Kenkou Integration Documentation

KenkouSDK 1.0

By Alexander Gorny/Ondej Vancak/Egor Gaydamak/Kristina Goryacheva/Jian Zhao

Version of this document: 02/10/2020

For iOS

Documentation

Version Support

- **KenkouSDK iOS 1.x.x**
  Supports **Swift 5**
  Works with **iOS**11

Language

KenkouSDK for iOS is written in Swift in combination with Kotlin. All public API is written in Swift.

Platform

We generally support n-2 versions of Apple's iOS Operating System.iOS is a fast moving target, and Apple are aggressive in migrating devices to the latest available version.

Devices

KenkouSDK for iOS supports all available iOS devices, and is optimised to support all form factors. This includes all form factors of iPhone and iPad Pro.

Our minimum supported version of iOS is currently iOS11.This means the baseline devices we support are the iPhone 5s.

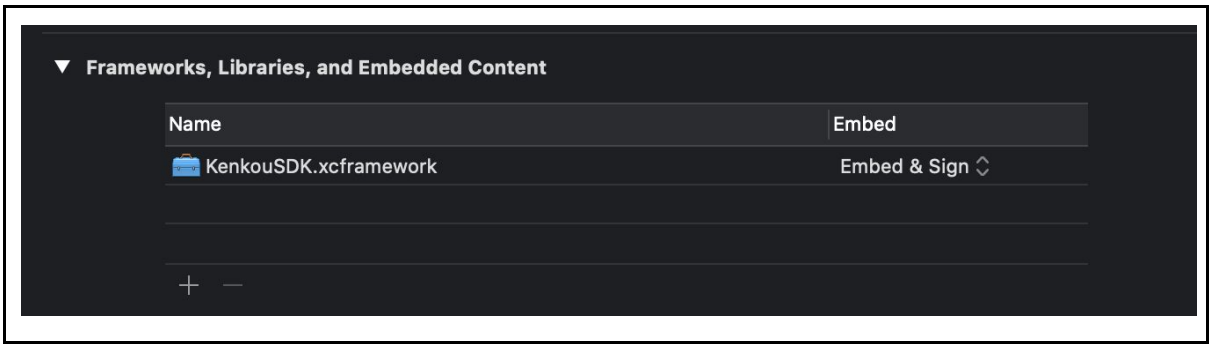
Getting Started

Integrating KenkouSDK Manually

You can manually integrate the KenkouSDK framework into your project

Integrating the Dynamic Framework

Drag `KenkouSDK.xcframework` into the `Frameworks, Libraries and Embedded Content` section of your target.



Basics

Once you've integrated KenkouSDK, there are a few basics to go over before you can fully utilise the features and capabilities.

Client Authentication

We use client credentials to authenticate you as a partner product with Kenkou. These credentials are **mandatory**, and the SDK will not function without providing them.

So before continuing, make sure you have the following:

- Client Identifier
- Client Secret

You can then start using KenkouSDK - we recommend starting in your `AppDelegate`. In addition to the credentials, you will also be required to provide a KenkouSDKDelegate.

```
KenkouSDK.start(withClientIdentifier: "MyClientIdentifier",
                          clientSecret: "MyClientSecret",
                              delegate: self)
```

Configuration

KenkouSDKConfiguration allows you to customize various elements of features and functionality within KenkouSDK, and can be set when starting the SDK. In the example above, .default is being used, and if you choose to not include the configuration parameter this will also be used.

KenkouDelegate

KenkouSDKDelegate contains a number of functions that any client app using KenkouSDK must conform and respond to.

```
extension AppDelegate: KenkouSDKDelegate {

        func userDidFinishedMeasurement(withMeasurementResults:
KenkouSDKMeasurementResults){

        }
    }
```

All Done

KenkouSDK is now set up and ready to go.

