

# Configuration de l'écran de démarrage

## Configuration de l'icône de lancement

Nous allons commencer par configurer l'icône de lancement de l'application sur [Android](#).

Pour ce faire, nous allons utiliser le [package](#) officiel [flutter\\_launcher\\_icons](#).

Nous allons l'ajouter aux dépendances de développement dans le fichier [pubspec.yaml](#) :

```
dev_dependencies:  
  flutter_launcher_icons: "^0.7.3"  
  
flutter_icons:  
  android: true  
  ios: true  
  image_path: "assets/icon/icon.png"  
  adaptive_icon_foreground: "assets/icon/adaptative_icon.png"  
  adaptive_icon_background: "#3498db"
```

Ensuite installez le avec [flutter pub get](#).

[android: true](#) permet de remplacer l'icône de lancement par défaut de [Flutter](#).

[adaptive\\_icon\\_foreground](#) : permet de passer une couleur ou une image qui sera utilisé comme arrière plan pour l'icône adaptative utilisée à partir d'[Android 8](#).

[adaptive\\_icon\\_background](#) : permet de passer une image qui sera utilisée comme image adaptative. Une image adaptative permet de changer la forme de l'icône en fonction des situations dans [Android](#). Il aura parfois une forme arrondie, parfois une forme plus carrée.

Il faut ensuite créer un dossier [icon](#) dans le dossier [assets](#) et y placer vos icônes avec le nom [icon.png](#) et [adaptative\\_icon.png](#).

Vous pouvez ensuite lancer la commande suivante pour générer les icônes :

```
flutter pub run flutter_launcher_icons:main
```

# Configuration de l'écran de démarrage

Lorsque votre application est entrain de charger, un écran blanc est présenté à l'utilisateur par défaut.

Il est cependant possible d'utiliser un écran que vous configurez.

Pour cela, il faut aller dans `android/app/src/main/res/drawable` et modifier le fichier `launch_background.xml` :

```
<item>
  <color
    android:color="#3498db" />
</item>

<item>
  <bitmap
    android:gravity="center"
    android:src="@drawable/splash" />
</item>
```

Il vous suffit ensuite de mettre votre image pour l'écran de lancement dans le dossier.

Vous pouvez également spécifier la couleur du fond en changeant la propriété `android:color`.