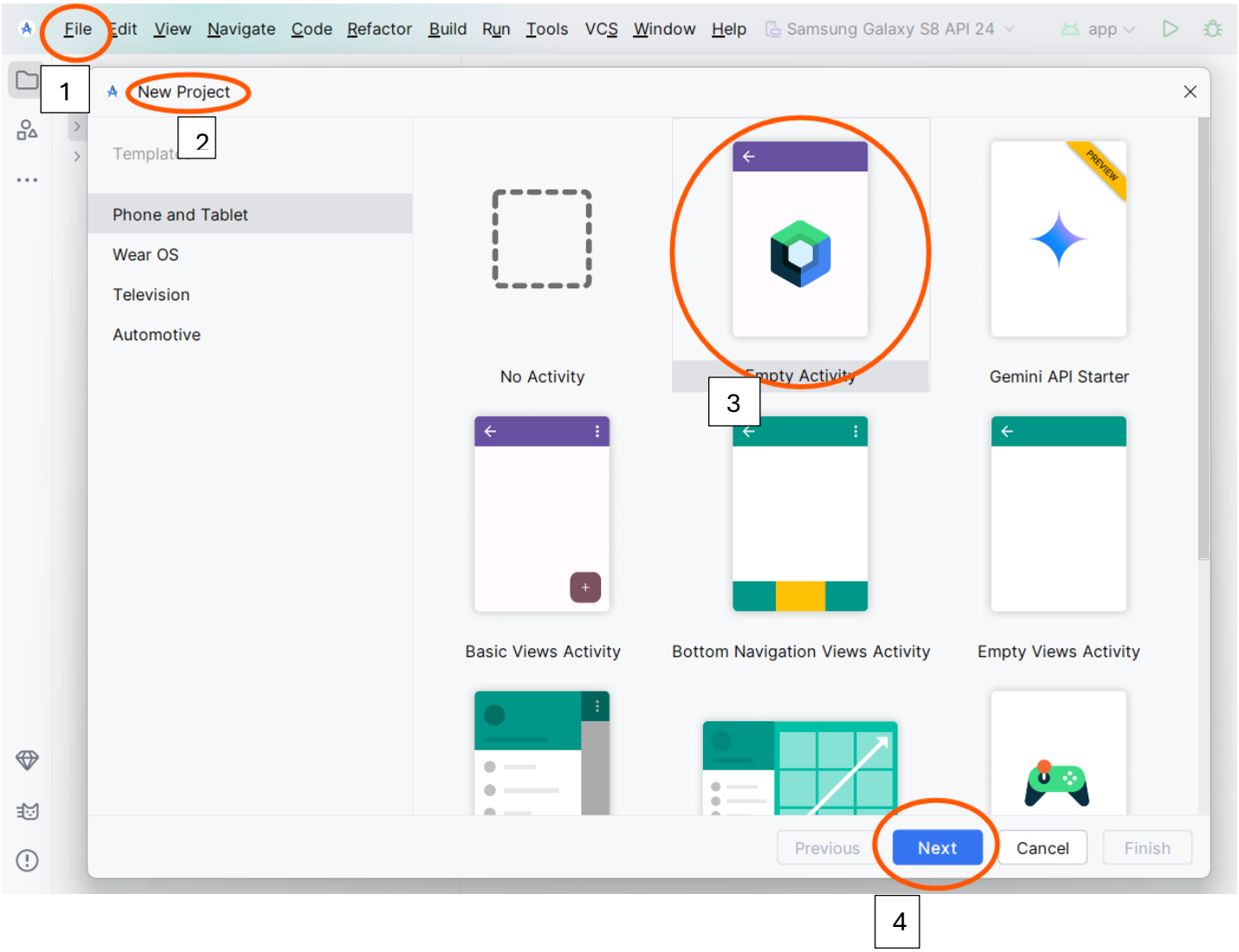
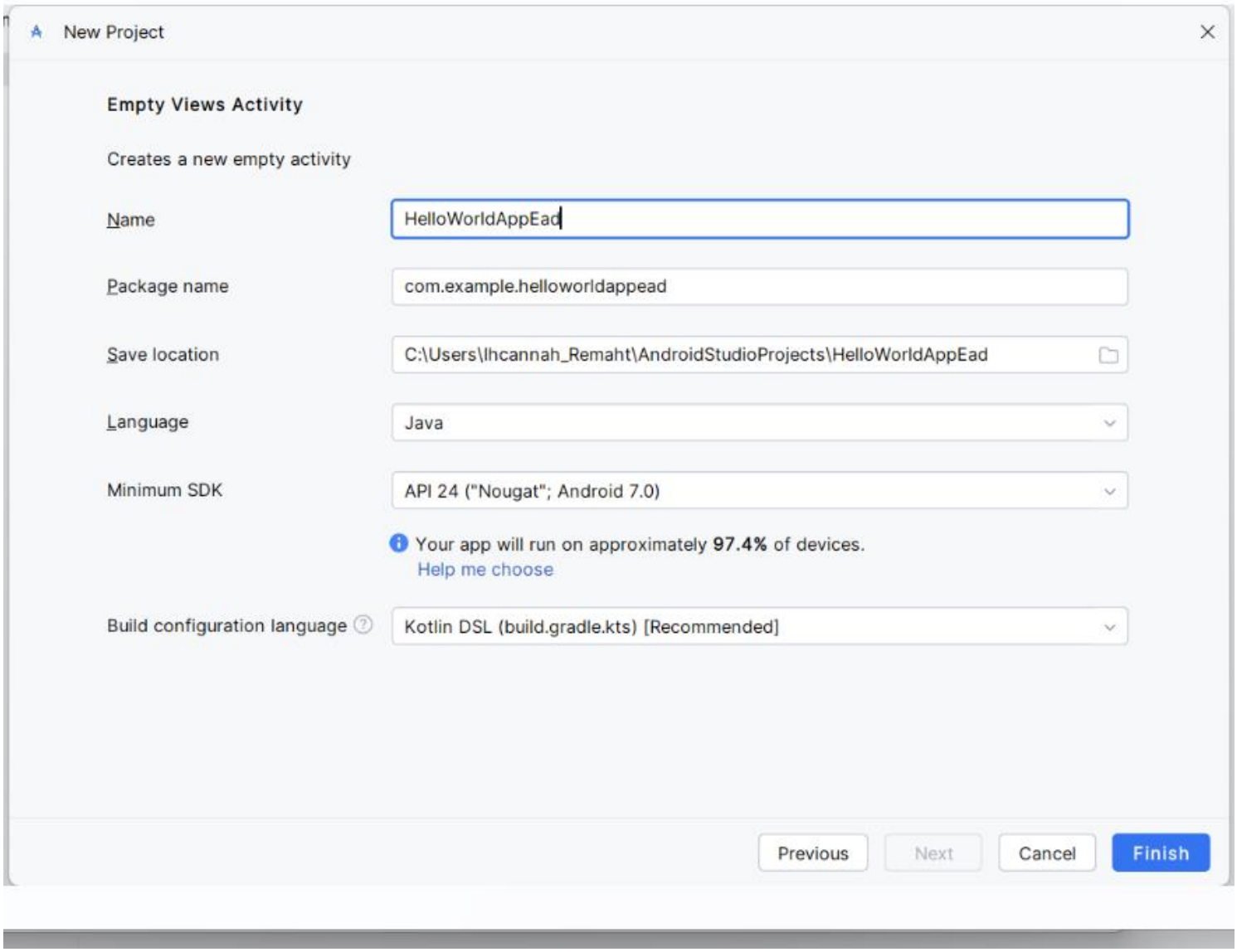
Étapes pour afficher un message puis avec un bouton:

