

Générateur de documentation

Description : Le générateur automatique de documentation est un outil qui va permettre de créer et d'exporter de la documentation technique en PDF ou en Markdown sur un projet en utilisant le code et ses commentaires à l'aide de Jspdf et File-saver.

Technologies :

- Node.js
- Next.js
- React

Librairies nécessaires :

- File-saver
- Jspdf
- Jest
- Mui

Fonctionnalités MVP :

- Génération de la documentation selon les commentaires.
- Prise en charge des fichiers REACT et JAVASCRIPT.
- Sélection du projet à documenter.
- **Stockage et utilisation UNIQUEMENT en local.**

Fonctionnalités :

- Affichage du nom des fichiers détecter.
- Prise en charge d'autres extensions.
- Sélection des extensions de code à analyser.
- Sélection du format de sortie (PDF, MARKDOWN).

Composants MVP :

- Bouton génération de la documentation (ButtonGenerateDocumentation.jsx)
- Bouton sélection des fichiers (ButtonFilesSelector.jsx)
- Bouton sélection du dossier (ButtonFolderSelector.jsx)
- Zone de saisie du nom du projet (TextBoxProjectName.jsx)

Composants :

- Bouton aide (ButtonHelp.jsx)
- Checkbox format de sortie (CheckBoxOutputFormat.jsx)
- Tableau des fichiers (TableDetectedFiles.jsx)
- Zone de saisie des extensions (TextBoxExtentions.jsx)

Fonctions MVP :

- Génération de la documentation (DocumentationGenerator.js)
- Parsing des fichiers (FileParsing.js)
- Tableau des fichiers (FilesArray.js)
- Dictionnaire des fichiers (FilesArraySelection.js)
- Sélection des fichiers (FilesSelector.js)
- Filtrage des fichiers (FilterFiles.js)
- Sélection du dossier (FolderSelector.js)
- Écriture du .MD (WriteMarkdown.js)
- Écriture du .pdf (WritePDF.js)

Fonctions :

- Capture récursive des fichiers (GetAllFiles.js)
- Nom du fichier PDF (OutputNameFilePDF.js)
- Nom du fichier MD (OutputNameFileMarkdown.js)
- Titre MD (OutputTitleMarkdown.js)
- Titre PDF (OutputTitlePDF.js)
- Tableau des extensions (ToArrayExtensions.js)

Composants :

Bouton de génération de la documentation

Bouton pour générer la documentation à l'aide de la fonction DocumentationGenerator. Au moins un fichier doit être sélectionné, le projet doit avoir un nom et au moins un format de sortie.

Bouton de sélection du dossier

Bouton qui fait appel aux fonctions FolderSelector, FilesArray et FilesArraySelection pour sélectionner un dossier sur le disque et mettre à jour le tableau.

Bouton de sélection des fichiers

Bouton qui fait appel aux fonctions FilesSelector, FilesArray et FilesArraySelection pour sélectionner un ou plusieurs fichiers sur le disque et mettre à jour le tableau.

Checkbox format de sortie

Sélection du format de sortie PDF ou markdown à l'aide de deux checkbox. Met à jour le dictionnaire formats et les booléens des checkbox.

Tableau des fichiers

Tableau avec les noms des fichiers et des extensions afin de vérifier la bonne détection de tous les fichiers sélectionnés.

Zone de saisie du nom du projet

Récupération du nom du projet dans la zone de texte afin de le mettre à jour dans formData. La saisie de ce champ est stockée dans la variable projectName.

Zone de texte (nom des extensions)

Récupération du nom des extensions à analyser pour mettre à jour le formData ainsi que le tableau des fichiers.

Bouton aide

Composant qui affiche un bouton afin d'aider à la compréhension du fonctionnement du générateur.

Fonctions :

FolderSelector ()

Permet de sélectionner le dossier à analyser à l'aide de showDirectoryPicker. Utilise la fonction GetAllFiles pour récupérer de manière récursive les fichiers des sous-dossiers. Retourne un array de fichiers (files) au format filehandle.

FileSelector ()

Permet de sélectionner un ou plusieurs fichiers à analyser à l'aide de showOpenFilePicker. Retourne un array de fichiers (files) au format filehandle.

FilesArray (array files, string extensionsName)

Prend en entrée un array de fichier et le string des extensions. La fonction va transformer les extensions en tableau et retourner un appel à la fonction FilterFiles.

FilterFiles (array files, array extensions)

Prend en entrée l'array des fichiers et l'array des extensions visées afin de retourner l'array des fichiers contenant le ou les extensions ciblées à l'aide de la fonction ToArrayExtensions.

ToArrayExtensions (string extensionsName)

Prend en entrée les extensions séparées par une virgule sous forme de string afin d'en faire un array.

OutputNameFilePDF (string projectName)

Prend en entrée le nom du projet afin de ressortir le nom du fichier de sortie avec '.pdf'.

OutputNameFileMarkdown (string projectName)

Prend en entrée le nom du projet afin de ressortir le nom du fichier de sortie avec '.md'.

OutputTitlePDF (string projectName)

Prend en entrée le nom du projet afin de le réécrire en tant que titre au début du fichier PDF de sortie à l'aide de la librairie jspdf sous la forme « Documentation de projectName ».

OutputTitleMarkdown (string projectName)

Prend en entrée le nom du projet afin de le réécrire en tant que titre au début du fichier Markdown de sortie sous la forme « # Documentation de projectName ».

