预览汉化—Web设计器

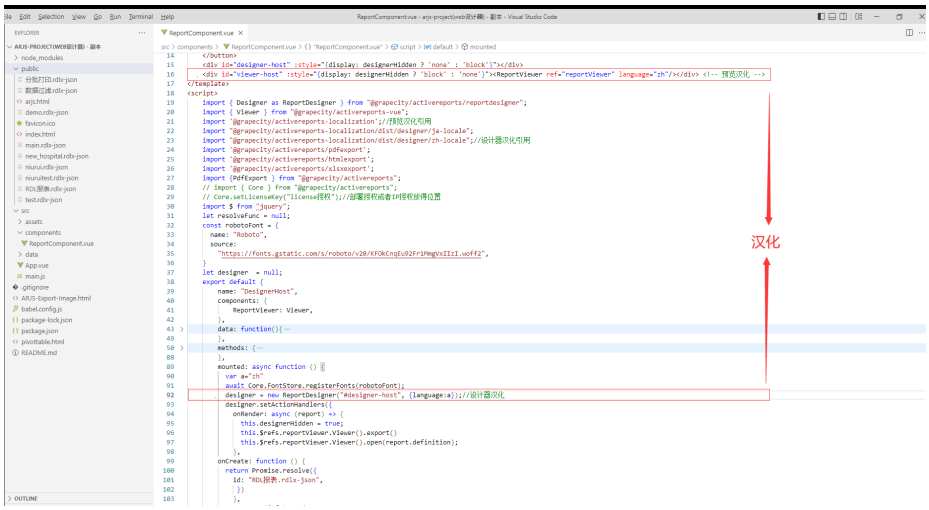
Web设计器默认是英文的，但是在实际应用中，经常需要进行汉化的操作。把工具栏、菜单栏、属性栏进行汉化。并且预览展示的时候也同事需要把预览结果的工具栏进行汉化

那么下面我们就来进行一个Web设计器的汉化操作

1、首先项目中引用Web设计器和预览的汉化文件

```
    <div id="viewer-host" :style="{display: designerHidden ? 'block' : 'none'}"><ReportViewer ref="reportViewer" language="zh"/></div> <!-- 预览汉化 -->
</template>
<script>
    import { Designer as ReportDesigner } from "@grapecity/activereports/reportdesigner";
    import { Viewer } from "@grapecity/activereports-vue";
    import '@grapecity/activereports-localization';//预览汉化引用
    import "@grapecity/activereports-localization/dist/designer/ja-locale";
    import "@grapecity/activereports-localization/dist/designer/zh-locale";//设计器汉化引用
    import '@grapecity/activereports/pdfexport';
    import '@grapecity/activereports/htmlexport';
    import '@grapecity/activereports/xlsxexport';
    import {PdfExport } from "@grapecity/activereports";
    // import { Core } from "@grapecity/activereports";
    // Core.setLicenseKey("license授权");//部署授权或者IP授权放得位置
    import $ from "jquery";
    let resolveFunc = null;
    const robotoFont = {
      name: "Roboto",
```

2、进行汉化配置



3、代码展示

```
<template>
  <div id="controls">
    <div class="line">
      <button id="export-pdf" @click="runPdf()">PDF</button>
    </div>
  </div>
    <button
            type="button"
            class="btn btn-secondary btn-sm col-sm-2 ml-1"
            v-on:click="onDesignerOpen()"
            :style="{display: designerHidden ? 'block' : 'none'}"
    >

    </button>
    <div id="designer-host" :style="{display: designerHidden ? 'none' :
'block'}"></div>
    <div id="viewer-host" :style="{display: designerHidden ? 'block' :
'none'}"><ReportViewer ref="reportViewer" language="zh"/></div> <!--
-->
</template>
```

