Tencent Cloud Mini Program Platform

Guidelines for Code Integration

Product Documentation

©2013-2022 Tencent Cloud. All rights reserved.

Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice

Tencent Cloud

All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Guidelines for Code Integration
- Get Demo and SDK
- Android
  - Android SDK Description
    - SDK introduction
    - SDK Integration
    - SDK integration FAQs
  - Android API
    - Mini program management APIs
      - Opening mini programs
      - Closing mini program
      - Deleting mini programs
      - Searching mini programs
      - Getting the list of recently accessed mini programs
      - Pre-downloading mini program package
      - Multi-process interaction
    - Mini program file management
    - Custom mini program capabilities
      - Custom mini program APIs
      - Custom API example
      - Custom mini program view components
    - Custom SDK Capabilities
      - Custom mini program UI
      - Custom mini program SDK APIs
      - Custom sharing capabilities
      - Custom authorization list
      - Custom permission requests
      - Custom User Attributes
      - Logging and event reporting
      - Open APIs
      - Others
    - API Description
    - SDK extension components
    - Preconfigure offline mini programs
  - Android SDK Description

©2013-2022 Tencent Cloud. All rights reserved.

Android FAQs
Android Error Codes
Android Privacy Compliance
iOS
iOS SDK Description
SDK Introduction
SDK Integration
SDK integration FAQs
iOS API
Mini program management APIs
Searching mini programs
Opening mini programs
Closing mini programs
Deleting mini programs
Getting recently accessed mini program list and information
Pre-downloading mini programs
Mini program file management
Preloading offline mini programs
Customize Mini Program Capabilities
Custom mini program APIs
Customize mini program view components
Customised SDK capabilities
Introduce
Customize mini program UI
Customize mini program information API
Customize sharing capability
Customize user attributes
Logging and event reporting
Open APIs
Others
API Introduction
iOS Extended component
iOS FAQs
iOS Error Codes
iOS Privacy Compliance
Flutter
Flutter SDK Description
SDK Overview

©2013-2022 Tencent Cloud. All rights reserved.

- SDK Quick Integration
- Common SDK Integration Issues
- Flutter API
  - Mini Program Management API
    - Opening the Mini Program
    - Closing the Mini Program
    - Deleting the Mini Program
    - Searching for the Mini Program
    - Get a list of recent visits to the mini program
    - Pre-downloading a Mini Program
  - Mini Program Capabilities Customization
    - Customizing Mini Program APIs
  - Plugin Customization
    - Customizing Plugin Capabilities
    - Customizing Mini Program Information API
    - Customizing the Sharing Capability
    - Customizing User Attributes
    - Event Reporting
    - Open APIs
    - Others
  - API Description
- Extensions
- Flutter Privacy Compliance

©2013-2022 Tencent Cloud. All rights reserved.

Guidelines for Code Integration
Get Demo and SDK

Last updated：2024-04-10 15:52:21

Android

Online integration method Demo acquisition

Offline integration method Demo and SDK acquisition

Note：

Follow Step One: Getting the Application SDK Configuration., replace it and run it. The project BundleId must be consistent with the BundleId set by the operating platform.

iOS

Get the address through CocoaPods integrated Demo

Manually integrate Demo and sdk to obtain the address

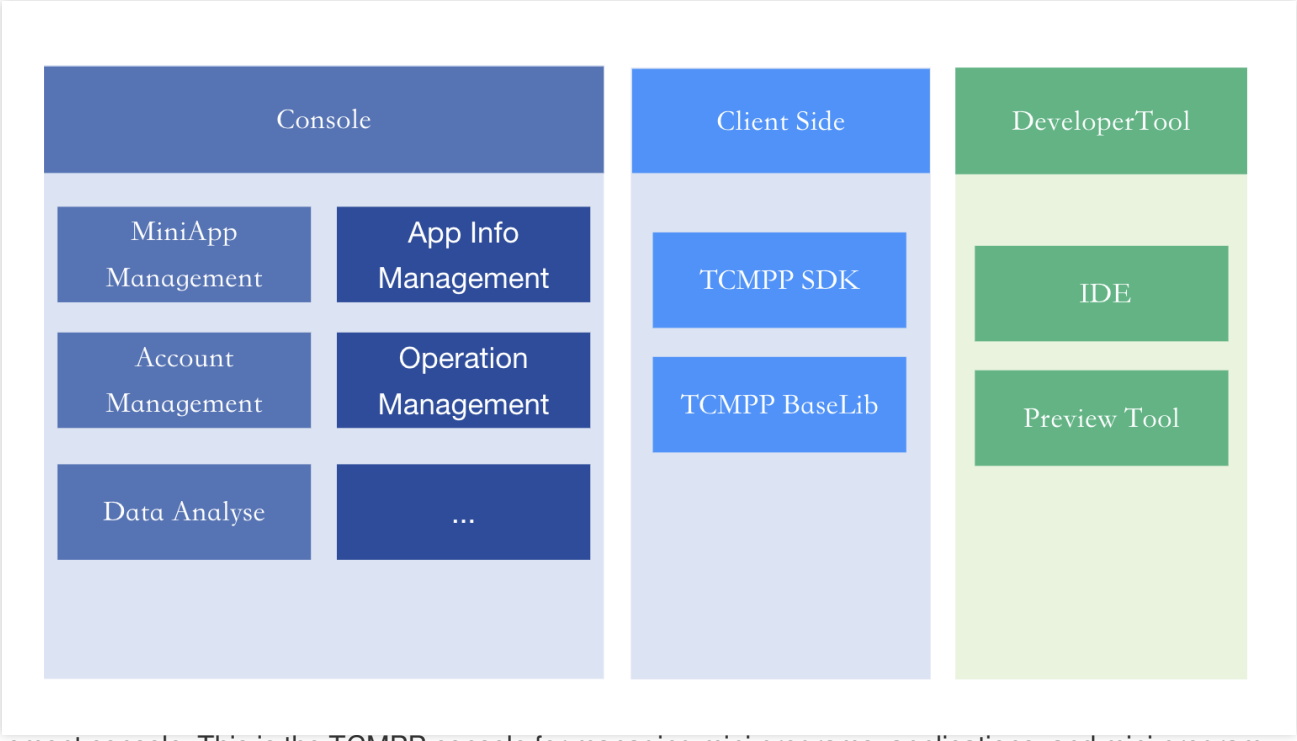
©2013-2022 Tencent Cloud. All rights reserved.

Android
Android SDK Description
SDK introduction

Last updated：2024-06-27 11:11:52

Introduction to TCMPP

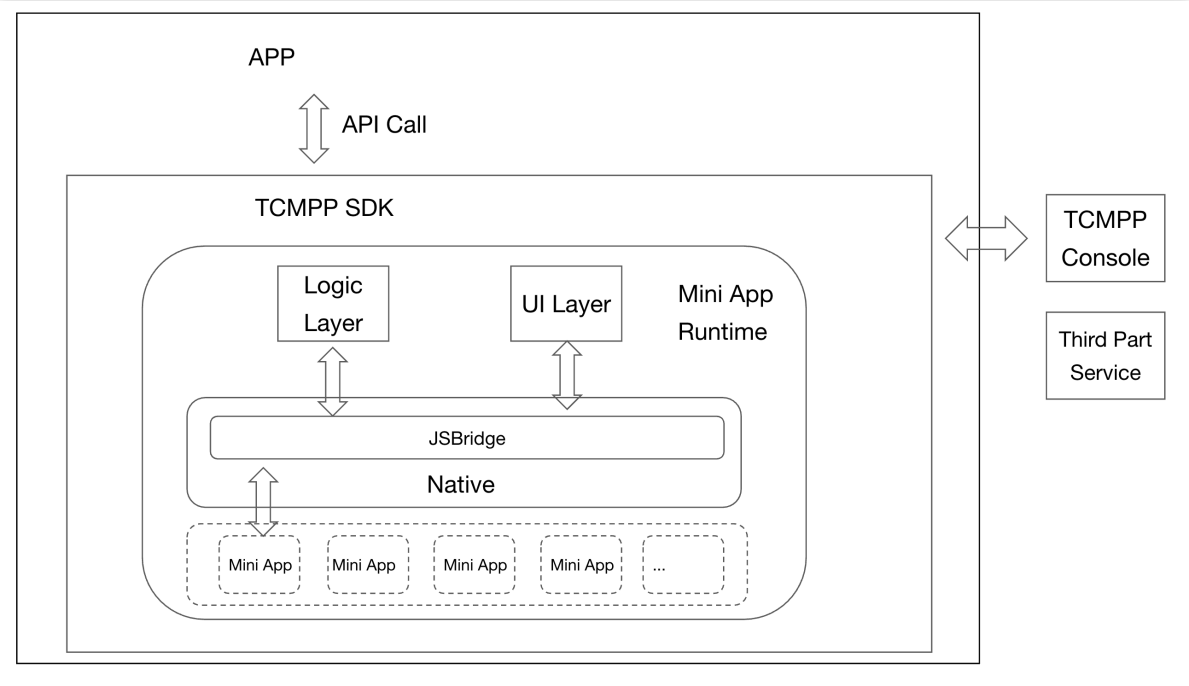
TCMPP consists of three parts: management console, client SDK and mini program development tool (TCMPP IDE).



Management console: This is the TCMPP console for managing mini programs, applications, and mini program operations.
Client: This refers to the mobile client application integrated with the TCMPP mini program SDK. The SDK provides a runtime environment for mini programs within the client application.
Developer tools: Developers use the mini program IDE for developing, debugging, and submitting versions of mini programs. The mini program preview assistant is used for previewing and validating mini programs on mobile clients.

How the TCMPP mini program SDK Works

©2013-2022 Tencent Cloud. All rights reserved.



The mini program SDK provides a runtime environment for mini programs within client apps. It communicates with the TCMPP backend to fetch mini program information and loads and runs the mini programs in the provided runtime environment.

Mini program SDK description

To keep the SDK lightweight and modular, it is divided into two categories: core SDK (mini_core) and extension SDKs.

Core SDK: Provides the minimal environment required to load and run mini programs. Users must integrate the core SDK to utilize basic mini program capabilities.

Extension SDKs: Provide additional features that users can integrate based on their needs.

Mini program SDKs:

SDK name	Description	Usage notes	Package size
Core SDK (mini_core)	Provides the minimal runtime environment for loading mini programs.	-	~4.7MB
Same-layer rendering extension SDK - Public version (mini_extra_public_x5)	Enables same-layer rendering for mini program components.	Requires public version kernel. For details, see SDK extension components.	~55 KB (excluding x5 kernel)
Same-layer rendering	Enables same-layer	Requires dynamic version	~56.7 KB (excluding

©2013-2022 Tencent Cloud. All rights reserved.

extension SDK - Dynamic version (mini_extra_dynamic_x5)	rendering for mini program components.	kernel. For details, see SDK extension components.	x5 kernel)
Same-layer rendering extension SDK - Static version (mini_extra_static_x5_new)	Enables same-layer rendering for mini program components.	Requires static version kernel. For details, see SDK extension components.	~54.4 KB (excluding x5 kernel)
QR code extension SDK (mini_extra_qrcode)	Provides QR code scanning capabilities.	For details, see SDK extension components	~ 35 KB
Map extension SDK - Tencent Maps version (mini_extra_map)	Supports Tencent Maps for location and map services.	Requires Tencent Maps SDK and dependencies. For details, see SDK extension components	About 156.4 KB, the volume does not include Tencent Map SDK
Map extension SDK - Huawei Maps version (mini_extra_huawei_map)	Supports Huawei Maps for location and map services.	Requires Huawei Maps SDK and dependencies. For details, see SDK extension components	~236.8KB (excluding Huawei Maps SDK)
Map extension SDK - Google Maps Version (mini_extra_google_map)	Supports Google Maps for location and map services.	Requires Google Maps SDK and dependencies. For details, see SDK extension components	~153.2KB (excluding Google Maps SDK)
Live streaming extension SDK (mini_extra_trtc_live)	Provides live streaming capabilities for mini programs.	Requires Tencent TRTC dependencies. For details, see SDK extension components	~91.4KB (excluding Tencent TRTC SDK)
LBS extension SDK (mini_extra_lbs)	Provides location-based services for mini programs.	For details, see SDK extension components	~ 73.8 KB
Bluetooth extension SDK (mini_extra_bluetooth)	Provides Bluetooth API access for mini programs.	For details, see SDK extension components	~ 115 KB
NFC extension SDK (mini_extra_nfc)	Provides NFC API access for mini programs.	For details, see SDK extension components	~ 65 KB
Biometric authentication extension SDK (mini_extra_soter)	Provides biometric authentication API	For details, see SDK extension components	~ 71 KB

©2013-2022 Tencent Cloud. All rights reserved.

	access for mini programs.		
Clipboard extension SDK (mini_extra_clipboard)	Provides clipboard access for mini programs.	For details, see SDK extension components	~7KB
Contacts extension SDK (mini_extra_contact)	Provides contacts API access for mini programs.	For details, see SDK extension components	~ 49 KB
Document engine extension SDK (mini_extra_doc)	Provides document preview capabilities for mini programs.	Requires Tencent Document SDK. For details, see SDK extension components	~40KB
Media extension SDK (mini_extra_media_support)	Provides media selection API access for mini programs.	For details, see SDK extension components	~ 568.1 KB

©2013-2022 Tencent Cloud. All rights reserved.

SDK Integration

Last updated：2024-06-27 11:14:07

This guide will help developers quickly integrate a minimal functionality mini program SDK to enable mini program launching.

Integration process

1. Configure online repository address

The configuration method depends on the Gradle plugin version used in your project.

For Gradle version earlier than 7.0, see Method 1.

For Gradle version 7.1 and later, see Method 2.

Method 1: For Gradle version earlier than 7.0

Add the following configuration to the project-level build.gradle:

```
buildscript {
    dependencies {
        classpath 'org.jetbrains.kotlin:kotlin-gradle-plugin:1.4.32'
    }
}

allprojects {
    repositories {
        maven {
            url 'https://tcmpp-work-maven.pkg.coding.net/repository/tcmpp/android'
        }
    }
}
```

Method 2: For Gradle versions 7.1 and later.

Add the following configuration to the project-level settings.gradle:

```
  pluginManagement {
    repositories {
        maven {
            url 'https://tcmpp-work-maven.pkg.coding.net/repository/tcmpp/android'
        }
    }
}
dependencyResolutionManagement {
    repositories {
```

©2013-2022 Tencent Cloud. All rights reserved.

```
        maven {
            url 'https://tcmpp-work-maven.pkg.coding.net/repository/tcmpp/android'
        }
    }
}
```

2. Kotlinp lugin configuration

Configure the Kotlin plugin based on the Gradle plugin version:

Method 1: For Gradle version earlier than 7.0:

```
apply plugin: 'kotlin-android'
apply plugin: 'kotlin-kapt'
```

Method 2: For Gradle versions 7.1 and later:

```
  plugins {
    id "org.jetbrains.kotlin.android"
    id "org.jetbrains.kotlin.kapt"
}
```

3. Configure packagingoptions

Add the following configuration to the app-level build.gradle:

```
android {
    defaultConfig {
        packagingOptions {
            pickFirst 'lib/arm64-v8a/libc++_shared.so'
            pickFirst 'lib/armeabi/libc++_shared.so'
            pickFirst 'lib/armeabi-v7a/libc++_shared.so'
            pickFirst 'lib/arm64-v8a/libmarsxlog.so'
            pickFirst 'lib/armeabi/libmarsxlog.so'
            pickFirst 'lib/armeabi-v7a/libmarsxlog.so'
            pickFirst 'lib/arm64-v8a/libv8jni.so'
        }
    }
}
```

4. Configure mini program SDK dependencies

```
dependencies {
    implementation 'com.google.android.material:material:1.3.0-alpha03'
    implementation 'androidx.core:core-ktx:1.6.0'
```

©2013-2022 Tencent Cloud. All rights reserved.

```
    //gosn
    implementation 'com.google.code.gson:gson:2.8.6'
    // ok-http
    implementation 'com.squareup.okhttp3:okhttp:3.12.13'

    // mini app start
    kapt 'com.tencent.tcmpp.android:mini-annotation_processor:${version}'// For ver
    implementation 'com.tencent.tcmpp.android:mini_core:${version}'//Please see And
    // mini app end
}
```

Notes:

The {version} in the configuration should be replaced with the version of the dependency; see Android SDK Updates.

If you have used annotationProcessor in your project, you need to change all annotationProcessors to kapt annotators.

If you want to use same-layer rendering, you need to integrate the x5 kernel,see SDK extension components.

After you introduce the dependency of mini program SDK, you can integrate the mini program SDK.

Preparations before integration

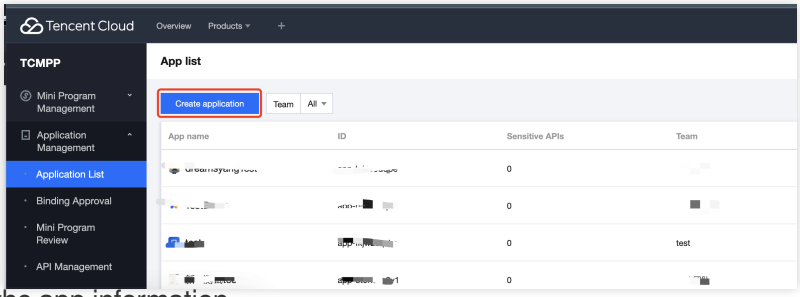
1. Get configuration file

The mini program SDK initialization requires a configuration file from the mini program console. Follow these steps:

How to get configuration file:

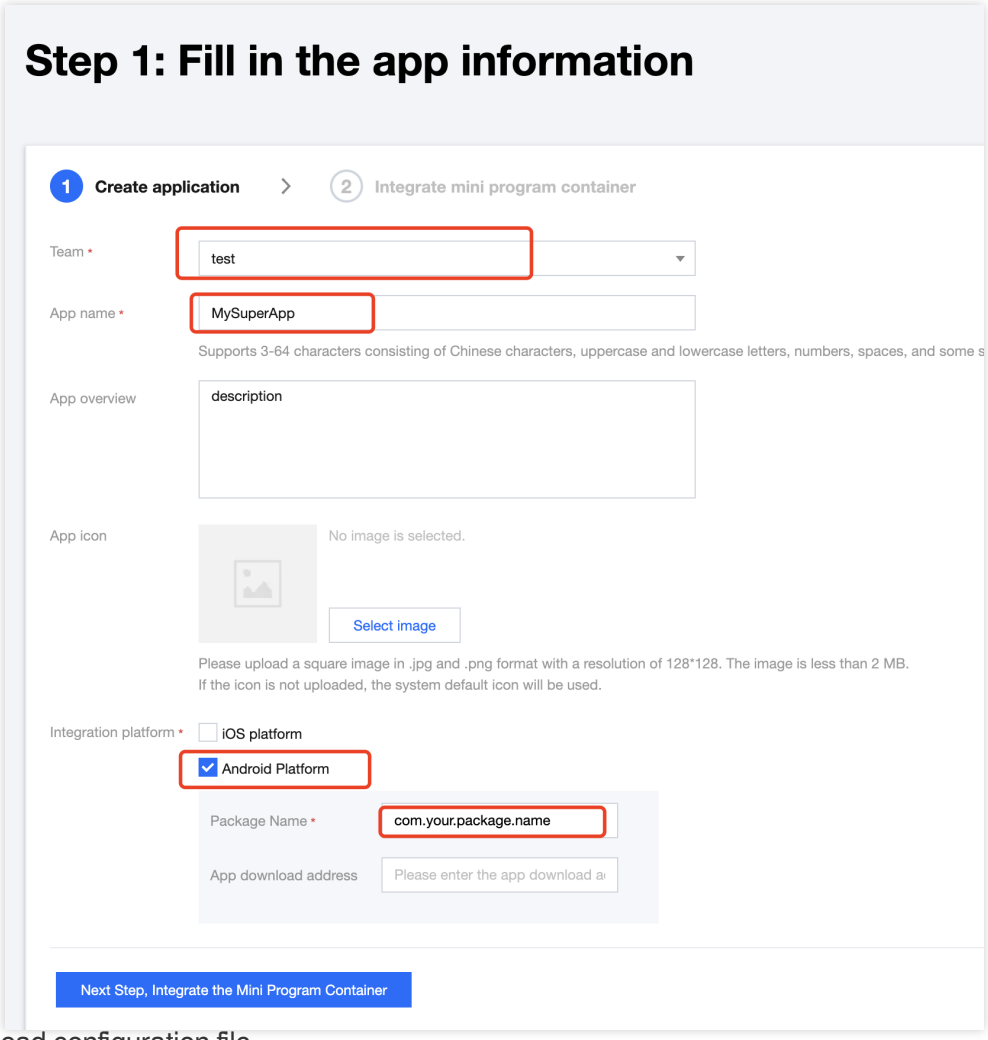
Log in to the mini program console

Create apps



Fill in the app information

©2013-2022 Tencent Cloud. All rights reserved.

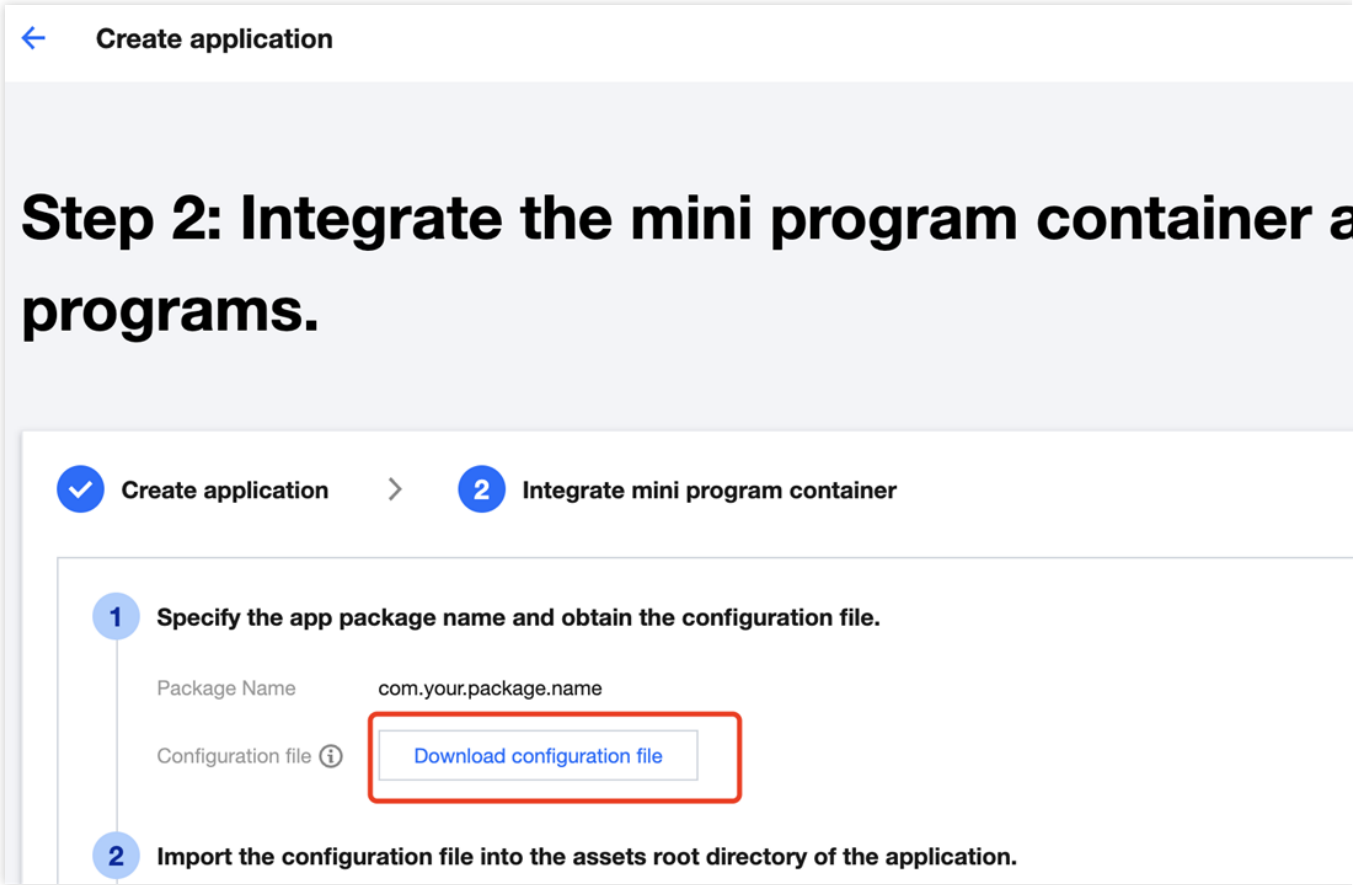


Download configuration file

Note：

Default name of the downloaded configuration file: tcmpp-android-configurations.json

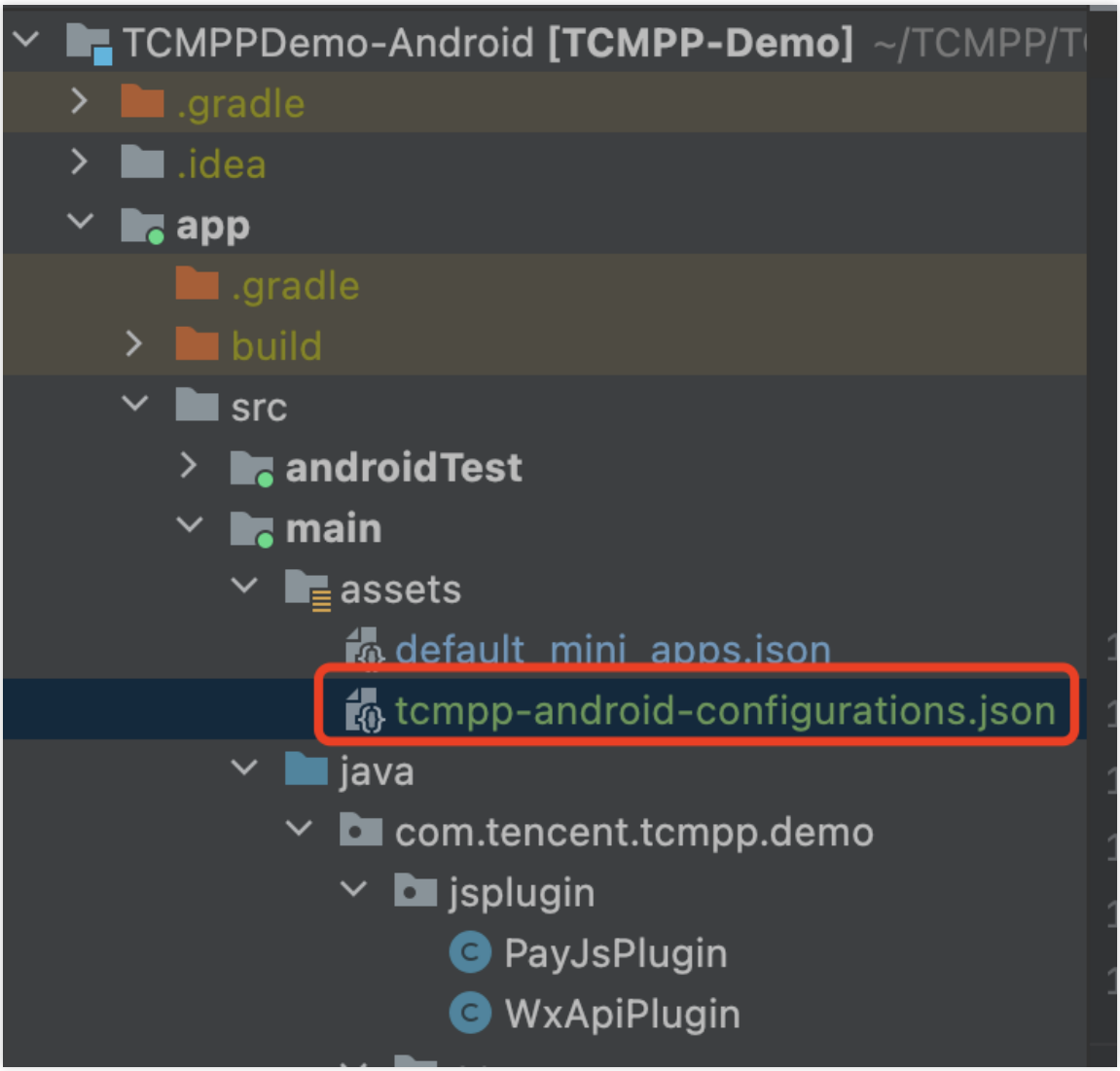
©2013-2022 Tencent Cloud. All rights reserved.



