

KMP로 Figma 아이콘 동기화, 클릭 한 번이면 끝!



옥수환

NAVER 밴드

발표자 소개

- 네이버 밴드 앱 안드로이드 개발자
- '찰스의 안드로이드' 블로그 및 오픈 채팅 채널 운영
- 드로이드나이트 외 연사, 강의, 행사 기획 다수 경험



목차

1 백문이 불여일견, 예제로 보는 아이콘 동기화

2 Figma API 맛보기

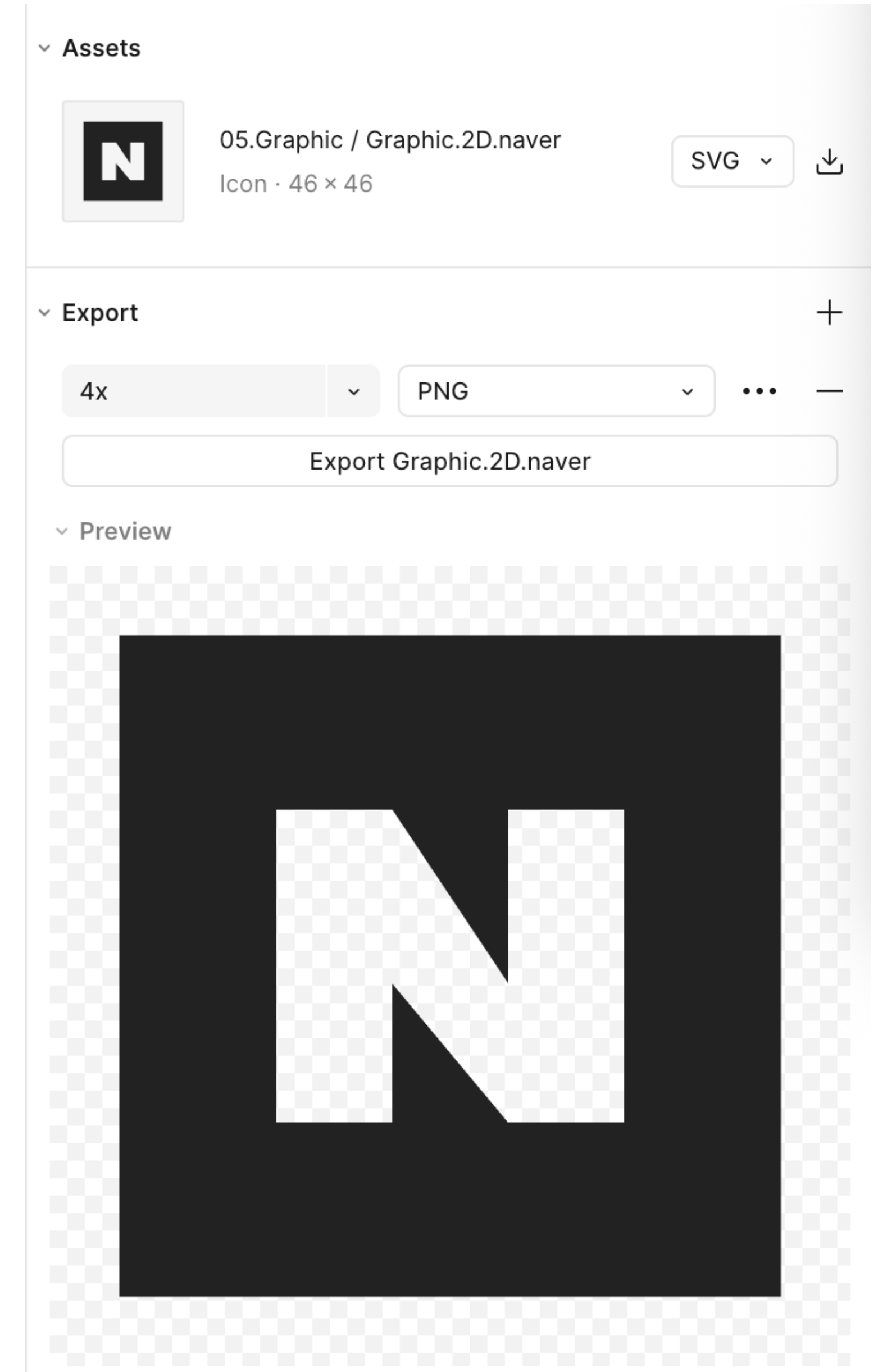
3 KMP로 아이콘 동기화 핵심 로직 만들기

4 빌드하기

1 백문이 불여일견, 예제 살펴보기

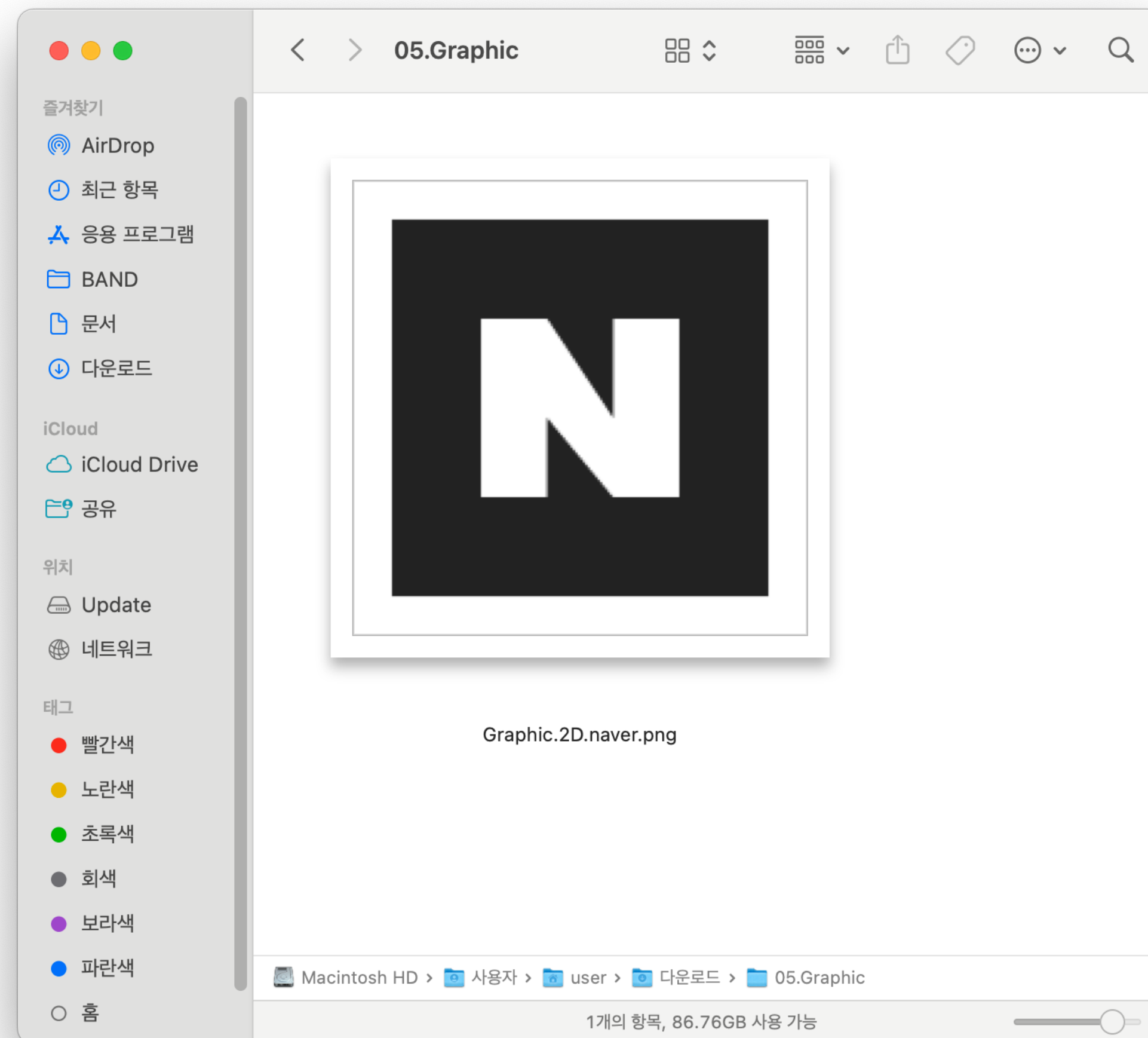
일반적인 아이콘 추가 방법

1. 아이콘을 다운로드 한다.



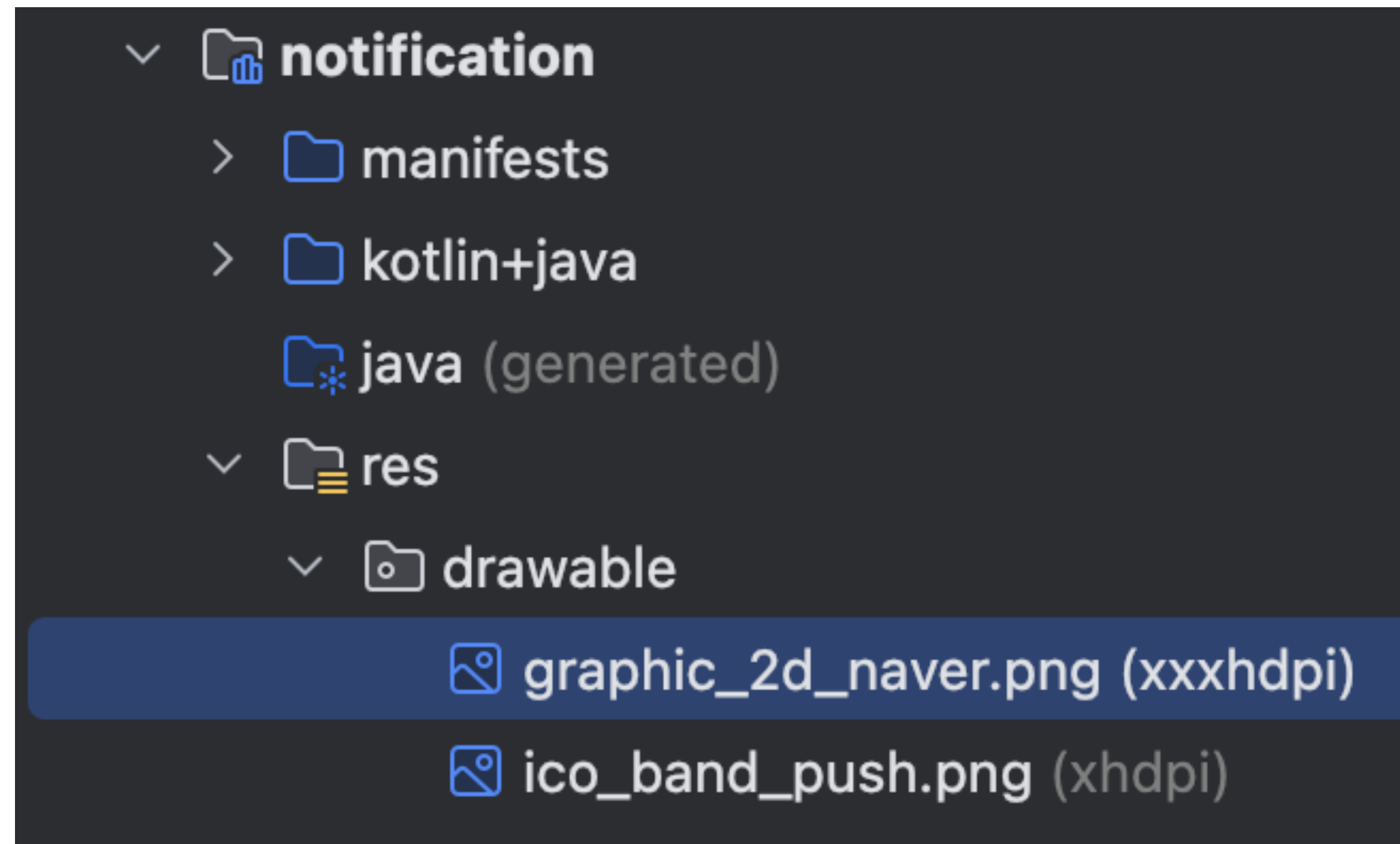
일반적인 아이콘 추가 방법

2. 다운로드 한 아이콘을 프로젝트로 이동한다.



