

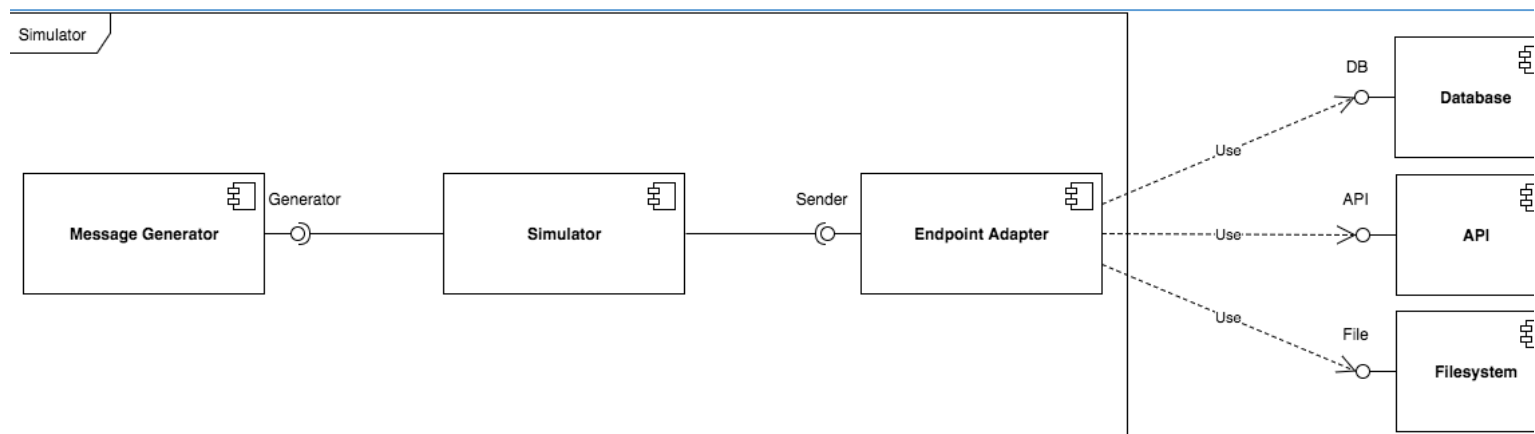
Kurze Beschreibung der Lösung

Die Idee ist, dass der Simulator aus einem Message Generator besteht, der Gabelstapler Nachrichten erzeugen kann. Der Message Generator ist unabhängig von der konkreten Nachricht konstruiert. Es lassen sich leicht neben Einsatznachrichten (Message/Usage) auch weitere Gabelstapler Nachrichten hinzufügen.

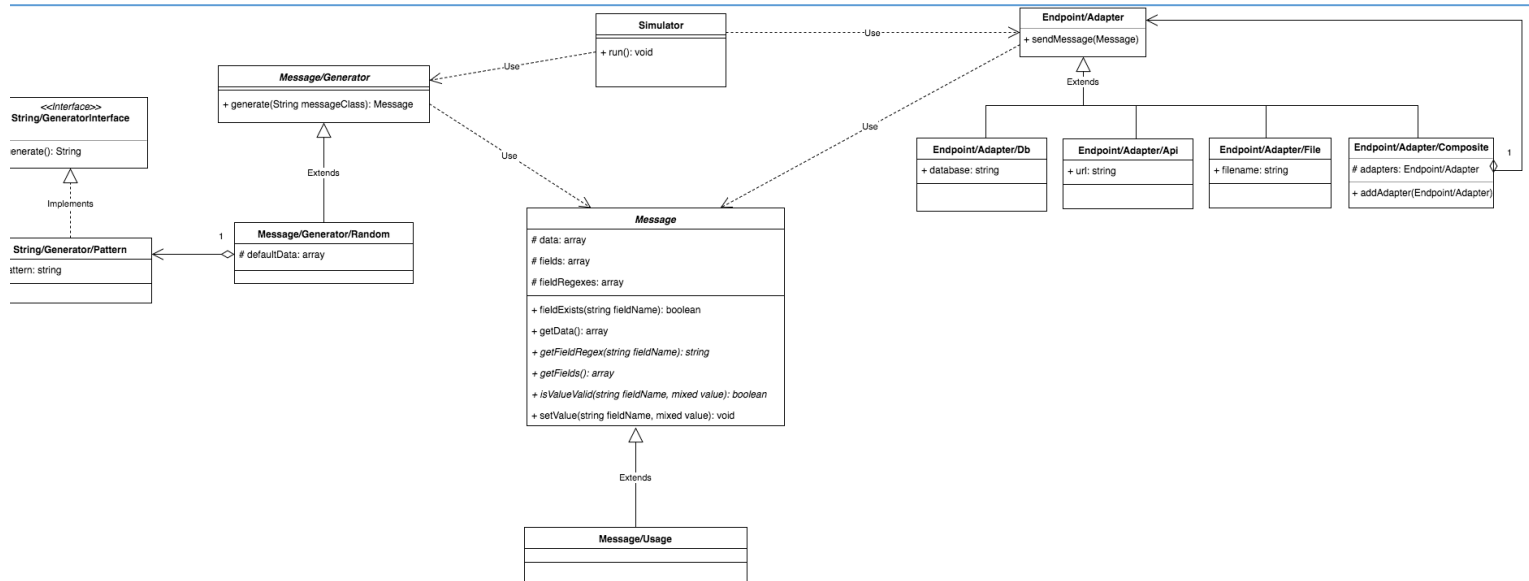
Die erzeugten Nachrichten werden an einen Endpoint Adapter gesendet, der von dem tatsächlichen Ziel der Nachricht abstrahieren soll. Es ist in der Aufgabenstellung nicht ganz klar, wohin die Nachrichten gesendet werden. Mit dem Ansatz des Endpoint Adapters ist der Simulator unabhängig davon, welche Ziele es tatsächlich gibt. Es lassen sich leicht neue Ziele hinzufügen ohne die Struktur verändern zu müssen. Ich habe mir in dieser Lösung drei Ziele überlegt: Datenbank, API und Datei. Laut Aufgabenstellung sollen Daten in einer imaginären Datenbank gespeichert werden. Man könnte mit meiner Lösung die Einsatznachrichten direkt in einer Datenbank speichern.

Verwendete Design-Pattern: Adapter, Composite

Komponentendiagramm



Klassendiagramm



Ausführen des Simulators

\$ php simulator.php

Unit Tests

Ich habe mit einer UnitTest Datei im Verzeichnis Tests angedeutet, wie ich systematisch die Message/Usage Klasse testen würde.

Ausführen der Unit Tests:

\$ vendor/bin/phpunit