com.fr.design.fun.ToolbarItemProvider

-
-
-
-
-
-
-
-
-

web//JSJS

1.ExportOperateProvider

2.

3.

ToolbarItemProvider.java

```
package com.fr.design.fun;

import com.fr.design.mainframe.JTemplate;
import com.fr.form.ui.Widget;
import com.fr.stable.Filter;
import com.fr.stable.fun.mark.Mutable;

/**
 * @author : focus
 * @since : 8.0
 * web
 */
public interface ToolbarItemProvider extends Mutable, Filter<JTemplate> {

    String XML_TAG = "ToolbarItemProvider";

    int CURRENT_LEVEL = 1;


    /**
     * webcom.fr.form.ui.ToolBarMenuButton  com.fr.form.ui.ToolBarButton;
     *
     * @return
     */
    Class<? extends Widget> classForWidget();

    /**
     * web
     *
     * @return
     */
    String iconPathForWidget();

    /**
     * web
     *
     * @return
     */
    String nameForWidget();

    /**
     *  or cpt
     * JTemplate
     * @return true, false
     */
    @Override
    boolean accept(JTemplate template);

}
```

ToolBarButton.java

```
package com.fr.form.ui;

import com.fr.data.core.DataCoreXmlUtils;
import com.fr.form.event.Listener;
import com.fr.general.data.Condition;
import com.fr.js.JavaScript;
import com.fr.js.JavaScriptImpl;
import com.fr.json.JSONException;
import com.fr.json.JSONObject;
import com.fr.script.Calculator;
import com.fr.stable.core.NodeVisitor;
import com.fr.stable.web.Repository;
import com.fr.stable.xml.XMLPrintWriter;
```

```
import com.fr.stable.xml.XMLableReader;

/*
 * TODO ToolBarWidget(afterload) Widget
 * Think in javaToolBarEmail, NextToolBar
 */
public abstract class ToolBarButton extends Button {
    // richer:,PrivilegeCondition
    protected Condition condition;

    public ToolBarButton(String text) {
        super(text);
        this.setEnabled(true);
    }

    public ToolBarButton(String text, String iconName) {
        super(text, iconName);
        // alex:disabled
        this.setEnabled(true);
    }

    public String widgetName() {
        return getClass().getSimpleName();
    }

    public Condition getCondition() {
        return this.condition;
    }

    public void setCondition(Condition condition) {
        this.condition = condition;
    }

    public Listener[] createListeners(Repository repo) {
        return new Listener[]{
                new Listener(AFTERINIT, initAction(repo)),
                new Listener(EVENT_CLICK, clickScript(repo)),
                new Listener(
                        WebContentUtils.getContentPanel(repo), WebContentUtils.EVENT_STARTLOAD, new
JavaScriptImpl(
                        getDisableAction()
                )
                ), new Listener(
                WebContentUtils.getContentPanel(repo), WebContentUtils.EVENT_AFTERLOAD, new JavaScriptImpl(
                onContentPanelAfterLoad(repo)
        )
        )
        };
    }

    /**
     *
     * @param repo Repository
     * @return
     * clickScript
     */
    @Deprecated
    protected abstract JavaScriptImpl clickAction(Repository repo);

    protected JavaScript clickScript(Repository repo){
        return clickAction(repo);
    }

    protected JavaScript initAction(Repository repo) {
        return new JavaScriptImpl("this.disable();");
    }

    /*
     * ContentPanel.afterload
     */
    protected String onContentPanelAfterLoad(Repository repo) {
```

```
        return getEnableAction();
    }

    @Override
    public JSONObject createJSONConfig(Repository repo, Calculator c, NodeVisitor nodeVisitor) throws
JSONException {
        JSONObject jo = super.createJSONConfig(repo, c, nodeVisitor);
        jo.put("widgetName", widgetName());
        return jo;
    }

    public void readXML(XMLableReader reader) {
        super.readXML(reader);
        if (reader.isChildNode()) {
            if (Condition.XML_TAG.equals(reader.getTagName())) {
                ToolBarButton.this.condition = DataCoreXmlUtils.readCondition(reader);
            }
        }
    }

    public void writeXML(XMLPrintWriter writer) {
        super.writeXML(writer);
        if (condition != null) {
            DataCoreXmlUtils.writeXMLCondition(writer, condition);
        }
    }

    public boolean equals(Object obj) {
        if (obj == null || !(obj instanceof ToolBarButton)) {
            return false;
        }
        return super.equals(obj);
    }
}
```

