Exposure Notification API

勉強会

有山　圭二

2021/06/09

目次

勉強会の目的

Exposure Notifications API v2 に対応しなければならない。

どうせ対応するなら楽な方がいい。

これまでの有山の取り組みを通じて、現時点でv2についてわかっていることを開発チームに共有することで、今後の作業内容を見通しやすくする。

Exposure Notifications APIとは

Exposure Notifications（EN）API とは？

Exposure Notifications（接触確認） APIは、GoogleとAppleが共同で開発しているスマートフォンのBluetoothを利用して、人と人が接触したことを検知、記録するための仕組み。

GAEN（Google Apple Exposure Notification）と呼ばれることもある。

EN APIを使ったアプリを「接触確認アプリ」と言う。

接触確認アプリの利用者は、利用者が新型コロナウイルス（COVID-19）感染症の陽性が確定したときに、一定期間内に接触した者に匿名で通知を行うことができる。

EN APIのアップデート

EN APIは、GoogleとAppleが共同で開発している。GoogleとAppleはEN APIをアップデートすることがある。

アップデートはGoogleとAppleが行い、接触確認アプリ側からは制御はできない。

GoogleとAppleでアップデート戦略が異なる。

EN APIのアップデート戦略

Google（Android）

APIは、プラットフォームから分離されたGoogle Mobile Services（GMS）のバージョンに紐付く。Androidのバージョンに依存しない

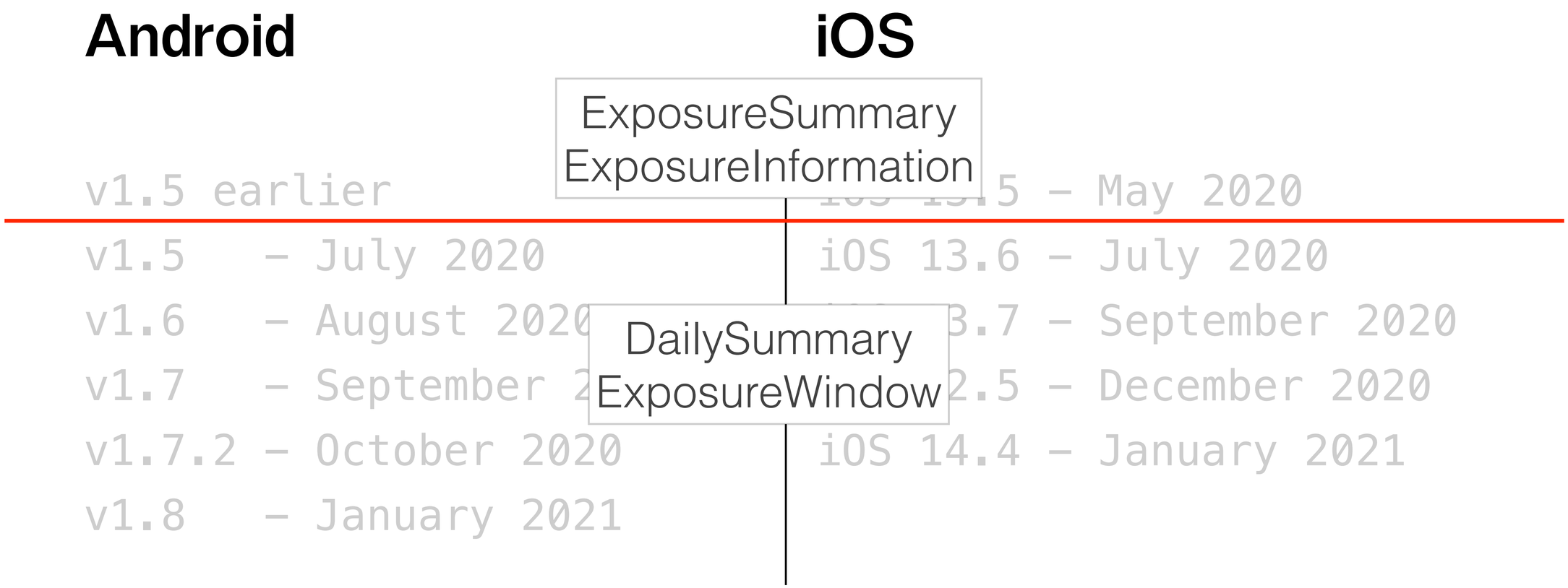
Apple（iOS）

iOSのバージョンに紐付く。古いバージョンのiOSでは、その時点で対応しているAPIしか使えない。

バージョンの変遷

Android	iOS
v1.5 earlier	iOS 13.5 – May 2020
v1.5 – July 2020	iOS 13.6 – July 2020
v1.6 – August 2020	iOS 13.7 – September 2020
v1.7 – September 2020	iOS 12.5 – December 2020
v1.7.2 – October 2020	iOS 14.4 – January 2021
v1.8 – January 2021	

2020年7月のアップデートでExposureWindowが導入



ExposureSummary
ExposureInformation

```
"exposure_summary": {
  "AttenuationDurationsInMinutes": [
    1800,
    480,
    0
  ],
  "DaysSinceLastExposure": 0,
  "MatchedKeyCount": 2,
  "MaximumRiskScore": 255,
  "SummationRiskScore": 54560
},
…
```

```
"exposure_informations": [
  {
    "AttenuationDurationsInMinutes": [
      1436,
      0,
      0
    ],
    "AttenuationValue": 26,
    "DateMillisSinceEpoch": 1622851200000,
    "Duration": 30.0,
    "TotalRiskScore": 256,
    "TransmissionRiskLevel": 4
  },
  …
```

https://github.com/keiji/chino/wiki/Sample-ExposureData-iOS-(V1)

https://github.com/keiji/chino/wiki/Sample-ExposureData-Android-(Legacy-v1)

DailySummary
ExposureWindow


```
"daily_summaries": [
  {
    "DaysSinceEpoch": 1622937600000,
    "DaySummary": {
      "MaximumScore": 1860.0,
      "ScoreSum": 51660.0,
      "WeightedDurationSum": 51660.0
    },
    "ConfirmedClinicalDiagnosisSummary": null,
    "ConfirmedTestSummary": {
      "MaximumScore": 1860.0,
      "ScoreSum": 51660.0,
      "WeightedDurationSum": 51660.0
    },
    "RecursiveSummary": null,
    "SelfReportedSummary": null
  },
```

```
"exposure_windows": [
  {
    "CalibrationConfidence": 3,
    "DateMillisSinceEpoch": 1622937600000,
    "Infectiousness": 1,
    "ReportType": 1,
    "ScanInstances": [
      {
        "MinAttenuationDb": 24,
        "SecondsSinceLastScan": 300,
        "TypicalAttenuationDb": 25
      },
      {
        "MinAttenuationDb": 24,
        "SecondsSinceLastScan": 240,
        "TypicalAttenuationDb": 25
      },
      …
```

https://github.com/keiji/chino/wiki/Sample-ExposureData-iOS-(V2)

https://github.com/keiji/chino/wiki/Sample-ExposureData-Android-(Exposure-Window-mode)

