Custom Address API Call Fix - Summary

Problem Fixed

CRITICAL ISSUE: The system was loading the system deposit address from the API even when the user had a custom address configured. This resulted in:

- ❌ Unnecessary API calls to `/api/finance/deposit/spot`
- ❌ System address being loaded (TDKVWv2NmbodhuSzw1NCL37LqUBqvuqf67)
- ❌ Custom address being overridden by system address
- ❌ Duplicate matching logic in component and store

Solution Implemented

Architecture Change

Moved ALL address selection logic from the component to the store, following the single-responsibility principle:

- ☑ **Store**: Handles all business logic (custom vs system address selection)
- ☑ **Component**: Only displays the address from the store

Changes Made

1. `src/stores/user/wallet/deposit.ts`

Updated `fetchDepositAddress()` function to check for custom addresses BEFORE making API calls:

```
fetchDepositAddress: async () => {
  // Get profile from dashboard store
  const profile = useDashboardStore.getState().profile;
  const customAddresses = profile?.customAddressWalletsPairFields || [];

  // ☑ STEP 1: Check for custom address FIRST
  if (Array.isArray(customAddresses) && customAddresses.length > 0) {

    // Priority 1: Exact match (currency + network)
    const exactMatch = customAddresses.find(
      (addr) => addr.currency === selectedCurrency &&
                addr.network === selectedDepositMethod
    );
    if (exactMatch) {
      set({ depositAddress: exactMatch });
      return; // ⚠ SKIP API CALL
    }

    // Priority 2: Currency-only match
    const currencyMatch = customAddresses.find(
      (addr) => addr.currency === selectedCurrency
    );
    if (currencyMatch) {
      set({ depositAddress: currencyMatch });
      return; // ⚠ SKIP API CALL
    }
  }

  // ✅ STEP 2: No custom address - fetch from API
  const { data, error } = await $fetch({ url, silent: true });
  if (!error && data) {
    set({ depositAddress: data });
  }
}
```

Key Features:

- ✅ Checks custom addresses BEFORE API call
- ✅ Priority 1: Exact match (currency + network)
- ✅ Priority 2: Currency-only match
- ✅ Only calls API if NO custom address found
- ✅ Comprehensive console.log debugging

2. `src/components/pages/user/wallet/deposit/DepositAddress/DepositAddress.tsx`

Simplified the component to remove duplicate matching logic:

BEFORE (70 lines of matching logic):

```
useEffect(() => {
  const customAddresses = profile?.customAddressWalletsPairFields;

  if (Array.isArray(customAddresses) && customAddresses.length > 0) {
    let currentAddress = null;

    // Priority 1: Exact match
    if (selectedDepositMethod) {
      currentAddress = customAddresses.find(/* ... */);
    }

    // Priority 2: Currency match
    if (!currentAddress) {
      currentAddress = customAddresses.find(/* ... */);
    }

    // Priority 3: System address
    if (!currentAddress) {
      currentAddress = depositAddress;
    }

    setUserDepositAddress(currentAddress);
  } else {
    setUserDepositAddress(depositAddress);
  }
}, [profile, selectedCurrency, selectedDepositMethod, depositAddress]);
```

AFTER (Simple - 15 lines):

```
useEffect(() => {
  // ✅ Simply use depositAddress from store
  // Store already handles all the logic!
  if (depositAddress) {
    console.log("[DepositAddress] ✅ Using address from store:",
depositAddress);
    setUserDepositAddress(depositAddress);
  }
}, [depositAddress]); // ⚠️ Only depends on depositAddress
```

Benefits:

- ✅ No duplicate logic
- ✅ Single source of truth (store)
- ✅ Simpler dependencies (only `depositAddress`)
- ✅ Easier to maintain and test

Expected Behavior

User WITH Custom Address

```
// Console logs:
[DepositStore] fetchDepositAddress START
[DepositStore] Checking for custom address...
[DepositStore] Custom addresses found, searching for match...
[DepositStore] Priority 1: Looking for exact match (currency + network)...
[DepositStore] ✅ Found exact match: TK5F8mcAHAcgetuhr25ypfQUEqDEUZMEtP
[DepositStore] Skipping API call, using custom address
[DepositStore] fetchDepositAddress END (custom address)

[DepositAddress] useEffect START
[DepositAddress] ✅ Using address from store:
TK5F8mcAHAcgetuhr25ypfQUEqDEUZMEtP
[DepositAddress] useEffect END

// ✅ NO API call to /api/finance/deposit/spot
// ✅ User sees ONLY their custom address
```

