Documentation des Classes et Méthodes

1. UserService (http://localhost:8085/api/user)

- **Responsabilités** : Gère les opérations CRUD pour l'entité User, telles que la récupération, l'ajout, la suppression et la mise à jour des utilisateurs.
- **Méthodes** :
  - `getAllUsers()` : Récupère tous les utilisateurs.

  - `getUserById(Long id)` : Récupère un utilisateur par son identifiant.
  - `saveUser(User user)` : Sauvegarde un nouvel utilisateur.

  - `deleteUser(Long id)` : Supprime un utilisateur par son identifiant.
  - `updateUser(Long id, User updatedUser)` : Met à jour les détails d'un utilisateur existant.

2. AccountService (http://localhost:8085/api/account)

- **Responsabilités** : Gère les opérations CRUD pour l'entité Account, ainsi que la récupération du solde et des transactions associées.
- **Méthodes** :
  - `getAllAccounts()` : Récupère tous les comptes.
  - `getAccountById(Long id)` : Récupère un compte par son identifiant.
  - `saveAccount(Account account)` : Sauvegarde un nouveau compte.
  - `deleteAccount(Long id)` : Supprime un compte par son identifiant.
  - `updateAccount(Long id, Account updatedAccount)` : Met à jour les détails d'un compte existant.
  - `getAccountSold(Long id)` : Récupère le solde d'un compte par son identifiant.

- `getAccountTransactions(Long id)` : Récupère les transactions associées à un compte par son identifiant.

3. CardService(http://localhost:8085/api/card)

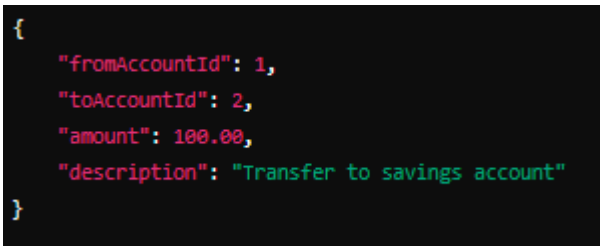
- **Responsabilités** : Gère les opérations CRUD pour l'entité Card, ainsi que la mise à jour de l'état des cartes.
- **Méthodes** :
  - `getAllCards()` : Récupère toutes les cartes.
  - `getCardById(Long id)` : Récupère une carte par son identifiant.
  - `saveCard(Card card)` : Sauvegarde une nouvelle carte.
  - `deleteCard(Long id)` : Supprime une carte par son identifiant.
  - `updateCard(Long id, Card updatedCard)` : Met à jour les détails d'une carte existante.

4. BeneficiaryService(http://localhost:8085/api/benef)

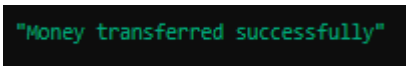
- **Responsabilités** : Gère les opérations CRUD pour l'entité Beneficiary, ainsi que la mise à jour des détails d'un bénéficiaire.
- **Méthodes** :
  - `getAllBeneficiaries()` : Récupère tous les bénéficiaires.
  - `getBeneficiaryById(Long id)` : Récupère un bénéficiaire par son identifiant.
  - `saveBeneficiary(Beneficiary beneficiary)` : Sauvegarde un nouveau bénéficiaire.
  - `deleteBeneficiary(Long id)` : Supprime un bénéficiaire par son identifiant.
  - `updateBeneficiary(Long id, Beneficiary updatedBenef)` : Met à jour les détails d'un bénéficiaire existant.

5. TransactionService(http://localhost:8085/api/transaction)

- **Responsabilités** : Gère les opérations CRUD pour l'entité Transaction, ainsi que les opérations de transfert externes /internes.
- **Méthodes** :
  - `getAllTransactions()` : Récupère toutes les transactions.
  - `transferMoney(Long fromAccountId, Long toAccountId, Double amount, String description)` : Effectue un transfert d'argent interne entre deux comptes.



- `transferExternally(Long ribB, Long ribA, Double amount, String description)` : Effectue un transfert d'argent externe vers un bénéficiaire.



Cette documentation inclut des exemples de requêtes et de réponses pour chaque endpoint, ce qui devrait vous aider à tester et à utiliser les services via Postman de manière efficace dans votre application.