プラットフォームでAPIの呼び方が異なる

Android

Legacy v1 mode

v1.5 earlier

v1.5 – July 2020

v1.6 – August 2020

ExposureWindow mode

v1.7.2 – October 2020

v1.8 – January 2021

iOS

Version 1

iOS 13.5 – May 2020

iOS 13.6 – July 2020

iOS 13.7 – September 2020

iOS 12.5 – [illegible] 2020

Version 2

iOS 14.4 – January 2021

ExposureWindow modeの検証

Exposure Notifications API: Android Reference Design

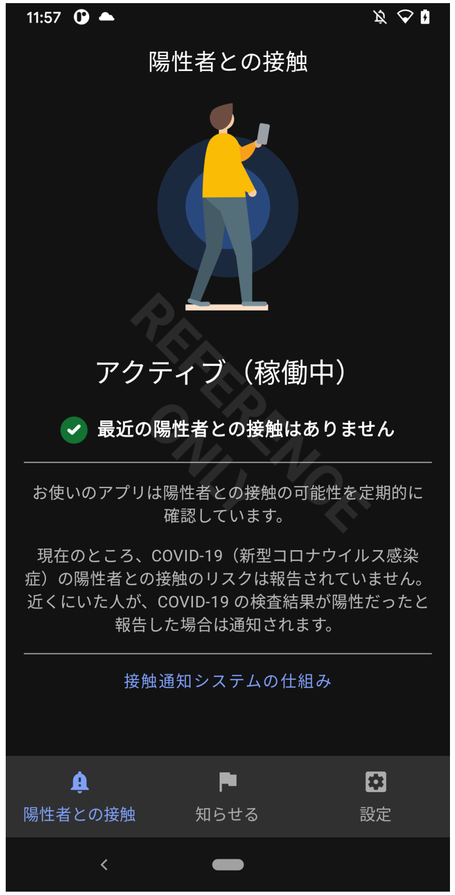
Google製

Java言語で書かれている

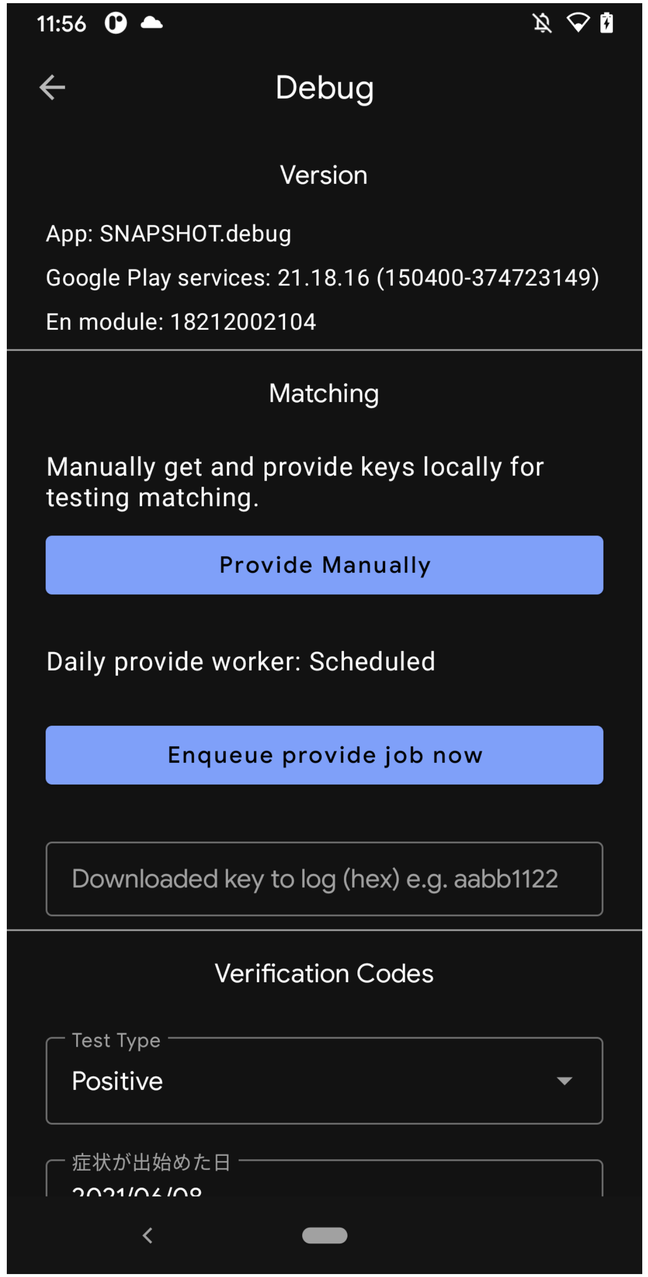
デバッグモードが充実している

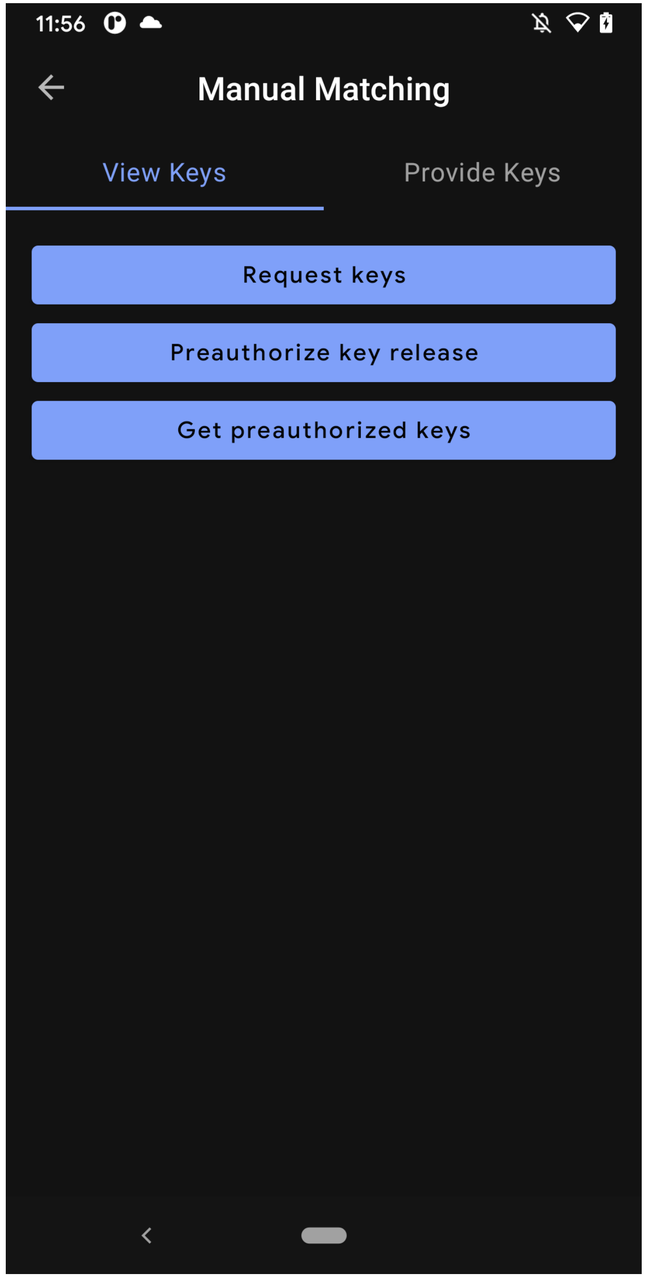
接触確認の設定値は固定（ビルド時）

https://github.com/google/exposure-notifications-android/blob/master/app/configs/sample.json

