Getting Started with ResearchKit

Starting with the basics of the ResearchKit framework, this book walks you through the steps of creating an iOS application that could serve as the basis of a clinical research mobile app.

This book will introduce readers to ResearchKit and explain how to turn your iPhone into a medical diagnostic tool. The book will start off by installing and building the research framework in line with the researcher’s needs. During this, the reader will learn to embed ResearchKit in the application and create a small task.

After this, the book will go a little deeper into creating modules for surveys, and so on. The book will also cover the various aspects of privacy and security with regard to participant data, and how to build dashboards for visualizing medical data and results in line with the researcher’s requirements: data backends, JSON serialization and deserialization, and so on. Readers will be able to fully utilize ResearchKit for medical research, will be able to get more and more patients to participate in their surveys, and will gain insights from the surveys using the dashboards that they have created.

Who this book is written for

If you want to gain practical knowledge for using the Sencha Touch mobile web application framework, and you are familiar with HTML and CSS, then this book is for you. It is assumed that you know how to use touchscreens, touch events, and operating systems such as Apple iOS and Google Android.

What you will learn from this book

- Interact with the hardware of the device including the gyro and the motion sensors
- Get introduced to two backend services: a simple backend server using Sinatra and Sage Bridge
- Build a custom task (a conditional survey example) and a custom active task with recorders
- Get an overview of ResearchKit’s open source repository and App Core
- Interact with the hardware of the device, including the gyro and the motion sensors
- Learn the basics of this revolutionary technology
- Get introduced to the barebones app and learn to write your first code


Edward Cessna
Dhanush Balachandran

Enter the era of medical research using mobile devices with the help of this guide on ResearchKit!
In this package, you will find:

- The author's biography
- A preview chapter from the book, Chapter 1 'Getting Started'
- A synopsis of the book’s content
- More information on Getting Started with ResearchKit
About the Authors

Dhanush Balachandran has a vast experience in creating mobile apps for healthcare industry that include several ResearchKit apps. He was the lead iOS engineer at Jiff, a healthcare start-up and later worked on ResearchKit-based apps for leading institutions. Currently, he is an iOS engineer at DJI. He is also the founder & CEO of mobile app startup, Sortly.

Edward Cessna is the software engineering director for Y Media Labs, a digital agency in Northern California that creates mobile applications. He has three decades of software engineering experience—ranging from embedded software providing cryptographic services to a multitude of iOS applications. He has been working with iOS since its first public release in July 2008. He has managed the development effort for a number of ResearchKit-based applications for leading research institutions.
Preface

This book helps you get started on creating ResearchKit-based applications. ResearchKit™ is an open source framework introduced by Apple that allows researchers and developers to create powerful apps for medical research.

What this book covers

Chapter 1, Getting Started, introduces ResearchKit and explains the anatomy of a ResearchKit-based application.

Chapter 2, ResearchKit Hello World, helps you create your first ResearchKit application and teaches how to integrate ResearchKit in your projects.

Chapter 3, Building Surveys, covers how to create and present clinical surveys. In this process, you'll also learn about the ResearchKit object model.

Chapter 4, ResearchKit Informed Consent, explains how to create, present, and obtain informed consent to participate in clinical studies from the end users.

Chapter 5, Active Tasks, covers how to use active tasks—one of the most important features of ResearchKit.

Chapter 6, Navigable and Custom Tasks, helps you create smart surveys that can skip questions based on the answers provided by users.

Chapter 7, Back End Service, explains how to serialize task results to send to a backend service.

Chapter 8, Where to go from here, helps you learn various tools and tips to create real-world ResearchKit applications.
On March 9, 2015, Apple introduced ResearchKit, a software framework that facilitates the development of health and clinical-based research applications for iOS. Doctors and researchers will be able to collect larger quantities of data frequently and with greater accuracy through the applications built with ResearchKit. The ultimate goal is to increase the research community's knowledge on diseases that could potentially lead to medical breakthroughs in the treatment of the studied diseases.

In conjunction with ResearchKit's announcement, Apple announced and released five ResearchKit-based applications. These applications are as follows:

1. Asthma Health: This is developed by Mount Sinai in order to study asthma triggers and help participants self-manage their asthma. This application makes heavy use of ResearchKit's survey capabilities and custom tasks.

