

2022 부트캠프 029

- CustomLists(use Express Route Parameter) POST & DELETE / Model.find~() / \$pull배열

!Tips & Links

- ※ 강의 <https://www.udemy.com/course/the-complete-web-development-bootcamp/>
- ※ mongoose API <https://mongoosejs.com/docs/api.html>

\$pull / \$pullAll : operator => mongoose remove document from array (pulls out elements)
 ※ res.redirect()에 **res** 까먹지 않기 ※ document를 CREATE => 모델.create() / 모델.insertOne() / 모델.save()

Model. Find~	findOne()	조건에 매치하는 doc를 찾아서 return one document (object)
	findOneAndUpdate()	조건에 매치하는 doc를 찾아서 update 하단에 정리 참고
	findById()	찾는 id value에 매치하는 _id를 찾음
	findByIdAndRemove()	찾는 id value에 매치하는 _id를 찾아서 doc삭제 // findByIdAndRemove(id, ~):
.etc	countDocuments()	counts num of documents that matches filter // 조건에 맞는 Doc 개수 count
	exists()	returns a document with _id only,(or null) // 존재하면 _id만 return

Find One And Update()

```

모델.findOneAndUpdate (
  { conditions }, // 조건: { query }
  { updates }, // 수정: { 수정할 거 }
  { options }, // 옵션: doc참고
  functions(err, result){
    if(!err){redirect( ~ )} // 콜백: { redirect등 }
  });

모델.findOneAndUpdate (
  { name : listName }, // 모델(collection)에서 doc찾음
  { $pull: { items: { _id : chkdId } } }, // item배열의 { _id: ~ }
  functions(err, result){
    if(!err){
      res.redirect("/") + listName
    }
  });
  
```

Express Route Parameter

```

[ get("/:custom") ]
[ req.params.~ ]
  
```

```

app.get("/:customListUrl", function(req, res){
  const paramsList=req.params.customListUrl;
  List.findOne({name: paramsList}, function(err, foundList){
    if(!err){
      // name이 paramsList(customListUrl)
      if(!foundList){
        const list = new List({
          name: paramsList, items: defaultItems
        });
        list.save(); // callback한 list가 없다면
        res.redirect("/") + paramsList // defaultItems로 새 List저장
      } else {
        res.render('list', { // list.ejs를 render
          listTitle: foundList.name, // 찾은 list이름
          listItems: foundList.items // 찾은 list아이템
        });
      }
    }
  });
});

// app.get("/:parameter", function(){})
// :Express의 Route Parameter

// req.params.parameter
// { parameter : ~~ }로 log

이름 설정:
"/city:cityName" => { cityName: ~ }

// 모델.findOne({ name: parameter },
// function( err, foundList ){})
  
```

Custom List에서 POST

```

[ 모델.findOne ]
[ foundList.~.push() ]
  
```

```

app.post('/', function(req, res){ // POST
  let itemName=req.body.newListItem;
  let listName=req.body.list; // hidden input의 name과 value사용
  const newItem = new Item({~~}); // POST할 value => new 모델
  if(listName === "today"){ // 메인 "/"
    newItem.save(); // 바로 저장 => "/"로 이동
    res.redirect("/");
  } else {
    // Custom Lists일 때 아이템 추가
    List.findOne({name: listName}, function(err, foundList){ // 모델서 찾을
      if(!err){
        foundList.items.push(newItem); // document의 items배열에 push
        foundList.save(); // document를 다시 저장한다.
        res.redirect("/") + listName; // 해당 /list로 이동
      }
    });
  }
});

// post req => req.body.list를 확인
// 저장한 listName이 "/"와 같으면
// if(listName === "today") => "/"
// else, 다르면 Custom List
// List.findOne({name: listName})찾아
// foundList의 items에 push하고
// foundList를 저장함
// "/" + listName으로 redirect

<input type="hidden" name="listName"
value="<%= listTitle %>">
  
```

Custom List에서 Delete

```

[ findByIdAnd~ ]
[ findOneAnd~ ]
  
```

```

app.post("/delete", function(req, res){
  const chkdId = req.body.chk_box; // chk_box value(<%= item_id %>)
  const listName = req.body.listName; // listName vlaue(<%= listTitle %>)
  if(listName === "today"){ // 메인 "/"
    Item.findByIdAndRemove(chkdId, function(){ // id로 remove
      res.redirect("/");
    });
  } else {
    // Custom List
    List.findOneAndUpdate( // 찾아서 Update
      {name: listName}, // document 찾을
      { $pull: { items: { _id : chkdId } } }, // 배열에서 pull
      function(err, result){ if(!err){ res.redirect("/") + listName;}});
  }
});

// chkdId로 넘어온 id value 저장
// id로 ① findByIdAndRemove (items에서)
// ② findOneAndUpdate (lists에서)

// $pull : 배열에서 배열 요소 지우기
  
```

```

/* -----<lang: en>----- */
list categorize // functionality for to-do list
  
```