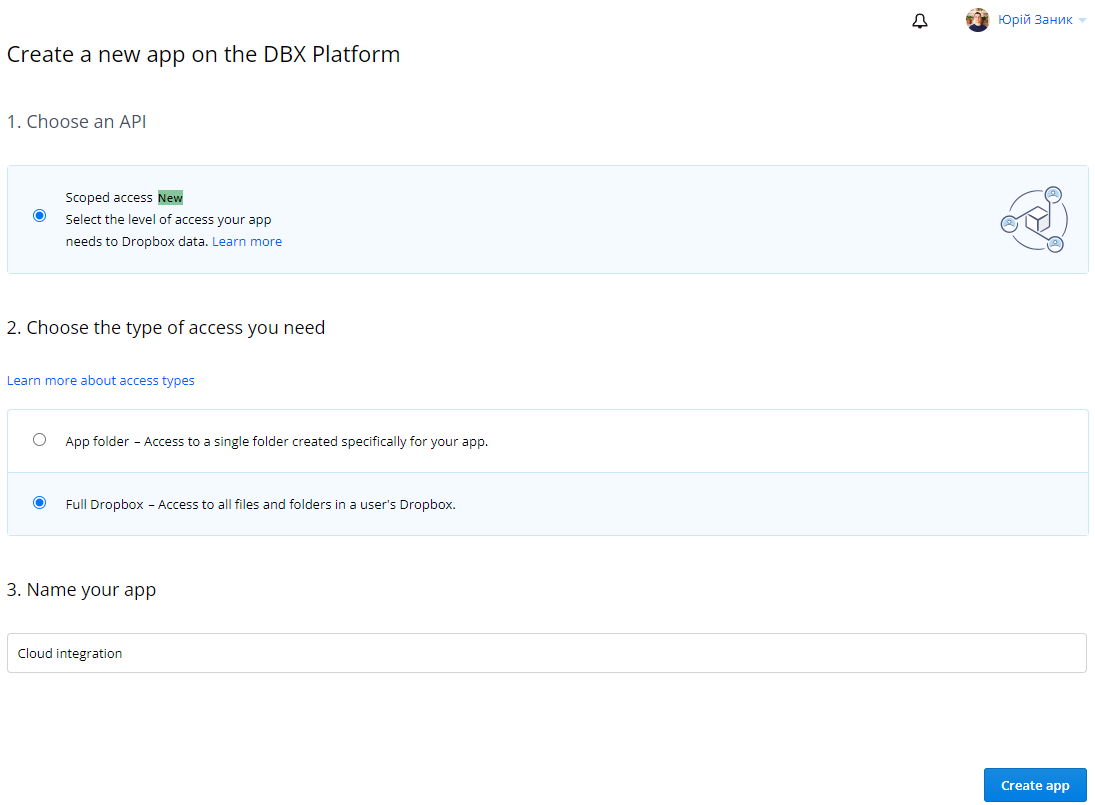
**Cloud services integration (Dropbox, Google drive, OneDrive)**

**Dropbox**

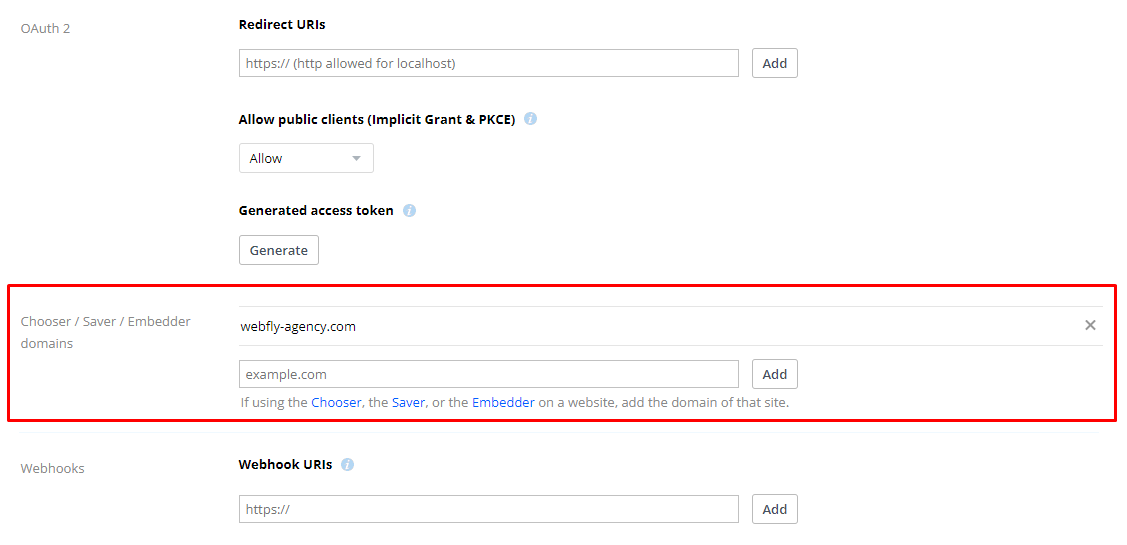
1. Create a new App in [Dropbox App Console](https://www.dropbox.com/developers/apps)



