

DynamicValidationService

August 18, 2026

DynamicValidationService

Kind: Class**Source:** Pams/Web/Pams.Web/Client/app/services/configurations/dynamic-validation/dynamic-validation.service.ts (line 86)**Part of:** Pams

DynamicValidationService manages dynamic validation configuration data in the client application. It retrieves lookup modules, submodules, and properties, and saves or loads mandatory-field settings for a selected module.

Methods

Method	Signature	Returns
getlookupModules	getlookupModules()	void
getlookupSubModules	getlookupSubModules(id: undefined)	void
getProperties	getProperties(id: undefined)	void
saveMandatoryFields	saveMandatoryFields(list: undefined)	void
getMandatoryFieldsByModule	getMandatoryFieldsByModule(moduleId: number)	void

Diagram

```
graph LR
    Component[Configuration Component] --> Service[DynamicValidationService]
    Service --> Modules[getlookupModules]
    Service --> SubModules[getlookupSubModules]
    Service --> Properties[getProperties]
    Service --> Save[saveMandatoryFields]
    Service --> Mandatory[getMandatoryFieldsByModule]
    Modules --> Api[Configuration API]
    SubModules --> Api
    Properties --> Api
```

Save --> Api
Mandatory --> Api

Usage

```
import { Component, OnInit } from '@angular/core';
import { DynamicValidationService } from './dynamic-validation.service';

@Component({
  selector: 'app-mandatory-fields',
  template: ''
})
export class MandatoryFieldsComponent implements OnInit {
  modules: unknown[] = [];
  mandatoryFields: unknown[] = [];

  constructor(
    private readonly dynamicValidationService: DynamicValidationService
  ) {}

  ngOnInit(): void {
    this.dynamicValidationService.getlookupModules().subscribe((modules) => {
      this.modules = modules;
    });
  }

  loadMandatoryFields(module: unknown): void {
    this.dynamicValidationService
      .getMandatoryFieldsByModule(module)
      .subscribe((fields) => {
        this.mandatoryFields = fields;
      });
  }

  saveMandatoryFields(fields: unknown): void {
    this.dynamicValidationService.saveMandatoryFields(fields).subscribe();
  }
}
```

AI Coding Instructions

- Keep calls to `getlookupModules` , `getlookupSubModules` , and `getProperties` in the configuration workflow that selects validation targets.
- Load existing values with `getMandatoryFieldsByModule` before editing mandatory-field configuration.

- Send the payload expected by `saveMandatoryFields` ; preserve the module, submodule, and property identifiers supplied by the UI.
- Handle observable success and error paths in consuming components, especially when saving validation settings.

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)

- `PamsServicesModule` — `Pams/Web/Pams.Web/Client/app/services/pams-services.module.ts :1`