Getting started with oauth2 for Google

Posted At : October 31, 2014 12:59 AM | Posted By : William Eatman
Related Categories: oauth2, coldfusion, Google Calendar

Github Project: https://github.com/billeatman/oauth2-Examples

For CommandBox: forgebox install oauth2-Examples

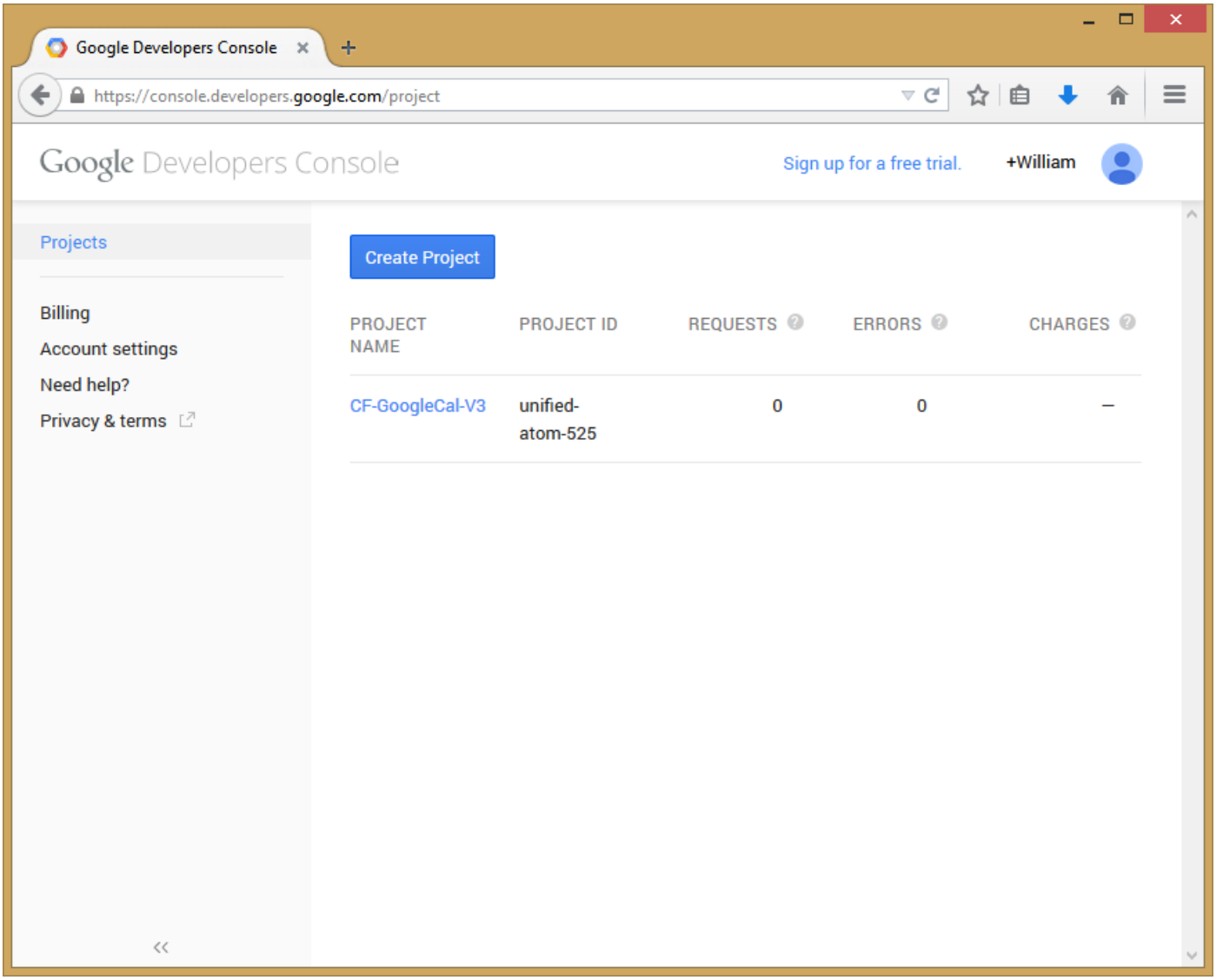
I'm going to focus on how to set up the oauth2 project to work with Google. It's a confusing process if you've never done it before.

