What if I told you I that my goal is to build an app with a framework that Apple won’t let me show you live?

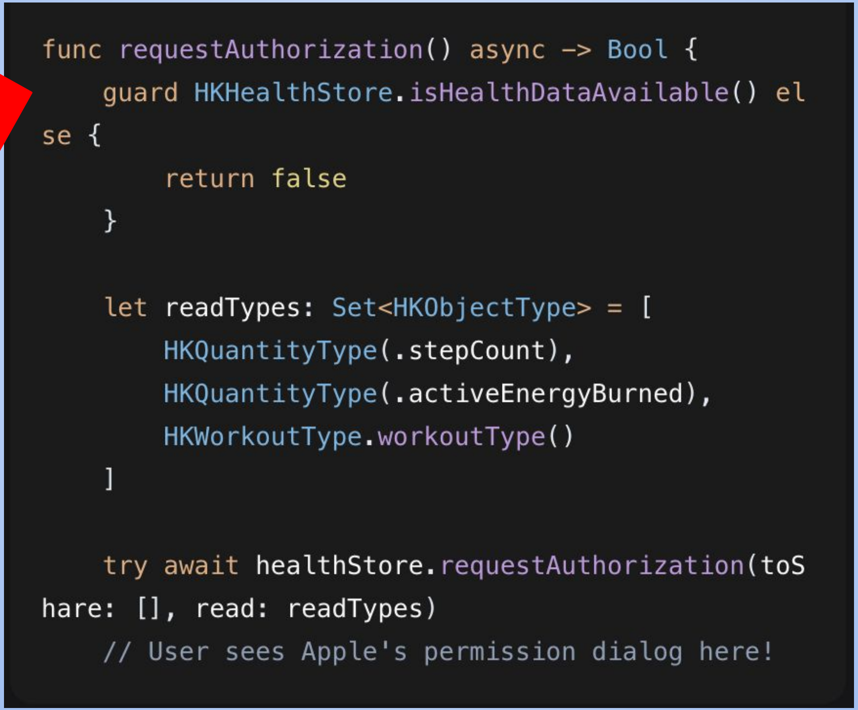
HealthKit

Apple’s creation for keeping users data safe

Health data is extremely sensitive, so Apple designed HealthKit with strict privacy rules. Apps can only access data the user explicitly allows.

I chose HealthKit as one of my frameworks because I want to give the users the ability to track their health and wellness data in a meaningful way. I want users to have access to their personal statistics while ensuring their data stays securely stored on their own phones and accessible only to them.

Everything starts with the users permission



```
func requestAuthorization() async -> Bool {
    guard HKHealthStore.isHealthDataAvailable() el
se {
        return false
    }

    let readTypes: Set<HKObjectType> = [
        HKQuantityType(.stepCount),
        HKQuantityType(.activeEnergyBurned),
        HKWorkoutType.workoutType()
    ]

    try await healthStore.requestAuthorization(toS
hare: [], read: readTypes)
    // User sees Apple's permission dialog here!
```

Health Metrics HealthKit Collects

❤️ Vital Signs

- Heart rate
- Resting heart rate
- Heart rate variability
- Blood pressure
- Respiratory rate
- Blood oxygen
- Body temperature

🧍 Activity & Fitness

- Step count
- Walking/running distance
- Flights climbed
- Active energy burned
- Exercise minutes
- VO$_2$ Max (cardio fitness)
- Workout types

😴 Sleep & Recovery

- Sleep duration
- Sleep stages (REM, core, deep)
- Time in bed

Health Metrics HealthKit Collects

⚖️ Body Measurements

- Height
- Weight
- Body fat percentage
- BMI
- Lean body mass

🩺 Medical & Health Records

(If user connects providers)

- Lab results
- Medications
- Allergies
- Immunizations
- Conditions

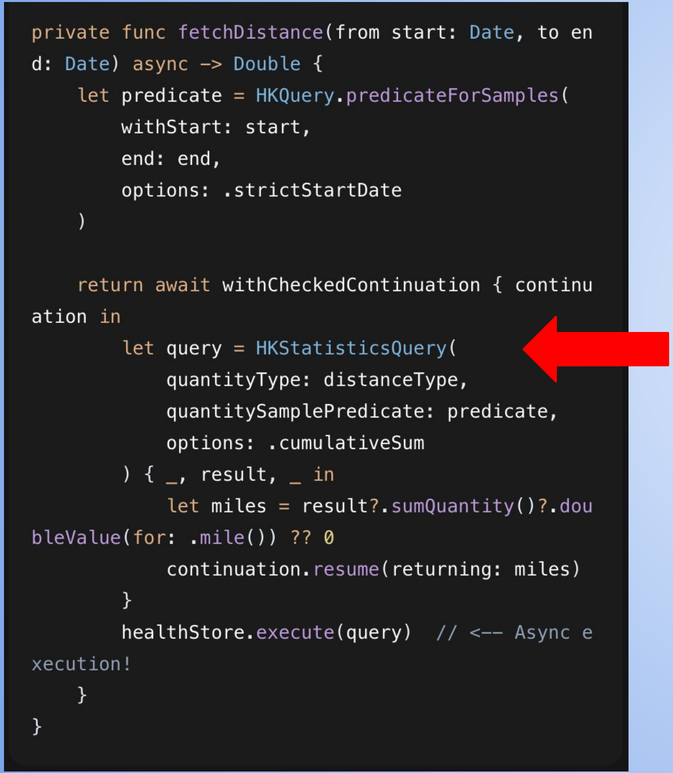
🍎 Nutrition

- Calories consumed
- Water intake
- Vitamins & minerals
- Macronutrients (protein, carbs, fat)

So HealthKit is basically a central health data hub where many apps and devices can store and read information (with permission of course).

Beyond just reading current data, HealthKit offers permanent, encrypted storage of users’ history. Imagine being able to look back at your fitness journey over the past year, seeing every milestone all the while knowing that your history is encrypted and stored on your own device, and not some external corporate server.

Timeless Queries, From Today to History



```
private func fetchDistance(from start: Date, to en
d: Date) async -> Double {
    let predicate = HKQuery.predicateForSamples(
        withStart: start,
        end: end,
        options: .strictStartDate
    )

    return await withCheckedContinuation { continu
ation in
        let query = HKStatisticsQuery(
            quantityType: distanceType,
            quantitySamplePredicate: predicate,
            options: .cumulativeSum
        ) { _, result, _ in
            let miles = result?.sumQuantity()?.dou
bleValue(for: .mile()) ?? 0
            continuation.resume(returning: miles)
        }
        healthStore.execute(query)  // <-- Async e
xecution!
    }
}
```

Think about what this means for users: No login to forget or password to rest.
Their fitness journey becomes a personal diary on their device, not a profile in some company's marketing database. And if they ever want to delete my app their health history remains intact.

As a developer here is how that long-term storage works in code. The same HKStatisticsQuery can fetch data from any date range - last week, last month, last yearn - using the same secure, permission-based system.

Now you’ve seen why: Health data deserves more than a simulator. It deserves architecture built for real devices and real trust.

Even though I couldn’t show you live data, I hope it’s clear how simple the code is, and how vast the possibilities are when you build your app with HealthKit.