



How to Add Push Notifications to the OTT Toolkit App? (iOS/Android)

×



To activate the push notification settings in your OTT Toolkit, you will need to create three separate Firebase files for



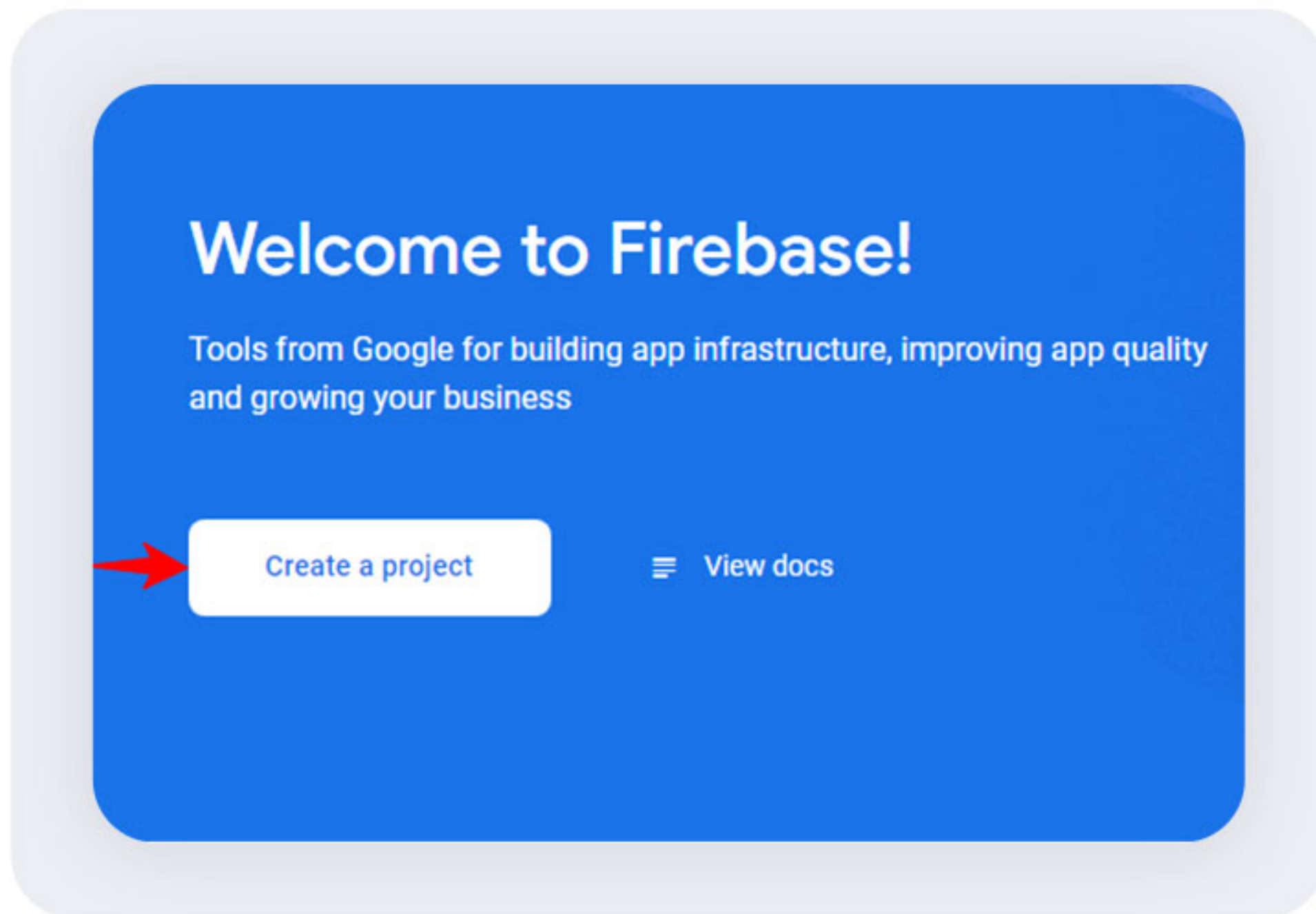
- ✔ The OTT server - private key .json file.
- ✔ Android App - Google-services.json file & package name.
- ✔ iOS App - GoogleService-Info.plist & Bundle ID.