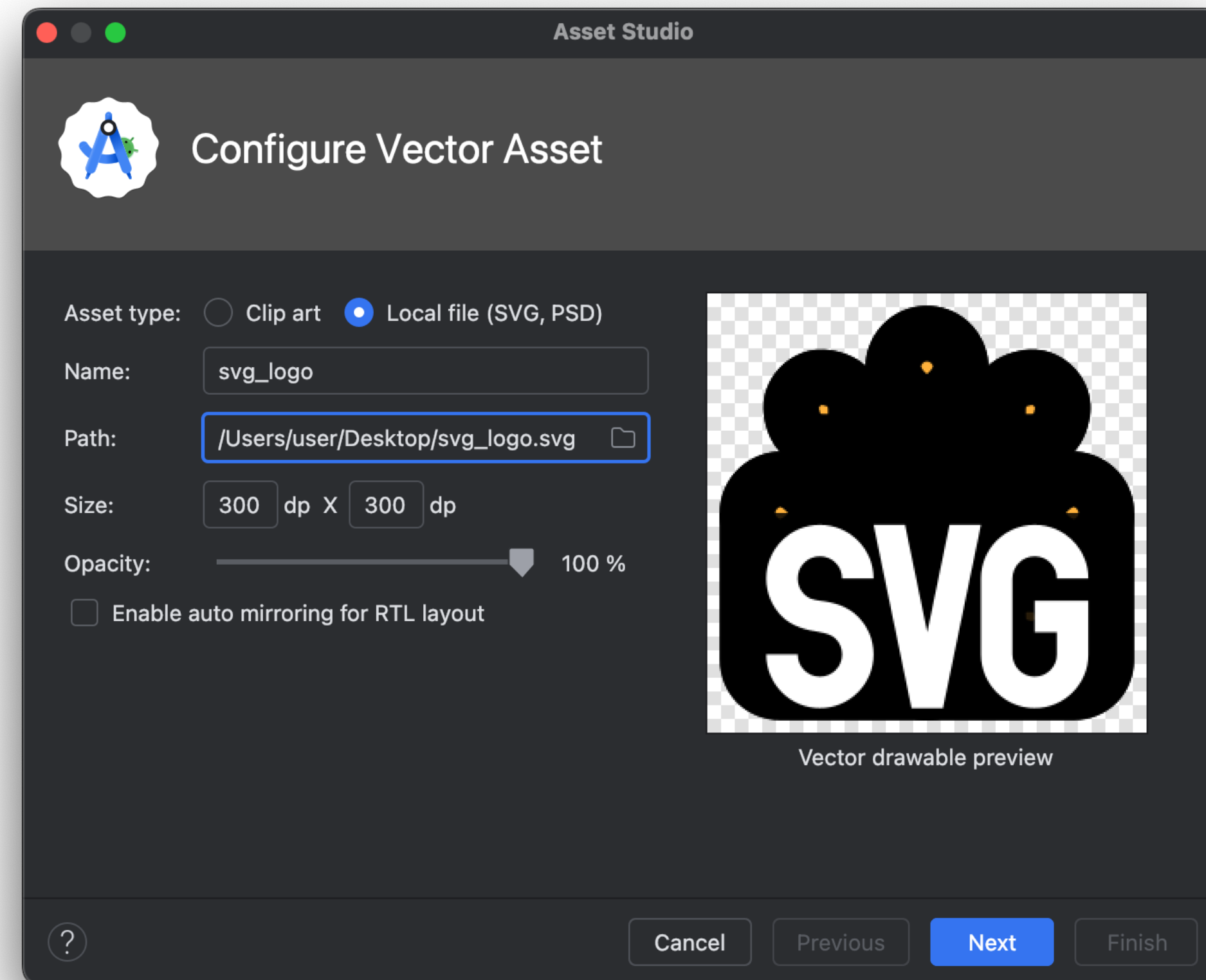
일반적인 아이콘 추가 방법

3. 다운로드 한 아이콘을 참조한다.