2. Add configuration file to project

After obtaining the configuration file, you need to copy the configuration file to the assets directory of your project.

©2013-2022 Tencent Cloud. All rights reserved.



Note：

Ensure the app package name matches the packageName in the configuration file. Otherwise, the mini program SDK cannot pass the package name verification when running, which will cause the SDK initialization error.

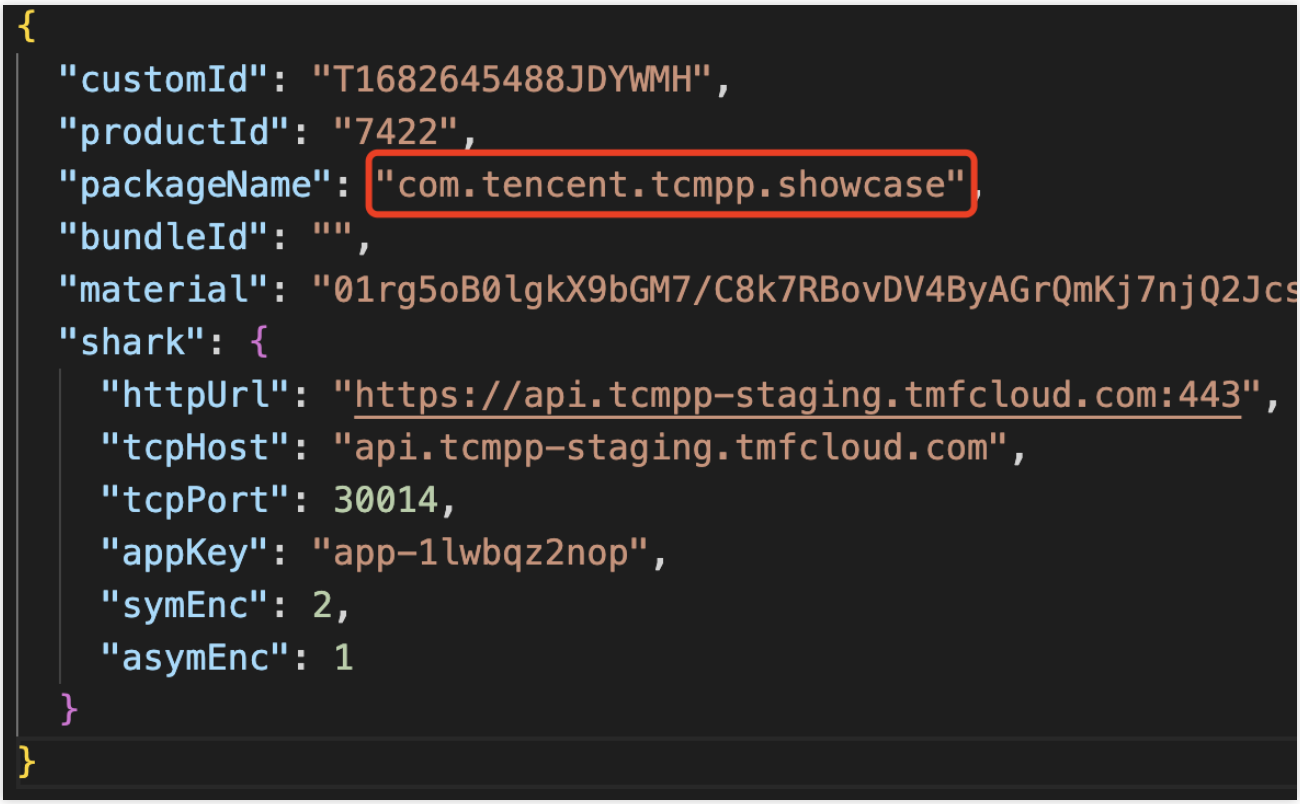
If they do not match, you can:

either change the project package name.

or update the package name in the console and re-download the configuration file.

The packageName field in the configuration file needs to be consistent with the package name of the current project.

©2013-2022 Tencent Cloud. All rights reserved.



3. Add mini program SDK Initialization configuration in source code

Inherit the implementation MiniConfigProxy class;

MiniConfigProxy implementation class, add the following annotation:

```
@ProxyService(proxy = MiniConfigProxy.class)
```

Sample code:

```
@ProxyService(proxy = MiniConfigProxy.class)
public class MiniConfigProxyImpl extends MiniConfigProxy {
    /**
     * App Application
     * @
     */
    @Override
    public Application getApp() {
        return "app Application";
     }

      /**
     * Create initialization configuration information
     * @
     */
    @Override
    public MiniInitConfig buildConfig() {
      MiniInitConfig.Builder builder = new MiniInitConfig.Builder();
```

©2013-2022 Tencent Cloud. All rights reserved.

```
        MiniInitConfig config = builder
                    .configAssetName("tcmpp-android-configurations.json") //Configuratio
                     .imei("IMEI") //(Optional) Device ID, which is used for grayscale r
                     .autoRequestPermission(true) //Configure whether the mini program s
                     .debug(true) //Log switch, disabled  by default
                     .build();
        }
    }
```

Note：

Please note that configAssetName of MiniConfig should specify the path and name of the configuration file in the project.

At this point, the mini program SDK is integrated. And now you can open and preview mini programs using the API provided by mini program SDK.

Launch mini programs

To launch a mini program, use the following code:

Note：

The appId is the mini program ID, which you need to obtain from the mini program developers.

```
TmfMiniSDK.startMiniApp(activity, appId, miniStartOptions);
```

©2013-2022 Tencent Cloud. All rights reserved.

SDK integration FAQs

Last updated：2024-06-27 11:10:09

When integrating the mini program SDK and compiling the project, you may encounter the following AAPT error

```
  AAPT: error: attribute android:requestLegacyExternalStorage not found.
```

Solution

Add the following configuration under the <application> tag in AndroidManifest.xml:

```
<application
    android:theme="@style/AppTheme"
    tools:replace="android:icon"
      tools:remove="android:requestLegacyExternalStorage">
 /application>
```

When integrating the mini program SDK and compiling the project, you may encounter the 'Duplicate class android.support.v4' error:

```
Duplicate class android.support.v4.app.INotificationSideChannel found in modules co
```

Solution

Add the following code to `gradle.properties`:

```
android.useAndroidX=true
android.enableJetifier=true
```

When integrating the mini program SDK and compiling the project, you may encounter the 'compileDebugJavaWithJavac' version matching error:

```
Execution failed for task ':app:kaptGenerateStubsDebugKotlin'.
> 'compileDebugJavaWithJavac' task (current target is 1.8) and 'kaptGenerateStubsDe
  Consider using JVM toolchain: https://kotl.in/gradle/jvm/toolchain
```

Solution

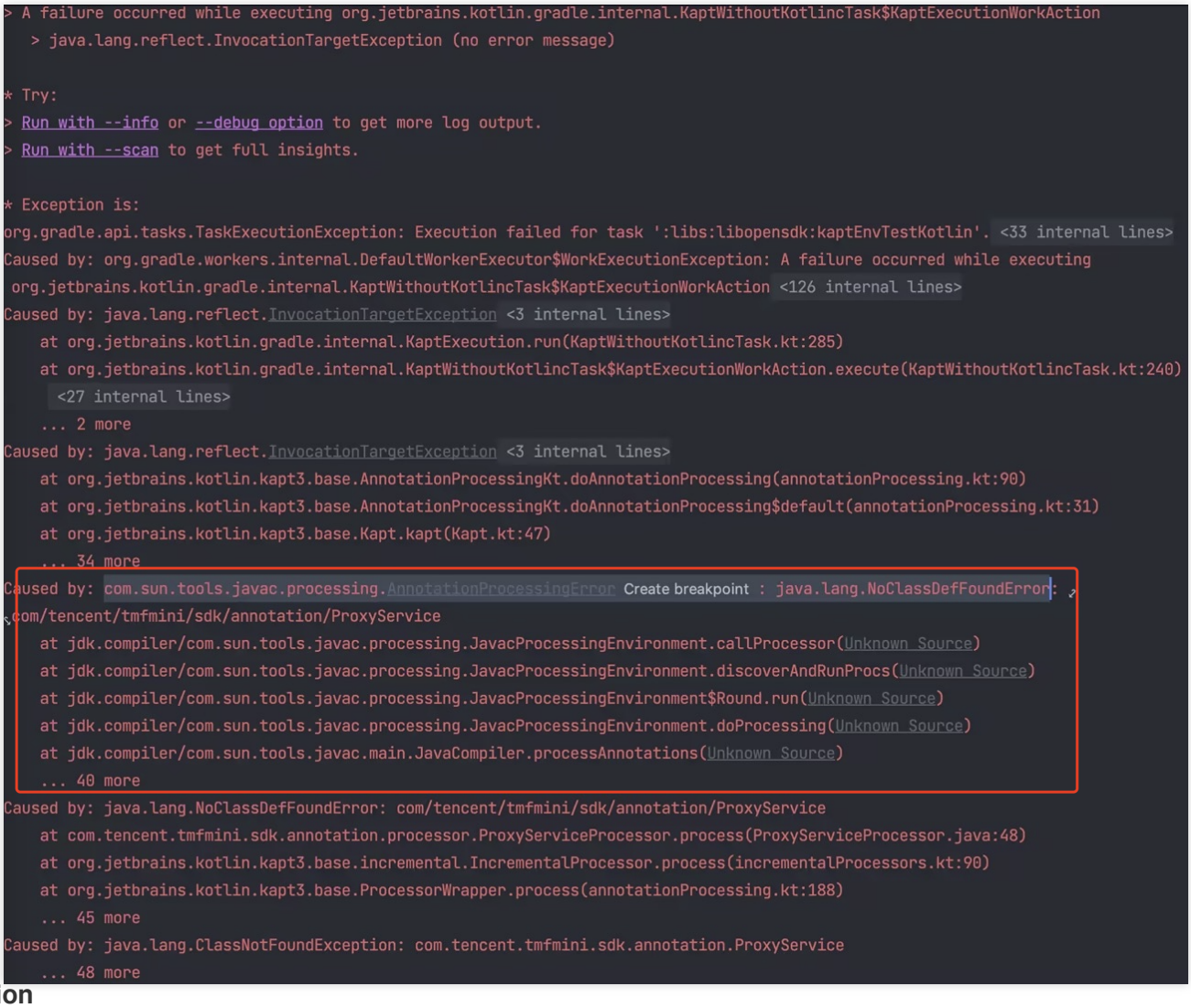
Change the JDK version in compileOptions in build.gradle to the appropriate version.

```
android {

    compileOptions {
        sourceCompatibility JavaVersion.VERSION_17 //Need to match kapt version
```

©2013-2022 Tencent Cloud. All rights reserved.

```
        targetCompatibility JavaVersion.VERSION_17 //Need to match kapt version
    }
}
```

After accessing the mini program SDK, the following 'java.lang.NoClassDefFound, ProxyService' issue occurs when executing the project compilation build, as shown below



Solution

Check if the following configurations are present in the project and remove them if they are present.

```
kapt.include.compile.classpath=false
```

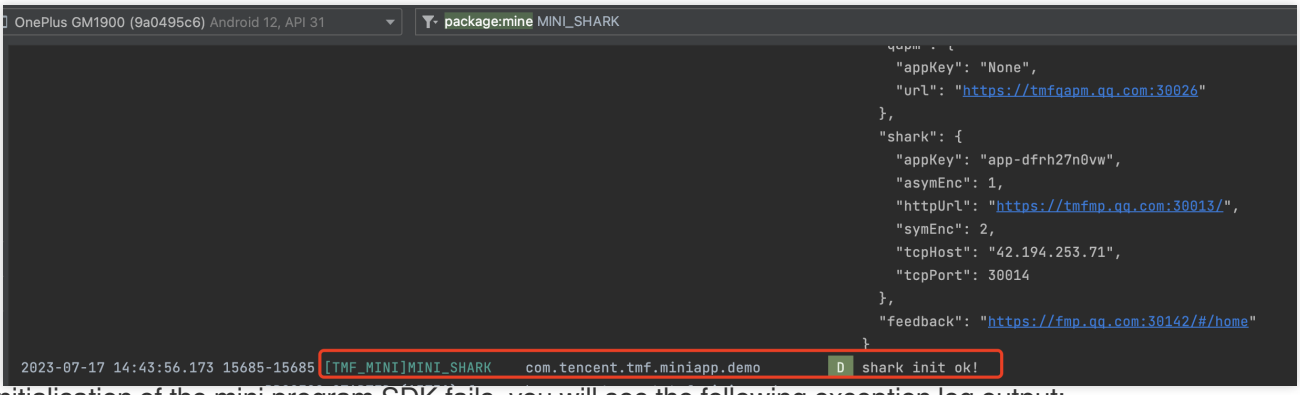
What is the minimum Android system version supported by the mini program SDK?

Minimum SDK version (minSdkVersion): 21, corresponding to Android version Android 5.0.

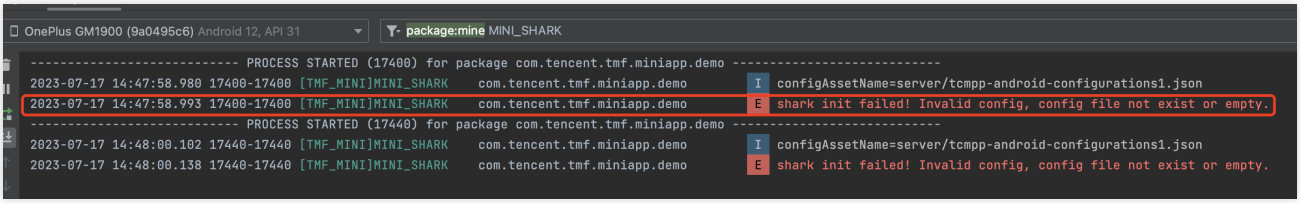
How to determine whether the mini program SDK has been initialised successfully?

©2013-2022 Tencent Cloud. All rights reserved.

After version 1.5.1.1 of the mini program SDK mini_core, you can filter the logs through MINI_SHARK to determine whether the initialisation was successful or not, and a successful initialization will output: 'shark init ok!', as shown in the figure:



If the initialisation of the mini program SDK fails, you will see the following exception log output:



What types of devices does the mini program SDK support for debugging?

The mini program SDK supports emulators and real devices with CPU architecture type arm. Emulators with x86 type are not supported.

©2013-2022 Tencent Cloud. All rights reserved.

Android API
Mini program management APIs
Opening mini programs

Last updated：2024-10-18 10:46:37

The mini program management APIs provide capabilities for searching, opening, and closing mini programs.

Opening mini programs

When opening a mini program, the system will check whether there is a locally cached one. If not, it will automatically download the mini program from the remote server and then open it. If a cached version is available, the local mini program will be opened first while the system checks in the background for any new versions on the server.

Notes:

If a new version is available, it will be downloaded, and the next time you open the mini program, the updated version will be used.

1. Opening a mini program by mini program ID (appId)

To open the released version of a mini program :

Notes:

The appId field is the ID of the mini program, which can be obtained from the mini program developer or via the mini program search API.

The miniStartOptions are the startup parameters for the mini program. For details, see API Description.

```
TmfMiniSDK.startMiniApp(activity, appId, miniStartOptions);
```

To open the Preview or development version of a mini program:

```
TmfMiniSDK.startMiniApp(activiy, appId, MiniScene.LAUNCH_SCENE_MAIN_ENTRY, appVerTy
```

Notes:

The appVerType should match the version of the mini program. The value of the appVerType can be obtained from the MiniApp object instance returned by the API (getRecentList).

The value of appVerType for the Preview of the mini program should be MiniApp.TYPE_PREVIEW.

The value of appVerType for the development version of the mini program should be MiniApp.TYPE_DEVELOP.

2. Opening a mini program through QR code

©2013-2022 Tencent Cloud. All rights reserved.

The mini program SDK provides an API to open mini programs based on QR code scanning in the console. Before using the scanning capability provided by the mini program SDK, you need to integrate the scanning extension capability. For details about the integration, refer to the scanning capability documentation.
After the scanning capability is integrated, you can start the QR code scanning and open the mini program with one of the options:
Option 1: Process the scanning results by host app's Activity
Start QR code scanning:

```
TmfMiniSDK.scan(activity);
```

Get the result of the QR code scanning:
Notes:
The result of the QR code scanning is returned through the Activity's onActivityResult method. To identify the QR code, you need to rewrite the Activity's onActivityResult method and call `TmfMiniSDK.getScanResult` to get the QR code.

```
@Override protected void onActivityResult(int requestCode, int resultCode, @Nullabl
    // Get QR code scanning result  JSONObject scanResult = TmfMiniSDK.getScanResul
```

Open the mini program according to the QR code scanning result:
Notes:
The getScanResult method returns data in a JSON structure. The mini program's startup parameters are stored in the `'result'` field of this JSON structure. You need to extract the value of the `result` field and use it to open the mini program.

```
@Override protected void onActivityResult(int requestCode, int resultCode, @Nullabl
    //Get QR code scanning result  JSONObject scanResult = TmfMiniSDK.getScanResult
    if (scanResult != null) { // Extract `result` field String result = scanResult.
            //Use the value of the `result` field to open the mini program TmfMiniS
```

Option 2: Process the scanning results automatically by the agent Activity and start up the mini program directly
Scan the QR code and start up the mini program:

```
TmfMiniSDK.startMiniAppByScan(activity);
```

3. Configuring Scheme to open the mini program

Mini programs can be opened by third-party apps and scanning tools via URL redirection. Before you can use this capability, you need to configure the URL Scheme of the host app by adding the following string resource to the app:

```
<string name="mini_sdk_intent_filter_scheme">your scheme</string>
```

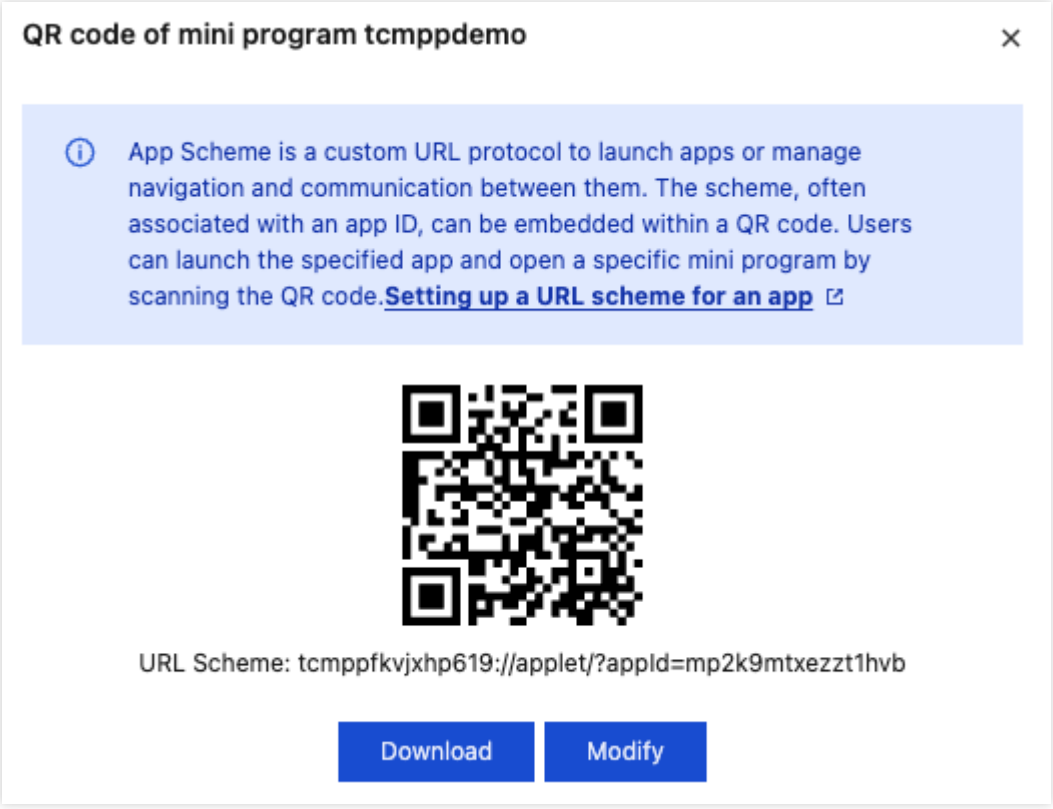
where `your scheme` should be replaced by the host's Scheme.

©2013-2022 Tencent Cloud. All rights reserved.

The host of the mini program URL is applet and the full format of the URL is: ${scheme}://applet?appid=${appId}&path=${encoded path}&param=${encoded param}. Parameters in the URL are as follows:

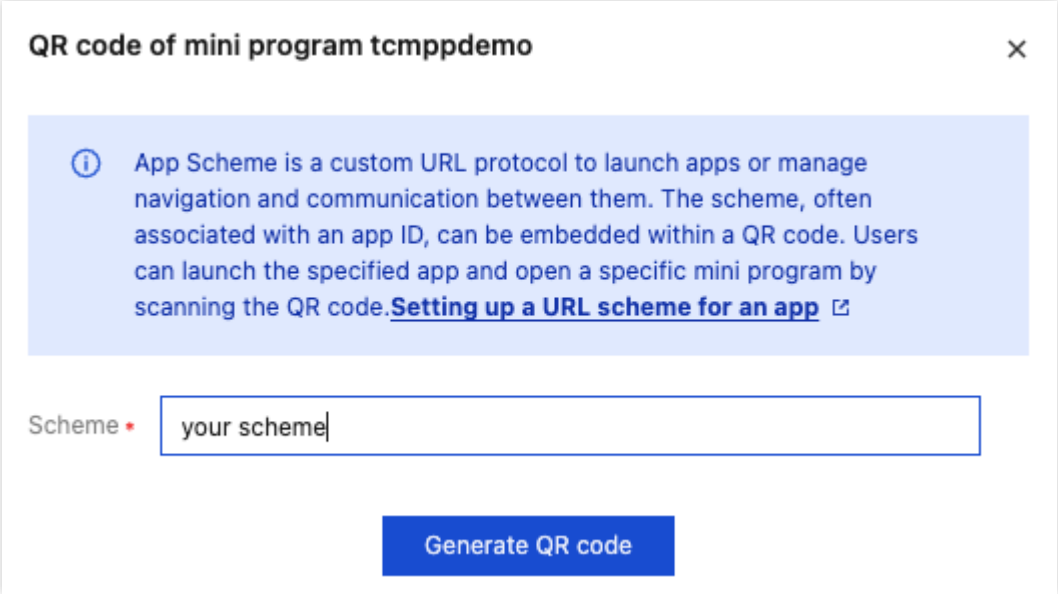
Parameter	Type	Required	Description
appId	String	Yes	Mini program ID
path	String	No	Mini program entry path, for which the URL encoding is required
param	String	No	The query passed to the mini program, for which the URL encoding is required

In the TCMPP console, a corresponding URL is created for the released version of each mini program and a QR code is generated based on the URL. When a URL is created, tcmpp+ host app's appkey is used as the Scheme by default.



To ensure that a correct URL is generated, please replace it with the host app's URL Scheme. Click Modify to change the Scheme used to generate the URL:

©2013-2022 Tencent Cloud. All rights reserved.



Once configured, you can open the mini program by scanning the generated QR code or via URL redirection.

4. Opening a mini program by search

Call the mini program search API:

Notes:

The parameters for SearchOptions are:

keyword: Mini program keyword

firstCate: Primary category of the mini program

sencondaryCate: Secondary category of the mini program

If both primary and secondary categories are provided, the search results will be the intersection of both categories.

If only one category is provided, the search will be based on that single category.

```
SearchOptions searchOptions = new SearchOptions(keyword, firstCate, sencondaryCate)

TmfMiniSDK.searchMiniApp(searchOptions, new MiniCallback&lt;List<MiniApp>>() { @Ove
          // Search failed or the mini program list is empty } } });
```

Open the mini program from the search results:

Notes:

The search results are returned via the value method of the MiniCallback API.

The parameter int code indicates the error code.

The parameter String msg indicates the returned search information.

The parameter List<MiniApp> data indicates the list of found mini programs.

```
SearchOptions searchOptions = new SearchOptions(keyword, firstCate, sencondaryCate)

TmfMiniSDK.searchMiniApp(searchOptions, new MiniCallback&lt;List<MiniApp>>() { @Ove
```

©2013-2022 Tencent Cloud. All rights reserved.

```
            MiniApp miniApp = data.get(0);
            // Open the first mini program in the search results TmfMiniSDK.startMin
           // Search failed or the mini program list is empty } } });
```

5. Mini program startup listener

To help developers troubleshoot issues, the mini program SDK provides a listener for startup errors. You can configure it with the resultReceiver in MiniStartOptions.

How to configure:

```
private ResultReceiver mResultReceiver = new ResultReceiver(new Handler()) {
    @Override
    protected void onReceiveResult(int resultCode, Bundle resultData) {
      if (resultCode != MiniCode.CODE_OK) {
                String errMsg = resultData.getString("errMsg");
                Toast.makeText(mActivity, errMsg + resultCode, Toast.LENGTH_SHORT
                }
         }
};
```

`The miniStartOptions.resultReceiver` can be used to receive error codes when a mini program is started up. For error code details, see API Description.

©2013-2022 Tencent Cloud. All rights reserved.

Closing mini program

Last updated：2024-06-27 11:09:11

The 'closing mini program' API terminates all activities of a mini program and ends its process.

Closing a specific mini program

API description: This API allows you to close a mini program with a specific ID.

```
/**
 * Terminate a mini program
 *
 * @param context
 * @param appId
 */
public static void stopMiniApp(Context context, String appId)
```

Sample code:

```
TmfMiniSdk.stopMiniApp(context,appId);
```

Closing all mini programs

API description: This API allows you to close all mini programs.

```
/**
 * Terminate all mini programs
 *
 * @param context
 */
public static void stopAllMiniApp(Context context)
```

Sample code:

```
TmfMiniSdk.stopAllMiniApp(context);
```

©2013-2022 Tencent Cloud. All rights reserved.