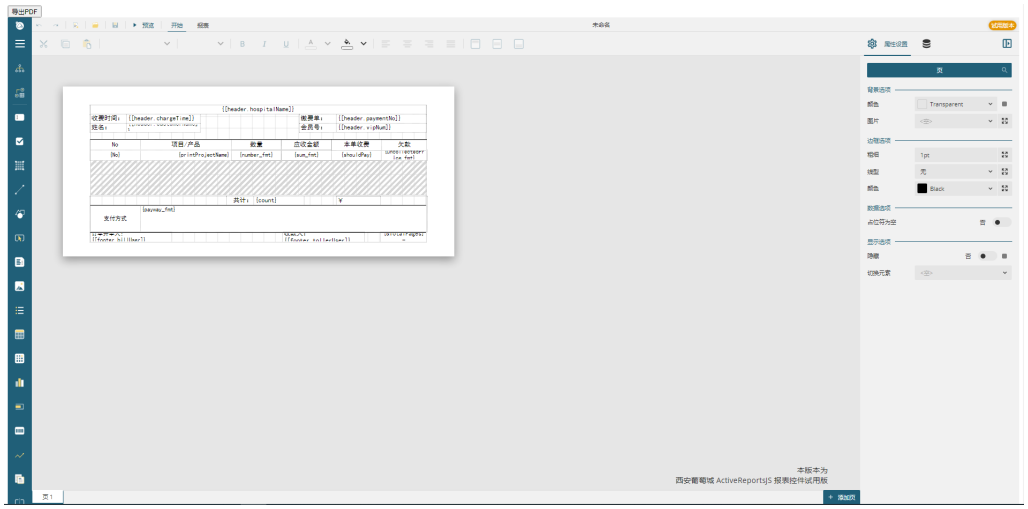
```
<script>
    import { Designer as ReportDesigner } from
"@grapecity/activereports/reportdesigner";
    import { Viewer } from "@grapecity/activereports-vue";
    import '@grapecity/activereports-localization';//
    import
"@grapecity/activereports-localization/dist/designer/ja-locale";
    import
"@grapecity/activereports-localization/dist/designer/zh-locale";//
    import '@grapecity/activereports/pdfexport';
    import '@grapecity/activereports/htmlexport';
    import '@grapecity/activereports/xlsxexport';
    import {PdfExport } from "@grapecity/activereports";
    // import { Core } from "@grapecity/activereports";
    // Core.setLicenseKey("license");//IP
    import $ from "jquery";
    let resolveFunc = null;
    const robotoFont = {
      name: "Roboto",
      source:

"https://fonts.gstatic.com/s/roboto/v20/KFOkCnqEu92Fr1MmgVxIIzI.woff2",
    }
    let designer  = null;
    export default {
        name: "DesignerHost",
        components: {
            ReportViewer: Viewer,
        },
        data: function(){
            return {
                designerHidden: false,
                counter: 0,
                reportStorage: new Map(),
            }
        },
        methods: {
            onDesignerOpen(){
                this.designerHidden = false;
            },
            onSelectReport(reportId) {
                if (resolveFunc) {
                    $("#dlgOpen").modal("hide");
                    resolveFunc({ definition:
this.reportStorage.get(reportId), id: reportId, displayName: reportId
});
                    resolveFunc = null;
                }
            },
            //
            download(fileName, blob) {
                const link =document.createElement('a');
                link.href = URL.createObjectURL(blob);
                link.download = fileName;
                link.click();
                link.remove();
                URL.revokeObjectURL(link.href);
```

```
            },
            runPdf() {
                let settings = {
                    info: {
                      title: 'test',
                      author: 'GrapeCity inc.',
                    },
                    pdfVersion:"1.7"
                }
                let pageReport = new Core.PageReport();
                pageReport.load('demo.rdlx-json').then(function() {
                    return pageReport.run();
                  }).then(function(pageDocument) {
                      return PdfExport.exportDocument(pageDocument,
settings)
                  }).then(function(result) {
                      console.log(result)
                      result.download('arjs-pdf')
                  });
            }
        },
        mounted: async function () {
          var a="zh"
          await Core.FontStore.registerFonts(robotoFont);
          designer = new ReportDesigner("#designer-host",
{language:a});//
          designer.setActionHandlers({
            onRender: async (report) => {
              this.designerHidden = true;
              this.$refs.reportViewer.Viewer().export()
              this.$refs.reportViewer.Viewer().open(report.definition);
            },
        onCreate: function () {
          return Promise.resolve({
            id: "RDL.rdlx-json",
             })
            },
            onSave: (info) => {
              console.log(info);
              const reportId = info.id || `NewReport${this.counter +
1}`;
              //
              const fileName = `NewReport${this.counter + 1}.rdlx-json`;
              const blob = new Blob([JSON.stringify(info.definition)],
{type: "application/json"})
              this.download(fileName, blob);
              this.counter++;
              return Promise.resolve({displayName: reportId});

            },
            onOpen: function () {
                    // const ret = new Promise(function (resolve) {
                    //     resolveFunc = resolve;
                    //     this.reportIds = this.reportStorage.keys();
                    //     $("#dlgOpen").modal("show");
                    // });
                    // return ret;
```

```
                }
            });
            //test.rdlx-json


            designer.setReport({id: ".rdlx-json"});//

           //designer.setReport({definition: report.define});//json
          },
    };
</script>

<style scoped>
    #designer-host {
        margin: 0 auto;
        width: 100%;
        height: 100vh;
    }
    #viewer-host {
        margin: 0 auto;
        width: 100%;
        height: 100vh;
    }
```

```
</style>
```

4、预览结果



5、demo