2. mPower: This is developed by the University of Rochester and Sage Bionetworks in order to study the variability in the symptoms of Parkinson's disease. This application utilizes ResearchKit's two-finger tapping, short walk, spatial-span memory, and custom tasks.

3. GlucoSuccess: This is developed by Massachusetts General Hospital to study how diet, physical activity, and medications affect blood glucose levels for participants with type 2 diabetes. This application utilizes ResearchKit's survey capability and custom tasks.

4. Share the Journey: This is developed by the Dana-Farber Cancer Institute, UCLA Fielding School of Public Health, Penn Medicine, and Sage Bionetworks to study the long-term effects of chemotherapy used in the treatment of breast cancer. This application utilizes ResearchKit's survey and custom tasks.

5. MyHeart Counts: This is developed by Stanford Medicine and the University of Oxford in order to study how a participant's lifestyle affects the risk of cardiovascular disease. This application utilizes ResearchKit's fitness-check task and surveys, and custom tasks.
The five initial ResearchKit-based applications shared similar user experience. Using a common application core that's independent of ResearchKit, these applications had a common on-boarding process to enroll new participants in the study, an activity list to present tasks that the researchers wish the participants to carry out, and a dashboard to present the results of the previously carried out tasks. Additionally, these applications used the same backend service to establish accounts, download task schedules and surveys, and upload the collected data in a secure manner.

Apple has open sourced all of the initial ResearchKit applications, and the application core that provides additional services and capabilities beyond these features of ResearchKit.

Links to the source code, documentation, and other information can be found on [www.apple.com/researchkit](http://www.apple.com/researchkit) and [www.researchkit.org](http://www.researchkit.org); the source code is hosted directly on GitHub at [https://github.com/researchkit](https://github.com/researchkit).

The open source applications serve as an example for researchers to undertake the development of their own ResearchKit-based applications. As examples, there are differences between these applications and the ones available from the App Store. In general, copyright material has been removed along with the cryptographic credentials that enable the applications to upload data to the researcher's servers.

### What is ResearchKit?

At its core, ResearchKit orchestrates the administering of tasks and recording of the results from each step of the task. ResearchKit's tasks are segregated into modules: surveys, informed consent, and active tasks. Surveys are questionnaires that prompt the participant to answer to a set of questions for the purpose of recording information that can be used for statistical analysis. Surveys support a variety of question types and answer formats. The informed consent module provides the basic mechanism that is necessary to conduct informed consent visually. Active tasks provide the framework that is necessary to allow applications to develop tasks for the participant. On its announcement, ResearchKit was shipped with a number of predefined tasks, as follows:

1. **Fitness check**: The participant is asked to walk for a specified duration while recording data from various sensors. If the heart-rate data is available at the conclusion of the task, the user is asked to sit down for a duration and the data recording continues.

2. **Short Walk**: The participant is asked to walk a short distance while data from the accelerometer and pedometer data is being recorded.
3. Audio recording: The participant is asked to record the sound that they make.

4. Finger tapping: The participant is asked to tap two targets on the screen as touch activity and accelerometer data is being recorded.

5. Spatial-span memory: The participant is asked to participate in a game-like task that tests their ability to repeat a pattern of increasing length.

Subsequent chapters will go to greater depths on how to use the predefined activities provided by ResearchKit and construct new active tasks.

**Privacy**

In a clinical research application, nothing is more important than the privacy of the user's personal information. Whether this information resides in a device or during transmittal to a data-collection server, personal health information must be safeguarded at each step. Information in the device must be protected irrespective of whether the application operating in the foreground, background, or not executing at all. If the information is transmitted to a data-collection server, it must be safeguarded during transmission. Safeguarding data in transit includes protecting the data during transmission as well as ensuring that the data is going to the correct destination.

