Writing OPass in KMM

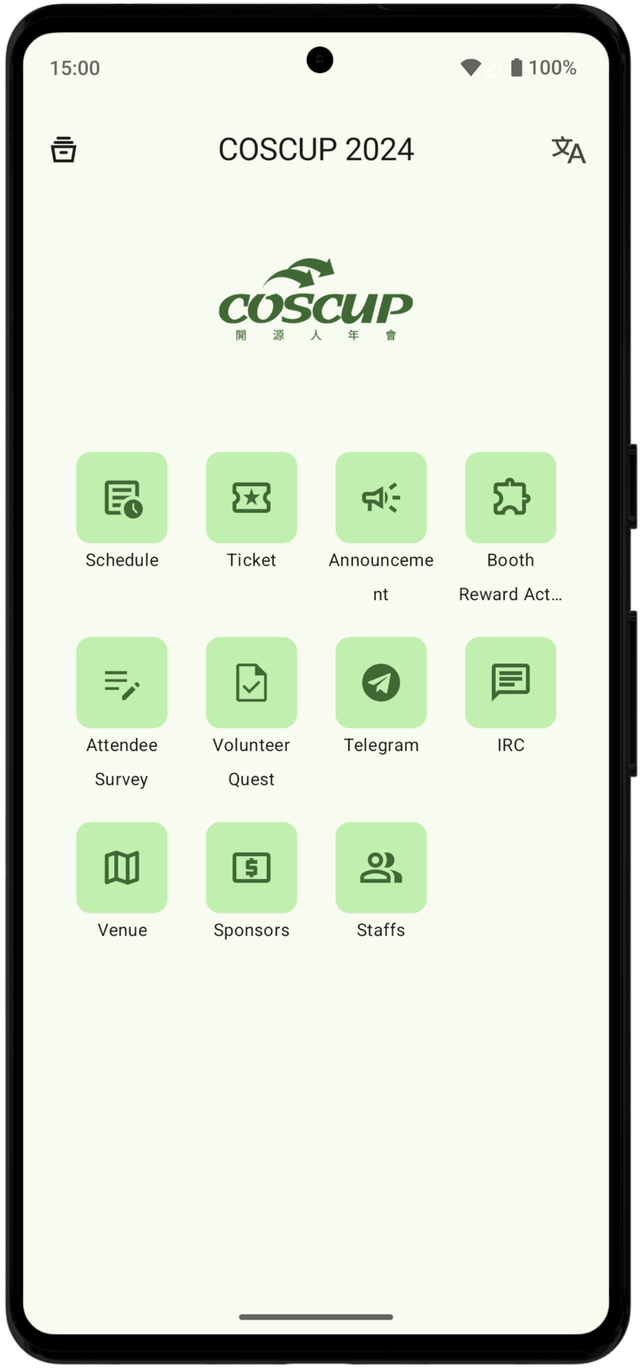
Adventures of an Android Developer with multi-platform development

Aayush Gupta

Development

Development

- During COSCUP 2024, discussed to rewrite Android app in compose with Material 3
- Want to learn KMM too
- Decided to rewrite in KMM while keeping UI native and sharing backend code



```
aayush@Aayushs-MacBook-Pro.local:~/StudioProjects/CCIP-KMP (main) $ tree -L 1 -I 'build|local.properties'
.
├── LICENSES
├── README.md
├── REUSE.toml
├── androidApp
├── build.gradle.kts
├── gradle
├── gradle.properties
├── gradlew
├── gradlew.bat
├── iosApp
├── settings.gradle.kts
└── shared
    ├── build.gradle.kts
    ├── consumer-rules.pro
    └── src
        ├── androidMain
        ├── commonMain
        └── iosMain

10 directories, 9 files
```

