

STRUCTURE COMPLÈTE DU PROJET TAXASGE

Structure racine du projet

```
taxasge-app/      # Racine du repository GitHub
├── .github/       # Configuration GitHub
│   └── workflows/ # Workflows GitHub Actions
│       ├── flutter_ci.yml
│       ├── android_build.yml
│       └── documentation_build.yml
├── android/       # Configuration Android (généré par Flutter)
├── ios/           # Configuration iOS (généré par Flutter)
├── lib/           # Code source Dart/Flutter
├── test/          # Tests
├── assets/        # Ressources statiques
├── scripts/       # Scripts utilitaires (à créer)
├── docs/          # Documentation (à créer)
├── backend/       # Code backend (à créer lors de la Phase 6)
├── pubspec.yaml   # Configuration Flutter
├── README.md      # Documentation principale
└── .gitignore     # Configuration Git
```

Structure détaillée du dossier lib/

```
lib/
├── main.dart      # Point d'entrée de l'application
├── api/           # Interaction avec l'API
│   ├── api_client.dart
│   └── endpoints.dart
├── auth/         # Authentification (Phase 5)
```

- | └─ auth_service.dart
- | └─ user_model.dart
- | └─ constants/ # Constantes de l'application
- | └─ app_constants.dart
- | └─ database/ # Gestion de la base de données locale
- | └─ database_service.dart
- | └─ schema.dart
- | └─ migration.dart
- | └─ dao/ # Data Access Objects
- | └─ ministerio_dao.dart
- | └─ sector_dao.dart
- | └─ categoria_dao.dart
- | └─ sub_categoria_dao.dart
- | └─ concepto_dao.dart
- | └─ document_requis_dao.dart
- | └─ mot_cle_dao.dart
- | └─ favori_dao.dart
- | └─ ml/ # Intégration du modèle ML (Phase 3)
- | └─ model_service.dart
- | └─ query_processor.dart
- | └─ response_generator.dart
- | └─ models/ # Modèles de données
- | └─ ministerio.dart
- | └─ sector.dart
- | └─ categoria.dart
- | └─ sub_categoria.dart
- | └─ concepto.dart
- | └─ document_requis.dart
- | └─ sync_record.dart
- | └─ navigation/ # Gestion de la navigation
- | └─ app_router.dart

```
|   ├── route_names.dart
|   └── navigation_service.dart
|   ├── providers/           # Gestion d'état (Provider)
|   |   ├── tax_provider.dart
|   |   ├── search_provider.dart
|   |   └── favorites_provider.dart
|   ├── screens/           # Écrans de l'application
|   |   ├── home_screen.dart
|   |   ├── categories_screen.dart
|   |   ├── tax_detail_screen.dart
|   |   ├── search_screen.dart
|   |   ├── chatbot_screen.dart
|   |   ├── profile_screen.dart
|   |   ├── advanced_search_screen.dart
|   |   ├── favorites_screen.dart
|   |   ├── auth/
|   |   |   ├── login_screen.dart
|   |   |   └── register_screen.dart
|   |   └── tools/
|   |       ├── tax_calculator_screen.dart
|   |       └── calendar_screen.dart
|   ├── services/         # Services de l'application
|   |   ├── offline_service.dart
|   |   ├── share_service.dart
|   |   ├── sync_service.dart
|   |   ├── update_service.dart
|   |   ├── chatbot_service.dart
|   |   ├── favorites_service.dart
|   |   ├── export_service.dart
|   |   └── notification_service.dart
```

- | └─ analytics_service.dart
- | └─ crash_reporting_service.dart
- | └─ feedback_service.dart
- | └─ ota_update_service.dart
- └─ theme/ # Thème de l'application
 - | └─ app_theme.dart
 - | └─ custom_widgets_styles.dart # Styles des widgets personnalisés
 - | └─ extensions.dart # Extensions pour faciliter l'utilisation du thème
- └─ utils/ # Utilitaires
 - | └─ delta_applier.dart
 - | └─ log_collector.dart
 - | └─ event_tracker.dart
 - | └─ error_boundary.dart
- └─ widgets/ # Widgets réutilisables
 - └─ tax_card.dart
 - └─ ministry_grid_item.dart
 - └─ search_bar.dart
 - └─ category_list_item.dart
 - └─ bug_report_dialog.dart
 - └─ chat/
 - | └─ message_bubble.dart
 - | └─ suggestion_chip.dart
 - | └─ chat_input.dart
 - └─ search/
 - | └─ filter_chips.dart
 - | └─ search_history.dart
- └─ feedback/
 - └─ rating_dialog.dart
 - └─ feedback_form.dart

