

Mvc Crypto 插件设计文档

V1.1

文档历史

修订历史

本次修订日期：2012-11-7	下次修订日期：
------------------	---------

修订编号	修订日期	摘要	标记变更
1	2012-11-7	创建版本<杨明光>	
2	2012-11-8	针对1.0版本的批注进行修改<杨明光>	
3	2012-11-9	完善1.1版本<杨明光>	
4			
5			
6			
7			

审批

名称	标题

分发

名称	标题

目录

文档历史	2
修订历史	2
审批	2
分发	2
目录	3
1. 简介	4
1.1 目的	4
1.2 定义	4
1.3 参考	4
2. 总体设计	5
2.1 需求规定	5
2.1.1 前端加密后端解密过程	5
2.1.2 后端加密前端解密过程	5
2.1.3 mvc-crypto加解密的具体形式	6
2.2 运行环境	6
2.3 基本概念和处理流程	6
2.3.1 基本概念	6
2.3.2 处理流程	7
2.4 结构	9
2.4.1 包结构	9
2.4.2 类结构	10
2.5 配置文件	11
2.5.1 web.xml配置	11
2.5.2 Crypto加解密URL配置	11
2.6 页面标签	13
2.6.1 cryptoForm标签	13
2.6.2 延时解密标签	13
2.7 密钥的更新	14
3. 尚未解决的问题	15
3.1 前端加密后端解密后request保留明文	15

1. 简介

1.1 目的

1.2 定义

1.3 参考

1. mvc-crypto需求
2. apache-commons-codec
3. com.ovea ovea-crypto js-java-crypto

2. 总体设计

2.1 需求规定

2.1.1 前端加密后端解密过程

- 1.form表单提供taglib形式。
- 2.Ajax提供Ajax的插件形式。
- 3.form和ajax都需要提供 include、exclude属性。
- 4.form的action和ajax的url要和配置文件匹配 才能正确进行。

配置文件：如下

```
<uncrypto url="mvc/uncrypto/test"/>
<uncrypto url="mvc/test/test"/>
```

2.1.2 后端加密前端解密过程

- 1.利用提供的Ajax插件进行解密 需要后台提供解密的标志。
- 2.非Ajax的解密提供标签的形式进行解密 例如： <xx:uncpt prop="name"/>
- 3.后端只对配置好的需要加密的url进行加密 配置文件：如下

```
<crpyto url="/demo/views/cryp1.jsp" >
    <property includes="name,money,main.common.s.cars.caritms...." name="user" />
    <property excludes="name,money" name="roles" />
    <property includes="name,money,info.address" name="groups" />
</crpyto>
<crpyto url="/demo/views/cryp2.jsp" >
    <property includes="name,money,main.common.s.cars.caritms...." name="user" />
    <property excludes="name,money" name="roles" />
    <property includes="name,money,info.address" name="groups" />
</crpyto>
```

2.1.3 mvc-crypto加解密的具体形式

- 1.采用过滤器的方式进行
- 2.独立于MVC框架。
- 3.核心加解密算法采用XXTEA和BASE64
- 4.需要加解密的url配置在后端的配置文件中
- 5.配置文件提供更新的API
- 6.配置文件加载后放置在application Scope中