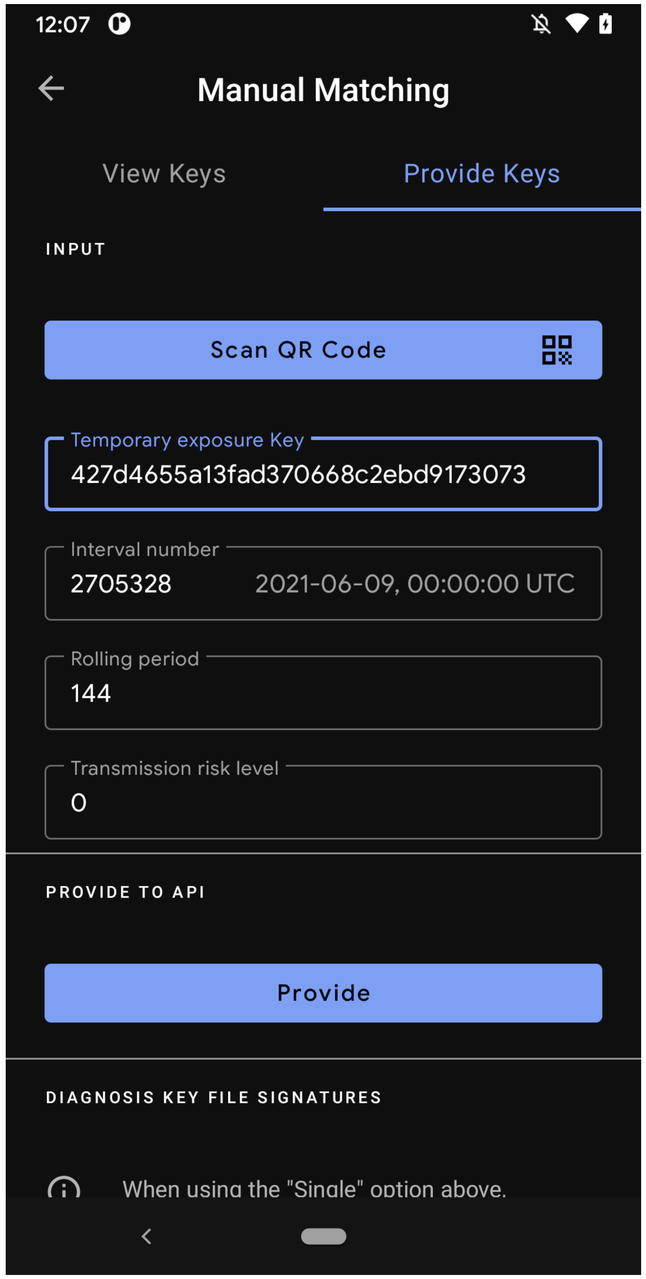


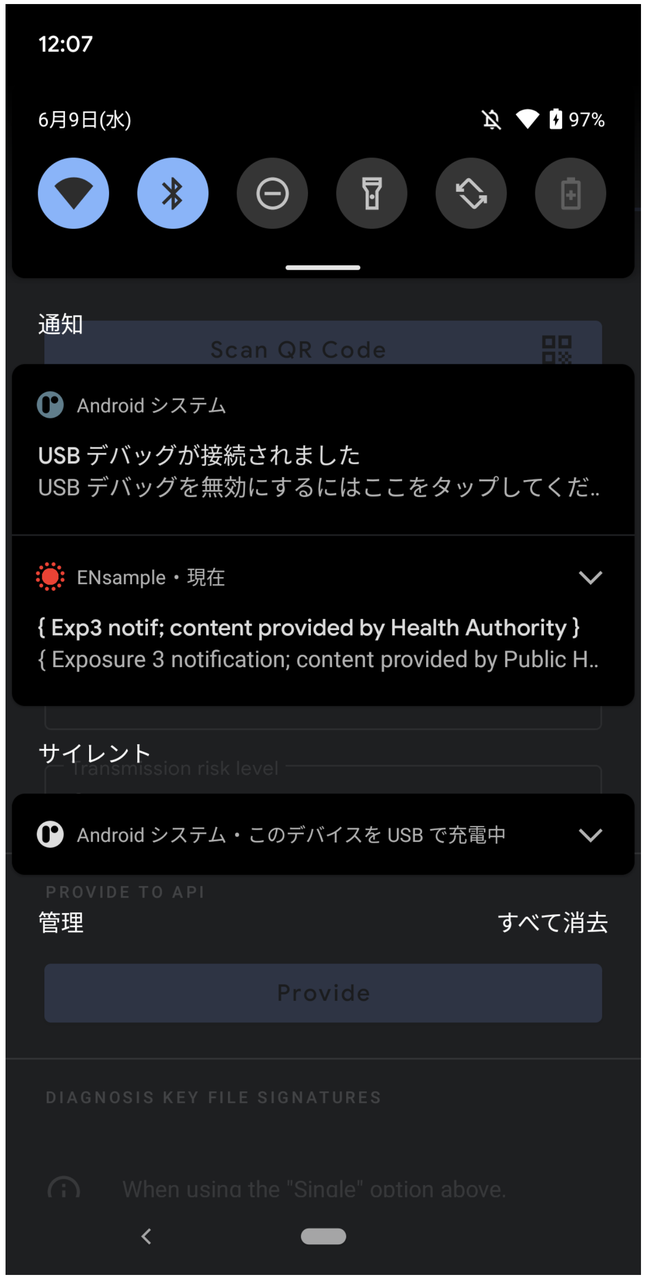
https://github.com/google/exposure-notifications-android

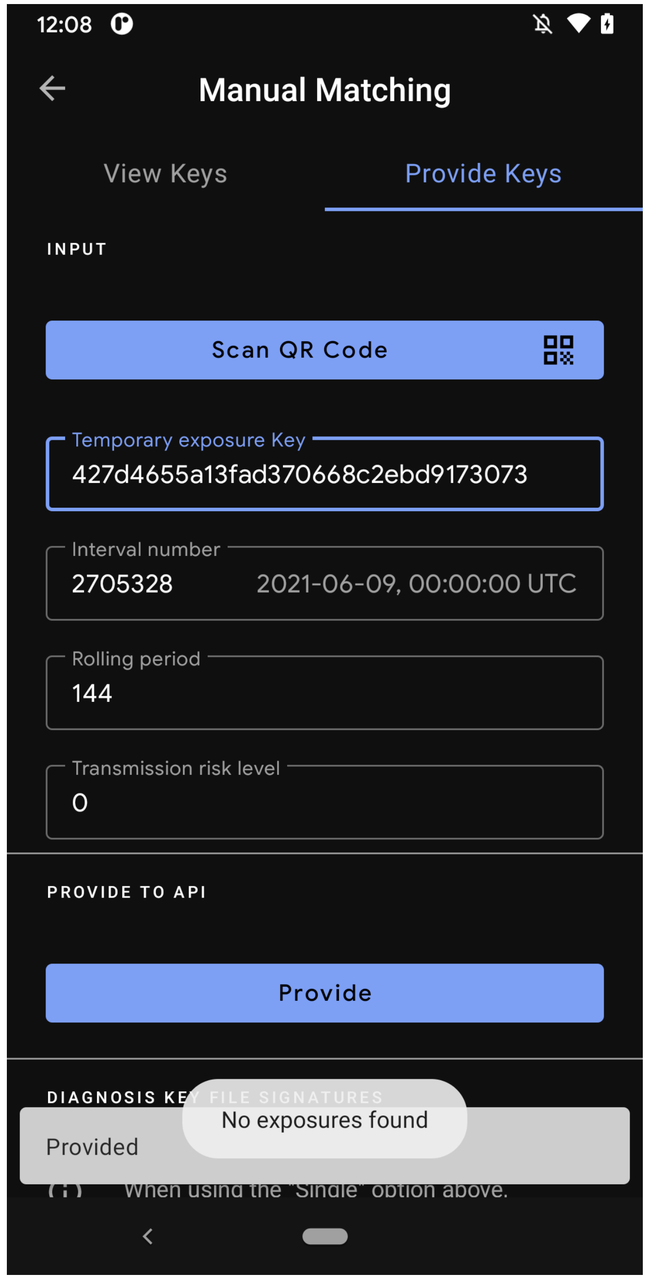



11:56
Manual Matching
View Keys
Provide Keys
Request keys
Preauthorize key release
Get preauthorized keys


12:02
Manual Matching
View Keys
Provide Keys
Date: June 3, 2021
KeyData: 5b1ae97654e5ff4ae260568498f1d856
RollingStartIntervalNumber: 2704464
RollingPeriod: 101
TransmissionRiskLevel: 0
Date: June 3, 2021
KeyData: ce6fc5c11272921e8193de5875a4f0c8
RollingStartIntervalNumber: 2704464
RollingPeriod: 98
TransmissionRiskLevel: 0
Date: June 3, 2021
KeyData: d7860276fbfba94136a04476bc18214d
RollingStartIntervalNumber: 2704464
RollingPeriod: 144
TransmissionRiskLevel: 0


12:07
Manual Matching
View Keys
Provide Keys
INPUT
Scan QR Code
Temporary exposure Key
427d4655a13fad370668c2ebd9173073
Interval number
2705328
2021-06-09, 00:00:00 UTC
Rolling period
144
Transmission risk level
0
PROVIDE TO API
Provide
DIAGNOSIS KEY FILE SIGNATURES
When using the "Single" option above.


12:07
6月9日(水)
97%
通知
Android システム
USB デバッグが接続されました
USB デバッグを無効にするにはここをタップしてくだ..
ENsample・現在
{ Exp3 notif; content provided by Health Authority }
{ Exposure 3 notification; content provided by Public H..
サイレント
Android システム・このデバイスを USB で充電中
管理
すべて消去


12:08
Manual Matching
View Keys
Provide Keys
INPUT
Scan QR Code
Temporary exposure Key
427d4655a13fad370668c2ebd9173073
Interval number
2705328
2021-06-09, 00:00:00 UTC
Rolling period
144
Transmission risk level
0
PROVIDE TO API
Provide
No exposures found
Provided

