

6주차 오픈소스SW개발도구 활용

20185138 신영수

- 소스코드
- MongoDB Atlas 설정
 - 계정 생성
 - 무료 공유 클라우드 데이터베이스 생성
 - IP 접속 허용
- 문제 001
 - 로그인 폼 작성
- 문제 002
 - req.body undefined 문제 발생
 - 분석
- 문제 003
- 문제 004

소스코드

<https://github.com/vl0011/HallymOpenSourceSW>

MongoDB Atlas 설정

계정 생성

MongoDB Atlas

- ✓ **Work with your data as code**
Documents in MongoDB map directly to objects in your programming language. Modify your schema as your apps grow over time.
- ✓ **Focus on building, not managing**
Let MongoDB Atlas take care of the infrastructure operations you need for performance at scale, from always-on security to point-in-time recovery.
- ✓ **Simplify your data dependencies**
Leverage application data for full-text search, real-time analytics, rich visualizations and more with a single API and minimal data movement.

sign up

See what Atlas is capable of for free

First Name*
YoungSoo

Last Name*
Shin

Company

Email*
vl0114@naver.com

Password*

☐ I agree to the [Terms of Service](#) and [Privacy Policy](#).

Create your Atlas account

or

[Sign up with Google](#)

MongoDB Atlas 홈페이지에서 계정을 생성 했습니다.

무료 공유 클라우드 데이터베이스 생성

영수's Org - ... Access Manager Billing

Project 0 Atlas App Services Charts

DEPLOYMENT

Database

Data Lake PREVIEW

DATA SERVICES

Triggers

Data API

Data Federation

SECURITY

Quickstart

Database Access

Network Access

Advanced

New On Atlas 2

영수'S ORG - 2022-11-13 > PROJECT 0

Security Quickstart

To access data stored in Atlas, you'll need to create users and set up network security controls. [Learn more about security setup](#)

1 How would you like to authenticate your connection?

Your first user will have permission to read and write any data in your project.

Username and Password Certificate

Create a database user using a username and password. Users will be given the *read and write to any database privilege* by default. You can update these permissions and/or create additional users later. Ensure these credentials are different to your MongoDB Cloud username and password.

Username

Enter username

Password

Enter password

Autogenerate Secure Password Copy

Create User


ID/PW 인증 방식의 유저를 생성 했습니다.


Username

Authentication Type

vl0011

Password

 EDIT

 REMOVE

IP 접속 허용

Only an IP address you add to your Access List will be able to connect to your project's clusters. You can manage existing IP entries via the [Network Access Page](#).

IP Address

Description

Enter IP Address

Enter description

Add My Current IP Address


Add Entry


IP Access List

Description

210.115.253.92/32

My IP Address

 EDIT

 REMOVE

현재 접속중인 IP를 입력 했습니다.

한림대학교에서 작업한 화면이라 한림대학교 IP대역이 나타났습니다.

문제 001

로그인 폼 작성

```

const express = require('express');
const app = express();

app.listen(8080, function() {
  console.log('listening on 8080')
})

app.get('/', function(req, res) {
  res.sendFile(__dirname + '/index.html')
})

app.get('/write', function(req, res) {
  res.sendFile(__dirname + '/write.html')
})

```

001 소스코드로 다음과 같은 구조의 소스코드 입니다.

| 메소드 | 요청 | 응답 |
|-----|--------|------------|
| GET | / | index.html |
| GET | /write | write.html |

write.html 은 로그인 폼임을 확인할 수 있습니다.

```

<form>
  <div class="mb-3">
    <label for="exampleInputEmail1" class="form-label">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp">
    <div id="emailHelp" class="form-text">We'll never share your email with anyone else.</div>
  </div>
  <div class="mb-3">
    <label for="exampleInputPassword1" class="form-label">Password</label>
    <input type="password" class="form-control" id="exampleInputPassword1">
  </div>
  <div class="mb-3 form-check">
    <input type="checkbox" class="form-check-input" id="exampleCheck1">
    <label class="form-check-label" for="exampleCheck1">Check me out</label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>

```

| 태그 | 타입 | 내용 |
|----|----|----|
|----|----|----|

| 태그 | 타입 | 내용 |
|--------|----------|--------------|
| Input | Text | E Mail |
| Input | Text | Password |
| Input | Checkbox | Check me out |
| Button | Submit | Submit |

문제 002

로그인폼에서 작성한 form 태그에서 submit button 을 클릭 하면
form 의 내용을 post 메소드 로 서버에 요청을 보냅니다.

