

Introduction to MongoDB

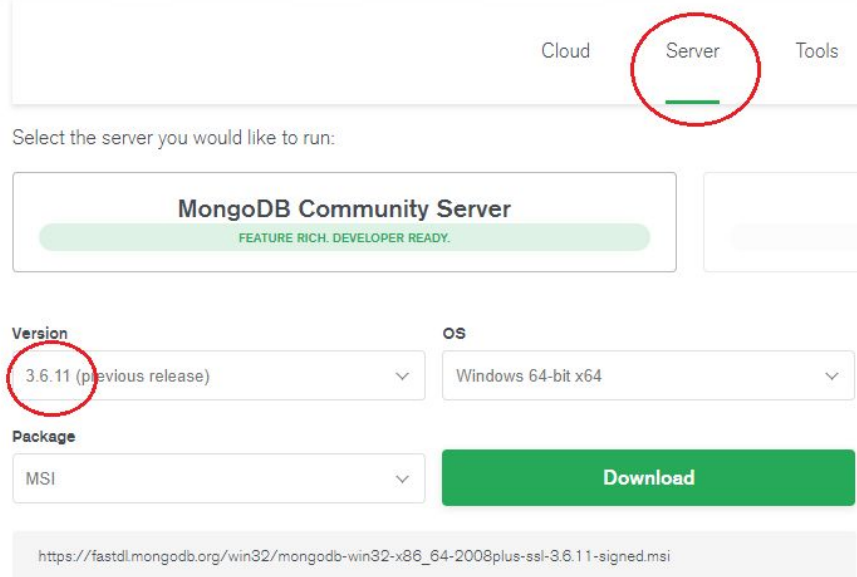
15/04/2019

What is MongoDB

- It's a free NoSQL document oriented database.
 - Not only SQL, since SQL queries are supported.
- It uses JSON-like documents to save data.
 - Flexible schemas - Allows missing fields.
- High performance, high availability, and automatic scaling.
 - Used by large companies like Amazon.

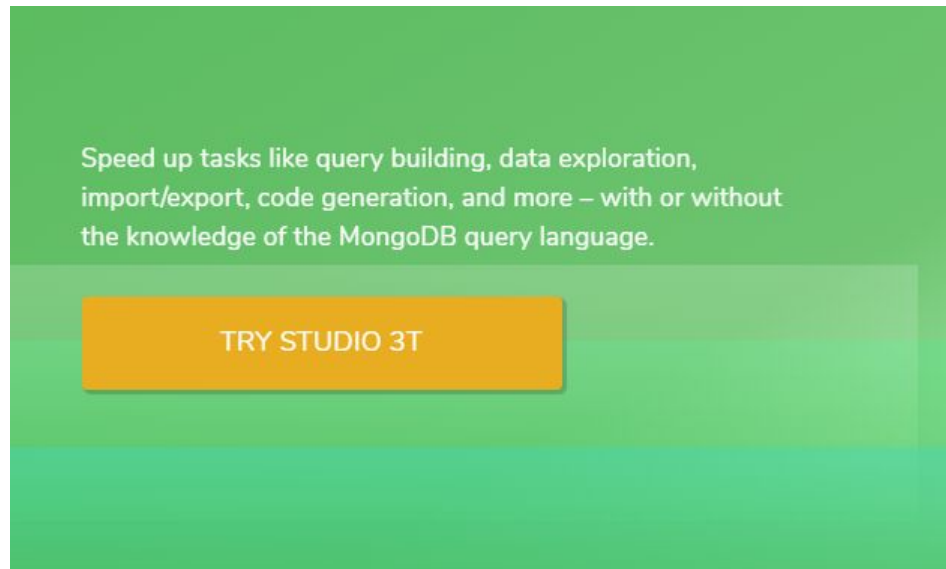
Installation

MongoDB community server



The screenshot shows the MongoDB download page. At the top, there are three tabs: 'Cloud', 'Server', and 'Tools'. The 'Server' tab is selected and highlighted with a red circle. Below the tabs, there is a section titled 'MongoDB Community Server' with the subtitle 'FEATURE RICH. DEVELOPER READY.' Below this, there are two dropdown menus: 'Version' and 'OS'. The 'Version' dropdown is set to '3.6.11 (previous release)' and is highlighted with a red circle. The 'OS' dropdown is set to 'Windows 64-bit x64'. Below these, there is a 'Package' dropdown set to 'MSI'. A green 'Download' button is visible. At the bottom, there is a link to the download file: https://fastdl.mongodb.org/win32/mongodb-win32-x86_64-2008plus-ssl-3.6.11-signed.msi

Studio 3T



Installation

Studio 3T 2/2

Complete the form to download:

☒ Send me helpful tips & tricks

Name *

First Name Last Name

Email *

Country Telephone Number

[Download for Windows](#)

By submitting this form I agree to 3T Software Labs [Privacy Policy](#).