If you are new to oauth2, I would first read through the docs on Google.

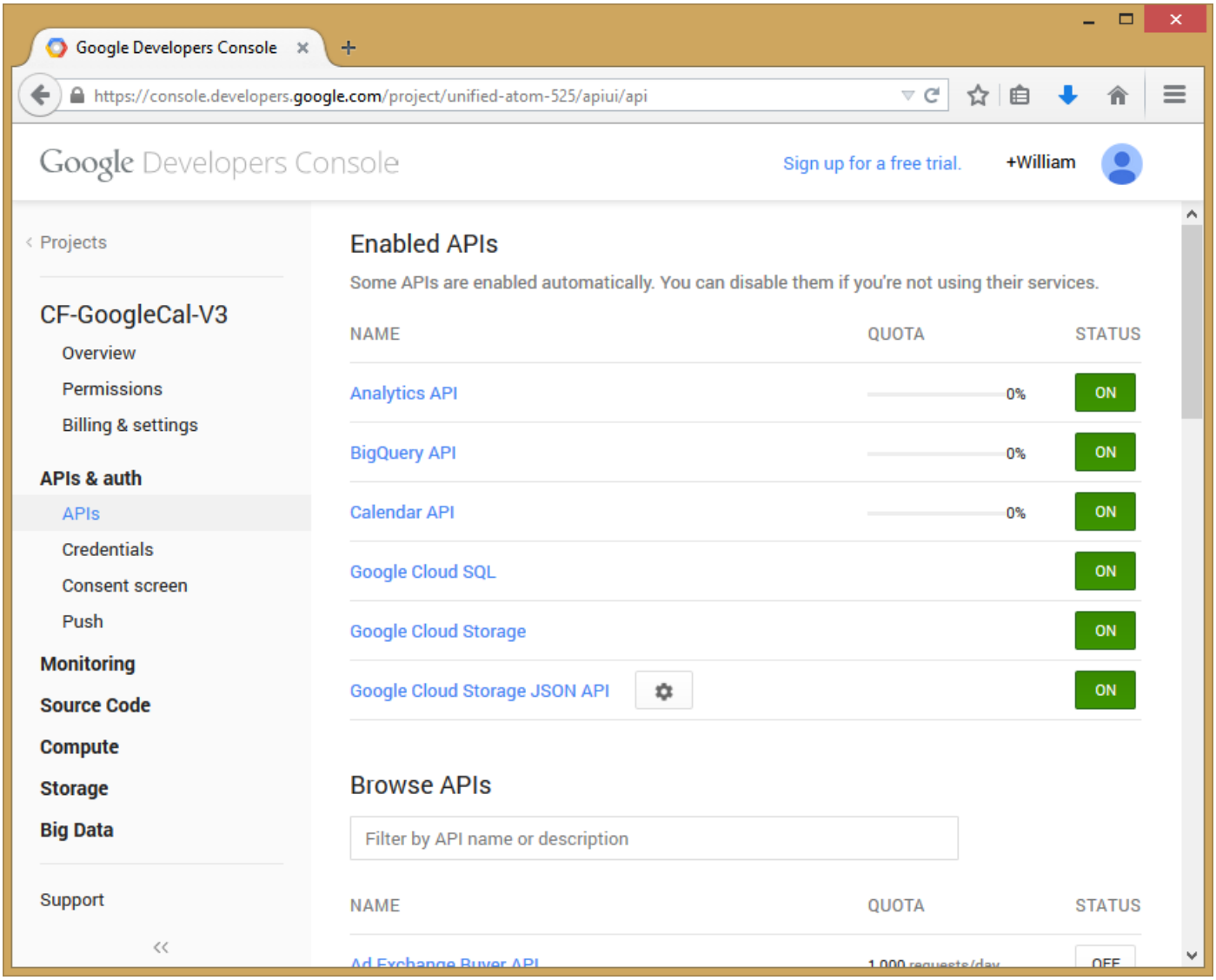
https://developers.google.com/accounts/docs/OAuth2

Let's get started!

