Get Wallet Details

User	Frontend	JWTFilter	WalletController	WalletService	SharedAccessRepository	WalletRepository	TransactionRepository	Database

User → **Frontend**: Open wallet details

Frontend → **JWTFilter**: GET /api/wallets/{id} + JWT

JWTFilter → **JWTFilter**: validateToken(jwt)

JWTFilter → **WalletController**: request + userId

WalletController → **WalletService**: getWalletDetails(userId, walletId)

WalletService → **SharedAccessRepository**: findByUserAndResource(userId, 'WALLET', walletId)

SharedAccessRepository → **Database**: SELECT * FROM shared_access WHERE...

Database ⇠ **SharedAccessRepository**: sharedAccess

alt

[No access]

WalletService → **WalletController**: throw ForbiddenException

WalletController ⇠ **Frontend**: 403 Forbidden

Frontend ⇠ **User**: Access denied

[Has access (OWNER or MEMBER)]

par [Parallel queries]

WalletService → **WalletRepository**: findById(walletId)

WalletRepository → **Database**: SELECT * FROM wallets WHERE id = ?

Database ⇠ **WalletRepository**: wallet

WalletService → **TransactionRepository**: findByWalletId(walletId, pageable)

TransactionRepository → **Database**: SELECT * FROM transactions WHERE wallet_id = ? ORDER BY date DESC

Database ⇠ **TransactionRepository**: transactions[]

WalletService → **SharedAccessRepository**: findMembersByWalletId(walletId)

SharedAccessRepository → **Database**: SELECT sa.*, u.email, u.first_name FROM shared_access sa JOIN users u...

Database ⇠ **SharedAccessRepository**: members[]

WalletService → **TransactionRepository**: getWalletSummary(walletId)

TransactionRepository → **Database**: SELECT SUM(CASE WHEN type='INCOME'...) FROM transactions

Database ⇠ **TransactionRepository**: {income, expense, balance}

WalletService → **WalletService**: buildWalletDetailsDTO()

WalletService ⇠ **WalletController**: WalletDetailsDTO

WalletController ⇠ **Frontend**: 200 OK

Frontend ⇠ **User**: Display wallet details