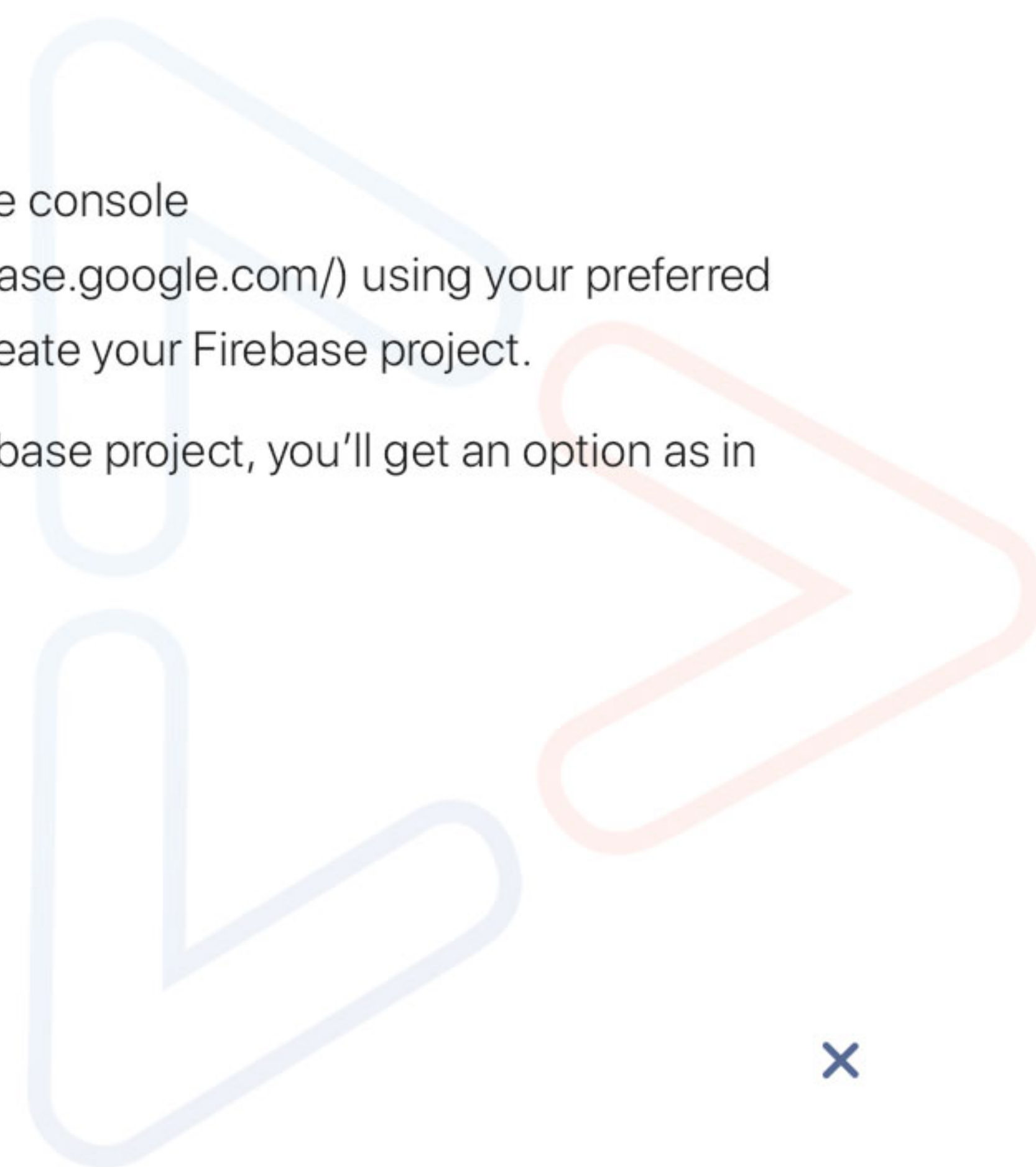


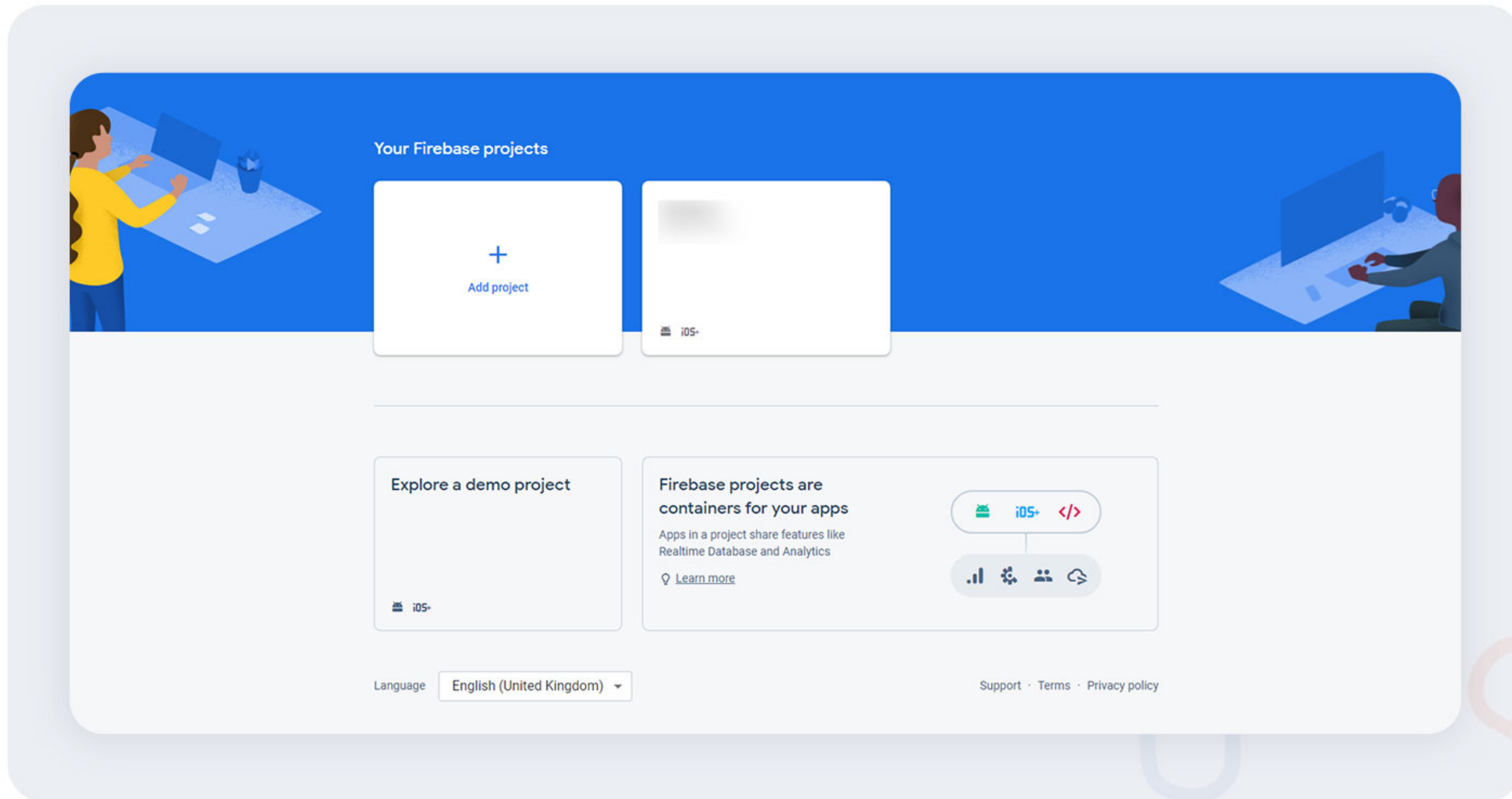
To add **Firebase push notifications** to an iOS/Android app, create a **Firebase project**.
Configure the following steps to create a Firebase project.

Create a project



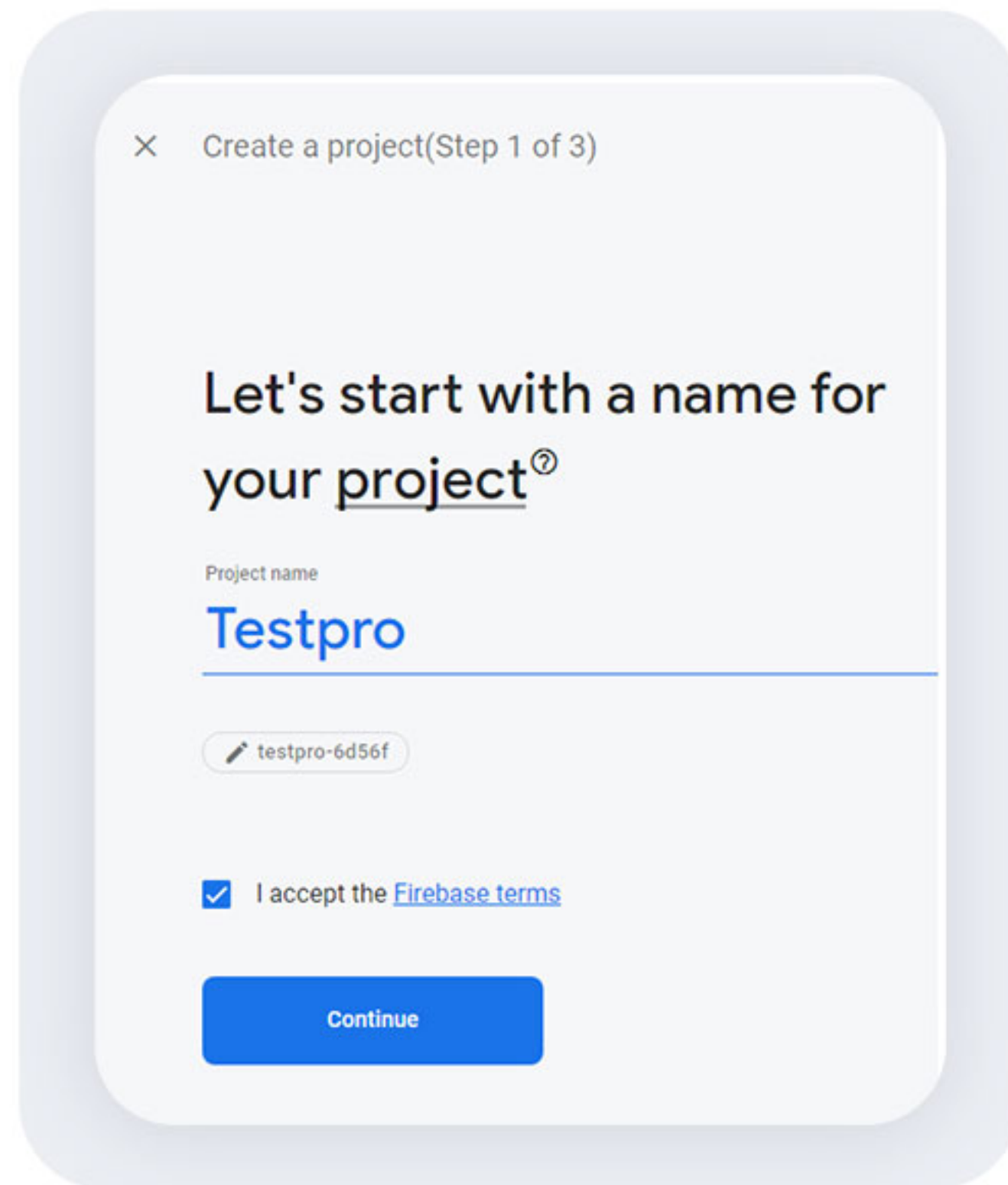
- ✓ Sign in to the Firebase console (<https://console.firebase.google.com/>) using your preferred Google account to create your Firebase project.
- ✓ If this is your first Firebase project, you'll get an option as in the image below:





Or, if you have already created a project, you can click on the "Add Project" to continue the process.

Process



× Create a project(Step 1 of 3)

Let's start with a name for your project

Project name

Testpro

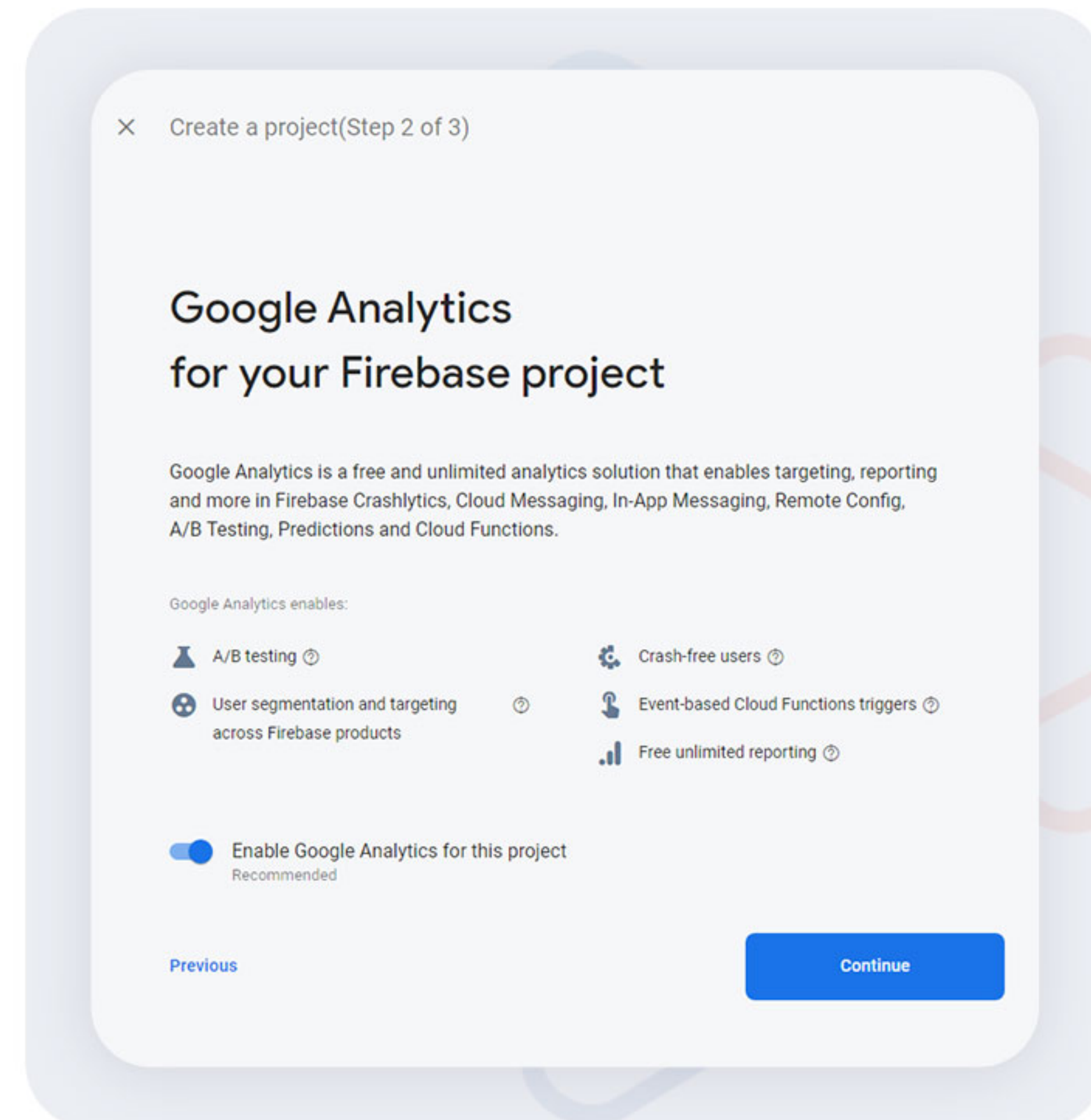
testpro-6d56f

I accept the [Firebase terms](#)

Continue

Now, enter or select a name for your project and click "Continue" after agreeing to the Firebase terms & conditions.