Deleting mini programs

Last updated：2024-06-27 11:08:59

Note：

Running a mini program stores the program package and information locally for faster access in the future. If you want to delete all information of a mini program, you can use the following API to delete a specific mini program or all mini programs.

Deleting a mini program by mini program ID

API description:

```
/**
 * Delete a mini app based on `appId`, regardless of the version (official/develope
 * @param appId
 */
public static void deleteMiniApp(String appId)
```

Sample code:

```
//Delete a mini program with a specific ID
TmfMiniSdk.deleteMiniApp(appId);
```

Deleting a mini program of a specified version type and version number

Delete a mini program of the specified type and the specified version。

API description:

```
/**
* Delete a mini program of the specified type and the specified version
* @param appId
* @param appVerType
* @param version
*/
public static void deleteMiniApp(String appId, int appVerType, String version)
```

Sample code:

Note：

©2013-2022 Tencent Cloud. All rights reserved.

The parameter `appId` is the ID of the mini program.

The parameter 'appVerType' refers to the type of the mini program. You can refer to the mini program type description.

The parameter 'version' is the version number of the mini program.

```
//Delete a mini program with a specific ID, type and version number
TmfMiniSdk.deleteMiniApp(appId, appVerType,version);
```

©2013-2022 Tencent Cloud. All rights reserved.

Searching mini programs

Last updated：2024-06-27 11:08:48

The mini program SDK provides an online search API for mini programs. This API allows you to search for mini programs by keywords and categories.

API description:

Note：

The 'SearchOptions' parameter is used to specify the keywords and category information for the mini program search;

The 'MiniCallback' parameter is used to retrieve the search results of the mini program.

```
/**
 * Search for a mini program
 *
 * @param searchOptions
 * @param callback
 */
public static void searchMiniApp(SearchOptions searchOptions, MiniCallback<List<Min
```

Searching mini program by keyword

Sample code:

```
SearchOptions searchOptions = new SearchOptions("yourkeyword");
TmfMiniSDK.searchMiniApp(searchOptions, new MiniCallback<List<MiniApp>>() {
   @Override
   public void value(int code, String msg, List<MiniApp> data) {
       if (code == MiniCode.CODE_OK && data != null) {
           //Search succeeded. The list is not empty
       }else{
           //Search failed. The list is empty
       }
    }
});
```

Searching by single category

Sample code:

```
SearchOptions searchOptions = new SearchOptions("","category name","");
```

©2013-2022 Tencent Cloud. All rights reserved.

```
TmfMiniSDK.searchMiniApp(searchOptions, new MiniCallback<List<MiniApp>>() {
   @Override
   public void value(int code, String msg, List<MiniApp> data) {
       if (code == MiniCode.CODE_OK && data != null) {
           //Search succeeded. The list is not empty
       }else{
           //Search failed. The list is empty
       }
    }
});
```

Searching by dual category

Note：

The results of a dual category search are the intersection of the two categories.

Sample code:

```
SearchOptions searchOptions = new SearchOptions("","category name","category name 2
TmfMiniSDK.searchMiniApp(searchOptions, new MiniCallback<List<MiniApp>>() {
   @Override
   public void value(int code, String msg, List<MiniApp> data) {
       if (code == MiniCode.CODE_OK && data != null) {
           //Search succeeded. The list is not empty
       }else{
           //Search failed. The list is empty
       }
    }
});
```

Searching by keyword and category

Note：

When both keyword and category are set as search parameters, the results will be the intersection of the keyword and category.

Sample code:

```
SearchOptions searchOptions = new SearchOptions("keyword","category name","category
TmfMiniSDK.searchMiniApp(searchOptions, new MiniCallback<List<MiniApp>>() {
   @Override
   public void value(int code, String msg, List<MiniApp> data) {
       if (code == MiniCode.CODE_OK && data != null) {
```

©2013-2022 Tencent Cloud. All rights reserved.

```
            //Search succeeded. The list is not empty
        }else{
            //Search failed. The list is empty
        }
    }
});
```

©2013-2022 Tencent Cloud. All rights reserved.

Getting the list of recently accessed mini programs

Last updated：2024-06-27 11:08:29

The mini program SDK provides the following API to get the list of recently accessed mini programs.

```
/**
 * Get the list of recently accessed mini programs
 * @param callback
 */
public static void getRecentList(IRecentMiniCallback callback)
```

Sample code:

```
TmfMiniSDK.getRecentList(new IRecentMiniCallback() {
    @Override
    public void get(List<MiniApp> data) {
        //data is the recently accessed mini programs
    }
});
```

©2013-2022 Tencent Cloud. All rights reserved.

Pre-downloading mini program package

Last updated：2024-06-27 11:08:02

When a mini program is launched, it synchronizes and updates its version, which may cause a waiting period. To reduce this waiting time and improve user experience, the following API is provided to pre-download mini program packages.

Pre-download a single mini program package

API description:

```
/**
 * Pre-download main package of a mini program
 *
 * @param preDownloadInfo Pre-Downloading Info
 * @param callback Download callback
 */
public static void preDownloadPkg(PreDownloadInfo preDownloadInfo, IDownloadCallbac
```

Sample code:

```
PreDownloadInfo downloadInfo = new PreDownloadInfo("appId");
TmfMiniSDK.preDownloadPkg(downloadInfo, new IDownloadCallback() {
    @Override
    public void onFinish(DownloadInfo downloadInfo) {

    }

    @Override
    public void onError(DownloadInfo downloadInfo) {

    }
});
```

Pre-download multiple mini program main packages

API description:

```
 /**
 * Pre-download main packages of mini programs
```

©2013-2022 Tencent Cloud. All rights reserved.

```
     * @param preDownloadInfos Pre-Downloading Info
     * @param callback Download callback
     */
    public static void preDownloadPkg(List<PreDownloadInfo> preDownloadInfos, IDownload
```

Sample code:

```
PreDownloadInfo downloadInfo = new PreDownloadInfo("appId");
PreDownloadInfo downloadInfo2 = new PreDownloadInfo("appId2");
ArrayList<PreDownloadInfo>infos = new ArrayList<>();
infos.add(downloadInfo);
infos.add(downloadInfo2);
TmfMiniSDK.preDownloadPkg(infos, new IDownloadCallback() {
    @Override
    public void onFinish(DownloadInfo downloadInfo) {

    }

    @Override
    public void onError(DownloadInfo downloadInfo) {

    }
});
```

©2013-2022 Tencent Cloud. All rights reserved.

Multi-process interaction

Last updated：2024-06-27 11:07:32

The mini program SDK creates a separate process for each mini program, isolating it from the host application's process.
The SDK provides a unified API for communication between the mini program process and the host process.

Check if current process is a mini program process

API description:

```
/**
 * Whether the current process is a mini program process
 * @param context
 * @
 */
public static boolean isMiniProcess(Context context)
```

Host process calls mini program process plugin

1. To implement a sub-process plugin extension

Implementation steps:
Inherit BaseIpcPlugin.
Annotate the class with @IpcMiniPlugin.
Annotate the invoke method with @IpcEvent and define the plugin event name.
Sample code:

```
@IpcMiniPlugin
public class MiniProcessIpcPlugin extends BaseIpcPlugin {
    public static final String EVENT = "MiniProcessPlugin";

    @Override
    @IpcEvent(EVENT)
    public void invoke(IpcRequestEvent req) {

        String value = req.data.getString("key");

        Log.d(ModuleApplet.TAG, "current process:" + ProcessUtil.getProcessName(req
        Bundle resp = new Bundle();
```

©2013-2022 Tencent Cloud. All rights reserved.

```
        resp.putString("key", "i am ok");
        req.ok(resp);
    }
}
```

2. Plug-ins that call mini program process in main process

API description

```
/**
 * Call a mini-program process plugin for mini program introduction, and this metho
 *
 ///@param appId Mini program ID
 * @param eventId
 * @param request
 * @param callback
 */
public static void callMiniProcessPlugin(String appId, String eventId, Bundle reque

/**
 * Call a mini program process plugin for develop/preview mini program, and this me
 *
 ///@param appId Mini program ID
 * @param appVerType Mini program type. See MiniApp
 * @param eventId
 * @param request
 * @param callback
 */
public static void callMiniProcessPlugin(String appId, int appVerType, String event
```

Sample code:

```
TmfMiniSDK.callMiniProcessPlugin("", MiniProcessIpcPlugin.EVENT, bundle, new IpcCal
    @Override
    public void result(boolean isSucc, Bundle response) {
        Log.d(ModuleApplet.TAG, "current process:" + ProcessUtil.getProcessName(Mod
    }
});
```

Calling host process plugin in mini program process

1. To implement a host process plugin extension

Implementation steps：

©2013-2022 Tencent Cloud. All rights reserved.

Inherit BaseIpcPlugin.

Annotate the class with @IpcMainPlugin.

Annotate the invoke method with @IpcEvent and define the plugin event name.

Sample code:

```
@IpcMainPlugin
public class MainProcessIpcPlugin extends BaseIpcPlugin {
    public static final String EVENT = "MainProcessPlugin";

    @Override
    @IpcEvent(EVENT)
    public void invoke(IpcRequestEvent req) {
        String value = req.data.getString("key");

        Log.d(ModuleApplet.TAG, "current process|" + ProcessUtil.getProcessName(req
        Bundle resp = new Bundle();
        resp.putString("key", "i am ok");
        req.ok(resp);
    }
}
```

2. Calling host process plugin in mini program process

API description

Parameters：

eventId: Indicates event name defined by host process

request: Indicates parameters passed from mini program process to host process plugin

ipcCallBack: Indicates callback to receive the host process

```
/**
 * Calls the main process Plugin, and this method can only be called in the process
 *
 * @param eventId
 * @param request
 * @param callback
 */
public static void callMainProcessPlugin(String eventId, Bundle request, IpcCallbac
```

Sample code:

```
TmfMiniSDK.callMainProcessPlugin(MainProcessIpcPlugin.EVENT, bundle, new IpcCallbac
    @Override
    public void result(boolean isSucc, Bundle response) {
        Log.d(ModuleApplet.TAG, "current process:" + ProcessUtil.getProcessName(Mod
    }
```

©2013-2022 Tencent Cloud. All rights reserved.

```
});
```

©2013-2022 Tencent Cloud. All rights reserved.

Mini program file management

Last updated：2024-06-27 11:00:34

The mini program file management API allows for the conversion and creation of file paths between the mini program and the host application's local storage.

Note：

Mini program file path: A path starting with wxfile://, used by mini program developers. The mini program SDK maps these paths to local file paths within the host application.

Host app local file path: An absolute path in the host application's storage, e.g., /data/data/com.tencent.miniapp.demo/app_T1701421723ASSNID/2121/files/mini/.

Get IMiniAppFileManager

All mini program files are managed by IMiniAppFileManager which can be obtained from the mini program context IMiniAppContext. IMiniAppContext can be accessed via BaseJsPlugin.

```
IMiniAppFileManager fileManager = mMiniAppContext.getManager(IMiniAppFileManager.cl
```

CREATING FILES IN THE MINI PROGRAM TEMPORARY DIRECTORY

The SDK supports creating files in the mini program's cache directory and returns the local path for the host app to use.

Sample code:

```
String tmpPath = fileManager.getTmpPath(".jpg");
```

Converting absolute paths to wxfile paths

The SDK can convert file paths created in the mini program's cache directory to paths usable by the mini program.

Sample code:

```
String wxfile = fileManager.getWxFilePath(path);
```

wxfile paths converted to absolute paths

The SDK supports converting wxfile paths in the cache directory to local paths.

Sample code:

```
String path = fileManager.getAbsolutePath(wxfile);
```

©2013-2022 Tencent Cloud. All rights reserved.

©2013-2022 Tencent Cloud. All rights reserved.

Custom mini program capabilities
Custom mini program APIs

Last updated：2024-06-27 10:58:43

If a mini program needs to call certain capabilities provided by the host app that are not implemented or cannot be implemented by the mini program SDK, developers can register custom APIs to enable these capabilities.

Implementing custom APIs in the host app

The host app implements the custom mini program API capability by inheriting the BaseJsPlugin.

Note：

Extend `BaseJsPlugin` and define it with the annotation @JsPlugin(secondary = true).

Define a method, which can have only one parameter of the `RequestEvent` type.

Then, define the annotation @JsEvent("event name") in the method. When the mini program JS calls the "event name", the method modified by @JsEvent will be called.

@JsEvent supports defining multiple event names.

The data can be returned synchronously or asynchronously (only one method can be selected for the same event).

You can use `sendState` to send multiple intermediate states to the mini program. After the final sendState call, you must call either ok or fail to indicate the end of the process.

Sample code:

```
@JsPlugin(secondary = true)
public class CustomPlugin extends BaseJsPlugin {
   @JsEvent("customAsyncEvent")
   public void custom(final RequestEvent req) {
       // Get the parameters
       //req.jsonParams
       // Return the data asynchronously
       //req.fail();
       //req.ok();
       JSONObject jsonObject = new JSONObject();
       try {
           jsonObject.put("key", "test");
       } catch (JSONException e) {
           e.printStackTrace();
       }
       req.ok(jsonObject);
    }
```

©2013-2022 Tencent Cloud. All rights reserved.

```
    @JsEvent("customSyncEvent")
    public String custom1(final RequestEvent req) {
        // Get the parameters
        //req.jsonParams
        //Return the data synchronously
        JSONObject jsonObject = new JSONObject();
        try {
            jsonObject.put("key", "value");
        } catch (JSONException e) {
            throw new RuntimeException(e);
        }
        return req.okSync(jsonObject);
     }

    /**
     * Test coverage system API
     * @param req
     */
    @JsEvent("getAppBaseInfo")
    public void getAppBaseInfo(final RequestEvent req) {
        // Get the parameters
        //req.jsonParams
        // Return the data asynchronously
        //req.fail();
        //req.ok();
        JSONObject jsonObject = new JSONObject();
        try {
            jsonObject.put("key", "test");
        } catch (JSONException e) {
            e.printStackTrace();
        }
        req.ok(jsonObject);
     }

    @JsEvent("testState")
    public void testState(final RequestEvent req) {

      try {
          //Call back intermediate state
          req.sendState(req, new JSONObject().put("progress", 1));
          req.sendState(req, new JSONObject().put("progress", 30));
          req.sendState(req, new JSONObject().put("progress", 60));
          req.sendState(req, new JSONObject().put("progress", 100));
      } catch (JSONException e) {
          e.printStackTrace();
      }
```

©2013-2022 Tencent Cloud. All rights reserved.

```
        JSONObject jsonObject = new JSONObject();
        try {
            jsonObject.put("key", "test");
            req.ok(jsonObject);
        } catch (JSONException e) {
          throw new RuntimeException(e);
        }
      }
}
```

Configurecustom APIs

Refer to the following template to configure the custom API parameters:

Note：

In the configuration template, custom API information is stored in a JSON file.

The extApi field in the outer layer of the JSON file holds an array of custom APIs. Each object within extApi represents the configuration of a custom API.

The overwriteWxApi field in the outer layer of the JSON file determines whether custom events override the default mini program API implementation.

Setting this to true means the SDK's built-in mini program APIs will be overridden.

Each custom API configuration must include the following three key attributes:

name: The event name of the custom API, which must match the event name defined by @JsEvent.

sync: Indicates whether the call is synchronous and should be consistent with the return method in the "Creating API" example.

params: Describes the parameters of the custom API.

Template：

```
{
 "extApi": [
    {
      //Event name, must match the event defined in the "Create API" example with @
     "name": "customSyncEvent",
     //Whether the call is synchronous, consistent with the return method in the "C
     "sync": true,
     //Define parameters
     //a. JSON format can be nested
     //b. string parameter value can be to ""
     //c.  numeric parameter value can be to 0
     "params": {
       "name": "",
       "age": 0,
```

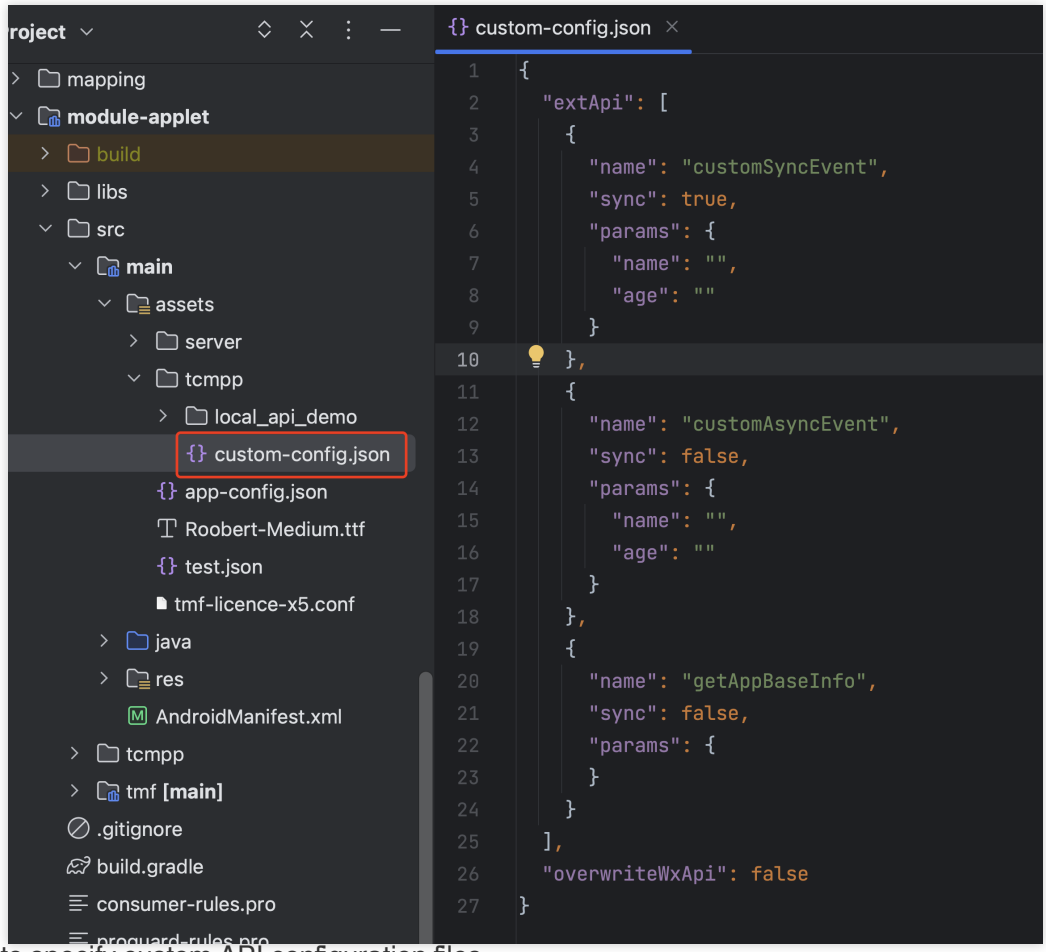
©2013-2022 Tencent Cloud. All rights reserved.

```
        "object": {
          "key": "",
        }
       }
    },
      {
      "name": "customAsyncEvent2",
      "sync": false,
      "params": {
        "name": "",
        "age": ""
      }
   ],
  //true: If the custom API event name matches a built-in method name in the mini pr
  //API, and the final call will invoke the developer's custom API.
  "overwriteWxApi": false
}
```

Specifying the custom API configuration file path

Place the defined API configuration file in the project's assets directory (the file name and path can be defined by the developer).

©2013-2022 Tencent Cloud. All rights reserved.



Code to specify custom API configuration files

```
@ProxyService(proxy = MiniAppProxy.class)
public class MiniAppProxyImpl extends BaseMiniAppProxyImpl {
     @Override
   public MiniConfigData configData(Context context, int configType, JSONObject par
        if(configType == MiniConfigData.TYPE_CUSTOM_JSAPI) {
            //Custom JsApi configuration
            MiniConfigData.CustomJsApiConfig customJsApiConfig = new MiniConfigData.
            customJsApiConfig.jsApiConfigPath = "tcmpp/custom-config.json";

            return new MiniConfigData
                    .Builder()
                   .customJsApiConfig(customJsApiConfig)
                    .build();
        }

      return null;
}
```

Mini program developer calling custom API

©2013-2022 Tencent Cloud. All rights reserved.

Example

The host app defines two custom APIs: customAsyncEvent and getAppBaseInfo. The configuration file is as follows:

```
{
  "extApi": [
    {
      "name": "customSyncEvent",
      "sync": true,
      "params": {
        "name": "",
        "age": ""
      }
    },
    {
      "name": "customAsyncEvent",
      "sync": false,
      "params": {
        "name": "",
        "age": ""
      }
    },
    {
      "name": "getAppBaseInfo",
      "sync": false,
      "params": {
      }
    }
  "overwriteWxApi": true
}
```

The mini program developers can access custom APIs as follows:

```
wx.customAsyncEvent({"name":"123","age":"18"})
wx.getAppBaseInfo()//his will override the system API and call the custom API.
```

Note：

Since the overwriteWxApi property in the custom API configuration file is set to true, calling wx.getAppBaseInfo in the mini program will invoke the host application's custom implementation of getAppBaseInfo.

Troubleshooting custom APIs

If there are errors when calling custom APIs, see Android FAQs.

©2013-2022 Tencent Cloud. All rights reserved.

Custom API example

Last updated：2024-06-27 10:55:40

Launching an Activity in a custom API

Start a new Activity in the plugin and retrieve the data returned by the Activity.

Sample code:

Extending BaseJsPlugin to determine the custom API event name:

```
@JsPlugin(secondary = true)
public class CustomPlugin extends BaseJsPlugin {
    @JsEvent("testStartActivityForResult")
    public void testStartActivityForResult(final RequestEvent req) {
  }
}
```

Add a listener for the Activity result and remove it once processing is complete:

Note：

It is recommended to pair the registration and removal of listeners to avoid memory leak.

```
@JsPlugin(secondary = true)
public class CustomPlugin extends BaseJsPlugin {
    @JsEvent("testStartActivityForResult")
    public void testStartActivityForResult(final RequestEvent req) {
      Activity activity = req.activityRef.get();
      TmfMiniSDK.addActivityResultListener(new IActivityResultListener() {
          @Override
          public boolean doOnActivityResult(int requestCode, int resultCode, Intent
              TmfMiniSDK.removeActivityResultListener(this);

              Log.i(ModuleApplet.TAG, data.getStringExtra("key"));
              req.ok();
              return true;
          }
      });
  }
}
```

To start a new activity:

Note：

The requestCode must be >= 1000000 to avoid conflicts with internal request codes, which could cause unexpected issues.

©2013-2022 Tencent Cloud. All rights reserved.

```
@JsPlugin(secondary = true)
public class CustomPlugin extends BaseJsPlugin {
    @JsEvent("testStartActivityForResult")
    public void testStartActivityForResult(final RequestEvent req) {
      Activity activity = req.activityRef.get();
      TmfMiniSDK.addActivityResultListener(new IActivityResultListener() {
          @Override
          public boolean doOnActivityResult(int requestCode, int resultCode, Intent
              TmfMiniSDK.removeActivityResultListener(this);

              Log.i(ModuleApplet.TAG, data.getStringExtra("key"));
              req.ok();
              return true;
          }
      });

      //Note: requestCode must be >= 1000000 to avoid conflicts with internal reque
      activity.startActivityForResult(new Intent(activity, TransActivity.class), 10
  }
}
```

Calling third-party apps in custom APIs

When calling other third-party apps (e.g., for sharing, payment, login) within a custom API, ensure that the operation returns directly to the mini program instead of the host app.

Firstly, developers create transparent auxiliary Activity.

Define a transparent helper activity, such as TransActivity. The transparency settings depend on the theme configuration, which is related to the system Activity that your app's Activity inherits from.

```
  <activity android:name=".activity.TransActivity"
    android:exported="false"
    android:screenOrientation="portrait"
    android:theme="@style/TransparentTheme"/>
```

And then invoke that business logic in the TransActivity.

```
public class TransActivity extends AppCompatActivity {
   @Override
   protected void onCreate(@Nullable Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.applet_activity_tran);

       /************************
```

©2013-2022 Tencent Cloud. All rights reserved.

```
                Business development
        ************************/

        //After completing the business logic, return data and call finish
        Intent intent = new Intent();
        intent.putExtra("key", "value");
        setResult(RESULT_OK, intent);
        finish();
    }
}
```

The plug-in starts the activity and listens for the data to return.

```
  @JsPlugin(secondary = true)
public class CustomPlugin extends BaseJsPlugin {
    @JsEvent("testStartActivityForResult")
    public void testStartActivityForResult(final RequestEvent req) {
      Activity activity = req.activityRef.get();
      TmfMiniSDK.addActivityResultListener(new IActivityResultListener() {
          @Override
          public boolean doOnActivityResult(int requestCode, int resultCode, Intent
              TmfMiniSDK.removeActivityResultListener(this);

              Log.i(ModuleApplet.TAG, data.getStringExtra("key"));
              req.ok();
              return true;
          }
      });

      //Note: requestCode must be >= 1000000 to avoid conflicts with internal reque
      activity.startActivityForResult(new Intent(activity, TransActivity.class), 10
  }
}
```

©2013-2022 Tencent Cloud. All rights reserved.

Custom mini program view components

Last updated：2024-06-27 10:55:30

Mini program view component extension description

Users of the mini program SDK can customize native client components to support mini programs. Mini program developers can create and manipulate native components on mini program pages and communicate with them using specific mini program APIs. Android native client components are divided into two types:
Non-layered native components: These components render above the mini program page, do not support zIndex, and always overlay other mini program components, potentially obscuring content. They are drawn as regular Views;
Layered native components: These components render within the mini program page, support zIndex, and respect the component hierarchy within the page. They need to be drawn as Surfaces within the mini program page.
Note：
To implement extended components, modifications are required both in the mini program by the mini program developer and in the host app by the host app developer. The combination of these modifications enables the extension of mini program view components.

