Search

× Hide

Lessons

WELCOME TO THE INTRODUCTION TO MONGODB COURSE

▾ Learn

Video: Welcome to the Introduction of MongoDB

Overview: Introduction to MongoDB

Lab: Introduction to the IDE

Progress 100%

100% Complete

Show Details

Resources & Forums

Assignments

Notes

Get Support

FAQ'S SIGN OUT

Welcome to the Introduction to MongoDB Course / Learn

Next

Next

Search

× Hide

Lessons

- Learn ✓
- Practice ✓

LESSON 3: MANAGING DATABASES, COLLECTIONS, AND DOCUMENTS IN ATLAS DATA EXPLORER

- Learn ✓
- Practice ✓

CONCLUSION

- Learn ✓
  - Video: Overview of MongoDB and the Document Model Review
  - Conclusion: Overview of MongoDB and the Document Model

Progress 100%

100% Complete

Show Details

Resources & Forums

Assignments

Notes

Get Support

Overview of MongoDB and the Document Model

In this unit, you learned about the relationship between MongoDB and Atlas. MongoDB is a general-purpose database that can be used for a variety of use cases. It's part of Atlas, the developer data platform. You also learned about the MongoDB document model. Specifically, you learned about these key features:

- Data is organized into documents, collections, and databases.
- Documents are stored in BSON, which supports a large range of data types, including all JSON data types, dates, numbers, and ObjectIds.
- Every document requires an `_id` field, which acts as a primary key or unique identifier. If an inserted document doesn't have an `_id` field, MongoDB automatically generates one.
- MongoDB has a flexible schema, which means that documents with different structures can be stored in the same collection.

Finally, you learned how to use Atlas Data Explorer to create a database, collection, and a document and begin storing your own data.

Resources

Use the following resources to learn more about inserting and finding documents in MongoDB:

Lesson 01: Introduction to MongoDB

- What Is MongoDB?
- MongoDB Use Cases

Lesson 02: The MongoDB Document Model

- MongoDB Docs: Documents
- MongoDB Docs: BSON Types
- Explaining BSON with Examples
- JSON and BSON

Lesson 03: Managing Databases, Collections, and Documents in Atlas Data Explorer

Search

× Hide

Lessons

- Learn ✓
- Practice ✓

LESSON 3: TROUBLESHOOTING A MONGODB CONNECTION IN JAVA APPLICATIONS

- Learn ✓
- Practice ✓

CONCLUSION

- Learn ✓
  - Video: Connecting to MongoDB in Java Review
  - **Conclusion: Connecting to MongoDB in Java**

Progress 100%

100% Complete

Show Details

Resources & Forums

Assignments

Notes

Get Support

Connecting to MongoDB in Java

In this unit, you learned how to:

- Install the MongoDB Java driver by using Maven.
- Connect to an Atlas cluster from a Java application.
- Use a MongoClient instance.
- Troubleshoot common connection issues when using the MongoDB Java driver.

Resources

Use the following resources to learn more about using the MongoDB driver to connect your Java application to MongoDB:

Lesson 01: Using MongoDB Java Client Libraries

- MongoDB Drivers
- MongoDB Java Driver Docs

Lesson 02: Connecting to an Atlas Cluster in Java Applications

- MongoDB Docs: Get Connection String
- MongoDB Docs: Connection String Format
- MongoDB Java Connection to Atlas

Lesson 03: Troubleshooting a MongoDB Connection in Java Applications

- MongoDB Docs: Troubleshoot Connection Issues

Using MongoDB with Java Course

By completing this unit, you've taken a step toward completing the Using MongoDB with Java course.

Search

× Hide

Lessons

▸ Practice

LESSON 3: COUNTING DOCUMENTS IN A MONGODB COLLECTION

▸ Learn ✓

▸ Practice

CONCLUSION

▾ Learn ✓

Video: MongoDB CRUD Operations: Modifying Query Results Review

Conclusion: MongoDB CRUD Operations: Modifying Query Results

Progress 100%

100% Complete

Show Details

Resources & Forums

Assignments

Notes

Get Support

MongoDB CRUD Operations: Modifying Query Results

In this unit, you learned how to modify query results with MongoDB. Specifically, you learned how to:

- Return query results in a specified order by using `cursor.sort()`.
- Constrained the number of results returned by using `cursor.limit()`.
- Specified fields to return by adding a projection document parameter in calls to `db.collection.find()`.
- Counted the number of documents that match a query by using `db.collection.countDocuments()`.

