

com.fr.web.core.WebActionsDispatcher

-
- [FineReport V10 2018-07-30](#)
-
-
-

//—FineReport V10 2018-07-30

WebActionsDispatcher.java

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

import com.fr.general.web.ParameterConstants;
import com.fr.plugin.injectable.PluginModule;
import com.fr.stable.StringUtils;
import com.fr.stable.plugin.ExtraClassManagerProvider;
import com.fr.stable.web.RequestCMDReceiver;
import com.fr.web.utils.WebUtils;

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

/**
 * @author kunsnat E-mail:kunsnat@gmail.com
 * @version 2011-12-7 10:52:44
 * : Service, Action. Action
 */
public class WebActionsDispatcher {

    private static final String PLUGIN_ID = "pid";

    /**
     * sessionidcmd
     *
     * @param req HTTP
     * @param res HTTP
     * @param sessionID id
     * @param actions
     *
     * @date 2014-9-25-6:12:55
     */
    public static void dealForActionCMD(HttpServletRequest req, HttpServletResponse res,
        String sessionID, RequestCMDReceiver[] actions) throws Exception {

        dealForActionDefaultCmd(req, res, sessionID, actions, null);
    }

    /**
     * cmd
     *
     * @param req HTTP
     * @param res HTTP
     * @param sessionID id
     * @param actions
     * @param defaultCmd
     *
     * @date 2014-9-25-6:12:55
     */
}
```

```

        */
        public static void dealForActionDefaultCmd(HttpServletRequest req, HttpServletResponse res,
                                                    String sessionID, RequestCMDReceiver[] actions, String
defaultCmd) throws Exception {
            String cmd = WebUtils.getHTTPRequestParameter(req, "cmd");
            if (StringUtils.isEmpty(cmd)) {
                cmd = defaultCmd;
            }
            dealForActionCMD(req, res, sessionID, actions, cmd);
        }

/**
 * sessionidcmd
 *
 * @param req HTTP
 * @param res HTTP
 * @param sessionID id
 * @param actions
 * @param cmd
 *
 * @date 2014-9-25-6:12:55
 */
        public static void dealForActionCMD(HttpServletRequest req, HttpServletResponse res, String sessionID,
RequestCMDReceiver[] actions, String cmd) throws Exception {

            String op = WebUtils.getHTTPRequestParameter(req, ParameterConstants.OP);
            String pluginID = WebUtils.getHTTPRequestParameter(req, PLUGIN_ID);
            ExtraClassManagerProvider pluginProvider = PluginModule.getAgent(PluginModule.ExtraCore);
            RequestCMDReceiver actionCMD = null;
            if (pluginProvider != null) {
                actionCMD = pluginProvider.getActionCMD(op, cmd, pluginID);
            }
            if (actionCMD != null) {
                actionCMD.actionCMD(req, res, sessionID);
            } else {
                for (RequestCMDReceiver action : actions) {
                    String acceptCmd = action.getCMD();
                    if (acceptCmd.equalsIgnoreCase(cmd)) {
                        action.actionCMD(req, res, sessionID);
                        return;
                    }
                }
            }
        }

/**
 * sessionidcmd
 *
 * @param req HTTP
 * @param res HTTP
 * @param actions
 *
 * @date 2014-9-25-6:12:55
 */
        public static void dealForActionNoSessionIDCMD(HttpServletRequest req, HttpServletResponse res,
ActionNoSessionCMD[] actions) throws Exception {

            String op = WebUtils.getHTTPRequestParameter(req, ParameterConstants.OP);

            String cmd = WebUtils.getHTTPRequestParameter(req, "cmd");

            String pluginID = WebUtils.getHTTPRequestParameter(req, PLUGIN_ID);
            ExtraClassManagerProvider pluginProvider = PluginModule.getAgent(PluginModule.ExtraCore);
            RequestCMDReceiver actionCMD = null;
            if (pluginProvider != null) {
                actionCMD = pluginProvider.getActionCMD(op, cmd, pluginID);
            }
        }

```

```

        if (actionCMD != null) {
            actionCMD.actionCMD(req, res);
        } else {
            for (ActionNoSessionCMD action : actions) {
                String acceptCmd = action.getCMD();
                if (acceptCmd.equalsIgnoreCase(cmd)) {
                    action.actionCMD(req, res);
                    return;
                }
            }
        }
    }
}

```

//

[WebServiceview/reportview/form,op cmd](#)

opWebServicecmdAction

43

```

RequestCMDReceiver [] actions = new RequestCMDReceiver[]{
    new ActionA(),
    new ActionB()
}
//sessionId,sessionId
WebActionsDispatcher.dealForActionCMD( request, response, sessionId, actions);

```

```

RequestCMDReceiver [] actions = new RequestCMDReceiver[]{
    new ActionA(),
    new ActionB()
}
//cmd
WebActionsDispatcher.dealForActionDefaultCmd( request, response, sessionId, actions, "cmdA");

```

```

ActionNoSessionCMD [] actions = new ActionNoSessionCMD[]{
    new ActionA(),
    new ActionB()
}
//ActionsessionId,8.0/9.0
WebActionsDispatcher.dealForActionNoSessionIDCMD( request, response, actions);

```

[com.fr.stable.fun.Service](#)

[com.fr.web.core.ActionNoSessionCMD](#)

[com.fr.stable.web.RequestCMDReceiver](#)