第27回 日本医療情報学会春季学術大会

チュートリアル6
二次利用ユーザーのためのデータ抽出、変換、登録（ETL）入門
－ETLの理解がデータ活用の近道－

PHRデータ利活用事始め

（Apple Healthcareからのデータ取得と医療情報との連携）

内村　祐之（東京医科歯科大学）

本日の内容

1. 利用可能なPHRデータについて
2. Apple Healthcareについて
3. Apple Healthcareからデータを取得するiPhoneアプリについて
4. iPhoneアプリから別のサーバへのPHRデータ送信について
5. サーバでのPHRデータ処理について
6. ヘルスケア情報と医療情報との連携について

利用可能なPHRデータについて

- iPhone「ヘルスケア」 - Apple Watchと連携
- Androidアプリ「ヘルスコネクト」- Fitbitと連携
- quick 非接触型体温計 → Apple ヘルスケアApp
- omron 血圧計、体重体組成計、活動量計、体温計、パルスオキシメータ、→ Apple ヘルスケアApp

- 医療情報との連携も始まっています
  - 札幌医科大学と富士通、ヘルスケア領域のデータポータビリティ実現に向けて、個人の健康データの活用推進に合意
    - https://pr.fujitsu.com/jp/news/2023/01/16.html

- 他にもたくさんあります

Apple Healthcare について

- Appleのヘルスケア機能はすべて、あなたのデータを安全に保護し、プライバシーを守るように作られています。デバイス上のあなたの健康データは暗号化され、自分のパスコード、Touch ID、Face IDのいずれかを使わない限りアクセスできません。iCloudと同期されている健康データは、iCloudとの通信中も保存時も暗号化されます。（Appleサイトより）

