

# com.fr.stable.fun.RequestInterceptor

•  
•  
•  
•  
•  
•  
•  
•  
•

WEBopcmd

RequestInterceptor8.09.010.0

1op/cmdAction

2opcmd

## RequestInterceptor.java

```
package com.fr.stable.fun;

import com.fr.plugin.injectable.SpecialLevel;
import com.fr.stable.StringUtils;
import com.fr.stable.fun.mark.Layer;
import com.fr.stable.fun.mark.Mutable;
import com.fr.stable.web.RequestCMDReceiver;

/**
 * Created by richie on 16/8/9.
 * ,opcmd
 */
public interface RequestInterceptor extends Mutable, RequestCMDReceiver, Layer {

    String MARK_STRING = SpecialLevel.RequestInterceptor.getTag();

    int CURRENT_LEVEL = 1;

    class ActionKey {

        private String op;
        private String cmd;
        private String pluginID;

        public static ActionKey create(String op, String cmd, String pluginID) {
            if (pluginID == null) {
                pluginID = StringUtils.EMPTY;
            }
            return new ActionKey(op, cmd, pluginID);
        }

        private ActionKey(String op, String cmd, String pluginID) {
            this.op = op;
            this.cmd = cmd;
            this.pluginID = pluginID;
        }

        public String getOp() {
```

```

        return op;
    }

    public String getCmd() {
        return cmd;
    }

    public String getPluginID() {
        return pluginID;
    }

    @Override
    public boolean equals(Object o) {

        if (this == o) {
            return true;
        }
        if (o == null || getClass() != o.getClass()) {
            return false;
        }

        ActionKey actionKey = (ActionKey) o;

        return op.equals(actionKey.op) && cmd.equals(actionKey.cmd) && pluginID.equals(actionKey.pluginID);
    }

    @Override
    public int hashCode() {
        int result = op.hashCode();
        result = 31 * result + cmd.hashCode();
        result = 31 * result + pluginID.hashCode();
        return result;
    }
}

```

### AbstractRequestInterceptor.java

```
package com.fr.stable.fun.impl;

import com.fr.stable.fun.RequestInterceptor;

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

/**
 * Created by richie on 16/8/9.
 */
public abstract class AbstractRequestInterceptor extends AbstractProvider implements RequestInterceptor {

    public String mark4Provider() {
        return getClass().getName();
    }

    public int currentAPILevel() {
        return CURRENT_LEVEL;
    }

    public void actionCMD(HttpServletRequest req, HttpServletResponse res, String sessionId) throws Exception {

    }

    @Override
    public void actionCMD(HttpServletRequest req, HttpServletResponse res) throws Exception {

    }

    @Override
    public int layerIndex() {
        return CURRENT_LEVEL;
    }
}
```

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

### plugin.xml

```
<extra-core>
    <RequestInterceptor class="your class name" op="", cmd="", pluginId="" />
</extra-core>
```

ExtraClassManager#mountSpecificACTIONaddHackActionCMDhackActionMap

[CMD](#)cmdop/cmd/pluginId

op/cmd

op [WebService](#)WebService[RequestCMDReceiver](#)cmdpluginId

pluginIdpluginID

cmdcmdopcmdpluginIdpluginIdpluginId

demodemo-request-interceptor

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

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

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

[web](#)

[filter](#)