Enabling Google Analytics is up to you.



× Create a project(Step 2 of 3)

Google Analytics for your Firebase project

Google Analytics is a free and unlimited analytics solution that enables targeting, reporting and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, Predictions and Cloud Functions.

Google Analytics enables:

- A/B testing
- User segmentation and targeting across Firebase products
- Crash-free users
- Event-based Cloud Functions triggers
- Free unlimited reporting

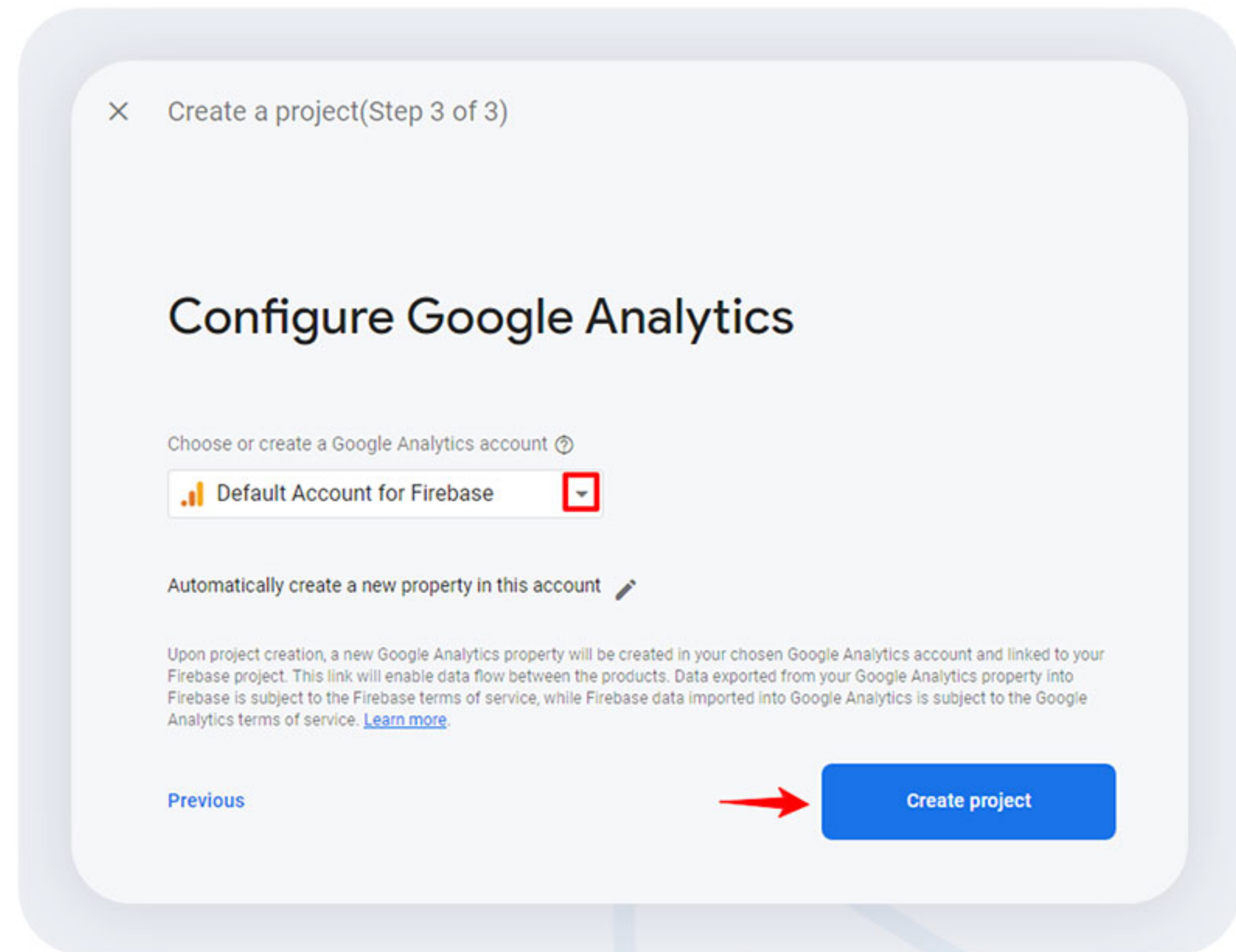
Enable Google Analytics for this project
Recommended

Previous Continue



Process

You can either create your project with a new account or an existing account.



× Create a project(Step 3 of 3)

Configure Google Analytics

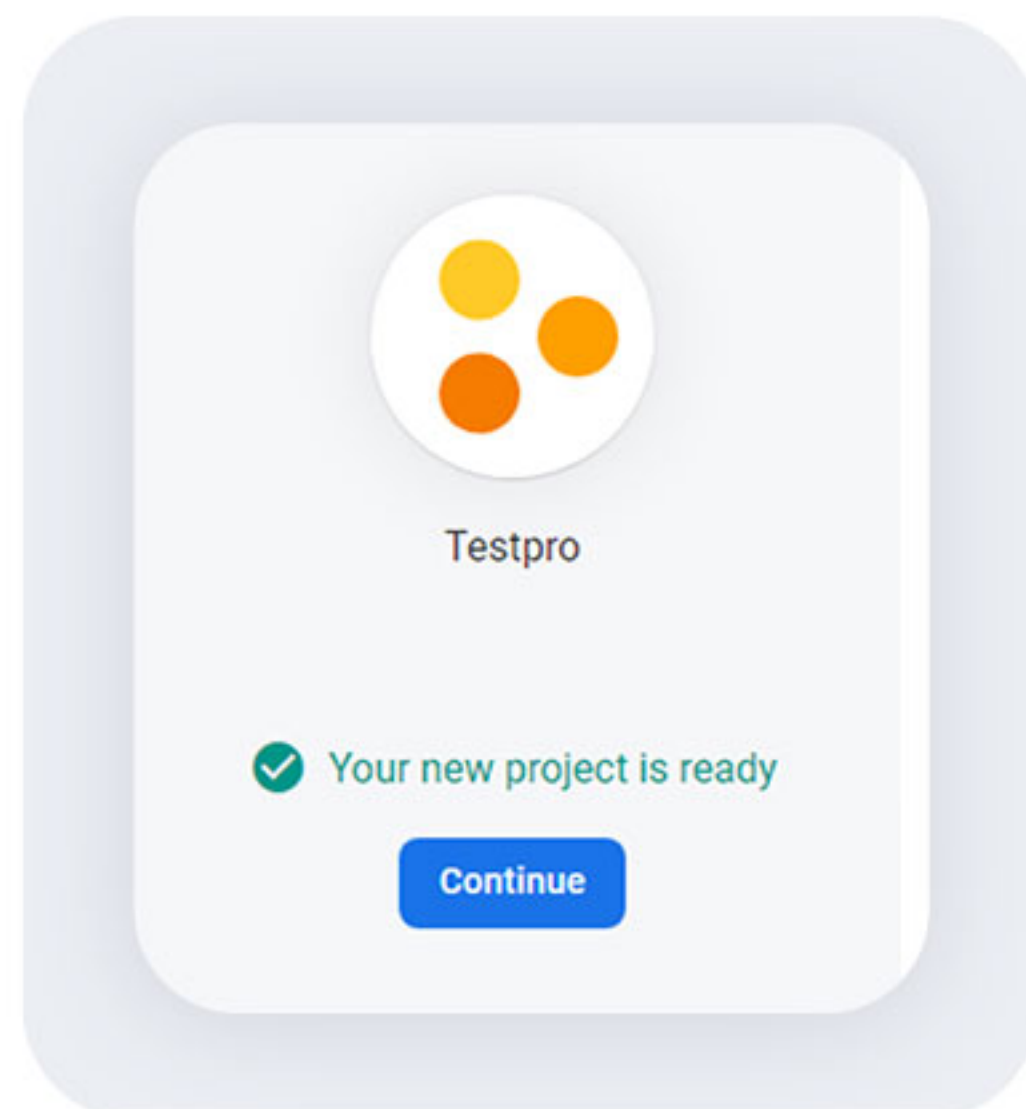
Choose or create a Google Analytics account ⓘ

📊 Default Account for Firebase ▾

Automatically create a new property in this account ✎

Upon project creation, a new Google Analytics property will be created in your chosen Google Analytics account and linked to your Firebase project. This link will enable data flow between the products. Data exported from your Google Analytics property into Firebase is subject to the Firebase terms of service, while Firebase data imported into Google Analytics is subject to the Google Analytics terms of service. [Learn more](#).

