JSDoc

Documentation Tool

Prepared by:
Samia Alam ,Abdus Salam

Contents

1 Introduction

1.1) What is Documentation tool ?

A documentation tool is used to generate organized and structured documentation from source code, making it easier for developers to understand and maintain projects. It automatically extracts comments and annotations from code to create readable guides, manuals, or technical documents. Popular examples include JSDoc for JavaScript and Doxygen for C++.

1.2) What is JSDcos ?

JSDoc is a documentation generator for JavaScript that allows developers to create documentation from their JavaScript source code comments. It provides a simple way to add inline documentation that can be extracted to generate HTML files or other formats, which makes it easier for developers and teams to understand, use, and maintain the code.

1.2) Key Features

a) Supports documenting functions, classes, methods, variables, and objects.
b) Works with most modern JavaScript (ES6+) features, including classes and modules.
c) Supports type annotations for better tooling and understanding, helping with type safety.
d) Can generate HTML pages for easy viewing of documentation.

For more information check here
`https://jsdoc.app/about-getting-started`

2 Installation and Usage

2.1 Installation:

1) Install Node.js if it's not already installed on your system. You can download it from the

`https://nodejs.org/en/download/package-manager`

2) Install JSDoc globally using this command:

```
npm install -g jsdoc
```

3) Or use the following command to install it for a single project:

```
npm install --save-dev jsdoc
```

For better understanding click here

`https://www.youtube.com/watch?v=Nqv6UkTROak`

2.2 How JSDoc Works:

JSDoc works by reading special comment blocks in your JavaScript code and then converting these comments into structured documentation.

example:

```
/**
 * Adds two numbers together.
 * @param {number} a - The first number.
 * @param {number} b - The second number.
 * @returns {number} The sum of `a` and `b`.
 */
function add(a, b) {
  return a + b;
}
```

2.3 JSDocs Tags

- @param - Describes a parameter of a function.
- @returns - Describes what a function returns.
- @type - Describes the type of a variable.
- @class - Marks a comment for a class.
- @constructor - Marks a constructor method
- @throws - Describes exceptions thrown by the function.

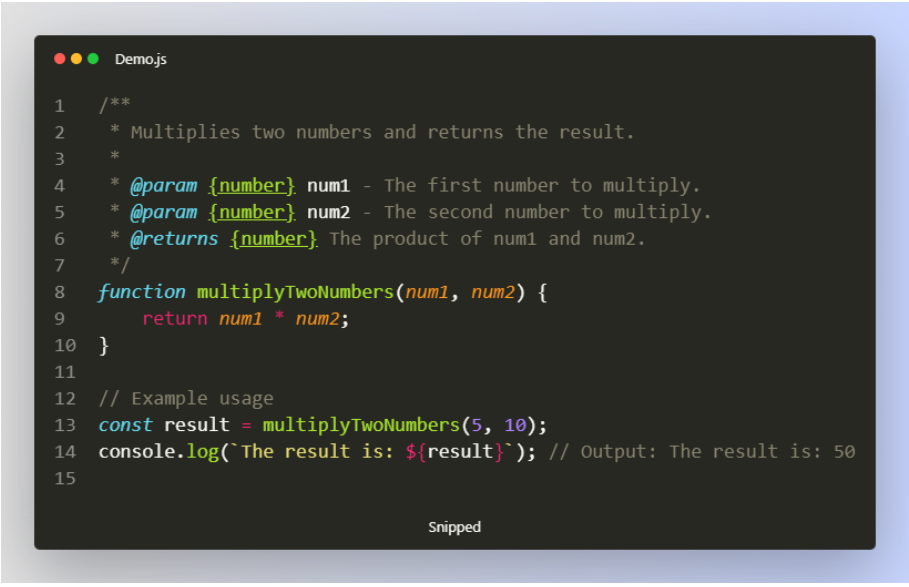
2.3 A quick guide for generating jsdoc documentation file:

