Котвицький О. В., магістрант, гр. ІПЗм-19-1,
Сергутін С. О., асистент кафедри КН
Державний університет «Житомирська політехніка»

СИСТЕМА АВТОМАТИЗАЦІЇ ГЕНЕРАЦІЇ ДОКУМЕНТАЦІЇ ПРОЄКТУ

WEB-проєкти на високорівневих мовах програмування, таких як Java або C#, які використовують Enterprise специфікації (наприклад Java EE), мають доволі громіздку структуру та відзначаються підвищеною складністю для розуміння того, які функції виконують ті чи інші методи. Зокрема, якщо проєкт реалізує набір API (Application Programming Interface) методів для сторонніх інтеграцій, такий набір методів потріб-но описати в документації для того, щоб користувачі, які будуть використовувати даний набір методів, розуміли які функції кожен з них виконує.

Опис методів зручно публікувати на окремій WEB-сторінці для легкого доступу для користувачів та розробників. Опис методів можна задавати вручну, але, якщо проєкт великий за об'ємом, кількість методів може рахуватись сотнями або ж тисячами.

Саме тому існує необхідність автоматизувати процес генерації документації. Опис методів зручніше зберігати у вихідному коді, безпосередньо там, де метод оголошується, а його параметри та тип даних, який він повертає – розпізнавати та описувати автоматично.

Отже, система має складатися з таких компонентів:

1. Модуль обробки метаінформації методів та класів
2. Модуль генерації веб-сторінки (наприклад JsDuck)
3. Система CI/CD(наприклад TeamCity)
4. Система контролю версій (наприклад Git)
5. Система контейнеризації (Docker)

Розглянемо кожен пункт детальніше:

Для реалізації модуля обробки метаінформації методів та класів мова програмування, на якій написано проєкт, обов'язково має підтримувати рефлексію (Reflection). Рефлексія – це процес, під час якого програма може відслідковувати і модифікувати власну структуру і поведінку під час виконання. Зокрема важливою є можливість пошуку і модифікації конструкцій початкового коду (методів та класів). В мові програмування Java – це використання анотацій, в мові програмування C# – атрибутів. В даних синтаксичних конструкціях ми задаємо метаінформацію методів або класів. Це, наприклад, чи є метод застарілим (Deprecated), чи описувати метод в документації для всіх користувачів проєкту (public), чи лише для розробників (private). Звісно, для обробки даної метаінформації має існувати модуль, який зчитує дану метаінфор-мацію та має алгоритм її обробки.

Модуль генерації веб-сторінки відповідає за створення HTML, CSS, JavaScript файлів WEB-сторінки. Можна використовувати готові рішення, наприклад JSDuck. Модуль обробки метаінформації методів та класів надає контент для WEB-сторінки.

Генерувати документацію зручно за допомогою CI/CD інструментів, адже вони дозволяють налаштувати готові кроки та зробити їх повторюваними, що забезпечить періодичну генерацію документації і розмежування її по версіям (наприклад для кожної версії проєкту – своя документація).

Згенеровану документацію потрібно десь зберігати та оновлювати. Для цього зручно використовувати системи контролю версій Git. До того ж, Git зручно інтегрувати в CI/CD інструменти.

Щоб отримати готову до розгортання WEB-сторінку з налаштованими шляхами, маршрутизацією URL та іншим – зручно створити Docker контейнер з веб-сервером Apache та контентом WEB-сторінки. За створення такого контейнера відповідає CI/CD інструмент.

Такий контейнер можна розгорнути на хостингу або хмарній платформі.

Таким чином, наведені вище кроки направлені на максимальну автоматизацію процесу та зменшення ризику людської помилки під час створення документації. Основні переваги даного підходу:

1. Документація генерується та заповнюється базуючись на вихідному коді.
2. Забезпечено автоматизацію процесу генерації за допомогою CI/CD інструментів.
3. Забезпечено розділення версій документації.
4. Забезпечена можливість розгортання готової WEB-сторінки разом з WEB-сервером за допомогою технології Docker.

Список літератури

1. Використання рефлексії на мові програмування Java [Електронний ресурс] – режим доступу до ресурсу: https://www.geeksforgeeks.org/reflection-in-java/.
2. Офіційний сайт Docker [Електронний ресурс] – режим доступу до ресурсу: https://github.com/senchalabs/jsduck.

JsDuck на GitHub[Електронний ресурс] – режим доступу до ресурсу: https://github.com/senchalabs/jsduck.