

BaseComponent

August 18, 2026

BaseComponent

Kind: Class**Source:** Pams/Web/portal/pams-web/src/app/components/core/base.component.ts (line 8)**Part of:** [Pams Web](#)

`BaseComponent` centralizes keyboard handling, validation, and form submission behavior for portal components. Subclasses can add validation rules through `customValidate()` and respond to validation completion with `onValidated()`, while `isValid()` and `validationError()` expose validation state.

Methods

Method	Signature	Returns
<code>handleKeyboardEvent</code>	<code>handleKeyboardEvent(event: KeyboardEvent)</code>	<code>void</code>
<code>validate</code>	<code>validate(groupNames: string[])</code>	<code>void</code>
<code>validateControl</code>	<code>validateControl(controlName: string, groupNames: string[])</code>	<code>void</code>
<code>isValid</code>	<code>isValid(controlName: string)</code>	<code>boolean</code>
<code>validationError</code>	<code>validationError(controlName: string)</code>	<code>any</code>
<code>onValidated</code>	<code>onValidated()</code>	<code>void</code>
<code>submit</code>	<code>submit()</code>	<code>void</code>
<code>submitForm</code>	<code>submitForm()</code>	<code>void</code>
<code>customValidate</code>	<code>customValidate()</code>	<code>void</code>

Properties

Property	Type
<code>_model</code>	<code>BehaviorSubject<T></code>
<code>loading</code>	<code>boolean</code>
<code>validated</code>	<code>any</code>

Diagram

```
graph LR
  KeyboardEvent[Keyboard event] --> Handle[handleKeyboardEvent()]
  Handle --> Submit[submit()]
  Submit --> SubmitForm[submitForm()]
  SubmitForm --> Validate[validate()]
  Validate --> ValidateControl[validateControl()]
  Validate --> CustomValidate[customValidate()]
  Validate --> OnValidated[onValidated()]
  Validate --> IsValid[isValid()]
  Validate --> ValidationError[validationError()]
```

Usage

```
import { BaseComponent } from 'src/app/components/core/base.component';

export class ProfileFormComponent extends BaseComponent {
  save(): void {
    this.submitForm();

    if (!this.isValid()) {
      console.error(this.validationError());
    }
  }

  customValidate(): void {
    // Add component-specific validation rules here.
  }

  onValidated(): void {
    // React after validation has completed.
  }
}
```

AI Coding Instructions

- Extend `BaseComponent` for components that need shared validation and submission behavior.
- Put component-specific validation rules in `customValidate()` instead of duplicating validation flow in submit handlers.
- Check `isValid()` before acting on submitted data and use `validationError()` when displaying or logging validation failures.
- Keep `onValidated()` focused on post-validation actions; do not bypass `validate()` or `submitForm()` with separate form flows.

Used by

1 reference from 1 file. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (1)

- `LoginFormComponent` — `Pams/Web/portal/pams-web/src/app/components/core/login-form/login-form.component.ts :1`