WHITEPAPER

Couchbase Evaluation Guide for Java Developers

Couchbase

Introduction

This paper guides Java developers through the investigation and evaluation process of Couchbase. It includes simple, step-by-step navigation and content consumption advice for how to become familiar with the core integration capabilities of Couchbase and its use as a distributed data store for Java applications.

The document assumes that the reader is familiar with Couchbase Server, and its benefits such as multimodal data access to allow developers to innovate faster; support for deployment anywhere – in the cloud, across clouds, across datacenters, onto mobile and edge devices of all kinds; and focus on exceptional user experience through performance at scale, helping to drive down operational costs.

As Java developers begin planning their 2021 application development projects, we have updated and overhauled much of our developer-targeted content, and started with Java because it is the enterprise developers' lingua franca. We intend to expand this effort every few months to cover most of our ten supported language SDKs.

Java-centric content curated by our developer advocates

When you navigate to developer.couchbase.com you will find constantly updated content. The site provides a single-page explanation of Couchbase, features an introductory video by Couchbase co-founder Steve Yen, and offers clear navigation to our language-specific developer content. Our “Get Started” content moves you quickly through installations and into language-specific tutorials.

For example, when you click on Java, you are immediately directed to our quickstart tutorial for using Couchbase with Java and Spring Boot. While you may be tempted to jump in and start downloading software, this paper is designed to systematically guide you through learning exercises before installing any Couchbase software and trying it yourself.

Understanding the flexibility of the JSON document data model

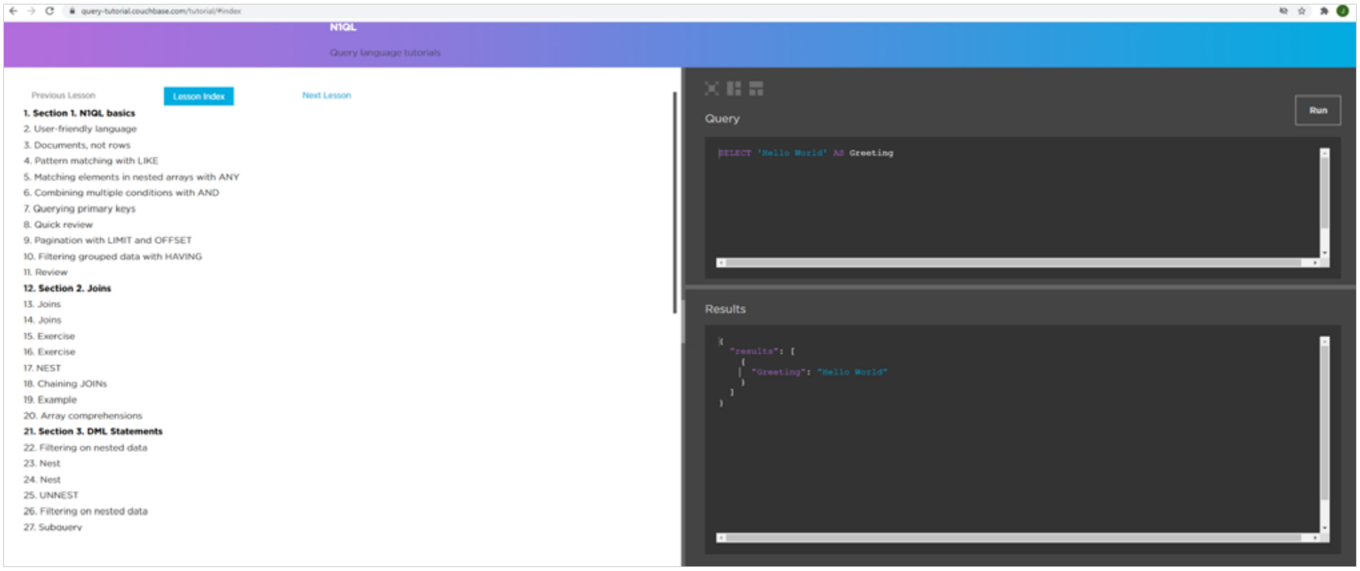
Document databases are different from relational databases because of their ability to change data structures as easily as changing data values. This capability inverts a long-standing control premise of data-driven development – that the database (and its DBA) controls the data, and not the application. JSON, however, offers Java developers the ability to have their application programmatically control the structure and the data in the database, rather than the database’s structure controlling what data is available to the application. To learn more, check out these articles on the value of a JSON database and advice on how to model JSON documents for performance.

