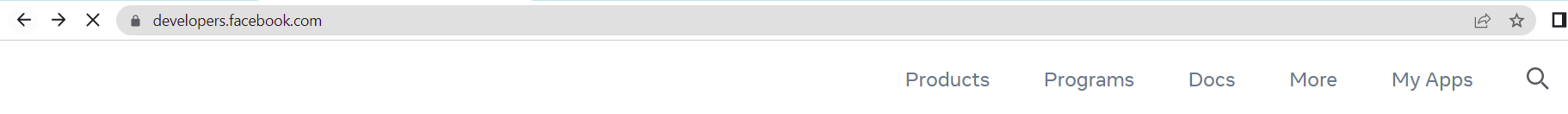
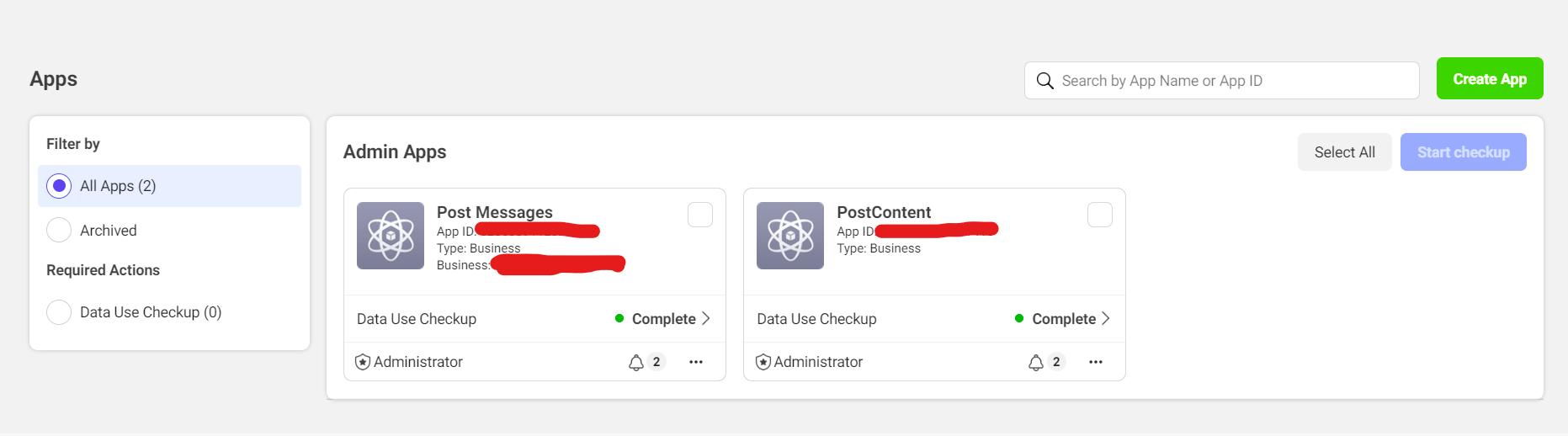
Documentation for USMP

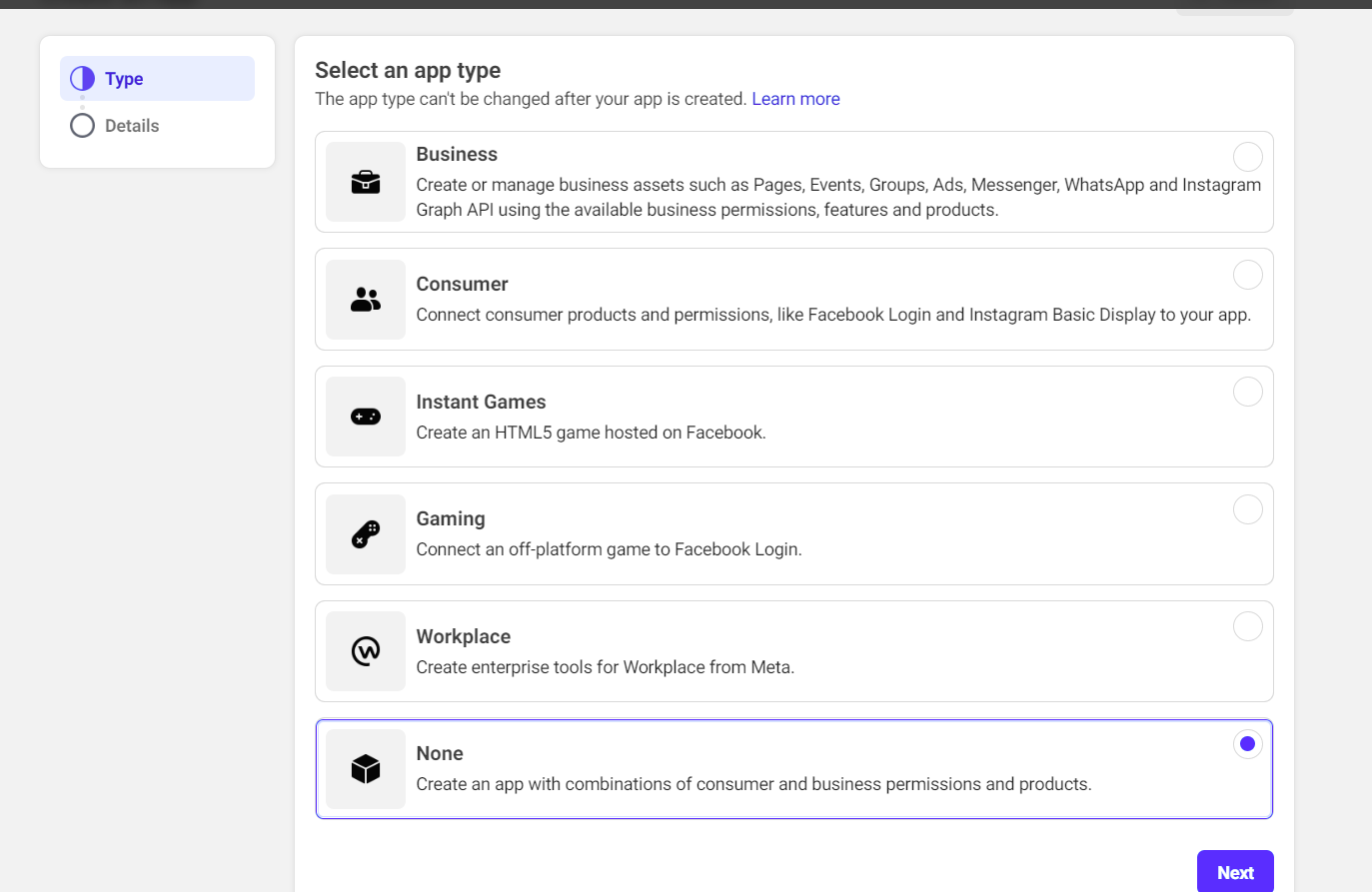
* Interaction with social media controllers

**For developers that want to create an app for Facebook and Instagram on the Meta For Developers site:**

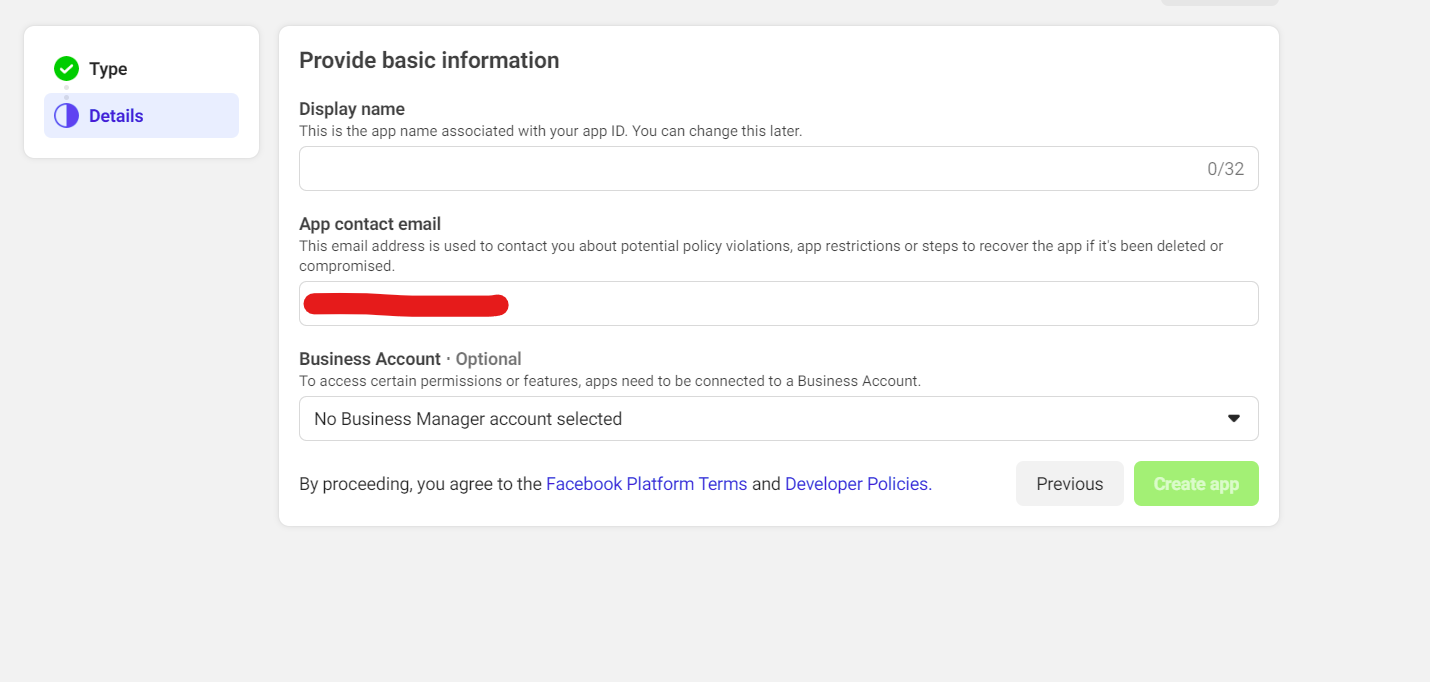
* Create an account on the Meta For Developers site.
* Go the My Apps portion of the Meta For Developers site.



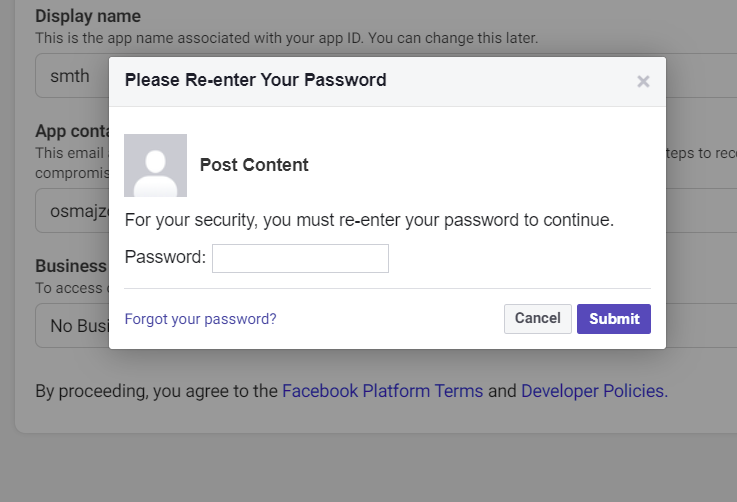
* Press on “Create App”.
* Choose None as the type.



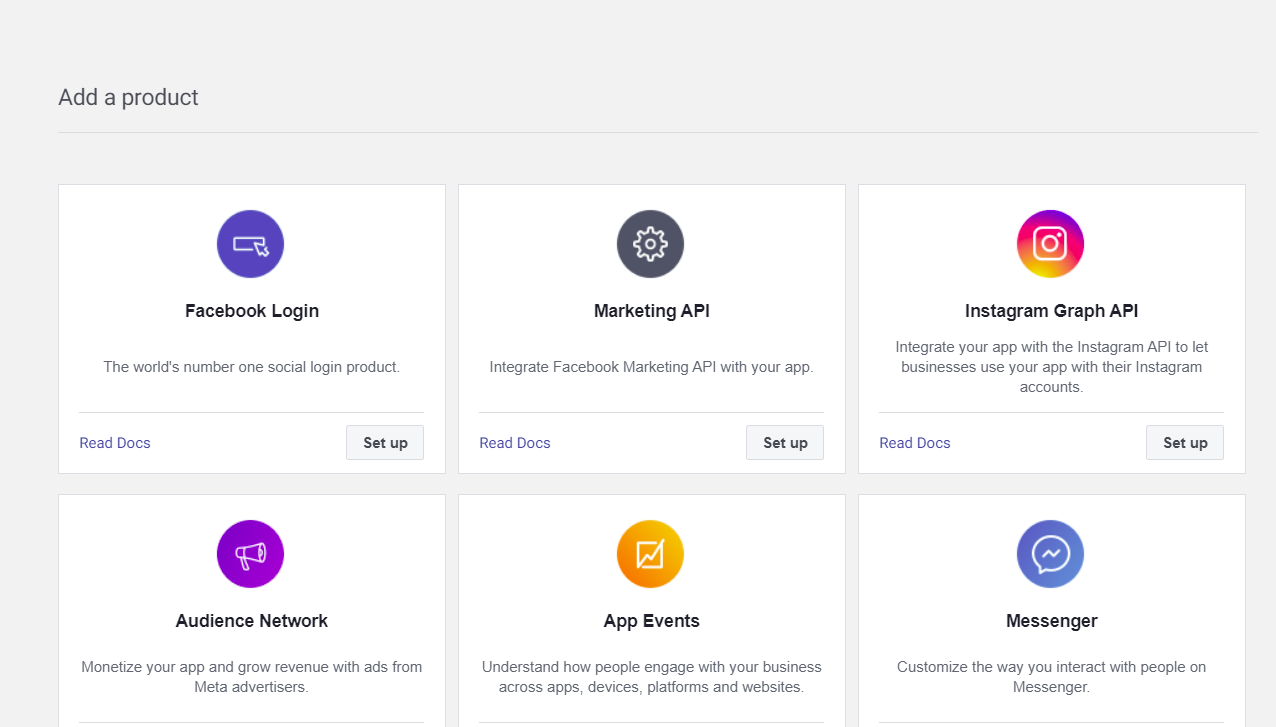
* Fill in the basic information and then press on “Create app”.