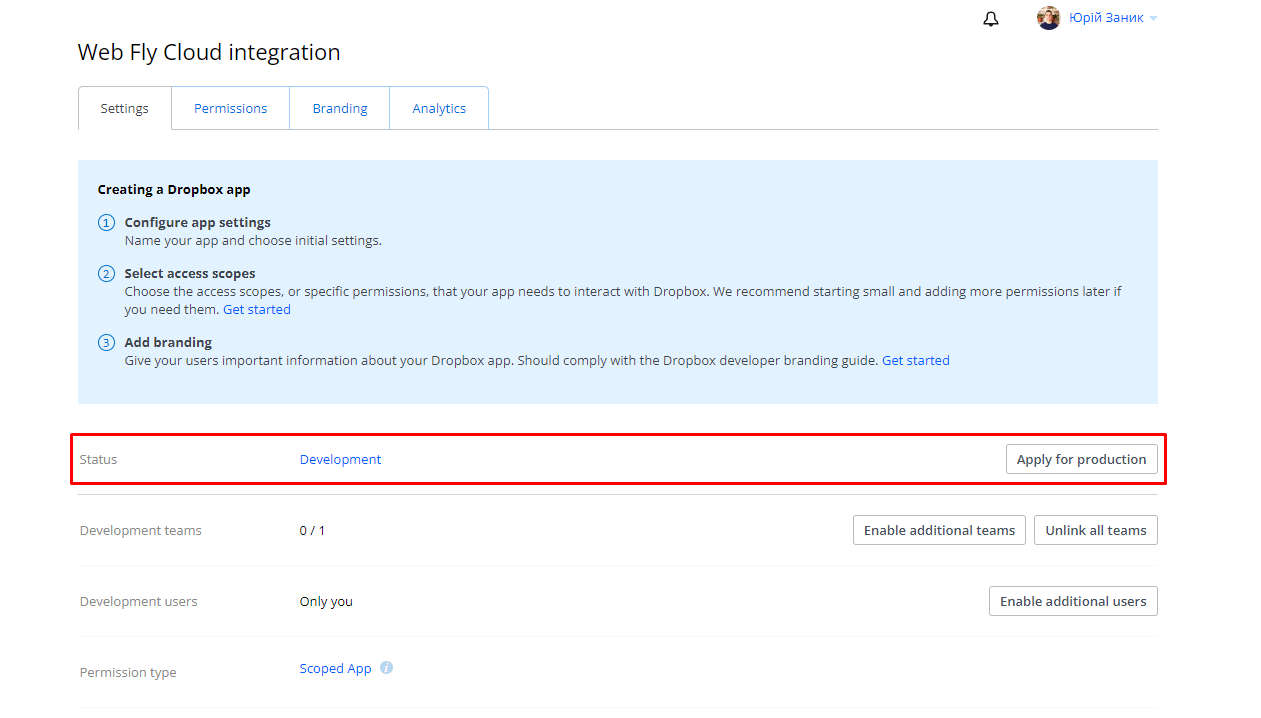
1. Fill needed fields in order to create a new App (Choose an API – Scoped access, Choose the type of access you needed – Full Dropbox, Name you app)



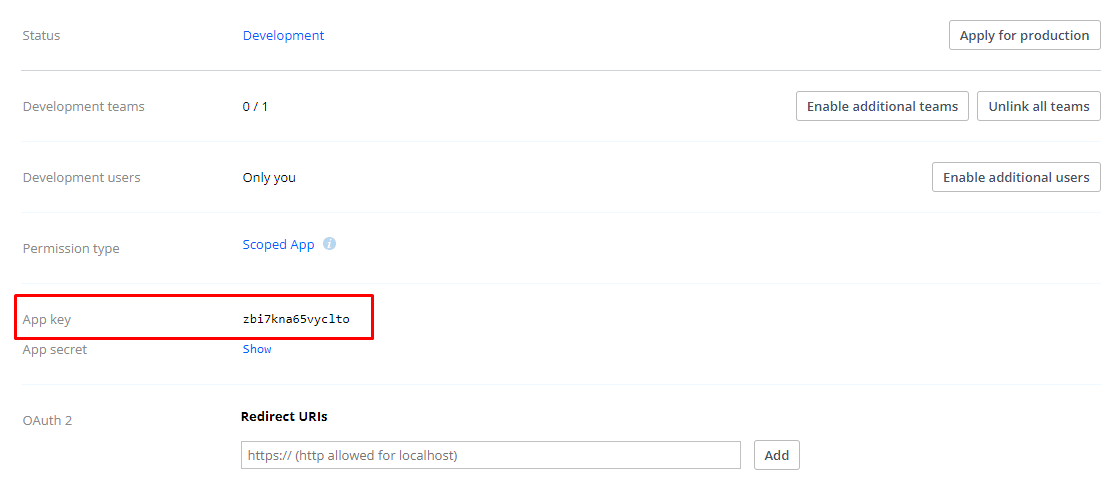
1. After the APP is created you have to set “Chooser / Saver/ Embedder domains” to one that you are going to use.



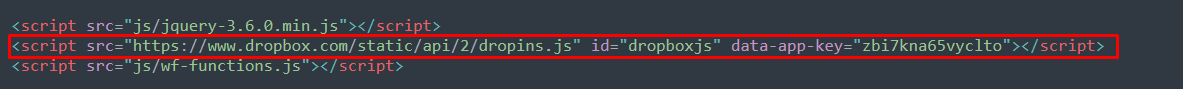
1. You have a “status” field that you have to apply for production after development stage is finished.



1. You can also change Permissions, set Branding and see Analytics (in tabs).
2. Having everything set you can get the app key that you will use while connecting Dropbox.



1. Connect [Dropbox JS File](https://www.dropbox.com/static/api/2/dropins.js) (set id to ”dropboxjs” and for the data-app-key use your app key)



