

Deposit Address Matching Logic - Debug & Fix

Проблема

Логіка matching custom addresses не працювала + `TypeError` при спробі читати `depositAddress.address` коли `depositAddress = null`.

Симптоми з логів:


```
// ☒ Custom addresses €:  
[DepositStore] customAddresses: [  
  
  {"address":"TK5F8mcAHAcgetuhr25ypfQUEqDEUZMEtP","currency":"USDT","network":"TRC20"},  
  
  {"address":"TN22swDaEr7vfZcn6UrKBECTkAU1fpHm31","currency":"USDT","network":"TRC20"}  
]  
  
// ☒ Але код каже:  
[DepositStore] No custom addresses configured  
  
// ☒ І потім:  
[DepositStore] Setting depositAddress to null  
  
// ☒ TypeError:  
TypeError: Cannot read properties of null (reading 'address')
```

Зміни

- ☒ Додано детальний debug logging в deposit store

Файл: `src/stores/user/wallet/deposit.ts`

Додано comprehensive logging в функцію `fetchDepositAddress()`:

```
fetchDepositAddress: async () => {  
  const { selectedWalletType, selectedCurrency, selectedDepositMethod } =  
    get();  
  
  //  DETAILED DEBUG: Log parameters with types  
  console.log("[DepositStore DEBUG] selectedCurrency:", selectedCurrency,  
    typeof selectedCurrency);  
  console.log("[DepositStore DEBUG] selectedDepositMethod:",  
    selectedDepositMethod, typeof selectedDepositMethod);  
  console.log("[DepositStore DEBUG] selectedWalletType:", selectedWalletType);
```

```

// Log customAddresses array details
console.log("[DepositStore DEBUG] customAddresses array:", customAddresses);
console.log("[DepositStore DEBUG] customAddresses length:",
customAddresses?.length);
console.log("[DepositStore DEBUG] customAddresses isArray:",
Array.isArray(customAddresses));

// Log each address comparison in detail
customAddresses.forEach((addr, index) => {
  console.log(`[DepositStore DEBUG] Address ${index}:`, {
    address_currency: addr.currency,
    address_currency_type: typeof addr.currency,
    address_network: addr.network,
    address_network_type: typeof addr.network,
    selected_currency: selectedCurrency,
    selected_currency_type: typeof selectedCurrency,
    selected_network: selectedDepositMethod,
    selected_network_type: typeof selectedDepositMethod,
    currency_match: addr.currency === selectedCurrency,
    network_match: addr.network === selectedDepositMethod,
  });
});

console.log("[DepositStore DEBUG] exactMatch result:", exactMatch);
}

```

Що логується:

- Типи даних selectedCurrency, selectedDepositMethod
- Структура customAddresses array
- Кожен custom address з деталями порівняння
- Результат exactMatch і currencyMatch

2. ☒ Виправлено TypeError в DepositAddress component

Файл: `src/components/pages/user/wallet/deposit/DepositAddress/DepositAddress.tsx`

Помилка #1 - Line 202:

```

// ✗ БУЛО:
{userDepositAddress?.info?.tag && (
  <div className="text-muted-400 text-sm">
    {t("Tag")} {depositAddress.info.tag} // ✗ depositAddress
  </div>
)}

// ✓ СТАЛО:
{userDepositAddress?.info?.tag && (
  <div className="text-muted-400 text-sm">
    {t("Tag")} {userDepositAddress.info.tag} // ✓ userDepositAddress

```

```
</div>
}}
```

Помилка #2 - Line 227:

```
// ✗ БУЛО:
onClick={async () => {
  if (contractType === "NO_PERMIT")
    await unlockAddress(depositAddress.address); // ✗ depositAddress може
бути null
  setSelectedDepositMethod(null, null);
  setStep(2);
}}

// ☑ СТАЛО:
onClick={async () => {
  if (contractType === "NO_PERMIT" && userDepositAddress?.address) // ☑
додано null check
    await unlockAddress(userDepositAddress.address); // ☑ використовується
userDepositAddress
  setSelectedDepositMethod(null, null);
  setStep(2);
}}
```

