

UI

UI

provider			&
bi.provider.conf.list.report.item	<pre>addReportOperatorItems(items: any[], position: number = 0)</pre>	tab	<p>tab</p> <p>blocked URL e.g. 3</p> <pre>BI.config('bi.provider.conf.list.report.item', (provider) => { provider.addReportOperatorItem([{ text: "", cls: 'inject-font', handler: function () { const refresh = this.attr("refresh"); // const { id } = this.attr("reportInfo"); // // do something refresh(); // }], 3) })</pre>
bi.provider.conf.list.widget.item	<pre>addWidgetOperatorItem(items: any[] , position: number = 0)</pre>	tab	<p>tab</p> <p>blocked URL e.g. 3</p> <pre>BI.config('bi.provider.conf.list.widget.item', (provider) => { provider.addWidgetOperatorItem({ text: "", cls: 'inject-font', handler: function () { const refresh = this.attr("refresh"); // const { id } = this.attr ("widgetInfo"); // // do something refresh(); // }], 3) })</pre>
bi.provider.conf.toolbars_analysis	<pre>addToolItems(items: any[])</pre>		<p>blocked URL</p> <p>e.g.</p> <pre>BI.config('bi.provider.conf.toolbars_analysis', (provider) => { provider.addTooItems([{ type: "bi.button", text: "", handler: () => { BI.Msg.alert('1') } }]); })</pre>

bi.provider.conf.toolbars.analysis_preview	addToolItems(items: any[])	UI	blocked URL e.g. <pre> BI.config('bi.provider.conf.toolbars.analysis_preview', (provider) => { provider.addToolItems([{ type: "bi.button", text: "", handler: () => { BI.Msg.alert('1') } }]); }); </pre>
bi.provider.conf.toolbars	addToolItems(items: any[])		blocked URL e.g. <pre> BI.config('bi.provider.conf.toolbars', (provider) => { provider.inject([{ type: 'bi.button', value: 10, // value openLayerType: 'bi.xxx.layer', // }]); }); </pre>
bi.provider.conf.pack.table.toolbars	inject() filter()		blocked URL e.g. <pre> BI.config('bi.provider.conf.toolbars', (provider) => { provider.inject([{ type: 'edit_custom_table', content: "" }, {key1}]); provider.filter("my-key"); }); </pre>
bi.provider.detail.widget.down_list_item	insertBefore(item: Obj) push(items: Obj)		blocked URL e.g. <pre> BI.config('bi.provider.detail.widget.down_list_item', (provider) => { provider.insertBefore({ type: 'bi.icon_button', cls: 'icon-font', handler: function(){ console.log('1'); } }); }); </pre>
bi.provider.design.widget_dashboard_toolbar.more_item	insertBefore(items: Obj)		blocked URL e.g. <pre> BI.config("bi.provider.design.widget_dashboard_toolbar.more_item", (provider) => { provider.push([{ value: De xtype, cls: "data-sentinel-font", text: BI.i18nText("Data- </pre>

	<code>push(items: Obj)</code>	<pre> Sentinel_Create_Sentinel"), title: BI.i18nText("Data- Sentinel_Create_Sentinel"), operator: function() { //do something } }) }); </pre>
<code>bi.provider.dashboard_toolbar</code>	<code>addToolItems(items: any[])</code>	<p>blocked URL</p> <p>e.g.</p> <pre> BI.config('bi.provider.dashboard_toolbar', (provider) => { provider.addToolItems({ type: 'bi.icon_button', cls: 'icon-font', handler: function(){ console.log('1'); } }); }); </pre>
<code>bi.provider.show.dashboard_toolbar</code>	<code>addToolItems(items: any[])</code>	<p>blocked URL</p> <pre> BI.config('bi.provider.show.dashboard_toolbar', (provider) => { provider.addToolItems({ type: 'bi.icon_button', cls: 'icon-font', handler: function(){ console.log('1'); } }); }); </pre>
<code>bi.provider.design.widget_dashboard_toolbar</code>	<code>addToolItems(items: any[])</code>	<p>blocked URL</p> <pre> BI.config('bi.provider.design.widget_dashboard_toolbar', (provider) => { provider.addToolItems({ type: 'bi.icon_button', cls: 'icon-font', handler: function(){ console.log('1'); } }); }); </pre>

bi.provider.design.widget.detail

injectCollapseWidget(func: Function)

BI

```
BI.config('bi.provider.design.widget.detail', function (provider) {
  provider.injectCollapseWidget(function (obj) {
    return {
      el: {
        type: 'bi.absolute',
        css: {
          'z-index': 1,
        },
        items: [
          {
            el: {
              type: BI.Layout xtype,
              cls: 'bi-plugin-push-button-
background',
              width: 90,
              height: 25,
            },
            top: 32,
            left: -32,
            bottom: 0,
            right: 0,
          },
          {
            el: {
              type: "bi.icon_change_button",
              iconCls: 'next-page-h-font',
              listeners: [{
                eventName: 'EVENT_CHANGE',
                action: function () {
                  this.setIcon(this.
isSelected() ? 'pre-page-h-font' : 'next-page-h-font');
                  if (this.isSelected()) {
                    obj.expand();
                  } else {
                    obj.collapse();
                  }
                }
              }],
              mounted: function () {
                obj.collapse();
              }
            },
            top: 0,
            right: 5,
            bottom: 0,
            left: 0,
          },
        ],
      },
      top: 0,
      right: -25,
      width: 25,
      height: 90,
    }
  });
});
```