

#9 비밀 설정 정보 관리

- 몽고db 연결 시 아이디와 비밀번호가 노출됨

→ 연결 정보만 따로 떼어놓은 후, .gitignore에 해당 폴더정보를 저장해 git에 올라가지 않도록 처리

```
ex.
const mongoose = require('mongoose')
mongoose.connect('mongodb+srv://몽고DB아이디:몽고DB비번@cluster0.donwl.mongodb.net/?retryWrites=true&w=majority',{
  useNewUrlParser : true, useUnifiedTopology : true
}).then(()=> console.log('MongoDB Connected'))
.catch(err => console.log(err))
```

- 운영 환경은 Local Development/ Deploy(배포 후) 2가지로 간주.

개발에선 폴더에서 설정 내용을 가져올 수 있지만 배포 후에는 사이트에서 직접 관리(?)하므로 환경에 따라 분기 처리 해줌

→ 환경변수 process.env.NODE_DEV 사용하여 분기

→ 개발/운영일 때 각각 설정 정보를 가져오는 파일 dev.js, prod.js를 생성한 후, 환경변수를 통해 개발/운영 버전에 따라 해당 파일을 연동하는 key.js 파일 생성.

```
//boiler-plate/config/dev.js
module.exports = {
  mongoURI : 'mongodb+srv://nuriKim:snflwkfl1!@cluster0.donwl.mongodb.net/?retryWrites=true&w=majority'
}
```

```
//boiler-plate/config/prod.js
module.exports = {
  mongoURI : process.env.MONGO_URI
}
```

```
//boiler-plate/config/key.js
if(process.env.NODE_ENV == 'production'){
  module.exports = require('./prod');
}else{ //NODE_ENV == 'development'
  module.exports = require('./dev');
}
```

```
//boiler-plate/index.js
const config = require('./config/key');
const mongoose = require('mongoose')
mongoose.connect(config.mongoURI,{
  useNewUrlParser : true, useUnifiedTopology : true
}).then(()=> console.log('MongoDB Connected'))
.catch(err => console.log(err))
```

→ git ignore dev.js 추가해줌(git에 설정정보 올라가지 않도록)

```
//boiler-plate/.gitignore
node_modules

dev.js
```