林老师，关于Google登录的过程，目前总结的情况是这样的：

1、Google的执行过程并没有具体的流程文档，只是强调自己的流程符合OIDC标准。

2、Google并不提供开源的代码或者SDK，想要从代码分析只能从网页中获取源代码，但这个是好几千行混淆过无注释的代码，短时间也分析不完。

3、目前从Analysing the Security of Google’s implementation of OpenID Connect这篇文章中发现了他们对于Google登录流程的研究。具体内容是这篇文章2.3.1的内容进行一些修改：

step 1：The user clicks the Google Sign-In button rendered on the RP website, and this causes the browser to send an HTTPS request to the RP.

step 2：The RP generates an OpenID Connect authorization request and sends it to the browser. The authorization request includes ***client id***, an identifier the RP registered with the IdP previously; ***response type***=id token which requests an id token be returned from Google; ***redirect uri***=postmessage (与redirect UX不同了), indicating that postMessage is being used; ***state***, an opaque value used by the RP Client to maintain state between the request and the callback (step 5 below); ***origin***, a URL without a path appended (代替原来的redirect uri); and ***scope***, the scope of the requested permission. 用这个脚本开了新的IdP页面。

step 3： If the user has already been authenticated by the IdP then this step and the next are skipped. If not, the IdP returns a login form which is used to collect user authentication information (e.g. user account and password). 在IdP页面执行，包含IdP脚本。

step 4： The user completes the login form and grants permission for the RP to access the attributes stored by the IdP.

step 5： After receiving the permission grant from the user, the IdP generates an HTML document (包含script) which contains the authentication response and sends it back to the browser. The authentication response contains the id token generated by the IdP; and state, the value sent in step 2.

step 6： The browser executes the JavaScript inside the html document it received in the previous step. The JavaScript sends the authentication response using postMessage to the RP script. After receiving the authentication response, the RP script extracts the id token and sends it back to the RP.

step 7： The RP verifies the id token; if the id token is valid, the RP now has evidence that the user has been authenticated.

其中step 6- step 7的修改，我是基于对Google官方开发者文档中的API介绍进行的，其文档中说明了，用户应该自行通过callback获得id token并且将id token上传至RP server。

对于RP script获取id token的介绍如下：

图形用户界面, 文本, 应用程序, 电子邮件

AI 生成的内容可能不正确。

说明需要在server端处理id token：

文本

AI 生成的内容可能不正确。

文档中说明在pop-up UX中，由开发者自行决定RP script获取id token之后如何上传服务器使用：

文本

AI 生成的内容可能不正确。