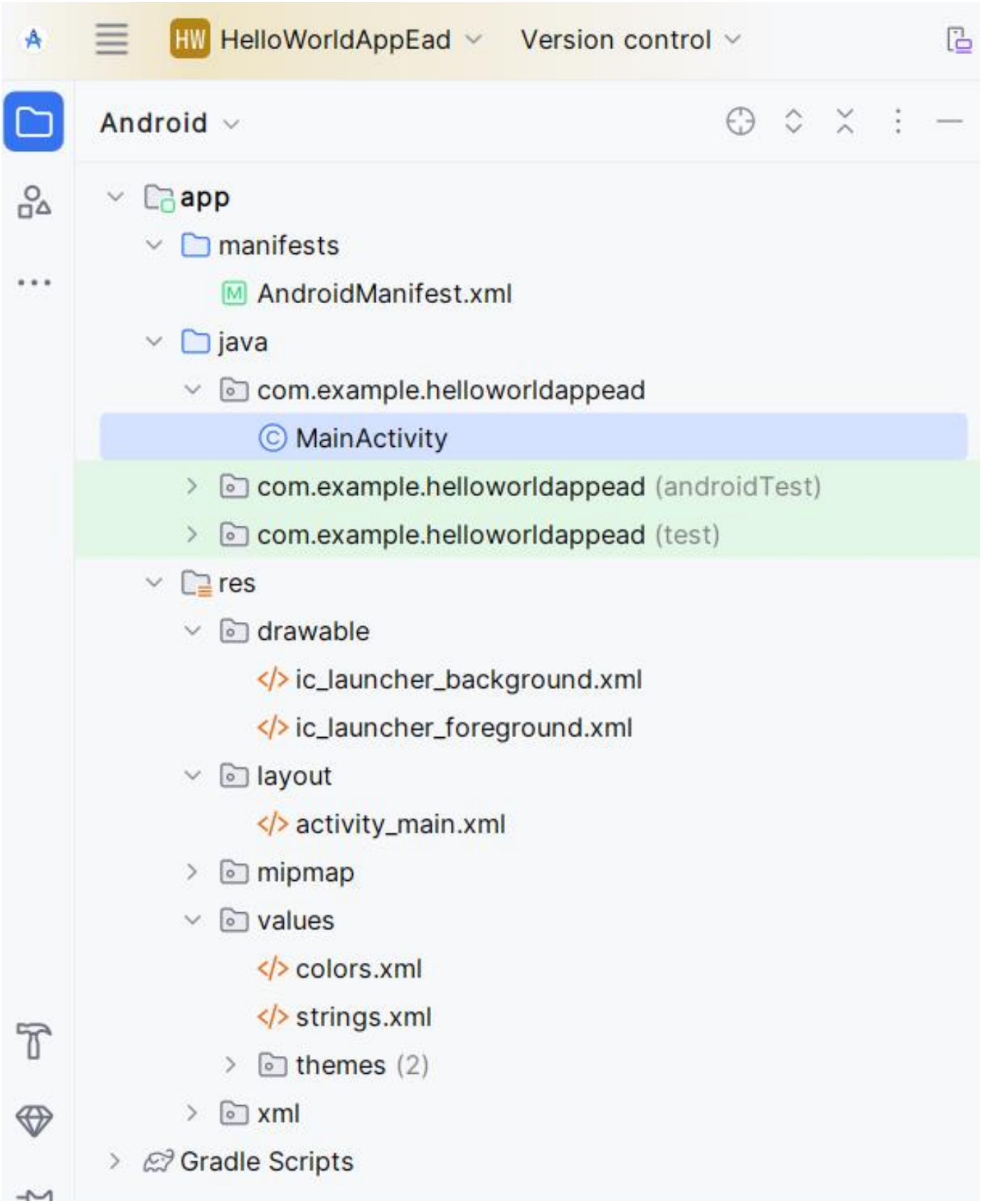
1. **Lancer Android Studio et créer un nouveau projet** :
    - Ouvrez Android Studio.
    - Choisissez **Créer un nouveau projet**.
    - Sélectionnez l'activité par défaut : **Activité vide**.