Safeguarding of personal health information is the responsibility of the application. ResearchKit provides little capability in this regard other than to ensure classes that may contain user information compile with the NSSecureCoding protocol. The following security and privacy-related issues must be considered by the application developer:

- The *Data Protection* service is the default level of protection for data files stored in the device by the iOS application.
- The *Data Protection* level for directories that may contain the uploaded data in the background.
- The protection of files containing the data that may be uploaded. The data to be uploaded will need additional levels of protection than those offered by the *Data Protection* service.
- The authentication of the endpoint where data will be uploaded.
Identification, authentication, and authorization

Identification, authentication, and authorization are the key concepts in information security. Identification is simply claiming you are somebody. Authentication is proving that you are who you say you are. Authorization is where an authority gives you the permission to carry out a specific task or set of tasks. It is highly likely that these concepts will come into play for a ResearchKit-based application, considering that the application may record, transmit, and display sensitive information about the participant.

ResearchKit does not provide any support for identification, authentication, or authorization; it is up to the application developer to implement the required functionality. Depending on the requirements of an application, the developer may need to implement features to identify and authenticate a participant for both the iOS device and web services (for example, data collection service). Identifying and authenticating to a device is to ensure that the right person is using the application. Identifying and authenticating to a web service is to ensure that the data is only collected from consented individuals.

For devices with Touch ID, Apple's fingerprint recognition feature, an application developer may use this technology to identify and authenticate a participant to the device. Available in all iOS devices, Apple's keychain technology may be used to store log-on credential for any web services. Once a participant has been authenticated to the device, the application may retrieve the web services' credentials from the keychain.

Informed consent

Informed consent is a standard practice for health providers and clinical researchers in providing health care or conducting clinical research; it is not a concept specific to ResearchKit. Obtaining informed consent from participants, prior to enrolling them in a clinical study, is to ensure that the permission has been given and the participant has a clear understanding of the facts, implications, and consequences of their participation in the study. As such, informed consent and its constituents are a vast topic that is beyond the scope of this book.
ResearchKit provides the mechanism that is necessary to present the informed consent document to the participant as an animated sequence of pages as well as the document in its totality. Each page may have more information that presents more details on the page’s topic. ResearchKit comes with a number of predefined sections, as shown in the following:

- Overview
- Data gathering
- Privacy
- Data use
- Time commitment
- Surveys
- Tasks
- Withdrawal

The informed consent module includes the ability to record the participant’s signature if that is required by a study’s informed consent process. This is not a digital signature (that is, verifiable and irrevocable) and ResearchKit has no provisions to collect such signatures. It’s the application developer’s responsibility to provide support for digital signature, if required by the study protocol.

Informed consent is a ResearchKit task. As such, it may be extended in order to comply with the requirements of the study. For example, the consent process for some of the initial ResearchKit applications presents a comprehension quiz that the participants were required to pass in order to be deemed consented.

**Relationship with HealthKit**

Apple promotes HealthKit as a technology that allows iOS applications providing health and fitness services to share data with each other. Effectively, HealthKit is a system-wide, health-specific database with developer services that allow the applications to read and write health data to HealthKit. Given the sensitivity of the data stored in the HealthKit repository, HealthKit will request permission from the user for each requested category of information and whether or not the application is allowed to read or write data of the requested category.
ResearchKit and HealthKit are separate but related technologies. ResearchKit utilizes HealthKit in a variety of ways. ResearchKit tasks may require access to information stored in HealthKit in order to present appropriate feedback to the participant or record such information for statistical analysis. A ResearchKit task may write information to HealthKit (for example, a participant's weight, blood pressure, and so on) after obtaining the participant's permission. Additionally, ResearchKit uses HealthKit to perform unit conversion on the data that is captured from various sensors or read from HealthKit.

Features not provided by ResearchKit
A ResearchKit-based application may need additional features beyond those provided by ResearchKit. The initial set of ResearchKit-based applications provide the following capabilities:

- Backend services: In order to be of any use, the recorded data must be transmitted somewhere for analysis. The initial ResearchKit-based applications used a service from a non-profit organization, Sage Bionetworks.
- User feedback of completed tasks: Appropriate levels of feedback to the user create engaging applications that encourages the user to continue using the application. This increases the likelihood of continued data streams from the participants.
- Data security and privacy: Applications must safeguard a participant's personal information by applying the appropriate level data and communication security.
- Passive data collection: Depending on the nature of the research study, it may be beneficial for the application to collect data in the background without direct participant involvement. For example, using location tracking at a low frequency, an application can obtain relative displacements and use it as a measure of socialization.
- Task scheduling: A study may want tasks performed at different frequencies and quantities.
The anatomy of a ResearchKit-based application

