

Bloggging your way to success in tech

Deepen your technical expertise, foster community ties, and advance your career
by sharing your knowledge online

Natalia Panferova, 13 April 2024

About me



Senior iOS developer at Trade Me



Founder at Nil Coalescing Limited

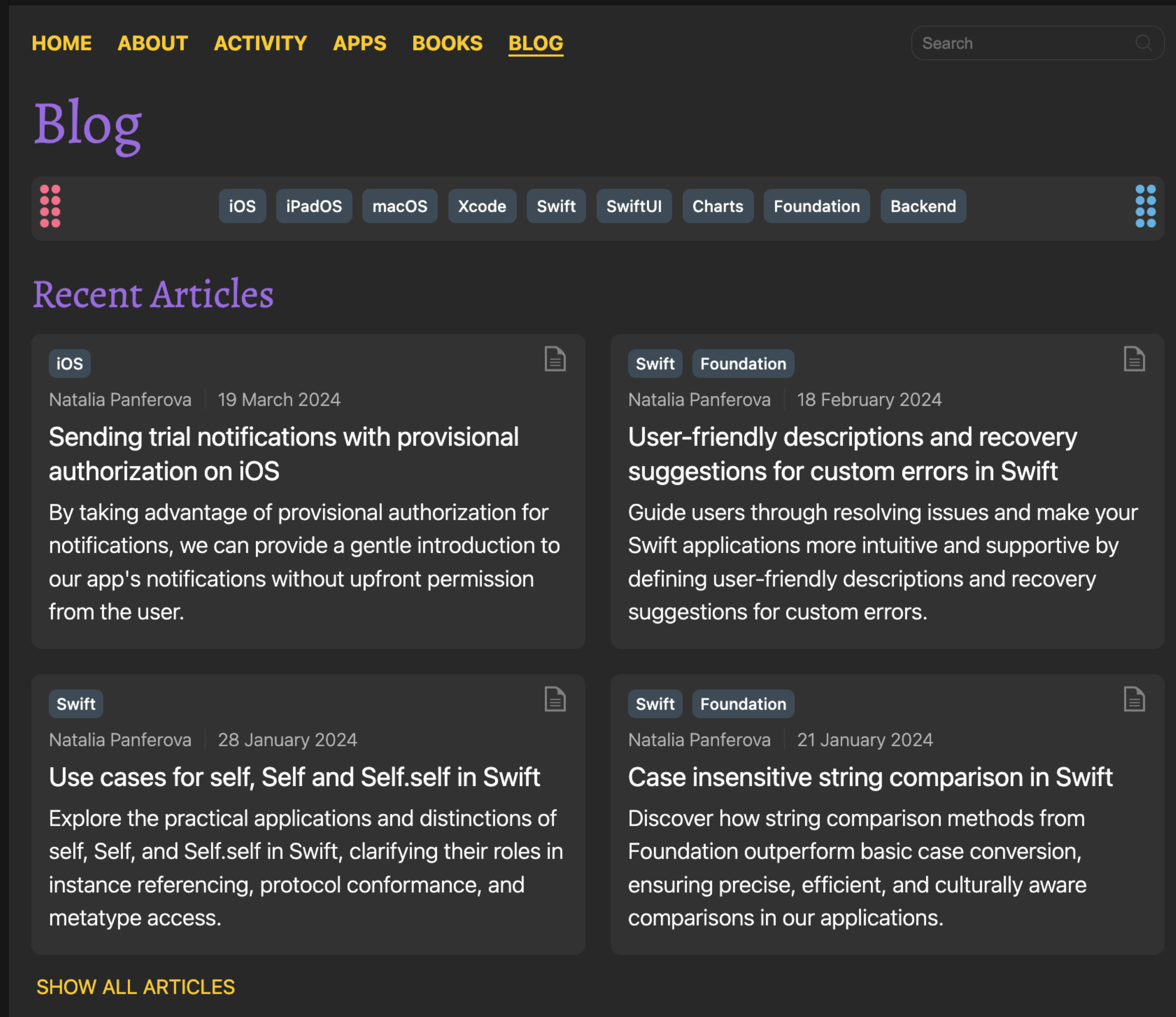


Based in Alexandra, Central Otago



Natalia Panferova

About my blog



Website:

nilcoalescing.com

Daily traffic:

3000 readers on average

Peak traffic:

10000 daily readers

Topics:

Swift, iOS, macOS, SwiftUI etc.

Content volume:

Over 100 articles

About my journey

Started the blog



Jan 2020

Nov 2020



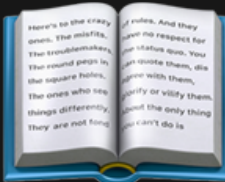
Joined the SwiftUI team
at Apple

Resumed blogging



Apr 2022

Sept 2022



Published a book

Joined Trade Me



Apr 2023

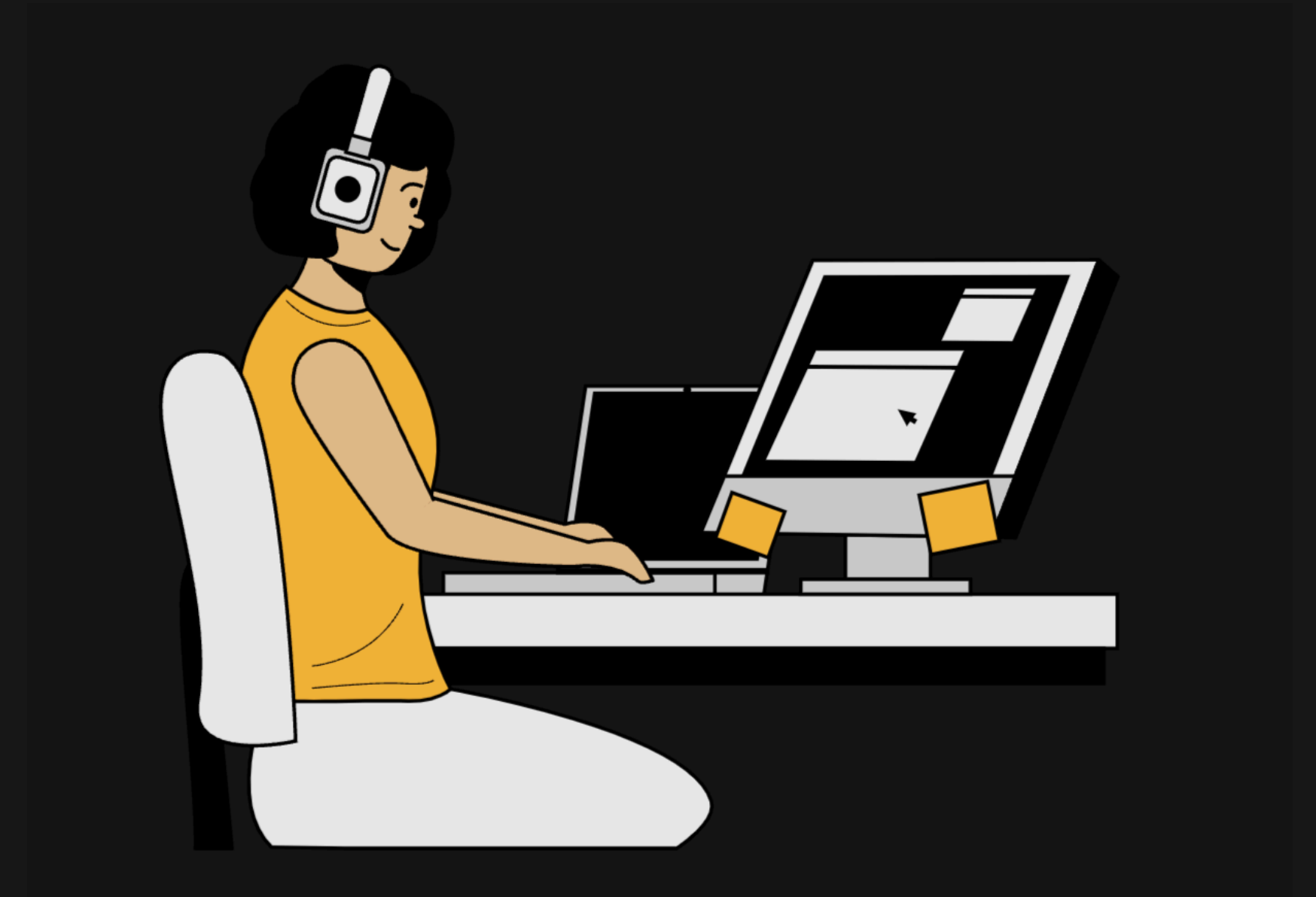
Agenda

👤 Benefits of sharing knowledge online

📝 Starting a technical blog

📢 Promoting your blog on social media and through newsletters

💰 Blog monetization strategies





Benefits of sharing knowledge online

Grow your skills and knowledge

1. Improve writing and communication skills



Transform complex ideas into clear and concise explanations

2. Deepen understanding of technical topics



Solidify your grasp of difficult concepts through teaching them to others

3. Create personal knowledge base



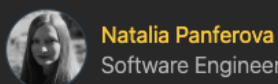
Compile your learnings into a go-to resource for future reference

Document your learnings in detail for future you

Example from my blog explaining push notification setup on iOS:

16 August 2023

iOS



iOS app setup for remote push notifications

Remote push notifications are messages that app developers can send to users directly on their devices from a remote server. These notifications can appear even if the app is not open, making them a powerful tool for re-engaging users or delivering timely information. They are different from local notifications, which are scheduled and triggered by the app itself on the device.

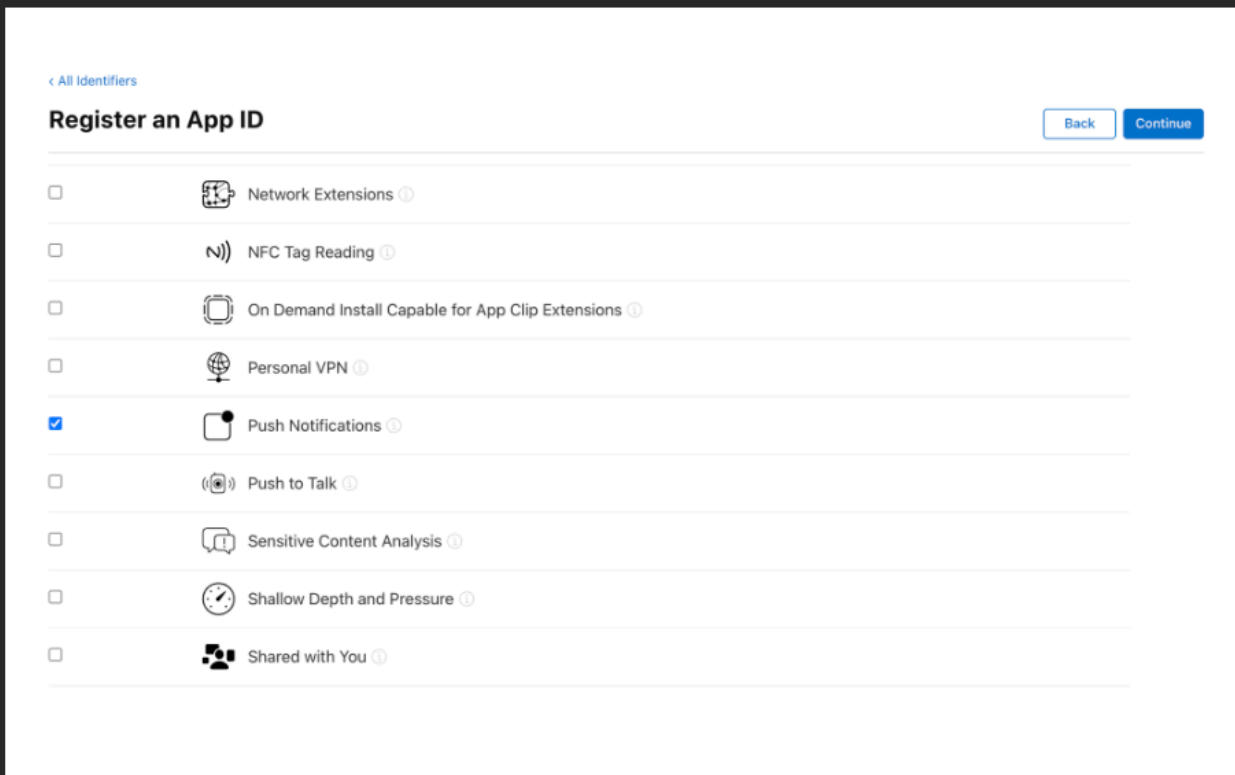
Adding remote notifications capability to an iOS app is a quite involved process that includes several steps and components. This post will walk you through all the necessary setup so that you can enable remote push notification functionality in your iOS project.