[Previous](#) ➔ [Create project](#)

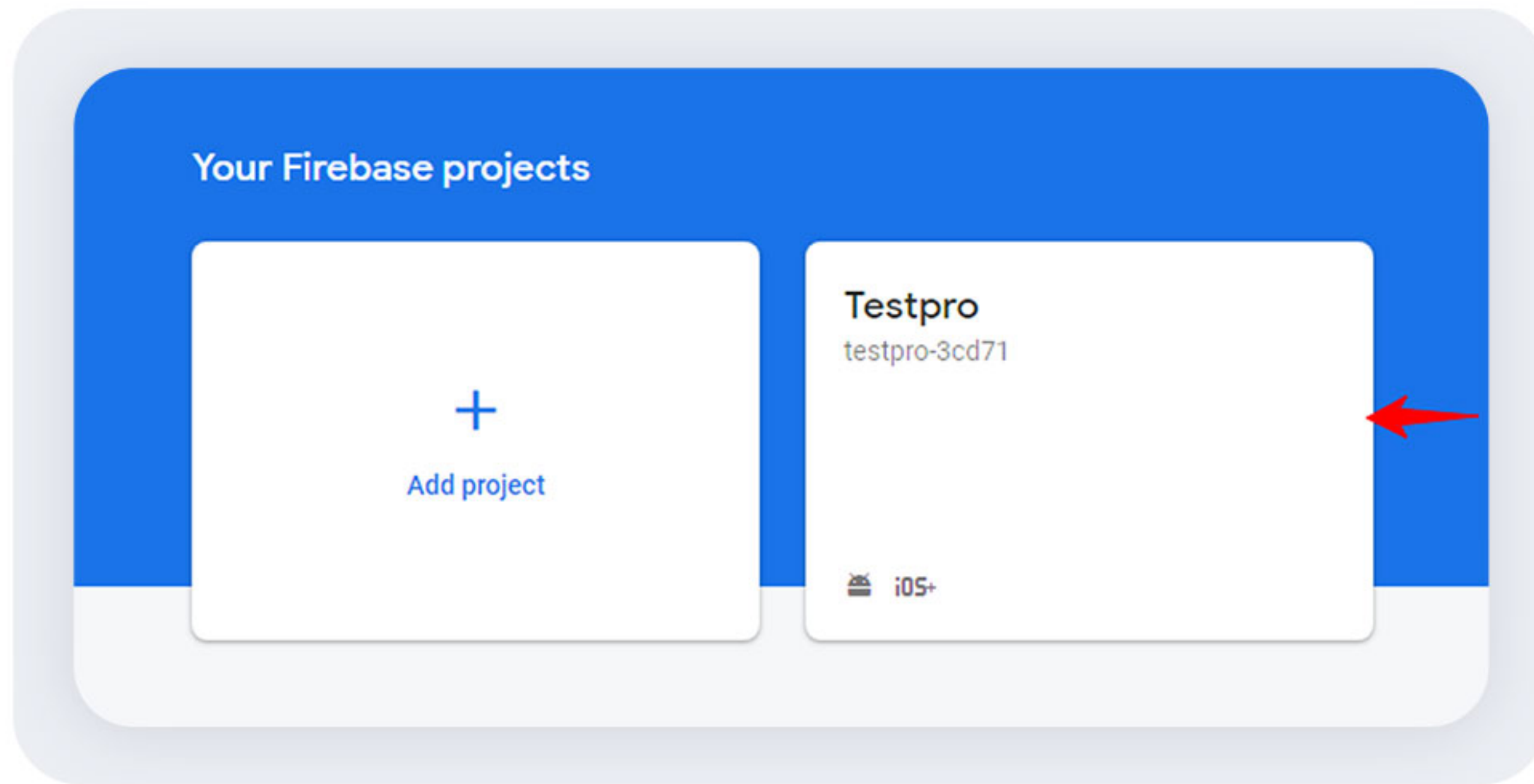


Then finish creating the project by clicking the "Continue" button.

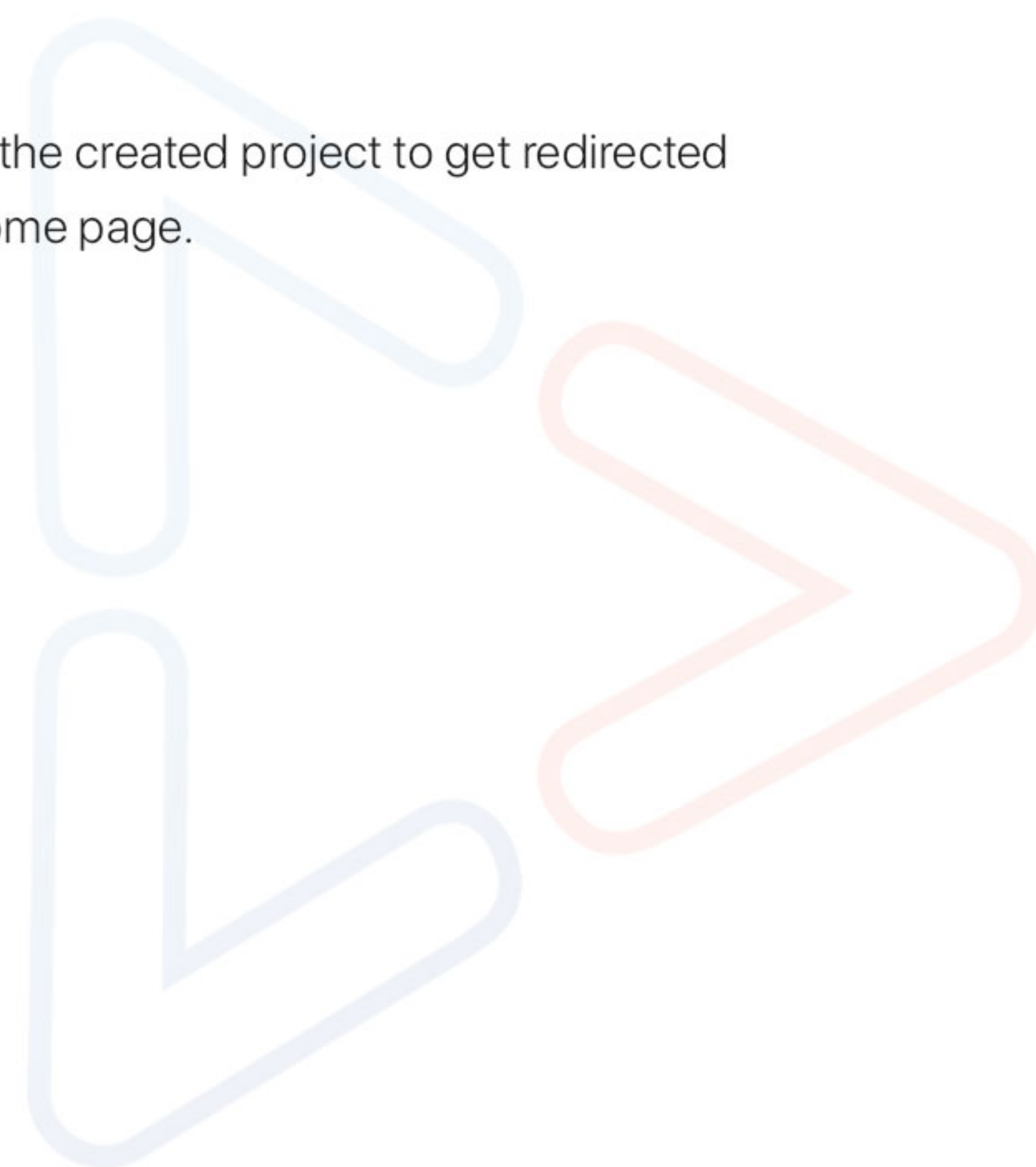


Your Firebase project is ready.

You'll get redirected to the project overview page. Go back to the project home page.

