

Proof Of Concept

Abrechnung (Rechnung)

Damit ein Benutzer in einer Abrechnung mit einbezogen werden kann, muss dieser entweder in den betreffenden Artikeln eines Kassenzettels oder allgemein in der Abrechnung referenziert werden. Sobald alle Teilnehmer in der Abrechnung referenziert und die benötigten Parameter angegeben wurden, kann der Gläubiger den Algorithmus starten. Auf dem Server können anhand der REST-Architektur im Vorfeld Kassenzettel und Abrechnungen erstellt, bearbeitet und in der Datenbank gespeichert werden. Der Server basiert auf Node.js, ist in Javascript geschrieben und verwendet dafür die Module "express" und "mongoose".

Der Algorithmus für die Berechnung der Kostenverteilung folgt dem Muster:

$\text{Artikel(gesamt)preis} * \text{prozentualer Verbrauch des jeweiligen Nutzers}$. Dadurch kann

nachgerechnet werden, ob der Algorithmus die korrekte Kostenverteilung bestimmt hat.

Wenn in einem Artikel keine teilhabenden Nutzer angegeben sind, wurde der Artikel nur von dem Gläubiger verwendet und somit nicht in die Abrechnung einbezogen.

Exit-Kriterium:

Der Algorithmus hat die Kostenverteilung anhand der referenzierten Kassenzettel und Nutzer sowie Parametern (z.B. prozentualer Verbrauch) in mind. 9 von 10 Fällen korrekt ausgerechnet. Die Durchführung des Algorithmus dauert nicht länger als 3 Sekunden.

Fail-Kriterium 1a:

Der Algorithmus hat die Kostenverteilung in 2 oder mehr Fällen falsch ausgerechnet, weil der Algorithmus falsch oder ineffizient ist.

Fallback 1a:

Der Algorithmus muss von den Entwicklern korrigiert und verbessert werden, indem überprüft wird, ob falsche Variablen im Code verwendet werden oder ob im Algorithmus ein mathematischer Fehler vorhanden ist.

Falls dies nicht möglich ist müssen die Nutzer die Abrechnung selbst ausrechnen und der PoC gilt als gescheitert.

Fail-Kriterium 1b:

Der Algorithmus hat die Kostenverteilung in 2 oder mehr Fällen falsch ausgerechnet, weil die falschen Kassenzettel referenziert wurden oder falsche Parameter angegeben wurden.

Fallback 1b:

Der Nutzer muss die korrekten Kassenzettel erneut referenzieren oder Parameter ändern können.

Fail-Kriterium 2:

Die Durchführung dauert länger als 3 Sekunden.

Fallback 2:

Der Algorithmus muss von den Entwicklern optimiert werden, indem überprüft wird, ob die schwache Leistung an dem Querying der Datenbank, dem Datenformat selbst oder dem Aufbau des Algorithmuses liegt. Falls dies keine Verbesserung mit sich bringt sollte der Nutzer währenddessen andere Aktionen ausführen können und vom Client nicht bis zur Antwort des Servers blockiert werden.

Abrechnung (Kommunikation)

Nachdem der Algorithmus zur Berechnung der Kostenverteilung auf dem Server vollzogen wurde, muss das Ergebnis an die Android-Clients aller Beteiligten gesendet werden. Diese müssen der Abrechnung entweder zustimmen oder widersprechen und die Antwort über den Server an den Gläubiger senden.

Exit-Kriterium:

Der Server hat das Ergebnis des Algorithmus in mind. 9 von 10 Fällen an den Client aller beteiligten Nutzer geschickt. Die Nachricht enthält alle benötigten Informationen zur Überprüfung der Abrechnung, wie Nutzer, Anteil, Artikel und zu zahlender Betrag. Die beteiligten Nutzer können dem Gläubiger über den Server mitteilen, ob die Abrechnung korrekt war und angenommen wird. Beide Kommunikationsschritte sollten in höchstens 2 Sekunden ihr Ziel erreichen, wenn der jeweilige Client erreichbar ist.

Fail-Kriterium 1a:

Das Ergebnis wurde in 2 oder mehr Fällen nicht an den Client aller Beteiligten gesendet, weil dieser keine Internetverbindung hatte oder der Client abgeschaltet war.

Fallback 1a:

Der Gläubiger sollte das Ergebnis mehrmals versenden können oder der Server sollte die Nachricht automatisch erneut versenden, falls dieser registriert, dass bestimmte Nutzer die Nachricht nicht erhalten haben.

Fail-Kriterium 1b:

Das Ergebnis wurde in 2 oder mehr Fällen nicht an den Client aller Beteiligten gesendet, weil der Algorithmus das Ergebnis nicht an alle referenzierte Nutzer sendet oder der Gläubiger falsche Nutzer referenziert hat.

Fallback 1b:

Der Gläubiger muss einsehen können, an wann das Ergebnis gesendet wurde und fehlerhafte Referenzierungen ändern

Fail-Kriterium 2:

Die Nachricht an die Beteiligten enthält nicht alle Daten um die Abrechnung nachvollziehen zu können.

Fallback 2:

Der Nutzer kann bei Bedarf die benötigten Daten separat vom Server anfragen

Fail-Kriterium 3:

Der Nutzer kann dem Gläubiger nicht mitteilen, ob er die Rechnung akzeptiert oder nicht, weil der Gläubiger keine Internetverbindung hatte oder der Client abgeschaltet war.

Fallback 3:

Der Nutzer sollte das Ergebnis mehrmals versenden können oder der Server sollte die Nachricht automatisch erneut versenden, falls dieser registriert, dass der Gläubiger die Nachricht nicht erhalten hat. Die Bestätigung/Absage könnte auch in der Abrechnung gespeichert werden, so dass der Gläubiger dies zu einem späteren Zeitpunkt separat einsehen kann.