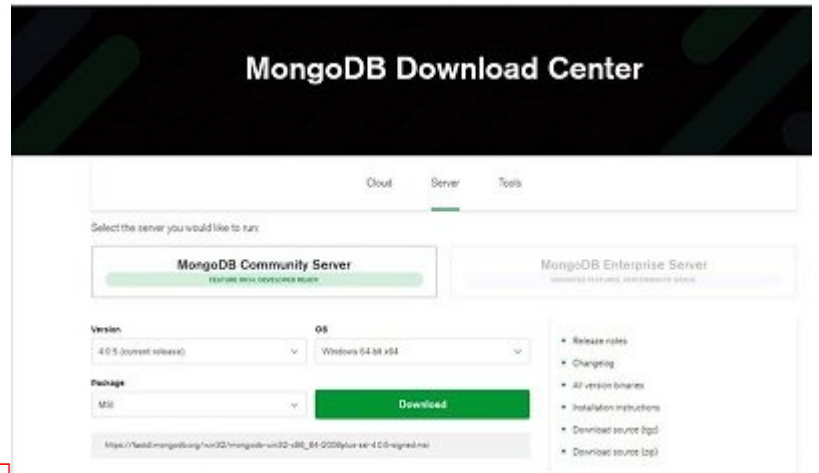


The Node.js logo, featuring the word "node" in a stylized, dark grey font. The letter "o" is replaced by a green, three-dimensional cube. Below the word "node" is a green hexagon containing the letters "JS" in a white, sans-serif font, followed by a registered trademark symbol (®).

: □□□□ □□ □□□□□ □□□□□ □□□ □□□□

<https://www.mongodb.com/download-center/community>

■ MongoDB 2020 年 12 月 10 日 10:00 分 現在、MongoDB 4.0.10 のインストールが完了しました。 (1 分)



MangoDB インストール 手順 1.1

■ MongoDB 2020 年 12 月 10 日 10:00 分 現在、MongoDB 4.0.10 のインストールが完了しました。 (1 分)  
C:\Program) インストール 手順 1.1  
■ MongoDB 2020 年 12 月 10 日 10:00 分 現在、MongoDB 4.0.10 のインストールが完了しました。 (1 分)

インストール 手順



インストール 手順 1.1

[MongoDB 4.0.10 のインストール 手順 1.1](#)

**mongod** と **mongo** のインストール 2.1

■ Node 2020 年 12 月 10 日 10:00 分 現在、MongoDB 4.0.10 のインストールが完了しました。 (1 分)  
nodetest1 インストール 手順 1.1  
nodetest1 インストール 手順 1.1  
mkdir data  
Express インストール 手順 1.1  
MongoDB インストール 手順 1.1





.object ( { &quot;nInserted&quot; : 1  
:object ( { &quot;nInserted&quot; : 1  
(db.usercollection.find()).pretty  
object ( { &quot;nInserted&quot; : 1  
(db.usercollection.find()).pretty  
.object ( { &quot;nInserted&quot; : 1

```

Command Prompt - mongo.exe
http://groups.google.com/group/mongodb-user
Server has startup warnings:
2015-02-01T11:05:25.885-0800 I CONTROL [initandlisten]
2015-02-01T11:05:25.885-0800 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2015-02-01T11:05:25.885-0800 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2015-02-01T11:05:25.885-0800 I CONTROL [initandlisten]
***
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).
The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URI with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.
To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
***
> use nodetest1
switched to db.nodetest1
> db.usercollection.insert({ "username" : "testuser1", "email" : "testuser1@testdomain.com" })
WriteResult({ "nInserted" : 1 })
> db.usercollection.find().pretty()
{ "_id" : ObjectId("5c54a2fb2b0a28907d595b4"),
  "username" : "testuser1",
  "email" : "testuser1@testdomain.com"
}

```

object ( { &quot;nInserted&quot; : 1

object ( { &quot;nInserted&quot; : 1

object ( { &quot;nInserted&quot; : 1  
:object ( { &quot;nInserted&quot; : 1  
(db.usercollection.find()).pretty  
object ( { &quot;nInserted&quot; : 1  
(db.usercollection.find()).pretty  
.object ( { &quot;nInserted&quot; : 1

```

newstuff = [{ "username" : "testuser2", "email" : "testuser2@testdomain.com" }, { "username" :
"testuser3", "email" : "testuser3@testdomain.com" }]
db.usercollection.insert(newstuff);

