



Users Stories – P_FUN-323-PlotThoseLines

#1 – \[EPIC\] Ingestion des données et API

#2 – Lister les marchés crypto



En tant que développeur

Je souhaite appeler l'API TokenInsight pour récupérer la liste des cryptos disponibles et normaliser la réponse en modèle `MarketInfo`.

Pour disposer d'une source de données que l'UI peut afficher.



Tests d'acceptance

- Le backend appelle `api/v1/coins/list`.
- Transformation en `MarketInfo` (id, symbole, nom, prix, volume 24h...).
- Gestion d'erreurs avec feedback clair.
- Exécution < 2s, tri par symbole alphabétique.

#3 – Récupération des données historiques



En tant que utilisateur

Je souhaite récupérer des séries temporelles historiques depuis TokenInsight.

Pour alimenter les graphiques avec de vraies données.



Tests d'acceptance

- Appel `/api/v1/history/coins/{id}`, transformation en `TimeSeriesPoint`.
- Gestion des erreurs (logs + UI informée).
- Séries récupérées < 2s, triées par timestamp.

#9 – \[EPIC\] Importation des données

#8 – Importation de données JSON local



En tant que utilisateur

Je souhaite importer un asset local depuis un fichier JSON.
Afin de visualiser mes propres données dans l'application.



Tests d'acceptance

- Fichier JSON valide → asset importé avec nom, symbole et logo.
 - L'asset apparaît dans ma liste locale.
-

#13 – Importation des données API



En tant que utilisateur

Je souhaite importer un asset via API en le recherchant par nom.
Afin de accéder à des données crypto à jour automatiquement.



Tests d'acceptance

- Je recherche un asset (ex. Ethereum) puis clique *Add new asset*.
 - L'asset apparaît dans ma liste API avec ses données.
-

#14 – Expérience utilisateur lors de l'importation d'assets



En tant que utilisateur

Je souhaite être guidé par l'interface lors de l'importation (états vides, erreurs, confirmation).

Afin de éviter les erreurs et savoir si l'ajout a réussi.



Tests d'acceptance

- Formulaire incomplet, fichier invalide ou doublon → message clair affiché.
 - Succès ou échec mis à jour dans la liste.
-

#15 – Clarté de la page d'Assets



En tant que utilisateur

Je souhaite que l'UI sépare clairement assets locaux et API.

Afin de gérer mes imports sans confusion.

📌 Tests d'acceptance

- Présentation claire : distinction visuelle avec noms, symboles, icônes.
-

#4 – \[EPIC\] Visualisation & Interaction

#5 – Affichage multi-séries temporelles

En tant que analyste

Je souhaite afficher plusieurs séries temporelles superposées.

Pour comparer prix, volume, indicateurs facilement.

📌 Tests d'acceptance

- Affichage simultané d'au moins 2 séries (couleurs distinctes, axe Y flexible).
 - Tooltips précis.
 - Séries alignées sur le temps même avec données manquantes.
-

#6 – Sélectionner des intervalles

En tant que utilisateur

Je souhaite sélectionner des intervalles prédéfinis (1d, 1w, 1m, 1y, All).

Pour explorer les données à différentes échelles.

📌 Tests d'acceptance

- Boutons visibles pour chaque intervalle.
 - Graphiques mis à jour automatiquement.
 - Données alignées correctement.
 - Intervalle sélectionné mis en évidence.
-

#7 – Superposer des périodes différentes

En tant que analyste

Je souhaite comparer des séries couvrant différentes périodes (ex. 2020 vs

2021).

Pour analyser saisonnalités et comportements périodiques.

Tests d'acceptance

- Séries superposées distinctement (style, couleur, légende).
 - Alignement temporel normalisé.
 - Tooltips indiquant date et valeur.
-

#10 – \[Task\] Maquettes Figma

 **En tant que** développeur

Je souhaite concevoir et utiliser des maquettes Figma de l'application.

Pour avoir une vision claire et validée avant de coder.

Tests d'acceptance

- Écrans *Importation de données*, *Graphiques* et *Paramètres* créés dans Figma.
 - Navigation visible et cohérente.
 - Composants identifiés.
 - Alignement avec les User Stories.
-

#16 – Page Settings

 **En tant que** utilisateur

Je souhaite renseigner et gérer ma clé API dans *Settings*.

Afin de connecter l'application à mes sources externes.

Tests d'acceptance

- Champ API visible dans *Settings*.
- Clé API sauvegardée et utilisée pour récupérer mes données.