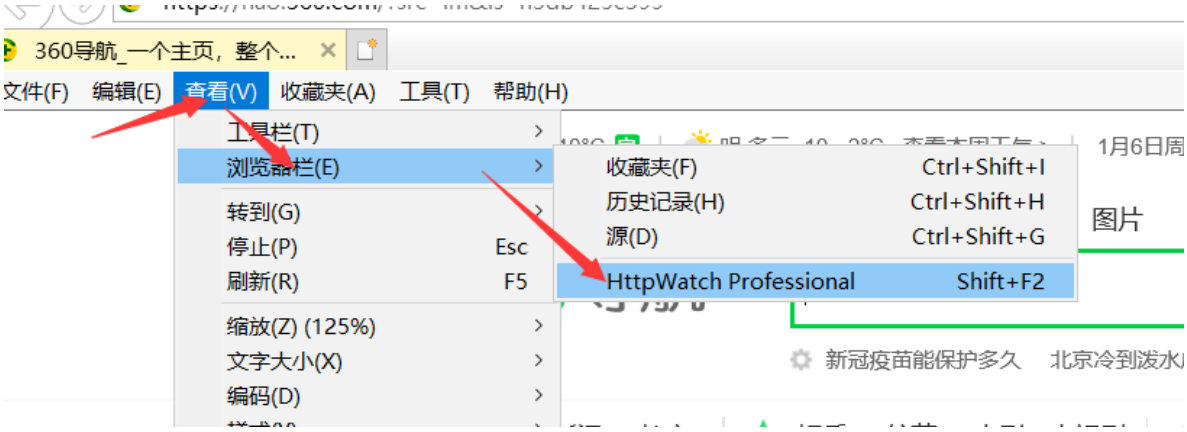
先安装HttpWatch软件，该软件可以对Http协议的监测，该软件只能在IE浏览器上运行监测，在IE浏览器上的打开步骤如下图，使用方法是点击Recored，然后运行网页，点击监测到的网页点击Stream模块



创建一个普通的web项目，新建一个在html目录下的index.html文件，在web.xml文件中配置为欢迎页面。

其中编写一个输入用户名和密码的页面，但是不需要将输入的数据与底层数据库匹配判断，只是形式上输入。再写一个超链接，点击login不是将输入的用户名与密码到数据库匹配，而是跳转到连接请求，然后请求方式设为GET，如下：