- Nommez l’ application, par exemple, "HelloWorldAppEad".
- choisissez le langage est **Java** et définissez le version API



le nouveaux projet crée aura cette arborescence :



2. **Configurer le texte du bouton dans `strings.xml` :**
    - Accédez au fichier `res/values/strings.xml`.
    - Ajoutez les chaînes de caractères pour les textes utilisés dans l'application :

```
<resources>
    <string name="app_name">MyFirstApplication</string>
    <string name="Click">Click Me</string>
</resources>
```

Remarques :

Utiliser le fichier strings.xml au lieu d'écrire directement des chaînes de caractères dans le code pour plusieurs raisons :

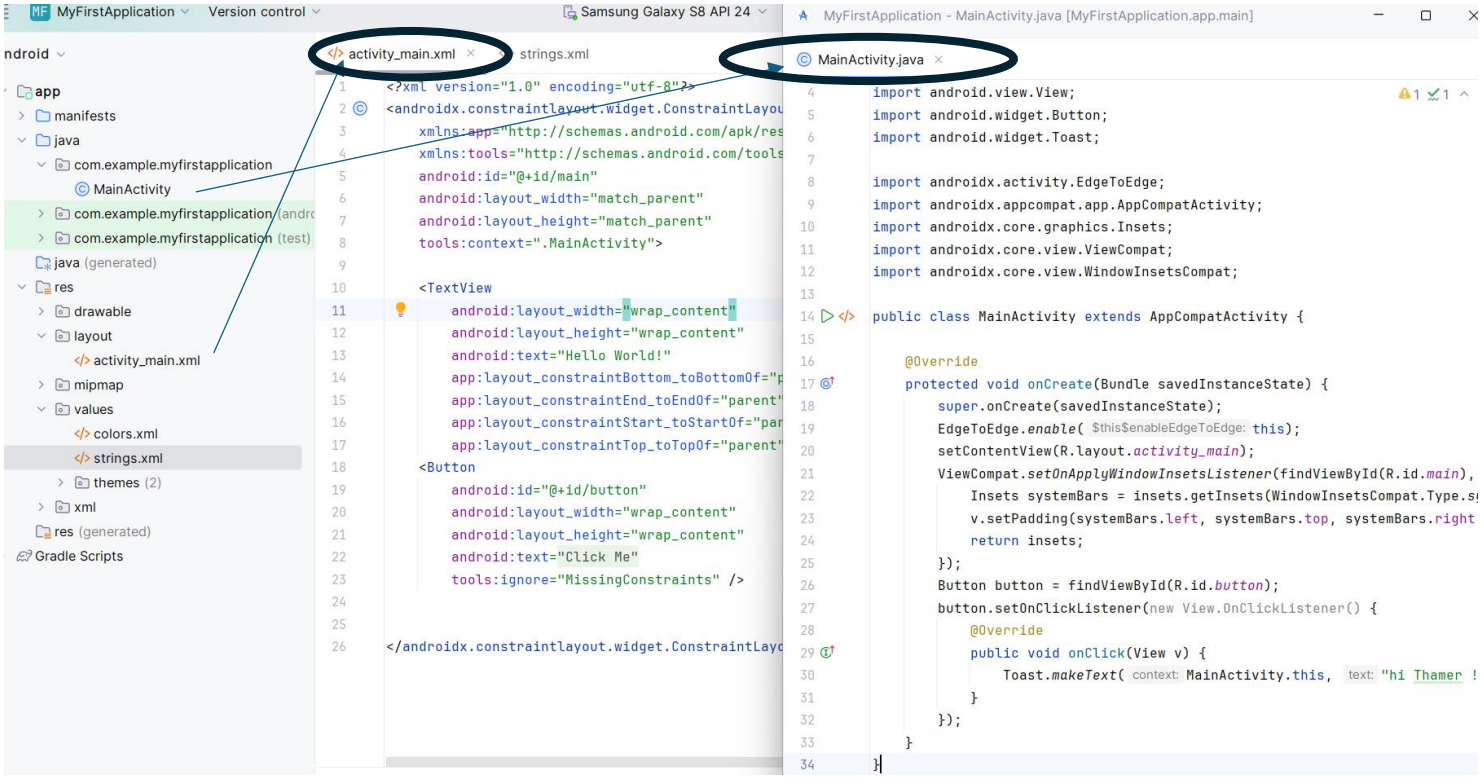
- Éviter la rigidité du code :
    - En écrivant directement des chaînes de caractères dans le code, vous rendez le code plus difficile à maintenir et à adapter. Les textes deviennent rigides et leur modification nécessite de chercher chaque occurrence dans le code, augmentant le risque d'erreurs.
    - Avec strings.xml, tous les textes de l'application sont centralisés, ce qui rend le code plus clair et simplifie les modifications. Si vous avez besoin de changer un libellé, il suffit de le faire dans strings.xml, et la modification sera appliquée partout.
- Faciliter la traduction et l'internationalisation :
    - strings.xml permet de gérer facilement les traductions en fonction de la langue de l'appareil de l'utilisateur.
    - Android détecte automatiquement la langue de l'appareil et utilise les chaînes de caractères appropriées. Cette structure permet de dynamiser l'interface selon la langue de l'utilisateur, sans nécessiter de modifications supplémentaires dans le code.
- Centralisation et réutilisation des chaînes :
    - En centralisant les chaînes dans strings.xml, chaque chaîne est facilement réutilisable dans plusieurs parties de l'application sans duplication. Cela rend l'application plus légère, car une même chaîne peut être référencée par plusieurs éléments d'interface.
- Respect des bonnes pratiques :
    - Utiliser strings.xml est une bonne pratique recommandée par Android et suit le principe de séparation des préoccupations : l'interface utilisateur est séparée du code métier. Cela rend votre application plus professionnelle et prête pour des améliorations futures.

