
안심본인인증 모바일연동 가이드 (Android)

Version 2.0A

2019년 9월 5일
NICE평가정보(주) 디지털개발실

목차

1. URL SCHEME 설정	- 2 -
1.1 URL SCHEME 설정이 필요한 경우	- 2 -
1.2 URL SCHEME 설정 방법	- 2 -
1.2.1 WebView 설정 추가 * MainActivity.java파일 참고.....	- 2 -
1.2.2 WebViewClient에 필요처리 추가 * DemoWebViewClient class 참고.....	- 3 -
2. WebView를 통한 인증 시 쿠키처리	- 4 -
2.1 쿠키를 통한 세션값 유지	- 4 -
2.2 세션값 저장 시 주의 사항	- 4 -
2.3 타 통신 라이브러리 이용 시.....	- 4 -
3. WebView의 결과 전달	- 4 -

1. URL SCHEME 설정

1.1 URL SCHEME 설정이 필요한 경우

WebView에서 통신사 인증앱(PASS앱) 호출 시 오류가 발생하거나, 앱 설치 안내 버튼 클릭 시 PlayStore 앱으로 연결이 안 되는 경우 URL SCHEME 설정이 필요합니다.

1.2 URL SCHEME 설정 방법

1.2.1 WebView 설정 추가 * MainActivity.java파일 참고

함께 첨부된 MainActivity.java 파일을 참고해 WebView 설정을 추가합니다.

아래 샘플코드의 붉은색 박스영역은 반드시 추가하셔야 합니다.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    mWebView = (WebView) findViewById(R.id.webView);

    //웹뷰의 설정을 다음과 같이 맞춰주시기 바랍니다.
    mWebView.getSettings().setJavaScriptEnabled(true); //필수설정(true)
    mWebView.getSettings().setDomStorageEnabled(true); //필수설정(true)
    mWebView.getSettings().setJavaScriptCanOpenWindowsAutomatically(true); //필수설정(true)

    mWebView.getSettings().setCacheMode(WebSettings.LOAD_NO_CACHE);
    mWebView.getSettings().setLoadsImagesAutomatically(true);
    mWebView.getSettings().setBuiltInZoomControls(true);
    mWebView.getSettings().setSupportZoom(true);
    mWebView.getSettings().setSupportMultipleWindows(true);
    mWebView.getSettings().setLoadWithOverviewMode(true);
    mWebView.getSettings().setUseWideViewPort(true);

    /**
     * !필수사항!
     *
     * 웹뷰 내 앱링크를 사용하려면 WebViewClient를 반드시 설정하여 주시기바랍니다. (하단 DemoWebViewClient 참고)
     */
    mWebView.setWebViewClient(new DemoWebViewClient());

    mWebView.loadUrl(URL_INFO);
}
```

<필수설정>

setJavaScriptEnabled(true);	// 웹뷰 내 javascript 이용 여부
setDomStorageEnabled(true);	// 웹뷰 내 로컬스토리지 이용 여부
setJavaScriptCanOpenWindowsAutomatically(true);	// 웹뷰 내 window.open() 허용 여부
setWebViewClient(new DemoWebViewClient());	// 웹뷰 내 URL별 분기 처리
	// (DemoWebViewClient 클래스 참고)

1.2.2 WebViewClient에 필요처리 추가 * DemoWebViewClient class 참고

아래 샘플코드의 shouldOverrideUrlLoading 함수같이 URL 별 분기처리를 해주셔야 통신사 인증앱 실행, 통신사앱 미설치 시 앱스토어로 연결에 오류가 발생하지 않습니다.

(WebViewClient를 상속받은 DemoWebViewClient 생성 후 shouldOverrideUrlLoading 함수 별도 처리)

```
@Override
public boolean shouldOverrideUrlLoading(WebView view, String url) {

    //웹뷰 내 표준창에서 외부앱(통신사 인증앱)을 호출하려면 intent:// URI를 별도로 처리해줘야 합니다.
    //다음 소스들 적용 해주세요.
    if (url.startsWith("intent://")) {
        Intent intent = null;
        try {
            intent = Intent.parseUri(url, Intent.URI_INTENT_SCHEME);
            if (intent != null) {
                //앱실행
                startActivity(intent);
            }
        } catch (URISyntaxException e) {
            //URI 문법 오류 시 처리 구간
        } catch (ActivityNotFoundException e) {
            String packageName = intent.getPackage();
            if (!packageName.equals("")) {
                // 앱이 설치되어 있지 않을 경우 구글마켓 이동
                startActivity(new Intent(Intent.ACTION_VIEW, Uri.parse("market://details?id=" + packageName)));
            }
        }
        //return 값을 반드시 true로 해야 합니다.
        return true;
    } else if (url.startsWith("https://play.google.com/store/apps/details?id=") || url.startsWith("market://details?id=")) {
        //표준창 내 앱설치하기 버튼 클릭 시 PlayStore 앱으로 연결하기 위한 로직
        Uri uri = Uri.parse(url);
        String packageName = uri.getQueryParameter("id");
        if (packageName != null && !packageName.equals("")) {
            // 구글마켓 이동
            startActivity(new Intent(Intent.ACTION_VIEW, Uri.parse("market://details?id=" + packageName)));
        }
        //return 값을 반드시 true로 해야 합니다.
        return true;
    }

    //return 값을 반드시 false로 해야 합니다.
    return false;
}
```

별도의 WebViewClient 사용하는 경우 붉은색 박스영역 코드를 shouldOverrideUrlLoading 함수 내부에 추가

<WebView 내 분기처리가 필요한 URI 목록>

- Intent:// 로 시작하는 링크 (예: intent://[통신사별 스키마])
- 스토어 주소 링크

https://play.google.com/store/apps/details?id=[통신사별 packagename]

market://details?id=[통신사 별 packagename]

2. WebView를 통한 인증 시 쿠키처리

2.1 쿠키를 통한 세션값 유지

인증 진행 시 안드로이드 CookieManager를 이용해 업체 도메인의 세션값을 유지시켜줘야 합니다.

예)

```
Private CookieManager cookie = CookieMnager.getInstance();  
Cookie.setCookie( url, "세션값");
```

위 샘플처럼 쿠키값을 넣어주면 해당 url의 쿠키 정보를 유지하게 됩니다.

2.2 세션값 저장 시 주의 사항

저장하는 세션값이 조금이라도 틀리면 유지되지 않으니 주의해주시고. 예를 들어, spring 으로 짜여진 web 호출 시 jsessionid="세션내용" 인데 ""를 누락하고 세션 내용만 쓰면 문제가 발생할 수 있습니다.

2.3 타 통신 라이브러리 이용 시

retrofit 혹은 volley와 같은 타 통신 라이브러리를 사용하는 경우, 해당 라이브러리에서 제공하는 cookie 설정에 맞춰 적용해주시고.

3. WebView의 결과 전달

WebView로 리턴된 인증결과는 Javascriptinterface 이용해 모바일앱으로 전달 가능합니다.