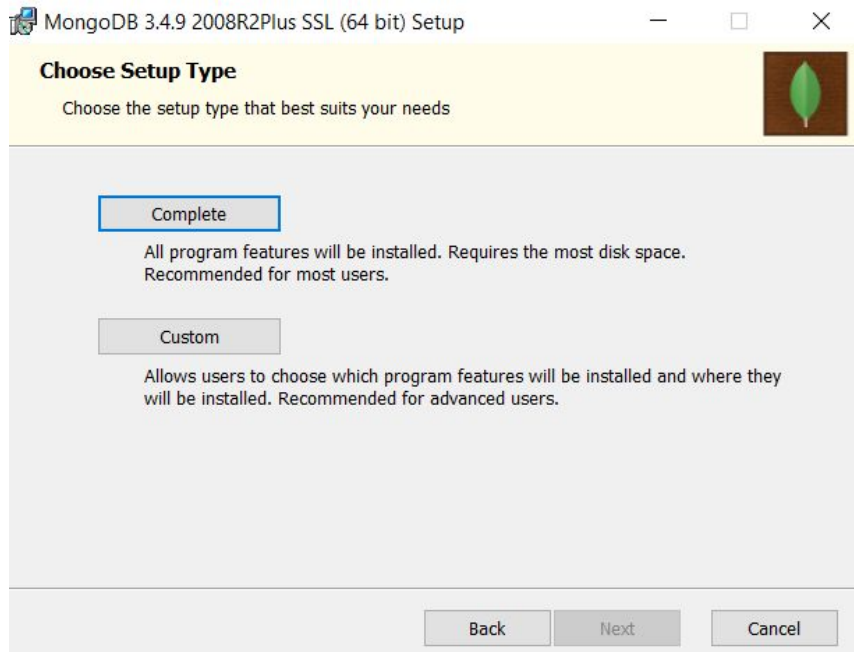
Full access to Core, Pro, and Enterprise editions for 30 days. Core edition always free for non-commercial use. [\(EU/UK\)](#)

+ More download options

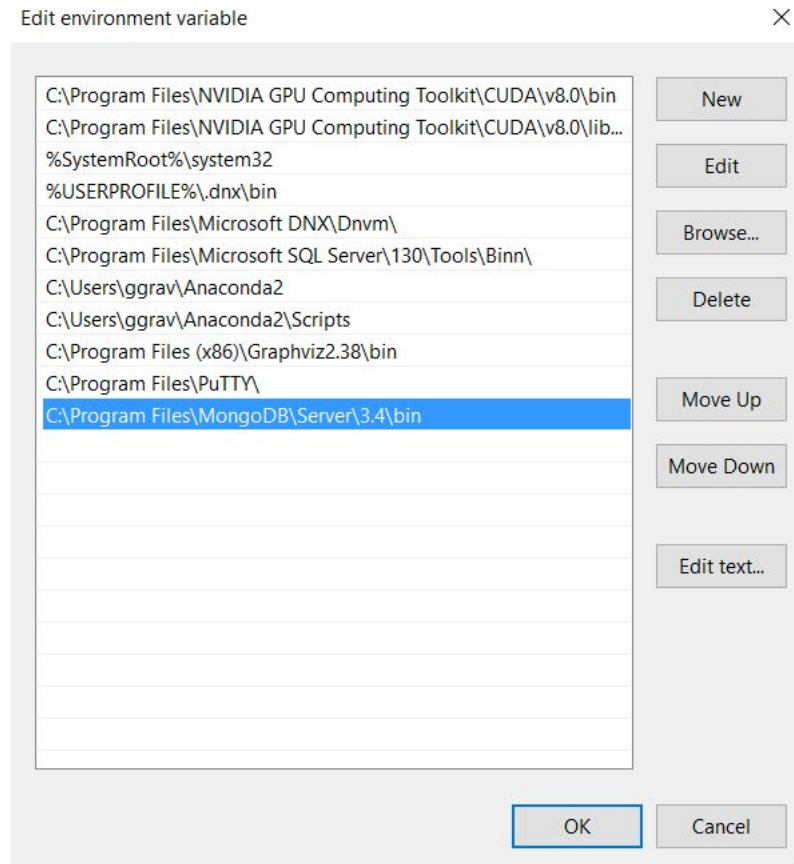
+ How to update?

