**项目背景**

**什么是数据可视化**

大数据已经被国家列入十三五规划，提倡开放，共享。开放共享的背后意味着人人都可以接触和进入大数据领域，企业不再为数据资源的垄断发愁，因为一切都是开放的，如何获取数据将不再是问题，困难在于数据有什么价值，用什么样的手段才能把数据的价值直观而清晰的表达出来。  
我之前看到过一篇文章《设计中的设计》，里面提出一个概念叫视觉对话。如果要两个语言、文字不通的陌生人进行沟通，给他们一张纸，一只笔，他们一定是用最简洁的方式把自己的想法画下来进行交流，这就是视觉对话。  
其实这也正是数据可视化的本质，通过可视化图表将用比文字快10倍的速度将陌生的读者带进门，大数据时代一个显著特征就是数据可视化的崛起。作为大数据最后一公里的展现环节，数据可视化将技术与艺术完美结合，借助图形化的手段，清晰有效地传达与沟通信息。  
一方面，数据赋予可视化以价值;另一方面，可视化增加数据的灵性，两者相辅相成，帮助企业从信息中提取知识、从知识中收获价值。

-- 摘自 《数据可视化的作用和实现方法》

数据可视化的几种常见模式如下：

[**商业软件的几种模式**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/preface?id=%e5%95%86%e4%b8%9a%e8%bd%af%e4%bb%b6%e7%9a%84%e5%87%a0%e7%a7%8d%e6%a8%a1%e5%bc%8f)

[**商业BI套件**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/preface?id=%e5%95%86%e4%b8%9abi%e5%a5%97%e4%bb%b6)

能够提供BI生命周期全套的解决方案，业界比较知名BI套件老牌的有IBM的**Cognos**、SAP的**BO**、Oracle的**BIEE**，以及新生军**Tableau**、**QlikView**等等，这些产品前端可视化都做得**非常强大**,报表设计、Dashboard设计、Report邮件发送、OLAP分析都不在话下

当然与之相对应的也是一笔少则几十万多则上百万的昂贵收费, 而且大数据/NoSQL/NewSQL产品五花八门、发展神速，想对所有数据源都提供支持也是很难做到的

[**第三方数据分析平台**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/preface?id=%e7%ac%ac%e4%b8%89%e6%96%b9%e6%95%b0%e6%8d%ae%e5%88%86%e6%9e%90%e5%b9%b3%e5%8f%b0)

通过SDK把数据发送到三方，如:友盟、百度统计、TalkingData、CNZZ、Google Analytics， 大概一般对数据比较敏感或者公司规模略大这种形式的数据处理都不能满足需求了,另外不乏一些优秀的创业公司或自己开发类似第一种BI套件打包软件和服务 这种解决方案比较适合没有研发能力和不想把精力过多花在软件解决方案上的小型、传统型企业

[**开源软件的几种模式**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/preface?id=%e5%bc%80%e6%ba%90%e8%bd%af%e4%bb%b6%e7%9a%84%e5%87%a0%e7%a7%8d%e6%a8%a1%e5%bc%8f)

[**图形库**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/preface?id=%e5%9b%be%e5%bd%a2%e5%ba%93)

优秀的开源的可视化开发组件，以JS库和开发包的形式提供，如：EChart、HighChart、D3、ChartJS等等，程序猿的利器，数据分析师无法直接使用

[**简单网页报表**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/preface?id=%e7%ae%80%e5%8d%95%e7%bd%91%e9%a1%b5%e6%8a%a5%e8%a1%a8)

简单网页版数据查询与图表展示，如：[**Art**](https://sourceforge.net/projects/art/)。 这类工具功能单一、对SQL查询出来的数据进行数据展示格式有严格要求，数据查询不能重用，图形简陋

[**报表工具**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/preface?id=%e6%8a%a5%e8%a1%a8%e5%b7%a5%e5%85%b7)

附录 [**12个最好的开源报表工具**](http://www.iteye.com/news/11361), 有兴趣的朋友可以自己去体验下

[**开源OLAP**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/preface?id=%e5%bc%80%e6%ba%90olap)

这一块最近比较流行的是基于Mondrian的[**Saiku**](http://www.meteorite.bi/products/saiku)和[**Pivot4J**](http://www.pivot4j.org/)。 这两种开源OLAP平台分析最大的好处是用户可以Self-Service的形式组织自己的数据、全方位多角度交互式的分析数据，我们从13年Saiku2.x的时候就开始使用，感觉非常不错，Saiku3之后并没有太多亮点，模型设计器和Saiku保存的查询组织成Dashboard也只有收费的企业版才有，想利用Saiku做多维分析，个人感觉2.6足以

[**开源BI组件**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/preface?id=%e5%bc%80%e6%ba%90bi%e7%bb%84%e4%bb%b6)

Pentaho, SpagoBI功能强大，但是系统架构非常笨重，组件复杂，难以驾驭

[**关于Dashboard**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/preface?id=%e5%85%b3%e4%ba%8edashboard)

* 免费的Dashboard设计平台，体验过如:[**dashbuilder**](http://dashbuilder.org/)、[**razorflow**](https://www.razorflow.com/)，两者在可视化控制方面做得够不够好，Razorflow Dashboard设计还需要开发人员介入，再如Python系[**Superset**](http://airbnb.io/caravel/index.html)之类产品没有细究过，个人感觉大数据产品主流还是JVM系语言开发的产品适应性比较好
* 业内另外一种普遍的Dashboard开发办法则是全定制，产品、运维提出需求想在一张页面上展示什么样的数据、以什么样的图形展示，前端工程师做好页面，后端服务准备数据。这种开发模式开发生命周期长，页面图表**不可重复使用**，比较适合业务和需求比较固定的公司。否则不断的页面新增与变更，代价非常之大

[**CBoard解决的问题**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/preface?id=cboard%e8%a7%a3%e5%86%b3%e7%9a%84%e9%97%ae%e9%a2%98)

在这样的背景下CBoard的出发点为打造一款：

* **及时数据交互:** 不需要/轻量级模型(Mondrian Schema, BO Universe)的“多维”数据图表设计Dashboard, 做到xxx广告宣称的"Type SQL, Get Chart"。
* **拖拽维度交互:** 借鉴了**Databrick Cloud Platform**和**Zeppelin**的可视化灵感，SQL或其他任何查询结构返回的二维数据表之后让用户通过简单拖拽列的组合方式进行数据透视、聚合与展示。
* **不仅仅限于JDBC开放的数据连接适配接口:** (后称DataProvider) 目前CBoard支持**JDBC数据源**，**Saiku2.x数据源**，Kylin1.6，Elasticsearch 1.x, 2.x, 5.x，但是用户可以在不用更改前端界面的前提下继承DataProvider接口，返回普通二维数组即可，交互所需参数前端自动生成相应交互界面。

**快速开始**

[**自己编译构建项目**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/install?id=%e8%87%aa%e5%b7%b1%e7%bc%96%e8%af%91%e6%9e%84%e5%bb%ba%e9%a1%b9%e7%9b%ae)

**注意:**

为了避免由于JDK版本不匹配引起的各种问题强烈建议自行编译构建项目。

[**项目要求**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/install?id=%e9%a1%b9%e7%9b%ae%e8%a6%81%e6%b1%82)

* JDK1.8 (Java环境)
* Maven3
* MySQL5+/SQLServer (系统元数据存储)
* Chrome (客户端主流浏览器访问现在只支持Chrome)
* 我们用的图表设计的拖拽现在测试下来只能支持chrome，我们是公司内部使用，对于兼容性没有太在意。由于我们没有专业前端开发，兼容性问题搞不定。拖拽使用的组件来自[**angular-drag-and-drop-lists**](https://github.com/marceljuenemann/angular-drag-and-drop-lists), 希望有前端开发资源的朋友能够帮忙一起解决兼容性问题
* 并配置好maven仓库，这里推荐一个阿里云的maven镜像，速度还不错

<mirror>

<id>alimaven</id>

<name>aliyun maven</name>

<url>http://maven.aliyun.com/nexus/content/groups/public/</url>

<mirrorOf>central</mirrorOf>

</mirror>

* [**PhantomJS**](http://phantomjs.org/) Version 2.1+ (用于看板导出与邮件发送)
* 基础的数据库、数据仓库、OLAP数据分析知识或者Excel透视表使用经验
* 系统维护与管理者需要有一定的J2EE项目经验

[**获取代码**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/install?id=%e8%8e%b7%e5%8f%96%e4%bb%a3%e7%a0%81)

github上下载或者clone项目源代码

**警告**

CBoard仓库中以branch-开头的分支为开发分支，经过内部测试使用之后在分支上发布可用版本。master仅用来同步分支最新代码，非发布、不稳定分支，请不要在master进行项目构建。

git clone https://github.com/yzhang921/CBoard.git

git checkout branch-0.4

[**准备CBoard元数据库**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/install?id=%e5%87%86%e5%a4%87cboard%e5%85%83%e6%95%b0%e6%8d%ae%e5%ba%93)

以MySQL为例

[**1. 从演示数据库开始**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/install?id=_1-%e4%bb%8e%e6%bc%94%e7%a4%ba%e6%95%b0%e6%8d%ae%e5%ba%93%e5%bc%80%e5%a7%8b)

* 你可以从用于演示的样本数据库开始，包含样本元数据库(cboard\_demo2)和样本DW库(foodmart2)
* 下载[**cboard\_demo & foodmart**](https://pan.baidu.com/s/1skOyPh7)
* 终端命令行进入下载这两个脚本的目录，并解压foodmart.zip
* 进入MySQL命令行工具，执行下面两个脚本文件
* mysql> source cboard\_demo.sql

mysql> source foodmart.sql

* 成功执行之后检查 cboard\_demo2 和 foodmart2 两个DB已经成功导入，最后查看一下数据

[**2. 从空白元数据库开始**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/install?id=_2-%e4%bb%8e%e7%a9%ba%e7%99%bd%e5%85%83%e6%95%b0%e6%8d%ae%e5%ba%93%e5%bc%80%e5%a7%8b)

执行元数据创建脚本：

$ cd cboard

mysql> create database cboard;

mysql> use cboard;

mysql> source sql/mysql/mysql.sql（项目源文件）

[**修改配置文件**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/install?id=%e4%bf%ae%e6%94%b9%e9%85%8d%e7%bd%ae%e6%96%87%e4%bb%b6)

src/main/resources/config.properties

validationQuery=SELECT 1

jdbc\_url=jdbc:mysql://localhost:3306/cboard # 修改指向第一步元数据库地址

jdbc\_username=root # 修改数据库用户名连接密码

jdbc\_password=111111

# Service configuration

dataprovider.resultLimit=300000

admin\_user\_id=1

phantomjs\_path=D:/phantomjs-2.1.1-windows/bin/phantomjs.exe # 修改为服务器上的phantomjs位置，linux环境下确保有执行权限

# 邮件发送配置

mail.smtp.host=127.0.0.1

mail.smtp.port=8825

mail.smtp.from=test@test.com

#mail.smtp.username=test@test.com

#mail.smtp.password=111111

#mail.smtp.ssl.checkserveridentity=false

# Cache Properties 使用Redis做数据缓存的配置

cache.redis.hostName=127.0.0.1

cache.redis.port=6379

[**编译构建**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/install?id=%e7%bc%96%e8%af%91%e6%9e%84%e5%bb%ba)

进入项目根目录(pom.xml所在目录)

**注意**

由于我们再maven依赖中默认包含了SQLServer的JDBC驱动，但是该驱动在公有的Maven仓库中不存在，所以在不修改maven依赖的前提下构建项目需要本地安装一次SQLServer JDBC Driver； 如果贵司使用环境中不涉及SQLServer数据库，可以选择把pom中下面依赖移除：

<dependency>

<groupId>com.microsoft.sqlserver</groupId>

<artifactId>sqljdbc4</artifactId>

<version>4.0</version>

</dependency>

$ mvn install:install-file -Dfile=lib/sqljdbc4-4.0.jar -DgroupId=com.microsoft.sqlserver -DartifactId=sqljdbc4 -Dversion=4.0 -Dpackaging=jar

$ mvn clean package

[**部署war到tomcat容器**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/install?id=%e9%83%a8%e7%bd%b2war%e5%88%b0tomcat%e5%ae%b9%e5%99%a8)

* 拷贝CBoard\target\cboard.war到tomcat的webapp目录
* 启动tomcat

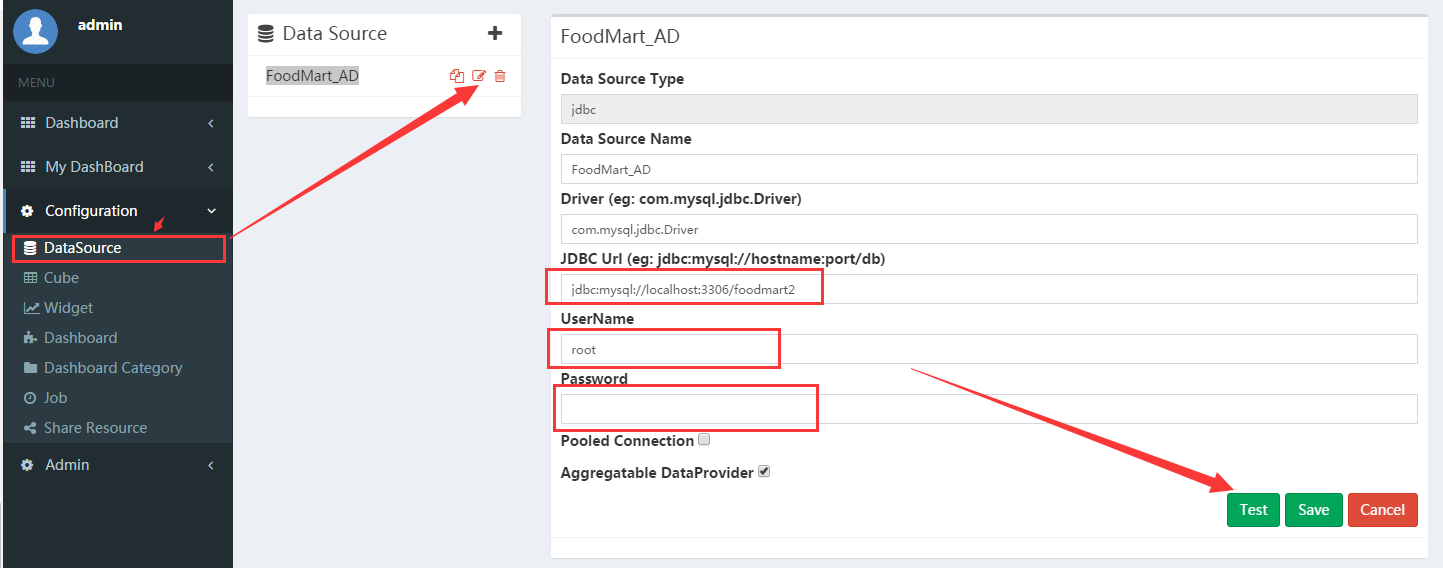
[**访问应用**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/install?id=%e8%ae%bf%e9%97%ae%e5%ba%94%e7%94%a8)