自分の健康情報は自分の端末からのみアクセス可能。
サードパーティアプリからアクセスする際には項目毎に許可設定が必要。

【Apple Healthcareについて】

iPhoneの健康データは公開されている？

【Apple Healthcareについて】

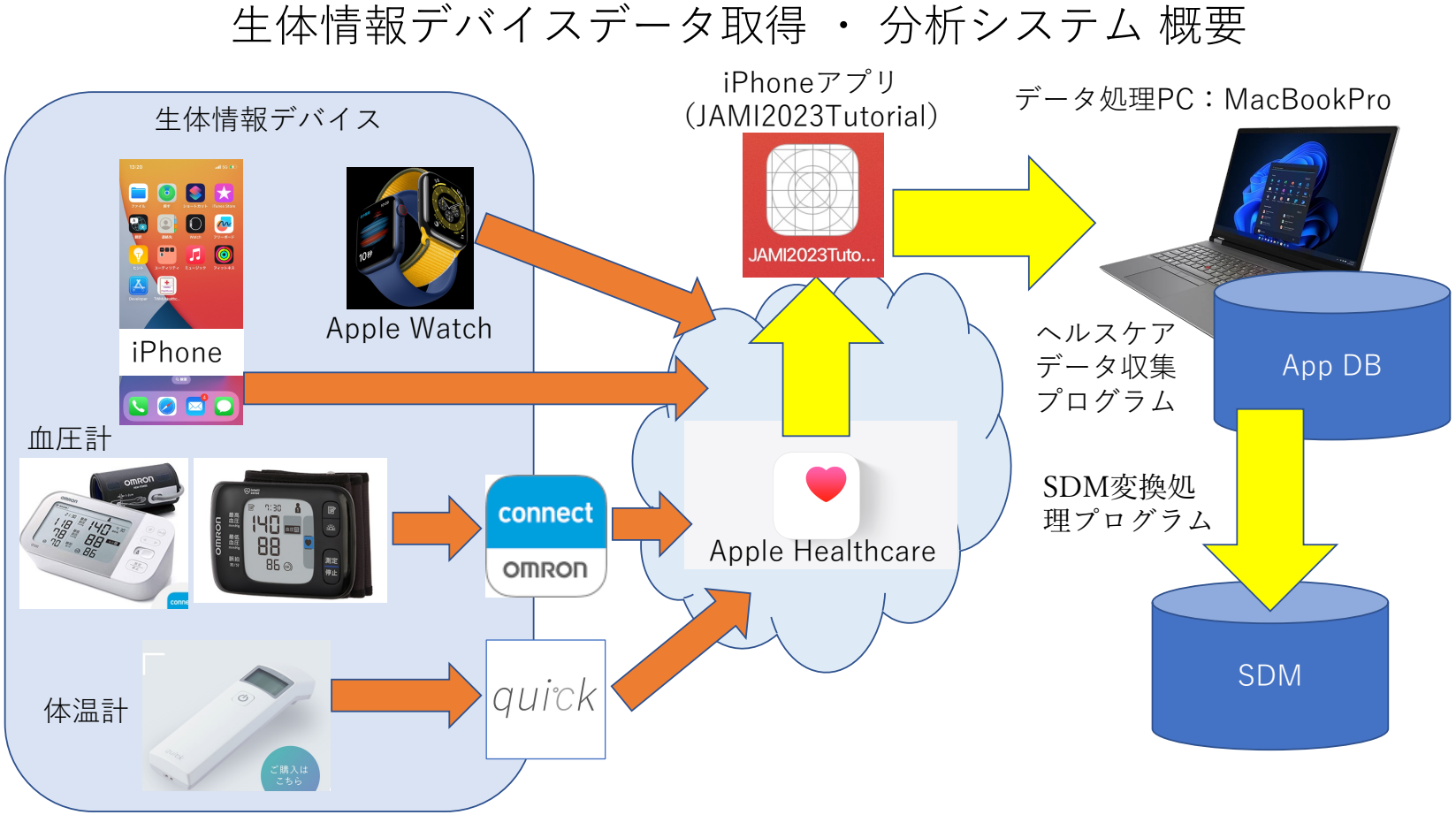
iPhoneのヘルスケア

【Apple Healthcareについて】

Apple Watch

	高心拍数と低心拍数の通知	不規則な心拍の通知	心電図アプリケーション	心肺機能レベルの低下の通知	体に取り込まれた酸素のレベル	転倒検出
センサー	光学式心拍センサー	光学式心拍センサー	電気心拍センサー	光学式心拍センサーと位置情報	光学式心拍センサー	加速度センサーとジャイロスコープ
Apple Watch Series 3	✓	✓	✗	✓	✗	✗
Apple Watch SE	✓	✓	✗	✓	✗	✓
Apple Watch Series 4、5	✓	✓	✓	✓	✗	✓
Apple Watch Series 6以降	✓	✓	✓	✓	✓	✓
Apple Watch Ultra	✓	✓	✓	✓	✓	✓

今回のチュートリアル説明用システム



【iPhoneアプリについて】

アプリ開発に必要なもの

- **iPhoneの実機**

  （シミュレータでは制限があります）

- **開発用PC**

  （Windowsでの開発もできなくはないですが、基本的にはMacになります。メモリは16GB以上が必要です。）

- **開発環境**

  （標準的なものはXcodeになります）

- **実機とPCを繋げるケーブル**

  （iPhoneに付属しているもので大丈夫ですが、別のケーブルを使う場合にはMFi認証Lightningケーブルでないとダメです）

【iPhoneアプリについて】

Xcodeについて

- AppleがiOSアプリ開発のために無償提供しているソフトウェアがXcodeです。
- iOSアプリの構築、実機検証、デバックという、アプリ開発を一元化した統合開発環境を持っており、iOSアプリ開発のためにXcodeは欠かせないソフトウェアとなっています。
- XcodeはApp Storeから入手可能です。

【iPhoneアプリについて】

Apple Developer Program

- Apple Developer ProgramはアプリをApp Storeに公開可能になるApple公式のメンバーシッププログラムです。
- 年額$99の費用がかかります。
- Apple Developer Programはさまざまな開発ツールやアプリの分析、最新テクノロジーが使用可能になります。
- 開発したiOSアプリはApp Storeを通じて販売・ダウンロードされますが、アプリ開発者が作成したiOSアプリをApp Storeで公開するためには、Apple Developer Programに登録を行う必要があります。
- Apple Developer Programを解約すると公開したアプリのダウンロードができなくなってしまいますので、継続してダウンロードしてもらうには登録の継続が必須となります。

