



中国报表软件领导品牌

---

**FineReport(V10.0)开发规范书**



## 前言

决定编写报表开发规范书的缘由有两个，一是有部分客户找我们，问是否有相关的报表开发规范书，供他们开发人员参考。二是 finereport 技术支持在处理客户的日常问题时发现，有不少问题是由于客户前期的一些不规范操作而引起的，部分是小问题，部分是会影响全局的严重问题。为了让客户更好的使用帆软报表，我们觉得确实有必要整理一版开发规范书，并向客户推广。通过推广报表开发规范，可以将一部分场景问题的排查手段前置，减少客户在开发阶段就可以避免的部分问题。

本文主要针对设计器、报表命名、功能实现方法、报表 UI 设计、平台和服务器这六个模块结合开发规范及历史经验，一定会导致问题操作的尽可能全量覆盖，可能会导致问题或者是优化建议择重点输出。

各位读者现在看到的第一版，可能会存在场景覆盖不全，叙述的角度跟用户理解的角度有偏差或者不够贴合实际业务场景等等问题，欢迎大家向小编（帆软技术支持）提意见或者建议。

## 目录

1. 设计器	5
1.1 分离开发	5
1.2 远程设计环境开发注意事项	5
1.3 协同开发规范	5
1.4 备份策略	6
1.4.1 工程备份	6
1.4.2 备份修改	7
2. 报表命名规范	8
3. 功能实现方法规范	10
4. 报表 UI 设计规范	15
4.1 字体	15
4.2 格式	17
4.3 配色	18
5. 报表平台篇	19
5.1 内置/外置数据库的使用规范	19
5.1.1 内置数据库需要遵守的规范	19
5.1.2 内置数据库迁移到企业内部数据库的规范操作	20
5.1.3 内置数据库和外置数据库都需要遵守的规范	20
5.1.4 更改外置数据库账号和密码流程	20
5.1.5 给当前工程换一个外置数据库的操作流程	24
5.2 备份策略	24
6. 服务器篇	24
6.1 服务器配置	25

6.1.1 服务器硬件配置 .....	25
6.1.2 环境参数配置 .....	25
6.1.3 环境检测工具 .....	27
6.2 系统环境 .....	28
6.2.1 websocket 端口 .....	28
6.3 字体规范 .....	33
6.3.1 异常类别 .....	33
6.3.2 字体展示逻辑 .....	33
6.3.3 优化方案 .....	33
6.4 系统运维工具 .....	36
6.4.1 云端运维 .....	36
6.4.2 宕机处理工具 .....	36
6.4.3 平台日志导出插件 .....	37

## 1. 设计器

### 1.1 分离开发

每次在开发（dev）环境开发，修改，内部测试通过后，发布到测试（UAT）环境供业务人员测试，关键用户测试（UAT）测试通过后，再发布到生产环境，供最终用户使用。

### 1.2 远程设计环境开发注意事项

- （1）模板开发过程中，在保证 Jar 及插件版本与服务器一致的前提下，尽量远程设计；
- （2）远程设计连接的目标服务器需要授权远程设计功能点，否则远程设计无法使用；
- （3）设计器和服务器的 jar 包和插件版本需要保持一致；
- （4）远程设计前保证服务器的端口对外开放，开发人员能通过浏览器正常访问决策平台；内网通过 vpn 连接远程设计时，需要检查网络是否通、vpn 权限是否正常；
- （5）协同开发过程中，注意分配账号及权限，开启报表角色权限控制。远程设计的权限区别于平台的模板权限控制，需要管理员给开发人员分配相关的模板及数据连接远程设计权限，否则切换远程设计报错用户名密码错误；

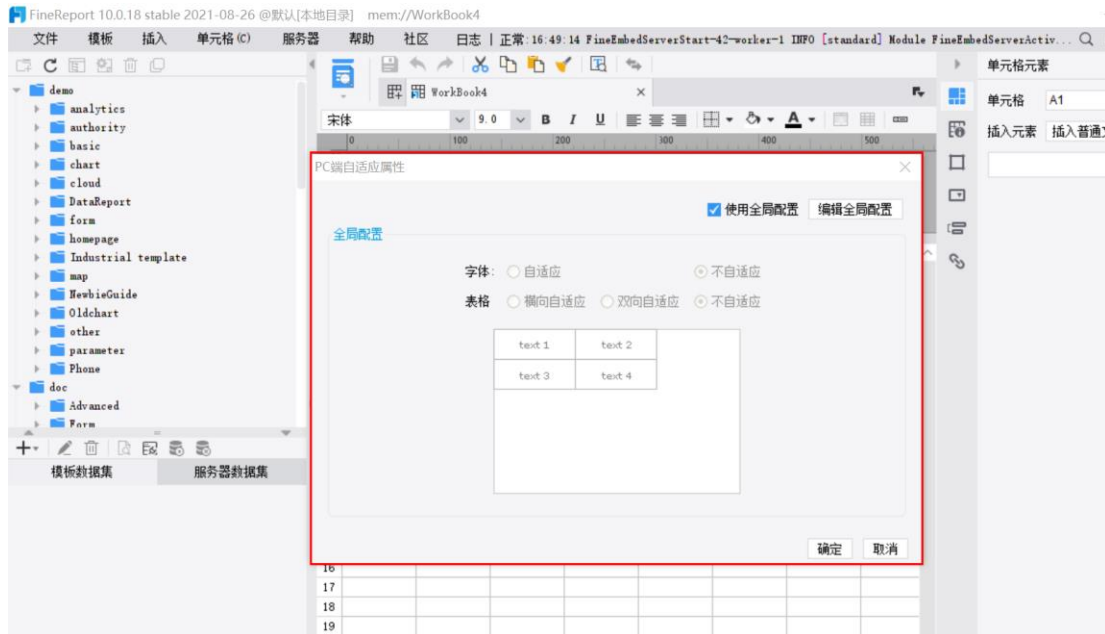
权限配置步骤参考文档：远程设计模板权限管理-  
<https://help.fanruan.com/finereport/doc-view-2845.html>

远程设计数据连接权限控制-<https://help.fanruan.com/finereport/doc-view-902.html>

- （6）启用 https 或修改 https 配置（证书路径、https 密钥），必须重启设计器才能生效。
- （7）远程设计不支持阿里云 pfx 格式的免费 ssl 证书以及 crt 证书，需要转成.keystore 或者 jks 证书。

### 1.3 协同开发规范

- （1）协同开发或模块开发过程中，需保证 UI 风格相对一致，配色可以使用设计器的预定义配色去控制；
- （2）大量决策报表的项目，必须考虑模板自适应配置，统一使用全局配置；



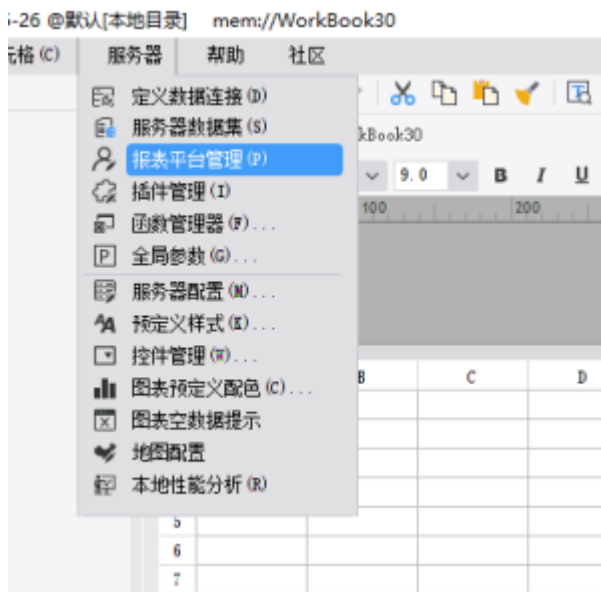
(3) 协同开发或模块开发过程中，报表中有多个组件、控件时，需保证组件命名的一致性、规范性。

