

2. API 개발 고급 - 준비

#1.인강/jpa활용편/활용편2/강의

- /소개
- /조회용 샘플 데이터 입력

소개

API 개발 고급 전반을 소개

조회용 샘플 데이터 입력

API 개발 고급 설명을 위해 샘플 데이터를 입력하자.

- userA
 - JPA1 BOOK
 - JPA2 BOOK
- userB
 - SPRING1 BOOK
 - SPRING2 BOOK

```
package jpabook.jpashop;

import jpabook.jpashop.domain.*;
import jpabook.jpashop.domain.item.Book;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Component;
import org.springframework.transaction.annotation.Transactional;

import javax.annotation.PostConstruct;
import javax.persistence.EntityManager;

@Component
@RequiredArgsConstructor
public class InitDb {

    private final InitService initService;
```

```
@PostConstruct
```

```
public void init() {  
    initService.dbInit1();  
    initService.dbInit2();  
}
```

```
@Component
```

```
@Transactional
```

```
@RequiredArgsConstructor
```

```
static class InitService {
```

```
    private final EntityManager em;
```

```
    public void dbInit1() {
```

```
        Member member = createMember("userA", "서울", "1", "1111");  
        em.persist(member);
```

```
        Book book1 = createBook("JPA1 BOOK", 10000, 100);  
        em.persist(book1);
```

```
        Book book2 = createBook("JPA2 BOOK", 20000, 100);  
        em.persist(book2);
```

```
        OrderItem orderItem1 = OrderItem.createOrderItem(book1, 10000, 1);  
        OrderItem orderItem2 = OrderItem.createOrderItem(book2, 20000, 2);  
        Order order = Order.createOrder(member, createDelivery(member),  
orderItem1, orderItem2);  
        em.persist(order);  
    }
```

```
    public void dbInit2() {
```

```
        Member member = createMember("userB", "진주", "2", "2222");  
        em.persist(member);
```

```
        Book book1 = createBook("SPRING1 BOOK", 20000, 200);  
        em.persist(book1);
```

```
        Book book2 = createBook("SPRING2 BOOK", 40000, 300);  
        em.persist(book2);
```

```
        Delivery delivery = createDelivery(member);
```

```

        OrderItem orderItem1 = OrderItem.createOrderItem(book1, 20000, 3);
        OrderItem orderItem2 = OrderItem.createOrderItem(book2, 40000, 4);
        Order order = Order.createOrder(member, delivery, orderItem1,
orderItem2);
        em.persist(order);
    }

    private Member createMember(String name, String city, String street,
String zipcode) {
        Member member = new Member();
        member.setName(name);
        member.setAddress(new Address(city, street, zipcode));
        return member;
    }

    private Book createBook(String name, int price, int stockQuantity) {
        Book book = new Book();
        book.setName(name);
        book.setPrice(price);
        book.setStockQuantity(stockQuantity);
        return book;
    }

    private Delivery createDelivery(Member member) {
        Delivery delivery = new Delivery();
        delivery.setAddress(member.getAddress());
        return delivery;
    }
}

```

참고: 주문 내역 화면에서는 회원당 주문 내역을 하나만 출력했으므로 하나만 노출된다.