

20211

1

2

3

4FR.downloadHyperlink( taskId ),typedownload

5web

FR.servletURL+"?op=fr\_attach&cmd=ah\_info&id="+taskId

{ success:true, id:"\$TASKID", filename: "()", size:"byte" } { success:false, error\_msg:"/" }

FR.servletURL+"?op=fr\_attach&cmd=ah\_download&id="+taskId

JS



plugin.xml

**plugin.xml**

```
<dependence>  
  <item type="plugin" key="com.tptj.extra.hg.attachment.service" />  
</dependence>
```

plugin.xml3JAR

demo

// 3Object

**FileDefine.java**

```

package com.tptj.bridge.hg.file.load.demo;

import com.fr.io.context.info.GetConfig;
import com.fr.stable.FCloneable;
import com.tptj.tools.hg.file.operator.dynamics.Column;
import com.tptj.tools.hg.file.operator.utils.BaseUtils;
import com.tptj.tools.hg.xml.fun.Config;

/**
 * @author
 * @version 10.0
 * Created by on 2021-01-07
 **/
public class FileDefine implements FCloneable {

    public static final String KEY_PATH = "path";
    public static final String KEY_FILE = "file";
    public static final String KEY_RENAME = "rename";

    @Config(KEY_PATH)
    @Column(title = "", idx = 0)
    private Object path;

    @Config(KEY_FILE)
    @Column(title = "", idx = 1)
    private Object file;

    @Config(KEY_RENAME)
    @Column(title = "", idx = 2)
    private Object rename;

    @GetConfig(KEY_PATH)
    public Object getPath() {
        return path;
    }

    public void setPath(Object path) {
        this.path = path;
    }

    @GetConfig(KEY_FILE)
    public Object getFile() {
        return file;
    }

    public void setFile(Object file) {
        this.file = file;
    }

    @GetConfig(KEY_RENAME)
    public Object getRename() {
        return rename;
    }

    public void setRename(Object rename) {
        this.rename = rename;
    }

    @Override
    public Object clone() throws CloneNotSupportedException {
        FileDefine obj = (FileDefine)super.clone();
        obj.file = BaseUtils.clone( file );
        obj.path = BaseUtils.clone( path );
        obj.rename = BaseUtils.clone( rename );
        return obj;
    }
}

```

/FCloneable

@ConfigXML

@Column

GetConfig

BaseUtils.clone

JARFileDownloadHyperlink

## DownloadHyperlink.java

```
package com.tptj.bridge.hg.file.load.demo;

import com.fr.log.FineLoggerFactory;
import com.fr.plugin.transform.ExecuteFunctionRecord;
import com.fr.plugin.transform.FunctionRecorder;
import com.fr.stable.StableUtils;
import com.fr.stable.StringUtils;
import com.tptj.tools.hg.file.operator.bridge.FileDownloadHyperlink;
import com.tptj.tools.hg.file.operator.utils.DynamicsConfigUtils;
import com.tptj.tools.hg.xml.fun.Config;

import java.io.FileInputStream;
import java.io.InputStream;

/**
 * @author
 * @version 10.0
 * Created by on 2021-01-04
 */
@FunctionRecorder
public class DownloadHyperlink extends FileDownloadHyperlink {

    @Config("file")
    private FileDefine file;

    public FileDefine getFile() {
        return file;
    }

    public void setFile(FileDefine file) {
        this.file = file;
    }

    @Override
    public String getOutputFilename() {
        String val = DynamicsConfigUtils.getConfig( file.getRename() );
        if( StringUtils.isEmpty(val) ){
            val = DynamicsConfigUtils.getConfig( file.getFile() );
        }
        return val;
    }

    @Override
    @ExecuteFunctionRecord
    public InputStream load() {
        try{
            return new FileInputStream( StableUtils.pathJoin(
                (String) DynamicsConfigUtils.getConfig( file.getPath() ),
                (String)DynamicsConfigUtils.getConfig( file.getFile() ) ) );
        }catch(Exception e){

        }
        return null;
    }

    @Override
    public Object clone() throws CloneNotSupportedException {
        DownloadHyperlink item = (DownloadHyperlink)super.clone();
        try{
            item.file = ( FileDefine )file.clone();
        }catch(Exception e){
            FineLoggerFactory.getLogger().error(e,e.getMessage());
        }
        return item;
    }
}
```