User WITHOUT Custom Address

```
// Console logs:
[DepositStore] fetchDepositAddress START
[DepositStore] Checking for custom address...
[DepositStore] No custom addresses configured
[DepositStore] No custom address match, fetching from API...
[DepositStore] API URL: /api/finance/currency/SPOT/USDT/TRC20
[DepositStore] ☑ System deposit address received from API:
TDKVWv2NmbodhuSzw1NCL37LqUBqvuqf67
[DepositStore] fetchDepositAddress END (system address)

[DepositAddress] useEffect START
[DepositAddress] ☑ Using address from store:
TDKVWv2NmbodhuSzw1NCL37LqUBqvuqf67
[DepositAddress] useEffect END

// ☑ API call made (because no custom address)
// ☑ User sees system address
```

Testing Checklist

☑ User with Custom Address

- ☐ Open deposit page
- ☐ Select currency that has custom address (e.g., USDT)
- ☐ Select network (e.g., TRC20)
- ☐ **Verify in console**: NO API call to `/api/finance/deposit/spot`
- ☐ **Verify**: Custom address displayed (TK5F8mcAHAcgetuhr25ypfQUEqDEUZMEtP)
- ☐ **Verify**: NOT system address (TDKVWv2NmbodhuSzw1NCL37LqUBqvuqf67)

☑ User WITHOUT Custom Address

- ☐ Login as user without custom address
- ☐ Open deposit page
- ☐ Select currency (e.g., USDT)
- ☐ Select network (e.g., TRC20)
- ☐ **Verify in console**: API call made to `/api/finance/deposit/spot`
- ☐ **Verify**: System address displayed
- ☐ **Verify**: QR code generated correctly

☑ Custom Address Priority Logic

- ☐ User has custom address for USDT TRC20
- ☐ Select USDT
- ☐ Select TRC20 network
- ☐ **Verify**: Exact match found (currency + network)

- ☐ **Verify**: Custom address displayed

☑ Currency-Only Match

- ☐ User has custom address for USDT (no specific network)
- ☐ Select USDT
- ☐ Select any network
- ☐ **Verify**: Currency match found
- ☐ **Verify**: Custom address displayed

Debug Logs

The implementation includes comprehensive logging for easy debugging:

Store Logs

- `[DepositStore] fetchDepositAddress START/END`
- `[DepositStore] Checking for custom address...`
- `[DepositStore] Priority 1: Looking for exact match...`
- `[DepositStore] Priority 2: Looking for currency-only match...`
- `[DepositStore] Skipping API call, using custom address`
- `[DepositStore] No custom address match, fetching from API...`

Component Logs

- `[DepositAddress] useEffect START/END`
- `[DepositAddress] Using address from store: {address}`

Files Modified

1. ☑ `src/stores/user/wallet/deposit.ts` - Added custom address check before API call
2. ☑ `src/components/pages/user/wallet/deposit/DepositAddress/DepositAddress.tsx` - Simplified to use store address only

Key Benefits

1. ☑ **Performance**: No unnecessary API calls for users with custom addresses
2. ☑ **Correctness**: Users see ONLY their custom address, never system address
3. ☑ **Maintainability**: Single source of truth (store), no duplicate logic
4. ☑ **Debugging**: Comprehensive console logs for easy troubleshooting
5. ☑ **Architecture**: Clean separation of concerns (business logic in store, display in component)

Migration Notes

Before

- ❌ Matching logic in BOTH store and component
- ❌ API always called, even with custom address
- ❌ Component had complex 70-line useEffect

- ❌ Multiple dependencies in useEffect

After

- ☑ Matching logic ONLY in store
- ☑ API called ONLY when needed
- ☑ Component has simple 15-line useEffect
- ☑ Single dependency: `depositAddress`

Related Documentation

- `CUSTOM_WALLET_ADDRESSES_DISPLAY.md` - Original custom address implementation
- `profile.customAddressWalletsPairFields` - User model definition

Status: ☑ Implementation Complete

Next Step: Testing with real users (both with and without custom addresses)