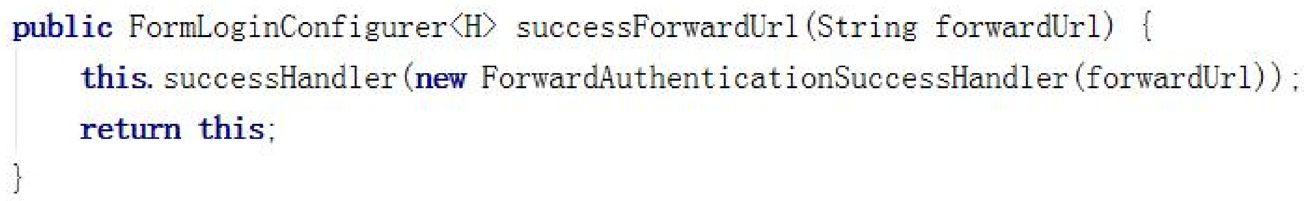
自定义登录成功处理器

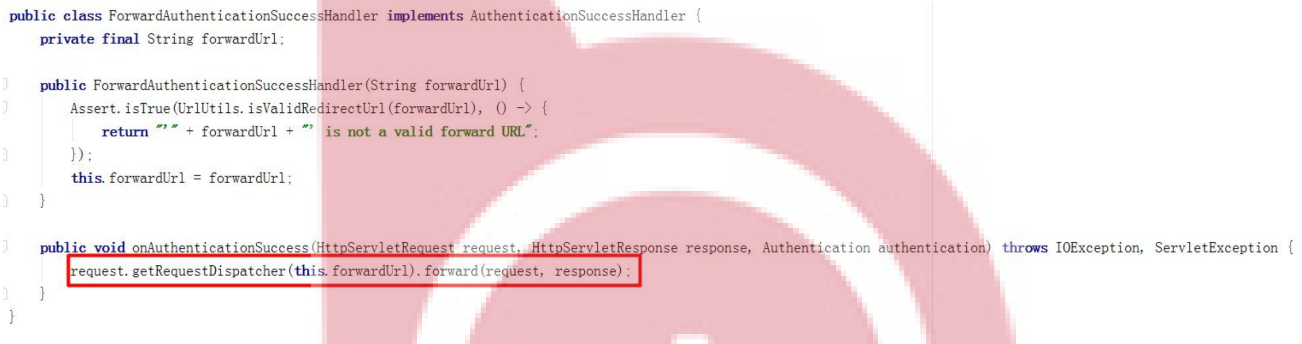
源码分析

使用successForwardUrl()时表示成功后转发请求到地址。内部是通过 `successHandler()` 方法进行控制成功后交给哪个类进行处理



```
public FormLoginConfigurer<H> successForwardUrl(String forwardUrl) {
    this.successHandler(new ForwardAuthenticationSuccessHandler(forwardUrl));
    return this;
}
```

ForwardAuthenticationSuccessHandler内部就是最简单的请求转发。由于是请求转发，当遇到需要跳转到站外或在前后端分离的项目中就无法使用了。



```
public class ForwardAuthenticationSuccessHandler implements AuthenticationSuccessHandler {
    private final String forwardUrl;

    public ForwardAuthenticationSuccessHandler(String forwardUrl) {
        Assert.isTrue(UrlUtils.isValidRedirectUrl(forwardUrl), () -> {
            return "'" + forwardUrl + "' is not a valid forward URL";
        });
        this.forwardUrl = forwardUrl;
    }

    public void onAuthenticationSuccess(HttpServletRequest request, HttpServletResponse response, Authentication authentication) throws IOException, ServletException {
        request.getRequestDispatcher(this.forwardUrl).forward(request, response);
    }
}
```

当需要控制登录成功后去做一些事情时，可以进行自定义认证成功控制器。

代码实现

自定义类

新建类 com.xxxx.handler.MyAuthenticationSuccessHandler 编写如下：

```
package com.xxxx.springsecuritydemo.handler;

import org.springframework.security.core.Authentication;
import org.springframework.security.core.userdetails.User;
import org.springframework.security.web.authentication.AuthenticationSuccessHandler;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;

/**
 * @author zhoubin
 * @since 1.0.0
 */
public class MyAuthenticationSuccessHandler implements AuthenticationSuccessHandler {

   private String url;
```

```
    public MyAuthenticationSuccessHandler(String url) {
        this.url = url;
    }

    @Override
    public void onAuthenticationSuccess(HttpServletRequest request, HttpServletResponse
response, Authentication authentication) throws IOException, ServletException {
        //Principal 主体，存放了登录用户的信息
        User user = (User) authentication.getPrincipal();
        System.out.println(user.getUsername());
        //输出null
        System.out.println(user.getPassword());
        System.out.println(user.getAuthorities());
        response.sendRedirect(url);
    }
}
```

修改配置项

使用 successHandler()方法设置成功后交给哪个对象进行处理

```
//表单提交
http.formLogin()
        //自定义登录页面
        .loginPage("/login.html")
        //当发现/login时认为是登录，必须和表单提交的地址一样。去执行UserServiceImpl
        .loginProcessingUrl("/login")
        //登录成功后跳转页面，POST请求
        // .successForwardUrl("/toMain")
        //和successForwardUrl不能共存
        .successHandler(new MyAuthenticationSuccessHandler("http://www.baidu.com"))
        //登录失败后跳转页面，POST请求
        .failureForwardUrl("/toError")
        .usernameParameter("myusername")
        .passwordParameter("mypassword");
```