EN Sample

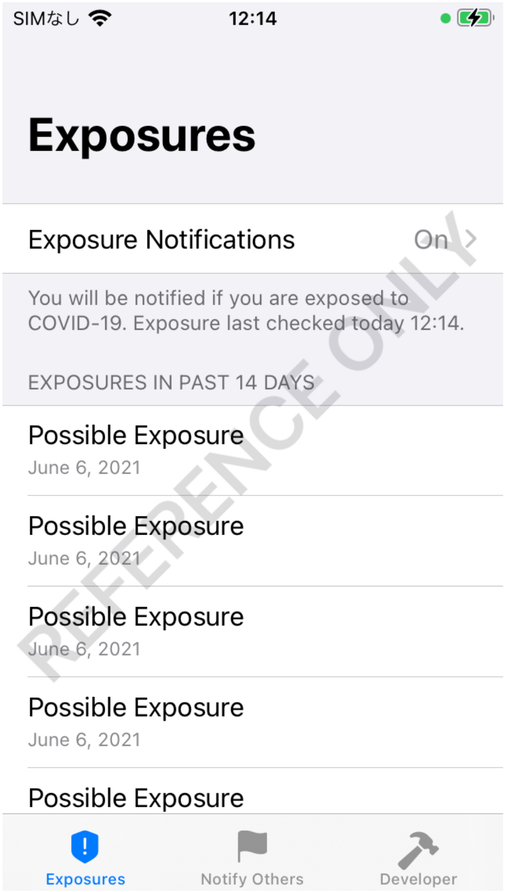
Apple製

Swiftで書かれている

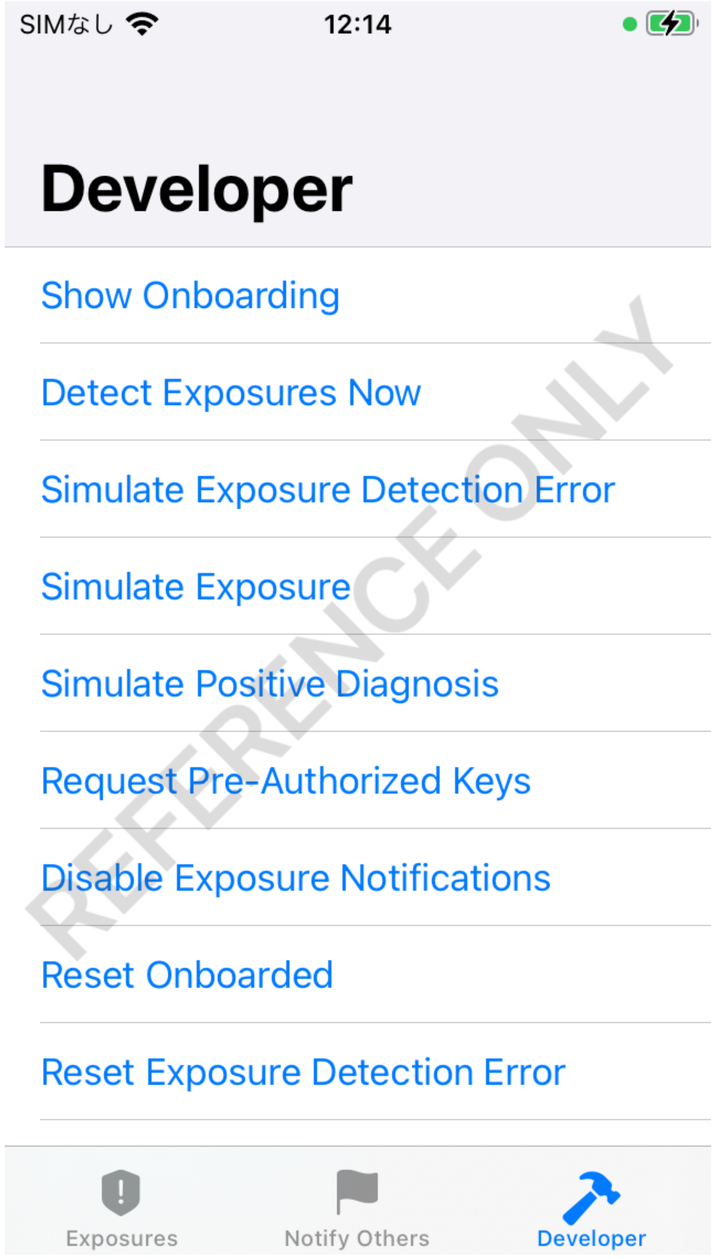
デバッグモードが充実している

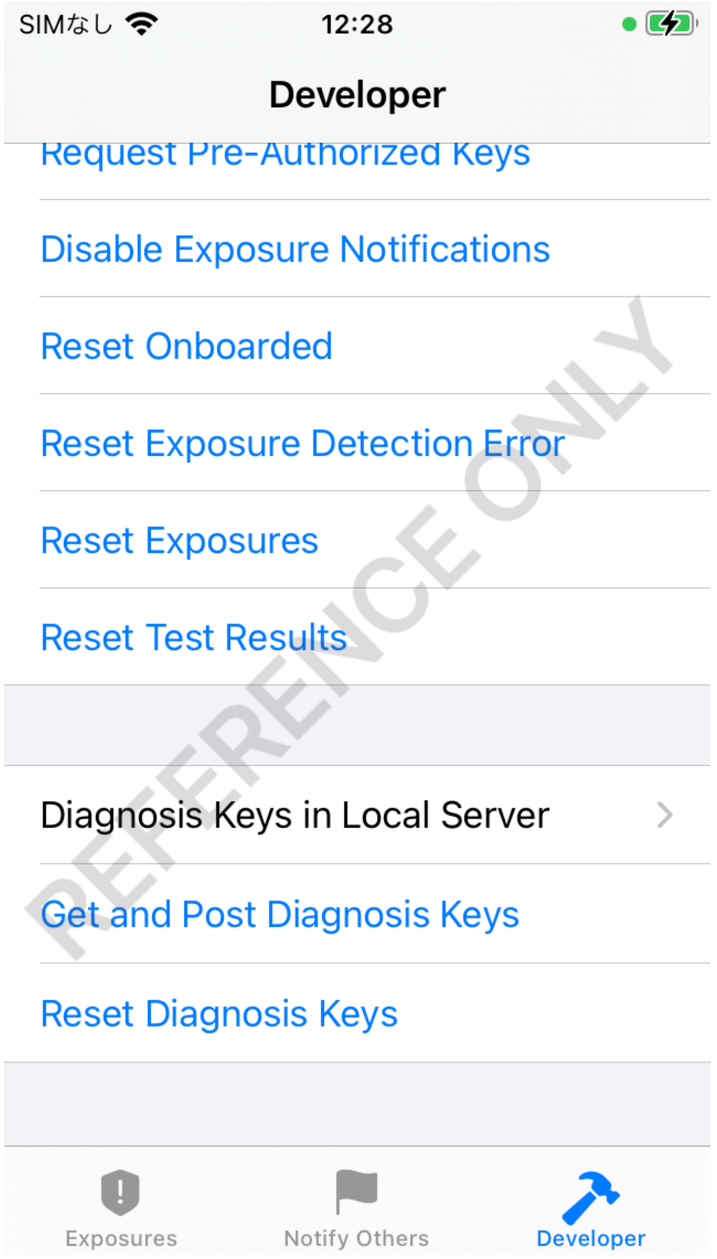
接触確認の設定値は固定（ビルド時）

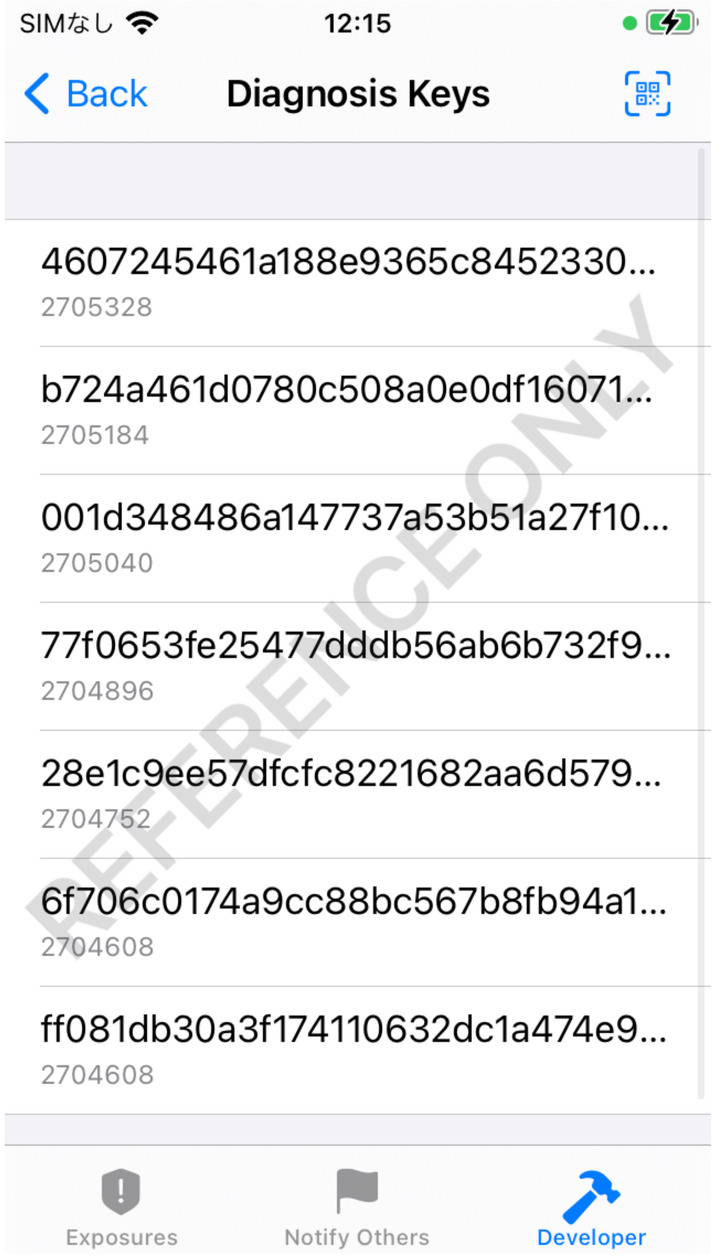
Common/Model/Server.swift

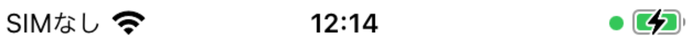


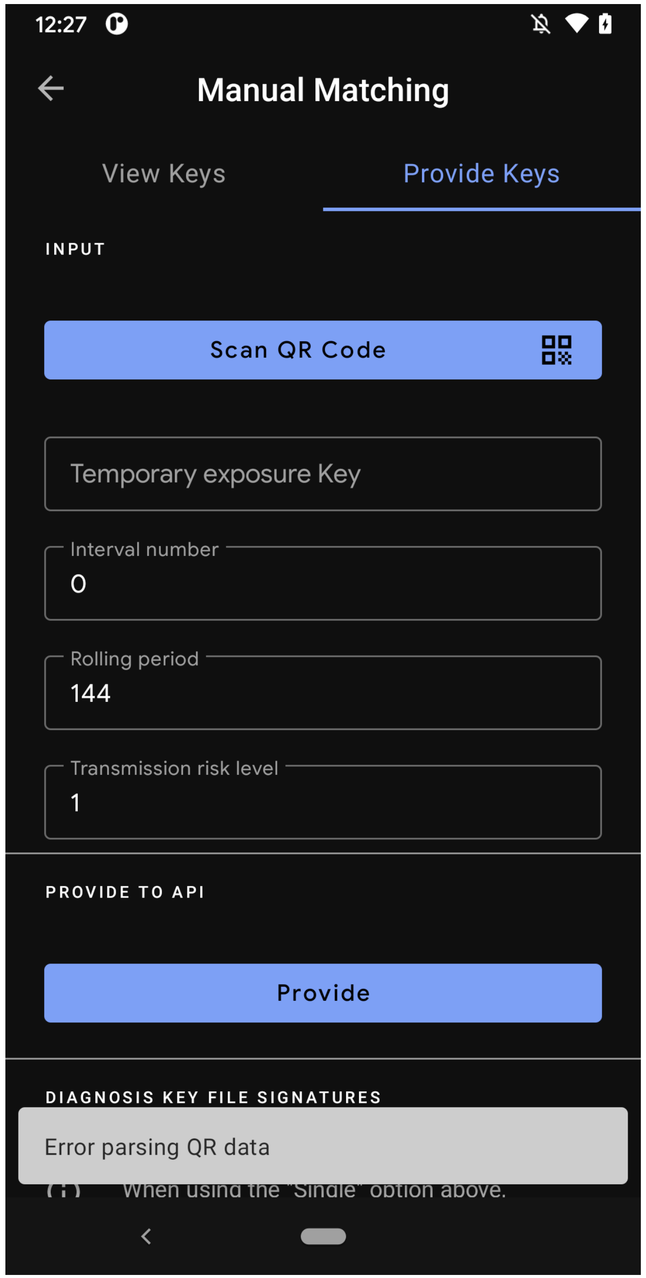
https://developer.apple.com/documentation/exposurenotification/building_an_app_to_notify_users_of_covid-19_exposure


SIMなし
12:14
Developer
Show Onboarding
Detect Exposures Now
Simulate Exposure Detection Error
Simulate Exposure
Simulate Positive Diagnosis
Request Pre-Authorized Keys
Disable Exposure Notifications
Reset Onboarded
Reset Exposure Detection Error
Exposures
Notify Others
Developer
REFERENCE ONLY

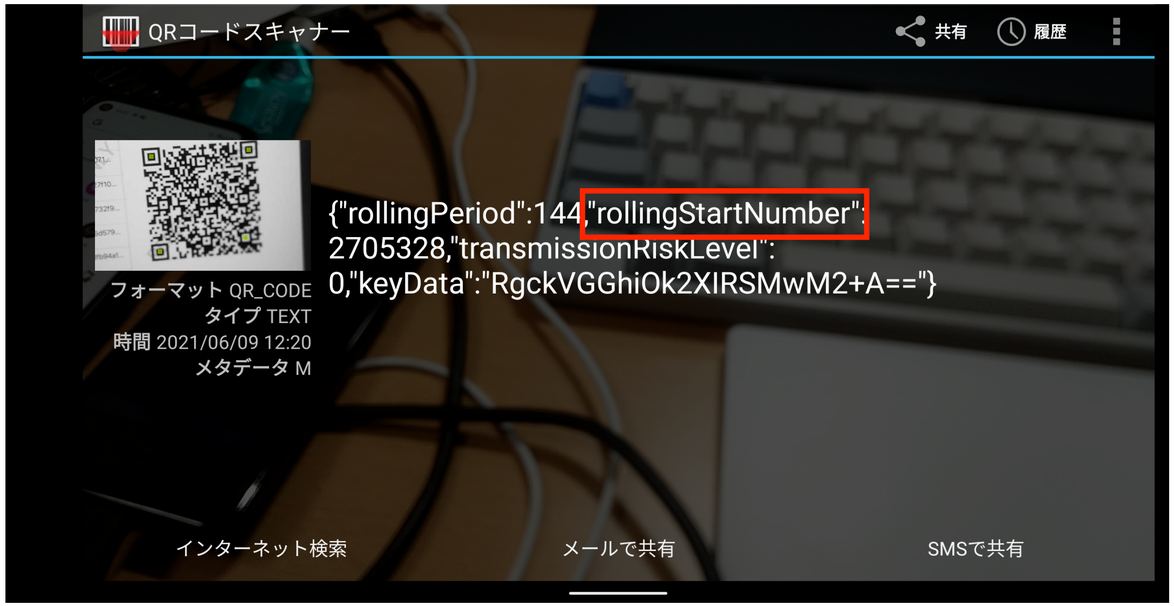

SIMなし
12:28
Developer
Request Pre-Authorized Keys
Disable Exposure Notifications
Reset Onboarded
Reset Exposure Detection Error
Reset Exposures
Reset Test Results
Diagnosis Keys in Local Server
Get and Post Diagnosis Keys
Reset Diagnosis Keys
Exposures
Notify Others
Developer
REFERENCE ONLY

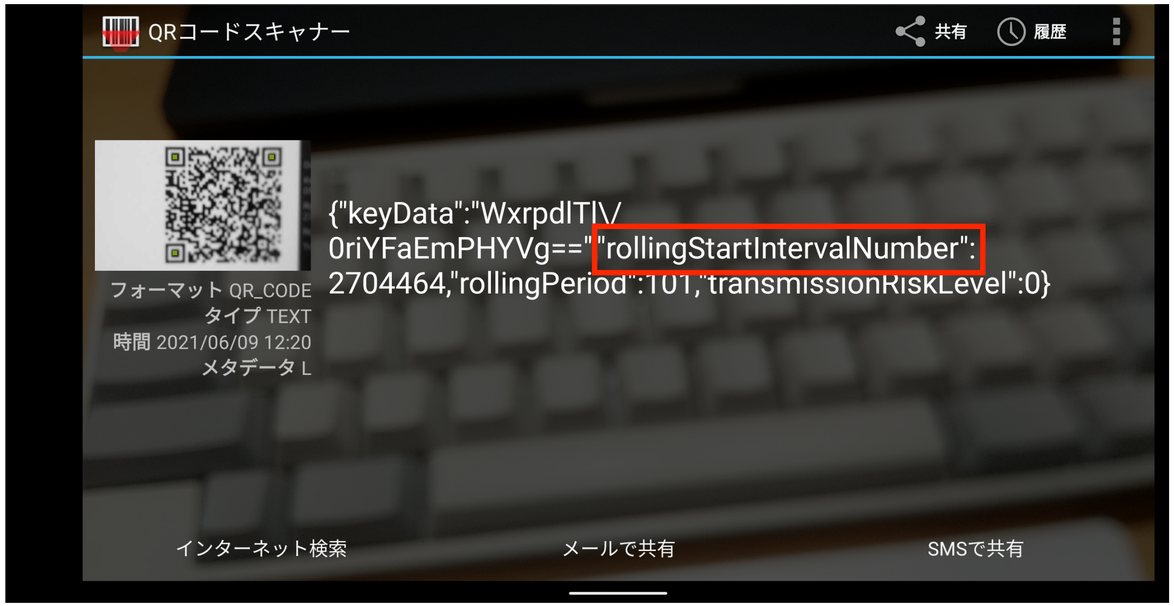






iOSで生成したQRコードは Androidで読み込めない


QRコードスキャナー
共有
履歴
フォーマット QR_CODE
タイプ TEXT
