

# How to Widget

Connor Nelson  
iOS Developer, Capital One

- Intro, hi
- I'm here to talk about widgets
- So first, let's take a look at what your phone will look like after becoming a widget master



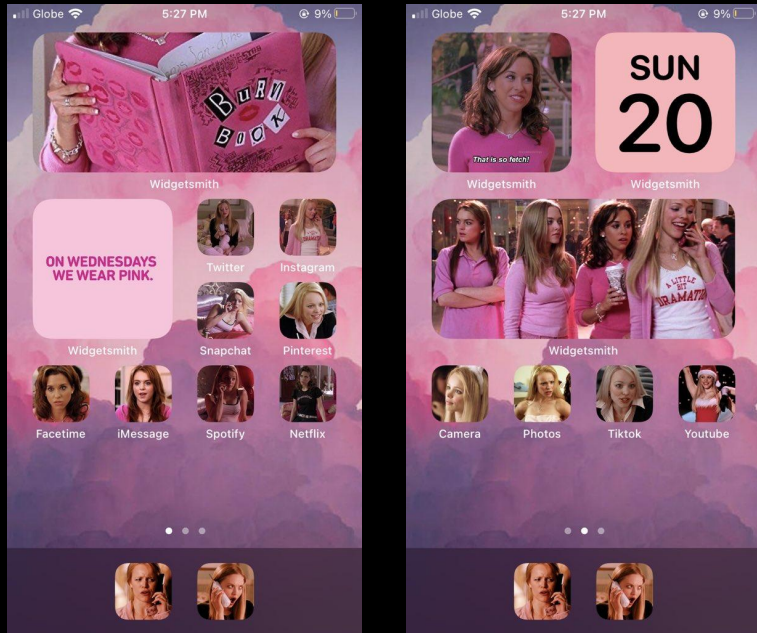
<https://twitter.com/okpng/status/1306850786085761025>

- For the animal crossing fans, you can finally have a real life nook phone



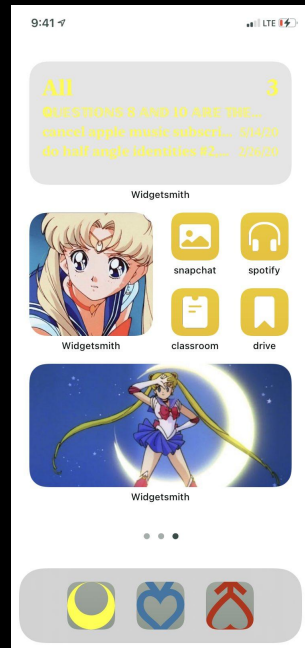
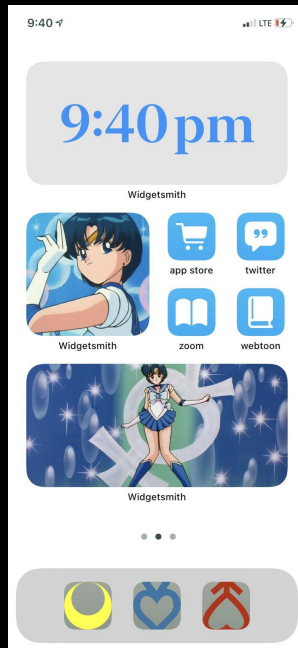
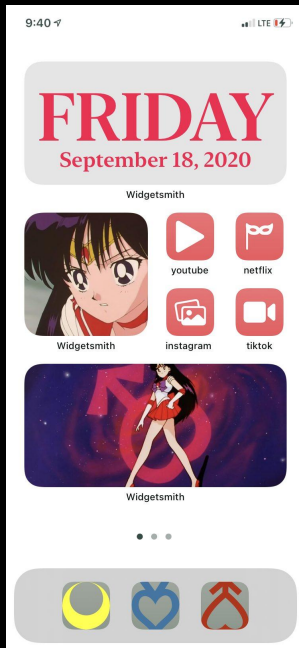
<https://twitter.com/wholelottajenni/status/1307868397787521027>

- Or take it back to gaming in the mid 90's



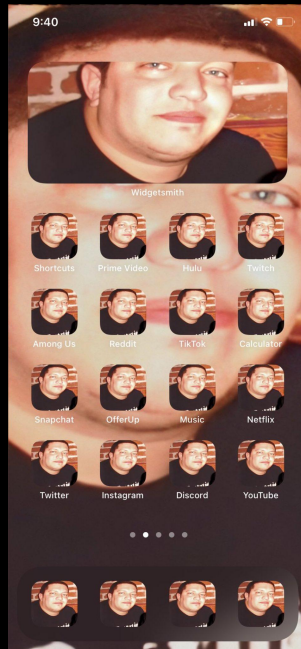
<https://twitter.com/lzldjem/status/1307907062421008385>

- I'm sure nobody has ever seen this movie



<https://twitter.com/hyakukisses/status/1307132730354872320>

- But maybe you're a big fan of anime



<https://twitter.com/shampagnepaki/status/1307995244907188224>

- And who doesn't love a good meme
- But seriously, today we aren't going to be creating an aesthetic home screen, although most of these home screens do incorporate widgets.
- Instead, we'll start off by going over some widget concepts at a high level, then get into coding.
  - Where we will create a fully functional widget from the ground up
  - This is going to include some live coding, so wish me luck
- And let's get started

# What is a Widget?

- A SwiftUI view that updates over time
- An extension built on your main app
- A relevant, glanceable experience
- Not a mini-app
- Widgets can support three sizes



<https://developer.apple.com/videos/play/wwdc2020/10028>

So, what is a widget?