Note that to be able to fully configure and test remote push notifications, you will need an active Apple developer account.

App ID configuration

To configure your app ID with push capabilities, go to **Certificates, Identifiers & Profiles** in your Apple developer account, create a new app ID or choose an existing one that corresponds to your project where you are adding push notifications support.

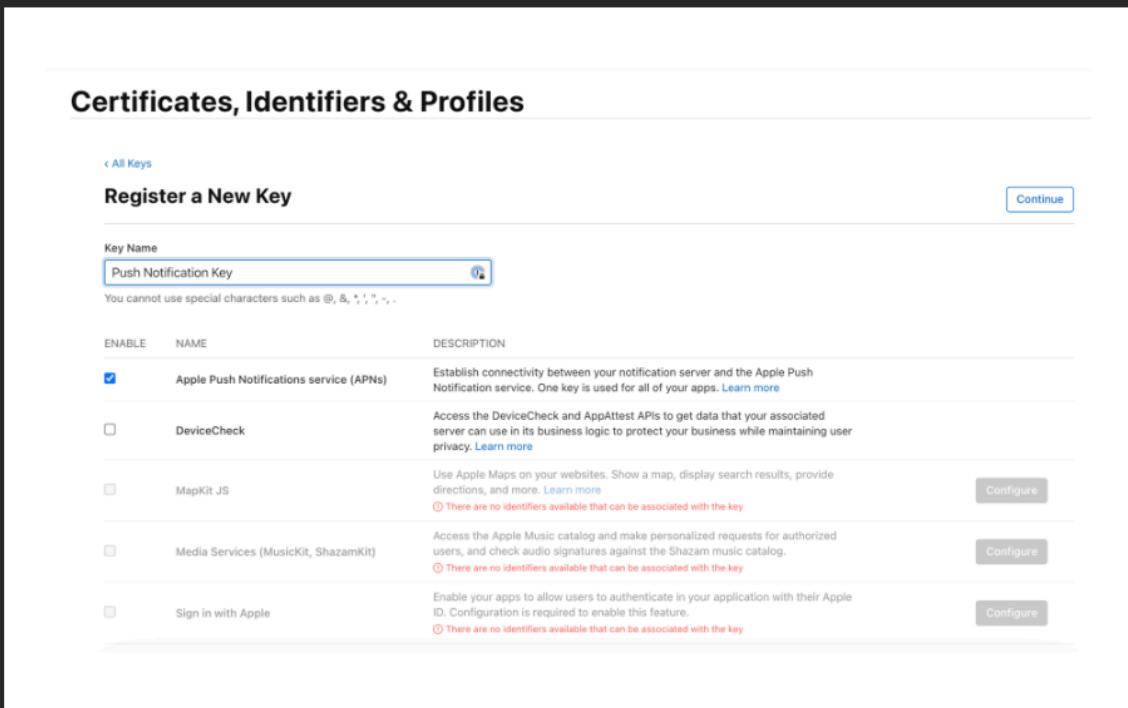
Make sure that the "Push Notifications" capability is selected.



Push certificate

If you don't have a push notifications key associated with your developer account yet under **Certificates, Identifiers & Profiles**, then you will need to create one.

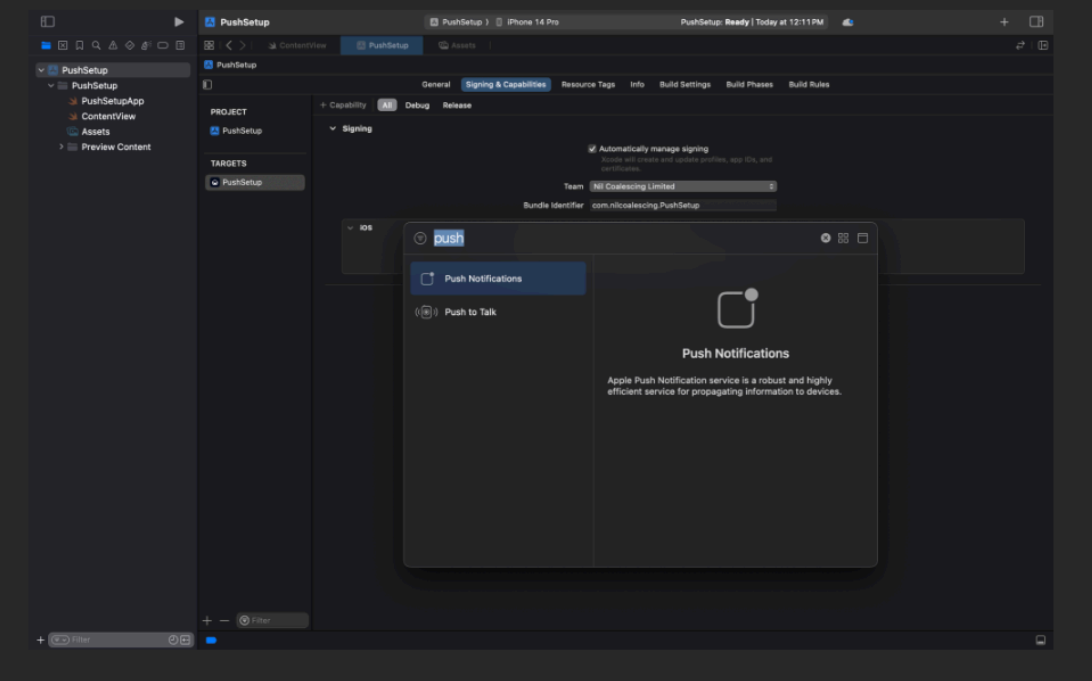
Name it "Push Notification Key" and make sure to check the "Apple Push Notifications service (APNs)" checkbox.



