

demo<https://code.fanruan.com/fanruan/demo-show-charts>

67so easy

echarts(DEMO)

1.

x	y	z
...	...	...
...	..	...

2.apiecharts.init(dom)api:instance.setOption(option)

- idea
- srccom.fr.plugin,com.fr.plugin(9.0)com.fr(8.0)
- jscom.fr.plugin
-

echarts

1. :
xxxDataConfigAbstractDataConfig

eg

xyzxyz

readXML&& writeXML&& clone && equals && hashCode && dataSetFields

readExtendedField&writeExtendedFieldkey

DemoDataConfig

```
package com.fr.plugin.demo;

import com.fr.extended.chart.AbstractDataConfig;
import com.fr.extended.chart.ExtendedField;
import com.fr.general.ComparatorUtils;
import com.fr.stable.xml.XMLPrintWriter;
import com.fr.stable.xml.XMLLabelReader;

/**
 * Created by shine on 2018/3/24.
 */
public class DemoDataConfig extends AbstractDataConfig {

    private ExtendedField x = new ExtendedField();
    private ExtendedField y = new ExtendedField();
    private ExtendedField z = new ExtendedField();

    public ExtendedField getX() {
        return x;
    }

    public void setX(ExtendedField x) {
        this.x = x;
    }

    public ExtendedField getY() {
        return y;
    }
```

```

    }

    public void setY(ExtendedField y) {
        this.y = y;
    }

    public ExtendedField getZ() {
        return z;
    }

    public void setZ(ExtendedField z) {
        this.z = z;
    }

    @Override
    protected void readAttr(XMLableReader reader) {
        readExtendedField(x, "x", reader);
        readExtendedField(y, "y", reader);
        readExtendedField(z, "z", reader);
    }

    @Override
    protected void writeAttr(XMLPrintWriter writer) {
        writeExtendedField(x, "x", writer);
        writeExtendedField(y, "y", writer);
        writeExtendedField(z, "z", writer);
    }

    @Override
    public ExtendedField[] dataSetFields() {
        return new ExtendedField[]{
            x,
            y,
            z
        };
    }

    @Override
    public DemoDataConfig clone() throws CloneNotSupportedException {
        DemoDataConfig result = new DemoDataConfig();
        result.setX(this.getX().clone());
        result.setY(this.getY().clone());
        result.setZ(this.getZ().clone());
        return result;
    }

    @Override
    public int hashCode() {
        return super.hashCode() + AssistUtils.hashCode(this.getX(), this.getY(), this.getZ());
    }

    @Override
    public boolean equals(Object obj) {
        return obj instanceof DemoDataConfig
            && AssistUtils.equals(this.getX(), ((DemoDataConfig) obj).getX())
            && AssistUtils.equals(this.getY(), ((DemoDataConfig) obj).getY())
            && AssistUtils.equals(this.getZ(), ((DemoDataConfig) obj).getZ())
            ;
    }
}

```

2.

xxxWrapper.js
egdemoWrapper

demoWrapper

```
demoWrapper = ExtendedChart.extend({
    _init:function (dom, option) {
        var chart = echarts.init(dom);
        chart.setOption(option);
        return chart;
    }
});
```

3.

xxxChartAbstractChartDemoDataConfigextends AbstractChart<DemoDataConfig>

eg:

getChartID ID&& getChartName() && demoImagePath && addJSON jsonoption && requiredJS js&& wrapperName

dataConfig.getX().getValues()