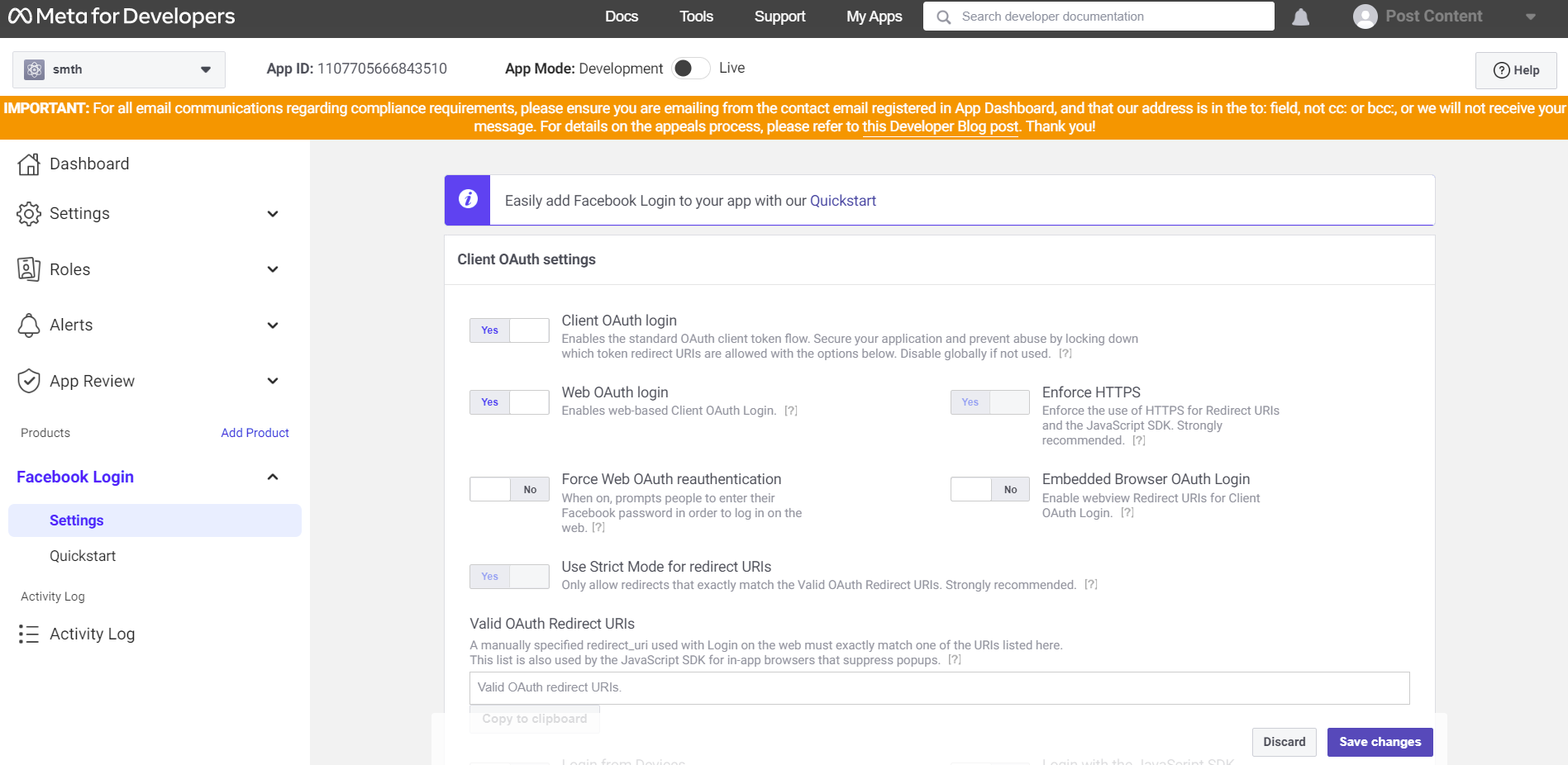
* Re-enter your password if prompted to.

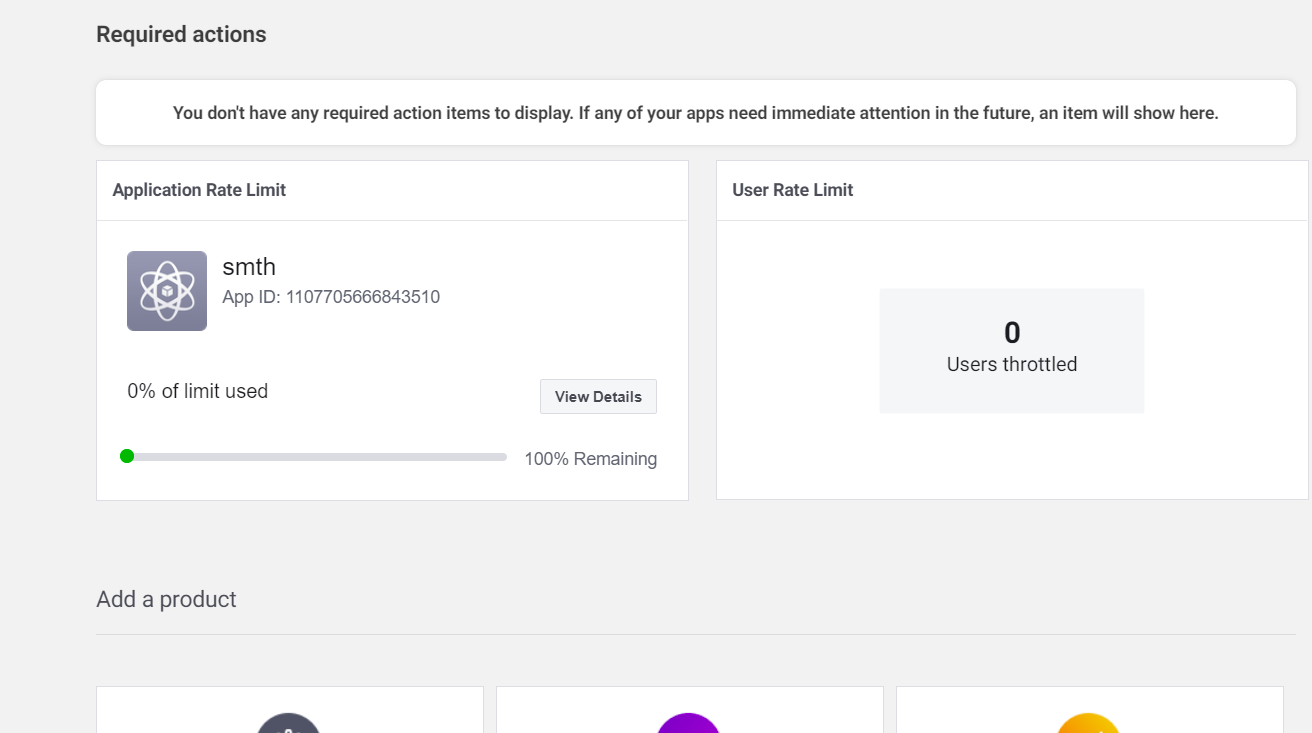


* Add the Facebook Login and Instagram Graph API products by setting up each of them.

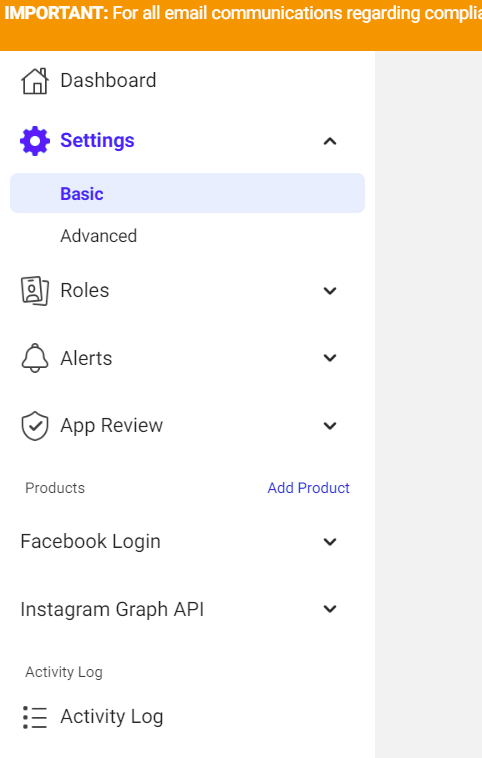


* You can start with Facebook Login where you can add Valid OAuth Redirect URIs (note: <http://localhost> endpoints are enabled by default in development mode for non-business type of apps) and then add Instagram Graph API by pressing on “Add Product” in the Products section, finding it under “Add a product” and pressing on the “Set up” button of that product.





* You can get the App ID and App Secret from the Basic section of Settings.

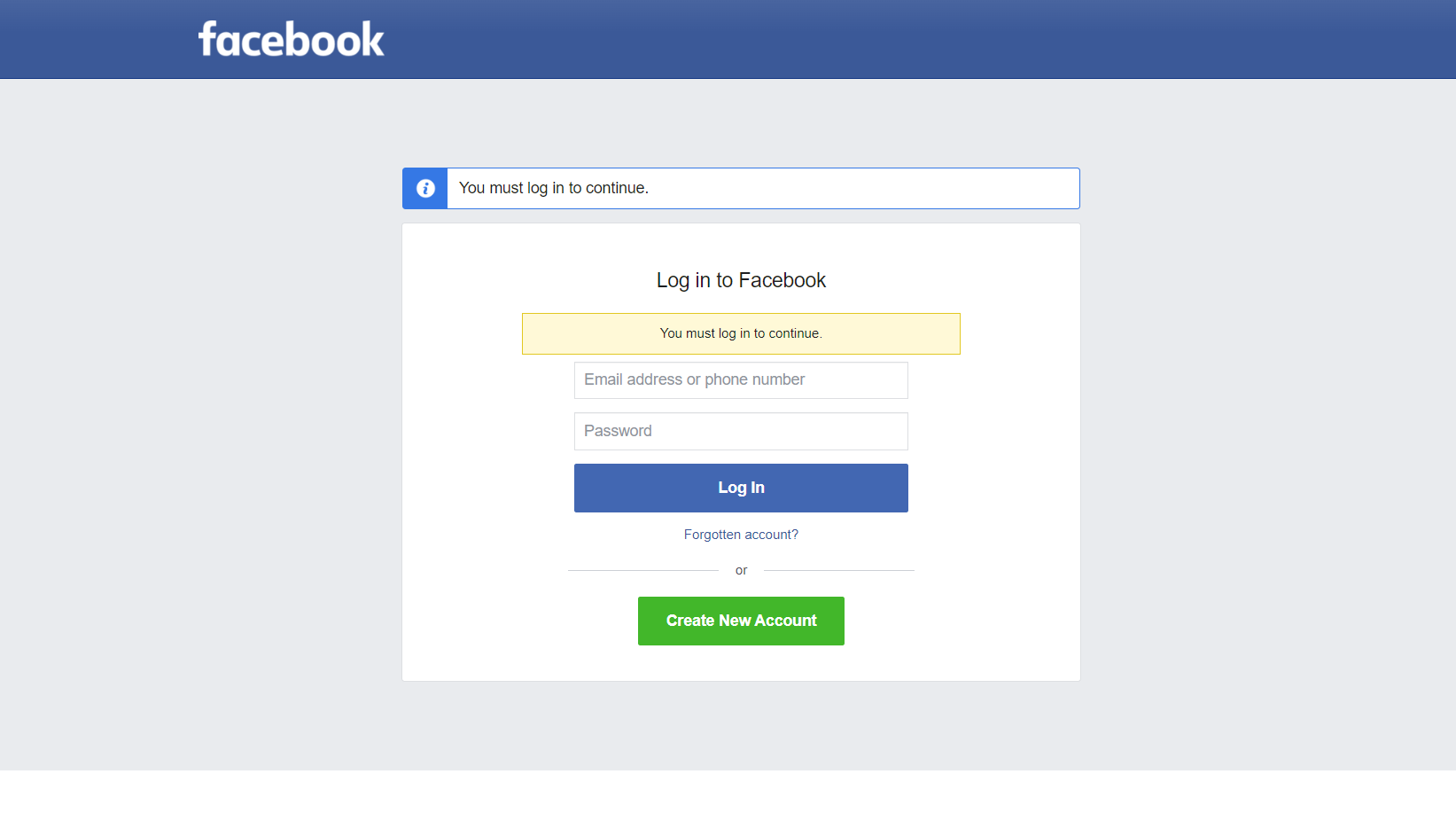


**To create an app for WhatsApp:**

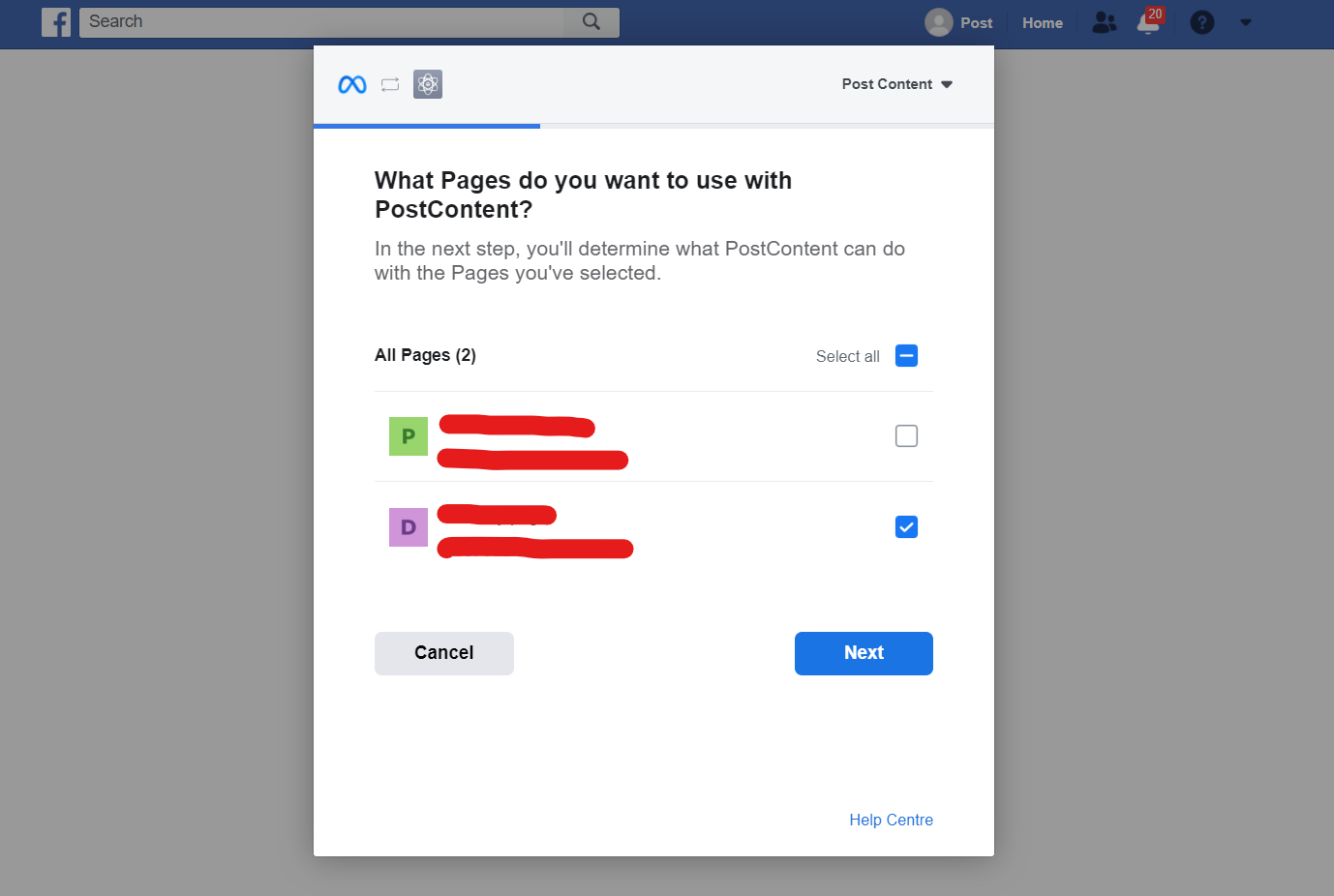
* Proceed in a similar fashion as the case for creating an app for Facebook and Instagram, but choose the type to be “Business” and choose the “WhatsApp” product. If you want to have Facebook Login for users of your app, please add the “Facebook Login” product as well.