FR	8.0		
FR	9.0		
FR	10.0		
BI	3.6		
BI	4.0		
BI	5.1		
BI	5.1.2		
BI	5.1.3		

plugin.xml

```
<extra-designer>
        <ToolbarItemProvider class="your class name"/>
</extra-designer>
```

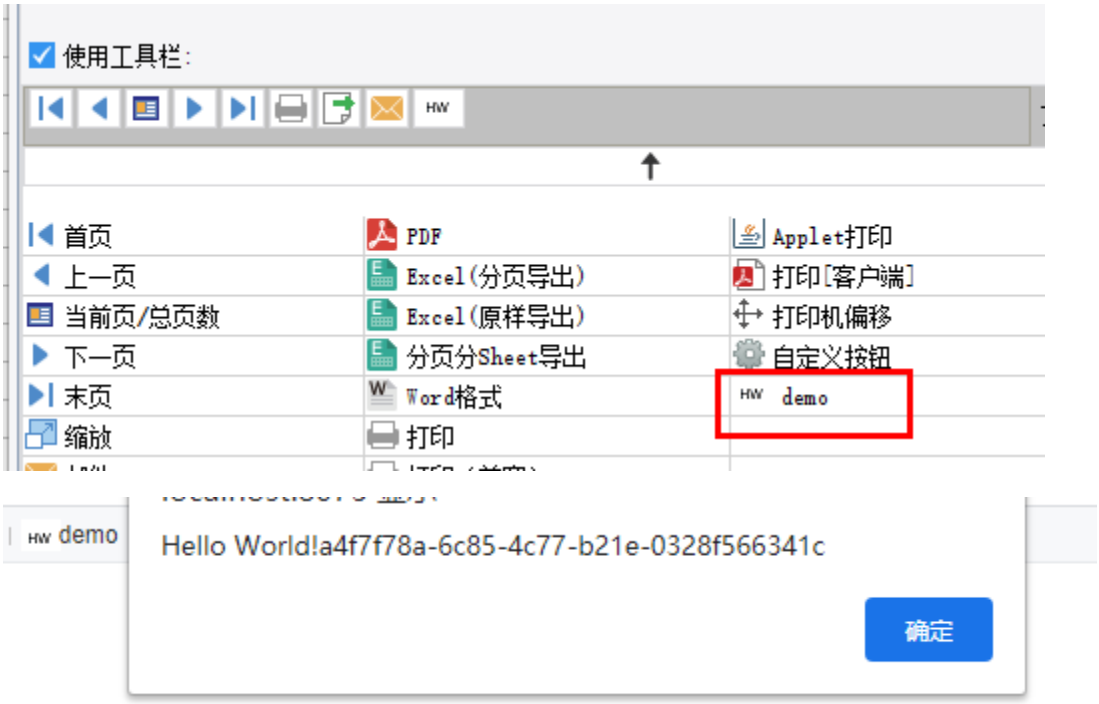
ExtraDesignClassManager#getWebWidgetOptionsToolbarItemProvidercpt

ToolBarButtonsuperToolBarButton(,) SundryKit.loadToolbarIcon(,);

ToolBarButton#clickAction(Repository repo)JavaScriptImpl new JavaScriptImpl("JS ");RepositoryJSsession

web

demodemo-toolbar-item-provider



demo-export-xml