

Utilisation de ButterKnife pour le binding de vues

Butter Knife est une bibliothèque de liaison de vues populaire pour Android qui simplifie le processus de liaison des éléments d'interface utilisateur aux champs dans votre code. Cependant, veuillez noter que Butter Knife est désormais obsolète, et le développement Android moderne recommande d'utiliser la liaison de vues (introduite dans Android Jetpack) à la place. Cependant, si vous devez utiliser Butter Knife 6.1.0 (une ancienne version publiée vers 2015) avec un fichier .jar dans un projet Android, voici comment vous pouvez le faire dans Android Studio à partir du 03 mars 2025.

Prérequis

- Android Studio installé (toute version récente devrait fonctionner, bien que le processus puisse légèrement différer en fonction de votre version).
- Un projet Android de base configuré.
- Le fichier `butterknife-6.1.0.jar` téléchargé. Vous pouvez généralement trouver des versions plus anciennes dans des dépôts comme Maven Central ou via des sources archivées si vous avez le fichier .jar localement.

Étapes pour utiliser `butterknife-6.1.0.jar` dans le développement Android

Étape 1 : Ajouter le fichier .jar à votre projet

1. Localiser le dossier `libs` :

- Dans votre projet Android Studio, accédez au module `app`.
- À l'intérieur du dossier `app`, trouvez ou créez un dossier nommé `libs`. S'il n'existe pas, faites un clic droit sur le dossier `app`, sélectionnez `New > Directory`, et nommez-le `libs`.

2. Copier le fichier .jar :

- Copiez le fichier `butterknife-6.1.0.jar` dans le dossier `libs`. Vous pouvez le faire en faisant glisser et en déposant le fichier dans le dossier `libs` dans Android Studio ou en le plaçant manuellement via votre explorateur de fichiers.

3. Synchroniser le fichier .jar avec Gradle :

- Ouvrez le fichier `build.gradle` pour le module `app` (situé à `app/build.gradle`).
- Ajoutez la ligne suivante sous le bloc `dependencies` pour inclure tous les fichiers `.jar` dans le dossier `libs` :

```
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
}
```

- Synchronisez votre projet en cliquant sur le bouton “Sync Project with Gradle Files” dans Android Studio.

Étape 2 : Configurer votre projet Puisque Butter Knife 6.1.0 utilise le traitement des annotations, vous n'avez pas besoin d'une dépendance de processeur d'annotations pour cette version spécifique (contrairement aux versions ultérieures telles que 8.x et au-dessus). Le fichier .jar inclut la bibliothèque d'exécution, et Butter Knife 6.1.0 s'appuie sur la réflexion d'exécution plutôt que sur la génération de code en temps de compilation pour la plupart de ses fonctionnalités.

Cependant, assurez-vous que votre projet est configuré pour prendre en charge les annotations Java :

```
- Dans votre app/build.gradle, assurez-vous que la version Java est compatible (Butter Knife 6.1.0 fonctionne avec Java 6+) : gradle android { compileOptions { sourceCompatibility JavaVersion.VERSION_1_6 targetCompatibility JavaVersion.VERSION_1_6 } }
```

Étape 3 : Utiliser Butter Knife dans votre code

1. Ajouter les annotations Butter Knife :

- Dans votre Activity ou Fragment, importez Butter Knife et annotez vos vues avec @InjectView (l'annotation utilisée dans la version 6.1.0). Par exemple :

```
import android.os.Bundle;
import android.widget.Button;
import android.widget.TextView;
import butterknife.InjectView;
import butterknife.ButterKnife;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    @InjectView(R.id.my_button)
    Button myButton;

    @InjectView(R.id.my_text)
    TextView myText;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ButterKnife.inject(this); // Lier les vues

        // Exemple d'utilisation
        myButton.setOnClickListener(v -> myText.setText("Button clicked!"));
    }
}
```

2. Mise en page XML :

- Assurez-vous que votre fichier de mise en page (par exemple, `res/layout/activity_main.xml`) contient les vues avec les IDs correspondants :

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical">  
  
    <TextView  
        android:id="@+id/my_text"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="Hello World" />  
  
    <Button  
        android:id="@+id/my_button"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="Click Me" />  
/</LinearLayout>
```

3. Lier les vues :

- L'appel `ButterKnife.inject(this)` dans `onCreate` lie les champs annotés aux vues de la mise en page. Notez que dans la version 6.1.0, `inject` est utilisé au lieu de `bind` (qui a été introduit dans les versions ultérieures comme 7.x et 8.x).

Étape 4 : Exécuter votre projet

- Construisez et exécutez votre projet. Si tout est configuré correctement, Butter Knife liera vos vues, et vous devriez voir l'interface utilisateur fonctionner comme prévu.

Notes importantes

- Limitation de version** : Butter Knife 6.1.0 est assez ancien (publié en 2015) et manque de fonctionnalités introduites dans les versions ultérieures, telles que la génération de code en temps de compilation (via des processeurs d'annotations) et le support pour `@BindView` au lieu de `@InjectView`. Il utilise la réflexion en temps d'exécution, ce qui peut légèrement affecter les performances par rapport aux alternatives modernes.
- Statut obsolète** : Butter Knife n'est plus maintenu (le développement s'est arrêté vers 2020), et la recommandation officielle est d'utiliser la liaison de vues ou Jetpack Compose pour les nouveaux projets.

- **Configuration alternative** : Si vous utilisez une version plus récente (par exemple, 10.2.3), vous devriez ajouter une dépendance de processeur d'annotations (par exemple, annotationProcessor 'com.jakewharton:butterknife-compiler:10.2.3'), mais pour 6.1.0 avec un fichier .jar, ce n'est pas nécessaire.

- **Dépannage :**

- Si vous obtenez une erreur "cannot resolve symbol" pour les classes Butter Knife, assurez-vous que le fichier .jar est correctement ajouté au dossier libs et inclus dans les dépendances Gradle.
- Si l'application plante avec une NoClassDefFoundError, vérifiez que le fichier .jar est dans le dossier libs et que le projet a été synchronisé.

Recommandation

Bien que ce guide vous montre comment utiliser butterknife-6.1.0.jar, envisagez de migrer vers la liaison de vues pour une approche plus moderne et prise en charge. Voici une comparaison rapide : - **Butter Knife 6.1.0** : Utilise @InjectView, réflexion en temps d'exécution, intégration manuelle du fichier .jar. - **Liaison de vues** : Intégrée à Android, aucune dépendance externe, génère des classes de liaison en temps de compilation, et est sécurisée contre les nullités.