excle导入

```
 public Map<String,Object> uploadSpaceUserExc(HttpServletRequest request) throws
Exception {
        Map<String,Object> map = new HashMap<>();
        Integer state = 1000;
        try {
            MultipartHttpServletRequest multipartRequest = (MultipartHttpServletRequest)
request;
            //获取excle文件
            MultipartFile file = multipartRequest.getFile("Filedata");
            if (file.isEmpty()) {
//              return error("上传文件不能为空");
            }

             /**
             *这里做优化，excle不用限制版本
             */
            Workbook workbook = null;
            String name=file.getOriginalFilename();//获取文件名称
            if(name.endsWith(".xlsx")){
                InputStream is = file.getInputStream();
                workbook = new XSSFWorkbook(is);//2007以上版本,扩展名是.xlsx
            }else if(name.endsWith(".xls")){
                InputStream is = file.getInputStream();
                workbook = new HSSFWorkbook(is);//2003以下版本,扩展名是.xls
            }
            Sheet sheet = workbook.getSheetAt(0);

            //获取最后一行行标
            int lastRow = sheet.getLastRowNum();
//          if (lastRow < 1)
//              return error("数据不能为空");
            List<String> datas = new ArrayList<String>();
            //从第一行遍历，如果不想遍历第一行就从第二行遍历，
            //改为for (int i = 1; i < lastRow + 1; i++) {
            for (int i = 0; i < lastRow + 1; i++) {
                Row row = sheet.getRow(i);
                row.getCell(0).setCellType(Cell.CELL_TYPE_STRING);
                Long userId = Long.parseLong(row.getCell(0).getStringCellValue());

                spaceUserExcMapper.insertIntoSpaceUserExc(userId);


            }
        }catch(Exception e){
            map.put("msg", "sql错误");
            state=2000;
        }
        map.put("state", state);
        return map;

    }
```

获取行和列

```
            Workbook workbook = null;
            String name=file.getOriginalFilename();//获取文件名称
            if(name.endsWith(".xlsx")){
                InputStream is = file.getInputStream();
                workbook = new XSSFWorkbook(is);//2007以上版本,扩展名是.xlsx
            }else if(name.endsWith(".xls")){
                InputStream is = file.getInputStream();
                workbook = new HSSFWorkbook(is);//2003以下版本,扩展名是.xls
            }

            //获得工作表
            Sheet sheet = workbook.getSheetAt(0);

            //获取最后一行行标(编号是从0开始)，比行数小1
            int lastRow = sheet.getLastRowNum();
            //如果遍历行的时候
            for (int i = 0; i <= lastRow; i++)


            //获取列数，比最后一列列标大1
            int lastCell = sheet.getRow(k).getLastCellNum();
            //如果遍历列的时候
            for (int i = 0; i < lastCell; i++)
```

这个在导入的时候遍历那块会用到，所以注意遍历时的变量长度

excle导出

```
    @RequestMapping(value = {"exportOutSpaceUserExc"})
    @ResponseBody
    public void exportOutSpaceUserExc(HttpServletResponse response, SpaceUserExc
spaceUserExc) throws IOException {
    Map<String,Object> map = new HashMap<>();
        List<Map<String,Object>> list = spaceUserExcMapper.findByPage();



        System.out.println("list是"+list.toString());
        String fileName = DateUtils.format(DateUtils.yyyyMMddHHmmssSSS) + ".csv";
        response.setContentType("application/octet-stream;charset=utf-8");
        response.setHeader("Pragma", "no-cache");
        response.setHeader("Cache-Control", "no-cache");
        response.setDateHeader("Expires", 0);
        response.setCharacterEncoding(DEFAULT_CHARSET);
        response.setHeader("Content-Disposition", "attachment; filename=" +
URLEncoder.encode(fileName, DEFAULT_CHARSET));
```

```
        String[] headerNames = {"用户VV号","用户昵称","活跃等级","歌手等级","财富等级","注册时间","认证类型","认证时间","是否实名认证","家族Id","家族名称","加入家族时间","家族负责人","自建歌房Id","自建歌房名称","家族歌房Id","家族歌房名称"};
        String[] columnNames = {"UserId","Nickname","level","levelSinger","levelWealth","registerTime","AuthType","AuthTime","liveAuthState","familyId","name","createtime","principal","userRoomId","userRoomName","familyRoomId","familyRoomName"};
        System.out.println("list是"+list);

        System.out.println("list是"+list.toString());
        logger.info(list.toString());
        System.out.println(columnNames.toString());
        ExcelUtils.writeEXCEL(list, headerNames, columnNames, response.getOutputStream());
        spaceUserExcMapper.delete();

    }
```

