



MONGODB FOR PYTHON DEVELOPERS

DURATION: 4 DAYS

COURSE OVERVIEW

This course is a deep exploration of building applications in MongoDB (the most popular NoSQL document-database). There are many benefits to choosing a NoSQL database over traditional RDBMSs such as SQL Server or Oracle. This course starts out by looking at why you should choose a NoSQL database in the first place. We will explore the native query language and capabilities of MongoDB. Then we will start working with MongoDB from our Python applications and look at several topics that explore the advanced aspects of the MongoDB Python API including MongoDB from Python, PyMongo, ODMs such as Mongo Engine, and more. We will see how to leverage the immense scalability of MongoDB using the aggregation framework, replica sets, and sharding. You will discover how to store and manage files of virtually unlimited size in MongoDB using GridFS. We will discuss how to properly design your entities and documents (both natively and in Python) to take full advantage of what MongoDB has to offer. Finally, we will round out the course with a few topics that you will need to be successful with MongoDB including Server administration for developers and Security and permissions.

Note that some of the topics cover in this course are technology agnostic (for example the native query lesson is equally applicable to Python developers as well as .NET or Java developers) and some are specifically optimized for Python developers (for example MongoEngine and ODMs).

Why should you choose DevelopMentor's MongoDB course? At DevelopMentor we have experience building and running large scale MongoDB deployments including our online learning platform LearningLine and develop.com itself. The course authors work closely with the MongoDB team and are in the MongoDB Masters program. We have been teaching MongoDB in our Python courses and our Guerrilla .NET course for several years. In short, we know what we're doing with MongoDB and Python and we'd love to share it with you.

COURSE OBJECTIVES

1. Learn why you should consider NoSQL as your database
2. Discover the MongoDB shell and 'native' query language and capabilities
3. Work with MongoDB from Python using PyMongo
4. Use a rich ODM (object data mapper) such as MongoEngine
5. Design your classes and entities to take full advantage of MongoDB

6. Build and tune high performance applications with indexing and profiling
7. Work with the file system (GridFS) within MongoDB from Python
8. Answer complex questions while leveraging MongoDB's scalability using the aggregation framework
9. Learn enough to administration MongoDB servers to be productive
10. Scale out using sharding and replica sets
11. Add security and permissions to MongoDB

COURSE CONTENT

Day 1

- **Why NoSQL and why MongoDB?**
We explore the history of NoSQL databases and why they are becoming so popular, so quickly. We discuss the various categories of NoSQL databases including document databases, key-value stores, column-store databases, and graph databases. While these databases initially seem to have little in common, we will examine a common thread that flows between them. Next, we dig into MongoDB including why it's the industry leader, the techniques it uses to become immensely scalable, and why it makes life much easier and simpler for development teams.
- **The shell and 'native' commands**
You will learn the native commands and query syntax for MongoDB. In order to fully appreciate what each language driver (database API) is doing, you will need to be fluent in working with MongoDB at the server level. That is the goal of this topic. We will see how to query, insert, and update data. We will explore the many operators in the JavaScript API (e.g. \$in).
- **MongoDB from Python**
Accessing MongoDB from Python applications is easy and familiar to most Python developers. You will see how PyMongo uses the rich dictionary support in Python to create a similar API as MongoDB's native JavaScript query syntax. However, there some gotchas (both for execution and performance) and this lesson will ensure you get the most out of MongoDB via Python. There is also a second API built upon the atomic operators of MongoDB (\$set, \$push, etc.) which truly leverages the full power of MongoDB and its aggregate-level atomicity. We will explore this API from Python as well.

Day 2

- **Real world MongoDB with MongoEngine**
Python and PyMongo allow direct programming against MongoDB from Python. This is most appropriately compared to programming at the level of raw SQL for RDBMSes. That level is a necessary building block, but for most applications working at a higher level and building upon custom classes is more appropriate. This module explores one of the most popular ODMs

(object-data mappers) for Python and MongoDB: MongoEngine. You will learn to build out your data model in rich classes and map them to and from MongoDB with ease and incredible performance.