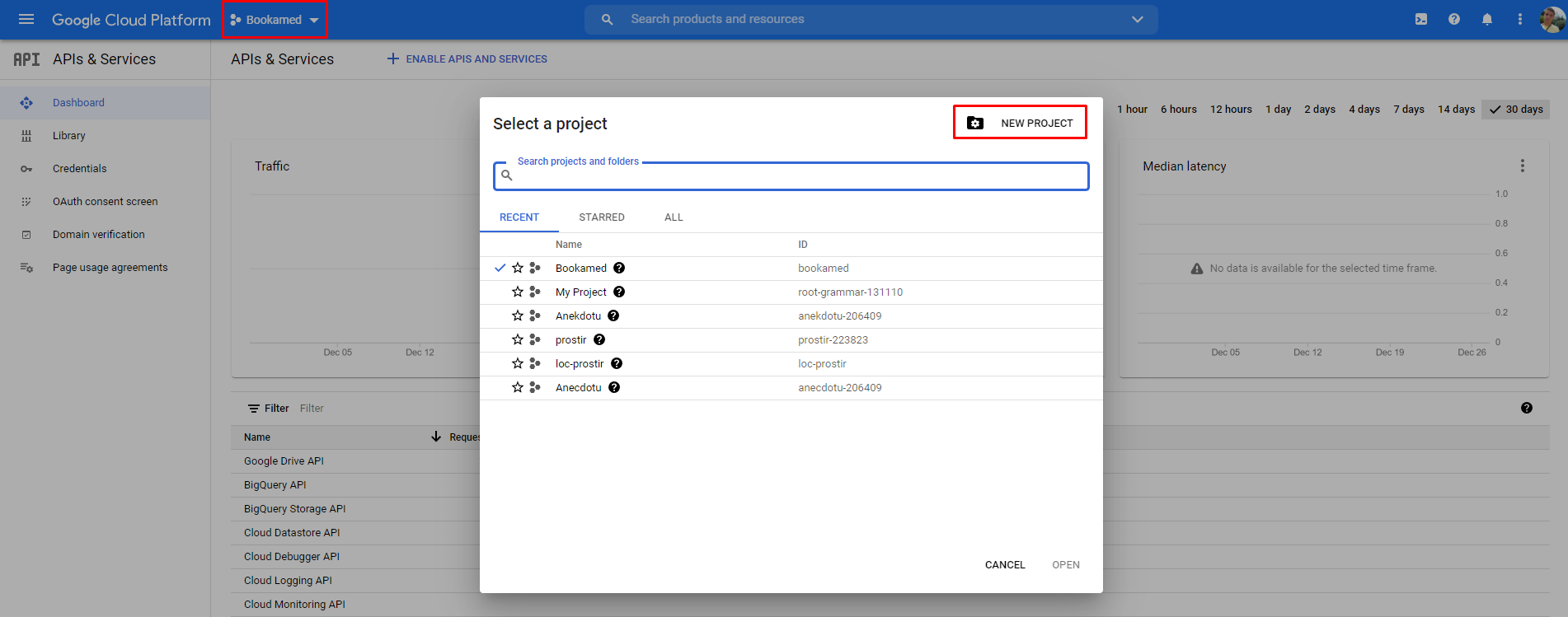
1. Now in JS you have “Dropbox” class and for that class you can use “choose” function with options. Here is an example.



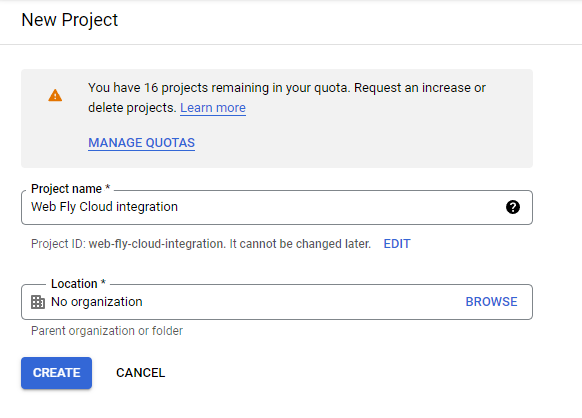
1. In the “success” function you will get an array of files with file name and downloadable URL.

**Google Drive.** [**Documentation example**](https://developers.google.com/picker/docs)

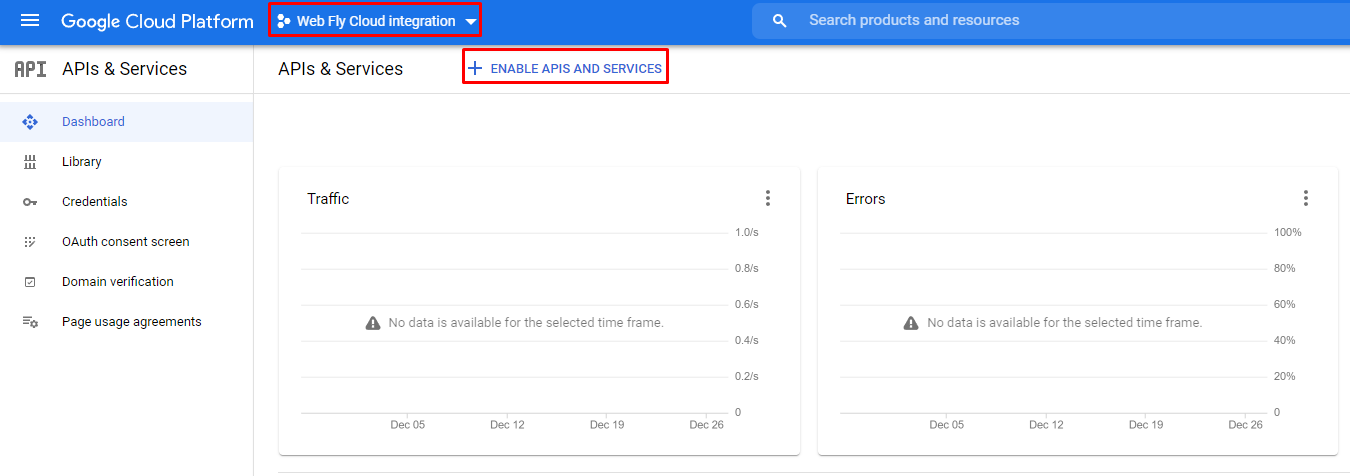
1. Create a new Project in [Google Developers Console](https://console.cloud.google.com/apis/dashboard)

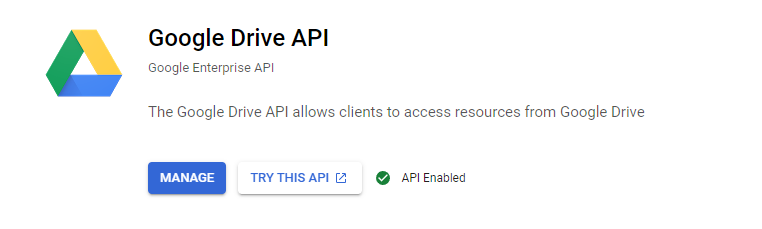


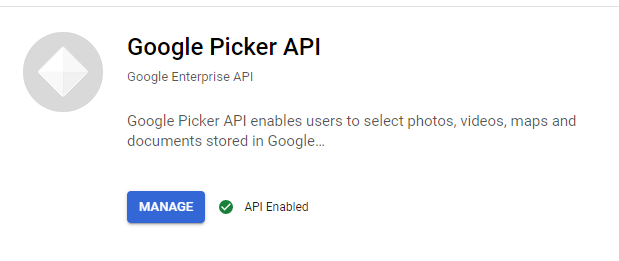
1. Set the name of the project and select organization if needed



