com.fr.report.fun.FormatActionProvider

FormatActionProviderfilter

viewlet=xxx.cpt&format=xxx

FormatActionProvider.java

```
package com.fr.report.fun;

import com.fr.stable.fun.mark.Mutable;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * format()jspdf
 * Created by zack on 2016/3/3.
 */
public interface FormatActionProvider extends Mutable{

    String XML_TAG = "FormatActionProvider";
    int CURRENT_LEVEL = 1;

    /**
     *
     * @param req
     * @param res
     * @throws Exception
     */
    void doAction(HttpServletRequest req, HttpServletResponse res) throws Exception;

    /**
     * actionkey,cmd,cmd
     * @return key
     */
    String getCMD();
}
```

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-report>
        <FormatActionProvider class="your class name"/>
</extra-report>
```

FormatActionFactory.java

```
package com.fr.web.core.reserve;

import com.fr.log.FineLoggerFactory;
import com.fr.report.ExtraReportClassManager;
import com.fr.report.fun.FormatActionProvider;
import com.fr.web.core.action.PDFPrintPrintAction;

import java.util.HashMap;
import java.util.Map;
import java.util.Set;

/**
 * Created by zack on 2016/3/3.
 */
public class FormatActionFactory {
    private static Map<String, Class<? extends FormatActionProvider>> detailMap = new HashMap<String, Class<?
extends FormatActionProvider>>();

    static {
        detailMap.put("pt_print", PDFPrintPrintAction.class);//pdf
        Set<FormatActionProvider> formatActionProviders = ExtraReportClassManager.getInstance().getArray
(FormatActionProvider.XML_TAG);
        for (FormatActionProvider provider : formatActionProviders) {
            detailMap.put(provider.getCMD(), provider.getClass());
        }
    }

    public static void registerActionProvider(String name, Class<? extends FormatActionProvider> clazz) {
        detailMap.put(name, clazz);
    }

    public static FormatActionProvider getReqProcessor(String format) {
        Class<? extends FormatActionProvider> c = detailMap.get(format);
        try {
            if (c != null) {
                return c.newInstance();
            }
        } catch (InstantiationException e) {
            FineLoggerFactory.getLogger().error(e.getMessage(), e);
        } catch (IllegalAccessException e) {
            FineLoggerFactory.getLogger().error(e.getMessage(), e);
        }
        return null;
    }
}
```

ReportletDealWith.java

```
package com.fr.web.core.reserve;

...

public class ReportletDealWith {
    ...
    /**
     *
     *
     * @param req    http
     * @param res    http
     * @param weblet
     * @throws Exception e
     */
    public static void dealWithReportlet(HttpServletRequest req, HttpServletResponse res, Weblet weblet) throws
Exception {
        String sessionID = SessionPoolManager.generateSessionID(req, res, weblet);

        ...

        if (WebUtils.getHTTPRequestParameter(req, "format") != null) {
            turnToExportWithSessionKept(req, res, sessionID);
        } else {
            dealWithReportlet(req, res, sessionID, null);
        }
    }


    public static void turnToExportWithSessionKept(HttpServletRequest req, HttpServletResponse res, String
sessionID) throws Exception {
        ExportSessionKeeper.getInstance().keepAlive(sessionID);
        turnToExport(req, res, sessionID);
        ExportSessionKeeper.getInstance().close(sessionID);
    }


    /**
     *
     *
     * @param req       request
     * @param res       response
     * @param sessionID session id
     * @throws Exception e
     */
    public static void turnToExport(HttpServletRequest req, HttpServletResponse res, String sessionID) throws
Exception {
        // ,PDF, PDF_Activex, Excel..
        String format = WebUtils.getHTTPRequestParameter(req, "format");
        FormatActionProvider requestProcessor = FormatActionFactory.getReqProcessor(format);
        if (requestProcessor != null) {
            requestProcessor.doAction(req, res);
        } else {
            String embedParameter = WebUtils.getHTTPRequestParameter(req, ParameterConstants.EXPORT_PDF_EMBED);
            boolean embed = "true".equals(embedParameter); // true,
            ExportService.dealWithExport(req, res, sessionID, embed);
        }
    }

    ...
}
```

FormatActionProvider

PDFPDFPrintPrintAction format=pt_printWriteAutoSubmitAction format=submitAction

FormatActionFactoryFormatActionProviderplugin.xmlFormatActionFactory.registerActionProvider("mycmd", MyAction.class);

FormatActionProviderMutableformat

Action

demodemo-format-action-provider

com.fr.report.fun.ExportOperateProvider

com.fr.report.fun.ExportExtensionProcessor

com.fr.stable.fun.ExcelExportCellValueProvider

com.fr.form.stable.FormExportProcessor

com.fr.report.fun.ExcelExportAppProvider

com.fr.report.fun.CommentExcelProcessor

com.fr.io.exporter.PDFExporterCreator