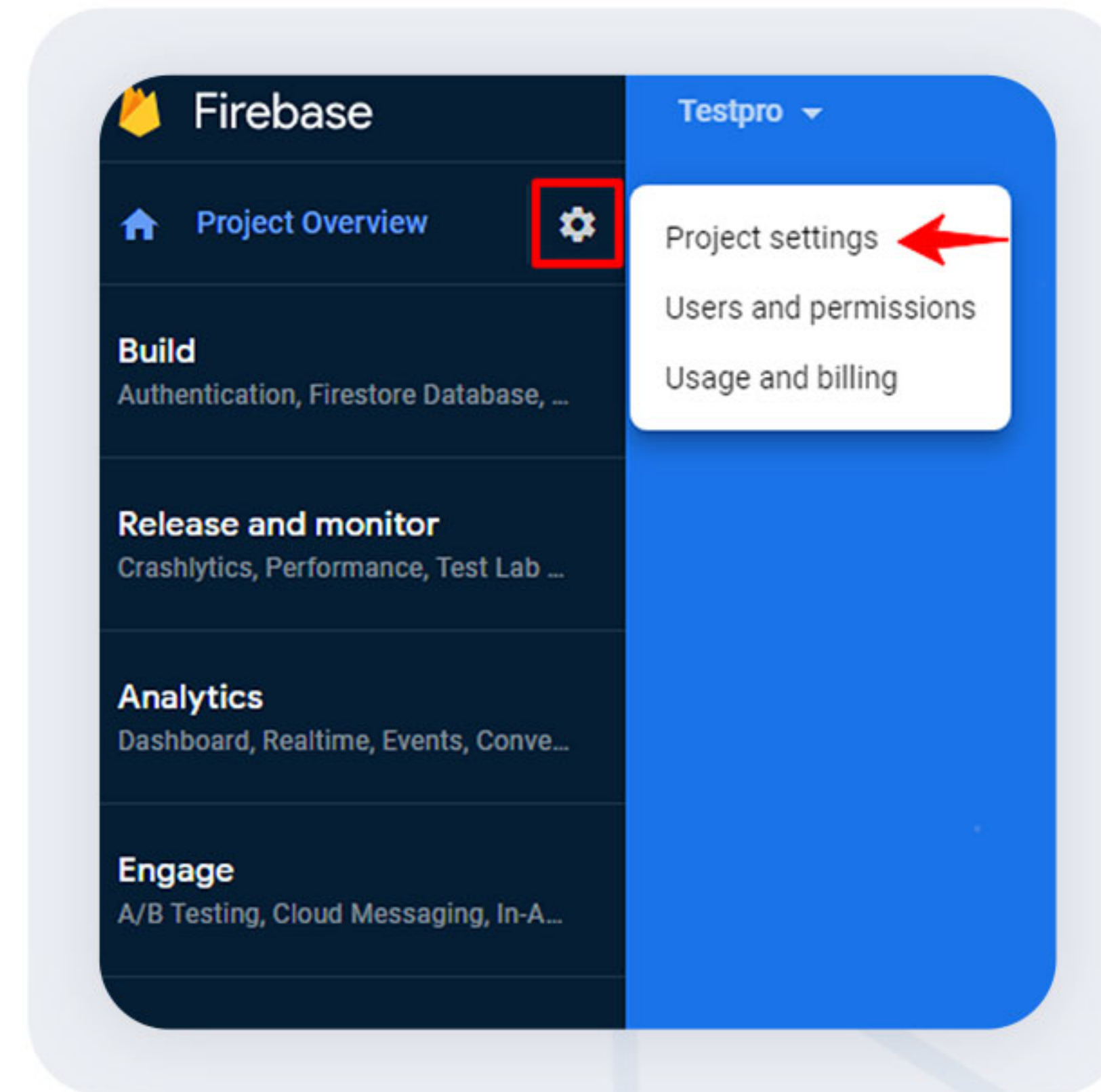


Click on the created project to get redirected to the home page.



Process

Click on the gear icon and go into the "Project settings."



Project settings

General Cloud Messaging Integrations Service accounts Data privacy Users and permissions App check

Now go into the "Service accounts" section.

Click on the "Generate new private key" button.

The screenshot shows the Firebase Admin SDK console interface. On the left, there is a sidebar with navigation options: 'Legacy credentials', 'Database secrets', and 'All service accounts' with a sub-option '4 service accounts'. The main content area is titled 'Firebase Admin SDK' and includes a link to 'Manage service account permissions'. Below this, there is a section for 'Admin SDK configuration snippet' with radio buttons for 'Node.js', 'Java', 'Python', and 'Go'. A code block contains the following JavaScript code:

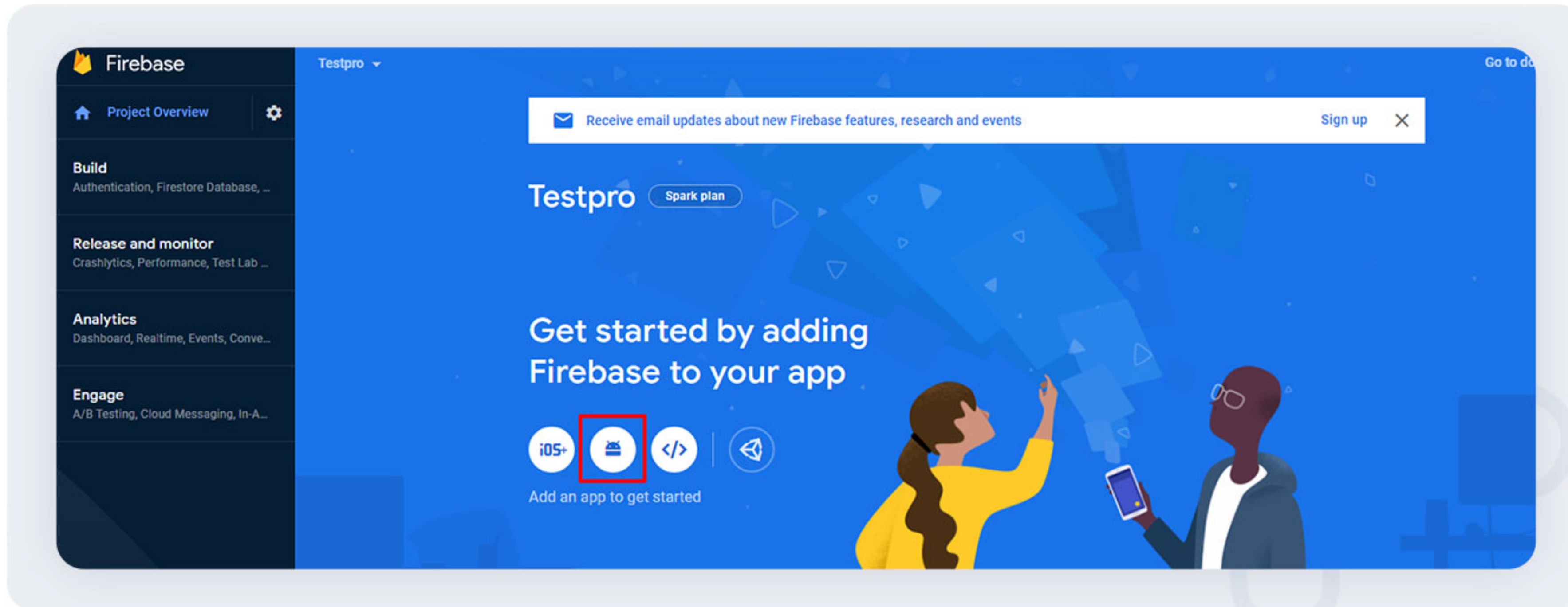
```
var admin = require("firebase-admin");  
var serviceAccount = require("path/to/serviceAccountKey.json");  
  
admin.initializeApp({  
  credential: admin.credential.cert(serviceAccount)  
});
```

At the bottom of the main content area, a blue button labeled 'Generate new private key' is highlighted with a red rectangular box.

A JSON file will be created. Save this file.

ANDROID

To enable the push notifications for the Android app,



On the project home page, click on the Android button.

Process

- ✓ Enter your package name in the Android package name field (other fields are optional)
- ✓ Then click the "Register app" button.

× Add Firebase to your Android app

1 Register app

Android package name ⓘ

App nickname (optional) ⓘ

Debug signing certificate SHA-1 (optional) ⓘ

Required for Dynamic Links, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.

Register app

2 Download config file

× Add Firebase to your Android app

✓ Register app
Android package name: com.OTTToolkit.Testpro, app nickname: Testpro

2 Download config file Instructions for Android Studio below | [Unity C++](#)

Download google-services.json

Switch to the Project view in Android Studio to see your project root directory.

Move the google-services.json file that you just downloaded into your Android app module root directory.

google-services.json

Next

Save the downloaded file along with the package name.

Then download the "Google-services.json" file.



Click the next button for the next three steps, then click the "Continue to the console" button.

× Add Firebase to your Android app

- ✓ Register app
Android package name: com.OTTToolkit.Testpro, app nickname: Testpro
- ✓ Download config file
- ✓ Add Firebase SDK
- 4 Next steps

You're ready!

Make sure that you take a look at the [documentation](#) to learn how to get started with each Firebase product that you want to use in your app.

You can also explore [sample Firebase apps](#).

