24/05/2017

Virginie PORTEMER, Benoit Guillaumin, Yannick Cappelle, Rémi Blavec

Isagri

formation GTM

Documentation technique

Application Job Overview

Sommaire

[I. Paramétrage de la chaine de connexion 2](#_Toc483410380)

[1. Restauration de la base de données 2](#_Toc483410381)

[2. Récupération de la chaine de connexion. 3](#_Toc483410382)

[II. Architecture globale 4](#_Toc483410383)

[III. Application 4](#_Toc483410384)

[IV. Gestion des tâches de production 5](#_Toc483410385)

[V. Gestion des tâches annexes 5](#_Toc483410386)

[VI. Gestion des versions 5](#_Toc483410387)

[VII. Tests Unitaires 6](#_Toc483410388)

[VIII. TODO 6](#_Toc483410389)

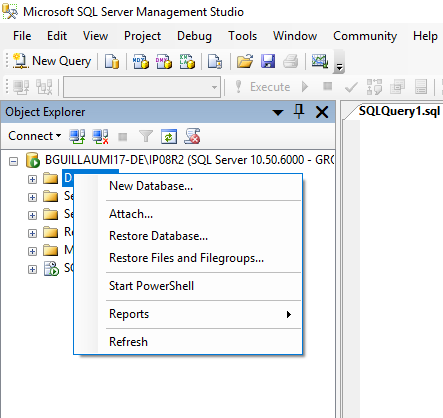
# Paramétrage de la chaine de connexion

Pour tous types d’utilisation avant de lancer l’application Job Overview, vous devez paramétrer votre chaine de connexion à votre base Job Overview.

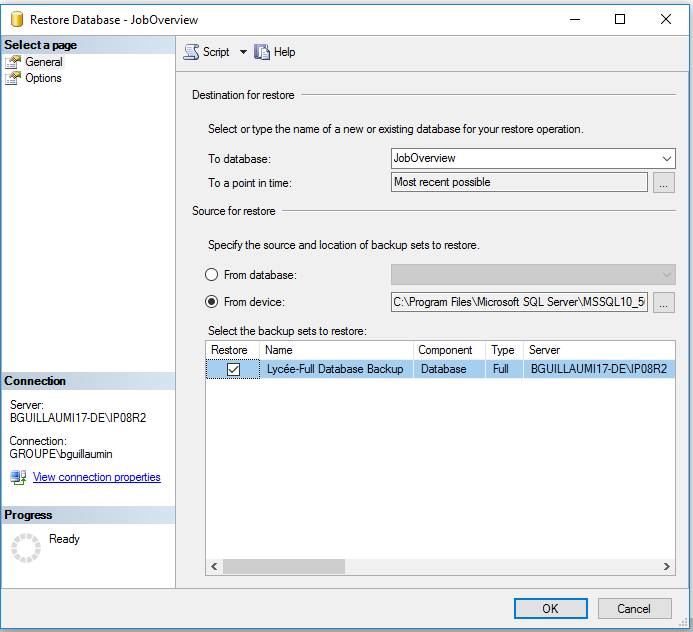
## Restauration de la base de données

Si votre base de données n’existe pas, vous devez en créer une. Pour ce faire, nous vous proposons de faire une restauration d’une base déjà existante fourni dans le livrable (.bak).

Dans SQL Server Management Studio (SSMS), cliquez sur Restore Database.



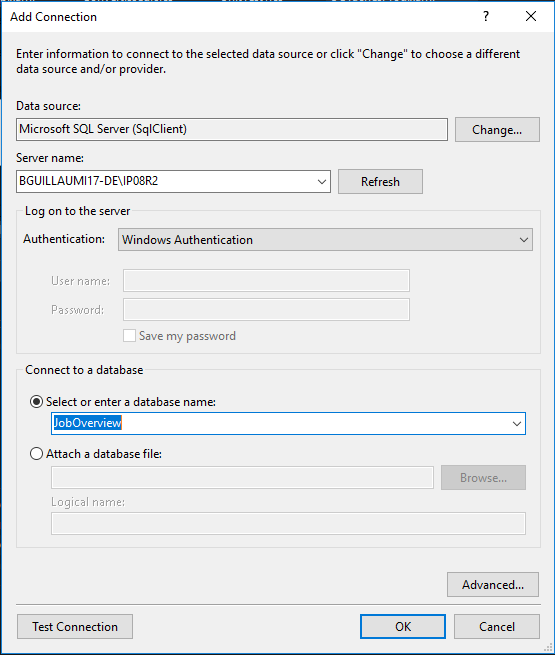
Ajouter un nom à votre base de données, par exemple « Job Overview ». Dans « from device » : mettre le chemin du fichier .bak et checker la case « Restore ».



