# Set up of MongoDB

1. Community Server can be downloaded from: <https://www.mongodb.com/try/download/community>
2. Install location is: C:\Program Files\MongoDB\Server\5.0\data\
3. Create a folder in C drive at: C:\data\db\
4. Add MongoDB to system PATH (with Git BASH installed) as follows
   1. cd ~
   2. touch .bash\_profile
   3. nano .bash\_profile
   4. Add the following lines to the file and hit ctrl-x to save:

alias mongod="/c/Program\ files/MongoDB/Server/5.0/bin/mongod.exe"

alias mongo="/c/Program\ files/MongoDB/Server/5.0/bin/mongo.exe"

* 1. Restart the terminal
  2. Open two terminals: one will be for the server and the other for the client.

# MongoDB Basic CRUD Commands - Intro

*CRUD = Create Read Update Delete*

To run these examples, open two terminals

1. In one terminal type ‘mongod’ to run the server
2. In the other terminal type ‘mongo’ to run the client

Reference: <https://docs.mongodb.com/manual/crud/>

Basics

|  |  |
| --- | --- |
| **>help** | Get help |
| **>show dbs** | Show available databases (must have at least one collection in them to be shown) |
| **>db** | Show currently selected (use’d) database |

## Create

|  |  |
| --- | --- |
| **>use <dbname>** | Select or (create if not exist) a database |
| **>db** | Show currently selected database |
| **>show collections** | Show the collections in the selected database |
| **>db.<collection>.insertOne( { JSON Obj })** | Insert a document (as a JSON object, with embedded JSON objects optionally) into the collection |

>use shopDB

>db

>db.products.insertOne( {\_id: 1, name: “Pen”, price: 1.2 } )

>db.products.insertOne( {\_id: 2, name: “Pencil”, price: 0.8 } )

>db.products.insertOne( {\_id: 3, name: “ruler”, price: 5.0,

reviews: [ { author: “Michael”, review: “Awesome ruler!” },

{ author: “Ali”, review: “Makes very straight lines” } ] } )

## Read

|  |  |
| --- | --- |
| **>db.<collection>.find( parameters, return fields )** | - Parameters and fields to show are optional. There is also a findOne() method as well |

>db.products.find() - Show all documents in the collection (no parameters)

>db.products.find( { name: “Pencil” } )

>db.products.find( {\_id: { $lt: 3 } } )

>db.products.find( { price: { $gt: 0.5 } } )

>db.products.find( { price: { $gt: 0.5 } }, { name: 1 } ) - Only show name (and \_id) field in results

## Update

|  |  |
| --- | --- |
| **>db.<collection>.updateOne( parameter, change)** |  |

>db.products.updateOne( { \_id: 2 }, { $set: { price: 0.6 } } )

>db.products.updateOne( { \_id: 2 }, { $set: { stock: 32 } } )

>db.products.updateOne( { \_id: 1 }, { $set: { stock: 16 } } )

>db.products.find()

## Delete

|  |  |
| --- | --- |
| **>db.<collection>.deleteOne(parameter)** |  |

>db.products.deleteOne( { \_id: 2 } )

## After getting experience with the terminal-based client

1. Now move on to open the ‘FruitsProject’ which will show you and example of how to use the native MongoDB driver (which directly uses the functions shown above) in NodeJS to communicate with the server and database. Driver and docs may be found at: <https://docs.mongodb.com/drivers/>
2. Later we may also learn to use Mongoose, which is simpler and less verbose than the native driver.