Next Steps

With everything set up, you can now move onto *Permissions.*

Permissions

The measurement feature of the Kenkou SDK requires additional permissions, which require you to modify your `Info.plist` with permission keys. The system will then use the key in the alert presented to a user when the specific subsystem is attempted to be accessed.

Apple enforces these rules statically when uploading to App Store Connect; meaning that even if the code is never executed, the mere reference triggers the analyzer to require the permission keys to be present. This means that in order to integrate Kenkou SDK, you are required to add these permission keys.

Camera permission

We offer the ability to measure stress level which requires access to the camera. This feature has to be explicitly enabled in a KenkouConfiguration and is disabled by default. Your `Info.plist` should have the following entries:

```
<key>NSCameraUsageDescription</key>
```

Measurement Combined Component

Kenkou SDK provides Stress measurement as a combined component including measurement onboarding, measurement, question after measurement and measurement

result. We provide full featured HRV analysis for scientific research and professional use, including:

- Supports wide range of ECG, PPG and RR interval data formats Accurate QRS and pulse wave detection
- Automatic artefact correction algorithm
- Automatic analysis sample generation
- Computes all commonly used time-domain, frequency-domain and nonlinear HRV analysis parameters

Here a screenshot of code will be provided

Data access

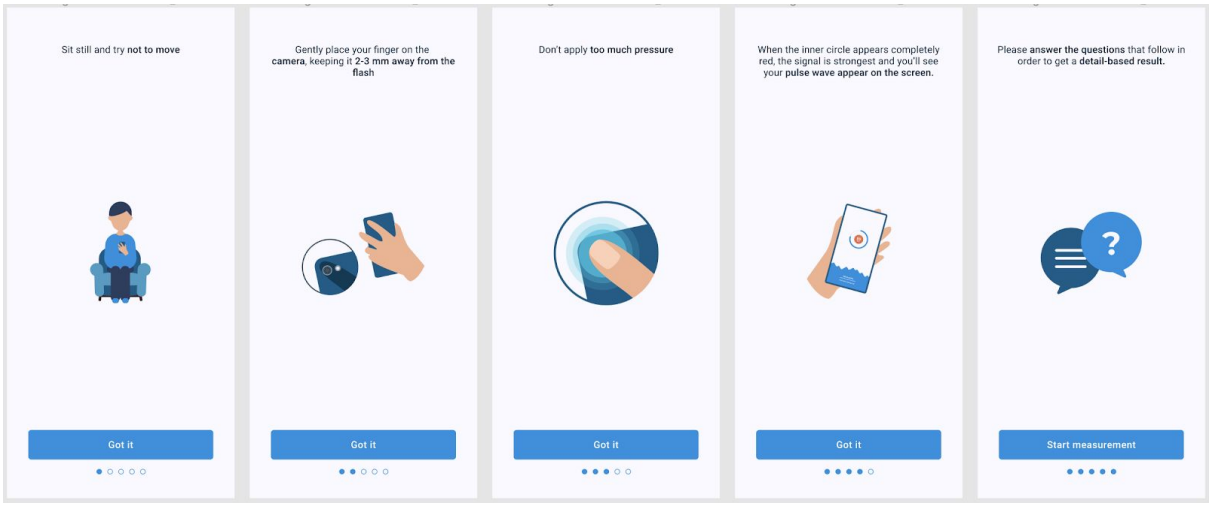
After measurement is finished, detailed stress and cardiovascular indexes would be shown in the measurement result. Current Indexes as well as indexes that we can provide in the future are listed in the reference at the end of this document. These data could be passed onto your app through conforming to KenkouSDKDelegate's userDidFinishedMeasurement method. `KenkouSDKMeasurementResults` is the structure which contains measurement indexes:

```
struct KenkouSDKMeasurementResults {
    public let rr: Double
    public let bpm: Double
    public let rmssd: Double
    public let sdnn: Double
    public let pnn50: Double
    public let stressIndex: Double
    public let normalizedStressIndex: Double
    public let sd1: Double
    public let sd2: Double
    public let pns: Double
    public let relaxationIndex: Double
}
```