Installation

Step 1



Step 2



Execution

Step 3 -> Create folder "C:\data\db\" -> Open cmd -> run "mongod"

```

C:\Users\ggrav>cmd
Command Prompt - mongod

(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\ggrav>mongod
2017-10-13T03:50:03.878-0700 I CONTROL [initandlisten] MongoDB starting : pid=17672 port=27017 dbpath=C:\data\db\ 64-bit host=
2017-10-13T03:50:03.878-0700 I CONTROL [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2017-10-13T03:50:03.880-0700 I CONTROL [initandlisten] db version v3.4.9
2017-10-13T03:50:03.880-0700 I CONTROL [initandlisten] git version: 876ebee8c7dd0e2d992f36a848ff4dc50ee6603e
2017-10-13T03:50:03.880-0700 I CONTROL [initandlisten] OpenSSL version: OpenSSL 1.0.1u-fips 22 Sep 2016
2017-10-13T03:50:03.880-0700 I CONTROL [initandlisten] allocator: tcmalloc
2017-10-13T03:50:03.880-0700 I CONTROL [initandlisten] modules: none
2017-10-13T03:50:03.880-0700 I CONTROL [initandlisten] build environment:
2017-10-13T03:50:03.880-0700 I CONTROL [initandlisten]     distmod: 2008plus-ssl
2017-10-13T03:50:03.880-0700 I CONTROL [initandlisten]     distarch: x86_64
2017-10-13T03:50:03.880-0700 I CONTROL [initandlisten]     target_arch: x86_64
2017-10-13T03:50:03.880-0700 I CONTROL [initandlisten] options: {}
2017-10-13T03:50:03.882-0700 I - [initandlisten] Detected data files in C:\data\db\ created by the 'wiredTiger' storage engine
the active storage engine to 'wiredTiger'.
2017-10-13T03:50:03.882-0700 I STORAGE [initandlisten] wiredtiger_open config: create,cache_size=7611M,session_max=20000,evict
threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=
0000),checkpoint=(wait=60,log_size=2GB),statistics_log=(wait=0),
2017-10-13T03:50:04.217-0700 I CONTROL [initandlisten]
2017-10-13T03:50:04.217-0700 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2017-10-13T03:50:04.218-0700 I CONTROL [initandlisten] **           Read and write access to data and configuration is unrestric
2017-10-13T03:50:04.218-0700 I CONTROL [initandlisten]
2017-10-13T13:50:04.469+0300 I FTDC [initandlisten] Initializing full-time diagnostic data capture with directory 'C:/data/c
2017-10-13T13:50:04.472+0300 I NETWORK [thread1] waiting for connections on port 27017
```

What is a document

- A sum of key / value pairs

```
{  
  name: "sue",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]  
}
```



← field: value
← field: value
← field: value
← field: value

What is a collection

- A grouping of MongoDB **documents**.
- A **collection** is the equivalent of an RDBMS table.
- A **collection** exists within a single database.
- **Collections** do not enforce a schema.
- **Documents** within a collection can have different fields.
- **Collections** are grouped into **databases**.

Example: **db** = tweets, **collections** = greek_tweets, english_tweets, etc.

Operations

- Run / Connect to mongo
 - Import documents
 - Insert documents
 - Create queries
 - Quick data lookups
 - Indexing
-

Connection

Studio 3T:

- Studio 3T -> Connect -> New connection -> (enter name) -> Save -> Connect
- Right click -> Add database -> (enter name) -> ok

Python:

```
1      import pymongo
2
3      client = pymongo.MongoClient('localhost', 27017)
4      db = client['yelp']
```

Import dataset

Studio 3T:

- Connect -> Select DB -> Import collections -> JSON -> Add sources -> Rename target collections -> Rename collections* -> Start import

* Rename collections according to the assignment.