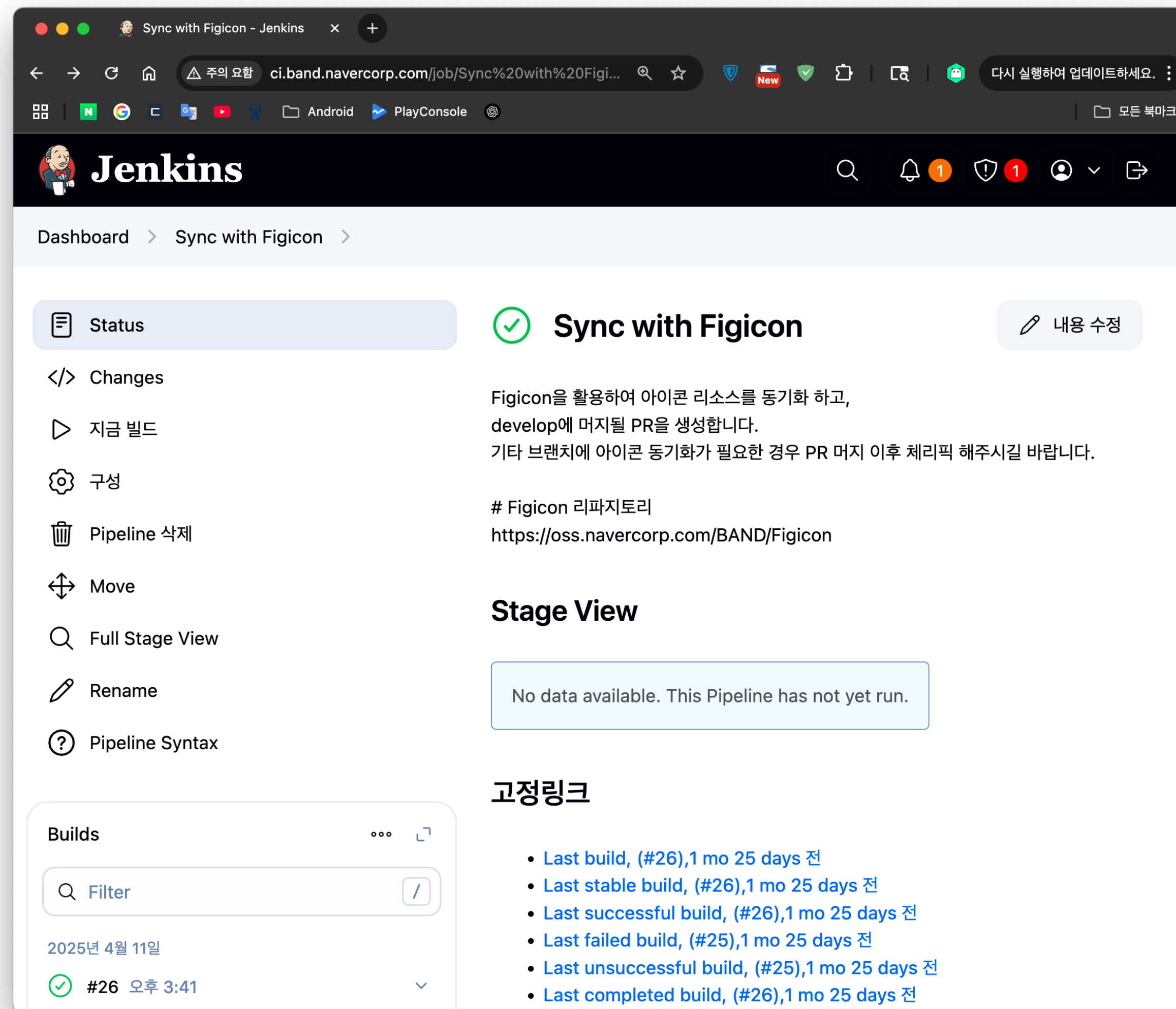


Asset Studio 활용

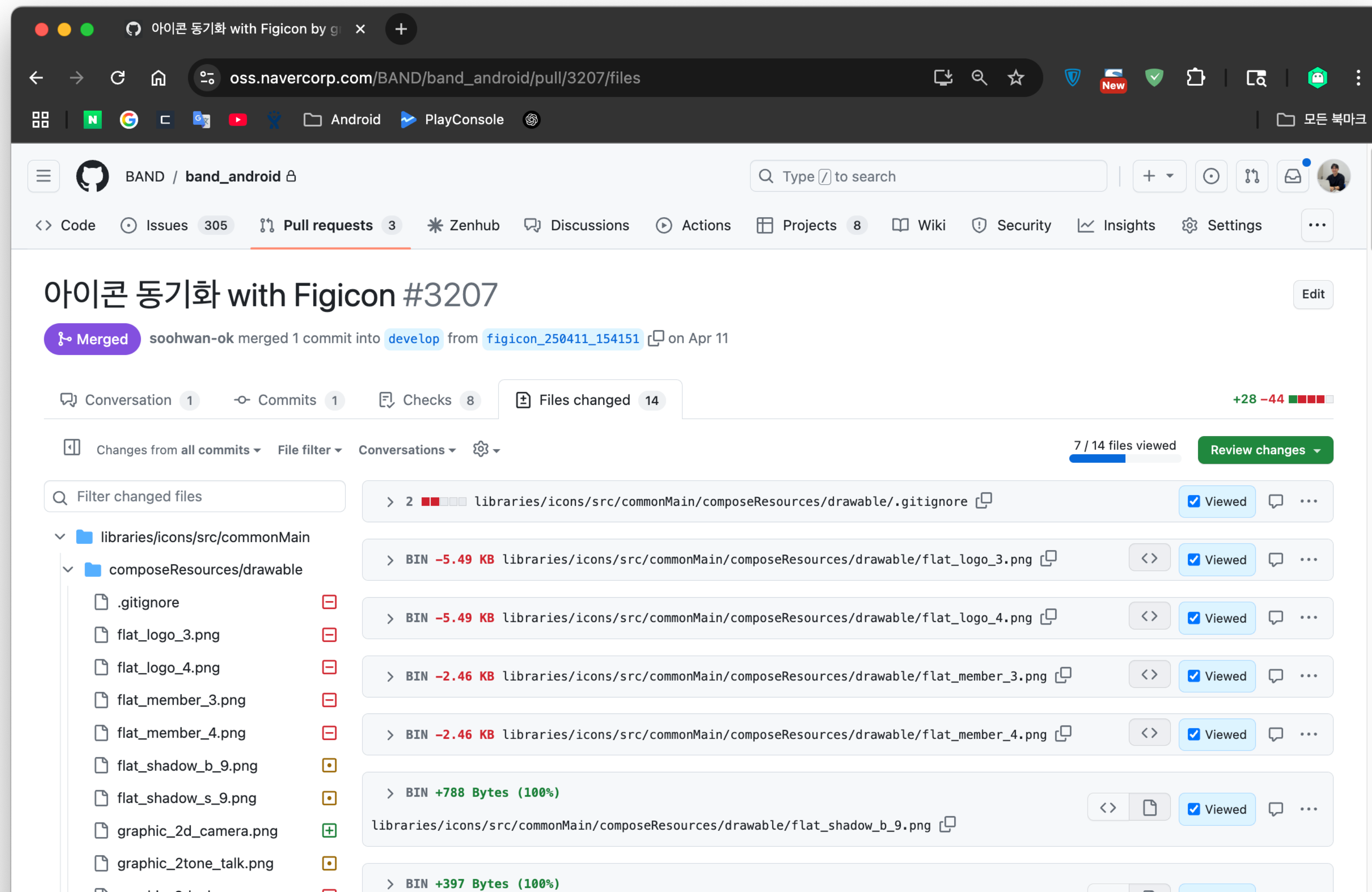
주로 SVG와 같은 벡터형식의 포맷을
프로젝트에 추가할 때 사용한다.



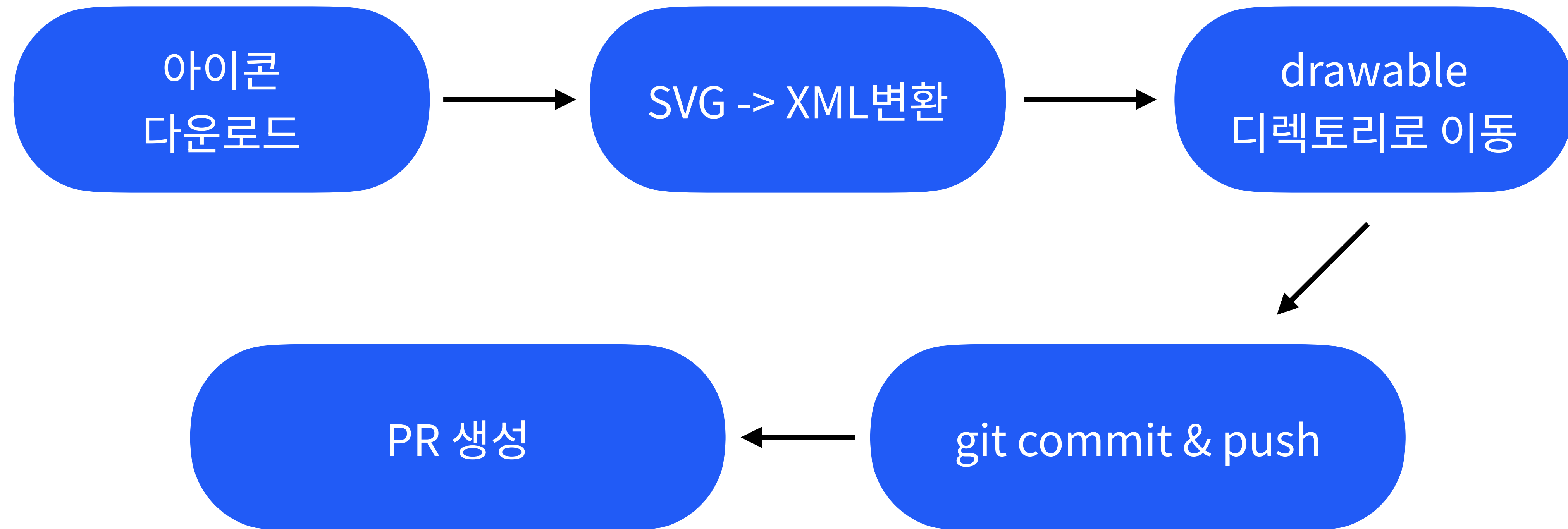
예제1 - 젠킨스로 아이콘 동기화



예제1 - 젠킨스로 아이콘 동기화



예제1 - 젠킨스로 아이콘 동기화



예제2- macOS 앱

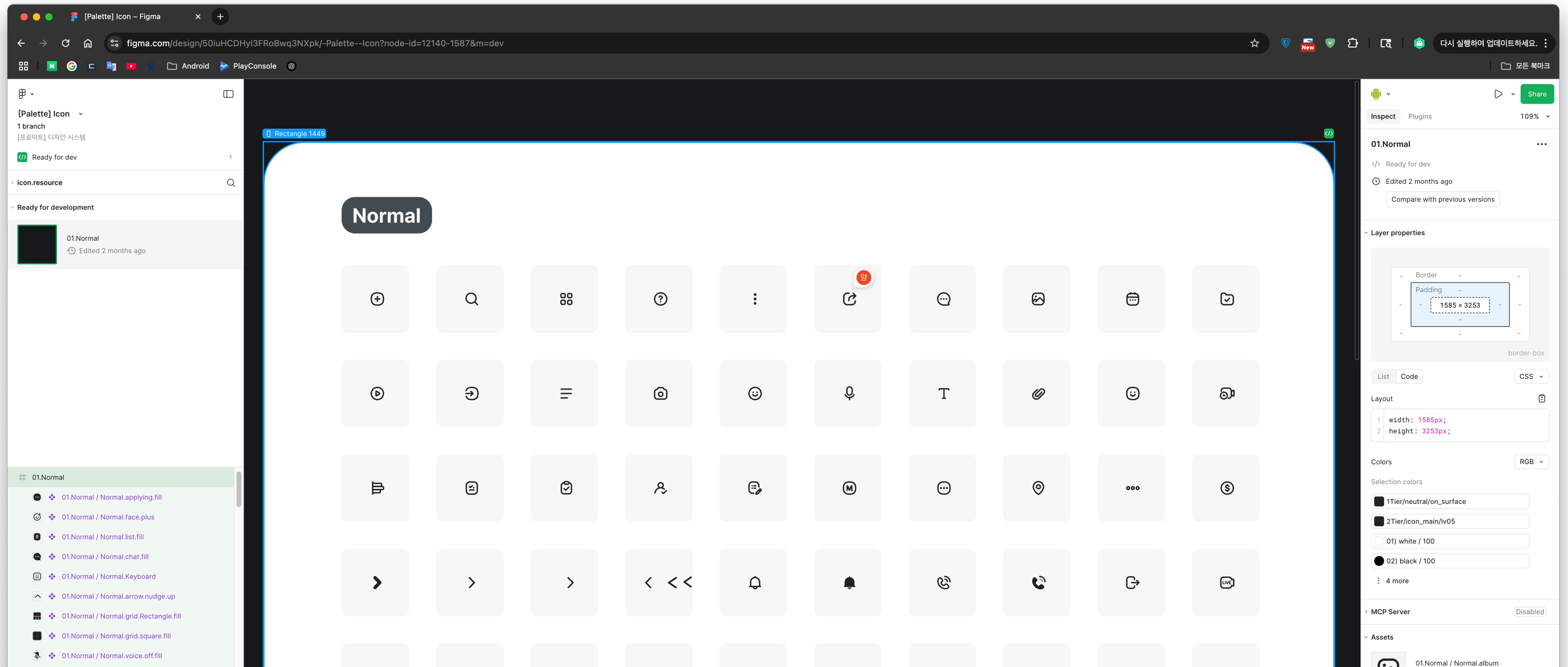
아이콘을 일괄 다운로드 받을 수 있는
macOS용 데스크탑용 애플리케이션 제공



2 Figma REST API 맛보기

피그마 파일 구조화

- * Figma REST API 응답에 영향을 주는 요소
- * 디자이너와 협의 필요

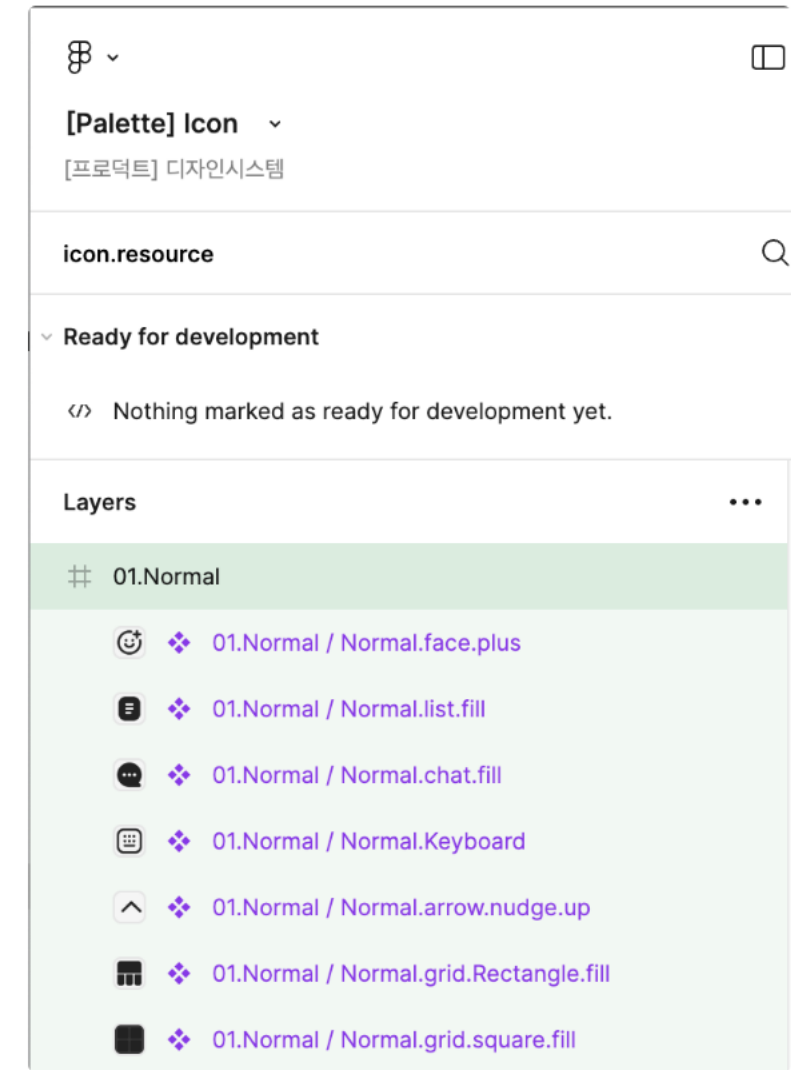


피그마 파일 구조화

- * Figma REST API 응답에 영향을 주는 요소
- * 디자이너와 협의 필요

Figma 아이콘 규칙

- 아이콘은 반드시 특정 프레임내 1-depth 이내에 속해 있어야 하며, 다음 이미지와 같이 구조화 되어 있어야 합니다.