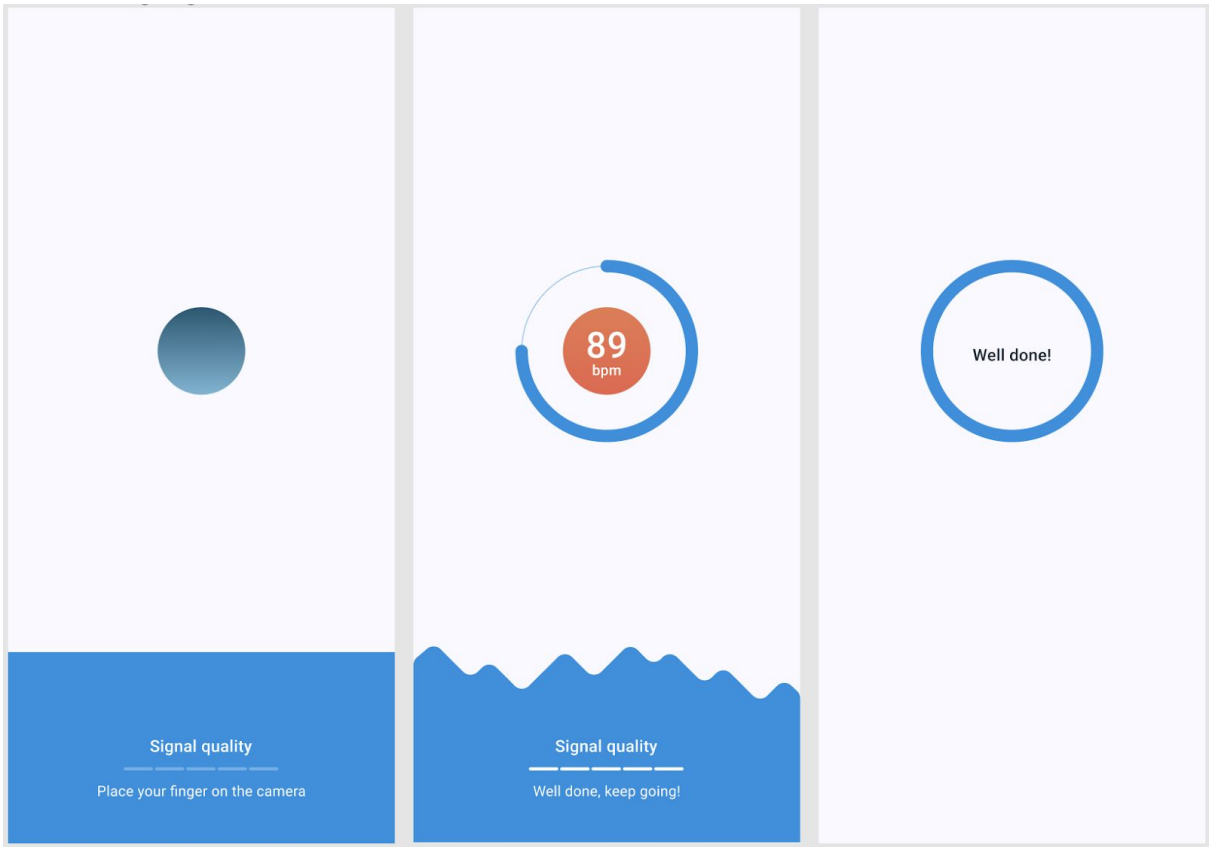
Color/Font

KenkouSDK provided customized color and font options built in every UI component, including measurement onboarding screens, measurement screens, questions after measurement screens and measurement result screens. KenkouColor/Font are designed to provide all the flexibility you need to make Kenkou feel at home in your app.

