

参考资源：

SpreadJS官网：<https://www.grapecity.com.cn/developer/spreadjs>

SpreadJS

API：<http://help.grapecity.com/spread/SpreadSheets11/webframe.html#welcome.html>

SpreadJS Spread.Sheets 中文学习指

南：<https://demo.grapecity.com.cn/spreadjs/TutorialSample/#/>

SpreadJS Spread.Sheets 在线编

辑：<https://demo.grapecity.com.cn/spreadjs/ExcelLikeSample/inspector.html>

原型：

<http://10.129.8.22:100/V3.9/prototype2/#g=1&p=%E4%BA%A7%E5%93%81%E6%96%B9%E6%A1%88>

项目：

ux-tms-setreport

<http://gitlab.beisencorp.com/ux-bi-ui/ux-tms-setreport>

<http://gitlab.beisencorp.com/ux-bi-ui/ux-tms-setreport/blob/master/README.md>

<http://gitlab.beisencorp.com/ux-bi-ui/ux-tms-setreport/blob/master/RUNREADME.md>

由于SpreadJS插件不支持自定义图表数据，只能从excel上读取，所以在原本的excel下隐藏了一个放置模拟数据的excel，图表也是插入到隐藏excel中的

代码整体分三个部分，隐藏excel部分、插入图表部分和配置图表部分

1. 隐藏excel的代码位于/ux-tms-

setreport/app/scripts/views/fixreport/spreadfile/excel-view.js

```

},
onShow: function() {
    var self = this;
    window.self = this;
    // 初始化spread
    this.init();
    // 初始化一个隐藏的excel, 此excel只是为了插入图表相关功能
    this.initChartSheet();
    // 为当前spread实例绑定SpreadJS提供的事件
    this.bindSpreadEvents();
    $(".excelShow").keyup(function(e) {

```

initChartSheet方法用于创建隐藏的excel

bindSpreadEvents方法用于在两个excel都创建好后，绑定所有与图表相关的事件（主要是同步两个excel之间的状态）

2. 插入图表的代码位于excel-view.js和/ux-tms-

setreport/app/scripts/views/fixreport/spreadfile/insert-chart-view.js

```

49  onRender: function() {
50      var insertChartView = new InsertChartView({
51          el: this.ui.insertChart,
52          onInsertChart: this.handleChartInsert.bind(this),
53      });
54      insertChartView.render();
55  },

```

excel-view的onRender中，渲染下拉的图表类型列表

```

245  handleChartInsert: function(chartType) {
246      var sheet = this.spread.getActiveSheet(),
247          chartSheet = this.chartSpread.getActiveSheet(),
248          chartInsertCell = this.getChartInsertCell(sheet),
249          chart = this.createChart(
250              chartSheet,
251              chartType,
252              chartInsertCell
253          );
254      if (chart) {
255          this.setActiveChart(chart);
256          this.openChartSetting(chart);
257      }
258  },

```

handleChartInsert方法是点选某个图表类型后的callback

3. 图形配置的代码位于excel-view.js和/ux-tms-setreport/app/scripts/views/fixreport/spreadfile/chart-setting-view.js

```
320  openChartSetting: function(chart) {
321      if (!this.chartSettingView) {
322          this.chartSettingView = new ChartSettingView({
323              el: this.ui.chartSetting,
324              onViewToggled: this.onChartSettingToggled.bind(this),
325              chartCoordinatesInfo: this.chartCoordinatesInfo,
326          });
327          this.chartSettingView.render();
328      }
329      this.chartSettingView.open(chart);
330  },
```

第一次打开配置面板时，实例化ChartSetting视图，所有图表共享同一个视图

```
31  initialize: function(props) {
32      this.settings = this.initSettings(props.chartCoordinatesInfo);
33      this.chart = null;
34      this.selectionTarget = null;
35      this.jPickers = {};
36      this.onViewToggled = props.onViewToggled;
37      this.model.set({
38          tabList: configs.tabList,
39          chartTypes: configs.chartTypes,
40          fontSizeList: configs.fontSizeList,
41          fontFamilyList: configs.fontFamilyList,
42          dataLabelPositionList: configs.dataLabelPositionList,
43          hasPercentageList: configs.hasPercentageList,
44          hasOutsideEndList: configs.hasOutsideEndList,
45          hasInsideEndList: configs.hasInsideEndList,
46          activeTab: configs.defaultActiveTab,
47          dataSource: {},
48          chartType: null,
49          title: {},
50          dataLabels: {},
51      });
52      this.listenTo(this.model, "change", this.render);
```

this.settings存储所有chart的配置信息（这里配置信息指与后端chartCoordinatesInfo对应的字段）

this.chart存储当前正在配置的chart

与当前页面相关数据都放在model中，model修改，自动出发re-render

父级视图使用到的API有：

1. open 打开配置面板

2. close 关闭配置面板

3. render 渲染配置面板

4. getData 获取所有图表的配置信息

其他function基本都是对控件event的callback