- First, it's a SwiftUI view that updates over time.
  - SwiftUI only no UIKit
- Widgets are built as extensions from a main app, and should present relevant, glanceable information from your main app
- This is very important: widgets are not mini apps
  - There is no scrolling or animations
  - You should think of them as a projection of content from your app
- Widgets can come in 3 sizes, as seen here, but you don't have to support all 3

# Widgets can appear on...

- iPhone home screen and Today View
- iPad Today View
- macOS Big Sur notification center



<https://developer.apple.com/videos/play/wwdc2020/10028>

Widgets can appear in a couple different different places, such as...

- The iOS home screen or Today View
- The Today View on iPad
  - Unfortunately not on the home screen
- The notification center on macOS
- Today we will be focusing on iOS



# Widget Configuration

- The back of a widget can display a configuration UI
- The UI is generated by the system, the parameters are defined by you
- Widgets don't have to be configurable

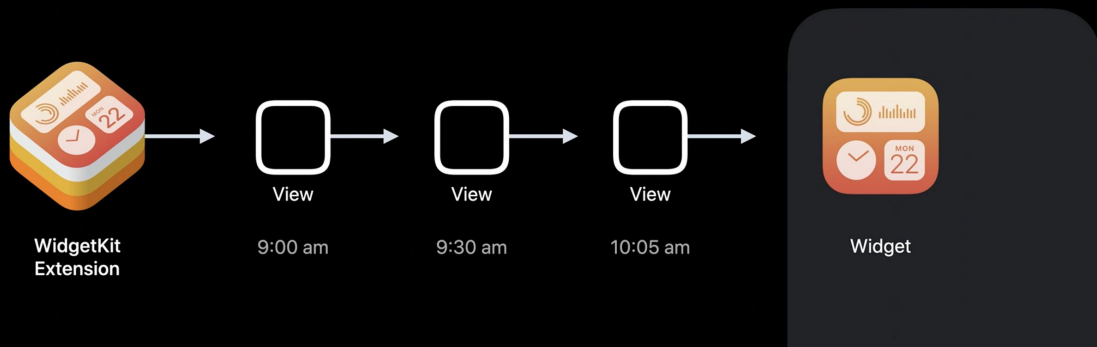


Widgets can also be configured by the user

- If a widget is configurable, the back will show a configuration UI
- This UI is generated by the system
- The configurable parameters are defined by you using Intents
- Widgets don't have to be configurable

# Widget Timelines

- A timeline is a series of views and when to display them
- Entries are serialized to disk and just-in-time rendered by the Home Screen
- The timeline Reload Policy defines when to request a new timeline
- You can manually reload the timeline from your main app

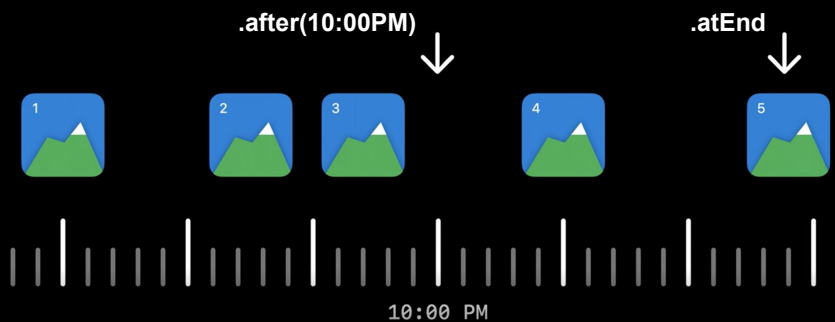


## Every widget has a timeline

- A timeline is a series of entries, and each entry has a view and when to display it
- Basically, these views are serialized and sent to the home screen, where they are just-in-time rendered
- Timelines typically don't stretch on forever, so each timeline has a reload policy, which specifies when to reload the timeline
- You can also manually reload the widget timeline from your main app
  - This will wake up your widget extension and generate a new timeline

# Timeline Reload Policies

- There are 3 different Timeline Reload Policies
- `.atEnd` requests a new timeline after the last timeline entry
- `.after(Date)` requests a new timeline after the provided Date
- `.never` will not request a new timeline



There are 3 different timeline reload policies

- `.atEnd` will reload the timeline after the last entry is displayed
- `.after(Date)` will reload after the provided date
- `.never` will not reload the timeline
  - This means that your main app will manually have to reload the timeline to generate new entries

