Create a Flex Widget

In this example we will create Flex Widget of type Card, that displays info of a Baja Component.

1) Create a JS file in ***src/rc***:

src/rc/flexwidget/MyFlexWidget.js

```
import { CARD } from "fw-types";
import FWProps, { parseToDefaultProps, parseToPropTypes } from "fw-props";
import useOrdResolve from "btibCore/hook/niagara/useOrdResolve";
import { isComponent } from "btibCore/util/compUtils";
import React from "react";

// React Component
function MyFlexWidget(props) {
  const { componentOrd, background, foreground, flexProps } = props;
  const base = flexProps.value;

  // In BajaScript the resolution of an ORD in asynchrone
  const { value, isLoading, error } = useOrdResolve({ ord: componentOrd, base });

  if (isLoading) {
    return <div>Loading...</div>;
  }

  if (error || !isComponent(value)) {
    return <div>{error.toString() || "The ORD doesn't target a baja:Component"}</div>;
  }

  return (
    <div style={{ background, color: foreground }}>
      <h3>Component Info:</h3>
      <ul>
        <li>Name: {value.getName()}</li>
        <li>Type: {value.getType().toString()}</li>
        <li>Handle: {value.getHandle()}</li>
        <li>Nav ORD: {value.getNavOrd().toString()}</li>
        <li>Permissions: {value.getPermissions().toString()}</li>
      </ul>
    </div>
  );
}

MyFlexWidget.fwTypes = CARD;

MyFlexWidget.fwProps = {
  componentOrd: FWProps.ord.isRequired,
  background: FWProps.color("#ebebeb"),
  foreground: FWProps.color("#4d4d4d"),
};

MyFlexWidget.defaultProps = parseToDefaultProps(MyFlexWidget.fwProps);

MyFlexWidget.propTypes = parseToPropTypes(MyFlexWidget.fwProps);

export default MyFlexWidget;
```

The types of the widget are defined in the ***fwTypes*** property and the properties of the widget, that will be used in the React Component, are defined in ***fwProps*** property.

We use **fw-props** lib to define a Flex Widget property. Many types are available : string, boolean, number, array...

The ***flexProps*** property is injected in all Flex Widgets and contains data to create our view:

- *flexProps.value*: the Baja Component on which the view is agent.
- *flexProps.isEditorMode*: indicates if the Flex View is in editor mode.
- *flexProps.portalRoot*: allow to use ***ReactDOM.createPortal*** to show a Modal for example.

- ...

2) Define the Flex Widget in ***active.flex.js***:

src/rc/active.flex.js

```
define({
    widgets: {
          // Without src folder and the .js extension
      MyFlexWidget: "nmodule/myModule/rc/flexwidget/MyFlexWidget"
    },
 });
```

This allow the Flex View to load the widget.

3) You can add your widget in the module palette to allow drag & drop:

module.palette

```
...
<p n="MyFlexWidget" m="bv=btibVision" t="bv:FlexWidget">
    <p n="id" f="r" t="b:String" v="myModule:MyFlexWidget"/>
    <!-- <p n="icon" f="hr" t="b:Icon" v="module://myModule/resource/icon/x16/flexwidget/MyFlexWidget.png"/> -->
</p>
...
```