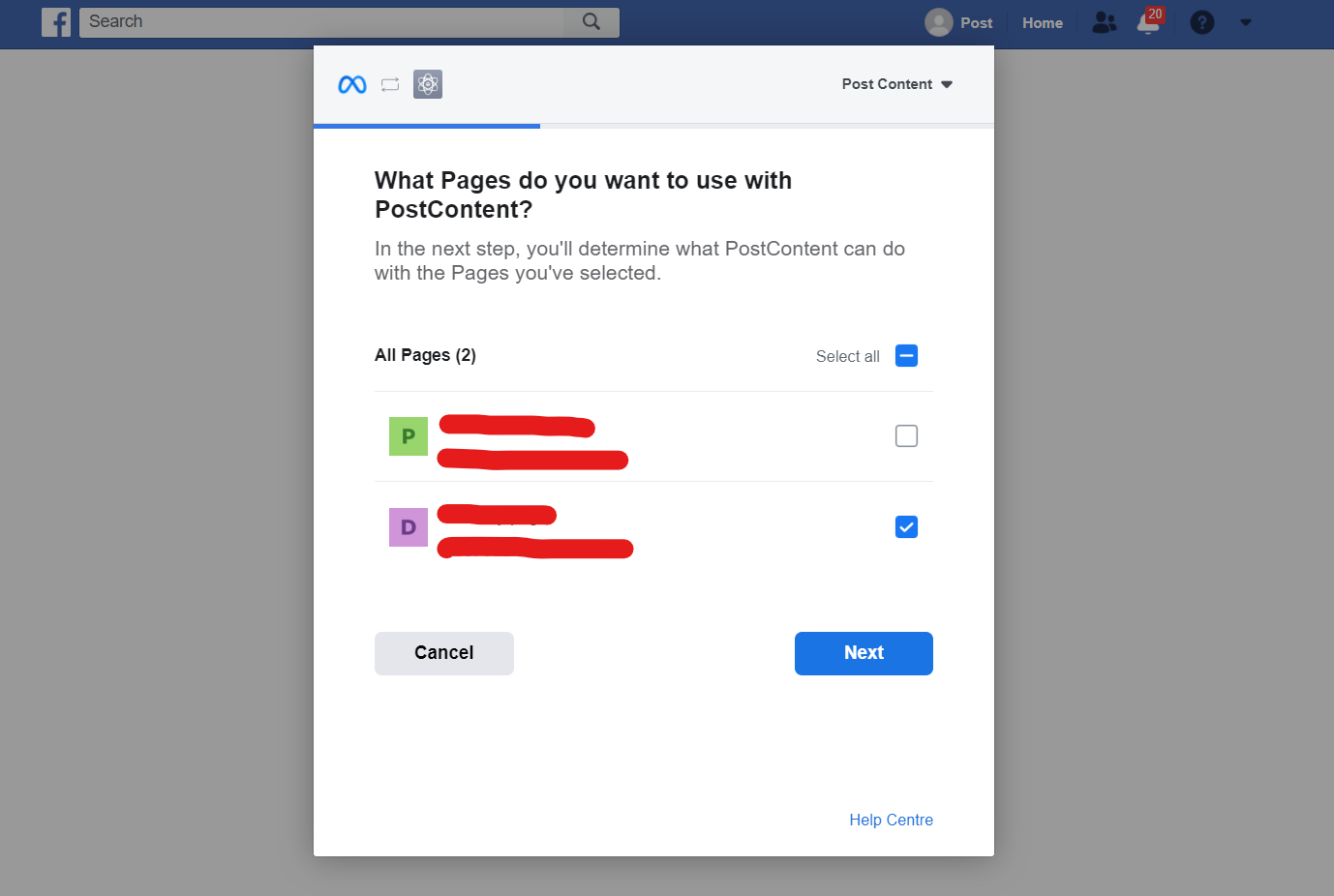
1. Facebook:

* To allow the API to post to your Facebook page, you must send a POST request to the /Facebook/Login endpoint of the API with a redirect Uri in the body. This call should return a URL. Please visit this URL. From there, you will be asked to log in to Facebook.

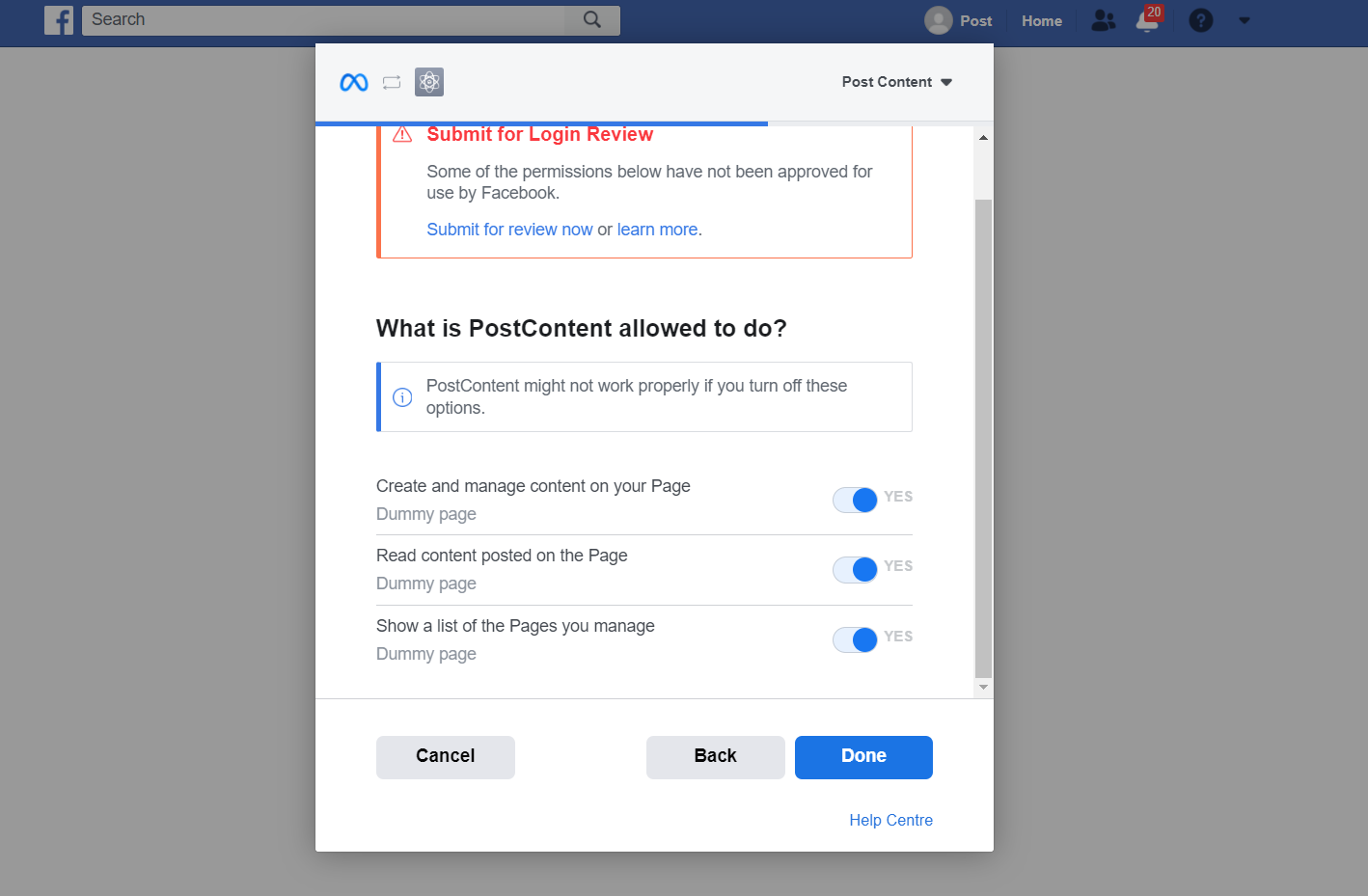


* Next, you must select the page you would like the API to post to. Please copy your page ID located below the Page name for later use in other requests made to the Facebook portion of the API.

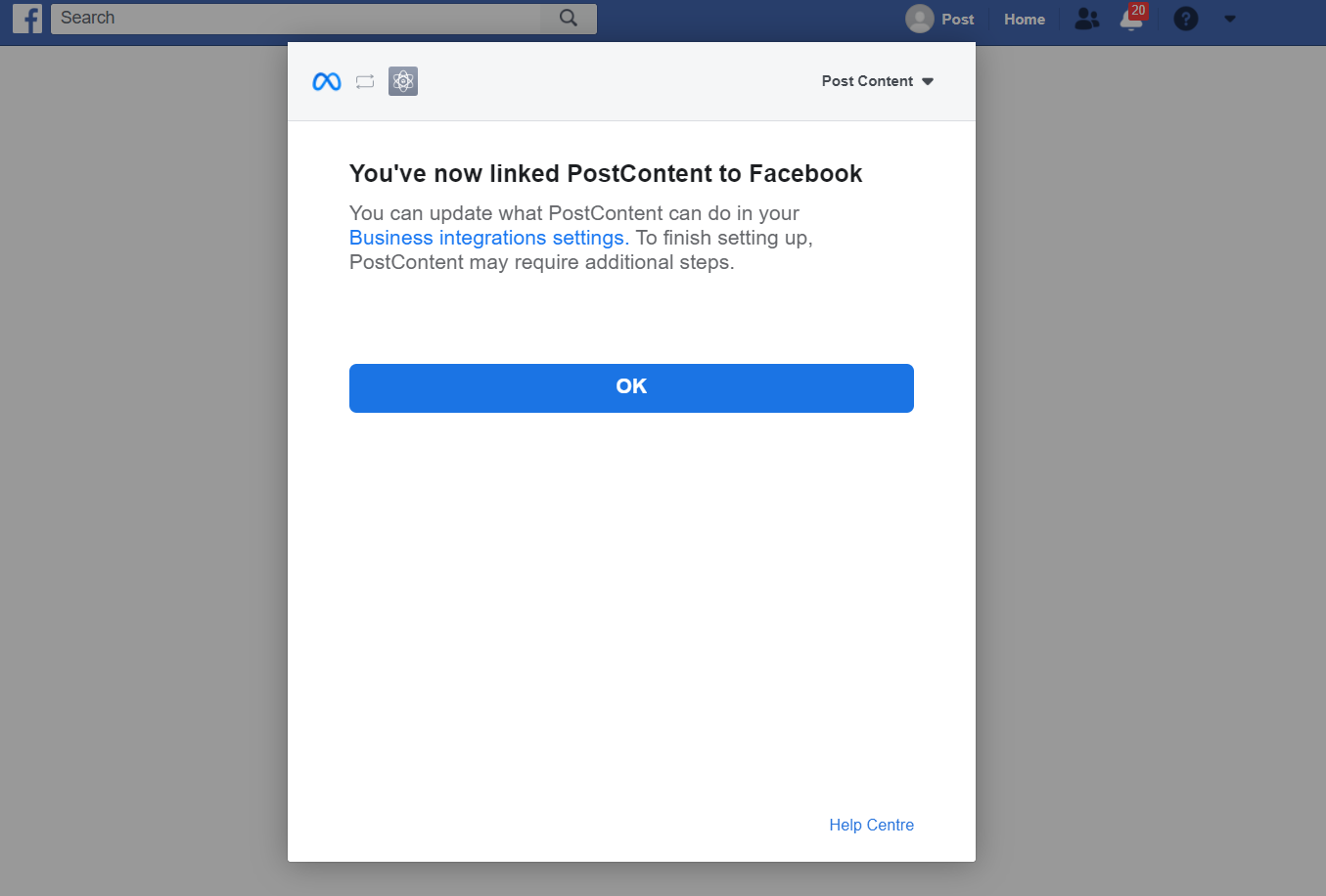




* Afterwards, you should choose the permissions that you want to grant to the API app and press “Done”.



