Problem 3: Messy React

Contents

First, I'll go through the entire code to edit some of the declared variables. Then I will consider logic and optimization.

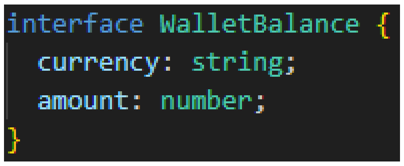
1. EDIT SOME OF THE DECLARED VARIABLES:

1. Add missing properties in the WalletBalance interface:

   Explain:

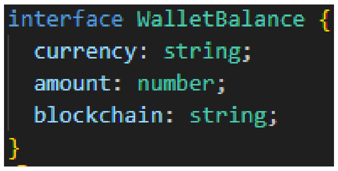
   - After looking through the entire code I noticed that the WalletBalance interface was missing blockchain properties.
   - And with getPriority function below which use parameter blockchain for switch case and case is only get string type.

   → So I guess blockchain is a string data type.

   ```
   interface WalletBalance {
     currency: string;
     amount: number;
   }
   ```

   to

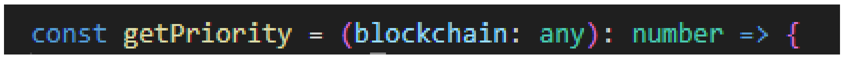
   ```
   interface WalletBalance {
     currency: string;
     amount: number;
     blockchain: string;
   }
   ```

2. Change type of parameter in getPriority function:

   Explain:

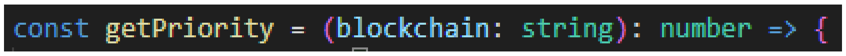
   - getPriority function use parameter blockchain for switch case and case is only get string type.

   → Change blockchain: any to blockchain: string

   ```
   const getPriority = (blockchain: any): number => {
   ```

   To

   ```
   const getPriority = (blockchain: string): number => {
   ```

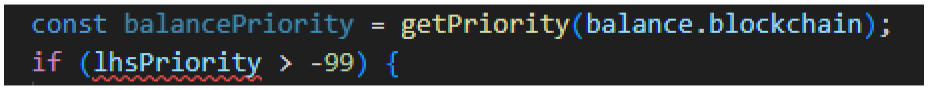
3. Change variable in sortedBalances in filter method:

   Explain:

   - balancePriority is declared but its value is never read and lhsPriority is undefined.

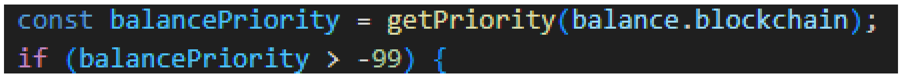
- balancePriority is get value return from getPriority.
- lhsPriority is compare with -99 (which return by default in getPriority)

→ Change lhsPriority to balancePriority



```
const balancePriority = getPriority(balance.blockchain);
if (lhsPriority > -99) {
```

To



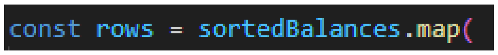
```
const balancePriority = getPriority(balance.blockchain);
if (balancePriority > -99) {
```

4. <u>Change variable in rows:</u>

   Explain:

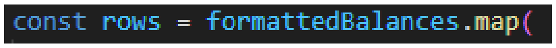
   - In sortedBalances.map, each balance element with FormattedWalletBalance type. But it must be WalletBalance type.
   - In component WalletRow, it get properties formattedAmount but balance in sortedBalances don't have formatted.