Implementation of native component extensions

App developers need to implement specific proxies to create native components and respond to operations on them when the mini program calls custom component APIs. The following sections describe the implementation for non-layered and layered components separately:

Non-layered native component extension

Override the proxy with the following settings to allow the host to customize the creation of non-layered native components:

```
@ProxyService(proxy = ExternalElementProxy.class)
public class MyExternalElementProxy extends ExternalElementProxy{}
```

The proxy needs to implement the following methods:
1. Insert non-layered component This method is called when the mini program inserts a non-layered native component into the page. Developers need to implement this method and add the custom component as a child View to the container provided by the parent parameter.

```
/**
 * Create non-layered components. When the mini program creates custom non-layered
```

©2013-2022 Tencent Cloud. All rights reserved.

```
 *
 * @param widgetId The unique ID of the component created by the mini program
 * @param widgetContext Context of the mini program component, which is used to pas
 * @ param type Name of the component type created by the mini program
 * @param parent Parent container hosting native components
 * @param params: Arguments passed when the mini program creates the component
 */
public abstract void handleInsertElement(long widgetId, ExternalWidgetContext widge
                                         String type, ViewGroup parent, JSONObject
```

2. Update Non-layered Component This method is called to notify the app when the position or size of the native component changes within the mini program. The parent container layout of the native component will adapt to the new style, and the native component itself can adjust as needed.

```
/**
 * update with layer component style, when the applet to update the style of the cu
 *
 * @ param widgetId Mini program component ID
 * @param widgetContext Context of the mini program component, which is used to pas
 * @param params Parameters passed when the mini program updates a component
 */
public abstract void handleUpdateElement(long widgetId, ExternalWidgetContext widge
                                         JSONObject params);
```

3. Operate non-layered component Operate Non-layered Component: This method is called when the mini program sends an event to the native component (e.g., button click, state change). The specific content of the event needs to be defined by the developer in the params.

```
/**
 * Operation with layer components, when the Mini programs need to send instruction
 *
 * @param widgetId Mini program component ID
 * @param widgetContext Context of the mini program component, which is used to pas
 * @param params Parameters passed when the mini program updates a component
 */
public abstract void handleOperateElement(long widgetId, ExternalWidgetContext widg
                                          JSONObject params);
```

4. Remove non-layered component This method is called to notify the app when the mini program deletes an added native component. At this point, the parent container of the component will be removed, and the application should destroy the component.

```
/**
 *Delete non-layered components
 *
 * @param widgetId Mini program component ID
```

©2013-2022 Tencent Cloud. All rights reserved.

```
 * @param widgetContext Context of the mini program component, which is used to pas
 */
public abstract void handleRemoveElement(long widgetId, ExternalWidgetContext widge
```

Mini program non-layered native component context

The context for mini program components includes methods that allow native components to send messages to the corresponding mini program components. The onExternalElementEvent method sends an onExternalElementEvent event directly to the mini program, which should capture and handle this event. The callbackSuccess and callbackFail methods are used to call the success or fail callbacks provided by the mini program when it invokes an API to send events to the application.

```
/**
 * Sends onExternalElementEvent event to the mini program.
 *
 * @param jsonObject JSON data carried by the event.
 */
public final void onExternalElementEvent(JSONObject jsonObject);

/**
 * Calls the success callback provided by the mini program.
 *
 * @param jsonObject JSON data carried by the callback, which can be null.
 */
public final void callbackSuccess(JSONObject jsonObject);

/**
 *  Calls the `fail` callback provided by the mini program.
 *
 * @param jsonObject JSON data carried by the callback, which can be null.
 /// @param message Error message description
 */
public final void callbackFail(JSONObject jsonObject, String message);
```

Same-layer native component extensions

If the mini program requests the creation of a same-layer native component, the application needs to implement the following proxy:

```
@ProxyService(proxy = ExternalEmbeddedWidgetClient.class)
public class MyExternalEmbeddedElementProxy extends ExternalEmbeddedWidgetClient{}
```

©2013-2022 Tencent Cloud. All rights reserved.

1. Unlike non-layered components, same-layer components involve the creation and destruction of a Surface embedded in the page, thus providing additional lifecycle callbacks. Developers can override these methods to handle lifecycle events of same-layer components:

```
 * @param tagName Tag of same-layer component in DOM
 * @param attributes Attributes of same-layer components
 */
public abstract void onInit(Context context, long widgetId, String tagName, Map<Str

/**
 * Callback when the Surface of the same-layer component is created.
 *
 * @param widgetId Same-layer component ID
 * @param surface Surface associated with the same-layer component
 */
public abstract void onSurfaceCreated(long widgetId, Surface surface);

/**
 * Callback when the Surface of the same-layer component is destroyed.
 *
 * @param widgetId Same-layer component ID
 * @param surface Surface associated with the same layer component
 */
public abstract void onSurfaceDestroyed(long widgetId, Surface surface);

/**
 * Callback for touch events on the same-layer component.
 *
 * @param widgetId Same-layer component ID
 * @param event The touch event
 * @return Whether the same-layer components consumed touch event
 */
public abstract boolean onTouchEvent(long widgetId, MotionEvent event);

/**
 * Callback when the drawing area of the same-layer component changes.
 *
 * @param widgetId Same-layer component ID
 * @param rect: New drawing area
 */
public abstract void onRectChanged(long widgetId, Rect rect);

/**
 * Requests a redraw of the same-layer component.
 *
 * @param widgetId Same-layer component ID
```

©2013-2022 Tencent Cloud. All rights reserved.

```
 */
public abstract void onRequestRedraw(long widgetId);

/**
 * The visibility of the same-layer component changes.
 *
 * @param widgetId Same-layer component ID
 Is * @param visibility Whether it is visible
 */
public abstract void onVisibilityChanged(long widgetId, boolean visibility);

/**
 * Same-layer components become available
 *
 * @param widgetId Same-layer component ID
 */
public abstract void onActive(long widgetId);

/**
 * Same-layer components become unavailable
 *
 * @param widgetId Same-layer component ID
 */
public abstract void onDeActive(long widgetId);

/**
 * Same-layer component node is destroyed
 *
 * @param widgetId Same-layer component ID
 */
public abstract void onDestroy(long widgetId);

/**
 * The current mini program page enters the pause state
 *
 * @param widgetId Same-layer component ID
 */
public abstract void webViewPause(long widgetId);

/**
 *The current mini program page enters the resume state.
 *
 * @param widgetId Same-layer component ID
 */
public abstract void webViewResume(long widgetId);

/**
```

©2013-2022 Tencent Cloud. All rights reserved.

```
 * The current mini program page is destroyed
 *
 * @param widgetId Same-layer component ID
 */
public abstract void webViewDestroy(long widgetId);

/**
 * Mini program enters resume state
 *
 * @param widgetId Same-layer component ID
 */
public abstract void nativeResume(long widgetId);

/**
 * Mini program enters pause state
 *
 * @param widgetId Same-layer component ID
 */
public abstract void nativePause(long widgetId);

/**
 * Mini program is destroyed
 *
 * @param widgetId Same-layer component ID
 */
public abstract void nativeDestroy(long widgetId);
```

2. When the mini program actually inserts an same-layer component into the page, it will call the application's handleInsertXWebExternalElement API. Client developers should implement this method to render the appropriate component type, as specified by type, onto the Surface associated with widgetId.

```
/**
 * Inserts a same-layer component.
 *
 * @param widgetId Same-layer component ID
 * @param widgetContext The mini program context associated with the same-layer com
 * @ param type The type of the same-layer component.
 * @param req Parameters passed through by the mini program.
 */
public abstract void handleInsertXWebExternalElement(long widgetId, ExternalWidgetC
```

3. When the size or style of the component changes, the application will be notified through this method:

```
/**
 * Update same-layer component style
 *
 * @param widgetId Same-layer component ID
```

©2013-2022 Tencent Cloud. All rights reserved.

```
 * @param widgetContext The mini program context associated with the same-layer com
 * @param req Parameters passed by the mini program.
 */
public abstract void handleUpdateXWebExternalElement(long widgetId, XWebExternalWid
```

4. When the mini program component needs to send events to the native component for operations, it will notify the application through this method:

```
/**
 * Operates on the same-layer component.
 *
 * @param widgetId Same-layer component ID
 * @param widgetContext The mini program context associated with the same-layer com
 * @param req Parameters passed by the mini program.
 */
public abstract void handleOperateXWebExternalElement(long widgetId, XWebExternalWi
```

5. When the mini program deletes a same-layer native component, it will notify the application through this method:

```
/**
 * Remove same-layer components
 *
 * @param widgetId Same-layer component ID
 * @param widgetContext The mini program context associated with the same-layer com

*/
public abstract void handleRemoveXWebExternalElement(long widgetId, XWebExternalWid
```

Mini program same-layer native component context

Similar to non-layered components, same-layer components can send onXWebExternalElementEvent events to the mini program through the component's context. Additionally, when the mini program calls APIs to operate on native components, it can use the provided callback methods.

```
/**
 * Sends onXWebExternalElementEvent event to the mini program.
 *
 * @param jsonObject JSON data carried by the event
 */
public final void onXWebExternalElementEvent(JSONObject jsonObject);

/**
 * Calls the success callback provided by the mini program.
 *
 * @param jsonObject JSON data carried by the callback, which can be null.
 */
```

©2013-2022 Tencent Cloud. All rights reserved.

```
public final void callbackSuccess(JSONObject jsonObject);

/**
 * Calls the `fail` callback provided by the mini program.
 *
 * @param jsonObject jsonObject JSON data carried by the callback, which can be nul
 /// @param message Error message description
 */
public final void callbackFail(JSONObject jsonObject, String message);
```

Mini program extension implementation

To insert a custom client-side component into the page, the mini program first needs to include an external-element node in the WXML:

```
<external-element
    id="comp1"
    type="maTestView"
    _insert2WebLayer="true"
    style="width: 200px;height: 100px;"
    bindexternalelementevent="handleEvent"
></external-element>
```

Here, type should match the component type name agreed upon with the application. _insert2WebLayer indicates whether the component is a same-layer component (true for same-layer, requiring client-side same-layer component proxy; false for non-layered, requiring client-side non-layered proxy). bindexternalelementevent can capture native onExternalElementEvent or onXWebExternalElementEvent events, with callback parameters including: OnXWebExternalElementEvent, callback parameters include:

```
{
    target,
    currentTarget,
    timeStamp,
    touches,
    detail, // Parameters passed by native
}
```

Next, the mini program creates a context associated with this node using its ID:

```
this.ctx = wx.createExternalElementContext('comp1');
```

This method notifies the application to create the corresponding native component at the node's position. The mini program can then use this context to send events to and operate on the native component:

©2013-2022 Tencent Cloud. All rights reserved.

```
this.ctx.call({
  params1: {
    name: 'name1',
    age: 11
  },
  params2: {
    name: 'name2',
    age: 22
  },
  success: (e) => {
    console.log('====operate success=====', e)
  },
  fail: (e) => {
    console.log('====operate fail=====', e)
  },
  complete: (e) => {
    console.log('====operate complete=====', e)
  }
})
```

©2013-2022 Tencent Cloud. All rights reserved.

Custom SDK Capabilities
Custom mini program UI

Last updated：2024-06-27 11:02:53

The mini program SDK allows developers to customize certain native UI elements. Follow the guidelines below to configure this customization.
To customize the mini program UI, developers need to extend and implement AbsMiniUiProxy. Sample code:

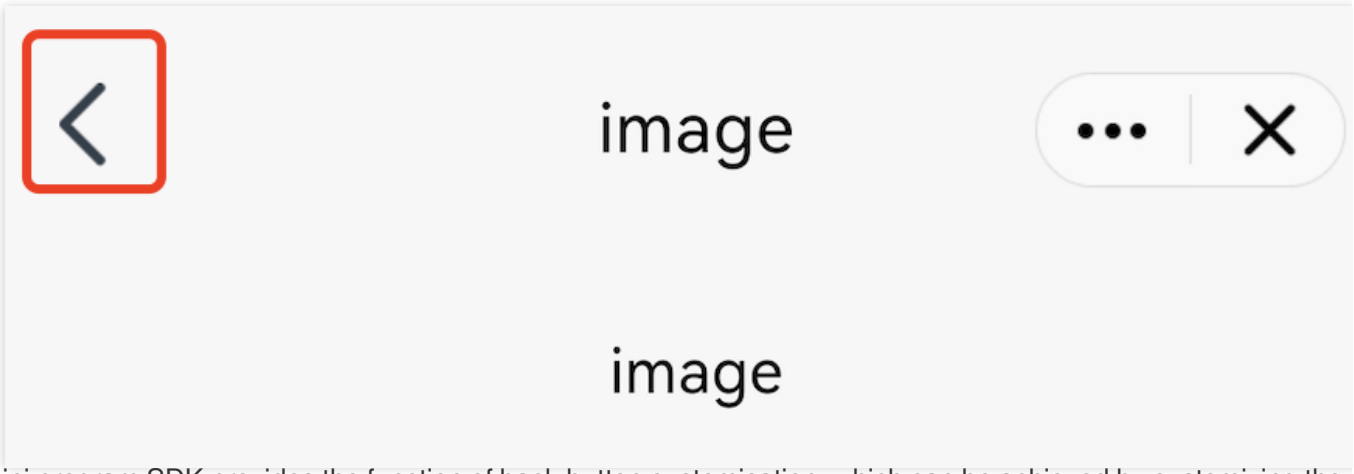
```
@ProxyService(proxy = IMiniUiProxy.class)
public class MiniUiProxyImpl extends AbsMiniUiProxy
```

Note：
You must use the @ProxyService(proxy = IMiniUiProxy.class) annotation to customize the UI.

Customizing navigation bar buttons

1. Back button

The figure of back button is as follows:



The mini program SDK provides the function of back button customisation, which can be achieved by customizing the back button and overriding the navBarBackRes method of AbsMiniUiProxy.
The API definition is as follows:
Method parameter mode: indicates the navBar title colour, 1 means black title, 0 means white title;
Method return value: the icon resource ID of the back button.

```
/**
 * Custom navigation bar return icon, aspect ratio = 24x43.
 * Call environment: subprocess
```

©2013-2022 Tencent Cloud. All rights reserved.

```
 /*** * @param mode navigation bar title.
 * @param mode navigation bar title colour, 1:black 0:white
 * @return
 */
@DrawableRes
int navBarBackRes(int mode);
```

Note：

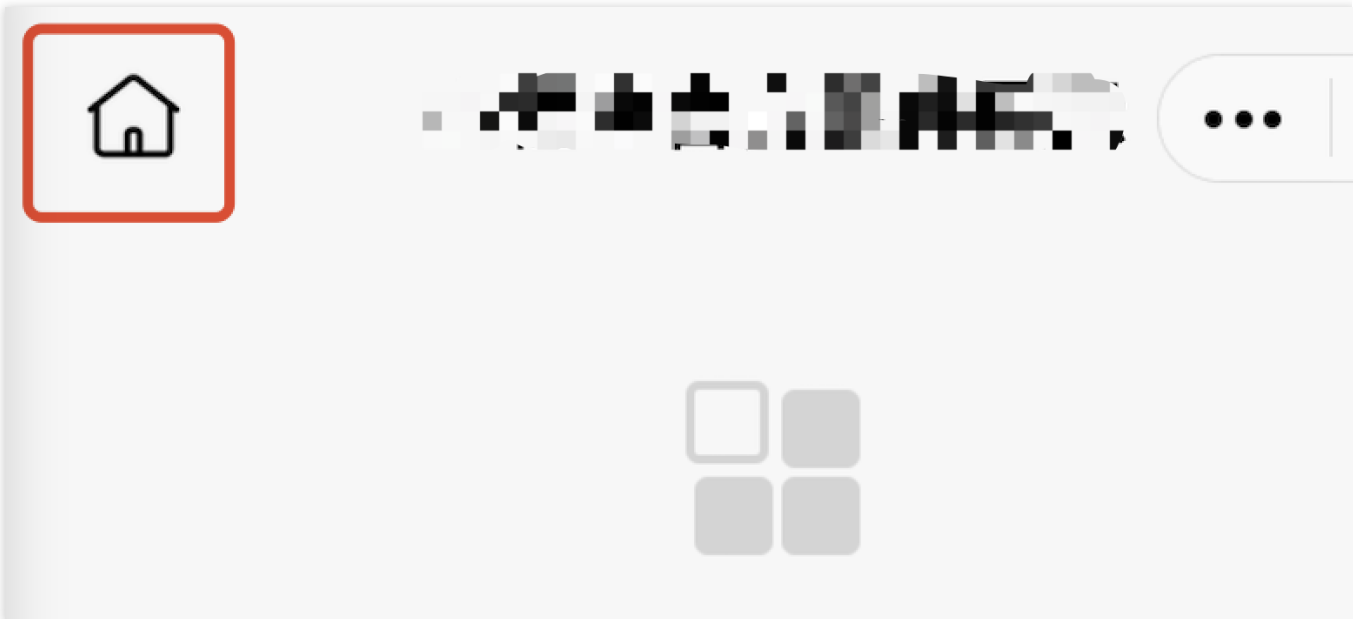
The aspect ratio required for the Back button icon resource is: 24x43.

Sample code:

```
@Override
public int navBarBackRes(int mode) {
    if(mode == 0) {//black
        return R.drawable.back_icon;//your black icon res
    }else {//white
        return R.drawable.white_icon;//your white icon res
    }
}
```

2. home button

The figure of the home button is as follows:



The mini program SDK provides home button customisation, which can be achieved by overriding AbsMiniUiProxy's homeButtonRes to customize the home button method.

The API definition is as follows:

Method parameter mode: indicates the title colour of the navigation bar, 1 means black title, 0 means white title;

Method return value: the icon resource ID of the returned button.

©2013-2022 Tencent Cloud. All rights reserved.

```
/**
 * Navigation bar back to home page icon, aspect ratio requirement = 48x48
 * Calling environment: subprocess
 * @param mode navigation bar title colour, 1:black 0:white
 * @param mode navigation bar title colour, 1:black 0:white
 * @return
 */
@DrawableRes
int homeButtonRes(int mode);
```

Note：

Size requirement: navigation bar home icon, aspect ratio = 48x48.

Sample code:

```
@Override
public int homeButtonRes(int mode) {

    if(mode == 0) {//black
        return R.drawable.home_black_icon;
    }else {
        return R.drawable.home_white_icon;
    }
}
```

3. More buttons

More buttons are shown below:

©2013-2022 Tencent Cloud. All rights reserved.

The mini program SDK provides more button customisation, which can be achieved by overriding AbsMiniUiProxy's moreButtonRes to customize the home button method.

The API definition is as follows:

Method parameter mode: indicates the title colour of the navigation bar, 1 means black title, 0 means white title;

Method return value: the icon resource ID of the returned button.

```
/**
 * Navbar return home icon, aspect ratio = 48x48.
 * Call environment: subprocess
 *
 * @param mode navigation bar title colour, 1:black 0:white
 * @return
 */
@DrawableRes
int moreButtonRes(int mode);
```

Note：

Size requirements: navigation bar home icon, aspect ratio = 80x59.

Sample code:

```
@Override
public int moreButtonRes(int mode) {

    if(mode == 0) {//black
        return R.drawable.more_black_icon;
    }else {
        return R.drawable.more_white_icon;
```

©2013-2022 Tencent Cloud. All rights reserved.

```
    }
}
```

4. Mini Program Close Button

The mini program close button is shown with the following intention:

The mini program SDK provides the function of close button customisation, which can be achieved in the following ways:

Customise the home button:

Override the closeButtonRes method of AbsMiniUiProxy.

API definition is as follows:

API Description:

Method parameter mode: means the title colour of the navigation bar, 1 means black title, 0 means white title.

Method return value: the icon resource ID of the returned button.

```
/**
 * Navigation bar return to home page icon icon, aspect ratio requirement = 48x48
 * Call environment: subprocess
 /* @param mode
 * @param mode navigation bar title colour, 1:black 0:white
 * @return
 */
@DrawableRes
int closeButtonRes(int mode);
```

Note：

The size of the icon is the same as the home icon in the navigation bar:

Size requirement: navigation bar home icon, aspect ratio = 80x59.

Sample code:

```
@Override
public int closeButtonRes(int mode) {

    if(mode == 0) {//black
```

©2013-2022 Tencent Cloud. All rights reserved.

```
        return R.drawable.close_black_icon;
    }else {
        return R.drawable.close_white_icon;
    }
}
```

5. Capsule Button Dividing Line

The intention of the mini program capsule button split line display is as follows:

The mini program SDK provides the function of custom capsule button split line, which can be achieved by overriding AbsMiniUiProxy's lineSplitBackgroundColor and customising the home button method.

The API definition is as follows:

Method return value: the background colour of the split line.

```
/**
 * Capsule button middle split line background colour
 * Calling environment: child process
 * @return
 * @return
 */
@DrawableRes
int lineSplitBackgroundColor();
```

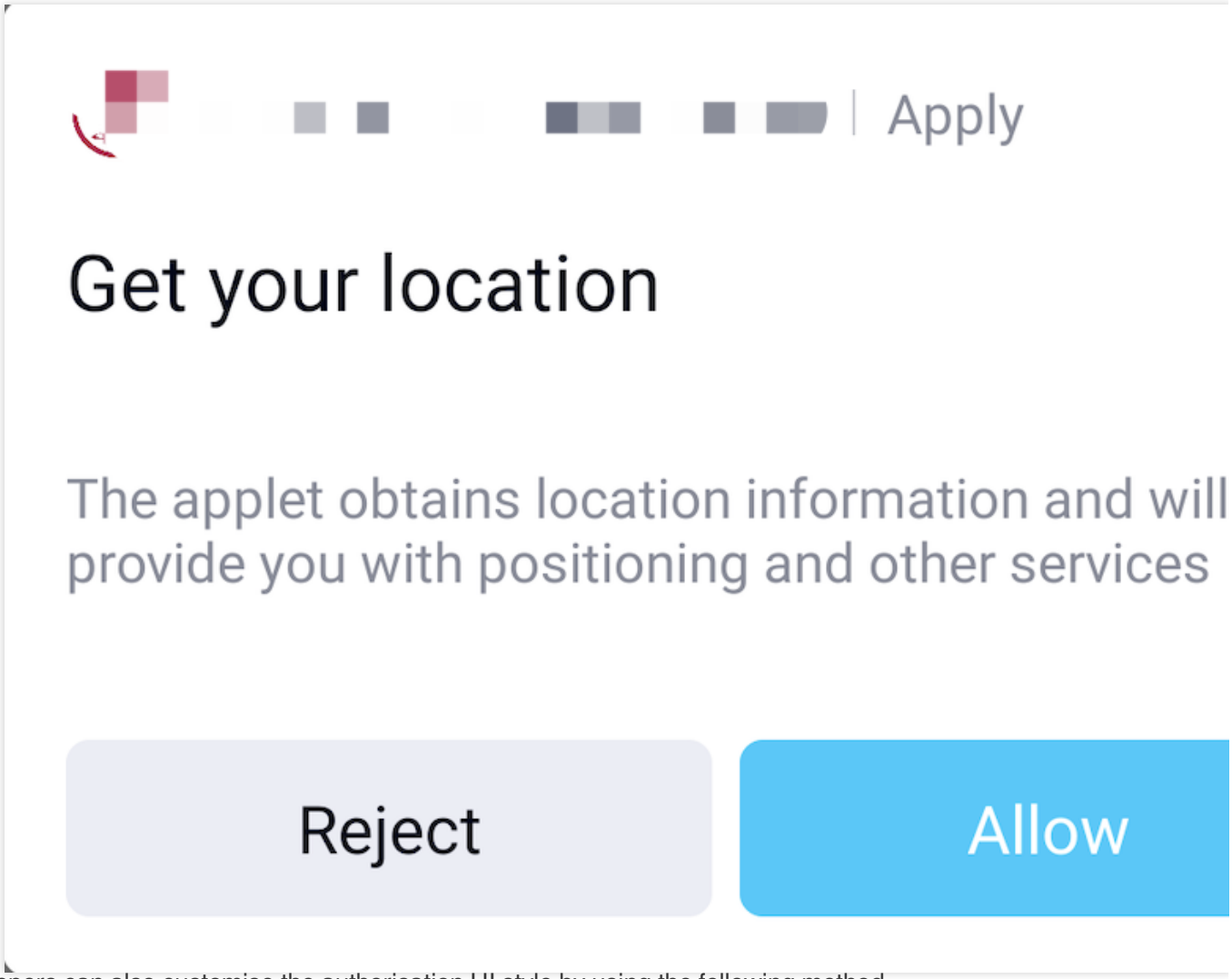
Sample code:

```
@Override
public int lineSplitBackgroundColor() {
return Color.RED;
}
```

Customised authorisation pop-ups

©2013-2022 Tencent Cloud. All rights reserved.

When the API called by the mini program requires authorisation, the SDK provides the following default authorisation UI style:



Developers can also customise the authorisation UI style by using the following method.

Customise the authorisation popup window:

Override the authView method of AbsMiniUiProxy;

The parameter Context represents the context of the mini program process;

The parameter MiniAuthInfo contains the authorisation information requested by the current mini program, see MiniAuthInfo for details;

The parameter IAuthView is used to set the custom popup view.

The API definition is as follows:

```
/**
 * Custom authorisation popup view
 * Call environment: subprocess
 *
 * @param context
 * @param authInfo
```

©2013-2022 Tencent Cloud. All rights reserved.

```
 * @param authView
 * @return true:custom authView;false:use built-in
 */
@Override
public boolean authView(Context context, MiniAuthInfo authInfo, IAuthView authView)
```

Note：

The return value of authView method indicates whether to use the custom authorisation view or not, the return value is false to use the default built-in authorisation view, the return value is true to use the custom authorisation view. RefuseListener and grantListener in MiniAuthInfo are the listeners for the authorisation popup window, must be set, otherwise it will cause the mini program authorisation exception.