DemoChart

```
package com.fr.plugin.extended.chart.demo;

import com.fr.base.BaseFormula;
import com.fr.extended.chart.AbstractChart;
import com.fr.extended.chart.HyperLinkPara;
import com.fr.extended.chart.export.ExportProcessor;
import com.fr.general.GeneralUtils;
import com.fr.json.JSON;
import com.fr.json.JSONArray;
import com.fr.json.JSONException;
import com.fr.json.JSONFactory;
import com.fr.json.JSONObject;
import com.fr.stable.web.Repository;

import java.util.List;

/**
 * Created by shine on 2018/3/24.
 */
public class DemoChart extends AbstractChart<DemoDataConfig>{

    private static final String ID = "DEMO_CHART";
    private static final String NAME = "DEMO";

    @Override
    protected String getChartID() {
        return ID;
    }

    @Override
    public String getChartName() {
        return NAME;
    }

    @Override
    protected String demoImagePath() {
        return "com/fr/plugin/extended/chart/demo/demo.png";
    }

    @Override
    protected void addJSON(DemoDataConfig dataConfig, JSONObject jsonObject, Repository repo, JSONPara
para) throws JSONException {
```

```

JSONArray array = JSONFactory.createJSON(JSON.ARRAY);

List<Object> xValues = dataConfig.getX().getValues();
List<Object> yValues = dataConfig.getY().getValues();
List<Object> zValues = dataConfig.getZ().getValues();

double maxValue = Double.MIN_VALUE;
for (int i = 0, len = xValues.size(); i < len; i++) {
    maxValue = Math.max(GeneralUtils.objectToNumber(zValues.get(i)).doubleValue(), maxValue);

    array.put(JSONFactory.createJSON(JSON.ARRAY).put(xValues.get(i)).put(yValues.get(i)).put(
zValues.get(i)));
}

JSONObject.put("series", JSONFactory.createJSON(JSON.OBJECT).put("type", "bar3D").put("data",
array)
    .put("bevelSize", 0.2).put("bevelSmoothness", 2).put("shading", "color"));

JSONObject.put("xAxis3D", JSONFactory.createJSON(JSON.OBJECT).put("type", "category"))
    .put("yAxis3D", JSONFactory.createJSON(JSON.OBJECT).put("type", "category"))
    .put("zAxis3D", JSONFactory.createJSON(JSON.OBJECT).put("type", "value"));

JSONObject.put("grid3D", JSONFactory.createJSON(JSON.OBJECT).put("boxWidth", 200).put(
"boxDepth", 80));

JSONObject.put("visualMap", JSONFactory.createJSON(JSON.OBJECT)
    .put("max", maxValue)
    .put("color", JSONFactory.createJSON(JSON.ARRAY).put("#d94e5d").put("#eac736").put(
"#50a3ba")));
}

@Override
protected String[] requiredJS() {
    return new String[]{
        "com/fr/plugin/extended/chart/demo/demowrapper.js",
        "com/fr/plugin/extended/chart/demo/echarts.js",
        "com/fr/plugin/extended/chart/demo/echarts-gl.js"
    };
}

@Override
protected String wrapperName() {
    return "demoWrapper";
}

@Override
protected HyperLinkPara[] hyperLinkParas() {
    return new HyperLinkPara[0];
}

@Override
protected List<StringFormula> formulas() {
    return null;
}

@Override
protected ExportProcessor createExportProcessor() {
    return null;
}
}

```

4. xxxTableDataPaneAbstractExtendedChartTableDataPane extends AbstractExtendedChartTableDataPane<DemoDataConfig>

eg

xyz

fieldLabels && filedComboBoxes && populate && update

DemoTableDataPane

```
package com.fr.plugin.extended.chart.demo;

import com.fr.design.gui.icombobox.UIComboBox;
import com.fr.extended.chart.AbstractExtendedChartTableDataPane;

/**
 * Created by shine on 2018/3/24.
 */
public class DemoTableDataPane extends AbstractExtendedChartTableDataPane<DemoDataConfig>{
    private UIComboBox xComboBox;
    private UIComboBox yComboBox;
    private UIComboBox zComboBox;

    @Override
    protected String[] fieldLabels() {
        return new String[]{
            "X",
            "Y",
            "Z"
        };
    }

    @Override
    protected UIComboBox[] filedComboBoxes() {
        if (xComboBox == null) {
            xComboBox = new UIComboBox();
            yComboBox = new UIComboBox();
            zComboBox = new UIComboBox();
        }
        return new UIComboBox[]{
            xComboBox,
            yComboBox,
            zComboBox
        };
    }

    @Override
    protected void populate(DemoDataConfig dataConf) {
        populateField(xComboBox, dataConf.getX());
        populateField(yComboBox, dataConf.getY());
        populateField(zComboBox, dataConf.getZ());
    }

    @Override
    protected DemoDataConfig update() {
        DemoDataConfig dataConfig = new DemoDataConfig();

        updateField(xComboBox, dataConfig.getX());
        updateField(yComboBox, dataConfig.getY());
        updateField(zComboBox, dataConfig.getZ());

        return dataConfig;
    }
}
```