## 1.4 备份策略

### 1.4.1 工程备份

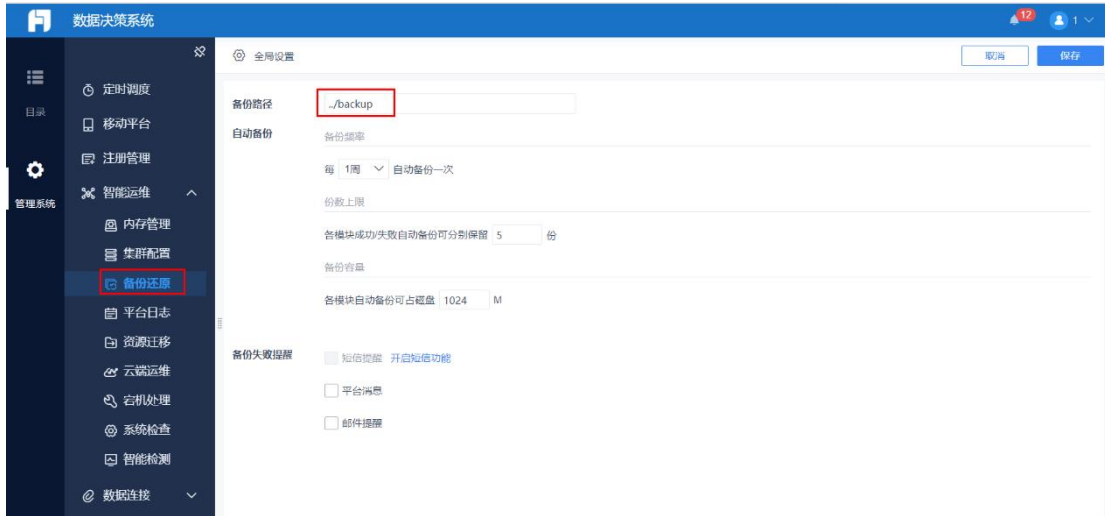
手动备份整个工程文件，或者在决策平台开启自动备份或者手动备份。

进入设计器自带的平台的方式：



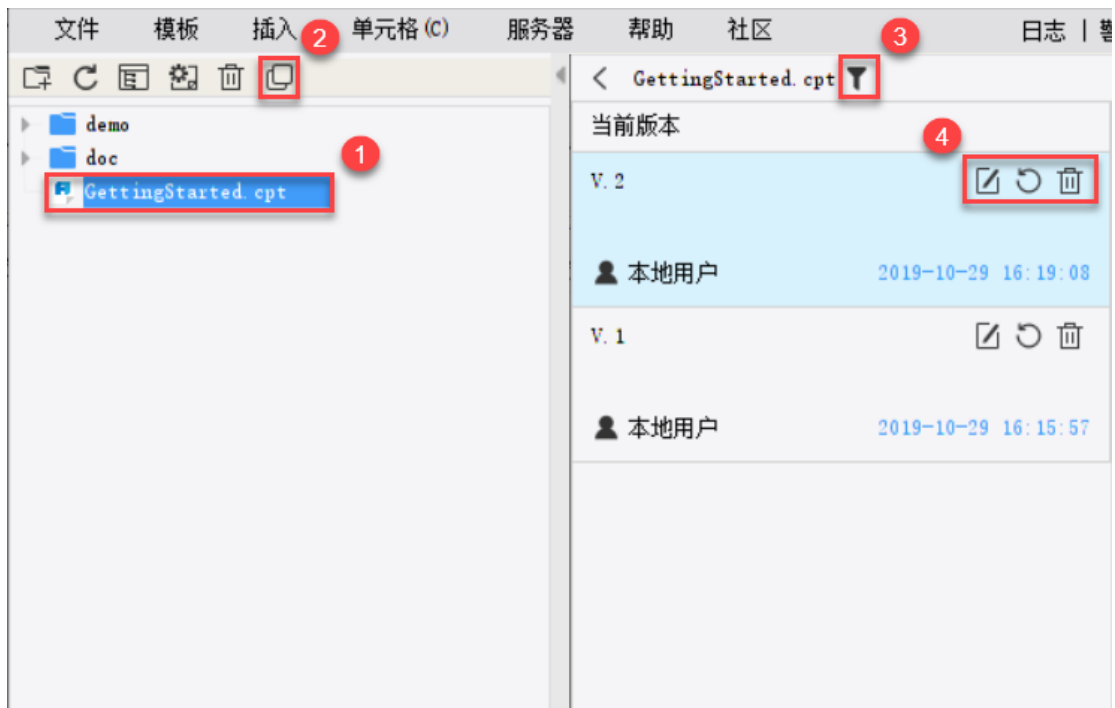
在开发的时候，在开发环境、测试环境开启自动备份，将备份的目录选在工程目录之外，然后通过 shell 脚本将备份文件移动到其他服务器或者其他目录，避免大量的备份文件导致应用过大，以及一台服务器出故障导致工程损坏。

### 备份目录选在工程之外的路径



### 1.4.2 备份修改

- (1) 优先在开发环境调整、验证与测试；
- (2) 如果没有开发环境，必须先模板备份(名称可以加上日期后缀，比如\*\*\*\_V20210627.cpt)，基于备份的模板做修改调整，不影响正式使用；
- (3) 模板调整修改需记录在案，方便追踪，可以使用设计器的模板版本管理功能。



## 2. 报表命名规范

表 2-1 报表命名规范说明表

项目	必须执行	建议执行
目录命名规则	1.不使用中文，名称整体由 26 个字母、数字、下划线组成，首位必须为字母	1.报表目录层级尽量不要超过四层
	2.目录名需做到见名知意，与业务模块相关。例如财务：FINANCE、人事：HR、营销：MARKET	2.建议独立创建测试目录 TEST，作为各实施人员或运维人员自己的测试空间，二级目录为 TEST+下划线+名字，例如：TEST_NAME
	3.下层目录建议用 ‘_’ 来分隔命名，例如营销的销售二级目录名：MARKET_SALE；营销的回款部分：MARKET_GETIN	3.测试模板一旦正式启用必须另存到正式目录中
模板命名规则	1.模块目录名+下划线+编号 例如：MARKET_SALE_001	/
	2.编号可以按照制作顺序递增，不可重复	/
	3.准备一张报表映射 excel，如附图 1-1，可以放在文件管理系统中，根据开发情况及时更新。	/
数据集命名规则	1.参数面板数据集：以 para_开头，例如：地区选择下拉框的数据集 para_area	/
	2.数据字典数据集：以 dic_开头，例如：产品映射字典数据集 dic_product	/
	3.报表主体数据集：以 report_开头，例如：客户销售数据 report_customerSaleData	/
	4.图表数据集：以 chart_开头，例如：产品占比图表 chart_productRatio	/



表 2-1 报表命名规范说明表（续）

项目	必须执行	建议执行
参数命名规则	1.全局参数：以 g 开头，例如： gSaleGroup, gProduct, gPerson	/
	2.模板参数：以 p 开头，例如： pDate, pCurrency, pUnit	/
	3.数据集参数：以 s 开头，例如： sCompany, sProject	/
条件属性命名规则	/	1.建议使用简短的中文命名，方便业务和开发
	/	2.格式：条件类型_具体功能，例如： 列宽_隐藏 projguid 列
超链接命名规则	/	1.建议使用简短的中文命名，方便业务和开发
	/	2.格式：超链接类型_具体功能，例如： 网络报表_跳转到明细
frm 组件命名规则	1.不允许使用复制组件自动填充的名称，比如“report0_c”、“report0_c_c”	1.每个组件名称中加上实际用途，比如“report0_sale_detail”、“chart0_sale_summary_column_chart”
	2.每个组件应按照组件类型+序号的形式，比如说“report0”、“report1”、“chart0”，方便后期维护调整	/
cpt 悬浮元素命名规则	/	2.每个悬浮元素名称中加上实际用途，比如 “Float0_sale_detail_line_chart”，方便后期维护调整

附图 1-1

模板英文名	中文名称	模板类型
MARKET_SALE_001	签约明细表	分析表
MARKET_SALE_002	销售汇总表	分析表
MARKET_SALE_003	项目首开信息查看表	分析表
MARKET_SALE_004	项目首开信息填报表	填报表

### 3. 功能实现方法规范

常见需求的实现方法规范，在满足功能的前提下，以性能、便于维护为目标。

表 3-1 功能实现方法规范说明表

项目	必须执行	建议执行
展示报表制作	1.图块多的报表（如驾驶舱、大屏）使用 frm 开发，一般来说使用绝对布局-适应区域，每个图块一个组件，无特殊要求不要使用组件叠加	1.减少跨组件获取单元格值使用
	2.减少前端过滤的使用，尽量在数据集中过滤	2.控件数据字典数据量大且重复数据多时，使用专用数据集
	3.报表不要使用隐藏行、列，如有需要使用条件属性隐藏	3.如非必要，不要在公式和 JS 中使用 SQL 函数
	4.公式中也要注意格式，如嵌套公式应做到每个公式换行，每个条件或结果换行	4.设计时尽量减少行列和格子数
	5.大屏模板制作，用外置图片，否则非常影响性能，可以使用外置图片插件比较方便	5.设计时尽量减少层次坐标、SQL 类函数在扩展行的使用
	6.数据量大的模板需要导出，使用大数据导出 excel 插件或者流式导出插件	6.设计时尽量减少公式调用图片
	7.预览数据量大的模板使用 sql 参数限制数据量，或者使用分页 sql、新引擎、行式引擎、新填报预览等功能	7.设计时尽量减少控件数量
	8.公式复杂计算慢的模板，使用定时调度推送或者抽数缓存插件	8.同一行不同格子设置过滤的时候，尽量只对第一个格子设置过滤
	/	9.所有格子尽量使用同一个父格、而不是依次继承
	/	10.控件数据字典尽量用数据集并开启缓存，尽可能不要直接用数据查询

表 3-1 功能实现方法规范说明表（续）

项目	必须执行	建议执行
填报报表制作	1.大数据量填报报表要尽量读写分离，查询报表和填报报表分离	<p>1.填报的数据库表设计（要有自己的主键，有主键在做数据的修改、流程流转时很有用，不用去用很多字段确定唯一的记录）</p> <p>设计思路：</p> <p>①要有自己的主键，有主键在做数据的修改、流程流转时很有用，不用去用很多字段确定唯一的记录</p> <p>②要有填报时间、填报用户。用于追溯数据的来源，时间等。</p>
	2.填报报表设置参数，新增导入时不展现数据，查询修改时尽可能添加过滤控件，减少查询结果集	<p>2.填报报表设计</p> <p>在页面的单元格中写参数 \$fine_username，公式 now()。分别记录填报的用户，页面打开的时间。</p> <p>如附图 3-1</p> <p>添加 E1、F1 两个单元格，分别写入公式 now()和\$fine_username，这两个单元格不扩展，并隐藏。</p> <p>如附图 3-2</p> <p>设置填报属性，这里要勾选上未修改不更新，注意使用该功能时填报属性中的值一定要是单元格，不能填公式，所以我们上一步特地新加了两个单元格</p>
	3.业务主键字段通常设置不可修改，如必须修改则应用数据库表中代理主键 UUID 字段做填报主键，新增数据时公式赋值新的 UUID	/
	4.没有导入需求时，尽可能在单元格做即时校验，有导入需求时则应将所有校验设置在提交校验中	/

表 3-1 功能实现方法规范说明表（续）

项目	必须执行	建议执行
参数开发	1.统一采用英文命名方式	1.涉及到不同 TAB 模块的同样指标的参数，参数名建议带上对应 TAB 模块的名称，以地区参数为例，例如： TAB1DIQU,TAB2DIQU
	2.采用 26 个英文字母和 0—9 这十个自然数，加上下划线_组成，共 36 个字符，不出现其他字符。采用英文单词或英文短语（包括缩写）作为名称，参照字典表给出的基础命名，没有的去翻译，不使用无意义的字符或汉语拼音。	2.对于多个模板都要使用的参数，可以设置成服务器参数，供多个模板使用
	3.大屏中设置参数要考虑到不同报表块参数加载速度的问题，避免一个大屏模块过多的不同参数	3.超级链接传递参数，设计模板时建议在子页面的单元格用=\$参数把参数值显示出来确认传递的值没有问题，设计完毕后删掉公式
	4.参数大小写一致，建议都用大写，不要用汉语命名	4.控件实际值统一用 id，控件显示值统一用名称，sql 过滤条件统一用 id 过滤
	5.多个数据集中的同一个参数默认值设置成相同，避免出现异常	/
	6.避免数据集参数和模板参数使用同一参数名	/
	7. 避免数据集参数和模板参数重名	/
	8.数据集 sql 中使用参数时，使用公式 if(len(参数)=0,"","sql 条件")规避参数为空的情况，避免特殊情况下出现报错或者打印、导出没有数据	/
	9.避免使用单元格的过滤功能，尽量用 sql 进行处理，单元格过滤会影响报表展示性能	/