時間 2021/06/09 12:20
メタデータ M
{"rollingPeriod":144,"rollingStartNumber":
2705328,"transmissionRiskLevel":
0,"keyData":"RgckVGGhiOk2XIRSMwM2+A=="}
インターネット検索
メールで共有
SMSで共有


QRコードスキャナー
共有
履歴
フォーマット QR_CODE
タイプ TEXT
時間 2021/06/09 12:20
メタデータ L
{"keyData":"WxrpdlTl\/
0riYFaEmPHYVg==","rollingStartIntervalNumber":
2704464,"rollingPeriod":101,"transmissionRiskLevel":0}
インターネット検索
メールで共有
SMSで共有

Xamarin.ExposureNotificationの

ExposureWindow mode対応状況

Xamarin.ExposureNotificationとは

Xamarin公式のEN API用のライブラリ。

COCOAとSmittestopp（デンマーク・ノルウェー）が採用している。

COCOA

https://github.com/cocoa-mhlw/cocoa

Smittestop

https://github.com/folkehelseinstituttet/Fhi.Smittestopp.App

https://github.com/xamarin/XamarinComponents/tree/main/XPlat/ExposureNotification

ExposureWindow mode対応状況

Xamarin.ExposureNotificationは、2021年6月9日時点でExposureWindow modeに対応していない。

2020年9月26日からPull Request「Implement the shared API for v2」で作業中（dev/api-v2ブランチ）。

https://github.com/xamarin/XamarinComponents/pull/955

最終更新は4月27日と1ヶ月以上、動きがない状況。

SmitteStoppはExposureWindow modeに対応

NuGetパッケージ（Xamarin.ExposureNotification.0.16.0-alpha.nupkg）をローカルに持っている。

https://github.com/folkehelseinstituttet/Fhi.Smittestopp.App/tree/dev/NDB.Covid19/TempNugetFeed