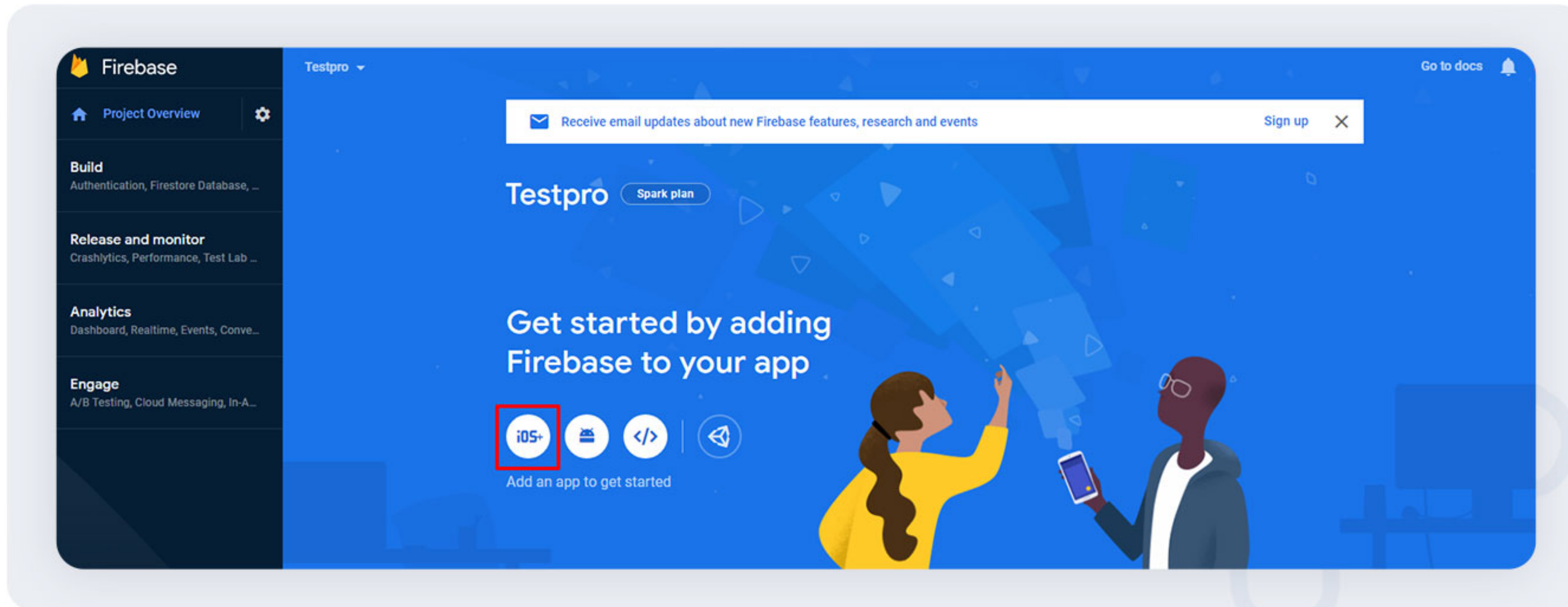
Or, continue to the console to explore Firebase.

Previous **Continue to the console**



iOS

To enable the push notifications for the iOS app,



On the project home page, click on the iOS button.

Process

- ✓ Enter OTT Toolkit's bundle ID in the iOS bundle ID field (other fields are optional).
- ✓ Then click the "Register app" button.

× Add Firebase to your Apple app

1 Register app

Apple bundle ID ⓘ
com.OTTToolkit.Testpro

App nickname (optional) ⓘ
Testpro

App Store ID (optional) ⓘ
123456789

Register app

2 Download config file

× Add Firebase to your Apple app

✓ Register app
Apple bundle ID: com.OTTToolkit.Testpro, app nickname: Testpro

2 Download config file Instructions for Xcode below | [Unity](#) [C++](#)

Download GoogleService-Info.plist

Move the GoogleService-Info.plist file that you just downloaded into the root of your Xcode project and add it to all targets.

Next

Save the downloaded file along with the Bundle ID.

Then download the "GoogleService-Info.plist" file.

Click the next button for the next three steps, then click the "Continue to the console" button.

× Add Firebase to your Apple app

- ✓ Register app
Apple bundle ID: com.OTTToolkit.Testpro, app nickname: Testpro
- ✓ Download config file
- ✓ Add Firebase SDK
- ✓ Add initialisation code
- 5 Next steps

You're ready!

Make sure that you take a look at the [documentation](#) to learn how to get started with each Firebase product that you want to use in your app.

You can also explore [sample Firebase apps](#).

Or, continue to the console to explore Firebase.

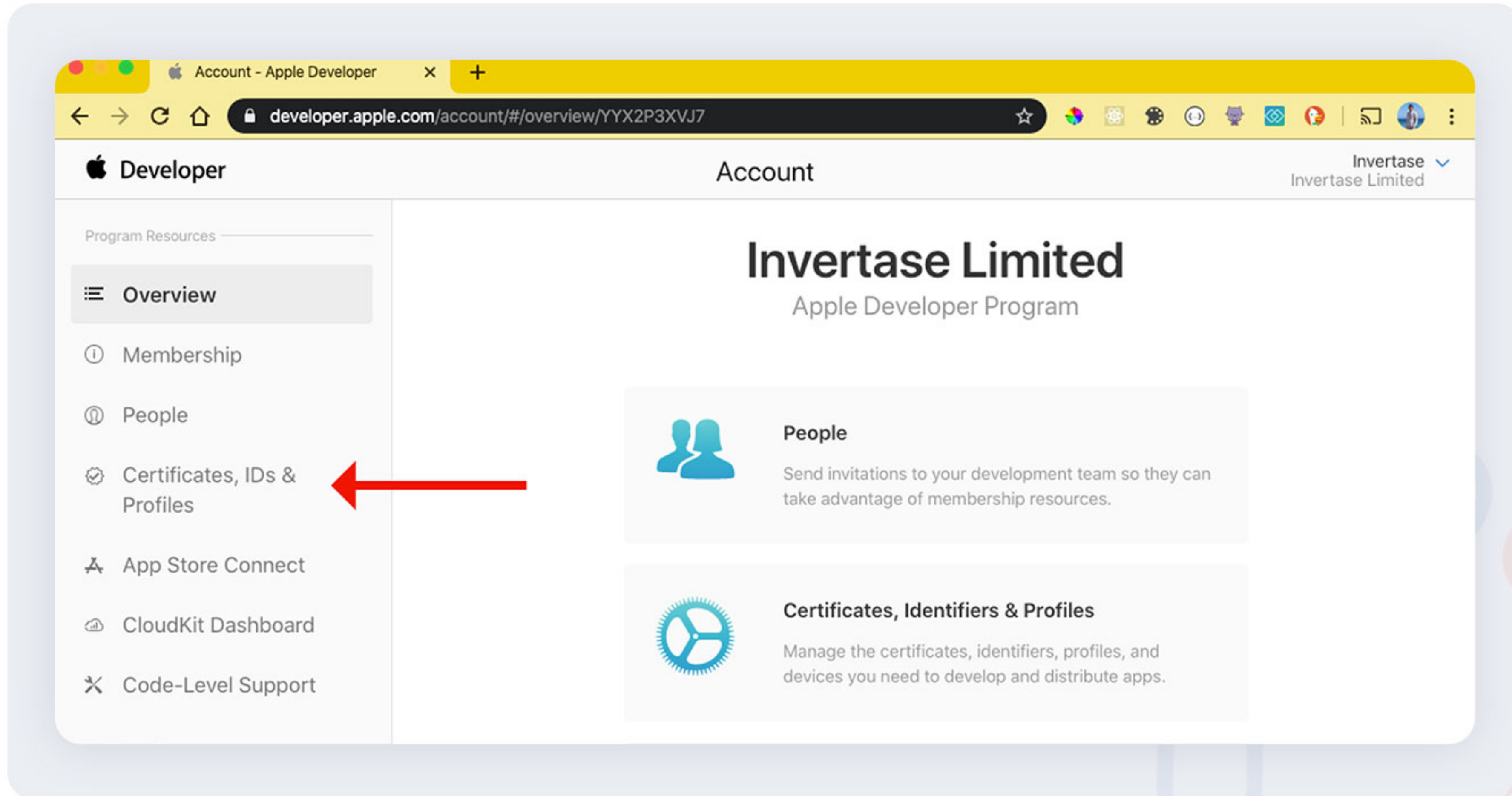
Previous [Continue to the console](#)