2.2 运行环境

mvc-crypto是运行在javaEE的mvc框架上的。需要web容器支持filter

2.3 基本概念和处理流程

2.3.1 基本概念

2.3.1.1 密钥的产生

SessionListener用于创建密钥。

在用户登录后需要复写这个密钥。

密钥生成的算法是 `sessionId [+userId]` md5一次 得到一个32位的16进制数。

根据sessionId或userId得到的密钥需要变化，以免还原。

2.3.1.2 前端加密后端解密

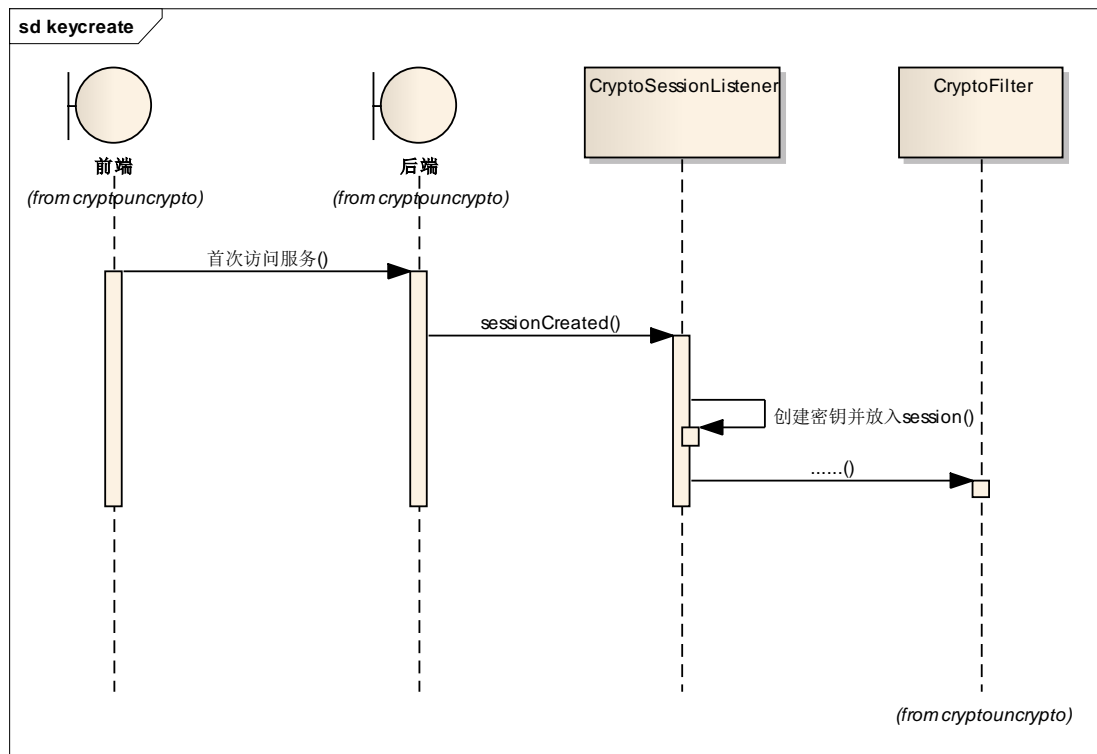
前端页面利用提供的JavaScript 控件或页面标签实现对应内容的加密，并将密文传送到后端，后端根据配置的解密url进行解密，解密后的数据供开发人员直接调用，整个过程开发人员对加密解密过程是透明的。