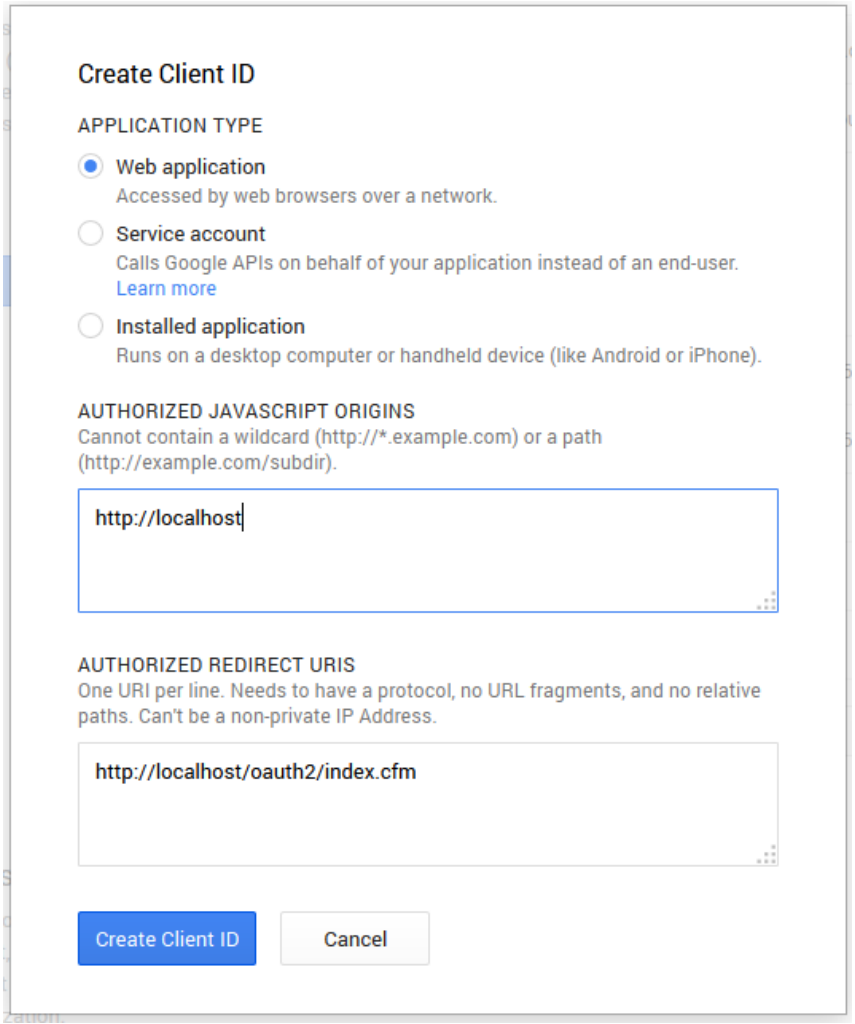
Our first step is to create a project using the Google Developers Console.



I've created a project for my GoogleCal-V3 project. Now we can enable the APIs need for our application. I've enabled a few things for this project including the Calendar API.



After enabling needed API, select "Credentials" then click the "Create new Client ID" button. You should see the dialog below. I've filled it in assuming that the oauth2 project is in a directory off of the root called oauth2.



Now you can copy the newly generated client_id and client_secret from the Google Developer Console to the application.cfc in the oauth2 project. In this example I'm using the scope for Google Calendar, but you can use whatever scope you want.

