How to Add a Tab to a Property Flow

Bhoomitalli Real Estate Platform

Overview

This guide provides step-by-step instructions for adding a new tab to an existing property flow in the Bhoomitalli Real Estate Platform.

1. Update Flow Definition in `flows.ts`

First, modify the property flow in the flow definitions file:

```
// src/modules/owner/components/property/wizard/constants/flows.ts

export const FLOW_STEPS = {
  // Existing flow
  COMMERCIAL_RENT: [
    STEP_DEFINITIONS.details,
    STEP_DEFINITIONS.location,
    STEP_DEFINITIONS.rental,  // Add new step here
    STEP_DEFINITIONS.commercial,
    STEP_DEFINITIONS.features,
    STEP_DEFINITIONS.review,
    STEP_DEFINITIONS.photos
  ],

  // Other flows...
};
```

2. Ensure Step Definition Exists in `common.ts`

Confirm that your tab is defined in the common step definitions:

```
// src/modules/owner/components/property/wizard/constants/common.ts

export const STEP_DEFINITIONS = {
  // Existing definitions...
  rental: {
    id: 'rental',
```

```
    title: 'Rental',
    icon: IndianRupee,
    description: 'Rental terms'
  },
  // Other definitions...
};
```

3. Update Navigation Filtering in `FormNavigation.tsx`

Modify the `visibleSteps` filtering to include your new tab:

// src/modules/owner/components/property/wizard/components/FormNavigation.tsx

```
const visibleSteps = useMemo(() => {
  return steps.filter(step => {
    // Common steps
    if (['features', 'photos', 'review'].includes(step.id)) {
      return true;
    }

    // For your target flow - make sure to include the new tab ID
    if (propertyType === 'commercialrent') {
      // Include the new tab 'rental' in the list
      return ['details', 'location', 'rental', 'commercial'].includes(step.id);
    }

    // Other flows...
  });
}, [steps, propertyType]);
```

4. Ensure Content Rendering in `FormContent.tsx`

Update the conditional rendering in FormContent to handle your tab properly:

// src/modules/owner/components/property/wizard/PropertyForm/components/FormContent.tsx

```
if (currentStepId === 'rental') {
  // Modify condition to include the flow where the tab should appear
  if (isSaleMode || isPGHostelMode || isCommercialSaleMode ||
      isCoworkingMode || isLandSaleMode || isFlatmatesMode) {
    return (
      <div className="py--6 text-center">
```

```
      <p className="text-muted-foreground">This section is not applicable for this property
type.</p>
    </div>
  );
}

  // This will now render for both standard rental and commercial rental
  return (
   <RentalDetails
     form={form}
     adType={effectiveAdType}
   />
  );
}
```

5. Handle Step Navigation in `useStepNavigation.ts`

If needed, update the step sequence in the navigation hook:

```
// src/modules/owner/components/property/wizard/PropertyForm/hooks/useStepNavigation.ts

const stepSequences = useMemo(() => ({
 // Make sure the sequence includes your new tab
 commercialRent: ['details', 'location', 'rental', 'commercial', 'features', 'review', 'photos'],

 // Other sequences...
}), []);
```

6. Test the Changes

After making these updates:

1. Run the application
2. Navigate to the flow where you added the tab
3. Verify that the tab appears in the correct position
4. Test navigation to and from the tab
5. Ensure data entry and validation work correctly

Key Tips

1. **Consistent Naming**: Keep step IDs consistent across files
2. **Flow Detection**: Make sure your flow detection logic is accurate

3. **Validation**: Update validation schemas if your new tab has required fields
4. **Debugging**: Use the console logs from FormNavigation for troubleshooting
5. **User Testing**: Always test the complete flow from start to finish

Common Issues

- **Tab not showing**: Check the filtering in FormNavigation.tsx
- **Can't navigate to tab**: Verify the step sequence in flows.ts and useStepNavigation.ts
- **Validation errors**: Ensure validation schemas are updated for new fields
- **Incorrect content**: Check the conditional rendering in FormContent.tsx

By following these steps, you can successfully add a new tab to any property flow in the application.