【iPhoneアプリについて】

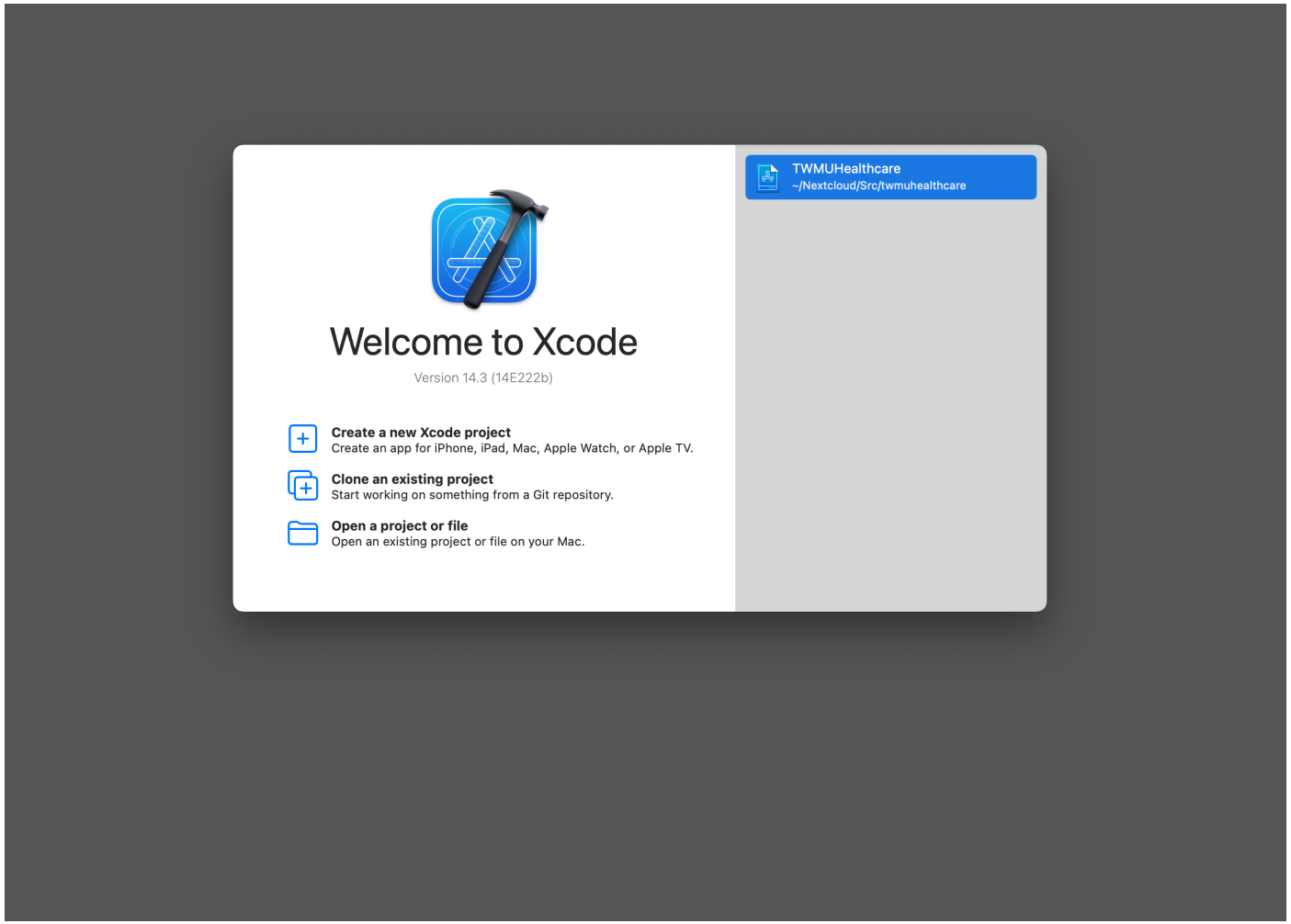
SwiftUI開発について

- iPhoneネイティブアプリを開発するには、MacでXcodeというツールを使って開発を行うのが一般的です。
- XcodeではSwift言語を使って開発を行いますが、Swift開発は従来のStoryboardを使ったSwift開発と2019年から始まったSwiftUIによる開発の２パターンがあります。
- Storyboard開発とは、例えばオブジェクトをドラッグ&ドロップで配置してレイアウトを作っていく開発方法で、SwiftUIは画面のレイアウトを全てコードで実装するというのが特徴です。
- SwiftUIは2019年にリリースされた開発方法なので、ネット上に情報が少なかったり、古い本には書かれていないので注意が必要です。