通过Chrome访问应用  
[**http://\_yourserverip\_:8080**](http://_yourserverip_:8080/)  
默认登录用户名和密码: admin/root123

[**样本数据源是否配置正确**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/install?id=%e6%a0%b7%e6%9c%ac%e6%95%b0%e6%8d%ae%e6%ba%90%e6%98%af%e5%90%a6%e9%85%8d%e7%bd%ae%e6%ad%a3%e7%a1%ae)

**注意**

为了数据安全，在编辑数据源的时候，密码不会被加载回前端。故每次测试数据连接和修改数据连接的时候请再次输入连接密码。



[**CBoard使用基本步骤**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/install?id=cboard%e4%bd%bf%e7%94%a8%e5%9f%ba%e6%9c%ac%e6%ad%a5%e9%aa%a4)

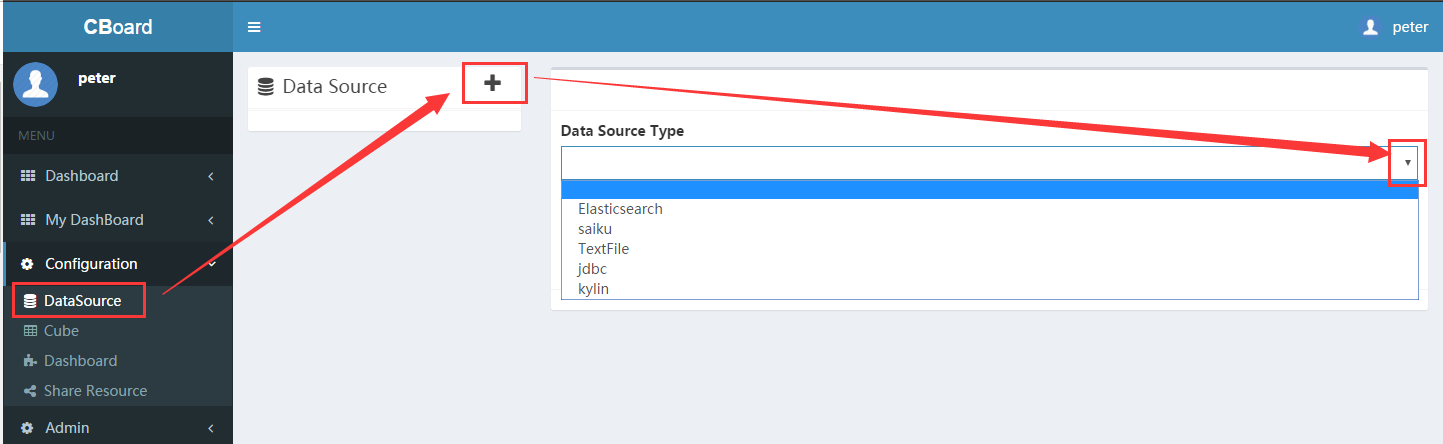


**数据源管理**

**目前CBoard支持以下数据源连接**

* JDBC数据源
* ElasticSearch 1.x, 2.x, 5.x (原生读取Index与Mapping，根据用户拖拽生成查询DSL)
* Kylin 1.6 (原生读取kylin Model，根据用户拖拽生成查询SQL)
* TextFile (文本文件，文本需要存放于CBoard应用服务器上面，读取本地文件)
* Saiku2.x (读取Saiku图表数据而非集成Saiku生成图表)
* Solr4.x, 5.x (读取Solr collection，根据用户拖拽生成Solr查询语法，4.x无法使用后台聚合功能)

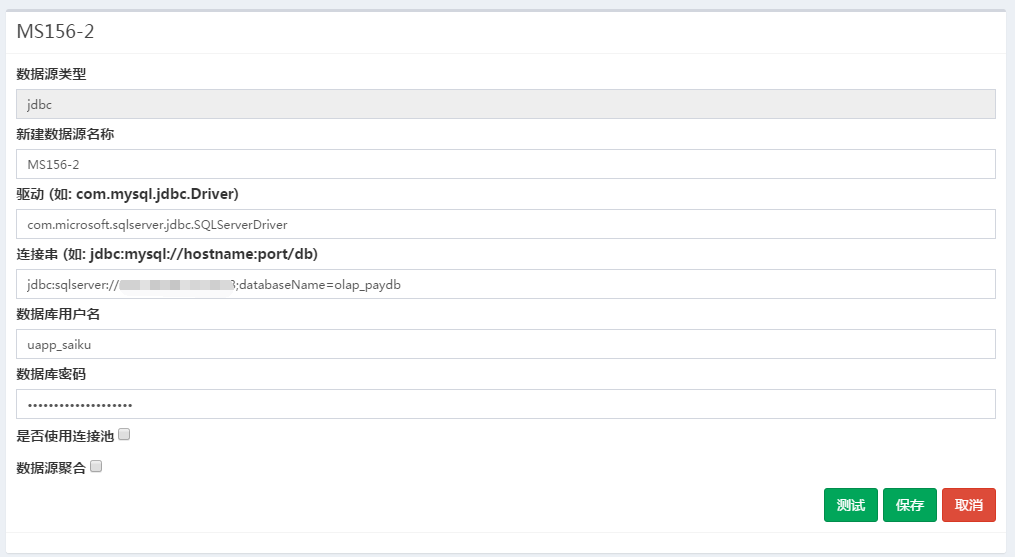
操作路径图如下：



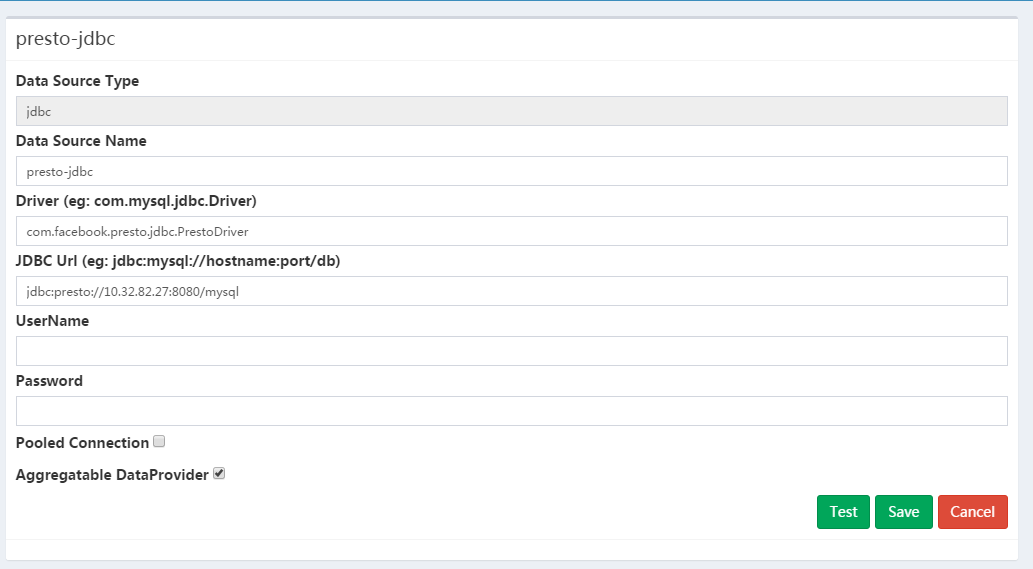
[**1、 JDBC数据源**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/datasource?id=_1-jdbc%e6%95%b0%e6%8d%ae%e6%ba%90)

我们来点实际的，在此我们不纯粹为了宣传而罗列各种数据库产品，懂JDBC原理的都知道没有意义。**基本上所有关系型数据库都能支持.**

或许会存在一点点语法兼容问题，但是修改起来都比较简单。遇到后台日志抛语法错误异常请在github上面提issue，附上详细的异常日志，和操作步骤，我们会协助解决，更加欢迎提PR合并解决方案到我们的版本。 目前我们生产环境上连接的是MySQL和SQLServer。



[**presto**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/datasource?id=presto)

另外，**Presto也是通过JDBC的方式连接**。Presto的JDBC连接认证**没有用户名密码认证机制**，用户名和密码输入留空即可。 

[**Impala**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/datasource?id=impala)

**Impala也是通过JDBC的方式连接**，Impala的JDBC连接驱动可以是Impala自己的驱动com.cloudera.impala.jdbc4.Driver ，也可以是Hive的驱动org.apache.hive.jdbc.HiveDriver。

使用Impala驱动时无认证的URL为jdbc:impala://localhost:21050/default，有认证的URL为jdbc:impala://localhost:21050/default;AuthMech=3。

使用Hive驱动时无认证的URL为jdbc:hive2://localhost:10000/default;auth=noSasl，有认证的URL为jdbc:hive2://localhost:10000/default。

[**关于Hive和Spark**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/datasource?id=%e5%85%b3%e4%ba%8ehive%e5%92%8cspark)

理论上Hive、Spark-SQL也可以通过JDBC的方式连接，考虑到二者执行效率都不是太高不适合做及时交互查询引擎。不推荐引入

[**驱动引入方式**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/datasource?id=%e9%a9%b1%e5%8a%a8%e5%bc%95%e5%85%a5%e6%96%b9%e5%bc%8f)

添加JDBC数据源之前确认WEB-INF\lib下面已经包含了对应的JDBC驱动包，**MySQL/SQLServer/Kylin的JDBC驱动不需要额外引入。**

* Maven方式在打包之前修改pom.xml(推荐),下面是mysql的maven依赖：

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.24</version>

</dependency>

* 应用在Web容器解包之后拷贝驱动到WEB-INF\lib，之后**重启Tomcat**
  + 填写完毕之后可以点击**测试**按钮，输入简单的sql查询测试数据源与CBoard之间的连通性
  + 0.3新增两项配置
  + 是否使用连接池 设置数据源是否通过druid连接池的方式连接数据源
  + 数据源聚合 0.3之后的版本支持聚合下推数据源

[**2 、Kylin**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/datasource?id=_2-kylin)

[**JDBC方式连接**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/datasource?id=jdbc%e6%96%b9%e5%bc%8f%e8%bf%9e%e6%8e%a5)

0.2版本添加了kylin的jdbc依赖，可以直接连接，连接方式如下

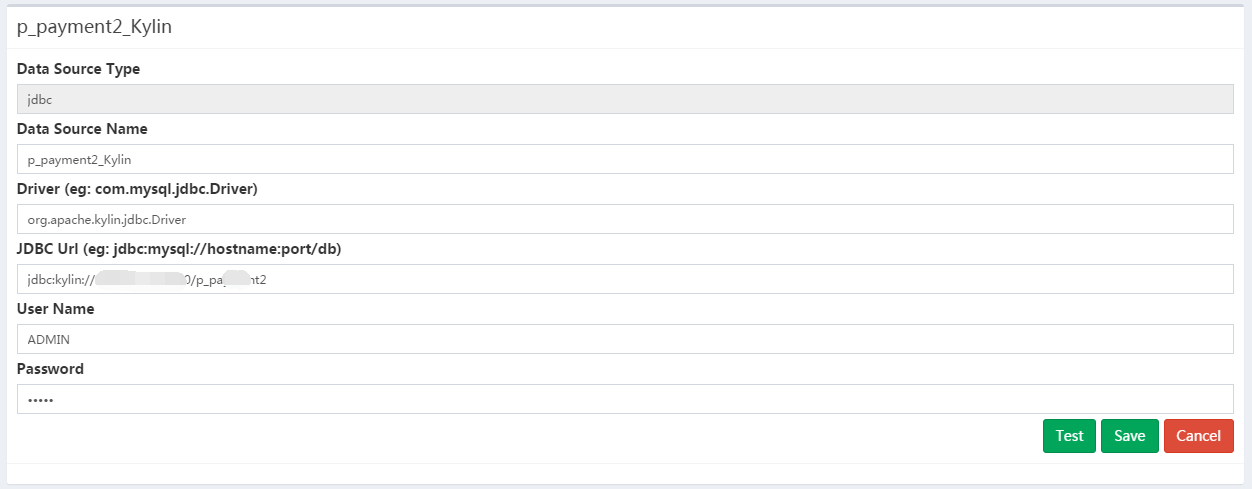
Driver ：org.apache.kylin.jdbc.Driver

JdbcUrl：jdbc:kylin://<host>:<port>/<project>

用JDBC方式连接Kylin可以选数据源聚合也就是聚合下推Kylin，或者使用CBoard提供的内置聚合器，通过查询预聚合返回小的结果集，结果集缓存在CBoard服务器上做二次聚合。 聚合下推原理参照[**性能优化章节**](https://peter_zhang921.gitee.io/cboard_docsify/#/../discuss/optimize)。

**注意:**

**非数据源聚合的连接，切记注意数据集或者Query查询结果集大小。**



**关于Kylin1.6对子查询支持结果异常问题：**

CBoard中采用的基于数据集定义的子查询嵌套之后group by的形式聚合下推数据源。 我们在实际使用中遇到过查询结果不准的bug，遇此情况可以通过图表设计模块下预览查询功能查看调试实际执行的Query。

或者改用下面Kylin原生数据源

[**Kylin原生连接**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/datasource?id=kylin%e5%8e%9f%e7%94%9f%e8%bf%9e%e6%8e%a5)

0.3版本之后支持连接Kylin1.6, 0.4兼容kylin 2.0, 2.1, 原生通过restful接口读取解析Kylin数据模型，

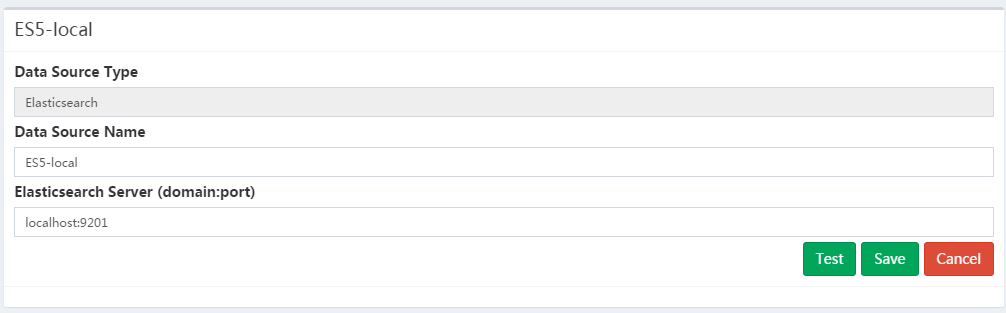
**原生连接模式下聚合操作只会在数据源进行。**



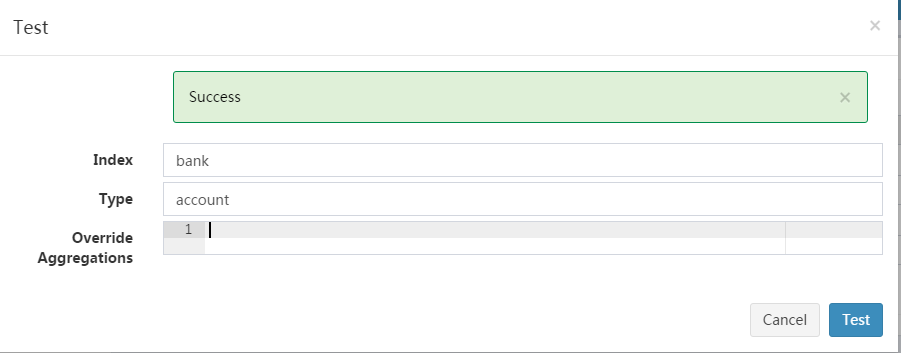
[**3 、Elasticsearch**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/datasource?id=_3-elasticsearch)

