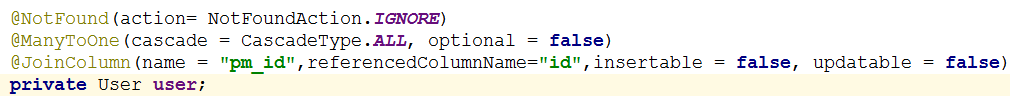
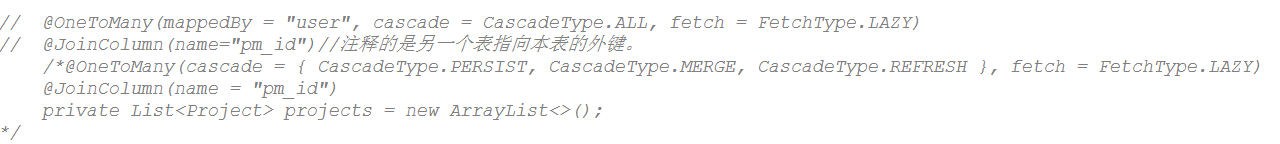
使用jpa自带的@Query查询(Project与User对应，多对一)

@Query(" select p from Project p left join p.user u where p.pmId=u.id")

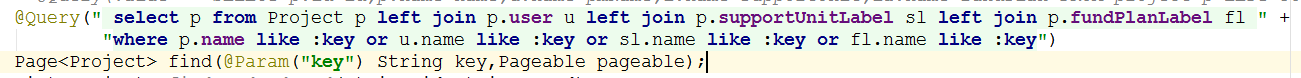
Project实例内



User实例内不要project对象(只需要Project内有User对象就可以，不然查询会报错)



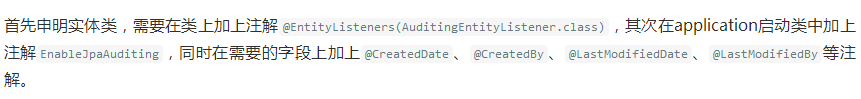
高级自主查询，终究是探索出来了奥妙



更为复杂的查询Predicate

|  |
| --- |
| Sort sort = new Sort(Sort.Direction.DESC, order);  Pageable pageable = new PageRequest(page - 1, rows, sort);  return demandRepository.findAll(new Specification<Demand>() {  public Predicate toPredicate(Root<Demand> root, CriteriaQuery<?> query, CriteriaBuilder cb) {  List<Predicate> andPredicates = new ArrayList<>();  query.distinct(true);//查询不重复项  if(null != state){  if(110==state){  Join<Solution, Demand> join = root.join("solutions", JoinType.LEFT);//联表查询  andPredicates.add(cb.equal(join.get("state"), state/10));//按解决方案进度状态查询  }else{  andPredicates.add(cb.equal(root.get("state"), state));//按需求进度状态查询  }  }  if(null != onYear && !"".equals(onYear)){  andPredicates.add(cb.equal(root.get("onYear"), onYear));//按需求年份查询  }  if(null != type && !"".equals(type)){  andPredicates.add(cb.like(root.get("typeId"), "%"+type+"%"));//按需求类型查询  }  if(null != industry && !"".equals(industry)){  andPredicates.add(cb.equal(root.get("industryId"), industry));//按需求所属行业查询  }  if (null != key && !"".equals(key)){  andPredicates.add(cb.like(root.get("name"), "%" + key + "%"));//按需求名称模糊查询  }  Predicate[] andArrayPredicates = new Predicate[andPredicates.size()];  return cb.and(andPredicates.toArray(andArrayPredicates));  }  }, pageable); |
|  |
|  |

@CreatedDate、@LastModifiedDate的使用



|  |
| --- |
| @CreatedDate  @JsonSerialize(using = CustomDateSerializer.class)  private Date createdTime;//创建时间  @LastModifiedDate  @JsonSerialize(using = CustomDateSerializer.class)  private Date updatedTime;//更新时间  /\*\*  \* @description 自定义返回JSON 数据格中日期格式化处理  \*/  public class CustomDateSerializer extends JsonSerializer<Date> {  @Override  public void serialize(Date value,  JsonGenerator jsonGenerator,  SerializerProvider provider)  throws IOException, JsonProcessingException {  SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");  jsonGenerator.writeString(sdf.format(value));  }  } |

自动生成UUID

|  |
| --- |
| @Entity  @Table(name = "demand")  @GenericGenerator(name = "jpa-uuid", strategy = "uuid")  @EntityListeners(AuditingEntityListener.class)//设置时间自动添加  public class Demand implements Serializable {  @Id  @GeneratedValue(generator = "jpa-uuid")  private String id;//需求编号 |

多字段排序方式

|  |
| --- |
| Sort.Order top = new Sort.Order(Sort.Direction.DESC,"top"); Sort.Order orderType = new Sort.Order(Sort.Direction.DESC,order); List<Sort.Order> orders = new ArrayList<>(); orders.add(top); orders.add(orderType); Sort sort = new Sort(orders); |