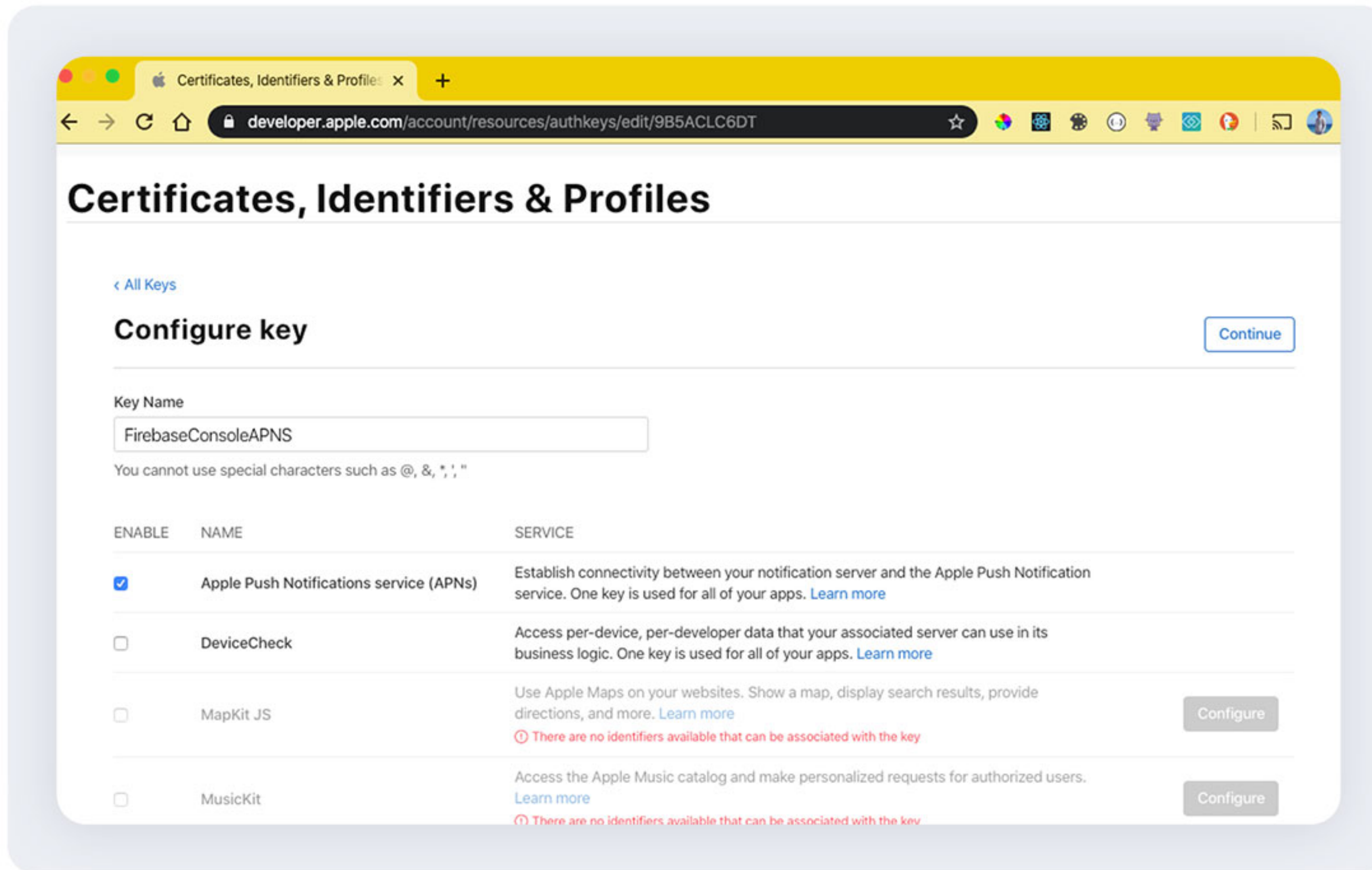
Now back on your iOS device,

Make sure you have access to your Apple Developer account and navigate to the "Certificates, ID & Profiles" menu.



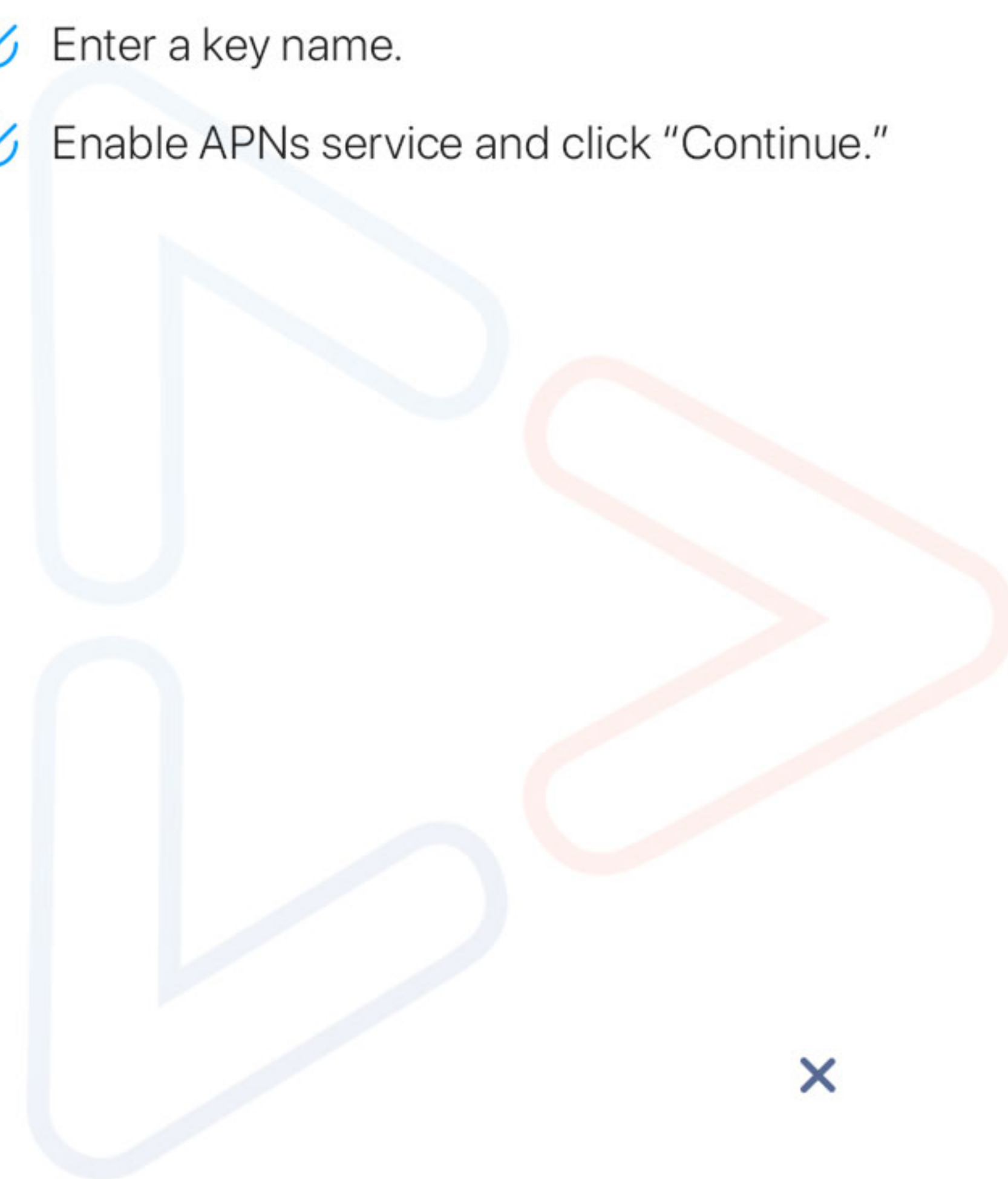
Now generate a key to give full access over the Apple push notification service (APNs).



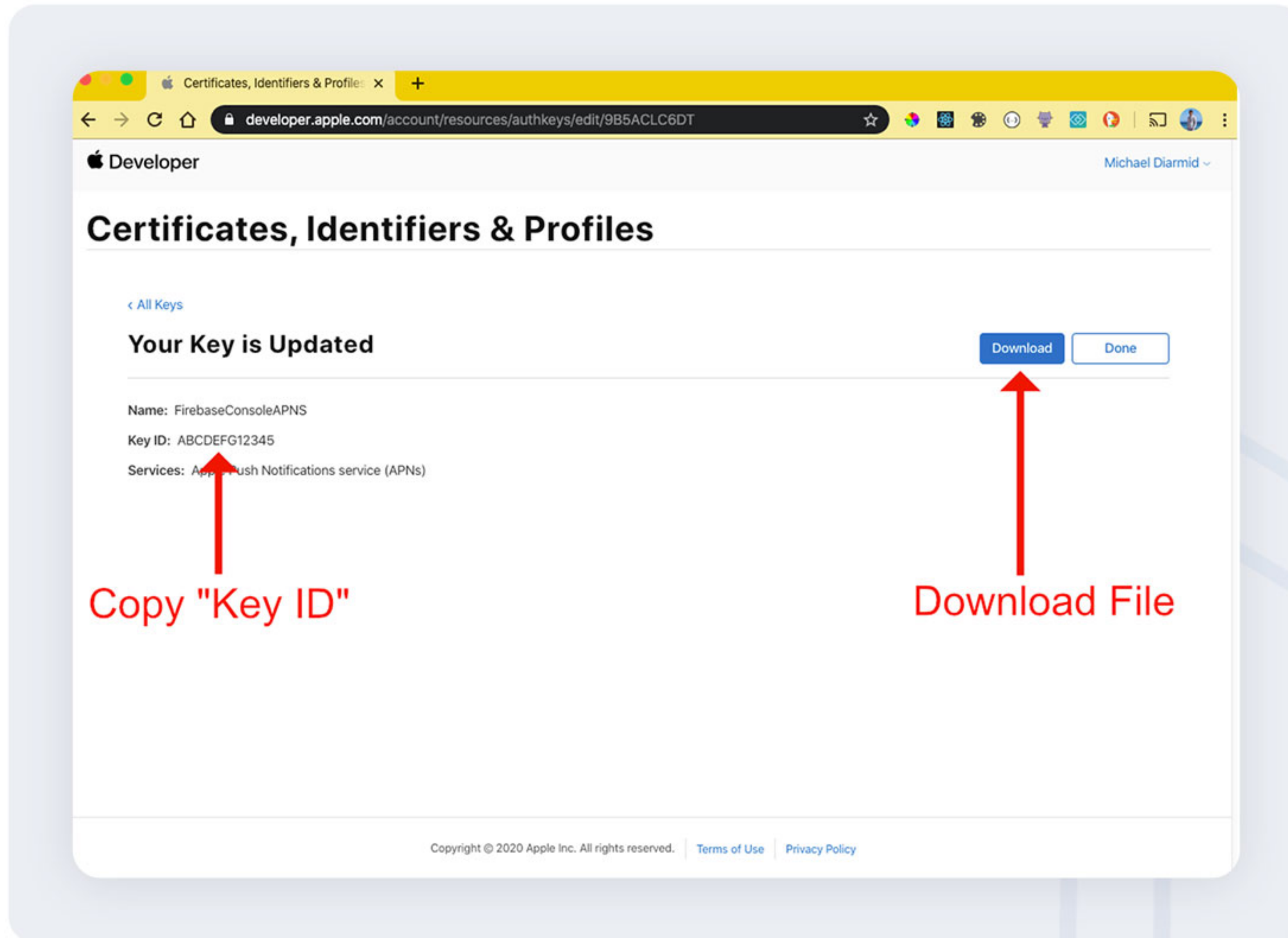


- ✓ Enter a key name.
- ✓ Enable APNs service and click "Continue."

