai sous quelle forme je peux ou ne peux pas vous y aider.



# IMPLEMENTATION

J'ai développé plusieurs exemples simples pour illustrer cette présentation. Vous les trouverez sur ce dépôt GitHub :  
<https://github.com/DeveloppeurPascal/DelphiSummit2024-Synchronize-Your-Databases>

Un serveur WebBroker prend en charge les bases de données synchronisées des projets clients. Le serveur tourne sur Linux.

Les programmes clients ont été testés sur Windows, macOS, iOS et Android.



# FINALIZATION

- Table Data Sync peut vous simplifier la vie pour simplifier celle de vos utilisateurs.
- La librairie pour Delphi est disponible sur <https://github.com/DeveloppeurPascal/TableDataSync4Delphi>
- La librairie pour PHP est disponible sur <https://github.com/DeveloppeurPascal/TableDataSync4PHP>



# FINALIZATION

- Pour me contacter passez par l'un de mes sites ou les liens présents sur <https://gravatar.com/patrickpremartinfr>
- Je suis aussi disponible entre deux sessions durant la conférence.



**Delphi Summit**

June 2024 | Amsterdam

delphisummit.com

13 & 14 June 2024

43





# Delphi Summit

June 2024 | Amsterdam

END.