【iPhoneアプリについて】

アプリ開発の流れ１

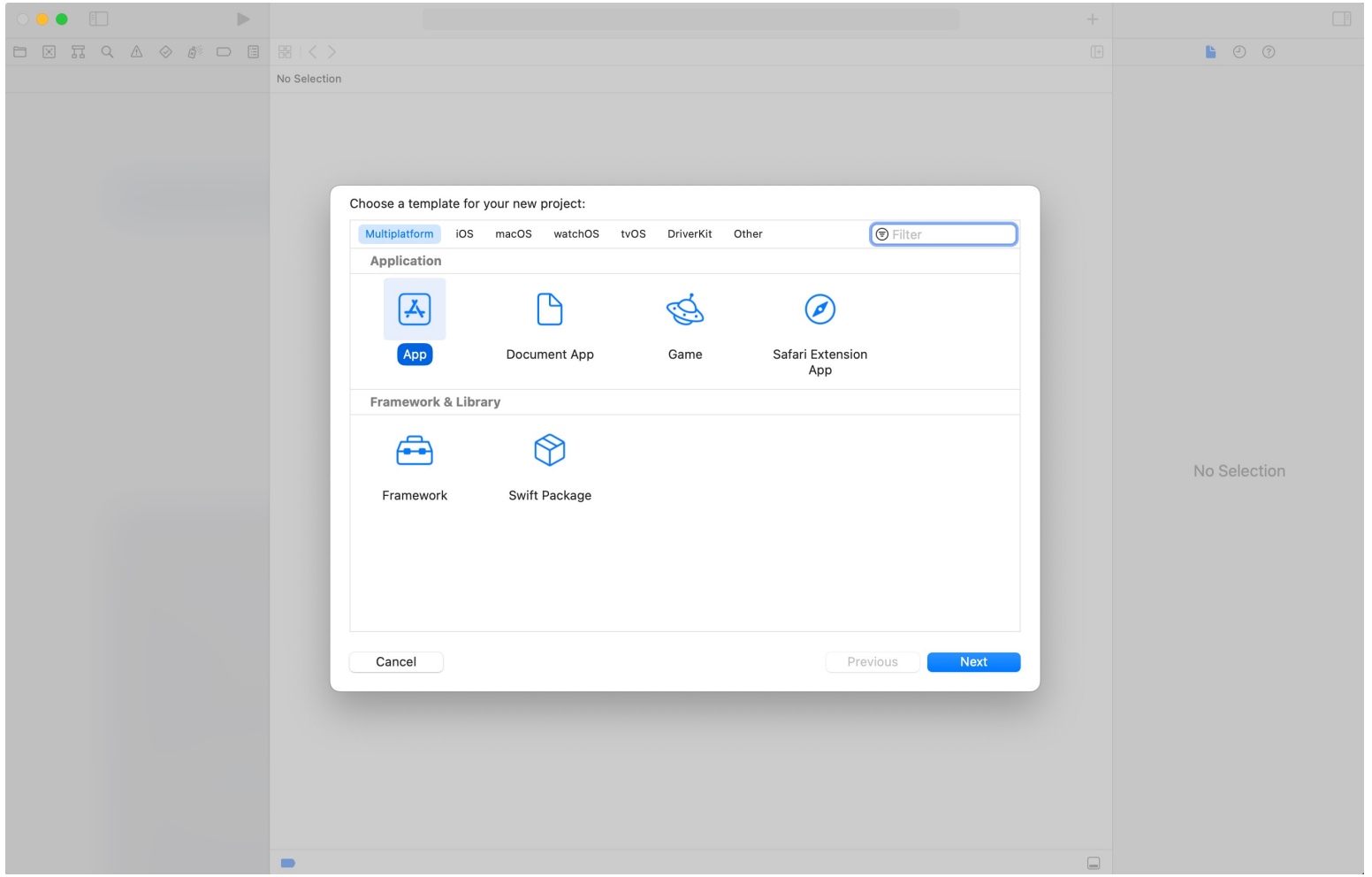
1. Xcodeで新規プロジェクトを作成します



【iPhoneアプリについて】

アプリ開発の流れ２

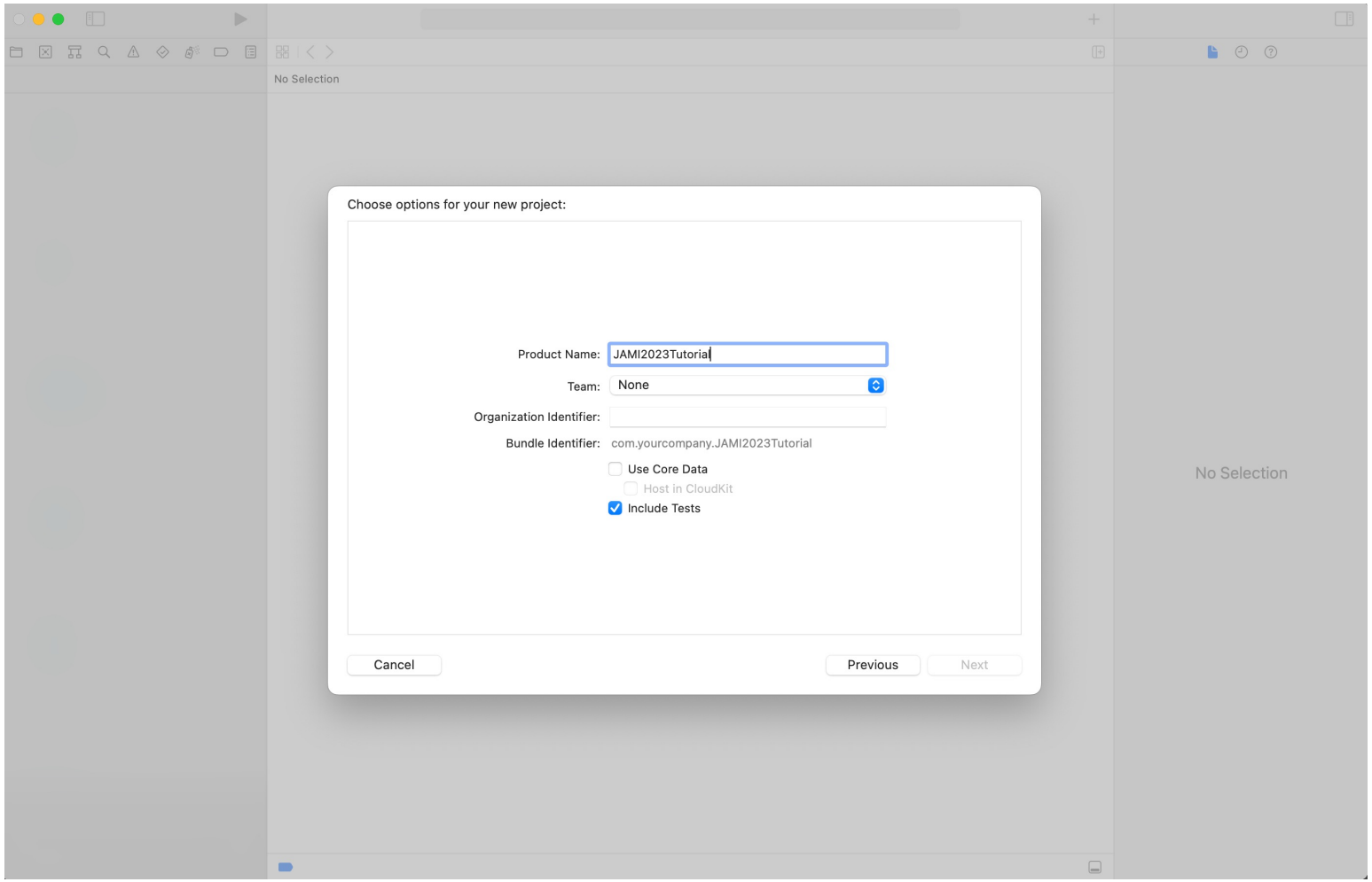
2. アプリケーションから「App」を選択します。



【iPhoneアプリについて】

アプリ開発の流れ３

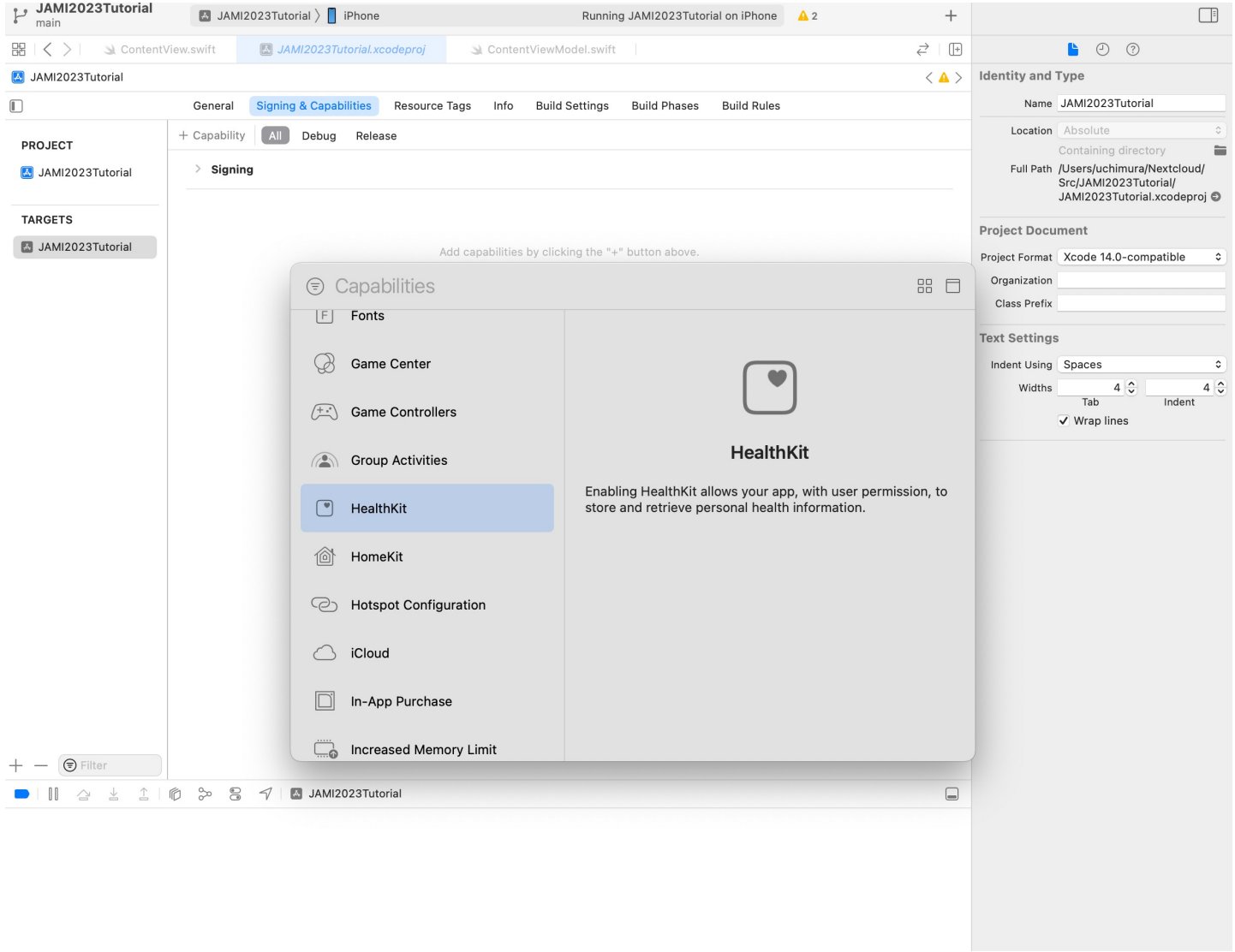
3. プロジェクト名を入力します（JAMI2023Tutorial）



【iPhoneアプリについて】

アプリ開発の流れ５

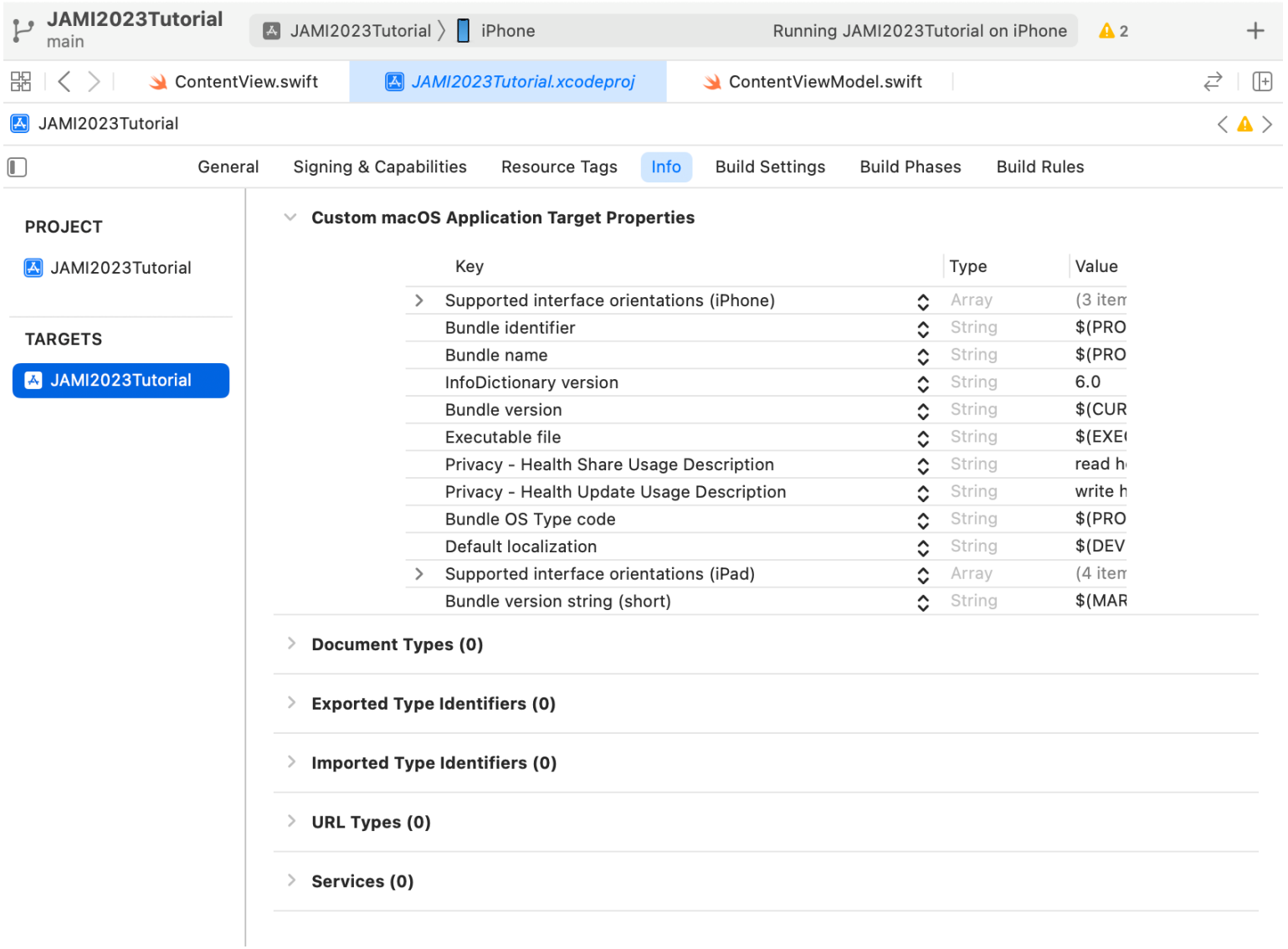
5. Healthkitが使えるようにします①



【iPhoneアプリについて】

アプリ開発の流れ５

5. Healthkitが使えるようにします②



【iPhoneアプリについて】

アプリ開発の流れ６

6. 項目毎にヘルスケアの利用許可を得ます①

HKHealthStoreのrequestAuthorization
メソッドを呼び出してヘルスケアの
利用許可を得る必要があります。
以下は「体温」の例です。

```
let readTypes = Set([
    HKQuantityType.quantityType(forIdentifier:
        HKQuantityTypeIdentifier.bodyTemperature)!
])
```