2.3.1.3 后端加密前端解密

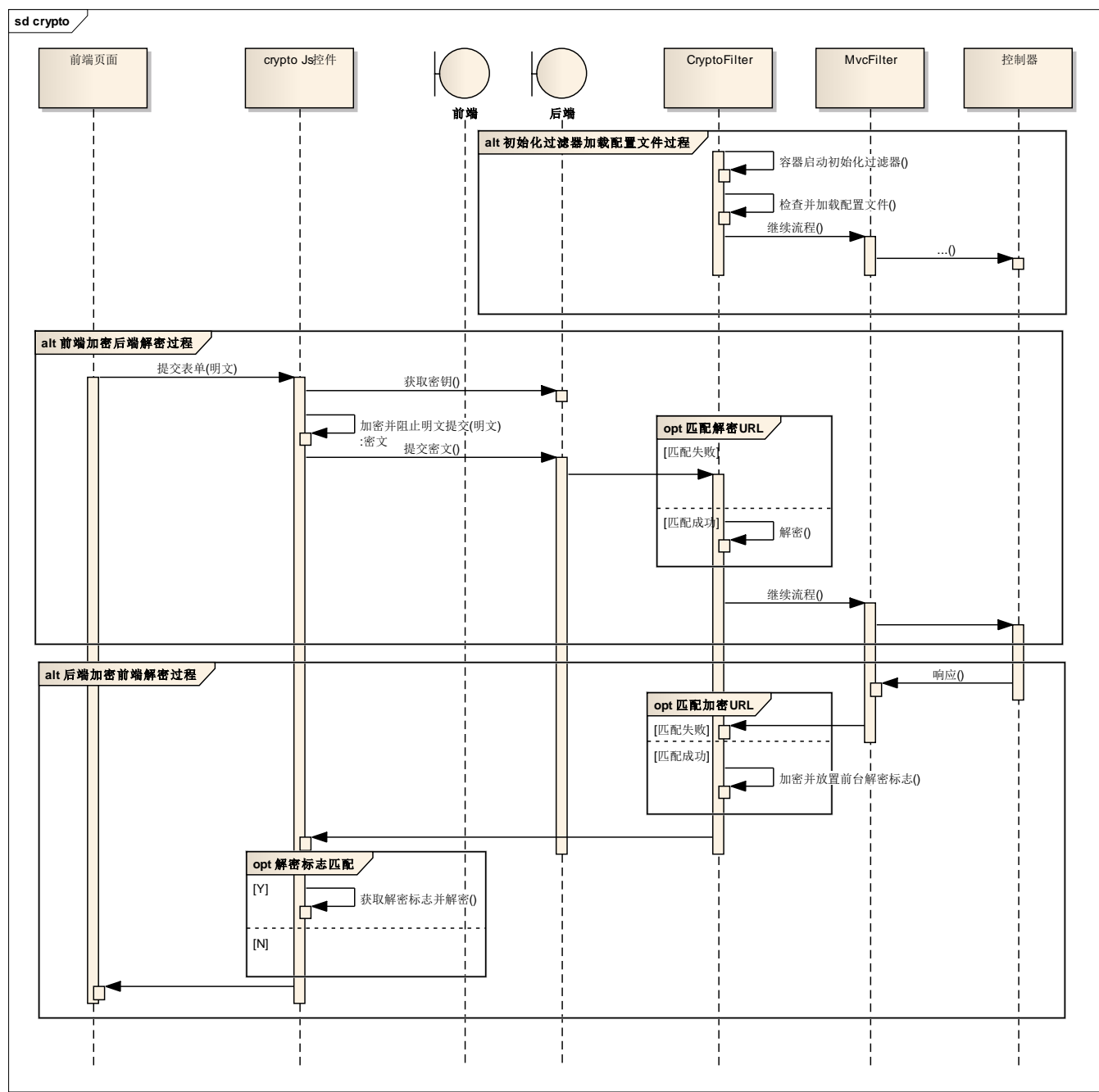
后端根据配置的加密url和详细信息进行加密，将密文响应到前端，前端的JavaScript控件会自动根据内容进行解密，或使用标签取得解密的内容。