Development

- Schedule requires storing fetching data from network and storing it locally
- Network calls are done using ktor library that uses okhttp and darwin under the hood
- Database management is done using sqldelight library

commonMain

```
-- SPDX-FileCopyrightText: 2024 OPass
-- SPDX-License-Identifier: GPL-3.0-only

-- Schmea defination
-- -----------------

PRAGMA user_version = 1;

-- Create event table
CREATE TABLE EventTable (
    id TEXT PRIMARY KEY NOT NULL,
    logoUrl TEXT NOT NULL,
    nameEn TEXT NOT NULL,
    nameZh TEXT NOT NULL
);

-- Named queries for event table

selectAllEvents:
SELECT * FROM EventTable;

deleteAllEvents:
DELETE FROM EventTable;

insertEvent:
INSERT INTO EventTable (id, logoUrl, nameEn, nameZh) VALUES (:id, :logoUrl, :nameEn, :nameZh);
```

commonMain

```
internal class OPassDatabaseHelper {

    companion object {
        const val FILE_NAME = "opass.db"
    }

    private val database = OPassDatabase(DriverFactory.createDriver())
    private val dbQuery = database.oPassDatabaseQueries

    suspend fun getAllEvents(): List<Event> {
        return withContext(Dispatchers.IO) {
            dbQuery.selectAllEvents().executeAsList().map { it.toEvent() }
        }
    }

    suspend fun addEvents(events: List<Event>) {
        withContext(Dispatchers.IO) {
            dbQuery.transaction {
                dbQuery.deleteAllEvents()
                events.forEach {
                    dbQuery.insertEvent(
                        id = it.id,
                        logoUrl = it.logoUrl,
                        nameEn = it._name.en,
                        nameZh = it._name.zh
                    )
                }
            }
        }
    }
}
```

commonMain

```
/*
 * SPDX-FileCopyrightText: 2024 OPass
 * SPDX-License-Identifier: GPL-3.0-only
 */

package app.opass.ccip.database

import app.cash.sqldelight.db.SqlDriver

internal expect object DriverFactory {
    fun createDriver(): SqlDriver
}
```

androidMain

```kotlin
/*
 * SPDX-FileCopyrightText: 2024 OPass
 * SPDX-License-Identifier: GPL-3.0-only
 */

package app.opass.ccip.database

import app.cash.sqldelight.async.coroutines.synchronous
import app.cash.sqldelight.db.SqlDriver
import app.cash.sqldelight.driver.android.AndroidSqliteDriver
import app.opass.ccip.database.OPassDatabaseHelper.Companion.FILE_NAME
import utils.AppContextWrapper

internal actual object DriverFactory {
    actual fun createDriver(): SqlDriver {
        return AndroidSqliteDriver(
            OPassDatabase.Schema.synchronous(),
            AppContextWrapper.appContext!!,
            FILE_NAME
        )
    }
}
```

iosMain

```kotlin
/*
 * SPDX-FileCopyrightText: 2024 OPass
 * SPDX-License-Identifier: GPL-3.0-only
 */

package app.opass.ccip.database

import app.cash.sqldelight.async.coroutines.synchronous
import app.cash.sqldelight.db.SqlDriver
import app.cash.sqldelight.driver.native.NativeSqliteDriver
import app.opass.ccip.database.OPassDatabaseHelper.Companion.FILE_NAME

internal actual object DriverFactory {
    actual fun createDriver(): SqlDriver {
        return NativeSqliteDriver(OPassDatabase.Schema.synchronous(), FILE_NAME)
    }
}
```