Querying

Studio 3T:

- Right click collection -> Open Intellishell
- Examples:
 - `db.restaurants.find({})`
 - `db.restaurants.find({"neighborhood": "Downtown"})`
 - `db.restaurants.find({ $and: [{"neighborhood": "Southeast"}, {"city": "Las Vegas"}]})`

Python:

```
24 db['restaurants'].find({'neighborhood': neighborhood})
25 db['restaurants'].find_one({'business_id': restaurant_id})
26 db['reviews'].find({"$and": [{"business_id": restaurant_id}, {"stars": 5}]})
```

Data lookups - For quick data checking

Studio 3T:

- Right click collection -> Open Intellishell
 - Ex: `db.collection_name.find({ "Search_Field": "value" }, { "Field_to_display": 1 })`
 - `db.restaurants.find({ "neighborhood": "Downtown" }, { "name": 1 })`
 - `db.restaurants.find({ "name": /.*pollos.* / }, { "text": 1 })`
 - `/*` is the regex equivalent for: “any single character, 0 or more times”

Python:

```
27     def find_reviews_that_contain_a_word(word):
28         return db['reviews'].find({'text': {'$regex': '.*' + word + '.*'}})
29
```

Indexing

- Basic indexing
 - Speeds up queries on specific fields
 - Mostly used fields should be indexed
 - Index responsibly : Too many indices might slow down the database
- Text indexing
 - Faster text search queries on string content
 - Term lookup
 - Exact phrase lookup
 - Automatic relevance sorting

How to - Text indexing

The screenshot displays the Studio 3T interface. On the left, a sidebar shows a tree view of the database structure, with 'existing_customers' selected under the 'test-db' collection. A context menu is open over the 'existing_customers' collection, listing various actions such as 'Open Collection Tab', 'Export Collection...', 'Import Data...', and 'Add Index...'. The 'Add Index...' option is highlighted. The main window shows the 'existing_customers' collection with a list of documents. Each document is represented by a hexagonal ID, a set of fields in curly braces, and a 'Type' of 'Document'. The bottom status bar indicates 'Count Documents' and '0.090s'.

ID	Fields	Type
686f57	{ 14 fields }	Document
687003	{ 13 fields }	Document
687078	{ 14 fields }	Document
686f9d	{ 13 fields }	Document
686f89	{ 13 fields }	Document
687081	{ 13 fields }	Document
686f6a	{ 13 fields }	Document
686fe5	{ 13 fields }	Document
686f58	{ 15 fields }	Document
687030	{ 14 fields }	Document
686f7b	{ 13 fields }	Document
687036	{ 13 fields }	Document
687022	{ 13 fields }	Document
68706b	{ 13 fields }	Document
686fb7	{ 13 fields }	Document
686f97	{ 13 fields }	Document
686ff0	{ 12 fields }	Document
686f88	{ 14 fields }	Document
686f9f	{ 12 fields }	Document
687028	{ 14 fields }	Document
686f68	{ 13 fields }	Document
686fc0	{ 13 fields }	Document
686f99	{ 14 fields }	Document
686f98	{ 13 fields }	Document
686fa2	{ 14 fields }	Document

How to - Text indexing

The screenshot shows the Studio 3T for MongoDB interface. The left sidebar displays the database structure: Studio 3T ReplicaSet [replica set] > test-db > Collections (3) > existing_customers. The main window is titled 'Add Index' and shows the 'existing_customers' collection in the 'test-db' database. The index name is set to 'compound_text_index'. The 'Indexed Fields' table is empty, and the 'Index Options' section shows checkboxes for Unique, Sparse, Drop Duplicates, Expire After, and Create In Background.

Studio 3T for MongoDB

Connect Collection IntelliShell SQL Aggregate Map-Reduce Export Import Users Roles Schema Compare Feedback

Search Open Connections (Cmd+F) Add Index

alice Studio 3T ReplicaSet (3 servers) test-db existing_customers

Name: compound_text_index

Fields Text Options Geo Options Partial Index Options JSON Collation

Indexed Fields:

Field	Index Type

Add Field(s)... Remove Field Move Up Move Down

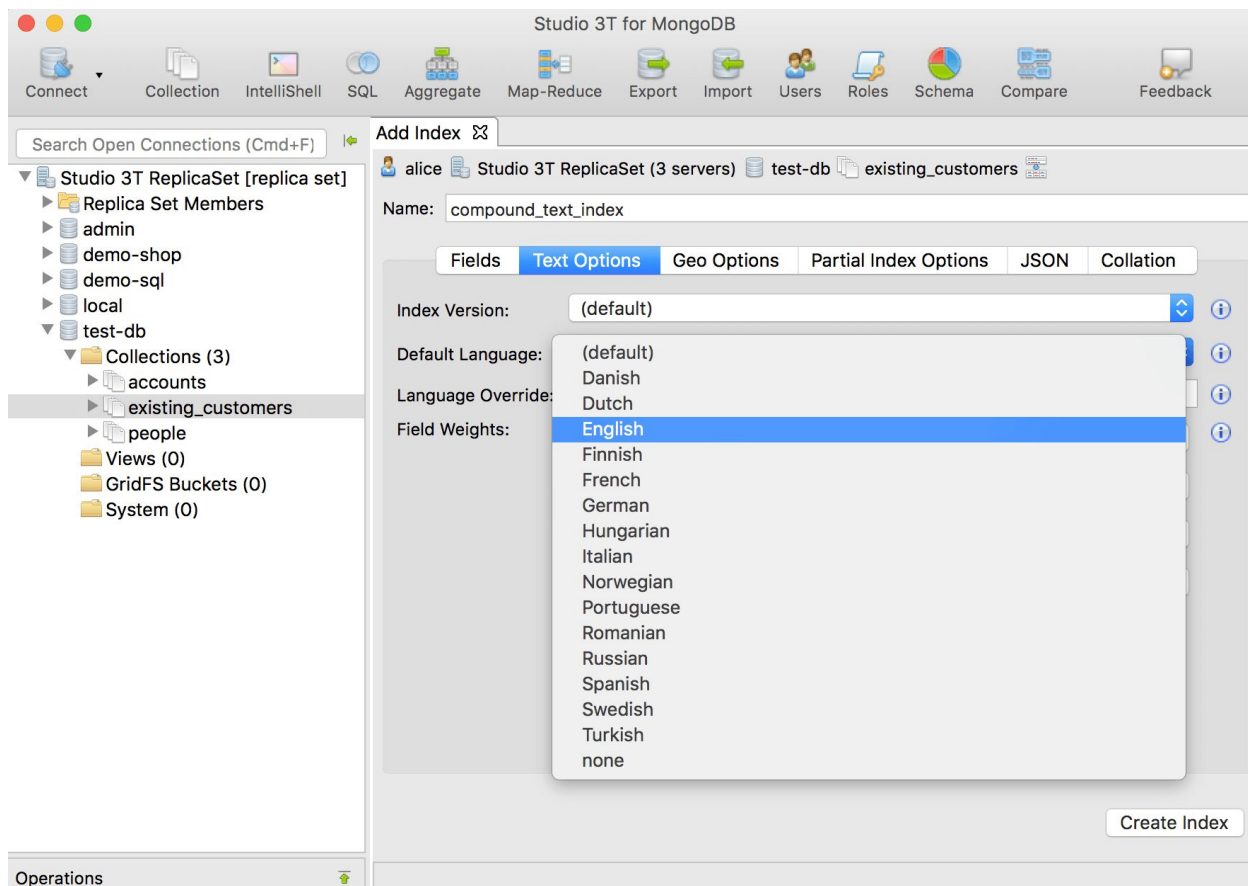
Index Options

☐ Unique ☐ Sparse ☐ Drop Duplicates ☐ Expire After Seconds ☐ Create In Background

Create Index

Operations

How to - Text indexing



Querying with text indices

- `db.stores.find({ $text: { $search: "java coffee shop" } })`
 - `$text` tokenizes the search string and performs a logical OR
 - Will search on all indexed fields
 - Results include a relevance score for each record
- `db.stores.find({ $text: { $search: "java \"coffee shop\"" } })`
 - Will match exact phrases “java” OR “coffee shop”
- `db.stores.find({ $text: { $search: "java shop -coffee" } })`
 - Will match (“java” OR “shop”) AND NOT(“coffee”)

Web app development with flask



A python-based web microframework, suitable for small-scale applications.

[Flask Documentation](#)

[Demo TODO app](#)

[Jinja2 Template Engine Documentation](#)

Resources

- [MongoDB](#)
- [Flask](#)
- [Flask-RESTful](#)
- [Python](#)

<https://github.com/vasisouv/flask-mongo-snippets>