2.3.2 处理流程

2.3.2.1 密钥创建过程



2.3.2.2 加密解密过程



2.3.2.3 更新配置文件过程

更新配置文件尚未完成。

class crypto1.1

```

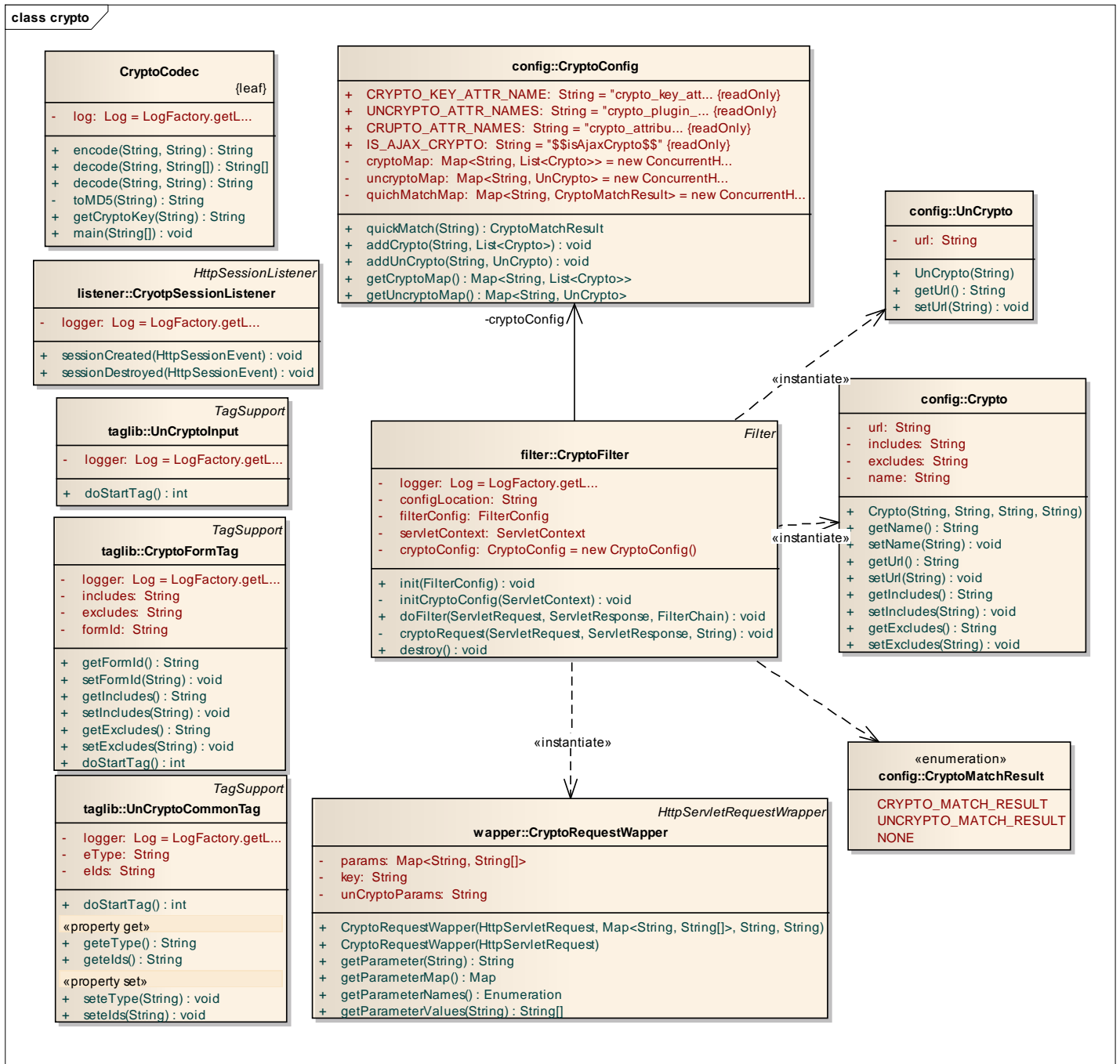
classDiagram
    package com {
        + sinosoft
    }
    package sinosoft {
        + one
    }
    package one {
        + mvc
    }
    package mvc {
        + crypto
    }
    package crypto {
        + CryptoCodec
        + config
        + filter
        + listener
        + taglib
        + wrapper
    }
    package config {
        + Crypto
        + CryptoConfig
        + CryptoMatchResult
        + UnCrypto
    }
    package filter {
        + CryptoFilter
    }
    package listener {
        + CryptoSessionListener
    }
    package taglib {
        + CryptoFormTag
        + UnCryptoCommonTag
        + UnCryptoInput
    }
    package wrapper {
        + CryptoRequestWrapper
    }
    com --> sinosoft
    sinosoft --> one
    one --> mvc
    mvc --> crypto
    crypto --> config
    crypto --> filter
    crypto --> listener
    crypto --> taglib
    crypto --> wrapper
  
```

The diagram illustrates the package structure for class `crypto1.1`. It shows a hierarchical arrangement of packages and their contents:

- com**
 - + sinosoft
- sinosoft** (from com)
 - + one
- one** (from sinosoft)
 - + mvc
- mvc** (from one)
 - + crypto
- crypto** (from mvc)
 - + CryptoCodec
 - + config
 - + filter
 - + listener
 - + taglib
 - + wrapper
- config** (from crypto)
 - + Crypto
 - + CryptoConfig
 - + CryptoMatchResult
 - + UnCrypto
- filter** (from crypto)
 - + CryptoFilter
- listener** (from crypto)
 - + CryptoSessionListener
- taglib** (from crypto)
 - + CryptoFormTag
 - + UnCryptoCommonTag
 - + UnCryptoInput
- wrapper** (from crypto)
 - + CryptoRequestWrapper