Sample code:

View Layout File

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingTop="16dp"
    android:paddingRight="16dp"
    android:paddingBottom="15dp"
    android:background="@android:color/white">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="custom auth view"
        android:layout_centerHorizontal="true"/>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="150dp">

        <Button
            android:id="@+id/mini_auth_btn_refuse"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1.0"
            android:text="@string/applet_mini_reject" />

        <Button
            android:id="@+id/mini_auth_btn_grant"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
```

©2013-2022 Tencent Cloud. All rights reserved.

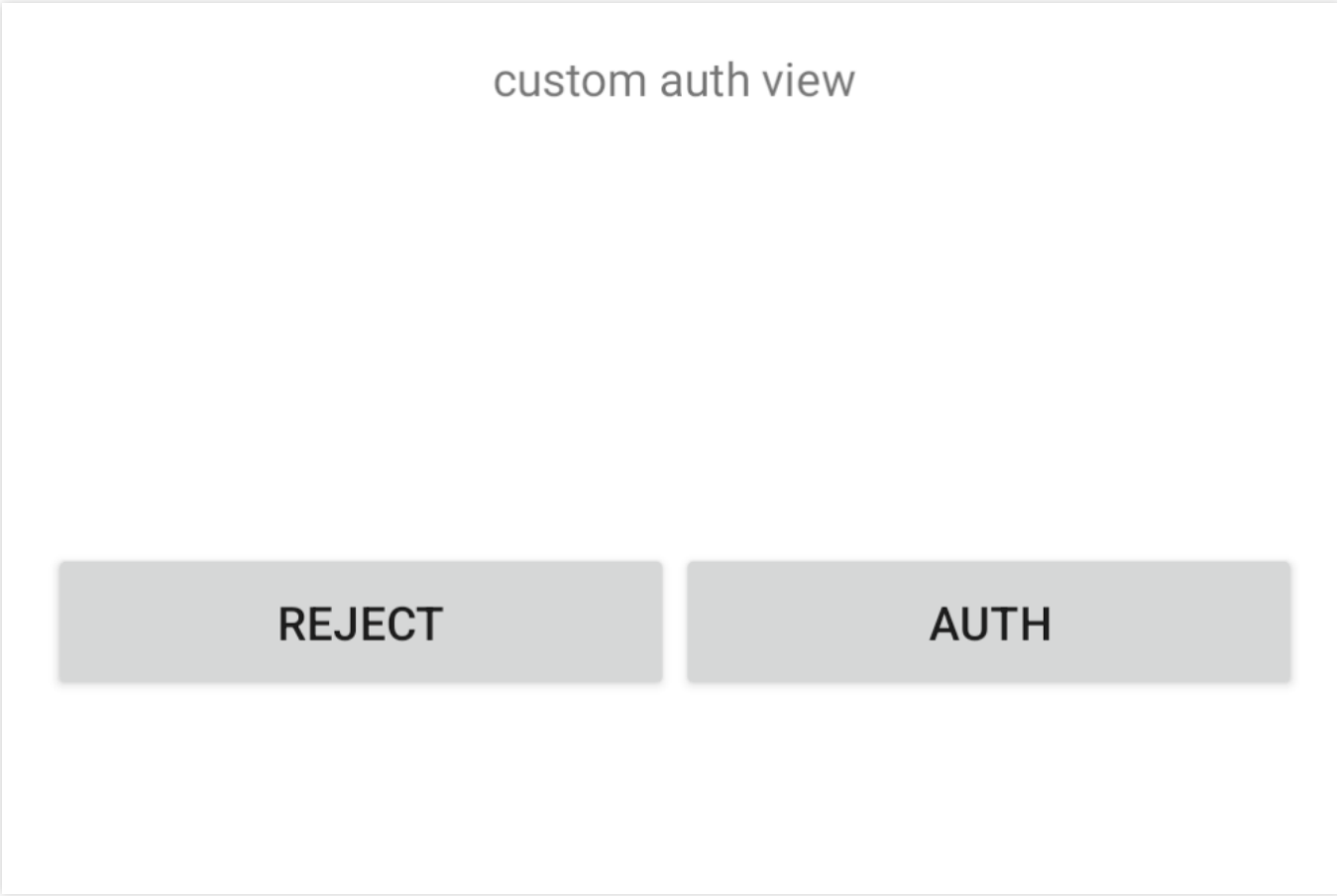
```
            android:layout_weight="1.0"
            android:text="@string/applet_mini_auth" />
    </LinearLayout>

</RelativeLayout>
```

Override the authView method (note that the method returns true):

```
@Override
public boolean authView(Context context, MiniAuthInfo authInfo, IAuthView authView)
    boolean isCustom = true;
    if (isCustom) {
        View view = LayoutInflater.from(context).inflate(R.layout.mini_auth_view, n
        //Must be set
        view.findViewById(R.id.mini_auth_btn_refuse).setOnClickListener(authInfo.re
        //Must be set
        view.findViewById(R.id.mini_auth_btn_grant).setOnClickListener(authInfo.gra

        //Return to Custom View
        authView.getView(view);
    }

    return isCustom;
}
```

Custom effects:

©2013-2022 Tencent Cloud. All rights reserved.



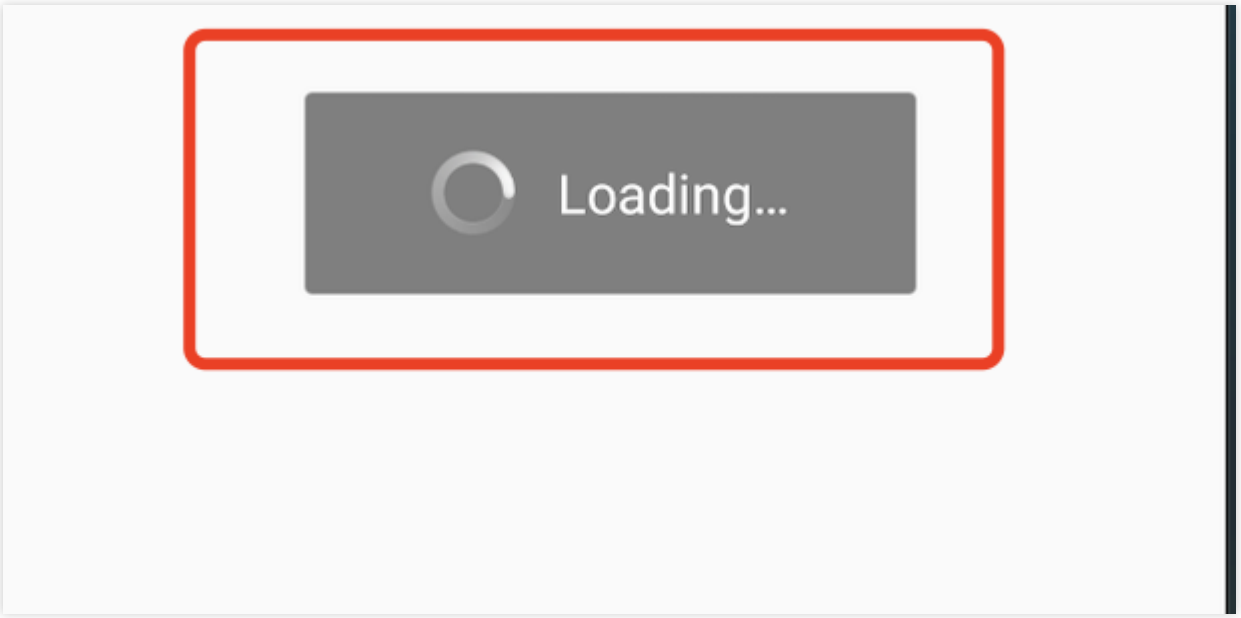
Customising the mini program Loading view

During the opening process of mini program, there are check update and start loading loading animations, both of which support customisation.

1. Customise the mini program check update Loading view

The default loading animation for checking update is shown in the following figure:

©2013-2022 Tencent Cloud. All rights reserved.



You can customise the check update animation by overriding the updateLoadingView method of AbsMiniUiProxy.

The API description is as follows:

Note：

The return value of updateLoadingView method is an instance of IMiniLoading type.

IMiniLoading method description:

create Creates a Loading view;

show The callback to show the Loading effect;

stop Callback to stop the Loading effect.

```
/**
 * Custom mini program to check and update the loading page.
 * Call environment: main process
 * @param context
 * @param context
 * @return
 */
public abstract IMiniLoading updateLoadingView(Context context);
```

Sample code:

```
@Override
public IMiniLoading updateLoadingView(Context context) {
    return new IMiniLoading() {
        @Override
        public View create() {
            return LayoutInflater.from(context).inflate(R.layout.applet_activity_cu
        }
```

©2013-2022 Tencent Cloud. All rights reserved.

```
        @Override
        public void show(View v) {

        }

        @Override
        public void stop(View v) {

        }
    };
}
```

2. Custom Mini Program Loading View

The customisation of the update animation can be checked by overriding the startLoadingView method of AbsMiniUiProxy.

The API description is as follows:

Note：

The return value of the updateLoadingView method is an instance of type IMiniLoading.

IMiniLoading method description:

create Creates a Loading view;

show Callback to show the Loading effect;

stop Callback to stop the Loading effec

```
/**
 * Custom mini program loading loading page
 * Calling environment: child process
 *
 * @param activityWeakRef Activity reference
 * @param app mini program information
 * @return return mini program loading UI
 */
public abstract IMiniLoading startLoadingView(WeakReference<Activity> activityWeakR
```

Sample code:

```
@Override
public IMiniLoading startLoadingView(Context context) {
    return new IMiniLoading() {
```

©2013-2022 Tencent Cloud. All rights reserved.

```
        @Override
        public View create() {
            return LayoutInflater.from(context).inflate(R.layout.applet_activity_cu
        }

        @Override
        public void show(View v) {

        }

        @Override
        public void stop(View v) {

        }
    };
}
```

Hide Capsule Button on Loading Page

By default, there will be a capsule button on the top right corner of the mini program loading page, you can control the hide and show of the capsule button by overriding the hideLoadingCapsule method of AbsMiniUiProxy.
Capsule button schematic:：

©2013-2022 Tencent Cloud. All rights reserved.

API description: true means hide the capsule button on the Loading page, false means don't hide it (default value).

```
/**
 * Whether to hide the capsule in the upper right corner of the loading page of the
 /* @return true:Hide the capsule on the top right corner of the loading page.
 * @return true:hide;false:don't hide (default value)
 */
boolean hideLoadingCapsule();
```

©2013-2022 Tencent Cloud. All rights reserved.

Custom mini program SDK APIs

Last updated：2024-09-11 16:10:27

To enhance usage scenarios, the mini program SDK exposes certain APIs that can be customized by the host app. This allows the host app to provide unique capabilities to the mini program.

Prerequisite

To customize the mini program SDK APIs, you need to define a custom MiniAppProxy as follows:

Note：

Define an implementation class that extends BaseMiniAppProxy and annotate it with @ProxyService(proxy = MiniAppProxy.class).

```
@ProxyService(proxy = MiniAppProxy.class)
public class MiniAppProxyImpl extends BaseMiniAppProxy{}
```

Once the prerequisites are completed, you can customize the mini program SDK APIs. The SDK mainly supports the following types of APIs:

1. Customizing capsule button feature

1.1 Customizing close button event listener

Customize the close button event listener for the capsule button, allowing the host app to receive callback events when the close button is clicked.

Close button figure:

API description: The return value of onCapsuleButtonCloseClick method indicates whether to customise the close button listener or not, a return value of true means customise it, false (default value) means use the default listener.

```
/**
 * Close option for clicking capsule button
```

©2013-2022 Tencent Cloud. All rights reserved.

```
     * Calling environment: child process
     *
     * @param miniAppContext The environment in which the mini programme runs (the mini
     * @param onCloseClickedListener Callback when clicking the mini program to close i
     * @return does not support this interface, please return false
     */
    public abstract boolean onCapsuleButtonCloseClick(IMiniAppContext miniAppContext,
            DialogInterface.OnClickListener onCloseClickedListener);
```

Sample code:

```
    @Override
    public boolean onCapsuleButtonCloseClick(IMiniAppContext miniAppContext,
                                             DialogInterface.OnClickListener onClos
        Log.e("TAG","onCapsuleButtonCloseClick"+miniAppContext.getMiniAppInfo());
        //handle close mini app event
        return true;
    }
```

1.2 Customising the More Buttons event listener

Customising the event listener of the More button allows the host application to listen to the callback event of the corresponding item when the More button is clicked.

Diagram of the More button:

API description: The getMoreItemSelectedListener method returns a value that indicates more button listeners.

```
/**
 * Return to Capsule More panel button click listener
 * Calling environment: child process
```

©2013-2022 Tencent Cloud. All rights reserved.

```
 *
 * @return listener
 */
public abstract OnMoreItemSelectedListener getMoreItemSelectedListener();
```

Sample code:

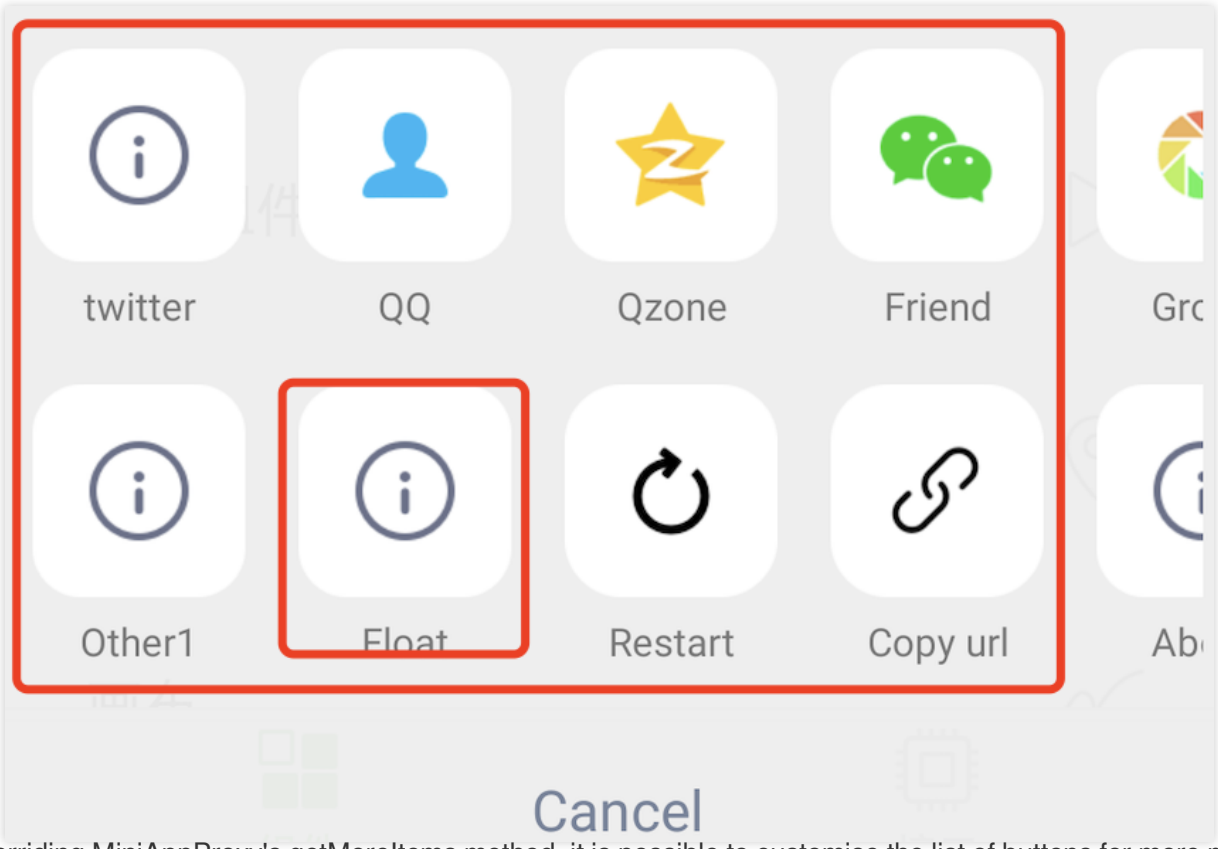
```
/**
 * Return to Capsule More panel button click listener
 * @return
 * @return
 */
@Override
public OnMoreItemSelectedListener getMoreItemSelectedListener() {
    return new DemoMoreItemSelectedListener();
}

/**
define of DemoMoreItemSelectedListener
**/
public class DemoMoreItemSelectedListener extends DefaultMoreItemSelectedListener {
    public static final int CLOSE_MINI_APP = 150;

    @Override
    public void onMoreItemSelected(IMiniAppContext miniAppContext, int moreItemId)

        switch (moreItemId) {
            case CLOSE_MINI_APP:
                close(miniAppContext);
                return;
            case OTHER_MORE_ITEM_1:
                miniAppContext.getAttachedActivity().runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        Toast.makeText(miniAppContext.getAttachedActivity(), "custo
                    }
                });
                return;
        }
        super.onMoreItemSelected(miniAppContext, moreItemId);
    }

    public void close(IMiniAppContext miniAppContext) {
        Activity activity = miniAppContext.getAttachedActivity();
        if (activity != null && !activity.isFinishing()) {
            boolean moved = activity.moveTaskToBack(true);
            if (!moved) {
```

©2013-2022 Tencent Cloud. All rights reserved.

```
                QMLog.e("Demo", "moveTaskToBack failed, finish the activity.");
                activity.finish();
            }
        }
    }
}
```

1.3 Customising the More Buttons Display List

By the time the user triggers the more buttons click event, the following list of optional extended buttons will pop up, the default list is shown below:



By overriding MiniAppProxy's getMoreItems method, it is possible to customise the list of buttons for more menu extensions.

API Description:

The return value of getMoreItems method represents the customised list of extension buttons;

The method parameter MoreItemList.Builder is used to add extension buttons, the display order of extension buttons is the same as the add order;

The id attribute of MoreItem is used to distinguish the buttons and needs to be unique.

```
/**
 * The button that returns the capsule to the More panel, the ID of the extended bu
 * Call environment: child process
 *
```

©2013-2022 Tencent Cloud. All rights reserved.

```
 * @param builder
 * @return
 */
public abstract ArrayList<MoreItem> getMoreItems(MoreItemList.Builder builder);
```

Warning：

The ID of the extension button needs to be set to a value in the interval [100, 200], otherwise the add is invalid.

Sample code:

```
@Override
public ArrayList<MoreItem> getMoreItems(IMiniAppContext miniAppContext, MoreItemLis
    MoreItem item1 = new MoreItem();
    item1.id = ShareProxyImpl.OTHER_MORE_ITEM_1;
    item1.text = getString(miniAppContext, R.string.applet_mini_proxy_impl_other1);
    item1.drawable = R.mipmap.mini_demo_about;

    MoreItem item2 = new MoreItem();
    item2.id = ShareProxyImpl.OTHER_MORE_ITEM_2;
    item2.text = getString(miniAppContext, R.string.applet_mini_proxy_impl_other2);
    item2.shareKey = SHARE_TWITTER;//Custom sharing key, must be set and unique, wi
    item2.drawable = R.mipmap.mini_demo_about;

    MoreItem item3 = new MoreItem();
    item3.id = DemoMoreItemSelectedListener.CLOSE_MINI_APP;
    item3.text = getString(miniAppContext, R.string.applet_mini_proxy_impl_float_ap
    item3.drawable = R.mipmap.mini_demo_about;

    MoreItem item4 = new MoreItem();
    item4.id = ShareProxyImpl.OTHER_MORE_ITEM_INVALID;
    item4.text = getString(miniAppContext, R.string.applet_mini_proxy_impl_out_of_e
    item4.drawable = R.mipmap.mini_demo_about;

    // Self-adjusting order.
    builder.addMoreItem(item1)
            .addMoreItem(item2)
            .addShareQQ("QQ", R.mipmap.mini_demo_channel_qq)
            .addMoreItem(item3)
            .addShareQzone(getString(miniAppContext, R.string.applet_mini_proxy_imp
                    R.mipmap.mini_demo_channel_qzone)
            .addShareWxFriends(getString(miniAppContext, R.string.applet_mini_proxy
                    R.mipmap.mini_demo_channel_wx_friend)
            .addShareWxMoments(getString(miniAppContext, R.string.applet_mini_proxy
                    R.mipmap.mini_demo_channel_wx_moment)
            .addRestart(getString(miniAppContext, R.string.applet_mini_proxy_impl_r
                    R.mipmap.mini_demo_restart_miniapp)
            .addAbout(getString(miniAppContext, R.string.applet_mini_proxy_impl_abo
```

©2013-2022 Tencent Cloud. All rights reserved.

```
                    R.mipmap.mini_demo_about)
            .addDebug(getString(miniAppContext, R.string.mini_sdk_more_item_debug),
                    R.mipmap.mini_demo_about)
            .addMonitor(getString(miniAppContext, R.string.applet_mini_proxy_impl_p
                    R.mipmap.mini_demo_about)
            .addComplaint(getString(miniAppContext, R.string.applet_mini_proxy_impl
                    R.mipmap.mini_demo_browser_report)
            .addSetting(getString(miniAppContext, R.string.mini_sdk_more_item_setti
                    R.mipmap.mini_demo_setting);

    return builder.build();
}

private String getString(IMiniAppContext miniAppContext, int id) {
    return miniAppContext.getContext().getString(id);
}
```

2. Customised Image Selection

The mini program SDK provides default image selection, which is done by default using the album selector that comes with Android, and is triggered by calling multimedia selection wx.chooseImage in the mini program.
Custom image selection can be achieved by overriding the openChoosePhotoActivity method of BaseMiniAppProxy.
API description.
Parameter Context: the current activity instance of the mini program;
Parameter maxSelectedNum: maximum number of selected pictures;
Parameter IChoosePhotoListner: callback interface for photo selection;
Return value boolean: a return value of false means using the default image selection implementation, a return value of true means using the customised image selection implementation.

```
/**
 * Open the selection screen
 *
 * @param context Current Activity
 * @param maxSelectedNum Maximum number of allowable selections
 * @param listner Callback interface
 * @return Do not support this interface, please return false
 */
@Override
public boolean openChoosePhotoActivity(Context context, int maxSelectedNum, IChoose
```

Note：

©2013-2022 Tencent Cloud. All rights reserved.

When the image path is returned by the onResult method of IChoosePhotoListner, the image path should be an absolute path.

Sample code:

```
@Override
public boolean openChoosePhotoActivity(Context context, int maxSelectedNum, IChoose
    Log.d("TAG","open choose photo activity ");
    Intent intent = new Intent(context, ChoosePhotosActivity.class);
    intent.putExtra("maxCount",maxSelectedNum);
    //not recommand to use static refs,just for example
    ChoosePhotosActivity.setChooseCallBack(listner);
    context.startActivity(intent);

    return true;
}


public class ChoosePhotosActivity extends Activity {
   static WeakReference<MiniAppProxy.IChoosePhotoListner>iChoosePhotoListnerWeakRef
    public static void setChooseCallBack(MiniAppProxy.IChoosePhotoListner choosePho
        iChoosePhotoListnerWeakReference = new WeakReference<>(choosePhotoListner);
    }

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //todo show your choose view
    }

    @Override
    protected void onResume() {
        super.onResume();
        startChooseImage();
    }

    private void startChooseImage(){
        //todo finish choose logic

        //response choose callback
        ArrayList<String>path = new ArrayList<>();
        //todo replace with real path of image
        path.add("picture local absolute path");
        path.add("picture local absolute path2");
        path.add("/data/user/0/com.tencent.tmf.miniapp.demo/files/userfiles/Screens
        iChoosePhotoListnerWeakReference.get().onResult(path);
```

©2013-2022 Tencent Cloud. All rights reserved.

```
        }
    }
```

3. Customising image preview

The mini program SDK provides a default image preview implementation, which is triggered when the mini program calls the multimedia selection wx.previewImage.
Custom image preview can be achieved by overriding the openImagePreview method of BaseMiniAppProxy.
The default image preview page is shown below:

©2013-2022 Tencent Cloud. All rights reserved.

API description:

Parameter Context: the current activity instance of the mini program;

parameter selectedIndex: position of the selected image in the list;

parameter pathList: list of paths of the images to be displayed;

Return value boolean: a return value of false means use default image to display, a return value of true means use custom image to display.

```
/**
 * Open the picture preview interface
```

©2013-2022 Tencent Cloud. All rights reserved.

```
 * Calling environment: child process
 *
 * @param context Current Activity
 * @param selectedIndex index of the currently selected image
 * @param pathList picture path list
 * @return Do not support this interface, please return false
 */
public abstract boolean openImagePreview(Context context, int selectedIndex, List<S
```

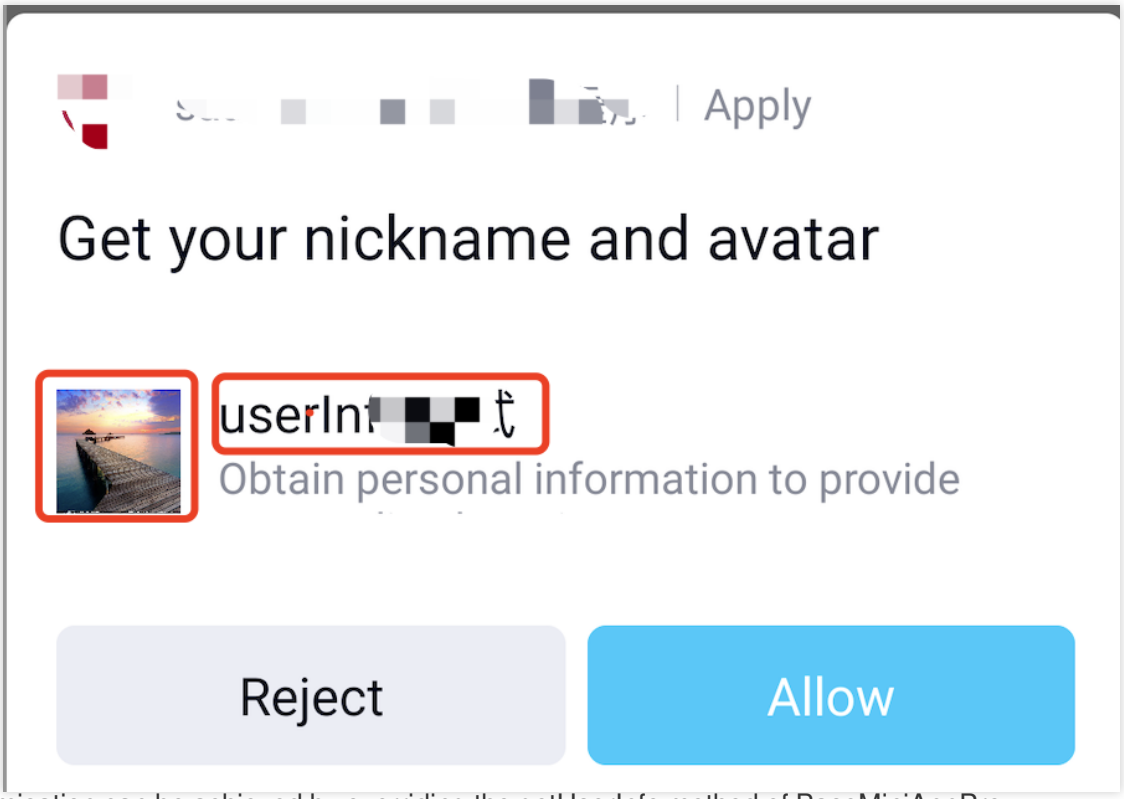
Sample code:

```
/**
 * Open the picture preview interface
 *
 * @param context Current Activity
 * @param selectedIndex index of the currently selected image
 * @param pathList picture path list
 * @return Do not support this interface, please return false
 */
@Override
public boolean openImagePreview(Context context, int selectedIndex, List<String> pa
    //todo start your image preview
    Intent intent = new Intent(context, CustomPreviewActivity.class);
    intent.putExtra("curIndex",selectedIndex);
    intent.putStringArrayListExtra("pathList", pathList);
    context.startActivity(intent);
    return true;
}
```

4. Customising the user information in the authorisation pop-up

When a mini program applies for user information, it will pop up the user authorisation pop-up window; the host application can customize part of the information in the user authorisation pop-up window, including: avatar image and user nickname, as shown in the following figure:

©2013-2022 Tencent Cloud. All rights reserved.



Customisation can be achieved by overriding the getUserInfo method of BaseMiniAppProxy.

API description:

Parameter appId: appId of the mini program currently requesting authorisation;

Parameter AsyncResult: used to return user information to the mini program SDK.

```
/**
 * Get scope.userInfo authorised user information
 * Call environment: subprocess
 *
 * @param appId
 * @param result
 */
@Override
public void getUserInfo(String appId, AsyncResult result)
```

Sample code:

```
/**
 * Get scope.userInfo authorised user information
 * Call environment: subprocess
 *
 * @param appId
 * @param result
 */
@Override
public void getUserInfo(String appId, AsyncResult result) {
```

©2013-2022 Tencent Cloud. All rights reserved.

```
    JSONObject jsonObject = new JSONObject();
    try {
        //return nickname
        jsonObject.put("nickName", "userInfo Test");;
        //return the avatar url
        jsonObject.put("avatarUrl", "https://gimg2.baidu.com/image_search/src=http%
        result.onReceiveResult(true, jsonObject);
    } catch (JSONException e) {
        e.printStackTrace();
    }
}
```

5. Custom Image Loader

Warning:

Since the mini program SDK does not have a default image loader implementation, the loading of images relies on the host application's implementation; if the host application does not implement an icon loader, this will result in abnormal image loading on some pages.

Host app developers, need to implement image loading customisation by overriding the getDrawable method of BaseMiniAppProxy.

API description:

Parameter context: the context of the mini program process currently requesting authorisation;

parameter source: the source of the image, it can be a local or network image;

parameter width: width of the image;

parameter height: height of the image;

parameter defaultDrawable: the default drawable, used in loading and loading failure;

Return value Drawable: the loaded drawable object.

