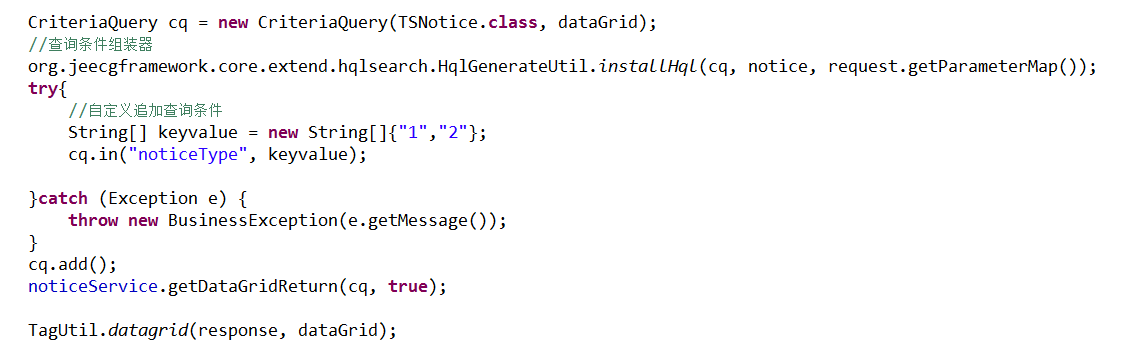
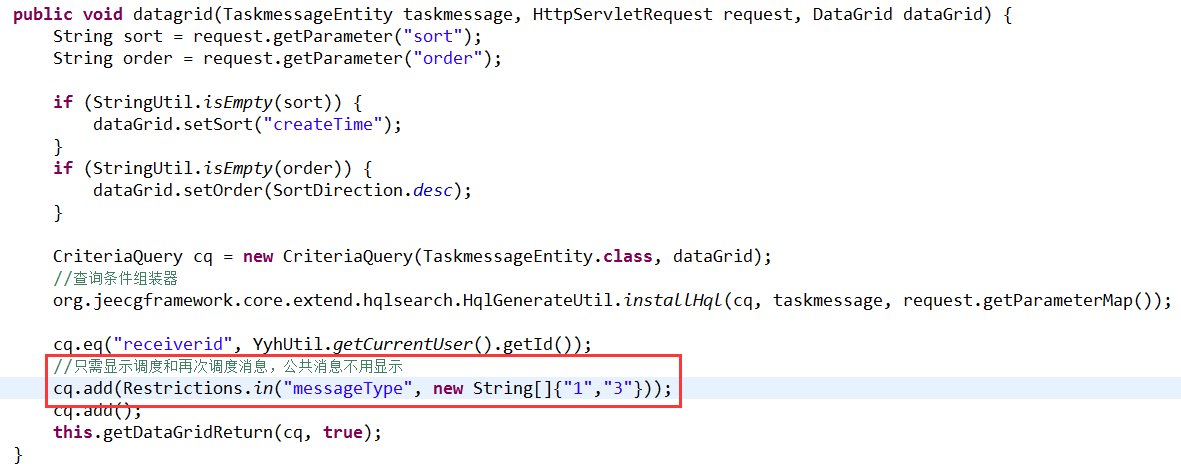
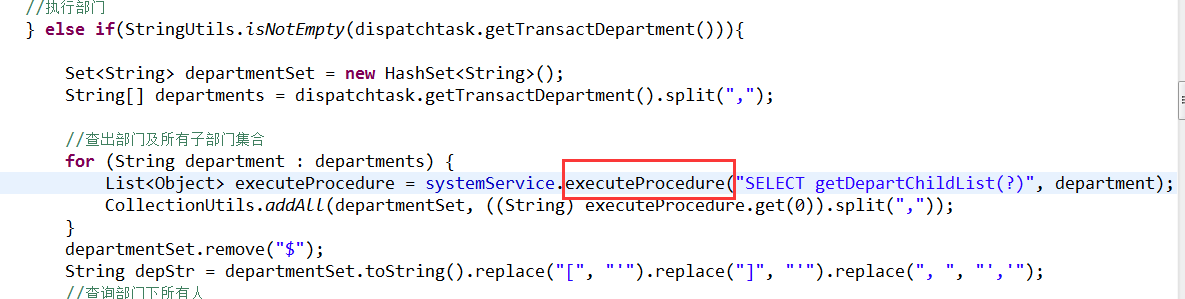
1.在jeecg 框架中对某个字段进行多值过滤。



