服务器端通过session来得知连接的客户端，因此需要okhttp与webview使用一套cookie，就可以实现在H5页面中与原生app同步登陆。  
通过阅读okhttp3源码得知，Okhttp是使用cookjar来设置cookie的，而在OKHttpClient的Builfer方法中.默认的cookjar是一个空的对象，没有设置任何cookie

/\*\* A cookie jar that never accepts any cookies. \*/

CookieJar NO\_COOKIES = new CookieJar() {

@Override public void saveFromResponse(HttpUrl url, List<Cookie> cookies) {

}

@Override public List<Cookie> loadForRequest(HttpUrl url) {

return Collections.emptyList();

}

};

所以只需要我们手动取得webview的cookie并设置在okhttp的cookJar上即可实现cookie的共享。webview是的引擎为webkit，在webkit是有cookieManager对cookie进行管理。

public class WebViewCookieHandler implements CookieJar {

private CookieManager mCookieManager = CookieManager.getInstance();

@Override

public void saveFromResponse(HttpUrl url, List<Cookie> cookies) {

String urlString = url.toString();

for (Cookie cookie : cookies) {

mCookieManager.setCookie(urlString, cookie.toString());

}

}

@Override

public List<Cookie> loadForRequest(HttpUrl url) {

String urlString = url.toString();

String cookiesString = mCookieManager.getCookie(urlString);

if (cookiesString != null && !cookiesString.isEmpty()) {

String[] cookieHeaders = cookiesString.split(";");

List<Cookie> cookies = new ArrayList<>(cookieHeaders.length);

for (String header : cookieHeaders) {

cookies.add(Cookie.parse(url, header));

}

return cookies;

}

return Collections.emptyList();

}

}

然后再创建okhttpclient的工具类中加上自定义的cookiejar

public static OKhttpClient getClient(){

return new OkhttpClient.Builder()

.cookieJar(new WebViewCookieHandler())

.build();

}