FileDefine file@ConfigXMLxml

getOutputFilename

load

API DynamicsConfigUtils.getConfig

### DownloadHyperlinkPane.java

```
package com.tptj.bridge.hg.file.load.demo;

import com.fr.design.beans.BasicBeanPane;
import com.fr.design.layout.FRGUIPaneFactory;
import com.tptj.tools.hg.file.operator.design.DynamicsListPane;
import com.tptj.tools.hg.file.operator.utils.DynamicsPaneUtils;

import java.awt.*;

/**
 * @author
 * @version 10.0
 * Created by on 2021-01-04
 */
public class DownloadHyperlinkPane extends BasicBeanPane<DownloadHyperlink> {

    DynamicsListPane<FileDefine> editor ;

    public DownloadHyperlinkPane(){
        setLayout(FRGUIPaneFactory.createM_BorderLayout());
        editor = DynamicsPaneUtils.createListPane(FileDefine.class);
        add( editor, BorderLayout.CENTER );
    }

    @Override
    public void populateBean( DownloadHyperlink link ) {
        if( null == link ){
            return;
        }
        editor.populateBean( link.getFile() );
    }

    @Override
    public DownloadHyperlink updateBean() {
        DownloadHyperlink rt = new DownloadHyperlink();
        rt.setFile( editor.updateBean() );
        return rt;
    }

    @Override
    protected String title4PopupWindow() {
        return "demo";
    }
}
```

