

TemplatesService

August 18, 2026

TemplatesService

Kind: Class

Source: Frontend/src/app/services/configurations/templates/templates.service.ts (line 43)

Part of: Frontend

TemplatesService handles template-related requests in the configuration area of the frontend. It provides methods to fetch, save, update, upload, and download templates and their variable lists.

Methods

Method	Signature	Returns
getAllTemplates	getAllTemplates()	void
getHTMLTemplateByType	getHTMLTemplateByType(templateType: number)	void
saveHTMLTemplate	saveHTMLTemplate(template: HTMLTemplate)	void
updateTemplate	updateTemplate(template: Templates)	void
uploadTemplate	uploadTemplate(templateFile: any)	void
uploadListOfVariable	uploadListOfVariable(templateFile: any)	void
downloadBasicTemplate	downloadBasicTemplate(typeId: number, APIURL: string)	void
downloadUserTemplate	downloadUserTemplate(typeId: number, APIURL: string)	void
downloadListOfVariables	downloadListOfVariables(APIURL: string)	void

Diagram

```
graph LR
  Component[Configuration Component] --> Service[TemplatesService]
  Service --> Templates[Template API]
  Service --> Uploads[Template and Variable Uploads]
  Service --> Downloads[Template and Variable Downloads]
```

Usage

```
import { Component, OnInit } from '@angular/core';
import { TemplatesService } from '../services/configurations/templates/templates.service';

@Component({
  selector: 'app-template-list',
  template: `
    <button (click)="downloadBasicTemplate()">
      Download basic template
    </button>
  `,
})
export class TemplateListComponent implements OnInit {
  templates: unknown[] = [];

  constructor(private readonly templatesService: TemplatesService) {}

  ngOnInit(): void {
    this.templatesService.getAllTemplates().subscribe((templates) => {
      this.templates = templates;
    });
  }

  downloadBasicTemplate(): void {
    this.templatesService.downloadBasicTemplate().subscribe();
  }
}
```

AI Coding Instructions

- Keep template API calls inside `TemplatesService` ; components should call service methods rather than construct HTTP requests directly.
- Follow the existing return-type pattern when consuming service methods, including observable subscriptions where applicable.
- Use `getHTMLTemplateByType()` before editing a template associated with a specific type.

- Send file uploads through `uploadTemplate()` or `uploadListOfVariable()` rather than mixing upload logic into UI components.
- Keep download handling aligned with the response format returned by `downloadBasicTemplate()` , `downloadUserTemplate()` , and `downloadListOfVariables()` .

Used by

4 references from 4 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (4)

- `TemplatesListComponent` — `Frontend/src/app/components/configuration/templates/templates-list/templates-list.component.ts :1`
- `ProcurementDetailsComponent` — `Frontend/src/app/components/procurement/procurement-details/procurement-details.component.ts :1`
- `PurchasingDetailsComponent` — `Frontend/src/app/components/purchasing/purchasing-details/purchasing-details.component.ts :1`
- `NewJobComponent` — `Frontend/src/app/components/sales/new-job/new-job.component.ts :1`