

<http://www.egovframe.go.kr/wiki/doku.php?id=egovframework:hyb3.5:hrtc:session>

Session Synchronization (Android)

While iOS is capable of synchronizing sessions, the developer must ensure the sessions are synchronized when calling the server.

Adding Cookie Manager.

- Administers Single-tone Instance of CookieManager.
- URL in the server is called out of SessionCommTask().execute(cookieManager,baseUrl); for storage in CookieManager.
- Cookie information stored in CookieManager is to be loaded for binding with URL.

```
public class MainActivity extends CordovaActivity
{
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        super.init();

        super.appView.getSettings().setAppCacheEnabled(false);
        super.appView.getSettings().setCacheMode(WebSettings.LOAD_NO_CACHE);
        //super.appView.setVerticalScrollbarOverlay(true);
        super.appView.clearCache(true);
        super.appView.getSettings().setJavaScriptEnabled(true);
        super.appView.getSettings().setDomStorageEnabled(true);

        //super.appView.getSettings().setSaveFormData(false);
        //super.appView.getSettings().setJavaScriptCanOpenWindowsAutomatically(true);
        //super.appView.getSettings().setBuiltInZoomControls(true);
        //super.appView.getSettings().setSupportZoom(true);
        //super.appView.getSettings().setSavePassword(false);

        CookieSyncManager.createInstance(this);
        CookieManager cookieManager = CookieManager.getInstance();
        CookieSyncManager.getInstance().startSync();
        //cookieManager.setCookie("http://192.168.100.120:8280",
        "JSESSIONID=998822185E5713BE7188A163F33C5433");

        String baseUrl = this.getString(R.string.SERVER_URL);
        new SessionCommTask().execute(cookieManager,baseUrl);

        // Set by <content src="index.html" /> in config.xml
        loadUrl(launchUrl);
    }

    public class SessionCommTask extends AsyncTask<Object, Integer, String> {
```

```

CookieManager cookieManager;
String baseUrl;

@Override
protected void onPreExecute() {

    super.onPreExecute();
}

@Override
protected String doInBackground(Object... params) {

    cookieManager = (CookieManager)params[0];
    baseUrl = (String)params[1];

int errorCode = ERROR_CODE_NOERROR;

try {
String addr = baseUrl+"/dvc/xml/deviceInfoList.do";
URL url = new URL(addr);
Log.d(this.getClass().getSimpleName(),"addr : "+addr);
URLConnection conn = (URLConnection) url.openConnection();
if( conn != null ) {
    Log.d(this.getClass().getSimpleName(),"http conn OK");
    int resCode = conn.getResponseCode();
    if( resCode == HttpURLConnection.HTTP_OK ) {

        String resultCookie = conn.getHeaderField("Set-Cookie");
        Log.d(this.getClass().getSimpleName(),"main >
resultCookie1 : "+resultCookie);

        if (resultCookie!=null) {
            String[] cookieItems = resultCookie.split(";");
            //if (cookieItems.length>0) {
                for(String cookieItem : cookieItems) {
                    //for(int i =0; i < cookieItems.length ;
i++) {

                        Log.d(this.getClass().getSimpleName(),"cookie Itme : "+cookieItem);

                        cookieManager.setCookie(baseUrl, cookieItem);
                    }
                //}
            }

            Log.d(this.getClass().getSimpleName(),"main >
resultCookie2 : "+cookieManager.getCookie(baseUrl));

            // DISCONNECT
            conn.disconnect();
        }
    } else {
        Log.d(this.getClass().getSimpleName(),"http conn FAIL");
    }
}
}

```

```

        errorCode = ERROR_CODE_HTTP_ELSE;
    }
    } catch(Exception e) {
    e.printStackTrace();
    errorCode = ERROR_CODE_HTTP_EXCEPTION;
    }

    return null;
}

protected void onProgressUpdate(Integer... progress) {
Log.d(this.getClass().getSimpleName(),"onProgressUpdate:["+progress[0]+"]");
}

@Override
protected void onPostExecute(String result) {

    super.onPostExecute(result);
}

@Override
protected void onCancelled() {
    // TODO Auto-generated method stub
    super.onCancelled();
}

}
Binding cookies to EgovInterfacePlugin