表 3-1 功能实现方法规范说明表（续）

项目	必须执行	建议执行
条件属性	1.设置条件属性时候需要考虑到性能问题，一行上面一个条件属性能解决的问题不要用多个条件属性	1.条件属性根据实际需求命名，不要用默认的“条件属性 1” “条件属性 2”
	2.尽量合并多个条件属性为一个，通过多写条件判断实现，条件属性过多降低前段展示速度	2.要考虑性能问题，能不使用条件属性则不使用
	3.隐藏行列一定不要直接隐藏行列！通过条件属性设置行高 or 列宽=0 达到隐藏行列，方便维护	/
防宕机	1.必须迁移内置数据库 可参照：配置外接数据库- <a href="https://help.fanruan.com/finereport/doc-view-2458.html">https://help.fanruan.com/finereport/doc-view-2458.html</a>	1.限制数据集的结果集在任何参数情况下都不可以查出大量数据，最优 sql 结果<100 ,最大尽量不要超过 1w
	2.sql 查询不要太复杂，数据库全部扫描次数过多，给数据库带来太大压力，导致数据库执行缓慢，从而反馈到应用无响应	2.服务器尽量不要跑其他任何东西，纯净运行报表服务，如果运行有其他东西，尽量设置 XMS=XXM
	3.在测试环境做模板，确保不做出笛卡儿积模板预览导致系统宕机	3.尽量避免使用 toImage 函数，能用 webimage 就用 webimage
	4.windows 下使用 tomcat 等容器，正式运行环境请后台运行，不要使用带黑窗的 tomcat 等，后台运行尽可能使用最小日志输出（非调试情况下，在数据决策系统把日志级别改成 ERROR）和按时间分割日志文件	4.安装防宕机插件，限制 sql 行数，目前不建议使用自动排队回收功能
	5.所有工程的连接池数量总和不可以超过数据库允许的值	5.使用数据字典的时候，尽量使用专用数据集(就两个字段最好)，而不是拿完整的 select*的数据集

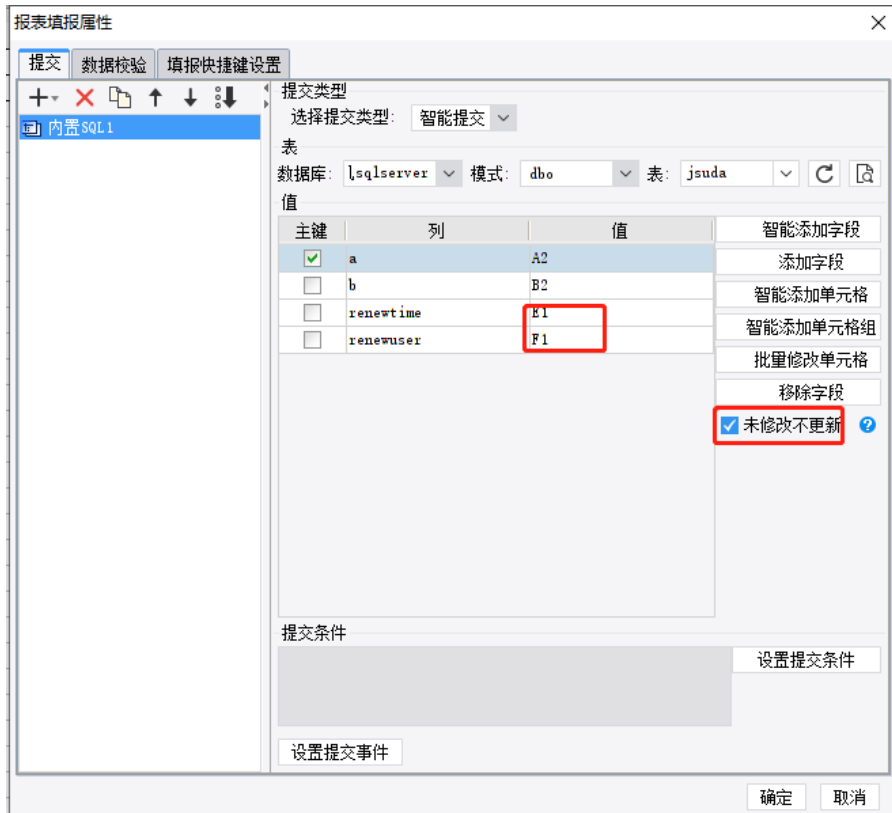
表 3-1 功能实现方法规范说明表（续）

项目	必须执行	建议执行
防宕机	6.配置 web 容器的最大线程数量略低于模板业务数据连接池的数量	6.配置一排同一个父格的时候尽量设置成所有格子使用一个父格而不是依次继承
	7.尽可能避开填报笛卡儿积问题，报表笛卡尔积的场景实例	/
	8.linux 上要用 nohup 启动工程 防止被 shell 注销干扰，后台运行尽可能使用最小日志输出（非调试情况下，在数据决策系统把日志级别改成 ERROR）和按时间分割日志文件	/
	9.jvm 内存不要超过 32g，尽量不要超过系统物理内存的 1/2，内存大于 10g 的时候尽量使用 G1 回收器	/
	10.在 linux 上修改应用最大打开文件数（默认 1024，至少 i 调整到 6w+）在遇到客户使用附件，或者缓存到磁盘的功能的时候会产生影响	/
	11.表单的背景不要重复设置，尽量不要用大的图片做背景，实在需要可以用 css+web 图片处理表单背景	/

附图 3-1

	A	B	C	D	E	F	G
1	a	b	更新时间	更新人	=now()	=\${fine_user}	
2	ds1.G(a)	ds1.G(b)	ds1.G(renewtime)	ds1.G(renew			
3							
4							
5							
6							
7							
8							
9							

附图 3-2



## 4. 报表 UI 设计规范

在 FR 报表的开发过程中，对于报表的 UI，需要注意字体、格式、配色这几方面。具体实施，不同用户会有不同需求，若您有专门的 UI 设计规范，则以您的设计规范为准，若没有则在报表的开发过程中，建议遵循本 UI 设计规范。

UI 设计规范方面，不同的人有不同的看法，核心是，风格一致，例如，标题都是黑体，正文都是宋体等；配色风格，如都是暗色系，则不应突然冒出个亮色系。在此基础上，字体、格式、配色，可因地制宜，根据实际情况处理。举例：财务的报表，要求字段都左对齐，那应当遵循其既有规则。

### UI 设计原则：

- (1) 风格一致，整齐清爽；
- (2) 字体方面，有些商用收费的字体应该避免使用，如微软雅黑等。

### 4.1 字体

中文字体里，建议使用宋体、仿宋、黑体、楷体、隶书、幼圆。另外，思源黑体、思源宋体、庞门正道标题体、文泉驿系列、站酷系列，以及方正的楷体、黑体、仿宋、书宋，都可以免费商用。

无论是明细表的标题还是图表的标题，建议加粗，且字号大于正文。

**明细表：**

**2018年公司利润明细表**

项目	第一季度	第二季度	第三季度	第四季度	合计
XIANGMU1	3, 399	1, 995	9, 049	6, 734	21, 177
XIANGMU2	7, 471	9, 113	1, 232	1, 957	19, 773
XIANGMU3	6, 320	2, 118	8, 537	1, 634	18, 609
XIANGMU4	8, 249	1, 311	1, 680	2, 387	13, 627
XIANGMU5	6, 742	1, 463	2, 847	7, 326	18, 378
合计	32, 181	16, 000	23, 345	20, 038	91, 564

使用字体：宋体

**2018年公司利润明细表**

项目	第一季度	第二季度	第三季度	第四季度	合计
XIANGMU1	3, 399	1, 995	9, 049	6, 734	21, 177
XIANGMU2	7, 471	9, 113	1, 232	1, 957	19, 773
XIANGMU3	6, 320	2, 118	8, 537	1, 634	18, 609
XIANGMU4	8, 249	1, 311	1, 680	2, 387	13, 627
XIANGMU5	6, 742	1, 463	2, 847	7, 326	18, 378
合计	32, 181	16, 000	23, 345	20, 038	91, 564

使用字体：黑体

**2018年公司利润明细表**

项目	第一季度	第二季度	第三季度	第四季度	合计
XIANGMU1	3, 399	1, 995	9, 049	6, 734	21, 177
XIANGMU2	7, 471	9, 113	1, 232	1, 957	19, 773
XIANGMU3	6, 320	2, 118	8, 537	1, 634	18, 609
XIANGMU4	8, 249	1, 311	1, 680	2, 387	13, 627
XIANGMU5	6, 742	1, 463	2, 847	7, 326	18, 378
合计	32, 181	16, 000	23, 345	20, 038	91, 564

推荐字体：思源黑体  
免费可商用字体

[注]上图所示表格样式，适用于数据量较少，行数≤5的明细表。

**2018年公司利润明细表**

项目	第一季度	第二季度	第三季度	第四季度	合计
XIANGMU1	3, 399	1, 995	9, 049	6, 734	21, 177
XIANGMU2	7, 471	9, 113	1, 232	1, 957	19, 773
XIANGMU3	6, 320	2, 118	8, 537	1, 634	18, 609
XIANGMU4	8, 249	1, 311	1, 680	2, 387	13, 627
XIANGMU5	6, 742	1, 463	2, 847	7, 326	18, 378
合计	32, 181	16, 000	23, 345	20, 038	91, 564

