<<Entity>> User

- userId : UUID

username : Stringpassword : Stringemail : StringfirstName : StringlastName : String

<<Entity>>

Account

accountld: UUIDuserld: UUID

- accountNumber: String

- accountType: AccountType (Enum)

- balance: double

- status: AccountStatus (Enum)

- lastTransactionDate: LocalDateTime

<<Entity>> LogEntry

- id: UUID

- message: String

- messageType: MessageType (Enum)

- dateTime: LocalDateTime

<<Entity>>

Transaction

transactionId: UUIDAccountId: UUIDamount: double

- description: String

- status: TransactionStatus (Enum)

- timestamp: LocalDateTime

USER SERVICE <<Entity>> User - userld : UUID - username : String - password : String - email : String - firstName : String - lastName : String <<use>> <<Interface>> UserRepository + findByUsername(username: String): User + existsByUsername(username: String): boolean + existsByEmail(email: String): boolean <<DTO>> <<DTO>> <<DTO>> <<DTO>> <<DTO>> RegisterRequest UserResponse **ErrorResponse** UserLoginResponse LoginRequest - userld: UUID - username: String - userld: UUID - userld: UUID - userld: UUID - password: String - username: String - username: String - username: String - username: String - email: String - firstName: String - lastName: String <<use>> <<use>> <<use>> <<use>>> <<use>> UserService + registerUser(request: RegisterRequest): UserResponse - passwordHash(password:string):string + login(request: LoginRequest): UserLoginResponse + getUserProfile(userId: UUID): UserResponse UserController + register(request: RegisterRequest): ResponseEntity<UserResponse> + login(request: LoginRequest): ResponseEntity<UserLoginResponse> + getProfile(userId: UUID): ResponseEntity<UserResponse>