为搜索而生的ES，最近几年在数据分析领域使用得越来越广，ELK技术栈中可视化工具，Kibana5之前的版本基本不具备多维分析功能，Kibana5具备了简单的指标分组split metric功能，但是还是缺少不同聚合指标的计算功能（常用于计算转化率与占比之类的需求）（P.S. 或许是本人还不会用，有说错的地方请指正），所以CBoard0.3版本更新特意在原计划之外评估比较和Saiku3数据源的支持之后决定选择先支持ES。

[**ES 数据源配置**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/datasource?id=es-%e6%95%b0%e6%8d%ae%e6%ba%90%e9%85%8d%e7%bd%ae)



[**连通性测试**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/datasource?id=%e8%bf%9e%e9%80%9a%e6%80%a7%e6%b5%8b%e8%af%95)

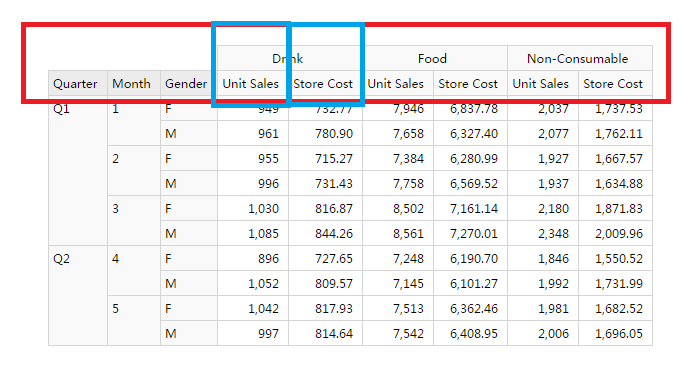


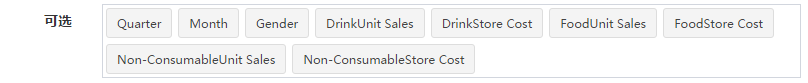
[**4 、Saiku2.x**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/datasource?id=_4-saiku2x)

CBoard实现了对Saiku2.6的数据源连接，Saiku3.x的Restful接口相比2.6有些变动，暂不支持。Saiku生成图表的样式和CBoard本身不太搭配，这里Saiku集成读取Saiku图表数据把Saiku保存之后的图表通过Restful请求查询，返回数据作为数据集，再在CBoard上渲染生成图表，而非集成Saiku生成图表。

[**Saiku数据获取说明**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/datasource?id=saiku%e6%95%b0%e6%8d%ae%e8%8e%b7%e5%8f%96%e8%af%b4%e6%98%8e)

用过Saiku的可能会在这里产生疑问，Saiku的交叉表行列表头往往有层级关系，下图展示了一个2级列表头

  
当前CBoard获取数据的默认以粒度最小的列作为表头，**两级表头会合并为一级**，合并之后新的表头效果如下图所示：



[**5 、TextFile 文本数据源**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/datasource?id=_5-textfile-%e6%96%87%e6%9c%ac%e6%95%b0%e6%8d%ae%e6%ba%90)

文本数据只能读取CBoard应用服务器本地文件，使用者需要自己架设ftp服务器单独维护文本文件的上传与管理。



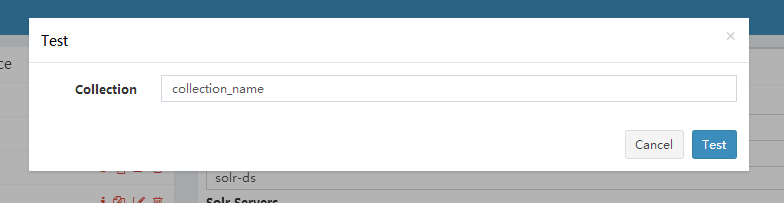
[**6、 Solr**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/datasource?id=_6-solr)

在CBoard0.4版本中加入了Solr数据源，Solr4.x没有分组集合功能，5.0+才可以使用数据源集合功能。

[**Solr数据源配置**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/datasource?id=solr%e6%95%b0%e6%8d%ae%e6%ba%90%e9%85%8d%e7%bd%ae)



[**Solr连通性测试**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/datasource?id=solr%e8%bf%9e%e9%80%9a%e6%80%a7%e6%b5%8b%e8%af%95)



**数据集管理**

**数据集**

**数据集**类似于OLAP分析的Cube(数据立方体)，可以提前定义查询、聚合表达式、动态时间漏斗。在用户数据模型比较稳定的前提下，可以减少相同数据集下不同表报设计时重复的填写查询脚本、新建聚合表达式工作。

CBoard的数据集为轻模型，任何一条简单的查询输出都可以当做一个cube来使用，cube修改避免了常规数据建模、修改、发布测试的繁琐过程。打个比方，第一次业务需求你可能选择了，a,b,c,d四个列作为聚合维度建立了一个Cube

SELECT a, b, c,

sum(xx) AS m1, count(yyy) AS m2

FROM fact a

JOIN dima a ON ...

JOIN dimb b ON ...

GROUP BY a, b, c

之后有了新的需求，需要引入维度e, f, g 列，常规的报表工具需要重新引入表、更新模型定义、发布，这个时候CBoard轻模型的优势就体现出来了只需要简单的修改或者复制之前的模型再修改一下查询

SELECT a, b, c, **e, f, g**

sum(xx) AS m1, count(yyy) AS m2

FROM fact a

JOIN dima a ON ...

JOIN dimb b ON ...

GROUP BY a, b, c, **e, f, g**

**0.2**之前的查询数据结果数据会全部读取到浏览器端，OLAP操作都在浏览器端进行，使用不当的情况下会造成客户端浏览器压力巨大。  
**0.3**引入数据源聚合，实现了JDBC、Kylin、Elasticsearch的数据源聚合功能。同时考虑到小企业存在原始数据大、数据源配置低，但是需要分析的最终数据较小的应用场景，所以保留了离线数据集特性，使用时只需在JDBC数据源创建时不要勾选数据源聚合即可。

[**1、数据集模型定义**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/dataset?id=_1-%e6%95%b0%e6%8d%ae%e9%9b%86%e6%a8%a1%e5%9e%8b%e5%ae%9a%e4%b9%89)

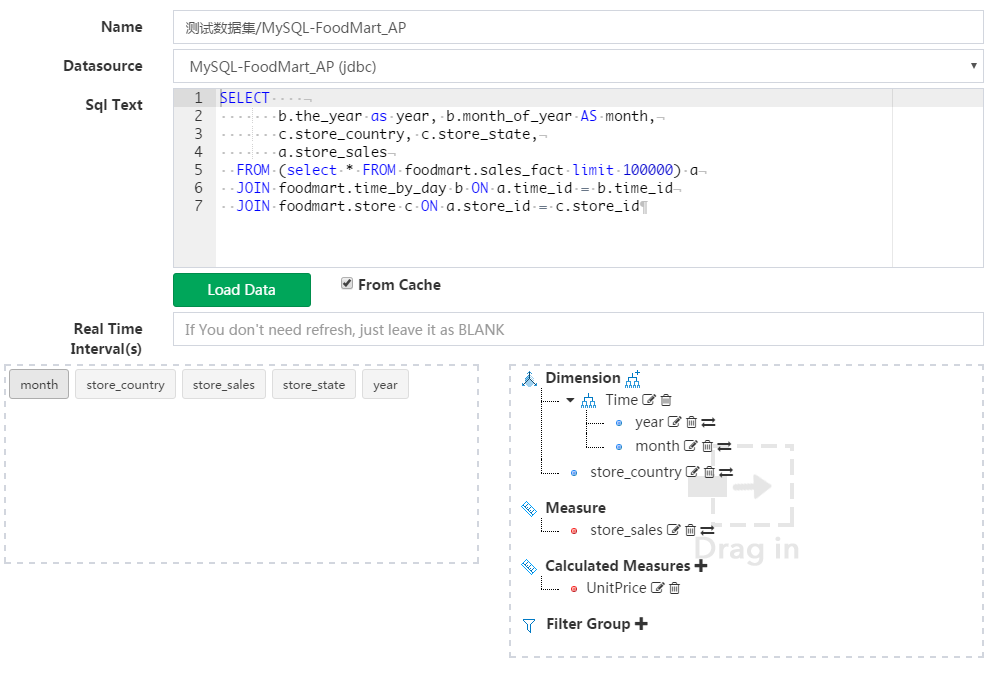
数据集的schema包含维度列、度量列、聚合表达式、预定义漏斗

* 选择数据源，填写对应的查询脚本，JDBC数据源为查询SQL，读取数据
* 读取数据成功之后，可选列和Schema空树出现在页面下方
* 拖拽左边方框的列到右边维度节点/度量节点下方，也可以通过点击左边的可选列，快速把列添加到schema，默认添加到维度节点，之后可通过功能键切换到度量
* 一个列可在不同的维度层级下多次使用，如：年->月->日，年->周->日
* 维度列在图表设计时只能拖拽到维度栏；指标(度量)列只能退拽到指标栏，聚合函数需要在设计时选定
* 加入Schema树的列可以编辑修改，之后拖拽相应的列到层级组节点下
* 层级是图表下钻基础
* 计算表达式和过滤组通过点击添加新建

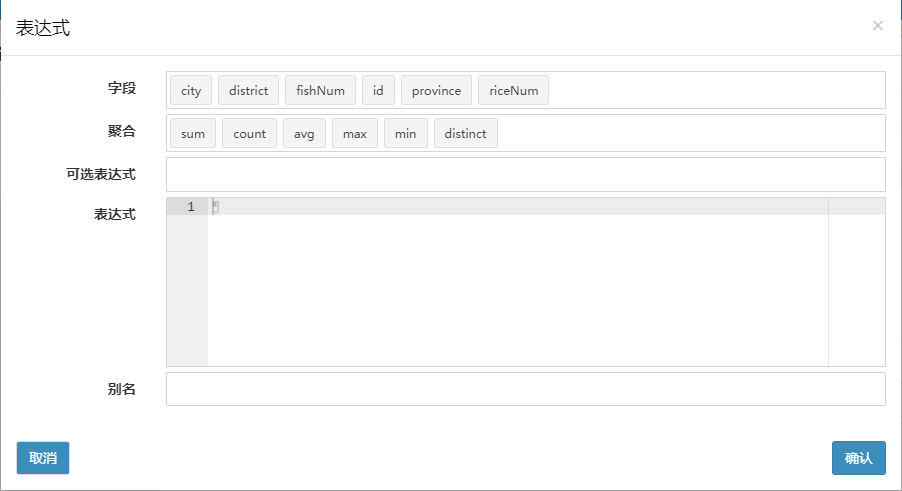
[**关于数据集更新间隔**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/dataset?id=%e5%85%b3%e4%ba%8e%e6%95%b0%e6%8d%ae%e9%9b%86%e6%9b%b4%e6%96%b0%e9%97%b4%e9%9a%94)

这个问题影响到看板是如何做刷新的? 我们知道一个看板里面可以组合来自不同数据集的图表, 但是不是所有的数据集的图表都需要刷新. 因此我们把看板上图表的刷新和数据集绑定在一起了, 在设计数据集的时候设置实时刷新时间间隔, 可以保持空，留空则不做后台刷新，填入大于0的值, 看板展示的时候里面要是存在设置了刷新间隔的数据集所创建的图表, 该图表就会按设置的时间间隔重新读取数据并更新

Real Time Interval(s): 数据集刷新间隔(秒), 有了数据集刷新间隔之后, 该数据集下的图表在看板会在设定的时间重新获取数据更新图表



[**2、预定义聚合表达式**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/dataset?id=_2-%e9%a2%84%e5%ae%9a%e4%b9%89%e8%81%9a%e5%90%88%e8%a1%a8%e8%be%be%e5%bc%8f)

1. 编辑聚合表达式并测试正确，表达式是用于聚合后再计算.  
   如: Math.log(sum(columnA)/count(columB)), 按钮只能检测到表达式的格式是否基本正确，复杂表达式不能保证验证成功之后后续100%能够工作。
2. 编辑栏替换为ace编辑器之后之前的点击辅助输入出现了一些兼容性问题，可以借助输入提示输入!  
   