常规图表样式  
运用间隔色区分行

不同报表的标题，字体、字号应一致。若出现表头，或是二级标题，字体应小于主标题，大于正文，建议加粗；正文，字体、字号应一致。



项目	第一季度	第二季度	第三季度	第四季度	合计
XIANGMU1	3, 399	1, 995	9, 049	6, 734	21, 177
XIANGMU2	7, 471	9, 113	1, 232	1, 957	19, 773
XIANGMU3	6, 320	2, 118	8, 537	1, 634	18, 609
XIANGMU4	8, 249	1, 311	1, 680	2, 387	13, 627
XIANGMU5	6, 742	1, 463	2, 847	7, 326	18, 378
合计	32, 181	16, 000	23, 345	20, 038	91, 564



错误范例

项目	第一季度	第二季度	第三季度	第四季度	合计
XIANGMU1	3, 399	1, 995	9, 049	6, 734	21, 177
XIANGMU2	7, 471	9, 113	1, 232	1, 957	19, 773
XIANGMU3	6, 320	2, 118	8, 537	1, 634	18, 609
XIANGMU4	8, 249	1, 311	1, 680	2, 387	13, 627
XIANGMU5	6, 742	1, 463	2, 847	7, 326	18, 378
合计	32, 181	16, 000	23, 345	20, 038	91, 564



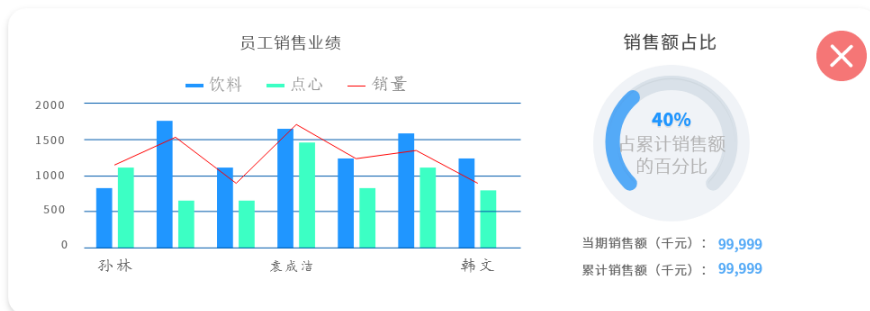
正确范例

标题字号需大于正文  
字体建议加粗

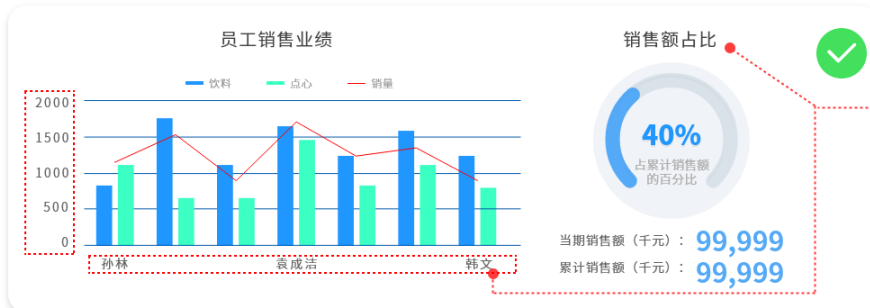
正文的字体、字号应  
保持一致

### 图表：

图表中涉及文字的地方有标题、标签、轴标签、提示。这四类，同类应保持字体、字号一致。标题字号一般大于其他三类。建议这四类，字体一致，后三类字号一致。



错误范例



正确范例

同类型的标题、标签  
应保持字体、字号一致

## 4.2 格式

所有标题居中；数字的小数位一致。

### 明细表：

明细文字，格式统一，都居左或居中；

明细表第一行、第一列空出来，留出点余白。行高最好一致，自动换行不建议开，字数不一致的情况下，会导致行高不一，显得很乱；

网格线方面，保持风格一致即可，即线型、颜色一致；

此外，明细表的行列不许直接隐藏，用条件属性实现。

### 图表：

标题居中。坐标轴标题，若有，如 Y 轴标题，则位置应保持一致，都居上或居中；

标签的位置，至少同一类图表，如柱形图，应保持一致，都在外侧或是内侧。

## 4.3 配色

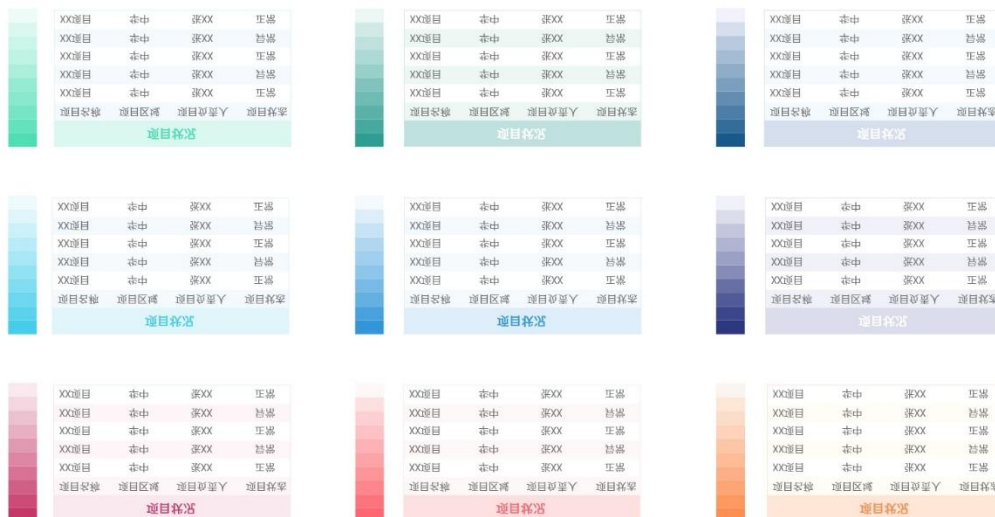
### 明细表：

字体颜色一般都是黑色，特殊情况，如数值预警高亮，因地制宜；

表头加背景色，不建议大红大紫那些过于鲜艳的颜色，建议蓝色或灰色等。表头字色根据背景色调整，建议白色、灰色等；

明细表还建议设置奇偶行的间隔色，以达到好区分不同行的目的。颜色同样不可过于鲜艳，建议灰色或淡蓝色等。

### 配色参考



## 图表：

根据背景颜色来设置标题等字体颜色，如图表都是暗色系，那文字的颜色建议白色这样的亮色系：

不同系列色，应遵循美观、舒适的原则，颜色不可太亮太刺眼，要让人看得舒服；

大屏的背景色建议暗色系，因为白色在大屏上的展示效果不好。

## 5. 报表平台篇

### 5.1 内置/外置数据库的使用规范

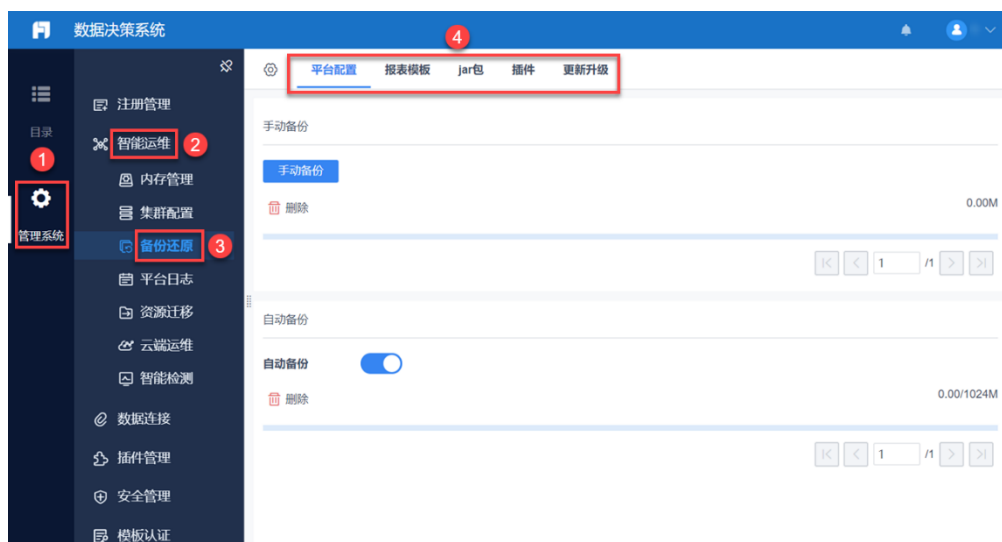
平台数据库（默认名称 finedb，下文使用 finedb 代指）十分重要，保存了包括用户、权限、数据决策系统界面的所有配置、设计器上的全局配置等信息，由于某些不规范操作，会导致数据丢失、错乱、产生脏数据等情况，影响数据决策系统的正常使用，因此规范使用及规范管理 finedb 是十分重要的。

#### 5.1.1 内置数据库需要遵守的规范

（1）finedb 存储在%FR\_HOME%\webapps\webroot\WEB-INF\embed 文件夹下，启动 FR 服务器的用户，要有这个文件夹的读写权限。

（2）不得删除%FR\_HOME%\webapps\webroot\WEB-INF\embed 文件夹下任何东西，若不小心点了多次启动 FR，会导致 finedb 被多个进程占用了，此时会提示产生了 db.lck 文件，需要删除，可以把这个文件从 finedb 文件夹移出来就行。

（3）定期备份数据库，10.0.11 版本前的工程需要管理员手动备份 finedb，10.0.11 版本后的工程可以使用平台提供的备份还原功能。



### 5.1.2 内置数据库迁移到企业内部数据库的规范操作

- (1) 手动备份一次内置数据库。
- (2) 在要迁移的数据库新建一个空的数据库，不得把内置数据库迁移到非空数据库。
- (3) 数据库服务器和报表服务器网络需要互通，ip 和端口都通畅，需要一直保持通畅，若网络不通或者数据库服务停了，会影响报表系统使用。
- (4) 在数据决策平台配置外置数据库，连接的数据库用户需要有这个库包括但不限于增删查改等全部的权限。
- (5) 把内置库迁移到外接数据库之后，让数据库管理员设置自动备份，建议每周在数据库服务器上对该库进行备份一次。
- (6) 同时需要在数据决策系统-管理系统-智能运维-备份还原-平台配置页面，设置自动备份，建议每周备份一次，以便应对数据库服务器的突发状况。