- 각 프레임(예: Normal, Bar 등)은 고유의 ID를 갖습니다. 프레임을 추가 또는 변경하는 경우 ID가 변경되므로, 변경 사실을 개발자에게 알려야 합니다.
- 개별 아이콘의 이름은 다음과 같은 형식입니다.

```
형식 : ${카테고리명} / ${아이콘명}예) 01.Normal / Normal.face.plus

카테고리 : normal
아이콘명 : face_plus

최종 파일명 : normal_face_plus.svg
```

개별아이콘이 위와 같은 형식의 이름을 갖지 않는 경우 프로그램이 실행 중 중단되거나, 아이콘이 다운로드 되지 않을 수 있습니다.

Figma URL 형식

<https://www.figma.com/design/50iuHCDHyl3FRoBwq3NXpk/-Palette--Icon>

Figma URL 형식

<https://www.figma.com/design/50iuHCDHyl3FRoBwq3NXpk/-Palette--Icon>



파일 타입



파일 키



파일 이름

Figma URL 형식

<https://www.figma.com/design/50iuHCDHyl3FRoBwq3NXpk/-Palette--Icon>



파일 키

간단한 Figma REST API 테스트

[https://www.figma.com/
developers/api#get-files-endpoint](https://www.figma.com/developers/api#get-files-endpoint)

Try it out for yourself

file_key

version (optional)

ids (optional)

depth (optional)

geometry (optional)

plugin_data (optional)

branch_data (optional)

[+ Get personal access token](#) What's this?

Your cURL command

```
curl -H 'X-FIGMA-TOKEN: <personal access token>'  
'https://api.figma.com/v1/files/:file_key'
```

Submit API Request

간단한 Figma REST API 테스트

file_key와 access_token을 넣어
Figma 파일 정보를 가져올 수 있다.

Try it out for yourself

file_key 50iuHCD*****NXpk

version (optional)

ids (optional)

depth (optional)

geometry (optional)

plugin_data (optional)

branch_data (optional)

[+ Get personal access token](#) What's this?

↑ 클릭해서 토큰 얻기

Your cURL command

```
curl -H 'X-FIGMA-TOKEN: <personal access token>'  
'https://api.figma.com/v1/files/:file_key'
```

Submit API Request

간단한 Figma REST API 테스트

API 요청에 대한 응답으로
JSON 형식의 Figma 파일 정보를 얻을 수 있다.

Your cURL command

```
curl -H 'X-FIGMA-TOKEN:  
figd_E0FK2tdHUd9z5mBTeh3y2Sjtm1p4W2SNqkrUV4Bx'  
'https://api.figma.com/v1/files/50iuHCDHy13FRoBw  
q3NXpk'
```

Submit API Request

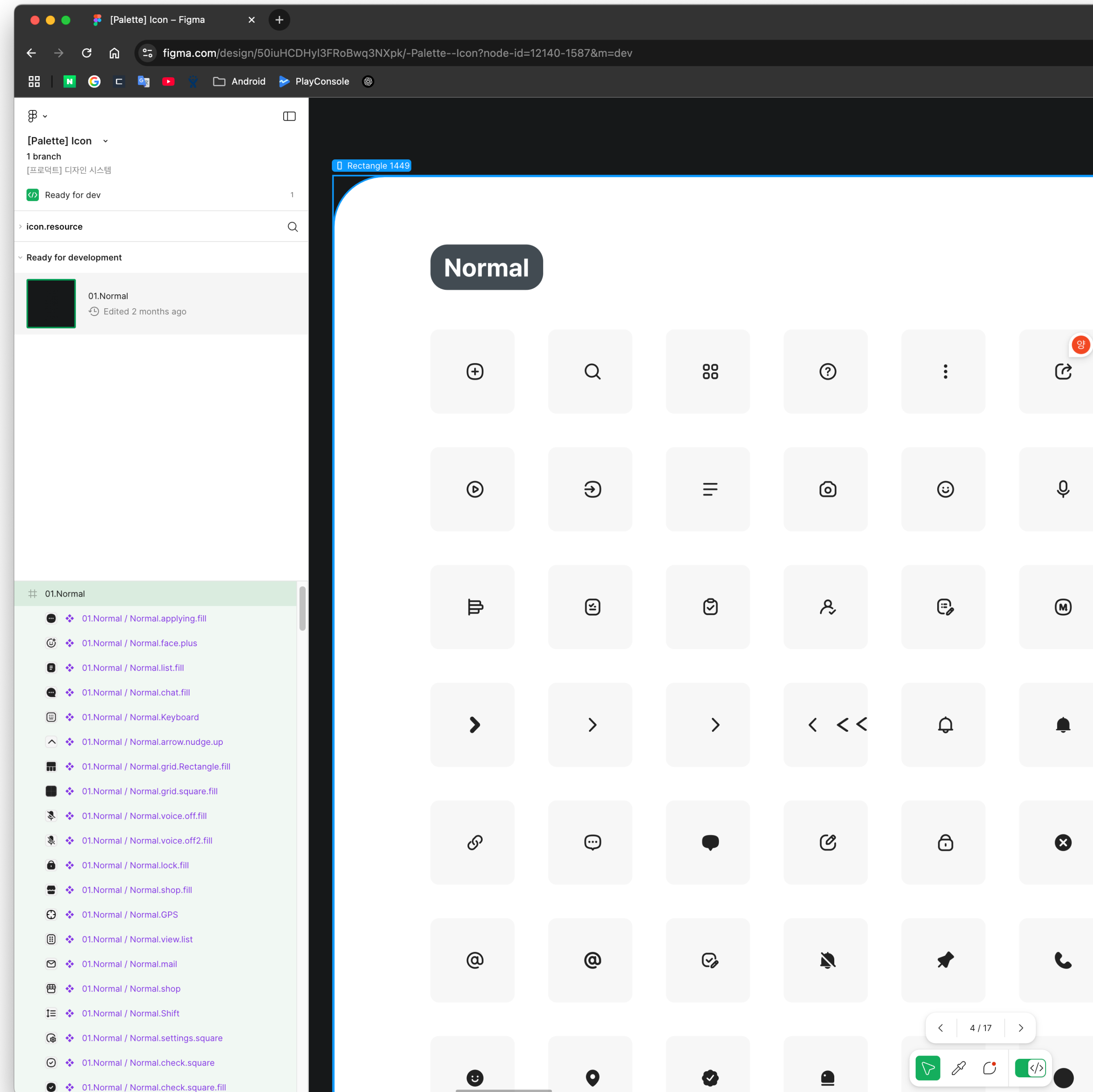
Copy to clipboard

```
{  
  "document": {  
    "id": "0:0",  
    "name": "Document",  
    "type": "DOCUMENT",  
    "scrollBehavior": "SCROLLS",  
    "children": [  
      {  
        "id": "2:33",  
        "name": "icon.resource",  
        "type": "CANVAS",  
        "scrollBehavior": "SCROLLS",  
        "children": [  
          {  
            "id": "1381:825",  
            "name": "# Create Band",  
            "type": "GROUP",  
            "scrollBehavior": "SCROLLS"
```

Figma에서 Node란?

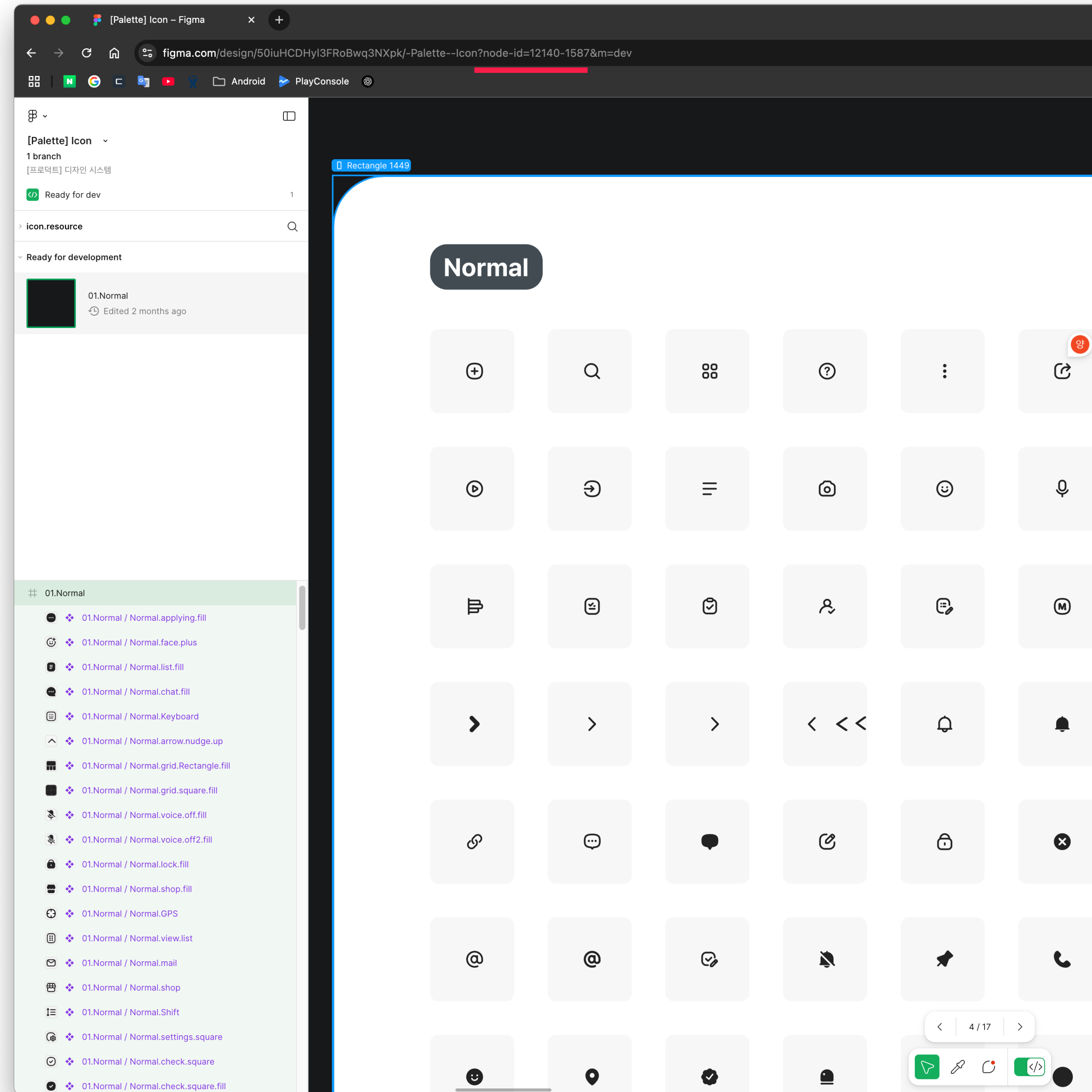
Figma 파일 내의 레이어 또는 객체.
모양, 텍스트, 컴포넌트까지 Figma 디자인
의 모든 요소는 노드로 표현된다.

