

Генерація документації для C#

1. XML-документація в C#

1. Увімкнення XML-документації:

- У файлі `.csproj` додайте:

```
<GenerateDocumentationFile>true</GenerateDocumentationFile>
```

- Це створює XML-файл (наприклад, `MyProject.xml`) під час збірки.

2. Написання XML-коментарів:

- Використовуйте `///` з тегами, такими як `<summary>`, `<param>`, `<returns>`.
- Приклад:

```
/// <summary>Обчислює суму двох цілих чисел.</summary>  
/// <param name="a">Перше ціле число.</param>  
/// <param name="b">Друге ціле число.</param>  
/// <returns>Сума цілих чисел.</returns>  
public int Sum(int a, int b) => a + b;
```

3. Збірка:

- Виконайте `dotnet build`, щоб створити XML-файл.

2. Використання DocFX

DocFX генерує статичний сайт із XML-коментарів та Markdown-файлів.

Налаштування

- Встановлення: `dotnet tool install -g docfx`
- Ініціалізація: У папці `docs` виконайте `docfx init -q`, щоб створити `docfx.json` та `toc.yml`.

Кроки

1. Налаштування `docfx.json`:

```
{  
  "metadata": [{ "src": [ "../MyProject.csproj" ], "dest": "api" }],
```

```
"build": { "content": ["api/**/*.yaml", "articles/**/*.md"], "dest": "_site" }
}
```

2. Налаштування змісту (ТОС):

- Відредагуйте `toc.yaml`, щоб замінити стандартний ТОС на власний.
- Приклад `toc.yaml`:

```
- name: Генерація документації
  href: articles/generate_docs.md
- name: Літинг
  href: articles/linting.md
- name: Довідка API
  href: api/index.yaml
```

- Переконайтеся, що файли (наприклад, `generate_docs.md`, `linting.md`) є в папці `articles/`.

3. Додавання Markdown-файлів:

- Розмістіть `generate_docs.md` та `linting.md` у папці `articles/`.
- Перевірте, чи шляхи в `toc.yaml` відповідають розташуванню файлів.

4. Генерація:

- Виконайте `docfx docfx.json` для створення метаданих та обробки ТОС.

5. Попередній перегляд:

- Виконайте `docfx docfx.json --serve` та відкрийте `http://localhost:8080`, щоб перевірити ТОС.

6. Збірка:

- Виконайте `docfx build docfx.json`, щоб створити сайт у папці `_site`.

3. Видалення DocFX

Якщо DocFX більше не потрібен, виконайте наступні дії для його видалення:

1. Видалення глобальної установки:

- Якщо DocFX встановлено глобально, видаліть його командою:

```
dotnet tool uninstall -g docfx
```

2. Видалення локальної установки:

- Якщо DocFX додано як залежність у проєкті, видаліть його з `.csproj`:

```
dotnet remove package DocFX
```

3. Очищення файлів DocFX:

- Видаліть папку `docs` (або іншу, де ініціалізовано DocFX) разом із файлами `docfx.json`, `toc.yml`, `articles/`, `api/`, `_site/` тощо.

4. Перевірка:

- Переконайтеся, що команда `docfx --version` не повертає результату (для глобального видалення).

Як інструмент було обрано SonarQube, через його підтримку C#, зручний інтерфейс та швидкість налаштування.

Лінтинг з SonarQube

Цей документ описує, як запустити SonarQube через Docker для аналізу якості коду.

Передумови

- Встановлений Docker і Docker Compose.
- Інструмент збірки проєкту (Maven, Gradle, npm тощо).

Запуск SonarQube через Docker

1. Створіть файл `docker-compose.yml`:

```
version: '3.8'
services:
  sonarqube:
    image: sonarqube:community
    container_name: sonarqube
    depends_on:
      - db
    environment:
      - SONAR_JDBC_URL=jdbc:postgresql://db:5432/sonar
      - SONAR_JDBC_USERNAME=sonar
      - SONAR_JDBC_PASSWORD=sonar
    ports:
      - '9000:9000'
    volumes:
      - sonarqube_data:/opt/sonarqube/data
      - sonarqube_logs:/opt/sonarqube/logs
    networks:
      - sonarnet
  db:
    image: postgres:13
    container_name: sonarqube_db
    environment:
      - POSTGRES_USER=sonar
      - POSTGRES_PASSWORD=sonar
      - POSTGRES_DB=sonar
    volumes:
      - sonarqube_db_data:/var/lib/postgresql/data
    networks:
      - sonarnet
```

```
networks:
  sonarnet:
    driver: bridge
volumes:
  sonarqube_data:
  sonarqube_logs:
  sonarqube_db_data:
```

2. Запустіть контейнери:

```
docker-compose up -d
```

3. Відкрийте SonarQube: <http://localhost:9000>.

4. Увійдіть (логін: `admin`, пароль: `admin`) і змініть пароль.

Налаштування проєкту

1. Створіть файл `sonar-project.properties` у корені проєкту:

```
sonar.projectKey=my-project-key
sonar.projectName=Мій Проєкт
sonar.sources=src
sonar.host.url=http://localhost:9000
sonar.login=<Ваш-Токен>
```

- Замініть `<Ваш-Токен>` на токен із SonarQube (`Account > Security`).

2. Встановіть SonarQube Scanner CLI та додайте до PATH.

3. Запустіть аналіз:

```
sonar-scanner
```

Перегляд результатів

- У браузері: Відкрийте <http://localhost:9000> для звітів.
- У IDE: Встановіть плагін SonarQube для IDE (SonarLint) для миттєвого аналізу.

Корисні команди

- Переглянути логи:

```
docker logs sonarqube
```

- Зупинити контейнери:

```
docker-compose down
```