3. 数据集中预定义的聚合表达式在图表设计时不支持修改

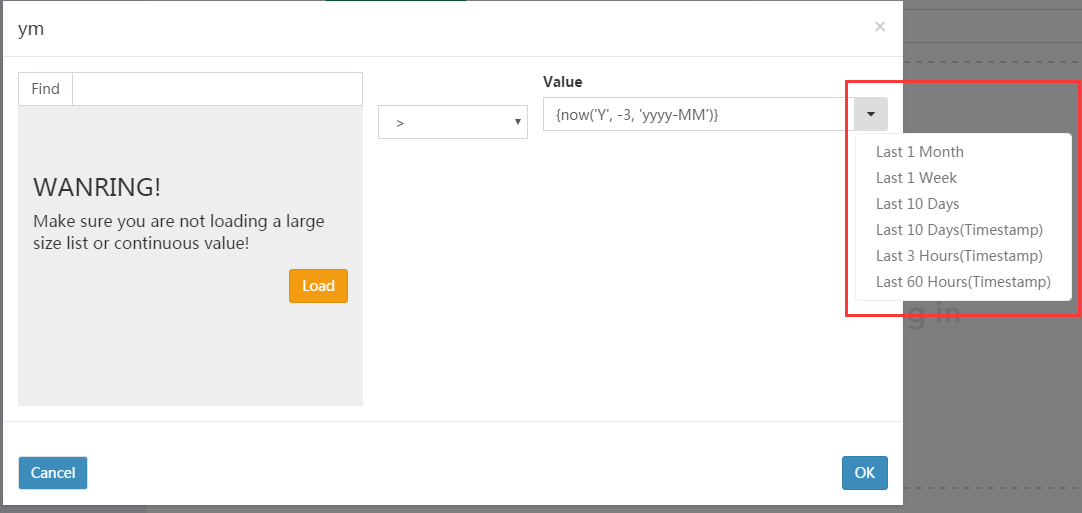
**sql的count distinct:**

表达式里面的distinct(col)与sql的count(distinct col)等价, 直接写count(distinct col)是错误的写法

**怎么实现count(case when col > 10 then 1 else 0 end)**

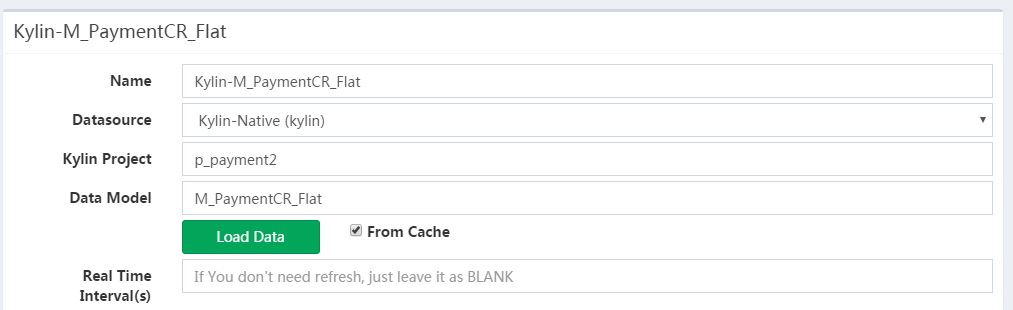
加个引号就可以了, 表达式会把里面的语句当成一个整体 count("case when col > 10 then 1 else 0 end")

[**3、预定义漏斗（过滤器）**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/dataset?id=_3-%e9%a2%84%e5%ae%9a%e4%b9%89%e6%bc%8f%e6%96%97%ef%bc%88%e8%bf%87%e6%bb%a4%e5%99%a8%ef%bc%89)

用于预定义动态日期窗口，点击下拉选择动态时间表达式模板，模板中的值可以编辑文本。用户可根据自己的需求改成任意大小时间窗口。 

[**4、特殊数据集的查询定义说明**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/dataset?id=_4-%e7%89%b9%e6%ae%8a%e6%95%b0%e6%8d%ae%e9%9b%86%e7%9a%84%e6%9f%a5%e8%af%a2%e5%ae%9a%e4%b9%89%e8%af%b4%e6%98%8e)

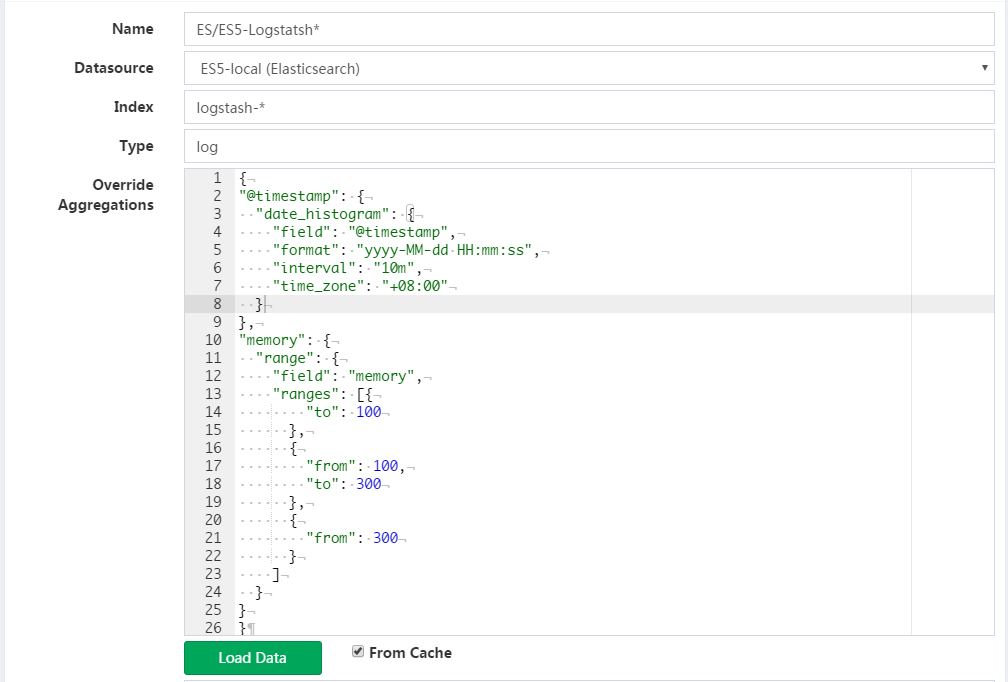
[**4.1 Kylin Native**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/dataset?id=_41-kylin-native)



使用Kylin数据源之前需对Kylin基本原理有所了解。 需要填写项, 以及解释如下：

* **Kylin Project**：对应Kylin本身的Project
* **Data Model**：对应Kylin Model

[**4.2 Elasticsearch**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/dataset?id=_42-elasticsearch)



为了减少对ES版本以及第三方ES库的依赖，CBoard采取restful + DSL连接与查询模式。所以ES的使用需要使用者（尤其是建模者）对ES基本概念有所了解，之外还需要掌握一些DSL语法。 ES查询需要参数需要精确到Type，Index名称可使用通配符原理与ES DSL查询URL参数一样。  
由于ES里面存储的数据大多为明细数据，时间维度聚合的时候粒度需要调整为时间段date\_histgram，而默认的聚合级别为term也就是关键词，所以在使用时间维度聚合之前需要调整聚合粒度。

[**4.2.1 Override Aggregations**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/dataset?id=_421-override-aggregations)

CBoard提供了三类常见的聚合Bulket覆盖辅助输入，具体可配置参数请参考官方文档[**Bucket Aggregations**](https://www.elastic.co/guide/en/elasticsearch/reference/current/search-aggregations-bucket.html)

1. date\_hist: 日期类型直方图聚合
2. number\_range: 数字区间聚合
3. number\_hist：数字直方图



语法如下(可以重复覆盖多个列的聚合):

{

"columnname":{

"<aggregation\_type>" : {

<aggregation\_body>

}

}

}

---------------------------------

样例：

对时间戳字段timestamp(long类型)按每10分钟一段进行聚合, 数字memory按自定义区间聚合

{

"timestamp": {

"date\_histogram": {

"field": "timestamp",

"format": "yyyy-MM-dd HH:mm:ss",

"interval": "10m",

"time\_zone": "+08:00"

}

},

"memory": {

"range": {

"field": "memory",

"ranges": [

{

"to": 10000

},

{

"from": 10000,

"to": 30000

},

{

"from": 30000

}

]

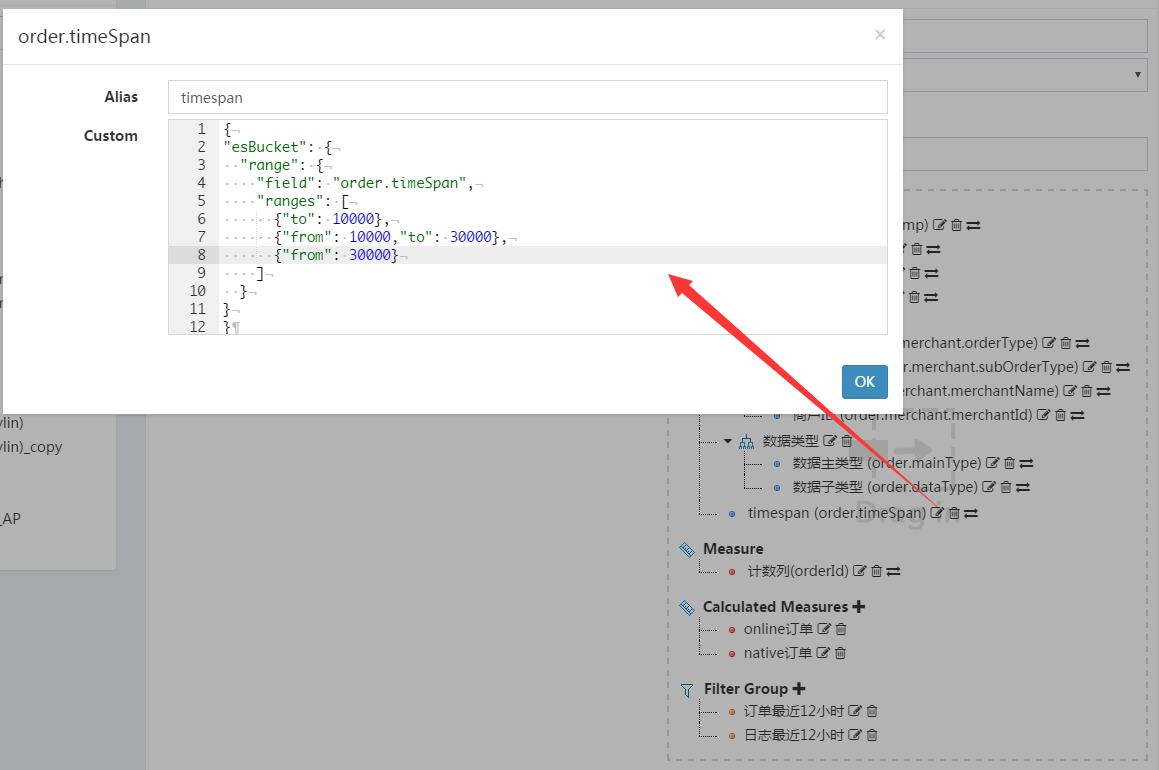
}

}

}

[**4.2.2 相同列不同聚合Bucket设置(Custom扩展)**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/dataset?id=_422-%e7%9b%b8%e5%90%8c%e5%88%97%e4%b8%8d%e5%90%8c%e8%81%9a%e5%90%88bucket%e8%ae%be%e7%bd%aecustom%e6%89%a9%e5%b1%95)

在4.2.1中的聚合覆盖设置在数据集查询之上，相当于全局查询列聚合覆盖；同时如果想对一个列采取多种分桶策略，可以在Schema树上多次引用可选列，然后编辑custom信息



{

"esBucket": {

"<aggregation\_type>" : {

<aggregation\_body>

}

}

}

**注意:**

esBucket为内置关键词，不能替换

[**4.2.3 高级聚合表达式**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/dataset?id=_423-%e9%ab%98%e7%ba%a7%e8%81%9a%e5%90%88%e8%a1%a8%e8%be%be%e5%bc%8f)

除了常规聚合统计之外，ES的聚合表达式还支持[**Filter Aggregation**](https://www.elastic.co/guide/en/elasticsearch/reference/current/search-aggregations-bucket-filter-aggregation.html), 用作条件统计，或者静态占比

语法如下:

count/sum/avg/max/min("{

'coulumn': '用于聚合的column',

'filter': <filter\_body>

}")

-------------------------------

count("{

'column': 'orderId',

'filter': {

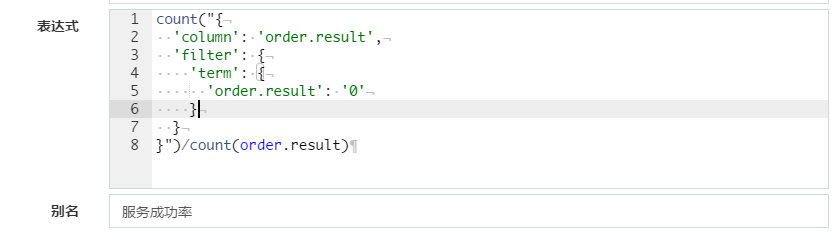
'term': {

'order.platform': 'online'

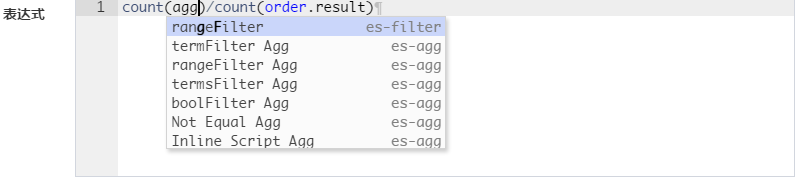
}

}

}")



CBoard也内建了一些常用过滤器输入辅助



**图表设计**

[**获取数据**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/widget?id=%e8%8e%b7%e5%8f%96%e6%95%b0%e6%8d%ae)

[**已有数据集**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/widget?id=%e5%b7%b2%e6%9c%89%e6%95%b0%e6%8d%ae%e9%9b%86)

已有数据集能够加载之前预定义的聚合表达式、过滤漏斗, 不能在图表设计页新建表达式

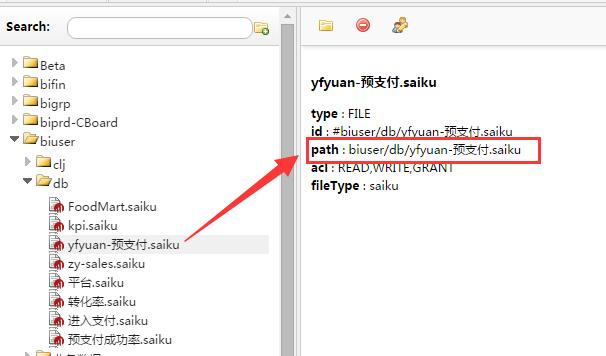
[**读取JDBC数据**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/widget?id=%e8%af%bb%e5%8f%96jdbc%e6%95%b0%e6%8d%ae)

默认获取数据方式为读取已有数据集，点击切换为查询获取数据之后，选择要查询的**JDBC数据源**需要填写读取数据对应的*SQL TEXT*，真正实现**Type SQL, Get Chart的体验。**

[**读取Saiku2.x数据**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/widget?id=%e8%af%bb%e5%8f%96saiku2x%e6%95%b0%e6%8d%ae)

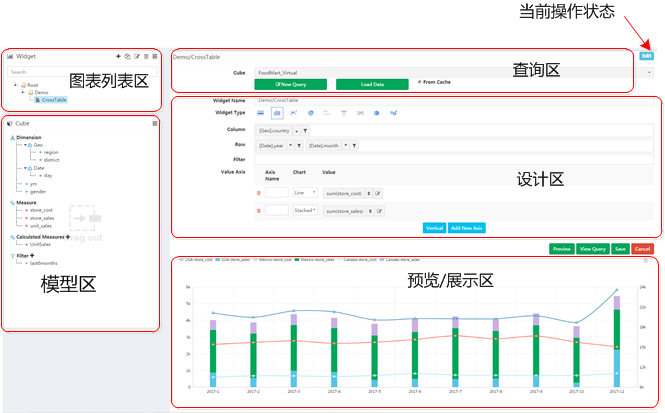
**Saiku数据**需要填写Saiku Repository下面保存查询的path

**Saiku Repository**



**CBoard Saiku Query**





图表设计功能区

[**界面模块介绍**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/widget?id=%e7%95%8c%e9%9d%a2%e6%a8%a1%e5%9d%97%e4%bb%8b%e7%bb%8d)

[**图表列表区**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/widget?id=%e5%9b%be%e8%a1%a8%e5%88%97%e8%a1%a8%e5%8c%ba)