Resources

Use the following resources to learn more about modifying query results in MongoDB:

Lesson 01: Sorting and Limiting Query Results in MongoDB

- MongoDB Docs: cursor.sort()
- MongoDB Docs: cursor.limit()

Lesson 02: Returning Specific Data from a Query in MongoDB

- MongoDB Docs: Project Fields to Return from Query
- MongoDB Docs: Projection Restrictions

Lesson 03: Counting Documents in a MongoDB Collection

- MongoDB Docs: db.collection.countDocuments()

Associate Certification Course

If you're interested in continuing your journey toward the Associate Developer Certification exam, the next step is to review the following units:

Search

× Hide

Lessons

- Learn ✓
- Practice ✓

LESSON 5: TROUBLESHOOTING MONGODB ATLAS CONNECTION ERRORS

- Learn ✓
- Practice ✓

CONCLUSION

- Learn ✓
  - Video: Connecting to your Database Review
  - **Conclusion: Connecting to your Database**

Progress 100%

100% Complete

Show Details

Resources & Forums

Assignments

Notes

Get Support

Connecting to a MongoDB Database

In this unit, you learned how to connect to Atlas by using:

- The MongoDB Shell
- MongoDB Compass
- Applications

Finally, you learned how to troubleshoot some common connection problems that can occur.

Resources

Use the following resources to learn more about connecting to your database.

Lesson 01: Using MongoDB Connection Strings

- MongoDB Docs: Get Connection String
- MongoDB Docs: Connection String URI Format

Lesson 02: Connecting to a MongoDB Atlas Cluster with the Shell

- MongoDB Docs: The MongoDB Shell

Lesson 03: Connecting to a MongoDB Atlas Cluster with Compass

- MongoDB Docs: MongoDB Compass

Lesson 04: Connecting to a MongoDB Atlas Cluster from an Application

- MongoDB Docs: Connect via Your Application

Lesson 05: Troubleshooting MongoDB Atlas Connection Errors

- MongoDB Docs: Troubleshoot Connection Issues

Search × Hide

Lessons

MODEL

- Learn ✓
- Practice ✓

LESSON 7: USING ATLAS TOOLS FOR SCHEMA HELP

- Learn ✓
- Practice ✓

CONCLUSION

- Learn ✓
  - Video: Introduction to MongoDB Data Modeling Review
  - **Conclusion: Introduction to MongoDB Data Modeling**

Progress 100%

100% Complete

Show Details

Resources & Forums

Assignments

Notes

Get Support

Conclusion / Learn

Previous Next

Introduction to MongoDB Data Modeling

In this unit, you learned how to:

- Explain the purpose of data modeling.
- Identify the types of data relationships (one to one, one to many, many to many).
- Model data relationships.
- Identify the differences between embedded and referenced data models.
- Scale a data model.
- Use Atlas Tools for schema help.

Resources

Use the following resources to learn more about the basics of data modeling:

Lesson 01: Introduction to Data Modeling

- Data Modeling Introduction
- Separating Data That is Accessed Together

Lesson 02: Types of Data Relationships

- Data Model Design
- Model Relationships Between Documents
- Embedding MongoDB
- MongoDB Schema Design Best Practices

Lesson 03: Modeling Data Relationships

- Data Model Design
- Model Relationships Between Documents

Lesson 04: Embedding Data in Documents

Search

× Hide

Lessons

Learn

Practice

LESSON 5: DELETING DOCUMENTS IN MONGODB

Learn

Practice

CONCLUSION

Learn

Video: MongoDB CRUD Operations: Replace and Delete Documents Review

Conclusion: MongoDB CRUD Operations: Replace and Delete Documents

Progress 100%

100% Complete

Show Details

Resources & Forums

Assignments

Notes

Get Support

MongoDB CRUD Operations: Replace and Delete Documents

In this unit, you learned how to modify query results with MongoDB. Specifically, you:

- Replaced a single document by using `db.collection.replaceOne()`.
- Updated a field value by using the `$set` update operator in `db.collection.updateOne()`.
- Added a value to an array by using the `$push` update operator in `db.collection.updateOne()`.
- Added a new field value to a document by using the `upsert` option in `db.collection.updateOne()`.
- Found and modified a document by using `db.collection.findAndModify()`.
- Updated multiple documents by using `db.collection.updateMany()`.
- Deleted a document by using `db.collection.deleteOne()`.

Resources

Use the following resources to learn more about modifying query results in MongoDB:

Lesson 01: Replacing a Document in MongoDB

- MongoDB Docs: replaceOne()

Lesson 02: Updating MongoDB Documents by Using `updateOne()`

- MongoDB Docs: Update Operators
- MongoDB Docs: $set
- MongoDB Docs: $push
- MongoDB Docs: upsert

Lesson 03: Updating MongoDB Documents by Using `findAndModify()`

- MongoDB Docs: findAndModify()

Lesson 04: Updating MongoDB Documents by Using `findAndModify()`

Search × Hide

Lessons

MONGODB

- Learn ✓
- Practice ✓

LESSON 5: FINDING DOCUMENTS BY USING LOGICAL OPERATORS

- Learn ✓
- Practice

CONCLUSION

- Learn ✓
  - CRUD 1: Inserting and Finding Documents with MongoDB
  - Conclusion
  - Conclusion: MongoDB CRUD

Progress 100%

100% Complete

Show Details

Resources & Forums

Assignments

Notes

Get Support

MongoDB CRUD Operations: Insert and Find Documents

In this unit, you learned how to insert and find documents in a MongoDB collection. You built queries by using the following comparison operators:

- `$gt` (greater than)
- `$lt` (less than)
- `$lte` (less than or equal to)
- `$gte` (greater than or equal to)

You also used the following logical operators:

- `$and`
- `$or`

Finally, you learned how to query elements in an array and how to use the `$elemMatch` operator.

Resources

Use the following resources to learn more about inserting and finding documents in MongoDB:

Lesson 01: Inserting Documents in a MongoDB Collection

- MongoDB Docs: insertOne()
- MongoDB Docs: insertMany()

Lesson 02: Finding Documents in a MongoDB Collection

- MongoDB Docs: find()
- MongoDB Docs: $in

Lesson 03: Finding Documents by Using Comparison Operators

- MongoDB Docs: Comparison Operators

Search

× Hide

Lessons

LESSON 2: CREATING AND DEPLOYING AN ATLAS CLUSTER

Learn

Practice

CONCLUSION

Conclusion

Video: Getting Started with MongoDB Atlas, the Developer Data Platform Review

Conclusion: Getting Started with MongoDB Atlas, the Developer Data Platform

Progress 100%

100% Complete

Show Details

Resources & Forums

Assignments

Notes

Get Support

Getting Started with MongoDB Atlas, the Developer Data Platform

In this unit, you learned that MongoDB Atlas is a developer data platform that stores your data in Atlas clusters, a global, multi-cloud database service. You also learned that the developer data platform has many features that enable you to build a wide variety of applications.

You created a free Atlas account and deployed a free tier Atlas cluster by doing the following:

- Chose a shared cluster.
- Chose a major cloud provider to host your cluster.
- Used the region recommended by MongoDB as the location of database deployment.
- Prepared to connect to your cluster by adding a database user and password and an IP address to the IP Access list

Finally, you successfully loaded sample data into your cluster and found a document in a collection by using Data Explorer.

Resources

Use the following resources to learn more about inserting and finding documents in MongoDB:

Lesson 01: Introduction to MongoDB Atlas, the Developer Data Platform

- MongoDB Docs: MongoDB Clusters
- MongoDB Docs: MongoDB Atlas