

JS Viewer API

初始化Viewer 的选项设置

以下方式是在初始化Viewer 中会被执行的

action

说明: JSViewer 实现报表导航, 如钻取, 跳转URL, 跳转书签, 折叠展开控件时所执行的回调函数。

类型: function(actionType, actionParams)回调函数

示例: :

```
action: (actionType, actionParams) => console.log('Action type: ' + actionType + '; Action parameters: ' + actionParams)
```

animation

说明: 报表中预览中的动画选项

类型: 该对象和可用的属性如下

- loadChart
Property: enabled
Type: Boolean (default 'false')
 - hoverChart
Property: enabled
Type: Boolean (default 'false')
 - hoverTable
Property: enabled
Type: Boolean (default 'false')
- Property: backgroundColor
Type: String
Accepted Value: Any
- Property: textColor
Type: String
Accepted Value: Any

示例代码:

```
animation: {  
  loadChart: {  
    enabled: true  
  },  
  hoverChart: {  
    enabled: true  
  },  
  hoverTable: {  
    enabled: true,  
    backgroundColor: 'LightGray',  
    textColor: 'DarkGray'  
  }  
}
```

availableExports

说明: 是用于设置所提供的导出文件的格式, 在不设置该选项时候, JSViewer 默认是支持所有格式导出, PDF, Excel, Word, JSON, XML, CSV, 图像

类型: Array(数组)

示例:

```
availableExports: ['Xml', 'Pdf']
```

displayMode

说明: 设置Viewer 是单页显示还是连续一页加载

类型: String (字符串)

可识别的值: 'single', 'continuous'

示例:

```
displayMode: 'Continuous'
```

documentLoaded

说明: 当报表文档从服务器中完全加载完成后执行的回调函数。

类型: function() 回调函数

示例:

```
documentLoaded: () => console.log('The document is loaded entirely on the server')
```

返回值: Void 空

element

说明 JQuery 选择器, 用于绑定JSViewer 的宿主对象。

类型: String

示例: :

```
element: '#viewerContainer'
```

error

说明 :

当加载报表发生错误时候, 所执行的的错误提示回调函数。如果回调函数返回的是True默认的错误提示面板就不会出现了。错误参数是一个包含message属性的对象, 开发人员可自定义错误提示信息。

类型: function(error) 回调函数

示例: 隐藏默认的错误面板

```
error: (error) => {  
    if (error.message) {  
        // show error message.  
        alert("Internal error! Please ask administrator.");  
        // do not show default error message.  
    }  
}
```

reportID

说明: 指定JS Viewer 加载报表的标识 (或路径)

类型: String

示例:

```
reportID: 'AnnualReport.rdlx'
```

reportLoaded

说明: JSViewer 获取关于请求报表时所执行的回调函数, 该回调函数会包含 reportInfo 对象, 包含报表名称目录信息, 参数信息, 以及渲染报表结果的链接。

类型: function(reportInfo)

Example:

```
reportLoaded: (reportInfo) => console.log('The report ' + reportInfo.name + ' was successfully loaded!')
```

reportParameters

说明: {name, value} 键值对的数组, 用于为报表传递对应的参数值。

类型: Array (数组)

示例:

```
reportParameters: [{ name: 'ReportParameter1', values: ['1']}]
```

reportService

说明: 连接 WebAPI 的设置

类型: 包含URL 及可选的securityToken 属性的对象

示例:

```
reportService: {  
    url: 'http://example.com/api/reporting',  
    securityToken: '42A9CD80A4F3445A9BB60A221D042FCC',  
};
```

可调用的API 方法及属性

方法

backtoParent

说明: 常用于报表钻取, 报表钻取到子报表时, 使用该方法可以让Viewer 返回到主报表。

类型: backToParent()Void

示例:

```
viewer.backToParent()
```

返回值: Void

create

说明: 新建一个新的Viewer 实例, 并在指定的 DOM 元素中加载它。

Syntax: create()Void

Parameters:

- params

Example:

```
const viewer = GrapeCity.ActiveReports.JSViewer.create({  
    element: '#root',  
    reportID: 'AnnualReport.rdlx',  
    availableExports: ['Xml', 'Pdf'],  
    // other properties
```

```
});
```

Return Value: Void

destroy

说明：从当前元素中删除Viewer

Syntax: `destroy()`Void

Example:

```
viewer.destroy()
```

Return Value: Void

export

说明：导出当前报表

Syntax: `export(exportType, callback, saveAsDialog, settings)`Void

Parameters:

- `exportType`: 指定导出格式.
- `settings`: `RenderingExtensions`所提供的导出设置
-

注意：区域报表的导出设置在使用rendering extensions时候不可用,。页面报表和RDL 报表使用Export Filter 导出PDF时不可用。

示例:

```
viewer.export('Pdf', downloadReport, true, { Title: '1997 Annual Report' })
```

Return Value: Void

getTOC

说明：获取报表目录 TOC.

