

ПРО ПІДХІД ДО СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПІДТРИМКИ УЗГОДЖЕНОСТІ СТРУКТУР ДАНИХ ПРИ РОЗРОБЦІ ПРОГРАМНИХ СИСТЕМ

М.О. Трибунський¹, М.С. Широкопетлева²

¹ студент кафедри ПІ, ХНУРЕ, Харків, Україна

² ст. викладач кафедри ПІ, ХНУРЕ, Харків, Україна

На сьогоднішній день тисячі розробників у всьому світі налагоджують обмін даними між клієнтським програмним забезпеченням та серверним.

Сфера підтримки розробки програмного забезпечення активно розвивається. У її рамках вирішуються такі проблеми, як транспортування, валідація та зберігання даних. Але, наприклад, при транспортуванні даних формати, у яких вони передаються однією стороною та приймаються іншою, можуть відрізнятися. Така ситуація може трапитись через погано налагоджену комунікацію між командами, помилку, допущену у процесі розробки, тощо. Дані передаються у форматі JSON, XML або у будь-якому іншому, але окремо жоден з них не дає змогу підтримки цілісності структур даних на різних частинах програмного забезпечення.

В роботі розглядається можливість автоматизації узгодження структури даних між застосунками, що дозволить вирішити проблему підтримки цілісності структури даних у декількох сервісах, розроблених на різних мовах, одночасно.

Наразі ця проблема вирішується мануально – шляхом описання структур даних у проектній документації та їх підтримки власноруч. Даний метод рішення не виключає ймовірність припущення людиною помилки, що є приводом для автоматизації цього процесу. Також вже існує програмне забезпечення, зокрема ASP.NET Core Built-in validation та Fluent Validation, що частково вирішує поставлену проблему шляхом валідування даних на дотримання вимогам, заданим розробниками. З одного боку, це рішення дозволяє автоматично перевіряти дані на наявність всіх членів структури, описаної у проектній документації. З іншого боку, воно не враховує вихідний код, який найбільш точно описує структуру даних, яка очікується при отриманні.

Для вирішення цієї задачі пропонується спроектувати систему, яка на підставі вихідного коду створює жорстку структуру даних, що очікується у різних частинах системи, та відповідає за її підтримку.

Складовими такої системи виступатимуть бібліотеки для Backend-застосунків розроблених з використанням ASP.NET Core, та Frontend-застосунків, розроблених за допомогою Angular. Backend бібліотека в якості вхідних даних повинна отримувати тип об'єкту C# (клас або структуру) та генерувати схему структури його полів у форматі JSON. Вихідний JSON-об'єкт матиме інформацію про всі члени класу та правила валідації щодо цих полів. Отриманий JSON-об'єкт далі має передаватись на Frontend-застосунок, де бібліотека, на підставі отриманих даних, автоматично має генерувати фасад [1], який дозволить лише заповнювати поля, описані схемою, та надасть механізми для валідації даних, на базі тих, що були сконфігуровані у Backend-частині системи. Після заповнення даних через фасад та валідацію, їх можна передавати до Backend-застосунку.

В роботі запропонований підхід до вирішення проблеми підтримки узгодженості структур даних на різних частинах програмного забезпечення в процесі розробки

Список літератури:

1. Refactoring Guru [Електрон. ресурс]. – Режим доступу: <https://refactoring.guru/ru/design-patterns/facade> – Фасад