

Issue Analysis Report – Event Processing System Refactoring

Student Name: Mu'men R Hijazi

Course: Advanced Topics in Software Development

1. Introduction

الموجودة في الكود الأصلي لنظام معالجة الأحداث قبل (Design Issues) يهدف هذا التقرير إلى تحليل المشاكل التصميمية (Refactoring). عملية إعادة الهيكلة هذه المشاكل تتعلق بالتصميم المعماري وجودة الكود، وليست أخطاء تشغيل أو ترجمة.

2. Overview of the Original Design

يقوم بجميع المهام (EventProcessor) الكود الأصلي كان يعتمد بشكل أساسي على كلاس واحد مركزي التخزين في قاعدة البيانات، وإرسال الإشعارات، payload مثل التحقق من الحدث، تنفيذ المنطق، بناء الـ. هذا الأسلوب أدى إلى تصميم متشابك وصعب التوسعة.

3. Identified Design Issues

3.1 Single Responsibility Principle (SRP) Violation

الشرح:

كان مسؤولاً عن أكثر من وظيفة في نفس الوقت EventProcessor كلاس، مما جعله كبير الحجم وصعب الصيانة، وأي تعديل صغير فيه قد يؤثر على أجزاء أخرى من النظام.

Resolution:

للمعالجة Pipeline تم فصل المسؤوليات إلى عدة كائنات مستقلة مثل للإشعارات Subscribers للتعامل مع أنواع الأحداث، و Handlers.

3.2 Open/Closed Principle (OCP) Violation

الشرح:

للتعامل مع أنواع الأحداث المختلفة if / switch الكود الأصلي استخدم شروط، وبالتالي إضافة نوع حدث جديد كانت تتطلب تعديل الكود الموجود.

Resolution:

بحيث يتم فصل منطق كل نوع حدث في كلاس مستقل Strategy Pattern تم تطبيق، وبالتالي يمكن إضافة أنواع جديدة بدون تعديل الكود الأساسي.

3.3 Tight Coupling

الشرح:

«Logger كان هناك اعتماد مباشر على كائنات محددة مثل قاعدة البيانات والد. وهذا يقلل من مرونة النظام ويصعب عملية الاختبار.

Resolution:

بين المكونات (Coupling) لتقليل الترابط Interfaces تم استخدام

3.4 Rigid Payload Construction

الشرح:

«كان مدمجًا داخل منطق المعالجة Payload بناء ال. مما جعل إضافة خصائص جديدة مثل التشفير أو الضغط أمرًا معقدًا.

Resolution:

بشكل مرن وديناميكي Payload لبناء ال Decorator Pattern تم استخدام

3.5 Inefficient Resource Management

الشرح:

الوصول إلى قاعدة البيانات لم يكن منظمًا ولم يستخدم أسلوب إعادة استخدام الموارد.

Resolution:

لتحسين إدارة الموارد Proxy و Repository مع Connection Pool تم تطبيق

4. Conclusion

«عملية إعادة الهيكلة حسّنت تصميم النظام بشكل كبير من حيث التنظيم. قابلية التوسعة، وسهولة الصيانة، وذلك من خلال تطبيق مبادئ هندسة البرمجيات الصحيحة.