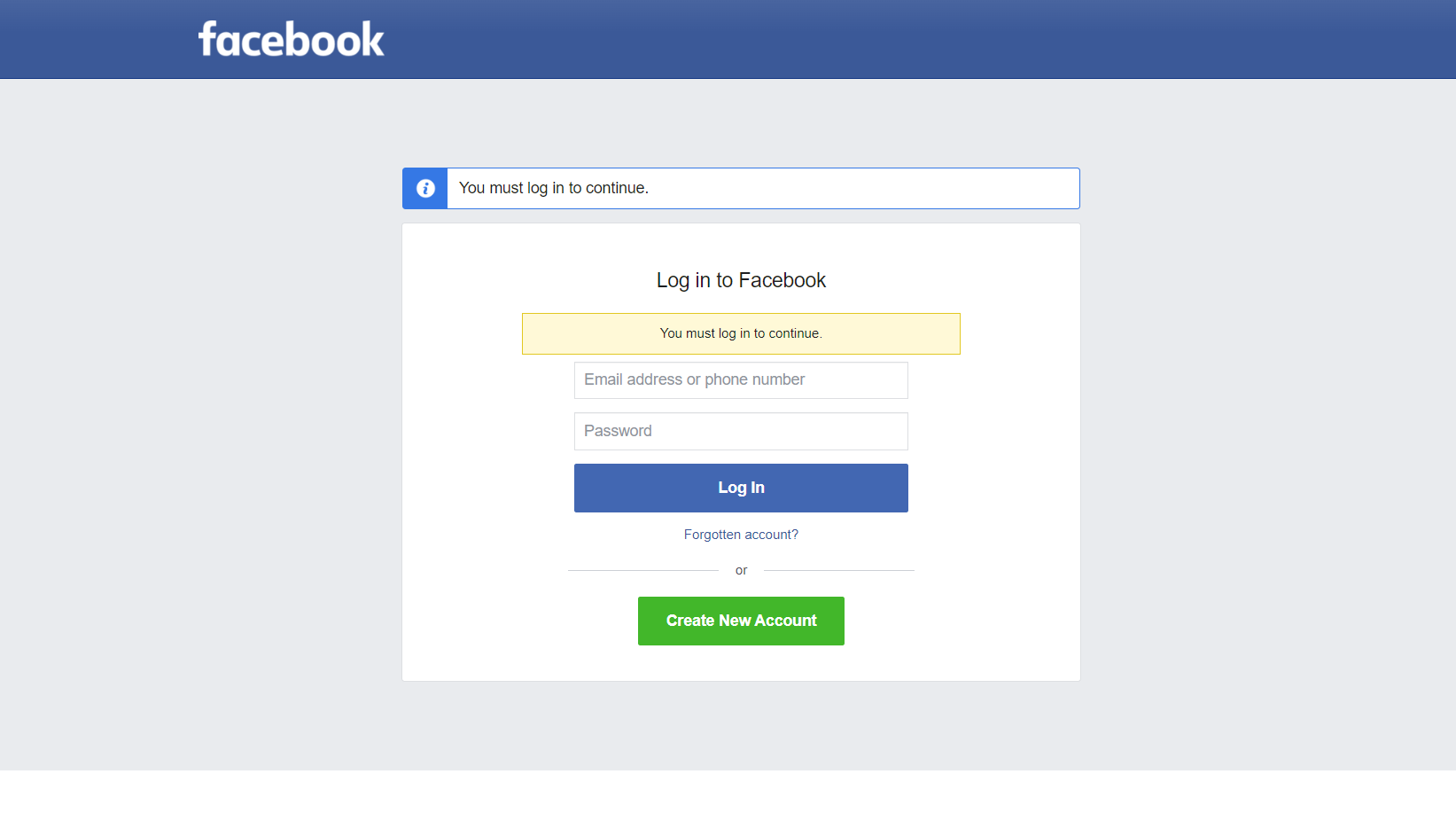
* On success, you should see the following:

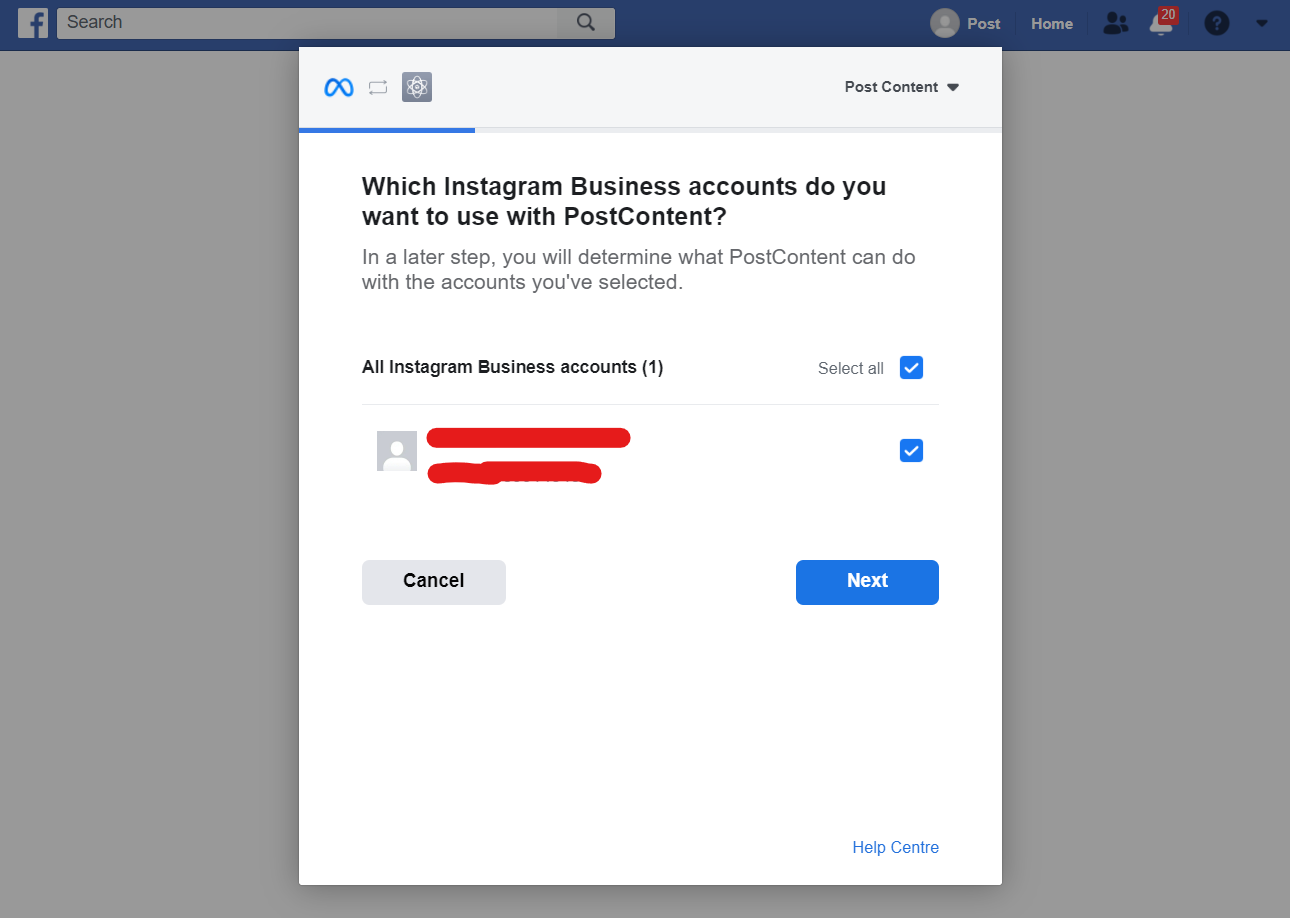


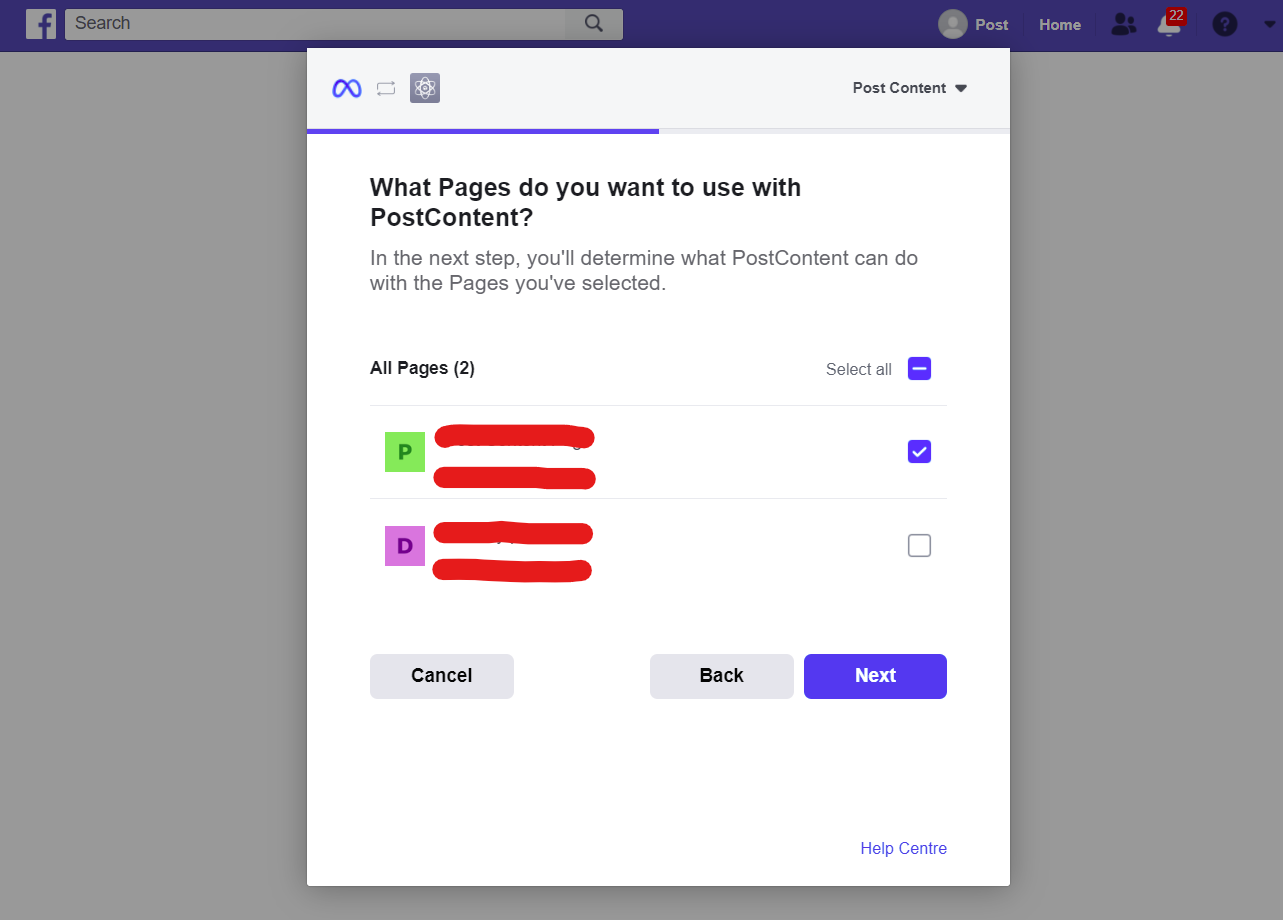
* After pressing “OK”, wait until you are no longer being redirected to other sites. The authentication code that is received in the URL after logging in is used in the POST request to the /Facebook/Access\_Token endpoint, which gives a long-lived access token. Please use this token and the Facebook page ID copied earlier in the body of the POST request to the /Facebook/Page\_Token endpoint to get a page token. This token along with the Page ID is used in the POST request to the /Facebook endpoint of the API.

1. Instagram

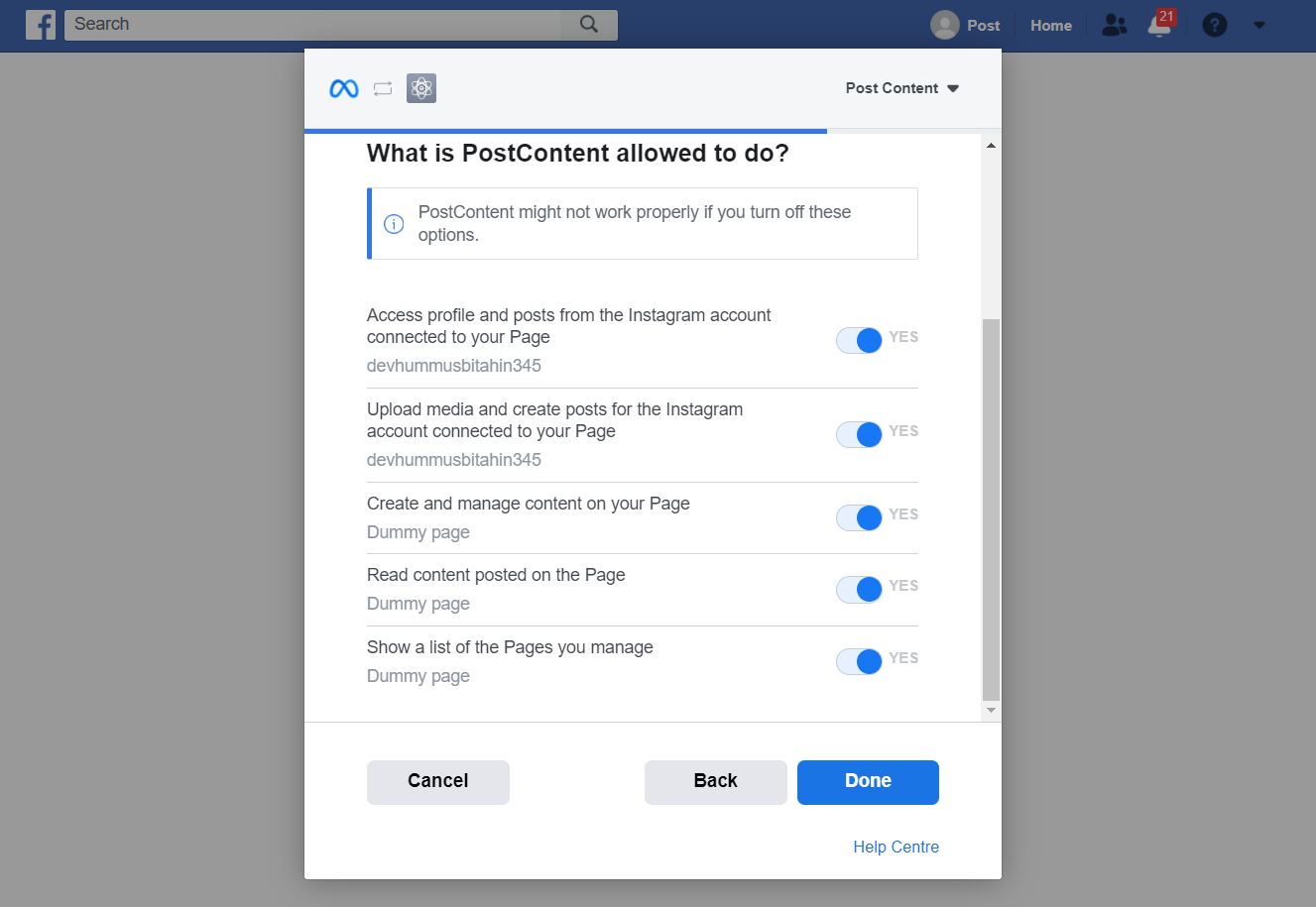
* To allow the API to post to your Instagram business account, you must send a POST request to the /Instagram/Login endpoint of the API with a redirect Uri in the body. This call should return a URL. Please visit this URL. From there, you will be asked to log in to Facebook. Note that your Instagram account should be a business account and connected to a Facebook page.