OutputFormat (boolean checkboxPdf, boolean checkboxMd)

Prend en entrée le booléen de la checkbox PDF et de la checkbox Markdown afin d'en faire un dictionnaire (array) qui servira à DocGenerator. Sert également à vérifier qu'au moins une des deux cases est cocher.

FileParsing (formData, rows)

Prend en entrée l'ensemble des data et les rows triés afin d'analyser l'extension et de choisir la bonne syntaxe des commentaires et des noms de fonctions à extraire. Génère également un dictionnaire avec le nom du fichier, le nom des différentes fonctions ainsi que leurs commentaires associés.

WritePDF (string outputTitlePDF, string outputNameFilePDF, array filesParsed)

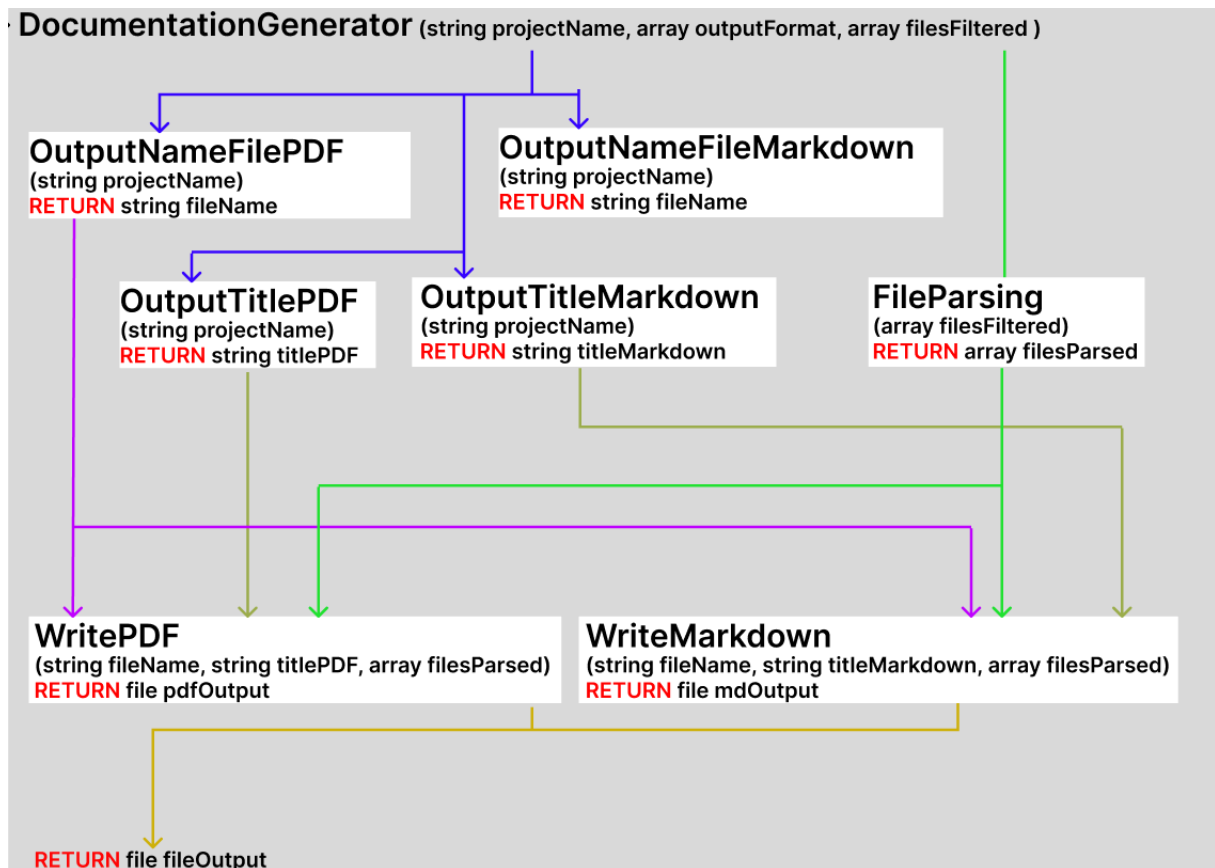
Prend en entrée le titre, le nom du fichier pdf ainsi que le tableau des fichiers triés afin de générer la documentation au format PDF.

WriteMarkdown (string outputTitleMarkdown, string outputNameFileMarkdown, array filesParsed)

Prend en entrée le titre, le nom du fichier markdown ainsi que le tableau des fichiers triés afin de générer la documentation au format Markdown.

DocumentationGenerator (formData, rows)

Prend en entrée l'ensemble du formulaire ainsi que rows afin de générer les fichiers de sorties en réutilisant les différentes fonctions vues ci-dessus (FileParsing, OutputTitlePDF, OutputNameFilePDF, WritePDF, OutputTitleMarkdown, OutputNameFileMarkdown, WriteMarkdown).



Interface Utilisateur

Générateur de documentation

?

SÉLECTION DU DOSSIER

SÉLECTION DES FICHIERS

?

?

?
Format de sortie :
☐ PDF
☐ Markdown

GÉNÉRER LA DOCUMENTATION

Files	Extensions

SCHEMA

Générateur de documentation

Outil permettant la génération automatique de documentation en fonction des commentaires et des liens entre les différentes méthodes