Step 1: Environment Setup:

- Install node.js if it is not installed
- Install jsdoc by using this command

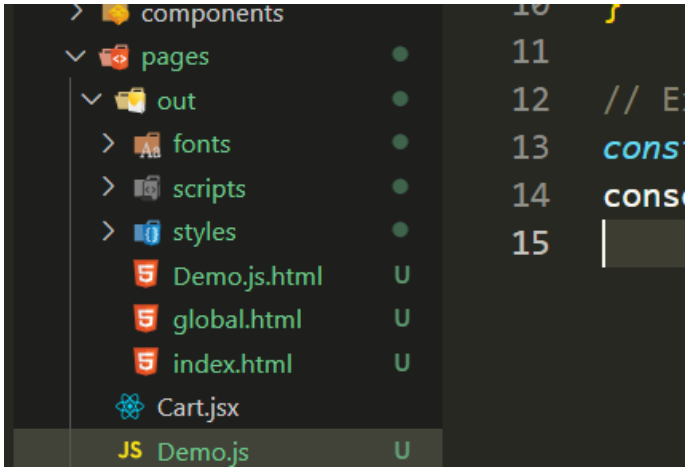
```
npm install -g jsdoc
```

Step 2: Create Your File(javascript) and Add comment like this:

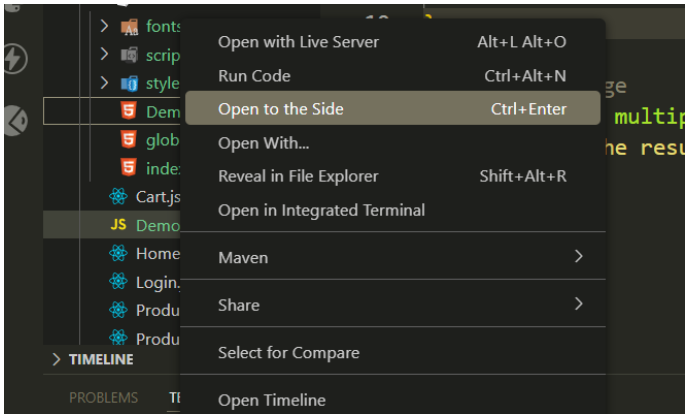


Step 3: Enter the file by this command and after that the folder structure below :

note:(Enter the folder where the file is created)
jsdoc Demo.js



Step 4:Right Click the file and Click on live server option:



After completing all steps our final output is:

3 Conclusion

3.1) Why we choose JSDoc as frontend and backend Documentation tool ?

JSDoc is a javascript-based documentation tool. As our frontend and backend both use js framework(react and node.js) it is easy for us to use and manage code easily.This is also comfortable for us.

3.2) Advantages

- **Improved Code Readability:** Inline comments with standard formats help developers and team members understand the functionality of the code better.
- **Supports Modern JavaScript:** JSDoc can work with ES6+ classes, modules, and other advanced features.
- **Type Checking and Tooling:** When paired with editors like VSCode, JSDoc annotations can provide in-line type checking and autocompletion, making development smoother.
- **Easy Documentation Generation:** Automatically generates well-structured and navigable HTML documentation.
- **Free and Open-Source:** JSDoc is free to use and open-source, which makes it accessible to all developers.

3.3) Disadvantages

- **Manual Maintenance:** Developers need to manually update the comments every time the code changes. Otherwise, the generated documentation might become outdated.
- **Limited Beyond JavaScript:** While primarily designed for JavaScript, it doesn't support languages outside of the JavaScript ecosystem as well as tools like Doxygen or Sphinx do.
- **Learning Curve:** For developers not familiar with documenting code, learning all the JSDoc tags can take time, especially in large codebases.
- **Not a Replacement for Full Comments:** JSDoc should be used alongside regular comments, as it focuses more on structure and metadata than explaining complex logic.

Credits

https://jsdoc.app/howto-es2015-classes

THANK YOU