TotalLINK

产品手册

上海朝识智能科技有限公司

2020 年 8 月

目 录

文档控制

■ 主要内容

本文介绍 Echarts 图表与 TotalLINK 看板功能的结合使用。通过在 TotalLINK 设计模型管理 Echarts 图表与 TotalLINK 数据列表等特色模块，利用全局模型编写 HTML5+CSS3 代码设计布局，在 TotalLINK 网站看板界面可以看到富有科技感的大屏数据。

总的设计思路如下：

1. 设计模型，配置 ECHARTS 图表或其它图表
2. 看板设置，整合一整套看板所需要的图表
3. 设计一整套看板的全局参数以及布局。9 号模型存放 UI 布局代码，10 号模型存放参数
4. 在看板网站输入第二步配置的看板关键字，即可查看数据。

■ 更改记录

日期	版本	作者	备注
2020-08-24	1.0	Jozey	初始发布

■ 支持版本

非特殊说明的功能，默认前后版本都支持

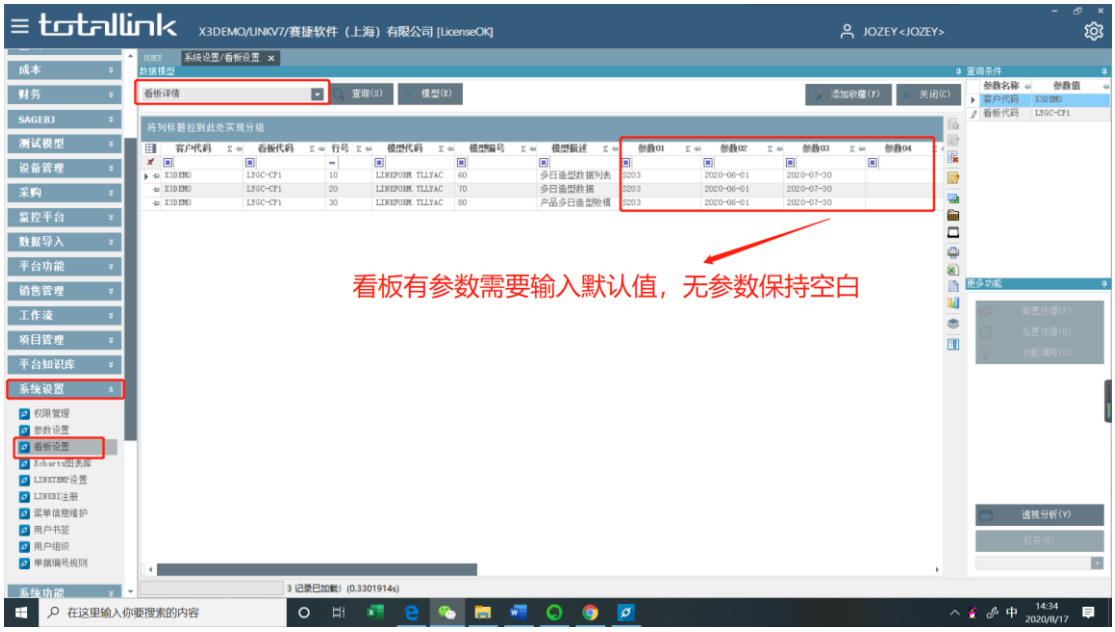
仅支持T20版本及以后版本的功能点
▪ 本文数据大屏功能仅支持 V20 及以上版本 ▪

1 看板设置

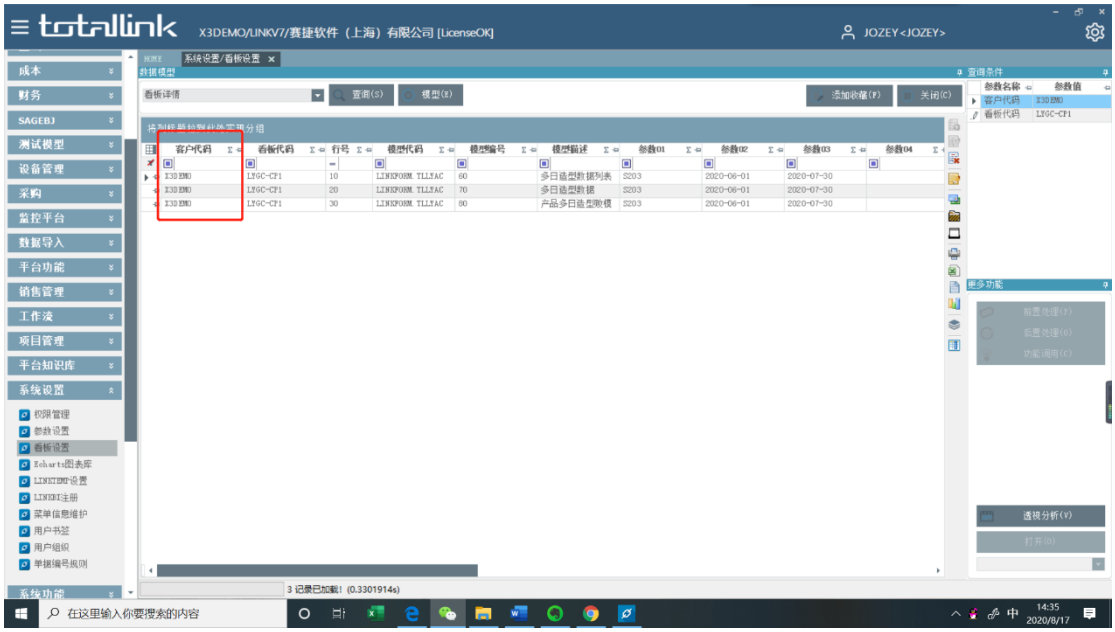
首先设计模型实现动态数据的输出，然后通过看板设置，将多个图表数据进行整合，自定义输出方式，可以得到高大上具有科技感的看板。

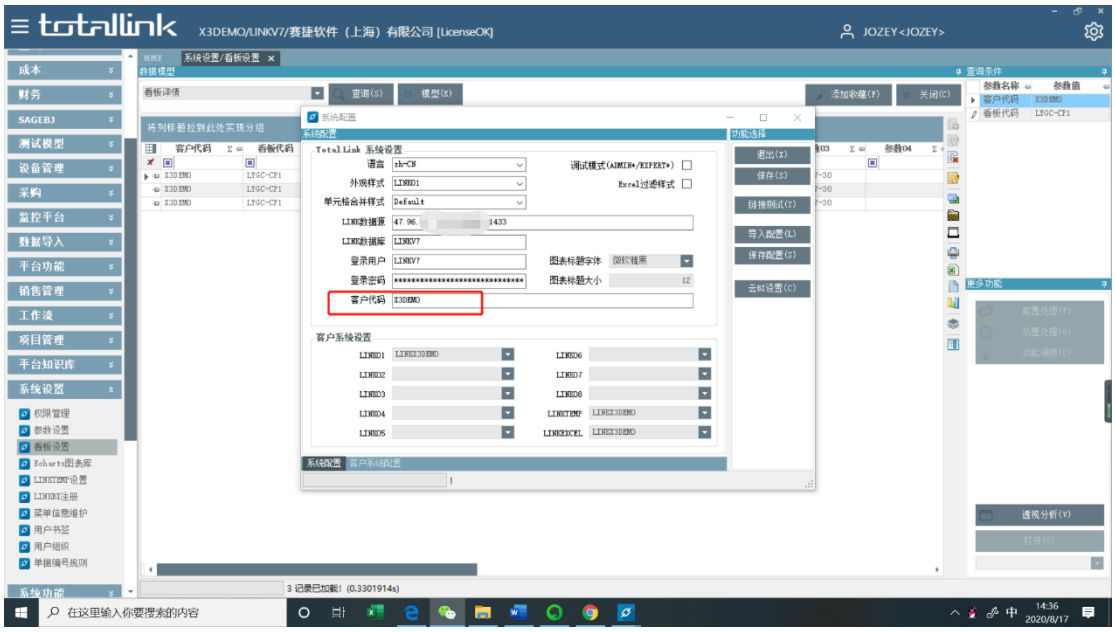
1.1 基本设置

- 先找出所要进行看板输出数据及图表的模型代码和顺序号。
- 路径：系统设置—看板设置—看板详情。

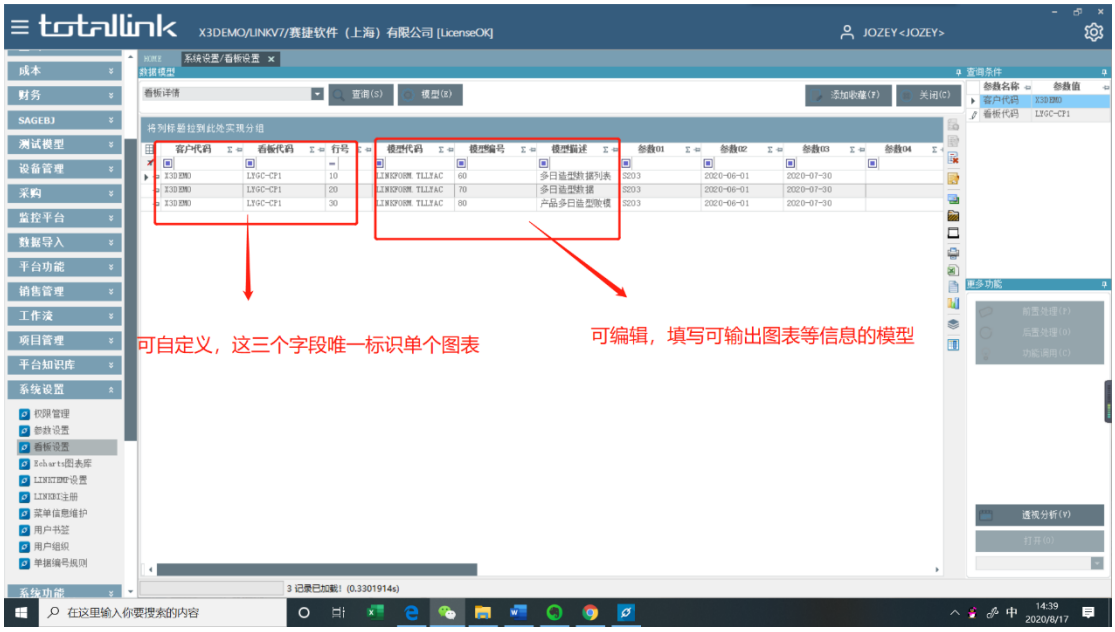


- 客户代码是指系统设置中的客户代码（在 TotalLINK 图标鼠标点击右键弹出“系统配置”对话框可看到客户代码）

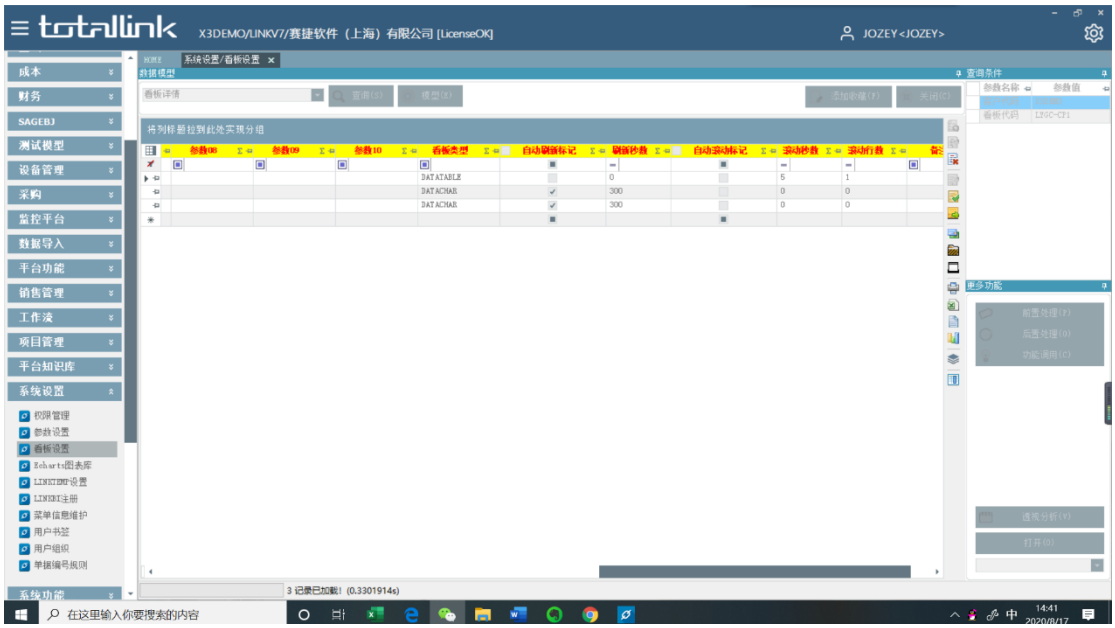




- 看板代码和行号可自定义编辑（在同一个看板上显示的模型，看板代码一致；行号不能相同，如下图所示），将要进行看板输出数据及图表的模型代码和顺序号编辑到模型中。



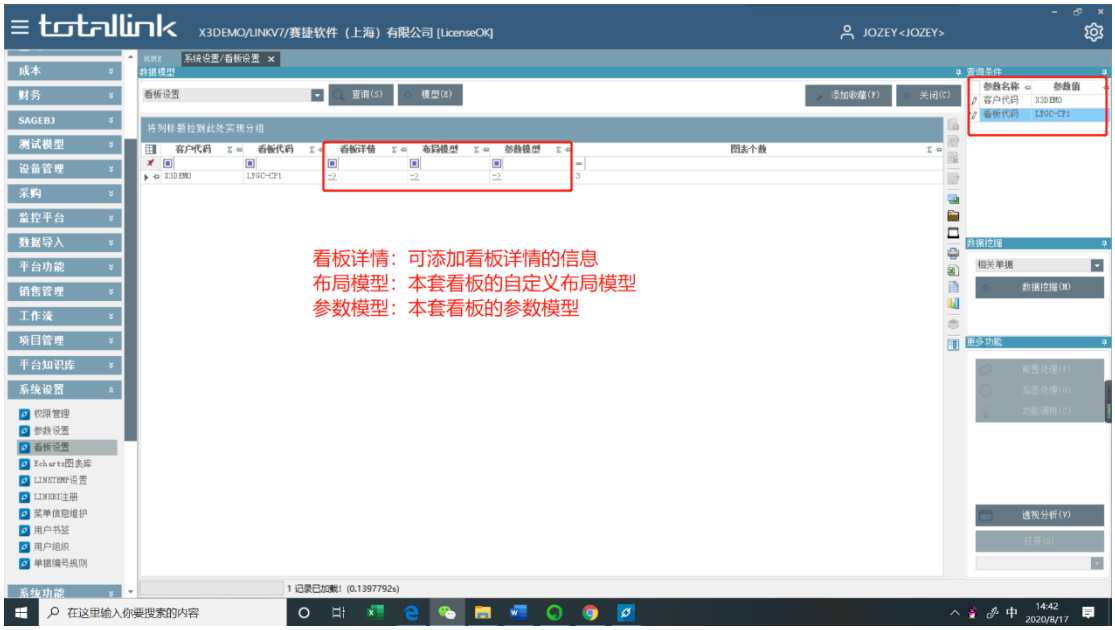
- 设置看板类型，并设置相应的自动刷新标记、刷新秒数、自动滚动标记、滚动秒数、滚动行数等，最后进行保存。



基本介绍

- 看板类型：具体见本文第二小节-看板类型介绍
- 自动刷新标记：标记是否自动刷新记录；
- 刷新秒数：自动刷新的秒数；
- 自动滚动标记：标记在自动刷新的过程中是否自动滚动进度条；
- 滚动秒数：自动刷新的过程中进度条自动滚动的秒数；
- 滚动行数：自动刷新的过程中进度条自动滚动的行数。