FRbeanpaneAPI DynamicsPaneUtils.createListPane

@Column@ConfigFCloneable

## DownloadHyperlinkBridge.java

```
package com.tptj.bridge.hg.file.load.demo;

import com.fr.design.fun.impl.AbstractHyperlinkProvider;
import com.fr.design.gui.controlpane.NameObjectCreator;
import com.fr.design.gui.controlpane.NameableCreator;
import com.fr.general.ComparatorUtils;

/**
 * @author
 * @version 10.0
 * Created by on 2021-01-04
 */
public class DownloadHyperlinkBridge extends AbstractHyperlinkProvider {
    //
    private NameableCreator nameableCreator = new NameObjectCreator("demo",
        DownloadHyperlink.class, DownloadHyperlinkPane.class);

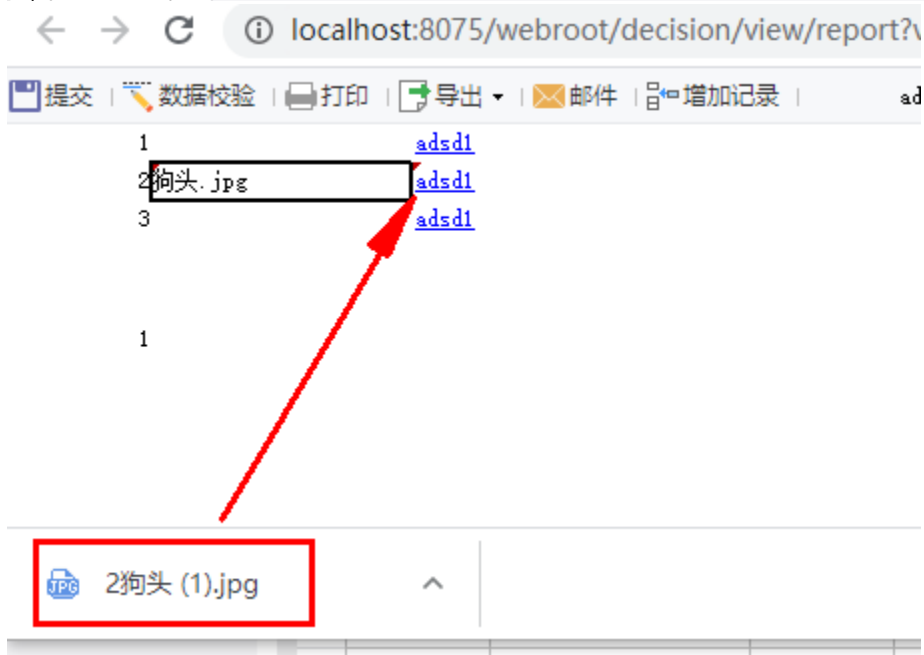
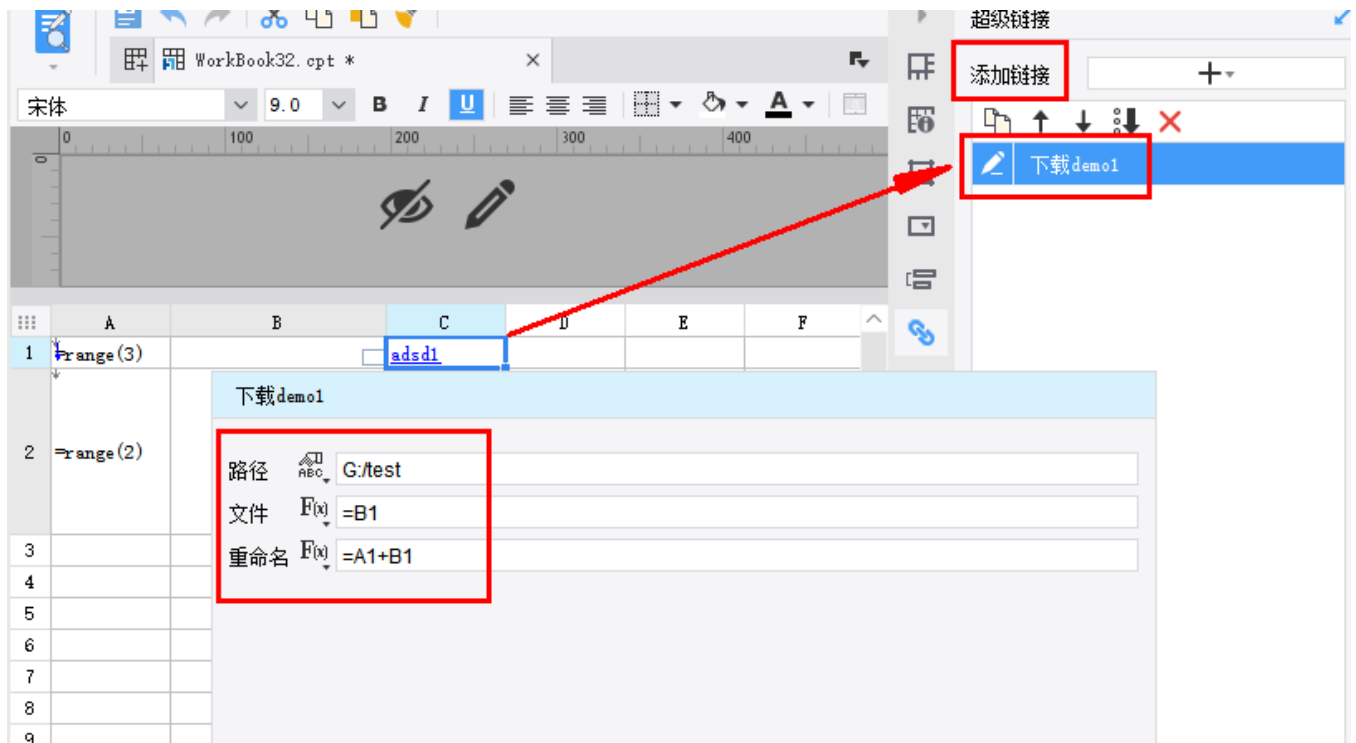
    @Override
    public int hashCode() {
        return nameableCreator.menuName().hashCode();
    }

    @Override
    public boolean equals(Object obj) {
        return (obj != null && obj instanceof DownloadHyperlinkBridge)
            && ComparatorUtils.equals(((DownloadHyperlinkBridge) obj).nameableCreator, nameableCreator);
    }

    @Override
    public NameableCreator createHyperlinkCreator() {
        return nameableCreator;
    }
}
```

