DocumentCreator

August 18, 2026

DocumentCreator

Kind: Class

Source: `Frontend/src/app/components/dashboard/_principalReport/docx-generator.ts` (line 22)

Part of: Frontend

`DocumentCreator` builds a `docx` `Document` for the principal report view. Its `create()` method assembles report sections from paragraph-producing helpers, including contact information, headings, institution details, skills, achievements, and interests.

Methods

Method	Signature	Returns
`create`	`create([experiences, educations, skills, achivements, chart]: undefined)`	`Document`
`createContactInfo`	`createContactInfo(phoneNumber: string, profileUrl: string, email: string)`	`Paragraph`
`createHeading`	`createHeading(text: string)`	`Paragraph`
`createSubHeading`	`createSubHeading(text: string)`	`Paragraph`
`createInstitutionHeader`	`createInstitutionHeader(institutionName: string, dateText: string)`	`Paragraph`
`createRoleText`	`createRoleText(roleText: string)`	`Paragraph`
`createBullet`	`createBullet(text: string)`	`Paragraph`
`createSkillList`	`createSkillList(skills: any[])`	`Paragraph`
`createAchivementsList`	`createAchivementsList(achivements: any[])`	`Paragraph[]`
`createInterests`	`createInterests(interests: string)`	`Paragraph`
`svgToBase64`	`svgToBase64(svg: string)`	`string`

Method	Signature	Returns
`splitParagraphIntoBullets`	`splitParagraphIntoBullets(text: string)`	`string[]`
`createPositionDateText`	`createPositionDateText(startDate: any, endDate: any, isCurrent: boolean)`	`string`
`getMonthFromInt`	`getMonthFromInt(value: number)`	`string`

Diagram

```
graph LR
  A[DocumentCreator] --> B[create(): Document]
  B --> C[createContactInfo]
  B --> D[createHeading]
  B --> E[createSubHeading]
  B --> F[createInstitutionHeader]
  B --> G[createRoleText]
  B --> H[createBullet]
  B --> I[createSkillList]
  B --> J[createAchivementsList]
  B --> K[createInterests]

  C --> L[Paragraph]
  D --> L
  E --> L
  F --> L
  G --> L
  H --> L
  I --> L
  J --> M[Paragraph[]]
  K --> L
  L --> N[docx Document]
  M --> N
```

Usage

```
import { Packer } from "docx";
import { DocumentCreator } from "./docx-generator";

const reportData = {
  // Supply the report data expected by DocumentCreator.
};

const creator = new DocumentCreator(reportData);
const document = creator.create();

const fileBlob = await Packer.toBlob(document);
```

```
const url = URL.createObjectURL(fileBlob);
window.open(url, "_blank");
```

AI Coding Instructions

- Keep document assembly inside `create()` and use the helper methods to produce `Paragraph` instances for each report section.
- Preserve the `createAchivementsList()` method name when calling or modifying it, even though its spelling differs from “achievements.”
- Return `Paragraph[]` from `createAchivementsList()` so multiple achievement entries can be added to the generated document.
- Use `docx` primitives such as `Document` and `Paragraph`; do not return HTML or React elements from this generator.
- Update the report-data mapping passed to `DocumentCreator` when dashboard report fields change.

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)

- `OverallAnalysisYearMap` — `Frontend/src/app/components/dashboard/_principalReport/principal-report.component.ts`:1