* 显示已保保存的图表
* 左上角功能键可以新建、复制、编辑、删除图表，除新建图表外，其他的操作需要先选中单个图表
* 删除也一样只能删除单个图表
* 用户在保存列表名称时候按照[Folder]?/[SubFolder]\*/[图表名称]的格式保存之后，会自动目录树结构
* 列表中的文件加为虚拟目录，实际上并不存在文件夹实体
* **鼠标双击图表**进入编辑状态
* 选中图表**键盘Delete**可以快捷删除一个图表

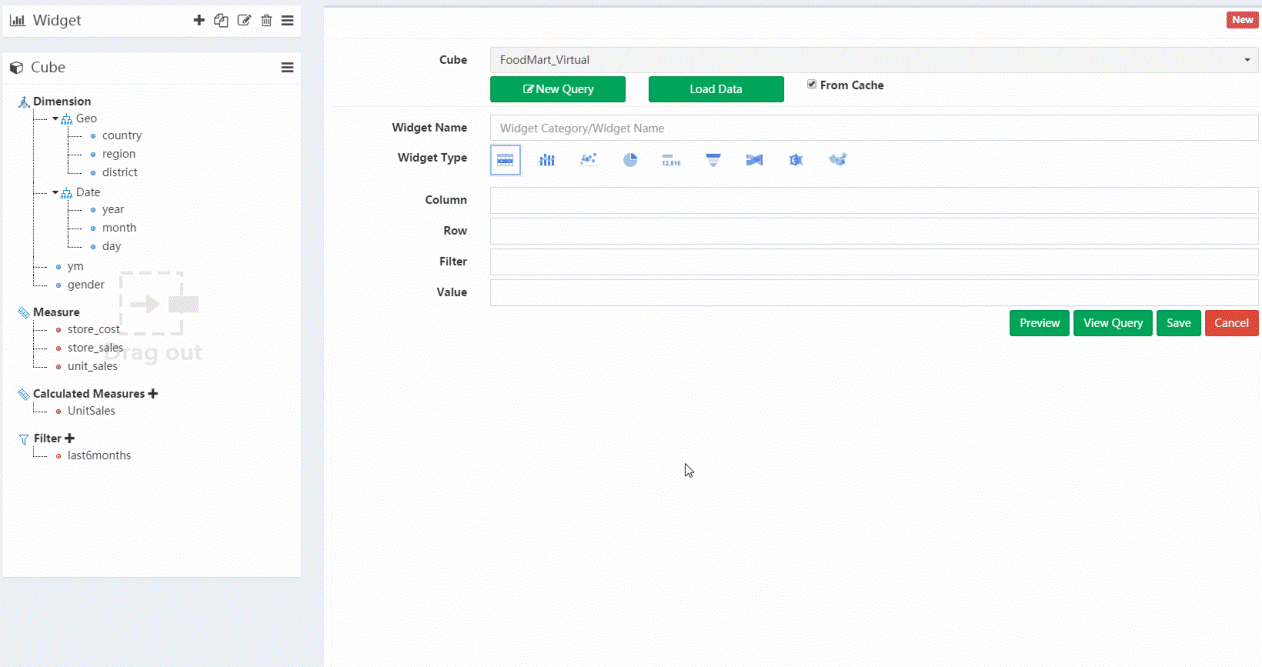
[**查询区**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/widget?id=%e6%9f%a5%e8%af%a2%e5%8c%ba)

* 查询区默认为选择已经保存的数据集(Cube)
* 下方两个功能按钮分别为切换查询方式(在数据集与新建AD-Hoc查询之间切换查询)，加载数据（JDBC离线数据源数据会加载至服务端缓存，支持数据源聚合的连接对应的查询只会加载模型）
* **【当前操作状态】**
  + **新建(New)**状态：新建状态下每次保存都会新建或者另存一个图表，方便连续设计不同粒度的不同展示形态的多张图表，当然也可以通过复制功能，复制之后再进入编辑
  + **编辑(Edit)**状态：修改保存至影响当前报表，可以重复点击保存

新建状态下如果不修改图表名称，连续点击保存会提示重名错误

[**模型区**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/widget?id=%e6%a8%a1%e5%9e%8b%e5%8c%ba)

* 模型区的层级结构(Hierarchy)需要在数据集设计页面提前定义，否则只能看到原始的可选列属性
* 模型包含4个一级分类节点
* **维度(Dimensions)**
  + 层级下面的列在支持下钻的图表组件下面可以进行下钻/上卷操作



* **度量(Measures)**
* **二次计算指标(Calulated Measures)**
  + 预先定义： 数据集预先定义只能使用，不能修改，不能查看具体内容；数据集上对指标进行修改会影响图表；数据集中删除已经在图表中使用的指标会导致图表运行失败
  + \*\*+\*\* 新建: 新建的计算指标可以修改，只在当前图表有效，其他图表中不可见
  + 只在单列上的聚合操作可以不使用计算维度
* **过滤器(Filter)**
  + 预先定义： 数据集预先定义只能使用，不能修改，不能查看具体内容

[**设计区**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/widget?id=%e8%ae%be%e8%ae%a1%e5%8c%ba)

* 图表名称格式如：[Folder]?/[SubFolder]\*/[图表名称]
* 图表类型切换：不同类型的图表对于输入有不同的限制，鼠标放置在图表类型ICON上面有详细提示，点亮状态下为理论可切换状态
* 通过拖拽至设计区输入栏进行OLAP设计

| **设计区目标框** | **模型区节点类型** |
| --- | --- |
| **行维**框、**列维**框、**过滤**框 | **维度**(Dimension)节点 |
| **指标/值**框 | 指标列节点、可选表达式节点 |
| **过滤**框 | 维度节点、过滤器节点 |

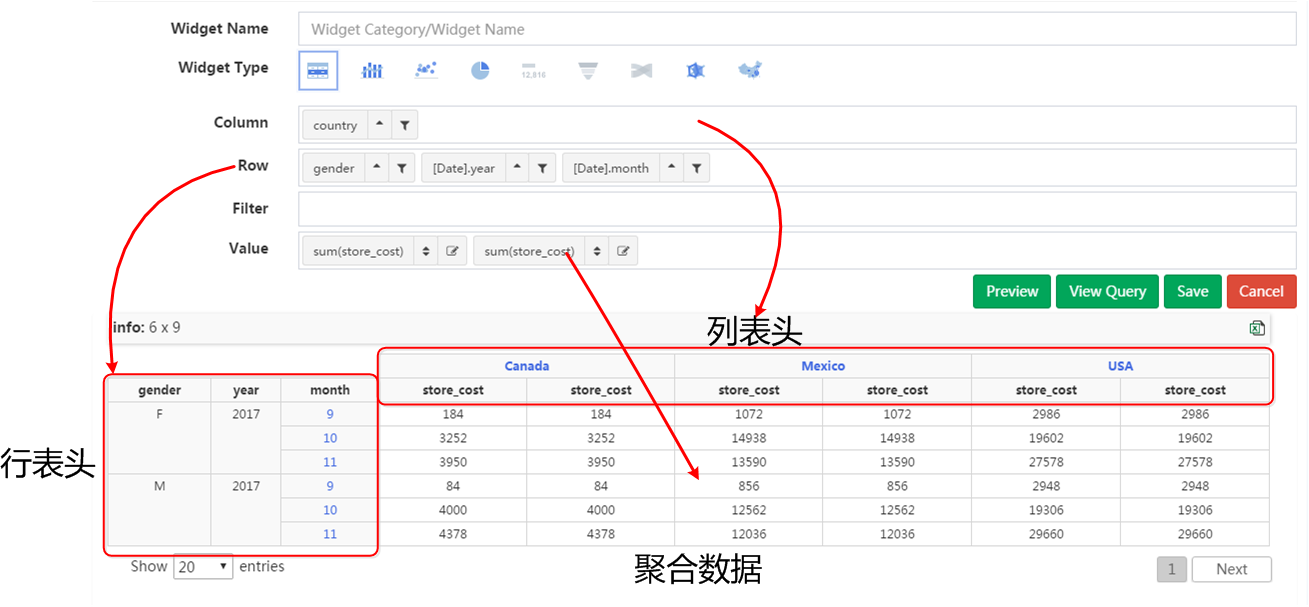
* 功能按钮
* 预览
* 预览查询：用于查看、调试数据源聚合查询下动态查询脚本
* 保存
* 取消：暂时没有功能
* **【预览区】** 预览图形

[**基本原理★**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/widget?id=%e5%9f%ba%e6%9c%ac%e5%8e%9f%e7%90%86%e2%98%85)

OLAP多维分析的数据理论上都基于多维数据，多个序列series加上相同的index，数据上与之对应最直观的展现为交叉表，交叉表原理同Excel的透视表

| **交叉表组成** | **图表设计栏** |
| --- | --- |
| 行表头 | 行维栏放置的胶囊 |
| 列表头 | **列维栏**放置的胶囊 与 **值栏**胶囊的组合 |
| 中间交叉处对应的聚合数据 | 交叉表头查询条件的数据集合+聚合函数 |

* 对应关系如下图所示，交叉表的表头合并只有在表头排序正确的情况下才能正确合并
* 开始图表设计之前请先理解此处的对应关系。后续所有图表对应的数据都和交叉表相关

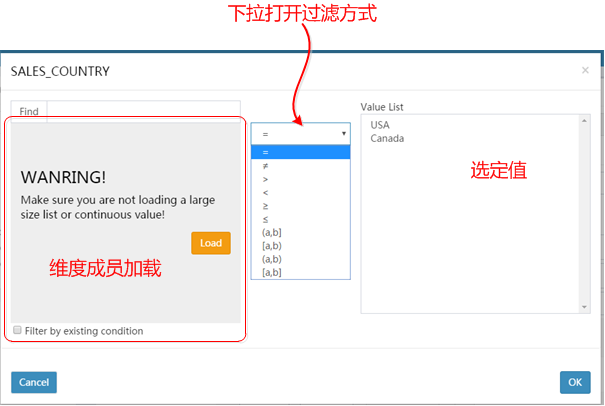


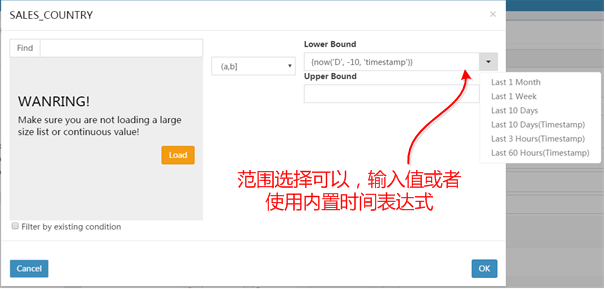
[**维度过滤/切片排序**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/widget?id=%e7%bb%b4%e5%ba%a6%e8%bf%87%e6%bb%a4%e5%88%87%e7%89%87%e6%8e%92%e5%ba%8f)

* 放置在**行维**与**列维**上面的**胶囊**点击编辑按钮可以进行数据过滤
* 也可以放置**维度**在**过滤**栏单纯对数据进行过滤

**注意:**

为了防止某个维度下面成员数量过大，或者误把连续值如时间戳之类的维度不经处理直接放置在维度框，引起服务端数据源查询压力过大，所以再加载维度成员之前加入了确认操



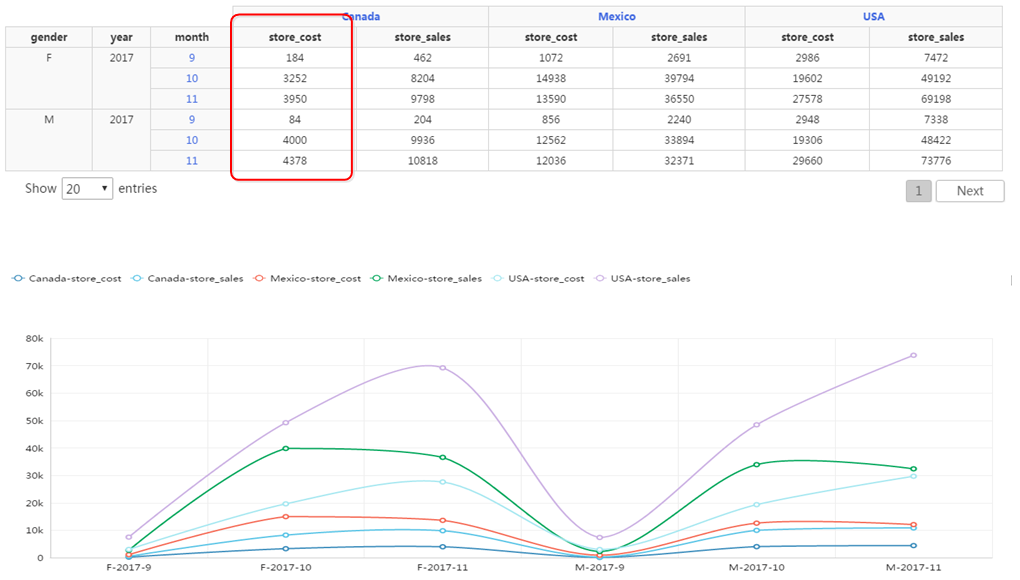


* 过滤比较只支持字符串和数值型比较，由于不同数据库的时间函数处理不一样，但是大多数数据库的时间都能与一个标准的日期字符串进行比较，可能某些数据库会有由于数据类型转换引起的索引失效，这里没有特殊处理，请数据准备的时候在这块进行一些预处理，时刻使用预览查询进行调试
* 值上的列也可以排序和过滤（0.3之后）
* 支持输入值域范围比较
* TOP N展示
* 从交叉表的形式可以看出，在值排序与行表头排序是冲突的，**行排序、各值排序有且仅有一个生效**

[**基础图形**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/widget?id=%e5%9f%ba%e7%a1%80%e5%9b%be%e5%bd%a2)

[**柱线图**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/widget?id=%e6%9f%b1%e7%ba%bf%e5%9b%be)

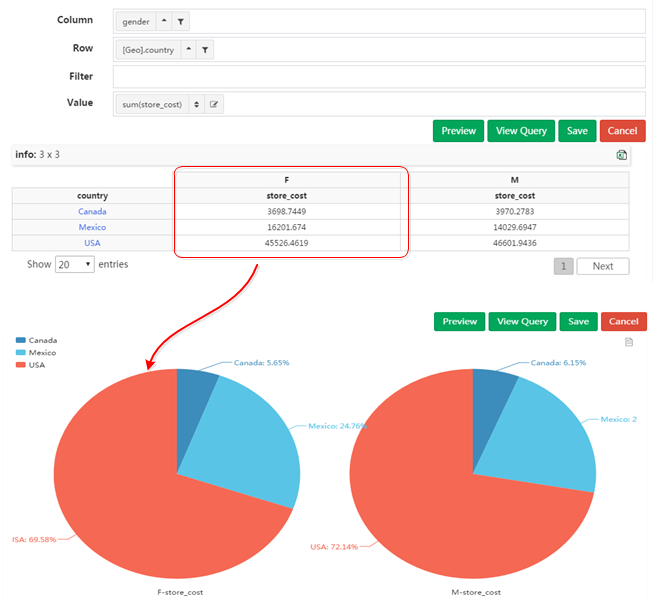
每列显示显示为一条线，或者一个柱形图序列。



| **设计区** | **图表** | **要求** |
| --- | --- | --- |
| 行维 | X轴 | 放置一个或者多个维度节点 |
| 列维 | 分类 | 放置0个或者多个维度节点 |
| 指标 |  | 放置1个或者多个指标节点(**请不要在没有放置指标的时候反复的问我们为什么没有图形出来**) |
| 添加轴 | 显示双轴 | 建议为不同的轴配置成不同的图形类别，如：柱线 |
| 垂直/水平 | X/Y轴位置互换 |  |