もともとバックグラウンド処理を独自実装するなど、Xamarin.ExposureNotificationに依存しない仕組みにしていた。

https://github.com/folkehelseinstituttet/Fhi.Smittestopp.App/blob/dev/NDB.Covid19/NDB.Covid19.iOS/Utils/BackgroundServiceHandler.cs

https://github.com/folkehelseinstituttet/Fhi.Smittestopp.App/issues/223

Cappuccinoを使った

ExposureWindow modeの検証

Cappuccino（カプチーノ）とは

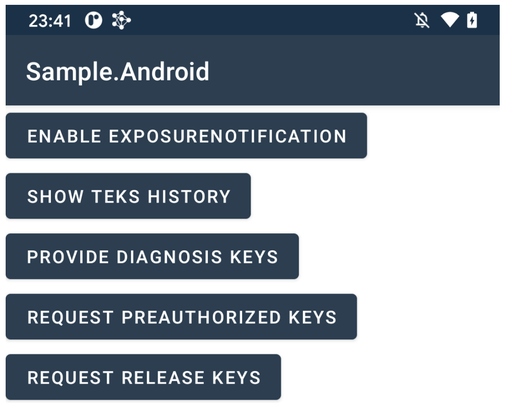
オープンソースのXamarin向け接触確認APIライブラリ。

Xamarin.ExposureNotificationのようにバックグラウンド処理をハンドリングしない。Android/iOSの薄いラッパーとして動作することを目指している。

MITライセンス。

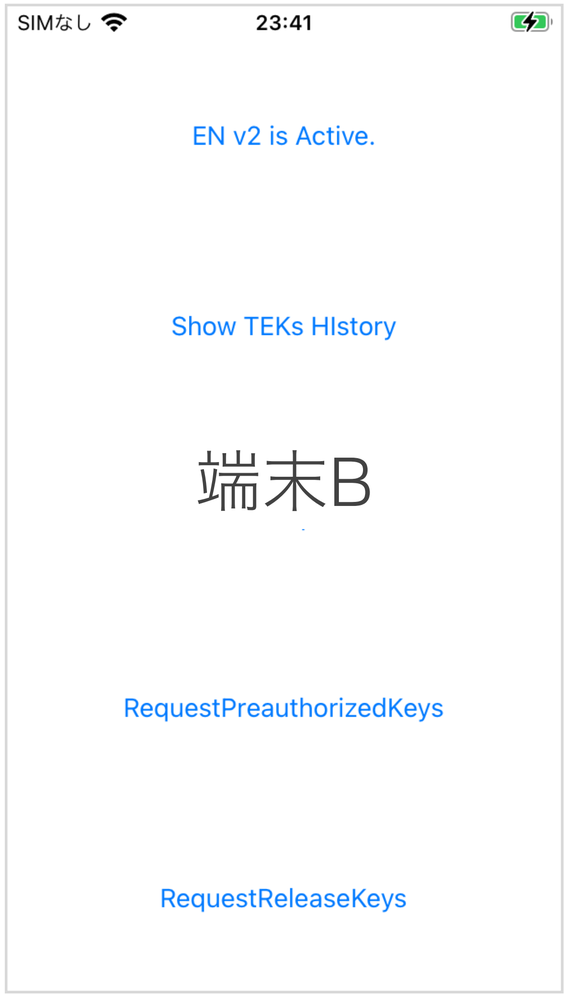
https://github.com/keiji/chino

Sampleアプリ（Android, iOS）



端末A

Sample.Android



端末B

Sample.iOS

デバッグ設定 - Android

デバッグ設定 - iOS

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/
PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>com.apple.developer.exposure-notification</key>
    <true/>
    <key>com.apple.developer.exposure-notification-test</key>
    <true/>
    <key>com.apple.developer.exposure-notification-test-skip-file-verification</key>
    <true/>
</dict>
</plist>
```

DiagnosisKeys（診断キー）の生成

日次キー（Temporary Exposure Key: TEK）を元に生成する。
デバッグ用の診断キー生成は、次の手順で行う。

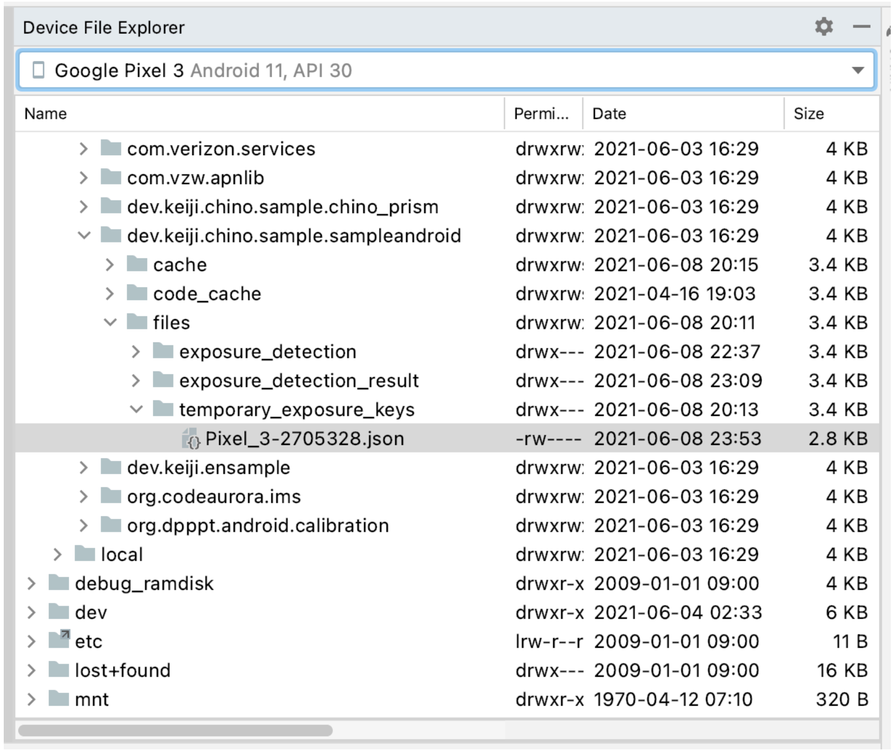
- 端末AからTEKを取得
- 署名用秘密鍵を生成（Docker）
- 診断キーを生成（Docker）

TEKの取得


23:41
Sample.Android
ENABLE EXPOSURENOTIFICATION
SHOW TEKS HISTORY
PROVIDE DIAGNOSIS KEYS
REQUEST PREAUTHORIZED KEYS
REQUEST RELEASE KEYS
23:53
Sample.Android
ENABLE EXPOSURENOTIFICATION
WXRPDLTL/0RIYFAEMPHYVG==
ZM/FWRJYKH6BK95YDATWYA==
14YCDVV7QUE2OER2VBGHTQ==
2LFG3J8XW6E8S9GVU4PTIG==
7IKZSYB+HLBJB6JPI/VDPQ==
J8M9HWALJNHM/SXWJZ2KZA==
NQATQOJP/077L2PLRM1AUA==
NI2P71DJI0WTXRCUG9D7IQ==
BZ1YE4NJHPXZHA+UPM/Q+W==
CQI4KLZ03QVPCBRHK6/BLA==
J7CZR0QHXNVWVW0+QIWJWA==
LUHKDAKEJB+COLGBKDG3RG==
WDFZO7W0FCZIWBIOVJO3TG==
JVK5E+RRLDGFLV2ODBOYLG==
3P59VGLZLH4AIKKEE7EE5W==
9YI+DG5VMFDXTUJALKR9UW==
PROVIDE DIAGNOSIS KEYS
REQUEST PREAUTHORIZED KEYS
REQUEST RELEASE KEYS

TEKの取得

端末Aのファイルシステムにアクセスして
dev.keiji.chino.sample.sampleandroid/files/temporary_exposure_keys にある Jsonファイルをダウンロードする



https://github.com/keiji/chino/wiki/Sample-TemporaryExposureKeys

署名用秘密鍵の生成（Docker）

ディレクトリ export-generate-diagnosis-keys で作業する。

```
$ cd export-generate-diagnosis-keys
$ docker build -t keiji/exposure-notifications-server v0.29.0

$ docker run --rm -v `pwd`/work:/work keiji/exposure-notifications-server \
    /bin/sh -c "cd /work && openssl ecparam -genkey -name prime256v1 -noout -out private.pem"
```

work に private.pem が生成される

診断キーの生成（Docker）

取得したTEKのJsonファイルを work ディレクトリに配置して次のコマンドを実行する

