



# Mémo Accessibilité Android

E-Accessibility Solutions for Everyone

## Table des matières

Navigation générale

Navigation au clavier

Couleurs et Contrastes

Agrandissement des caractères

Navigation vocale

Checklists accessibilité

Gestes de base Talkback

Outil de test automatique



Recommandations Orange <https://ally-guidelines.orange.com/fr/>

Documentation sous licence CC BY SA 3.0, le logo Orange et les images sont la propriété d'Orange

## Navigation générale

Définir des zones de clic suffisamment grandes ✍

Se référer aux préconisations de la plateforme.  
48dp largeur/hauteur +8dp de marge autour de l'élément.

Adapter l'application aux 2 orientations d'écran ✍

Ne pas dépendre de l'orientation de l'écran (portrait ou paysage), sauf si l'orientation particulière est essentielle pour la compréhension ou l'utilisation du contenu.

Proposer une alternative aux actions gestuelles ✍

Les actions déclenchées par un geste ou un mouvement de l'appareil doivent disposer d'une alternative.

Utiliser des composants standards

Utiliser en priorité les composants natifs. Si aucun composant ne répond au besoin, créer un composant dédié basé sur un composant standard en veillant à conserver la cohérence de navigation et l'accessibilité.

Proposer des textes lisibles ✍

Les polices d'écriture avec empâtement, les tailles de texte trop petites, ainsi que l'italique sont à éviter.

Permettre d'adapter les limites de temps

Lorsque le contexte change après une certaine limite de temps (ex: overlay qui disparaît sur un lecteur vidéo après plusieurs secondes), il faut permettre à l'utilisateur d'adapter ce délai ou proposer une alternative.

```
getRecommendedTimeoutMillis()
```

Fournir des alternatives textuelles

Pour les images porteuses d'information et tout élément qui nécessite une description complémentaire.  
À noter : pas d'alternative pour les images décoratives.

```
contentDescription
```

Indiquer l'état des éléments

Tout élément dont le statut est modifié durant l'utilisation de l'application doit restituer son état à travers son alternative textuelle (élément sélectionné, section dépliée, onglet sélectionné...).

## Ordre de lecture

Par défaut, les lecteurs d'écran lisent les éléments dans un ordre logique. Si besoin, il est possible d'ajuster cet ordre.

```
setAccessibilityTraversalAfter  
setAccessibilityTraversalBefore
```

Prévenir l'utilisateur des changements de contenu

Déclencher la vocalisation d'un message par le lecteur d'écran si un changement dynamique du contenu survient (rafraîchissement de liste, compteur de temps, etc.).

```
announceForAccessibility  
setAccessibilityLiveRegion
```

Gérer la navigation par entêtes

Pour que Talkback puisse reconnaître les entêtes et offrir une navigation plus rapide, ajouter un attribut dans le layout XML pour les éléments considérés comme headers.

```
accessibilityHeading
```

Indiquer les défilements horizontaux

Donner une indication visuelle aux utilisateurs lors de la présence de « scroll » horizontaux (flèches ou des points indiquant une pagination). Penser également à compléter l'alternative textuelle de ces éléments.

Labelliser les champs de formulaire

Lier les champs de formulaires avec leurs labels.

```
android:labelFor ou setLabelFor  
android:hint ou setHint (pour les TextView uniquement)
```

Identifier les erreurs de saisie

En cas d'erreur de saisie, il est important d'indiquer clairement l'erreur, afin que le champ en question soit identifié sans difficulté (indication visuelle, suggestion de correction, message d'erreur explicite).

Définir un titre pour chaque écran ✍

Chaque écran doit avoir un titre qui lui est propre et pertinent, pour se repérer dans l'application (couplé avec le bouton de retour).

## Laisser le contrôle à l'utilisateur

Tout contenu en mouvement, mis à jour, clignotant, doit pouvoir être stoppé, caché ou mis en pause par l'utilisateur. Celui-ci doit rester maître de l'application (pas de vidéo qui se lance automatiquement, etc.).

## Navigation au clavier

### Permettre la navigation au clavier

Pour permettre la navigation au focus, rendre les éléments interactifs focusables.

Attribut `focusable` à `true`

L'ordre du focus doit être précisé à l'aide des options.

`nextFocusDown`, `nextFocusUp`

`nextFocusRight`, `nextFocusLeft`

La position du focus doit être visible à tous moments. Définir l'état **state-focused** des éléments interactifs.

## Couleurs et contrastes

### Vérifier le contraste des couleurs

- 4,5/1 min. pour du texte de taille standard.
- 3/1 min. pour texte de grande taille (150% ou 120% si gras).

### Ne pas s'appuyer seulement sur la couleur

Utiliser un autre moyen que la couleur pour passer l'information.

## Agrandissement des caractères

### Permettre l'agrandissement des caractères

Afin de permettre à l'option « grands caractères » d'interagir correctement avec l'application.

Utiliser une taille de police dynamique : le « **sp** »

Gérer les débordements de contenu : **min-height**, **height** et **wrap\_content**.

### Masquer des éléments à l'accessibilité

Il est possible de masquer des éléments aux outils d'assistance (Talkback ou Voiceover par exemple).

## Navigation vocale

### Déclencher une vocalisation

Il est possible de faire parler le lecteur d'écran (lorsque celui-ci a été activé par l'utilisateur).

`announceForAccessibility`

### Connaître l'état du lecteur d'écran

Dans certains cas il peut être intéressant de savoir si l'utilisateur navigue à l'aide du lecteur d'écran.

### Méthode `isEnabled` de l'objet `AccessibilityManager`

### Vocalisation des listes

Il arrive parfois que les listes soient mal vocalisées (vocalisation de la liste en une seule fois). Pour pallier ce problème, il faut positionner l'attribut **focusable** à **true** pour chaque item de la liste.

## Checklists accessibilité

### Rapide checklist pour une pull request

- Images et icônes ont une alternative textuelle
- L'orientation de l'écran est supportée
- Les champs de saisie sont labellisés
- Les textes peuvent être agrandis
- L'information ne s'appuie pas que sur la couleur
- Les contrastes sont respectés
- Les zones interactives font 48dp par 48dp minimum
- Il n'y a pas de flash lumineux (épilepsie)

### Checklist complète avant mise en production

Avant de mettre en production votre application, il vous est conseillé de faire, pour chaque écran, une vérification de l'ensemble des critères indiqués sur cette fiche. Vous trouverez aussi une liste de critères plus exhaustive sur le site des recommandations accessibilité Orange, ainsi que sur la va11ydette :

<https://la-va11ydette.orange.com/?list=wcag-android&lang=fr>

## Gestes de base Talkback

### Activer le lecteur d'écran

Paramètres > accessibilité > Talkback



Déplacement avec un doigt sur l'écran

Pour explorer l'écran et entendre la vocalisation de l'élément qui est touché.



Double-clic n'importe où sur l'écran

Pour ouvrir ou activer l'élément qui a été touché (vocalisé) en dernier.



Slide vers la droite en utilisant un doigt

Pour déplacer le focus sur l'élément suivant.



Slide vers la gauche en utilisant un doigt

Pour déplacer le focus sur l'élément précédent.

## Outil de test automatique

### Accessibility Scanner

Application disponible sur le Play Store permettant d'effectuer des tests d'accessibilité sur votre application.



Critères d'accessibilité à vérifier dès la phase de conception