# WATCHOS BY TUTORIALS pdf

## 1: watchOS 4 Tutorial Part 1: Getting Started | www.enganchecubano.com

*watchOS by Tutorials is a whopping 27 chapters and www.enganchecubano.com's take a quick look at what's inside.*

Read on to see how you can get a copy! What is watchOS by Tutorials? Learn how to send data and control instructions directly from the Watch to a BLE device! Chapter 1, Hello, Apple Watch!: Chapter 2, Designing Great Watch Apps: Apple has repeatedly emphasized glanceable, actionable, and responsive as the design goal of watchOS apps. From icon design to the new interactivity APIs, make your apps stand out from the rest. This chapter will teach you everything you need to know about this unique aspect of watch apps. Chapter 4, UI Controls: Springs and Struts then? Tables are the staple ingredient of almost any watchOS app. Learn how to set them up, how to populate them with data, and just how much they differ from UITableView. Chapter 9, Digital Crown and Gesture Recognition: Explore the rich set of physical interactions with the Watch, including the Digital Crown, pan gestures, and force touch! Chapter 10, Snapshot API: Glances are out, snapshots are in. Learn how to make your app appear in the new Dock â€" and update the icon dynamically! The way you animate your interfaces has changed in watchOS. This chapter will walk you through the process of setting up your first complication, along with introducing each of the complication families and their corresponding layout templates. Chapter 16, Watch Connectivity: With the introduction of native apps, the way the watch app and companion iOS app share data has fundamentally changed. Out are App Groups, and in is the Watch Connectivity framework. Chapter 17, Audio Recording: You can now record audio directly on the Apple Watch inline in your apps, without relying on the old-style system form sheets. Build a simple game that you can control with just your wrist â€" using SpriteKit and SceneKit. Chapter 19, Advanced Watch Connectivity: In earlier chapters, you learned how to set up a Watch Connectivity session and update the application context. Chapter 20, Advanced Complications: Now that you know how to create a basic complication, this chapter will walk you through adding Time Travel support, as well giving you the lowdown on how to efficiently update the data presented by your complication. Chapter 21, Handoff Video Playback: Want to allow your watch app users to begin a task on their watch and then continue it on their iPhone? Sure you do, and this chapter will show exactly how to do that through the use of Handoff. Chapter 22, Core Motion: This chapter will walk you through incorporating HealthKit into your watch app, from managing authorization to recording a workout session. Chapter 24, Core Location: With watchOS, developers have exactly that via the Core Location framework. Learn everything you need to know about using the framework on the watch in this chapter. Chapter 25, Core Bluetooth: Learn how to send control instructions and other data directly over Bluetooth. Learn how to expand your reach and grow a truly international audience by localizing your watch app using the tools and APIs provided by Apple. You want as many people as possible to enjoy your watch app, right? Learn all about the assistive technologies available in watchOS, such as VoiceOver and Dynamic Type, so you can make your app just as enjoyable for those with disabilities as it is for those without. One thing you can count on: About the Authors Of course, our book would be nothing without our team of experienced and dedicated authors: Ehab Amer is a software developer in Cairo, Egypt. In the day, he leads mobile development teams create cool apps, In his spare time he spends dozens of hours improving his imagination and finger reflexes playing computer gamesâ€¦ or at the gym! In his day job, he is a software developer at Capital One. When not writing software, he spends time rowing on the Potomac river or exploring new restaurants and cooking great food. Soheil Azarpour is an engineer, developer, author, creator, husband and father. He enjoys bicycling, boating and playing the piano. Matthew Morey is an engineer, author, hacker, creator and tinkerer. When not developing apps he enjoys traveling, snowboarding, and surfing. He blogs about technology and business at matthewmorey. Ben Morrow delights in discovering the unspoken nature of the world. He produces beauty by drawing out the raw wisdom that exists within each of us. Audrey Tam retired in from a 25yr career as a computer science academic. Her teaching included many programming languages, as well as UI design and evaluation. Audrey now teaches iOS app development to non-programmers. Jack Wu has built dozens of iOS apps and enjoys it very much. Outside of work, Jack enjoys coding on the beach, coding by the pool, and sometimes just having a quick code in the park. Now

