

2022 부트캠프 032

- Heroku 연결+설정 / Heroku에 Atlas연결(blogDB)

!Tips & Links

- ※ 강의 <https://www.udemy.com/course/the-complete-web-development-bootcamp/>
- ※ Heroku Deploying NodeJS <https://devcenter.heroku.com/articles/deploying-nodejs>
- ※ Atlas <https://cloud.mongodb.com/v2/>

※ Heroku 연결 <https://devcenter.heroku.com/articles/getting-started-with-nodejs#deploy-the-app>

git init / git add . / git commit -m "initial commit here" / **heroku login** => logged in as ~~@~~
:로그인 확인 / **heroku create** 새 앱 생성 / touch Procfile / **Procfile에 web: node app.js** 작성 /
Listen 작성(리모트, 또는 리모트+로컬) / app에 engine작성 / .gitignore touch, 기본설정 복사 /
git push heroku master

※ Listen on the port:
포트 설정

only listen on process.env.PORT	app.listen(process.env.PORT);
리모트+로컬 포트 같이 쓰기 Heroku port and Local port	let port = process.env.PORT; if(port == null port == "") { port = 3000 // Heroku포트 없으면 로컬 }; app.listen(port, function(){ console.log("Server started") });

※ file set-up
환경 파일

package.json	.gitignore
language-specific set-up, nodeJS ver. // Specify the version of node "engines": { "node": "17.4.0" }, node —version해서 그대로 작성	/node_modules npm-debug.log .DS_Store // mac설정파일 /*.env
	상단과 같이 추가

- ※ heroku는 dependencies의 node module들을 확인 후 heroku의 서버 단(end)에서 설치(install)하여 작동
=> node module들은 .gitignore에 추가해준다.(편의성, 절약)
- ※ git push heroku master => deploy with URL, ex) <http://radiant-refuge-39602.herokuapp.com/>
- ※ blog를 서버에 연결 // 서버에 POST 완료되기 전에 redirect하는 예러 => save()의 callback으로 해결

```
npm i mongoose 몽구스설치 / const mongoose = require("mongoose"); 몽구스require
mongoose.connect("mongodb+srv://user1:pass1@cluster0.cw4wk.mongodb.net/testBlogDB") DB에 URI연결
const postSchema = new mongoose.Schema({ ~~ })    새 Schema
const Post = mongoose.model("Post", postSchema)    새 컬렉션
const newPost = new Post({ ~~ })                      괄호 + 중괄호(method)
newPost.save() /
```

※ title로 검색(기존)에서
변경> DB의 id로 검색

ejs수정
parameter수정
get에서 findById사용
res.render(해당 post)

- 1) a의 href를 <%= post.title %>에서 <%= post_id %>로 렌더되게 수정
- 2) 주소 parameter는 app.get("post/:post_id")로 바꿔줌
- 3) 주소에서 req.params.post_id로 리퀘스트하는 id(post_id)가져오고
- 4) Post.findById(리퀘스트한 id, (err, post)=>{ // _id 매칭 post찾기
 res.render('post', { // 찾은 post로
 title:title // response로 'post.ejs'를 render해줌
 })