

Contrôleur et Javadoc

Le contrôleur **Controle** est une classe singleton utilisée pour gérer les interactions entre l'interface utilisateur et le modèle (ici, la classe **Profil**). Il fournit des méthodes pour créer un profil, récupérer les informations du profil et gérer la persistance locale.

1. Étapes du Fonctionnement

a. Gestion du Singleton

La classe utilise le patron de conception **Singleton** pour garantir qu'une seule instance de **Controle** est créée.

- **Instance unique** : La méthode `getInstance` initialise l'instance si elle n'existe pas.
- **Accès aux données locales** : Utilise **AccesLocal** pour récupérer le dernier profil enregistré.

```
// singleton
public static final Controle getInstance(){ no usages
    if (Controle.instance==null) {
        Controle.instance=new Controle();
    }
    return Controle.instance;
}
```

b. Création du Profil

La méthode **creerProfile** crée un nouvel objet **Profil** et le sauvegarde localement.

- **Paramètres** :
 - Poids, taille, âge, sexe : données nécessaires pour instancier un objet **Profil**.

```
public void creerProfile( Integer poids, Integer taille, Integer age, Integer sexe , Context contexte){ 1 usage
    profil=new Profil(new Date(),poids, taille, age, sexe);
    accesLocal.ajout(profil);
}
```

c. Récupération des Informations

Le contrôleur fournit des méthodes pour récupérer des données spécifiques du profil.

```
    public Integer getPoids(){ 2 usages
        if (profil==null){return null;}
        else {return profil.getPoids();}
    }

    public Integer getTaille(){ 1 usage
        if (profil==null){return null;}
        else {return profil.getTaille();}
    }

    public Integer getAge(){ 1 usage
        if (profil==null){return null;}
        else {return profil.getAge();}
    }

    public Integer getSexe(){ 1 usage
        if (profil==null){return null;}
        else {return profil.getSexe();}
    }
}
```

2. Ajout des Commentaires

1. Tapant /** juste au-dessus des méthodes ou de classe.
2. Appuyant sur Entrée.
3. L'IDE génère automatiquement un squelette de commentaire avec des sections pour :
 - a. La description de la méthode.
 - b. Les paramètres (@param).
 - c. La valeur de retour (@return).
 - d. Les exceptions (@throws), si la méthode en déclare.