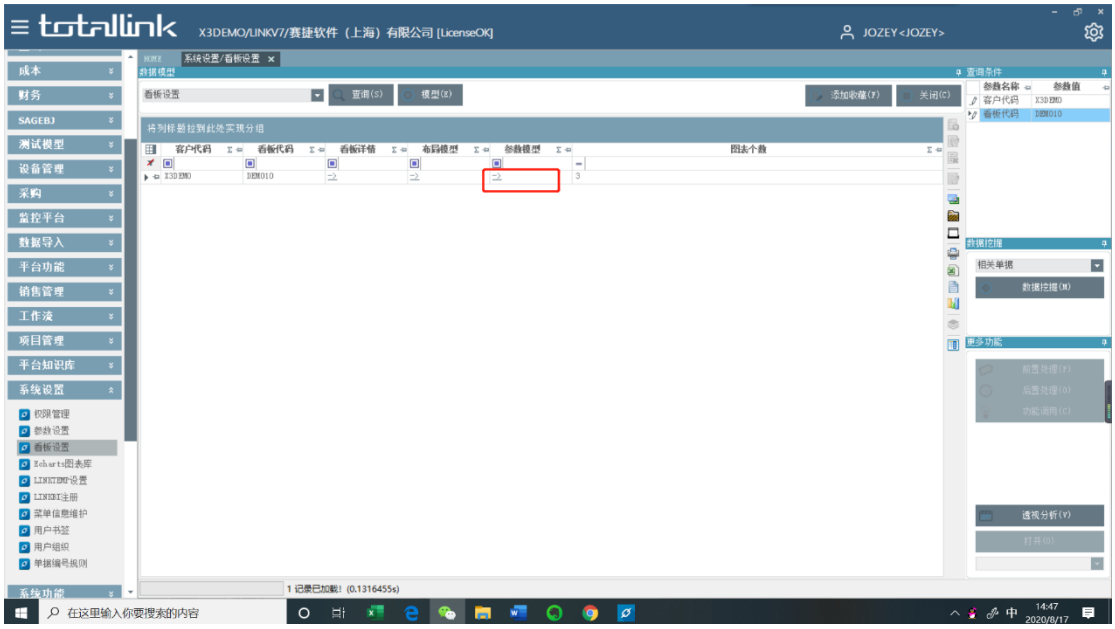
- 查询时必须输入看板参数，否则无数据输出。



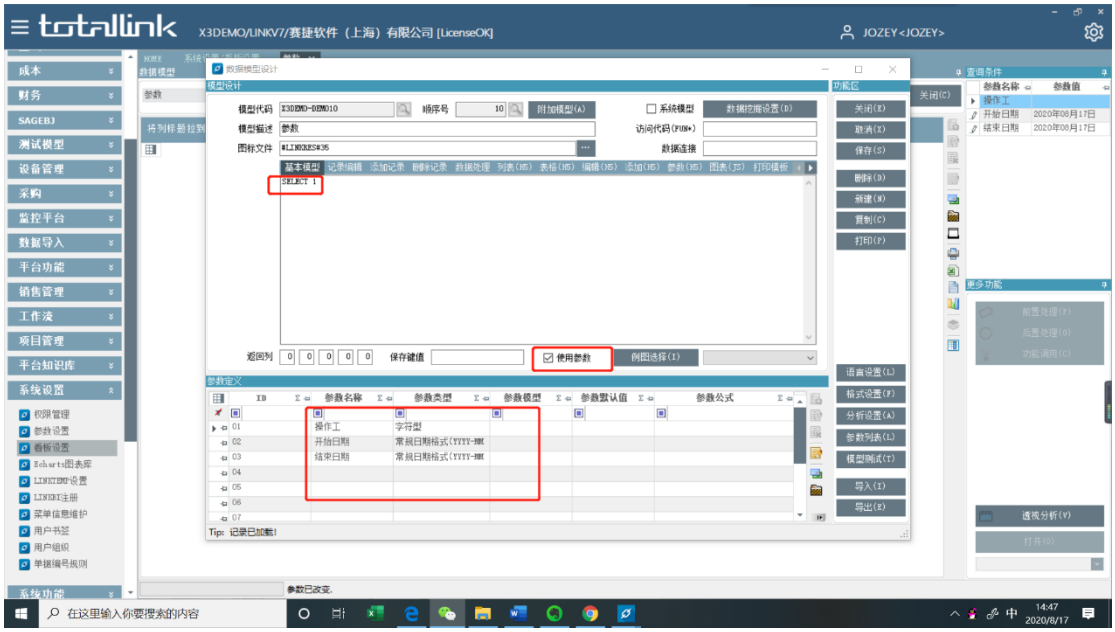
看板详情：可添加看板详情的信息
布局模型：本套看板的自定义布局模型
参数模型：本套看板的参数模型

1.2 参数布局

- 设置好看板参数之后，点击挖掘，新建一个对应的 9、10 号模型。
- 其中 9 号模型用来放编写看板布局及 UI 代码，10 号模型设置参数（无参数的看板也需要建立 10 号模型，不写任何代码也可以）
- 注意，看板布局如果不需要自定义，9 号模型不写任何代码
- 挖掘过去新建的模型代码默认为客户代码+看板代码

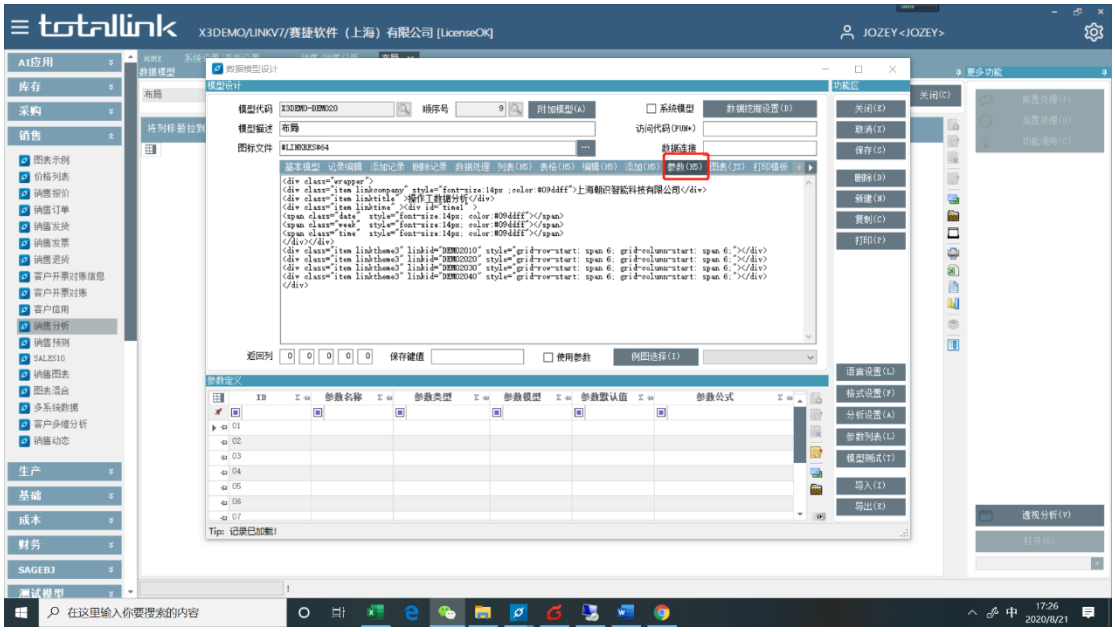


- 如下方的 10 号模型，由于看板输出需要带参数，则该模型需要新建对应的参数，并且需要写一个查询，查询仅起到初始化该模型的作用



- 如果 9 号模型不编写代码，默认是系统自定义布局，效果如下

- 如果需要自定义布局，9 号模型编写布局 UI 代码
- “参数”模块—Html+CSS 代码：用栅格布局自定义页面布局；栅格布局用法见本文第三节《栅格布局》



参考代码：

```
<div class="linkbox">
<div class="item linkcompany" style="font-size:14px ;color:#09ddff">上海朝识智能科技有限公司</div>
<div class="item linktitle" >操作工数据分析</div>
<div class="item linktime" ><div id="time1" >
<span class="date"  style="font-size:14px; color:#09ddff"></span>
<span class="week"  style="font-size:14px; color:#09ddff"></span>
<span class="time"  style="font-size:14px; color:#09ddff"></span>
</div></div>
<div class="item linktheme3" linkid="DEMO2010" style="grid-row-start: span 6; grid-column-start: span 6;"></div>
```

```
    <div class="item linktheme3" linkid="DEMO2020" style="grid-row-start: span 6; grid-column-start: span 6;"></div>
    <div class="item linktheme3" linkid="DEMO2030" style="grid-row-start: span 6; grid-column-start: span 6;"></div>
    <div class="item linktheme3" linkid="DEMO2040" style="grid-row-start: span 6; grid-column-start: span 6;"></div>
</div>
```

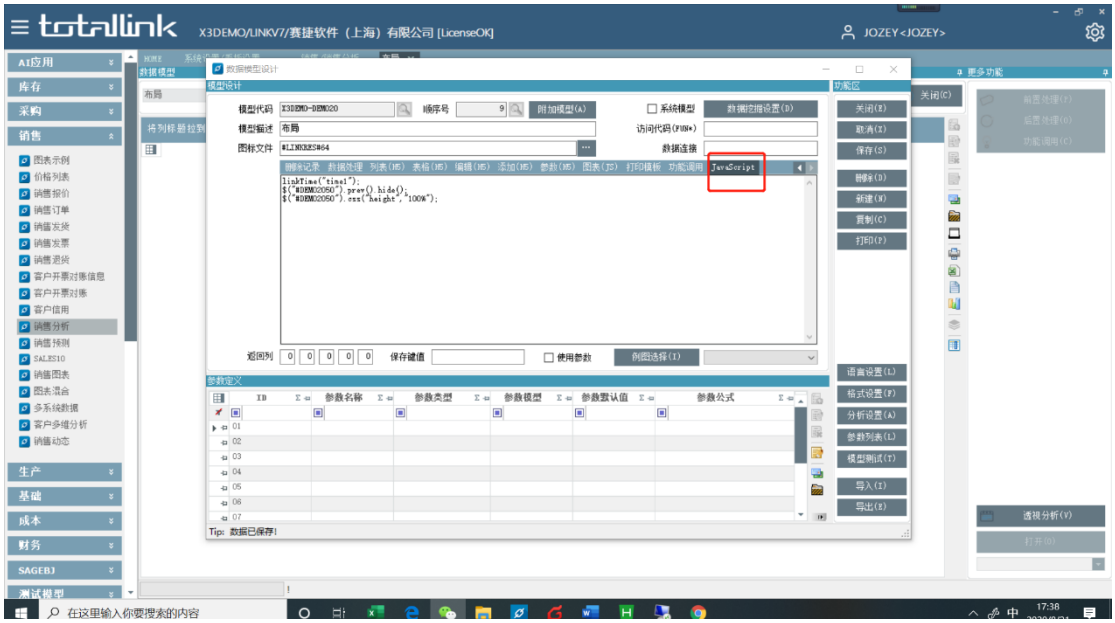
代码解析：

- class：类名，我们默认为”item+linkthemeX”。 linkthemeX 代表图表边框的主题，目前的主题有 linktheme1，linktheme2，linktheme3，linktheme4，效果如下图

- linkid: 看板代码+行号，根据看板设置的参数维护
- linkcompany\ linktitle\inktime:
  这三个类默认占 1 行，每个类占 4 列，如果只要标题，则其它两个 div 标签的内容去掉即可
  参考代码：

```
<div class="linkbox">
<div class="item linkcompany" style="font-size:14px ;color:#09ddff"> </div>
<div class="item linktitle" >操作工数据分析</div>
<div class="item linktime" ></div>
```

- Javascript 模块:写 JS



参考代码：

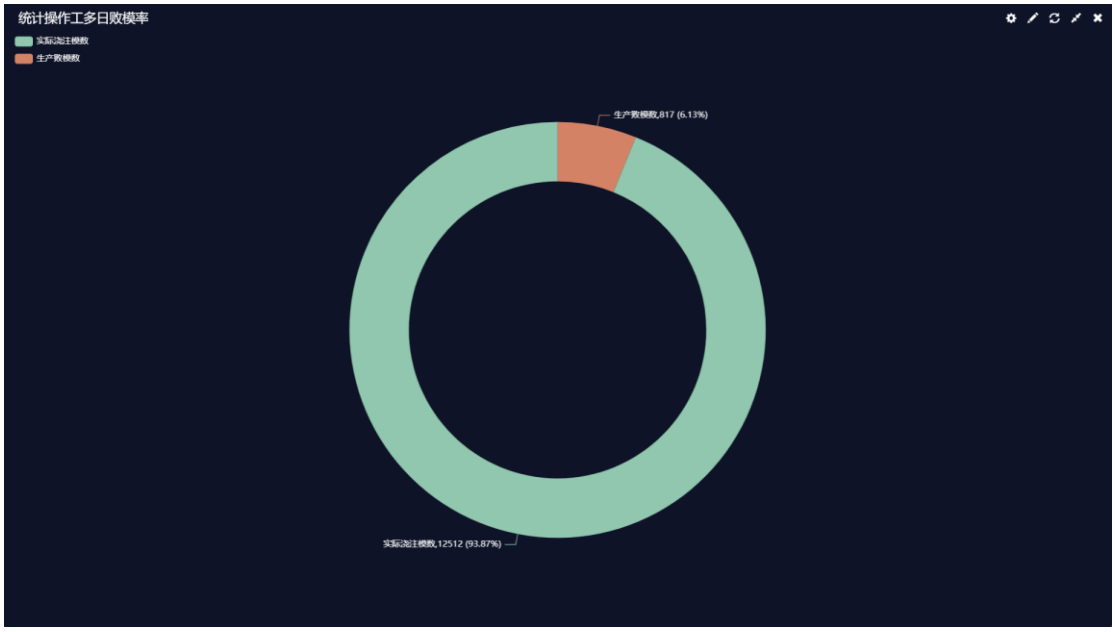
```
linkTime("time1");
$("#DEMO2050").prev().hide();
$("#DEMO2050").css("height","100%");
```

代码解析：

- 红色背景代码：JS 调用时间系统函数，作用于参数模块里 id=”time1”的标签
- 黄色背景代码：隐藏图表的标题工具栏，这里隐藏的是 linkid=”DEMO2050”的通知图表

- 自定义看板效果

- 放大单个图表效果



1.3 数据展现

- 打开看板网页，输入客户代码及看板代码，点击查询
- 无参数的看板，直接点击“查询”按钮即可看到图表
- 如果有参数并且在看板设置那设置默认的参数，点击查询可以直接看到默认的参数。输入参数后，再点击“提交参数”可以查看相应看板数据

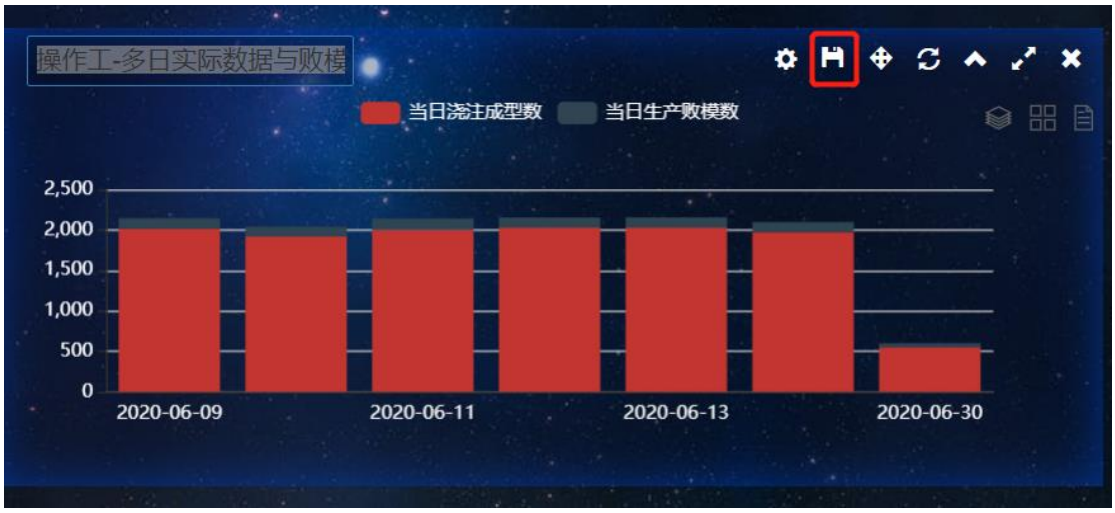
- 单个图表按钮介绍

➢ 1.设置

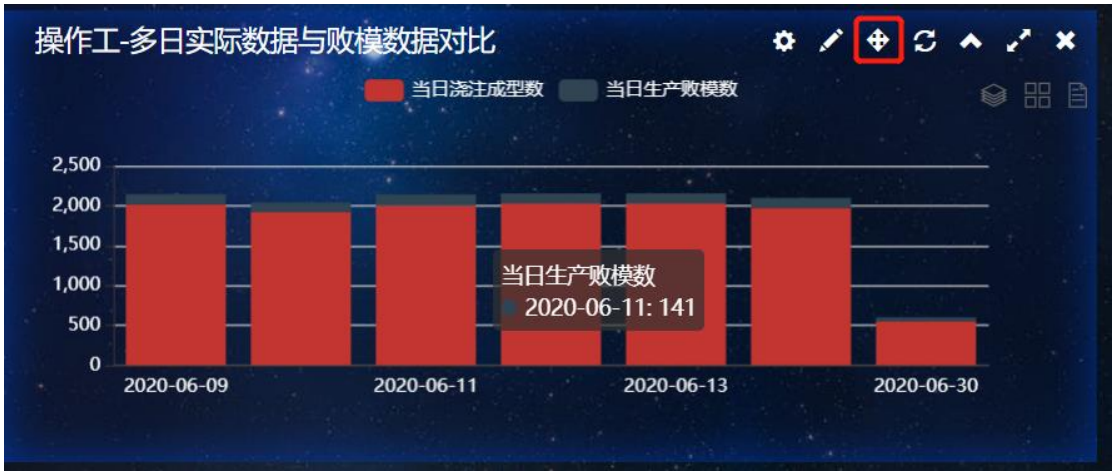
系统默认加载看板设置中的刷新、滚动参数，如果在网页这对单个图表再次设置，则以网页上设置的参数优先。当退出网页之后再进入看板，刷新、滚动参数仍以后台设置的值为主

