

# ( 디지털 컨버전스 ) 스마트 콘텐츠와 웹 융합 응용 SW개발자 양성과정

훈련기간 : 2021.05.07 ~ 2021.12.08

Debug

디버깅버튼을 누르고 빨간색표시를 누른다. 맵핑된 url을 요청해본다. 값이 잘들어오는 지 확인해본다.

랜덤 던전을 요청한 클라이언트한테 보내기위한 처리

```
@Autowired
private JdbcTemplate jdbcTemplate;

public List<Dungeon> randomList() throws Exception {

    private final int DUNGEON_NUM = 3 으로 초기화되어있음
    int cnt = 0; 요청한 곳에 랜덤 던전을 3개 보내기위해 정해놓음
    int[] randArr = new int[DUNGEON_NUM];

    List<Dungeon> tmp; 던전타입의 리스트를 만든다.

    int max = jdbcTemplate.queryForObject( 현재 테이블의 튜플갯수를 max값으로 초기화한다.
    sql: "select count(1) from vuedungeon", Integer.class);

    for(int i = 0; i < DUNGEON_NUM; i++) {
        randArr[i] = (int) (Math.random() * max) + 1;
    }

    List<Dungeon> results = new ArrayList<Dungeon>();
}
```

저장할 배열 선언

selcct count(10) from vuedungeon과 selcct max from vuedungeon 의차이는 max의 경우 데이터의 수가 아닌 마지막 번호만 기억하고 번호를 반환하지만 count의 경우는 갯수를 반환한다.

```
List<Dungeon> results = new ArrayList<Dungeon>();

for (int i = 0; i < DUNGEON_NUM; i++) { 3번 반복 하고 특정정보를 읽어오는 read와 구조유사
    tmp = jdbcTemplate.query(
        sql: "select dungeon_no, name, description, monster_amount, reg_date " +
            "from vuedungeon where dungeon_no = ?",
        new RowMapper<Dungeon>() {
            @SneakyThrows
            @Override
            public Dungeon mapRow(ResultSet rs, int rowNum) throws SQLException {

                log.info("rowNum: " + rowNum);

                Dungeon dungeon = new Dungeon();

                dungeon.setDungeonNo(rs.getInt( columnLabel: "dungeon_no"));
                dungeon.setName(rs.getString( columnLabel: "name"));
                dungeon.setDescription(rs.getString( columnLabel: "description"));
                dungeon.setMonsterAmount(rs.getInt( columnLabel: "monster_amount"));

                dungeon.setRegDate(rs.getDate( columnLabel: "reg_date"));

                return dungeon;
            }
        }, randArr[i] 랜덤번호에가 쿼리조건문에 dungeon_no = ? 에 들어가서 랜덤 던전에 대한 정보를 가져온다.
    );

    results.add(tmp.get(0));

    리스트형태로 값을 저장한다. tmp.get(0)
    (tmp값이 누적이아니라 초기화됨에 get(0))

return results;
}
```

중복데이터 처리는 없기때문에 클라이언트쪽에서 index값에 주의해야한다.

# Debugging

디버깅버튼을 누르고 빨간색표시를 누른다. 맵핑된 url을 요청해본다. 값이 잘들어오는 지 확인해본다.

2

```
for(int i = 0; i < DUNGEON_NUM; i++) { i: 0  
    randArr[i] = (int) (Math.random() * max) + 1; randArr: [0, 0, 0] max: 5 i: 0
```

1

www.localhost:7777/vuedungeon/randomAlloc

3

4

"http-nio-7777-exe...up "main": RUNNING

randomList:37, VueDungeonRepository (com.example.demo.repo  
invoke:-1, VueDungeonRepository\$\$FastClassBySpringCGLIB\$\$4c  
invoke:218, MethodProxy (org.springframework.cglib.proxy)  
invokeJoinpoint:779, CglibAopProxy\$CglibMethodInvocation (org  
proceed:163, ReflectiveMethodInvocation (org.springframework.a  
proceed:750, CglibAopProxy\$CglibMethodInvocation (org.spring  
invoke:137, PersistenceExceptionTranslationInterceptor (org.spring  
proceed:186, ReflectiveMethodInvocation (org.springframework.a  
proceed:750, CglibAopProxy\$CglibMethodInvocation (org.spring  
intercept:692, CglibAopProxy\$DynamicAdvisedInterceptor (org.s  
randomList:-1, VueDungeonRepository\$\$EnhancerBySpringCGLIB  
randomList:20, VueDungeonServiceImpl (com.example.demo.ser  
getRandomListsDungeon:30, VueDungeonController (com.examp  
invoke0:-1, NativeMethodAccessorImpl (jdk.internal.reflect)

this = {VueDungeonRepository@6726}

cnt = 0

randArr = {int[3]@6725} [1, 0, 0]

0 = 1

1 = 0

2 = 0

max = 5

i = 1

jdbcTemplate = {JdbcTemplate@6727}

randArr[i] = 0

[

{  
 "dungeonNo": 2,  
 "name": "뇌전도원",  
 "description": "폭풍 벼락 매우위험한 던전",  
 "monsterAmount": 5,  
 "regDate": "2021-07-23"  
},  
{  
 "dungeonNo": 5,  
 "name": "잊혀진 고대의 신전",  
 "description": "섬광탄 스팀 조심",  
 "monsterAmount": 3,  
 "regDate": "2021-07-23"  
},  
{  
 "dungeonNo": 5,  
 "name": "잊혀진 고대의 신전",  
 "description": "섬광탄 스팀 조심",  
 "monsterAmount": 3,  
 "regDate": "2021-07-23"  
}

results.add(tmp.get(0));  
}  
  
return results;

Frames

"http-nio-7777-exe...up "main": RUNNING

randomList:67, VueDungeonRepository (com.example.demo.repo  
invoke:-1, VueDungeonRepository\$\$FastClassBySpringCGLIB\$\$4c  
invoke:218, MethodProxy (org.springframework.cglib.proxy)  
invokeJoinpoint:779, CglibAopProxy\$CglibMethodInvocation (org  
proceed:163, ReflectiveMethodInvocation (org.springframework.a  
proceed:750, CglibAopProxy\$CglibMethodInvocation (org.spring  
invoke:137, PersistenceExceptionTranslationInterceptor (org.spring  
proceed:186, ReflectiveMethodInvocation (org.springframework.a  
proceed:750, CglibAopProxy\$CglibMethodInvocation (org.spring  
intercept:692, CglibAopProxy\$DynamicAdvisedInterceptor (org.s  
randomList:-1, VueDungeonRepository\$\$EnhancerBySpringCGLIB  
randomList:20, VueDungeonServiceImpl (com.example.demo.ser

this = {VueDungeonRepository@6726}

cnt = 0

randArr = {int[3]@6725} [4, 5, 1]

max = 5

results = {ArrayList@6772} size = 2

0 = {Dungeon@6777} "Dungeon(dungeonNo=4, name=망자들의 쉼터, description=망령 계열의 몬스터 출몰 던전, monsterAmount=2, regDate=2021-07-23)"

1 = {Dungeon@6798} "Dungeon(dungeonNo=5, name=잊혀진 고대의 신전, description=섬광탄 스팀 조심, monsterAmount=3, regDate=2021-07-23)"

i = 2

tmp = {ArrayList@6796} size = 1

0 = {Dungeon@6805} "Dungeon(dungeonNo=1, name=불타는 대지, description=끝없이 화염이 폭풍이 몰아치는 과거 전장 지역, monsterAmount=2, regDate=2021-07-23)"