   → Change sortedBalances.map to formattedBalances.map



```
const rows = sortedBalances.map(
```

To



```
const rows = formattedBalances.map(
```

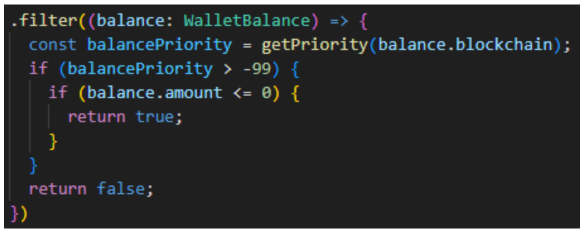
2. LOGIC:

5. <u>Unclear logic in sortedBalance:</u>

   Explain:

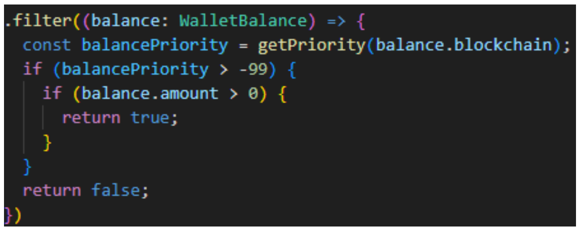
   - <u>Unclear filter method:</u>
     - I think the logic of the filter method is to filter out 'balance' which exist in getPriority and amount > 0

       → So, condition balance.amount should be '> 0' instead of '<=0'



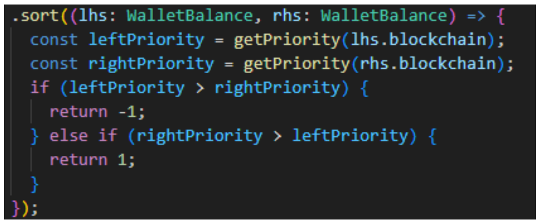
```
.filter((balance: WalletBalance) => {
  const balancePriority = getPriority(balance.blockchain);
  if (balancePriority > -99) {
    if (balance.amount <= 0) {
      return true;
    }
  }
  return false;
})
```

To



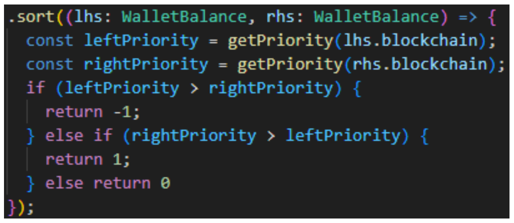
```
.filter((balance: WalletBalance) => {
  const balancePriority = getPriority(balance.blockchain);
  if (balancePriority > -99) {
    if (balance.amount > 0) {
      return true;
    }
  }
  return false;
})
```

- <u>Condition return in sort method:</u>
  - I think the logic of the sort method is sorting obj by a Priority in decreasing order.
  - In cases where leftPriority equals rightPriority. This could lead to unexpected behavior or errors (undefined).
    → Add condition when it equals or use another logic.



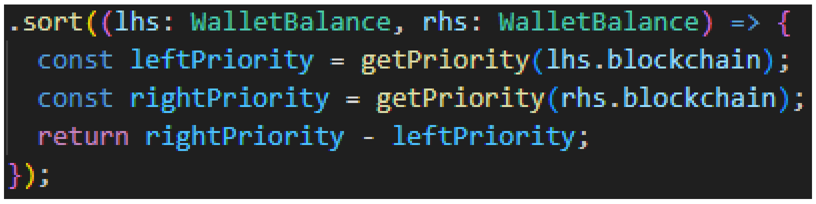
```
.sort((lhs: WalletBalance, rhs: WalletBalance) => {
  const leftPriority = getPriority(lhs.blockchain);
  const rightPriority = getPriority(rhs.blockchain);
  if (leftPriority > rightPriority) {
    return -1;
  } else if (rightPriority > leftPriority) {
    return 1;
  }
});
```

To



```
.sort((lhs: WalletBalance, rhs: WalletBalance) => {
  const leftPriority = getPriority(lhs.blockchain);
  const rightPriority = getPriority(rhs.blockchain);
  if (leftPriority > rightPriority) {
    return -1;
  } else if (rightPriority > leftPriority) {
    return 1;
  } else return 0
});
```

Another logic:



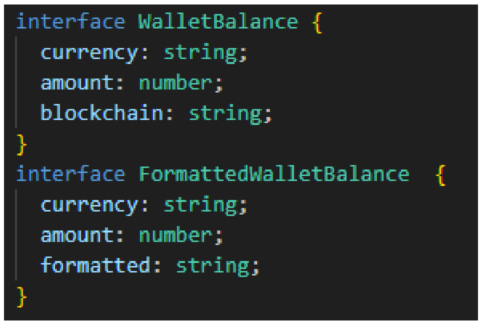
```
.sort((lhs: WalletBalance, rhs: WalletBalance) => {
  const leftPriority = getPriority(lhs.blockchain);
  const rightPriority = getPriority(rhs.blockchain);
  return rightPriority - leftPriority;
});
```