xxxReportDataPane,AbstractExtendedChartReportDataPane

DemoReportDataPane

```
package com.fr.plugin.extended.chart.demo;

import com.fr.design.formula.TinyFormulaPane;
import com.fr.extended.chart.AbstractExtendedChartReportDataPane;

/**
 * Created by shine on 2018/3/24.
 */
public class DemoReportDataPane extends AbstractExtendedChartReportDataPane<DemoDataConfig> {

    private TinyFormulaPane xPane;
    private TinyFormulaPane yPane;
    private TinyFormulaPane zPane;

    @Override
    protected String[] fieldLabel() {
        return new String[]{
            "X",
            "Y",
            "Z"
        };
    }

    @Override
    protected TinyFormulaPane[] formulaPanes() {
        if (xPane == null) {
            xPane = new TinyFormulaPane();
            yPane = new TinyFormulaPane();
            zPane = new TinyFormulaPane();
        }
        return new TinyFormulaPane[]{
            xPane,
            yPane,
            zPane
        };
    }

    @Override
    protected void populate(DemoDataConfig dataConf) {
        populateField(xPane, dataConf.getX());
        populateField(yPane, dataConf.getY());
        populateField(zPane, dataConf.getZ());
    }

    @Override
    protected DemoDataConfig update() {
        DemoDataConfig dataConfig = new DemoDataConfig();

        updateField(xPane, dataConfig.getX());
        updateField(yPane, dataConfig.getY());
        updateField(zPane, dataConfig.getZ());

        return dataConfig;
    }
}
```

5. xxxAbstractExtentChartProvider
eg
`createChart` xxxChart

Demo

```
package com.fr.plugin.extended.chart.demo;

import com.fr.extended.chart.AbstractChart;
import com.fr.extended.chart.AbstractExtentChartProvider;

/**
 * Created by shine on 2018/3/24.
 */
public class Demo extends AbstractExtentChartProvider {
    @Override
    protected AbstractChart createChart() {
        return new DemoChart();
    }
}
```

xxxUIAbstractExtendedChartUIProvider

eg `getTableDataSourcePane` && `getReportDataSourcePane` && `getIconPath()`

DemoUI

```
package com.fr.plugin.extended.chart.demo;

import com.fr.design.mainframe.chart.gui.data.report.AbstractReportDataContentPane;
import com.fr.extended.chart.AbstractExtendedChartTableDataPane;
import com.fr.extended.chart.AbstractExtendedChartUIProvider;

/**
 * Created by shine on 2018/3/24.
 */
public class DemoUI extends AbstractExtendedChartUIProvider {
    @Override
    protected AbstractExtendedChartTableDataPane getTableDataSourcePane() {
        return new DemoTableDataPane();
    }

    @Override
    protected AbstractReportDataContentPane getReportDataSourcePane() {
        return new DemoReportDataPane();
    }

    @Override
    public String getIconPath() {
        return "com/fr/plugin/extended/chart/demo/icon.png";
    }
}
```

6.

plugin.xml

id: `com.fr.plugin.demoChart` `com.fr.plugin.xxxChart`

name: DEMO

`function-recorder` class `@FunctionRecorderaddJSON@ExecuteFunctionRecord`

`IndependentChartProvider` class `xxxplotID` getChartID

`IndependentChartUIProviderUI` class `xxxUIplotID` getChartID

plugin.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><plugin>
  <id>com.fr.plugin.demoChart</id>
  <name><![CDATA[DEMO]]></name>
  <active>yes</active>
  <version>1.0.0</version>
  <env-version>10.0</env-version>
  <jartime>2018-3-31</jartime>
  <vendor>finereport.shine</vendor>
  <description><![CDATA[demo]]></description>
  <change-notes><![CDATA[]]></change-notes>

  <function-recorder class="com.fr.plugin.demo.DemoChart"/>

  <extra-chart>
    <IndependentChartProvider class="com.fr.plugin.demo.Demo" plotID="DEMO_CHART"/>
  </extra-chart>

  <extra-chart-designer>
    <IndependentChartUIProvider class="com.fr.plugin.demo.DemoUI" plotID="DEMO_CHART"/>
  </extra-chart-designer>
</plugin>
```

function-recorder

```
/**
 * Created by shine on 2018/3/24.
 */
@FunctionRecorder
public class DemoChart extends AbstractChart<DemoDataConfig>{

    @ExecuteFunctionRecord
    @Override
    protected void addJSON(DemoDataConfig dataConfig, JSONObject jsonObject, Repository repo) throws
    JSONException {

        JSONArray array = JSONFactory.createJSON(JSON.ARRAY);

        List<Object> xValues = dataConfig.getX().getValues();
        List<Object> yValues = dataConfig.getY().getValues();
        List<Object> zValues = dataConfig.getZ().getValues();

        .....@FunctionRecorder && @ExecuteFunctionRecord
```

ororor