➢ 2.编辑标题

修改界面看板的标题后保存，当退出网页之后再进入看板，标题仍以后台设置的值为主

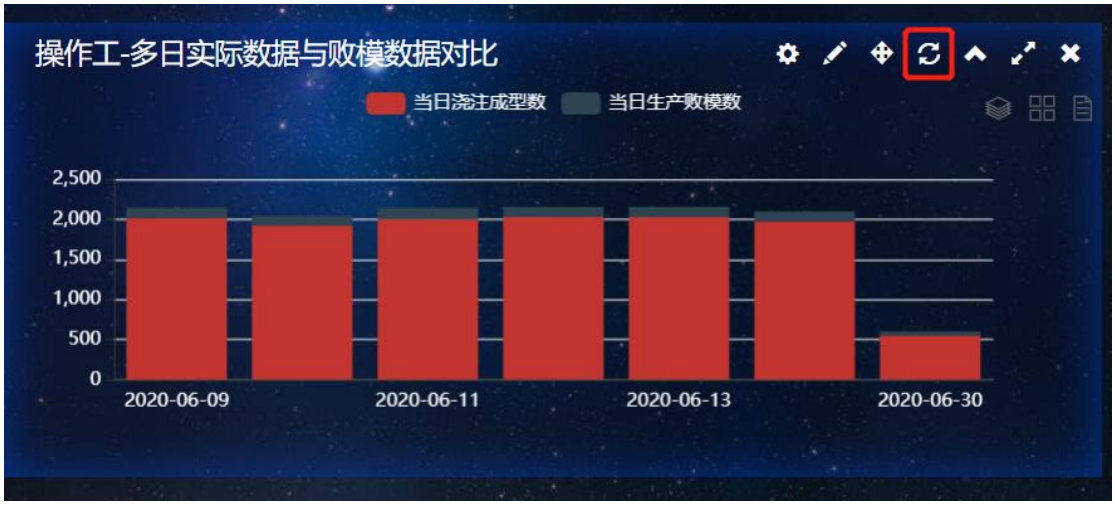


➢ 3.移动位置



➢ 4.刷新

手动刷新当前数据



➢ 5.伸缩

将报表暂时隐藏


操作工-多日实际数据与败模数据对比

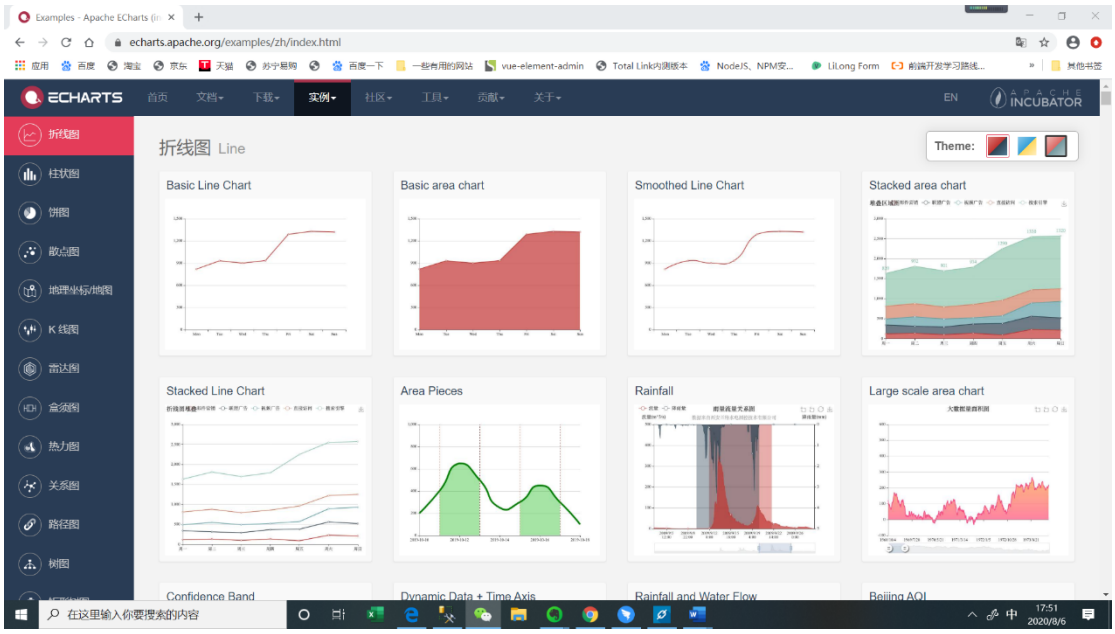
➢ 6.全屏


操作工-多日实际数据与败模数据对比
当日浇注成型数
当日生产败模数
2,500
2,000
1,500
1,000
500
0
当日浇注成型数
2020-06-11: 1,999
2020-06-09
2020-06-11
2020-06-13
2020-06-30

2 看板类型

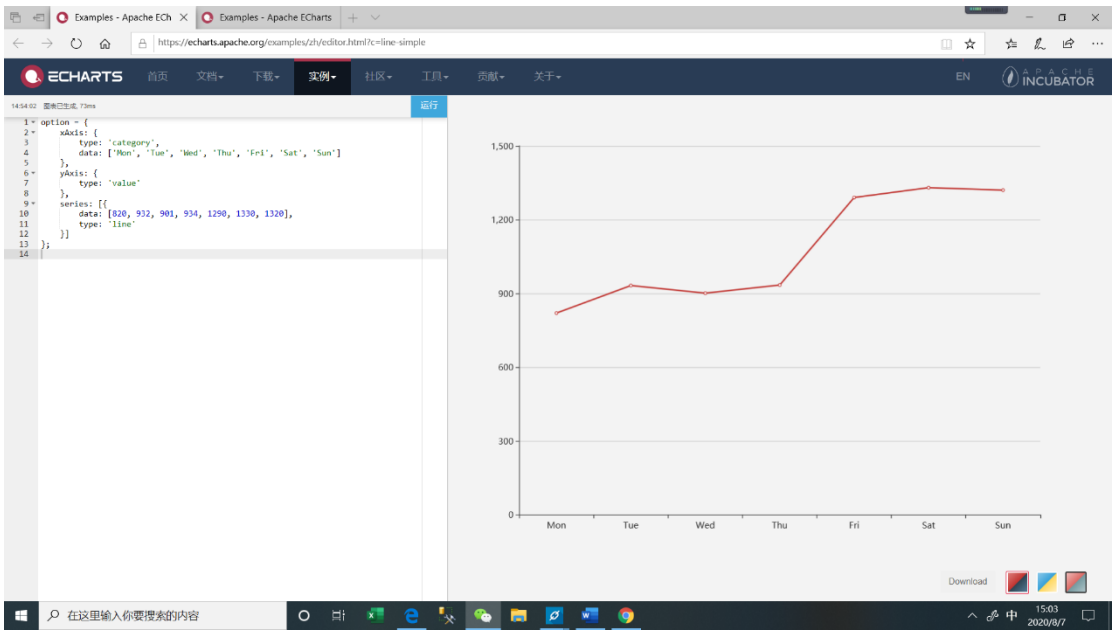
2.1 Echarts 图表—DATACHART

2.1.1 介绍



百度 Echarts 官网：https://echarts.apache.org/zh/index.html

- 在 Echarts 挑选需要展示的图表，需要调整成 dataset 的数据格式，比如一个折线图：

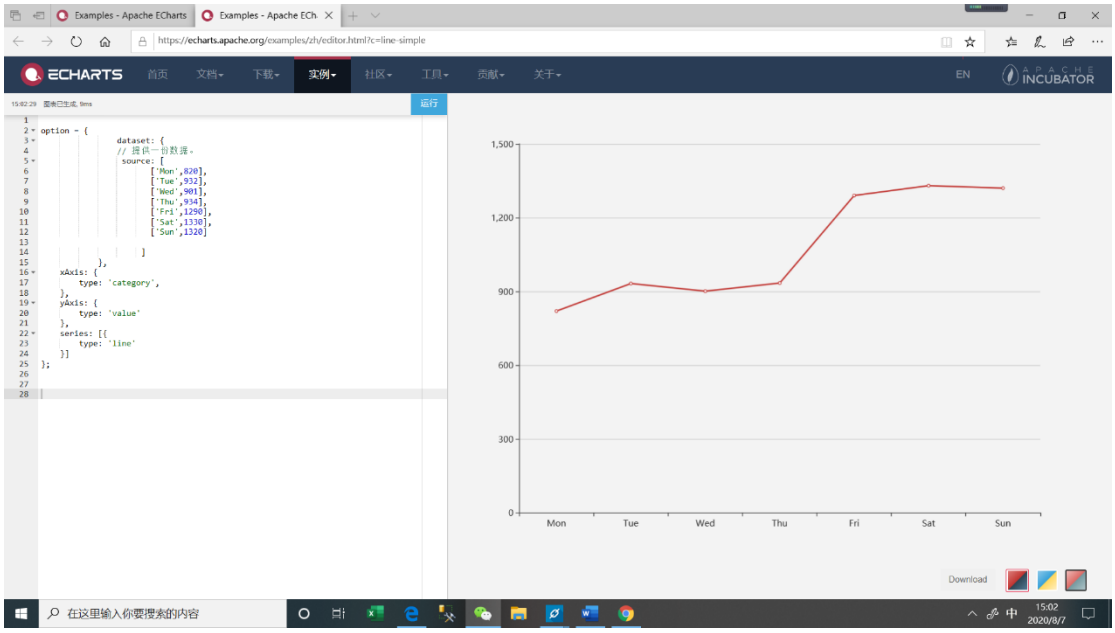


参考代码：

```
option = {
```

```
    xAxis: {
        type: 'category',
        data: ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']
    },
    yAxis: {
        type: 'value'
    },
    series: [{
        data: [820, 932, 901, 934, 1290, 1330, 1320],
        type: 'line'
    }]
};
```

- 调整之后

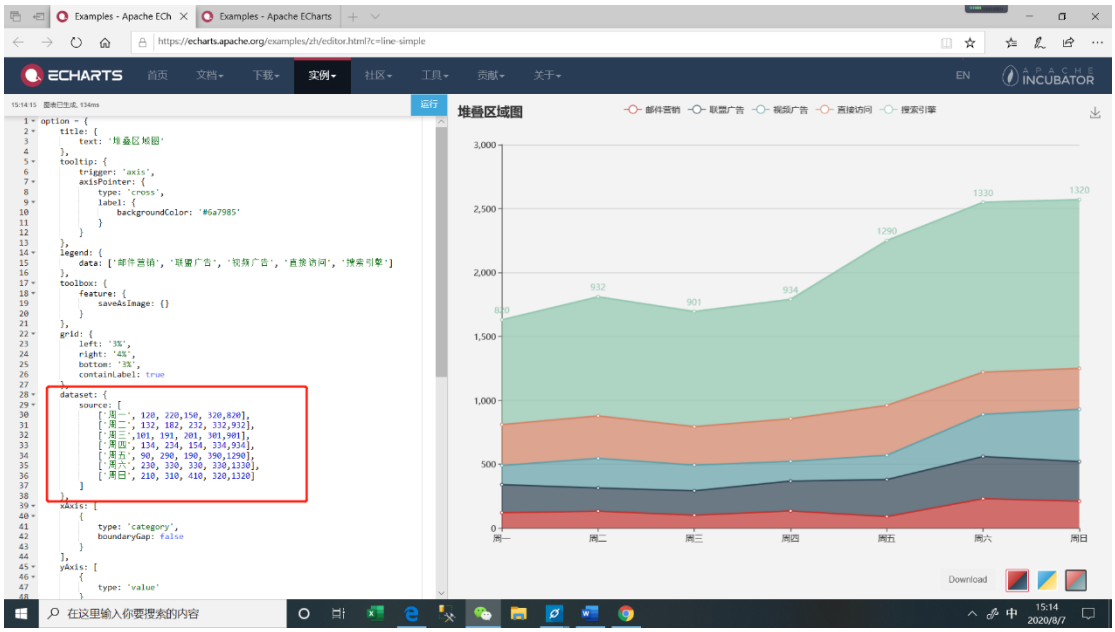



```
    option = {
            dataset: {

                    source: [
                        ['Mon',820],
                        ['Tue',932],
                        ['Wed',901],
                        ['Thu',934],
                        ['Fri',1290],
                        ['Sat',1330],
                        ['Sun',1320]

                    ]
                },
        xAxis: {
```

```
            type: 'category',
        },
        yAxis: {
            type: 'value'
        },
        series: [{
            type: 'line'
        }]
};
```

代码解析：

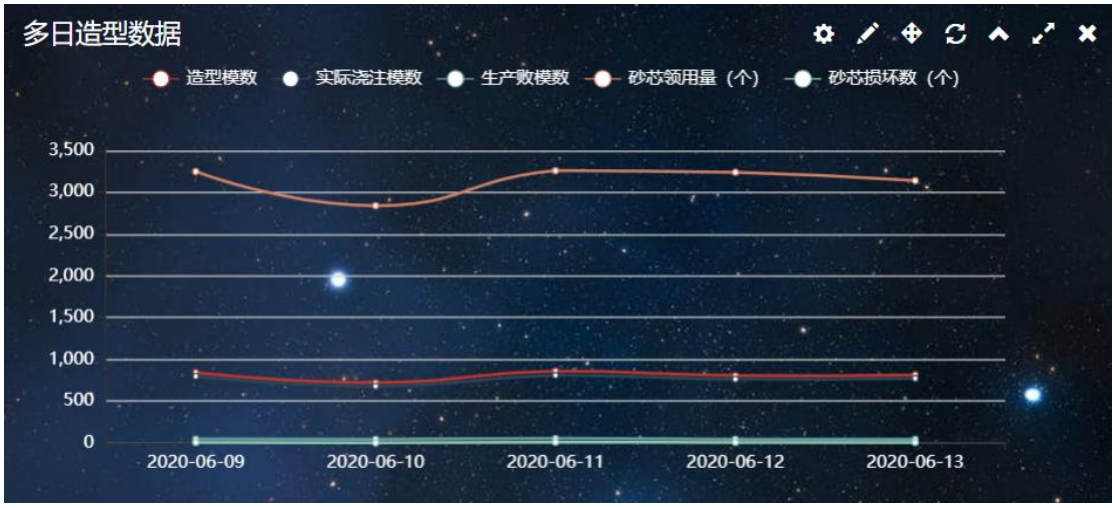
- dataset 数据默认第一列为 X 轴的分类，第二列开始是 Y 轴的数据。如果同一个分类，Y 轴有多个系列，则可以从第二列开始，增加第三列、第四列..管理 Y 轴的数据
  比如下面区域折线图的 dataset 数据，Y 轴有多个系列，则对应 dataset 可设计多列管理起来：