# Widget Snapshots

- Used when the system needs to quickly display an entry
- Often the same view as the first timeline entry



<https://developer.apple.com/videos/play/wwdc2020/10028>

Every widget also has a snapshot

- You can think of this as a single entry from a timeline
- This is used when the system needs to quickly display a widget entry
- Primarily this happens when a user opens the widget gallery to add your widget to the home screen
- Normally you want your snapshot to be the same as the first entry in your timeline
  - So when the user adds your widget to the home screen, and the timeline takes over, it's a continuous experience

# Widget Placeholders

- The default content for your widget
- Don't display actual user data
- Typically only retrieved when the device environment changes



<https://developer.apple.com/videos/play/wwdc2020/10028>

Finally, widgets have placeholders

- This is the default content for your widget
- It displays what your widget looks like, but without any actual user info
  - You shouldn't provide any real user data to this view
  - It will be redacted anyway
- Apple doesn't really say when this will be called, but it's typically requested in response to environment changes
  - For instance, when the user changes from light to dark mode

# Thank you!

[connor.nelson@capitalone.com](mailto:connor.nelson@capitalone.com)

## References (Apple Content)

- <https://developer.apple.com/documentation/widgetkit/creating-a-widget-extension>
- <https://developer.apple.com/documentation/widgetkit/keeping-a-widget-up-to-date>
- <https://developer.apple.com/documentation/widgetkit/making-a-configurable-widget>
- <https://developer.apple.com/design/human-interface-guidelines/ios/system-capabilities/widgets>
- <https://developer.apple.com/documentation/widgetkit/widgetcenter>
- <https://developer.apple.com/tutorials/swiftui/composing-complex-interfaces>
- <https://developer.apple.com/videos/play/wwdc2020/10194> - Add configuration and intelligence to your widgets
- <https://developer.apple.com/videos/play/wwdc2020/10034> - Widgets Code-Along, part 1
- <https://developer.apple.com/videos/play/wwdc2020/10035> - Widgets Code-Along, part 2
- <https://developer.apple.com/videos/play/wwdc2020/10036> - Widgets Code-Along, part 3
- <https://developer.apple.com/videos/play/wwdc2020/10087/> - Design for intelligence: Make friends with "The System"
- <https://developer.apple.com/videos/play/wwdc2020/10028> - Meet WidgetKit

## References (Apple Content, continued)

- [https://developer.apple.com/documentation/sirikit/adding\\_user\\_interactivity\\_with\\_siri\\_shortcuts\\_and\\_the\\_shortcuts\\_app](https://developer.apple.com/documentation/sirikit/adding_user_interactivity_with_siri_shortcuts_and_the_shortcuts_app)
- <https://developer.apple.com/documentation/widgetkit/timelineprovider>



## References (Other Articles)

- <https://www.simpleswiftguide.com/how-to-expand-swiftui-views-to-span-across-entire-width-or-height-of-screen/>
- <https://www.hackingwithswift.com/quick-start/swiftui/how-to-present-a-new-view-using-sheets>
- <https://www.raywenderlich.com/11303363-getting-started-with-widgets#toc-anchor-001>
- <https://medium.com/better-programming/deep-links-universal-links-and-the-swiftui-app-life-cycle-e98e38bcef6e>
- <https://www.donnywals.com/handling-deeplinks-in-ios-14-with-onopenurl/>
- [https://medium.com/@Tho\\_Stark/better-comparing-dates-in-swift-using-calendar-67d2499aab49](https://medium.com/@Tho_Stark/better-comparing-dates-in-swift-using-calendar-67d2499aab49)
- <https://www.atomicbird.com/blog/sharing-with-app-extensions/>
- <https://medium.com/@abedalkareemomreih/what-is-main-in-swift-bc79fbee741c#:~:text=The%20new%20%40main%20used%20on.including%20the%20platform%20specific%20code.>
- <https://swiftwithmajid.com/2020/08/19/managing-app-in-swiftui/>
- <https://github.com/ArthurGuibert/SwiftUI-Particles>
- <https://kahoot.com/tech-blog/how-we-built-the-ios-14-widget-for-kahoot/>

## References (iOS 14 Home Screens)

- <https://twitter.com/lzldjem/status/1307907062421008385> - mean girls home screen
- <https://twitter.com/wholelottajenni/status/1307868397787521027> - ps1 home screen
- <https://twitter.com/okpng/status/1306850786085761025> - animal crossing home screen
- <https://twitter.com/monokecenter/status/1307106017331863555> - studio ghibli home screen
- <https://twitter.com/hyakukisses/status/1307132730354872320> - sailor moon home screen
- <https://twitter.com/edgygukkie/status/1307112029585629187> - kirby home screen
- <https://twitter.com/shampagnepaki/status/1307995244907188224> - sal home screen
- <https://mashable.com/article/ios-14-iphone-home-screen-ideas/>