SWAGGER

Set of open-source tools around OpenAPI that can be used to design,build,document and consume REST API.

Major tools:

- Swagger Editor – browser-based editor where you can write OpenAPI definitions.
- Swagger UI – renders OpenAPI definitions as interactive documentation.
- Swagger Codegen – generates server stubs and client libraries from an OpenAPI definition.
- Swagger Editor Next (beta) – browser-based editor where you can write and review OpenAPI and AsyncAPI definitions.
- Swagger Core – Java-related libraries for creating, consuming, and working with OpenAPI definitions.
- Swagger Parser – standalone library for parsing OpenAPI definitions
- Swagger APIDom – provides a single, unifying structure for describing APIs across various description languages and serialization formats.

Use Swagger API in NodeJs:

- Install swagger: *npm install swagger-jsdoc--save*
- *npm install swagger-ui-express –save*

Project Title: CRUD Swagger

Scope: Implement CRUD operations using Swagger for API documentation.

Key Components:

- Swagger: Used for API documentation and testing.
- CRUD Operations: Create, Read, Update, delete operations for interacting with data.
- API Endpoints: Define routes for different CRUD operations.

Specification:

- Use Swagger to document API endpoints for CRUD operations.
- Implement controllers for handling CRUD operations.
- Define models for data representation.
- Use appropriate HTTP methods for each CRUD operation.

Usage:

- Access Swagger UI to view and test API endpoints.
- Perform CRUD operations using the defined API routes.

Directory Structure:

Swagger Configuration File:

- File: swagger.js
- Description: Configures Swagger documentation for the API.
- Functionality: Sets up Swagger options with API metadata, servers, and paths to API routes.

Swagger UI File:

- File: index.js
- Description: Entry point of the CRUD API server.
- Functionality: Connects to MongoDB, sets up Swagger documentation, defines middleware, routes, error handling, and starts the server.

Model File:

- File: product.js
- Description: Defines the schema for the product model.
- Functionality: Specifies the structure of a product including name, brand, RAM, camera, network, fingerprint, and price.

Route File:

- File: products.js
- Description: Defines API endpoints for CRUD operations on products.
- Functionality: Handles GET, POST, PUT, DELETE requests for products, interacts with the database, and sends appropriate responses.