



## Module 1 Développer l'interface d'une application informatique

|                          |        |     |          |        |
|--------------------------|--------|-----|----------|--------|
| Composants accès données | Séance | S01 | Activité | A-004A |
|--------------------------|--------|-----|----------|--------|

Vous allez découvrir dans cet atelier comment séparer les différentes couches de nos composants afin de faciliter la maintenance et la réutilisabilité du code.

Ce sera aussi l'occasion de découvrir de nouveaux patrons de conception sollicités couramment dans l'industrie du logiciel dans l'objectif de produire un code DRY, robuste et performant.

# 1 Appliquer le patron de conception Singleton

Pour vous familiariser avec ce design pattern (de catégorie Création), découvrez comment l'utiliser pour limiter le chargement d'un formulaire à un seul exemplaire.

Vous avez précédemment mis en place des dialogues utilisateur de type modal. Ce type de dialogue oblige l'utilisateur à finir la transaction en cours avant de pouvoir passer à une autre transaction. Très restrictif, il n'assure pas toujours un bon confort d'usage de l'application.

La possibilité offerte par le multifenêtrage apporte quant à elle beaucoup de souplesse mais peut aussi être source de confusions et d'erreurs.

Nous allons ici proposer un compromis entre ces deux approches en recourant au design pattern Singleton appliqué à la gestion des fenêtres.

A partir de la maquette mise à disposition, faire en sorte que le formulaire Singleton ne soit chargé qu'une seule fois. Il faut aussi s'assurer que ce dernier, s'il avait été réduit en icône, s'affiche en mode plein écran. Actuellement, dans la version initiale, il peut être chargé plusieurs fois.