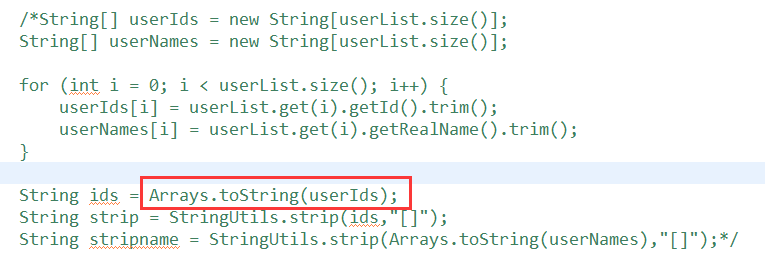
第二种



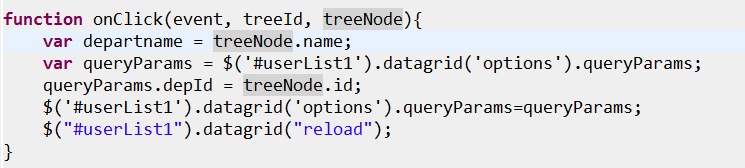
2.在jeecg 框架中执行存储过程。

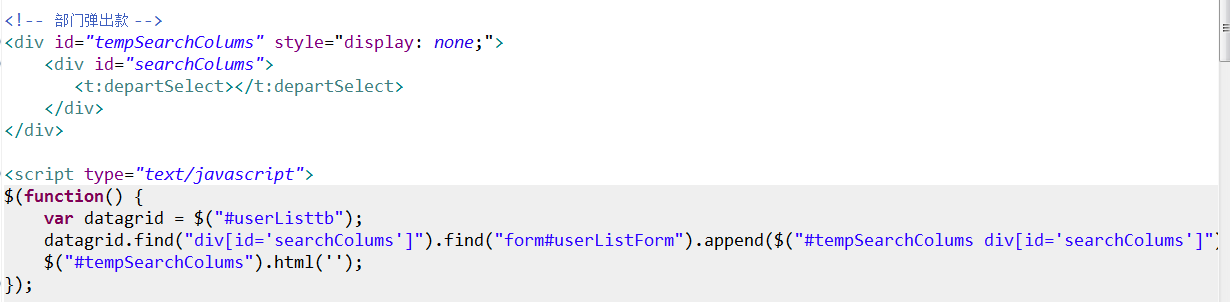


3.Arrays中toString方法的特性：默认在逗号后面加空格



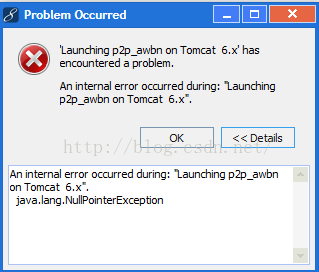
4.前端技术





5.[myeclipse 启动tomcat时报错：](http://blog.csdn.net/u012922219/article/details/42869361)