### 5.1.3 内置数据库和外置数据库都需要遵守的规范

- (1) 因为 finedb 数据库表与表之间有复杂的级联关系，若直接删除某个表字段，或者修改某个表字段等操作，会导致级联关系混乱。

因此非特殊情况，不能连接 finedb 库直接修改表字段，而是登录数据决策系统，或者使用设计器连接 FR 服务器工程，在平台界面或者设计器界面修改相关配置。

如果需要修改字段值，请安装 FINE\_CONF\_ENTITY 可视化配置-

<https://help.fanruan.com/finereport/doc-view-2471.html> 插件，在决策系统上修改字段值。

- (2) 需要在数据决策系统-管理系统-智能运维-备份还原-平台配置页面，设置自动备份，至少每周备份一次。
- (3) 未配置集群的情况下，finedb 库只能被一个 FR 工程连接，不允许被多个工程连接。

### 5.1.4 更改外置数据库账号和密码流程

由于安全要求，部分企业的数据库会定期更改密码或者账号。系统配置了外接数据库后，工程每次启动时都需要连接外接数据库。如果直接修改外接数据库的账号密码，工程则无法成功连接外接库，导致启动失败。

#### (1) 修改方法

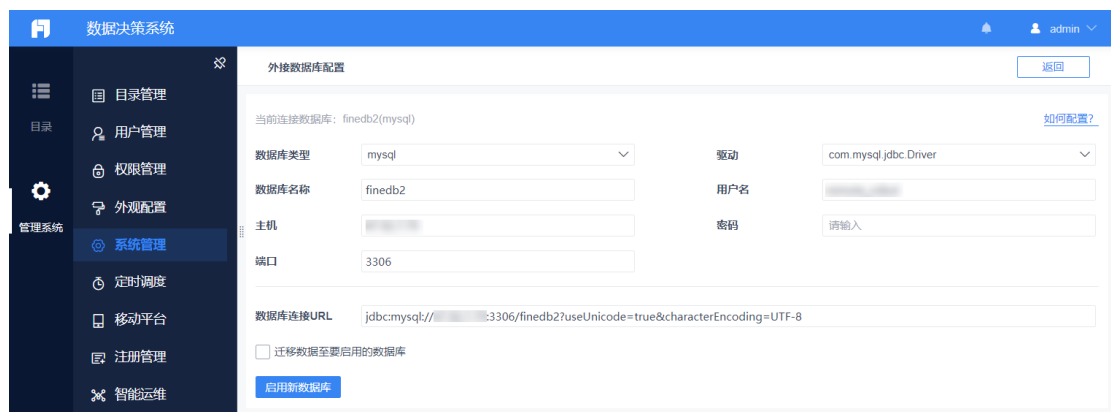
本文提供两种正确修改外接数据库账号密码的方法。

方法一：

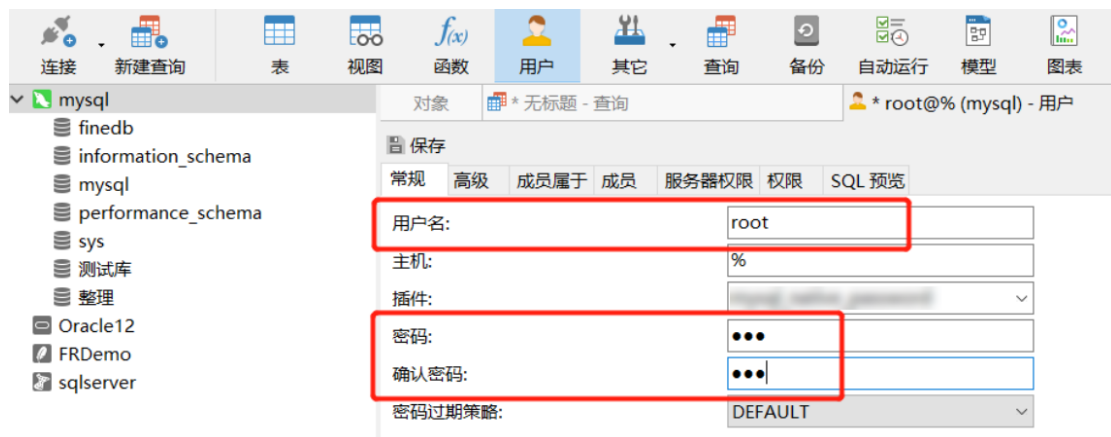
步骤一：启动工程，管理员登录数据决策系统，点击「管理系统>系统管理>常规>外接数据库>已配置」，如下图所示：



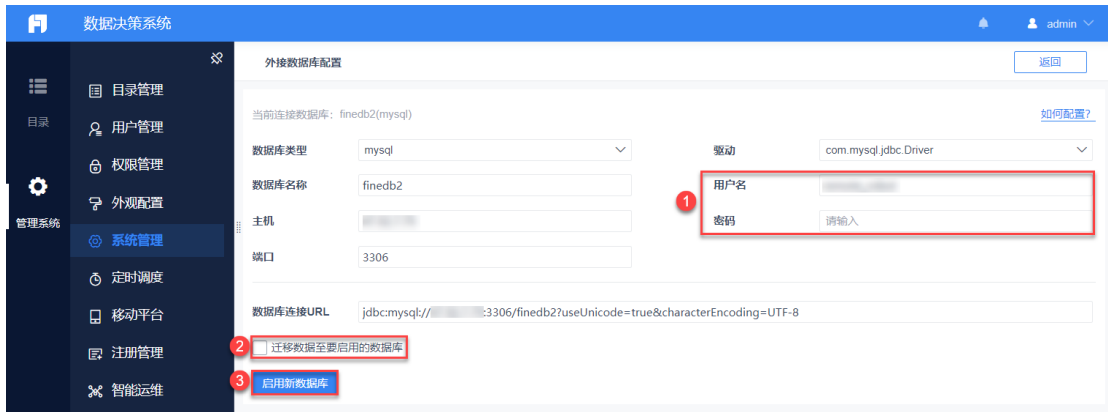
步骤二：确保打开了「外接数据库配置」页面。



步骤三：此时可以去修改数据库的用户名密码。



步骤四：在「外接数据库配置」页面，填入新用户名和密码，切勿勾选「迁移数据至要启用的数据库」，点击「启用新数据库」，启用成功即可。



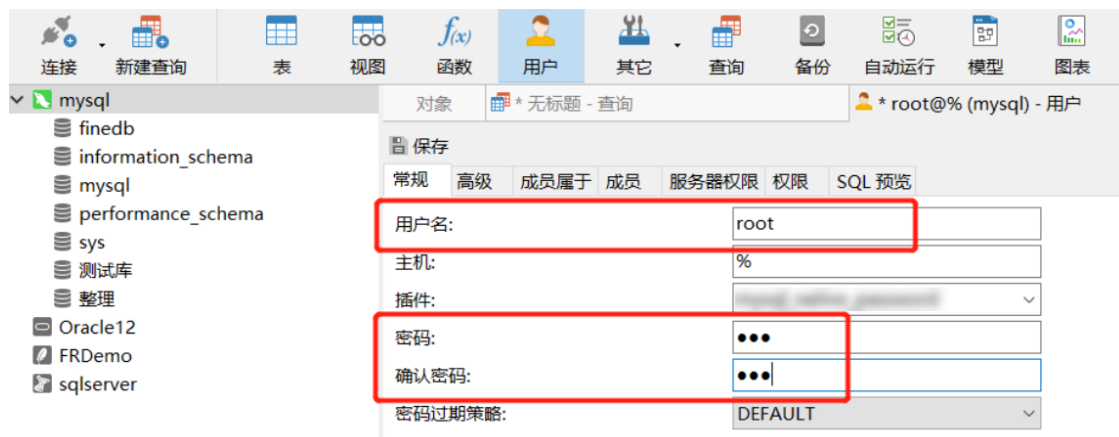
**方法二：**

**步骤一：** 准备一个临时数据库，数据库内容为空，与需要修改账号密码的外接数据库无关联即可。

**步骤二：** 按照文档，将外接数据库迁移至新的数据库。迁移时无需勾选「迁移数据至要启用的数据库」。

数据库类型	迁移方法
MySQL5 RDS MySQL	<a href="#">配置 MySQL5 外接数据库</a>
MySQL8	<a href="#">配置 MySQL8 外接数据库</a>
SQL Server	<a href="#">配置 SQL Server 外接数据库</a>
Oracle	<a href="#">配置 Oracle 外接数据库</a>
DB2	<a href="#">配置 DB2 外接数据库</a>

**步骤三：** 此时可以去修改原外接数据库的用户名密码。



**步骤四：**改完成后再次按照步骤 2，将外接数据库迁移回原数据库。迁移时无需勾选「迁移数据至要启用的数据库」。

## （2）未按规范操作流程导致外置数据库连接不上的解决方案

若未按照上述的（1）修改方法里提供的两种方法进行操作，外接数据库的账号密码已被修改。

工程启动时失败，自动跳转到服务器部署向导页面，提示「数据库连接异常，外接数据库连接失败」，如下图所示：

本章将简单介绍出现该情况时的解决方法。



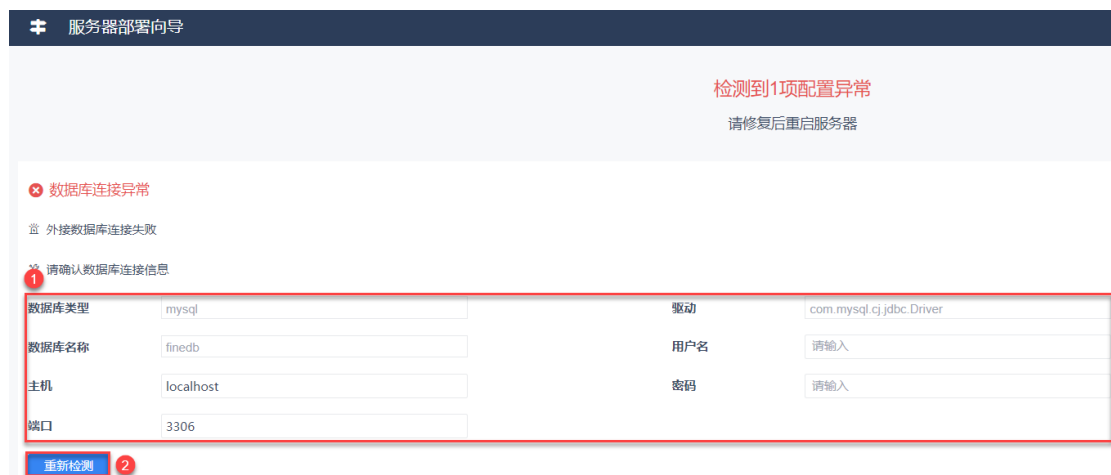
### 方法一：回退账号密码

**步骤一：**联系数据库管理员，将外接数据库的账号密码回退。

**步骤二：**工程正常启动后，按照第二章的方法修改外界数据库的账号密码。

### 方法二：服务器部署向导

在服务器部署向导页面，重新配置外接数据库信息。



## 5.1.5 给当前工程换一个外置数据库的操作流程

当前工程 A 已经连接了一个外置数据库了，但是想启用另一个工程 B 的外置数据库，以达到两个环境的用户、权限等都一致的情况，应按以下流程操作：

**步骤一：** 备份当前工程 A 所用的数据库。

**步骤二：** 复制一份工程 B 所用的数据库。

**步骤三：** 删除工程 A 的 %FR\_HOME%\webapps\webroot\WEB-INF\config\db.properties 文件夹，重启工程 A 所用的 web 容器（如 web 容器使用的是 tomcat，那么需要重启 tomcat），超管登录数据决策系统，进入管理系统-系统管理-常规-外置数据库-待配置页面，配置好数据库信息，连接到工程 B 所用的数据库的副本上，点击启用新数据库，会弹出提示：

