**Chapter 06. 데이터베이스 사용하기**

* 몽구스로 DB 다루기(app3.js 사용)
* NoSQL 컬렉션의 경우, 속성이 다를 수 있어 RDB처럼 조회 조건을 공통으로 적용하기 어렵다. e.g)users 컬렉션에 ‘name’속성이 있는 문서와 없는 문서가 있을 수 있음
* 이런 문제 때문에 **Schema(스키마)**를 만들고 스키마에 따라 문서 객체를 저장한다. Schema는 DB의 구조를 사전에 (엑셀의 칼럼처럼) 정의하는 것을 의미한다.
* Schema를 통해 DB의 문서객체를 Javascript의 객체로 바꿀 수 있다. 이 때, Object Mapper를 통해 이렇게 DB 문서 객체를 Javascript 객체로 변환할 수 있다.
* 기존에는 MongoClient.connection( .. )에 databaseUrl을 전달해 데이터베이스에 연결했는데, 몽구스 사용 시에는 데이터 베이스 연결 함수에서 mongoose.connect(databaseUrl)을 사용한다. database 변수에는 mongoose.connection을 할당하고, 이 database변수를 이용해 연결 이벤트를 관리하는 방식으로 DB 연결을 관리한다.
* 연결 데이터베이스 open 이벤트 발생 시, 우선 스키마를 정의 한다(id, name, password)
* 스키마 정의 후 User Model을 정의한다.  
  UserModel = mongoose.model(‘users’, UserSchema);

즉, 모델에 ‘users’ 컬렉션을 설정하고 상기 선언한 User Schema를 사용하도록 설정한다.

* Mongoose의 model( ) 메소드는 **모델명(==컬렉션 명과 매칭됨)**, 스키마를 파라미터로 받는다. (UserModel과 UserSchema를 변수로 설정해두면 다른 함수에서도 사용이 가능하다)
* 몽구스로 사용자 인증
* 기존 authUser는 users 컬렉션에서 직접 find( )했지만, 모델 객체를 사용할 경우 **모델객체를 사용해서 만들어낸 인스턴스 객체 내에 정의된** DB 컬렉션 문서 조회/수정/삭제 메소드를 사용한다.
* 사용자 인증 시 UserModel.find( … )는 즉시 사용할 수 있지만, save( )를 통해 유저를 추가할 경우 인스턴스 생성 시 사용이 가능하다!