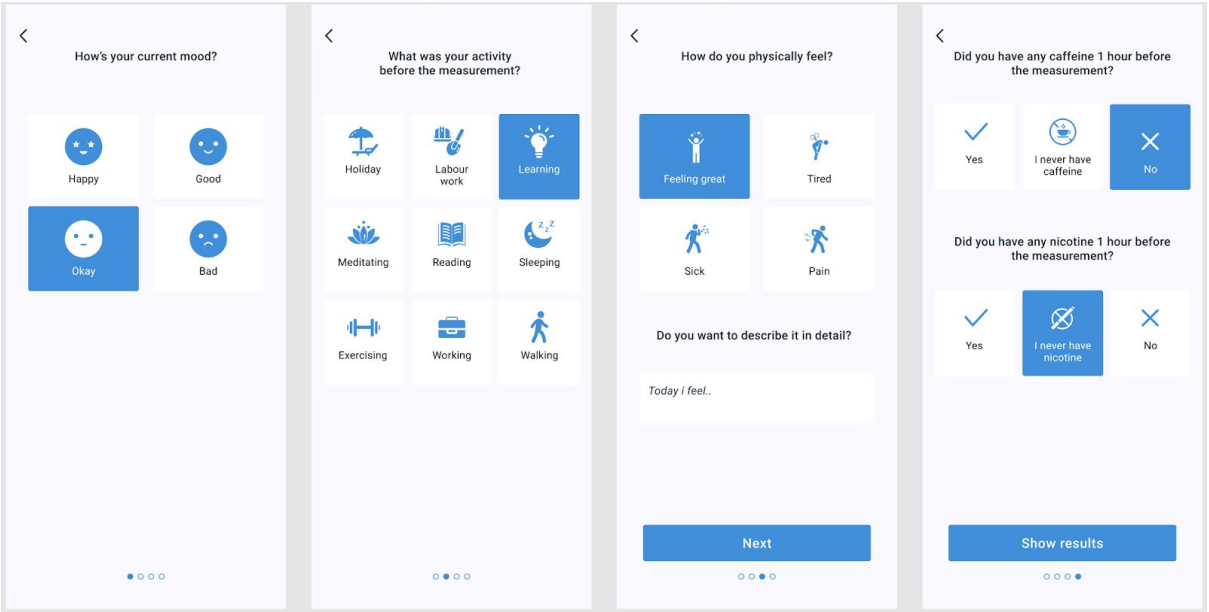
Measurement onboarding screen



Measurement screen



Questions after measurement screen



Contents

- KenkouColorBook
    - ColorPrimary
    - ColorSurface
    - ColorOnSurface
    - ColorBackground
    - ColorOnBackground
    - ColorValue
        - ColorA
        - ColorB
        - ColorC
        - ColorD
    - ColorOnValue
- KenkouFontBook

KenkouColorBook

`KenkouColorBook` provides a list of colors that can be customized to provide custom tinting to your Kenkou experience. The color book will use default colors without configuration, and you are able to override as many or as few as desired.

```
struct KenkouColorBook {
    var colorPrimary: UIColor
    var colorOnPrimary: UIColor
    var colorSurface: UIColor
    var colorOnSurface: UIColor
    var colorBackground: UIColor
    var colorOnBackground: UIColor
    var colorOnValue: UIColor
    var colorA: UIColor
    var colorB: UIColor
    var colorC: UIColor
    var colorD: UIColor
}

//for example you can create default color book and change it in future

var colorBook: KenkouColorBook = .defaultBook
```

ColorPrimary

The primary color that is applied to the button, page slider, pulse wave background, measurement progress indicator, icons tint on questions after measurement screens. The default primary color code is #2688E5.

Examples:

```
colorBook.colorPrimary
```

ColorOnPrimary

ColorOnPrimary that is applied to text on buttons, text on pulse wave, Signal quality indicator, text on selected cards. The default ColorOnPrimary is #FFFFFF.

Examples:

```
colorBook.colorOnPrimary
```

ColorSurface
ColorSurface that is applied to cards background and text field background. The default ColorSurface is #FFFFFF.