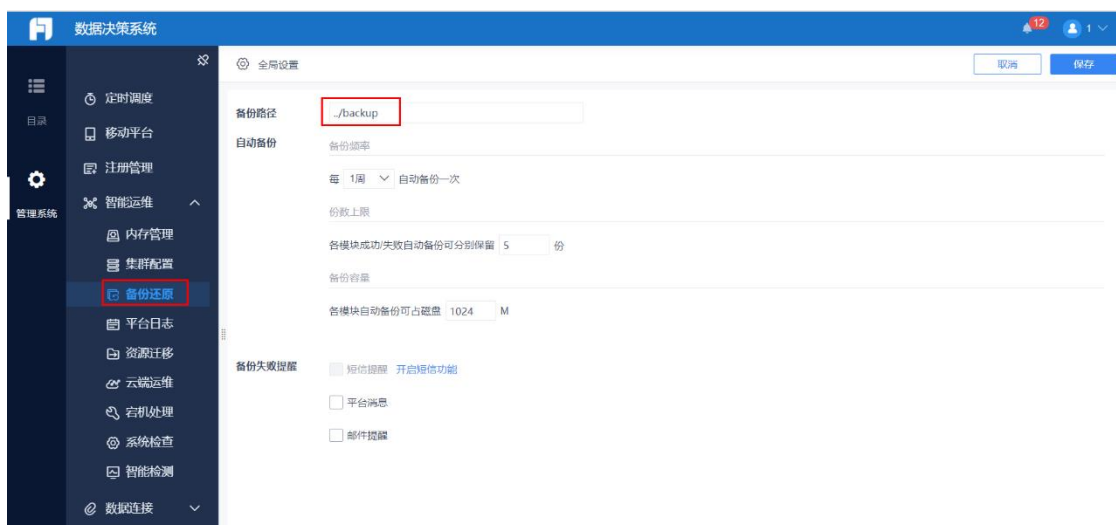
该数据库已存在平台数据，导入数据前将清空原有平台数据，确认连接该数据库？点确认就行。

## 5.2 备份策略

手动备份整个工程文件，或者在决策平台开启自动备份或者手动备份。

以 linux 环境为例：在开发的时候，在开发环境、UAT 环境开启自动备份，将备份的目录选在工程目录之外，然后通过 shell 脚本将备份文件 move 到其他服务器或者其他目录，避免大量的备份文件导致应用过大，以及一台服务器出故障导致工程损坏。

**备份目录选在工程之外的路径。**



## 6. 服务器篇



## 6.1 服务器配置

### 6.1.1 服务器硬件配置

主要针对操作系统、数据库、服务器、浏览器、单机或者集群环境下系统配置等场景给出推荐方案

硬件配置-<https://help.fanruan.com/finereport/doc-view-1164.html>

浏览器配置-<https://help.fanruan.com/finereport/doc-view-1940.html>

### 6.1.2 环境参数配置

主要针对 win 或者 linux 环境下 GC 日志、dump 日志、堆栈日志等配置给出说明

表 6-1 日志配置说明表

项目	Windows	Linux	适用场景
gc	<p><b>【解压版本】</b> 在 Catalina.bat 文件中设置 set CATALINA_OPTS=- Xloggc:gc.%%p.log - XX:+PrintGCDateStamps - XX:+PrintGCTimeStamps - XX:+PrintGCDetails 如附图 6-1 生成的日志在 bin 目录下，也可以设置自定义路径 -Xloggc:E:\tomcat8.5\gc.%%p.log</p> <p><b>【安装版本】</b> 在 java options 添加 -Xloggc:gc.%%p.log -XX:+PrintGCDateStamps -XX:+PrintGCTimeStamps -XX:+PrintGCDetails</p>	<p>找到 tomcat/bin 目录下的 Catalina.sh，在文件头加上这一段，-Xloggc 路径自定义</p> <p>CATALINA_OPTS='- XX:+PrintGCDateStamps - XX:+PrintGCTimeStamps - XX:+PrintGCDetails - Xloggc:/usr/local/java/tomcat1 /gc.%%p.log'</p>	gc 日志可以准确的反映系统一段时间内内存使用情况。

表 6-1 日志配置说明表（续）

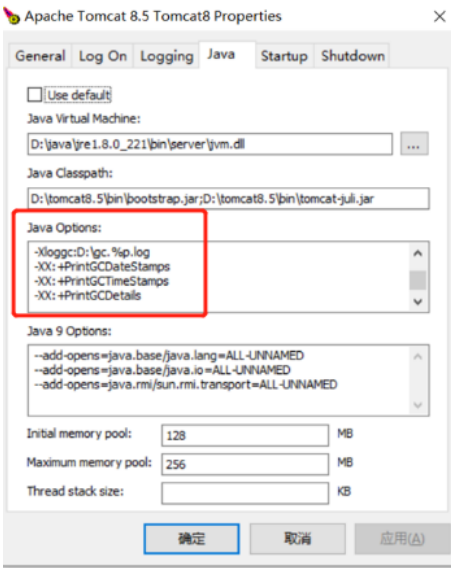
项目	Windows	Linux	适用场景
gc		/	/
dump	<p>注意点：需要配置好 jdk 环境变量或者切换到 jdk 的 bin 目录下运行</p> <p><b>【手动生成】</b></p> <p>需要 cd 到该 tomcat 使用的 jdk 的 bin 目录下执行，其中 pid 是对应 java 进程号,若为制定路径默认导出的文件在 jdk 的 bin 目录下</p> <pre>jmap - dump:format=b,file="D:\a.dump" pid</pre> <p><b>【自动生成】</b></p> <p>(1) 参考文档将服务器 jar 版本更新到制定版本自动生成 dump</p> <p>宕 机 处 理 -</p> <p><a href="https://help.fanruan.com/finereport/doc-view-3921.html">https://help.fanruan.com/finereport/doc-view-3921.html</a></p> <p>(2) win 压缩包在 catalina.bat 中添加参数</p> <pre>set JAVA_OPTS= -Xms1024m - Xmx1024m - XX:+HeapDumpOnOutOfMemoryError -</pre>	<p>注意点：需要配置好 jdk 环境变量或者切换到 jdk 的 bin 目录下运行</p> <p><b>【手动生成】</b></p> <p>需要 cd 到该 tomcat 使用的 jdk 的 bin 目录下执行，其中 pid 是对应 java 进程号,若为制定路径默认导出的文件在 jdk 的 bin 目录下</p> <pre>jmap - dump:format=b,file=/home/he apdump/a.dump pid</pre> <p><b>【自动生成】</b></p> <p>linux 中在 Catalina.sh 的参数中加入以下两句参数即可</p> <pre>- XX:+HeapDumpOnOutOfMemoryError ( OutOfMemory 的时候打 dump ) - XX:HeapDumpPath=/home/he apdump (对应路径)</pre>	内存溢出时用于分析。

表 6-1 日志配置说明表（续）

项目	Windows	Linux	适用场景
dump	<p>XX:HeapDumpPath=D:\heapdump</p> <p>（3）win 安装版在在 java options 添加，参数位置必须按照顺序来，不然服务起不来</p> <p>-</p> <p>XX:+HeapDumpOnOutOfMemoryError</p> <p>or</p> <p>-XX:HeapDumpPath=D:\heapdump</p> <p>-Xloggc:gc.%p.log</p> <p>-XX:+PrintGCDateStamps</p> <p>-XX:+PrintGCTimeStamps</p> <p>-XX:+PrintGCDetails</p>	/	/
堆栈	<p>操作方式：在发生宕机或者觉得服务器卡顿的时候，输入如下指令，一般每隔 3~5 秒打一个，打三至四个：</p> <p>注：需要配置好 jdk 环境变量或者切换到 jdk 的 bin 目录下运行</p> <p>jstack pid &gt;&gt;1.txt</p> <p>jstack pid &gt;&gt;2.txt</p> <p>jstack pid &gt;&gt;3.txt</p>	<p>操作方式：在发生宕机或者觉得服务器卡顿的时候，输入如下指令，一般每隔 3~5 秒打一个，打三至四个：</p> <p>注：需要配置好 jdk 环境变量或者切换到 jdk 的 bin 目录下运行</p> <p>jstack pid &gt;&gt;1.txt</p> <p>jstack pid &gt;&gt;2.txt</p> <p>jstack pid &gt;&gt;3.txt</p>	宕机时可以辅佐 dump 或其他日志进行深度判断。

附图 6-1

```

setlocal
rem Suppress Terminate batch job on CTRL+C
set JAVA_OPTS=-Xms2048m -Xmx2048m -Dfile.encoding=UTF-8
set CATALINA_OPTS=-Xloggc:gc.%p.log -XX:+PrintGCDateStamps -XX:+PrintGCTimeStamps -XX:+PrintGCDetails
if not "%1" == "run" goto mainEntry
if "%TEMP%" == "" goto mainEntry
if exist "%TEMP%\%-na0.run" goto mainEntry
echo Y>%TEMP%\%-na0.run
    
```