Structure détaillée du dossier assets/

assets/

- └─ data/ # Données JSON
- | └─ taxes_schema.json # Schéma de données
- | └─ taxes.json # Données extraites
- └─ fonts/ # Polices de caractères
- | └─ Roboto-Regular.ttf
- | └─ Roboto-Bold.ttf
- | └─ Roboto-Mono.ttf
- └─ images/ # Images et graphiques
- | └─ logo.png
- | └─ splash.png
- | └─ ministries/ # Icônes des ministères
- └─ icons/ # Icônes de l'application
- └─ ml/ # Modèles de machine learning
 - └─ taxasge_model.tflite # Modèle TensorFlow Lite
 - └─ training_corpus.json # Corpus d'entraînement

Structure détaillée du dossier scripts/

scripts/ # Scripts utilitaires

- └─ excel_to_json_converter.py # convertit les données du fichier Excel en format JSON
- └─ corpus_generator.py # Génération du corpus pour NLP
- └─ train_model.py # Entraînement du modèle NLP
- └─ signing/ # Scripts de signature
 - └─ android_signing.sh
 - └─ ios_provisioning.sh

Structure détaillée du dossier test/

test/

- └─ setup.dart # Configuration commune pour les tests

```
└─ unit/                # Tests unitaires
|   └─ models/
|   └─ services/
|   └─ utils/
|   └─ providers/
└─ integration/         # Tests d'intégration
    └─ database_test.dart
    └─ api_test.dart
    └─ sync_test.dart
    └─ auth_flow_test.dart
└─ performance/         # Tests de performance
    └─ startup_test.dart
    └─ search_benchmark.dart
    └─ memory_profile.dart
    └─ battery_usage.dart
```

Structure détaillée du dossier docs/

```
docs/                  # Documentation
└─ architecture/
|   └─ technical_architecture.md
|   └─ database_schema.md
└─ api/                # Documentation de l'API
    └─ swagger.yaml
└─ screenshots/        # Captures d'écran de l'application
└─ test_plan.md        # Plan de test
└─ test_cases.md       # Cas de test détaillés
└─ beta_testing_guide.md # Guide pour les beta-testeurs
└─ release_process.md   # Processus de release
└─ rollback_procedure.md # Procédure de rollback
```

Structure détaillée du dossier backend/ (pour la Phase 6)

```
backend/          # Code du serveur Node.js
├── server.js      # Point d'entrée du serveur
├── package.json   # Configuration npm
├── config/        # Configuration
│   ├── db.config.js    # Configuration de la base de données
│   └── security.config.js # Configuration de sécurité
├── routes/        # Routes de l'API
│   ├── auth.routes.js
│   ├── taxes.routes.js
│   ├── sync.routes.js
│   └── feedback.routes.js
├── controllers/   # Contrôleurs
│   ├── auth.controller.js
│   ├── taxes.controller.js
│   └── sync.controller.js
├── middleware/    # Middleware
│   ├── auth.middleware.js
│   └── rate-limiter.js
├── models/        # Modèles Sequelize
│   ├── user.model.js
│   ├── tax.model.js
│   └── ministry.model.js
├── services/      # Services
│   ├── sync.service.js
│   ├── delta.service.js
│   ├── encryption.service.js
│   ├── audit.service.js
│   └── release.service.js
└── database/      # Base de données
```

```
|   ├── schema.sql
|   ├── seed.js
|   └── migrations/
└── monitoring/      # Monitoring
    └── alerts.js
```

Structure détaillée du dossier store_assets/ (pour la Phase 8)

```
store_assets/      # Assets pour les stores
|   ├── google_play/      # Assets pour Google Play
|   |   ├── feature_graphic.png
|   |   ├── icon.png
|   |   └── screenshots/
|   ├── app_store/       # Assets pour App Store
|   |   ├── app_icon.png
|   |   └── screenshots/
|   ├── descriptions/    # Descriptions multilingues
|   |   ├── fr.txt
|   |   └── es.txt
|   ├── screenshots/     # Screenshots optimisés
|   |   ├── phone/
|   |   └── tablet/
└── metadata.yaml      # Métadonnées communes
```

Cette structure complète vous servira de guide pour organiser tous les fichiers de votre projet de manière cohérente.