

1. Tests unitaires des utilitaires (/utils)

Pourquoi ? Ils sont utilisés partout (download, auth, state...). Les valider tôt évite de fausses erreurs dans les tests supérieurs.

crypto-utils.js

file-operations.js

async-queue.js

event-throttler.js

2. Tests des constantes et événements

Vérifie event-types.js et error-codes.js pour t'assurer qu'il n'y a pas d'erreurs de référence ou de logique.

Teste aussi event-bus.js + event-logger.js (cohérence, journalisation, propagation).

3. Tests unitaires des composants "core"

Tu poses ici les fondations.

state-manager.js : essentiel pour la gestion des statuts de téléchargement.

config-manager.js : teste lecture/écriture + fallback.

error-handler.js : vérifie les messages retournés + erreurs custom.

4. Tests des modules critiques en logique métier

C'est le cœur fonctionnel :

download-manager.js (logique, événements, gestion de file)

download-queue.js (concurrency, débordements, priorités)

format-converter.js (vérifie les conversions et formats supportés)

5. Tests des adaptateurs (/adapters)

Un par un. Commence par les plus utilisés (YouTube, Tidal, Spotify).

Utilise des mocks d'API pour simuler le comportement sans te connecter réellement.

Vérifie les erreurs typiques (token expiré, 404, débit limité...).

6. Tests du module d'authentification

Important pour tout ce qui est sécurisé :

auth-manager.js : succès/échec + refresh de token

secure-token-store.js : intégrité, persistance

7. 🎨 Tests de l'interface utilisateur

Teste avec des simulateurs ou des frameworks comme Playwright ou Electron-mocha :

ui-manager.js + component-registry.js

Interactions basiques, clics, affichage des erreurs...

8. 🔄 Tests d'intégration (flux complets)

Une fois les modules validés :

Ex. : connexion → téléchargement → conversion → fichier final

Valide la communication entre modules.

9. 🚀 Tests de build & cross-plateformes (/build)

Tu vérifies que tout fonctionne sur macOS, Windows, Linux.

Lance des builds automatisés avec des scripts de test pour chaque OS.