* 어떠한 요소를 클릭하면 url에서 노드 아이디를 확인할 수 있다.



Figma에서 Node란?

요소(아이콘 또는 프레임 등)를 클릭하면,
브라우저 주소창에 node-id가 노출 됩니다.



Figma의 Node Id

.../50iuHCDHyl3FRoBwq3NXpk/-Palette--Icon?node-id=12580-1814



노드 아이디

Figma File REST API

(GET) https://api.figma.com/v1/files/파일키?ids=node_id

// 응답 샘플

```
{
  ...
  "components": {
    "13084:1482": { → node_id
      "key": "4e189da2203df00284cc38f43b0927c19c626c46",
      "name": "01.Normal / Normal.applying.fill",
      "description": "",
      "remote": false,
      "documentationLinks": []
    },
  },
}
```

Figma Image REST API

(GET) <https://api.figma.com/v1/images/파일키>

[벡터 아이콘인 경우]

<https://api.figma.com/v1/images/파일키?ids=노드아이디&format=svg>

[비트맵 아이콘인 경우]

<https://api.figma.com/v1/images/파일키?ids=노드아이디&format=png&scale=4>

Figma Image API

// 응답 샘플

```
{  
  "images": {  
    "13084:1482": "https://figma.amazonaws.com/images/6253c8fc-b981-b770e06bcdd0"  
  }  
}
```

Figma Image API

// 응답 샘플

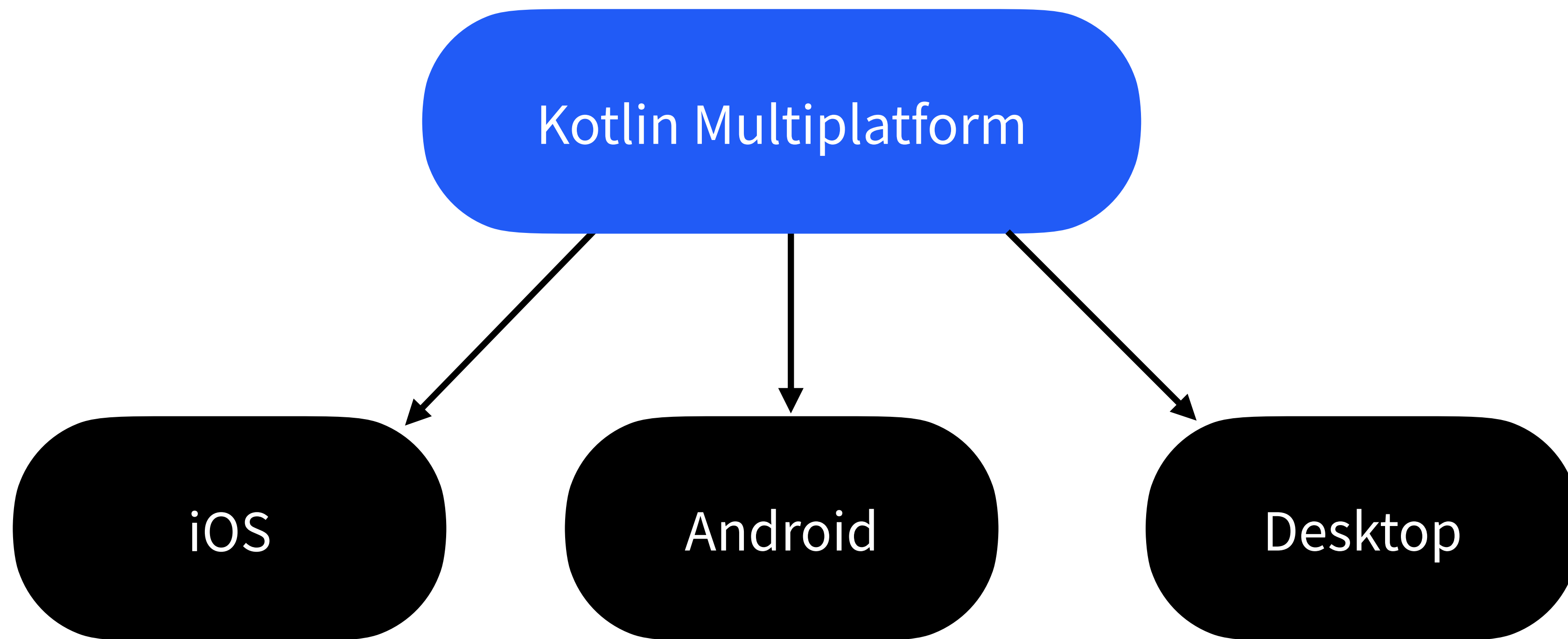
```
{  
  "images": {  
    "13084:1482": "https://figma.amazonaws.com/images/6253c8fc-b981-b770e06bcdd0"  
  }  
}
```

아이콘 노드 아이디

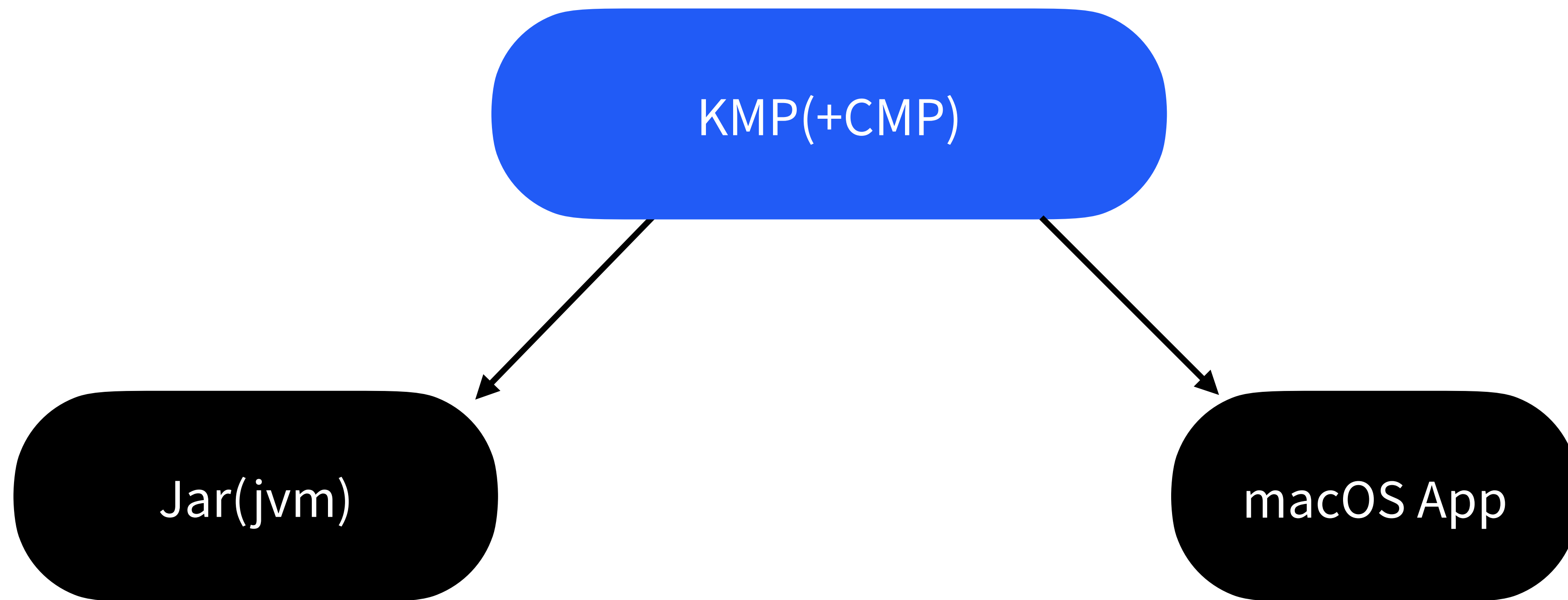
이미지 URL

3 KMP로 아이콘 동기화 핵심 로직 만들기

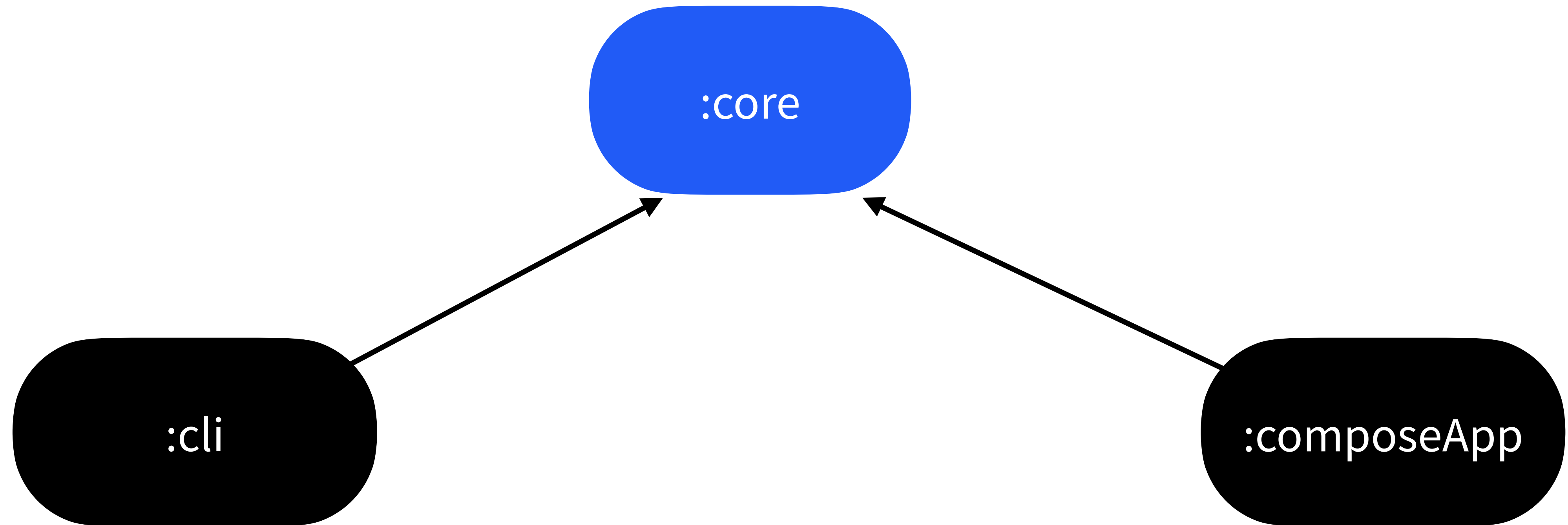
Why KMP?