## plugin.xml

```
<extra-designer>
    <HyperlinkProvider class="com.tptj.bridge.hg.file.load.demo.DownloadHyperlinkBridge"/>
</extra-designer>
```



FileUploadSubmitTask

## FileUploadSubmit.java

```
package com.tptj.bridge.hg.file.load.demo;

import com.fr.cache.Attachment;
import com.fr.general.GeneralUtils;
import com.fr.io.repository.base.fs.FileSystemRepository;
import com.fr.log.FineLoggerFactory;
import com.fr.plugin.transform.ExecuteFunctionRecord;
import com.fr.plugin.transform.FunctionRecorder;
import com.fr.stable.Primitive;
import com.fr.stable.StableUtils;
import com.fr.stable.StringUtils;
import com.tptj.tools.hg.file.operator.bridge.FileUploadSubmitTask;
import com.tptj.tools.hg.file.operator.dynamics.UploadRowData;

/**
 * @author
 * @version 10.0
 * Created by on 2021-01-04
 */
@FunctionRecorder
public class FileUploadSubmit extends FileUploadSubmitTask<FileDefine> {

    public final static String Job_Type = "upload_demo";

    @Override
    @ExecuteFunctionRecord
    protected void submit( UploadRowData<FileDefine> file ) {
        file.isDelete();//
        file.isModify();//
        FileDefine conf = file.getRowData(FileDefine.class);
        try{
            Object val = conf.getFile();
            if( !(val instanceof Attachment) ){
                return;
            }
            Attachment attachment = (Attachment) val;
            val = conf.getPath();
            String path = StringUtils.EMPTY;
            if( !(val instanceof Primitive) ){
                path = GeneralUtils.objectToString( conf.getPath() );
            }
            val = conf.getRename();
            String rename = StringUtils.EMPTY;
            if( !(val instanceof Primitive) ){
                rename = GeneralUtils.objectToString( conf.getRename() );
            }
            String full_path =StableUtils.pathJoin( path ,
                StringUtils.isNotEmpty( rename )? rename: attachment.getFilename() );
            FileSystemRepository.getSingleton().write( full_path, attachment.getInputStream() );
        }catch(Exception e){
            FineLoggerFactory.getLogger().error(e,e.getMessage());
        }
    }

    @Override
    public String getJobType() {
        return Job_Type;
    }
}
```

FileDefinesubmit

UploadRowData<FileDefine> file

```
FileDefine conf = file.getRowData(FileDefine.class); getget@GetConfig
```

### FileUploadSubmitPane.java

```
package com.tptj.bridge.hg.file.load.demo;

import com.fr.design.beans.BasicBeanPane;
import com.fr.design.layout.FRGUIPaneFactory;
import com.tptj.tools.hg.file.operator.design.DynamicsTablePane;
import com.tptj.tools.hg.file.operator.utils.DynamicsPaneUtils;

import java.awt.*;
import java.util.List;

/**
 * @author
 * @version 10.0
 * Created by on 2021-01-04
 */
public class FileUploadSubmitPane extends BasicBeanPane<FileUploadSubmit> {

    private DynamicsTablePane<FileDefine> editor;

    public FileUploadSubmitPane() {
        setLayout(FRGUIPaneFactory.createM_BorderLayout());
        editor = DynamicsPaneUtils.createTablePane(FileDefine.class);
        add( editor, BorderLayout.CENTER );
    }

    @Override
    public void populateBean( FileUploadSubmit data ) {
        if( null == data ){
            return;
        }
        List<FileDefine> files = data.GetFiles();
        if( null == files ){
            return;
        }
        editor.populate( files.toArray( new FileDefine[files.size()] ) );
    }

    @Override
    public FileUploadSubmit updateBean() {
        FileUploadSubmit task = new FileUploadSubmit();
        task.setFiles( editor.update() );
        return task;
    }

    @Override
    protected String title4PopupWindow() {
        return "demo";
    }
}
```

```
DynamicsPaneUtils.createTablePane(FileDefine.class);
```

```
@Column@ConfigFCIoneable
```

