

MEAN 스택 활용 웹 개발

Express 라우팅



2016년도 2학기 2학년 방과후학교

Express Routing 기능

- 라우팅(routing)은 요청이 들어오면, 요청에 맞는 미들웨어가 동작하도록 분배해주는 동작을 의미한다.
- Express의 라우팅은 HTTP 메소드에 따라, URL의 경로에 따라서 동작하는 미들웨어를 설정할 수 있다.

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

```
app.get('/', function (req, res) {  
  res.send('Get 요청!');  
});
```

```
app.post('/', function (req, res) {  
  res.send('Post 요청!');  
});
```

정규식을 이용한 라우팅

- `?` : 앞 문자가 0번 또는 1번
- `+` : 앞 문자 1번 이상 반복
- `*` : 임의의 문자 반복

```
// /abcd, /acd  
app.get('/ab?cd', function (req, res) { })
```

```
// /abcd, /abbbcd, /abbbbcd  
app.get('/ab+cd', function (req, res) { })
```

```
// /abcd, /abxcd, /ab123cd  
app.get('/ab*cd', function (req, res) { })
```

```
// /abe, /abcde  
app.get('/ab(cd)?e', function (req, res) { })
```

route() 함수를 이용한 라우팅

- 같은 경로에 메소드만 다른 경우 router() 함수를 이용하여 라우팅할 수 있다.
 - ✓ `app.get('/music', handler1);`
 - ✓ `app.post('/music', handler2);`
 - ✓ `app.put('/music', handler3);`
 - ✓ `app.delete('/music', handler4);`

```
app.route('/music')  
  .get(function (req, res) { })  
  .post(function (req, res) { })  
  .put(function (req, res) { })  
  .delete(function (req, res) { });
```

Router 클래스를 이용한 라우팅

- Router 클래스는 모듈화하고 마운트할 수 있는 route 핸들러를 만들기 위해 사용한다.

```
[ birds.js ] // 모듈화
var express = require('express');
var router = express.Router();

router.use(function timeLog(req, res, next) {
  console.log('Time: ', Date.now());
  next();
});
// /birds 호출 시
router.get('/', function(req, res) {
  res.send('Birds home page');
});
// /birds/about 호출시
router.get('/about', function(req, res) {
  res.send('About birds');
});

module.exports = router;
```

```
[ app.js ]
var birds = require('./birds');

...
// 마운트
app.use('/birds', birds);
```

라우팅 핸들러 모듈 분리

- /hello 요청에 대한 route 핸들러 모듈 작성 : hello.js
- /hi 요청에 대한 route 핸들러 모듈 작성 : hi.js
- app.js
 - ✓ `app.use('/hello', require('./hello'));` // hello.js route 모듈 사용
 - ✓ `App.use('/hi', require('./hi'));` // hi.js route 모듈 사용
- hello.js
 - ✓ `router.get('/', 함수명);`
- hi.js
 - ✓ `router.get('/', 함수명);`

에러 처리 미들웨어

- 에러 발생시 에러처리 미들웨어에게 위임하면 에러 처리가 중복되지 않으며, 에러 처리 방법을 일관적으로 작성할 수 있는 장점이 있다.
- 에러 처리 미들웨어는 미들웨어들 중 후순위에 위치하게 한다.

```
// 에러 핸들러 등록
app.use(errorHandler);

function task(req, res, next) {
  // 에러 발생시 처리
  var error = new Error('메시지');
  next(error);
  ...
}

// 에러 처리 미들웨어
function errorHandler(err, req, res, next) {
  res.status(500);
  res.send('error 발생');
}
```

[실습] 영화 관리 어플리케이션

- 라우팅 핸들러 모듈을 분리하여 작성해보자.
- postman을 이용하여 CRUD 테스트를 수행한다.
- 파일명 : 22_routing/movie/app.js, movie/routes/movie.js