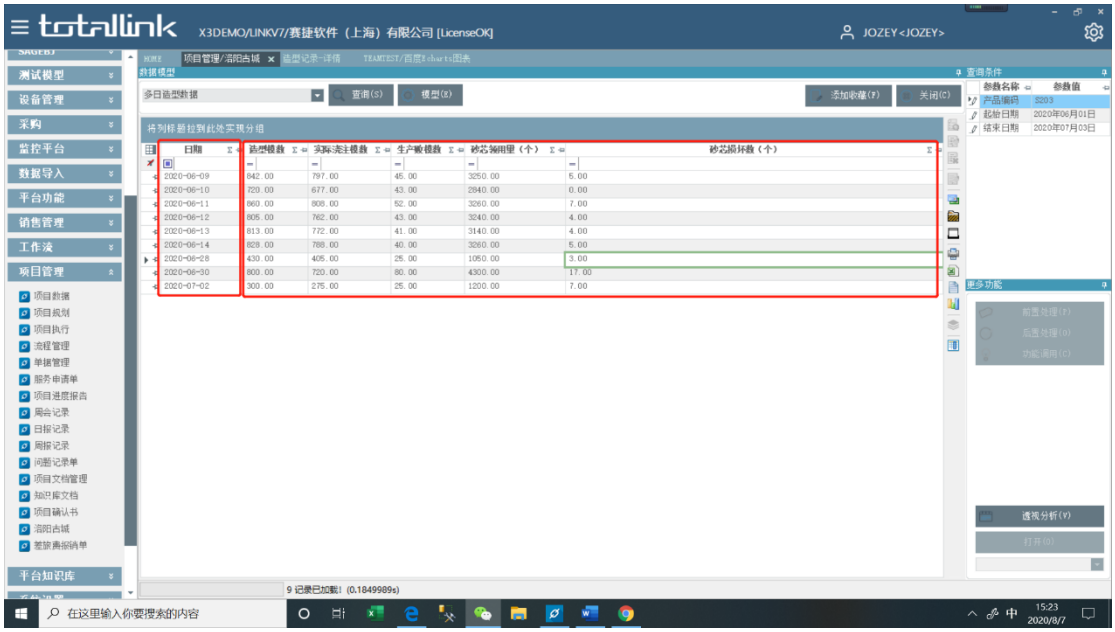
- 本例子中将数据转换成 dataset 之后，相对应 X 轴 Y 轴 series 内的 data 要去掉，并且在 dataset 中转置显示
- 由于 echarts 控制，并非所有的图表都支持 dataset，使用时请到 TotalLINK 知识库挑选对应图表代码直接使用即可

2.1.2 应用

比如想输出下面的图表



- 新建模型，让数据以 dataset 标准格式输出

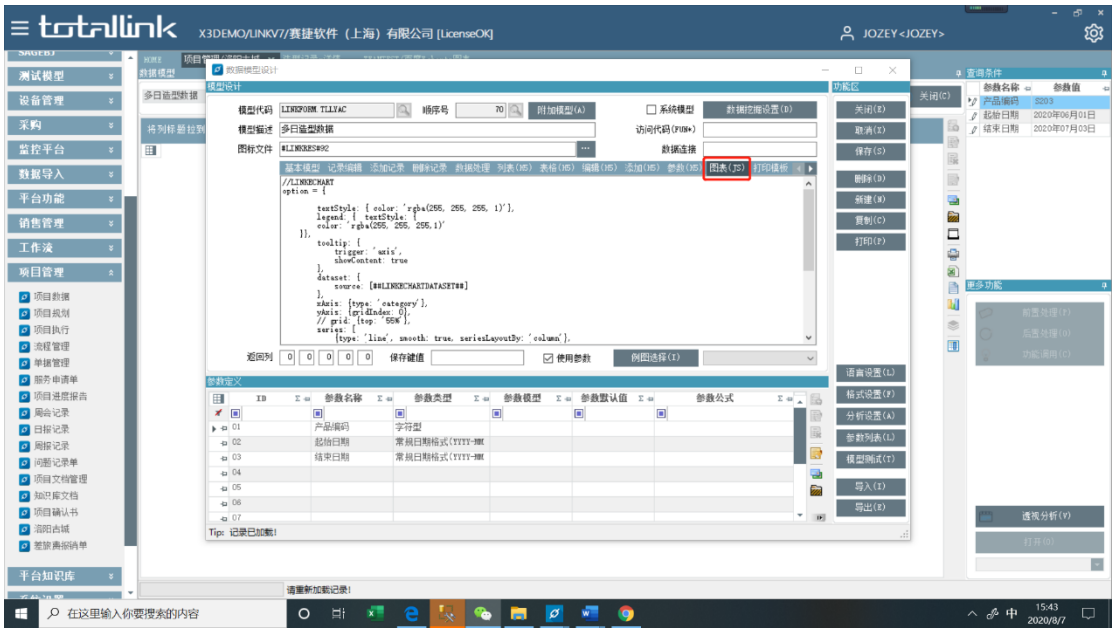


参考代码：

```
SELECT
CONVERT(varchar(10), H.DAT01, 23) AS 日期
,SUM(MOLDMOD)  AS 造型模数
,SUM(ACTCASMOD)  AS 实际浇注模数
,SUM(PROFAILMOD)  AS 生产败模数
,SUM(SANDCORNO)  AS '砂芯领用量（个）'
,SUM(DAMGCORNO)  AS '砂芯损坏数（个）'
FROM [LINKFORMLYAC] YC
INNER JOIN [LINKFORMH] H
ON YC.[DOCNUM]=H.[DOCNUM] AND H.[FORMTYP]='TLLYAC'
WHERE PROCODE='{0}' AND CONVERT(varchar(10), H.DAT01, 23) BETWEEN '{1}'
AND '{2}'
GROUP BY CONVERT(varchar(10), H.DAT01, 23)
```

```
ORDER BY CONVERT(varchar(10), H.DAT01, 23)
```

- 在该模型的“图表（JS）”模块写入多系列折线图的 dataset 格式的代码



参考代码：

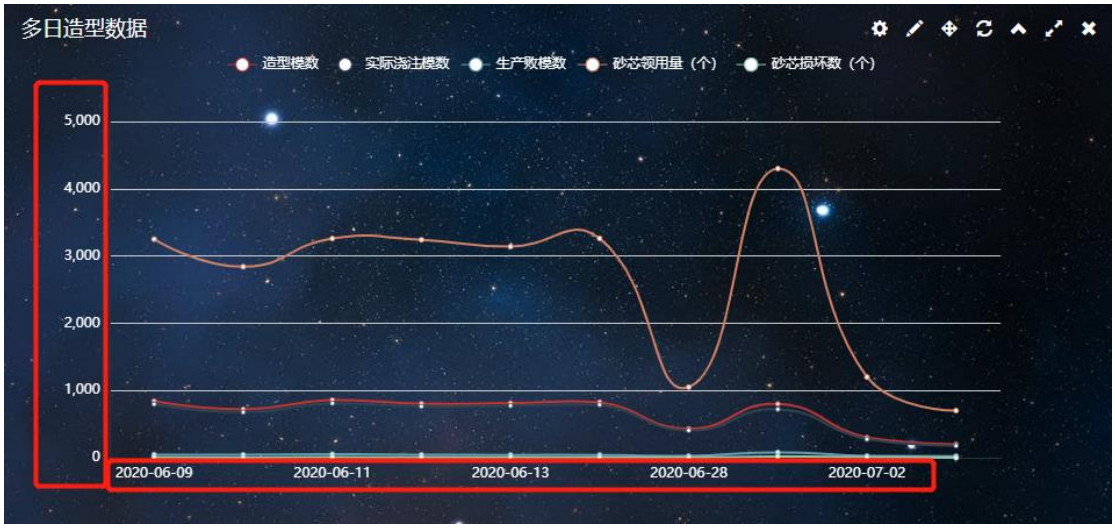
```
//LINKECHART
option = {

        textStyle: {
        color: 'rgba(255, 255, 255, 1)'
        },
        legend: {    textStyle: {

        color: 'rgba(255, 255, 255,1)'

    }},
        tooltip: {
            trigger: 'axis',
            showContent: true
        },
        dataset: {
            source: [##LINKECHARTDATASET##]
        },
        xAxis: {type: 'category'},
        yAxis: {gridIndex: 0},
        // grid: {top: '55%'},
        series: [
            {type: 'line', smooth: true, seriesLayoutBy: 'column'},
            {type: 'line', smooth: true, seriesLayoutBy: 'column'},
            {type: 'line', smooth: true, seriesLayoutBy: 'column'},
```

```
                    {type: 'line', smooth: true, seriesLayoutBy: 'column'},
                    {type: 'line', smooth: true, seriesLayoutBy: 'column'}

                ]
            };


myChart.setOption(option);
```

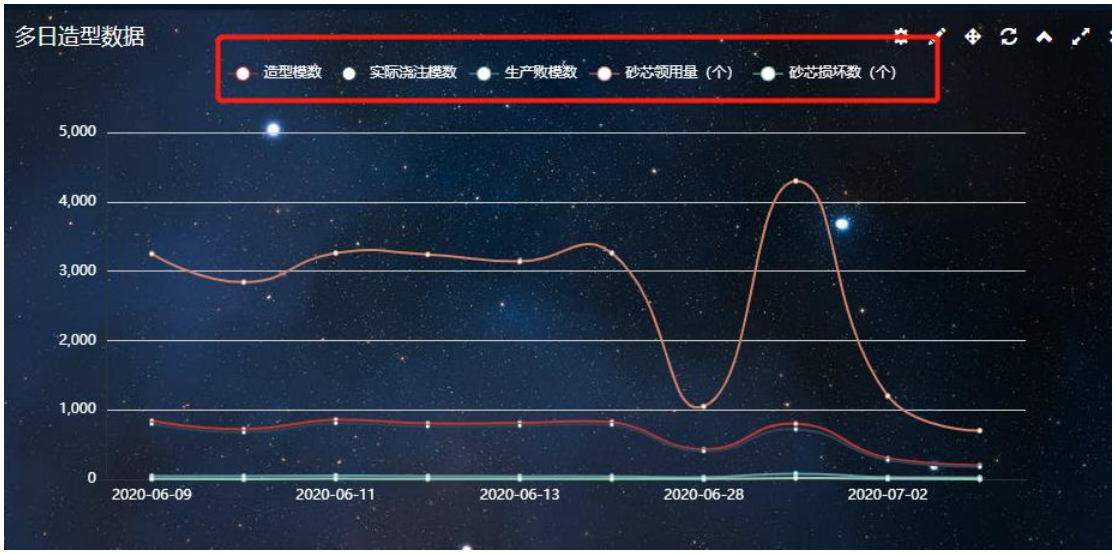
- **关键代码**
  - //LINKECHART
    写在代码最上方
  - myChart.setOption(option);
    写在代码最下方
  - ##LINKECHARTDATASET##
    调用该查询模型的数据
- **文字颜色**

```
textStyle: {
color: 'rgba(255, 255, 255, 1)'
},
```

在 option 方法加上面的代码，实现 X 轴、Y 轴描述变白色



在 legend 方法加上面的代码，实现图例说明变白色

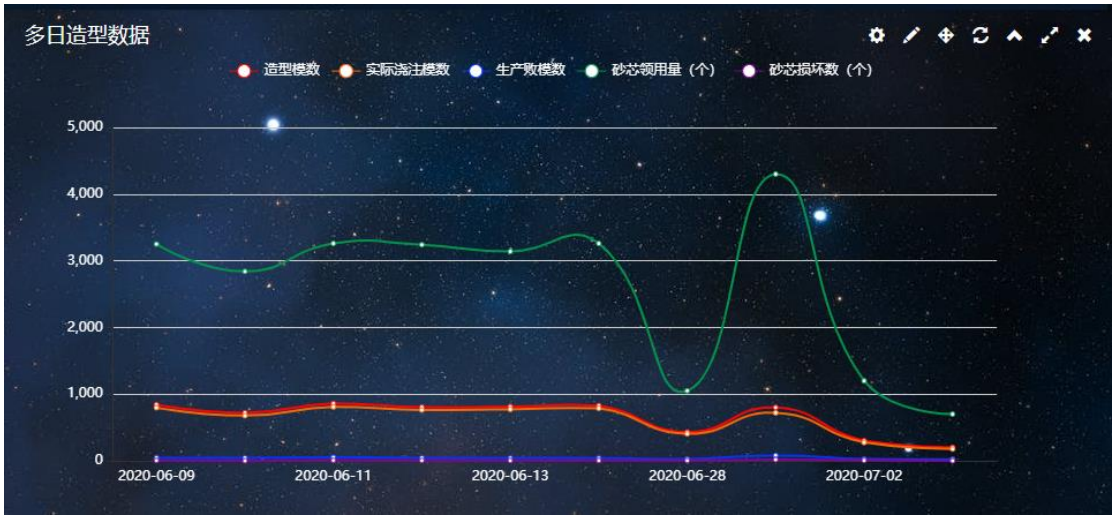


目前系统默认集成深色模式，文字默认白色，如果需要更改为其它颜色，可以根据此处代码修改

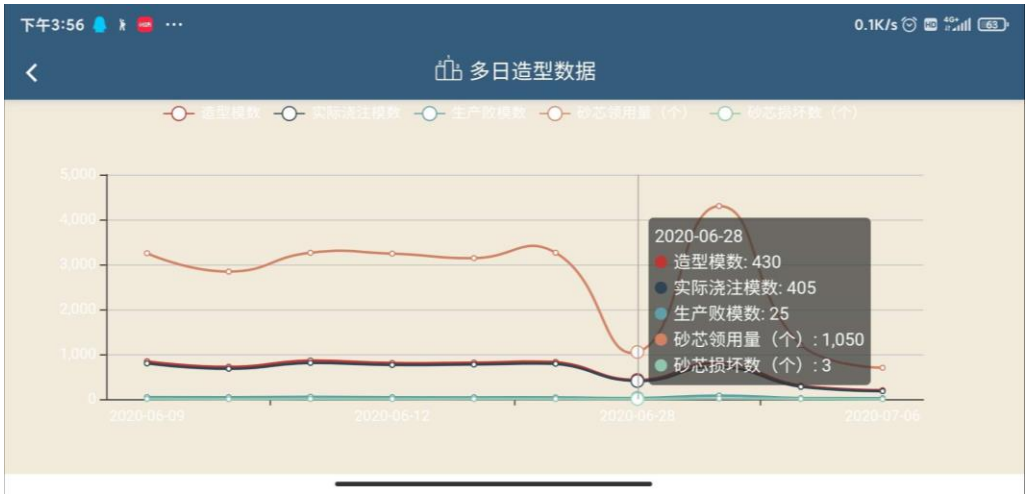
- **图表颜色**

```
color: [ '#F90606', '#F96806','#0635F9','#06934D','#7E0693'],
```

在 option 方法添加上面的代码，实现更换系列的颜色，多个系列的颜色用英文逗号隔开,比如本例根据图例顺序， 修改折线颜色分别为 红 橙 蓝 绿 紫,最终效果如下：