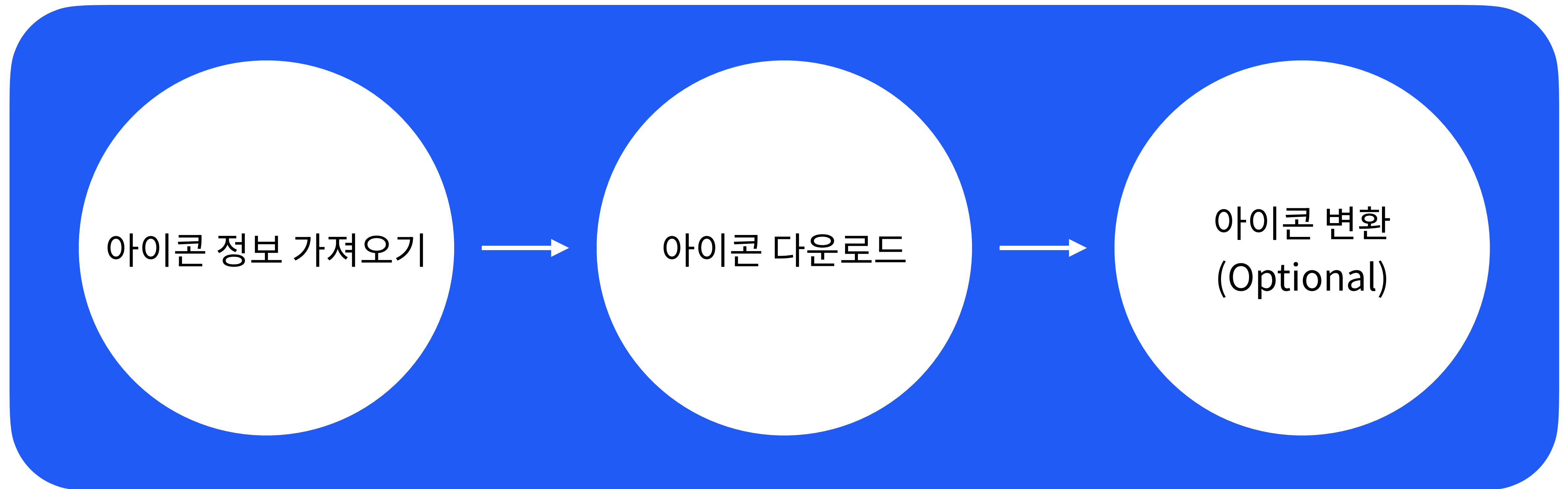
Why KMP?



프로젝트 모듈 구성



아이콘 동기화 프로세스 개요 (:core 모듈)



```
// 파일정보 가져오기
```

```
@Serializable
```

```
data class FileResponse(val components:Map<String, Component>) // File API 응답 모델
```

```
class RetrieveFileUseCase(private val client:HttpClient) { // Ktor Http Client
```

```
    suspend operator fun invoke(
```

```
        fileKey: String,
```

```
        ids: String
```

```
    ): FileResponse {
```

```
        return client.get("files/$fileKey") {
```

```
            parameter("ids", ids)
```

```
        }.body<FileResponse>()
```

```
    }
```

```
}
```

// 아이콘 정보 가져오기

@Serializable

data class ImageResponse(**val** images: Map<String, String?>) // Image API 응답 모델

```
class RetrievalconsUseCase(private val client:HttpClient) {  
    suspend operator fun invoke(  
        fileKey:String, // 파일키  
        ids: String, // 아이콘 노드 아이디 목록, 예)12345:12, 23423:354, 34342:123  
        format:String, // svg 또는 png  
        scale:Int? = null  
    ): ImageResponse {  
        return client.get("images/${fileKey}") {  
            parameter("ids", ids)  
            parameter("format", format)  
            if(scale!=null) parameter("scale", scale.toString())  
        }.body<ImageResponse>()  
    }  
}
```

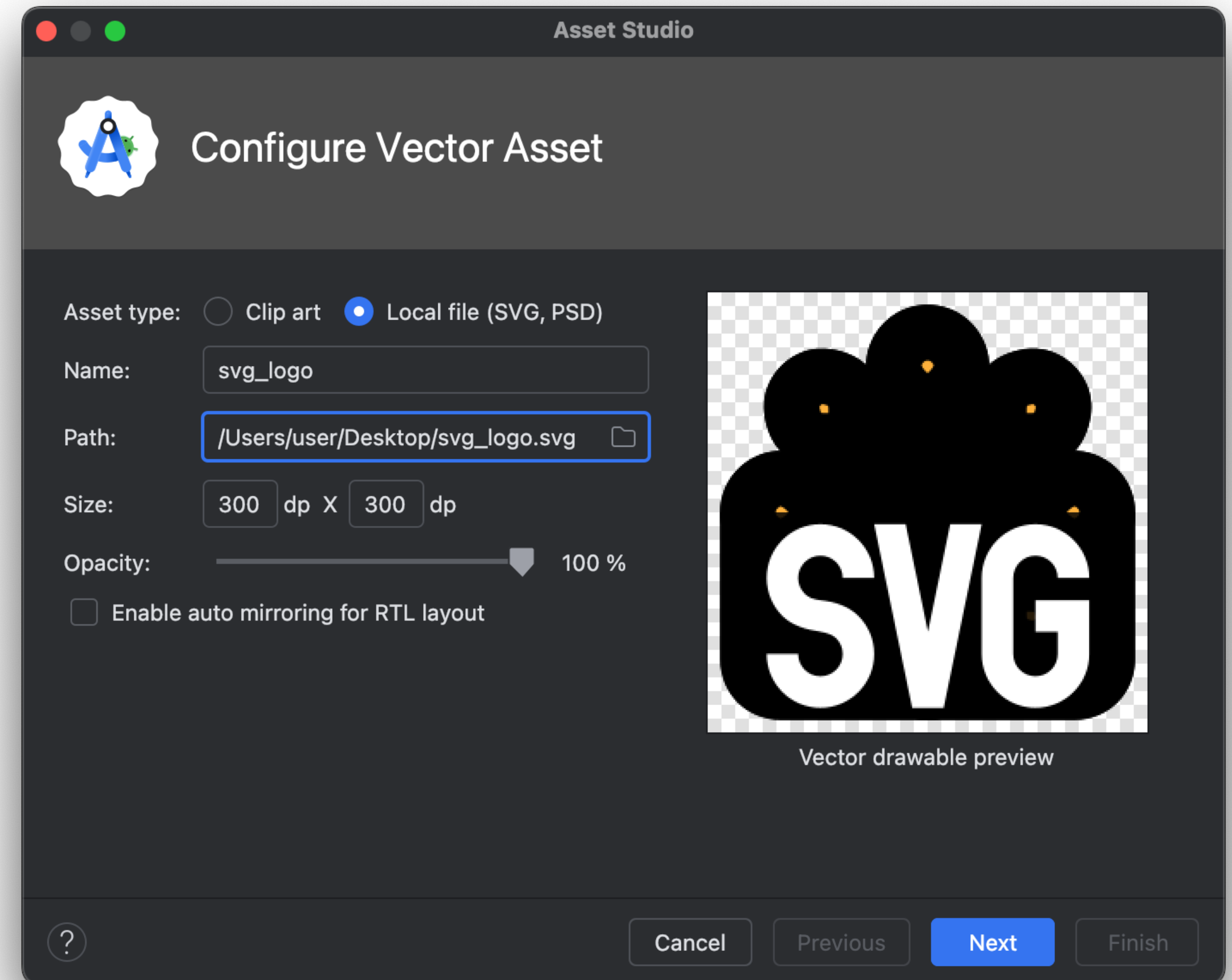
// 아이콘 다운로드

```
class DownloadIconsUseCase(
    private val client: HttpClient,
) {
    suspend operator fun invoke(
        imageResponse: ImageResponse,
    ) {
        imageResponse.images.forEach { entry ->
            val iconUrl = entry.value?: return // 아이콘 url
            val iconFile: File = ... // 다운로드 받을 경로

            client.get(iconUrl)
                .bodyAsChannel()
                .copyAndClose(iconFile.writeChannel()) // Ktor 클라이언트를 통해 아이콘 다운로드 받기
        }
    }
}
```

SVG 변환하기

안드로이드에서는 SVG 파일을 직접적으로 참조할 수 없으므로,
Vector Asset Studio가 생성한 XML 을
참조해야 한다.



Svg2Vector.java

1. ASOP에서 SVG를 VectorDrawable로 변환하는 Java 툴을 제공한다.
2. 직접 빌드하여 CLI 툴로 사용할 수 있다.
3. Svg2Vector.java가 포함된 코드는 Maven Central 및 Google Maven에서 공식적으로 배포되는 아티팩트로는 제공되지 않는다.

tools.base:sdk-common 의존하기 (비공식적/불완전)

// 최신 버전은 Google Maven에서 확인 가능
// 이 라이브러리는 공식적인 API가 아니며 언제든지 깨질 수 있습니다.
// 의존성 트리가 매우 크고 Android Studio의 내부 도구와 tightly coupled 되어 있습니다.

```
dependencies {  
    implementation("com.android.tools:sdk-common:31.7.2")  
}
```

// SVG 파일 XML로 변환하기

```
class ConvertSvg2VectorUseCase{

    operator fun invoke(
        srcPath: String, // SVG 파일 경로
        outputDir: String, // XML 파일이 저장될 디렉토리 경로
        fileName: String, // 변환될 XML 파일명
    ): String {
        val xmlFile = File(outputDir, "${fileName}.xml")
        val srcFile = File(srcPath)
        Svg2Vector.parseSvgToXml(
            java.nio.file.Path.of(srcFile.toURI()),
            xmlFile.outputStream()
        ) // SVG파일을 Android Vector 형식의 XML로 변환
        return xmlFile.absolutePath
    }

}
```

4 빌드하기 (JAR 및 macOS 앱)

애플리케이션 Build 하기

- CLI(Command Line Interface) 툴로 이용할 JAR 빌드
- GUI 툴로 이용할 macOS 데스크톱 앱 빌드

4-1 CLI 툴로 이용할 JAR 빌드

// JAR로 만들기 위해 메인 메서드 만들기

```
public class Runner {  
    public static void main(String[] args) {  
        // 여기에서 아이콘 동기화 프로세스를 수행  
    }  
}
```