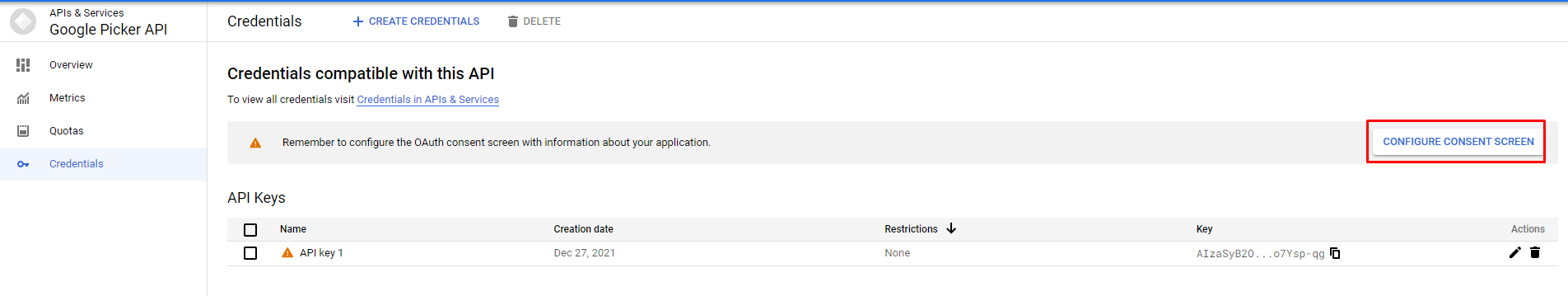
1. Enable Google Drive and Google Picker APIs for the project



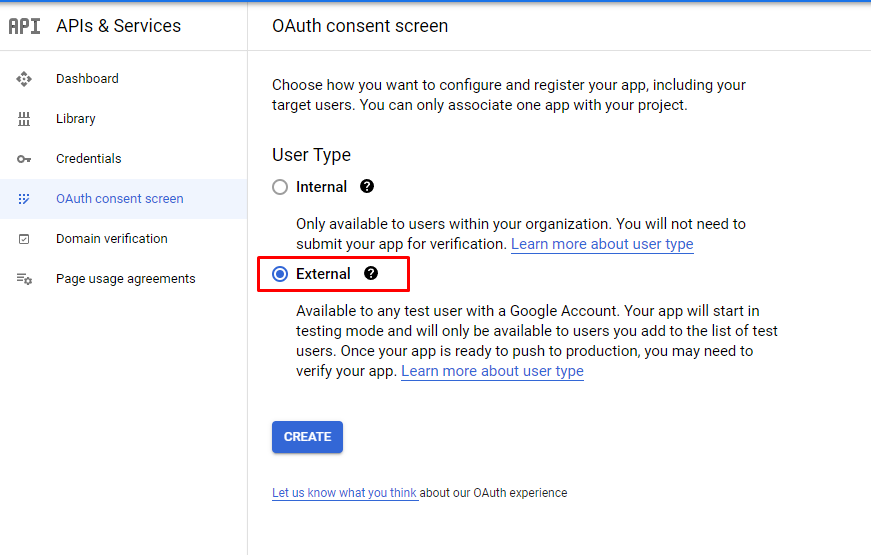




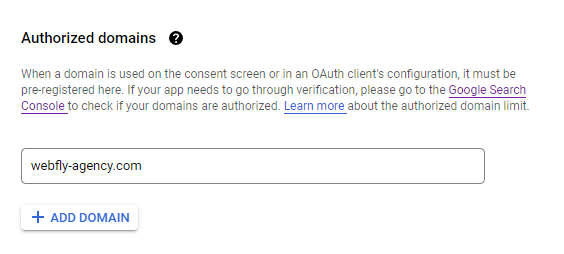
1. Now with having Google drive and Google Picker APIs enabled you have to configure consent screen with information about the application.



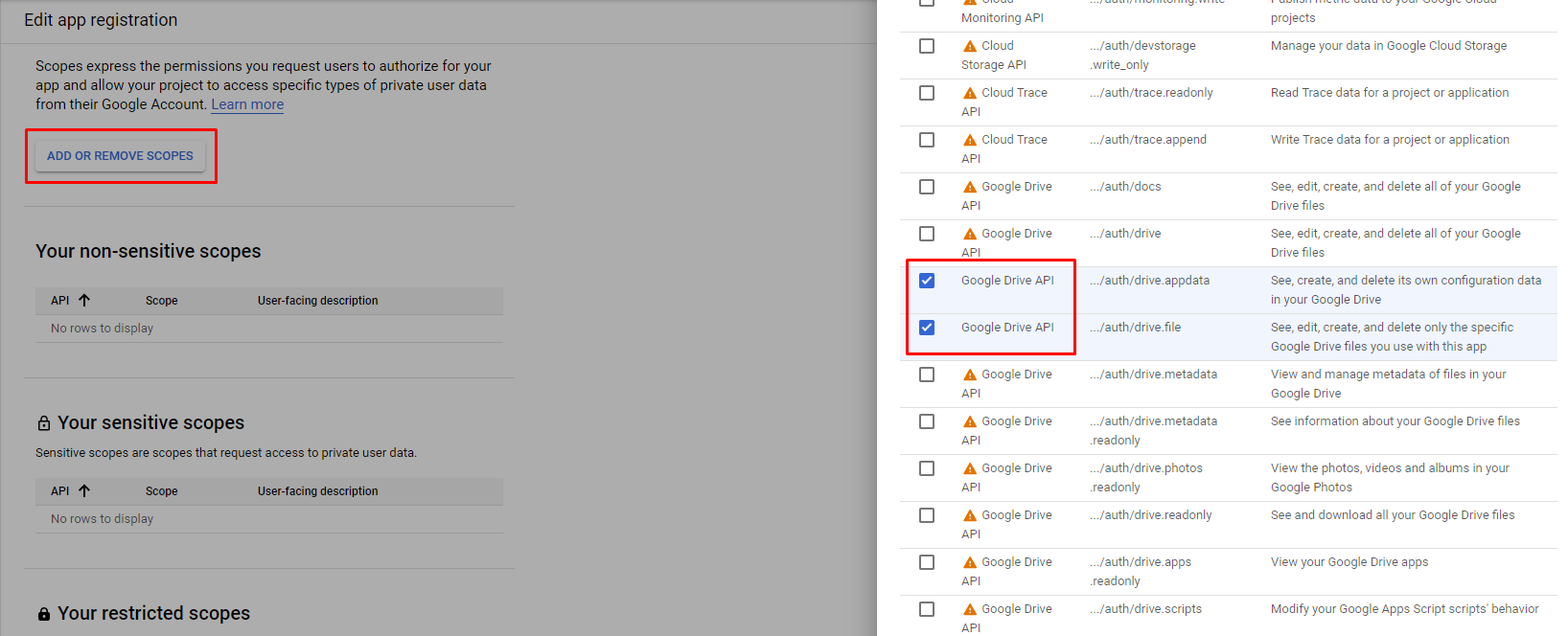
1. Make it “External” unless it is within the organization only



