

# TP Qt Introduction

(2,5 séances)

Yannick Kergosien

---

**Objectif du TP :** *se familiariser avec Qt Creator et de poser les bases du projet que vous réaliserez au cours des prochaines séances.*

Lien utile : <http://doc.qt.io/qt-5/classes.html>.

## 1 Préambule

- Créez un projet (application graphique Qt) sous Qt Creator.
- Lors de cette création, n’oubliez pas de créer également une QMainWindow.
- Soyez curieux, regardez les fichiers de votre projets (.pro, main.cpp, mainwindow.cpp, etc.).

## 2 Une première fenêtre : l’identification

Avant d’afficher la QMainWindow, votre application doit d’abord posséder une étape d’identification. Pour cela :

- Créez une nouvelle fenêtre de type Dialog without button.
- Ajoutez des éléments qui permettent de saisir un login et un mot de passe (QLabel, QLineEdit, QPushButton, etc.). Deux boutons devront être disponibles : Connecter et Annuler.
- N’oubliez pas l’utilisation des Layout et le paramétrage de vos fenêtres (tailles, etc.)

Les fonctionnalités de cette fenêtre doit être les suivantes :

- Si l’utilisateur appuie sur Annuler (ou ferme la fenêtre), l’application se termine (sans instancier la fenêtre principale).
- Si l’utilisateur appuie sur Connecter alors :
  - Si le login et le mot de passe sont correctes alors la fenêtre d’identification se ferme et la fenêtre principale s’ouvre.
  - Sinon une fenêtre type QMessageBox : :warning s’affiche en précisant “Login ou Mot de passe incorrecte ! ” et à sa fermeture on revient sur la fenêtre d’identification (le champ login doit rester le même mais le champ mot de passe doit être à nouveau vide).

Pour vérifier si le Login et le Mot de passe sont corrects, vous réaliserez une méthode privée qui renverra vrai ou faux. Cette méthode appellera une autre méthode externe permettant de se connecter à la base de données et de vérifier les informations (à réaliser plus tard).

## 3 La QMainWindow : les premières briques graphiques

Vous allez maintenant définir la structure de base (uniquement graphique) de votre QMainWindow. Elle sera complétée au fur et à mesure. Voici les premiers principaux éléments à réaliser (à vous de choisir les composants graphiques adéquats, une ergonomie simple, etc.) :

- Dans le menu principal (*Menu Bar*), il sera possible d’effectuer les actions :
  - Fichier → Quitter : pour quitter l’application.
  - Fichier → Ajouter → Client : permet d’ouvrir une nouvelle fenêtre de saisie d’un nouveau client. Pour l’instant, vous pouvez créer une fenêtre “vide” avec uniquement un bouton Ok et Annuler (cette fenêtre doit être “bloquante”).

- Fichier → Ajouter → Personnel : permet d’ouvrir une nouvelle fenêtre de saisie d’une nouvelle personne (banquier, assureur, etc.). Comme précédemment, vous pouvez créer une fenêtre “vide” avec uniquement un bouton Ok et Annuler (cette fenêtre doit être “bloquante”).
- ? → A propos : permet d’ouvrir une nouvelle fenêtre affichant une image (le logo Polytech) et des informations sur la version du projet et leurs créateurs (vous).
- Dans le menu barre d’outils (*Toolbars*), deux icônes différentes permettront de réaliser les mêmes actions qu’ajouter un client et du personnel.
- La barre de statuts affichera des messages liés aux dernières actions (ex : Vous venez de vous connecter, Client ajouté, Personne ajoutée, etc.).
- Enfin dans la partie centrale, deux parties devront apparaître non simultanément (utiliser Tool Box ou Tab Widget ou Stacked Widget ou etc.) :
  - Recherche de Client : permettra de rechercher un client dans la base grâce à son nom, prénom, date de rendez-vous, ou numéro d’identification. L’affichage s’effectuera grâce à une Table View. D’autres actions seront également permises, elles seront détaillées plus loin.
  - Planification : permettra d’afficher les personnes actives pour effectuer une planification des rendez-vous à un jour donné. Cette partie sera également détaillée dans les prochains TPs.

## 4 Les clients

Des clients peuvent donc être ajoutés à la base. À l’ajout d’un client, les saisies suivantes sont obligatoires :

- un nom (chaîne de caractères dont la première lettre est en majuscule, le reste en minuscule),
- un prénom (chaîne de caractères dont la première lettre est en majuscule, le reste en minuscule),
- une adresse (chaîne de caractères libre),
- une ville (chaîne de caractères dont la première lettre est en majuscule, le reste en minuscule),
- un code postal (uniquement des chiffres pour simplifier),
- le jour de rendez-vous (peut être le jour même mais pas un jour antérieur),
- une durée de rendez-vous totale estimée en minute (par défaut : 1h),
- une priorité (de 5 à 1 avec 5 comme plus prioritaire et 1 valeur par défaut),
- une liste d’identifiants de ressource qui seront modélisés comme par les personnes (banquiers, assureurs, etc.) pour simplifier (graphiquement elles seront représentées par le nom d’une personne disponible dans la base de données, l’identifiant étant l’identifiant dans la base de données).

Les saisies non obligatoires sont :

- Un champ libre pour des commentaires,
- un numéro de téléphone (uniquement des chiffres pour simplifier).

Une recherche de client sera donc possible grâce à son nom (les premiers caractères), prénom (les premiers caractères), date de rendez-vous (entre deux dates à saisir), ou un numéro d’identification (identification dans la base). Pour rechercher un client, au moins un de ces champs doit être rempli et les clients trouvés doivent correspondre à au moins un de ces critères. La liste de ces clients est à afficher dans une Table View composée d’une colonne nom, prénom, date de rendez-vous, numéro d’identification, et de la possibilité de faire deux actions (boutons, liens, etc.) : modifier la fiche client et supprimer la fiche client.

Même si l’interface avec la base de données ne sera vue qu’aux prochains TPs (apprenez à faire vos propres données pour vos tests unitaires), implémentez la partie Modèle/Vue/Contrôleur d’ajout de client, de recherche de client, de suppression de client et de modification de client. Codez proprement (MVC) et de manière efficace (ex : l’ajout et la modification de client peut être rassemblés). L’ergonomie

et la simplicité du système n'est pas à négliger (ex : le traitement des majuscules et minuscules pour les noms et les prénoms peuvent se faire automatiquement).

## 5 Le personnel

Des personnels peuvent donc être aussi ajoutés à la base. A l'ajout d'une personne, les saisies suivantes sont obligatoires :

- un nom (chaîne de caractères dont la première lettre est en majuscule, le reste en minuscule),
- un prénom (chaîne de caractères dont la première lettre est en majuscule, le reste en minuscule),
- un type parmi : Banquier type A, Banquier type B, Assureur logement, Assureur voiture, Assureur vie, Informaticien et divers,
- seulement pour le type informaticien, un login et un mot de passe devront être saisies (pour la connexion au système). Les autres types ne pourront pas se connecter au système.

Tous le personnel sera à afficher (pas de fonctionnalité recherche) dans la fenêtre principale. Cette affichage s'effectuera grâce à une Tree View à deux niveaux : les types puis les noms. A la sélection du nom, deux actions seront possibles : modifier la fiche du personnel et supprimer la fiche du personnel.

Comme précédemment, même si l'interface avec la base de données ne sera vu qu'aux prochains TPs, implémentez la partie Modèle/Vue/Contrôleur d'ajout de personnel, de suppression de personnel et de modification de personnel. Codez proprement (MVC) et de manière efficace. L'ergonomie et la simplicité du système n'est pas à négliger.