// :cli 모듈의 build.gradle.kts

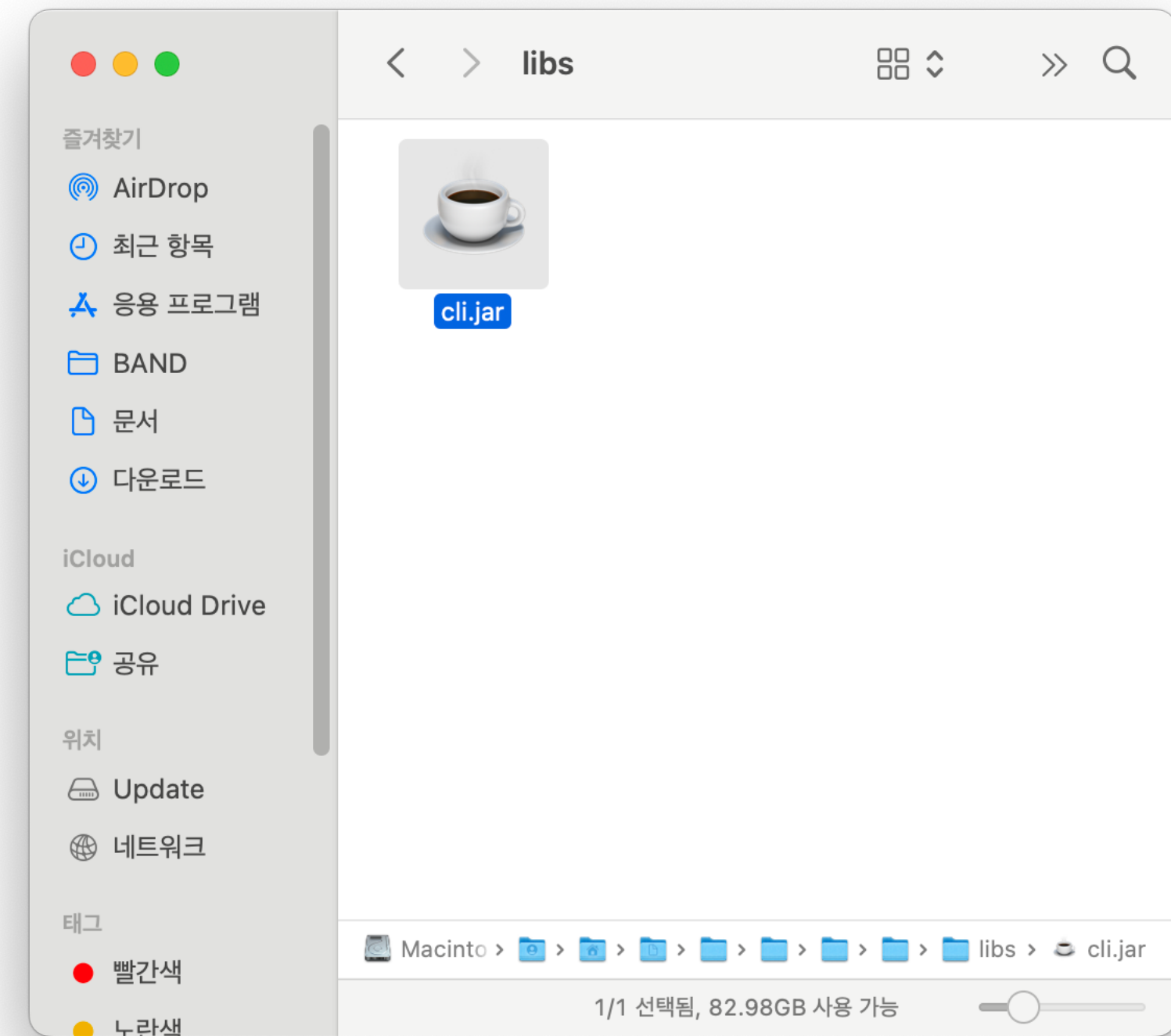
```
val fatJar = task("fatJar", type = Jar::class) {  
    manifest { attributes["Main-Class"] = "us.band.figicon.cli.Runner" }  
    from(  
        configurations.runtimeClasspath.get()  
        .map({ if (it.isDirectory) it else zipTree(it) })  
    )  
    with(tasks.jar.get() as CopySpec)  
    duplicatesStrategy = DuplicatesStrategy.EXCLUDE  
}  
  
tasks {  
    "build" {  
        dependsOn(fatJar)  
    }  
}
```

JAR 빌드하기

```
$ ./gradlew :cli:build
```

빌드된 JAR 파일 경로

예) $\text{\$}\{\text{프로젝트 경로}\}/\text{cli/build/lib/cli.jar}$

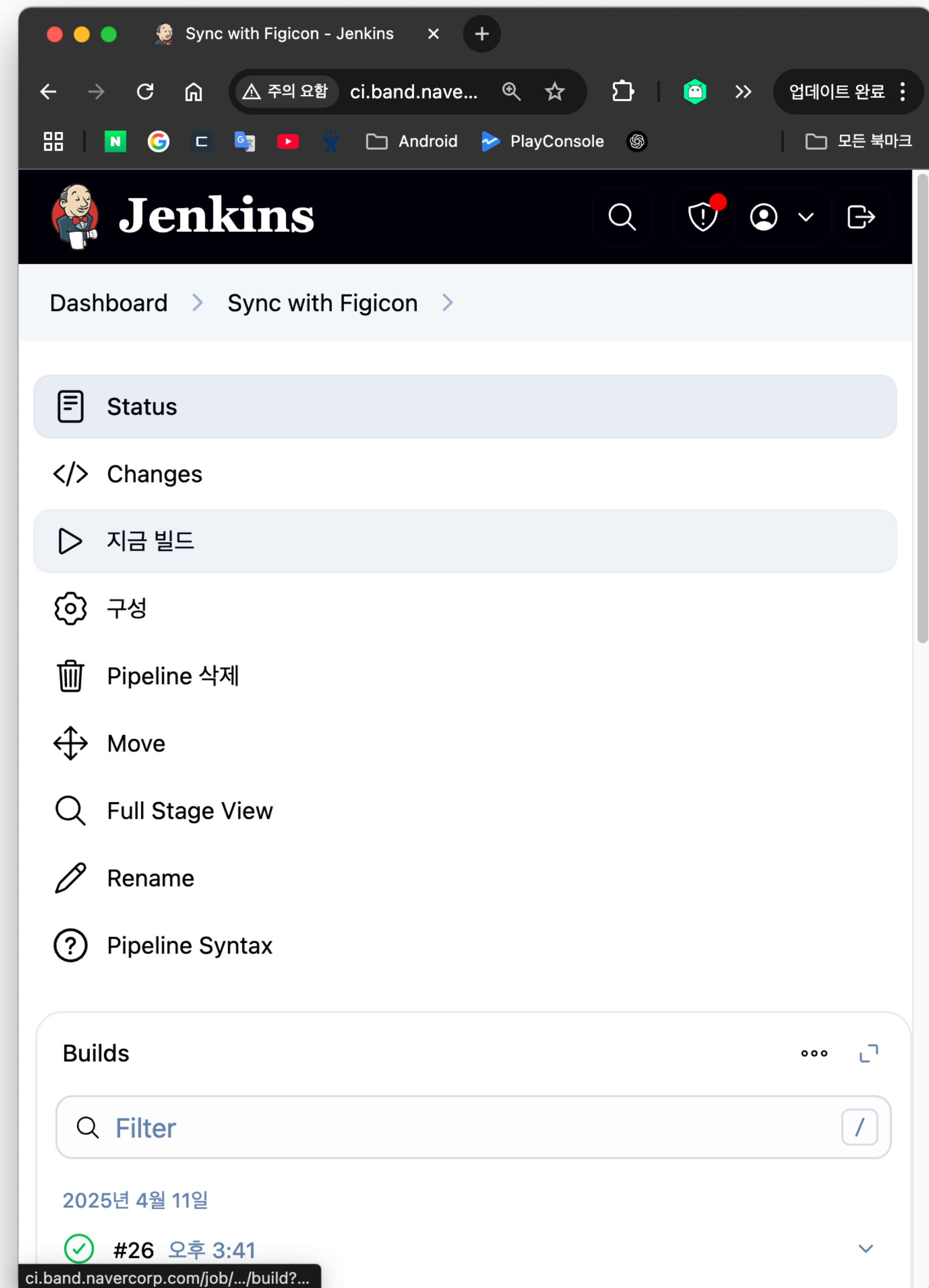


JAR 실행

```
$ java -jar cli.jar
```

Jenkins로 응용 예시

```
pipeline {
  stages {
    stage('Sync') {
      steps {
        script {
          sh "java -jar /Users/user/Documents/cli.jar"
        }
      }
    }
  }
}
```



4-2 macOS용 앱 빌드

// :composeApp 모듈의 build.gradle.kts

```
plugins {  
    id("org.jetbrains.kotlin.multiplatform")  
    id("org.jetbrains.compose")  
    id("org.jetbrains.kotlin.plugin.compose")  
}
```

```
// :composeApp 모듈의 build.gradle.kts
```

```
plugins { ... }
```

```
kotlin {  
    jvm("desktop")
```

```
    sourceSets {  
        val desktopMain by getting  
        desktopMain.dependencies {  
            implementation(project(":core"))  
        }  
    }  
}
```

```
// :composeApp 모듈의 build.gradle.kts
```

```
kotlin {  
    jvm("desktop")  
  
    sourceSets {  
        val desktopMain by getting  
        desktopMain.dependencies {  
            implementation(project(":core"))  
            // compose 관련 의존성 추가  
            implementation(compose.runtime)  
            implementation(compose.foundation)  
            implementation(compose.material)  
            implementation(compose.ui)  
            implementation(compose.components.resources)  
            implementation(compose.components.uiToolingPreview)  
            implementation(compose.desktop.currentOs)  
            implementation(libs.kotlinx.coroutines.swing)  
            ...  
        }  
    }  
}
```

// :composeApp 모듈의 build.gradle.kts

```
compose.desktop {  
    application {  
        mainClass = "us.band.figicon.MainKt"  
  
        nativeDistributions {  
            targetFormats(TargetFormat.Dmg, TargetFormat.Msi, TargetFormat.Deb)  
            packageName = "Figicon"  
            packageVersion = "1.0.0"  
            macOS {  
                val iconPath = project.file("src/desktopMain/ic_launcher.icns")  
                iconFile.set(iconPath)  
            }  
        }  
    }  
}
```

