

Mapping TMP Tokens to DataCore Cards

After receiving the token list from TMP API, the application should match the response with the previously retrieved DataCore card identifiers.

The mapping should be done using:

```
externalCardId
```

Mapping Rules

- If a card ID is not present in the TMP response, the card has not been added to Apple Wallet.
- If a card ID is present in the TMP response, the card has already been added to Apple Wallet:
 - If `tokenStatus == "ACTIVE"`, the card has already been added and activated.
 - If `tokenStatus == "INACTIVE"`, the card has been added but requires activation.

Swift Example

```
enum WalletCardStatus {
    case notAdded
    case active
    case requiresActivation
}

let tokensByCardId = Dictionary(
    uniqueKeysWithValues: tokens.compactMap { token in
        token.externalCardId.map { ($0, token) }
    }
)

let walletStatuses: [String: WalletCardStatus] = Dictionary(
    uniqueKeysWithValues: cardIds.map { cardId in
        guard let token = tokensByCardId[cardId] else {
            return (cardId, .notAdded)
        }
    }
)
```

```
switch token.tokenStatus {
case "ACTIVE":
    return (cardId, .active)
case "INACTIVE":
    return (cardId, .requiresActivation)
default:
    return (cardId, .notAdded)
}
}
)
```

UI Handling

Based on the resolved wallet status, the application should display the appropriate user action:

- `notAdded` → display the “Add to Apple Wallet” button
- `requiresActivation` → display the “Activate Card” button
- `active` → display information that the card has already been added to Apple Wallet

Revision #1

Created 19 May 2026 07:33:40 by Bartłomiej Jończy

Updated 19 May 2026 07:34:15 by Bartłomiej Jończy