

CONTEXTE

Dougs, un cabinet d'expertise-comptable, confronté à la nécessité de garantir l'intégrité des données bancaires de ses clients, a identifié un défi crucial lié à la synchronisation des opérations bancaires. La récupération de ces données est essentielle pour la tenue correcte des comptabilités clients, mais elle est souvent entravée par des erreurs dues à des doublons ou des omissions dans les données remontées par les prestataires externes.

L'objectif est de développer une solution capable de valider l'exactitude des opérations bancaires synchronisées avec les relevés bancaires des clients. En utilisant les soldes de fin de période mentionnés dans chaque relevé comme points de contrôle, la solution doit pouvoir identifier et rectifier automatiquement les anomalies telles que les doublons ou les transactions manquantes. Cette validation permettra de minimiser les interventions manuelles des comptables.

SOLUTION API

API de Validation des Mouvements Bancaires

Endpoint : /POST /movements/validation

Request : La requête attendue par cet endpoint comprend deux principaux éléments de données :

- movements : un tableau d'objets représentant les opérations bancaires. Chaque objet contient un id, une date, un label (libellé de l'opération), et un amount (montant de l'opération).
- balances : un tableau de points de contrôle avec date et balance (solde à la date donnée).

Responses :

- 202 Accepted : renvoyé si les données des mouvements et des soldes correspondent sans incohérences.
- 409 Conflict Detected : renvoyé en cas d'anomalies, accompagné d'un objet reasons détaillant les erreurs spécifiques trouvées (par exemple, doublons, opérations manquantes, etc.).

Process de l'API

1. Point d'entrée : les données entrent dans l'API /POST /movements/validation, où elles sont structurées en deux parties principales : les mouvements et les soldes de contrôle.
2. Traitement : Après la réception des données, les fonctions décrites ci-dessus traitent les informations pour calculer les soldes, détecter les doublons et identifier les écarts par rapport aux soldes de contrôle.
3. Sortie : En fonction des résultats du traitement, une réponse est générée et renvoyée à l'utilisateur. Si tout est correct, une réponse 202 Accepted est envoyée; si des anomalies sont détectées, une réponse 409 est envoyée avec des détails explicatifs.

Sorties de l'API

```
{
  "message": "Accepted",
  "isValid": true,
  "reasons": {
    "duplicates": [],
    "balanceErrors": {}
  }
}
```

Cette réponse est renvoyée lorsque tous les mouvements et balances soumis sont validés sans aucune erreur. Cela signifie que :

- Aucun doublon n'a été trouvé parmi les mouvements soumis.
- Aucune erreur de balance n'a été détectée, ce qui implique que les soldes calculés à partir des mouvements correspondent exactement aux balances attendues pour chaque période.

Cette situation est idéale et indique que les données financières soumises sont entièrement cohérentes et fiables.

```
{
  "code": 409,
  "message": "Conflict detected",
  "isValid": false,
  "reasons": {
    "duplicates": [],
    "balanceErrors": {
      "2024-05": {
        "expected": 1200,
        "actual": 1500,
        "difference": 300
      }
    }
  }
}
```

Cette réponse est renvoyée lorsque les soldes calculés à partir des mouvements ne correspondent pas aux balances attendues fournies pour certaines périodes. L'exemple montre une erreur de balance pour le mois de mai 2024, où le solde calculé est supérieur de 300 à ce qui était attendu. Cette situation peut indiquer des transactions non enregistrées, des erreurs de saisie ou d'autres anomalies financières qui nécessitent une enquête plus approfondie.

```

{
  "code": 409,
  "message": "Conflict detected",
  "isValid": false,
  "reasons": {
    "duplicates": [
      {
        "identifiant": "2024-05-29T12:00:00Z:Achat en ligne:150",
        "movements": [
          {
            "id": 1,
            "date": "2024-05-29T12:00:00Z",
            "label": "Achat en ligne",
            "amount": 150
          },
          {
            "id": 2,
            "date": "2024-05-29T12:00:00Z",
            "label": "Achat en ligne",
            "amount": 150
          },
          {
            "id": 4,
            "date": "2024-05-29T12:00:00Z",
            "label": "Achat en ligne",
            "amount": 150
          }
        ]
      }
    ]
  },
  "balanceErrors": {}
}

```

Cette réponse indique un conflit dans les données, spécifiquement des doublons dans les mouvements bancaires. Les détails des mouvements doublonnés sont fournis, y compris leurs identifiants, dates, libellés et montants. Cela permet d'identifier et de résoudre facilement les erreurs de saisie ou de transmission des données.

SOLUTION PROJET

Structure du projet

La solution implémente un service (BankService) et un contrôleur (BankController) dans NestJS. Le service gère la logique de validation tandis que le contrôleur expose cette fonctionnalité via une API REST.

Description des Fonctions Principales

1. BankService

- validateMovements(movements, balances) : Cette fonction prend en entrée des mouvements et des balances, utilise des sous-fonctions pour calculer les doublons et les erreurs de balance, et retourne un objet indiquant si les données sont valides et détaillant les erreurs trouvées.

2. BankController

- movementsValidation(dto, res) : Fonction du contrôleur qui gère la requête HTTP pour la validation des mouvements. Elle appelle validateMovements du service et renvoie une réponse HTTP appropriée.

Traitements des données dans le service

Calcul des soldes mensuels : implémentation d'une fonction pour accumuler les montants des opérations par mois, permettant de comparer ces totaux aux soldes donnés dans les points de contrôle.

- Cette fonction est cruciale pour déterminer le solde à la fin de chaque mois à partir des opérations bancaires. Elle agrège les montants par mois en utilisant une Map où chaque clé est un mois (format 'YYYY-MM') et chaque valeur est le solde total calculé pour ce mois.

Recherche de doublons : une méthode spécifique pour identifier les opérations enregistrées plus d'une fois.

- Pour identifier les doublons dans les données des mouvements, chaque opération est transformée en une chaîne de caractères unique basée sur la date, le libellé et le montant. Cette chaîne sert de clé dans une Map.
- Si une opération avec la même clé est déjà présente dans la Map, elle est considérée comme un doublon. Les opérations identifiées comme doublons sont stockées dans une liste pour une éventuelle révision ou correction.

Analyse des opérations manquantes et excessives : des méthodes pour déterminer si des opérations n'ont pas été enregistrées en excès ou s'il en manque, basées sur les écarts entre les soldes calculés et les soldes contrôlés.

- Cette fonction compare les soldes calculés à ceux fournis dans les points de contrôle. L'écart entre ces soldes indique soit un manque (solde calculé inférieur), soit un excès (solde calculé supérieur).
- Pour chaque mois, si l'écart est négatif, la différence est enregistrée comme opération manquante; si l'écart est positif, elle est enregistrée comme opération excessive. Cela aide à cibler spécifiquement les mois et les montants des ajustements nécessaires. On ne peut évidemment pas déterminer quelles opérations sont manquantes ou en trop.

TESTS

Les tests unitaires ont été écrits pour chaque composant fonctionnel principal :

- Validation de l'absence de doublons.
- Vérification des erreurs de balance.
- Tests de réponses API pour différents cas (données valides, présence de doublons, erreurs de balance).

Le swagger est accessible sur ce endpoint : /api