```
$ docker build -t keiji/exposure-notifications-server $1

$ docker run --rm -v `pwd`/work:/work keiji/exposure-notifications-server \
    /bin/sh -c "go run ./tools/export-generate \
        --signing-key=/work/private.pem \
        --tek-file=/work/Pixel_3-2705328.json \
        --region=440 \
        --key-id=440 \
    && cp /tmp/*.zip /work/"
```

work に診断キー testExport-*-records-*.zip が生成される

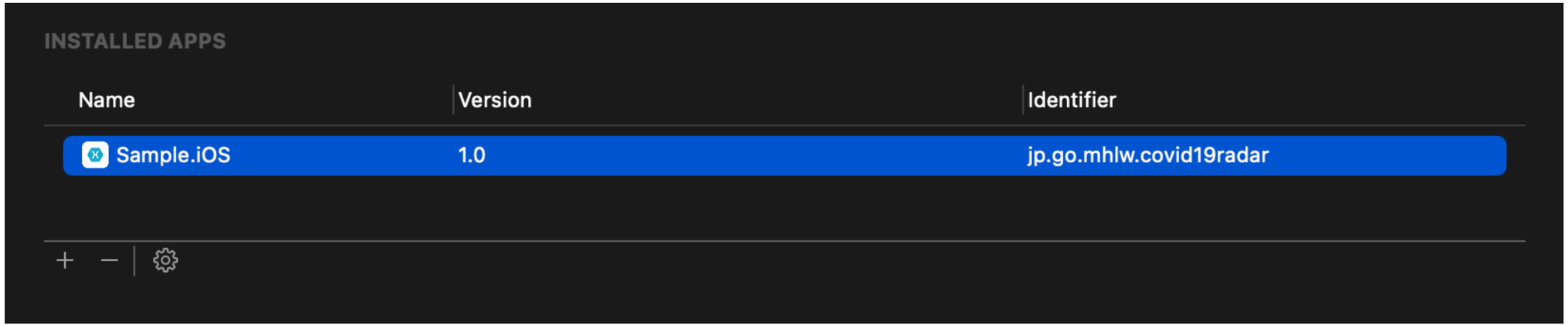
接触確認

生成した診断キーを端末Bにコピーして接触確認を行う。
接触確認は、次の手順で行う。

- 端末Bのアプリ（iOS）のコンテナをダウンロードする
- コンテナに生成した診断キーをコピーする
- 接触確認に使う設定ファイルを書き換える
- コンテナを端末Bにアップロード（置き換え）する
- 接触確認を実行する
- 端末Bのアプリ（iOS）のコンテナをダウンロードする
- 確認結果を取得する

アプリのコンテナをダウンロード

Xcodeの Devices and Simulators を開いて端末Bからアプリ Sample.iOS のコンテナ（xcappdata）をダウンロードする。



診断キーの配置（コピー）

ダウンロードしたコンテナ（xcappdata）を Finder から右クリックして Show Package Contentsを選択する。

ファイルシステムの AppData/Documents/exposure_detection の下に、生成した診断キーファイル（zip）を配置（コピー）する。

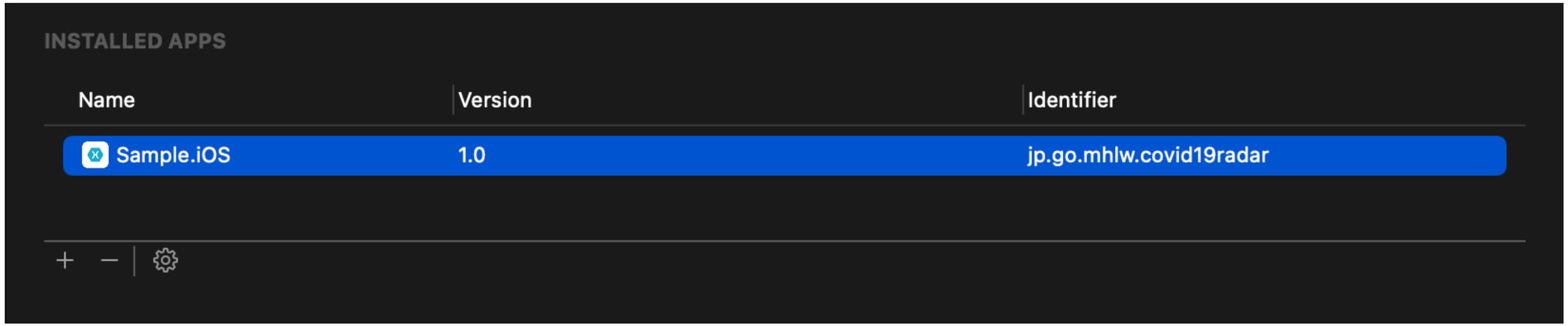
設定ファイルの書き換え

ファイルシステムの AppData/Documents/exposure_detection の下にある exposure_configuration.json を必要に応じて書き換える。

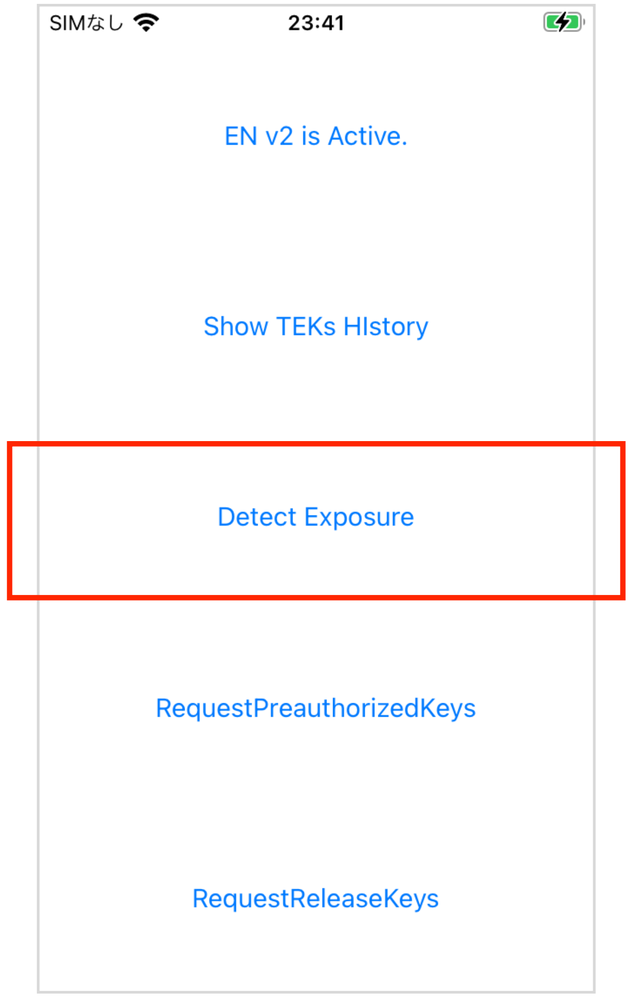
https://github.com/keiji/chino/wiki/Default-ExposureConfiguration

アプリのコンテナをアップロード（置き換え）

Xcodeの Devices and Simulators を開いて端末Bにアプリ Sample.iOS のコンテナ（xcappdata）をアップロードして置き換える。

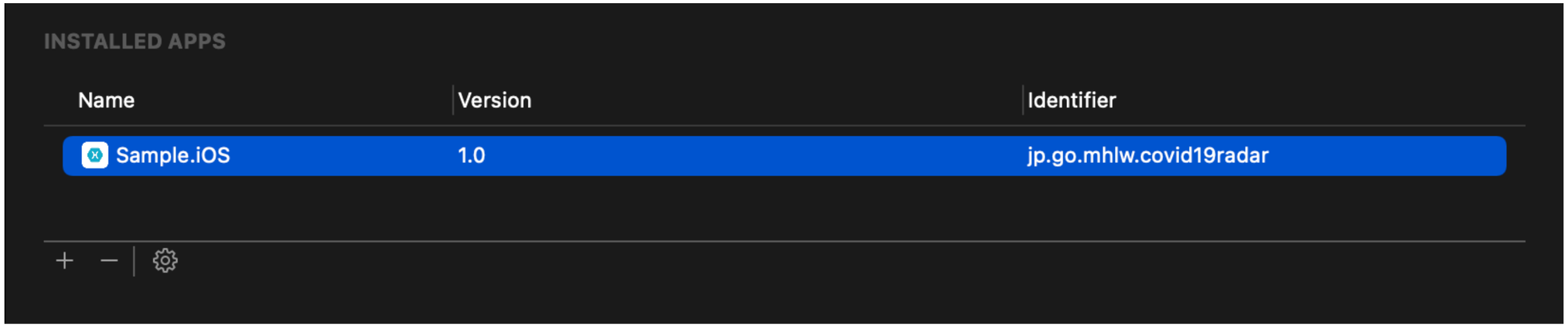


接触確認する


SIMなし
23:41
EN v2 is Active.
Show TEKs HIstory
Detect Exposure
RequestPreauthorizedKeys
RequestReleaseKeys

接触確認の結果取得

Xcodeの Devices and Simulators を開いて端末Bからアプリ Sample.iOS のコンテナ（xcappdata）をダウンロードする。



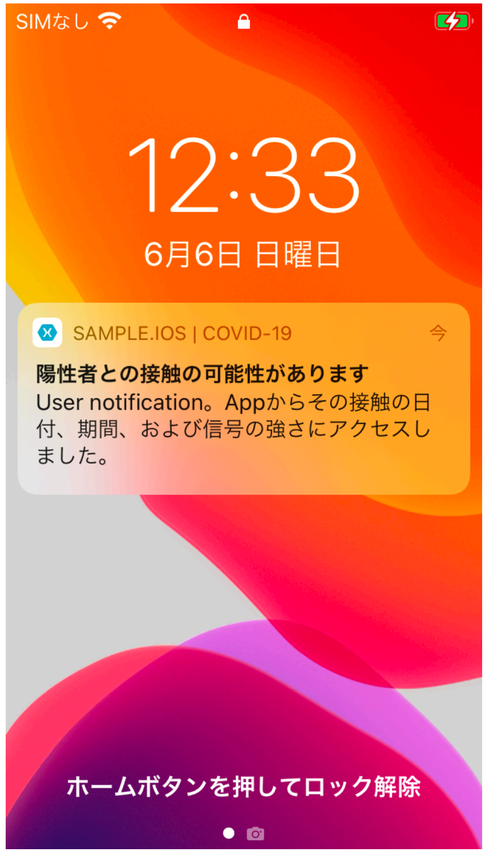
ファイルシステムの AppData/Documents/exposure_detection_result の下にあるJsonファイルを開く

https://github.com/keiji/chino/wiki/Sample-ExposureData-iOS-(V2)

ExposureWindow mode移行に

あたっての注意

ローカル通知の実装



接触情報を取得するときに通知が表示されない

→　ローカル通知を実装する必要がある

ExposureWindow modeへの切り換え（Android）

プログラム的に切り換える。

ExposureNotificationClientの接触確認でどのメソッドを呼び出すかで結果が変わる。

ExposureSummary ExposureInformation

