

Fontys Hogeschool Techniek en Logistiek

Informatik

Software Factory

Work - to - Students
Research Report - Merve Sahin
Firebase

16. Januar 2018

Informationen

Autor	Merve Sahin (2621363)
Modul	Software Factory
Betreuender Dozent	Ferd van Odenhoven
Institut	Fontys Hogeschool Techniek en Logistiek
Studiengang	Informatik
Studienjahr	2017/2018
Ort und Datum	Venlo, 16. Januar 2018

Firestore

Aufgrund der parallelen Entwicklung der mobilen und der Webapplikation, werden Ressourcen wie Datenbank erstmals für die mobile Applikation nicht zur Verfügung gestellt. Die mobile Version soll bestimmte Serverseitige Voraussetzungen erfüllen. Funktionalitäten wie Datenabfrage, Synchronisierung der Daten durch Events sollen vorhanden sein. Zudem soll die Applikation in der Lage sein Push Notifications zu senden und unterschiedliche Dateien hochzuladen.

Firestore stellt all diese Funktionalitäten innerhalb einer einzigen Plattform zur Verfügung.

Was ist Firestore?

Firestore ist eine Entwicklungsplattform für mobile und Web-Applikationen. Sie bietet diverse Micro-Services, wie z.B. Datenbank, Storage, Hosting, Push Notifications, App Engine, Cloud Functions, etc. Die Standard-Datenbank ist eine in der Cloud betriebene No-SQL Datenbank. Daten werden im JSON-Format abgespeichert und in Echtzeit synchronisiert. Die Firestore Schnittstelle ist eine Rest-API, die über HTTP Methoden mit Endgeräten kommunizieren kann. Für unterschiedliche Plattformen wie z.B. Browser, Android oder iOS werden Implementierungsschnittstellen zur Verfügung gestellt.

Vorteile

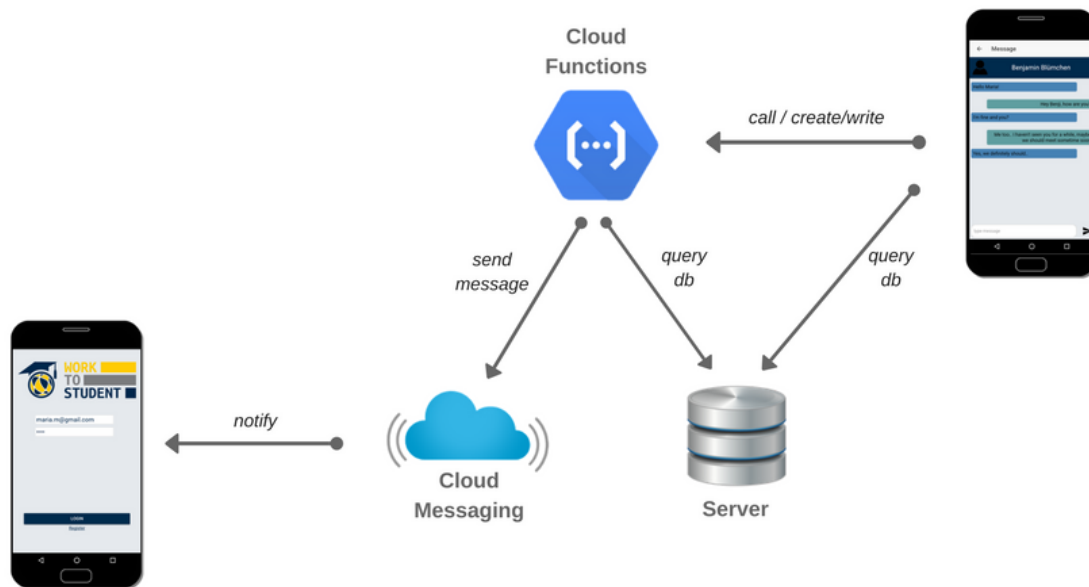
Die Einarbeitungszeit wird um einiges gekürzt, da Firestore sehr gut dokumentiert ist und viele Code-Beispiele zur Verfügung stellt.

Für die Implementierung einer hybriden Anwendung eignet es sich sehr gut, weil es genauso wie Ionic Funktionalitäten für unterschiedliche Anwendungen, sowohl mobil als auch web, zur Verfügung stellt. Firestore generiert eine Konfigurationsdatei für Android und iOS, um über die Implementierungsschnittstelle Zugang zu mehreren Services zu verschaffen. Diese sind Datenbank, Storage, Authentication und Push Notifications.

Neben der normalen Datenabfrage, bietet diese Schnittstelle Funktionalität für das Synchronisieren der Daten, beispielsweise werden Listen automatisch aktualisiert. Dies erspart bei der Implementierung viel Arbeit und erhöht die Performanz der Applikation.

Mit dem *Cloud Functions Service* kann man Serverseitig viele personalisierte Funktionalitäten einbauen, wie z.B. um Ergebnisse nach bestimmten Parametern zu sortieren. Somit bietet es

mehr Flexibilität und erhöht wiederum die Performanz, da Sortier-Operationen oder ähnliches Serverseitig laufen und dementsprechend die Applikation nicht überlasten. Dieser Service kann zu mehreren Zwecken dienen, eins davon wäre als Datenbank-Trigger. Man kann Trigger für bestimmte Endpunkte in der JSON-Struktur implementieren, wie z.B. beim Erstellen und speichern einer neuen Nachricht. Zudem kann man in der *Cloud Functions* auf mehrere Services zugreifen, beispielsweise die Push Notification. D.h. jedes Mal wenn jemand eine neue Nachricht in der Datenbank erstellt, wird der Trigger aufgerufen, welcher auf die Push Notification zugreift und eine Nachricht an ein anderes Gerät sendet. Die folgende Abbildung stellt den Prozess grob dar:



Nachteile

Im Gegensatz zu einer SQL-Datenbank hat Firebase deutlich weniger Abfrage-Parameter und muss daher mithilfe der *Cloud Functions* ergänzt werden. Somit nimmt die Abfrage länger Zeit in Anspruch und der Implementierungsaufwand ebenso.

Das Anlegen von neuen Testdaten über die Benutzeroberfläche scheint anfangs kompliziert und nimmt jegliche Zeit in Anspruch. Eine Andere Alternative wäre das Importieren einer JSON-Datei, welche jedoch alle vorhandenen Daten überschreibt.

Das Erstellen eines Datenbank-Designs für eine NO-SQL Datenbank wird erschwert, da abhängige Tabellen bzw. Dokumente schwer darzustellen sind. Zudem kann man für einzelne Felder keine

Constraints erstellen, was beim Hinzufügen von neuen Daten problematisch wird, weil Felder, die Nummern erwarten sollten, auch Strings annehmen.

Fazit

Firebase bietet diverse Funktionalitäten, die in Kombination einen großen Einfluss auf den Implementierungsaufwand haben, Es nimmt die meiste Arbeit ab, jedoch auf Kosten der verminderten Anzahl der Abfrage-Parameter. Abhängig davon ob man viele Komplexe Abfragen ausführt kann es entweder seinen Zweck erfüllen oder auf *Cloud Functions* angewiesen sein. Firebase ist kostenfrei und eignet sich perfekt als eine Testdatenbank. Für einen kostenpflichtigen Tarif bietet Firebase auch SQL-Datenbanken.