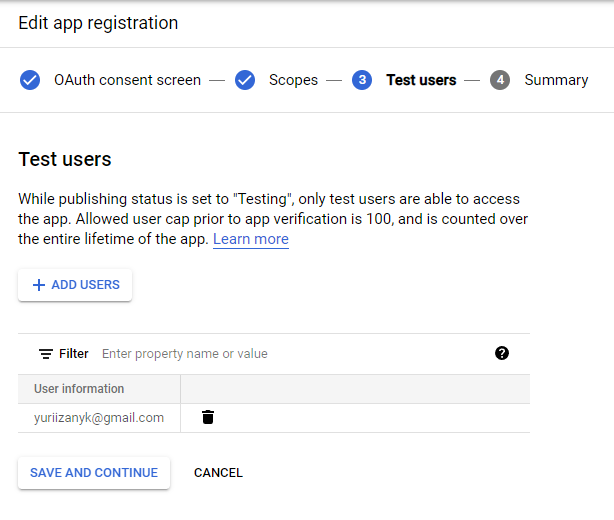
1. Fill all required fields. Don’t forget to set Authorized domains to the one that will be used for this integration (could be a root domain instead of subdomain if needed, the root domain will work for all subdomains)



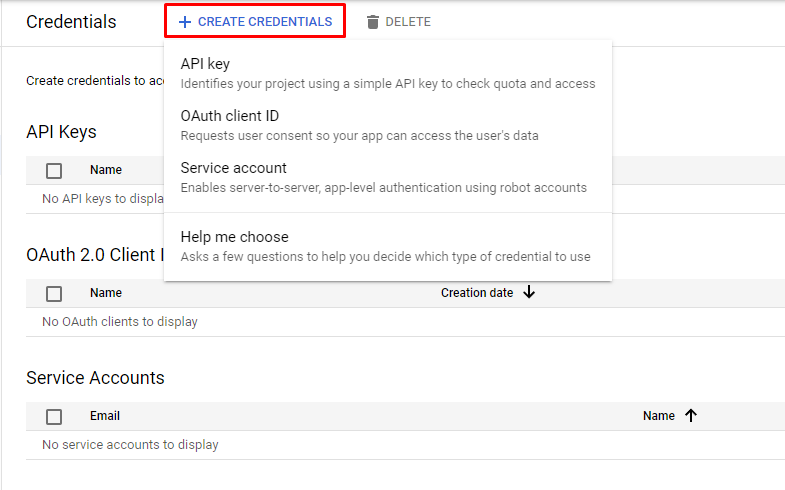
1. Add Google Drive API scope



1. In the “Test users” step add the user you are going to use for testing. While app is not verified only test users will be able to use integration.

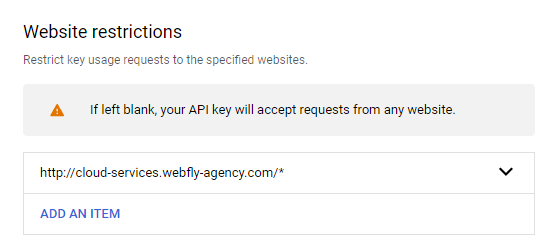


1. You need have API Key and OAuth client ID and create them you can by clicking “+ Create Credentials”.

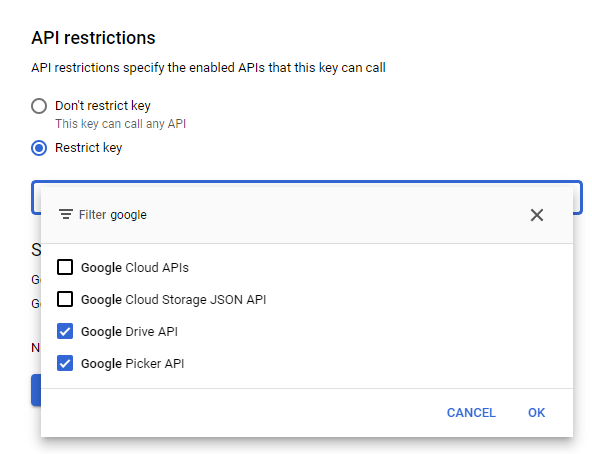


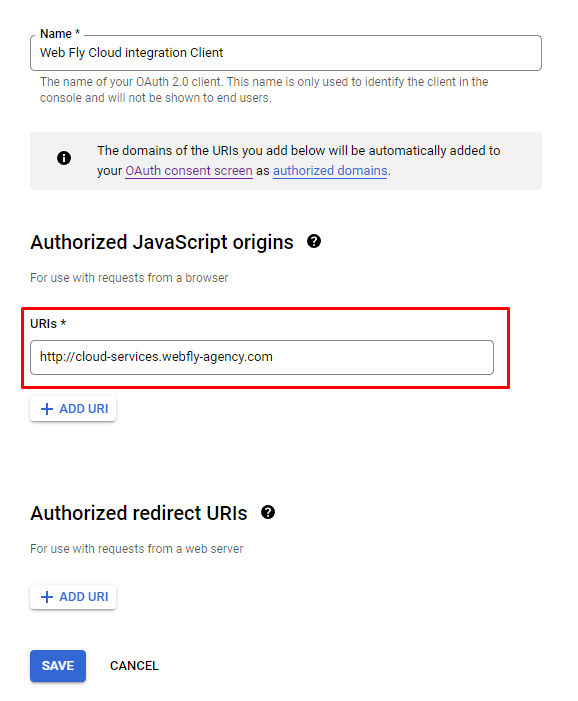
1. Also for API Key it is recommended to set API restrictions. In our case it will be

