

1

```
public interface PluginServiceCreator{
    //
    public void startService();

    //tomcat
    public void destroyService();

    //
    public String serviceAction(String req) throws Exception;

    //
    public boolean serviceStarted();

    //,
    public void cleanServiceLog();
}
```

2phantom

**phantom**

```
<!--service-->
<!--WebInfo-->
<!--dependence>
<!--dependenceID : IDdependenceType dependenceDir class-->
<!---->
<dependence>
    <DependenceUnit dependenceID="plugin.phantomjs" dependenceType="service" dependenceDir="/resources
/phantomjs" class="com.fr.plugin.chart.phantom.PhantomService" />
</dependence>
```

3phantomID

IDIDMapID

3phantom

## phantom

```
package com.fr.plugin.chart.phantom;
import com.fr.base.FRContext;
import com.fr.plugin.chart.phantom.serverpool.PhantomServerFactory;
import com.fr.plugin.dependence.PluginServiceCreator;

public class PhantomService implements PluginServiceCreator {

    public PhantomServerFactory phantomServerFactory = null;
    public static final String DEFAULT_EXE = "default server";
    public static final String LOG_PATH = FRContext.getCurrentEnv().getPath() + "/service";
    //
    private String ip = "127.0.0.1";
    private int port = 8088;
    private int serverNum = 2;
    private String exe = DEFAULT_EXE;

    //
    @Override
    public void startService() {
        if (phantomServerFactory == null) {
            phantomServerFactory = new PhantomServerFactory();
            //
            phantomServerFactory.init(ip, port, serverNum, exe);

            //
            if (phantomServerFactory.hasServer()) {
                return;
            }
            phantomServerFactory.startServer();
        }
    }

    @Override
    public void destroyService() {
        if (phantomServerFactory != null) {
            phantomServerFactory.destroyServer();
        }
    }

    /**
     *
     * @param req
     * @return
     * @throws Exception
     */
    @Override
    public String serviceAction(String req) throws Exception {
        return phantomServerFactory.requestServer(req);
    }

    @Override
    public boolean serviceStarted() {
        return phantomServerFactory != null;
    }

    @Override
    public void cleanServiceLog() {
        phantomServerFactory.cleanServerLog();
    }
}
```

```
//ID
try {
    //
    response = FRContext.getCurrentEnv().pluginServiceAction("plugin.phantomjs", json.toString());
} catch (Exception e) {
    response = null;
    FRContext.getLogger().info(e.getMessage());
}
```