이를 처리하기위 index.js 에 다음 내용을 추가하여 유저의 로그인 폼의 내용을
로그로 출력하도록 구성 했습니다.

```
app.post('/add', function(req, res){
  console.log(req.body);
  res.send('complete....')
});
```

write.html 에서 /add 주소로 form 의 내용을 전달하도록 form 의 action 을 설정 했습니다.

```
<form action="/add" method="POST">
  <div class="mb-3">
    <label for="exampleInputEmail1" class="form-label">Email address</label>
    <input type="email" class="form-control" name="email" id="exampleInputEmail1" aria-
describedby="emailHelp">
    <div id="emailHelp" class="form-text">We'll never share your email with anyone else.</div>
  </div>
  <div class="mb-3">
    <label for="exampleInputPassword1" class="form-label">Password</label>
    <input type="password" class="form-control" name="password" id="exampleInputPassword1">
  </div>
  <div class="mb-3 form-check">
    <input type="checkbox" class="form-check-input" id="exampleCheck1">
    <label class="form-check-label" for="exampleCheck1">Check me out</label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

req.body undefined 문제 발생

```
listening on 8080
undefined
undefined
```

```
console.log(req.body);
```

위 명령어 에서 req.body 가 undefined 로 출력되는 문제가 발생 했습니다.

분석

```
app.post('/add', function(req, res){
+   console.log(req.get('Content-Type'))
    console.log(req.body);
    res.send('complete....')
});
```

```
listening on 8080
application/x-www-form-urlencoded
{}
Content-Type: application/x-www-form-urlencoded
```

우선 요청 헤더의 Content-Type 을 확인하여 application/x-www-form-urlencoded 임을 확인 했습니다.
또한, Chrome Network 분석으로 요청 헤더의 Content-Type 을 확인하여 application/x-www-form-urlencoded 임을 확인 했습니다.

```
const express = require('express');
const app = express();

app.listen(8080, function() {
    console.log('listening on 8080')
})

+ app.use(express.urlencoded({ extended: true })); // x-www-form-urlencoded post body 처리 미들웨어

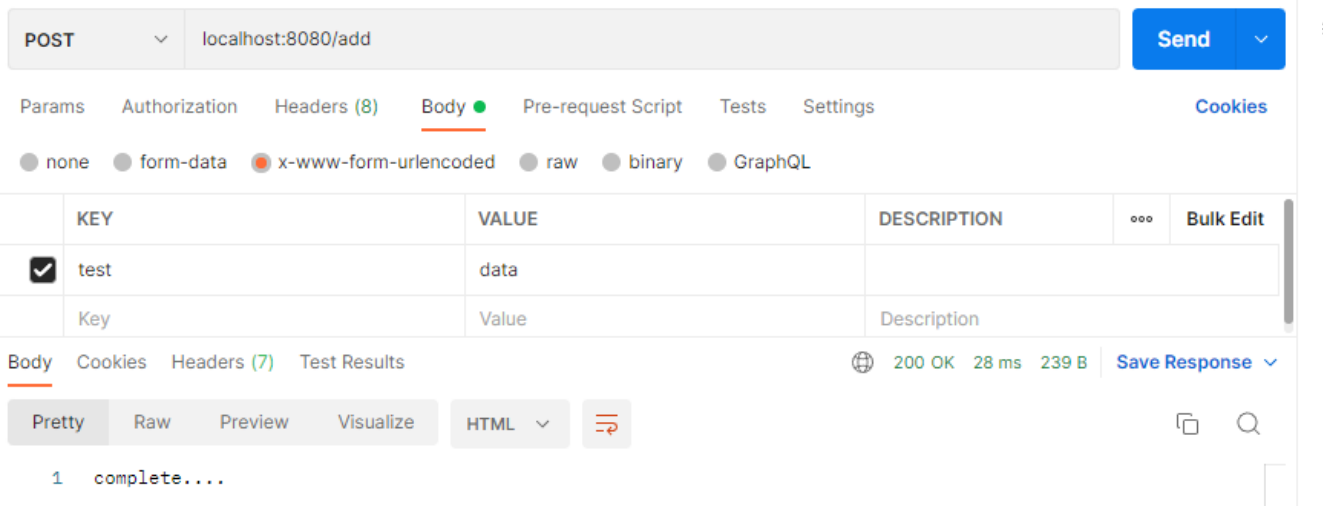
app.get('/', function(req, res) {
    res.sendFile(__dirname + '/index.html')
})

app.get('/write', function(req, res) {
    res.sendFile(__dirname + '/write.html')
})

app.post('/add', function(req, res){
    console.log(req.body)
    res.send('complete....')
});
```

x-www-form-urlencoded 파싱 미들웨어를 추가 한 후
/add 엔드포인트에 write.html 과 Postman 을 사용하여

로그를 확인해 보았습니다.



```
listening on 8080
{ test: 'data' }
```

우선 Postman에서의 테스트 환경에서 결과는 정상적으로 출력 되었습니다.