Examples:

```
colorBook.colorSurface
```

ColorOnSurface
ColorOnSurface is applied to text on unselected cards and text in the text field. The default secondary color is #121212.

Examples:

```
colorBook.colorOnSurface
```

ColorBackground
ColorBackground that is applied to every screen background. The default primary ColorBackground is #FFFFFF.

Examples:

```
colorBook.colorBackground
```

ColorOnBackground
ColorOnBackground that is applied to texts on the background, navigation elements, chips on the measurement details screen. The default ColorOnBackground is #121212.

Examples:

```
colorBook.colorOnBackground
```

ColorValue

ColorValue refers to the color indicating the level of stress index/recovery ability/heart rate and mood on the measurement result screen. Four ColorValue are set in the current version (ColorA/ColorB/ColorC/ColorD). You can change these by overriding them in your resources.


```
colorBook.colorValue
```

ColorA is applied to measurement values showing “low stress index/excellent recovery ability/HR normal/mood happy; The default ColorA is #31BFDE.

```
colorBook.colorA
```

ColorB is applied to measurement values showing medium stress index/good recovery ability/HR medium/mood good; The default ColorB is #36CB83.

```
colorBook.colorB
```

ColorC is applied to measurement values showing high stress index/okay recovery ability/HR high/mood okay; The default ColorC is #FFAE10.

```
colorBook.colorC
```

ColorD is applied to measurement values showing very high stress index/poor recovery ability/HR very high/mood bad; The default ColorD is #FF6C4C.

```
colorBook.colorD
```

ColorOnValue

ColorOnValue is applied to texts on measurement result values. The default ColorOnValue is #FFFFFF.

```
colorBook.colorOnValue
```

KenkouFontBook

`KenkouFontBook` provides a collection of fonts that can be customized to provide text styles to your Kenkou experience. The font book will default to system font. The fonts we recommend to use include Lato, Roboto, Open Sans, SF pro and Helvetica Neue.

Examples:

Here a screenshot of code will be provided

Reference

Parameter	unit	Definition	comment
Cardio			
HR	bpm	Heart rate	
HRmean	bpm	Average heart rate	
HRmax	bpm	Maximum heart rate	
HRmin	bpm	Minimum heart rate	
HRV			
RR	ms	weighted average of RR intervals	
RMSSD	ms	Square root of the squared mean of the sum of all differences in successive RR intervals	RMSSD expresses how much the heart rate changes from one heartbeat to the next. Indicator of parasympathetic activity. Error-prone with artifacts and arrhythmias.
SDNN	ms	Standard deviation of all RR intervals of a measurement (total variability)	The SDNN is the "gold standard" for medical stratification of cardiac risk when recorded over a 24h period, it is more accurate when calculated over 24h than during shorter periods monitored.
PNN50	ms	Percentage of consecutive RR intervals that differ from each other by more than 50ms.	Indicator of parasympathetic activity.
stress index	ms^-1	Baevsky’s stress index	Mathematical description of the histogram (see also BR. M. Baevsky. Methodical recommendations use kardivar system for determination of the stress level and estimation of the body adaptability standards of measurements and physiological interpretation. 2009)
normalized	0-100	normalized version of stress index	normalized version of previous quantity ranging from 0 to 100

stress index			
SD1	ms	Standard deviation of the orthogonal distances of the RRi / RRi + 1 points to the transverse diameter of the ellipse	Width of the point cloud; more sensitive to rapid, higher frequency changes in heart rate.
SD2	ms	the standard deviation along the line-of-identity in the Poincaré plot	Length of the point cloud; quantifies the long-term HRV.
PNS	0-100	Parasympathetic Nervous System index	Parasympathetic nervous system activity compared to normal resting values
histogram		aggregated distribution of RR intervals in 50 ms bins	normalised RR interval histogram with bin width 50 msec