- 可以在 app 实时打开该模型查看调试的效果

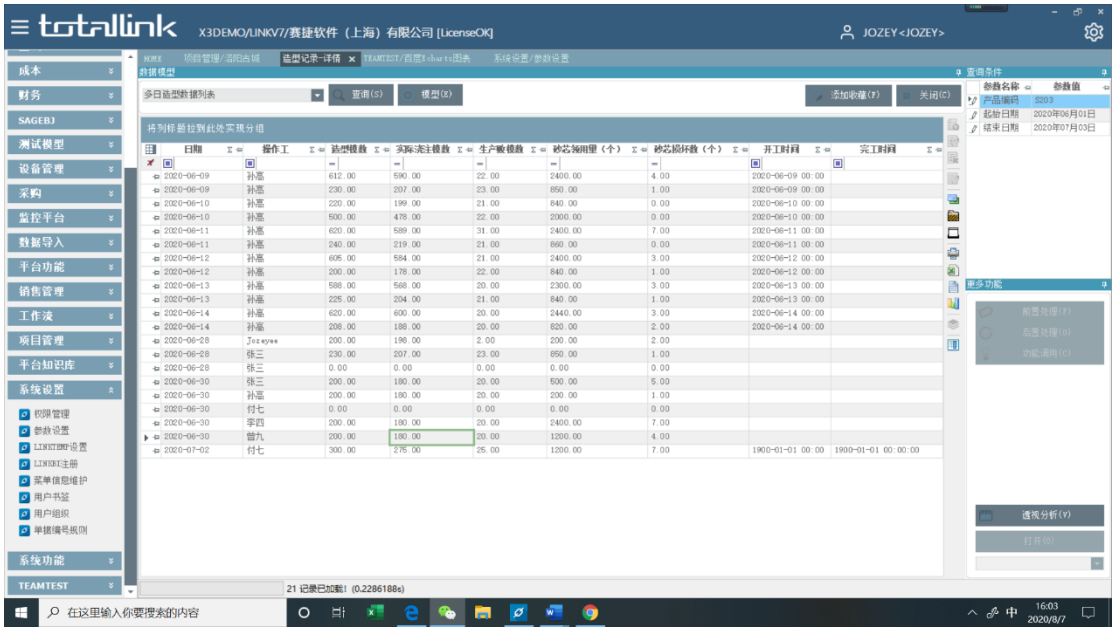


2.2 数据列表—DATATABLE

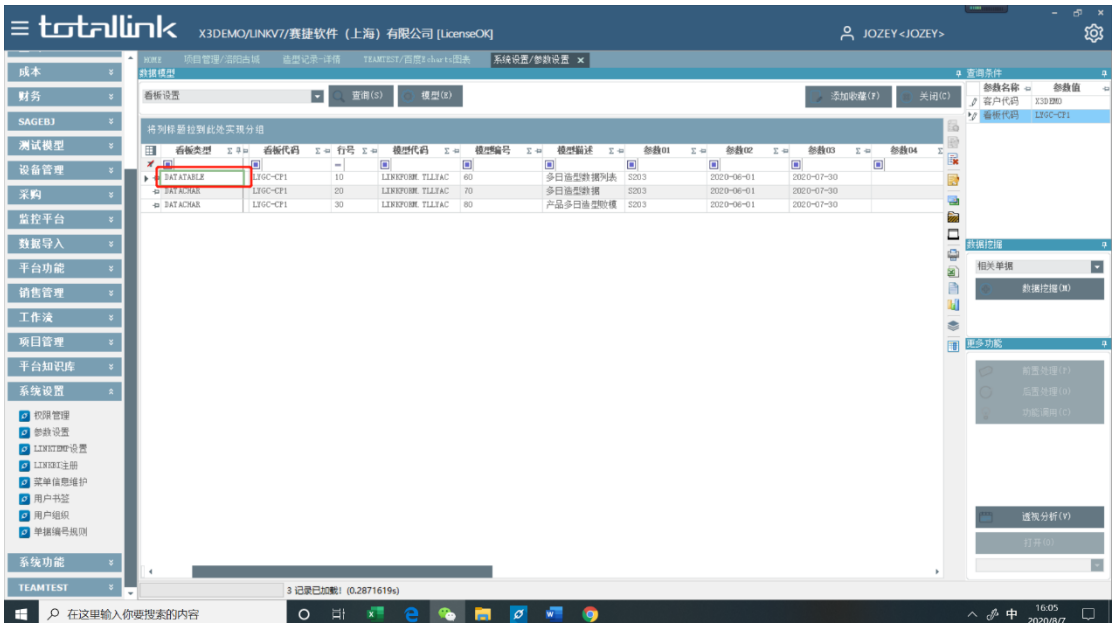
下面是列表在看板的输出

#	日期	操作工	造型模数	实际浇...	生产败...	砂芯领...	砂芯损...	开工时间	完工时间
1	2020-06...	孙高	612.00	590.00	22.00	2400.00	4.00	2020-06...	
2	2020-06...	孙高	230.00	207.00	23.00	850.00	1.00	2020-06...	
3	2020-06...	孙高	220.00	199.00	21.00	840.00	0.00	2020-06...	
4	2020-06...	孙高	500.00	478.00	22.00	2000.00	0.00	2020-06...	
5	2020-06...	孙高	620.00	589.00	31.00	2400.00	7.00	2020-06...	

- 在模型编写正常的查询语句



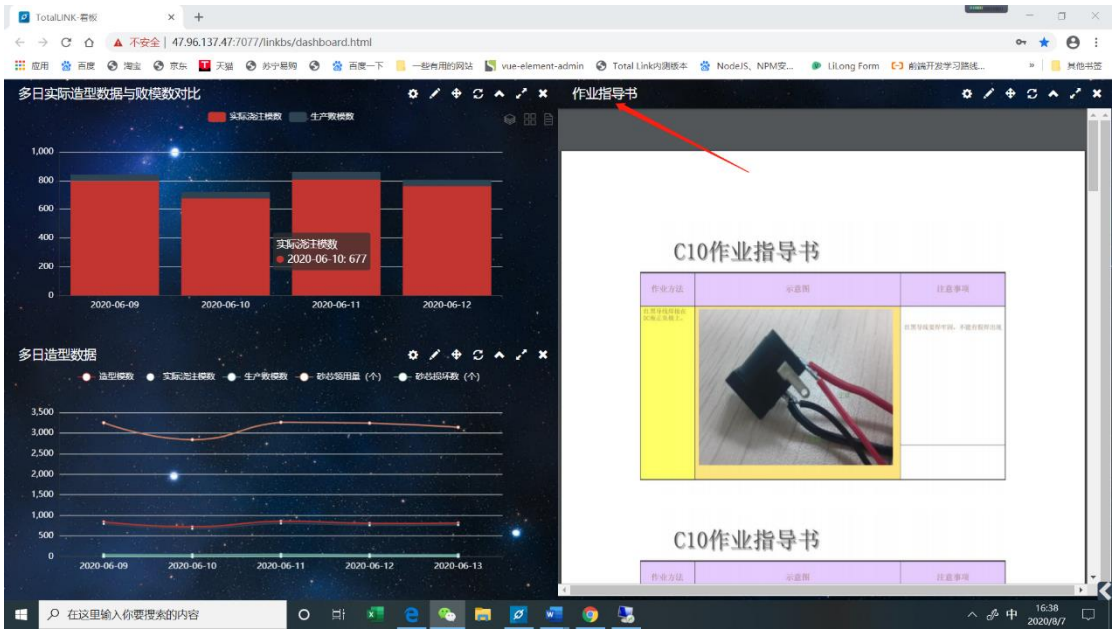
- 在看板设置时将看板类型设置为“DATATABLE”



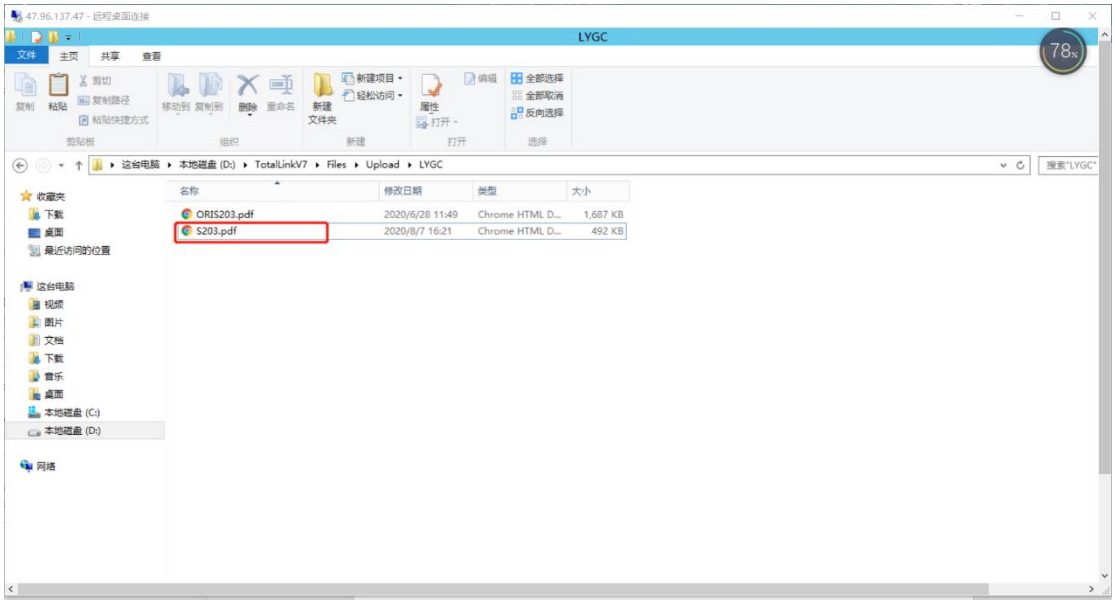
- 注意：列表模式的自动刷新时间与滚动秒数不要同时设置或间隔长些，否则会出现同一个表数据重复的问题，两者设置的时间应该相隔久点。
- 下面是自动刷新设置了 3 秒，自动滚动设置了 5 秒，当数据刷新时，数据往上滚动了，这个不好控制，建议两者不要同时设置。

2.3 PDF 类型—LINKURL

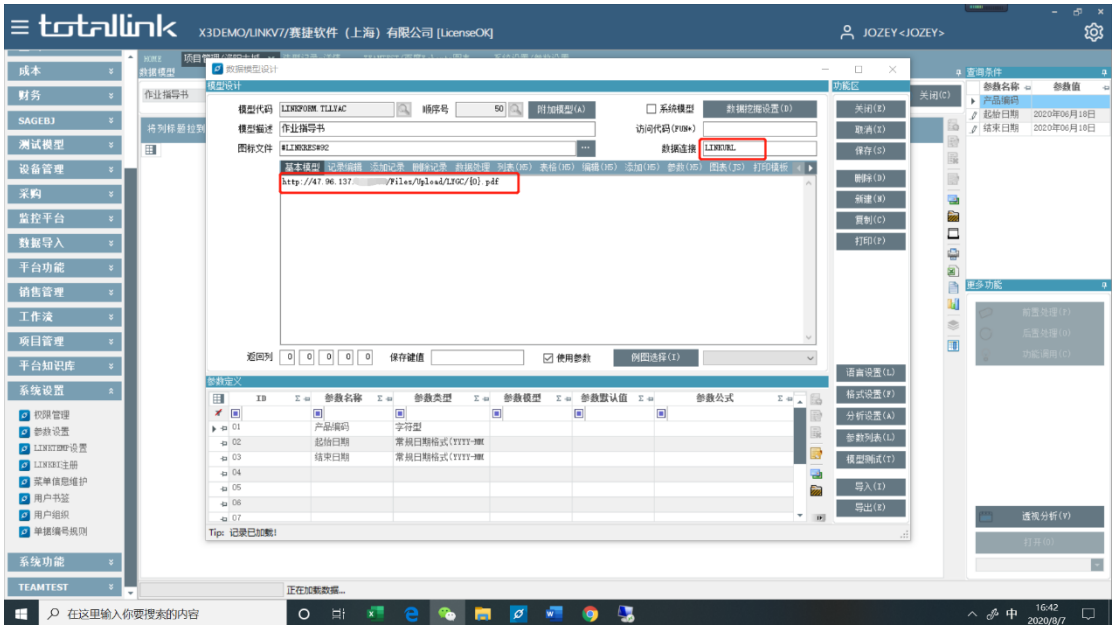
下面的看板嵌入了一个 PDF 类型的作业指导书



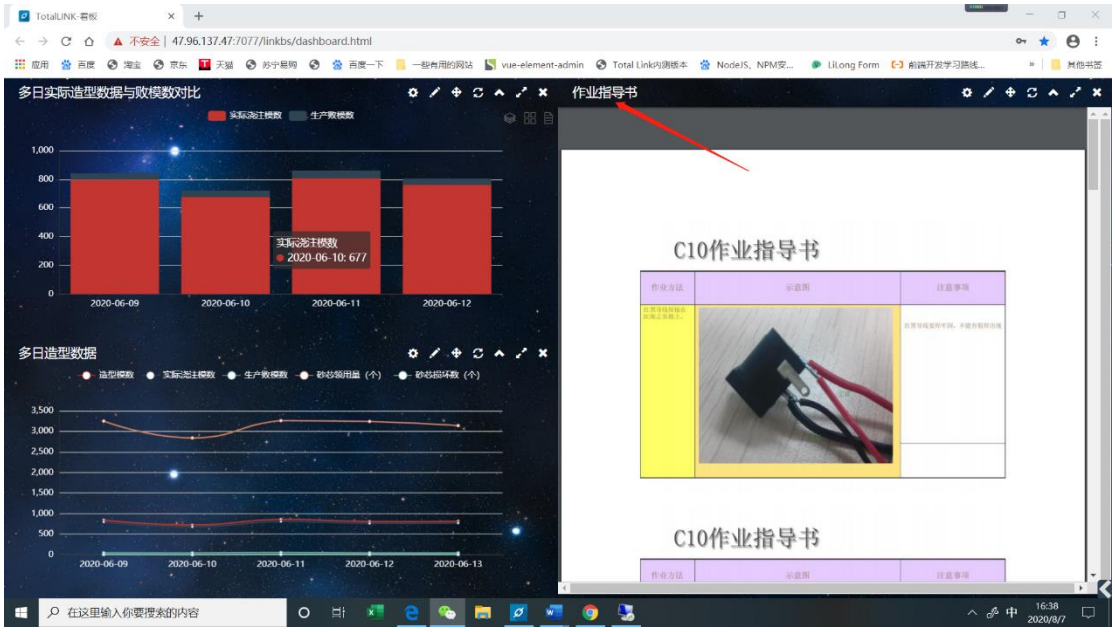
- 首先将 PDF 文件放到服务器固定文件夹下，确保客户端能够访问到该 PDF



- 在客户端基本模型模块直接填写服务器 PDF 的路径，数据链接为 LINKURL
- 如果看板带参数，并且 PDF 需要根据参数变化，可以参考本例编写代码，如果看板不带参数，请直接编写服务器 PDF 文件代码
- 代码参考：
  - 有参数：http://47.96.137.xx:xxxx/Files/Upload/LYGC/{0}.pdf
  - 无参数：http://47.96.137.xx:xxxx/Files/Upload/LYGC/S203.pdf



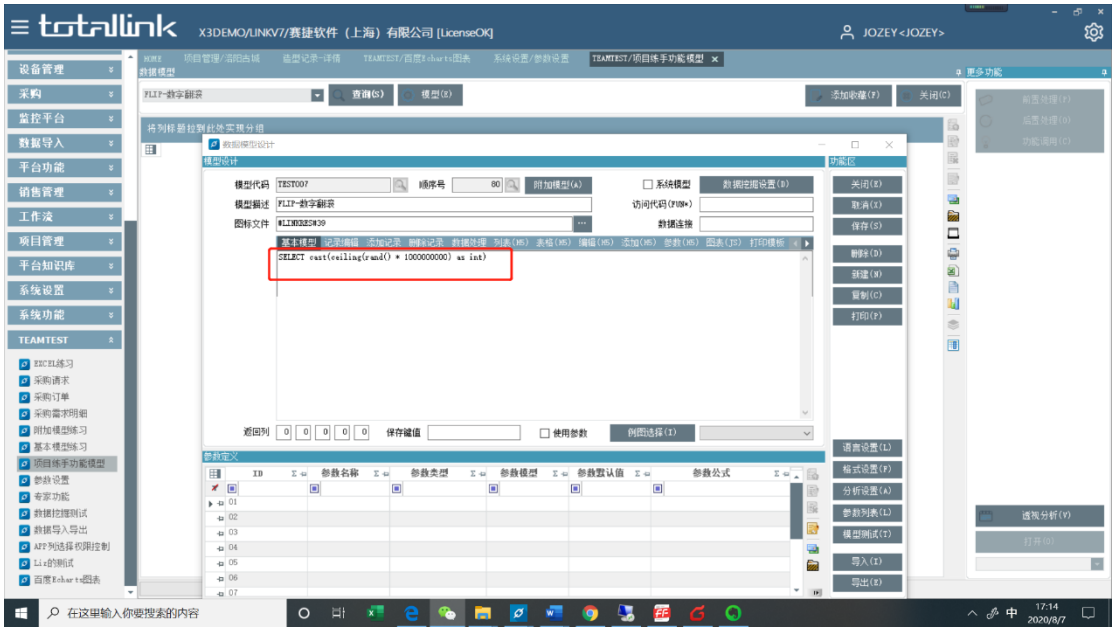
- 当在看板网站输入参数查询时，可以访问到对应的 PDF



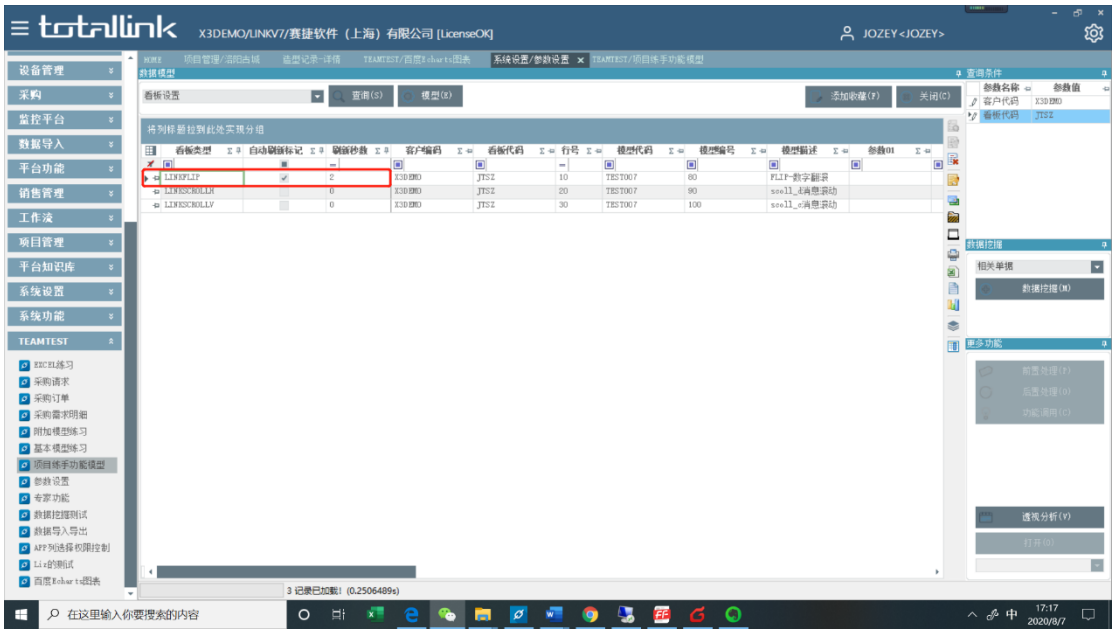
2.4 静态数据翻滚—LINKFLIP

当看板需要数据随着数据更新，可以使用下面的代码，并且设置刷新时间，可实现看板数据的滚动刷新，带来不一样的视觉效果。应用场景有电商订单总量、车间零件个数、图书馆当前人数等