```
provideDiagnosisKeys(List<File> keyFiles, ExposureConfiguration configuration, String token)
```

DailySummary ExposureWindow

```
provideDiagnosisKeys(List<File> keyFiles)
provideDiagnosisKeys(DiagnosisKeyFileProvider provider)
```

https://developers.google.com/android/reference/com/google/android/gms/nearby/exposurenotification/ExposureNotificationClient

ExposureWindow modeへの切り換え（iOS）

アプリの設定ファイル（info.plist）で切り換える。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/
PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>ENAPIVersion</key>
    <integer>2</integer>
</dict>
</plist>
```

https://developer.apple.com/documentation/exposurenotification/enmanager/3650384-detectexposures

ExposureWindowの取得（Android）

ExposureSummary ExposureInformation

```
getExposureSummary(String token)
getExposureInformation(String token)
```

DailySummary ExposureWindow

```
getDailySummaries(DailySummariesConfig dailySummariesConfig)
getExposureWindows()
```

https://developers.google.com/android/reference/com/google/android/gms/nearby/exposurenotification/ExposureNotificationClient

ExposureWindowの取得（iOS）

ExposureSummary ExposureInformation

```
getExposureInfo(summary: ENExposureDetectionSummary, userExplanation: String, )
```

DailySummary ExposureWindow

```
getExposureWindows(summary: ENExposureDetectionSummary, )
```

ENExposureDetectionSummaryはdetectExposureの実行結果として取得

```
detectExposure(configuration: ENExposureConfiguration, diagnosisKeyURLs: [URL], )
```

https://developer.apple.com/documentation/exposurenotification/enmanager

設定の取り扱い（Android）

ExposureConfigurationはdeprecated

```
provideDiagnosisKeys(List<File> keyFiles)
provideDiagnosisKeys(DiagnosisKeyFileProvider provider)
```

DailySummaries取得用にDailySummariesConfigが追加されている

```
getDailySummaries(DailySummariesConfig dailySummariesConfig)
```

設定の取り扱い（iOS）

引き続きENConfigurationを使う

```
detectExposure(configuration: ENExposureConfiguration, diagnosisKeyURLs: [URL], )
```

AndroidのDailySummariesConfigに対応する設定値がENConfigurationに追加されている。

ENConfiguration/ExposureConfigurationの取り扱い

Android

ExposureConfigurationは使わない

```
provideDiagnosisKeys(List<File> keyFiles)
provideDiagnosisKeys(DiagnosisKeyFileProvider provider)
```

iOS

ENConfigurationを使う

```
detectExposure(configuration: ENExposureConfiguration, diagnosisKeyURLs: [URL], )
```

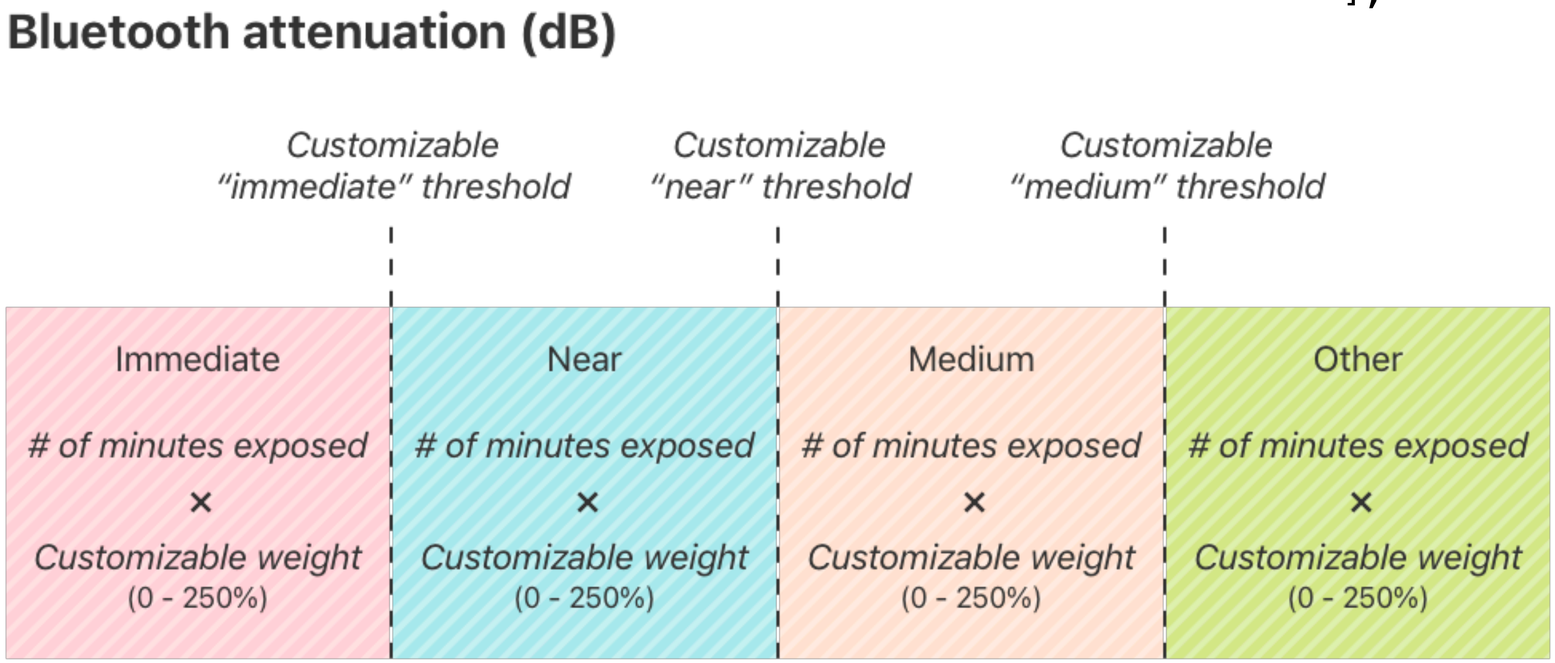
接触判定の設定

Legacy-v1の設定は再利用できないので、再度設定を設計し直す必要がある。

https://developer.apple.com/documentation/exposurenotification/enexposureconfiguration

Attenuation Duration Threshold

```
"apple_exposure_config_v2": {
 "attenuation_duration_thresholds": [
    50,
    70,
    90
  ],
```



https://developer.apple.com/documentation/exposurenotification/enexposureconfiguration

AndroidとiOSで設定可能な値の範囲が違う

値の範囲が異なる項目があるので注意が必要。

```
"infectiousness_weights": {
  "High": 1.0,
  "Standard": 1.0,
  "None": 1.0
},
"report_type_weights": {
  "ConfirmedClinicalDiagnosis": 1.0,
  "ConfirmedTest": 1.0,
  "SelfReport": 1.0,
  "Recursive": 1.0,
  "Revoked": 1.0,
  "Unknown": 1.0
}
```

1.0 - 2.5 (Android)

```
"infectiousness_high_weight": 100.0,
"infectiousness_standard_weight": 100.0,
"report_type_confirmed_clinical_diagnosis_weight": 100.0,
"report_type_confirmed_test_weight": 100.0,
"report_type_recursive_weight": 100.0,
"report_type_self_reported_weight": 100.0,
```

100 - 250 (iOS)

設定値の調整にあたって

COCOAをExposureWindowに対応する前に、Cappuccinoなどを使って、あらかじめ設定値を詰めておく。

設定の調整（確認）は次の順序で行う（陽性者端末 - 接触確認端末）。

- iOS - iOS
- Android - iOS
- Android - Android

iOSの場合、デバッグモードであれば同じ診断キーを使って設定を変えて何度でも手軽に接触確認が試行できる。

質疑応答