Wizard V2 Implementation Standards

Overview

This document defines the coding and folder structure standards for the Property Wizard V2 implementation. These standards must be followed by all developers and code generation tools when working on the wizard redesign.

Folder Structure Standards

Base Path

All wizard-related code resides under:

src/modules/owner/components/property/wizard/

Version 2 Folder Naming Convention

RULE 1: All new folders must use the `_v2` suffix

When creating new folders for V2 implementation, always append `_v2` to the folder name:

✅ **Correct:**

- `hooks_v2/`
- `utils_v2/`
- `database_v2/`
- `services_v2/`
- `components_v2/`

❌ **Incorrect:**

- `hooks/` (conflicts with existing)
- `hooksV2/` (wrong naming convention)
- `hooks-v2/` (wrong separator)

File Naming Convention

RULE 2: No version numbers in file names

Files within v2 folders should NOT include version numbers in their names:

✅ **Correct:**

- `usePropertyFlow.ts`
- `tempPropertyService.ts`
- `flowNavigation.ts`

❌ **Incorrect:**

- `usePropertyFlow.v2.ts`
- `tempPropertyService_v2.ts`
- `flowNavigation-v2.ts`

Standard Folder Structure for V2

```
src/modules/owner/components/property/wizard/
├── PropertyForm/                # Existing v1 folder (unchanged)
├── hooks/                       # Existing v1 hooks (unchanged)
├── utils/                       # Existing v1 utils (unchanged)
├── services/                    # Existing v1 services (unchanged)
├── components/                  # Existing v1 components (unchanged)
├── hooks_v2/                    # New v2 hooks
│   ├── usePropertyFlow.ts
│   └── useFlowNavigation.ts
├── utils_v2/                    # New v2 utilities
│   ├── flowNavigation.ts
│   └── urlParameterUtils.ts
├── services_v2/                 # New v2 services (non-database)
│   ├── flowStateService.ts
│   └── stepValidationService.ts
├── components_v2/               # New v2 components
│   ├── FlowWrapper.tsx
│   └── StepNavigator.tsx
└── database_v2/                 # New v2 database layer
    ├── services/
    │   ├── tempPropertyService.ts
    │   └── migrationService.ts
    └── migrations/
        └── create_temp_property_listing.sql
```

Implementation Guidelines

When to Create V2 Folders

Create a new v2 folder when:

1. Adding new functionality that doesn't exist in v1

2. Creating alternative implementations of existing functionality
3. Building services that interface with the new database structure

When NOT to Create V2 Folders

Do NOT create v2 folders when:

1. Making minor bug fixes to existing v1 code
2. Adding features that extend existing v1 functionality without breaking changes
3. Working with shared constants or types that both versions use

Code Import Standards

When importing from v2 folders in your code:

```
// Correct - explicit v2 import
import { usePropertyFlow } from '../hooks_v2/usePropertyFlow';
import { tempPropertyService } from '../database_v2/services/tempPropertyService';

// Correct - relative path that clearly shows v2 source
import { flowNavigation } from './utils_v2/flowNavigation';
```

Migration Strategy

During the transition period:

1. V1 and V2 implementations should coexist
2. V2 folders allow for parallel development without breaking V1
3. Once V2 is stable and tested, V1 folders can be gradually deprecated
4. V2 folders can later be renamed to remove the `_v2` suffix when V1 is fully retired

Code Generation Tool Instructions

When generating code for the Property Wizard:

1. **Always check the context** - Are you working on V1 (existing) or V2 (new) implementation?
2. **For V2 work**: Create or use folders with `_v2` suffix
3. **For V1 work**: Use existing folder structure without version suffixes
4. **File names**: Never include version numbers in individual file names
5. **Imports**: Ensure imports reference the correct version of services/utilities

Examples

Correct V2 Implementation

Creating new database service:
📁 database_v2/services/tempPropertyService.ts

Creating new hook:
📁 hooks_v2/usePropertyFlow.ts

Creating new utility:
📁 utils_v2/flowNavigation.ts

Correct V1 Extension

Bug fix in existing service:
📁 services/propertyService.ts (existing file)

Adding to existing hook:
📁 hooks/usePropertyForm.ts (existing file)

Enforcement

These standards are mandatory for:

- All human developers working on the wizard
- All AI/automated code generation tools
- All code review processes
- All documentation and examples

Following these standards ensures clean separation between implementations and smooth migration from V1 to V2.