The N1QL query language tutorial

When building a Couchbase application, Java developers will likely toggle between the Java APIs in Couchbase and the database's query language, N1QL (pronounced "nickel"). You'll wish you had a nickel every time you think that N1QL is just like standard SQL. That is intentional, as we felt it was important to carry over decades of practical experience for millions of database programmers rather than teach them a new language when SQL is both expressive and universal. [Our N1QL tutorial] is available online and free to use.

It guides you through the basics of N1QL, including how it retrieves and presents data in JSON format. You'll notice the familiar syntax and keywords of N1QL including "SELECT," "WHERE," "LIKE," "HAVING," and others that match ANSI SQL. [The N1QL language reference guide] has a complete list of N1QL statements and their usage.

According to [TechValidate research], N1QL is the No. 1 capability that Couchbase users cite as their favorite feature once they start using it.



1. Section 1. N1QL basics
2. User-friendly language
3. Documents, not rows
4. Pattern matching with LIKE
5. Matching elements in nested arrays with ANY
6. Combining multiple conditions with AND
7. Querying primary keys
8. Quick review
9. Pagination with LIMIT and OFFSET
10. Filtering grouped data with HAVING
11. Review
12. Section 2. Joins
13. Joins
14. Joins
15. Exercise
16. Exercise
17. NEST
18. Chaining JOINs
19. Example
20. Array comprehensions
21. Section 3. DML Statements
22. Filtering on nested data
23. Nest
24. Nest
25. UNNEST
26. Filtering on nested data
27. Subquery

```
SELECT 'Hello World' AS Greeting
```

```
{
  "results": [
    {
      "Greeting": "Hello World"
    }
  ]
}
```

Develop transactional Java applications

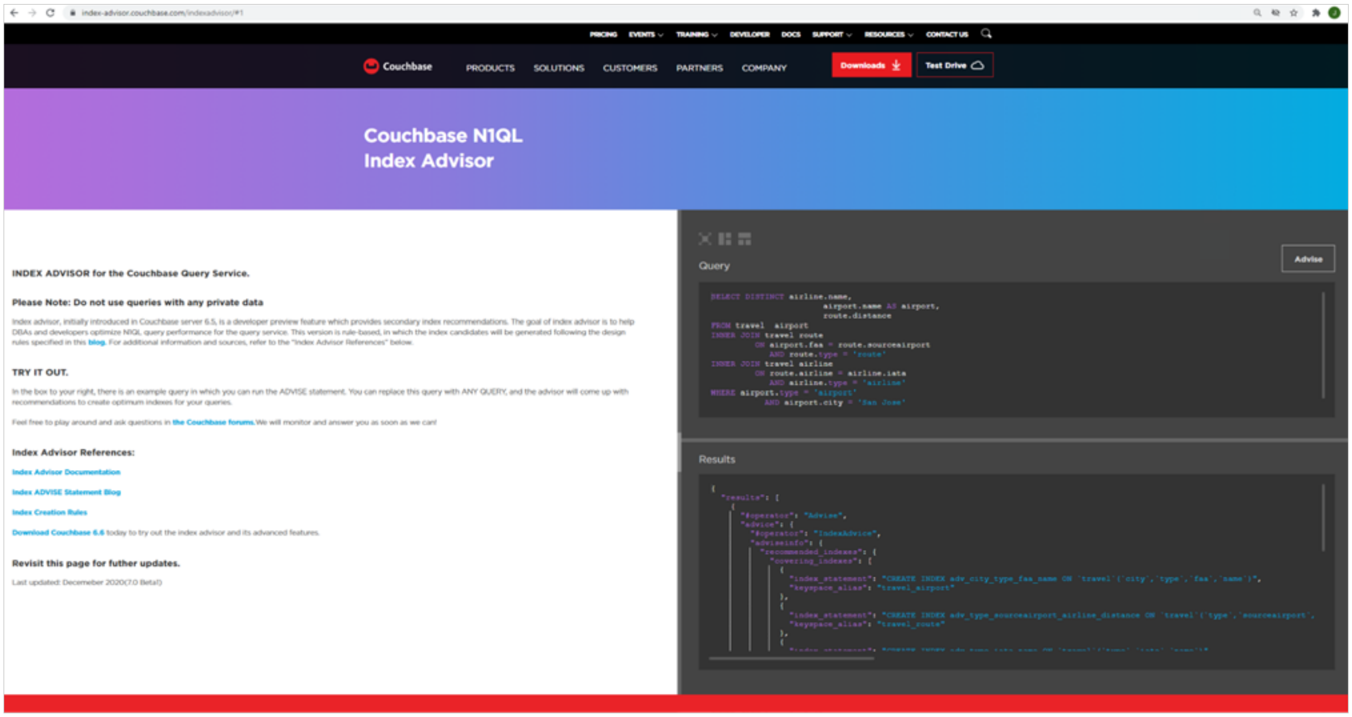
ACID transactions were first introduced in the Couchbase Java SDK, which is well documented. Java developers only need to add the couchbase-transactions library into their project, which allows the application to dictate where transactions are necessary and interact directly with the Key-Value (KV) engine. There are extensive examples of transactions, a great blog post, and a 2020 Connect.ONLINE session that demonstrate how to use transactions.