Once created, download the key and keep it somewhere safe. You will need this key to connect to APNs from your server when you are ready to send remote push notifications to your app users.

Xcode project configuration

Open your Xcode project and go to the project settings. In "Signing and Capabilities" settings add the "Push Notifications" capability.



Requesting user permission

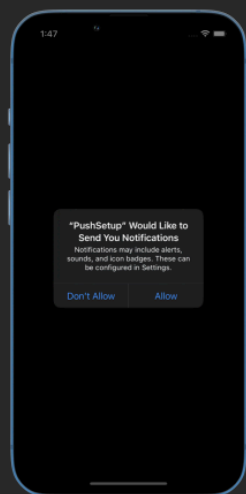
This step is only necessary if you are going to send alert push notifications, sometimes called foreground notifications. These display an alert, sound, or a badge on the app's icon and used to inform users of new content, updates, or other important information that requires immediate attention. Background push notifications or sometimes called silent notifications don't require user permission as they are not visible to the user and are not disruptive. They are great for updating content, syncing data, or performing maintenance tasks without interrupting the user.

To be able to show alerts on the user's device we first need to ask for permission. We should do it in response to a user action so that they are not surprised by the shown dialog. It's important that the user has the context to understand why the app needs authorization for notifications.

We can request permission with the following code.

```
Task {
    let center = UNUserNotificationCenter.current()
    let success = try await center.requestAuthorization(options: [.alert])

    if success {
        // proceed with registration
    }
}
```



The system prompts the user to grant or deny the request the first time, subsequent requests don't prompt the user. Once we obtain the user authorization we can proceed to register our app to receive remote push notifications.

Registering for push notifications

To get the device token from APNs we need to register for remote push notifications. This can be done after getting user permission for alerts. It should also be done on every app start, because old tokens can be no longer valid.

If your app uses the SwiftUI app lifecycle you'll need to add an application delegate to your project. With the UIKit lifecycle you can just use your existing UIApplicationDelegate.

In a SwiftUI app create a new file called AppDelegate.swift, declare a class conforming to UIApplicationDelegate and add application(_:didFinishLaunchingWithOptions:) method. If you are planning to send alerts rather than background push notifications, then make sure to check for notifications authorization status on the user's device before registering.

Handling the device token

Once registered, the app will receive a device token which needs to be sent to your backend server. This token will be used by your server to send notifications to this specific device.

To retrieve the token add the application(_:didRegisterForRemoteNotificationsWithDeviceToken:) method to your app delegate. Depending on what your server expects, you might need to convert the token to a string before sending it. Either way, it will be useful to have it in a string format for testing.

```
class AppDelegate: NSObject, UIApplicationDelegate {
    func application(
        _ application: UIApplication,
        didRegisterForRemoteNotificationsWithDeviceToken deviceToken: Data
    ) {
        let token = deviceToken.reduce("") { $0 + String(format: "%02x", $1) }
        print("device token: \(token)")

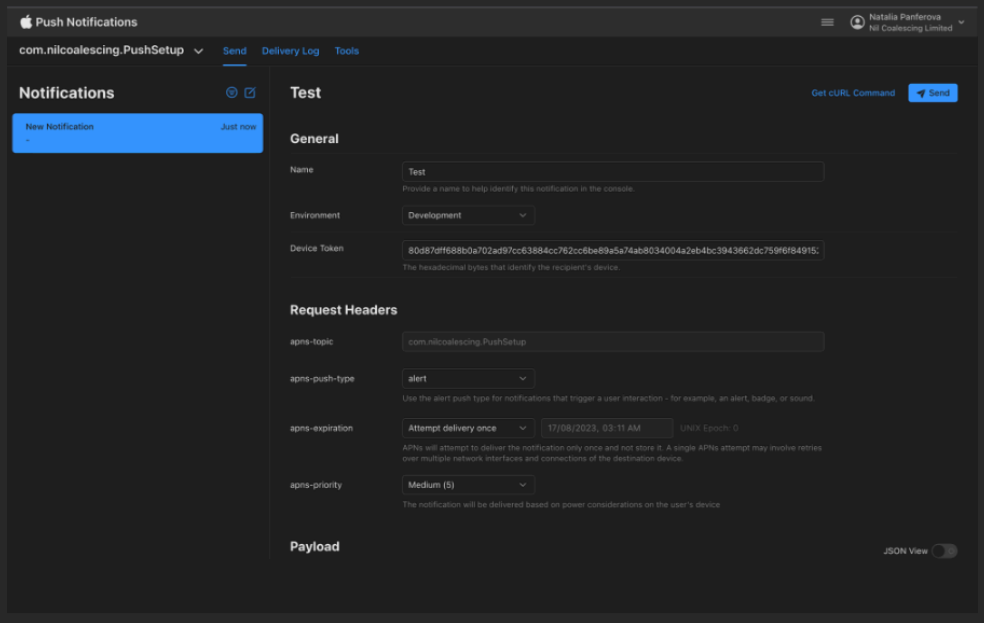
        // send the token to your server
    }
}
```

Testing remote push notifications

At this point your app is configured to receive basic push notifications. You will need to go through additional setup steps if you want to support rich push notifications with images, actions buttons, deep links or custom UI. I may cover these later in future posts.