2.4.2 类结构

2.4.2.1 类结构图



2.4.2.2 说明

CryptoFilter是加解密的核心过滤器，需要在web.xml中配置。

CryptoSessionListener负责监听session的创建，同时创建密钥。

CryptoConfig是加解密配置文件的加载结果类。

2.5 配置文件

2.5.1 *web.xml*配置

2.5.1.1 CryptoFilter

```
<filter>
    <filter-name>cryptoFilter</filter-name>
    <filter-class>com.sinosoft.one.mvc.cryptofilter.CryptoFilter</filter-class>
    <init-param>
        <param-name>cryptoConfigLocation</param-name>
        <param-value>/WEB-INF/crypto/crypto_config.xml</param-value>
    </init-param>
</filter>
<filter-mapping>
    <filter-name>cryptoFilter</filter-name>
    <url-pattern>/*</url-pattern>
    <dispatcher>REQUEST</dispatcher>
    <dispatcher>FORWARD</dispatcher>
    <dispatcher>INCLUDE</dispatcher>
</filter-mapping>
```

2.5.1.2 CryptoSessionListener

```
<listener>

<listener-class>com.sinosoft.one.mvc.crypto.listener.CryptSessionListener</listener-class>

</listener>
```

2.5.2 *Crypto*加解密URL配置

crypto-config.xml主要配置的是加解密的url。CryptoFilter会根据cryptoConfigLocation属性在容器启动时候进行加载。

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<CryptoConfig xmlns="http://com.sinosoft.one/schema/mvc/cryptoconfig"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://com.sinosoft.one/schema/mvc/cryptoconfig ">
```

```
<crpyto url="/demo/views/cryp1.jsp" >
    <property includes="name,money,main" name="user" />
    <property excludes="name,money" name="roles" />
    <property includes="name,money,info" name="groups" />
```

```
</crpyto>
```

```
<crpyto url="/demo/views/cryp2.jsp" >
    <property includes="name,money,main" name="user" />
    <property excludes="name,money" name="roles" />
    <property includes="name,money,address" name="groups" />
```

```
</crpyto>
```

```
<crpyto url="/demo/views/uncrypto.jsp">
    <property includes="id,name,info2" name="user"/>
```

```
</crpyto>
```

```
<uncrypto url="/demo/send"></uncrypto>
```

```
<uncrypto url="/demo/uncrypto"></uncrypto>
```

```
<uncrypto url="/demo/ajaxParam"></uncrypto>
```

```
<uncrypto url="/demo/cryp/uncrpy1"></uncrypto>
```

```
</CryptoConfig>
```

2.5.2.1 说明

- crypto 加密，url是前台页面的url。crypto的property代表的是一组属性。例如一个form的input有id和name需要加密可以这样写<prperty includes="id,name" />

- uncrypto解密，url指的是form的action或ajax的url。

2.6 页面标签

2.6.1 *cryptoForm* 标签

form的加密标签

2.6.1.1 标签的引用

```
<%@ taglib uri="http://mvc.one.sinosoft.com/crypto/form" prefix="f"%>
```

2.6.1.2 标签的使用

```
<form id="fff" action="/demo/ajaxParam" method="post"
      onsubmit="<f:cryptoForm formId="fff" includes="id,name" />" >
```

2.6.1.3 说明

cryptoForm标签必须写在form的onsubmit方法上面。form必须有id

2.6.2 *延时解密* 标签

页面元素的解密标签，包括inputs的解密和其他元素的解密。

2.6.2.1 标签的引用

```
<%@ taglib uri="http://mvc.one.sinosoft.com/crypto/inputs" prefix="x"%>
<%@ taglib uri="http://mvc.one.sinosoft.com/crypto/commons" prefix="co"%>
```

2.6.2.2 标签的使用

inputs标签

```
<x:inputs/>
```

unCmn标签

```
<co:unCmn elds="un1,un2,un3,un4"/>
```

2.6.2.3 说明

两个标签都属于延时加载，所以需要写在被解密的元素的后面。建议写在页面底部。

inputs不需要配置解密哪些属性，是根据后台的配置自动解密的。

unCmn是UnCryptoCommon的缩写意思是解密其他常用元素。目前支持td,textarea,div,..等无value属性的元素。需要配置解密元素的id。

2.7 密钥的更新

由于密钥的声明周期是绑定到`session`中的，不需要特别的更新方案。如果有特殊需求可以考虑增加更新方案。

3. 尚未解决的问题

3.1 前端加密后端解密后request保留明文