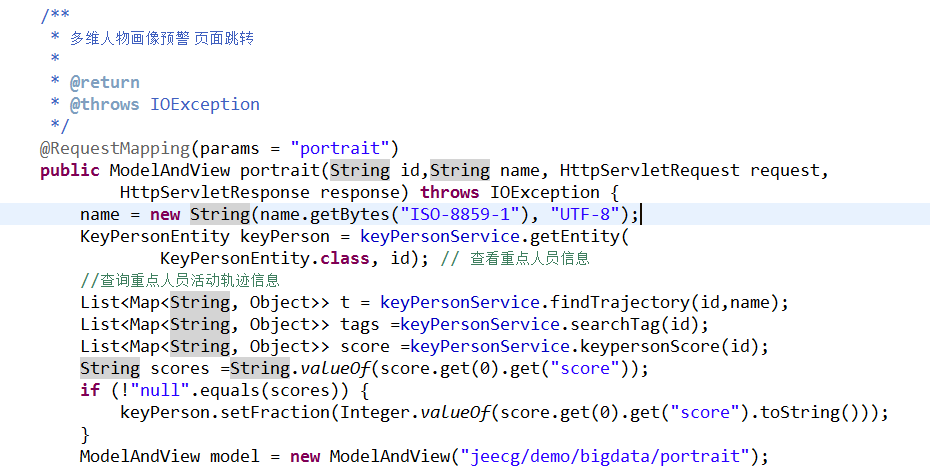
**java.lang.NullPointerException**



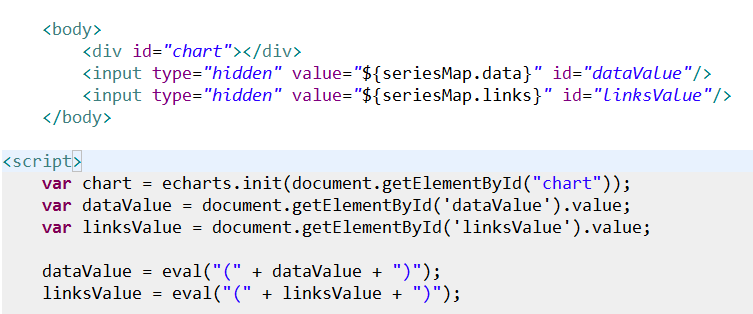
**解决办法：**

1、首先关闭MyEclipse工作空间。  
2、然后删除工作空间下的  
“/.metadata/.plugins/org.eclipse.core.runtime/.settings/com.genuitec.eclipse.ast.deploy.core.prefs”  
这个文件。在这个文件中存储了一些部署项目的信息。删了这个部署的项目的信息就没有错误了。  
3、重启启动MyEclipse即可解决了。

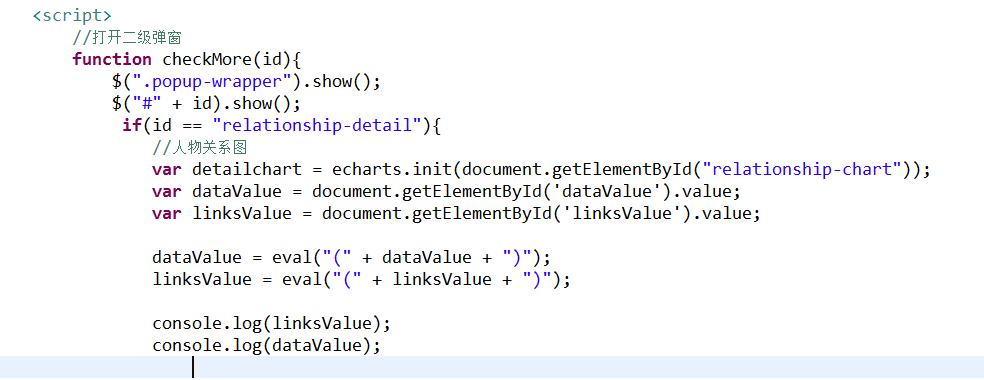
6.处理中文乱码



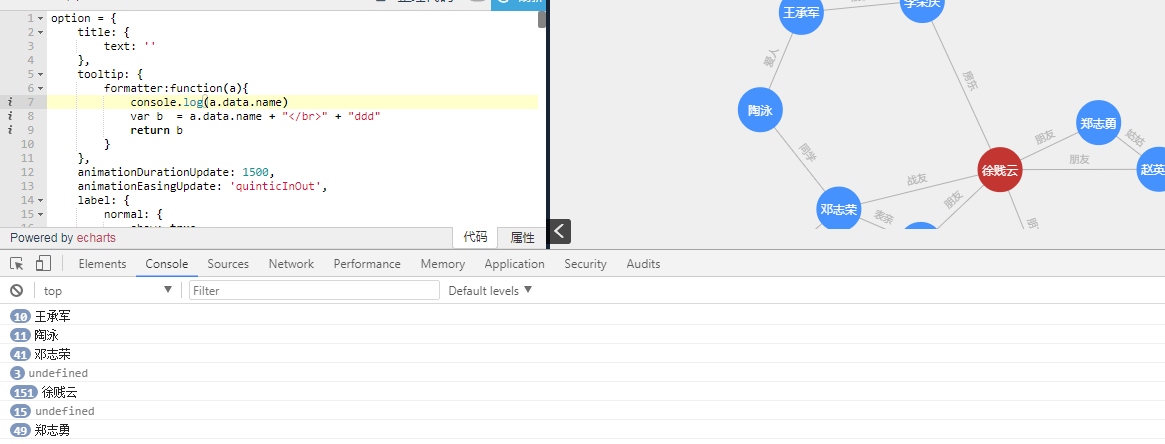
7.js中获取作用域里的值并转为json



8.前端向控制台打印数据



前端在控制台打印东西



9.MySQ相关函数

## trim

MySQL实现删除数据左右空格trim() 左空格ltrim() 右空格rtrim()

