Einführung in Key-Value Stores

Einleitung

Was ist ein Key-Value Store?

- **Definition**: NoSQL-Datenbank mit einfachem Key-Value-Paar-Mechanismus zum Speichern von Daten.
- **Struktur**: Basiert auf zwei Spalten:
  - Key | Value
- **Merkmale**: Einfachste und populärste NoSQL-Datenbank.

Beispiele für Anwendungsbereiche

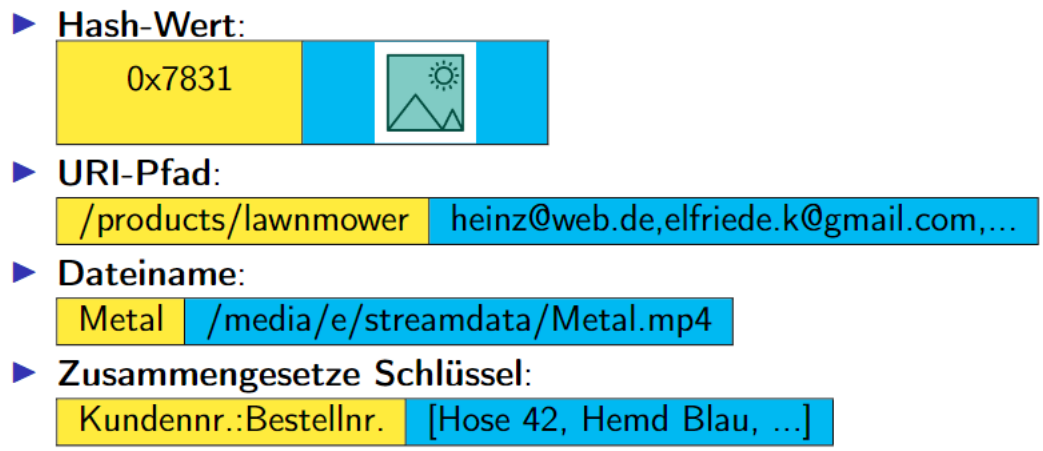
- **Caching**: Speicherung häufig genutzter Daten für schnellen Zugriff.
- **Session-Management**: Verfolgung von Benutzersitzungen in Webanwendungen.
- **Configuration Management**: Speichern von Konfigurationseinstellungen.

Arten von Key-Value Stores

- **In-Memory Key-Value Stores**: Daten im RAM für schnellen Zugriff (z. B. Memcached).
- **Persistent Key-Value Stores**: Daten auf Festplatte gespeichert (z. B. RocksDB).
- **Hybrid Key-Value Stores**: Kombination aus In-Memory und Persistent (z. B. Redis).

Technische Details

Aufbau Schlüssel und Werte



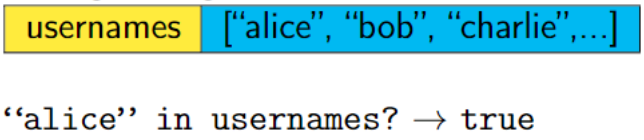
Speichern der Werte

- **Datenformate**: Bilder, JSON-Dateien als Byte-Arrays (z. B. BLOB).
- **Umwandlung**: Serialisierung/Deserialisierung durch den Client.

Abfragen der Werte

- **Einschränkungen**: Struktur erlaubt kein direktes Abfragen der Werte.

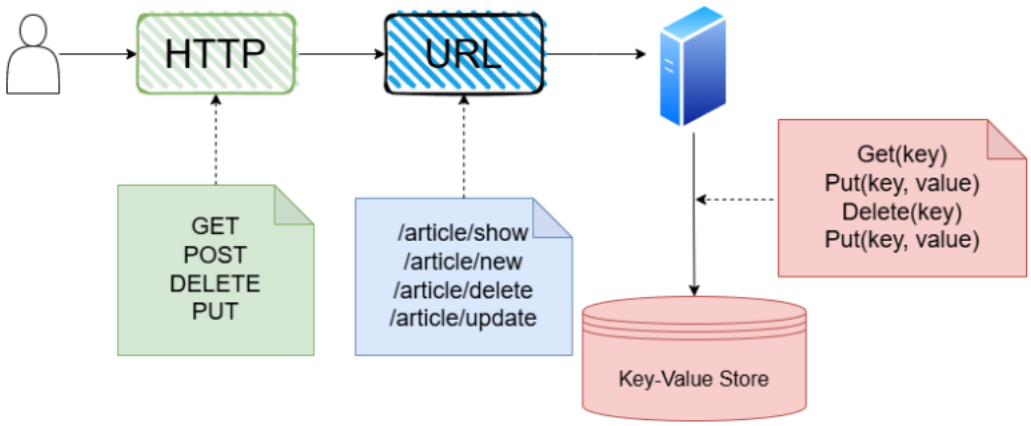
- **Clientseitige Abfrage**: Möglich.
- **Erweiterte K-V-Stores**: Z. B. Redis erlaubt Listenabfragen.



Die Abfrage-Operationen

- **Get(key)**: Lesen
- **Put(key, value)**: Schreiben / Aktualisieren
- **Delete(key)**: Löschen

Umsetzung Abfrage als REST API



Fazit

Vorteile von Key-Value Stores

- **Einfach**: Bedienung mittels Get/Put-Operationen.
- **Performance**: Hohe Geschwindigkeit durch niedrige Latenzzeiten, besonders bei In-Memory-Lösungen.
- **Skalierbarkeit**: Einfache horizontale Skalierbarkeit.
- **Flexibilität**: Schemafreies Design ermöglicht dynamische Datenspeicherung.

Nachteile von Key-Value Stores

- **Begrenzte Abfragemöglichkeiten**: Keine komplexen Abfragen wie bei relationalen Datenbanken.
- **Datenintegrität**: Weniger Fokus auf ACID-Eigenschaften.
- **Komplexe Datenbeziehungen**: Verwaltung kann aufwändig sein.
- **Potenzielle Datendopplungen**: Gefahr der mehrfachen Speicherung derselben Daten.

Quellen

- Ali Davoudian, Liu Chen, and Mengchi Liu. A survey on nosql

  stores. ACM Computing Surveys, 51(2):1–43, 2019. ISSN

  0360-0300. doi: 10.1145/3158661.

- IONOS. Key-value-store: Wie funktionieren schlüssel-werte-datenbanken? URL

  https://www.ionos.de/digitalguide/hosting/hosting-technik/key-value-store/.

- Andreas Meier and Michael Kaufmann. SQL- & NoSQL-Datenbanken. eXamen.press. Springer Vieweg, Berlin and Heidelberg, 8., überarbeitete und erweiterte auflage edition, 2016.

  ISBN 9783662476635. URL http://www.springer.com/.

- Redis. Redis documentation. URL https://redis.io/docs/latest/develop/data-types/

Letzer Abruf aller Webseiten: 31.05.2024 16:40 Uhr