```
@Override
public Drawable getDrawable(Context context, String source, int width, int hight, D
```

Sample code:

```
@Override
public Drawable getDrawable(Context context, String source, int width, int hight, D
    //Access to access their own ImageLoader
    //sample code to use the open source universalimageloader
    UniversalDrawable drawable = new UniversalDrawable();
    if (TextUtils.isEmpty(source)) {
        return drawable;
    }
    drawable.loadImage(context, source);
```

©2013-2022 Tencent Cloud. All rights reserved.

```
    return drawable;
}
```

6. Custom mini program data storage segregated by account

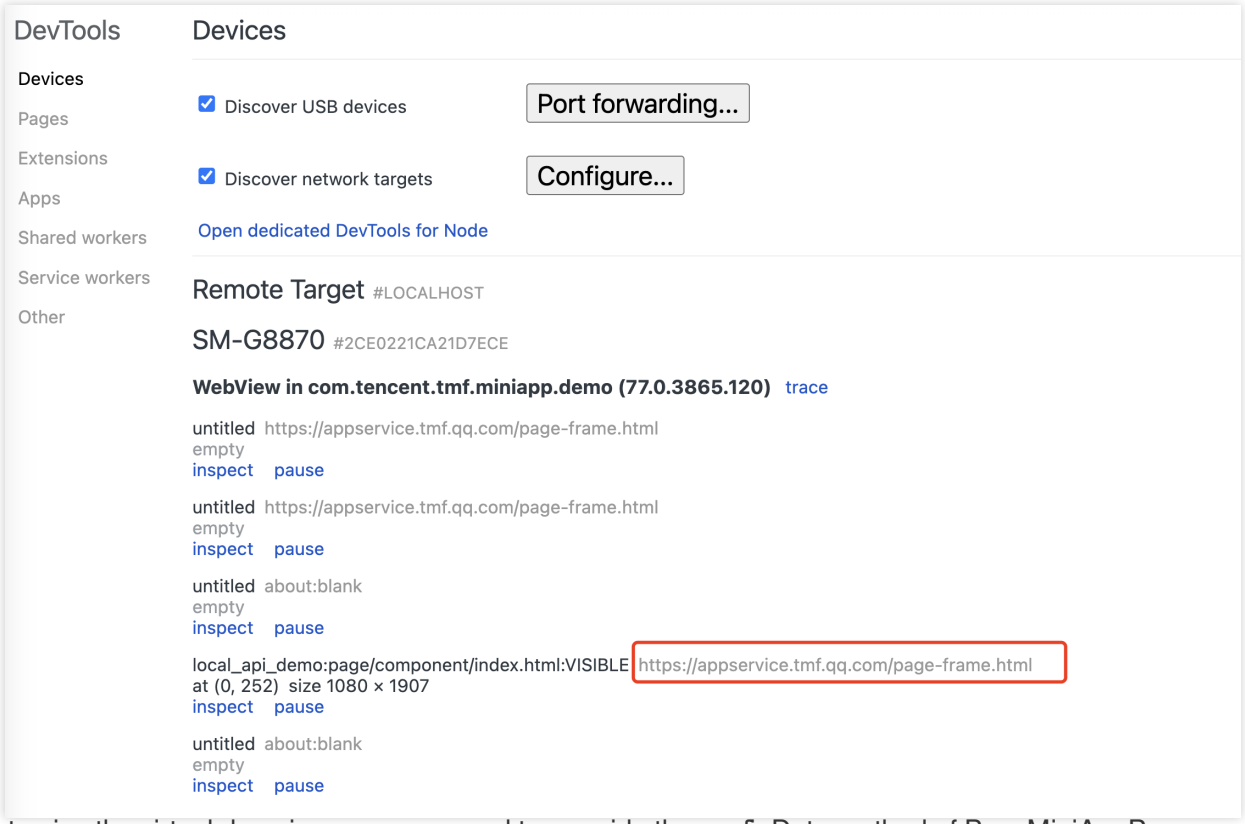
Override the getAccount method of BaseMiniAppProxy to achieve isolated storage of mini program data.
API description: The return value is the user account (must be unique), the data will be stored in isolation according to the account after setting, it is not recommended to use user sensitive information.

```
/**
 * user account, must be unique, after setting the data will be stored according to
 * Call environment: the main process
 */
@Override
public String getAccount() {
    return "tmf_test";
}
```

7. Customised Virtual Domains

Mini program to play to the default virtual domain name, this domain name is not a real domain name, in the browser is not accessible, if you need to customise can be set up in accordance with the following configuration.
The default virtual domain name is shown below:

©2013-2022 Tencent Cloud. All rights reserved.



To customise the virtual domain name, you need to override the configData method of BaseMiniAppProxy and intercept the configType to MiniConfigData.TYPE_DOMAIN to achieve customisation. Mini program to play to the default virtual domain name, this domain name is not a real domain name, in the browser is not accessible, if you need to customise can be set up in accordance with the following configuration.

The default virtual domain name is shown below:

Sample code:

```
@Override
public MiniConfigData configData(Context context, int configType, JSONObject params
    if(configType == MiniConfigData.TYPE_DOMAIN) {
        //Virtual Domain Configuration
        MiniConfigData.DomainConfig domainConfig = new MiniConfigData.DomainConfig(
        domainConfig.domain = "test.com";

        return new MiniConfigData
                .Builder()
                .domainConfig(domainConfig)
                .build();
    }

    return new MiniConfigData
                .Builder()
                .build();
}
```

©2013-2022 Tencent Cloud. All rights reserved.

8. Customise the userAgent of the WebView in the mini program.

Customisation can be achieved by overriding the configData method of BaseMiniAppProxy and intercepting the configType as MiniConfigData.TYPE_WEBVIEW.

Sample code:

```
@Override
public MiniConfigData configData(Context context, int configType, JSONObject params
    if(configType == MiniConfigData.TYPE_WEBVIEW) {
        //webView userAgent
        String ua = params.optString(MiniConfigData.WebViewConfig.WEBVIEW_CONFIG_UA
        MiniConfigData.WebViewConfig webViewConfig = new MiniConfigData.WebViewConf
        //Set the userAgent that the developer needs to append, note: don't set the
        webViewConfig.userAgent = "key/value";

        return new MiniConfigData
                .Builder()
                .webViewConfig(webViewConfig)
                .build();
    }

    return new MiniConfigData
            .Builder()
            .build();
}
```

9. Customised scanning capabilities

The mini program SDK provides a default scanning implementation (refer to Extension Library Support - Sweep), and also provides an interface for users to customise the code scanning ability.

To customise the scanning ability, you need to override the enterQRCode method of BaseMiniAppProxy.

API description:

Parameter context: the current context of the mini program process that initiates the code scanning;

parameter onlyFromCamera: if or not only camera is allowed to sweep code;

parameter AsyncResult: the callback of code scanning result;

Return value boolean: true means custom code scanning ability, false means use built-in code scanning ability (need to integrate code scanning extension library).

```
/**
 * Scanning QR code
 *
```

©2013-2022 Tencent Cloud. All rights reserved.

```
 * @param context context
 * @param onlyFromCamera Only allow camera to scan code
 * @param result Scanning result
 * @return true:custom scanning;false:use built-in scanning
 */
@Override
public boolean enterQRCode(Context context, boolean onlyFromCamera, AsyncResult res
```

10. Specify the storage path where the mini program saves the file

By overriding the getSaveFileDir method of BaseMiniAppProxy, we can implement the mini program API to save pictures and videos to the specified directory in the system album.

Sample code:

```
/**
 * Specify the directory where the mini program API saves images and videos to the
 *
 */
public  String getSaveFileDir(){
return "myapp";
}
```

Note：

The default path is tcmpp.

11. Listening to the mini program lifecycle

You can listen to the mini program lifecycle by overriding the onAppStateChange method of BaseMiniAppProxy.

API description:

Parameter appState: life cycle state of the current mini program, refer to AppState;

parameter MiniAppEvent: life cycle event of the current mini program.

```
/**
 * Mini-program lifecycle event callbacks
 * Call context environment: main process UI threads
 *
 * @param appState event state
 * @param event event
 */
public abstract void onAppStateChange(@AppState int appState, MiniAppEvent event);
```

©2013-2022 Tencent Cloud. All rights reserved.

12. Customising the base library update policy and listening

You can override BaseMiniAppProxy's isUpdateBaseLib method to listen for updates to the mini program's base library.

API description:

You can override BaseMiniAppProxy's isUpdateBaseLib method to listen for updates to the mini program's base library.

```
/**
 * Whether or not to update when the base library detects an update
 * @param context
 * @param data Base library information data
 * @return true:update;false:don't update, default is true
 */
public abstract boolean isUpdateBaseLib(Context context, JSONObject data);
```

©2013-2022 Tencent Cloud. All rights reserved.

Custom sharing capabilities

Last updated：2024-07-29 16:52:20

The mini program SDK provides an API for sharing feature. Users can customize the sharing capability as follows:

1. Add a custom share item to the capsule control panel via MiniAppProxyImpl:

```
private static final String SHARE_TWITTER = "twitter";
/**
 * Returns a map of custom share data.
 * Call environment: sub-process
 *
 * key: Matches the MoreItem.id added in the getMoreItems method.
 * value: Matches the MoreItem.shareKey added in the getMoreItems method.
 * @
 */
@Override
public Map<String, Integer> getCustomShare(){
   Map<String, Integer> objects = new HashMap<>();
   objects.put(SHARE_TWITTER, ShareProxyImpl.OTHER_MORE_ITEM_2);
   return objects;
}

/**
 * Go back to the **More panels** button of the capsule. Set the ID of the extensio
 * Call environment: sub-process
 *
 * @param miniAppContext Mini program runtime environment (mini program process but
 * @param builder
 * @
 */
@Override
public ArrayList<MoreItem> getMoreItems(IMiniAppContext miniAppContext, MoreItemLis
 MoreItem item2 = new MoreItem();
 item2.id = ShareProxyImpl.OTHER_MORE_ITEM_2;
 item2.text = getString(miniAppContext,
                   R.string.applet_mini_proxy_impl_other2);
 item2.shareKey = SHARE_TWITTER;//Custom share key, must be set and unique, used wh
 item2.drawable = R.mipmap.mini_demo_about;

 builder.addMoreItem(item2)
 return builder.build();
}
```

2. TCMPP SDK has built-in sharing buttons for QQ, Qzone, WeChat, and WeChat Circle of Friends, which can be quickly configured by overriding getDefaultShare.

©2013-2022 Tencent Cloud. All rights reserved.

```
 /**
 * Returns a list of built-in share buttons
 * Calling environment: child process
 *
 * @return Enumeration list of built-in share buttons
 */
@Override
public List<String> getDefaultShare() {
    ArrayList<String> defaultShare = new ArrayList<>();
    defaultShare.add(MoreItemList.SHARE_QQ);
    defaultShare.add(MoreItemList.SHARE_QZONE);
    defaultShare.add(MoreItemList.SHARE_WX_FRIENDS);
    defaultShare.add(MoreItemList.SHARE_WX_MOMENTS);
    return defaultShare;
}
```

3. To create a capsule, in theMore panel, clickListeners.

```
 /**
 * Return to Capsule More Panel Button Click Listener
 *
 * Return
 */
Override
public OnMoreItemSelectedListener getMoreItemSelectedListener() {
    return new DemoMoreItemSelectedListener(); }
}


public class DemoMoreItemSelectedListener extends DefaultMoreItemSelectedListener {
    public static final int CLOSE_MINI_APP = 150; @Override


    @Override
    public void onMoreItemSelected(IMiniAppContext miniAppContext, int moreItemId)
        //Handle developer-defined click events (except for custom sharing events)
        switch (moreItemId) {
            case CLOSE_MINI_APP: close(mini_APP)
                close(miniAppContext); return
                Returns;
            case OTHER_MORE_ITEM_1:
                miniAppContext.getAttachedActivity().runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        Toast.makeText(miniAppContext.getAttachedActivity(), "Custo
                    }
```

©2013-2022 Tencent Cloud. All rights reserved.

```
                });
                Return; }
        }


        // Handle built-in sharing and developer custom sharing, e.g. microblogging
        super.onMoreItemSelected(miniAppContext, moreItemId)}; return; }// Handle b
    }
}
```

4. Receive sharing according to the following types, click the event, the developer can get the sharing data in the share method, and call the third-party SDK to implement the sharing.

```
@ProxyService(proxy = ShareProxy.class)
public class ShareProxyImpl extends BaseShareProxy {
  /**
   * Share
   *
   * @param shareData shareData
   */
  @Override
  public void share(Activity activity, ShareData shareData) {
  //todo share
  }
}
```

©2013-2022 Tencent Cloud. All rights reserved.

Custom authorization list

Last updated：2024-06-27 10:53:15

To get the authorization list, use the TmfMiniSDK as shown below:

```
/**
 * Get the mini program authorization list
 * @param appId
 * @param appVerType Mini program version type
 * @
 */
public static List<MiniAuthState> getAuthStateList(String appId, int appVerType)

/**
 * Set the authorization status
 * @param appId
 * @param appVerType Mini program version type
 * @param scopeName Permission name
 * @param grant Whether to authorize
 */
public static void setAuthState(String appId, int appVerType, String scopeName, boo
```

©2013-2022 Tencent Cloud. All rights reserved.

Custom permission requests

Last updated：2024-06-27 10:52:40

Mini program requests system permissions

When using mini programs, some APIs require not only mini program authorization but also corresponding Android system permissions to function correctly. By default, the SDK will automatically prompt the user to grant these system permissions when needed. However, you can disable this automatic permission request during SDK initialization. In this case, the host app must handle these system permission requests to ensure the mini program APIs work properly.

```
@ProxyService(proxy = MiniConfigProxy.class)
public class MiniConfigProxyImpl extends MiniConfigProxy {
    /**
     * App Application
     * @
     */
    @Override
    public Application getApp() {
        return "app Application";
     }

      /**
     * Create initialization configuration information
     * @
     */
    @Override
    public MiniInitConfig buildConfig() {
      MiniInitConfig.Builder builder = new MiniInitConfig.Builder();
      MiniInitConfig config = builder
                .configAssetName("tcmpp-android-configurations.json") //Configuratio
                 .imei("IMEI") //(Optional) Device ID, which is used for grayscale r
                 .autoRequestPermission(false) //onfigure whether to automatically r
                 .debug(true) //Log switch, turned off by default.
                 .build();
     }
}
```

Customize system permission requests

By implementing the IPermissionManagerProxy API, you can intercept and customize the logic for mini program system permission requests. The IPermissionManagerProxy API includes three methods: isPermissionGranted to check if the host has a specific system permission, and requestForPermission and requestForPermissions to notify the host to request one or multiple permissions.

©2013-2022 Tencent Cloud. All rights reserved.

Note：

Once you customize system permission requests, the autoRequestPermission setting in the initialization configuration will no longer be effective.

```
@ProxyService(proxy = IPermissionManagerProxy.class)
public class PermissionProxyImpl implements IPermissionManagerProxy {

    /**
     * Check if the host has a specific system permission.
     * @param context   Android context
     * @param permission    The system permission to check, refer to android.Manifes
     * @return              Whether the host has permission
     */
    @Override
    public boolean isPermissionGranted(Context context, String permission) {
        return ContextCompat.checkSelfPermission(context, permission) == PackageMana
    }

    /**
     * Notify the host to request a system permission.
     * @param activity      The mini program Activity requesting the permission
     * @param permission     The system permission to request, refer to android.Mani
     * @param callbacks      Callback to return the permission request result to the
     */
    @Override
    public void requestForPermission(Activity activity, String permission, RequestPe
        Toast.makeText(activity,"applying for" + permission + "permission", Toast.LE
    }

    /**
     * Notify the host to request multiple system permissions.
     * @param activity      The mini program Activity requesting the permissions
     * @param permissions    The list of system permissions to request, refer to and
     * @param callbacks     Callback to return the permission request result to the
     */
    @Override
    public void requestForPermissions(Activity activity, String[] permissions, Reque
        Toast.makeText(activity,"applying for" + permissions[0] + "permissions," Toa
    }
}
```

```
/**
 * Notify the result of system permission requests.
 */
interface RequestPermissionCallback {
    /**
```

©2013-2022 Tencent Cloud. All rights reserved.

```
     All system permissions are successfully granted.
     */
    void onSuccess();

    /**
     * Some or all system permissions are denied.
     * @param rejected      List of denied system permissions
     */
    void onFail(String[] rejected);
}
```

©2013-2022 Tencent Cloud. All rights reserved.

Custom User Attributes

Last updated：2024-06-27 10:52:29

Host app developers can set user attributes (such as region or account information) using methods provided by the SDK. This is useful for scenarios like targeted push notifications.

Setting user ID

API description: The userId parameter is used to set account information.

```
/**
 * Set the account information
 * @param userId
 */
 public static void setUserId(String userId)
```

Set location information

API description:

Parameter country indicates the user's country.

Parameter province indicates the user's province.

Parameter city indicates the user's city.

```
/**
 * Set the location information
 * @param country
 * @param province
 * @param city
 */
 public static void setLocation(String country, String province, String city)
```

©2013-2022 Tencent Cloud. All rights reserved.

Logging and event reporting

Last updated：2024-06-27 10:52:17

Custom log output implementation

By implementing the LogProxy API, you can control the internal logging of the mini program SDK.

Sample code:

```
@ProxyService(proxy = LogProxy.class)
public class LogProxyImpl extends LogProxy {

    @Override
    public void log(int logLevel, String tag, String msg, Throwable t) {
        switch (logLevel) {
            case Log.DEBUG:
                if (t == null) {
                    android.util.Log.d(tag, msg);
                } else {
                    android.util.Log.d(tag, msg, t);
                }
                break;
            case Log.INFO:
                if (t == null) {
                    android.util.Log.i(tag, msg);
                } else {
                    android.util.Log.i(tag, msg, t);
                }
                break;
            case Log.WARN:
                if (t == null) {
                    android.util.Log.w(tag, msg);
                } else {
                    android.util.Log.w(tag, msg, t);
                }
                break;
            case Log.ERROR:
                if (t == null) {
                    android.util.Log.e(tag, msg);
                } else {
                    android.util.Log.e(tag, msg, t);
                }
                break;
            default:
```

©2013-2022 Tencent Cloud. All rights reserved.

```
                if (t == null) {
                    android.util.Log.v(tag, msg);
                } else {
                    android.util.Log.v(tag, msg, t);
                }
                break;
            }
        }

        @Override
        public boolean isColorLevel() {
            return true;
        }
    }
```

Custom Event Reporting

The host APP can override the internal reporting logic of TCMPP SDK by implementing the reportMiniAppEvent method of MiniAppProxy to customize the mini program event reporting.

Note：

Includes mini program operation events and the data reported by calling wx.reportEvent inside the mini program.

API is as follows:

```
/**
 * Agent for reporting mini program events.
 * @param eId eventId
 * @param eventName event name
 * @param props event property
 * @param app Mini program information.
 * @return is true, the SDK internal reporting logic is no longer executed, and the
 */
public abstract boolean reportMiniAppEvent(int eId, String eventName, JSONObject pr
```

The built-in event IDs are described below:

```
/** Custom event */
public static final int EID_None = 0;

/** Open a mini program */.
public static final int EID_OPEN_MINIAPP = 1;

/** Update mini program */ public static final int EID_OPEN_MINIAPP = 1.
public static final int EID_UPDATE_MINIAPP = 2;
```

©2013-2022 Tencent Cloud. All rights reserved.

```
/** Download mini program */
public static final int EID_DOWNLOAD_MINIAPP = 3;

/** Mini-program page view */
public static final int EID_MINIAPP_PAGE_VIEW = 4;

/** Exit mini program */ public static final int EID_MINIAPP_PAGE_VIEW = 4.
public static final int EID_EXIT_MINIAPP = 5;

/** Behavioural event of mini program */
public static final int EID_MINIAPP_ACTION = 6;
```

Customised Log Reporting

Real Time Log Reporting

The host APP can **customize the mini program real-time event log reporting logic by implementing the reportRealTimeLog method of MiniAppProxy.**

Note：

Include the log data written by the mini program internal call wx.getRealtimeLogManager.

API is as follows:

```
/**
 * Mini program real-time log upload agent
 * @param page Current page
 * @param jsLibVersion base library version
 * @param app mini program information
 * @param filterMsgs filter keywords
 * @param logs logs
 * @return is true, the SDK internal reporting logic is no longer executed.
 */
public abstract boolean reportRealTimeLog(String page, String jsLibVersion, MiniApp
                                          ArrayList<RealTimeLogItem> logs);
```

Internal log reporting for mini programs

The host APP can **customise the complaint feedback page of the mini program to collect the log upload logic by implementing the uploadUserLog method of MiniAppProxy.**

Note：

This includes log data written by the mini program's internal call to wx.getLogManager, and the user can upload printed logs by using the button component's open-type="feedback".

```
/**
```

©2013-2022 Tencent Cloud. All rights reserved.

```
     * feedback log upload
     *
     * @param appId mini program appId
     * @param logPath The log path.
     * @return is true, the SDK internal upload logic is not executed.
     */
    public abstract boolean uploadUserLog(String appId, String logPath);
```

©2013-2022 Tencent Cloud. All rights reserved.

Open APIs

Last updated：2024-06-27 10:52:01

The following table shows the mapping between Weixin mini program methods and native events. When developers call these methods in the mini program, the corresponding native events are triggered, and developers need to listen to these events and return data.

Mini Program Methods	MiniOpenApiProxy Methods	Description of the method
wx.login	login	Login interface
wx.getUserInfo	getUserInfo	Get basic user information
wx.getUserProfile	getUserProfile	Get user profile information
wx.getPhoneNumber	getPhoneNumber	Get phone number
wx.requestPayment	requestPayment	Initiate a payment
wx.checkSession	checkSession	Checks if the login status has expired.

Users can implement MiniOpenApiProxy proxy to correlate the data interactions between mini programs and the host application, the sample code is as follows:

Note：

Parameter IMiniAppContext is the context information of the current mini program;

The parameter JSONObject is the parameter passed when the mini program calls the open interface;

The parameter AsyncResult is the asynchronous callback interface through which the user needs to return the return value of the host application's implementation of the open interface to the mini program.

```
@ProxyService(proxy = MiniOpenApiProxy.class)
public class MiniOpenApiProxyImpl implements MiniOpenApiProxy {
    private static final String TAG = "MiniOpenApiProxyImpl";
    @Override
    public void login(IMiniAppContext miniAppContext, JSONObject params, AsyncResul
        QMLog.d(TAG, "login:" + params);
        JSONObject jsonObject = new JSONObject();
        try {
            jsonObject.put("key", "wx.login");
        } catch (JSONException e) {
            e.printStackTrace();
        }
        result.onReceiveResult(true, jsonObject);
    }
```

©2013-2022 Tencent Cloud. All rights reserved.

```
    @Override
    public void checkSession(IMiniAppContext miniAppContext, JSONObject params, Asy
        QMLog.d(TAG, "checkSession:" + params);
        JSONObject jsonObject = new JSONObject();
        try {
            jsonObject.put("key", "wx.checkSession");
        } catch (JSONException e) {
            e.printStackTrace();
        }
        result.onReceiveResult(true, jsonObject);
    }

    @Override
    public void getUserInfo(IMiniAppContext miniAppContext, JSONObject params, Asyn
        QMLog.d(TAG, "getUserInfo:" + params);
        JSONObject jsonObject = new JSONObject();
        try {
            final JSONObject userInfo = new JSONObject();

            userInfo.put("nickName", "userInfotest");
//            userInfo.put("avatarUrl", bundle.getString("avatarUrl"));

            userInfo.put("gender", 0);
            userInfo.put("country", "CN");
            userInfo.put("province", "BeiJing");
            userInfo.put("city", "BeiJing");
            userInfo.put("language", "en");
            jsonObject.put("userInfo", userInfo);
        } catch (JSONException e) {
            e.printStackTrace();
        }
        result.onReceiveResult(true, jsonObject);
    }

    @Override
    public void getUserProfile(IMiniAppContext miniAppContext, JSONObject params, A
        QMLog.d(TAG, "getUserProfile:" + params);
        JSONObject jsonObject = new JSONObject();
        try {
            jsonObject.put("key", "wx.getUserProfile");
        } catch (JSONException e) {
            e.printStackTrace();
        }
        result.onReceiveResult(true, jsonObject);
    }
```

©2013-2022 Tencent Cloud. All rights reserved.

```
    @Override
    public void getPhoneNumber(IMiniAppContext miniAppContext, JSONObject params, A
        QMLog.d(TAG, "getPhoneNumber:" + params);
        JSONObject jsonObject = new JSONObject();
        try {
            jsonObject.put("key", "wx.getPhoneNumber");
        } catch (JSONException e) {
            e.printStackTrace();
        }
        result.onReceiveResult(true, jsonObject);
    }

    @Override
    public void requestPayment(IMiniAppContext miniAppContext, JSONObject params, A
        QMLog.d(TAG, "requestPayment:" + params);
        JSONObject jsonObject = new JSONObject();
        try {
            jsonObject.put("key", "wx.requestPayment");
        } catch (JSONException e) {
            e.printStackTrace();
        }
        result.onReceiveResult(true, jsonObject);
    }
```

©2013-2022 Tencent Cloud. All rights reserved.

Others

Last updated：2024-09-10 18:48:14

Process special URLs in web-view component

If the webpage displayed in the web-view component contains special URLs, such as those starting with a custom scheme like tcmpp://host/path, the host app can intercept and customize the redirections of the URLs.
To intercept URL navigation, create a class that implements the IWebViewLinkProxy API and add the ProxyService annotation:

```
@ProxyService(proxy = IWebViewLinkProxy.class)
public class CustomWebViewLinkProxy implements IWebViewLinkProxy {

    /**
     * Handle custom url loading in web-view component.
     * Custom url is url with any custom scheme (scheme not network protocol such a
     * @param miniContext context of mini-program
     * @param url the url which is loading
     * @return return true if the url loading is handled, otherwise false the web-v
     */
    @Override
    public boolean webViewCustomUrlLoading(IMiniAppContext miniContext, String url)
        Uri uri = Uri.parse(url);
        if ("tcmpp".equals(uri.getScheme())) {
            Intent intent = new Intent(Intent.ACTION_VIEW, uri);
            miniContext.getAttachedActivity().startActivity(intent);
            return true;
        }
        return false;
    }
}
```

In the implementation, the mini program context and the loaded URL are provided. If the developer handles the URL redirections and returns true, the web-view component will no longer process the URL. If false is returned, the web-view component will process the URL according to normal logic.

©2013-2022 Tencent Cloud. All rights reserved.

API Description

Last updated：2024-07-29 16:28:12

MiniStartOptions

Parameter configuration for mini program launching

```
Parameter configuration for mini program launching
/**
 * Whether to force update when opening the mini program (effective only the first
 */
public boolean isForceUpdate = false;
/**
 * Entry path, supports adding parameters: path?key=value&key1=value1
 */
public String entryPath;
/**
 * Receive mini program startup errors
 */
public ResultReceiver resultReceiver;

/**
 // Parameters for starting the mini program
 */
public String params;

/**
 * Set the task mode for opening the mini program.
 *
 * false:  Multi-task mode, true:  Single task mode
 */
public boolean isSingleTask;
```

MiniCode

Error code description

```
/**
 * Successful
 */
public static final int CODE_OK = 0;
```

©2013-2022 Tencent Cloud. All rights reserved.