select trim(字段) from 表         删除左右空格

select ltrim(字段) from 表         删除左空格

select rtrim(字段) from 表         删除右空格

## CONCAT

CONCAT（）函数用于将多个字符串连接成一个字符串，是最重要的mysql函数之一，下面就将为您详细介绍mysql CONCAT（）函数，供您参考

mysql CONCAT(str1,str2,…)                          
返回结果为连接参数产生的字符串。如有任何一个参数为NULL ，则返回值为 NULL。或许有一个或多个参数。 如果所有参数均为非二进制字符串，则结果为非二进制字符串。 如果自变量中含有任一二进制字符串，则结果为一个二进制字符串。一个数字参数被转化为与之相等的二进制字符串格式；若要避免这种情况，可使用显式类型 cast, 例如： SELECT CONCAT(CAST(int\_col AS CHAR), char\_col)

mysql> SELECT CONCAT(’My’, ‘S’, ‘QL’);

-> ‘MySQL’

mysql> SELECT CONCAT(’My’, NULL, ‘QL’);

-> NULL

mysql> SELECT CONCAT(14.3);

-> ‘14.3′

mysql CONCAT\_WS(separator,str1,str2,…)   
CONCAT\_WS() 代表 CONCAT With Separator ，是CONCAT()的特殊形式。   第一个参数是其它参数的分隔符。分隔符的位置放在要连接的两个字符串之间。分隔符可以是一个字符串，也可以是其它参数。如果分隔符为 NULL，则结果为 NULL。函数会忽略任何分隔符参数后的 NULL 值。

mysql> SELECT CONCAT\_WS(’,',’First name’,'Second name’,'Last Name’);

-> ‘First name,Second name,Last Name’

mysql> SELECT CONCAT\_WS(’,',’First name’,NULL,’Last Name’);

-> ‘First name,Last Name’

mysql CONCAT\_WS()不会忽略任何空字符串。 (然而会忽略所有的 NULL）。

## group\_concat

完整的语法如下：

**group\_concat([DISTINCT] 要连接的字段 [Order BY ASC/DESC 排序字段] [Separator '分隔符'])**

## IFNULL

IFNULL函数的使用进行了详细的叙述，供您参考学习，如果您在MYSQL IFNULL函数使用方面遇到过类似的问题，不妨一看。

MYSQL IFNULL(expr1,expr2)            
如果expr1不是NULL，IFNULL()返回expr1，否则它返回expr2。IFNULL()返回一个数字或字符串值，取决于它被使用的上下文环境。            
mysql> select IFNULL(1,0);        
                     -> 1        
mysql> select IFNULL(0,10);        
                     -> 0        
mysql> select IFNULL(1/0,10);        
                     -> 10        
mysql> select IFNULL(1/0,yes);        
                     -> yes       
            
IF(expr1,expr2,expr3)            
如果expr1是TRUE(expr1<>0且expr1<>NULL)，那么IF()返回expr2，否则它返回expr3。IF()返回一个数字或字符串值，取决于它被使用的上下文。            
mysql> select IF(1>2,2,3);        
                     -> 3        
mysql> select IF(1<2,yes,no);        
                     -> yes       
mysql> select IF(strcmp(test,test1),yes,no);        
                     -> no       
       
expr1作为整数值被计算，它意味着如果你正在测试浮点或字符串值，你应该使用一个比较操作来做。            
       
mysql> select IF(0.1,1,0);        
                     -> 0        
mysql> select IF(0.1<>0,1,0);        
                     -> 1        
       
在上面的第一种情况中，IF(0.1)返回0，因为0.1被变换到整数值, 导致测试IF(0)。这可能不是你期望的。在第二种情况中，比较测试原来的浮点值看它是否是非零，比较的结果被用作一个整数。            
       
CASE value WHEN [compare-value] THEN result [WHEN [compare-value] THEN result ...] [ELSE result] END            
　            
CASE WHEN [condition] THEN result [WHEN [condition] THEN result ...] [ELSE result] END            
第一个版本返回result，其中value=compare-value。第二个版本中如果第一个条件为真，返回result。如果没有匹配的result值，那么结果在ELSE后的result被返回。如果没有ELSE部分，那么NULL被返回。            
mysql> SELECT CASE 1 WHEN 1 THEN "one" WHEN 2 THEN "two" ELSE "more" END;        
                 -> "one"       
mysql> SELECT CASE WHEN 1>0 THEN "true" ELSE "false" END;        
                 -> "true"       
mysql> SELECT CASE BINARY "B" when "a" then 1 when "b" then 2 END;        
-> NULL