

2022 부트캠프 035

- RESTful (특정documents) HTTP Verbs / .get(), .put(), .patch(), .delete()

!Tips & Links

※ 강의 <https://www.udemy.com/course/the-complete-web-development-bootcamp/>

※ res랑 log구분 > res는 클라이언트(html, postman) log는 서버사이드(터미널)

※ app.route()랑 .get() .post() 같은 다단에 놓고
세미콜론 마무리 잘하기

ex) app.route()	// route() 잘 확인
.get((req, res)=>{})	// 필요하면 줄 띄어쓰기
.post((req, res)=>{});	// 세미콜론 잘 간수하기!

※ RESTful: Handle a Specific Article. // `app.route("/articles/:제목")`

R E A D	.get() findOne()	모델.findOne({조건}, (err, 검색결과){ //검색결과 처리 })	.get(Article.findOne({title: req.params.title}, (err, foundArticle){ !err?res.send(foundArticle):""; })
	.put() replaceOne()	replace the entire document (by whole) 모델.replaceOne({조건}, {update내용}, (err, result)=>{});	.put(Article.replaceOne({title: req.params.title}, // 찾은 doc을 {title: req.body.title}, // 전체교체 {content: req.body.content})
U P D A T E	.patch() updateOne()	replace specific values 모델.updateOne({조건}, {\$set: update내용})	.patch(Article.updateOne({title: req.params.title}, {\$set: req.body},)
		req.body == \$set{ ~~ } // 구조가 똑같기 때문에, 동적인 표현 가능 \$set: {title: req.body.title, content: req.body.content} == \$set: req.body	
D E L E T E	.delete() deleteOne()	모델.deleteOne({조건}, 콜백)	.delete(Article.deleteOne({title: req.params.title}, //콜백)
	etc	빈 문서 지우기 // 콜백의 result.deletedCount로 지운 아이템 개수 확인가능 Article.deleteMany({title: {\$eq: null}}, {title: {\$eq: ""}}, (err)=>{!err?res.send("지우기 완료"):""};	

※ (mongoose) update()는 deprecated({overwrite: true} 썼었음) => updateOne은 rewrite 막혀있음(disallows)
☞ replaceOne() 사용 https://mongoosejs.com/docs/api.html#model_Model.updateOne (\$set 안됨 => 빈문서 생김)
updateOne() 사용 + mongoDB의 {\$set:{}} 사용하기

```
/* -----<lang: en>----- */
encompass[inˈkæmpəs] 포괄 / chaining methods /
by default Mongoose will prevent properties being overwritten and deleted.
try to build a project that demonstrates what you can do – based on your skills.
the interpersonal skills is very important / stack overflow reputation ~ doing some code reviews ~ etc.
```