- 如下图，新建一个模型，查询可随时间变化的数据数字
- 本例做了一个随机查询测试



- 看板类型设置为 LINKFLIP，并且自动刷新，这里每 2 秒查询一次该模型

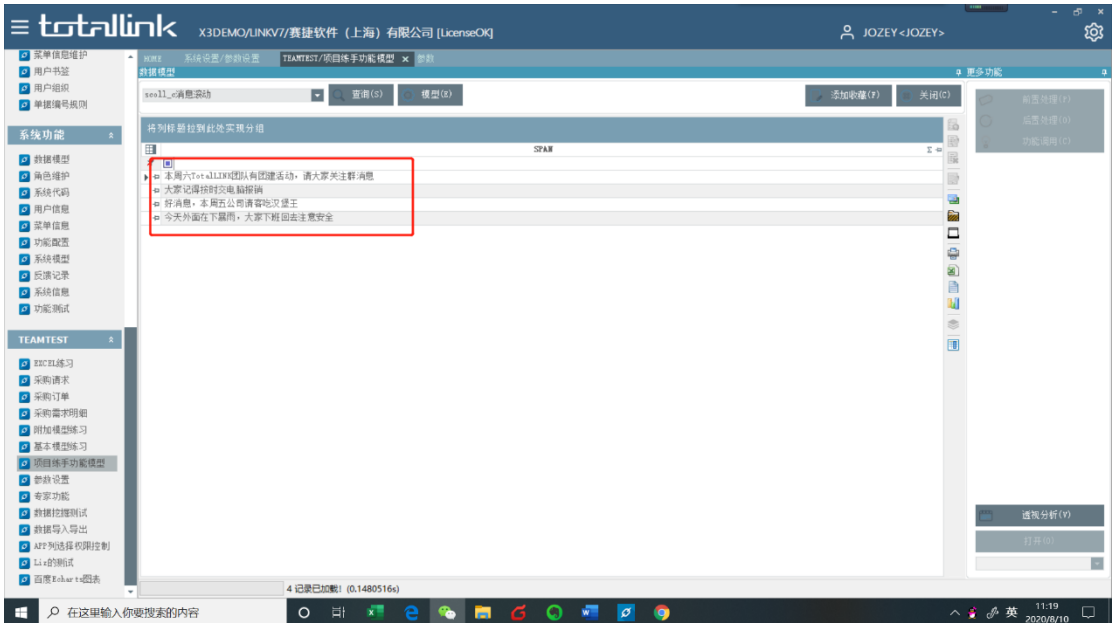


- 以上设置完毕可以看到随时间翻滚的数字

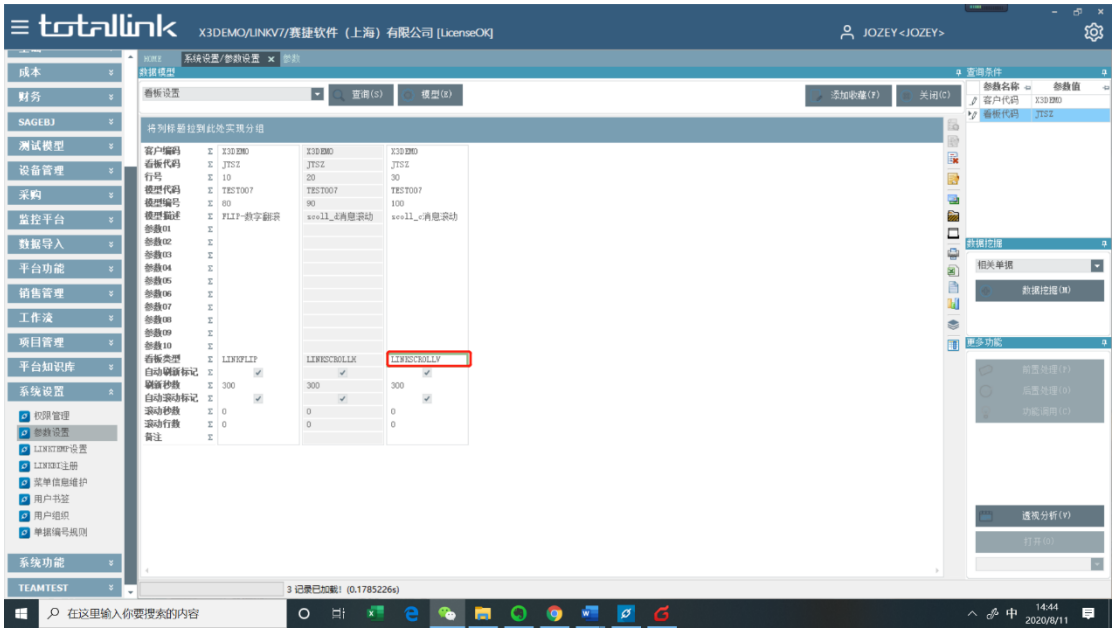
2.5 竖向消息滚动—LINKSCROLLV

上下消息滚动可在大屏上得到上下定时滚动的消息通知

- 如下图，新建一个模型，查询通知消息



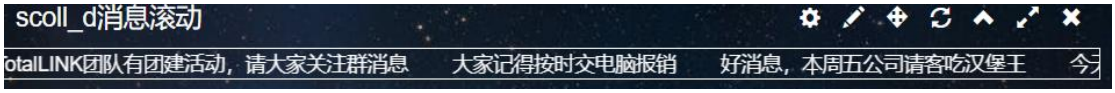
- 看板类型设置为 LINKSCROLLV



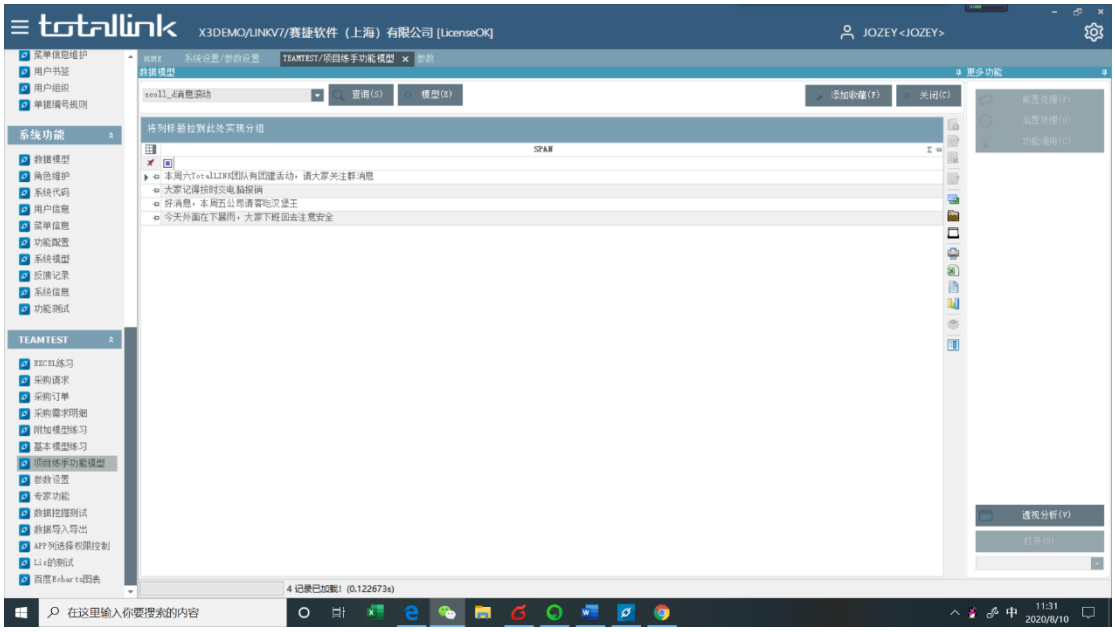
- 以上设置完毕可以看到消息慢慢往上滚动

2.6 横向消息滚动--LINKSCROLLH

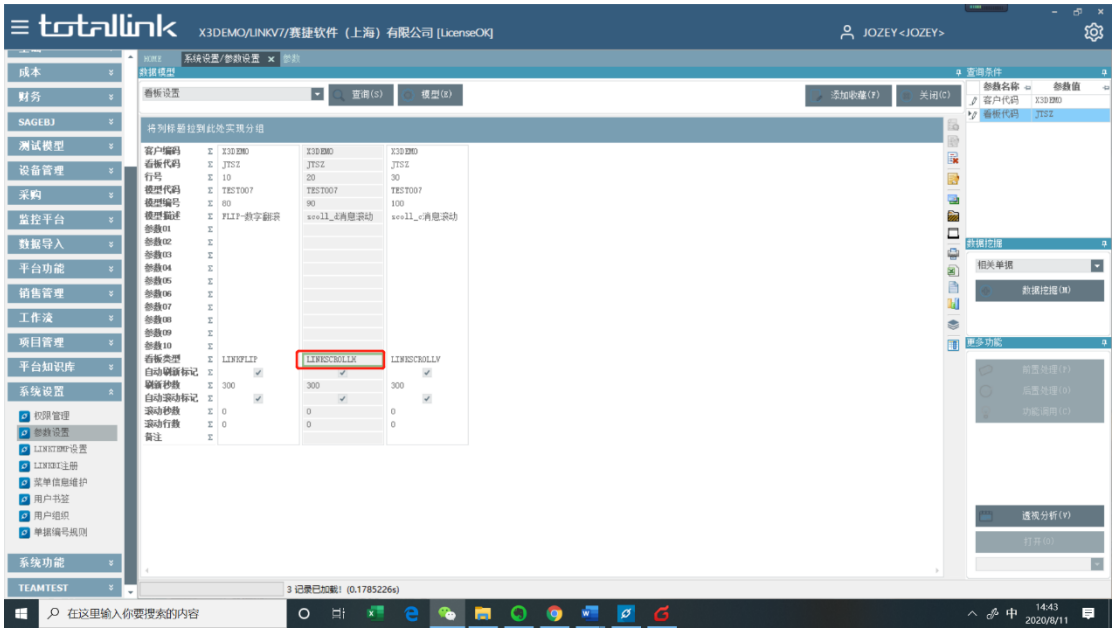
左右消息滚动可在大屏上得到左右定时滚动的消息通知



- 如下图，新建一个模型，查询通知消息



- 看板类型设置为 LINKSCROLLH



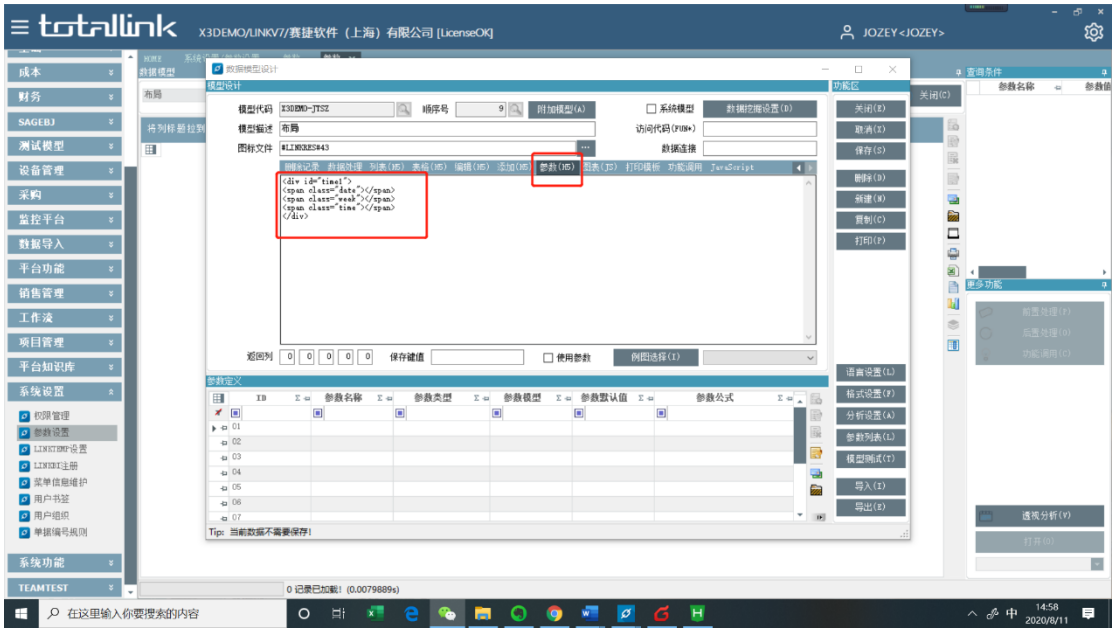
- 以上设置完毕可以看到消息慢慢往左滚动

2.7 其它静态效果

2.7.1 系统时间

- 在挖掘过来的全局 9 号模型“参数”、“Javascript”模块加相应代码，可以实现在看

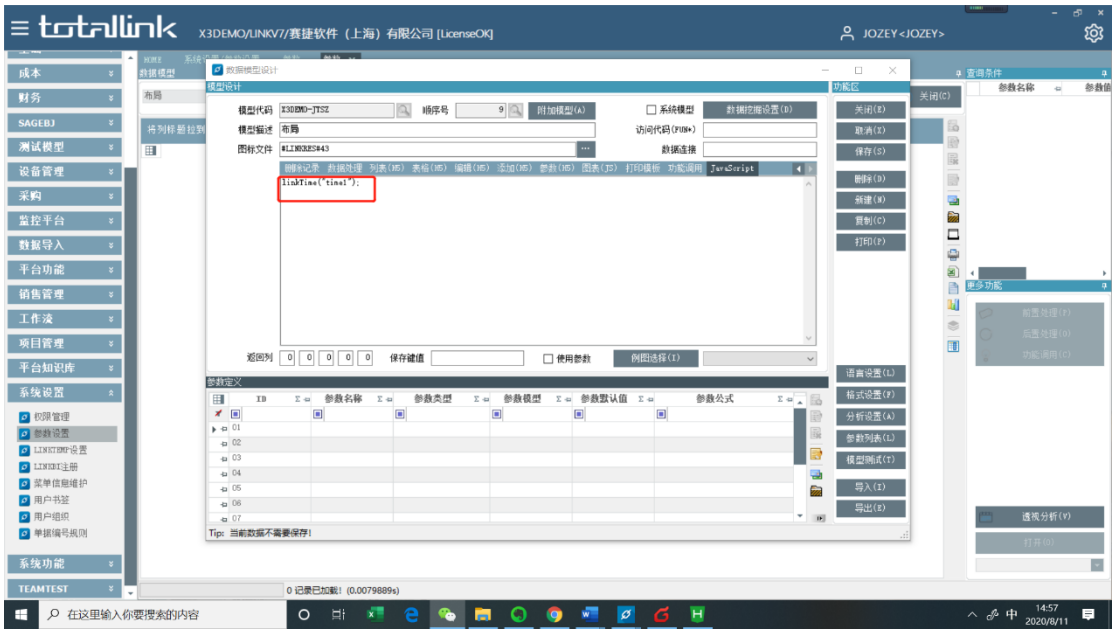
板上显示当前系统时间



参考代码：

```
<div id="time1">
    <span class="date"></span>
    <span class="week"></span>
    <span class="time"></span>
</div>
```