```

object ( { &quot;nInserted&quot; : 1  
:object ( { &quot;nInserted&quot; : 1  
(db.usercollection.find()).pretty  
object ( { &quot;nInserted&quot; : 1  
(db.usercollection.find()).pretty  
.object ( { &quot;nInserted&quot; : 1

```

Command Prompt - mongo.exe
> db.usercollection.find().pretty()
{ "_id" : ObjectId("5c54a2fb2b0a28907d595b4"),
  "username" : "testuser1",
  "email" : "testuser1@testdomain.com"
}
> newstuff = [{ "username" : "testuser2", "email" : "testuser2@testdomain.com" }, { "username" : "testuser3", "email" :
"testuser3@testdomain.com" }]
> db.usercollection.insert(newstuff);
WriteResult({ "nInserted" : 2,
  "writeErrors" : [ ],
  "writeConcernErrors" : [ ],
  "nInserted" : 2,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})

```

object ( { &quot;nInserted&quot; : 1

object ( { &quot;nInserted&quot; : 1

object ( { &quot;nInserted&quot; : 1  
:object ( { &quot;nInserted&quot; : 1  
(db.usercollection.find()).pretty  
object ( { &quot;nInserted&quot; : 1  
(db.usercollection.find()).pretty  
.object ( { &quot;nInserted&quot; : 1

**Node** □□ □□□□□ □□□□ □□□□ .5 □□□□□

00000000 00 .000 0000 00 00 00000000 0000 00000000 00 0000 00000 000000 0000 000 00000 000 00  
 :000 000 000 00 000 0000 0000

```
<ul>
  <li><a href="mailto:testuser1@testdomain.com">testuser1</a></li>
  <li><a href="mailto:testuser2@testdomain.com">testuser2</a></li>
  <li><a href="mailto:testuser3@testdomain.com">testuser3</a></li>
</ul>
```

[illegible]

```
var logger = require('morgan');
```

: □□□□ □□□□ □□ □□□□ □□ □□ □□ □□□□ □□□□ □□□□

```
var express = require('express');
var path = require('path');
var favicon = require('serve-favicon');
var logger = require('morgan');
var cookieParser = require('cookie-parser');
var bodyParser = require('body-parser');

// New Code
var mongo = require('mongodb');
var monk = require('monk');
var db = monk('localhost:27017/nodetest1');
```

00000000 00000000 MongoDB 00 Monk 0000 00 00000000 000000 00000000 00 app.js 0000 00 0000 000000 000000  
 00000000 000000 27017 0000 0000 .000 localhost:27017/nodetest1 00000000000000 00 00000000 0000 .0000  
 0000 000000 00 app.js 0000 00 0000 00 000000 .000 MongoDB 00000000000000 00 00000000 00000000 0000  
 :000000 00000000 0000 00 0000

```
app.use('/', indexRouter);
app.use('/users', usersRouter);
```

app.js  
 app.use  
 .  
 Express  
 :

```
// Make our db accessible to our router
app.use(function(req,res,next){
  req.db = db;
  next();
});
```

app.use('/',) 是 最常用 的 中间件 函数 之一。它 接收 一个 路由 规则 和 一个 中间件 函数 作为 参数。Mongo 数据库 中间件 函数 通常 会 接收 一个 db 对象 和 一个 中间件 函数 作为 参数。;(indexRouter 是 一个 中间件 函数，它 接收 一个 路由 规则 和 一个 中间件 函数 作为 参数。app.use 是 一个 中间件 函数，它 接收 一个 路由 规则 和 一个 中间件 函数 作为 参数。Monk 是一个 中间件 函数，它 接收 一个 路由 规则 和 一个 中间件 函数 作为 参数。app.is 是一个 中间件 函数，它 接收 一个 路由 规则 和 一个 中间件 函数 作为 参数。