3. ☑ Додано можливість редагування в admin wallet page

Файл: `src/pages/admin/finance/wallet/index.tsx`

```
// ☑ Додано можливість редагувати користувача
<DataTable
  canCreate={false}
  canView={false}
  canEdit={true} // ☑ Додано
  editPath="/admin/crm/user?email=[user.email]" // ☑ Редирект на сторінку
користувача
  columnConfig={columnConfig}
  endpoint={api}
  hasAnalytics
  title={t("Wallets")}
/>
```

Тепер адмін може:

1. Перейти на `/admin/finance/wallet`
2. Побачити всі гаманці і custom addresses
3. Натиснути "Edit" і перейти на сторінку користувача

4. Додати/редагувати `customAddressWalletsPairFields`

Як тестувати

1. Deploy змін на сервер

```
# На сервері:  
git pull  
npm run build  
pm2 restart all
```

2. Перевірити console logs

1. Відкрити браузер з DevTools
2. Перейти на deposit page
3. Обрати USDT
4. Обрати TRC20 network
5. Перевірити console logs:

Що шукати в логах:

```
// 🔍 Перевірити типи даних:  
[DepositStore DEBUG] selectedCurrency: "USDT" string  
[DepositStore DEBUG] selectedDepositMethod: "TRC20" string  
  
// 🔍 Перевірити custom addresses:  
[DepositStore DEBUG] customAddresses array: [...]  
[DepositStore DEBUG] customAddresses length: 2  
  
// 🔍 Перевірити кожен address comparison:  
[DepositStore DEBUG] Address 0: {  
  address_currency: "USDT",  
  address_network: "TRC20",  
  selected_currency: "USDT",  
  selected_network: "TRC20",  
  currency_match: true, // ☒ Має бути true  
  network_match: true, // ☒ Має бути true  
}  
  
// 🔍 Результат:  
[DepositStore DEBUG] exactMatch result: {address: "...", currency: "USDT",  
network: "TRC20"}  
[DepositStore] ☒ Found exact match!
```

3. Можливі причини якщо не працює

Якщо exactMatch НЕ знаходиться, перевірити в логах:

Причина A: selectedDepositMethod порожній

```
[DepositStore DEBUG] selectedDepositMethod: null undefined  
[DepositStore DEBUG] ⚠ selectedDepositMethod is empty/null/undefined!
```

Fix: Перевірити `SelectNetwork.tsx` - чи правильно встановлюється `item.chain`

Причина B: Типи даних не співпадають

```
[DepositStore DEBUG] Address 0: {  
  selected_network: "TRC20",  
  selected_network_type: "string",  
  address_network: "trc20", // ✗ lowercase!  
  address_network_type: "string",  
  network_match: false // ✗  
}
```

Fix: Перевірити case sensitivity в базі даних

Причина C: Whitespace в даних

```
[DepositStore DEBUG] Address 0: {  
  selected_network: "TRC20",  
  address_network: "TRC20 ", // ✗ пробіл в кінці!  
  network_match: false // ✗  
}
```

Fix: Очистити дані в базі або додати `.trim()` в matching logic

4. Додати custom address через admin

1. Перейти на `/admin/finance/wallet`
2. Знайти користувача
3. Натиснути "Edit" (перейде на `/admin/crm/user?email=...`)
4. Знайти поле `customAddressWalletsPairFields`
5. Додати JSON:

```
[  
  {  
    "address": "TK5F8mcAHAcgetuhr25ypfQEqDEUZMEtP",  
    "currency": "USDT",  
    "network": "TRC20"  }  
]
```

```
}  
]
```

Наступні кроки

1. ☒ Deploy змін на сервер
2. ☒ Перевірити console logs в браузері
3. ☒ Знайти причину чому exactMatch не спрацьовує (з детальних логів)
4. ☒ Виправити matching logic якщо потрібно (після аналізу логів)
5. ☒ Протестувати deposit flow end-to-end

Файли змінено

1. `src/stores/user/wallet/deposit.ts` - додано детальний debug logging
2. `src/components/pages/user/wallet/deposit/DepositAddress/DepositAddress.tsx` - виправлено 2 TypeError
3. `src/pages/admin/finance/wallet/index.tsx` - додано можливість редагування

Важливо

Після deploy на сервер, **обов'язково** перевірте console logs в браузері DevTools, щоб побачити:

- Чи `selectedDepositMethod` має правильне значення
- Чи типи даних співпадають
- Чи є case sensitivity або whitespace проблеми
- Що саме повертає exactMatch

З цими детальними логами ми зможемо точно визначити причину проблеми!