Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №3

ДІАГРАМА РОЗГОРТАННЯ. ДІАГРАМА КОМПОНЕНТІВ. ДІАГРАМА ВЗАЄМОДІЙ ТА ПОСЛІДОВНОСТЕЙ. JSON Tool

Виконав
студент групи ІА–22:
Щаблевський Д. Е.
Перевірив:
Мягкий Михайло Юрійович

Київ 2024

Зміст

Тема: Діаграма розгортання. Діаграма компнентів. Діаграма взаємодій та послідовностей.

Мета: Проаналізувати тему. Розробити діаграму розгортання, діаграму компонентів та діаграму послідовностей для проектованої системи.

Хід роботи

..28 JSON Tool (ENG) (strategy, command, observer, template method, flyweight)

Display JSON schema with syntax highlight. Validate JSON schema and display errors. Create user friendly table\list box\other for read and update JSON schema properties metadata (description, example, data type, format, etc.). Auto save\restore when edit, maybe history. Can check JSON value by schema (Put schema and JSON = valid\invalid, display errors). Export schema as markdown table. JSON to "flat" view.

Теоретичні відомості

У лабораторній роботі 3 розглядається створення UML-діаграм, які використовуються для моделювання фізичного розгортання, компонування та взаємодії елементів програмної системи.

1. **Діаграма розгортання (Deployment Diagram)** відображає фізичне розташування програмного забезпечення на апаратних вузлах. Основними елементами є вузли, які представляють фізичні пристрої або середовища виконання. Вузли з'єднуються зв’язками, що демонструють шляхи передачі даних.
2. **Діаграма компонентів (Component Diagram)** ілюструє розподіл системи на окремі модулі чи компоненти. Вона показує зв’язки між компонентами, сприяючи структуризації коду та забезпечуючи можливість його повторного використання.

3. **Діаграма послідовностей (Sequence Diagram)** моделює хронологію взаємодії об'єктів. Вона відображає, які об'єкти беруть участь у взаємодії та в якому порядку передаються повідомлення між ними.

Ці діаграми дозволяють детально вивчити архітектуру системи, оптимізувати її структуру та спланувати ефективне розгортання і подальший розвиток.

Реалізація частини функціональної системи

Рисунок 1. – Реалізована частина проекту

Реалізовані компоненти для редагування JSON:

- editor-base.component.ts
  Базовий компонент, який служить основою для редактора.
- json-editor/
  - json-editor.component.html
    HTML-шаблон компонента JSON-редактора.
  - json-editor.component.ts
    Типовий файл для логіки компонента JSON-редактора.
- json-toolbar/
  Директорія для панелі інструментів редактора JSON.
- json-tool.component.html
  Головний HTML-шаблон для JSON-інструменту.
- json-tool.component.ts
  Головний файл логіки для JSON-інструменту.

Діаграма послідовностей

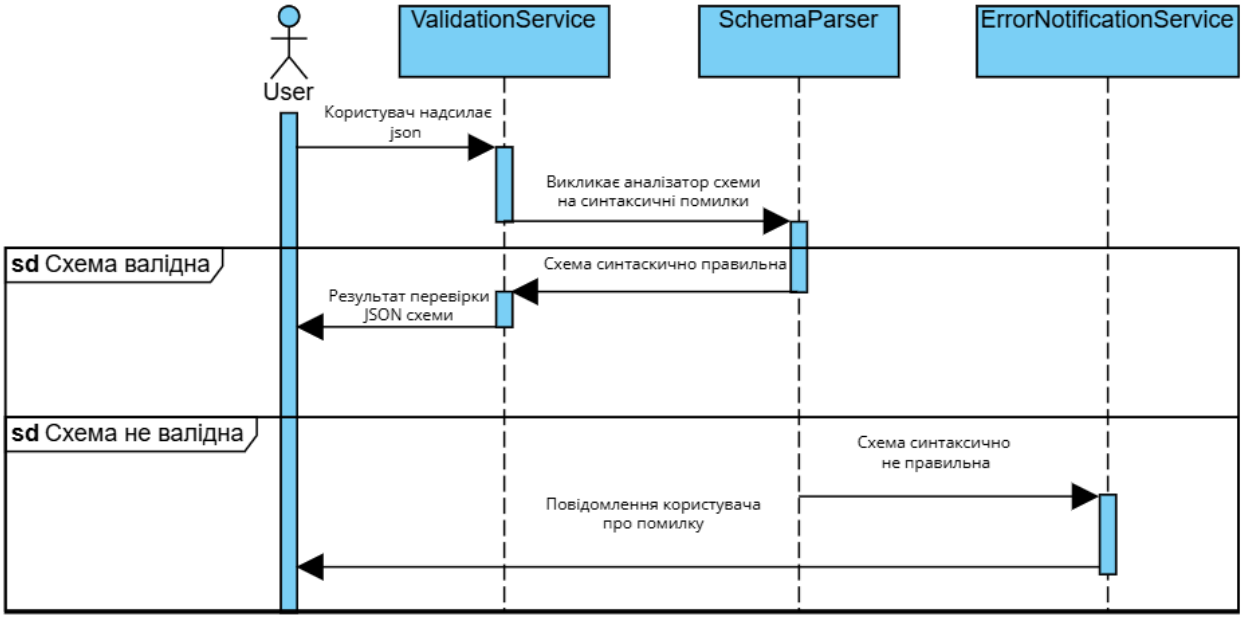

Рисунок 2. - Діаграма послідовностей

1. Початкові дії

- **Користувач** надсилає JSON до системи для перевірки.
- **ValidationService** отримує запит і передає дані до **SchemaParser** для перевірки синтаксичної коректності схеми.