- **Entity Design**

Designing entities in MongoDB (and document databases more generally) is very different than 3rd-normal-form from SQL tables. To be successful with MongoDB, you will need to master this skill. Getting your entity design correct is key to high performance and flexible applications. We will start by discussing some foundational OOD background on these rich documents and how you can leverage them for transactional safety. There are inherent limitations to documents (e.g. total size) and this lesson covers how work within those limits. We will talk about modeling relationships between data (embedding, foreign keys, data duplication). Finally, we discuss how to correctly deal with binary data and large data elements in MongoDB entities.

- **Indexing and profiling**

Simply put, indexes are the single biggest contributor (or culprit) for extremely high performance MongoDB deployments and applications. This lesson will ensure your applications use indexes to full advantage. We will cover how to create indexes and the assorted types (simple, multi-key, compound, unique, sparse, and hashed indexes). You will see how to add indexes to subdocuments for high performance deep queries. In practice, finding the queries that need optimized can be tricky, especially when there is a translation layer in the middle such as MongoEngine and an ODM. You will see how to use profiling to find the slow queries (note that in MongoDB you can reasonably consider 10ms a slow query). We will then use MongoDB's explain method to see the query plan and determine which index (if any) is used.

Day 3

- **GridFS - A file system within MongoDB**

MongoDB has a facility to store, classify, and query files of virtually unlimited size (binary data, text data, etc.). This lesson will introduce GridFS and show you how to work with it from Python. We will upload, download, and list files in GridFS. We will create custom metadata definitions (classes) and store them within our GridFS files which can then be used for rich reporting and querying that does not exist in standard file systems. We will alert you to a few pitfalls that lurk within GridFS and show you how to fill them in and avoid any trouble. Finally, we look at how GridFS is particularly well suited for web applications which the files (images, etc.) can be cached on the client-side.

- **Aggregation framework**

The aggregation framework in MongoDB allows you to execute rich queries and transformations on the server (potentially scaled out across shards). While normal queries leverage documents in the exact structure, aggregation (similar to map-reduce) is much more flexible. It can transform, group, and query data as well as act as a data pipeline on the server. In this lesson, you will learn to choose between aggregation and queries.

- **Server administration for developers**

Being successful with MongoDB means you must keep it running smoothly.

This lesson is designed to ensure that everyone on the team has enough knowledge of the admin side of MongoDB for that purpose. You will learn how to setup MongoDB as a Windows Service / System Daemon. We will discuss various backup strategies (using mongodump / mongorestore, MMS backup, and more) as well as potential tradeoffs associated with these strategies. You will learn how to use mongostat, mongotop, MMS, and other tools to diagnose and monitor MongoDB's performance.

Day 4

- **Replication**

Replication is key to MongoDB's fault tolerance. It can also be used for data locality across data centers, scaled-out reads, offsite backups, reporting without performance degradation, and more. In this lesson we will start by learning how to create a replicate set and how to access primary and secondary replicas from our applications. You will learn about potential 'eventual consistency' issues you may encounter when scaling reads and how to mitigate any issues from them. We will discuss how our applications can automatically fail-over if there is a problem with the primary server. Finally we will configure hidden and delayed replicas for non-core application services.

- **Sharding**

MongoDB is a high performance database even in single-server mode. However, to truly leverage MongoDB's performance potential, you will need to use sharding. This technique allows you to run a cluster of MongoDB servers working in concert to each hold some portion of the data and share some portion of the queries. It is sharding that gives MongoDB the ability to scale horizontally on commodity hardware. This lesson will show you how to configure and monitor sharding and then connect to it from your application.

- **Security and permissions**

Security and data go hand-in-hand. You will learn how to enable security in MongoDB. We will explore how to create and manage user accounts and how to grant and revoke roles to users. You will see that users can be database-wide or server-wide users and we will discuss how to create and manage both. You will learn about basic auditing in MongoDB and how to implement practices which enhance security for MongoDB. Finally we'll end this lesson by exploring major security features coming in the next version of the server.