[**饼图**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/widget?id=%e9%a5%bc%e5%9b%be)

每列显示一个饼图

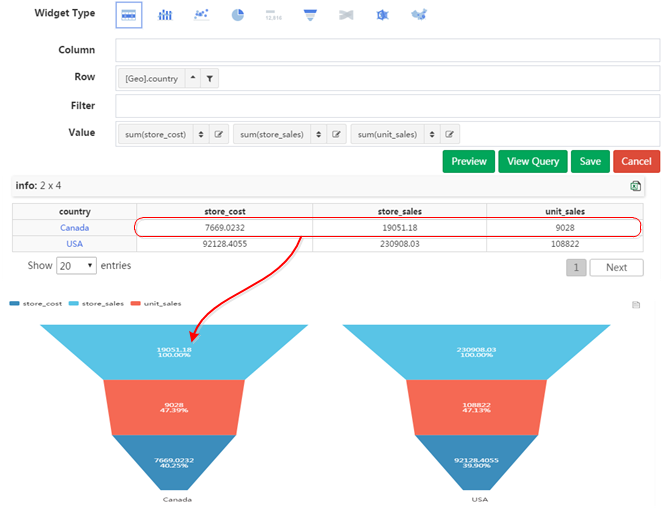


[**KPI**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/widget?id=kpi)

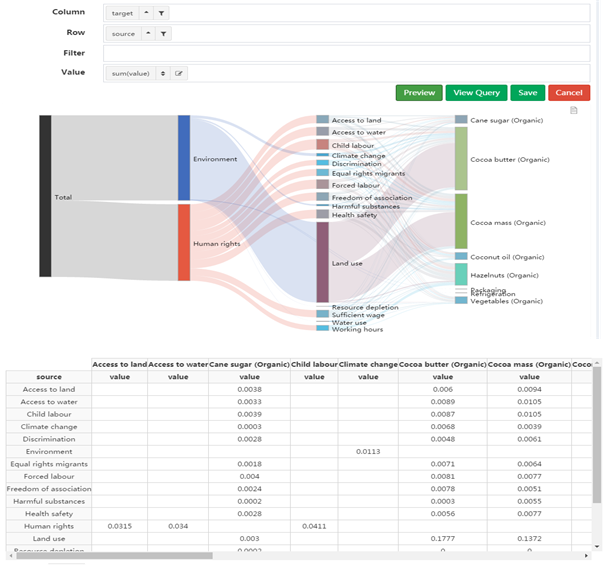
KPI输入没有维度信息，只有**一个度量值**，可选择颜色和数字格式化，格式化Formtter参考[**numbro做format**](http://numbrojs.com/format.html)

[**漏斗图**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/widget?id=%e6%bc%8f%e6%96%97%e5%9b%be)

一般情况下一个漏斗需要显示如 \_展示->点击->提交->付款, 一连串不同度量的数值， 在交叉表里面为值轴放置的多个列 行里面的值会按大小自动排序之后形成漏斗 **所以漏斗图一行一个漏斗**， 下面的Demo没有实际意义，仅仅作为演示说明



[**桑基图**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/widget?id=%e6%a1%91%e5%9f%ba%e5%9b%be)



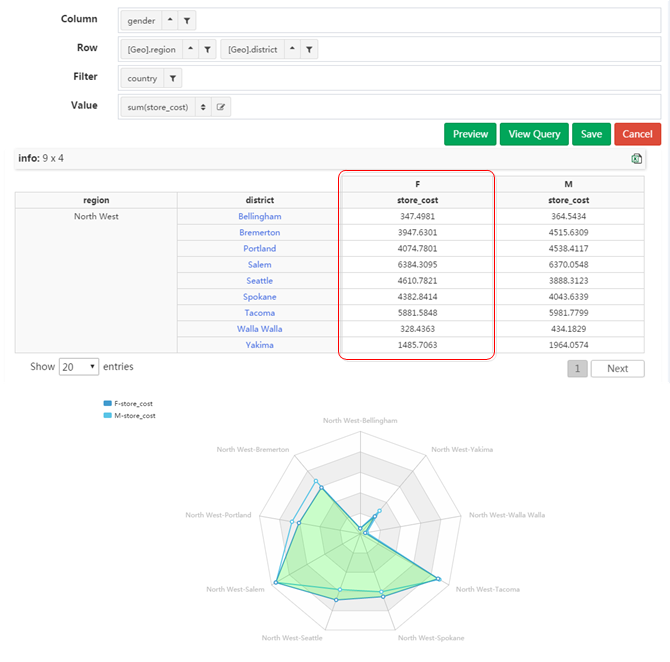
以行值和列值为节点，单元格为**行到列**的连接进行画图，交叉表可以视为一个连接矩阵

**为什么我的桑基图没有层级？**

有很多人问为什么自己的桑吉图没有层级，其实桑基图的层级和你的数据本身有关.

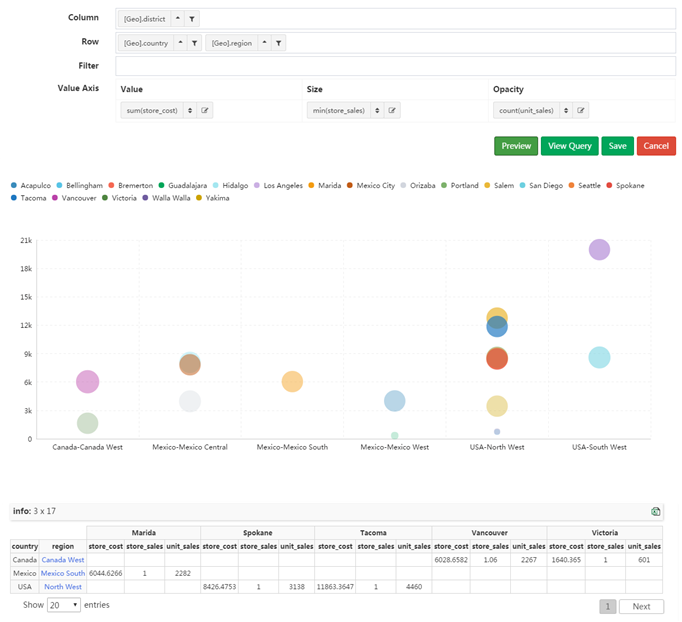
数据里面有 A -> B 和 B -> C， B为中间层，就会自动适配出两层 另外注意一点EChart对数据要求，不能成环 (A->B..->A)

[**雷达图**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/widget?id=%e9%9b%b7%e8%be%be%e5%9b%be)

一列数据在雷达上绘制一个网 

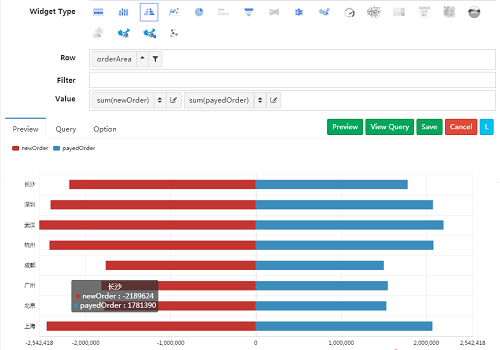
[**气泡图**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/widget?id=%e6%b0%94%e6%b3%a1%e5%9b%be)

| **设计区** | **图表** | **要求** |
| --- | --- | --- |
| 行维 | X轴 | 放置一个或者多个维度节点 |
| 列维 | 分类 | 放置0个或者多个维度节点 |
| 值 | Y轴、气泡大小、颜色深度 | 每个指标节点对应一个属性 |



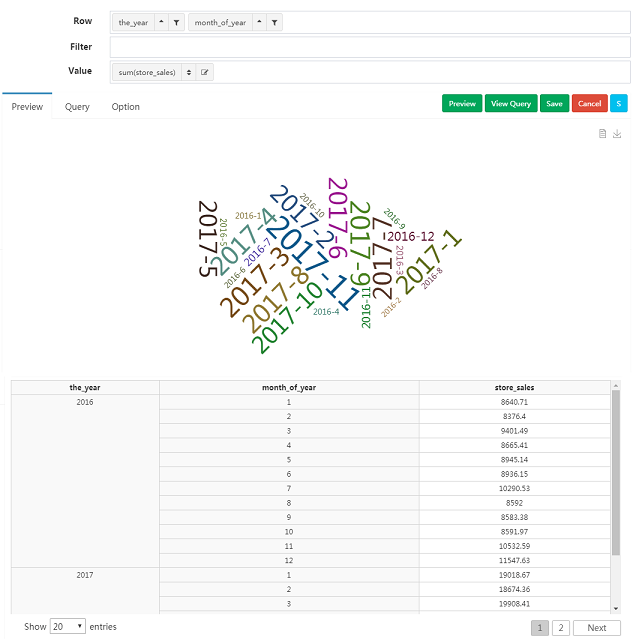
[**对比图**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/widget?id=%e5%af%b9%e6%af%94%e5%9b%be)

| **设计区** | **图表** | **要求** |
| --- | --- | --- |
| 行维 | Y轴 | 只能放置一个维度节点 |
| 指标 | X轴左右两边 | 只能放置两个指标节点 |



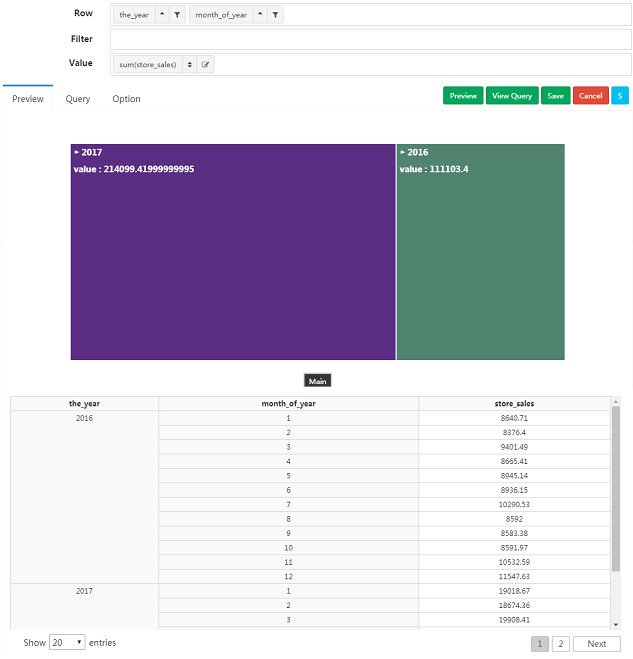
[**标签云**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/widget?id=%e6%a0%87%e7%ad%be%e4%ba%91)

只有能一个或多个行维节点



[**矩形树图**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/widget?id=%e7%9f%a9%e5%bd%a2%e6%a0%91%e5%9b%be)

| **设计区** | **图表** | **要求** |
| --- | --- | --- |
| 行维 | 多个行维代表多层，用颜色来区分类目 | 放置一个或者多个维度节点 |
| 指标 | 用面积来表示数值 | 放置1个指标节点 |



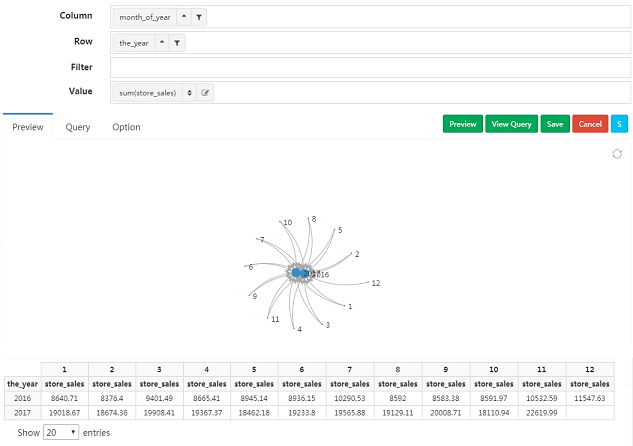
[**热点图**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/widget?id=%e7%83%ad%e7%82%b9%e5%9b%be)

| **设计区** | **图表** | **要求** |
| --- | --- | --- |
| 行维 | x轴 | 放置1个或者多个维度节点 |
| 列维 | 分类 | 放置0个或者多个维度节点 |
| 指标 |  | 放置1个指标节点 |



[**关系图**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/widget?id=%e5%85%b3%e7%b3%bb%e5%9b%be)

| **设计区** | **图表** | **要求** |
| --- | --- | --- |
| 行维 | 中心点集 | 放置1个或者2个维度节点 |
| 列维 | 分类 | 放置1个或者2个维度节点 |
| 指标 |  | 放置1个指标节点 |

