JSON

A High Performance Document Store for Modern Applications

JSON is a high performance document store that allows developers to build modern applications. It stores and processes JSON in-memory, supporting millions of operations per second with sub-millisecond response times. Native indexing, querying, and full-text search of JSON documents allows developers to create secondary indexes, and query data at lightning speed.

Enterprises are unable to meet the needs of modern applications

Rigid database schema limits agility

Relational database management systems (RDBMSes) have rigid data schemas that are hard to update and scale.

Disk-based document stores cause bottlenecks

Document stores allow developers to iterate faster by using a flexible JSON data model. However, they suffer from slow writes and high read latencies due to disk i/o resulting in application bottlenecks.

Bolt-on indexing and full-text search increases complexity

RDBMSes and existing document stores lack native indexing and full-text search capabilities. Bolt-on technologies increase complexity due to the need for integration.

JSON allows you to build modern applications with interactive experiences

In-memory JSON document store

- Store and process scheme-free JSON in-memory, supporting millions of operations per second with sub-millisecond response times

Native indexing, querying, and full-text search of JSON*

- Create secondary indexes, query your data, and run full-text search at lightning speeds
  *Using RediSearch

Enterprise-grade availability and scalability

- 99.999% availability with support for active-active geo-distribution

Available anywhere

- Flexible deployment options - on-premises, hybrid, and multicloud offerings. Fully managed support for AWS, and Google Cloud
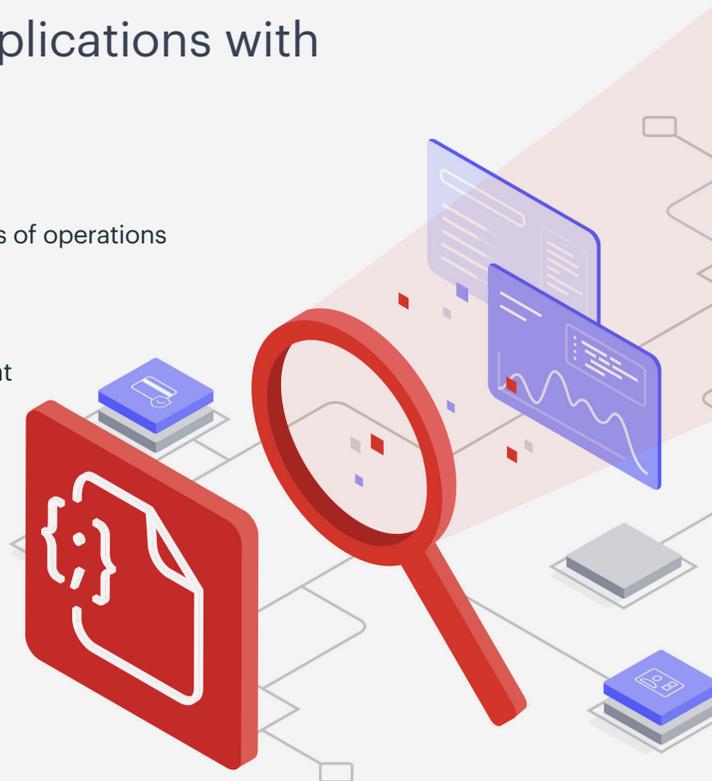
Usage Patterns

JSON makes it possible to undertake non-disruptive modernization of RDBMSes and slow document stores by using the principles of caching. Customers deploy JSON using different usage patterns across their data stacks.

Cache
Use JSON as a high-speed cache to store frequently accessed JSON data and manipulate sub-elements using atomic operations.

Primary Database
Distributed, in-memory JSON document database.

Query Accelerator
Use JSON as an in-memory data fabric on top of one or more data stores to accelerate queries while offloading production systems.

Use Cases

JSON is ideal for modern applications delivering real-time performance for use cases such as Customer360, Content Management, Product Catalogs, Mobile App Development, Session Management, and more.

Customer 360
Search, find and store critical information on customers for a product /service, profile, and history to match specific profiles and behaviors for better support

Content Management
Store and serve information assets and associated metadata to a range of applications like websites, online publications, and archives

Product Catalogs
Manage and search for thousands of products with different attributes and SKU combinations

Mobile App Development
Build responsive mobile apps while keeping your data in sync across client apps

Key Features

In-memory JSON store
- Native JSON data structure in Redis
- Implements JSON data interchange standard as a native data type (ECMA-404)
- In-memory manipulation of JSON documents at high velocity and volume

Server-side atomic operations
- Atomic updates to JSON sub-elements in-memory
- Supports JSONPath syntax for selecting fields within documents
- Partial and conditional document indexing in atomic operations

Linear scalability and geo-distribution
- Linear scale to any size and high throughput
- Globally distributed using CRDTs-based Active-Active Redis
- Indexes are automatically distributed across all replicas
- Guaranteed 99.999% uptime

Fully-managed, hybrid, and multicloud
- Fully managed support for AWS, and Google Cloud
- Scalable queries across entire clusters for indexing billions of JSON documents on hundreds of servers

Native indexing, querying, and full-text search (using RediSearch)
- Real-time indexing approach, allows you to instantly query documents that have been indexed
- Filtering by numeric properties and geographical distances
- Support for prefix, fuzzy, synonyms, and phonetic searches
- Stemming-based query expansion in multiple languages (using Snowball)
- Aggregation operations like groupby/reduce, map, sort and filter