commonMain

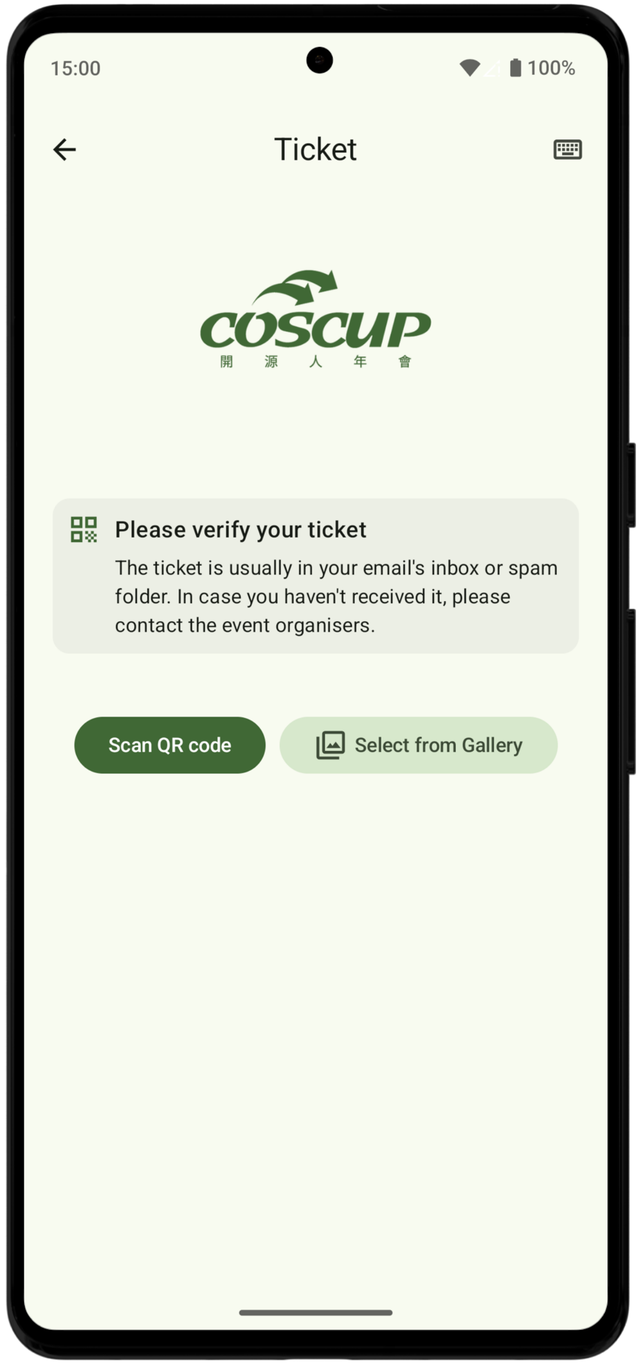
```kotlin
import app.opass.ccip.network.models.event.Event
import io.ktor.client.HttpClient
import io.ktor.client.call.body
import io.ktor.client.plugins.contentnegotiation.ContentNegotiation
import io.ktor.client.plugins.defaultRequest
import io.ktor.client.request.get
import io.ktor.serialization.kotlinx.json.json
import kotlinx.serialization.json.Json

internal class PortalClient {

    private val BASE_URL = "https://portal.opass.app"
    private val json = Json {
        prettyPrint = true
        ignoreUnknownKeys = true
        coerceInputValues = true
    }

    private val client = HttpClient {
        defaultRequest { url(BASE_URL) }
        install(ContentNegotiation) { json(json) }
    }

    suspend fun getEvents(): List<Event> {
        return client.get("/events/").body()
    }
}
```

commonMain

```kotlin
/**
 * Helper class to interact with OPass portal
 */
class PortalHelper {

    private val dbHelper = OPassDatabaseHelper()
    private val client = PortalClient()

    /**
     * Fetches list of [Event] from OPass portal
     * @param forceReload Whether to ignore cache, false by default
     */
    suspend fun getEvents(forceReload: Boolean = false): List<Event> {
        val cachedEvents = dbHelper.getAllEvents()
        return if (cachedEvents.isNotEmpty() && !forceReload) {
            cachedEvents
        } else {
            client.getEvents().also { dbHelper.addEvents(it) }
        }
    }
}
```

Development

- Ticket verification requires scanning QR codes
- Android provides CameraX library to work with camera
- For reading and writing QR, ZXing is a good FOSS library
- ZXing’s CPP port has wrapper for different languages including kotlin/native (snapshot)



androidApp

```kotlin
@Module
@InstallIn(SingletonComponent::class)
object CommonModule {

    @Provides
    @Singleton
    fun provideBarcodeReaderInstance(): BarcodeReader {
        return BarcodeReader().apply {
            options.tryRotate = true
            options.formats = setOf(BarcodeReader.Format.QR_CODE)
        }
    }

    @Provides
    @Singleton
    fun providesBackgroundExecutor(): ExecutorService {
        return Executors.newSingleThreadExecutor()
    }
}
```