Available in ePub! And as another exciting announcement, by popular request, watchOS by Tutorials is now available in ePub format. Take it on the go with you on your iPad, iPhone or other digital reader and enjoy all the mobile reading benefits that ePub has to offer! Where To Go From Here? To enter, simply retweet this post using the ios11launchparty hashtag by using the button below:

## 2: Tutorial : How To Master Apple Watch Series 3 | Apple Watch 3 Manual

*The Tutorial Team is a group of app developers and authors who write tutorials at the popular website www.enganchecubano.com We take pride in making sure each tutorial we write holds to the highest standards of quality.*

As you can see in the image above. Contact me by Email: Every single chapter in this book has been updated to Swift 4 to ensure it works flawlessly with Xcode 9 and watchOS 4. Chapter 17, Recording Audio: This replaces the "Playing Audio and Video" chapter from the previous edition. Chapter 21, Handoff Video Playback: Chapter 25, Core Bluetooth: Learn how to send data and control instructions directly from the Watch to a BLE device! This chapter replaces the Haptic Feedback" chapter from the previous edition. Introduction In the old days you always had to install iTunes on your PC to update or sync your iPhone. There were neither over-the-air updates nor iCloud backup or synchronizations. Just as you thought those days were gone, Apple came out with Apple Watch â€" the tiniest member of the personal devices family. Despite its coolness and elegance, Apple Watch was clumsy. Except for telling time, you needed a paired iPhone to do literally anything. You were always presented with a loading spinner that would sometime go on for seconds, and the navigation was awkward and you had to drill down few pages on a tiny screen to get to your info. In watchOS 2 we saw a leap forward and Watch apps started running on the Watch itself rather than the phone. We see the direction Apple is going with Apple Watch is to ultimately make the watch independent of your iPhone. In watchOS 3 overall performance of the system is optimized and improved and apps are faster and more responsive. After you leave an app, the OS keeps the app in the background for approximately 8 minutes so it can be relaunched faster. Apple introduced a theme for this: Apps should present purposeful data to user so user can take action. For example, taking your umbrella because the Weather app tells you it will rain is an action, and with rich animations available in watchOS 3 now, thanks to SpriteKit and SceneKit, it is even easier to communicate more complex concepts within 2 seconds. For example, a flight app may show an animated plane taking off to communicate departure. Instead, we suggest a more pragmatic approach â€" pick and choose the chapters that interest you the most, or the chapters you need immediately for your current projects. The chapters are self-contained, so you can go through the book in any order you wish. Looking for some recommendations of important chapters to start with?

## 3: watchOS by Tutorials: Making Apple Watch apps with watchOS 3 and Swift 3, 2nd Edition â€“ ScanLibs

*This is a comprehensive tutorial on watchOS, build on top of your existing iOS skills. If you have doubt about your iOS skills, I would recommend the iOS Apprentice book from the same publisher. Most of the chapters in this watchOS tutorials are independent of one another so you could pick and choose based on your immediate need.*