Transactions in N1QL are newly supported as well, making it even easier to adopt this capability. To familiarize yourself with transactions, it is best to review some of its deployment considerations, especially when designing systems for performance or ones that use cross-datacenter replication (XDCR). With the addition of multi-document, distributed ACID transactions built into both the SDKs and the N1QL query language, Couchbase offers Java developers several options to support transactional applications that perform at distributed scale. We feel that this is a primary requirement of any modern database, and a capability that sets Couchbase apart from other database vendors.

Index construction and utilization

Once a Java developer is comfortable with N1QL capabilities, it is important to understand how to index the database to ensure high performance. We have a number of resources to teach these capabilities.

First, we created the Index Advisor, which evaluates a N1QL query and identifies the best index to use or create for optimal performance. You can check out this release blog post to learn more about this service. And building indexes has never been faster, now that they are built in parallel. You can learn more about index statements in Couchbase documentation.



Couchbase N1QL
Index Advisor

INDEX ADVISOR for the Couchbase Query Service.

Please Note: Do not use queries with any private data

Index advisor, initially introduced in Couchbase server 6.5, is a developer preview feature which provides secondary index recommendations. The goal of index advisor is to help DBAs and developers optimize N1QL query performance for the query service. This version is rule-based, in which the index candidates will be generated following the design rules specified in this **blog**. For additional information and sources, refer to the "Index Advisor References" below.

TRY IT OUT.

In the box to your right, there is an example query in which you can run the ADVISE statement. You can replace this query with ANY QUERY, and the advisor will come up with recommendations to create optimum indexes for your queries.

Feel free to play around and ask questions in **the Couchbase forums.** We will monitor and answer you as soon as we can!

Index Advisor References:

Index Advisor Documentation

Index ADVISE Statement Blog

Index Creation Rules

Download Couchbase 6.6 today to try out the index advisor and its advanced features.

Revisit this page for futher updates.

Last updated: Decemeber 2020(7.0 Beta!)

Query

```
SELECT DISTINCT airline.name,
                       airport.name AS airport,
                       route.distance
FROM travel  airport
INNER JOIN travel route
         ON airport.faa = route.sourceairport
            AND route.type = 'route'
INNER JOIN travel airline
         ON route.airline = airline.iata
            AND airline.type = 'airline'
WHERE airport.type = 'airport'
           AND airport.city = 'San Jose'
```

