

1. 모듈 설치

```
npm install multer
```

2. template/admin/form.html

폼 전송타입 추가 (enctype)

```
<form action="" method="post" enctype="multipart/form-data">
```

```
{% set title = "제품 등록" %}
{% extends "layout/base.html" %}

{% block content -%}
    <form action="" method="post" enctype="multipart/form-data">
        <input type="hidden" name="_csrf" value="{{ csrfToken }}" />
        <table class="table table-bordered">
            <tr>
                <th>제품명</th>
                <td><input type="text" name="name" class="form-control">
            </tr>
            <tr>
                <th>가격</th>
```

필드 추가

```
<tr>
    <th>섬네일</th>
    <td>
        <input type="file" name="thumbnail" />
    </td>
</tr>
```

```

1  {% set title = "상점 등록" %}
2  {% extends "layout/base.html" %}
3
4  {% block content -%}
5      <form action="" method="post" enctype="multipart/form-data">
6          <table class="table table-bordered">
7              <tr>
8                  <th>이름</th>
9                  <td><input type="text" name="name" class="form-control" value="{{ shop.name }}">
10             </td>
11             </tr>
12             <tr>
13                 <th>섬네일</th>
14                 <td>
15                     <input type="file" name="thumbnail" />
16                 </td>
17             </tr>
18             <tr>
19                 <th>주소</th>
20                 <td><input type="text" name="address" class="form-control" value="{{ shop.address }}">

```

3. Shops 모델에 thumbnail 필드 추가

models/Shops.js

```
thumbnail : { type: DataTypes.STRING , comment: '상점 사진' },
```

```

1  const moment = require('moment');
2
3  module.exports = (sequelize, DataTypes) => {
4      const Shops = sequelize.define('Shops',
5          {
6              id: { type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true },
7              name : { type: DataTypes.STRING , comment: '상점명' },
8              thumbnail : { type: DataTypes.STRING , comment: '상점사진' } ,
9              address : { type: DataTypes.STRING , comment: '주소' },
10             location : { type: DataTypes.STRING , comment: '상세주소' },
11             phone : { type: DataTypes.STRING , comment: '전화번호' },
12             open_time : { type: DataTypes.STRING , comment: '운영시간' },
13             cell_phone : { type: DataTypes.STRING , comment: '핸드폰번호' }
14         }
15     );

```

app.js -> drop -> sync

위치 : app.js

```
// DB authentication
db.sequelize.authenticate()
.then(() => {
  console.log('Connection has been established')
  // return db.sequelize.sync();
  return db.sequelize.drop();
})
.then(() => {
  console.log('DB Sync complete.');
```

```
[nodemon] starting `node app.js`
Express listening on port 3000
Executing (default): SELECT 1+1 AS result
Connection has been established successfully.
Executing (default): DROP TABLE IF EXISTS `ProductsMemo`;
Executing (default): DROP TABLE IF EXISTS `Products`;
DB Sync complete.
█
```

```
// DB authentication
db.sequelize.authenticate()
.then(() => {
  console.log('Connection has been est
  return db.sequelize.sync();
  // return db.sequelize.drop();
})
.then(() => {
  console.log('DB Sync complete.');
```