【iPhoneアプリについて】

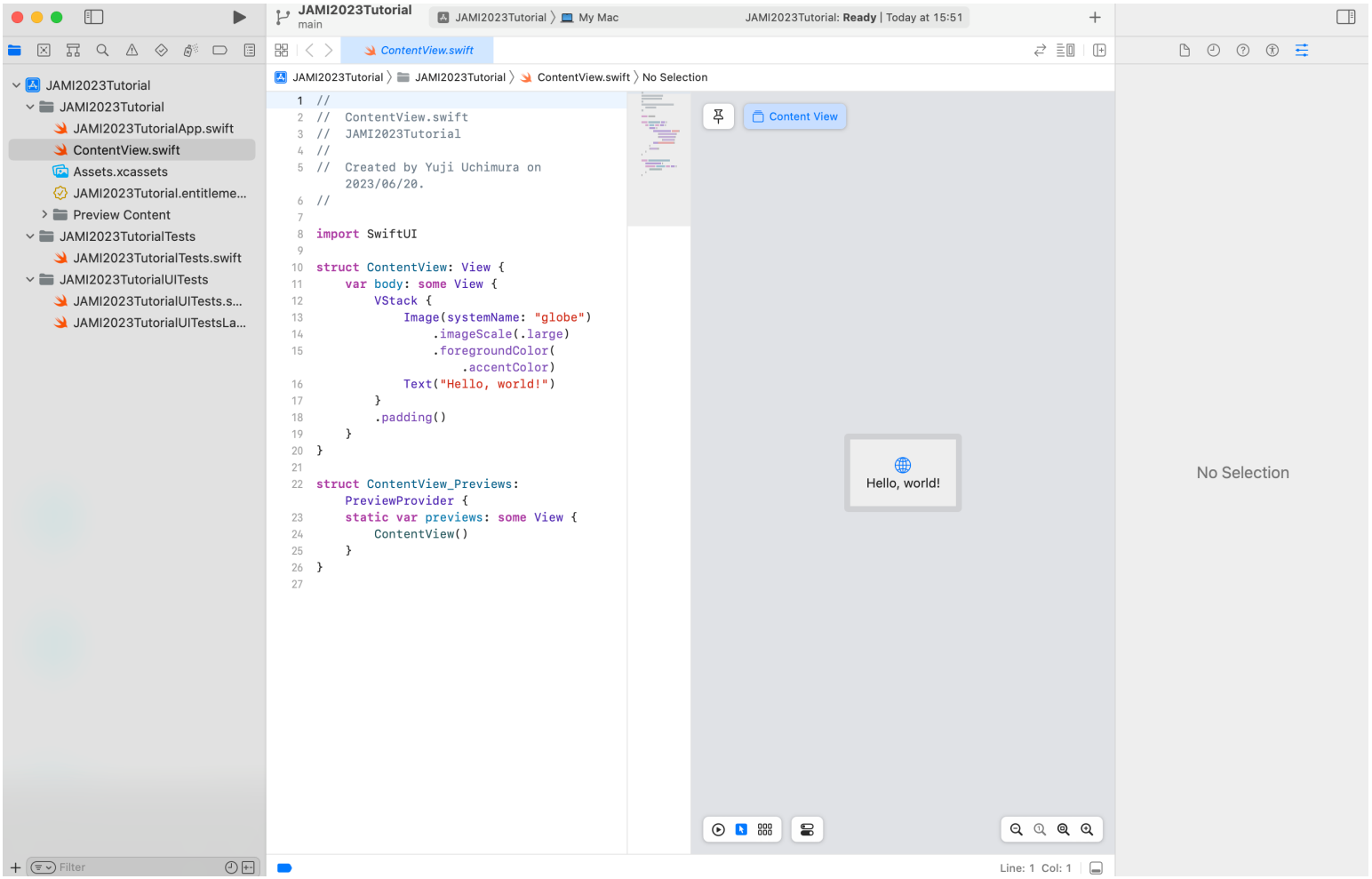
アプリ開発の流れ６

6. 項目毎にヘルスケアの利用許可を得ます②

【iPhoneアプリについて】

アプリ開発の流れ7

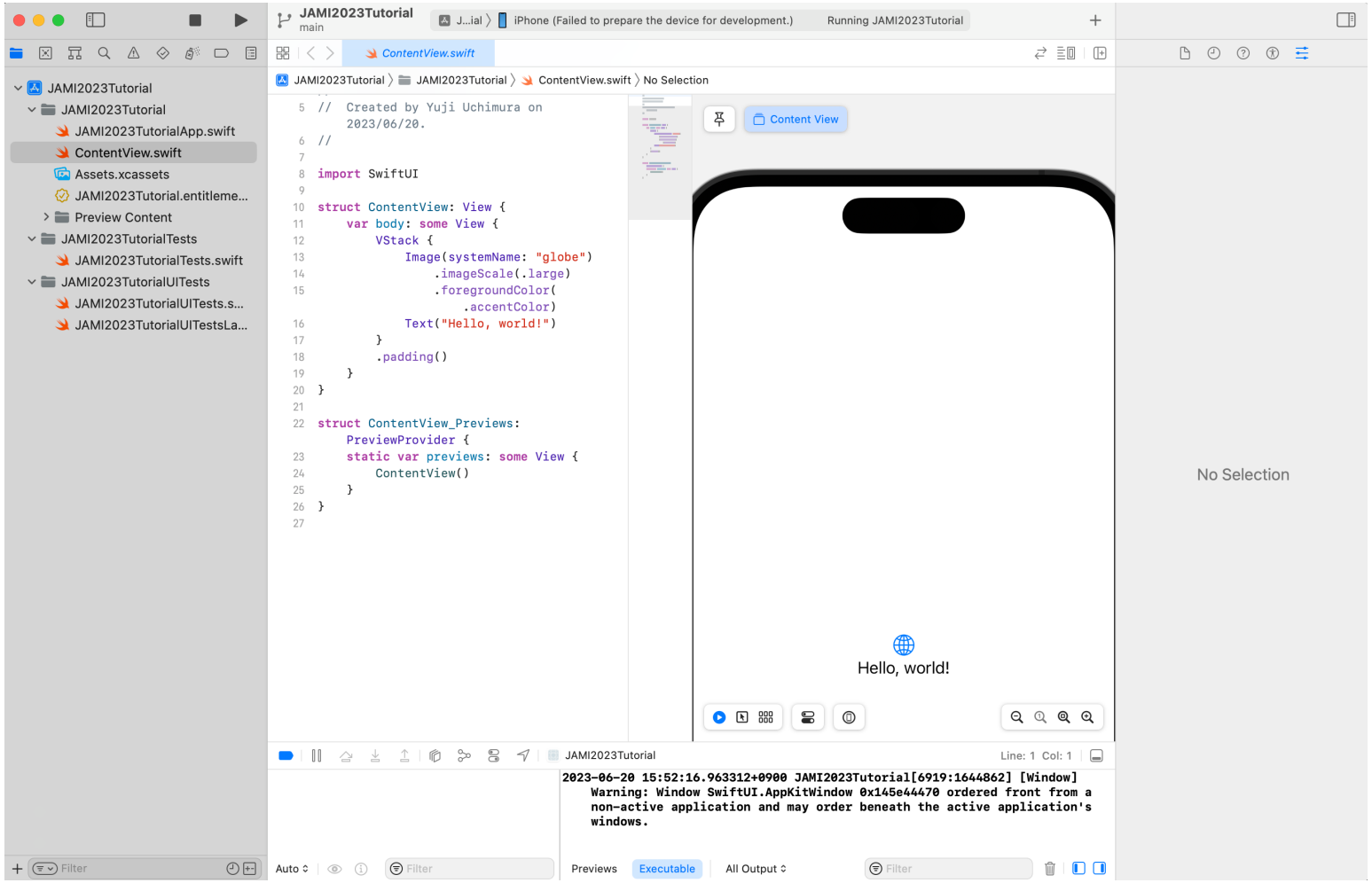
7. あとは普通にコードを書きます



【iPhoneアプリについて】

アプリ開発の流れ８

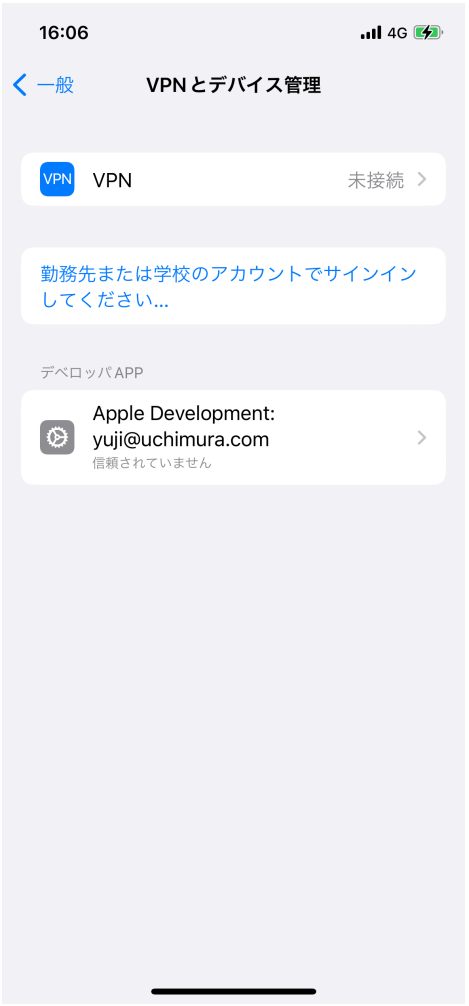
8. 画面も作っていきます



【iPhoneアプリについて】

アプリ開発の流れ９

9. 実機開発の場合、初回起動時に「信頼されていないデベロッパ」ダイアログが表示されるので設定します。



「設定」-「一般」-「VPNとデバイス管理」

で許可設定します。

【iPhoneアプリについて】

アプリ開発の流れ９

9. 実機開発の場合、初回起動時に「信頼されていないデベロッパ」ダイアログが表示されるので設定します。

「信頼されていません」から「検証済み」に変わったことを確認します。

標準データモデルを用いたPHRデータと医療情報との連携

- 東京医科歯科大学の病院情報システムのデータはDWHへ蓄積されている。
- 医系・歯系の電子カルテ、医事会計システム、主要部門システムの約５０システムのデータを蓄積している。
- DWHは標準データモデルの１つである「SDM（Semantic Data Modeling）」を採用した「JUST DWH」を採用した。
- PHRデータをSDM（SDM_VITAL_RAW）へ変換を行う事で、医療情報との連携が可能となる。
- 今回はiPhoneアプリからREST-APIでPHR情報を取得し、DWHデータとの連携を見据えてサーバにPHRデータを集約するサンプルプログラムを作成した。

