

# 웹개발실무

과제

담당교수: 김희숙  
(jasmin11@hanmail.net)



# 과제02: Node.js(영화정보)

담당교수: 김희숙  
(jasmin11@hanmail.net)



제출방법:  
LMS [과제] 제출

## [과제02]

[과제02] 프로그램 작성하고 실행결과 제출하시오

1. http, formidable 모듈 사용
2. 폼 화면: 영화정보 입력하고 새 영화 추가 확인

1) 제출방법: LMS-과제

첨부파일로 제출

2) 제출파일명: 워드 파일 제출(소스와 웹브라우저 실행 화면캡처)  
(과제02-분반\_이름\_학번-제출날짜.doc)

소스 압축하여 제출(폴더 중에서 node\_modules 는 삭제하고 압축하시오)

# [HW02] POST 요청(영화정보+포스터업로드)

## ❖ POST 서버 실습

- 목록 출력
- POST 요청 처리
- 영화 포스터
- 포스터 업로드 기능

## ● 영화정보

- ✓ 영화제목(title)
- ✓ 영화감독(director)
- ✓ 연도(year)
- ✓ 포스터(poster)

← → ↺ localhost:3000

### Favorite Movie

-  Avata(제임스 카메론,2009)
-  Starwars1(조지 루카스,1977)
-  Interstella(크리스토퍼 놀란,2014)

새 영화 입력

- 영화 제목
- 영화 감독
- 연도
- 포스터  파일 선택

- ① package.json 확인  
**"formidable": "^1.2.6"**
  - ② npm install
- node app

# [HW02] POST 요청(영화정보+포스터업로드)

← → ↻ localhost:3000

## Favorite Movie

-  Avata(제임스 카메론,2009)
-  Starwars1(조지 루카스,1977)
-  Interstella(크리스토퍼 놀란,2014)

### 새 영화 입력

- 영화 제목
- 영화 감독
- 연도
- 포스터  파일 선택

← → ↻ localhost:3000

## Favorite Movie

-  Avata(제임스 카메론,2009)
-  Starwars1(조지 루카스,1977)
-  Interstella(크리스토퍼 놀란,2014)

### 새 영화 입력

- 영화 제목
- 영화 감독
- 연도
- 포스터  파일 선택

← → ↻ localhost:3000

## Favorite Movie

-  Avata(제임스 카메론,2009)
-  Starwars1(조지 루카스,1977)
-  Interstella(크리스토퍼 놀란,2014)
-  Notebook(닉 카사베츠,2020)

### 새 영화 입력

- 영화 제목
- 영화 감독
- 연도
- 포스터  파일 선택



## initialDB.json

```
// ./initialDB.json
[
  {
    "title": "Avata",
    "director": "제임스 카메론",
    "year": 2009,
    "poster": "/images/Avata.jpg"
  },
  {
    "title": "Starwars1",
    "director": "조지 루카스",
    "year": 1977,
    "poster": "/images/Starwars1.jpg"
  },
  {
    "title": "Interstella",
    "director": "크리스토퍼 놀란",
    "year": 2014,
    "poster": "/images/Interstella.jpg"
  }
]
```

## [실습] 영화정보(데이터베이스 사용 X)

1) 모듈 설치 (단, formidable 버전 **^1.2.6**)  
npm install formidable

```
// package.json
{
  "name": "quiz01_movie",
  "version": "1.0.0",
  "description": "",
  "main": "app.js",
  "scripts": {
    "test": "echo W\"Error: no test specifiedW\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "formidable": "^1.2.6"
  }
}
```

app.js

```
// ./quiz01_movie/app.js
var http = require('http');
var fs = require('fs');
var querystring = require('querystring');
var pathUtil = require('path');
var url = require('url');
var formidable = require('formidable');

var uploadDir = __dirname + '/upload';
var imageDir = __dirname + '/images';

var initialDB = fs.readFileSync('./initialDB.json');
var movieList = JSON.parse(initialDB);
```

app.js

```
var server = http.createServer(function (req, res) {  
  if ( req.method.toLowerCase() == 'get' && req.url == '/' ) {  
    showList(req, res);  
  }  
  else if ( req.method.toLowerCase() == 'get' ) {  
    // 이미지  
    var parsed = url.parse(req.url);  
    // 127.0.0.1:3000/images/Avata.jpg?12331  
    var path = __dirname + parsed.pathname; // /images/Avata.jpg  
    fs.access(path, function(err) {  
      if ( err ) {  
        res.statusCode = 404;  
        res.end('Not Found');  
        return;  
      }  
      var is = fs.createReadStream(path);  
      is.pipe(res);  
    });  
  }  
  else if ( req.method.toLowerCase() == 'post' ) {  
    addNewMovie(req, res);  
  }  
  else {  
    res.statusCode = 400;  
    res.end('Error');  
  }  
});
```



app.js

```
server.listen(3000, () => {
  console.log('server on : 3000 port')
});

function addNewMovie(req, res) {
  var form = new formidable.IncomingForm();
  form.keepExtension = true;
  form.uploadDir = uploadDir;
  form.parse(req, function(err, fields, files) {
    if ( err ) {
      res.statusCode = 404;
      res.end('Error');
      return;
    }

    var title = fields.title;
    var director = fields.director;
    var year = fields.year;

    var poster = files.poster;
    //poster.path;
    var ext = pathUtil.extname(posters.name);
    var newFileName = title + ext;
    var newPath = uploadDir + pathUtil.sep + newFileName;
```

app.js

```
fs.renameSync(posters.path, newPath);
var url = '/images/' + newFileName; // /images/Starwars7.jpg

var info = {
  title : title,
  director : director,
  year : year,
  poster : url
};
movieList.push(info);

res.statusCode = 302;
res.setHeader('Location', '.');
res.end();
});
}
```

app.js

```
function showList(req, res) {
  var html = '<html>';
  html += '<head>';
  html += '<meta charset="utf-8">';
  html += '<style>';
  html += 'form label { width:100px; display:inline-block; }'
  html += 'li img { height:100px }';
  html += '</style>';
  html += '</head>';
  html += '<body>';
  html += '<h1>Favorite Movie</h1>';
  html += '<div>';
  html += '<ul>';
  movieList.forEach(function(movie) {
    html += '<li>';
    if ( movie.poster ) {
      html += '';
    }
    html += movie.title + '(' + movie.director + ',' + movie.year + ')' + '</li>';
  });
  html += '</ul>';
  html += '</div>';
}
```

app.js

```
html += '<form method="post" action="." enctype="multipart/form-data">';
html += '<h4>새 영화 입력</h4>';
html += '<ul>';
html += '<li><label>영화 제목</label><input type="text" name="title"></li>';
html += '<li><label>영화 감독</label><input type="text" name="director"></li>';
html += '<li><label>연도</label><input type="number" name="year"></li>';
html += '<li><label>포스터</label><input type="file" name="poster"></li>';
html += '</ul>';
html += '<input type="submit" value="올리기">';
html += '</form>';

html += '</body>';
html += '</html>';
res.writeHead(200, { 'Content-Type': 'text/html; charset=utf-8' });
res.end(html);
}
```