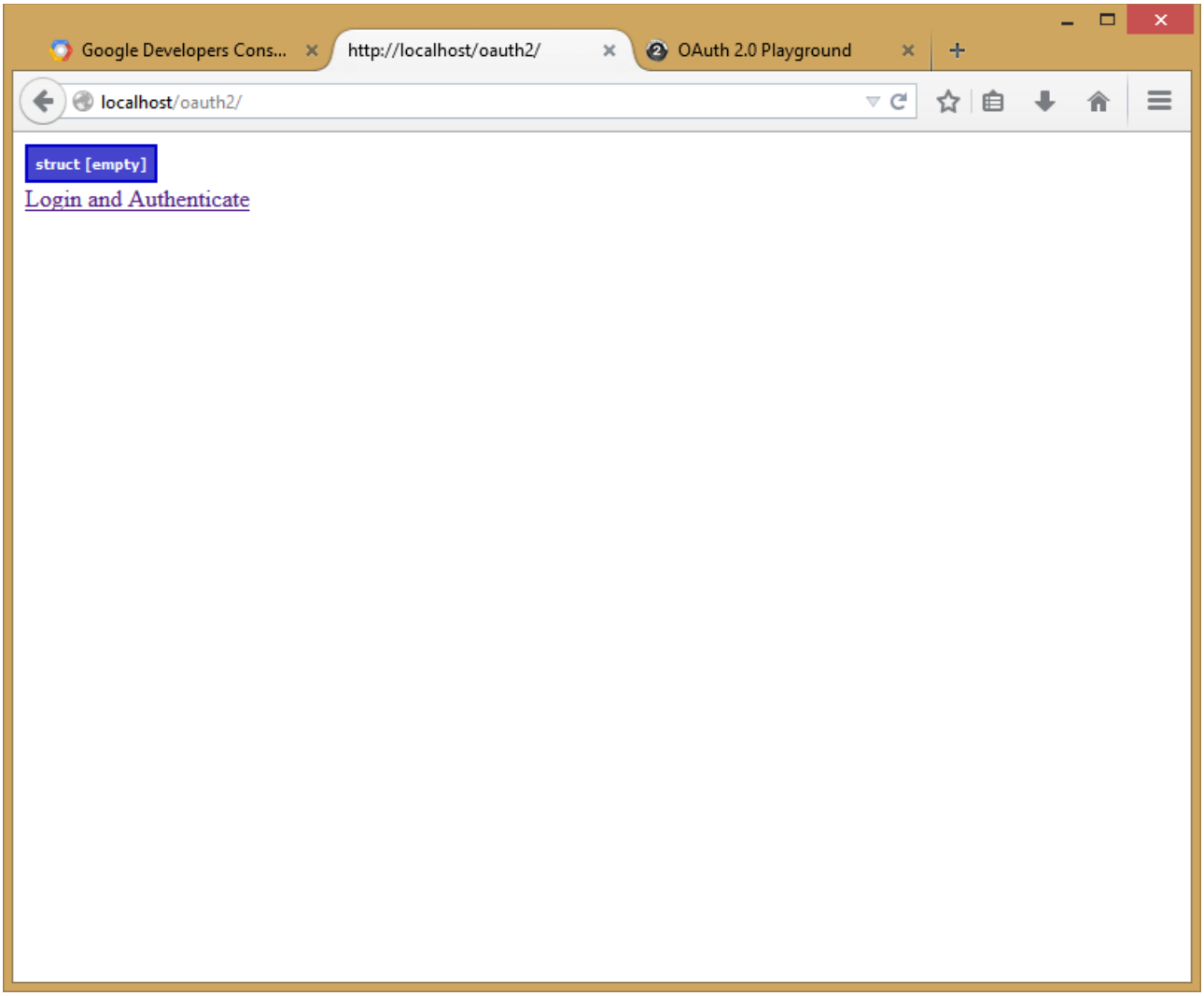
List of scopes: **https://developers.google.com/oauthplayground/**

```
<cffunction name="onApplicationStart"
            access="public"
            returntype="boolean"
            output="false"
            hint="Fires when the application is first created.">

        <!--- remove previous access and struct --->
        <cftry>
        <cfset application.oauth2.revokeAccess()>
        <cfcatch>
        </cfcatch>
        </cftry>

        <!--- CHANGE ME --->
        <!--- Fill in your own client_id, client_secret and scope --->
        <cfset application.oauth2 = new oauth2.oauth2(
            client_id       =   '[redacted]',
            client_secret   =   '[redacted]',
            redirect_uri    =   'http://localhost/oauth2/index.cfm',
            scope           =   'https://www.googleapis.com/auth/calendar',
            state           =   '',
            access_type     =   'offline', <!--- Use offline for refresh tokens
            approval_prompt =   'force'
        ) />

    <cfreturn true />
</cffunction>
```