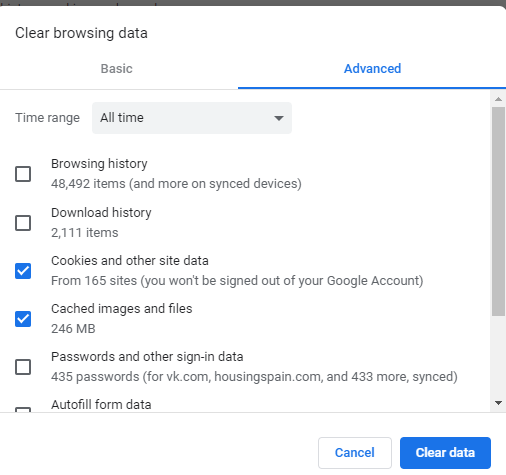
* Application restrictions – HTTP referrers (web sites)
* Website restrictions – root of the domain URL with \* after last /



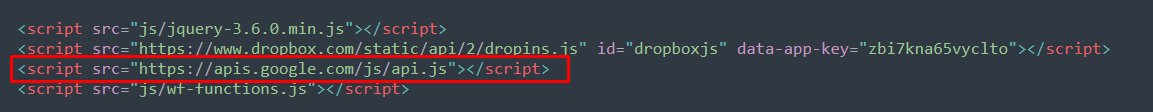
* API restrictions – Google Drive API and Google Picker API



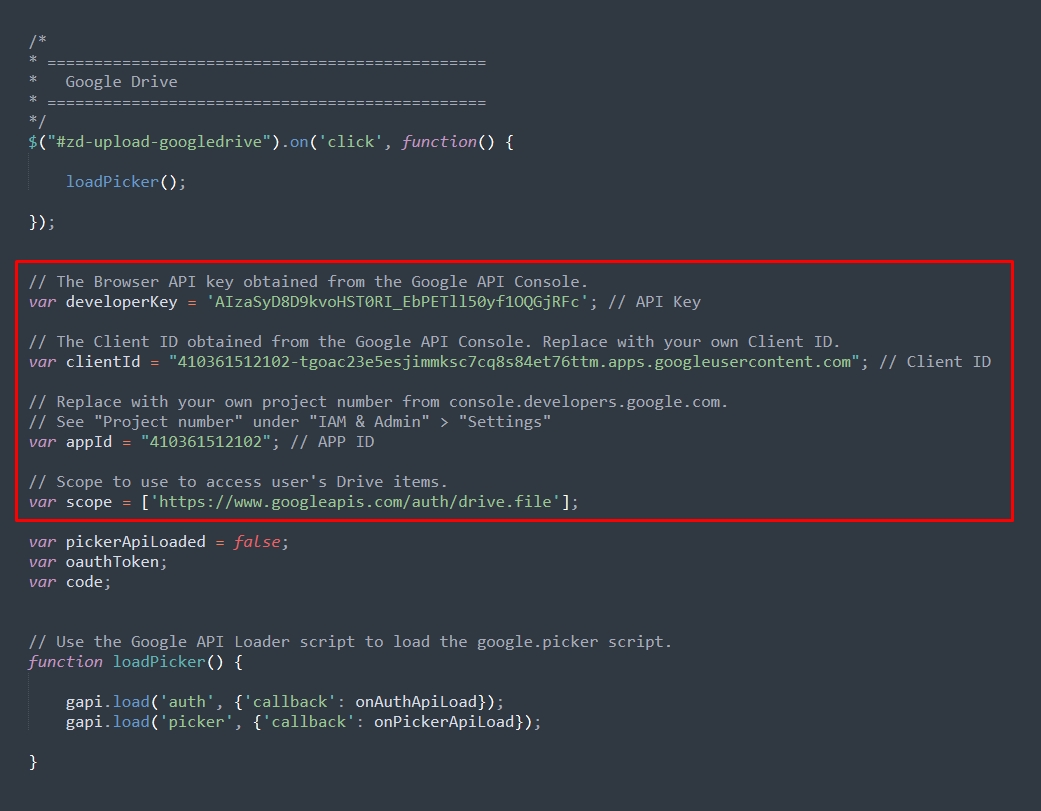
1. At the same time for the OAuth client ID you have to set Authorized JavaScript origins to the project’s home page.
2. Before using API Credentials you have to clear cookies or use another browser (Firefox)

