

BaseEditComponent

August 17, 2026

BaseEditComponent

Kind: Class

Source: Pams/Web/Pams.WebApp/Client/app/core/components/base-edit.component.ts (line 21)

Part of: Pams

`BaseEditComponent` is a shared base class for edit screens in the PAMS web application. It coordinates form construction, data population, validation, save actions, reset and cancel behavior, navigation, and unsaved-change checks.

Implements: `OnInit` , `CanComponentDeactivate`

Methods

Method	Signature	Returns
<code>buildForm</code>	<code>buildForm()</code>	<code>void</code>
<code>populateComponentDataAsync</code>	<code>populateComponentDataAsync()</code>	<code>void</code>
<code>customValidate</code>	<code>customValidate()</code>	<code>void</code>
<code>ngOnInit</code>	<code>ngOnInit()</code>	<code>void</code>
<code>onCancel</code>	<code>onCancel()</code>	<code>void</code>
<code>goBack</code>	<code>goBack()</code>	<code>void</code>
<code>onReset</code>	<code>onReset()</code>	<code>void</code>
<code>canDeactivate</code>	<code>canDeactivate()</code>	<code>`Observable</code>
<code>onSaveandClose</code>	<code>onSaveandClose()</code>	<code>void</code>
<code>onSave</code>	<code>onSave()</code>	<code>void</code>
<code>unloadNotification</code>	<code>unloadNotification(\$event: any)</code>	<code>void</code>
<code>getItem</code>	<code>getItem()</code>	<code>void</code>

Method	Signature	Returns
populateDataForNewItem	populateDataForNewItem()	void
setItemId	setItemId()	void

Properties

Property	Type
id	number
componentName	string
item	T
isLoading	boolean
isSaving	boolean
isSimpleLoad	boolean
isSaveAttempted	boolean
isSelfContained	boolean
closeOnSave	boolean
momentFunction	any
isNewItem	boolean
guidKey	string
errorMessage	string
errorMessageList	string[]
isNewEnabled	boolean
isEditEnabled	boolean
isDeleteEnabled	boolean
FormSaved	any
FormClosed	any
FormDataLoaded	any
ItemDataLoaded	any

Diagram

```
graph LR
  Init[ngOnInit] --> Build[buildForm]
  Init --> Populate[populateComponentDataAsync]
  Populate --> Edit[Edit form data]
  Edit --> Validate[customValidate]
  Validate --> Save[onSave]
  Save --> SaveClose[onSaveandClose]
  Edit --> Reset[onReset]
  Edit --> Cancel[onCancel]
  Cancel --> Back[goBack]
  Edit --> Guard[canDeactivate]
```

Usage

```
import { Component, OnInit } from '@angular/core';
import { BaseEditComponent } from './base-edit.component';

@Component({
  selector: 'app-customer-edit',
  template: `
    <button type="button" (click)="onSave()">Save</button>
    <button type="button" (click)="onSaveandClose()">Save and close</button>
    <button type="button" (click)="onReset()">Reset</button>
    <button type="button" (click)="onCancel()">Cancel</button>
  `
})
export class CustomerEditComponent extends BaseEditComponent implements OnInit {
  override buildForm(): void {
    // Create the customer edit form.
  }

  override async populateComponentDataAsync(): Promise<void> {
    // Load the customer record and assign form values.
  }

  override customValidate(): boolean {
    // Add screen-specific validation before saving.
    return true;
  }
}
```

AI Coding Instructions

- Extend `BaseEditComponent` for edit pages that need the shared form, save, reset, cancel, and navigation flow.

- Override `buildForm()` before reading or assigning form values in `populateComponentDataAsync()` .
- Put page-specific validation in `customValidate()` so `onSave()` and `onSaveandClose()` follow the same validation path.
- Keep `canDeactivate()` aligned with form changes so navigation guards can detect unsaved edits.