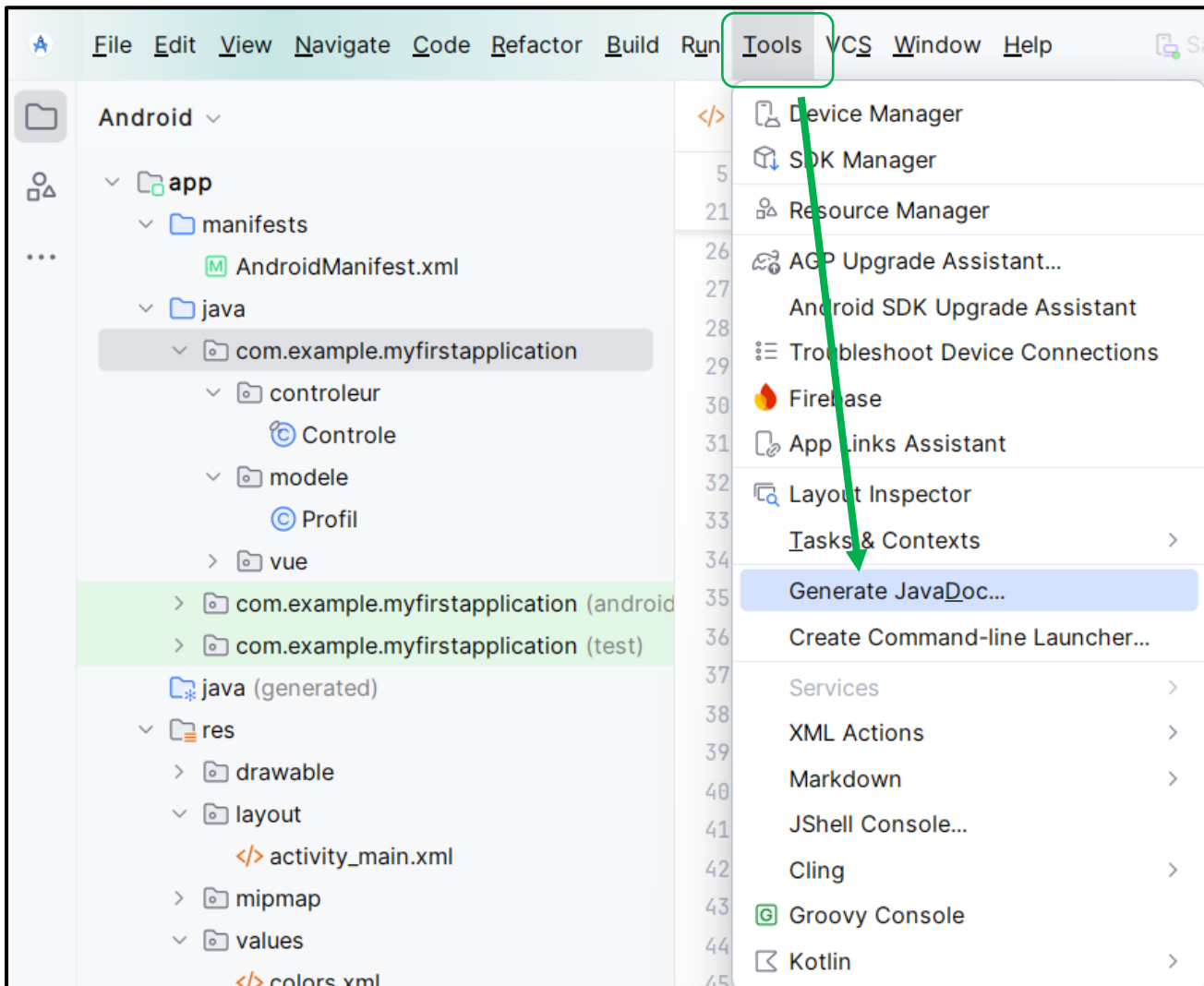
```
/**
 * creation de profil
 * @param poids
 * @param taille est en centimetre
 * @param age
 * @param sexe 1 pour homme et 0 pour femme
 */
public void creerProfil( Integer poids, Integer taille, Integer age, Integer sexe) {
    profil=new Profil(new Date(),poids, taille, age, sexe);
    accesLocal ajout(profil);
}
```

3. Génération de la Javadoc

1. Ajoutant des commentaires Javadoc (comme montré ci-dessus).

2. Générer la Javadoc :

- Dans Android Studio : **Tools > Generate JavaDoc.**
- Configurez l'emplacement de sortie.
- Cliquez sur **OK** pour générer la documentation.



Pourquoi utiliser Javadoc ?

1. **Documentation du code** : Permet d'expliquer clairement le but des classes, méthodes, et attributs.
2. **Génération automatique de documentation** : Javadoc est un outil standard de Java pour générer des documents HTML à partir des commentaires.
3. **Aide aux développeurs** : Les IDE affichent les commentaires Javadoc comme des infobulles lorsque vous survolez une méthode ou une classe.

Package com.example.myfirstapplication.controleur

Class Controle

java.lang.Object
com.example.myfirstapplication.controleur.Controle

public final class Controle
extends Object

Method Summary

All Methods	Static Methods	Instance Methods	Concrete Methods
Modifier and Type	Method		Description
void	creerProfile(Integer poids, Integer taille, Integer age, Integer sexe)		creation de profil
float	getImg()		recuperer img de profil
static final Controle	getInstance()		creation d el'instance
String	getMessage()		recuperation de message

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Method Details

getInstance		
public static final Controle getInstance()		
creation d el'instance		
Returns: instance		

Method Summary

All Methods	Static Methods	Instance Methods	Concrete Methods
Modifier and Type	Method		Description
static final Controle	getInstance()		creation d el'instance

All Methods	Static Methods	Instance Methods	Concrete Methods
Modifier and Type	Method		Description
void	creerProfile(Integer poids, Integer taille, Integer age, Integer sexe)		creation de profil
float	getImg()		recuperer img de profil
static final Controle	getInstance()		creation d el'instance
String	getMessage()		recuperation de message

Method Details

getInstance

```
public static final Controle getInstance()
```

creation d el'instance

Returns:

instance

creerProfile

```
public void creerProfile(Integer poids,  
                          Integer taille,  
                          Integer age,  
                          Integer sexe)
```

creation de profil

Parameters:

poids -

taille - est en centimetre

age -

sexe - 1 pour homme et 0 pour femme

getImg

```
public float getImg()
```

recuperer img de profil

Returns:

img

