

# 웹 서버 측 프로그래밍 2

## 4장 템플릿 엔진

- 템플릿 엔진

⇒ html을 동적으로 생성하는 엔진 : 다양한 layout 파일을 결합할 수 있음

⇒ 다양한 템플릿 엔진 : EJS, Pug, Handlebar

⇒ EJS 템플릿 엔진 : `<%=...>` 등의 형태 scriptlet 사용

⇒ 설치 : `npm install ejs --save`

⇒ 사용 등록 : `app.use('view engine', 'ejs');`

- EJS 사용하기

⇒ 템플릿 엔진 등록 : `app.set('view engine', 'ejs');`

⇒ 파일 보내기 : `res.render('index')` 메서드 사용, views 폴더에서 `index.ejs`를 찾아서 html로 변환 후 사용자에게 전송

- layouts

⇒ layout 파일 : 페이지들에 공통적으로 사용되는 내용을 저장하고 있는 파일 : `header`, `scripts`

⇒ layout 파일을 이용한 리팩터링 : views 폴더에 layouts 폴더 작성

( `header.ejs`, `navbar.ejs`, `footer.ejs`, `scripts.ejs` )

⇒ `<%- .. %>` ejs raw output 태그 사용하여 layout 파일 포함 (`index.ejs`)

## mondoDB 소개

- mongoDB

: 대표적인 NoSQL 데이터 베이스 ⇒ relation DB가 아님, 다양한 기관에서 사용

## SQL v.s NoSQL

SQL	NoSQL
규칙에 맞는 데이터 입력	자유로운 데이터 입력
테이블 사이의 JOIN 지원	컬렉션 사이의 JOIN 지원하지 않음
안정성, 일관성	확장성, 가용성
테이블, 행, 열	컬렉션, 문서, 필드

- mongoDB 구조

⇒ 데이터 베이스: 컬렉션의 집합, 컬렉션: 응용에서의 하나의 entity, 문서: instance of entity

---

### Database

#### Product Collection

##### Product document

```
{  
  "price":30,  
  "title": "Web Programming",  
  "description": "best web programming text",  
  "publication date": "2020-3-15"  
}
```

##### Product Document

...

#### User Collection

##### User Document

```
{  
  "username": "123abc",  
  "contact": {  
    "phone": "111-222-3333",  
    "email": hello@world.com  
  }  
}
```

...

- mongoose : node.js에서 mongoDB를 연결하는 라이브러리

( 설치 : npm install mongoose)

⇒ DB 연결하기

<코드>

```
1  const mongoose=require('mongoose');
2  const BlogPost=require('./models/BlogPost')
3
4  mongoose.connect('mongodb://localhost/Blog_DB')
```

- 모델 정의하기

⇒ 스키마 정의 : 스키마는 하나의 컬렉션에 속하는 문서의 모양 정의

⇒ models 폴더 생성 후 blogpost.js 작성

```
1  const mongoose=require('mongoose');
2  const Schema=mongoose.Schema;
3
4  const BlogPostSchema=new Schema({
5    title:String,
6    body:String
7  });
```

⇒ 모델 정의

: 스키마를 이용하여 db의 컬렉션과 연결

<코드>

```
const BlogPost = mongoose.model('BlogPost' , BlogPostSchema );
module.exports=BlogPost
```

- DB를 연결하여 테스트 하기

<코드>

```

1  const mongoose=require('mongoose');
2  const BlogPost=require('./models/BlogPost')
3
4  mongoose.connect('mongodb://localhost/Blog_DB')
5
6  BlogPost.create({
7    title: "처음 포스팅하는 블로그",
8    body: "옛날에 금잔디 동산에 메기 같이 앉아서 놀던 곳. 물레방아 \
9    소리 들리다 메기, \
10   내 사랑하는 메기야."
11 }, (error, blogpost) => {
12   console.log(error, blogpost);
13 });

```

⇒ compass의 확인 결과

```

_id: ObjectId('638c6e6dc0be9bbb1c427526')
title: "처음 포스팅하는 블로그"
body: "옛날에 금잔디 동산에 메기 같이 앉아서 놀던 곳. 물레방아
__v: 0

```

| \_id 필드 자동 생성

- CRUD 연산

⇒ retrieve 연산

```

BlogPost.find({title:/포스팅/},(err, blogpost)=>{
  console.log(blogpost);
})

```

⇒ update 연산

```

BlogPost.findByIdAndUpdate("638c6e6dc0be9bbb1c427526",
  {title:'수정된 제목'}, (err, blogpost)=>{
    console.log(blogpost);
  })

```

⇒ delete 연산

```
BlogPost.findByIdAndDelete("638c6e6dc0be9bbb1c427526",  
  (err, blogpost)=>{  
    console.log(err, blogpost)  
  })  
)
```