4. multer middleware 작성

위치 : /controllers/admin/index.js

```
//이미지 저장되는 위치 설정
const path = require('path');
const uploadDir = path.join( __dirname , '../..../uploads' ); // 루트의 uploads위치에 저장한다.

//multer 셋팅
const multer = require('multer');
const storage = multer.diskStorage({
  destination : (req, file, callback) => { //이미지가 저장되는 도착지 지정
    callback(null, uploadDir );
  },
  filename : (req, file, callback) => { // shops-날짜.jpg(png) 저장
    callback(null, 'shops-' + Date.now() + '.'+ file.mimetype.split('/')[1] );
  }
});
const upload = multer({ storage: storage });
```

```
1  const { Router } = require('express');
2  const router = Router();
3  const ctrl = require('./admin.ctrl');
4
5  //이미지 저장되는 위치 설정
6  const path = require('path');
7  const uploadDir = path.join( __dirname , '../..../uploads' ); // 루트의 uploads위치에 저장한다
8
9  //multer 셋팅
10 const multer = require('multer');
11 const storage = multer.diskStorage({
12   destination : (req, file, callback) => { //이미지가 저장되는 도착지 지정
13     callback(null, uploadDir );
14   },
15   filename : (req, file, callback) => { // shops-날짜.jpg(png) 저장
16     callback(null, 'shops-' + Date.now() + '.'+ file.mimetype.split('/')[1] );
17   }
18 });
19 const upload = multer({ storage: storage });
20
21 router.get('/shops', ctrl.get_shops );
22
23 router.get('/shops/write', ctrl.get_shops_write );
```

```
upload.single('thumbnail')
```

```
22
23 router.get('/shops/write', ctrl.get_shops_write );
24
25 router.post('/shops/write', upload.single('thumbnail') , ctrl.post_shops_write );
26
27 router.get('/shops/detail/:id', ctrl.get_shops_detail );
28
29 router.get('/shops/edit/:id', ctrl.get_shops_edit );
30
31 router.post('/shops/edit/:id', upload.single('thumbnail') , ctrl.post_shops_edit );
32
33 router.get('/shops/delete/:id', ctrl.get_shops_delete );
34
35 // 메뉴 작성
```

POST /admin/shops/write thumbnail 필드명 저장

위치 : /controllers/admin/admin.ctrl.js

```
req.body.thumbnail = (req.file) ? req.file.filename : "";
```

```
17 exports.get_shops_write = ( _ , res ) => {
18   res.render( 'admin/form.html' );
19 }
20
21 exports.post_shops_write = async (req,res) => {
22
23   try{
24     req.body.thumbnail = (req.file) ? req.file.filename : "";
25     await models.Shops.create(req.body);
26     res.redirect('/admin/shops');
27   }catch(e){
28     console.log(e);
29   }
30 }
31 };
32
```

POST /admin/shops/edit/:id

```

const fs = require('fs');
const path = require('path');
const uploadDir = path.join( __dirname , '.././uploads' );

//...종락

const shop = await models.Shops.findByPk(req.params.id);

if(req.file && shop.thumbnail){ //요청중에 파일이 존재 할시 이전이미지 지운다.
  fs.unlinkSync( uploadDir + '/' + shop.thumbnail );
}

// 파일요청이면 파일명을 담고 아니면 이전 DB에서 가져온다
req.body.thumbnail = (req.file) ? req.file.filename : shop.thumbnail;

```

```

73  exports.post_shops_edit = async(req, res) => {
74
75      const fs = require('fs');
76      const path = require('path');
77      const uploadDir = path.join( __dirname , '.././uploads' );
78
79      try{
80
81          const shop = await models.Shops.findByPk(req.params.id);
82
83          if(req.file && shop.thumbnail){ //요청중에 파일이 존재 할시 이전이미지 지운다.
84              fs.unlinkSync( uploadDir + '/' + shop.thumbnail );
85          }
86
87          // 파일요청이면 파일명을 담고 아니면 이전 DB에서 가져온다
88          req.body.thumbnail = (req.file) ? req.file.filename : shop.thumbnail;
89
90          await models.Shops.update(
91              req.body ,
92              {
93                  where : { id: req.params.id }
94              }

```

5. view 작성

template/admin/detail.html

```
{% if shop.thumbnail %}
  <p>
    
  </p>
{% endif %}
```

6. 모듈 분리

middleware/multer.js

```
//이미지 저장되는 위치 설정
const path = require('path');
const uploadDir = path.join( __dirname , '../uploads' ); // 루트의 uploads위치에 저장한다.

//multer 셋팅
const multer = require('multer');
const storage = multer.diskStorage({
  destination : (req, file, callback) => { //이미지가 저장되는 도착지 지정
    callback(null, uploadDir );
  },
  filename : (req, file, callback) => { // shops-날짜.jpg(png) 저장
    callback(null, 'shops-' + Date.now() + '.'+ file.mimetype.split('/')[1] );
  }
});
module.exports = multer({ storage: storage });
```

controllers/admin/index.js

```
const upload = require('../middleware/multer');
```

```
1  const { Router } = require('express');
2  const router = Router();
3  const ctrl = require('./admin.ctrl');
4
5  const upload = require('../..../middleware/multer');
6
7  router.get('/shops', ctrl.get_shops );
8
9  router.get('/shops/write', ctrl.get_shops_write );
```