To test that the basic setup is working properly we can use the new **Push Notifications Console** from Apple to send ourselves a test push.

Once you open the console, select your app ID in the top left corner and create a test push notification. To target your test device or simulator, you can copy the printed device token and paste it in the "Device Token" field of the notification form.



Grow the number of free online resources

Free resources for iOS developers:

Danijela Vrzan

iOS DEVELOPER

WRITER. SPEAKER.

USE YOUR LOAF



HACKING WITH SWIFT

SwiftLee

A weekly blog about Swift, iOS and Xcode Tips and Tricks



Kodeco

THE NEW RAYWENDERLICH.COM



Nil Coalescing

[SC]()

iOS. Apple. Indies. Plus Things.



TANASCHITA.COM

Swift with Majid

Connect with developers across the world

Messages of support I got over the years:

I've been your Moroccan loyal reader for months now! I always come to your content for any iOS/SwiftUI matter.

Hey Natalia, just wanted to say your posts are so great. The information you share is so helpful and clear.

I just wanted to reach out and say that some of your posts have been really, really helpful to me! Namely the one about divider line insets!
Thank you!

Hi Natalia, I want to thank you for your awesome articles and tips. I enjoy them very very much.


Hi Natalia! I'm a big fan of your blog and YouTube videos. I'm an iOS engineer working at Hyundai Motor Company. Hope we can connect!

Thanks for the great article on custom EnvironmentKeys in Swift! That really came in handy for me today adding a deep linking feature.

Share new approaches and perspectives


- Explain common topics in your own unique way
- Enhance available information with new details and examples
- Help people with different learning styles and experience levels to grasp new concepts

Reading multiple explanations of Swift closures helped me to grasp the concept better

 Hacking with Swift
<https://www.hackingwithswift.com/sixty/closures-su...>


Closures summary - a free Hacking with Swift tutorial

Closures can accept parameters and return values, like regular functions. You can pass closures into functions as parameters, and those closures can have ...

 LinkedIn · Mehmet Ali Demir
30+ reactions · 7 months ago


Understanding Closures in Swift

Closures in Swift are like self-contained blocks of functionality that can be assigned to variables or passed as arguments to functions.

 CodeWithChris
<https://codewithchris.com/introduction-to-swift-closure...>


Introduction to Swift Closures in 2023 - A Beginner's Guide

11 May 2023 — A Closure in Swift is basically a block of code that are executed together without creating a named function. Closure Expression Syntax.

 GeeksforGeeks
<https://www.geeksforgeeks.org/closures-in-swift>


Closures in Swift (With Examples)

17 Mar 2023 — In Swift, Closures are known as self-contained blocks of functionality that can be passed around and used inside your code.

 Apple
<https://forums.developer.apple.com/forums/thread>

Understanding Swift closures

So you closure takes an UIAlertAction (the parameter named action) and returns nothing. Inside your closure definition you may call this parameter as you like ...

 Medium · Lingeswaran Kandasamy
1 year ago

Swift Closures Explained

closure is a self-contained block of code that can be assigned to a variable or passed as an argument to a function. Closures are similar to ...

Open up new career opportunities

- Make your work visible
- Grow your online following and personal brand
- Receive invites to podcasts, workshops and conferences
- Get headhunted by top companies

When Apple invited me for an interview,
they explicitly mentioned my blog

Hi Natalia! The SwiftUI team is hiring. From your blog posts it's clear that you're passionate about using the framework. Would joining Apple to help build the future of SwiftUI be something you're interested in learning more about?

Let me know! I can also answer any questions about the position and our interview process

Jul 1, 2020, 11:02 PM



Starting a technical blog

Select a blogging platform

Services that host your blog on their servers:



Content management systems:



Self-hosted static site generators:



A closer look at Publish

Publish - a static site generator built for Swift developers.

When using Publish:

- each website is defined as a Swift package
- content is generated based on Markdown files

Repository: github.com/JohnSundell/Publish

```
struct DeliciousRecipes: Website {
    enum SectionID: String, WebsiteSectionID {
        case recipes
        case links
        case about
    }

    struct ItemMetadata: WebsiteItemMetadata {
        var ingredients: [String]
        var preparationTime: TimeInterval
    }

    var url = URL(string: "https://cooking-with-john.com")!
    var name = "Delicious Recipes"
    var description = "Many very delicious recipes."
    var language: Language { .english }
    var imagePath: Path? { "images/logo.png" }
}
```