### FileUploadSubmitBridge.java

```
package com.tptj.bridge.hg.file.load.demo;

import com.fr.design.beans.BasicBeanPane;
import com.fr.design.fun.impl.AbstractSubmitProvider;

/**
 * @author
 * @version 10.0
 * Created by on 2021-01-04
 */
public class FileUploadSubmitBridge extends AbstractSubmitProvider {

    @Override
    public BasicBeanPane appearanceForSubmit() {
        return new FileUploadSubmitPane();
    }

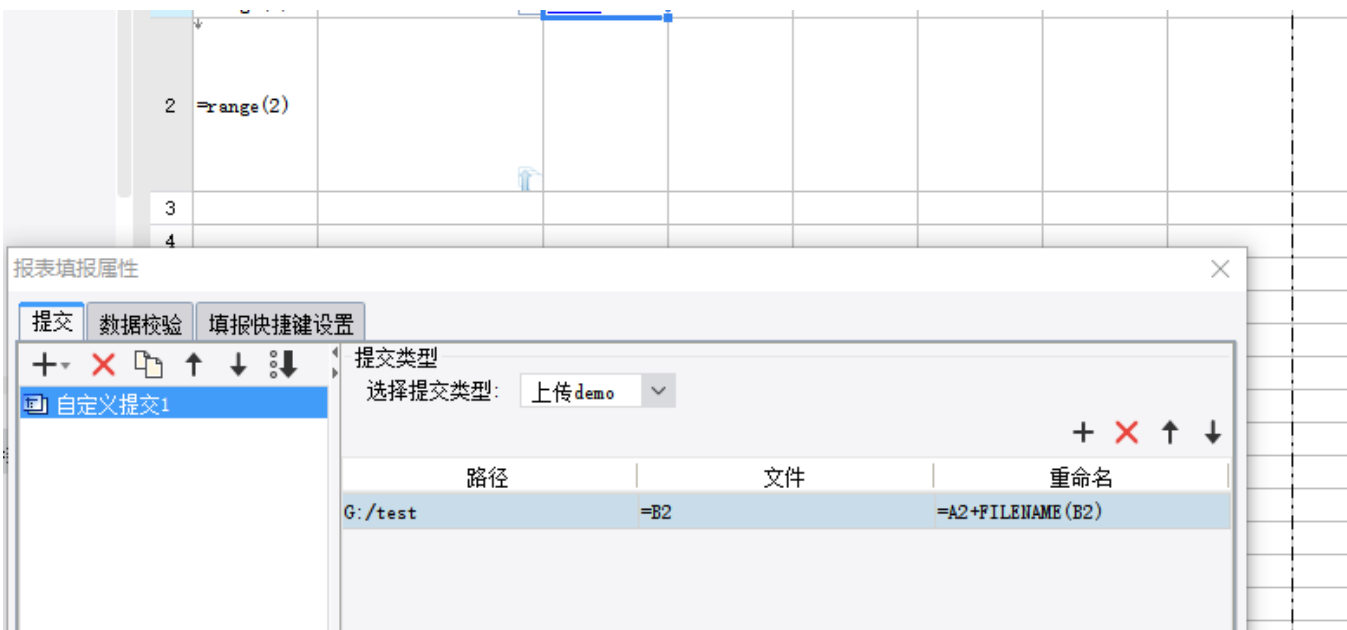
    @Override
    public String dataForSubmit() {
        return "demo";
    }

    @Override
    public String keyForSubmit() {
        return FileUploadSubmit.Job_Type;
    }
}
```

keyForSubmitGetJobType

### plugin.xml

```
<extra-designer>
    <SubmitProvider class="com.tptj.bridge.hg.file.load.demo.FileUploadSubmitBridge"/>
</extra-designer>
```



demodemo

demo



plugin-download-bridge.zip