```
//////////////////////////////Server error////////////////////////////////////
public static final int C_SERVER = -11000;
/**
 * shark network error
 */
public static final int C_SERVER_SHARK_ERROR = -11001;
/**
 * The server returns the code error
 */
public static final int C_SERVER_RET_CODE_ERROR = -11002;
/**
 * The server returns the response is null
 */
public static final int C_SERVER_RESPONSE_NULL = -11003;
/**
 * The server returns that the update type of mini program error.
 */
public static final int C_SERVER_UPDATE_TYPE_ERROR = -11004;
/**
 * The server returns that an exception occurred while parsingdata
 */
public static final int C_SERVER_PARSE_DATA_ERROR = -11005;
/**
 * Mini program does not exist or has been removed
 */
public static final int C_SERVER_TAKE_OFF = -11006;

/**
 * Maximum monthly activity reached
 */
public static final int C_SERVER_MAU_LIMIT = -11007;
/**
 * Resource limits
 */
public static final int C_SERVER_RES_LIMIT = -11008;
/**
 * Frequency of single interface requests is too high
 */
public static final int C_SERVER_FREQ_LIMIT_API = -11011;

//////////////////////////////Client-side error////////////////////////////////////
public static final int C_CLIENT = -12000;
/**
 * The Shark instance is empty
 */
public static final int C_CLIENT_SHARK_IS_NULL = -12001;
```

©2013-2022 Tencent Cloud. All rights reserved.

```
/**
 * Previewing mini program requires login
 */
public static final int C_CLIENT_NEED_LOGIN_PREVIEW_APP = -12002;
/**
 * Data parsing exception
 */
public static final int C_CLIENT_JSON_EXCEPTION = -12003;
/**
 * Code scanning exception
 */
public static final int C_CLIENT_SCAN_ERROR = -12004;
/**
 * Mini program information is missing
 */
public static final int C_CLIENT_MINI_APP_INFO_ERROR = -12005;
/**
 * Code scanning error
 */
public static final int C_CLIENT_QRCODE_ERROR = -12006;
/**
 * The code is not a TMF mini program QR code
 */
public static final int C_CLIENT_QRCODE_INVALIDATE = -12007;
/**
 * The `appId` is empty
 */
public static final int C_CLIENT_APPID_EMPTY = -12008;
/**
 * businessId null
 */
public static final int C_CLIENT_QRCODE_BUSINESSID_NULL = -12009;
/**
 // Mini program launching error
 */
public static final int C_CLIENT_START_MINI_APP_THROWABLE = -12010;
/**
 * json parsing exception
 */
public static final int C_CLIENT_JSON_ERROR = -12011;
/**
 * Failed to download mini program
 */
public static final int C_CLIENT_MINI_APP_DOWNLOAD_FAIL = -12012;
/**
 * Mini program parsing failed
 */
```

©2013-2022 Tencent Cloud. All rights reserved.

```
public static final int C_CLIENT_MINI_APP_PARSE_FAIL = -12013;
```

MiniApp

Mini program information classes

```
/**
 * Official mini program
 */
public static final int TYPE_ONLINE = MiniSDKConst.ONLINE;
/**
 * Debug mini program
 */
public static final int TYPE_DEVELOP = MiniSDKConst.DEVELOP;
/**
 * Preview mini program
 */
public static final int TYPE_PREVIEW = MiniSDKConst.PREVIEW;
/**
 Try out the mini program demo
 */
public static final int TYPE_EXPERIENCE = MiniSDKConst.EXPERIENCE;
/**
 * Mini program ID
 */
public String appId;
/**
 * Mini program version type (official/preview/developer)
 */
public int appVerType;
/**
 * Mini program version
 */
public String version;
/**
 * Mini program name
 */
public String name;
/**
 * Mini program icon
 */
public String iconUrl;
/**
 * Mini program overview
```

©2013-2022 Tencent Cloud. All rights reserved.

```
 */
public String appIntro;
/**
 * Developer enterprise name
 */
public String appDeveloper;
/**
 * Timestamp
 */
public long time;
```

MiniScene

Scenarios for opening a mini program.

```
/**
 * Main entry of the mini program, list of [recently used mini programs]
 */
public static final int LAUNCH_SCENE_MAIN_ENTRY = 1001;
/**
 * Scan the QR code to open
 */
public static final int LAUNCH_SCENE_QR_CODE_FROM_SCAN = 1011;
/**
 * Search and open
 */
public static final int LAUNCH_SCENE_SEARCH = 2005;
```

SearchOptions

```
/**
 * Search keyword. When it is empty, all mini programs are searched for.
 */
public String keyWord = "";
/**
 * Specify primary category
 */
public String firstLevelCate;

/**
 * Specify the secondary category
```

©2013-2022 Tencent Cloud. All rights reserved.

```
 */
public String secondaryLevelCate;
```

ShareData

```
/**
    Source of the share, which is value from ShareSource
    */
   public int shareSource;
   /**
    * Target of the share, which is value from ShareTarget
    */
   public int shareTarget;
   /**
    * ID for distinguishing share channels
    */
   public int shareItemId;
   /**
    * Share title
    */
   public String title;
   /**
     * Share summary
    */
   public String summary;
   /**
    * Path to the share image, either local or network path Whether the image is lo
    */
   public String sharePicPath;
   /**
    * Whether it is a local image. If it is True, sharePicPath is the path to the l
    */
   public boolean isLocalPic;
   /**
    * Share link obtained from the server
     */
   public String targetUrl;
   /**
    View the mini program package details.
    */
   protected MiniAppInfo miniAppInfo;
```

©2013-2022 Tencent Cloud. All rights reserved.

ShareSource

```
public static class ShareSource {

   public static final int INNER_BUTTON = 11; // From mini program|internal button
   public static final int MORE_BUTTON = 12;  // From "More" option in the capsule
}
```

ShareTarget

```
public static class ShareTarget {
   Public static final int QQ = 0;//Share to QQ contacts
   Public static final int QZONE = 1;//Share to QQ Zone
   public static final int WECHAT_FRIEND = 4;//Share to Weixin contacts
   public static final int WECHAT_MOMENTS = 4;// Sahre to Weixin Moments
}
```

ShareResult

```
public static class ShareResult {
   public static final int SUCCESS = 0;// Shared successfully
   public static final int FAIL = 1;// Share failed
   public static final int CANCEL = 2;// Share canceled
}
```

MiniStartLinkOptions

```
public class MiniStartLinkOptions {
    /**
     * Whether to force the mini program to check for updates when it is opened (va
     */
    public boolean isForceUpdate = false;
    /**
     * Entry address
     */
    public String entryPath;
```

©2013-2022 Tencent Cloud. All rights reserved.

```
    /**
     * Receive error messages during mini program startup.
     */
    public ResultReceiver resultReceiver.


    /**
     * Mini-program startup parameters.
     */
    public String params;
}
```

MiniInitConfig

```
/**
 * Configuration file name in assets
 */
private String configAssetName;
/**
 Customize configuration file path
 */
private String configFilePath;
/**
 * Profile Content
 */
private String configJsonStr;
/**
 * imei, for background mini program push configuration
 */
private String imei;
/**
 * SDK log switch
 */
private boolean debug;
/**
 * Set up external shark instance
 */
private IShark shark;
/**
 * When loading configuration file, whether to check the package name in the config
 */
private boolean verifyPkg;
/**
 * Whether to use x5 kernel
```

©2013-2022 Tencent Cloud. All rights reserved.

```
 */
private boolean isUserX5Core = true;
/**
 * Whether to force use the kernel base library
 */
private boolean forceUseBaseLibInAsset;
/**
 * Path to preset offline package in assets
 */
private String assetPathOfPresets;
```

IMiniAppContext

```
/**
 * callback miniprogram info
 */
MiniAppInfo getMiniAppInfo();
```

MiniAppInfo

```
public String appId;//appID of the mini program.
public String name;//Name of the mini program.
public String iconUrl;//url of the mini program icon.
public String version;//mini program version number
public int verType;//Type of mini program: development, preview, official version.
```

IpcCallback

```
public interface IpcCallback {
   /**
    * Callback for process communication
    * @param isSucc Whether the call is successful
    * @param response Returns data
    */
   void result(boolean isSucc, Bundle response);
}
```

©2013-2022 Tencent Cloud. All rights reserved.

IpcRequestEvent

```
public Context context;
//data
public Bundle data;
//returns callback
public IpcCallback callback;
```

RequestEvent

```
//Mini program activity
public WeakReference<Activity> activityRef;
//Event name
public String event;
//Event parameters
public String jsonParams;
```

AppState

```
/**
 * Mini program launched
 */
int STATE_START = 1;
/**
 * Mini program swithed to the foreground
 */
int STATE_FOREGROUND = 2;
/**
 * Close mini program via capsule
 */
int STATE_CLOSE = 3;
/**
 * Mini program switched to the background
 */
int STATE_BACKGROUND = 4;
/**
 * Mini program destroyed
 */
int STATE_DESTROY = 5;
```

©2013-2022 Tencent Cloud. All rights reserved.

MiniAppEvent

```
/**
 * Mini program information
 */
public MiniApp miniApp;
/**
 * Whether it is a hot start
 */
public boolean isHotStart;
```

PreDownloadInfo

```
/**
 * Mini program appId
 */
public String appId;
/**
 * Whether to download mini program package
 */
public boolean isDownload;
```

IDownloadCallback

```
/**
 * Callback for successful download
 *
 * @param downloadInfo
 */
void onFinish(DownloadInfo downloadInfo);

/**
 * Callback failed
 *
 * @param downloadInfo
 */
void onError(DownloadInfo downloadInfo);
```

©2013-2022 Tencent Cloud. All rights reserved.

DownloadInfo

```
//Download IOEXCEPTION and IllegalAccessException exceptions
public static final int CODE_DOWNLOAD_IOEXCEPTION = -100001;
//Download EXCEPTION exception
public static final int CODE_DOWNLOAD_EXCEPTION = -100002;
//No internet
public static final int CODE_NO_NETWORK = -100003;
//Download parameter error
public static final int CODE_PARAM_ERROR = -100004;
/// Failed to create the download directory
public static final int CODE_DOWNLOAD_DIR_CREATE_FAIL = -100005;
/// Failed to create the download directory
public static final int CODE_MINI_APP_PARSE_FAIL = -12013;

/**
 * Mini program ID
 */
private String appId;
/**
 * Error code
 */
private int errCode;
/**
 * Error msg
 */
private String message;
```

IMiniAppFileManager

```
/**
 * Get wxfile absolute path
 * Parameters wxFilePath
 * Returns
 */
String getAbsolutePath(String wxFilePath);

/**
 * Converts an absolute path to a wxfile.
 * Parameters path
```

©2013-2022 Tencent Cloud. All rights reserved.

```
 * Returns
 */
String getWxFilePath(String path);

/**
 * Get temporary directory
 * Parameter suffix
 * Returns
 */
String getTmpPath(String suffix);
```

BaseJsPlugin

```
/**
 * :: Mini program contexts
 */
protected IMiniAppContext mMiniAppContext;
/**
 * Mini program information.
 */
protected MiniAppInfo mMiniAppInfo;
/**
 * Mini program package information.
 */
protected ApkgInfo mApkgInfo; /** ** Applet package information */ protected ApkgIn
```

©2013-2022 Tencent Cloud. All rights reserved.

SDK extension components

Last updated：2024-06-27 16:41:57

This document will guide you on how to integrate components by using extension libraries. The supported components are detailed in the following sections.

Co-Level Rendering Extension SDK

Note:
The same-layer rendering capability of the mini program SDK relies on the Tencent X5 kernel. In order to implement the same-layer rendering capability, the host application needs to integrate the same-layer rendering extension component.
If the developer's original project has integrated the Tencent X5 kernel, the Mini Program SDK supports multiple versions of the X5 kernel, and customers need to choose based on the actual conditions of their own projects.
Note：
After integrating the Mini Program SDK, developers should remember not to call the initialization of X5 themselves, because the Mini Program SDK will be initialized. If the developer calls the initialization by themselves, it may cause X5 initialization to fail, thus affecting the same-layer rendering capabilities of the Mini Program.

Public network version X5 kernel

The public version of the X5 kernel is the kernel provided by Tencent X5 for external use. It is a shared kernel and will access Tencent-related background services.

```
implementation 'com.tencent.tbs:tbssdk:${version}' //See Android SDK Updates for ve
implementation 'com.tencent.tcmpp.android:mini_extra_public_x5:${version}' //See An
```

static kernel

The static x5 kernel packages the entire kernel in aar, and supports multiple architectures such as armeabi, arm64-v7a, and arm64-v8a, but the aar package size is relatively large; it also supports same-layer rendering of small programs.

```
//Introduce dependencies into the project
implementation 'com.tencent.tmf.android:tbscore:${version}' //See Android SDK Updat
implementation 'com.tencent.tcmpp.android:mini_extra_static_x5_new:${version}' //Se

//Enable extractNativeLibs configuration in the project application and must be set
<application
    android:extractNativeLibs="true">
```

©2013-2022 Tencent Cloud. All rights reserved.

```
</application>
```

Through MiniInitConfig configuration kernel LicenseKey, configuration code is as follows (LicenseKey need to find the relevant personnel to apply).

```
ndk { abiFilters "armeabi" "armeabi-v7a" "arm64-v8a" }
```

Dynamic kernel developers need to set the LicenseKey and kernel download address through MiniInitConfig, the configuration code is as follows:

```
@ProxyService(proxy = MiniConfigProxy.class)
public class MiniConfigProxyImpl extends MiniConfigProxy {
      /**
     * Create initial configuration information
     * @return
     */
    @Override
    public MiniInitConfig buildConfig() {
      MiniInitConfig.Builder builder = new MiniInitConfig.Builder();
      MiniInitConfig config = builder
                .x5LicenseKey("")//Set LicenseKey
                .build();
    }
}
```

Dynamic X5 core

The SDK integrated into the app with dynamic kernel is small in size, and the kernel is downloaded from the server.

```
implementation 'com.tencent.tmf.android:dynamicx5:${version}'
implementation 'com.tencent.tcmpp.android:mini_extra_dynamic_x5:${version}' //See A
```

Dynamic kernel developers need to set the LicenseKey and kernel download address through MiniInitConfig, the configuration code is as follows:

```
@ProxyService(proxy = MiniConfigProxy.class)
public class MiniConfigProxyImpl extends MiniConfigProxy {
      /**
     * Create initial configuration information
     * @return
     */
    @Override
    public MiniInitConfig buildConfig() {
      MiniInitConfig.Builder builder = new MiniInitConfig.Builder();
      MiniInitConfig config = builder
           .x5LicenseKey("")//Set LicenseKey
```

©2013-2022 Tencent Cloud. All rights reserved.

```
            .coreUrl32("")//32-bit kernel download address
            .coreUrl64("")//64-bit kernel download address
            .build();
    }
}
```

Deploy the kernel file to the server and generate the corresponding URL address according to the rules.

The rules for generating kernel file URL addresses are as follows:

```
(http or https) ://domain/path/{versionCode}/{tbscore.tbs}
```

{versionCode}: The kernel version number provided when obtaining the kernel.

{tbscore.tbs}: is the core file.

It is necessary to ensure that when the URL is intercepted with "/", the kernel file is the last digit and the versionCode is the second to last digit.

Examples are as follows:

32-bit kernel download address

64-bit kernel download address

In this example, the kernel version number is 46471 and the kernel file is tbs_core_release.tbs.

X5 kernel event callback

For dynamic kernels, you can listen to the download and initialisation results of dynamic kernels in the following way:

```
@ProxyService(proxy = IX5EventProxy.class)
public class X5EventProxyImpl implements IX5EventProxy {
    /**
     * Dynamic kernel download progress
     * @param progress
     */
    @Override
    public void onDownloadProgress(int progress) {
        Log.d(ModuleApplet.TAG, "X5EventProxyImpl onDownloadProgress=" + progress);
    }

    /**
     * Dynamic kernel download failed
     * @param code
     * @param msg
     */
    @Override
    public void onDownloadFailed(int code, String msg) {
        Log.d(ModuleApplet.TAG, "X5EventProxyImpl onDownloadFailed=" + code + " " +
    }

    /**
```

©2013-2022 Tencent Cloud. All rights reserved.

```
     * Dynamic kernel download finish
     */
    @Override
    public void onDownloadFinish() {
        Log.d(ModuleApplet.TAG, "X5EventProxyImpl onDownloadFinish");
    }

    /**
      * Kernel initialization callback
      * @param isX5 true: x5 initialization is successful; false: x5 initialization
     */
    @Override
    public void init(boolean isX5) {
        Log.d(ModuleApplet.TAG, "X5EventProxyImpl isX5=" + isX5);
    }
}
```

QR Code Extension SDK

Component description: If the developer's mini-program uses the code scanning capability, it is necessary to add the following SDK to support this function. Otherwise, it is recommended to refrain from adding it, as this can help reduce the overall size of the App package.
Integration Method: Add the QR code extension library dependency as follows:

```
//Code Scanning Extension Component
implementation 'com.tencent.tcmpp.android:mini_extra_qrcode:${version}' // See Andr
```

Upon adding the Code Scanning Extension SDK, the list of supported mini-program APIs is expanded as follows:

API name	Description
wx.scanCode	Invoke the client's scanning interface for code scanning.

Involved permissions:

Permission name	Description
Camera permission	It is necessary to request camera permissions for code scanning.
File read/write permissions	It is essential to request file read and write permissions to identify QR codes in local images.

©2013-2022 Tencent Cloud. All rights reserved.

Tencent Location Map Extension SDK

Component description: For App development in Chinese mainland, if the developer's mini-program utilizes the mini-program map capability, it is required to add the following SDK to support Tencent map functionality. Otherwise, it is recommended to refrain from adding it, as this can help reduce the overall size of the App package.
Integration Method: Add the map extension library dependency as follows:

```
implementation 'com.tencent.tcmpp.android:mini_extra_map:${version}' // See Android
implementation 'com.tencent.map:tencent-map-vector-sdk:4.5.10'  // See Tencent Map
implementation 'com.tencent.map:sdk-utilities:1.0.7'
implementation 'com.tencent.map.geolocation:TencentLocationSdk-openplatform:7.4.7'
```

You need to configure your project in the Tencent Location Service Console and obtain the API key required to access Tencent Map Services. For detailed operations, see the Development Guide.
Upon completion of the aforementioned steps, you need to configure your API key in the Android project. Add the following meta-data in the AndroidManifest.xml file, and fill in your API key at the (YOUR_API_KEY) location:

```
<application
    ...
    <meta-data
    android:name="TencentMapSDK"
    android:value="(YOUR_API_KEY)" />
    ...
</application>
```

Upon adding the Tencent Map Extension SDK, the list of supported mini-program APIs is expanded as follows:

API name	Description
Map	Supports map-related interfaces, including map display, using the map to select locations, and querying POI, among others.

Involved permissions:

Permission name	Description
Positioning Permissions	Positioning permissions are required for displaying map positioning.

Google and Huawei Positioning Map Extension SDK

Component description: For App development in outside of China, if the developer's mini-program utilizes the mini-program map capability, it is required to add the following SDK to support Google Map functionality. Otherwise, it is

©2013-2022 Tencent Cloud. All rights reserved.

recommended to refrain from adding it, as this can help reduce the overall size of the App package.
Integration Method: Add the map extension library dependency as follows:

```
implementation 'com.tencent.tcmpp.android:mini_extra_google_map:${version}' // See
implementation 'com.google.android.gms:play-services-maps:18.1.0'          //See Go
implementation 'com.google.maps.android:android-maps-utils:2.3.0'
```

Some Huawei devices do not support embedded Google Maps, which may result in the map not being displayed. You can additionally integrate Petal Map as a supplementary solution, and the mini-program framework will prioritize the use of Petal Map on Huawei devices.

```
repositories {
  maven {url 'https://developer.huawei.com/repo/'}
}
implementation 'com.tencent.tcmpp.android:mini_extra_huawei_map:${version}' // See
implementation 'com.huawei.hms:maps:6.9.0.300'          //See Huawei Maps Documenta
implementation 'com.huawei.hms:maps-basic:6.9.0.300'
implementation 'com.huawei.hms:site:6.5.1.300'
```

In the case of using Google Maps, you need to configure your Google Cloud project in your Google Cloud Console and obtain the API key required to access Google Map services. See the specific operational steps in **Setting up in Google Cloud Console** and **Using API Key**.
Upon completion of the aforementioned steps, you need to configure your API key in the Android project. Add the following meta-data in the AndroidManifest.xml file, and fill in your API key at the (YOUR_API_KEY) location:

```
<application
    ...
    <meta-data
        android:name="com.google.android.geo.API_KEY"
        android:value="(YOUR_API_KEY)" />
     ...
</application>
```

In the scenario of using Petal Map, you need to establish a project in your Huawei management console, activate the map and location services, and obtain the API key used by the location services. For specific operational steps, see Configuring AppGallery Connect. Then, follow the guide of Integrating HMS Core SDK to download the "agconnect-services.json" file to your project and configure the Huawei AGC plugin.
You need to add the following meta-data in the AndroidManifest.xml file, and fill in your API key at the (YOUR_API_KEY) location to use Huawei's location services normally:

```
<application
    ...
    <meta-data
    android:name="HuaweiApiKey"
    android:value="(YOUR_API_KEY)" />
```

©2013-2022 Tencent Cloud. All rights reserved.

```
    ...
</application>
```

Note:

For security reasons, it is recommended that you apply for an API key specifically for location services.

After adding the Google and Huawei map extension SDKs, the list of supported mini-program APIs is expanded as follows:

API name	Description
Map	Supports map-related interfaces and components, including map display, using the map to select locations, and querying POI, among others.

Involved permissions:

Permission name	Description
Positioning Permissions	Positioning permissions are required for displaying map positioning.

Live Streaming Component Extension SDK

Component Description: If you need to use the live streaming components (live-player and live-pusher) for developing live streaming push and pull scenarios, you need to add the following SDKs to support the implementation of related live streaming component features.

Integration Method: Add live streaming component dependencies:

```
// Live Streaming Component Support Library
implementation 'com.tencent.tcmpp.android:mini_extra_trtc_live:${version}' // See A
// Live Streaming Component Library
implementation 'com.tencent.liteav:LiteAVSDK_Professional:latest.release' //See Ten
```

In addition to completing the addition of the above dependencies, you also need to override and implement the following methods of BaseMiniAppProxyImpl, providing the LicenseUrl and LicenseKey required by the live streaming component, to complete the initialization information configuration of the live streaming component; if you do not configure the correct LicenseUrl and LicenseKey, the live streaming component functionality will be unavailable.

Note:

For the method of obtaining LicenseUrl and LicenseKey, see Add and Renew License.

```
@ProxyService(proxy = MiniAppProxy.class)
public class MiniAppProxyImpl extends BaseMiniAppProxyImpl {
      @Override
    public MiniConfigData configData(Context context, int configType, JSONObject pa
        if(configType == MiniConfigData.TYPE_LIVE) {
```

©2013-2022 Tencent Cloud. All rights reserved.

```
            //Live Streaming Configuration
            MiniConfigData.LiveConfig liveConfig = new MiniConfigData.LiveConfig();
            //The following key and url can only be used for demo purposes.
            liveConfig.licenseKey = "";
            liveConfig.licenseUrl = "";

            return new MiniConfigData
                    .Builder()
                    .liveConfig(liveConfig)
                    .build();
        }

        return null;
    }
}
```

Upon adding the Live Streaming Extension SDK, the list of supported mini-program APIs is expanded as follows:

API name	Description
wx.createLivePusherContext	Creating live streaming push context
LivePusherContext	Supports relevant LivePusherContext interfaces
wx.createLivePlayerContext	Creating live streaming pull context
LivePlayerContext	Supports relevant LivePlayerContext interfaces
Component	-
live-pusher	Push stream tag
live-player	Broadcast tag

The permissions involved are as follows:

Permission name	Description
Camera permission	-
Audio recording permission	-

LBS Extension SDK

Component Description: The LBS component provides capabilities related to location information, compass, accelerometer, positioning, and device orientation.

©2013-2022 Tencent Cloud. All rights reserved.

Integration method: Add the LBS extension library dependency as follows:

```
implementation 'com.tencent.tcmpp.android:mini_extra_lbs:${version}' // See Android
```

Upon adding the LBS Extension SDK, the list of supported mini-program APIs is expanded as follows:

API name	Description
Location Information	Supports location information related interfaces
Compass	Supports compass related interfaces
Accelerometer	Supports accelerometer related interfaces
Device Orientation	Supports device orientation related interfaces
Gyroscope	Supports gyroscope related interfaces

The LBS Extension SDK involves the following permissions:

Permission name	Description
Positioning	Obtaining positioning depends on positioning permissions.

Bluetooth Extension SDK

Component Description: After adding the Bluetooth extension library, you can use the Bluetooth-related APIs.

Integration method: Add the Bluetooth extension library dependency as follows:

```
implementation 'com.tencent.tcmpp.android:mini_extra_bluetooth:${version}'//Please
```

Upon adding the LBS Extension SDK, the list of supported mini-program APIs is expanded as follows:

API name	Description
Bluetooth - General	General Bluetooth Interface
Bluetooth - Low Energy Peripheral Device	Peripheral Device Related Interface
Bluetooth - Low Energy Central Device	Central Device Related Interface
Bluetooth - Beacon	Bluetooth Beacon Related Interface

The Bluetooth Extension SDK involves the following permissions:

©2013-2022 Tencent Cloud. All rights reserved.

Permission name	Description
Bluetooth	Operating Bluetooth requires the Bluetooth permissions.
Positioning	Bluetooth device search relies on positioning permissions.

NFC Extension SDK

Component Description: By incorporating the NFC Extension SDK, it is possible to achieve capabilities related to NFC read and write operations.
Integration method: Add the NFC Extension SDK dependency as follows:

```
implementation 'com.tencent.tcmpp.android:mini_extra_nfc:${version}'//Please refer
```

Upon adding the NFC Extension SDK, the list of supported mini-program APIs is expanded as follows:

API name	Description
wx.getNFCAdapter	Acquiring the NFC Operation Management Instance Object
NFCAdapter	Supports NFCAdapter related interfaces.
NFC Instances (NfcA, NfcB, NfcV, NfcF, Ndef, IsoDep, MifareUltralight, MifareClassic)	Supports interfaces related to NFC tag instances.

Involved permissions:

Permission name	Description
NFC Permissions	Requires acquisition of NFC permissions.

Biometric Authentication Extension SDK

Component description: The Biometric Authentication Extension SDK provides capabilities related to fingerprint and facial recognition.
Integration Method: Add the Biometric Authentication Extension Library dependency as follows:

```
implementation 'com.tencent.tcmpp.android:mini_extra_soter:${version}'//Please refe
```

Upon adding the Biometric Authentication Extension SDK, the list of supported mini-program APIs is expanded as follows:

©2013-2022 Tencent Cloud. All rights reserved.

API name	Description
wx.startSoterAuthentication	-
wx.checkIsSupportSoterAuthentication	-
wx.checkIsSoterEnrolledInDevice	-

Involved permissions:

Permission name	Description
Fingerprint Access	Fingerprint access permission is required.

Clipboard Extension SDK

Component Description: Provides the capability to access the clipboard.

Integration Method: Add the extension library dependency as follows:

```
implementation 'com.tencent.tcmpp.android:mini_extra_clipboard:${version}' // See A
```

Upon adding the LBS Extension SDK, the list of supported mini-program APIs is expanded as follows:

API name	Description
wx.getClipboardData	-
wx.setClipboardData	-

Involved permissions:

Permission name	Description
Clipboard Permissions	Clipboard access permission is required.

Address Book Extension SDK

Component Description: Provides capabilities related to contact access.

Integration Method: Add the extension library dependency as follows:

```
implementation 'com.tencent.tcmpp.android:mini_extra_contact:${version}' // See And
```

Upon adding the Address Book Extension SDK, the list of supported mini-program APIs is expanded as follows:

©2013-2022 Tencent Cloud. All rights reserved.

API name	Description
wx.addPhoneContact	Add contact
wx.chooseContact	Select contact

Involved permissions:

Permission name	Description
Contacts read/write permissions	Access and write permissions for contacts are required.

Document Engine Extension SDK

Component Description: Provides the capability to open documents (such as PDF, Word, Excel, PPT, etc.).

Integration Method: Add the extension library dependency as follows:

```
implementation 'com.tencent.tmf.android:tbs-doc-support:${version}'//See Android SD
implementation 'com.tencent.tcmpp.android:mini_extra_doc:${version}'//See Android S
```

Configure the Document Engine LicenseKey. For detailed configuration code instructions, refer to SDK Update News.

The LicenseKey requires application through relevant personnel.