The five initial ResearchKit-based applications share a common software architecture. One based on a layered architecture using common software components in order to provide a similar user experience and application functionality. While ResearchKit offers a wealth of opportunities for different types of research applications, the following figure describes a generic architecture that's common to what the initial ResearchKit-based application employed. Using a layered architecture and common components, a developer may create the basis for a family of applications. Such a framework could be used to address the needs for multiple applications at a reduced cost and increased software quality, as shown in the following diagram:

The figure describes a generic architecture that can form the basis of a ResearchKit-based application. Using a layered approach, the architecture has features described in the following paragraphs.
Getting Started

A central Data Model that services the needs for the upper layers and components. This layer includes the following components:

- A *data upload* component that is responsible to upload data, track the success or failure of the upload attempts, retry failed upload attempts, and clean up the data that has been successfully uploaded.

- A *data download* component that either periodically or on-demand, downloads new task schedules, survey contents, and news about the study that the researchers desire to share with participant.

- A *data archiving* component that packages the data to be uploaded to the studies backend server or other destinations. This component can support one or more formats. A key feature of this component is to ensure the confidentiality of the data prior to uploading it to the data collection service. The five initial ResearchKit-based applications used the Cryptographic Message Syntax (RFC 5652) in order to *wrap* the data for safeguarding.

- One or more *passive data collector* components. With the participant's explicit permission, these components could collect the data from the various sensors on the device and then trigger a data upload. For example, in order to determine whether the participants were socializing or homebound, a number of the initial ResearchKit-based applications collected relative displacement of the device location. Relative displacement allows the researchers to determine socialization, while avoiding collecting sensitive location data.

An on-boarding process will be common for ResearchKit-based applications. During this process, the applications could present background information about the study, collect any required demographic information, and perform the consent process (a ResearchKit-based activity). This layer includes the following components:

- A *consent process* component that performs the informed-consent process. ResearchKit supports this activity and provides many features to shape this feature.

- A *service signup* component, if it is necessary for the participant to provide some kind of login credential or identity with a data collection service.

- For applications that retrieve data from HealthKit, a *HealthKit permissions* component could inform the participant in one place as to which data will be collected and why. This would be an excellent place to allow the participant to opt out of the collection for one or more HealthKit parameters.
Given the potential sensitive nature of ResearchKit-based applications, the identity and authentication of the application user as a study participant should always be determined prior to allowing a user to use the application. As a variety of data, events, time triggers, and so on could be input in to the authentication process, this component may best be implemented as a state machine.

Once a user has been authenticated (gone through the on-boarding process and authenticated), the heart of this theoretical application's user interface is the dashboard. The dashboard layer includes the following components:

- A **to-do activities** component that displays a list of activities that the researchers wish the study participants to accomplish. These activities can include ResearchKit-based tasks (for example, surveys) or custom tasks that do not employ ResearchKit. An example of the latter would be a news component, where the researchers share ongoing information about the study in general.

- A **task results** component, where the application shares the results or summary of the results for completed tasks. If task-aggregated results are available (either baked in the application or downloaded via the data download complement), the application could inform the participant how they are performing with respect to the study's population norm.

- A **preference** component that allows the participant to customize the application to serve their need. This would be an excellent place to allow the participant to opt out or opt in to the collection of specific data parameters. Additionally, this component could serve as the vehicle to allow the participants to withdraw from the study.

The top component in this layered architecture is the **user interface** component. This component provides the heart and soul from a user’s experience point of view. This component will also address any branding requirements levied by the research institution.

**Summary**

In this chapter, you were introduced to ResearchKit framework and the different aspects of a ResearchKit-based application.

In the next chapter, we will write a simple Hello World ResearchKit-based application.
Where to buy this book
You can buy Getting Started with ResearchKit from the Packt Publishing website.
Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.
Click here for ordering and shipping details.