代码解析：显示时分秒



参考代码：

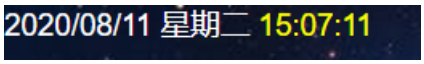
```
linkTime("time1");
```

代码解析：js 返回调用的时间函数

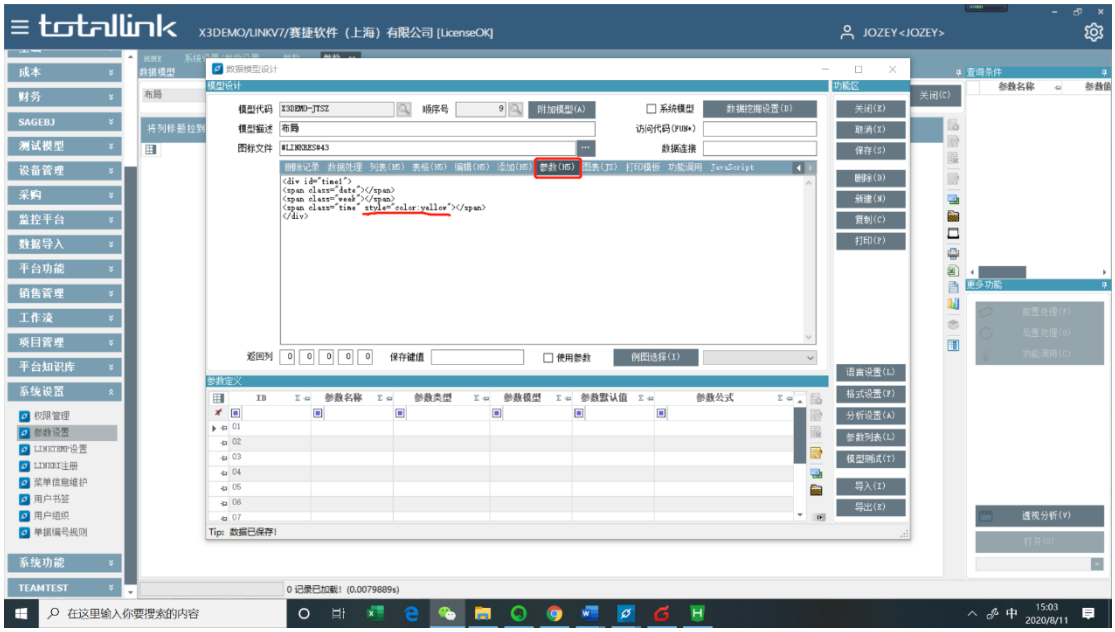
- 显示效果：

拓展：如果需要修改显示的 UI，有两种方法：

比如修改时分秒部分的 UI 为黄色



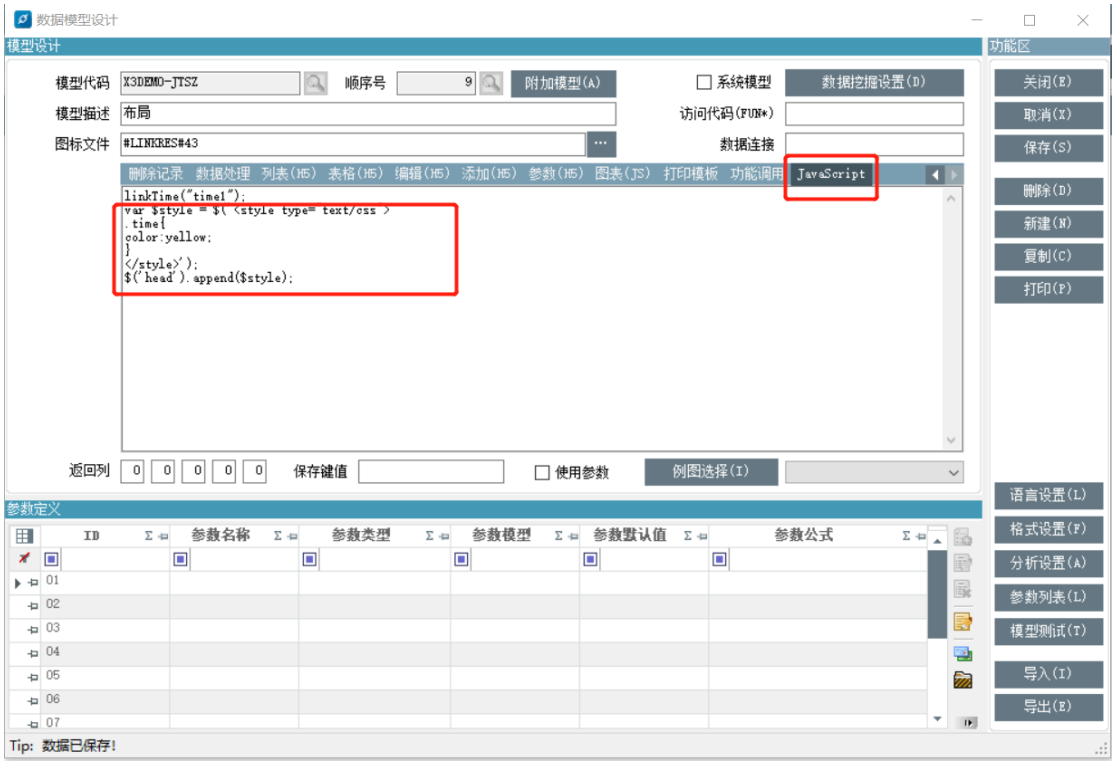
方法一：直接在标签上增加 CSS



参考代码：

```
    <div id="time1">
    <span class="date"></span>
    <span class="week"></span>
    <span class="time" style="color:yellow"></span>
</div>
```

方法二：JS 动态生成当前标签的 CSS

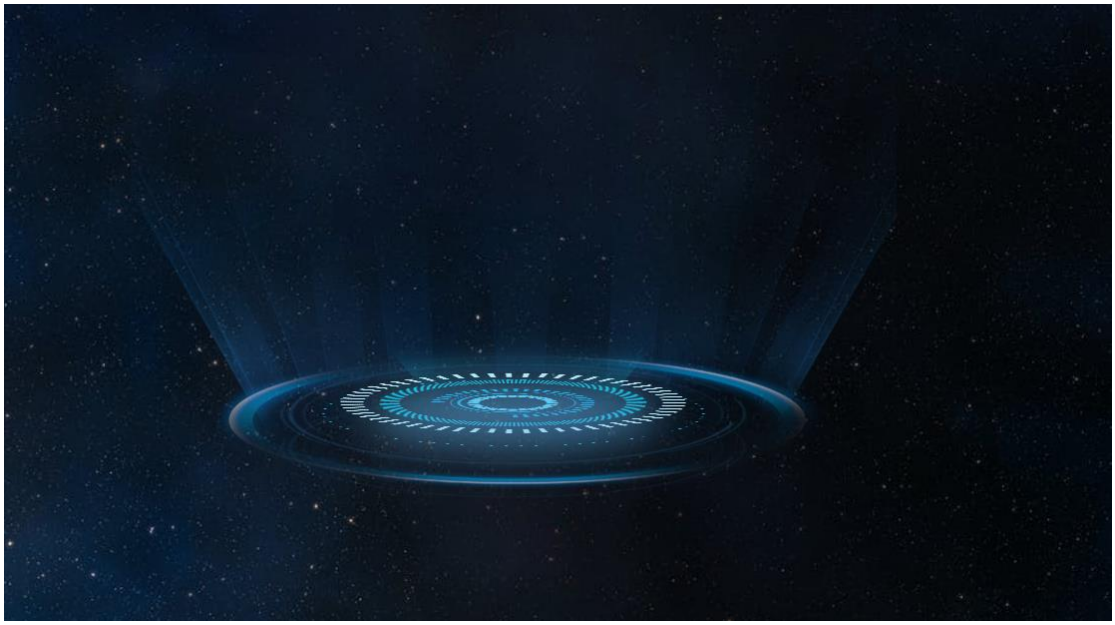


参考代码：

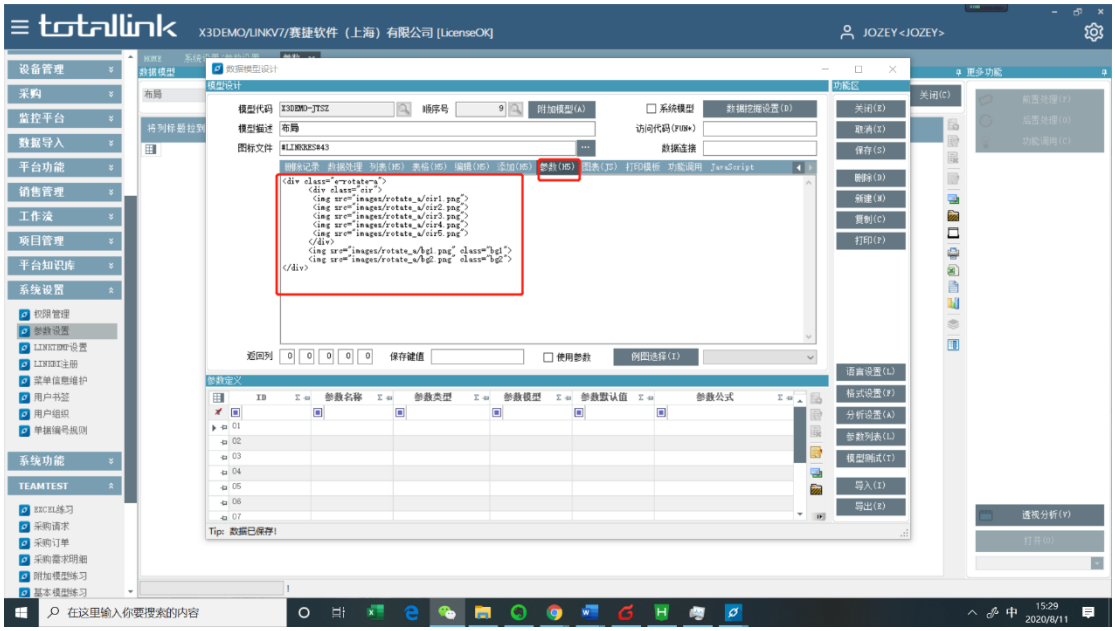
```
linkTime("time1");
var $style = $('<style type="text/css">
      .time{
      color:yellow;
      }
      </style>');
$('head').append($style);
```

实际和图表结合会使用第二种方式统一管理 CSS 代码，调整时间显示的位置

2.7.2 转盘



- 在挖掘过来的全局 9 号模型“参数”、“Javascript”模块加相应代码，可以实现在看板上显示转盘
- 在“参数”模块添加代码



参考代码：

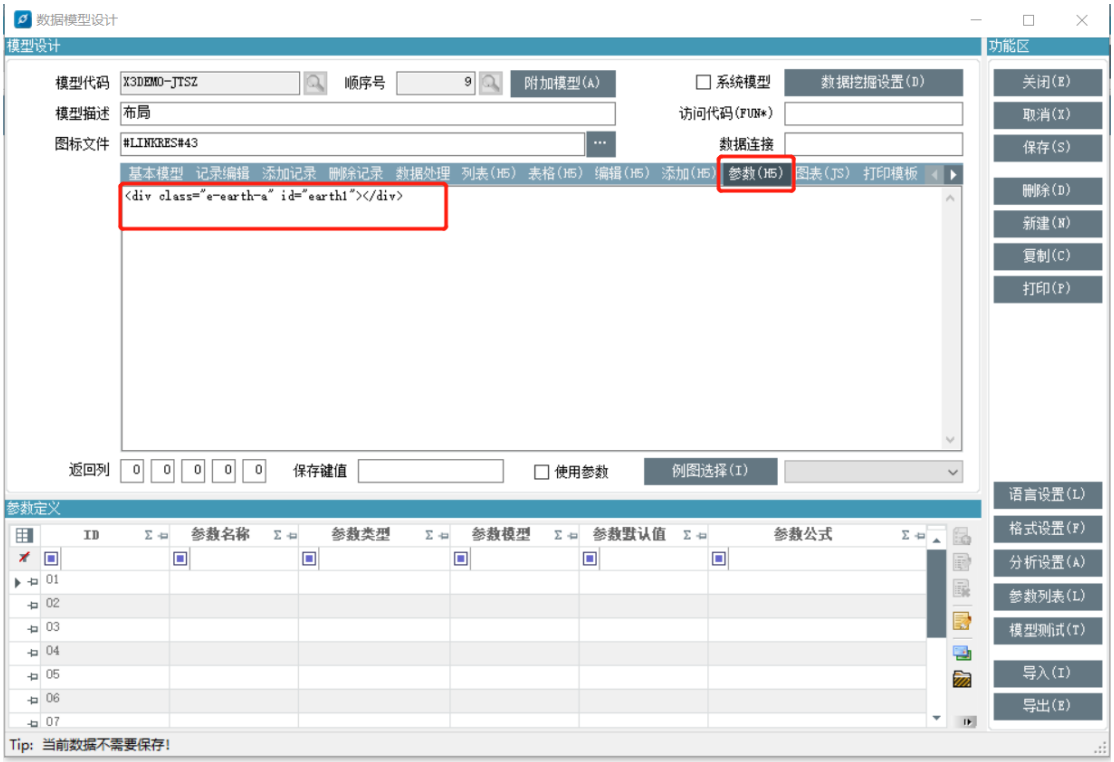
```
<div class="e-rotate-a">
        <div class="cir">
          <img src="images/rotate_a/cir1.png">
          <img src="images/rotate_a/cir2.png">
          <img src="images/rotate_a/cir3.png">
          <img src="images/rotate_a/cir4.png">
```

```
        <img src="images/rotate_a/cir5.png">
      </div>
      <img src="images/rotate_a/bg1.png" class="bg1">
      <img src="images/rotate_a/bg2.png" class="bg2">
</div>
```

- “Javascript”模块可以添加动态生成 css 的代码，可以调整该转盘的位置，此处不举例，注意类名固定为 e-rotate-a

2.7.3 地球

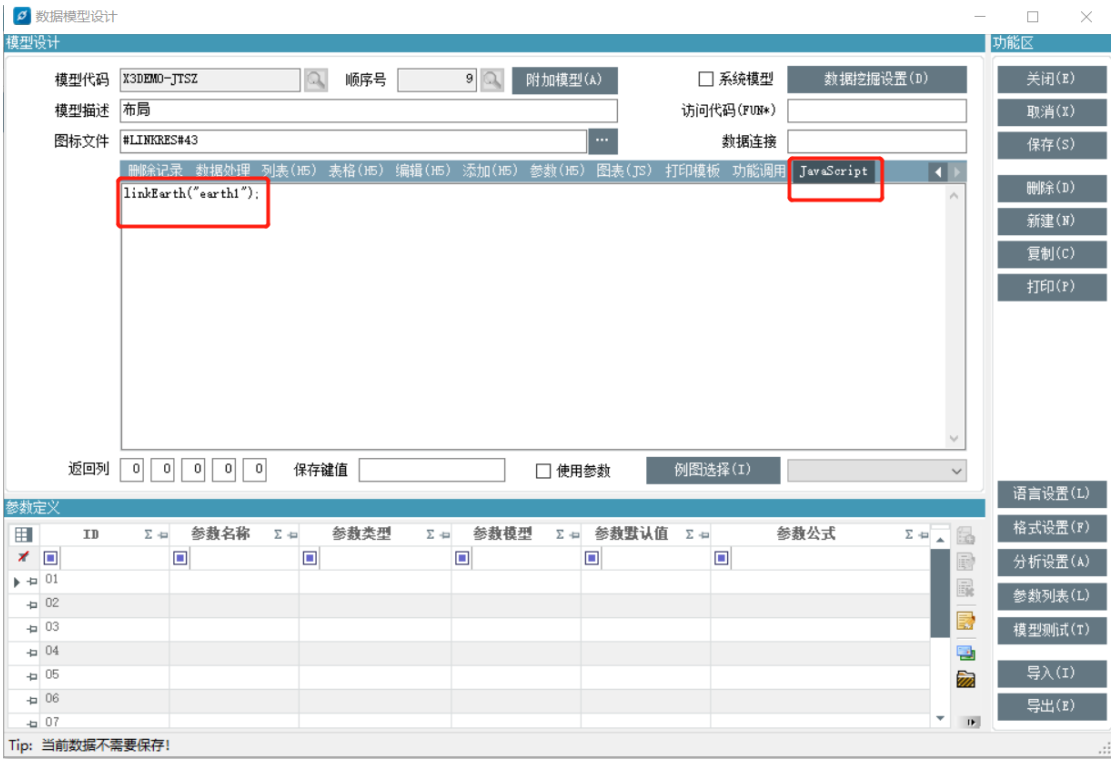
- 在挖掘过来的全局 9 号模型“参数”、“Javascript”模块加相应代码，可以实现在看板上显示地球
- 在“参数”模块添加代码



