

(가이드)

장표 종류



오리엔테이션 장표

전체 강의 중 1번 / 강사마다 1번 등장



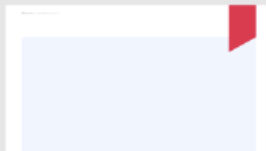
챕터 시작 장표

현재 15챕터까지 제작완료



화면전환 장표

클립 속의 클립, 부제등 사용 가능



내지 장표

기본 장표, 내지, 본문, PPT내용 부분

예시 : (1) 배경 속 숫자 = 챕터 넘버
(2) 빨간색 텍스트 숫자 = 클립 넘버



초격차 패키지 Online.

안녕하세요. 시그니처 백엔드 강의 Course 2. 백엔드 웹 개발 입문/실전을 진행하는 예상국입니다.

[Course 2] 백엔드 웹 개발 입문/실전

PART1 | 웹 개발 입문과 데이터베이스

PART2 | 웹 서비스 개발 실전

PART3 | 최신/심화 웹 개발 실전

Spring Data JPA

Spring Data 소개

JDBC

6.

JPA

JDBC란? (Java DataBase Connectivity)

자바 언어로 데이터베이스 프로그래밍을 하기위한 라이브러리

JDBC

6.

JPA

```
String jdbcDriver = "com.mysql.cj.jdbc.Driver";
String jdbcUrl = "jdbc:mysql://localhost:3306/user?serverTimezone=Asia/Seoul";
try {

    Class.forName(jdbcDriver).newInstance();
    Connection con = DriverManager.getConnection(jdbcUrl, "root", "root123!@#");
    Statement st = con.createStatement();

    String sql = "SELECT * FROM user";
    ResultSet rs = st.executeQuery(sql);

    while(rs.next()){
        String name = rs.getString(1);
        String age = rs.getString(2);
        String email = rs.getString(3);

        System.out.println(name + " " + age + " " + email);
    }

    rs.close();
    st.close();
    con.close();

} catch (Exception e) {
    e.printStackTrace();
}
```

JPA







JPA란 JAVA ORM(Object Relational Mapping) 기술에 대한 인터페이스

ORM

객체와 데이터베이스의 관계를 맵핑 하는 방법

EX)

```
public class User {  
    private String name;  
    private int age;  
    private String email;  
}
```

Result Grid				
Filter Rows: <input type="text"/>				
Edit:   				
Export/Import:  				
Wrap Cell Content: 				
	id	name	age	email
▶	1	홍길동	10	hong@gamil.com
	2	유관순	20	y@gmail.com
	3	강감찬	20	kang@gmail.com
*	NULL	NULL	NULL	NULL

Hibernate

JPA 의 인터페이스를 구현한 라이브러리

Hibernate 외에 EclipseLink, DataNucleus, OpenJPA, TopLink 등등등

```
public class User {  
    private String name;  
    private int age;  
    private String email;  
}
```

```
EntityManager entityManager= entityManager.getTransaction().begin();
```

```
var user = new User("홍길동", 20, "hong@gmail.com");
```

```
entityManager.persist(user);  
entityManager.getTransaction().commit();  
entityManager.close();
```


Spring Data JPA

Hibernate 외에 등등 어떠한 라이브러리를 써도 반복되는 작업의 발생

이를 편리하게 사용하고, Transaction 관리도 Spring에서 관리해주는 형태

```
@Transactional
public User save(User user) {
    return userRepository.save(user);
}
```

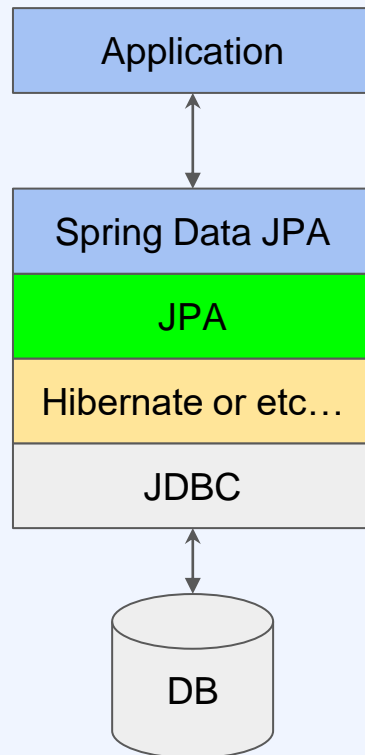
```
@Transactional
@Override
public <S extends T> S save(S entity) {

    Assert.notNull(entity, "Entity must not be null.");

    if (entityInformation.isNew(entity)) {
        em.persist(entity);
        return entity;
    } else {
        return em.merge(entity);
    }
}
```

Spring Data JPA

6. JPA



Spring Data 적용 및 개발

application.yaml

```
spring:
  jpa:
    show-sql: true
    properties:
      hibernate:
        format_sql: true
        dialect: org.hibernate.dialect.MySQL8Dialect
    hibernate:
      ddl-auto: validate
  datasource:
    url: jdbc:mysql://localhost:3306/user?useSSL=false&useUnicode=true&allowPublicKeyRetrieval=true
    driver-class-name: com.mysql.cj.jdbc.Driver
    username: root
    password: root1234!!
```