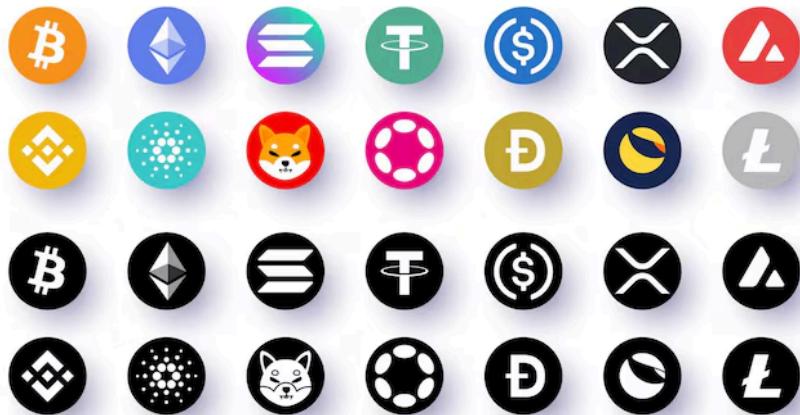


Jean Boinot

Documentation - Projet Api Crypto



SOMMAIRE

SOMMAIRE

I - Présentation	p.2
• A - Objectif du projet	p.2
II - Architecture du projet	p.2
• A - Schéma	p.2
• B - Quel est le rôle d'un composant	p.3
• C - Quel est le rôle d'un service	p.3
• D - Quel est le rôle du garde	p.3
III - Technologies utilisées	p.4
• A - Listage des technologies	p.4
IV - Étapes d'installation	p.4
• A - Commande pour installation avec terminal	p.4
V - Guide d'utilisation	p.5
• A - Navigation dans l'application	p.5
• B - Cas d'usage concrets	p.6

I - Présentation :

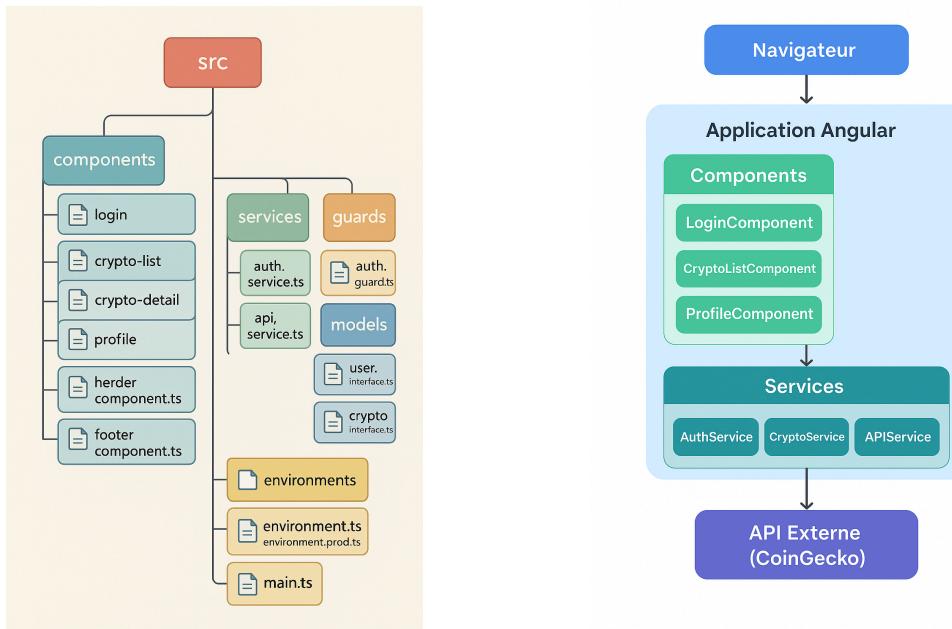
A - Objectif du projet :

Ce projet a pour but de partager les informations relatives aux cours des cryptomonnaies à ses utilisateurs. L'objectif est donc d'avoir les informations en temps

réelles en communiquant avec une api, tout en ayant une interface dit accessible et simple d'utilisation.

II - Architecture du projet :

A - Schéma :



A - Quel est le rôle d'un composant :

Un Component est une classe TypeScript qui contrôle une portion de l'écran appelée vue. C'est l'unité de base de l'interface utilisateur dans Angular.

- Son rôle est donc d'afficher les données.
- Capturer les actions utilisateurs
- Gérer l'état local du composant

B - Quel est le rôle d'un service :

Un Service est une classe qui contient la logique métier, les appels API, et l'état partagé de l'application. Il est injectable dans n'importe quel composant.