iPhoneアプリで取得したPHRデータをサーバへ送信する

8.　iPhoneアプリからREST-APIでサーバへ転送します

```
func sendServer(uuid: String, typeid: Int, type: String, startdate: String,
    device: String, personalId: String, serverUrl: String) {
    // API call
    guard let url = URL(string: serverUrl) else { return }

    let data: [String: Any] = [
        "uuid": uuid,
        "personalid": personalId,
        "typeid": typeid,
        "type": type,
        "startdate": startdate,
        "enddate": enddate,
        "result": result,
        "unit": unit,
        "source": source,
        "device": device
    ]

    guard let httpBody = try? JSONSerialization.data(withJSONObject: data, o

    var request = URLRequest(url: url)
    request.timeoutInterval = 200*1000
    request.httpMethod = "POST"
    request.addValue("application/json", forHTTPHeaderField: "content-type")
    request.httpBody = httpBody

    let numberOfTimes = 100

    let task =  URLSession.shared.retryDataTask(withRequest: request, times:
        DispatchQueue.main.async {
            self.count += 1
        }
        guard let response = response as? HTTPURLResponse,
            (200...299).contains(response.statusCode) else {
            return
        }
    }
```

ヘルスケアから取得したPHR情報をまとめています。

REST-APIでPOSTします

iPhoneアプリで取得したPHRデータをサーバへ送信する

8. iPhoneアプリ内で各種設定できるように設定画面を作っています

サーバ情報

連携項目の設定

iPhoneアプリで取得したPHRデータをDBへ格納する①

8. REST-APIでiPhoneからPHRデータを取得してDBへ格納します。

- サーバプログラムはPythonの軽量WebアプリケーションフレームワークであるFlaskを使用して開発しました。
- DBはPostgreSQLを使っています。

```python
@api.route('/results', methods=['POST'])
def post_result():
    # jsonリクエストから値取得
    payload = request.json
    uuid = payload.get('uuid')
    personalid = payload.get('personalid')
    typeid = payload.get('typeid')
    type = payload.get('type')
    startdate = payload.get('startdate')
    enddate = payload.get('startdate')
    result = payload.get('result')
    unit = payload.get('unit')
    source = payload.get('source')
    device = payload.get('device')

    # レコードの登録 新規作成したオブジェクトをaddしてcommit
    result = Result(uuid, personalid, typeid, type, startdate, enddate, result, unit, source, device)
    db.session.add(result)
    db.session.commit()
```

iPhoneアプリで取得したPHRデータをDBへ格納する②

8. PHRデータを標準データモデル（SDM）でDBへ格納します。（SDM_VITAL_RAWの抜粋）

	record_id [PK] character	group_id character
1	VTLRAW,12345678,00000002,4,20230310100355	VTLRAW,12345678,00000002,4,20230310100355
2	VTLRAW,12345678,00000002,4,20230310140045	VTLRAW,12345678,00000002,4,20230310140045
3	VTLRAW,12345678,00000002,4,20230310144531	VTLRAW,12345678,00000002,4,20230310144531
4	VTLRAW,12345678,00000002,4,20230315181128	VTLRAW,12345678,00000002,4,20230315181128
5	VTLRAW,12345678,00000002,4,2023	
6	VTLRAW,12345678,00000002,4,2023	

measurement_item_code character	measurement_item character varying	measurement_device character varying	measurement_bodypart character varying
4	取り込まれた酸素のレベル	N/A	
4	取り込まれた酸素のレベル	N/A	
4	取り込まれた酸素のレベル	N/A	
4	取り込まれた酸素のレベル	N/A	
4	取り込まれた酸素のレベル	N/A	
4	取り込まれた酸素のレベル	N/A	

measurement_method character varying	measurement_result character varying	measurement_unit character varying	measurement_value real	measurement_previous_value real	ratio_previous_value real
Airec0さんのApple Watch	97	%	97	0	0
Airec0さんのApple Watch	97	%	97	0	0
Airec0さんのApple Watch	96	%	96	0	0
Airec0さんのApple Watch	100	%	100	0	0
Airec0さんのApple Watch	100	%	100	0	0
Airec0さんのApple Watch	96	%	96	0	0

参考文献
参考サイト

- Apple HealthKit関連
  - https://developer.apple.com/jp/design/human-interface-guidelines/healthkit
  - https://developer.apple.com/documentation/healthkit
- 札幌医科大学と富士通、ヘルスケア領域のデータポータビリティ実現に向けて、個人の健康データの活用推進に合意
  - https://pr.fujitsu.com/jp/news/2023/01/16.html
- Apple「Apple Watchを使用した最大酸素摂取量による心肺機能の推定」
  - https://www.apple.com/jp/healthcare/docs/site/Using_Apple_Watch_to_Estimate_Cardio_Fitness_with_VO2_max_JP.pdf
- Apple「Apple Watchを使用した6分間の歩行距離の推定」
  - https://www.apple.com/jp/healthcare/docs/site/Using_Apple_Watch_to_Estimate_Six-Minute_Walk_Distance_JP.pdf
- Apple「iPhoneの歩行指標を使用した歩行の質の測定」
  - https://www.apple.com/jp/healthcare/docs/site/Measuring_Walking_Quality_Through_iPhone_Mobility_Metrics_JP.pdf

ご清聴ありがとうございました