Cliquez sur OK. Votre base de données est restaurée.

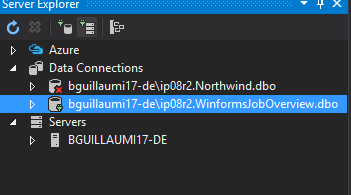
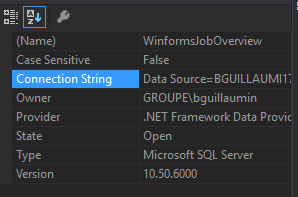
## Récupération de la chaine de connexion.

Votre chaine de connexion est accessible depuis votre logiciel Visual Studio dans la fenêtre Server Explorer, clic droit sur Data Connections, cliquez sur Add Connection.



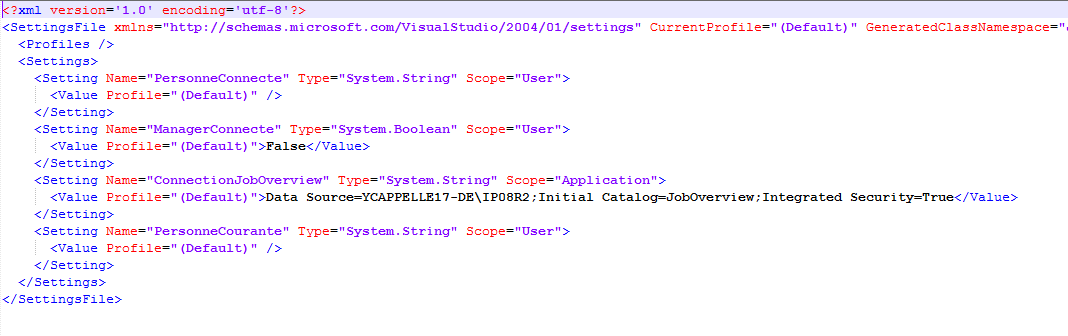
Sélectionner le nom de votre serveur (celui-ci porte généralement votre nom) ainsi que le nom de la BDD (pour ce cas-ci sélectionné Job Overview).

Récupérez ensuite la chaine de connexion par un clic droit sur la connexion puis propriété. Faire un copier de la connection string.

Dans le fichier Settings.Settings au chemin suivant : JobOverview\JobOverview\Properties

Remplacer la chaine de connexion existante par la votre.



Enregistrez le fichier. Vous pouvez lancer l’application.

# Architecture globale

L’architecture est de type MVVM. Le visuel est séparé des méthodes des DAL (Data Access Layer). Les échanges intermédiaires sont réalisés par des vues Modèles. Cela permet de scinder le visuel du code pur et est plus évolutif et réutilisable. Néanmoins certaines fonctionnalités plus complexes ont été réalisés en code behind. Le diagramme de classe joint décrit les relations entre les entités POCO. Les DAL regroupent les méthodes de requêtage.

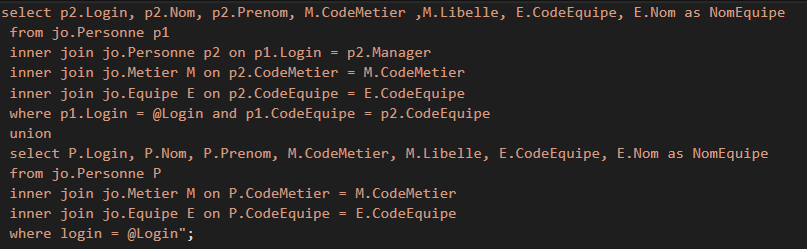
# Application

Pour cette application nous avons décidé de répartir les différentes fonctionnalités en menu. Le choix du menu a été fait plutôt qu’un tabControl car la fenêtre d’un menu n’appelle pas la base s’il est n’est pas ouverte. Contrairement à un tabControl, ceci évite le chargement de toutes les requêtes nécessaire à l’ouverture des différentes fenêtres. Par exemple si un utilisateur ouvre l’application que pour saisir des temps, seules les requêtes de cette fenêtre seront appelées.

Un visuel semblable a été appliqué dans les différentes fenêtres pour garder un ensemble cohérent et ne pas perdre l’utilisateur lorsqu’il change simultanément de fenêtre.

Chaque menu correspond à un besoin client. Certaines fonctionnalités ne sont disponibles que par le manager, comme l’ajout et la suppression des taches, ou encore l’exportation des taches de production au format XML.