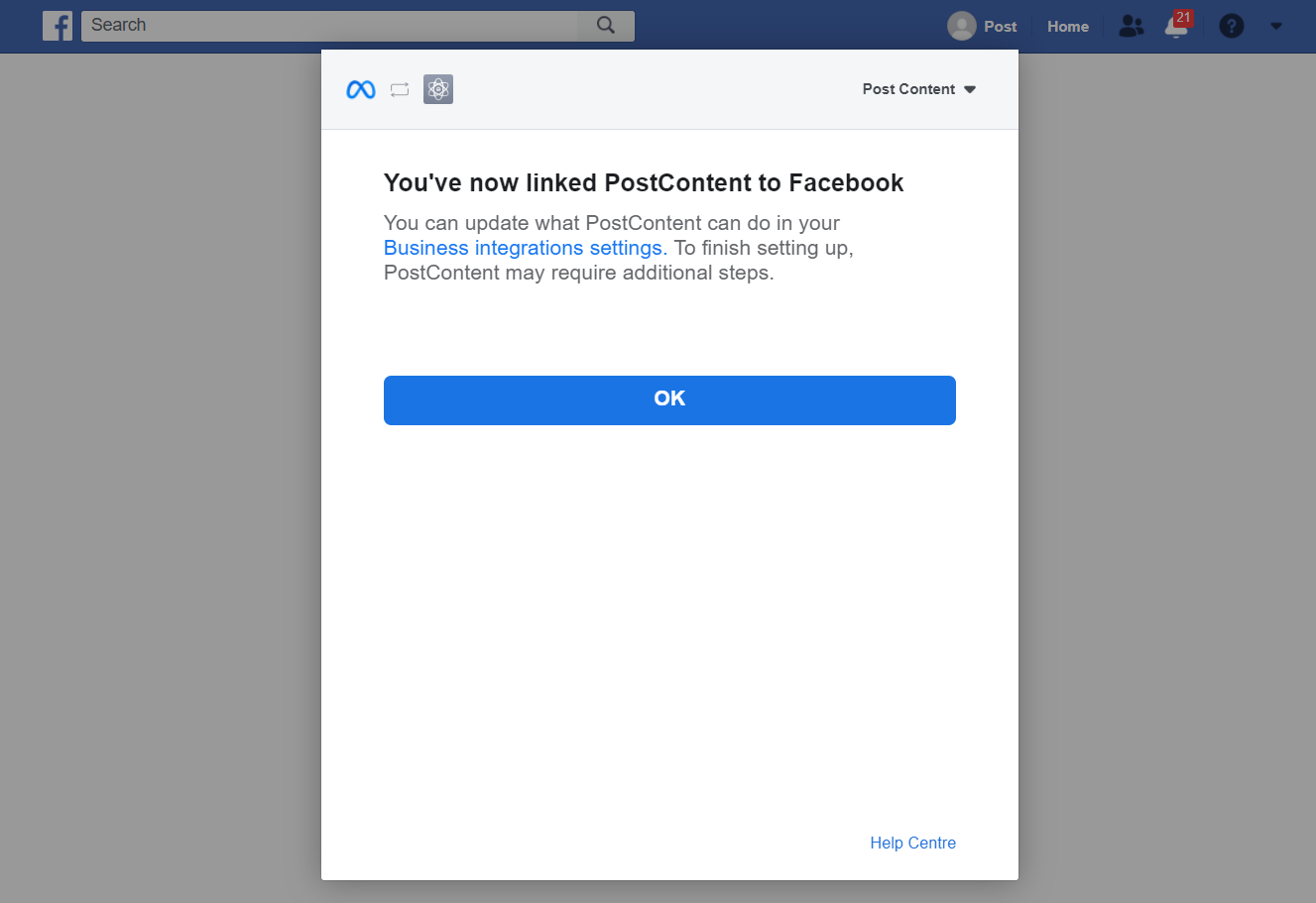
* Next, choose an Instagram Account that you want the API to post to. 
* Afterwards, choose the Facebook page connected to your Instagram account. Please copy the ID of the Facebook page you choose located below the Page name.



* Next, provide our app with the necessary permissions and press “Done”.



* After that, you should be able to see the following:

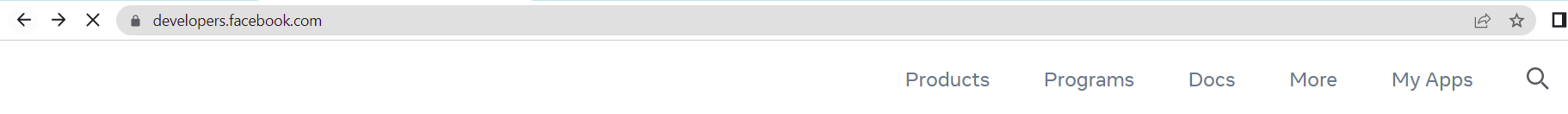


* Press “OK” and wait until you are no longer being redirected to other sites. The authentication code that is received in the URL after logging in is used in the POST request to the /Instagram/Access\_Token endpoint, which gives a long-lived access token. Please use this token and the Facebook page ID copied earlier in the body of the POST request to the /Instagram/Page\_Token endpoint to get a page token. The token along with the Facebook page ID is used in the POST request to the /Instagram endpoint of the API.

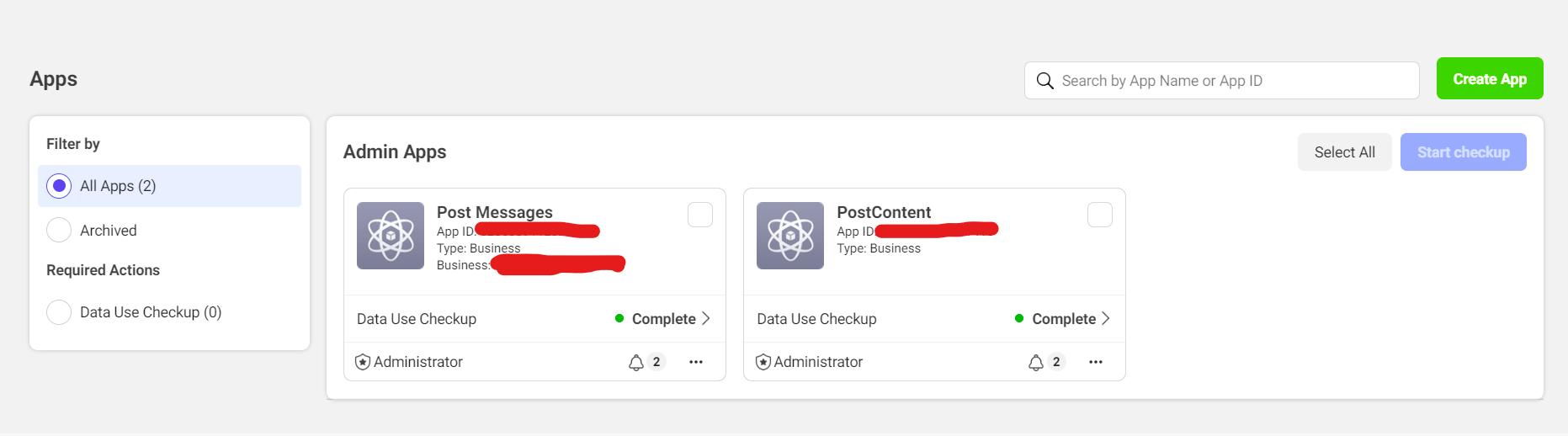
1. WhatsApp

**For a developer with access to the business app created on the Meta For Developers site** **and having the WhatsApp product added to it:**

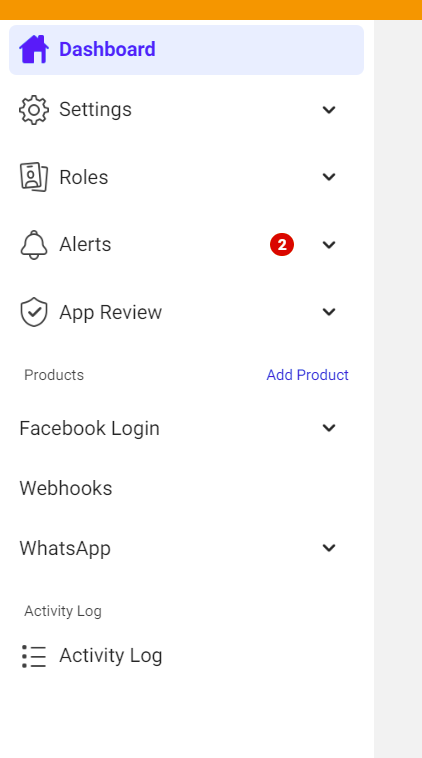
* Go the My Apps portion of the Meta For Developers site.



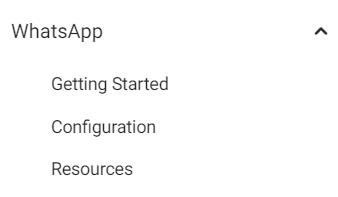
* Choose the business app created for WhatsApp.



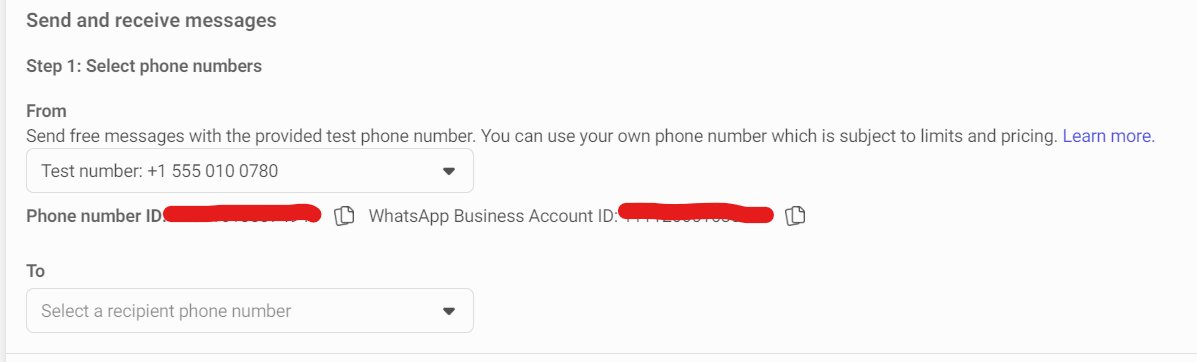
* Press on “WhatsApp” under Products on the left side of the screen.



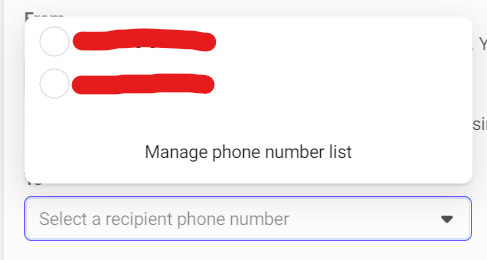
* Then, press on “Getting Started”.

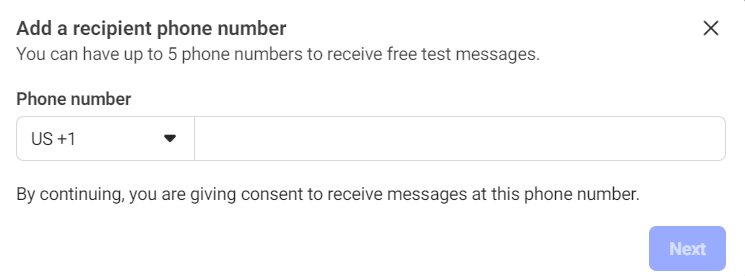


* In the “Send and receive messages” portion of the “Getting Started” page, the phone number ID of the test number provided by Meta is found. Please copy it as it is used in the POST request to the /WhatsApp endpoint of the API.



* To add a recipient phone number for testing, press on the “Select a recipient phone number” field, and then on “Manage phone number list”, where you will be prompted to add a number. Please follow the process on the website that follows that step to be able to add this number.