2. Сценарій "Схема валідна"

- **SchemaParser** перевіряє схему і виявляє, що вона **синтаксично правильна**.
- **ValidationService** отримує результат перевірки й надсилає **користувачу** повідомлення про успішну перевірку JSON схеми.

Результат: Користувач отримує підтвердження, що JSON відповідає коректній схемі.

3. Сценарій "Схема не валідна"

- **SchemaParser** виявляє, що схема містить синтаксичні помилки.
- **ValidationService** викликає **ErrorNotificationService**, який створює **повідомлення про помилку**.
- Повідомлення надсилається **користувачу**, щоб проінформувати про некоректність схеми.

Результат: Користувач отримує детальне повідомлення про наявність синтаксичної помилки у схемі.

Діаграма розгортання

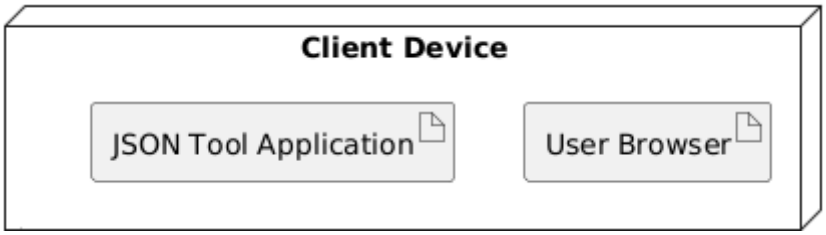

Рисунок 3. - Діаграма розгортання

Client Device (Клієнтський пристрій):

Містить два основні компоненти:

Додаток для роботи з JSON: додаток, який дозволяє користувачу обробляти JSON-схеми.

Браузер користувача: інтерфейс для взаємодії з додатком на клієнтському пристрої.

Діаграма компонентів

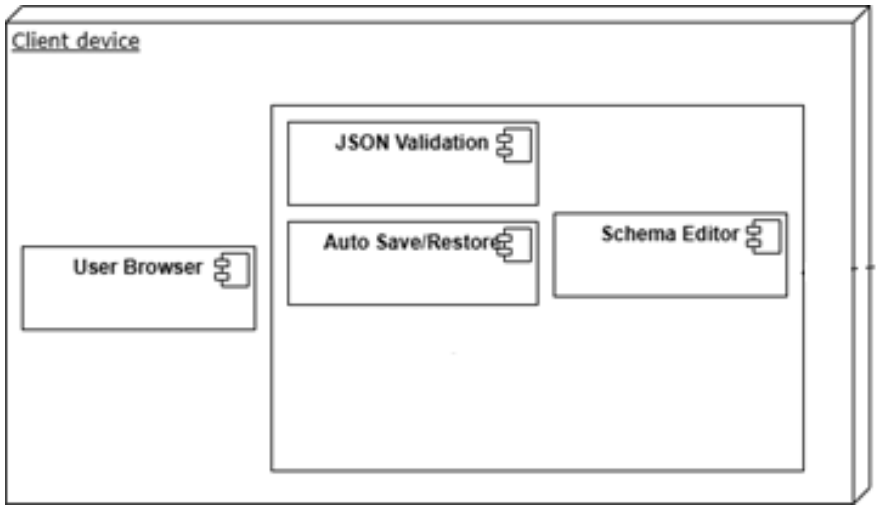

Рисунок. 4. – Діаграма компонентів

Опис діаграми компонентів:

На діаграмі компонентів зображено архітектуру взаємодії між клієнтським пристроєм та сервером у проєкті **JSON Tool**.

1. **Клієнтський пристрій**:
    - **User Browser**: інтерфейс для взаємодії користувача із JSON Tool.
    - **JSON Validation**: компонент, що перевіряє JSON на відповідність заданій схемі та виводить помилки.
    - **Auto Save/Restore**: автоматично зберігає зміни в JSON-схемі та відновлює їх за потреби.
    - **Schema Editor**: основний компонент для редагування JSON-схем та їх властивостей (опис, приклади, формат тощо).

Посилання на репозиторій проекту:

https://github.com/neodavis/json-tool

Висновок: В данній лабораторній роботі я розробив діаграми розгортання, компонентів та діаграми взаємодій та послідовностей.