### 6.1.3 环境检测工具

表 6-2 环境检测工具说明表

场景	文档
模板限制推荐插件	<p>智能运维&gt;内存管理&gt;模板限制」中，管理员不清楚模板限制相关数值具体设置多少，需要一个推荐值。开启了模板限制后，管理员希望知道哪些模板触发了限制。有一些被限制了造成不能访问的模板，业务人员又必须需要，管理员希望知道如何优化这些模板。参考文档：模板限制增强插件</p> <p>-<a href="https://help.fanruan.com/finereport/doc-view-3630.html">https://help.fanruan.com/finereport/doc-view-3630.html</a></p> <p>注：未注册购买「智能运维」功能点时，「模板限制增强」插件安装后不可用。</p>
设计器内存监控插件	<p>此插件主要是为了实时监控与手动清理设计器内存。避免碰到内存占据过多，突然卡死的现象，以及生成 dump.hprof 文件用于内存分析。</p> <p>参考文档：设计器内存监控插件 -<a href="https://help.fanruan.com/finereport/doc-view-3037.html">https://help.fanruan.com/finereport/doc-view-3037.html</a></p> <p>注：安装插件后，双击内存悬浮球时，会弹出启用内存回收设计器将会卡死，请耐心等待提示框，点击是（Y）后，才会开始生成 dump.zip 文件用于内存分析</p>

## 6.2 系统环境

### 6.2.1 websocket 端口

#### （1）WebSocket 的作用

WebSocket 可以让服务器端主动向客户端推送数据。

在 WebSocket API 中，客户端和服务端只需要完成一次握手，两者之间就直接可以创建持久性的连接，并进行双向数据传输。

WebSocket 主要用于刷新 token、用户被踢出、平台消息、内存和 CPU 显示、平台日志处当前系统在线人数、数据连接编辑状态的确定。

#### （2）socket 异常相关场景

- a.智能运维—内存管理—实时内存，两张图显示空白（正常工程刚启动需要等待一两分钟才会显示）
- b.智能运维—平台日志—访问统计—当前系统在线人数，无法正确显示，显示 0
- c.数据连接编辑状态的确定，可能存在多人同时编辑同一个数据连接的情况

- d.用户收不到右下角消息弹窗且小铃铛处无消息提示，但是可以点击小铃铛进入消息面板查看，消息列表有新消息
- e. 踢出用户失效，不会实时踢出。在框架内显示登录页面，并没有跳出框架直接跳转到登录界面
- f.其它场景：单一登录踢出、禁用用户、修改用户密码、修改认证方式、切换同步导入、平台使用用户禁用用户/打开限制开关，更换 lic，切换同步数据集等

**(3) socket 默认信息：**

端口	JAR 包	ID	默认值	设置范围	是否支持设置多个值
Websocket 端口	-	WebSocketConfig.port	["38888", "39888"]	参数值为端口数组 ["port1","port2"] port均属于区间(1024,65535]	支持
Websocket 转发端口	2019-11-08 之前	WebSocketConfig.requestPort	38889		支持
	2019-11-08 及之后	WebSocketConfig.requestPorts	38889		支持

- a.默认端口号是【38888, 38889】，若想更改为其他端口号，端口号可设置范围是：1024~65535，若为多个值，设置格式为：[端口号 1,端口号 2,端口号 3]。
- b.建议「WebSocket 端口」设置多个值，作为备用，防止一台服务器部署了多个工程，端口被占用。
- c.不要设置端口号为服务器远程连接端口 3389。
- d.不要重复设置某个端口号既是 WebSocket 端口，又是 WebSocket 转发端口。
- e.若 WebSocketConfig.port、WebSocketConfig.requestPort、WebSocketConfig.requestPorts 字段后面有空格，配置不生效。
- f.若 WebSocketConfig.port、WebSocketConfig.requestPort、WebSocketConfig.requestPorts 大小写错误，配置不生效。
- g.WebSocketConfig.requestPort 和 WebSocketConfig.requestPorts 不能同时存在于 fine\_conf\_entity 表中，否则会出错。

**(4) 开启方案（尽量保证 jar 为最新版本）：**

**1) 单机系统**

**a.端口号**

默认端口号是【38888, 38889】，若想更改为其他端口号，端口号可设置范围是：1024~65535

## b. 开放端口

若防火墙开启，可关闭防火墙，或者单独开放端口（确保端口正常开启，且未被占用）

若云服务器有安全组或者类似的内容，需要设置端口对外开放（阿里云/华为云/微软云服务器，除了服务器自带的防火墙端口限制外，还有一层安全组的限制，需要在安全组里加一条对应端口的允许开放规则，相当于两层防火墙端口都要开放）

## 2) 集群环境

### a. 工程开放 websocket 端口 38888

b.nginx 开放代理端口，nginx.conf 文件中，server 的 listen 端口需要与「WebSocket 转发端口」配置一致

监听端口和这边的设置一致



e.g.

### FR+BI 环境

FR 默认端口号【38888, 38889】，工程需要开放 38888，若做转发，转发接口开放监听即可

BI 默认端口号【48888、49888】，工程需要开放 48888，若做转发，转发接口开放监听即可

若做了负载均衡转发，转发配置的监听端口需要正常开放，同集群环境下 nginx 配置

## 3) 是否配置代理

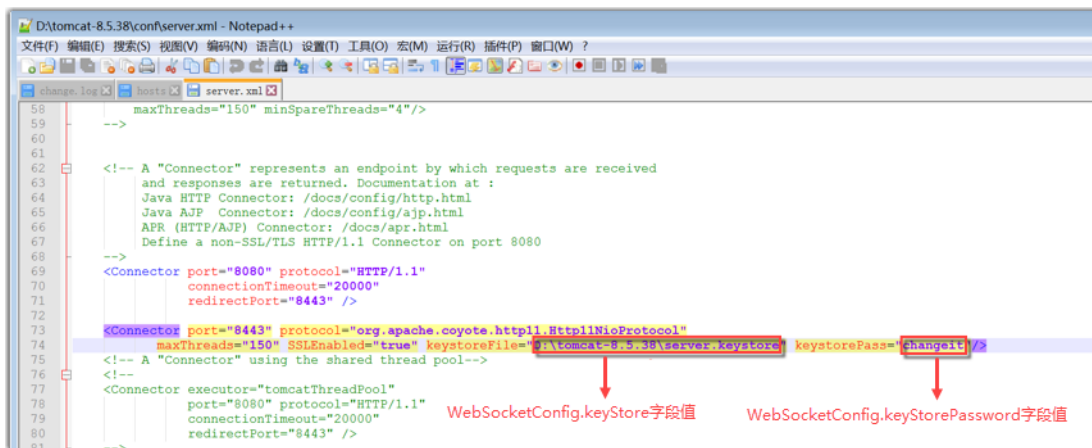
### ① 未配置代理

在 fineDB 的 fine\_conf\_entity 表中添加以下 4 个参数项，参数信息如下表所示：

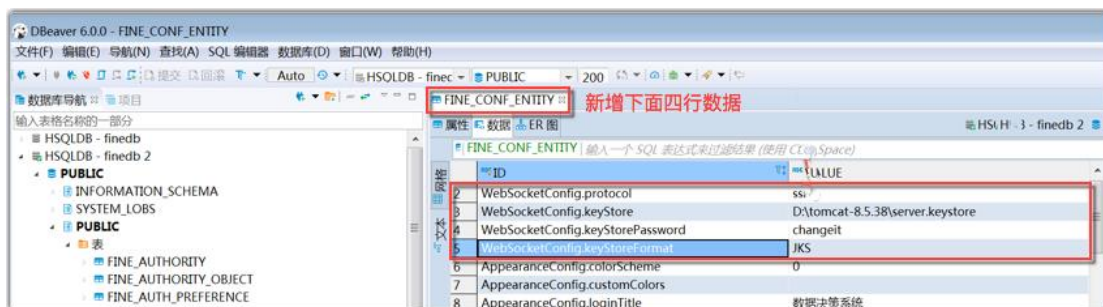
参数项	参数值
WebSocketConfig.protocol	ssl (默认)
WebSocketConfig.keyStore	来自 %TOMCAT_HOME%\conf\server.xml 中的 keystoreFile 字段的值
WebSocketConfig.keyStorePassword	来自 %TOMCAT_HOME%\conf\server.xml 中的 keystorePass 字段的值
WebSocketConfig.keyStoreFormat	JKS (默认)

其中 WebSocketConfig.keyStore 和 WebSocketConfig.keyStorePassword 的配置如下图所示：

注：fine\_conf\_entity 表中的 WebSocketConfig.keyStore 的值必须是绝对路径，不可以是相对路径。



数据库表中新增信息如下图所示：



配置好后，重启服务即可生效。

## ② 配置了代理

HTTP 默认端口是 80，HTTPS 默认端口是 443

### a. 配置 443 端口

```

#禁止在 header 中出现服务器版本，防止黑客利用版本漏洞攻击
    #server_tokens off;

    #如果是全站 HTTPS 并且不考虑 HTTP 的话，可以加入 HSTS 告诉浏览器本网站全站
    加密，并且强制用 HTTPS 访问

    #fastcgi_param    HTTPS                on;
    #fastcgi_param    HTTP_SCHEME          https;

    location / {
        proxy_http_version 1.1;
        proxy_set_header Connection "";

        proxy_buffering off;
        proxy_next_upstream http_500 http_502 http_503 error timeout
        invalid_header non_idempotent;

        proxy_redirect off;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        # 把 HTTPS 的协议告知容器（如 tomcat），否则容器可能认为是 HTTP 的请求
        proxy_set_header X-Forwarded-Proto $scheme;

        proxy_connect_timeout    20;
        proxy_read_timeout       1000;
        proxy_send_timeout       300;

        proxy_pass http://FR.com;
    }

