

SI5125 – JEE – TP2 – Filtre, servlet et JSP (*Version Java 21*)

Introduction

Dans le cadre de ce TP, nous allons mettre en place une application web basique permettant à un utilisateur de s'authentifier et de consulter des données sur des personnes. Pour cela nous réaliserons les actions suivantes :

- Mise en place d'objet de représentation des données
- Mise en place d'un gestionnaire de données sans persistance
- Mise en place d'un filtre d'authentification avec session utilisateur
- Mise en place d'une servlet d'authentification
- Mise en place d'un formulaire d'authentification
- Mise en place d'une page web permettant la consultation des personnes
- Mise en place d'une page web permettant la consultation d'une personne

Livrable attendu

Le TP n'est pas à rendre mais sa réalisation est nécessaire à la bonne compréhension du cours.

Prérequis

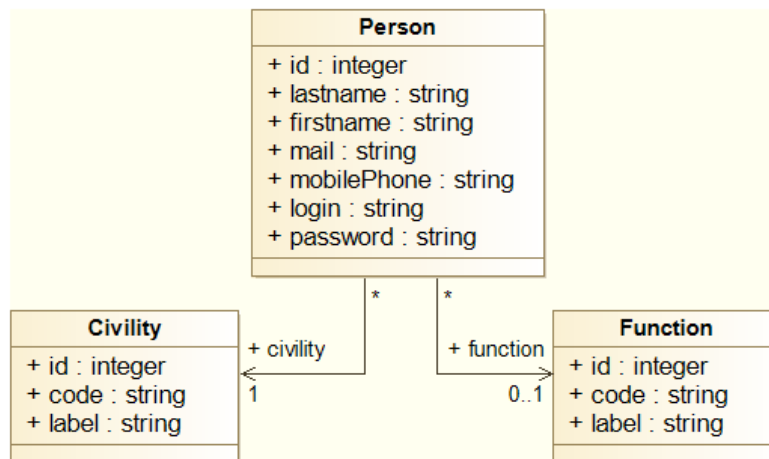
Afin de commencer la réalisation de ce TP, vous devez avoir préparé un projet java similaire à celui du TP1 à la différence que celui-ci se nomme TP2. Afin de le construire vous pouvez vous référer à l'énoncé du TP1 en adaptant le nommage.

A la pratique

Partie 1 – Mise en place de la couche de données

Dans cette partie, nous allons constituer un modèle de données minimaliste qui ne possédera aucune persistance de données, c'est-à-dire que les données manipulées seront réinitialisées après arrêt du serveur.

Voici le modèle à réaliser :



Afin de construire ce modèle dans l'application nous allons réaliser trois « Bean » java. En d'autres termes trois classes Java permettant de manipuler chaque attribut de la classe par des getters/setters accompagnés du constructeur par défaut.

Ces « **Bean** » permettant la représentation d'un élément de données seront ajoutés dans le package « **com.esiee.tp2.domain** ».

Partie 2 - Mise en place d'un gestionnaire de données sans persistance

Maintenant que nous avons nos trois types d'objets de données, nous allons mettre en place un gestionnaire permettant d'en manipuler des instances.

Le gestionnaire se nommera « **Datamodel** » et sera un singleton. Les instances d'objets seront initialisées dans le constructeur.

Le gestionnaire devra proposer les méthodes suivantes :

List<Person> getPersons()	Retourne la totalité des personnes présentes dans le model de données.
Person getPerson(Long id)	Retourne la personne correspondant à l'id
List<Civility> getCivilities()	Retourne la totalité des civilités présentes dans le model de données.
Civility getCivility(Long id)	Retourne la civilité correspondant à l'id
List<Function> getFunctions()	Retourne la totalité des fonctions présentes dans le model de données.
Function getFunction(Long id)	Retourne la fonction correspondant à l'id

Le choix de l'implémentation ainsi que l'ajout de méthodes est libre.

Je conseil cependant pour des raisons de simplicité et d'efficacité l'utilisation d'une « **map** » d'indexation.

La classe « **Datamodel** » devra être placée dans le package « **com.esiee.tp2.model** »

Partie3 - Mise en place d'un filtre d'authentification avec session utilisateur

Nous allons considérer que l'accès aux données nécessite une authentification. Pour faire ce control d'authentification, nous allons utiliser un filtre et la session.

L'idée est qu'à chaque accès à l'application, nous vérifierons si l'utilisateur est authentifié, pour cela, nous utiliserons une variable de session contenant le nom de l'utilisateur, si cette variable n'existe pas, c'est que l'utilisateur n'est pas authentifié. Dans ce cas, nous retournerons un message d'erreur de type « **unauthorized access** ».

Le filtre sera ajouté dans un package nommé : « **com.esiee.tp2.filter** »

Partie 4 - Mise en place d'une servlet d'authentification

Maintenant nous nous disposons de notre filtrage d'authentification, nous allons mettre en place une servlet permettant d'authentifier l'utilisateur, cette ci devra réceptionner deux valeurs. Le login et le mot de passe. Ces valeurs seront comparées avec les données du Datamodel. C'est-à-dire que nous devons avoir en base un objet « Person » avec le même login et mot de passe. Si tel est bien le cas, nous enregistrons l'identifiant de l'utilisateur dans la variable session, validant ainsi l'authentification.

La servlet sera ajoutée dans un package nommé « **com.esiee.tp2.servlet** »

Remarque : Des adaptations sur le code précédant sont à prévoir

Partie 5 - Mise en place d'un formulaire d'authentification

Bien que la servlet mise en place dans la partie 4 permette l'authentification d'un utilisateur. Aucune interface ne permet de le faire aisément.

Mettez en place le formulaire d'authentification qui utilisera la servlet.

Partie 6 - Mise en place d'une page web permettant la consultation des personnes

L'accès à cette page sera sécurisé par l'authentification.

La page permettra de consulter les personnes contenues sur la base. Pour cela vous devrez mettre en place une page JSP.

L'implémentation ainsi que les fonctionnalités de cette page est libre. L'utilisation de tag JSP est un mieux.

Partie 7 - Mise en place d'une page web permettant la consultation d'une personne

L'accès à cette page sera sécurisé par l'authentification.

A l'inverse de la page précédente, cette page permettra uniquement la consultation d'une personne. Pour cela vous devrez mettre en place une page JSP.

L'implémentation ainsi que les fonctionnalités de cette page est libre. L'utilisation de tag JSP est un mieux.

Allez plus loin

Afin d'aller plus loin dans ce TP, vous pouvez compléter les fonctionnalités afin de présenter les différents concepts étudiés et de vous approprier le projet.

Voici quelques idées :

- Mise en place de page JSP permettant de consulter les autres types d'objets
- Utilisation de tags JSP custom
- Mise en place d'une page de recherche de personnes
- Enrichissement du modèle de données
- Utiliser une interface pour spécifier le comportement du gestionnaire de données
- Utiliser la librairie JSTL pour minimiser les inclusions de code java
- ...