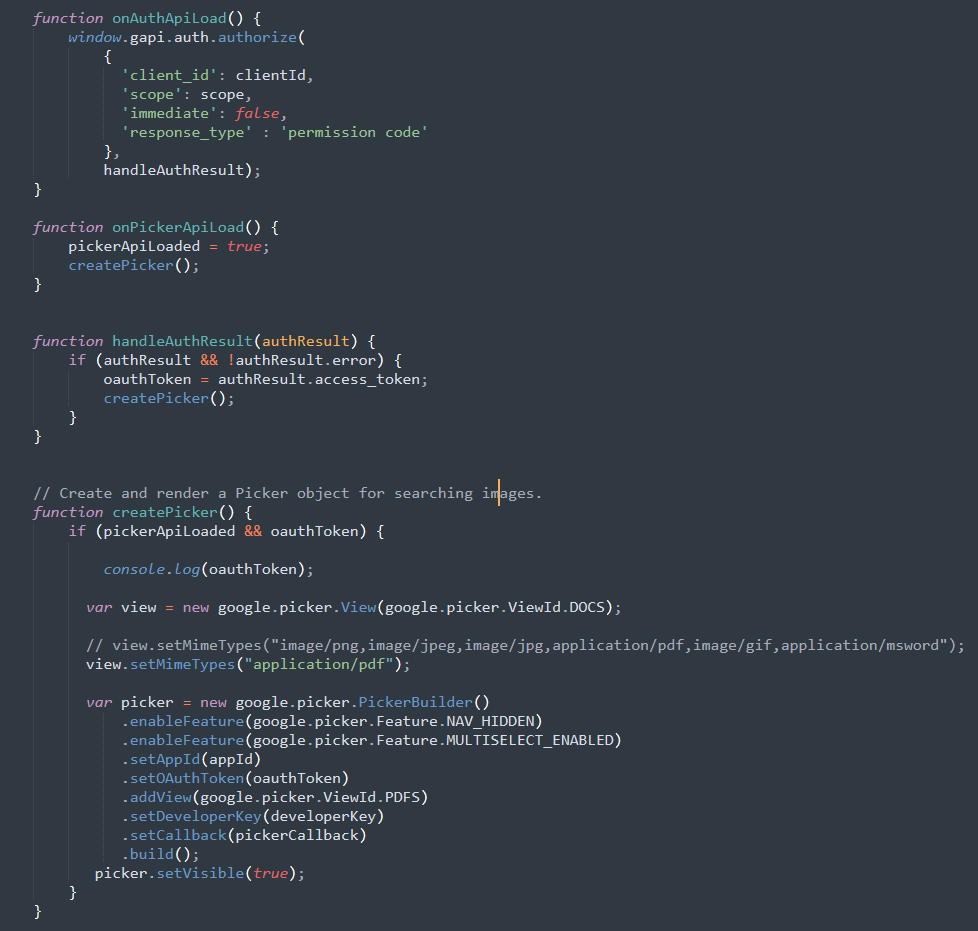


1. After you created 2 needed Credentials we’re moving to the coding part. First of all you have to connect [Google API JS file](https://apis.google.com/js/api.js)



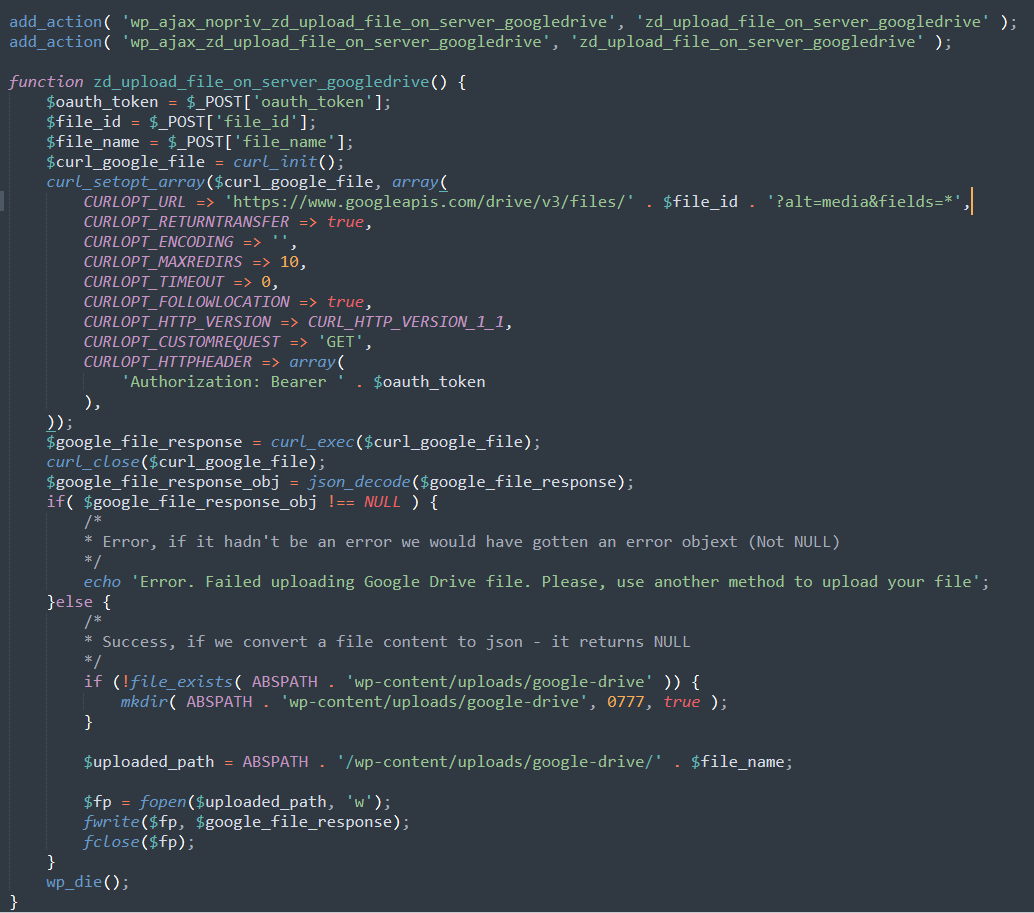
1. In the code you have to set variables to ones that we created in the Google Console and use code from the documentation example. Here it is.