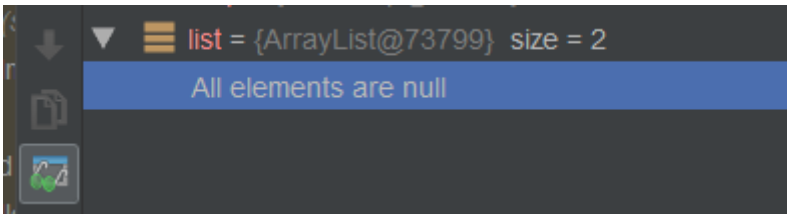
ExcelUtils.writeEXCEL(list, headerNames, columnNames, response.getOutputStream());这个底层代码

```
 public static void writeEXCEL(List<Map<String, Object>> datas, String[] headerNames, String[] columnNames, OutputStream os) throws IOException {
        HSSFWorkbook workbook = new HSSFWorkbook();
        HSSFSheet sheet = workbook.createSheet();
        HSSFRow row = sheet.createRow(0);

        int count;
        for(count = 0; count < headerNames.length; ++count) {
            HSSFCell cell = row.createCell(count);
            cell.setCellValue(headerNames[count]);
        }

        count = 0;
        if (datas != null && datas.size() > 0) {
            Iterator i = datas.iterator();

            while(i.hasNext()) {
                Map<String, Object> data = (Map)i.next();
                ++count;
                row = sheet.createRow(count);

                for(int i = 0; i < columnNames.length; ++i) {
                    HSSFCell cell = row.createCell(i);
                    String value = StringUtils.ifnull(data.get(columnNames[i]), "");
                    cell.setCellValue(value);
                }
            }
        }

        workbook.write(os);
        os.flush();
    }
```

导出数据如图：

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	用户VV号	用户昵称	活跃等级	歌手等级	财富等级	注册时间	认证类型	认证时间	是否实名认	家族Id	家族名称	加入家族时	家族负责人	自荐歌房Id	自荐歌房名	家族歌房Id	家族歌房名称	
2	20003	20003	8	43	37	2018-11-09	1	2019-01-09		12				538	12321的歌	548	对对对的房间	
3	20003	20003	8	43	37	2018-11-09	1	2019-01-09		15				538	12321的歌	548	对对对的房间	
4	20003	20003	8	43	37	2018-11-09	1	2019-01-09		137	对对对	2018-05-04		538	12321的歌	548	对对对的房间	
5	20002	20002	5	12	37	2015-01-29	3	2018-06-11		139	let it	2018-05-08	暖阳					
6	20002	20002	5	12	37	2015-01-29	3	2018-06-11		166	精钢葫芦娃	2018-08-06	2121					
7																		

导出数据逻辑这里如果要导出的顺序不能变，这里就要保证每一条数据都要查出来，如果查出来的为null，则会出现如下错误：



导出接口直接报错，为了防止以上bug,在查询的时候指定一个默认值，类似

```
select (case when su.UserId is null then '-1' else su.UserId end) as UserId from user
```

如果查出来为null，则赋值一个默认值'-1',就ok了！