

2022 퍼블리셔 부트캠프 026

- Mongoose / Schema & Model / insertMany([model]) / find(=>{err, callback-data})

!Tips & Links

- ※ 강의 <https://www.udemy.com/course/the-complete-web-development-bootcamp/>
- ※ 몽구스JS <https://mongoosejs.com/> Schema <https://mongoosejs.com/docs/guide.html#schemas>
Model <https://mongoosejs.com/docs/models.html>

Schema	= maps to a collection, defines the shape of the doc
Model	to use a schema definition -> need a Model to work with. for collection name, mongoose will automatically lowercase + pluralize your singular model name . ex) const Model = mongoose.model('Model', schema) => models

※ Schema & Model 사용

```
new mongoose.Schema({
  ~ : ~,
  ~ : ~
});
mongoose.model(
  modelName, schema
);
modelName.save();
```

```
const 스키마 = new mongoose.Schema({
  name: String,
  age: Number
});
const 모델 = mongoose.model( '모델', 스키마 )
const model1 = new 모델 ({
  name: ' ~~ ',
  age: xx
});
모델.save();
```

스키마 / 모델의 구조를 정의

```
const 스키마 = new mongoose.Schema({ 키: 타입, ~~ });
```

새 모델 / 새 모델 하나 만들기

```
const Model = mongoose.model( Model , 스키마)
// collection name => models
```

값 넣기

```
const model1 = new Model ({ 키: 값, ~~ });
```

```
모델.save();
```

※ insertMany([~, ~, ~]) 배열 안에 모델(스키마 有)

```
Model.insertMany([ 모델1, 모델2, 모델3 ], function( err ){
  if( err ){ console.log(err); } else { console.log('data saved.') }
})
```

callbacks!

※ 컬렉션.find(); callback data=>foreach

```
Fruit.find((err, data)=>{
  if(err){
    console.log(err);
  } else {
    data.forEach(( each )=>{
      console.log( each.name );
    });
  }
});
```

find(=>{①err, ②callback-data})

// find()한 data는 object의 배열
// foreach로 각 index 확인하기

※ close(); 서버 닫기: 서버 작업이 끝난 위치에서 mongoose.connection.close(); -> 작업 후 연결종료

※ use db명 > show collections > db.collections.find()으로 db의 컬렉션의 document 전부 확인!

※ _id : default property to schemas, Objectid type.

/* -----<lang: en>----- */
singular[**siNGgyələr**]단수 <-> plurals[**plōōrəl**]복수 // underlying 아래 쪽의