이제 이 코드를 실행하면 .env 파일을 통해 MongoDB 연결 정보를 설정하고 HTTP 서버를 실행할 수 있습니다. 이 코드는 app.js 파일에 포함되어 있습니다. 실행하려면 node app.js 명령어를 입력하면 됩니다.

```
var createError = require('http-errors');
var express = require('express');
var path = require('path');
var cookieParser = require('cookie-parser');
var logger = require('morgan');

// New Code
var mongo = require('mongodb');
var monk = require('monk');
var db = monk('localhost:27017/nodetest1');

var indexRouter = require('./routes/index');
var usersRouter = require('./routes/users');

var app = express();

// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'jade');

app.use(logger('dev'));
app.use(express.json());
app.use(express.urlencoded({ extended: false }));
app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));

// Make our db accessible to our router
app.use(function(req,res,next){
  req.db = db;
  next();
});

app.use('/', indexRouter);
app.use('/users', usersRouter);

// catch 404 and forward to error handler
app.use(function(req, res, next) {
  next(createError(404));
});

// error handler
app.use(function(err, req, res, next) {
  // set locals, only providing error in development
  res.locals.message = err.message;
  res.locals.error = req.app.get('env') === 'development' ? err : {};

  // render the error page
  res.status(err.status || 500);
  res.render('error');
```

```
});  
module.exports = app;
```

## 00000 000000 0 **Mongo** 00 0000000 000000 **.6** 000000

0 index 00000000 00 00 00000 0000 00000 .00000 0000 00 C:\node\nodetest1\routes\index.js 00000 000000 00  
.00000 0000000 00000 0000 00 00000 00000 000000 .000000 000000000 helloworld/

```
/* GET Userlist page. */  
router.get('/userlist', function(req, res) {  
  var db = req.db;  
  var collection = db.get('usercollection');  
  collection.find({}, {}, function(e, docs) {  
    res.render('userlist', {  
      "userlist" : docs  
    });  
  });  
});
```

00 00000 .000000000 0000000 **HTTP** 000000000 00 00 00 00000 .000000 000000000 **db** 00 000000000 00000 0000 00  
000000 .000000 00000000000 000000000000 0000000000 00 **docs** 000000 00000000000 00000 **db** 000000 00 0000 000000  
00 000000 0000 00 00 000000000 0000 000000000 00000000 00 00 0000 0000 000000000 000000 00 00 00000 00000000  
**docs** 0000000 00000 00 000000 00000000000 0000000000 00000 00000 00 00 0 00000 000000000 **usercollection** 0000000  
.00000 00000

00000 00 00 000000 0000 00 .00000 00000000 00 **userlist** 00000000 00000000 00 000000000 00 000000 0000 000000 00  
0 00000 0000 0000 000000 00 00000000 00 00000 000000 00 00 00 **userlist** 000000 00 0000000 00000 **Jade** 00 00000000  
00000 00 .00000 0000000 **Jade** 00000 00 00000 000000 .0000 0000000 000000 00 00 00000000000 00000 0000000000  
**userlist.jade** 0000 00 00 00000 0000 0000000 .00000 0000 00 **index.jade** 00000 0 00000 \C:\node\nodetest1\views  
(.00000 00000 00000 0000 00000 00 0000 0000000) .00000 00000000 0000 00000 00 00 0000 00000 000000 .00000 0000000

```
extends layout  
block content  
h1.  
  User List  
ul  
  each user, i in userlist  
    li  
a(href="mailto:#{user.email}")= user.username
```

00000000 00000 **userlist** 00000 00000 00 00 00 00000000000 00000000 0000000 00000 0000 00 00 000000 00000 00 0000  
00000000 00 0000 00 00000000 0000 0 0000000 00000 00 0000000 0000 (000000000 0000) 0000000000 00 0000 0 00000 00000000  
0000000 00 0000000000 0000000 0000 00 **i** 0000000000 00 000000000 .000000 000000 0000 **html** 000000 000000 00 00000 0 00000  
00000000 0000000 00 00000 0000 0000000 .00000000 00 00 0000000 0000000 0000 00 0000000 000000000 000000000  
000000 00 0 0000000 00 00 .000000 **Node** 000000000000 0 000000 000000 0000000 00 00000 0 00000000  
0000000 00 0000 000000000 0 000000 0000000 00 00000 0000000 **Ctrl+C** 0000000000 000000 00 0 000000 C:\node\nodetest1  
.000000 000000 00 00 00000000000 0000 0000000

C:\node\nodetest1>npm start

0000000 00 00 0000000000 0000000 00 0000000 <http://localhost:3000/userlist> 00000 00 0 00000 0000 00 0000 00000000  
(7 0000) .000000 000000000 000000000000