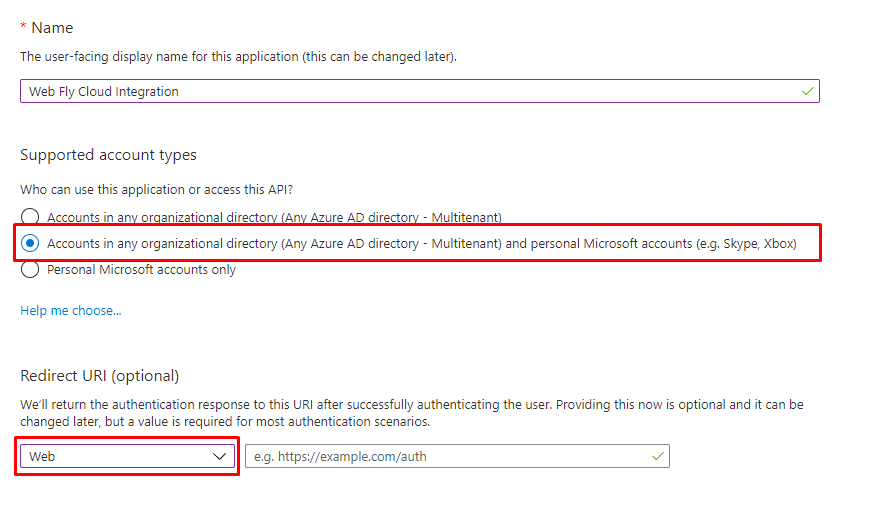


WP AJAX to upload file by ID to the server



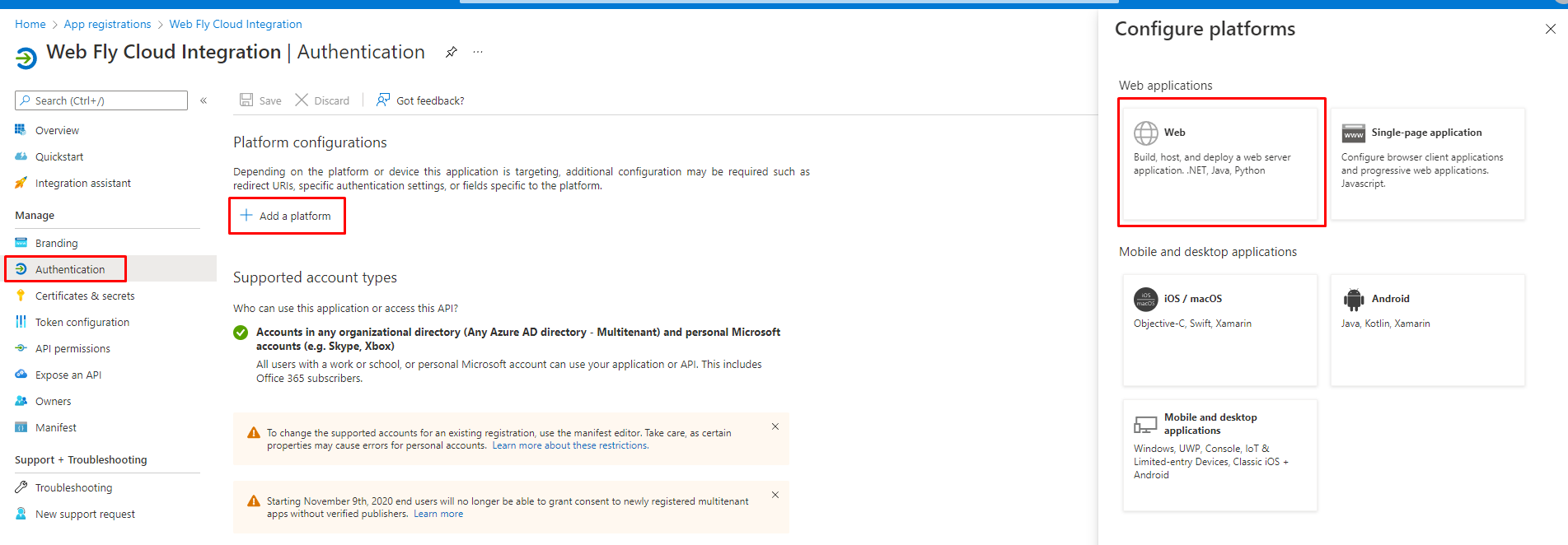
OneDrive

1. Create a new APP in [Microsoft Azure Portal](https://aka.ms/AppRegistrations).
2. Set the name and select support account types to “Accounts in any organizational directory and personal Microsoft accounts”. That is important. Also, you can restrict it for Web only and it is required to set redirect URI – the one that you will use to log in (add picker) on the website but you can always add it later too (see next step).

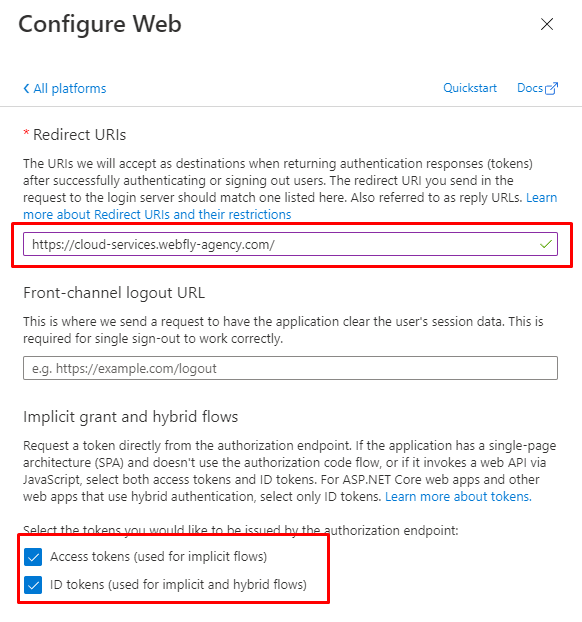


1. You have to add Redirect URIs (pages that you will use to log in with Microsoft accounts). The protocol should be https.

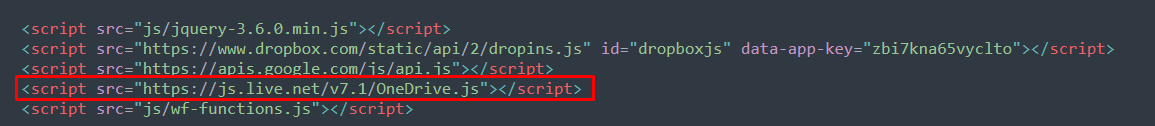
Go to Authenitcation -> “+ Add a platform” -> “Web”



1. Provide Redirect URIs and check both tokens to be issued by the authorization endpoint.



1. After you created an APP we can move to coding part. First of all you have to connect [OneDrive API JS file](https://js.live.net/v7.1/OneDrive.js)



1. After it insert this part of code. We will use OneDrive object which exist our we connected the JS file in the step above and “open” function to open a picker.