Results

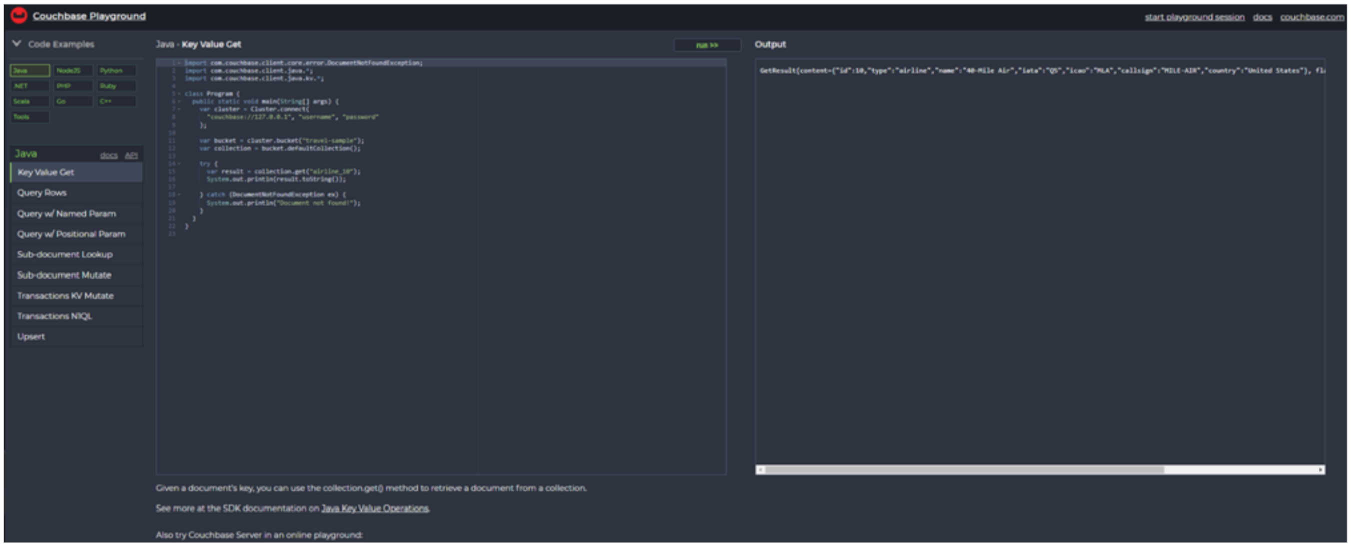
```
{
  "results": [
    {
      "#operator": "Advise",
      "advice": {
        "#operator": "IndexAdvice",
        "adviseinfo": {
          "recommended_indexes": {
            "covering_indexes": [
              {
                "index_statement": "CREATE INDEX adv_city_type_faa_name ON `travel`(`city`,`type`,`faa`,`name`)",
                "keyspace_alias": "travel_airport"
              },
              {
                "index_statement": "CREATE INDEX adv_type_sourceairport_airline_distance ON `travel`(`type`,`sourceairport`,
                "keyspace_alias": "travel_route"
              },
              {
                "index_statement": "CREATE INDEX adv_type_iata_name ON `travel`(`type`,`iata`,`name`)"
```

Using Couchbase for Java development

Before you invest time installing any software, we recommend that you try Couchbase Playground, which is a pre-installed online environment created by our developer advocacy team, and tailored for the Java developer. Here, we spin up a time-sensitive container of Couchbase so that you can explore the database using language-specific examples and follow a step-by-step guide to become familiar with Couchbase.

The playground teaches you the basics of Couchbase, including executing Key-Value operations, getting familiar with N1QL, and even teaching you about N1QL transactions. It also navigates you to the Couchbase dashboard interface so you can execute your own queries and explore what it is like to run the Couchbase GUI.

The playground is an excellent way to familiarize yourself with the operations and capabilities of Couchbase without having to install software on your local machines.



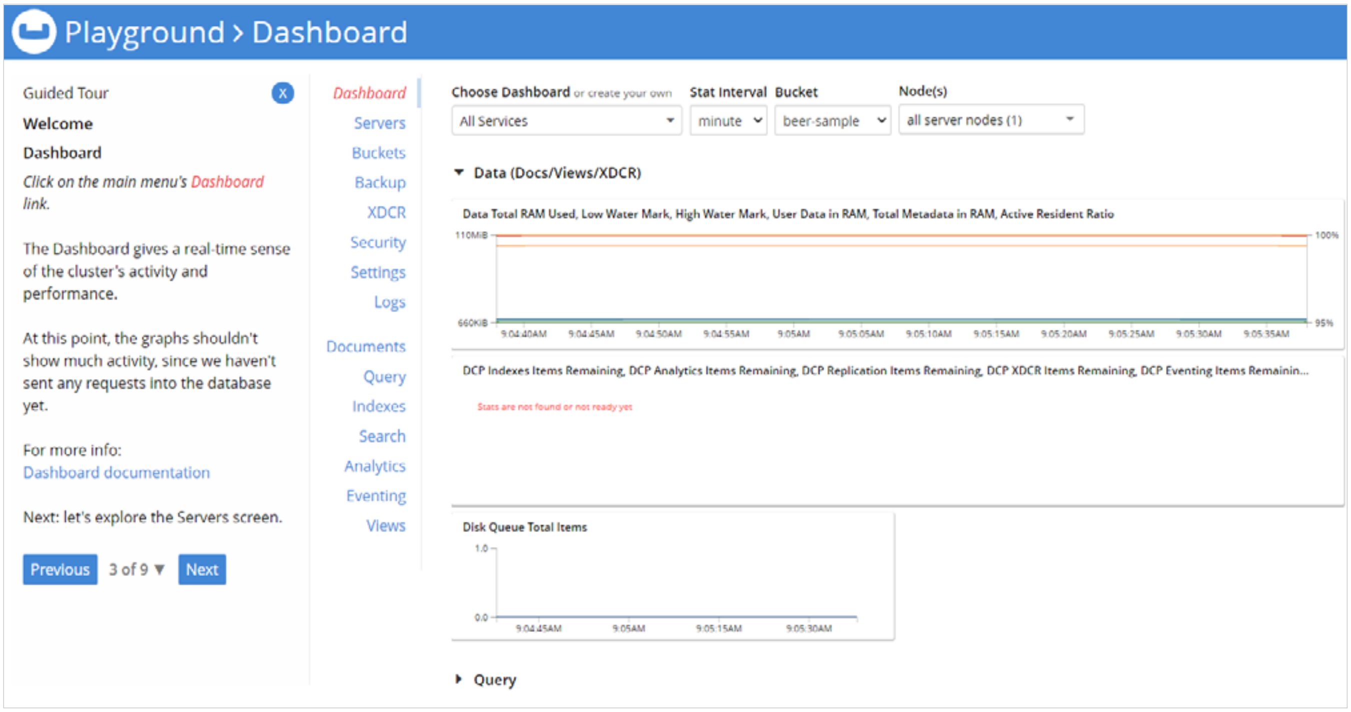
```
import com.couchbase.client.core.error.DocumentNotFoundException;
import com.couchbase.client.java.*;
import com.couchbase.client.java.kv.*;

class Program {
  public static void main(String[] args) {
    var cluster = Cluster.connect(
      "couchbase://127.0.0.1", "username", "password"
    );

    var bucket = cluster.bucket("travel-sample");
    var collection = bucket.defaultCollection();

    try {
      var result = collection.get("airline_10");
      System.out.println(result.toString());

    } catch (DocumentNotFoundException ex) {
      System.out.println("Document not found!");
    }
  }
}
```

```
GetResult{content={"id":10,"type":"airline","name":"40-Mile Air","iata":"Q5","icao":"MLA","callsign":"MILE-AIR","country":"United States"}, fl
```