androidApp

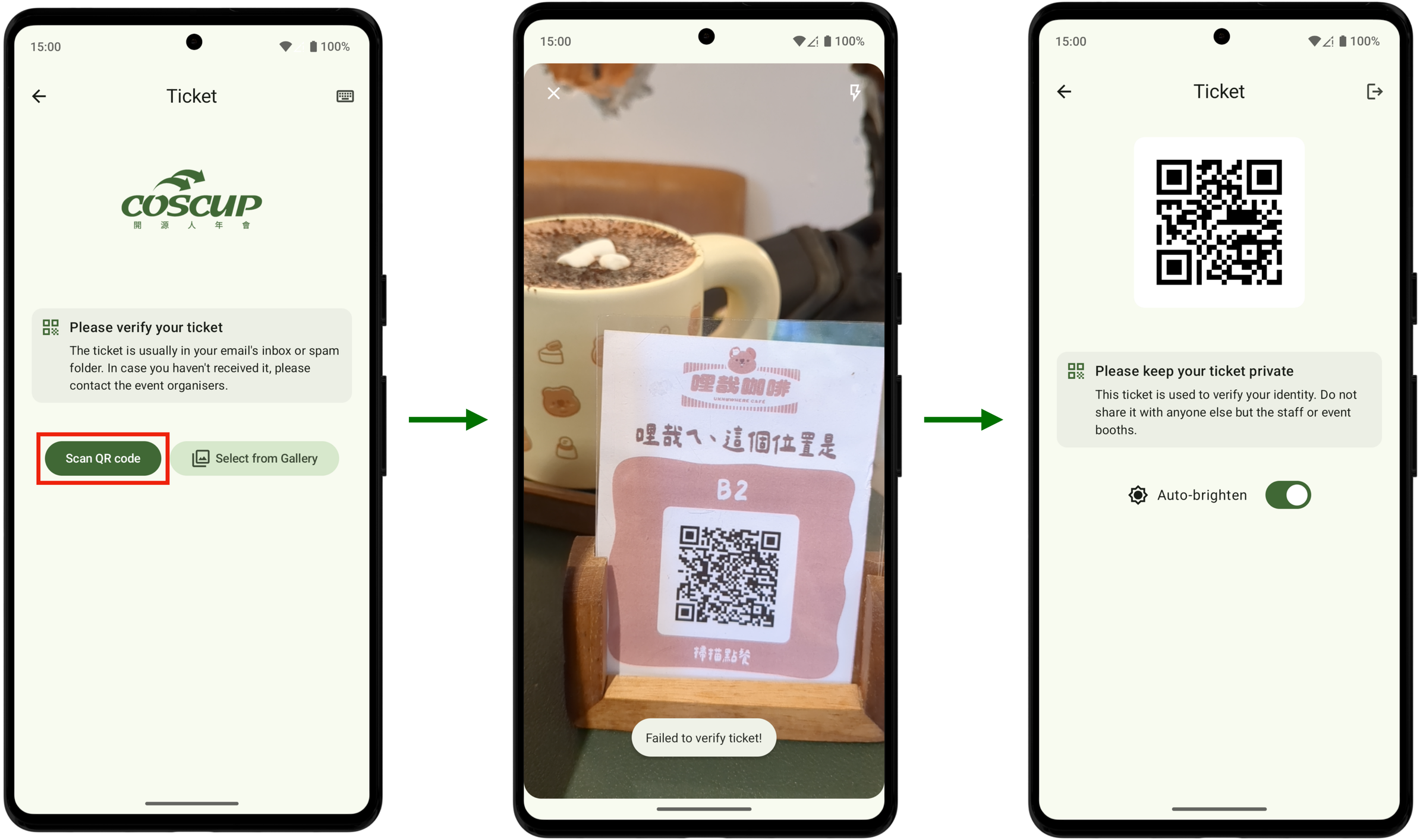
```
private val _surfaceRequest = MutableStateFlow<SurfaceRequest?>(null)
val surfaceRequest = _surfaceRequest.asStateFlow()

private lateinit var cameraControl: CameraControl
private val cameraPreview = Preview.Builder().build().apply {
    setSurfaceProvider { newSurfaceRequest ->
        _surfaceRequest.update { newSurfaceRequest }
    }
}

private val imageAnalysis = ImageAnalysis.Builder()
    .setBackpressureStrategy(ImageAnalysis.STRATEGY_KEEP_ONLY_LATEST)
    .build().apply {
        setAnalyzer(executorService) { imageProxy ->
            imageProxy.use { input ->
                barcodeReader.read(input).firstOrNull()?.text?.let { token ->
                    makeOneShotVibration()
                    runBlocking { getAttendee(eventId, token) }

                    // Avoid scanning the QR multiple times
                    Thread.sleep(2000)
                }
            }
        }
    }
```