Au démarrage de l’application, une requête ramène tous les employés de la base. La personne sélectionnée (dernier utilisateur de l’application) dans la fenêtre de login est mémorisé dans un paramètre utilisateur ainsi que son statut (manager ou non). Ainsi nous avons rendu visible ou non les fonctionnalités en fonction de l’utilisateur. De plus, une requête implémente une liste de personne en fonction de la personne logué. Pour un simple employé, seul ses taches seront ramenées, alors que pour un manager, on récupère toutes les taches de son équipe et y compris lui-même grâce à cette requête :



# Gestion des tâches de production

Pour cette fonctionnalité, nous utilisons une requête pour ramener les informations concernant la personne logué, une pour ramener les taches de production et d’autres simples pour alimenter les combobox (logiciel, version, activité). De simples filtres sont appliqués pour trier en fonction du logiciel, de la version, et de la personne liée à la tache (cas pour le manager) et des tâches non terminées.

Nous avons appliqué un maître détail entre la listview et l’expander qui sert de fiche récapitulative pour chaque tache.

Le bouton XML, appel une fonction de sérialisation qui permet de créer un fichier XML à partir de la liste des taches de production.

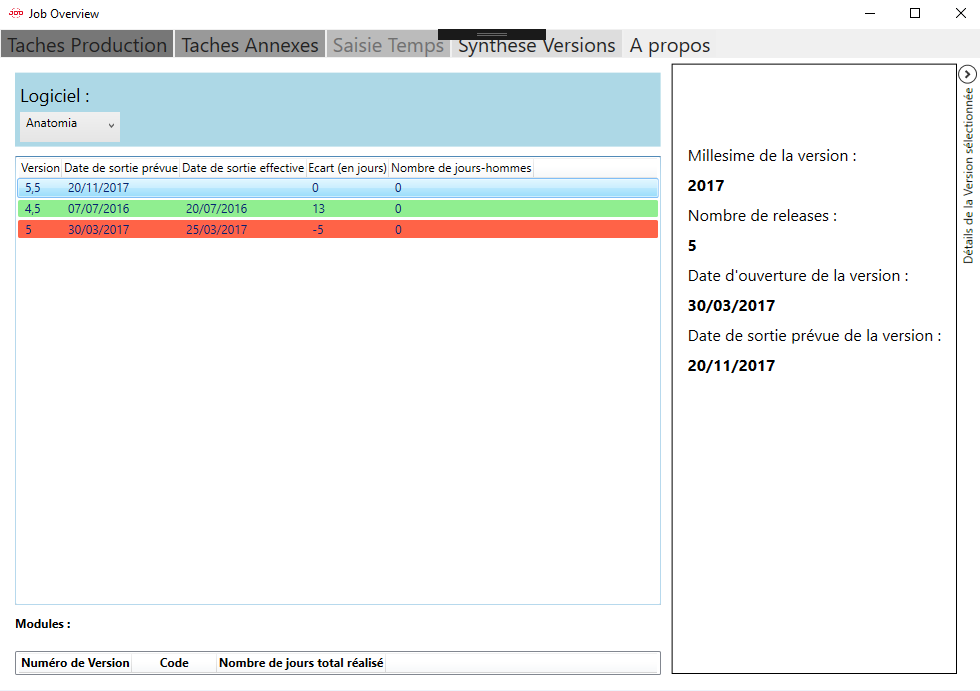
# Gestion des tâches annexes

Cette fonctionnalité de l’application s’appuie sur la gestion des taches de production. Néanmoins il est possible de supprimer des taches annexes n’ayant pas de contraintes de clé étrangère avec la classe Travail. Si on essaie de supprimer une tache lié à un temps, un message d’erreur apparait.

# Gestion des versions

Cette fonctionnalité permet de consulter la liste des versions et des modules par rapport à leur logiciel associé. Pour cela nous avons utilisé deux requêtes. Une par listview en fonction du logiciel sélectionné dans le filtre.

Un convertisseur est appliqué sur les versions pour mettre en valeur les versions sur lequel il y a du retard et sur celles où il y a de l’avance. Exemple :



# Tests Unitaires

Dans la partie test, nous avons testé les fonctionnalités d’ajout, d’enregistrement et de suppression des tâches (production / annexes). Il faut lancer tous les tests ensembles dans le test unitaire de suppression ne peut être validé si le test de création n’a pas été effectué.

# TODO

* Bloquer les saisies pour empêcher des mauvaises saisies (ex que des chiffres quand on attend un numéro)
* Faire la fenêtre Saisie Temps
* Faire en sorte que l’ajout de tâches de production fonctionne