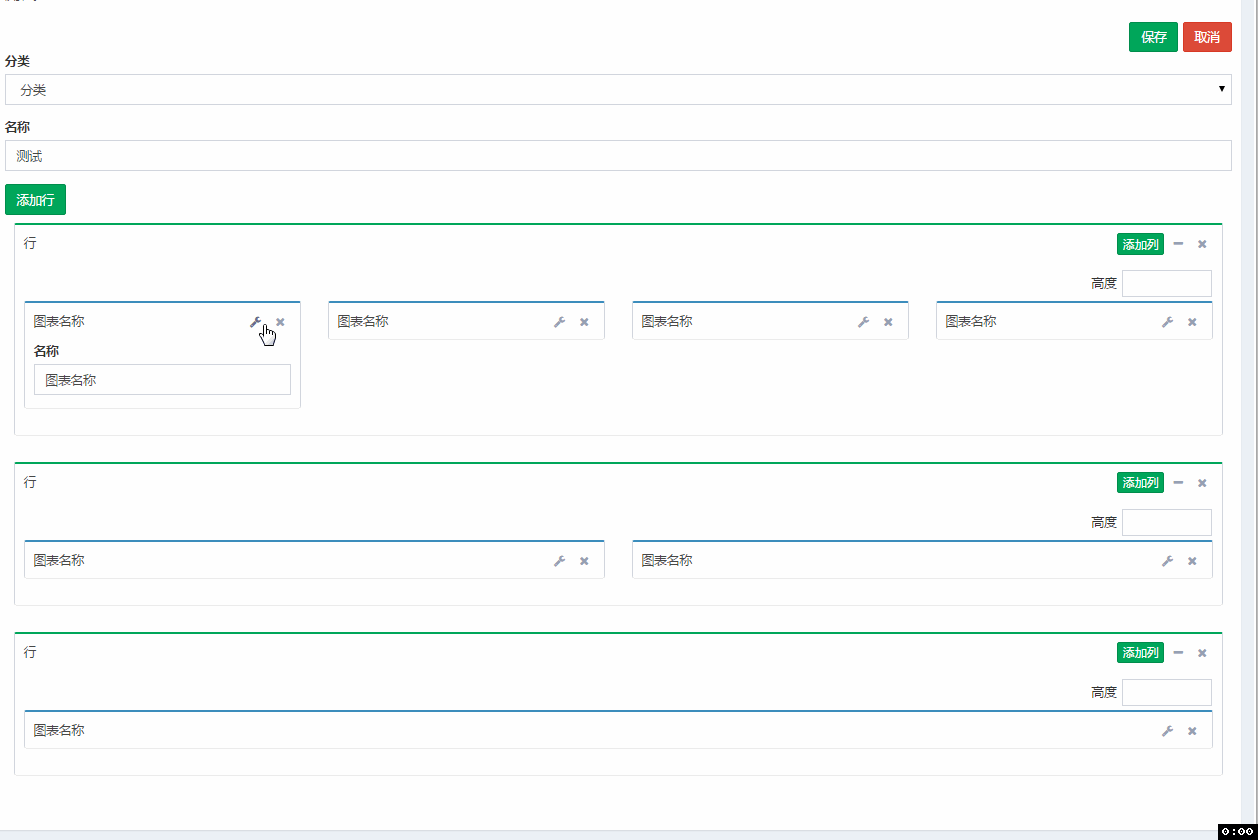


**看板设计**

**我的看板**为当前用户创建的看板，普通看板分类通过**分类管理**维护，保存看板时选择对应分类, 看板分类的添加需要有管理权限，建议超级管理员统一维护

[**网格布局**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/dashboard?id=%e7%bd%91%e6%a0%bc%e5%b8%83%e5%b1%80)

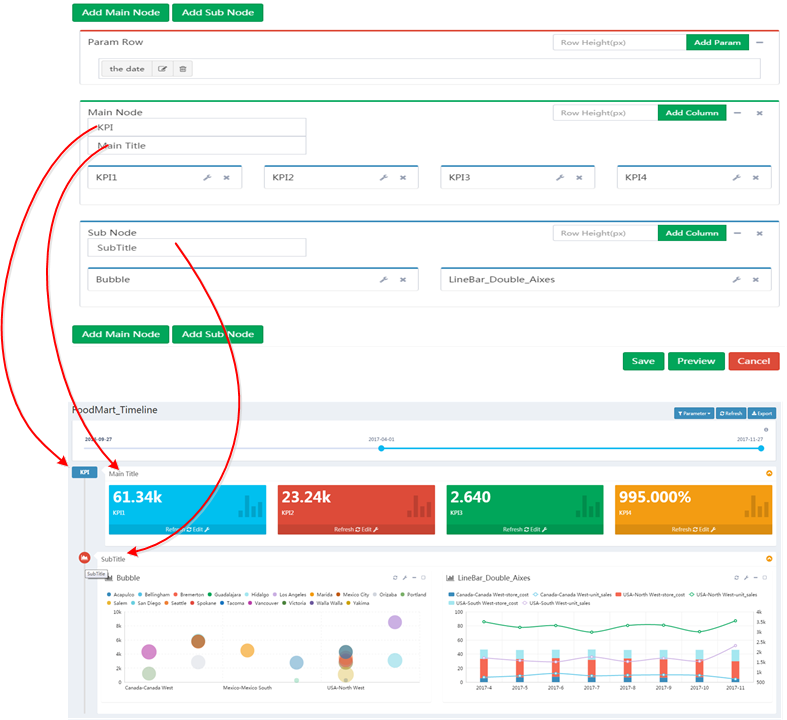
* 网格看板设计采用简单Row+Column布局模式，每行总长度为12，每列对应一个图表，行高度可以调节单位为像素，列高度继承行高；
* 交叉表默认行高要比普通图表高一些，使用时建议把表格单独放置一行，如果表格里面数据行过少，再自己调整合适的行高
* 演示动画和最新版本会有一些微小的差别



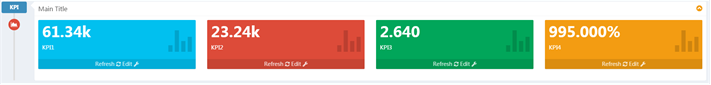
[**时间轴布局**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/dashboard?id=%e6%97%b6%e9%97%b4%e8%bd%b4%e5%b8%83%e5%b1%80)

* 时间轴布局适合总分结构、有强业务先后步骤的看板设计
* 看板结构组成
* **看板参数行（Param Row）**：有且仅有一行，而且只能在看板最顶端，参数栏为空的时候展示页面不会显示参数行
* **主节点（Main Node）**
* 一个看板下面可以有多个主节点
* 主节点可以有节点名称，在左侧时间线上可以看到，节点名称尽可能精简，太长会导致显示溢出
* 主节点对应看板：需要填写主**标题（Main Title）**
* **子节点（Sub Node）**
* 一个主节点下面可以有0到多个子节点，每个子节点都可以视为一个独立的看板
* 子节点收起状态数据不会加载

[**节点展开状态**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/dashboard?id=%e8%8a%82%e7%82%b9%e5%b1%95%e5%bc%80%e7%8a%b6%e6%80%81)



[**节点收起状态**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/dashboard?id=%e8%8a%82%e7%82%b9%e6%94%b6%e8%b5%b7%e7%8a%b6%e6%80%81)



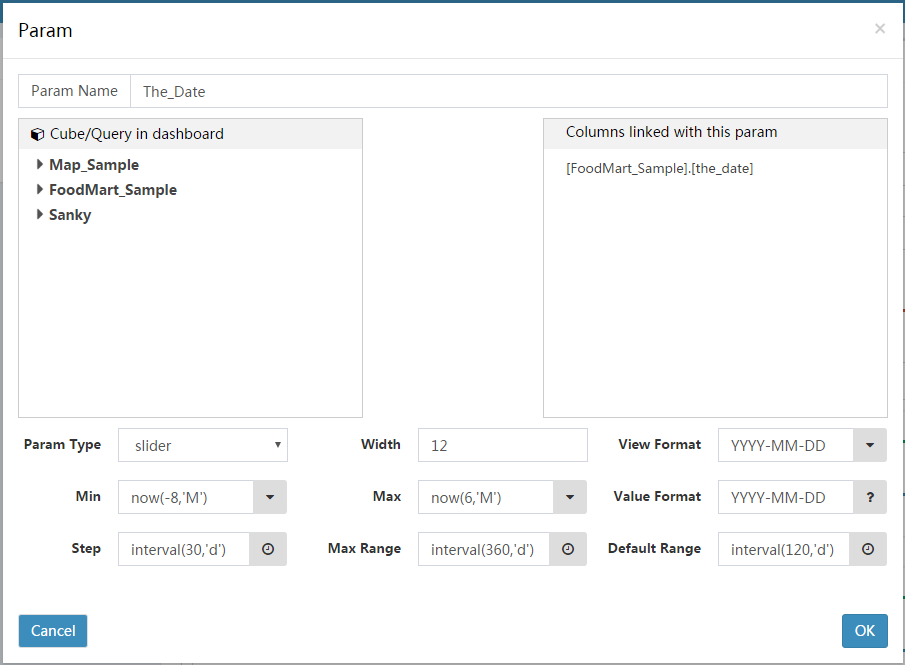
[**看板参数**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/dashboard?id=%e7%9c%8b%e6%9d%bf%e5%8f%82%e6%95%b0)

v0.2新增面板参数功能，利用面板参数可以在Dashboard展示页面支持用户数据交互；面板参数实质是关联在当前看板包含图表的数据集或者查询的列之上

v0.3.2新增滑动条交互的参数

* 我们在使用Elasticsearch展示实时数据时，发现需要使用到时间戳作为查询参数，而时间戳是连续变量，而且值内容相对可以预判，没有必要从数据源查询获取，而且select的交互方式不太直观，这促使我们加入了滑动条参数交互
* 滑动条的序列值是自动生成的**数字**，数字能够用作时间戳格式化为日期字符串
* 我们知道自动生成一个序列需要知道序列的
* **最小值(Min)**、**最大值(Max)**：内置时间戳变量*now(N,'M/d/h')*，当前时间偏移N个时间单位(M月/d天/h小时)对应的时间戳
* **步长(Step)**：内置步长时间戳变量*interval(N,'d/h/m/s')*，d天/h小时/m分钟/h小时
* 滑动调要设置区间(interval)
* **初始化范围（Default Range）**
* **最大允许范围（Max Range）**
* 显示在页面上可以控制滑动条
* **宽度(Width最大12)**、为了让时间戳有更好的可读性可以指定**显示格式化(View Format)**
* 时间戳交互是不用设置**值格式化(Value Format)**；用于日期类型交互时需要设置值格式化

[**下拉ParamType(交互类型)选择Slider**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/dashboard?id=%e4%b8%8b%e6%8b%89paramtype%e4%ba%a4%e4%ba%92%e7%b1%bb%e5%9e%8b%e9%80%89%e6%8b%a9slider)

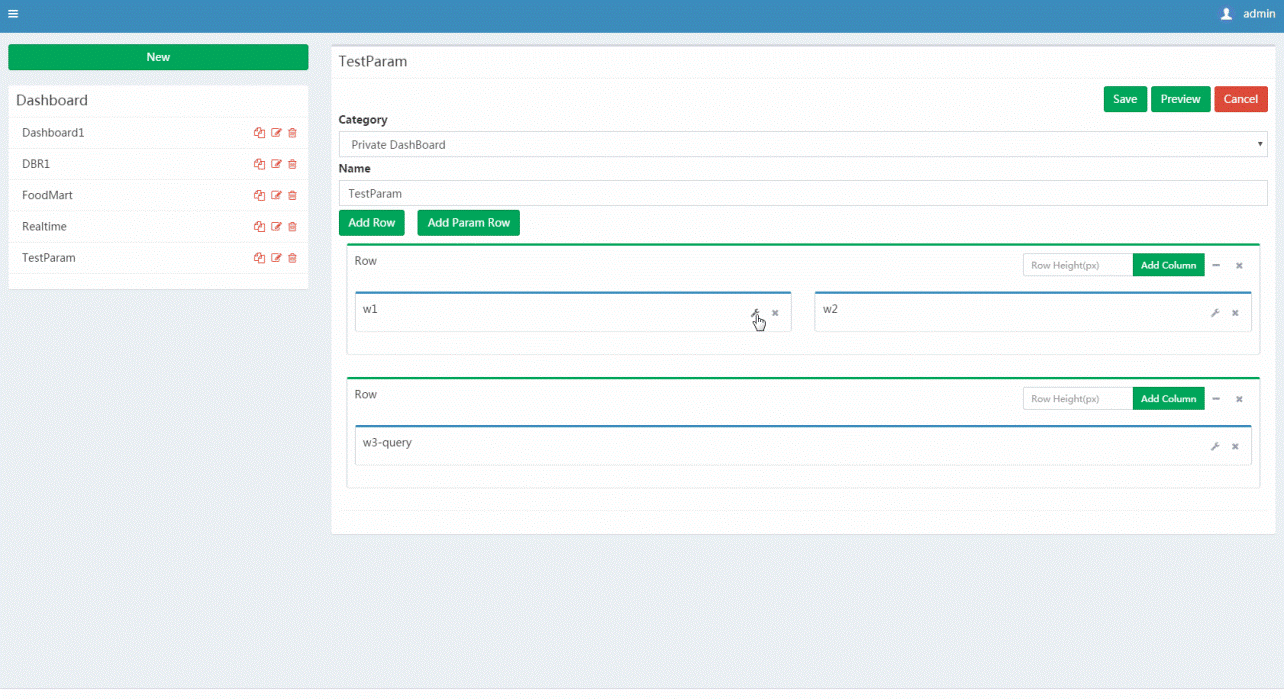


[**利用面板参数实现不同数据集之前的联动**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/dashboard?id=%e5%88%a9%e7%94%a8%e9%9d%a2%e6%9d%bf%e5%8f%82%e6%95%b0%e5%ae%9e%e7%8e%b0%e4%b8%8d%e5%90%8c%e6%95%b0%e6%8d%ae%e9%9b%86%e4%b9%8b%e5%89%8d%e7%9a%84%e8%81%94%e5%8a%a8)

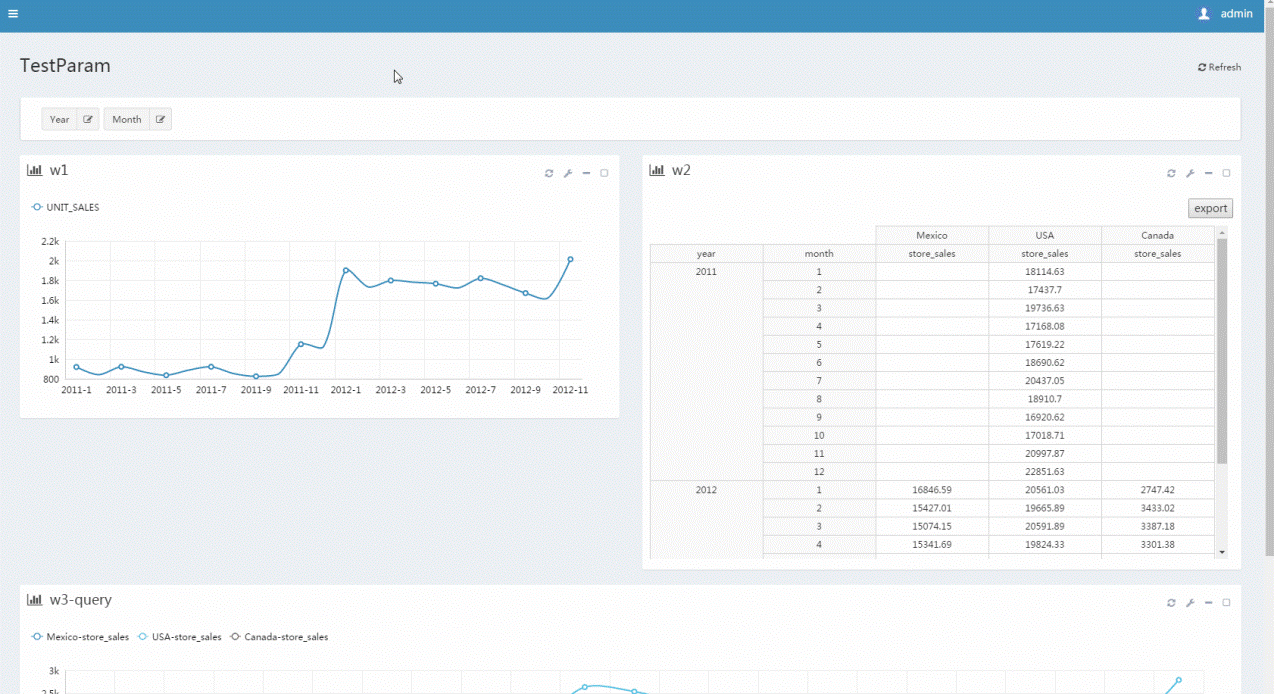
一个参数可以关联到多个Cube/查询的相同或者不同列之上，面板上选择设置参数值的操作类型(>, <, >=, <=, =, ≠, 区间范围)之后，该值会以过滤形式过滤所有关联cube的值。如：下图新建一个名称为**year**的参数，同时关联到了三个cube的不同名称的列之上。