androidApp

```
fun bindToCamera(lifecycleOwner: LifecycleOwner) {
    viewModelScope.launch {
        val processCameraProvider = ProcessCameraProvider.awaitInstance(context)
        processCameraProvider.bindToLifecycle(
            lifecycleOwner, DEFAULT_BACK_CAMERA, imageAnalysis, cameraPreview
        ).also {
            cameraControl = it.cameraControl
        }

        try { awaitCancellation() } finally { processCameraProvider.unbindAll() }
    }
}

fun toggleFlash(on: Boolean) {
    cameraControl.enableTorch(on)
}
```

androidApp

```
val surfaceRequest by viewModel.surfaceRequest.collectAsStateWithLifecycle()

surfaceRequest?.let { CameraXViewfinder(surfaceRequest = it) }
```


15:00
100%
Ticket
COSCUP
開 源 人 年 會
Please verify your ticket
The ticket is usually in your email's inbox or spam folder. In case you haven't received it, please contact the event organisers.
Scan QR code
Select from Gallery
15:00
100%
哩哉咖啡
UKNOWHERE CAFE
哩哉ㄟ、這個位置是
B2
Failed to verify ticket!
15:00
100%
Ticket
Please keep your ticket private
This ticket is used to verify your identity. Do not share it with anyone else but the staff or event booths.
Auto-brighten

Licensing

Licensing

- OPass uses code from various sources with different licenses and copyright
- REUSE makes it easy to maintain license and copyright information
- System Package Data Exchange (SPDX) is a freely available international open standard
- REUSE provides a easy to use tool to check and adapt recommendations

```
aayush@Aayushs-MacBook-Pro.local:~/StudioProjects/CCIP-KMP (main) $ reuse lint
# MISSING COPYRIGHT AND LICENSING INFORMATION

The following files have no copyright and licensing information:
* iosApp/OPass/Common/BlurView.swift
* iosApp/OPass/Common/CachedAsyncImage.swift
* iosApp/OPass/Common/CenterLabelStyle.swift
* iosApp/OPass/Common/SafariView.swift

# SUMMARY

* Bad licenses: 0
* Deprecated licenses: 0
* Licenses without file extension: 0
* Missing licenses: 0
* Unused licenses: 0
* Used licenses: GPL-3.0-only, GPL-2.0-or-later, Apache-2.0
* Read errors: 0
* Files with copyright information: 4 / 202
* Files with license information: 4 / 202

Unfortunately, your project is not compliant with version 3.3 of the REUSE Specification :-(


# RECOMMENDATIONS

* Fix missing copyright/licensing information: For one or more files, the tool
  cannot find copyright and/or licensing information. You typically do this by
  adding 'SPDX-FileCopyrightText' and 'SPDX-License-Identifier' tags to each
  file. The tutorial explains additional ways to do this:
  <https://reuse.software/tutorial/>
```

```kotlin
/*
 * SPDX-FileCopyrightText: 2025 OPass
 * SPDX-License-Identifier: GPL-3.0-only
 */

package app.opass.ccip.android.ui.extensions

import androidx.compose.runtime.Composable
import androidx.compose.runtime.remember
import androidx.hilt.navigation.compose.hiltViewModel
import androidx.lifecycle.ViewModel
import androidx.navigation.NavBackStackEntry
import androidx.navigation.NavController

/**
 * Gets viewModel from the parent composable
 */
@Composable
inline fun <reified T : ViewModel> NavBackStackEntry.sharedViewModel(navController: NavController): T {
    val parentEntry = remember(this) { navController.getBackStackEntry(this.destination.route!!) }
    return hiltViewModel<T>(parentEntry)
}
```

```toml
# SPDX-FileCopyrightText: 2024 OPass
# SPDX-License-Identifier: GPL-3.0-only

# This file describes the licensing and copyright situation for files that
# cannot be annotated directly, for example because of being simply
# uncommentable. Unless this is the case, a file should be annotated directly.
#
# This follows the REUSE specification: https://reuse.software/spec-3.2/#reusetoml

version = 1

[[annotations]]
path = [
    "README.md",
    "gradle/wrapper/*",
    "androidApp/src/main/**/*.png",
    "androidApp/src/main/res/resources.properties"
]
precedence = "closest"
SPDX-FileCopyrightText = "2024 OPass"
SPDX-License-Identifier = "GPL-3.0-only"
```

