

Aufgabe 3

23.04.2014

Das Softwareprojekt

1 Message Queue

In dieser Aufgabe sollen mit Hilfe des Konzeptes der losen Kopplung und Message Queues Nachbarsysteme simuliert werden. Frachtabrechnungen sollen nun nicht mehr von Hand erzeugt werden, sondern außerhalb des HLS. Die eingegangenen Frachtabrechnungen werden dann automatisch verarbeitet und eine entsprechende Gutschrift an die Bank verschickt. Der Datenaustausch zwischen den externen Komponenten findet per Message Queues statt.

Folgendes ist hierfür zu tun:

- Erweitern Sie das Nachbarsystem `FrachtfuehrerExternal` um den Versand von Frachtabrechnungen über die Message Queue an das HLS. Hierfür muss die schon bestehende Logik zum Empfang von Frachtaufträgen in einen Task (`System.Threading.Tasks`) ausgelagert werden, damit das Nachbarsystem nebenläufig sowohl Nachrichten empfangen als auch senden kann. Für die Vorführung der Lösung sollen drei Frachtabrechnungen verschickt werden.
- Die Komponente `FrachtfuehrerAdapter` muss eingehende Frachtabrechnungen aus der Queue lesen und dann an die Buchhaltungskomponente weiterleiten. Dazu muss ebenfalls ein neuer Task angelegt werden. Dieser Task wird durch die Methode `StarteEmpfangVonFrachtabrechnungen` nach dem Zusammenbau des Application Core explizit gestartet.
- Erstellen Sie eine Adapterkomponente **BankAdapter**, die die Kommunikation mit der Bank implementiert. Hier soll beim Bezahlen einer Frachtabrechnung (siehe Aufgabe 2) eine Gutschrift an den BankAdapter geschickt werden, welcher per

Message Queue die Gutschrift an das Nachbarsystem sendet.

- Erstellen Sie das Nachbarsystem **BankExternal** als Konsolenanwendung, welches die vom HLS verschickten Gutschriften aus der Queue liest und ausgibt. Legen Sie dazu ein neues Projekt im Ordner “4 - External Systems” an.
- Erstellen Sie für die Vorführung die Konsolenanwendung “HLS Konsole” im Ordner “0 - Client”. Hier soll sowohl der Application Core zusammengebaut werden als auch eine Befüllung mit Testdaten stattfinden, die notwendig sind um Frachtabrechnungen zu verarbeiten.

Hinweise:

- Folgende Namen sollen für die Queues verwendet werden:
 - Frachtführer zu HLS **HLS.Queue.Frachtabrechnung.TeamX**
 - HLS zu Bank **HLS.Queue.Gutschrift.TeamX**
- Alle Zugriffe auf die Message Queue sollen über in die in Util.Common vorhandene MessagingServices Schnittstelle passieren.
- WebAPI der Message Queue: <http://win-devel.informatik.haw-hamburg.de:15672/#/queues> (Login: teamX u. entsprechendes Teampasswort).

2 Neo4j

Im Moment wird die Neo4j Datenbank dazu genutzt Informationen über die Lokationen und deren Beziehung zueinander zu speichern. Bei jedem Testdurchlauf und jedem Starten des HLS werden die Informationen zurückgesetzt.

Ihre Aufgabe:

- Stellen Sie Funktionen in der Komponente Transportplanung zur Verfügung die:
 1. Lokationen anhand Ihrer Namen aus der Datenbank suchen.

2. Transportbeziehungen aus der Datenbank zurückgeben.

- Stellen Sie sicher, dass das HLS-System und die Tests verschiedene Datensätze verwenden. (Z.B. In den Tests Lokationen und Beziehungen Namen wie Test-Hamburg geben). In den Tests sollen diese Datensätze, wie bisher, nach jedem Einsatz gelöscht werden.
- Das HLS soll prüfen, ob es bereits eine Lokation mit dem entsprechenden Namen gibt. Ansonsten soll eine neue angelegt werden.
- Analog zu den Lokationen soll mit den Transportbeziehungen verfahren werden.

3 Abnahme

Die Abnahme besteht aus einem Pflicht- und einem Zusatzteil. Der Pflichtteil muss für eine erfolgreiche Abnahme komplett erfüllt werden. Jeder erfüllte Zusatzteil gibt extra Punkte und fließt in die abschließende Benotung ein.

Pflichtteil

- Aufgabenteil 1 muss vollständig erfüllt sein.
- Ihr Code muss lauffähig sein.
- Fragen zum Thema Scrum müssen beantwortet werden können.
- Desweiteren sind folgende Fragen zu beantworten:
 - Wofür werden Messaging Queues verwendet? Warum handelt es sich um eine Form der losen Kopplung?
 - Welches Format haben die Nachrichten innerhalb der Queue?
 - Was sind Verbundtests?

Zusatzteil

- Setzen Sie Aufgabenteil 2 um.
- Ausführliche Codetests.
- Instandhaltung der Dokumentation.
- Eingehaltene Codekonventionen.
- Professionelle Nutzung von Jira. Teilen Sie die Aufgaben in Arbeitspakete und Unteraufgaben auf und schätzen Sie die Aufwände entsprechend dem Scrum-Vorgehensmodell.
- Professionelle Versionsverwaltung. Hier wünschen wir uns diesmal einen Tag über den abzugebenen Code bis 20 Uhr des Vorabends.

Abgabetermin: 14.05.2014

Viel Spaß und Erfolg mit der dritten Aufgabe!