Given a document's key, you can use the collection.get() method to retrieve a document from a collection.

See more at the SDK documentation on Java Key Value Operations.

Also try Couchbase Server in an online playground:



Playground > Dashboard

Guided Tour

Welcome

Dashboard

Click on the main menu's Dashboard link.

The Dashboard gives a real-time sense of the cluster's activity and performance.

At this point, the graphs shouldn't show much activity, since we haven't sent any requests into the database yet.

For more info:
Dashboard documentation

Next: let's explore the Servers screen.

Choose Dashboard or create your own: All Services
Stat Interval: minute
Bucket: beer-sample
Node(s): all server nodes (1)

Data Total RAM Used, Low Water Mark, High Water Mark, User Data in RAM, Total Metadata in RAM, Active Resident Ratio

DCP Indexes Items Remaining, DCP Analytics Items Remaining, DCP Replication Items Remaining, DCP XDCR Items Remaining, DCP Eventing Items Remainin...

Stats are not found or not ready yet

Disk Queue Total Items

Once you are finished with the playground, we suggest you start working with Couchbase yourself. You need to download Couchbase Server, your Java SDK, and have your favorite IDE editor handy. Once those steps are completed, you can review our tutorial on Couchbase and Spring Boot.

Quickstart for Couchbase with Java and Spring Boot

To extend Couchbase into your application development stack, we recommend that you try our Quickstart for Couchbase with Java and Spring Boot along with your favorite Java development IDE to begin your project. Once it is built, look for information about how to deploy Couchbase in your cloud of choice.

Forum-based help from Java developers

Help is available via Couchbase Forums and you can already find the answers to the top five questions asked.

Couchbase for Java developers training and certification

If you would like training on how to develop in Java with Couchbase, we offer multi-level training through Couchbase Academy. These skill-based training courses are broken out into Associate, Professional, and Expert developer levels. After you complete a course, you can take a certification exam and earn a badge for each skill level.

Couchbase is a modern, multicloud-to-edge, JSON document database for building applications with agility without compromising performance and scale. And we make it easy for Java developers by offering the tools they need to get their projects off the ground quickly. Start your Java development journey today with Couchbase.

Couchbase

At Couchbase, we believe data is at the heart of the enterprise. We empower developers and architects to build, deploy, and run their mission-critical applications. Couchbase delivers a high-performance, flexible and scalable modern database that runs across the data center and any cloud. Many of the world's largest enterprises rely on Couchbase to power the core applications their businesses depend on.

For more information, visit www.couchbase.com.