000000000 00000 00 00 0000 .7 000

.00000 0000 0 000000 00 000000000000

.0000 0000 00 0000 000 00 0000 0 0000 0000000 000000000000 00 00 0000000 00000000000 000000

00000000000000 00 00000000 0000 00000000 0000

00 GET 000000 00 0000 000000 00000 0000 .00000 0000 00000000 000000000 0000 0000 00000000 000000  
.0000 000000 00 000 0000000 000000 0000 0000 .000 0000000 00 POST 000000

000000 00000000 0000 .1 000000

.0000 000000 00000000 0000000 00 0000 00 0000 00 0 0000 000000 00 000000 0 0000 0000 00 0000 0000 00  
000000 00 0000 0000 000000000 0000000 0000000 90 0000 00 00 0000 0000 0000000 0000 0000 00 0000 0000  
000000 0000 00000000 0000 00000 000000 0000 000000 00 .00000000 00 0000000000000 000000 000000 00000 0000  
(module.exports) 00000 0000 00 0000000 00 0000 00 0000 000000 0 00000 0000 00 routers/index.js 00000 .00000  
.00000 00000

```
/* GET New User page. */
router.get('/newuser', function(req, res) {
  res.render('newuser', { title: 'Add New User' });
});
```

0000 00 00 00 0 00000 0000 00 index.jade 00000 View 00000 00 .000000 00000 0000 00 00 0000 000000 00  
.00000 00000 0000 0000000 00000000 00 0000 000000 0 00000 000000 newuser.jade

extends layout

block content

h1= title

form#formAddUser(name="adduser",method="post",action="/adduser")

input#inputUserName(type="text", placeholder="username", name="username")

input#inputUserEmail(type="text", placeholder="useremail", name="useremail")

button#btnSubmit(type="submit") submit

00000 000000 00 00000 POST 0000 0000 0000 .000000000 000000 fromAddUser 000000 00 00000 00 00000 0000 00  
00000 0000 .000000000 000000 00000 00 00 000000 000000 00000 00 000000 00 .0000 adduser 00 000000 00 action  
00000000 00 0000 0000 0000000 <http://localhost:3000/newuser> 00000 00 0 00000 00000000 000000000000 00 Node  
.00000000 00000000 8 0000



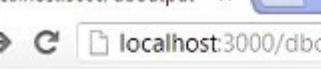


localhost:3000/dbinput

## Add User

noderocks arocks@rockingnode.com submit

□□□ □□□□□□□□□□ □□ □□□□ □□□



localhost:3000/dboutput x

localhost:3000/dboutput

## User List

- [testuser1](#)
- [testuser2](#)
- [testuser3](#)
- [noderocks](#)

[illegible][illegible]

000000 00 000000 00 MongoDB 00000000000000 00 000000 00 Jade 0 Node.js 0 Expree 00 00000 00000 00  
 00 00000000000 0 0000000 00 00000 00 0000000000 00000 00000 00000000000000 00 00 00 000000  
 00 0000 .000000000000 0000 00000000 00000 00 00 00000 000000 0000 000000 0000 .00000 000000 00000000000000  
 00 0000000000 000000000 00000000 00000 0000 0000 00 0000000 000000 00 0000000000 0 0000000000 00 00000000  
 0000000000 00000 000000 0000 .0000000000 0000 00 00000000000000 00 000000 00000 0 000000 00000000 00 00000000  
 000000 000000 00 00000 00 0000 00 .000000 00000 0000 00 000 000000000 000000000000 00000 00000 00 0000  
 .00000000 0000 00 00000000000 000000000 00000 00 00000 00000000

•

: □□□□ □□□□

[illegible]

07:30 - 19/02/1398

• ☐ ☐ ☐ ☐ ☐

[Express](#) - [MongoDB](#) - [Node.js](#) - [API](#) - [60](#) - [API](#)

□□□□□

<https://www.shabakeh-mag.com/workshop/15217/%D8%A8%D8%AF%D9%88%D9%86%D9%87:%D8%AF%D8%B1-60-%D8%AF%D9%82%DB%8C%D9%82%D9%87-%DB%8C%D8%A9-%D8%B5%D9%81%D8%AD%D9%87-%D9%88%D8%A8-%D8%A8%D8%B3%D8%A7%D8%B2%DB%8C%D9%85%D8%9F-%D8%A8%D8%AE%D8%B4-%D9%BE%D8%A7%DB%8C%D8%A7%D9%86%DB%8C>