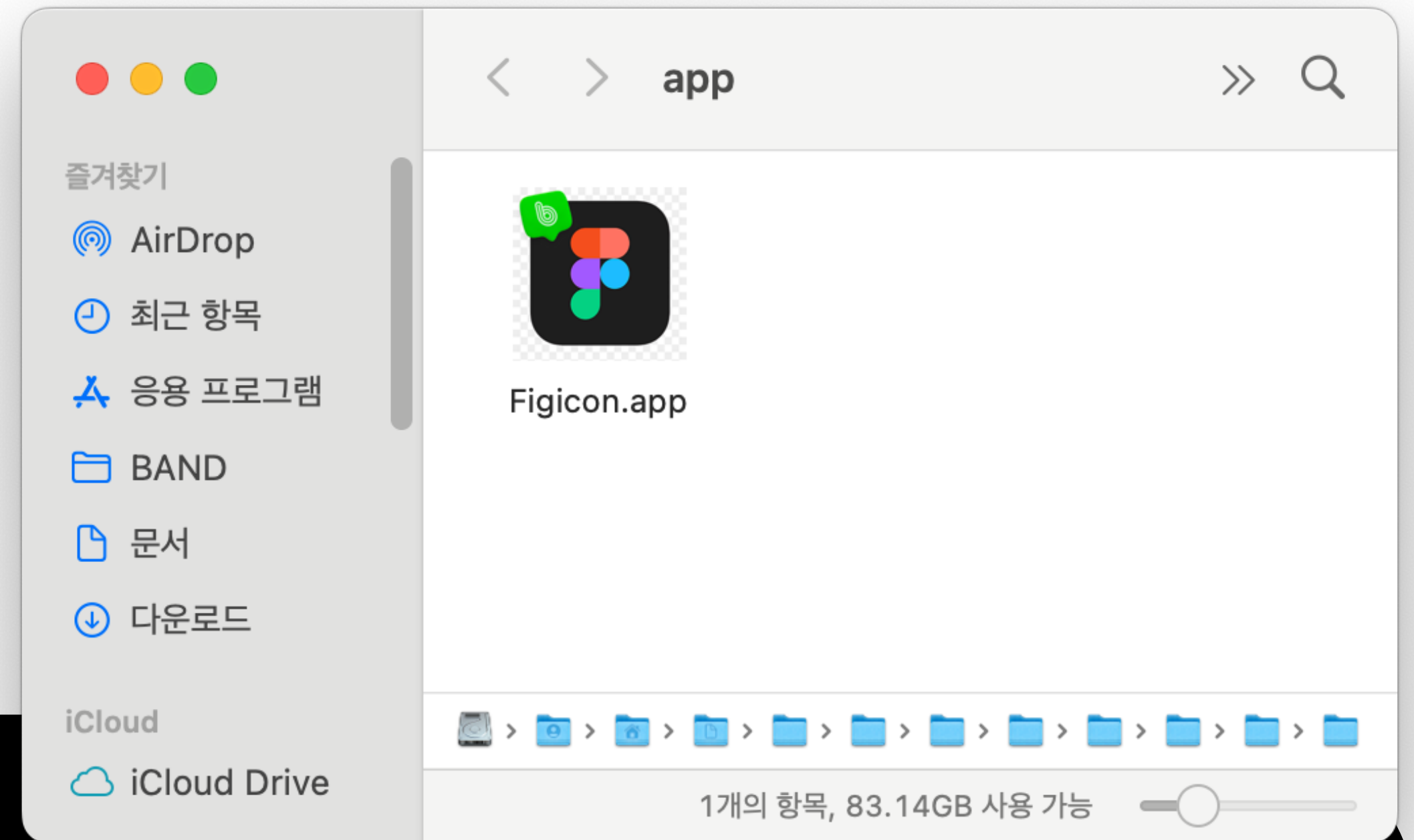
```
// src/desktopMain/kotlin/us/band/figicon/Main.kt
```

```
fun main() = application {  
  
    Window(  
        onCloseRequest = ::exitApplication,  
        title = "Figicon",  
    ) {  
        val navController = rememberNavController()  
        NavHost(navController = navController, startDestination = Route.MAIN.name) {  
            composable(route = Route.MAIN.name) {  
                MainScreen(...)  
            }  
            composable(route = Route.SETTINGS.name) {  
                SettingsScreen(...)  
            }  
        }  
    }  
}
```

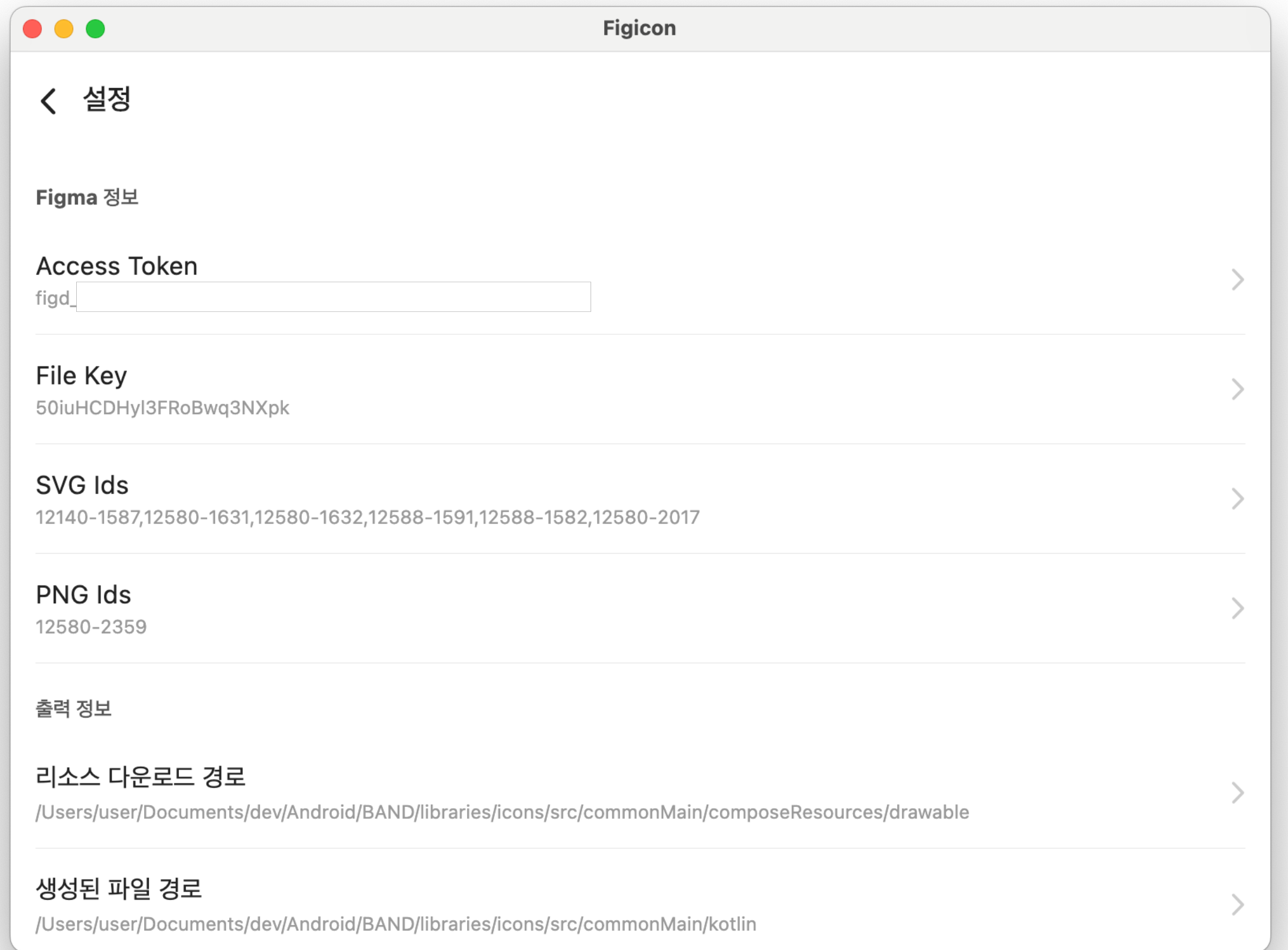
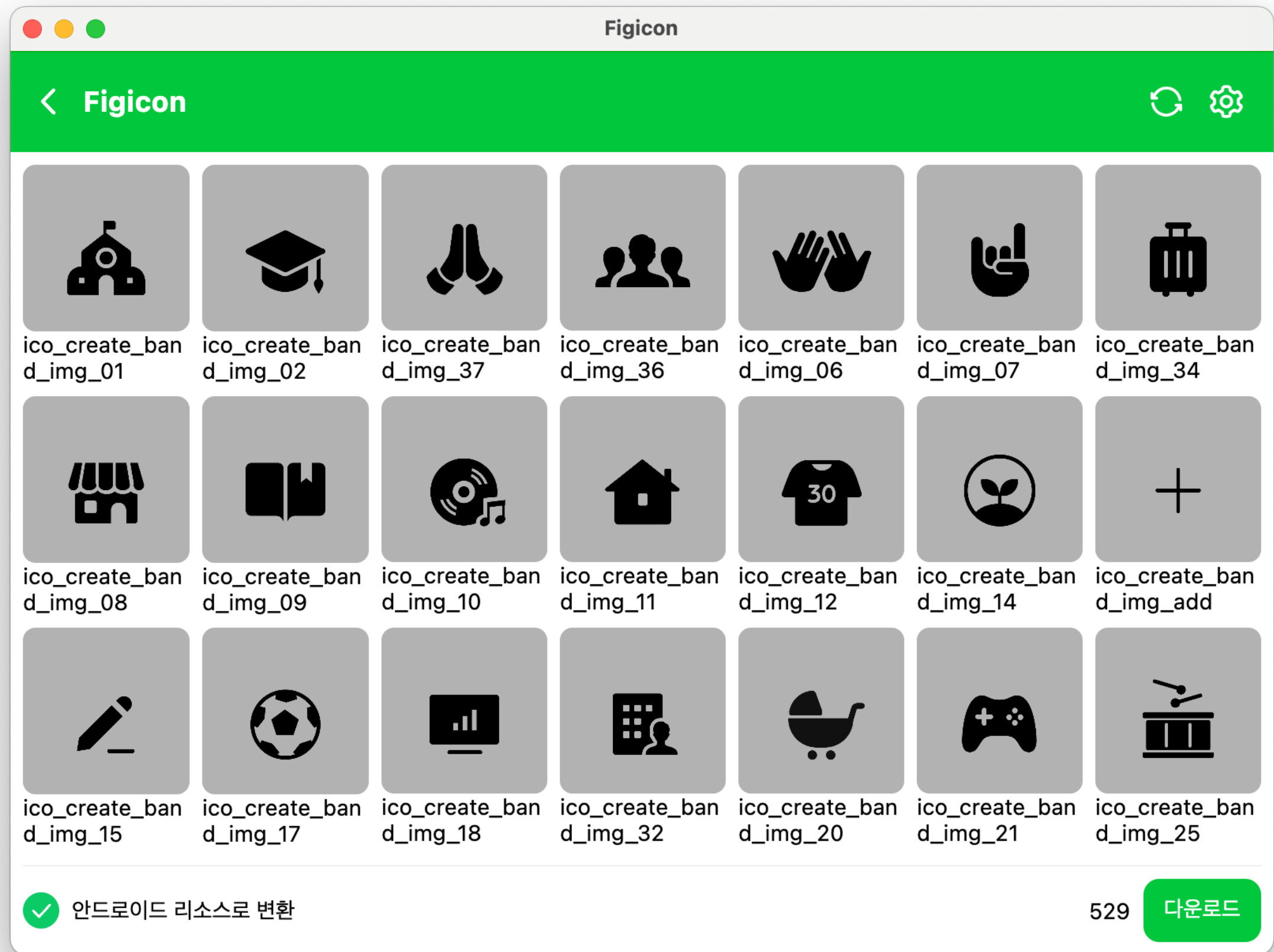
macOS 앱 빌드하기

```
$ ./gradlew :composeApp:packageMsi
```

빌드 된 macOS 앱 경로



예) $\{\text{프로젝트 경로}\}$ /composeApp/build/compose/binaries/main/app/Figicon.app





Thank You