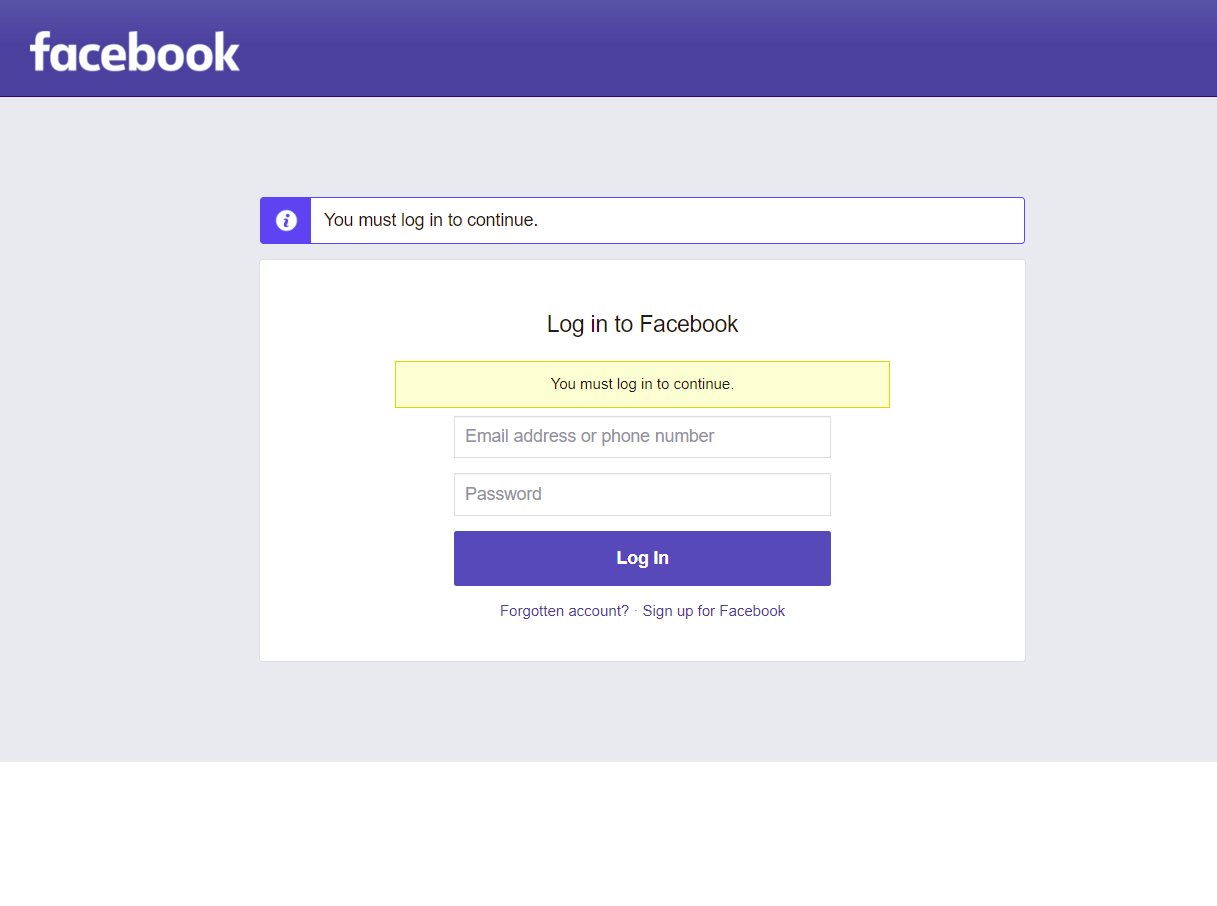




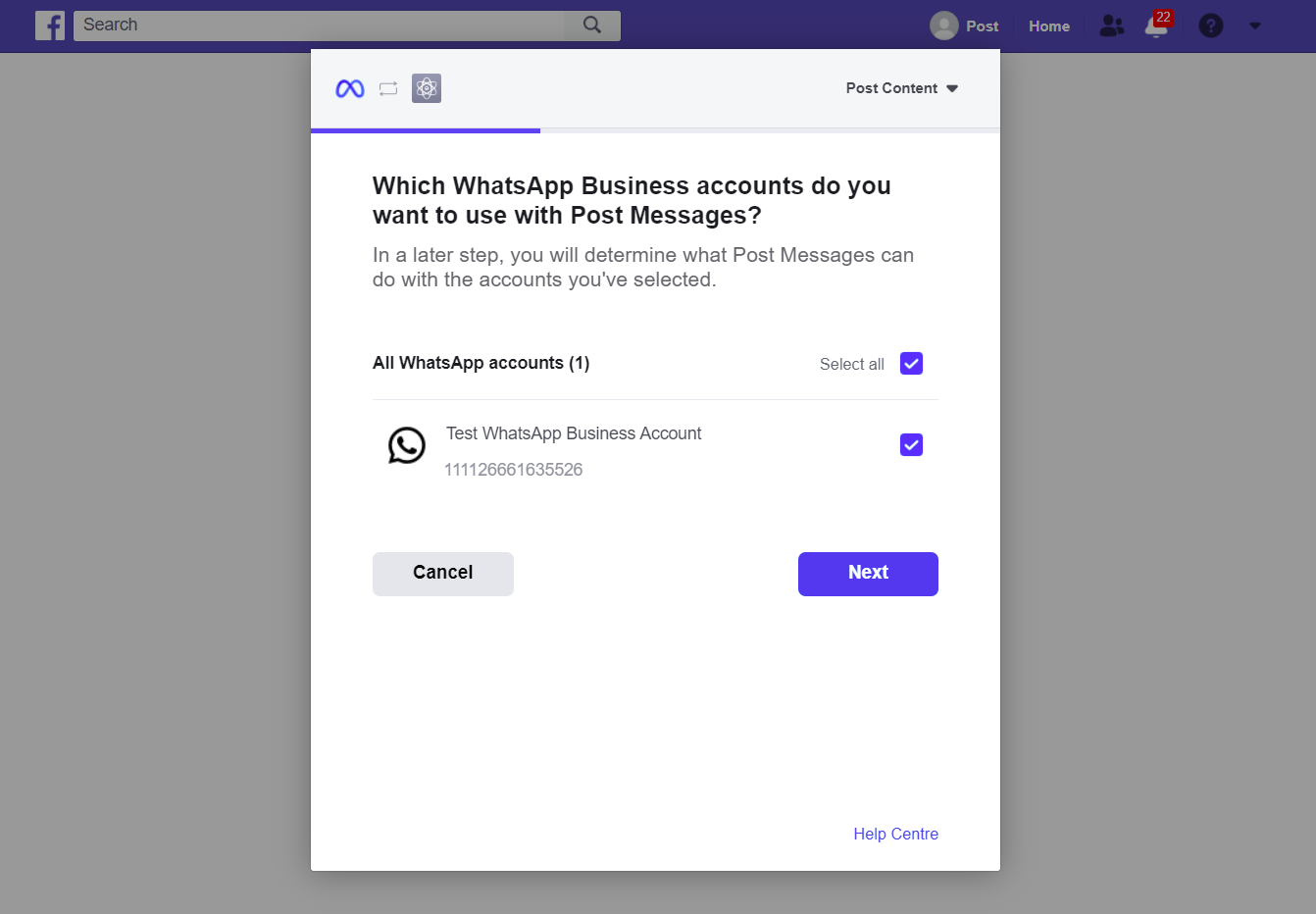
* Note that when using a usable recipient phone number in the POST request, to avoid issues, it is advised to type in the number with only numbers and without spaces. Ex: 96101234567.

**For users of the API**:

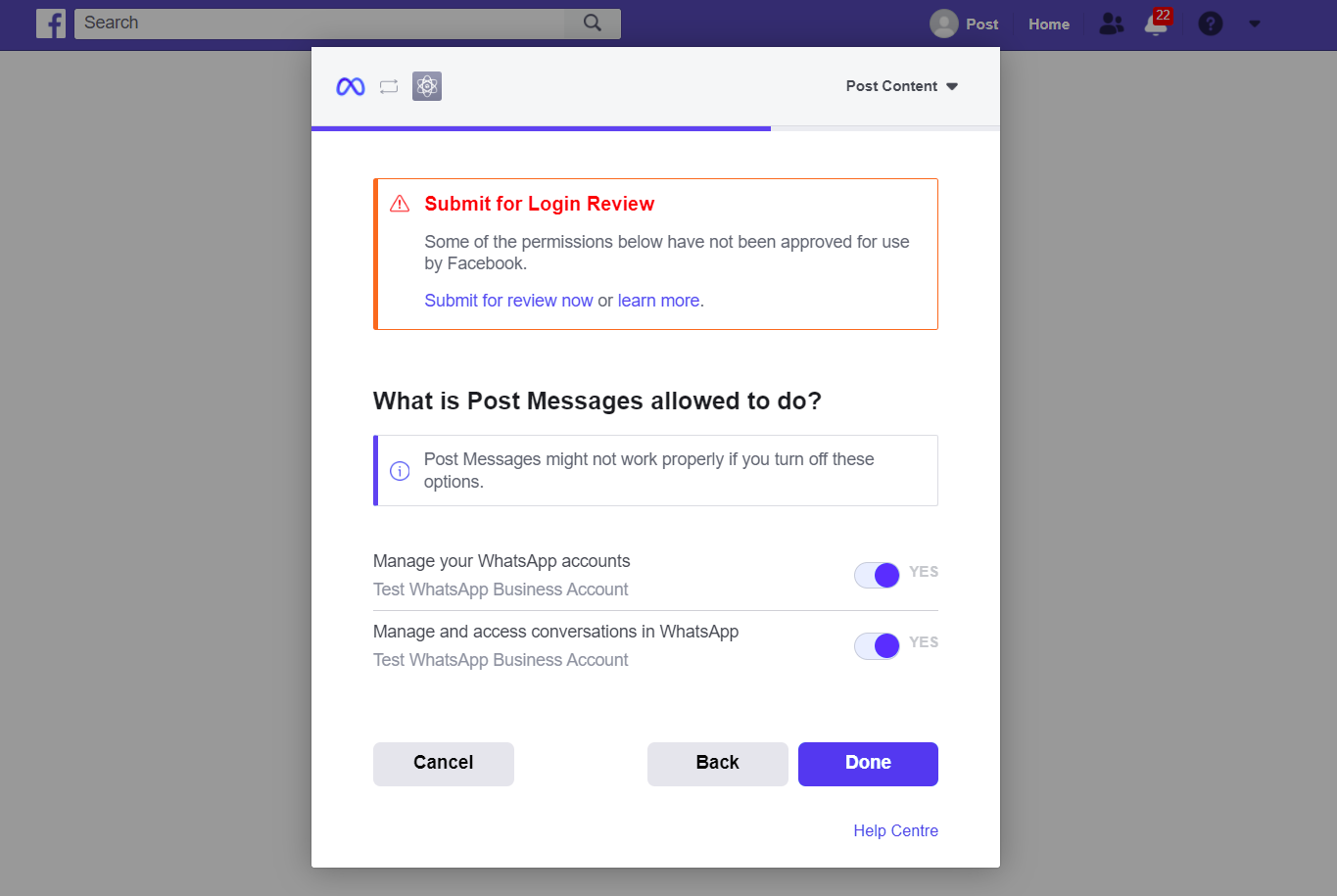
* To be able to use our API for WhatsApp, please send a POST request to the /WhatsApp/Login endpoint of the API which will return a URL. Please visit this URL to login into Facebook if you aren’t already and to authorize our app.
* Log in if prompted to do so (this is done when you aren’t already logged in).



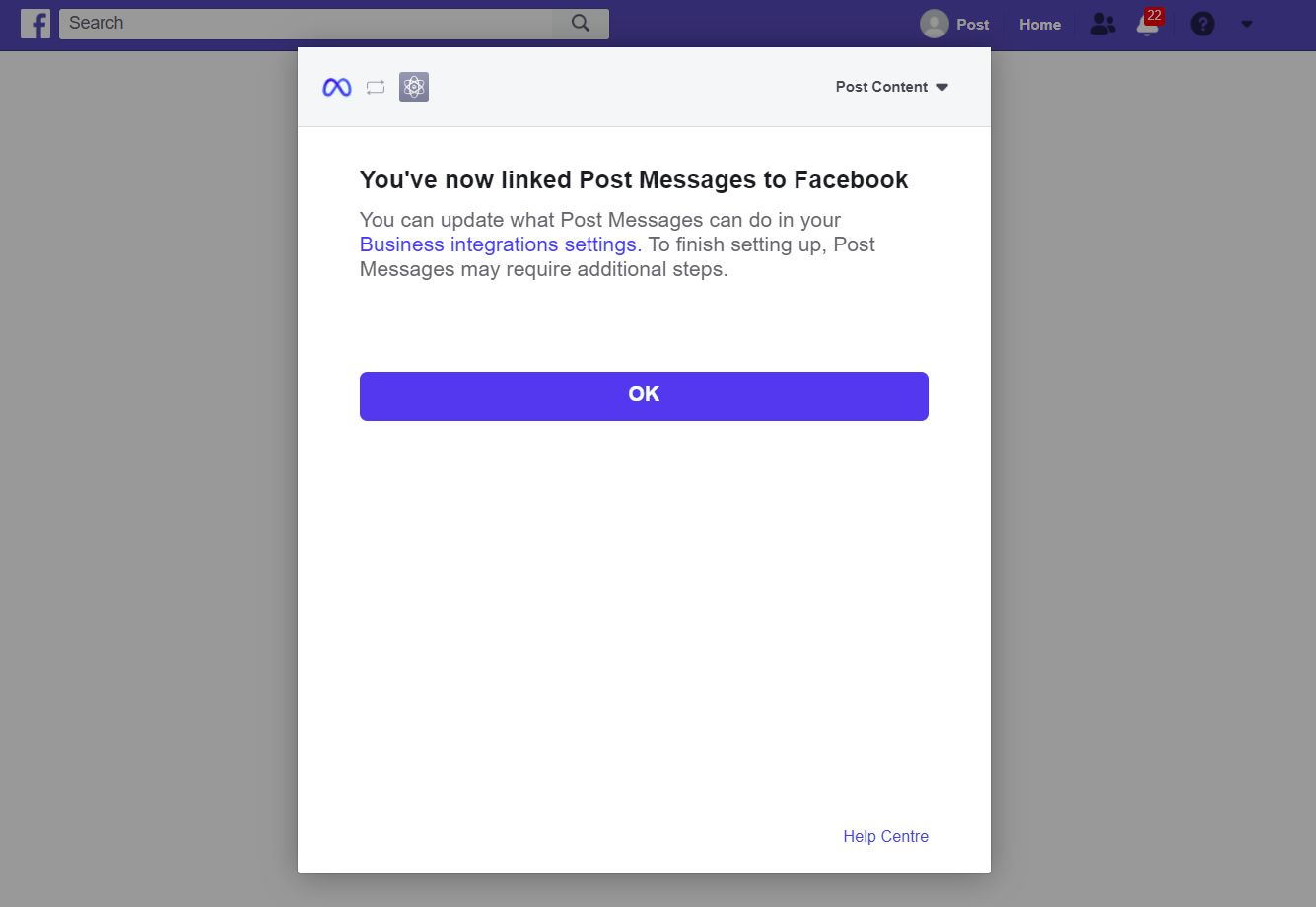
* Then, choose a WhatsApp Business Account to use with the API and press “Next”.



* Afterwards, please provide us with the listed permissions and press “Done”.



* On success, you should be able to see the following on your screen.

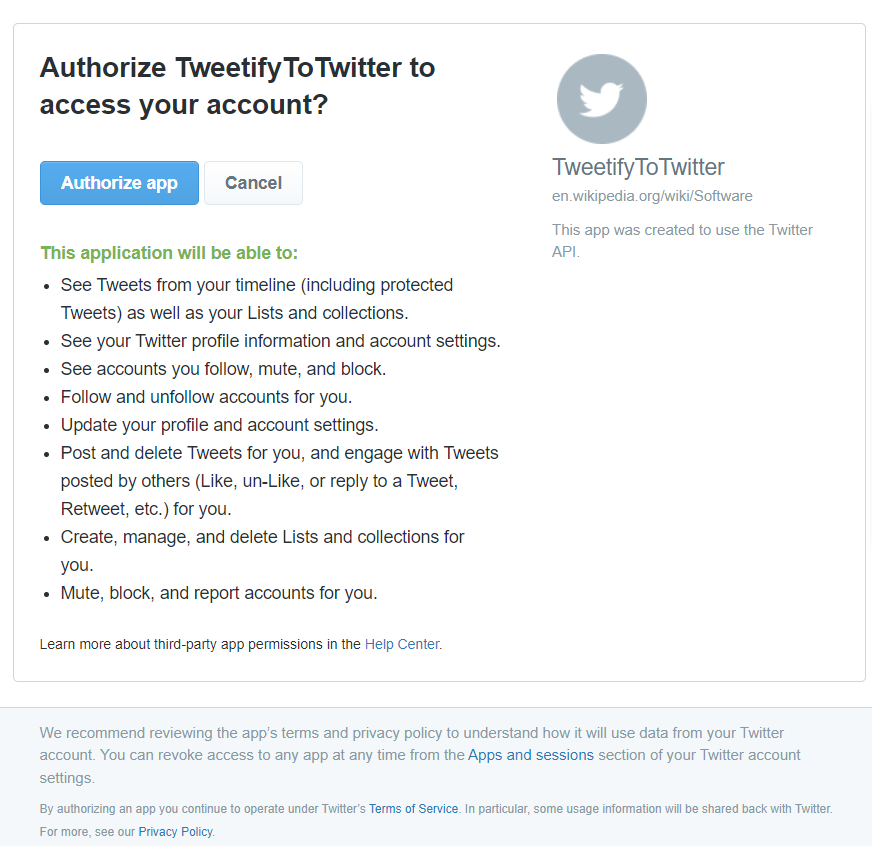


* Press “OK” and wait until you are no longer being redirected to other sites. The authentication code that is received in the URL after logging in is used in the POST request to the /WhatsApp/Token endpoint, which gives a long-lived access token. Please use this token and the phone number ID of the phone number you want to send messages from is used in the POST request to the /WhatsApp endpoint of the API.