Here are everything we know about WatchOS 4 features, with guide and tutorial to update your Apple Watch. Finally become available for general download around September  To view and remove profiles: Tap the beta profile that you want to remove, then tap Delete Profile. Enter your iPhone passcode if requested. Tap the beta profile that you want to remove, then tap Delete profile. To continue using the public beta of iOS, you can install the beta profile again. Daily Update The notifications in Activity are now more personalised to you that you receive daily update information from the moment on your watch. Activity and Workout Activity and Workout are almost certainly the most popular apps on the watch. Apple is continuing to tweak and add to these apps. As elsewhere, Apple is gradually increasing the influence of personalisation and machine learning. Apple says that taking a rest at one end of the pool will automatically be detected and be marked as the end of one set of laps and the beginning of another - handy for accurate timings. The accuracy of treadmill-based workouts. This is likely to take a while to filter out into the world outside. This is a demanding and highly effective form of workout that uses your heart rate to push you to your limits. The new Music app on watchOS4 again tries to anticipate all you needs. Now the Apple Watch also supports multiple playlists for the first time. This new feature automatically see music and playlists that you listen to regularly, freshly updated on your watch. The Siri face watchOS 4 displays information using machine learning, to match your routine. You can rotate the Digital Crown to scroll vertically through the various notifications and apps on the face. Apple also revealed two other groups of watch faces. First up is the kaleidoscope face, which you can rotate using the Digital Crown, producing a trippy effect. The visual elements rotate and change shape, just like those weird toys everybody apparently loved in the sixties. You can ask Siri to send money and will recive a notification if your friend accept your money. Woody, Jessie and Buzz Lightyear. These seem to be a bit more advanced, visually speaking, than the previous faces. Mickey and Minnie were animated, but in a fairly limited way. But the new faces incorporate multiple, longer animations that behave differently each time you rotate your wrist. Apple called these vignettes for the character climbing on to the watch face. Update your iPhone to the latest version of iOS. Make sure your Apple Watch is on its charger and has at least a 50 percent charge. Connect your iPhone to Wi-Fi. Keep your Apple Watch on its charger until the update completes. General screen on iPhone Download the update. If asked for your iPhone passcode or Apple Watch passcode, enter it. Wait for the progress wheel to appear on your Apple Watch. It could take from several minutes to an hour for the update to complete. After you delete the file, try to download and install Apple WatchOS 4 by follow this tutorial: Make sure that your Apple Watch connects to its charger. Restart your Apple Watch. Press and hold the side button until you see Power Off, then drag the slider. To turn your watch on again, press and hold the side button. Restart your paired iPhone. Try to start the update again. Leave your Apple Watch on the charger while the update completes.

## 4: How to disable the red dot indicator for unread Apple Watch notifications

*About the e-Book watchOS by Tutorials 2nd Edition pdf Make Apple Watch apps with Swift 3! With the announcement of watchOS 3, Apple is clearly striving to make the Apple watch as independent of your iPhone as possible.*

The original tutorial was written by Mic Pringle. Specifically, you will work on a Watch app for a fictional airline called Air Aber. How to add a watchOS 4 target to an iOS app. How to share data across the two targets. How to add a watchOS 4 interface controller to the Storyboard, and lay out the interface objects. How to create the WKInterfaceController subclass, and wire everything up. Open it in Xcode, and build and run. You should see a blank white screen: In the following screen, set Product Name to Watch, make sure Language is set to Swift, and uncheck any checkboxes that are checked. It really is that easy. This is because the code of a Watch app actually runs as an extension bundled within the Watch app, in much the same way Today extensions on iOS work. When prompted, choose Move to Trash to make sure the file is actually removed from the project: This should leave you with an empty storyboard, or as I prefer to think of it, a blank canvas. Expand the Shared group in the Project navigator, and select Flights. The file is now included in both the AirAber and Watch Extension targets. Repeat the process for the other file in the Shared group, Flight. And with that done, you can finally begin building the flight details interface! With the interface controller selected, open the Attributes inspector, and set Identifier to Flight, and check Is Initial Controller. Uncheck Activity Indicator On Load: Checking Is Initial Controller simply informs WatchKit this is the interface controller you want to display when the Watch app first launches. In order to simplify this tutorial, you will build your layout only for the 42mm watch. At the bottom left of the storyboard pane, ensure that it says View as: Watch app layout is completely different from iOS layout. When you drag an object onto the controller, it slots in under the previous objects, and the screen fills up pretty fast. To organize objects side-by-side, you use groups, which are a lot like stack views in iOS and macOS. So first, drag a group from the Object Library onto the interface controller: With the new group selected, head over to the Attributes inspector, and change Insets to Custom. Four text fields appear, where you can manually set the top, bottom, left and right insets of the group. Change Top to 6: This simply gives the layout group a little extra padding at the top. Next, drag an image into the group. If your group shrank in response to changing the top inset thanks Xcode! Now you need an image to display. This creates a new image set called Logo, with the actual image in the 2x slot: Using the Attributes inspector, make the following changes: Set Width to Fixed, with a value of  Set Height to Fixed, with a value of  The Attributes inspector should now look like the following: Next, drag another group into the existing group, making sure it appears to the right of the image, and use the Attributes inspector to change its Layout to Vertical. Next, drag two labels into the new group. Because you set layout to vertical, the labels appear one above the other: Your interface controller should now look like the following: Next, drag three labels into this new group: Check in the document outline to make sure all three labels are inside the group, not siblings of the group! Next, select the middle label, and change its Text to  Finally, change Height to Fixed, with a value of  Now you only have to add one more group, before you create the outlets, and have this interface display some real data. Next, add two labels to it. Your complete interface object hierarchy should now look like this: Use the Attributes inspector to set Text to Gate 1A for the left label. The completed interface should now look like the following: Click Next, and then Create. Add the following outlets to the top of FlightInterfaceController: Next, add the following property and property observer below the outlets: You declare an optional property of type Flight. This class is declared in Flight. You add a property observer that is triggered whenever the property is set. You only want to proceed with configuring the labels when you know you have a valid instance of Flight. You configure the labels using the relevant properties of flight. If the flight is delayed, you change the text color of the label to red. Now you need to set flight when the controller is first shown. Add the following below the declaration of flight: Next, use your favorite method to connect the outlets according to the list below:

## 5: watchOS by Tutorials Updated for Swift 4 and watchOS 4 | www.enganchecubano.com

*watchOS by Tutorials, Third Edition is now % complete, fully updated for Swift 4, watchOS 4 and Xcode 9 â€" and available today. If you've already bought the watchOS by Tutorials PDF, you can download the new book immediately on the the store page for the book page.*

## 6: Watchos By Tutorials â€“ BookDL

*In this watchOS 4 tutorial for complete beginners, you'll learn how to create the user interface for a fictional airline called Air Aber. In this watchOS 4 Tutorial, you'll build a simple but fully functional watchOS 4 app. Specifically, you will work on a Watch app for a fictional airline.*

## 7: watchOS by Tutorials Updated for Swift 4 and watchOS 4 â€“ Iyar

*Watchos By Tutorials is published by Razeware LLC in December This book has pages in English, ISBN This book has pages in English, ISBN Now you can build apps that interact even more deeply with the rich features and hardware features of watchOS 3.*

## 8: watchOS by Tutorials | www.enganchecubano.com Store

*watchOS by Tutorials Second Editon: Making Apple Watch apps with watchOS 3 and Swift 3 Make Apple Watch apps with Swift 3! With the announcement of watchOS 3, Apple is clearly striving to make the Apple watch as independent of your iPhone as possible.*

## 9: Apple WatchOS 4 Tutorial Started With New Features | Apple Watch 3 Guide

*Hallo und herzlich willkommen auf technikWERK! Hier nun Folge 3 des watchOS Tutorials: Themen: Soll ich die Apple Watch Ã¼ber Nacht ausschalten? - SOS Funktion verwenden - Homescreen anpassen.*

# WATCHOS BY TUTORIALS pdf

*Historical praxis as christopraxis Be a super sleuth with The case of the face at the window Embracing the Bible and truth Cwc engineering guide for wood frame construction Section 10 : Commercial air-conditioning and chilled-water systems. The origin and history of Sufism Adding and subtracting money worksheet Dreampower Tarot Deck Six sigma demystified a self-teaching guide Reliability assessment using stochastic finite element analysis African refugee resettlement in the United States The wedding march sheet music Soil and vegetation systems Disturbing ground Requiem for Yugoslavia Multistate Guide to Regulation and Taxation of Nonprofits (2008) I met a bear piano music An illustrated review of the digestive system The Adventure of the Gold Hunter The Disney Treasures Electronics Design with Off the Shelf Integrated Circuits Golf courses of Colorado In search of Steinbeck Destination transformation Tanzania and the Imf Understanding protein Hadoop in practice 2ed manning Drive to win carroll smith Memoirs of Mrs. Letitia Pilkington, 1712-1750 The complete chronicles of Conan Generator and lead selection Samuel J. Asirvatham, David L. Hayes, Paul A. Friedman Uh huh! Alia Starkweather Of Divine Economy Cilt 1. 1964-1965 Critical observations Richard Scarrys chuckle with Huckle! Rhetoric and communication Microbial Contamination in Parenteral Manufacturing (Drugs and the Pharmaceutical Sciences) Daily problems and weekly puzzlers Fiber industries from northern New England : ethnicity and technological traditions during the woodland p*