Android Integrations

1. **Add Retrofit to your Android project** (build.gradle):

implementation 'com.squareup.retrofit2:retrofit:2.9.0'

implementation 'com.squareup.retrofit2:converter-gson:2.9.0'

implementation 'com.squareup.okhttp3:logging-interceptor:4.9.0'

1. **Create API Service Interface**:

interface CertificateApiService {

@GET("api/applications/")

suspend fun getApplications(): Response<List<BirthCertificateApplicationDto>>

@POST("api/applications/")

suspend fun createApplication(

@Body application: BirthCertificateApplicationDto

): Response<BirthCertificateApplicationDto>

@Multipart

@POST("api/documents/")

suspend fun uploadDocument(

@Part("application") applicationId: Int,

@Part("document\_type") documentType: String,

@Part file: MultipartBody.Part

): Response<DocumentDto>

}

1. **Create Retrofit Client**:

object RetrofitClient {

private const val BASE\_URL = "http://your-django-server.com/"

private val okHttpClient = OkHttpClient.Builder()

.addInterceptor { chain ->

val original = chain.request()

val requestBuilder = original.newBuilder()

.header("Authorization", "Bearer ${getAuthToken()}")

val request = requestBuilder.build()

chain.proceed(request)

}

.build()

val instance: CertificateApiService by lazy {

Retrofit.Builder()

.baseUrl(BASE\_URL)

.addConverterFactory(GsonConverterFactory.create())

.client(okHttpClient)

.build()

.create(CertificateApiService::class.java)

}

private fun getAuthToken(): String {

// Retrieve your auth token from shared preferences

return ""

}

}

1. **Example Usage in Android**:

// Create new application

val application = BirthCertificateApplicationDto(

childName = "Test Child",

parentName = "Test Parent",

// ... other fields

)

try {

val response = RetrofitClient.instance.createApplication(application)

if (response.isSuccessful) {

// Handle success

} else {

// Handle error

}

} catch (e: Exception) {

// Handle exception

}

// Upload document

val file = File("path/to/document.jpg")

val requestFile = file.asRequestBody("image/jpeg".toMediaTypeOrNull())

val body = MultipartBody.Part.createFormData("file", file.name, requestFile)

try {

val response = RetrofitClient.instance.uploadDocument(

applicationId = 1,

documentType = "birth\_cert",

file = body

)

if (response.isSuccessful) {

// Handle success

}

} catch (e: Exception) {

// Handle exception

}

1. **Android Login Implementation**:

interface AuthApiService {

@POST("api/token/")

suspend fun login(@Body credentials: LoginRequest): Response<TokenResponse>

}

data class LoginRequest(val username: String, val password: String)

data class TokenResponse(val access: String, val refresh: String)

// Usage

val response = RetrofitClient.authInstance.login(LoginRequest(username, password))

if (response.isSuccessful) {

val tokens = response.body()!!

// Store tokens securely

}