



Plan pour intégrer plusieurs langues dans l'app Kazidomo Confiance de manière structurée et efficace :

Étape 1 — Créer un inventaire des éléments à traduire

On va regrouper tous les éléments textuels de l'interface pour l'MVP :

Éléments de navigation

- Boutons : “Suivant”, “Retour”, “Valider”, “Annuler”
- Menu : “Profil”, “Tableau de bord”, “Certifications”, “Services”, “Langue”

Pages principales

- Page d'accueil (accroche, description du service)
- Formulaires d'inscription et de connexion
- Fiche utilisateur / talent (bio, compétences, avis)
- CGU et politique de confidentialité

Messages système

- Erreurs : “Mot de passe invalide”, “Connexion impossible”
- Succès : “Compte créé avec succès”, “Demande envoyée”

ajouter d'autres sections spécifiques à mon app comme des “FAQs”, “candidatures”, ou “disponibilité géographique”

Étape 2 — Structuration des fichiers de traduction

Selon mon stack, on peut structurer ça sous forme de fichiers JSON, YAML, ou autres.

Exemple pour le français :

```
`json
{
  "home.welcome": "Bienvenue sur Kazidomo Confiance",
  "button.next": "Suivant",
  "error.invalidPassword": "Mot de passe invalide"
},
`
```

Et pour l'anglais :

```
`json
{
  "home.welcome": "Welcome to Kazidomo Confiance",
  "button.next": "Next",
  "error.invalidPassword": "Invalid password"
},
`
```

Étape 3 — Mise en œuvre du changement de langue

- Intégrer un sélecteur de langue dans le header ou le menu latéral.
- Stocker la langue choisie localement (via localStorage ou SharedPreferences, selon la techno).
- Garantir que le contenu se met à jour automatiquement après le changement de langue.

Étape 4 — Feuille de route multilingue

1. Semaine 1 : Inventorier le contenu + créer les premiers fichiers de traduction FR/EN.

2. Semaine 2 : Intégration du système i18n dans le code + sélecteur de langue.
3. Semaine 3 : Test utilisateur pour valider la clarté et la cohérence des traductions.
4. Après MVP : Ajouter Swahili, Espagnol ou d'autres langues selon les retours terrain.