# FOXWEB

- config DNS hébergeur (o2switch) : router sous-domaine vers IP publique

- installer VirtualBox sur PC/serveur

- ajouter le répertoire des exécutables VirtualBox au path

- copier VM sur PC/serveur

- configurer les partages de dossiers entre la VM et le PC/serveur

- fixer IP de la VM contenant le serveur web (dhcp de la box)

- NAT/PAT diriger port https 443 vers IP de la VM

- installer driver postgresql 32 bits sur le PC hébergeant la VM, via installeur Stack Builder

- configurer ODBC

- modifier fichiers de la VM : /scripts/db\_backup.sh (changer backupdir), /etc/opt/dj/settings.py (IP ds ALLOWED\_HOSTS),

    /etc/nginx/sites-available/dj.monkey-soft.fr (changer IP)

- installer autoit

- copier script autoit startVM ds répertoire démarrage de Windows (Win+R shell:startup) + sur bureau

- copier script autoit stopVM sur bureau + à la fermeture via gpedit.msc

- modifier IP dans fonctions et procédures VBA ADO de Foxweb

A = FOX (application Fox)

B = FOXWEB (foxweb.mdb)

C = WEB (base postgresql dj + site Django)

base access : foxweb.mbd

base postgresql : dj

# 05/09/2022

Màj B => v2-8

- Modification InitPostgresqlConnection (module global) : l’enregistrement du pb ds table TW\_Histo s’effectue désormais avant l’appel à on error resume next

- les appels à on error resume next ne s’effectuent désormais plus que dans P\_exit, sauf si instruction ADOCnx.RollbackTrans qui peut planter ; en tout état de cause la fonction AddToHisto est toujours appelée AVANT on error resume next ; ainsi on récupère les infos de l’objet Err avant qu’il ne soit réinitialisé par l’instruction on error resume next.

# 31/08/2022

Màj B => v2.7

* Ajout appel à AddToHisto (historique des erreurs) dans fonctions maj\_postgres et maj\_access

# 22/08/2022

màj B => v2.6

* remplacement fonction CopyTable1ToTable2 par Copy\_public\_foxweb\_client\_To\_TW\_client() et Copy\_public\_foxweb\_contrat\_To\_TW\_contrat() :

au lieu de supprimer les tables Access TW\_client et TW\_contrat et de les recréer, on supprime les enregistrements et on utilise une requête de type ajout (insert into destination(champ1, champ2,...) select champA, champB, ... from source ;)

Le résultat (en termes de performances) est tout aussi efficace et on peut conserver les noms des champs des tables access différents de ceux de postgres.

Copy\_public\_foxweb\_client\_To\_TW\_client() et Copy\_public\_foxweb\_contrat\_To\_TW\_contrat() sont appelées par insert\_clients\_access et insert\_contrats\_access

màj B => v2.5

* réécriture fonctions insert\_clients\_access et insert\_contrats\_access qui ne passent plus par ADO mais par les tables liées et la fonction CopyTable1ToTable2 basée sur une requête action (select ...into...from) exécutée via docmd.runsql ;  
  la requête action écrase (silencieusement) les tables TW\_client et TW\_contrat déjà existantes (l’appel aux procédures de suppression des enregistrements dans ces tables n’est donc plus nécessaire).  
  le travail n’est plus séquentiel (enregistrement par enregistrement en parcourant via ado chaque ligne des tables postgres pour les insérer dans les tables access) mais global (de niveau table cette fois), d’où le gain de performance astronomique.

# 21/07/2022

màj C =>

* affichage date en français en resultat de recherche par stand ( form home2.html)

# 20/07/22

màj C =>

* search by stand : view search\_stand ajoutée et formulaire home2 modifiée
* vidage resultats du formulaire recherches au focus qd changement type de recherche - passage search by client à search by stand et vice versa
* correction valeur None ds champ commentaires

exemple requête postgres permettant de récupérer contrat le plus actuel (sera géré via l’orm) :Une image contenant texte

Description générée automatiquement

# 19/7/22

maj B v2.3 => optimisations suite tests

maj C => modif champ recherche par code slt; ajout info n° stand ds lignes résultats

# 17 /7/22

Màj C : version incluant gestion utilisateurs (login + admin django), recherche multicritères (code, nom, tel, ville, code postal)  avec clé étrangères supprimées.

Views et templates mis à jour

# 08/7/22

màj B

# BTWEB

## 1.0 du 8/11/22

Création de BTWeb à partir de FoxWeb 5/9/22 de jour de Pat

Ajout des 3 tbl pour la gestion des BC

T\_bl

T\_pieces\_bl

T\_produits\_bl

Une image contenant table

Description générée automatiquement