Localisation

Localisation

Translating OPass

- Currently being translated into 9 languages through Weblate
- Android app uses per-app language feature with AppCompat APIs
- Android doesn’t supports all locales e.g. Hokkien

Get involved in **OPass**

Hello and thank you for your interest — OPass is being translated using Weblate, a web tool designed to ease translating for both developers and translators.

182	9	72.3%
STRINGS	LANGUAGES	TRANSLATED

The translation project for OPass currently contains **182 strings** for translation. It is being translated into **9 languages**. Overall, these translations are **72.3% complete**. If you would like to contribute to translation of OPass, you need to register on this server.

View project languages

OPass

translated 72%

Component	Translated	Unfinished	Unfinished words	Unfinished characters	Untranslated	Checks	Suggestions	Comments
CCIP-Android GPL-3.0-or-later	78%	220	888	5,234	215	23		
CCIP-KMP-Android GPL-3.0-only	59%	198	855	4,948	195	8		
Glossary OPass GPL-3.0-or-later	✓							

Translation

Source change

Search for events

English

Key search_event

Search for events

Chinese (Traditional Han script)

NBS … 「 」 『 』 (-) - – —

搜尋活動

Needs editing

4/170 · 17

Save and continue | Save and stay | Suggest | Skip

Nearby strings 8 | Similar keys 10 | Comments | Automatic suggestions | Other languages 8 | History

Key	English	Chinese (Traditional Han script)	Actions
app_name	OPass		
select_event	Select event	選擇活動	
search_event	Search for events	搜尋活動	
switch_language	Switch language	切換語言	

Glossary

English | Chinese (Traditional Han script)

No related strings found in the glossary.

Add term to glossary

String information

146965470

Screenshot context
No screenshot currently associated.
Add screenshot

Explanation
No explanation currently provided.

Key
search_event

Labels
No labels currently set.

Flags
java-printf-format

String age
3 weeks ago

Translations update from Hosted Weblate #12

Edit Code

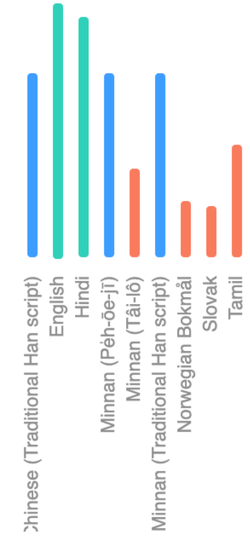
Merged **theimpulson** merged 2 commits into `CCIP-App:main` from `weblate:weblate-opass-ccip-kmp-android` last week

Conversation 0 Commits 2 Checks 1 Files changed 2 +28 −2

weblate commented last week

Translations update from Hosted Weblate for OPass/CCIP-KMP-Android.

Current translation status:



winstonsung added 2 commits last week

Reviewers
theimpulson ✓

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Successfully merging this pull request may close these issues.
None yet

Notifications Customize

Thoughts

Thoughts

- KMM provides ability to write code in native language of platform
- Some KMM libraries are locked to CMP limiting choices
- Namespace pollution is an issue with generated code
- Not all Kotlin code has counterpart in generated native code
- Weblate does not support new iOS string catalog format
- Sharing backend code in Kotlin while keeping UI code native seems like a perfect balance for a start

Thank You!