3. **Modifier le fichier XML de l'interface (`activity_main.xml`) :**
   - Dans le dossier **res/layout**, ouvrez `activity_main.xml`.
   - Ajouter **Button** pour afficher un message au clic :

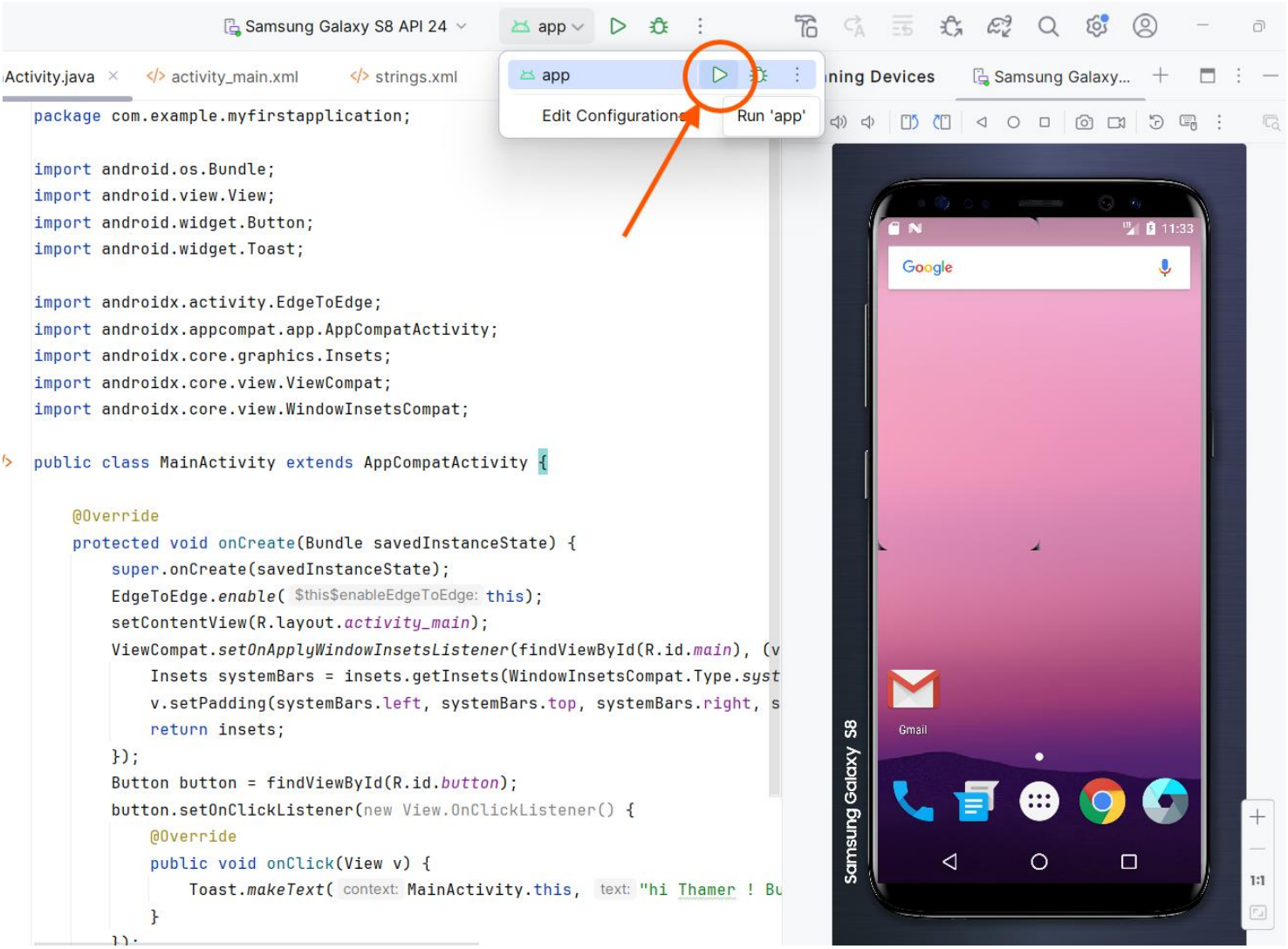
```
<Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/Click"
        tools:ignore="MissingConstraints" />
```

4. **Ajouter la logique dans `MainActivity.java` :**
   - Ouvrez `MainActivity.java` et configurez le bouton pour afficher un message lorsqu'il est cliqué :

```
Button button = findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Toast.makeText(MainActivity.this, "hi Thamer !
Button clicked!", Toast.LENGTH_SHORT).show();
    }
});
```



5. **Exécuter l'application** :
    - Connectez votre émulateur Galaxy S8



6. **Tester le bouton** :
    - Cliquez sur le bouton "Cliquez" pour voir s'afficher le message "hi Thamer Bouton clicked!" en bas de l'écran.