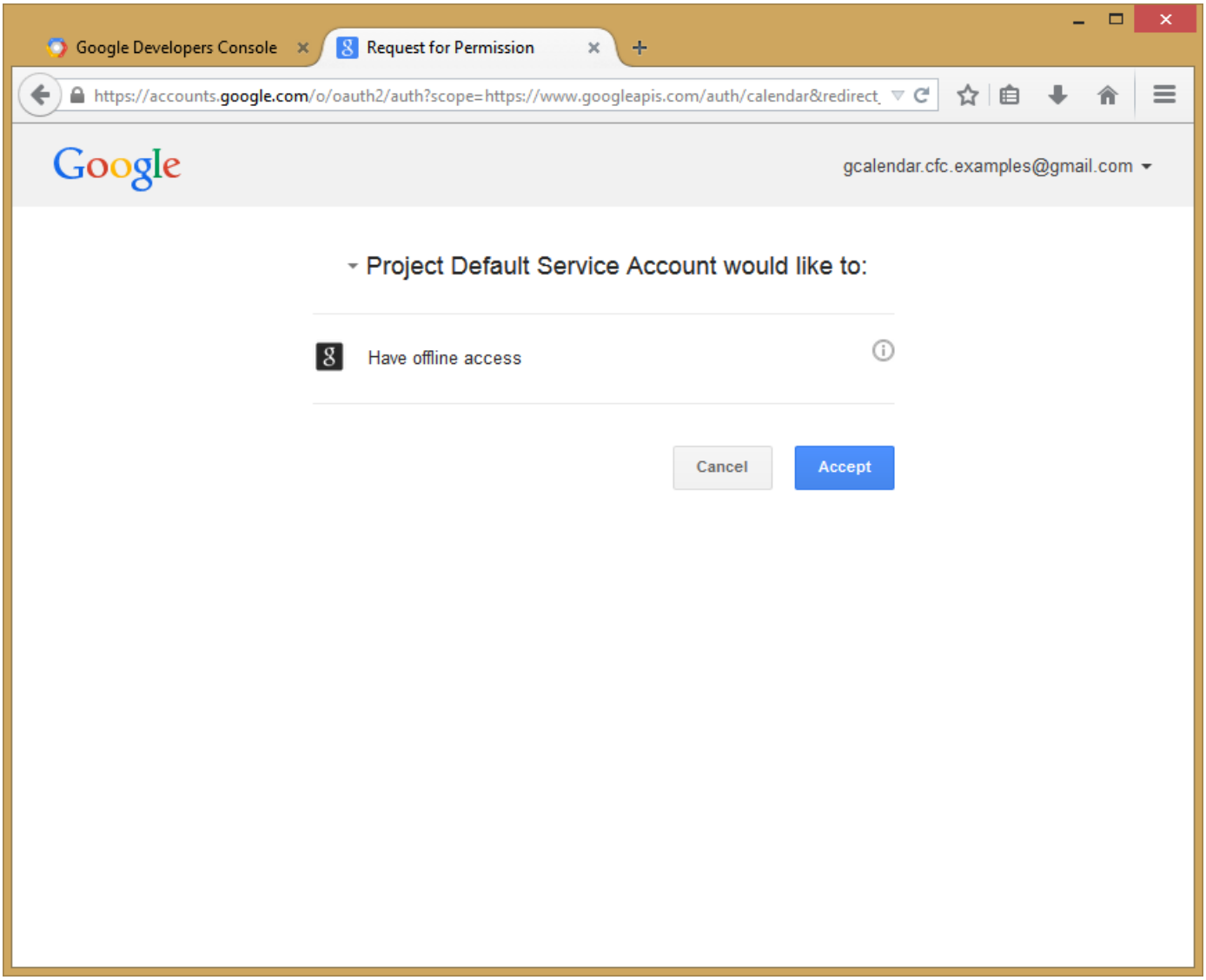
Once this is filled in, let's reinit the application.