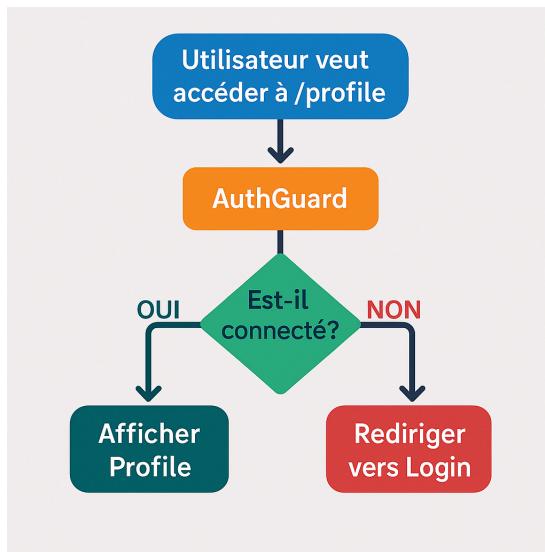
- Une logique métier dit réutilisable
- Communique avec les api
- Partage les données entre les composants

- Gestion de l'état

C - Quel est le rôle du garde :

Un Guard est une classe qui détermine si une route peut être activée, désactivée ou si l'utilisateur peut la quitter. C'est le "videur" de l'application.

- Son rôle est donc la gestion des accès routes



III - Technologies utilisées :

A - Listage des technologies :

Frontend: Angular 18, TypeScript 5

Graphiques: Chart.js 4.4

Styles: CSS natif avec design responsive

API: CoinGecko API (gratuite)

IV - Etapes d'installations :

A- Commande pour installation avec terminal

Cloner le repository

[git clone https://github.com/votre-username/crypto-tracker.git](https://github.com/votre-username/crypto-tracker.git)

[cd crypto-tracker](#)

Installer les dépendances

[npm install](#)

Lancer le serveur de développement
ng serve

Ouvrir dans le navigateur
<http://localhost:4200>

V - Guide d'utilisation

A - Navigation dans l'application

Ce guide détaille les étapes pour utiliser efficacement toutes les fonctionnalités de l'application Crypto Tracker.

Processus de connexion

Étape 1 : Accès initial L'utilisateur accède à l'application via l'URL <http://localhost:4200>. La page de connexion s'affiche automatiquement pour les utilisateurs non authentifiés.

Étape 2 : Authentification

- Saisir l'email dans le champ correspondant
- Entrer le mot de passe
- Cliquer sur le bouton "Se connecter"
- En cas d'erreur, un message rouge s'affiche sous le formulaire

Étape 3 : Redirection automatique Après connexion réussie, l'utilisateur est automatiquement redirigé vers le dashboard principal affichant la liste des cryptomonnaies.

Structure de navigation

Header persistant Une fois connecté, un header apparaît en haut de l'application avec :

- Logo/Nom de l'application à gauche
- Menu de navigation au centre (Cryptos, Profil)
- Bouton de déconnexion à droite

Navigation principale

- Dashboard (</cryptos>) : Vue d'ensemble du marché
- Détail (</crypto/:id>) : Analyse approfondie d'une crypto
- Profil (</profile>) : Informations utilisateur
- Déconnexion : Retour à la page de login

Utilisation de la liste des cryptos

Affichage par défaut Au chargement, les 50 principales cryptomonnaies s'affichent automatiquement, triées par capitalisation boursière décroissante.

Interprétation des données

- Carte verte : La crypto est en hausse sur 24h
- Carte rouge : La crypto est en baisse sur 24h
- Prix : Toujours affiché en USD
- Pourcentage : Variation sur les dernières 24 heures

Actions disponibles

- Scroll vertical pour parcourir la liste complète
 - Clic sur une carte pour accéder aux détails
 - Utilisation de la barre de recherche pour filtrer
-

B - Cas d'usage concrets

Cette section présente des scénarios d'utilisation typiques pour tirer le meilleur parti de l'application.

Cas d'usage 1 : Suivre l'évolution du Bitcoin

Objectif : Analyser la tendance du Bitcoin sur le mois dernier

Procédure :

1. Sur le dashboard, utiliser la barre de recherche
2. Taper "bitcoin" ou "btc"
3. Cliquer sur la carte Bitcoin qui apparaît
4. Sur la page de détail, cliquer sur le bouton "30j"
5. Observer le graphique et les statistiques

Conseil :

- Noter le prix le plus haut et le plus bas du mois
- Comparer avec l'ATH historique affiché
- Vérifier le volume d'échange pour évaluer l'activité

Cas d'usage 2 : Analyse comparative

Objectif : Comparer l'évolution de plusieurs cryptos sur la même période

Procédure :

1. Ouvrir la première crypto (ex: Bitcoin)
2. Noter les valeurs clés sur la période choisie
3. Retourner au dashboard via le bouton retour

4. Répéter pour d'autres cryptos (ex: Ethereum, Cardano)
5. Comparer les performances relatives

Information relative :

- Variation en pourcentage plutôt qu'en valeur absolue
- Même période pour toutes les cryptos
- Prise en compte du volume pour la liquidité

Cas d'usage 3 : Recherche rapide quotidienne

Objectif : Vérifier rapidement ses cryptos préférées chaque jour

Procédure optimisée :

1. Se connecter à l'application
2. Utiliser directement la recherche
3. Taper les 3 premières lettres (ex: "bit", "eth", "car")
4. Vérifier prix et variation
5. Cliquer uniquement si mouvement significatif

Gain de temps : Cette méthode permet de vérifier 5-6 cryptos en moins d'une minute.