Once saved, you'll get provided with a Key ID and an option to download the file.



Copy the ID and download the file.



Copy "Key ID"

Download File



Project settings

General **Cloud Messaging** Integration Service accounts Data privacy Users and permissions App Check (BETA)

Project credentials

[Add server key](#)

Key	Token
Server key	AAAAbhFyVVk:APA91bEDj1LpNQsVnEsiU4O2-56AP8xzbiXULrciKJgxUW6GIRnCQomWaHXcpj3JUJ5poRH9vDD6jgMnvfta nul62xiXKz4w4jEPbDr5gBsAcmmOoMG6CBI7zAm3PobTvvoXuVAEX_A2g
Sender ID	472739108185

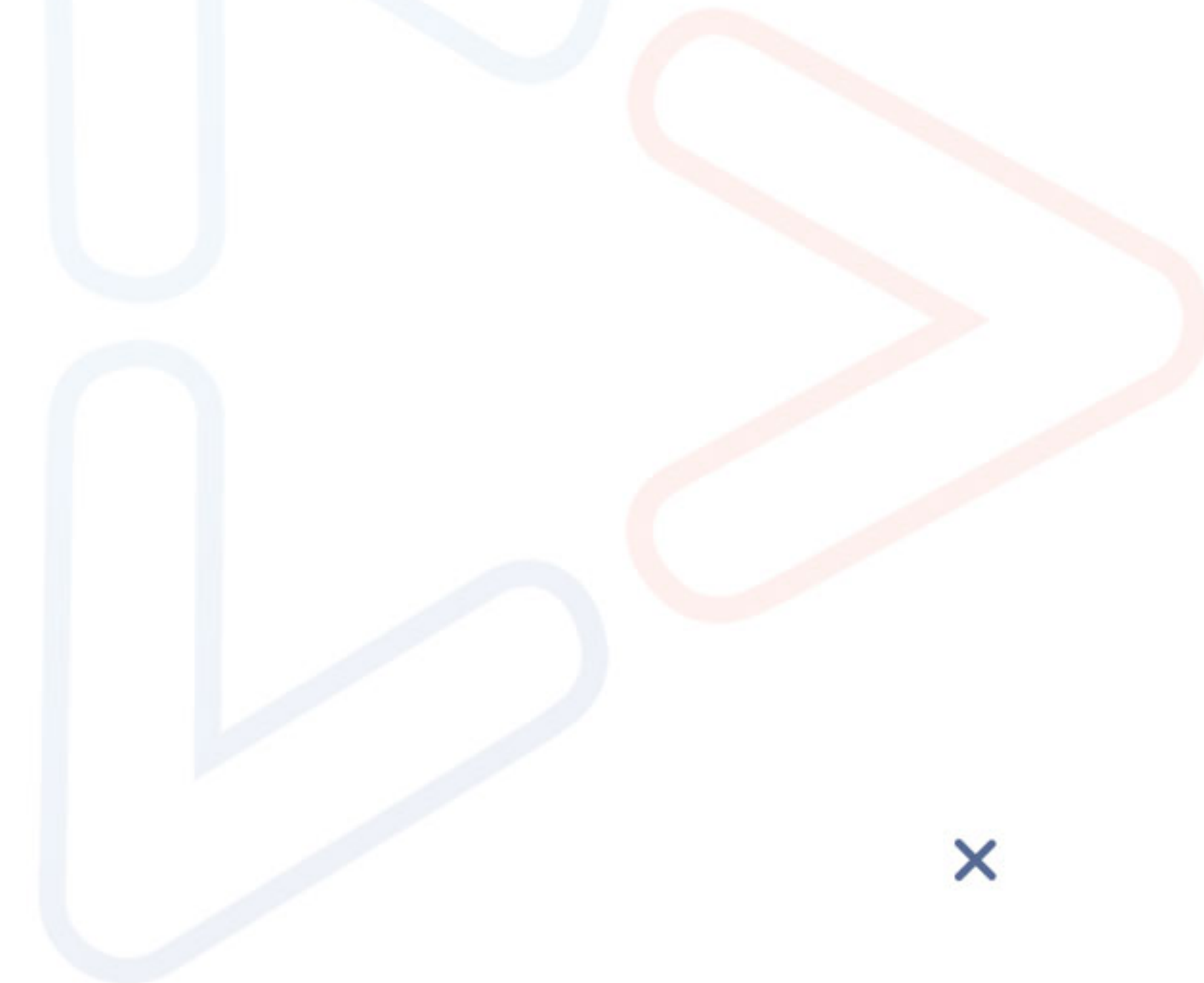
Apple app configuration

Apple apps	APNs Authentication Key						
<p>Testpro com.OTTToolkit.Testpro</p>	<p>★ Configuration with auth keys is recommended as they are the more current method for sending notifications to Apple devices</p> <table border="1"><thead><tr><th>File</th><th>Key ID</th><th>Team ID</th></tr></thead><tbody><tr><td>No APNs auth key</td><td></td><td></td></tr></tbody></table> <p>Upload</p>	File	Key ID	Team ID	No APNs auth key		
File	Key ID	Team ID					
No APNs auth key							



Now get back to the Firebase project settings,

- ✓ Go into the "Cloud Messaging" section.
- ✓ Click on the "Upload" button to upload the key and the file.





Upload APNs auth key ✕

APNs auth key ?

Drag file here to preview
Supports P8

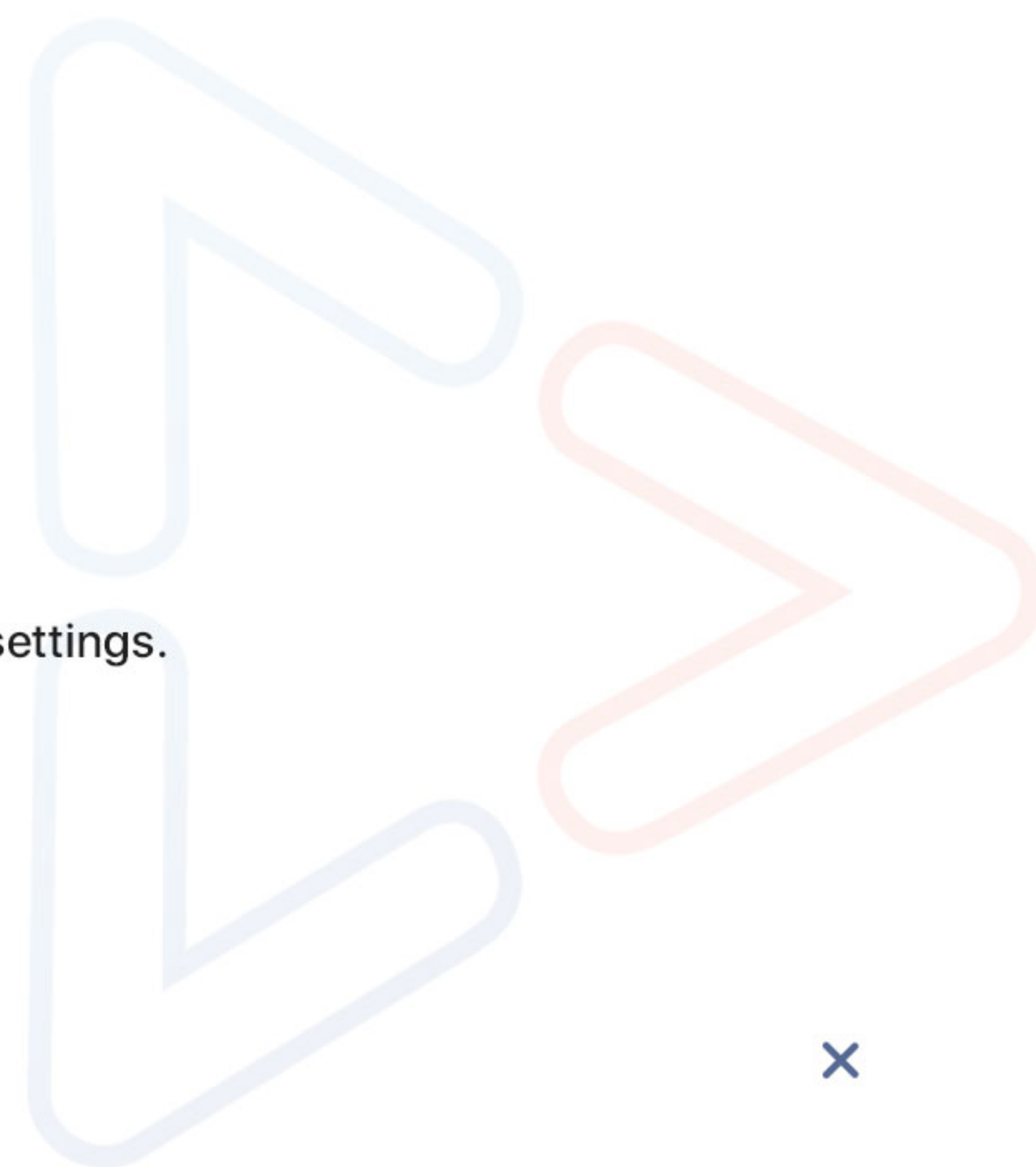
[Browse](#)

Key ID (required) ?

Team ID (required) ?

Cancel Upload

You have completed the initial setup for push notification settings.





**You have successfully
completed the initial setup for push notification settings.**