Example website configuration

Important considerations for a blogging platform

1. Accessibility support

Alt text for images, keyboard navigation, contrast, font sizes

2. Design and user experience

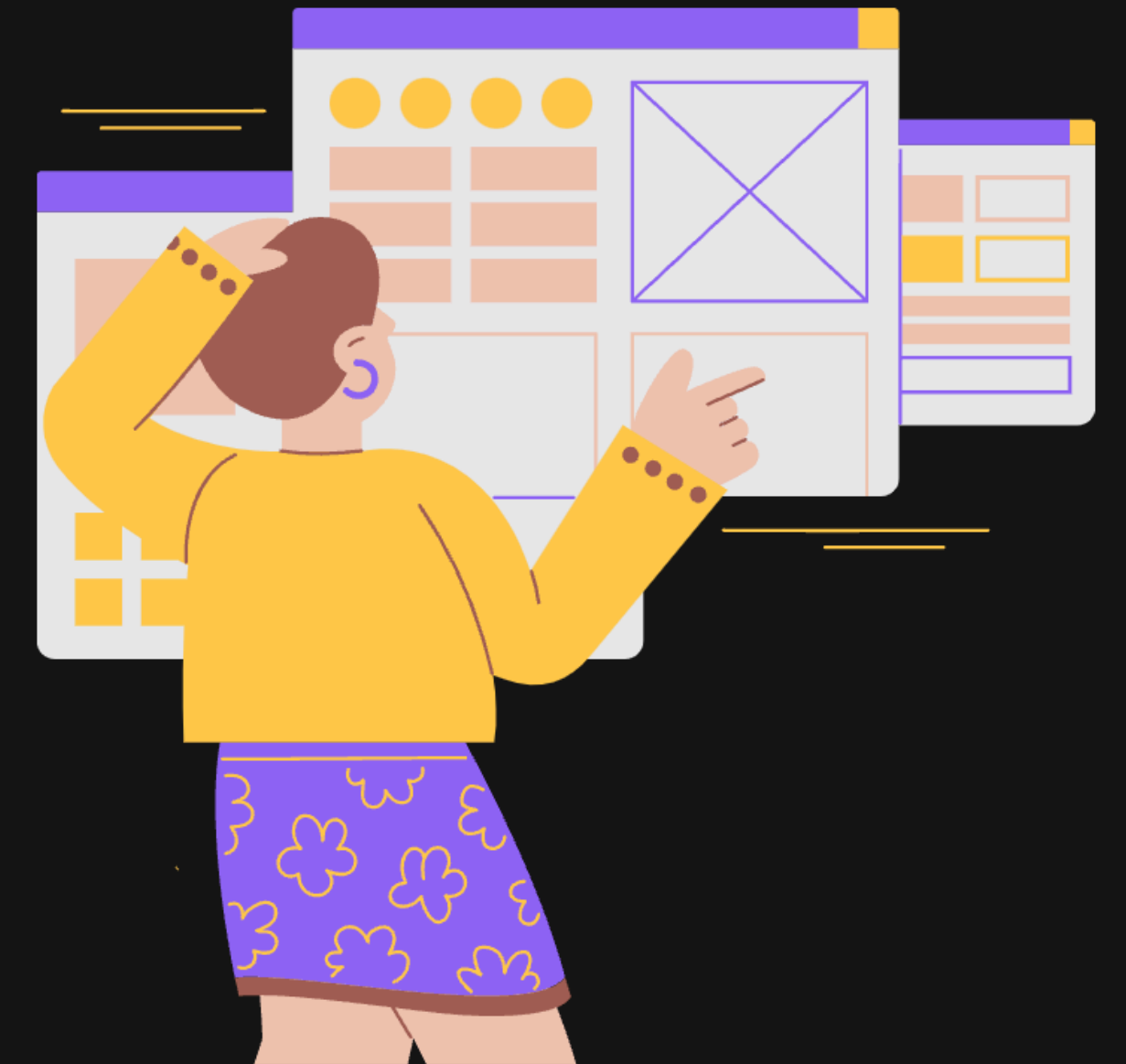
Support for different screen sizes, dark and light mode

3. Technical content presentation

Code snippets and syntax highlighting

4. Engagement and reach

SEO, analytics, familiarity for your audience



Finding inspiration for blog posts

 Personal learnings

“Exporting and importing localizations in a SwiftUI app”, “5 Lessons learned from my first year in cybersecurity”

 Project walkthroughs

“Building a Chat App with WebSocket”

 Latest developments in your field

“Overview of the new SwiftUI navigation APIs”, “Exploring the New Features of Angular 12”

 Solutions to common problems

“Align y-axis to leading edge in Swift Charts”, “Fixing memory leaks in iOS”

 Tools and technology reviews

“Serverless architectures: AWS Lambda vs. Azure Functions”

Types of blog posts

💡 **Short tips:** quick, actionable advice or solutions to common problems, easy to digest and implement.


📝 **Full articles:** in-depth exploration of topics, providing comprehensive insights and analysis.

📁 **Series of articles:** a deep dive into complex subjects, broken down into manageable, interconnected pieces for thorough coverage.


Short tips - an easy starting point

- Provide a quick solution to a common question
- Write a short paragraph to introduce the problem
- Include a code snippet that can be copied and integrated
- Add a link to more information on the topic

iOS macOS SwiftUI



Natalia Panferova
Software Engineer

 Quick Tip

Set custom actions for links in Text views

Since iOS 15 and macOS 12 SwiftUI Text views can contain interactive links created with Markdown or `AttributedString`.

To customize actions of such links we can use `openURL` environment value.

```
struct ContentView: View {
    var body: some View {
        Text("Visit our [website](https://example.com)")
            .environment(\.openURL, OpenURLAction { url in
                handleURL(url)
                return .handled
            })
    }

    func handleURL(_ url: URL) {
        // handle URL here
    }
}
```

A short tip example

Articles - detailed content pieces

- Start with a descriptive introduction
- Structure the post into subsections with titles
- Include links to documentation, code examples, screenshots etc.
- Provide a downloadable sample project
- Conclude with a summary

Series of articles - comprehensive subject breakdown

- Divide complex topics into manageable interconnected pieces
- Organize the series in a logical order
- Maintain a uniform structure through the series for coherence
- Summarize previously covered information in the introduction of subsequent posts
- Provide links to all the posts in the series