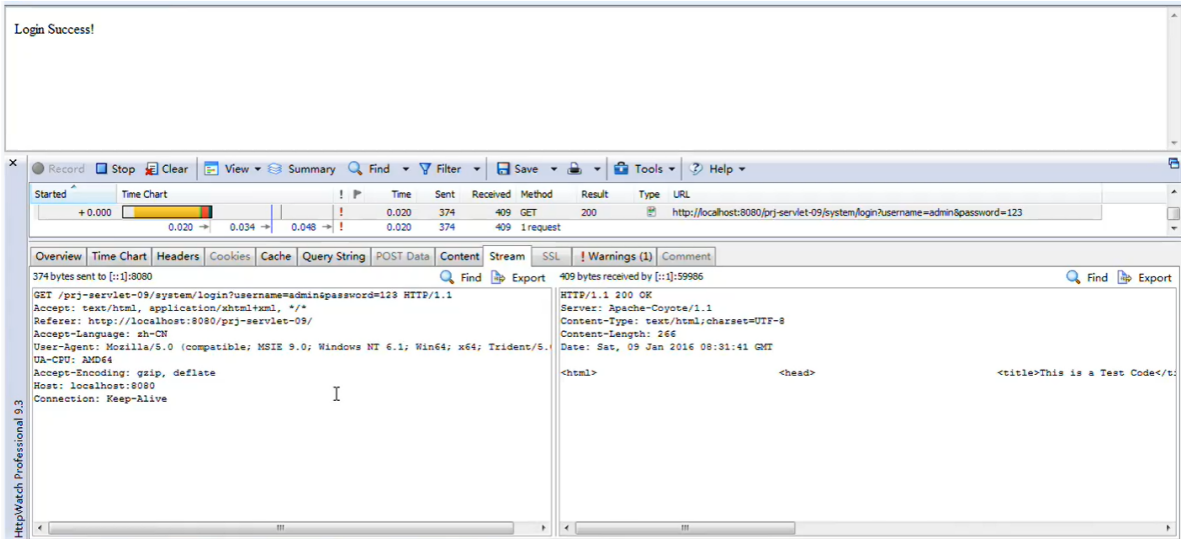
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>login page</title>
</head>
<body>
    <!--下面中的连接是在form标签中输入相应用户名和密码跳转到某个页面-->
    <!--这里输入用户名和密码只是个形式，不进行连接数据库匹配，只要点击登录login都能跳转-->
    <!--这里只是test-->
    <form action="/idea_servlet_http_05_war_exploded/system/hello" method="get">
        <!--前面username是名字，输入内容是text类型，该内容是value，而key值是name中的
username-->
        username<input type="text" name="username"><br>
        password<input type="text" name="password"><br>
        <input type="submit" value="login">
    </form>
</body>
</html>
```

创建HelloServlet类继承适配器抽象类GenericServlet，实现service方法。

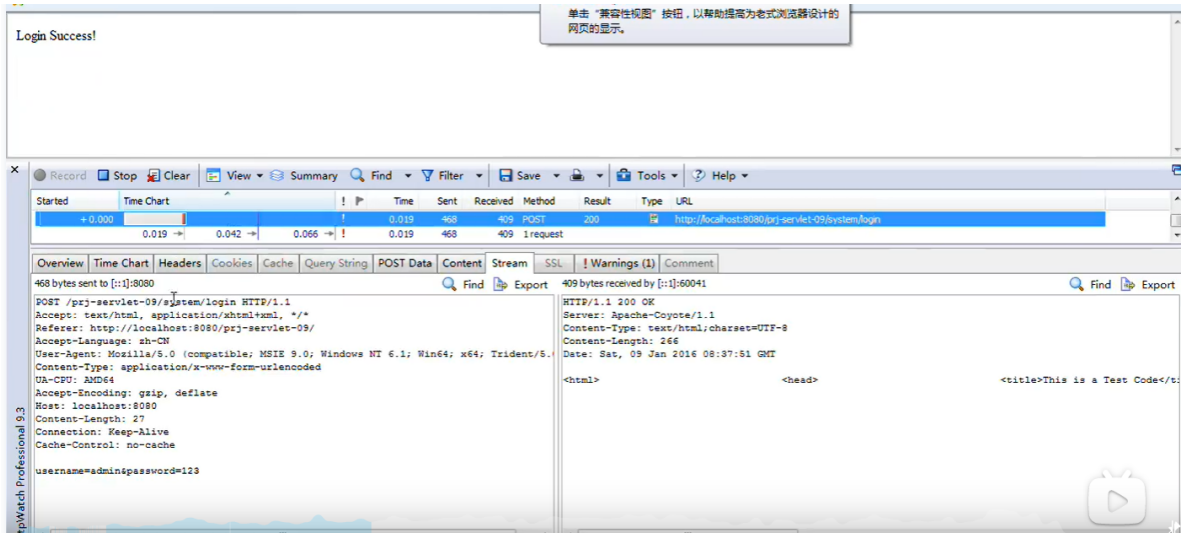
部署项目，在IE浏览器中打开欢迎页面，同时启动监测，随便输入用户名和密码点击login

由于HttpWatch的版本与win10的IE浏览器版本不兼容，开启Recored后点击login无法打开网页，无法测试，将教学中的截图作为分析理解。下图中的左边是请求，右边是响应

GET请求截图：



POST请求截图



--1、HTTP协议的详细内容

1.1 什么是HTTP协议?

--该协议是一直超文本传输协议

--是浏览器和服务器之间的一种通讯协议

--由W3C制定，本质是一种提前制定好的数据传送格式。服务器和浏览器必须遵从这种数据格式传送和发送数据

1.2 HTTP协议版本号

--上面截图的请求行中可以看到HTTP/1.1，就是HTTP协议的版本号

1.3 HTTP协议的几部分

--请求协议：从Browser发送到Server的数据格式

--响应协议：从Server发送到Browser的数据格式

1.4 请求协议

请求协议包括四部分：

--请求行：上两图的第一行是请求行

--消息报头：在请求行到空白行之间的内容就是消息报头

--空白行：GET请求截图中不明显，看POST请求中有一行空白，就是空白行

--请求体：如POST截图中的最后一行信息就是请求体，GET请求中请求体为空

由截图左部分可以看到请求行的信息由请求方式，请求路径（URL），和HTTP协议的版本号。并且对比GET请求 和POST请求，GET请求中发送的数据在请求行中，就是说在请求路径中。POST请求发送的数据在请求体中。空白行是 专门用来分离消息报头和请求体的

1.5 响应协议

四部分

--状态行

--响应报头

--空白行

--响应体、正文

状态行：协议版本号、状态码、状态描述信息

空白行：分离响应报头和响应体

--该协议中重点掌握状态码：

200：响应成功正常结束

404：资源未找到

500：服务器内错误

...

--2、GET请求和POST请求的区别

2.1 什么情况下浏览器发送的请求是GET还是POST请求

使用表单form。将form标签中的method属性设置为method="post"

此时才是POST的请求方式，其余的所有请求都是基于GET方式的

2.2 GET与POST请求的区别

--GET请求是在请求行上提交数据，格式为：URL?name=value&name=value...

这种方式最终提交的数据会显示到浏览器的地址栏上

--POST请求是在请求体中提交数据,是一种相对安全的方式,格式为:name=value&name=value...

该方式最终提交的数据不会显示在浏览器的地址栏上

--POST请求在请求体中提交数据,该数据是没有长度限制的(可以提交大数据)

--GET请求在请求行上提交数据,所以请求的长度是有限制的

--GET请求只能提交字符串数据,而POST请求可以提交任何类型,如视频、图片等等，且这些文件必须用POST请求

--GET请求到的最终结果,会被浏览器收纳。而POST请求的最终结果不会被浏览器缓存

查看上面POST请求方式的监测截图，在其请求报头中有一信息 Cache-Control：no-cache 即表示不缓存请求结果

原因是GET请求多数情况下是到服务器中读取资源，该资源短时间内不会发生改变，所以请求到的结果被浏览器缓存

起来了，下次再进行一模一样的该请求路径时效率就很高。POST请求是为了修改服务器端的资源，修改后的结果都

不会相同，也就没必要缓存到浏览器中。

2.3 GET与POST的选择

--有敏感数据必须使用POST

--数据不是字符串使用POST

--传送的数据非常多，使用POST

--请求是为了修改服务器中的资源使用POST
其余情况使用基于GET请求方式的请求

--注意：
GET请求后，浏览器将结果资源缓存，该缓存资源就与该路径绑定，下一次该浏览器再次发送该请求时，这时浏览器就会在对应的缓存中寻找资源，不再访问服务器，这样缓解了服务器的压力，提高了用户体验。但是有些时候我们希望每次GET请求都访问服务器，可以在请求路径中添加时间戳，例如：
http://ip:port/oa/syatem/logout?timestamp=112121
JS获取毫秒：new Data().getTime();