参考代码：

```
<div class="e-earth-a" id="earth1"></div>
```

- “Javascript”模块添加调用地球方法的代码，也可以添加动态生成 css 的代码，可以调整该地球的位置，此处不举例，注意类名固定为 earth1



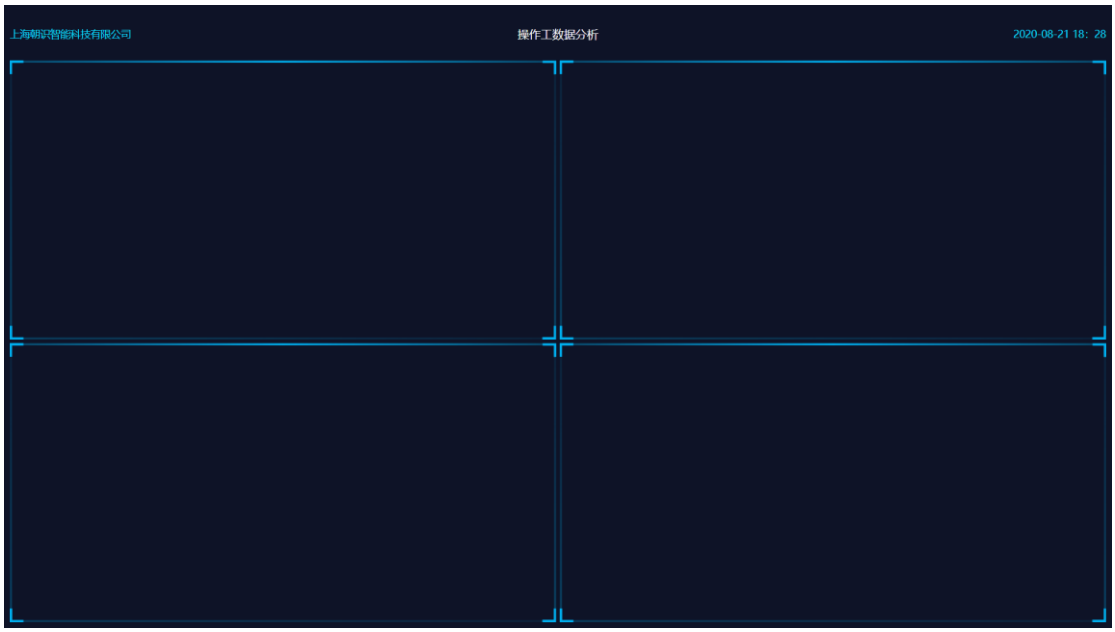
参考代码：

```
linkEarth("earth1");
```

3 栅格布局

本小节介绍两个案例（整个 html 代码参考附件），帮助大家理解栅格布局。

3.1.1 案例一



- 我们给屏幕预留了 4rem 高度、整行的宽度用来放标题、公司名称、系统时间等内容。屏幕剩下的部分利用 12*12 的分布设计栅格（意思是把剩下的屏幕分成 12*12 个格子）

grid-row-start：宽度占整个屏幕的几份
grid-column-start: 高度占整个屏幕的几份

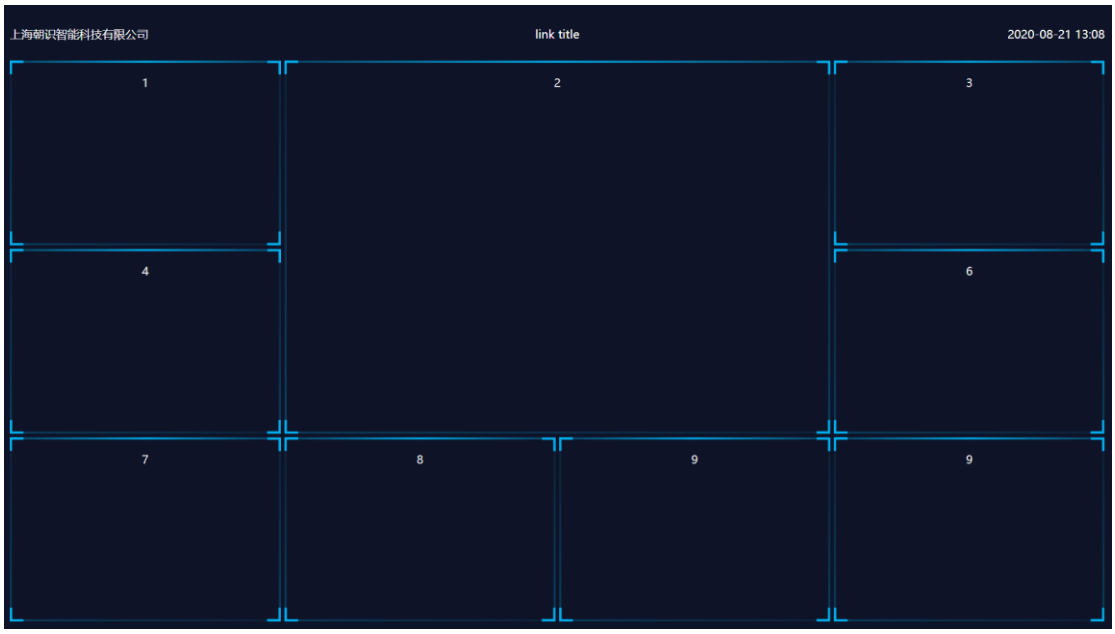
参考代码：

```
<div class="linkbox">

    <div class="item linkcompany" style="font-size:14px ;color:#09ddff">上海朝识智能科技有限公司</div>
    <div class="item linktitle" >操作工数据分析</div>
    <div class="item linktime" style="font-size:14px ;color:#09ddff" >2020-08-21 18:28</div>

    <div class="item linktheme3" linkid="xxx" style="grid-row-start: span 6; grid-column-start: span 6;"></div>
    <div class="item linktheme3" linkid="xxx style="grid-row-start: span 6; grid-column-start: span 6;"></div>
```

```
    <div class="item linktheme3" linkid="xxx" style="grid-row-start: span 6; grid-column-start: span 6;"></div>
    <div class="item linktheme3" linkid="xxx" style="grid-row-start: span 6; grid-column-start: span 6;"></div>

</div>
```

解析：
当布局画好后，将上方代码放进 9 号模型的参数模块，将 linkid 替换掉即可
总结：
分成多少个格子，就有多少个 class=”item”的 div

3.1.2 案例二



参考代码：

```
<div class="linkbox">
                    <div class="item linkcompany" >上海朝识智能科技有限公司</div>
                    <div class="item linktitle">link title</div>
                    <div class="item linktime" >2020-08-21 13:08</div>

                    <div class="item linktheme3" linkid="llll" style="grid-row-start: span 4; grid-column-start: span 3;">1</div>
                    <div class="item linktheme3" linkid="llll" style="grid-row-start: span 8; grid-column-start: span 6;">2</div>
                    <div class="item linktheme3" linkid="llll" style="grid-row-start: span 4; grid-column-start: span 3;">3</div>
```

```
    <div class="item linktheme3" linkid="llll" style="grid-row-start: span 4; grid-column-start: span 3;">4</div>
                    <div class="item linktheme3" linkid="llll" style="grid-row-start: span 4; grid-column-start: span 3;">6</div>



                    <div class="item linktheme3" linkid="llll" style="grid-row-start: span 4; grid-column-start: span 3;">7</div>
                    <div class="item linktheme3" linkid="llll" style="grid-row-start: span 4; grid-column-start: span 3;">8</div>
                    <div class="item linktheme3" linkid="llll" style="grid-row-start: span 4; grid-column-start: span 3;">9</div>
                    <div class="item linktheme3" linkid="llll" style="grid-row-start: span 4; grid-column-start: span 3;">9</div>
            </div>
```

4 附件:

4.1.1 附件一

栅格布局案例一 html 调试代码:

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <style type="text/css">
            html,
            body {
                width: 100%;
                height: 100%;
            }

            body {
                margin: 0px;
                background-color: #0E1327;
            }

.linkbox {
    /* background-color: aquamarine; */
    width: calc(100% - 16px);
    height: calc(100% - 16px);
    padding: 8px;
    /* margin: 8px; */
    /*  声明一个容器  */
    display: grid;
    /*    声明列的宽度    */
    grid-template-columns: repeat(12, 1fr);
    /*    声明行间距和列间距    */
    grid-gap: 5px;
    /*    声明行的高度    */
    grid-template-rows: 4em repeat(12, 1fr);
    justify-content: stretch;

}

.linktitle{
    grid-column-start: span 4;
    font-size: 20px;
```

```
        /*flex 布局*/
        display: flex;
        /*实现垂直居中*/
        align-items: center;
        /*实现水平居中*/
        justify-content: center;
        text-align: justify;

    }

    .linkcompany {
        grid-column-start: span 4;
        color: white;
        font-size: 20px;
        /*flex 布局*/
        display: flex;
        /*实现垂直居中*/
        align-items: center;
        /*实现水平居中*/
    /*  justify-content: center;
        text-align: justify; */
    }

    .linktime {
        grid-column-start: span 4;
        /*flex 布局*/
        display: flex;
        /*实现垂直居中*/
        align-items: center;
        /*实现水平居中*/
    /*  justify-content: center;
        text-align: justify; */
        margin-left:auto;
    }
    .item {
        text-align: center;
        font-size: 100%;
        color: #fff;

    }
    .linktheme1 {
        box-shadow: 0 0 3rem rgba(0, 85, 255, 0.5) inset;
        background: rgba(2, 17, 60, .3);
```

```
    }

    .linktheme2 {
        border: 1px solid rgba(0, 180, 220, 0.5);
        box-shadow: 0 0 2rem rgba(0, 180, 220, .5) inset
    }

    .linktheme3 {
        color: rgba(255, 255, 255, .9);
        border: 20px solid transparent;
        border-image:
url(http://47.96.137.47:7077/linkbs/images/frames/frame1.png) 5%;
        /*                                                    background-image:
url(http://www.imaoda.com/s/img/tpl/border.png) 5%; */
        background: rgba(14, 19, 39, 1);
        /* box-shadow: 0 0 5rem rgb(0, 110, 150) inset */
    }

    .linktheme4 {
        background:
url(http://47.96.137.47:7077/linkbs/images/frames/frame2.png) no-repeat;
        background-size: contain;
    }

    </style>
    </head>
    <body>




    <div class="linkbox">

    <div class="item linkcompany" style="font-size:14px ;color:#09ddff"></div>
    <div class="item linktitle" >操作工数据分析</div>
    <div class="item linktime" style="font-size:14px ;color:#09ddff" ></div>

    <div class="item linktheme3" linkid="DEMO2010" style="grid-row-start: span
6; grid-column-start: span 6;">1</div>
    <div class="item linktheme3" linkid="DEMO2020" style="grid-row-start: span
6; grid-column-start: span 6;">2</div>

    <div class="item linktheme3" linkid="DEMO2010" style="grid-row-start: span
6; grid-column-start: span 6;">3</div>
```

```
    <div class="item linktheme3" linkid="DEMO2020" style="grid-row-start: span 6; grid-column-start: span 6;">4</div>
    </div>
    </body>
</html>
```

4.1.2 附件二

栅格布局案例二 html 调试代码：

```
    <!DOCTYPE html>
    <html>
        <head>
            <meta charset="utf-8">
            <title></title>
            <style type="text/css">
                html,
                body {
                    width: 100%;
                    height: 100%;
                }

                body {
                    margin: 0px;
                    background-color: #0E1327;
                }

                .linkbox {
                    /* background-color: aquamarine; */
                    width: calc(100% - 18px);
                    height: calc(100% - 18px);
    /*                  width: 100%;
                    height: 100%; */
                    padding: 8px;
            /*      margin: 8px; */
                    /* 声明一个容器 */
                    display: grid;
                    /*  声明列的宽度  */
                    grid-template-columns: repeat(12, 1fr);
                    /*  声明行间距和列间距  */
                    grid-gap: 5px;
                    /*  声明行的高度  */
                    grid-template-rows: 4em repeat(12, 1fr);
```

```
        justify-content: stretch;

    }

    .linktitle{
        grid-column-start: span 4;
        font-size: 20px;
        /*flex  布局*/
        display: flex;
        /*实现垂直居中*/
        align-items: center;
        /*实现水平居中*/
        justify-content: center;
        text-align: justify;

    }

    .linkcompany {
        grid-column-start: span 4;
        color: white;
        font-size: 20px;
        /*flex  布局*/
        display: flex;
        /*实现垂直居中*/
        align-items: center;
        /*实现水平居中*/
    /*  justify-content: center;
        text-align: justify; */
    }

    .linktime {
        grid-column-start: span 4;
        /*flex  布局*/
        display: flex;
        /*实现垂直居中*/
        align-items: center;
        /*实现水平居中*/
    /*  justify-content: center;
        text-align: justify; */
        margin-left:auto;
    }
    .item {
        text-align: center;
```

```
                font-size: 100%;
                color: #fff;

            }
            .linktheme1 {
                box-shadow: 0 0 3rem rgba(0, 85, 255, 0.5) inset;
                background: rgba(2, 17, 60, .3);
            }

            .linktheme2 {
                border: 1px solid rgba(0, 180, 220, 0.5);
                box-shadow: 0 0 2rem rgba(0, 180, 220, .5) inset
            }

            .linktheme3 {
                color: rgba(255, 255, 255, .9);
                border: 20px solid transparent;
                border-image:
url(http://47.96.137.47:7077/linkbs/images/frames/frame1.png) 5%;
                /*                                          background-image:
url(http://www.imaoda.com/s/img/tpl/border.png) 5%; */
                background: rgba(14, 19, 39, 1);
                /* box-shadow: 0 0 5rem rgb(0, 110, 150) inset */
            }

            .linktheme4 {
                background:
url(http://47.96.137.47:7077/linkbs/images/frames/frame2.png) no-repeat;
                background-size: contain;
            }

        </style>
    </head>
    <body>


        <div class="linkbox">
            <div class="item linkcompany"></div>
            <div class="item linktitle">link title</div>
            <div class="item linktime"></div>
            <div  class="item  linktheme3"  linkid="llll"  style="grid-row-start:
span 4; grid-column-start: span 3;">1</div>
            <div  class="item  linktheme3"  linkid="llll"  style="grid-row-start:
span 8; grid-column-start: span 6;">2</div>
```

```
                    <div class="item linktheme3" linkid="llll" style="grid-row-start:
span 4; grid-column-start: span 3;">3</div>


                    <div class="item linktheme3" linkid="llll" style="grid-row-start:
span 4; grid-column-start: span 3;">4</div>
                    <div class="item linktheme3" linkid="llll" style="grid-row-start:
span 4; grid-column-start: span 3;">6</div>


                    <div class="item linktheme3" linkid="llll" style="grid-row-start:
span 4; grid-column-start: span 3;">7</div>
                    <div class="item linktheme3" linkid="llll" style="grid-row-start:
span 4; grid-column-start: span 3;">8</div>
                    <div class="item linktheme3" linkid="llll" style="grid-row-start:
span 4; grid-column-start: span 3;">9</div>
                    <div class="item linktheme3" linkid="llll" style="grid-row-start:
span 4; grid-column-start: span 3;">9</div>
                </div>
```