```

### b. 配置 WebSocket 端口

Nginx 转发默认监听 38889 端口，节点的 WebSocket 默认监听 38888 端口



智能运维处，websocket 端口配置如图



## 6.3 字体规范

### 6.3.1 异常类别

- a、中文乱码（地图乱码，中文模板乱码，图表中文字乱码等）
- b、打印导出字体异常（与预览效果不一致）
- c、预览效果异常（不同电脑，预览效果存在差异）
- d、设计器字体异常（模糊）

### 6.3.2 字体展示逻辑

系统字体：设计器、服务器、浏览器所在设备的系统字体。Windows 系统所在位置：  
C:/Windows/Fonts；Linux 系统所在位置：/usr/share/fonts

服务器依赖的 JRE 下的字体：%Tomcat%jre/lib/fonts

安装「性能优化插件」后对应的字体文件：%FR\_HOME%/webroot/WEB-INF/fonts

使用方式	字体来源
设计器设计时使用的字体	设计器所在电脑的系统字体+设计器工程依赖的 JRE 下的字体+性能优化插件字体（若安装了插件）
Web 端预览时使用的字体	Web 浏览器所在电脑的系统字体
导出时使用的字体	服务器的系统字体+服务器依赖的 JRE 下的字体+性能优化插件字体（若安装了插件）
打印时使用的字体	零客户端打印时和导出相同，本地软件打印时使用本地系统字体和打印软件 JRE 下的字体

### 6.3.3 优化方案

#### 设计器

将字体拷贝到设计器的 `jre\lib\fonts` 中，重启设计器；或者下载对应的字体导入到 `C:\Windows\Fonts` 文件夹中，保证设计器可以选到该字体

## 服务器

### Windows :

检查服务器字体，若缺失，将对应字体下载好后，放到服务器依赖的 JRE 下的字体文件夹下，如：`%Tomcat%jre/lib/fonts` 文件夹下

### Linux:

#### (1) 安装语言包

##### 1) 检查中文语言包

`echo $LANG` 查看系统语言环境；

若非 `zh_CN.UTF-8`，则需要使用 `locale` 查看是否有中文语言包，有 `zh_cn` 则表示已经安装了中文语言；

如果没有中文语言包，则需执行 `yum groupinstall chinese-support` 安装中文环境

配置系统语言为中文（临时更改：`LANG=zh_CN.UTF-8`）；

##### 2) 字体包缺失检查 (`fc-list/`)

若发现缺失字体，可以直接在 `/usr/share/fonts` 目录下新建一个目录 `chinese`，把本地 Windows 系统 `C:\Windows\Fonts` 中的字体，上传到 `/usr/share/fonts/chinese` 目录下（需要注意上传工具编码必须是 `utf8`）

##### 3) 字体权限

给字体包文件夹 755 权限（`chmod -R 755 /usr/share/fonts/chinese`）

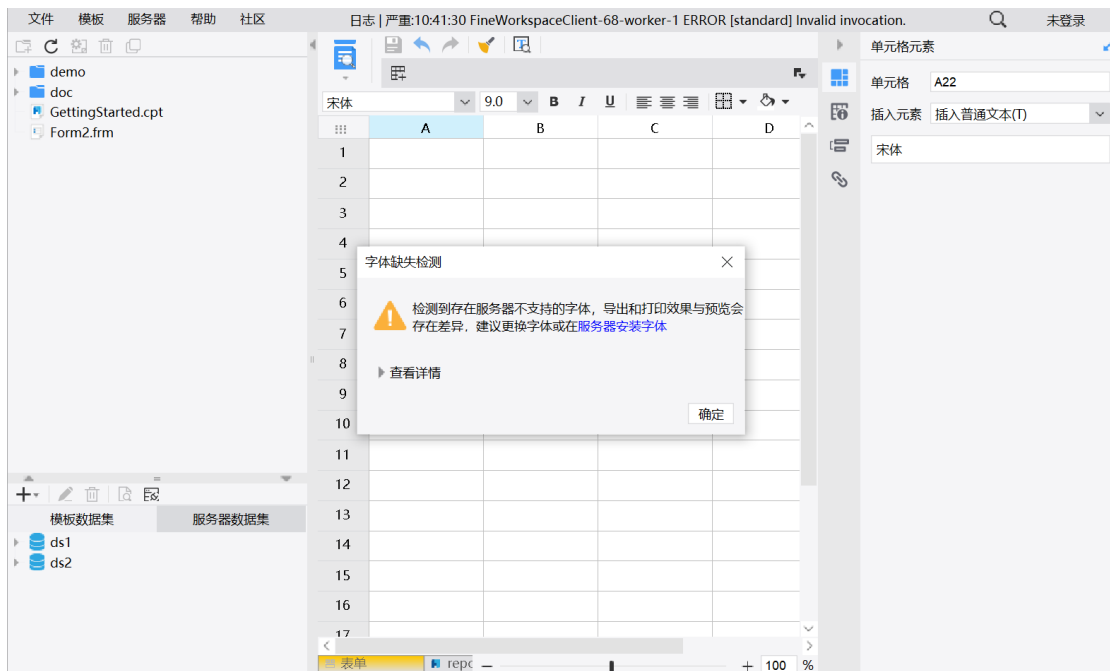
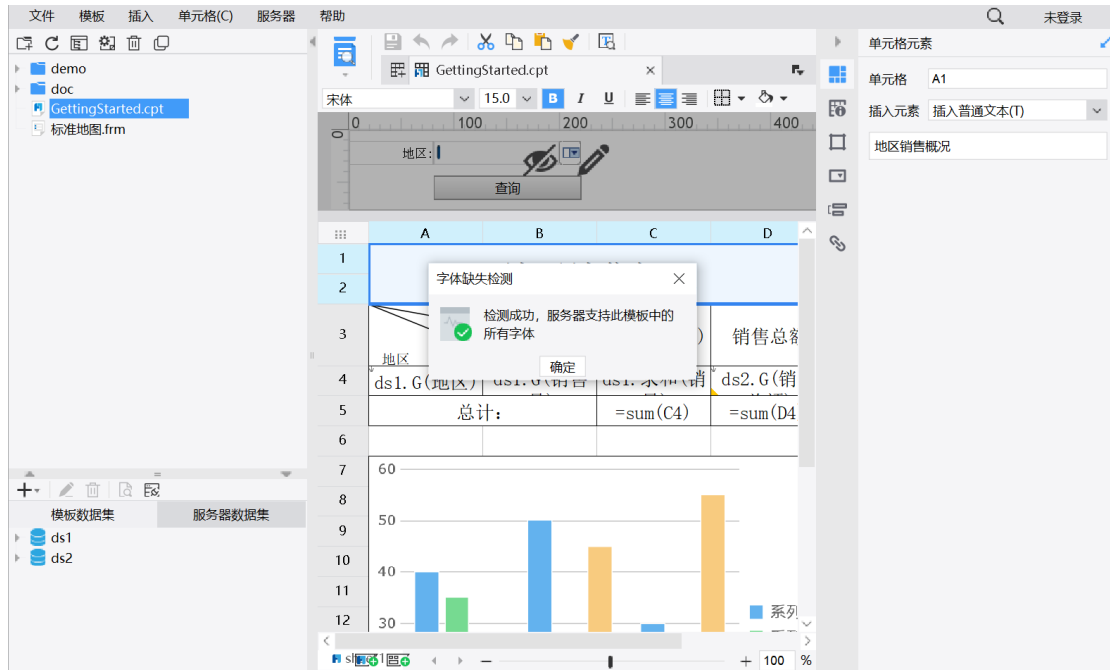
##### 4) 清空缓存

```
cd /usr/share/fonts/chinese
mkfontscale #字体扩展
mkfontdir #新增字体目录
fc-cache -fv #刷新缓存，或者执行
fc-cache /usr/share/fonts/chinese/fc-list #查看系统字体
```

##### 5) 重启服务器

#### (2) 插件优化

10.0.16 及之后版本即可安装字体检测插件，若服务用到了系统没有的字体，会提示安装，安装完重启服务即可生效，详情可参考：<https://help.fanruan.com/finereport/doc-view-4063.html?source=1>



## 6.4 系统运维工具

### 6.4.1 云端运维

可以依靠云端运维日志获取系统运行报告，从而了解系统整体使用情况、系统性能状况、排查宕机原因、了解报表系统数据、查看系统卡顿次数和内存负载的情况等。

云端运维报告-<https://help.fanruan.com/finereport/doc-view-3124.html>

云端运维使用步骤-<https://help.fanruan.com/finereport/doc-view-2789.html>

注：此功能为 10.0 版本内置的免费功能，9.0 及以下版本不支持。

### 6.4.2 宕机处理工具

部分时候宕机后为了不影响系统使用会直接重启，没有获取有效信息，对于宕机原因的排查却缺少相关的线索。10.0.15 版本后内置宕机功能，报表工程可以自动生成 DUMP 文件、重新启动系统，同时便于从 DUMP 文件中快速定位宕机原因。

宕机处理-<https://help.fanruan.com/finereport/doc-view-3921.html>

注：

(1) 10.0.15 及之后的版本新增此功能

10.0.15 以下的版本，可选择升级至支持的版本。如无法升级，可选择手动导出信息后重启。

手动导出信息方式：

#### a. thread

`jstack pid > 1.txt` pid 为工程进程号，隔 3-5 秒输出 3-5 次，每次输出的文件命名成不同名称。

#### b. dump

**windows:** `jmap -dump:format=b,file="D:\a.dump" pid`，路径要双引号

**linux:** `jmap -dump:format=b,file=/usr/a.dump pid`

(2) 仅限 Tomcat 容器部署

(3) 报表与 BI 集成的用户在使用此功能时，优先调用 BI 工程的「宕机处理」功能，禁用报表工程的「宕机处理」功能，以保证不影响 BI 工程的运行。

### 6.4.3 平台日志导出插件

工程日志默认位置：tomcat/logs

针对没权限、不方便登录服务器将日志发送技术支持查看的情况，可使用日志导出插件获取对应日志信息。

平台日志导出插件-<https://help.fanruan.com/finereport/doc-view-3160.html>

注：

- 1、导出日志默认最大存储限制 1GB，超过 1GB 后会自动清理最早产生的日志，保持最大 1GB 大小的日志文件
- 2、不会对系统默认的日志输出产生影响
- 3、仅限 10.0 版本且 jar 包版本为 2018-07-31 及之后