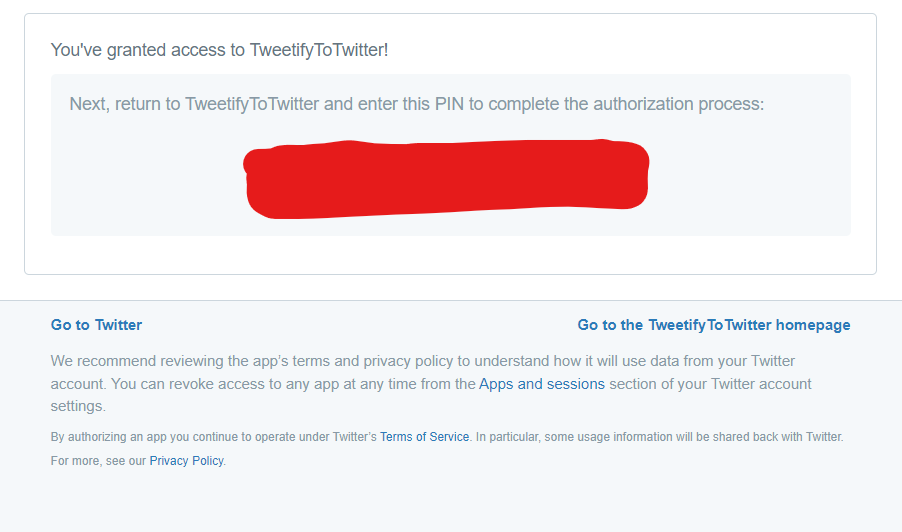
Side note: When using the API, you cannot send messages other than a template message to a recipient before they send a message to your number.

1. Twitter

* You must first send a GET request to the /Twitter/Login endpoint of the API. Log in to Twitter if you aren’t already and press “Authorize app” to be able to use the API.

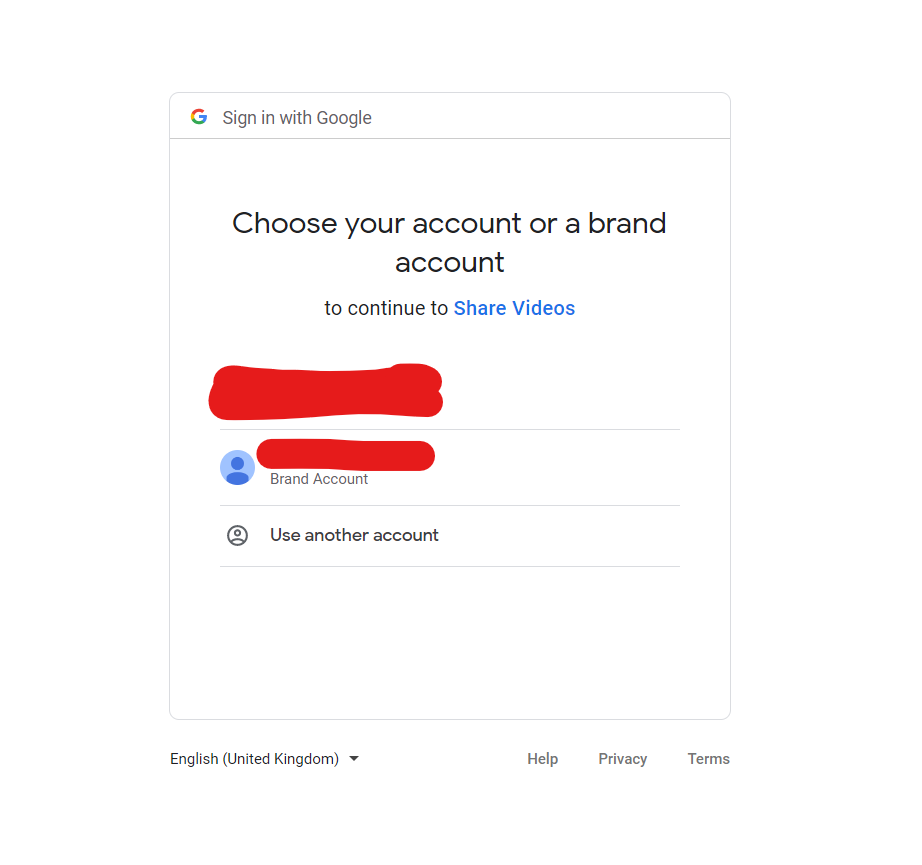


* Afterwards, copy the PIN you get and send a POST request with this PIN in its body to the /Twitter/Tokens endpoint of the API to get the access token and the access token secret. These credentials are used in POST request to the /Twitter endpoint of the API.

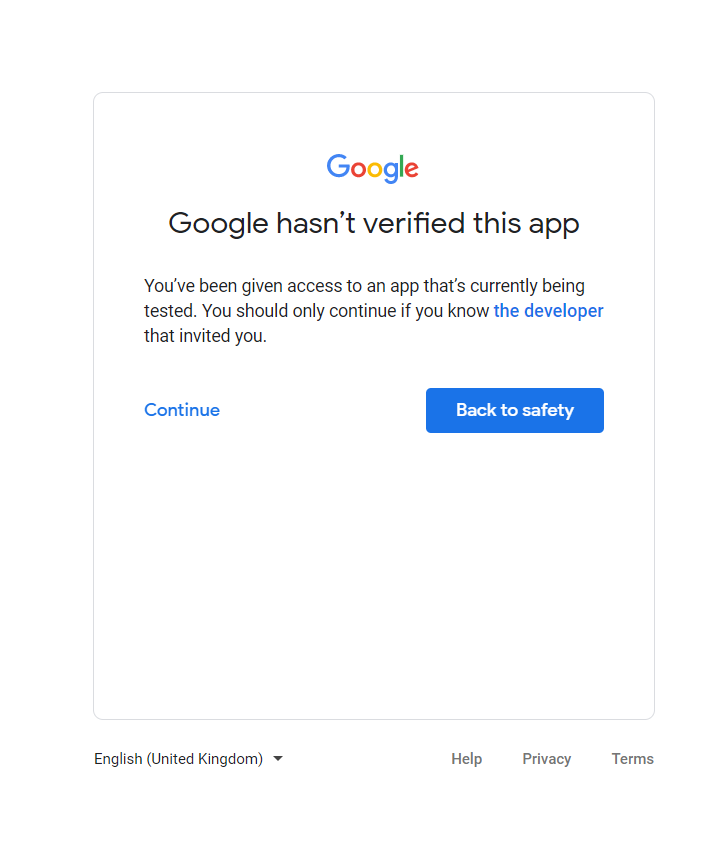


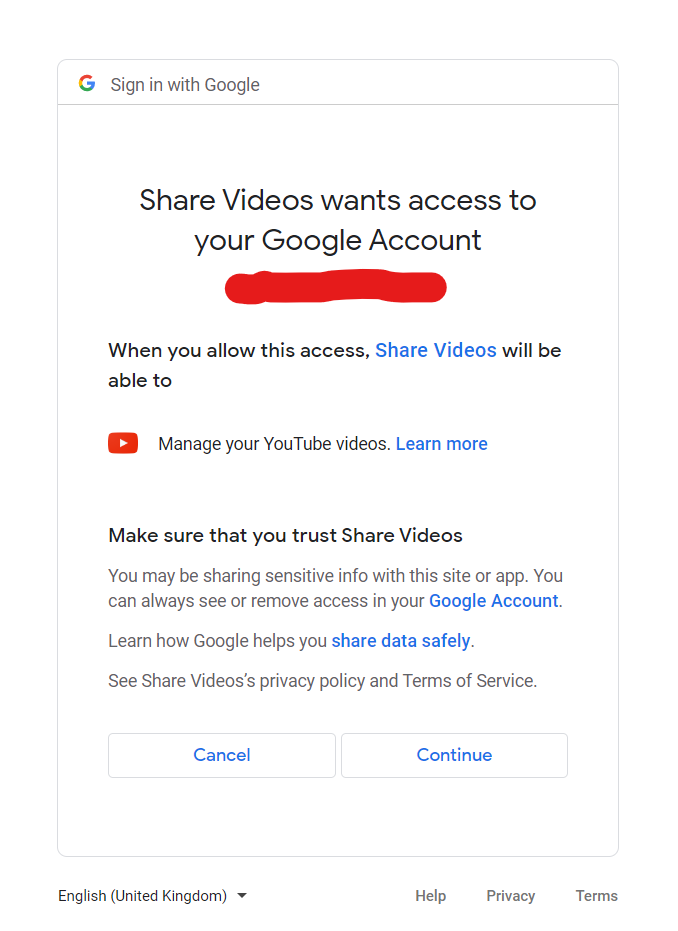
1. YouTube

* Send a POST request to the YouTube endpoint of the API with the file path of the video, the title and the description of the video. Choose an account or a brand account to proceed with.

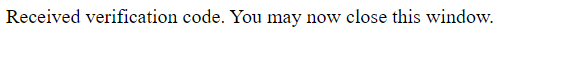


* Press “Continue” afterwards:

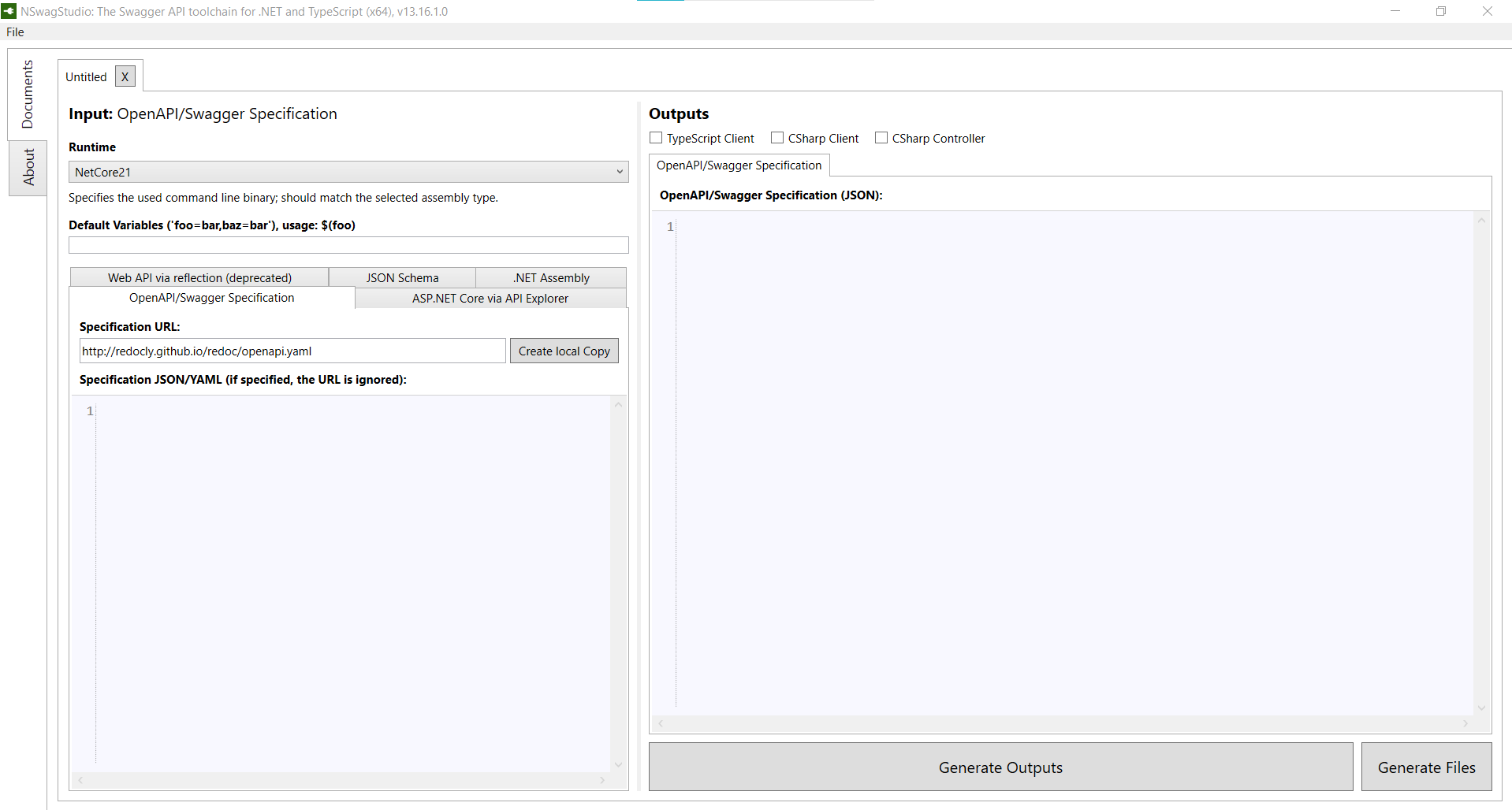
- Next, give the app the needed permission by pressing “Continue”

