**Chapter 07. 익스프레스 프로젝트를 모듈화 하기**

* 설정파일 만들기
* 모듈화 진행 시, 모듈이 수정되었거나 추가돼도 메인 파일에서는 이를 자동으로 인식할 수 있어야 한다.
  + - * **설정 파일을 만들고, 메인 파일이 설정 파일을 불러오도록 한다**(설정 파일만 수정해도 메인 파일에서 변경된 내용을 알 수 있도록)
* config.js : 설정 정보 정의한 후, 서버 실행 시 설정을 로드하도록 한다.
  + - * 서버 정보(포트, url) / DB정보(연결 url, 인증 정보) / 스키마 생성 파일의 로딩 정보 / 라우팅 로딩 정보 포함

1. config.js는 module.exports에 객체 한 개를 할당하는 구조  
   객체 안에 우선 기존 사용하던 server\_port와 db\_url을 선언한다.  
   그리고 app2.js에서 config 모듈을 require로 로드한 다음 app.set에서 ‘port’를 설정
2. DB 스키마 정보 선언  
   config.js 내의 db\_schemas 속성은 배열이며, file / collection / schemaName / modelName을 설정한다.
3. DB 스키마 생성 실행  
   [database/database.js] 설정 : 이 파일에서는 config의 db\_url을 사용해 DB를 연결하고, 연결 성공 시 스키마 및 모델 생성하는 함수를 실행하는 역할을 완료한다.

connect(app, config) 함수를 실행하면 기본적인 DB 연결 🡪 createSchema( ) 함수 실행. 파라미터 config는 사전 정의된 config.js를 넣어주게 될 것이다.

* + - * createSchema( ) 함수를 실행하면 config.db\_schemas 속성의 길이를 산출하고 var 문을 통해 스키마 배열을 curItem에 할당한다. 그리고 curItem.file(== user\_schema.js)의 createSchema(mongoose)를 실행해 리턴된 스키마를 curSchema에 저장한다.
      * 그리고 curItem.collection / curSchema를 이용해 모델을 생성한다.
      * 생성된 curSchema와 curModel을 database 객체의 속성으로 넣어준다. 그리고 app.set(‘database, database)로 app객체에 database 객체를 설정한다.

그리고 module.exports = database 선언해주면 이후 req.app.get(‘database’) 본 객체 참조 가능.

* 이후 유저 스키마에 새로운 정보를 넣고 싶다면, user\_schema.js에 스키마 추가 후 config.js에 스키마 객체 배열을 추가한다.

1. 라우팅 모듈 분리

config.js에 route\_info 배열 객체를 생성한다. file / path / method / type