Make your content findable online

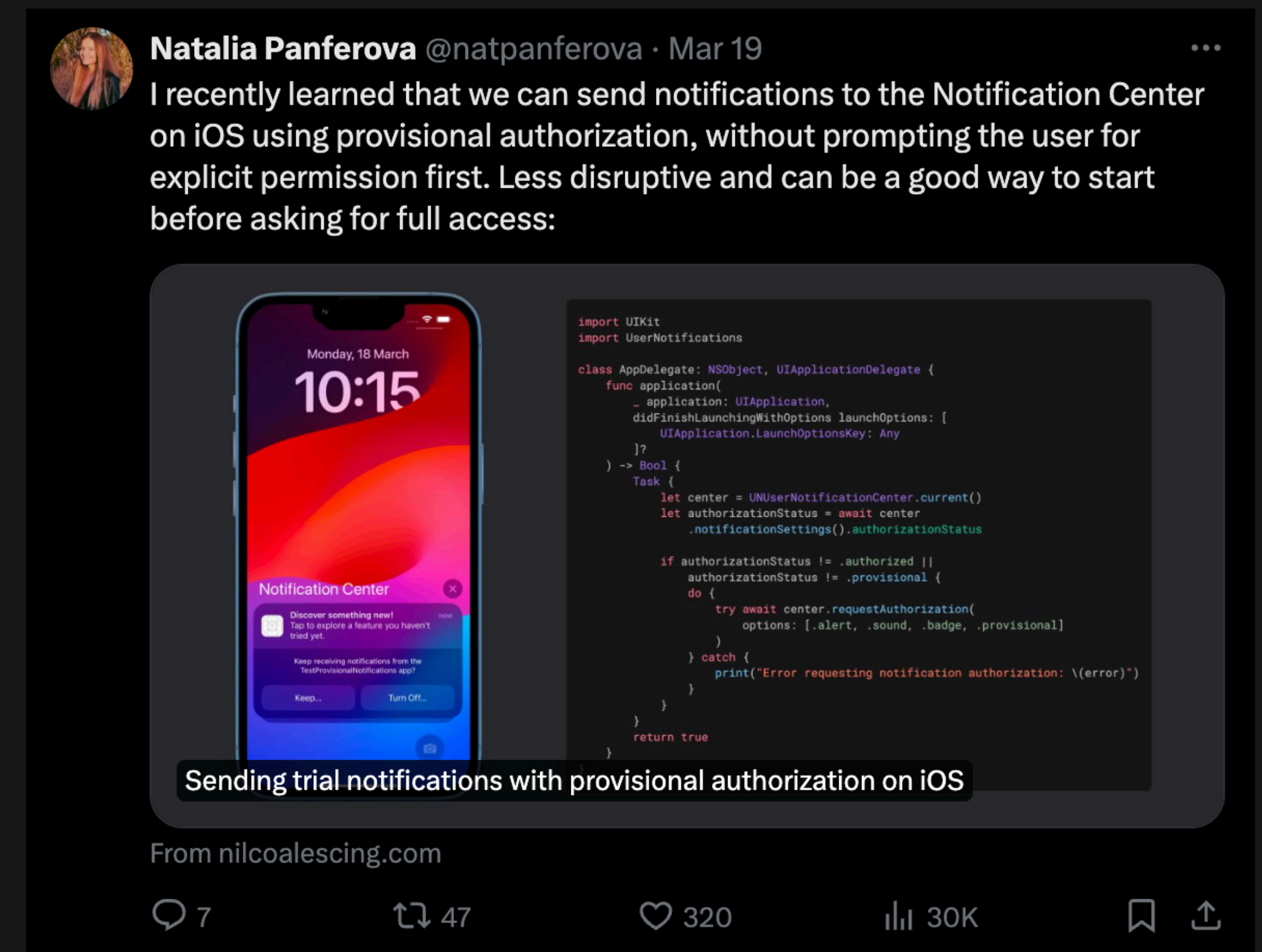
- 🔍 Craft clear and relevant titles, that include key search terms
- 🔧 Optimize your blog's HTML structure
- ➡️📱 Ensure mobile responsiveness & speed
- 🔑 Incorporate keywords throughout content
- 🔗 Implement internal linking



 Promoting your blog

Promote your posts on social media

- Set up your profile with an image and bio
- Post on as many platforms as you can
- Choose an appropriate time based on your audience's location
- Make sure your post includes the article banner
- Use relevant hashtags
- Engage with comments, answer questions



Example promotional post on X

Get featured in technical newsletters

- Curate a list of newsletters covering topics are you interested in
- Reach out to newsletter authors, share a link to your RSS feed
- Submit your article links to open-source newsletters on GitHub

Newsletter recommendations from the iOS community:



 iOS Dev Weekly

My recent post got featured in
iOS Dev Weekly

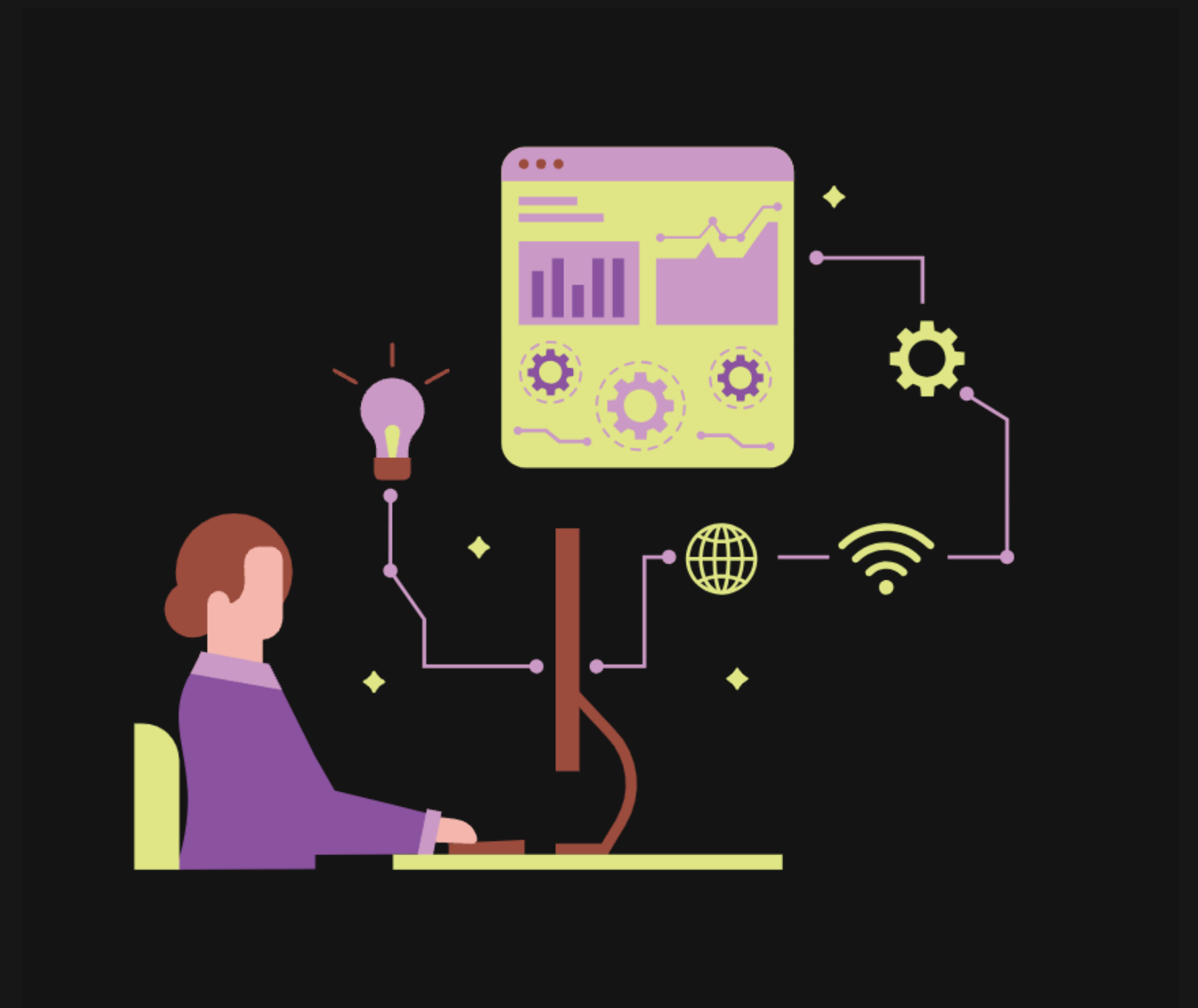
[Sending trial notifications with provisional authorization on iOS](#)

As [Natalia Panferova](#) explains in this article, “provisional” push notifications are a powerful way to show your users that your app can provide useful information via push before without needing any opt-in from users at all. This doesn’t use a new API, it was introduced with iOS 12, but I don’t see enough apps taking advantage of it.

nilcoalescing.com

Understand your blog's performance

- Choose a tool to get website traffic data (for example, Google Analytics)
- Look at page views, unique visitors, and average time spent on page
- Understand where your traffic comes from
- Identify popular posts
- Use analytics to optimize content strategy



Blog monetization strategies

💰 Ask for tips and donations

Encourage reader support through platforms like GitHub Sponsorship.

Example from Nil Coalescing:

If you are enjoying our blog and would like to support us, you can [sponsor us on GitHub](#).



Become a sponsor to Nil Coalescing



Nil Coalescing

NilCoalescing

📍 New Zealand

We are a small team of two software engineers living in New Zealand. We build applications for Apple platforms and try to share as much knowledge as we can with the developer community.

We have a blog where we regularly post detailed articles and shorter tips that everyone can access for free: <https://nilcoalescing.com/blog/>

We are also active on Twitter where we engage with the community, answer questions and share what we learn. You can reach out to us [@natpanferova](#) and [@hishnash](#).

We post code samples on GitHub that developers can use in their own projects, and work on open source projects such as [Django Channels Rest Framework](#).

By sponsoring us you could allow us to spend more time to share even more useful resources with the community.

Select a tier

Monthly

One-time

\$ 3  a month

A Public Sponsor achievement will be added to your profile.

\$5 a month

Select

🐶 cover our website hosting costs

\$10 a month

Select

🐶 support our team dog Phoebe with some monthly treats

Promote personal projects or products

Leverage your audience for personal projects, turning readers into customers.

Example from Nil Coalescing:



CHECK OUT OUR BOOK!

Integrating SwiftUI into UIKit Apps


UPDATED FOR iOS 17!

- ◆ Discover various ways to add SwiftUI views to existing UIKit projects
- ◆ Use Xcode previews when designing and building UI
- ◆ Update your UIKit apps with new features such as Swift Charts and Lock Screen widgets
- ◆ Migrate larger parts of your apps to SwiftUI while reusing views and controllers built in UIKit

Offer sponsorship opportunities

Allow businesses to reach your audience through your blog or social media channels.

Examples from other iOS blogs:



Timing Is Everything

Boost user engagement with precise context-aware prompts using ContextSDK. Over 170 signals at your fingertips. Transform your iOS app today.

[Try for Free](#)

Sponsored

This sponsored message helps keep the content on this site free. Please check out this sponsor as it directly supports me and this site.

\$UPERWALL

SPONSORED Superwall lets you build & test paywalls without shipping updates. Run experiments, offer sales, segment users, update locked features and more at the click of button. Best part? It's FREE for up to 250 conversions / mo and the Superwall team builds out 100% custom paywalls – free of charge.

[Learn more](#)

Sponsor Hacking with Swift and reach the world's largest Swift community!

You can easily support sarunw.com by checking out this sponsor.

SPONSOR



Note Taking: Quickly take note using Voice, Camera, or Typing

[Try it for Free](#)

Sponsor sarunw.com and reach thousands of iOS developers.

Let's recap!

Summary

Why have a blog?

- Enhance your grasp of complex subjects
- Join a network of like-minded people across the world
- Open doors to new professional paths

How to get started?

- Choose the right blogging platform
- Share your insights as short tips or longer articles
- Utilize social media and newsletters to amplify your reach
- Explore various ways to generate income



Where to find me



X (Twitter):

[@natpanferova](https://twitter.com/natpanferova)



Mastodon:

[@natpanferova@mastodon.cloud](https://mastodon.cloud/@natpanferova)



LinkedIn:

linkedin.com/in/nataliapanferova



Website:

nilcoalescing.com