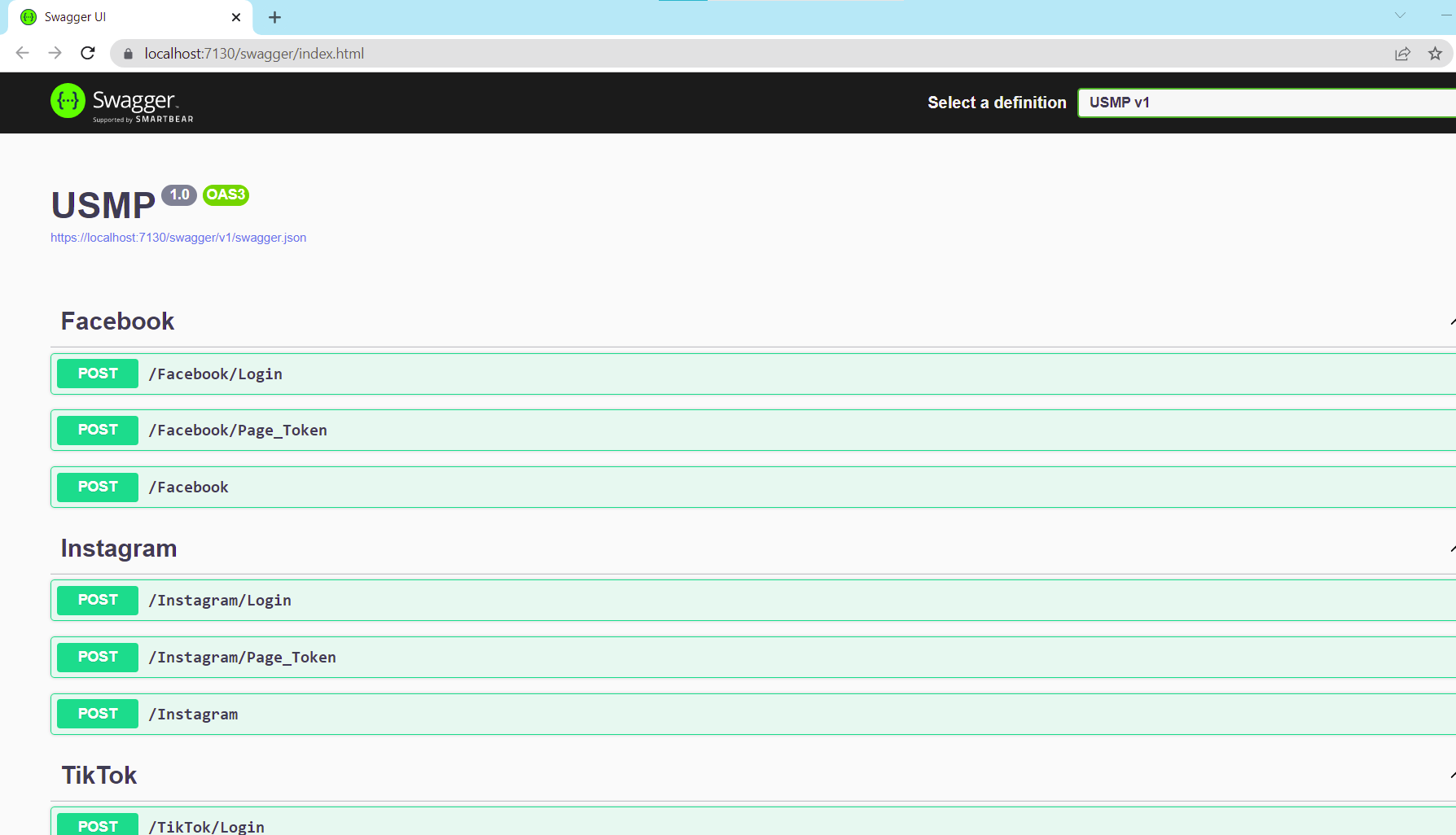


* Afterwards, if you receive the following message in your browser, you may close the window and expect the video to be uploading/uploaded.

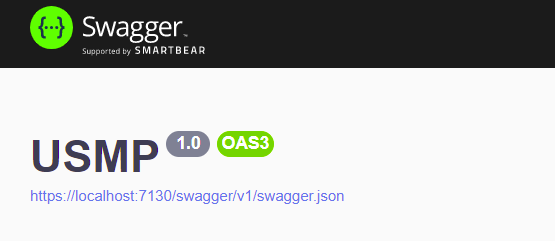


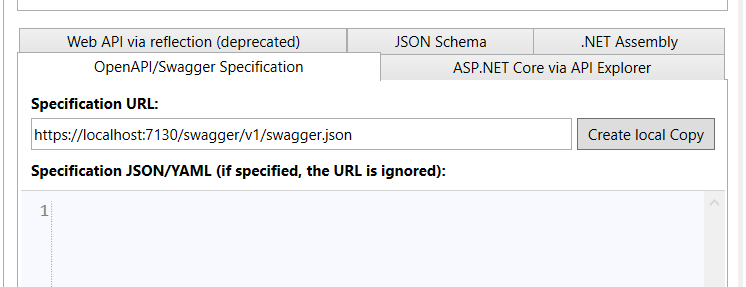
* Generating client code from API using NSwagStudio to use in Angular
* After creating the Web API with ASP.NET, NSwagStudio was used to generate a client code from the API.
* Open NSwagStudio, choose the appropriate runtime (NetCore60 in our case) and select the OpenAPI/Swagger specification tab if it isn’t selected. Also, run the Swagger Web API in parallel.



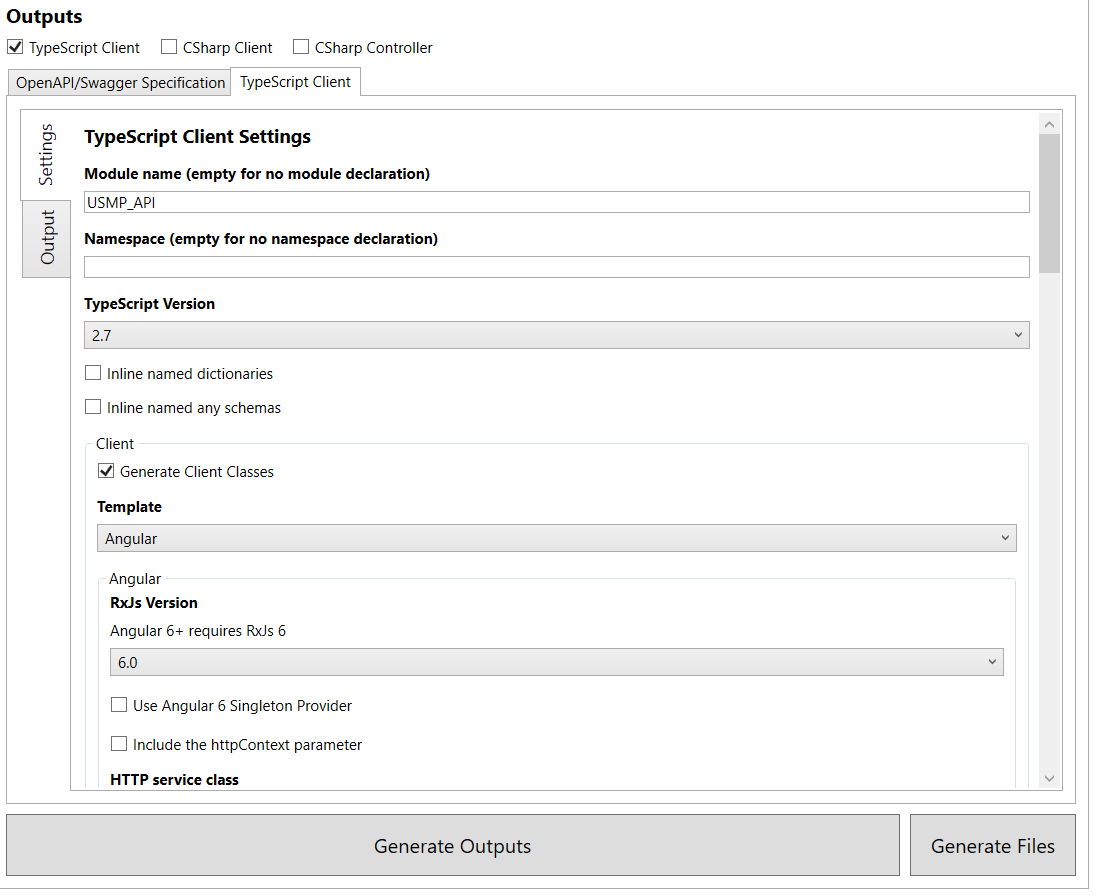


* Copy the link to the swagger.json found in the Swagger UI and paste it in the “Specification URL” field inside NSwagStudio.

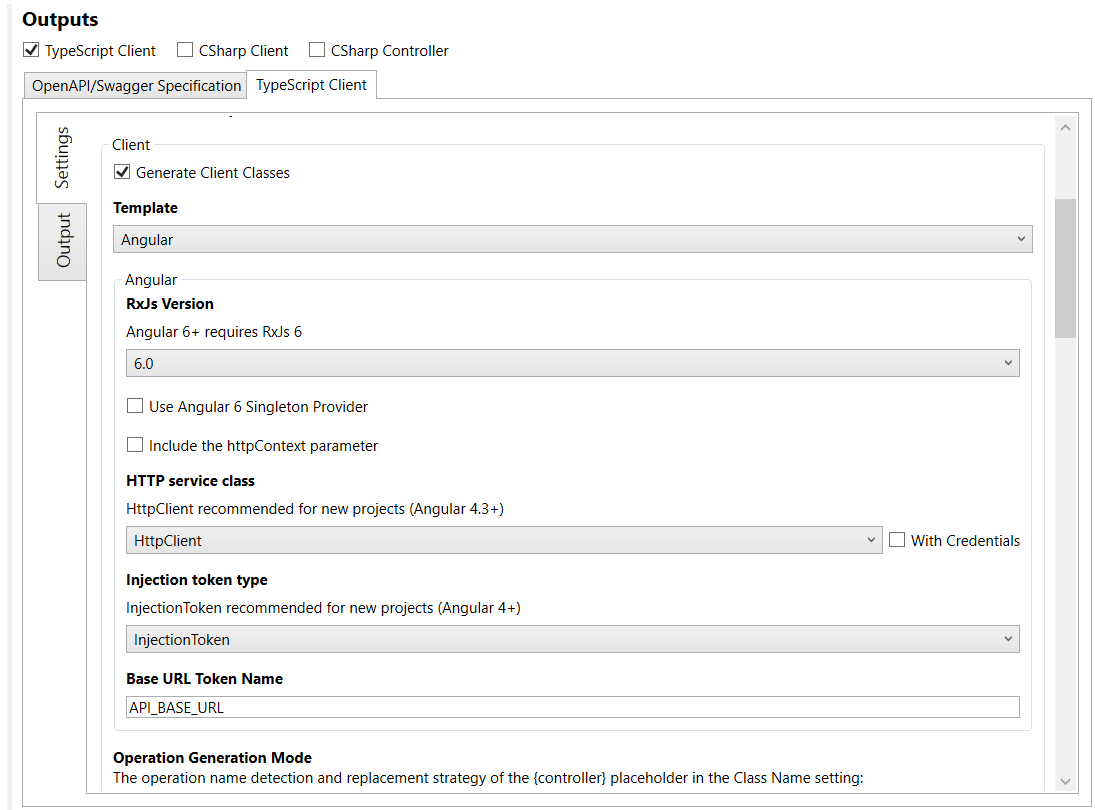




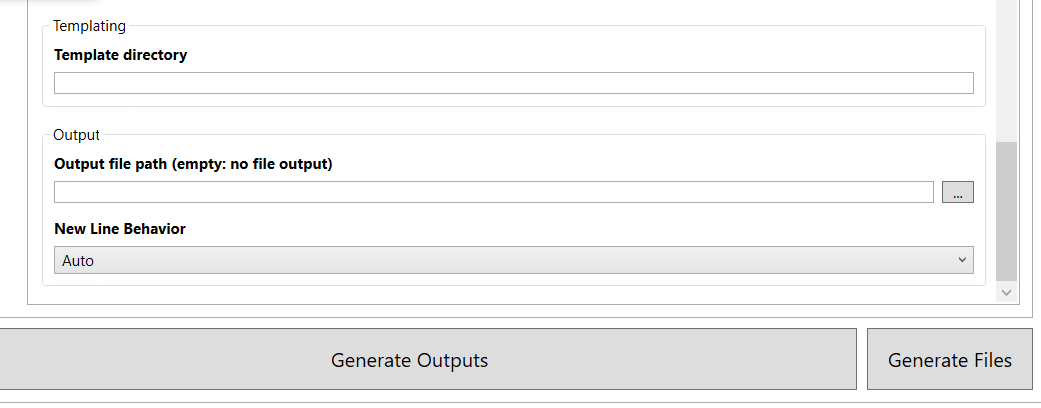
* Under outputs select TypeScript Client and go the TypeScript Client tab. Type in a name for the module of the client under “Module name” in the settings, make sure “Generate Client Classes” is selected under the Client section and choose the template to be Angular.

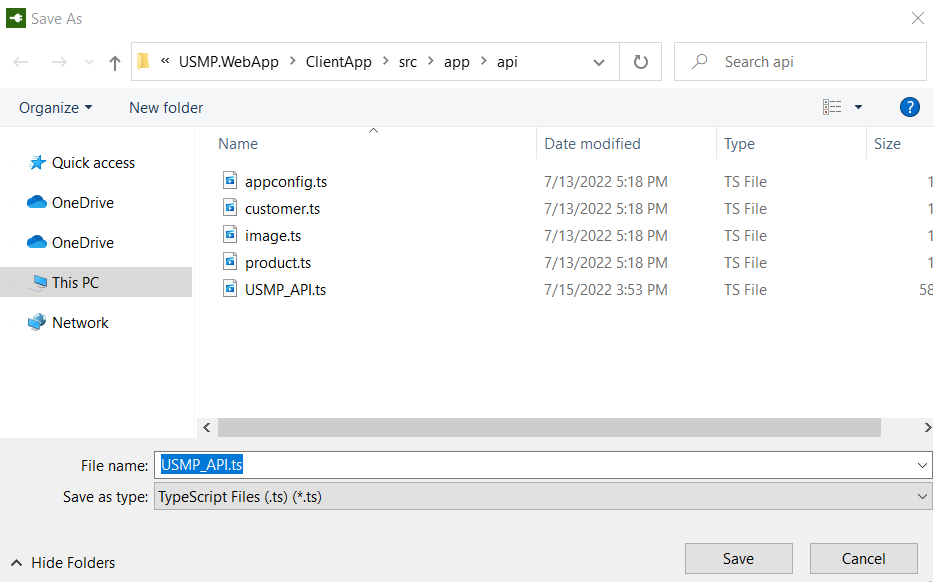


* Afterwards, under “Injection Token Type” select InjectionToken in case you are using an Angular 4+ version.



* Scroll down to find the Output section. You can press on the three dots next to the “Output file path” input field to open the File Explorer if you are on Windows, navigate to where you want your file to be placed in your Angular project and give the TypeScript file you want to generate a name. After saving the file path, press on “Generate Files” in NSwagStudio.





* In the generated TypeScript file, you can find a class/classes with a method/methods that you can use from your Angular app to interact with the API.
* Generating a MySQL database using the code-first strategy in EF Core
* Install the MySql.Data.EntityFrameworkCore package in your project.
* Refer to step 2 in the link <https://dev.mysql.com/doc/connector-net/en/connector-net-entityframework-core-example.html> to see an example of a model (including classes and properties) created in code as well as the database context class which must include your own connection string in the optionsBuilder.UseMySQL method. Use these examples to learn how to write your own code for the database.
* In the link, in part c of step 2, an example C# code to be placed in the existing Program.cs file is provided. It shows an example of using the context and inserting data to the database and then fetching data from it. Use this example to learn about inserting data to and fetching data from the database you create.
* To restore dependencies and run the application from the command line, you can use “dotnet restore” followed by “dotnet run”.