

Chaîne comptable — journaux, ledger et états financiers

Vue d'ensemble

Le moteur Accounting est construit en chaîne. Les journaux représentent les écritures source, le ledger est dérivé de ces journaux, puis la trial balance et les états financiers consomment les soldes produits.

```
flowchart LR
  B[Bills / Forms] --> J[General + Auxiliary Journals]
  J --> L[General Ledger]
  L --> TB[Trial Balance]
  TB --> IS[Income Statement]
  TB --> EQ[Statement of Equity / Retained Earnings]
  TB --> BS[Balance Sheet]
  J --> BR[Bank Reconciliation]
```

Synchronisation automatique du ledger

`utilsByModule.shouldSyncLedger()` active la synchronisation lorsque la préférence `AUTO_REPORT_TO_LEDGER` est active et que la page modifiée fait partie des sources reconnues.

Les pages sources incluent notamment :

- `general_journal`;

- `purchases_journal`;
- `sales_journal`;
- `withdrawals_journal`;
- `cashings_journal`;
- `payroll_journal`.

rebuildLedgerFromJournals()

Le serveur copie d'abord le Module Type Data et évalue les formules, efface uniquement la page dérivée `ledger`, puis rejoue les écritures des journaux en respectant `sectionsIndex` et `rowsIndex`. Les journaux source ne sont pas modifiés.

Les écritures du general journal et des journaux auxiliaires passent par des extracteurs différents avant d'être regroupées dans le ledger.

Quand le ledger est reconstruit

Les custom commands de sauvegarde de valeur, options, formule, suppression/réordonnancement de row/section vérifient si la page impacte le ledger. Si oui, elles reconstruisent le ledger puis envoient un message RTT spécialisé.

Answer Key

L'Answer Key utilise la même logique de reconstruction afin que la version correcte du ledger suive les mêmes règles que le ledger de l'étudiant.

UI Unity

Les contrôles spécialisés incluent notamment `TrialBalanceControl`, `StatementIncomeControl`, `StatementEquityControl`, `BalanceSheetControl` et `BankReconciliationControl`.

Debug d'une incohérence

1. Vérifier la bill/form source.

2. Vérifier l'écriture générée dans le journal.
3. Vérifier si `AUTO_REPORT_TO_LEDGER` est actif.
4. Inspecter le ledger reconstruit.
5. Comparer la trial balance.
6. Seulement ensuite inspecter l'état financier final.

Ajouter une nouvelle source comptable dans la chaîne

Une nouvelle opération comptable doit être branchée à l'endroit le plus en amont possible. Si elle produit une écriture, le ledger et les états doivent la recevoir via la chaîne existante plutôt que via un patch direct sur l'état financier.

Nouvelle écriture dans un journal

1. Définir la source : bill, form, régularisation ou saisie étudiant.
2. Définir debit/credit + comptes.
3. Ajouter la réponse attendue dans l'Answer Key du journal.
4. Ajouter le grading de l'écriture.
5. Si `AUTO_REPORT_TO_LEDGER` doit l'inclure, vérifier `shouldSyncLedger()` et l'extracteur utilisé par `rebuildLedgerFromJournals()`.
6. Vérifier trial balance et états financiers sans code spécial additionnel.

Ne pas patcher le ledger à deux endroits

Le ledger auto-report est une donnée dérivée. Une custom command modifie le journal puis appelle le rebuild. Ajouter en parallèle une mutation manuelle du ledger crée des doublons et des divergences entre Answer Key et décisions étudiant.

Diagnostic par invariants

Invariant cassé	Zone à inspecter
Débit $\neq$ crédit dans l'écriture source	Bill/form/regularization.
Journal bon, ledger incomplet	<code>shouldSyncLedger</code> / extracteur / rebuild.
Ledger bon, trial balance fausse	Account state / signed values.
Trial balance bonne, Income Statement faux	Mapping des comptes vers état.

Invariant cassé	Zone à inspecter
Income Statement bon, Balance Sheet faux	closing/equity/retained earnings.

Revision #2
Created 2026-09-18 19:17:32 UTC by Mathis
Updated 2026-09-18 20:13:44 UTC by Mathis