Syntax: `viewer.getToc()`

Example:

```
console.log(viewer.getToc())
```

Return Value: any

goToPage

说明：指定Viewer 展示特定页的报表

Syntax: `goToPage(number, offset, callback)`Void

Parameters:

- `number`: The number of pages to go to.

Example:

```
viewer.goToPage(1)
```

Return Value: Void

openReport

说明：加载指定的报表文件报表

Syntax: `openReport(reportID: string, reportParameters?: Array<Parameter>): void`

Parameters:

- reportID: The id of the report.

Example:

```
viewer.openReport('Invoice.rdlx')
```

Return Value: Void

Print

说明: 打印当前展示的报表.

Syntax: `print()`Void

Example:

```
viewer.print()
```

Return Value: Void

Refresh

说明: 刷新报表预览页面

Syntax: `option(name, [value])`Object

Example:

```
viewer.refresh()
```

Return Value: Void

Version

说明: 获取当前JSViewer 的版本。

Syntax: `viewer.version`

Example:

```
console.log(viewer.version)
```

Return Value: string

renderMode

说明: 获取或设置JS Viewer 的渲染模式

Type: String

Accepted Value: 'Galley', 'Paginated'

Example:

```
viewer.renderMode = 'Galley';
```

toolbar.addItem

说明: 为工具栏添加自定义按钮

Type: item: ToolbarItem

Example:

```
ToolbarItem = {  
    key: string;  
    iconCssClass?: string;  
    icon?: Icon;  
    text?: string;
```

```

    title?: string;

    checked?: boolean;

    enabled?: boolean;

    action?: (toolbarItem) => function;

    onUpdate?: (args: ChangedEventArgs, toolbarItem) => function;
}

```

```

Icon = FontIcon | SVGIcon;

```

```

SVGIcon = {
    type: 'svg';
    content: any; //(string | JSX.element)
    size?: Size;
    rotate?: '90' | '180' | '270';
};

```

```

FontIcon = {
    type: 'font';
    iconCssClass: string;
    fontSize?: string;
}

```

```

var pdfExportButton = {
    key: '$pdfExportButtonKey',
    iconCssClass: 'mdi mdi-file-pdf',
    enabled: true,
    action: function(item) {
        console.log('Export to PDF function works here');
    },
    onUpdate: function(arg, item) {
        console.log('Something in viewer was updated, check/update button state here');
    }
};

```

```

//OR button with svg as icon

```

```

var icon = `

```

```

`;

```

```

var svgPdfExportButton = {

```

```

    key: '$pdfExportButtonKey',
    icon: {type: 'svg', content: icon},
    enabled: true,
    action: function(item) {
        console.log('Export to PDF function works here');
    },
    onUpdate: function(arg, item) {
        console.log('Something in viewer was updated, check/update button state here');
    }
}

```

```

viewer.toolbar.addItem(pdfExportButton);
//or
viewer.toolbar.addItem(svgPdfExportButton);

```

toolbar.updateItem

说明： 更新设置某一工具栏按钮的布局，功能或状态

Type: String: key, ToolbarItem item

Example:

```

var xlsxExportButton = {
    key: '$xlsxExportButtonKey',
    iconCssClass: 'mdi mdi-file-pdf',
    enabled: true,
    action: function(item) { console.log('Export to XLSX function works here'); },
    onUpdate: function(arg, item) { console.log('Something in viewer was updated, check/update button state here'); }
};

viewer.toolbar.updateItem('$pdfExportButtonKey', xlsxExportButton); //from now button will behave as described in new
properties

```

toolbar.removeItem

说明： 删除工具栏的自定义的按钮。如果要删除默认的工具栏按钮，需要调用 `updateLayout` 函数

Type: String: key

Example:

```

viewer.toolbar.removeItem('$pdfExportButtonKey');

```

toggleSidebar

说明： 设置侧边栏可见或隐藏。如果不传递值，默认是隐藏。

Type: boolean

Example:

```

viewer.sidebar.toggle(true); //will hide it viewer.toggleSidebar(false); //will show it

```

可用属性

currentPage

说明：获取当前加载页码

Syntax: viewer.currentPage

Example:

```
console.log(viewer.currentPage())
```

Return Value: An integer representing currently displayed page number.

pageCount

说明：获取当前报表的总页数

Syntax: viewer.pageCount

Example:

```
console.log(viewer.pageCount())
```

Return Value: An integer representing page count.

相关资源：

注意

如需寻求在线帮助，请访问 [ActiveReports 求助中心](#)

如需了解更多ActiveReports产品特性，请访问 [ActiveReports 官方网站](#)

ActiveReports 官方技术交流群：109783140

下载产品体验产品功能：<http://www.gcpowertools.com.cn/products/download.aspx?pid=16>