1.orphanRemoval

删除关联对象按照官方给出的字面意思是：如果删除关系，是否删除关系上的实体，true表示会删除，false表示不删除。我们来举个例子说明一下，现在分别有User和Address两个实体，User类上有一个addresses的字段，表示引用多个Address实例，User和Address是一对多关系。大致的代码如下：  
public class User{  
 @OneToMany(orphanRemoval = true)  
 private Set<Address> addresses;  
}  
public class Address{

//…  
}

我们操作User实体，并删除一个address实例。

User user = em.find(User.class,1L);  
Address address = em.find(Address.class,1L);  
user.getAddresses().remove(address);  
em.save(user);

如果orphanRemoval = true，那么这个操作会删除address对象，如果为false，则会删除他们的关系，将address对user的引用设置为null。

2.CascadeType 级联关系

CascadeType.PERSIST 保存，常用  
CascadeType.MERGE 合并，常用  
CascadeType.REMOVE 删除  
CascadeType.ALL 包含所有

3.多对多级联保存报错

//先保存一次，这样Role才有id 不然直接多对多保存会导致Role一方id没有初始化

Role userRole = new Role("user");  
roles.add(superuserRole);  
roles.add(userRole);  
roleService.saveAll(roles);  
//superUser  
superuserRole.setPermissions(permissionService.findAll());  
//user  
userRole.setPermissions(permissionService  
 .findPermissionsByEntityName\_NameAndOperationLessThanEqual(Arrays.*asList*("User"), "MODIFY"));  
roleService.saveAll(roles);

4.jackson 序列化的问题

首先是jpa懒加载或者循环引用的问题

1 @JsonIgnore

可以过滤掉 不需要序列化的字段

2 @JsonIgnoreProperties(ignoreUnknown = true)

可以过滤类中不存在的字段,实际上不知道为什么不生效

所以最好不要随意用get为开头给方法命名