```
listening on 8080
{ test: 'data' }
{ }
```

그러나 write.html 에서는 빈 오브젝트가 리턴 되었습니다.

클라이언트 웹 페이지에서 문제가 있다고 판단되어 Postman으로 간단한 웹서버를 열어 클라이언트가 보내는 요청의 body를 분석해 보았습니다.

| TIME | REQUEST | RESPONSE |
|------------------------------------|---------|----------|
| Today, 11:01 pm | POST / | Default |
| > REQUEST HEADERS | | |
| > REQUEST BODY | | |
| This request does not have a body. | | |

요청 바디의 내용이 없음을 확인할 수 있었습니다.

input 태그에 name을 설정하지 않아서 생긴오류로 name을 설정하여 문제를 해결 했습니다.

```
<form action="/add" method="POST">
  <div class="mb-3">
    <label for="exampleInputEmail1" class="form-label">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp"
name="email">
    <div id="emailHelp" class="form-text">We'll never share your email with anyone else.</div>
  </div>
  <div class="mb-3">
    <label for="exampleInputPassword1" class="form-label">Password</label>
    <input type="password" class="form-control" id="exampleInputPassword1" name="password">
  </div>
  <div class="mb-3 form-check">
    <input type="checkbox" class="form-check-input" id="exampleCheck1" name="check">
    <label class="form-check-label" for="exampleCheck1">Check me out</label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

```
listening on 8080
```

```
{ email: 'ex@ex.com', password: 'pass', check: 'on' }
```

문제 003

Express 실행 전 MongoDB 가 실행되도록 설정 했습니다.



Connect to Cluster0

✓ Setup connection security

Choose a connection method

Connect

Connect to your application



MongoDB Drivers

Connect your application to your cluster using MongoDB's native drivers



MongoDB Atlas Data API

Access your Atlas data over HTTPS via a fully managed and secure API



Access your data through tools



MongoDB Compass

Explore, modify, and visualize your data with MongoDB's GUI



MongoDB Shell

Interact with your cluster using MongoDB's interactive Javascript interface



[Learn more](#)

Go Back

Close

×

Connect to Cluster0

✓ Setup connection security

✓ Choose a connection method

Connect

1

Select your driver and version

DRIVER

Node.js

VERSION

4.1 or later

2

Add your connection string into your application code

☐ Include full driver code example

mongodb+srv://vl0011:<password>@cluster0.hdy4v5g.mongodb.net/?retryWrites=true&w=majority

📋

Replace **<password>** with the password for the **vl0011** user. Ensure any option params are [URL encoded](#).

Having trouble connecting? [View our troubleshooting documentation](#)

Go Back

Close

MongoDB Drivers 를 클릭하여 node 에서 MongoDB 를 접속하기 위한 주소를 얻었습니다.

MongoDB URL을 설정하고, 문제 002에서 발견한 문제를 수정 했습니다.

```
const MongoClient = require('mongodb').MongoClient
app.use(express.urlencoded({ extended: true })); // x-www-form-urlencoded post body 처리 미들웨어

MongoClient.connect("mongodb+srv://vl0011:shin0114@cluster0.hdy4v5g.mongodb.net/?
retryWrites=true&w=majority", function(err, client){
  if (err) return console.log(err)
  app.listen(8080, function() {
    console.log('listening on 8080')
  })
})

PS C:\Users\vl0011\Desktop\hallym_univ_work\oss\003> npm run test

> 003@1.0.0 test
> node index.js

listening on 8080
```

이상없이 실행되었습니다. 4

문제 004

```
let db;
MongoClient.connect("mongodb+srv://vl0011:shin0114@cluster0.hdy4v5g.mongodb.net/?
retryWrites=true&w=majority", function(err, client){
  if (err) return console.log(err)
  db = client.db('nodejs');
  app.listen(8080, function() {
    console.log('listening on 8080')
  })
})
```

MongoDB 커넥션을 생성하여 다른 메소드가 사용할 수 있도록 전역변수 `db` 에 커넥션을 저장 했습니다.

```
app.post('/add', function(req, res){
  console.log(req.body);
  db.collection('login').insertOne({
    email: req.body.email,
    password: req.body.password,
    is_checked: (req.body.check) === "on"
  })
  res.send('complete....')
});
```

POST 요청 시 데이터베이스에 값을 추가하도록 수정 했습니다.

웹에서 로그인 요청을 하였고 로그의 출력과 MongoDB 에 추가된 값 입니다.

```
listening on 8080
{ email: 'test@email.com', password: 'qwer', check: 'on' }
```

```
_id: ObjectId('637102c3629e2c02fe820610')
email: "test@email.com"
password: "qwer"
is_checked: true
```