http://localhost/oauth2/?reinit=true

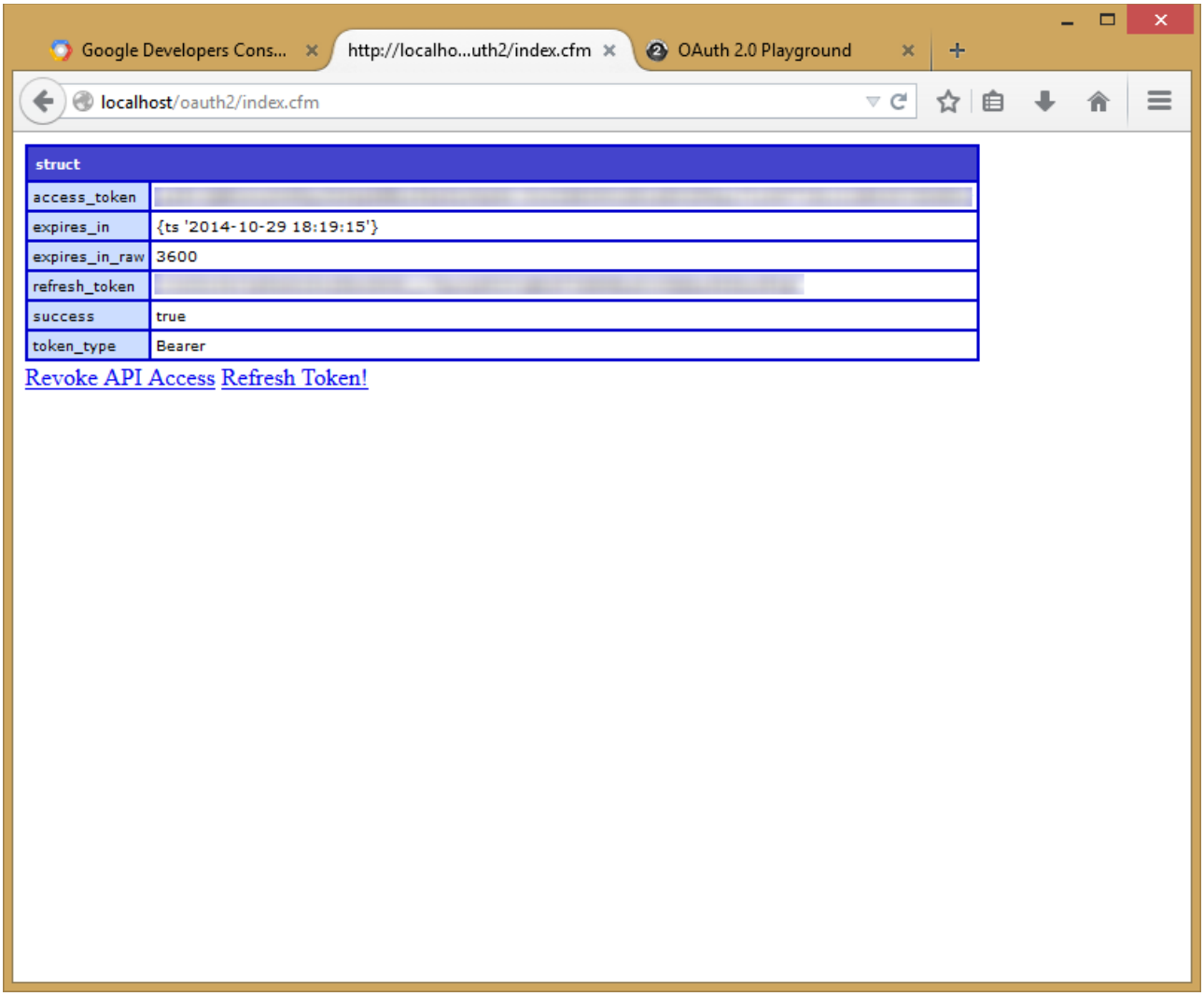
We should now be at the screen shown below.



Make sure at this point that your browser is logged into the Google account After clicking "Login and Authenticate"



After clicking "Accept", you should be at a screen that shows your access. :-)



If anyone is interested in helping me finish the Calendar V3 project before the middle of next month, let me know.