[**添加面板参数**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/dashboard?id=%e6%b7%bb%e5%8a%a0%e9%9d%a2%e6%9d%bf%e5%8f%82%e6%95%b0)

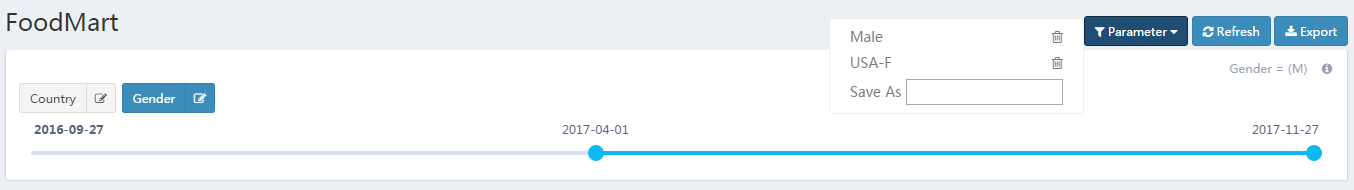
**注意面板参数只能用户维度列上面，不能用于值列之上**



[**使用面板参数**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/dashboard?id=%e4%bd%bf%e7%94%a8%e9%9d%a2%e6%9d%bf%e5%8f%82%e6%95%b0)



[**参数模板**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/dashboard?id=%e5%8f%82%e6%95%b0%e6%a8%a1%e6%9d%bf)



当看板参数过多的，或者参数成员组成比较复杂的时候，可以把当前参数保存为模板。第二次登陆，或者会话过期之后可以快速加载参数 看板参数绑定和用户绑定，每个用户可以保存自己特定的参数模板

[**看板刷新**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/dashboard?id=%e7%9c%8b%e6%9d%bf%e5%88%b7%e6%96%b0)

**看板刷新**

看板的刷新不需要特意设定, 系统会自动根据图表后台配置的对应的数据集是否有更新, 自动刷新图表  
具体配置见 [**关于数据集更新间隔**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/dataset?id=%E5%85%B3%E4%BA%8E%E6%95%B0%E6%8D%AE%E9%9B%86%E6%9B%B4%E6%96%B0%E9%97%B4%E9%9A%94)

**定时任务**

[**定时邮件发送**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/job?id=%e5%ae%9a%e6%97%b6%e9%82%ae%e4%bb%b6%e5%8f%91%e9%80%81)

[**邮件发送配置**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/job?id=%e9%82%ae%e4%bb%b6%e5%8f%91%e9%80%81%e9%85%8d%e7%bd%ae)

phantomjs\_path=D:/phantomjs-2.1.1-windows/bin/phantomjs.exe

mail.smtp.host=127.0.0.1

mail.smtp.port=8825

mail.smtp.from=test@test.com

#mail.smtp.username=test@test.com

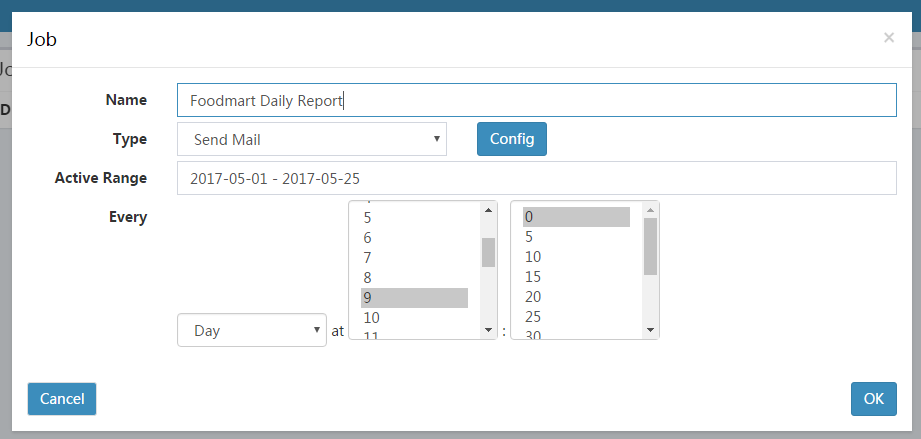
#mail.smtp.password=111111

#mail.smtp.ssl.checkserveridentity=false

* 配置邮件任务之前确保系统环境已经正确配置
* 邮件服务器配置正确，并且应用环境能够正常调用邮件发送接口
* [**phantomjs**](http://phantomjs.org/)安装配置正确: 正常情况下如果Dashboard能够正常导出表示安装配置正确

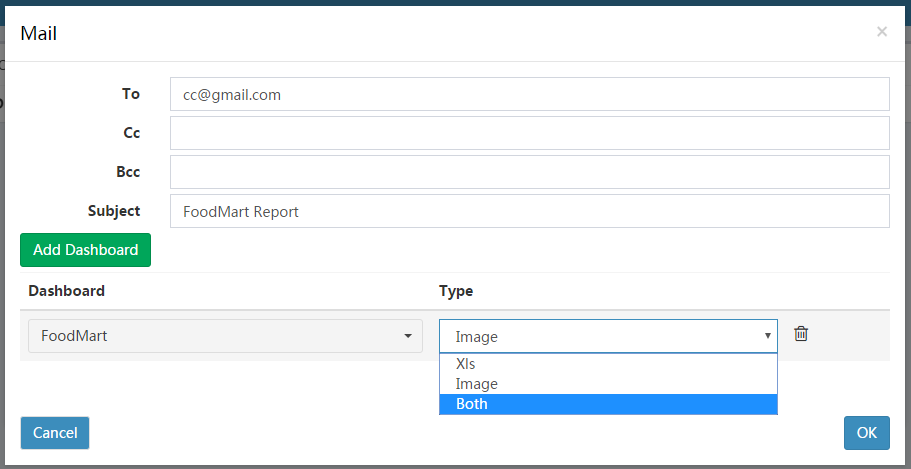
[**新建任务，填写任务表单**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/job?id=%e6%96%b0%e5%bb%ba%e4%bb%bb%e5%8a%a1%ef%bc%8c%e5%a1%ab%e5%86%99%e4%bb%bb%e5%8a%a1%e8%a1%a8%e5%8d%95)

* 任务名称
* 任务类型(邮件发送Send Mail)
* 任务有效期
* 周期



[**配置邮件内容**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/job?id=%e9%85%8d%e7%bd%ae%e9%82%ae%e4%bb%b6%e5%86%85%e5%ae%b9)

* 收件人(必填)/抄送/密送/邮件主题
* 添加看板：可以添加多个看板
* 配置看板展示形式
  + **Xls**：看板以Excel附件形式展示
  + **Image**：看板以页面截图形式在邮件正文展示
  + **Both**：两种形式共存



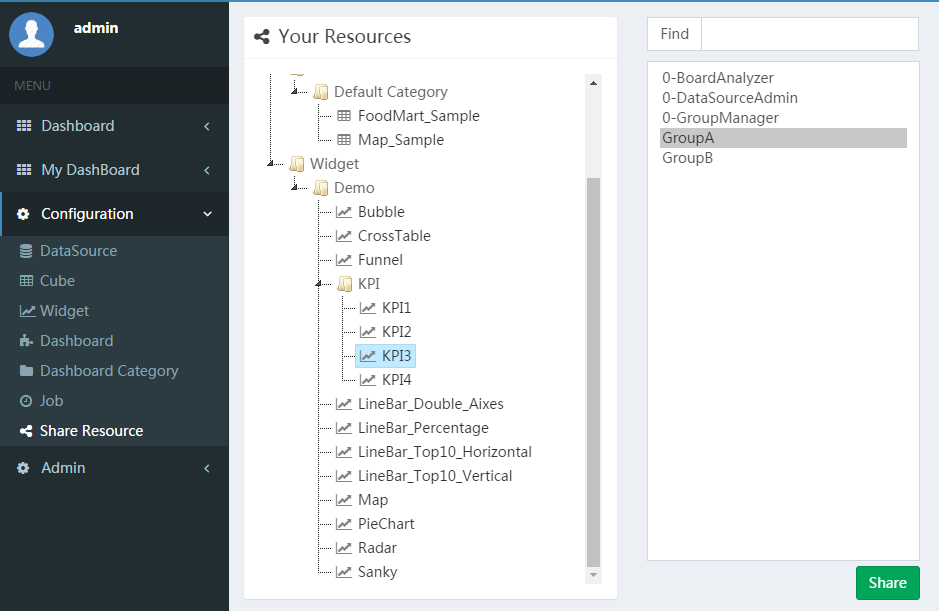
[**立即运行**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/job?id=%e7%ab%8b%e5%8d%b3%e8%bf%90%e8%a1%8c)

邮件添加完整可以立即运行，运行失败点击点击debug按钮查看出错信息

[**资源分享**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/shareResource?id=%e8%b5%84%e6%ba%90%e5%88%86%e4%ba%ab)

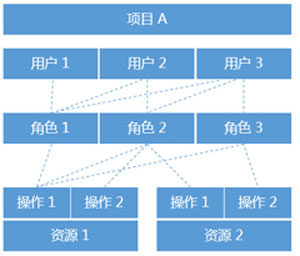
权限管理功能定位于给超级管理员与组管理员管理组用户与权限。  
对于没有用户管理页面的权限的普通用户，需要把自己新建的图表分享给其他组的时候，需要使用到资源分享功能

* 左边栏用户可以看到自己创建的所有资源(数据集、数据源、图表、看板)
* 右边栏可以看到所有组
* 选中左边资源右边用户组会联动选中
* 非增量授权，要给资源增加分享组需要，需要选中所有想要分析的用户组
* 利用非增量授权原理，可以取消授权
* CTRL+鼠标左键多选



[**权限管理**](https://peter_zhang921.gitee.io/cboard_docsify/#/zh-cn/manual/userAdmin?id=%e6%9d%83%e9%99%90%e7%ae%a1%e7%90%86)

RBAC (Role Based Access Control)基于角色的权限管理，由用户、角色、操作/资源，三部分组成，访问列表也称之为资源列表被赋予角色/用户组之上，赋予用户一个或多个组让用户获得所在组的资源访问权限



与之相对应CBoard的用户管理界面分也为三块

**用户：**

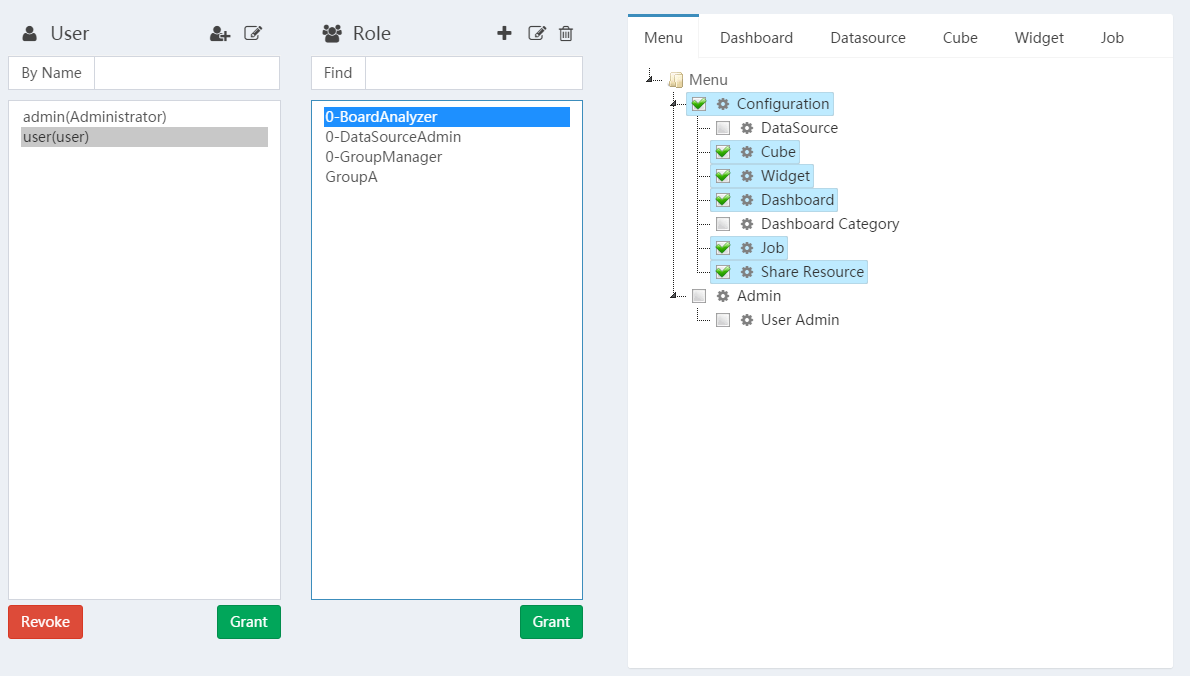
* 用户列表
* 新建、编辑用户
* 搜索用户：默认通过用户名（包含登陆名）搜索，点击搜索框前面的按名称按钮可以切换为按用户组搜索
* 用户多选框下面功能按钮分别为
* 按住Ctrl鼠标左键选中多个用户或者角色
* Grant： 选中用户与角色，对用户进行角色授权，用户授权非增量授权，每次Grant会更新用户角色为当前角色栏选中的角色列表，给用户增加授权时，按住Ctrl鼠标左键点击增加或者取消选中角色
* Revoke：选中用户与角色，把用户从角色中移除
* 资源不能授权给用户
* 选中一个用户之后，该用户对应的角色可以被联动选中

**角色(Role)：**

* 点击＋号添加角色
* 角色管理员用于管理当前角色所属
* 除了Admin之外，角色管理员只能看到自己管理的角色

**资源：**

* 菜单：一级菜单与二级菜单分开控制，没有级联(技术问题，jstree第三状态无法程序获取回选)
* 看板/数据源/数据集/图表: 由于文件夹为虚拟路径，可以级联，但是即使文件夹被选中，在新增了资源在文件夹里面之后，还是要手动添加授权
* 管理员只能对自己创建的资源添加到组



项目地址：<https://github.com/TuiQiao/CBoard>

文档地址：https://tuiqiao.github.io/CBoardDoc/#/