```

- The cookie corresponding with the URL for Restful communication is loaded out of CookieManager.
- The cookie information is added to the header information for binding in the form of requestHeaders.add("Cookie", cookies);
- Keep in mind Android is not capable of session synchronization when the session is not bound. When the session has not been synchronized, the servers cannot recognize a multitude of attempts out of the same application, leading to authentication failure.

```

@Override
protected String doInBackground(Object... params) {

    action = (String)params[0];
    url = (String)params[1];
    data = (JSONArray)params[2];
    callbackContext = (CallbackContext)params[3];

    if (action.equals(HTTP_METHOD_GET)) {
try {

    String uri = data.getString(0);
    JSONObject param = data.getJSONObject(2);
    Log.d(this.getClass().getSimpleName(), " $$$$ : "+uri);
    Log.d(this.getClass().getSimpleName(), " $$$$ : "+param.toString());
    Log.d(this.getClass().getSimpleName(), " $$$$ : "+data.getString(1));
}
}
}

```

```

HttpHeaders requestHeaders = new HttpHeaders();
requestHeaders
    .setContentType(MediaType.APPLICATION_FORM_URLENCODED);
if ("json".equals(data.getString(1))) {
    requestHeaders.setAccept(Collections
        .singletonList(MediaType.APPLICATION_JSON));
} else if ("xml".equals(data.getString(1))) {
    requestHeaders.setAccept(Collections
        .singletonList(MediaType.APPLICATION_XML));
    List<Charset> charset = new ArrayList<Charset>(Charset
        .availableCharsets().values());
    charset.add(Charset.forName("UTF-8"));
    requestHeaders.setAcceptCharset(charset);
} else {
    callbackContext.error(ERROR_MESSAGE_PARAM);
    return null;
}

```

```

String cookies = cookieManager.getCookie(url);
Log.d(this.getClass().getSimpleName(), "cookies > "+cookies);
//requestHeaders.add("Cookie",
"JSESSIONID=E7A4058A9BD6B26D8A9D5F869F1E24E4");
requestHeaders.add("Cookie", cookies);

```

```

HttpEntity<String> requestEntity = new HttpEntity<String>("",
    requestHeaders);

```

```

// Create a new RestTemplate instance
RestTemplate restTemplate = new RestTemplate();

```

```

// Add the String message converter
restTemplate.getMessageConverters().add(
    new StringHttpMessageConverter());

```

```

if (url == null) {
    url = "";
}

```

```

if (uri == null) {
    uri = "";
}

```

```

// Make the HTTP GET request, marshaling the response to a
// String
ResponseEntity<String> responseEntity = restTemplate.exchange(
    url + uri + "?" + parseParameter(param),
    HttpMethod.GET, requestEntity, String.class);

```

```

String result = responseEntity.getBody();

```

```
Log.d(this.getClass().getSimpleName(),"cookies result > "+responseEntity.getHeaders());
```

```
        callbackContext.success(result);  
    } catch (RestClientException e) {  
        callbackContext.error(e.getLocalizedMessage());  
    } catch (JSONException e) {  
        callbackContext.error(ERROR_MESSAGE_JSON);  
    } catch (Exception e) {  
        callbackContext.error(ERROR_MESSAGE_IO);  
    }  
}
```

... etc.

Adding Interceptor

- For additional authentication, you may add an interceptor in dispatcher-servlet.xml.
- First of all, designate the URL for interception, in the form of <mvc:mapping>.
- Designate the URL to skip interception, in the form of <mvc:exclude-mapping>.

```
<mvc:interceptors>  
<mvc:interceptor>  
<mvc:mapping path="/**/*.*do" />  
<mvc:exclude-mapping path="/uat/uia/**" />  
<mvc:exclude-mapping path="/index.do" />  
<bean class="egovframework.com.cmm.interceptor.AuthenticInterceptor" />  
</mvc:interceptor>  
</mvc:interceptors>
```

Verify authentication in the Interceptor.

- Check if Cookie is properly bound, using session.getId(). The very same ID is returned when properly bound.

```
public class AuthenticInterceptor extends HandlerInterceptorAdapter {
```

```
    /**  
     * The system checks the authentication by referring to LoginVO in the session.  
     * The system moves on to the login page when no LoginVO is available.  
     */  
    @Override  
    public boolean preHandle(HttpServletRequest request, HttpServletResponse response,  
Object handler) throws Exception {  
  
        boolean isPermittedURL = false;  
        HttpSession session = request.getSession();  
  
        System.out.println(">>>>> AuthenticInterceptor sessionID : "+session.getId());  
        System.out.println(">>>>> AuthenticInterceptor~~~~~");  
        /*  
        LoginVO loginVO = (LoginVO) EgovUserDetailsHelper.getAuthenticatedUser();  
        loginVO.setId("OK");  
  
        if(loginVO != null){  
            return true;  
        }  
    }  
}
```

```
        } else if(!isPermittedURL){
            ModelAndView modelAndView = new
ModelAndView("redirect:/dvc/logout.do");
            throw new
ModelAndViewDefiningException(modelAndView);
        } else {
            return true;
        }
    }
    */
    return true;
}
}
```