3. OPTIMIZE:

6. <u>Extends Interface:</u>

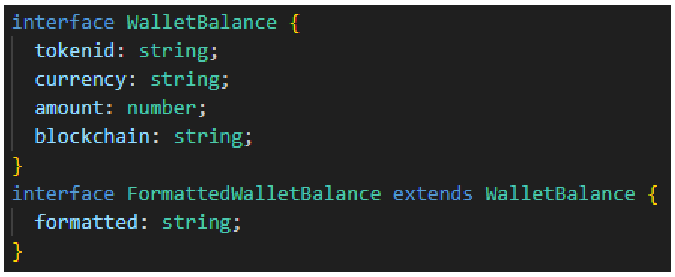
   Explain:

   - formattedBalances is using pread operator to copies all properties from the balance object into the new object.
     → formattedBalances is the same as sortedBalances just adding formatted attribute.



```
interface WalletBalance {
  currency: string;
  amount: number;
  blockchain: string;
}
interface FormattedWalletBalance  {
  currency: string;
  amount: number;
  formatted: string;
}
```

To



```
interface WalletBalance {
  tokenid: string;
  currency: string;
  amount: number;
  blockchain: string;
}
interface FormattedWalletBalance extends WalletBalance {
  formatted: string;
}
```

7. <u>Dependencies in useMemo:</u>

   Explain:

   - The dependency array for useMemo includes prices, but prices is not used in the memoized computation.

     → Consider removing it unless there is a specific reason for its inclusion.

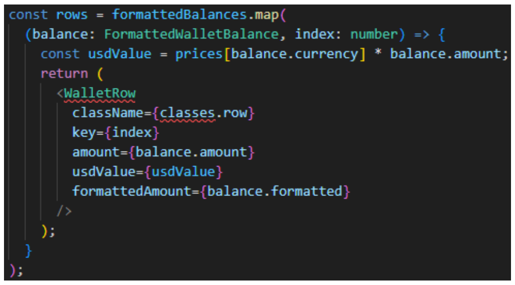
8. <u>Use index as key in map function.</u>

Explain:

- The key is used by React to identify which elements have changed which can help improve the performance.
- Use index as the key can cause issues when the order of the list is changed.

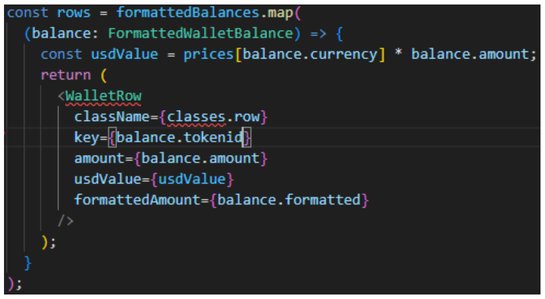
(References: https://react.dev/learn/rendering-lists#why-does-react-need-keys)

→ I guess each token will have unique id (or another unique attribute). I'm adding template tokenid.



```
const rows = formattedBalances.map(
  (balance: FormattedWalletBalance, index: number) => {
    const usdValue = prices[balance.currency] * balance.amount;
    return (
      <WalletRow
        className={classes.row}
        key={index}
        amount={balance.amount}
        usdValue={usdValue}
        formattedAmount={balance.formatted}
      />
    );
  }
);
```

To



```
const rows = formattedBalances.map(
  (balance: FormattedWalletBalance) => {
    const usdValue = prices[balance.currency] * balance.amount;
    return (
      <WalletRow
        className={classes.row}
        key={balance.tokenid}
        amount={balance.amount}
        usdValue={usdValue}
        formattedAmount={balance.formatted}
      />
    );
  }
);
```