

Pages, composants et structure UI

Vue d'ensemble

Le frontend sépare globalement les **écrans complets**, les **composants réutilisables** et les **popups**. Cette page sert de guide pour savoir où ajouter ou modifier une interface.

Structure UI

```
flowchart TD
    APP[AppModule] --> ROUTES[RouterModule.forRoot]
    ROUTES --> PAGES[Pages]
    PAGES --> COMP[Components]
    PAGES --> POP[Popup]
    PAGES --> SERVICES[Services Angular]
    COMP --> SERVICES
```

Pages

`src/app/Pages/` contient les écrans routables ou les grosses vues fonctionnelles. Les dossiers actuellement présents incluent notamment :

- `PageLogin`, `PageForgotPassword`, `PageUnauthorized`
- `PageIndex`
- `PageWebgl`
- `PageSupport` et `PageSupportAdmin`
- `PageAdminPresence`
- `PageTableList` et `PageUserInformation`
- `PageSettings`, `PageDemoSearch`, `PageActiveSimulation`, `PageSimulationComments`

Components

`src/app/Components/` contient les composants destinés à être utilisés par d'autres vues. On y trouve notamment `CommunicationOverlay` et `WebPanel`.

Un élément devrait devenir un composant partagé lorsqu'il possède une responsabilité UI autonome ou qu'il est utilisé par plusieurs pages.

Popup

`src/app/Popup/` regroupe les interfaces modales. Lorsqu'une interaction est temporaire et ne mérite pas une route complète, vérifier d'abord si elle appartient à cette couche.

Routing

Le routing est centralisé directement dans `src/app/app.module.ts` avec `RouterModule.forRoot(...)`. Il n'existe pas de `app-routing.module.ts` séparé.

Les routes sont regroupées par niveau d'accès : publiques, utilisateur, utilisateur Uniksime, admin et true admin. Voir **Structure et routing Angular** pour la liste détaillée.

Créer une nouvelle vue

flowchart LR

```
NEED[Nouvelle interface] --> ROUTE{Doit avoir une URL ?}
ROUTE -- Oui --> PAGE[Nouvelle Page]
PAGE --> APP[Ajouter déclaration + route dans app.module.ts]
ROUTE -- Non --> REUSE{Réutilisable ?}
REUSE -- Oui --> COMPONENT[Composant]
REUSE -- Non --> POPUP{Interaction modale ?}
POPUP -- Oui --> P[Popup]
POPUP -- Non --> LOCAL[Composant local à la page]
```

Conventions pratiques

- Une Page orchestre l'écran; elle ne devrait pas absorber toute la logique métier.
 - Les appels backend réutilisables passent par les services Angular.
 - Un composant partagé ne devrait pas dépendre inutilement d'une route précise.
 - Lorsqu'une page devient très volumineuse, extraire les parties UI cohérentes plutôt que créer une deuxième page sans raison.
 - Avant de créer un nouveau composant, vérifier si un équivalent existe déjà dans `Components/` ou `Popup/`.
-

Revision #1

Created 2026-09-18 17:46:21 UTC by Mathis

Updated 2026-09-18 17:46:25 UTC by Mathis