```
@ProxyService(proxy = MiniConfigProxy.class)
public class MiniConfigProxyImpl extends MiniConfigProxy {
      /**
     * Create initial configuration information
     * @return
     */
    @Override
    public MiniInitConfig buildConfig() {
      MiniInitConfig.Builder builder = new MiniInitConfig.Builder();
      MiniInitConfig config = builder
                .docLicenseKey("")//Set LicenseKey
                .build();
    }
}
```

Media extension SDK

Component description: Provides default implementations of chooseMedia and previewMedia.

©2013-2022 Tencent Cloud. All rights reserved.

Integration method: Add extension library dependencies as follows:

```
implementation 'com.tencent.tcmpp.android:mini_extra_media_support:${version}'//See
```

Implement the MediaImageLoaderProxy proxy and use a custom image loading implementation for image loading in the mini_extra_media_support library.

Note：

You can implement custom chooseMedia logic by implementing the MediaChooseJsProxy proxy.

```
@ProxyService(proxy = MediaImageLoaderProxy.class)
public class CustomMediaImageLoaderProxy implements MediaImageLoaderProxy {
    private GlideImageEngine glideImageEngine = new GlideImageEngine();

    @Override
    public ImageEngine getCustomImageEngine() {
        return glideImageEngine;
    }

    static class GlideImageEngine implements ImageEngine {

        @Override
        public void loadPhoto(@NonNull Context context, @NonNull Uri uri, @NonNull
            Glide.with(context).load(uri).transition(withCrossFade()).into(imageVie

        }

        @Override
        public void loadGifAsBitmap(@NonNull Context context, @NonNull Uri gifUri,
            Glide.with(context).asBitmap().load(gifUri).into(imageView);

        }

        @Override
        public void loadGif(@NonNull Context context, @NonNull Uri gifUri, @NonNull
            Glide.with(context).asGif().load(gifUri).transition(withCrossFade()).in

        }

        @Override
        public Bitmap getCacheBitmap(@NonNull Context context, @NonNull Uri uri, in
                throws Exception {
            return Glide.with(context).asBitmap().load(uri).submit(width, height).g
        }
    }
}
```

©2013-2022 Tencent Cloud. All rights reserved.

©2013-2022 Tencent Cloud. All rights reserved.

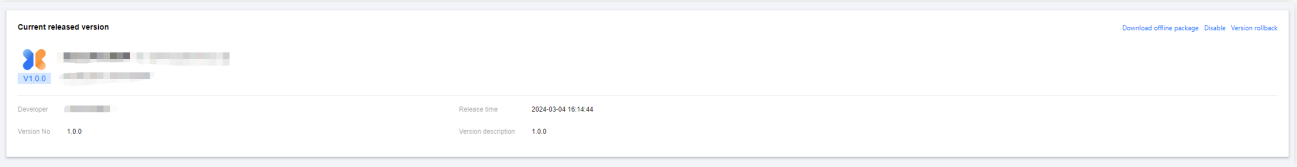
Preconfigure offline mini programs

Last updated：2024-06-27 10:50:02

Offline mini programs are embedded within the host app. You need to download the mini program package from the console and import it into the host app project, bundling it with the app. Users can open and run these mini programs without downloading them from the backend, even without an internet connection.

Steps

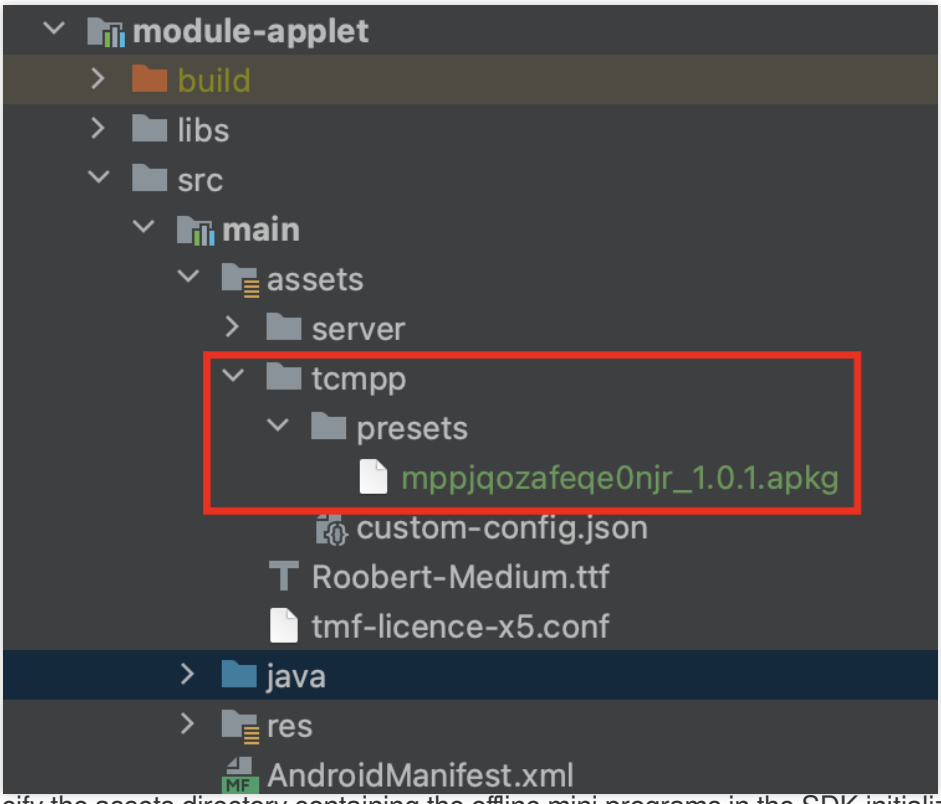
1. Download the required mini program from the console.



2. Copy the downloaded mini program package to a custom assets directory. Follow strict naming conventions and do not change it.

> Naming conventions: `{miniAppId}_{miniAppVersion}.apkg`

©2013-2022 Tencent Cloud. All rights reserved.



3. Specify the assets directory containing the offline mini programs in the SDK initialization configuration.

```
@ProxyService(proxy = MiniConfigProxy.class)
public class MiniConfigProxyImpl extends MiniConfigProxy {
    robincxiao
    @Override
    public Application getApp() { return ModuleApplet.sApp; }

    1 usage  robincxiao +1 *
    @Override
    public MiniInitConfig buildConfig() {
        MiniInitConfig.Builder builder = new MiniInitConfig.Builder();
        return builder
                .assetPathOfPresets("tcmpp/presets")
                .build();
    }
}
```

Notes

Offline mini programs must follow the standard binding and publishing process. Only published miniprograms can be downloaded as offline packages;

©2013-2022 Tencent Cloud. All rights reserved.

Offline mini programs adhere to version management logic, including new releases, version rollbacks, and deprecations. If the online version differs from the preloaded version, the client will fetch the online version;
If a mini program is deprecated, the corresponding offline mini program in the app will also become unusable.

©2013-2022 Tencent Cloud. All rights reserved.

Android SDK Description

Last updated：2024-06-27 10:49:45

Why use multi-process for Android SDK?

The Android runtime SDK uses a multi-process mechanism, where the mini-program and the host app run in separate processes. The processes do not interfere with each other and transfer data through cross-process communication. This approach offers several benefits:
Does not occupy the memory of host app The system allocates separate memory for the mini program process, preventing memory overflow issues in the host app.
Ensure safe and stable operation of the host App Any exceptions in the mini program process do not affect the host app, ensuring continuous operation even if the mini program crashes.
Data decoupling Processes are naturally isolated in memory, preventing direct data access between processes and avoiding data coupling issues.

How is multi-process implemented?

Android does not provide APIs for dynamic process creation. Instead, processes are created by binding them to Activities using the android:process attribute in AndroidManifest.xml. When an Activity is first launched, the system creates a new process and assigns the Activity to it.
To enhance user experience, different mini programs are displayed in separate positions in the recent tasks list. This is achieved by setting the android:taskAffinity attribute in AndroidManifest.xml, placing Activities of different mini-program processes in separate task stacks.

How to manage mini program processes?

The SDK can create up to 5 mini program processes simultaneously. If the number of mini program processes is below this limit, each new mini program will start in a new process. Once the limit is reached, any additional mini programs will reuse existing processes.
When opening the same mini program, the SDK first checks if it is already running. If it is, the corresponding process is reactivated.
The main process handles the creation, reuse, and activation of mini program processes, managing all these operations centrally.

How does mini program processes communicate with the main process?

Bidirectional communication between mini program processes and the host process is achieved using AIDL (Android Interface Definition Language). Both the mini program and main processes bind to the same remote service. Through the Binder mechanism, they obtain a proxy object for the service, which allows them to call AIDL interfaces for inter-process communication.

©2013-2022 Tencent Cloud. All rights reserved.

How to ensure process and thread security

Currently, all resources that need to be written during the mini program's runtime are stored in the application's sandbox directory specific to the mini program. This setup avoids writing to globally shared resources, thus eliminating inter-process competition for the same system resources. For potential process security issues, we use process-safe methods for reading and writing data during development, such as file locks for file access and ContentProvider instead of SharedPreferences for data sharing.

There are no direct thread security issues between processes. Thread security concerns arise only when different mini program processes concurrently access the main process's memory via inter-process communication or when there is concurrent read/write access within a process. The SDK addresses these thread security issues using locks, synchronized methods, and Java concurrency utilities.

©2013-2022 Tencent Cloud. All rights reserved.

Android FAQs

Last updated：2024-09-11 16:02:59

What is same-layer rendering?

Most mini program content is rendered on a WebView. If you consider the WebView as a separate layer, the system's native components are on a higher layer. These two layers are completely independent, so you can't simply use z-index to control the relative layering of native and non-native components. As shown below, non-native components are on the WebView layer, while native components and cover-view/cover-image are on a higher layer.
Same-layer rendering means using certain technical methods to render native components directly on the WebView layer. In this case, the native component layer no longer exists, and native components are directly mounted to WebView nodes. You can use native components almost like non-native ones, such as using `view` and `image` to overlay native components, using z-index to specify the layer of native components, placing native components in containers like scroll-view, swiper, movable-view, and setting styles via WXSS.

How is same-layer rendering implemented?

On Android, mini programs use Chromium as the WebView rendering layer. The WebView on Android renders separately, producing a complete view. Therefore, other methods are needed to achieve "same-layer rendering." Our research found that Chromium supports the WebPlugin mechanism, a browser kernel plugin mechanism mainly used to parse and describe embed tags. Android's same-layer rendering is implemented based on the embed tag combined with Chromium (customized x5 kernel) extensions.

```
<embed id="web-plugin" type="plugin/video" width=300 height=300>
```

The general process for same-layer rendering on Android is as follows:
The WebView creates an embed DOM node and specifies the component type.
The Chromium kernel creates a WebPlugin instance and generates a RenderLayer.
The Android client initializes a corresponding native component.
The Android client draws the image of the native component on "SurfaceTexture" that is bound to "RenderLayer".
Notify the Chromium kernel to render the RenderLayer.
The Chromium renders the embed node and displays it on the screen.
This method can be used for rendering native components like map, video, canvas, camera, textarea, and input.

Troubleshooting mini program startup failures

Mini program startup failures can occur for several reasons:
Incorrect CONFIGURATION FILE Path: The configAssetName should be the full path to the file in the assets directory. If the configuration file is in a subdirectory, include the directory path, e.g., server/tcmpp-android-configurations.json.

©2013-2022 Tencent Cloud. All rights reserved.

Modification of configuration file: Do not modify the mini program configuration file, as this will prevent the mini program from running correctly.

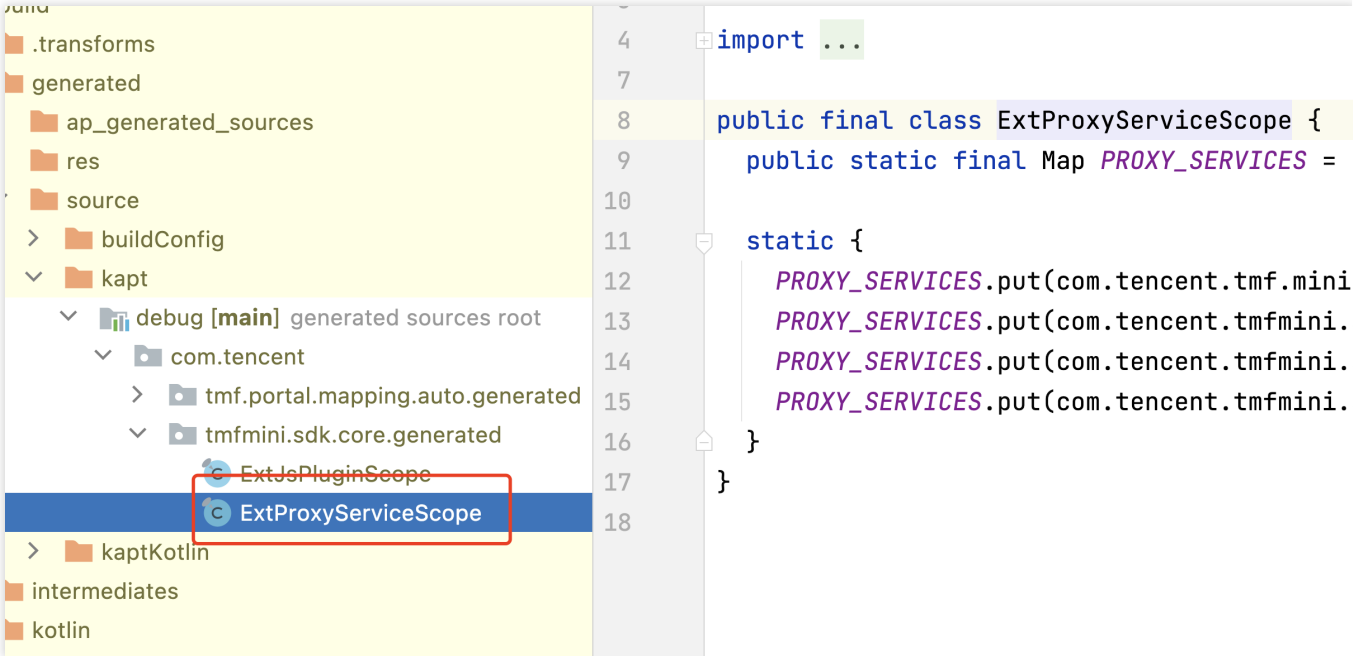
Package name mismatch: The **packageName** in the configuration file must match the application's **applicationId**. The app will fail to run if they do not match because the **packageName** is verified during initialization. You can disable package name verification with the following setting:

```
MiniInitConfig config = builder
        .verifyPkg(false) // Ignore the package name verification
        .build();
```

©2013-2022 Tencent Cloud. All rights reserved.

```
android {
    compileSdkVersion 30
    defaultConfig {
        applicationId "com.tencent.tmf.mir
        //feed back
//          applicationId "tmf.tencent.tmf.c
        minSdkVersion 21
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        ndk { NdkOptions it -> abiFilters "ar
```

Initialization annotation: Ensure that the initialization annotation @ProxyService has generated the ExtProxyServiceScope class.



How does the SDK ensure privacy compliance?

©2013-2022 Tencent Cloud. All rights reserved.

The mini program SDK initializes when the developer calls methods from the TmfMiniSDK class. Therefore, you should call these methods only after the user has agreed to the privacy compliance authorization.

How to troubleshoot customized mini program API errors?

Check if XxxJsPluginScope is generated in the compilation path, as shown below:

Ensure that the event names defined on the client side match the method names called by the mini program.**Note that custom mini-program API event names are case-sensitive**.

Mini program domain and privacy API validation logic

During usage, the mini program will validate the legality of the API request domain names. If privacy APIs are set in the management backend, authorization checks will also be performed. However, validation is skipped in the following scenarios:

The mini program is running in a non-production version and the mini program debugging is enabled, see Mini program debugging.

Modular project support

When multiple modules in a modular project use the @JsPlugin or @ProxyService annotations simultaneously, the following error may occur during the Make Project process:

©2013-2022 Tencent Cloud. All rights reserved.

```
Execution failed for task ':DEMO:mergeLibDexDebug'.
> A failure occurred while executing com.android.build.gradle.internal.tasks.Workers$Acti
   > com.android.builder.dexing.DexArchiveMergerException: Error while merging dex archiv
     Learn how to resolve the issue at https://developer.android
      .com/studio/build/dependencies#duplicate_classes.
     Program type already present: com.tencent.tmfmini.sdk.core.generated.ExtJsPluginScop

* Try:
Run with --stacktrace option to get the stack trace. Run with --info or --debug option to
 output. Run with --scan to get full insights.
```

To support multi-module projects, configure as follows:

1. Add the following code to the build.gradle file of each module using the @JsPlugin or @ProxyService annotation:

```
android {
    defaultConfig {
        javaCompileOptions {
            annotationProcessorOptions {
                // Configure module name: Define a unique name following Android cl
                arguments = [tcmppModuleName: "Demo"]
            }
        }
    }
}
```

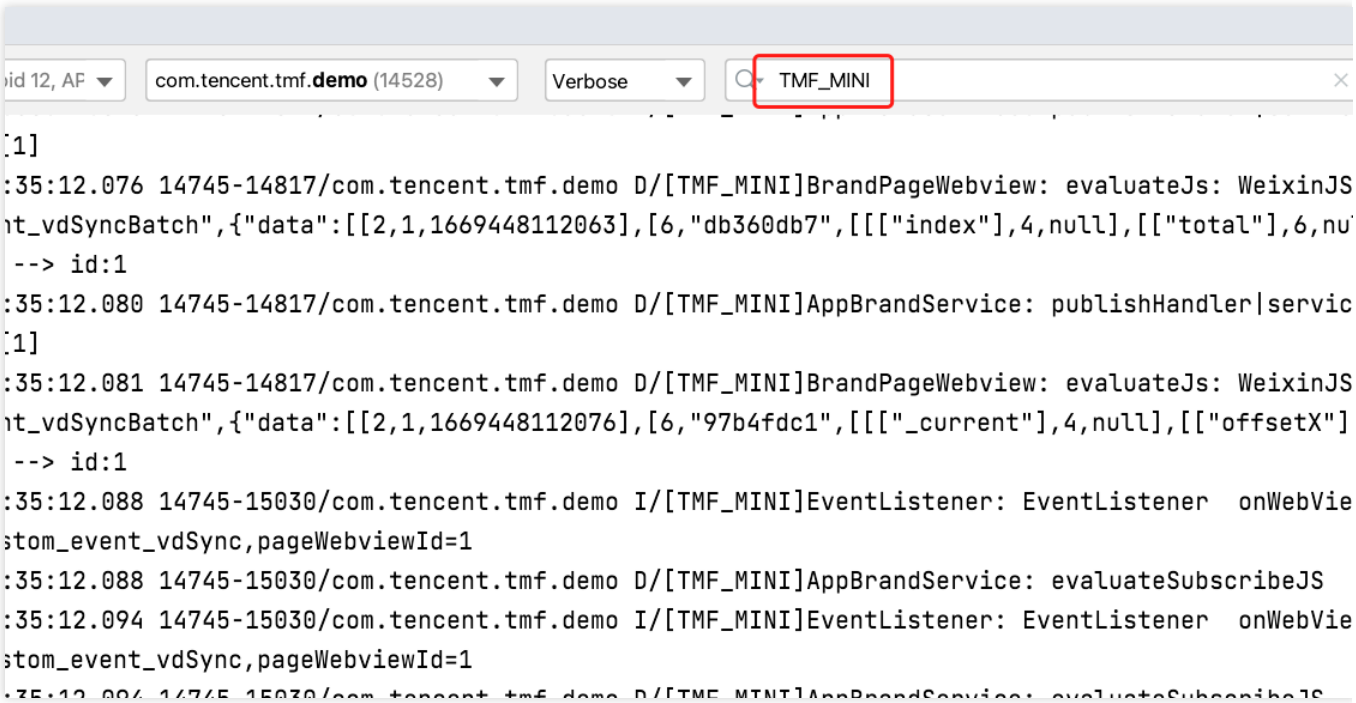
2. Register the module in the initialization code:

```
@ProxyService(proxy = MiniConfigProxy.class)
public class MiniConfigProxyImpl extends MiniConfigProxy {
    @Override
    public MiniInitConfig buildConfig() {
        MiniInitConfig.Builder builder = new MiniInitConfig.Builder();

        // Register all modules defined above; the registerModule parameter should
        return builder
```

How to view SDK log output?

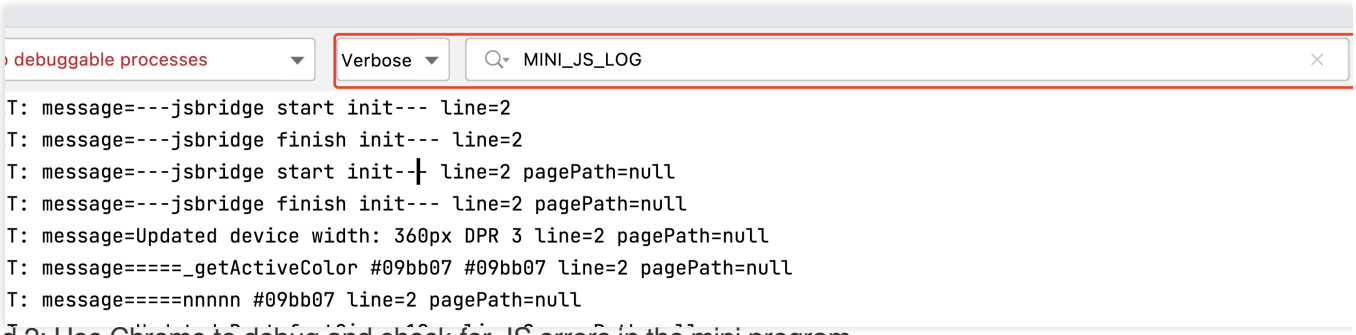
1. Developers can filter SDK logs using the keyword TMF_MINI in the development tools.

©2013-2022 Tencent Cloud. All rights reserved.



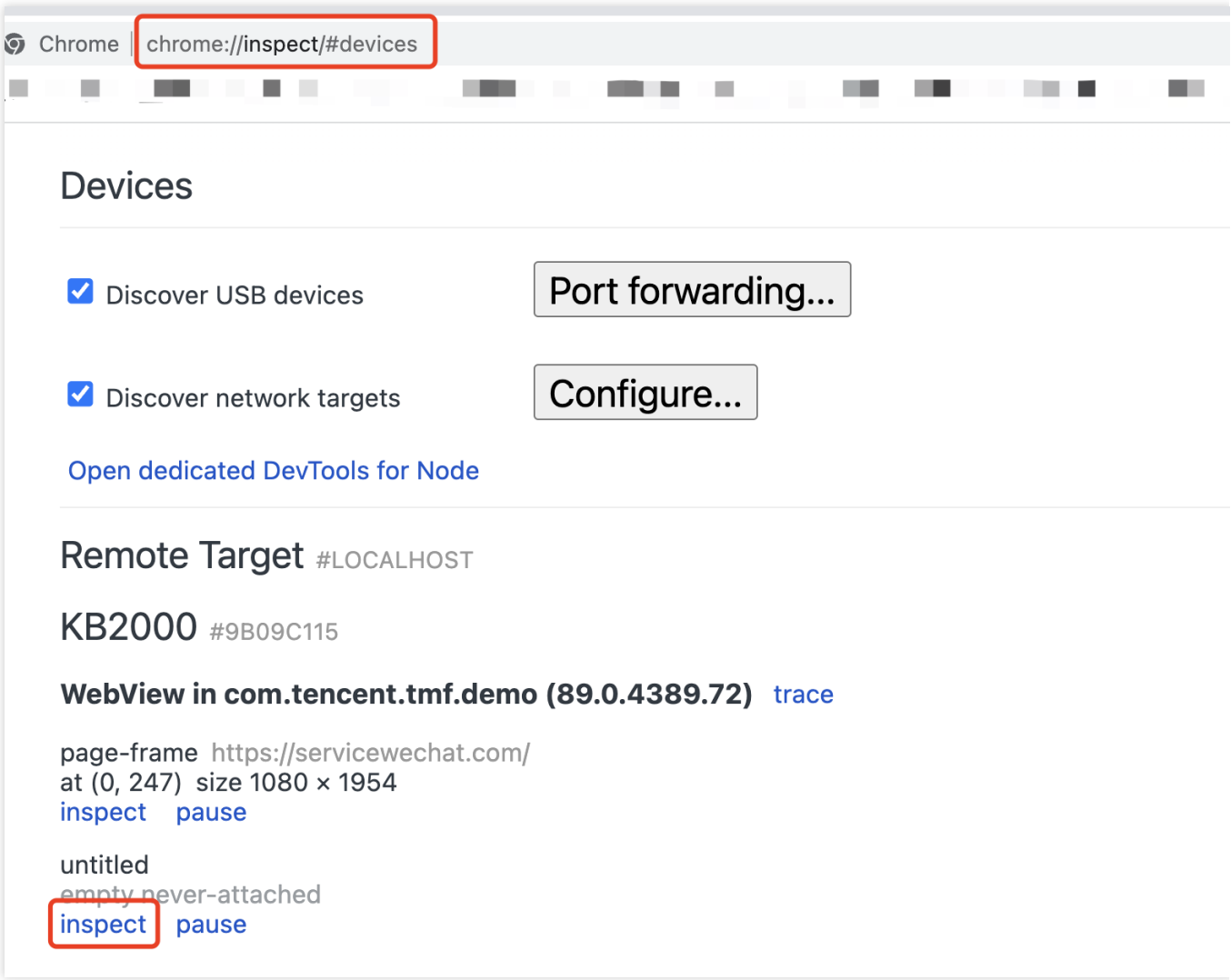
2. To view mini program JS error logs.

Method 1: Filter JS logs using the keyword MINI_JS_LOG.



Method 2: Use Chrome to debug and check for JS errors in the mini program.

©2013-2022 Tencent Cloud. All rights reserved.



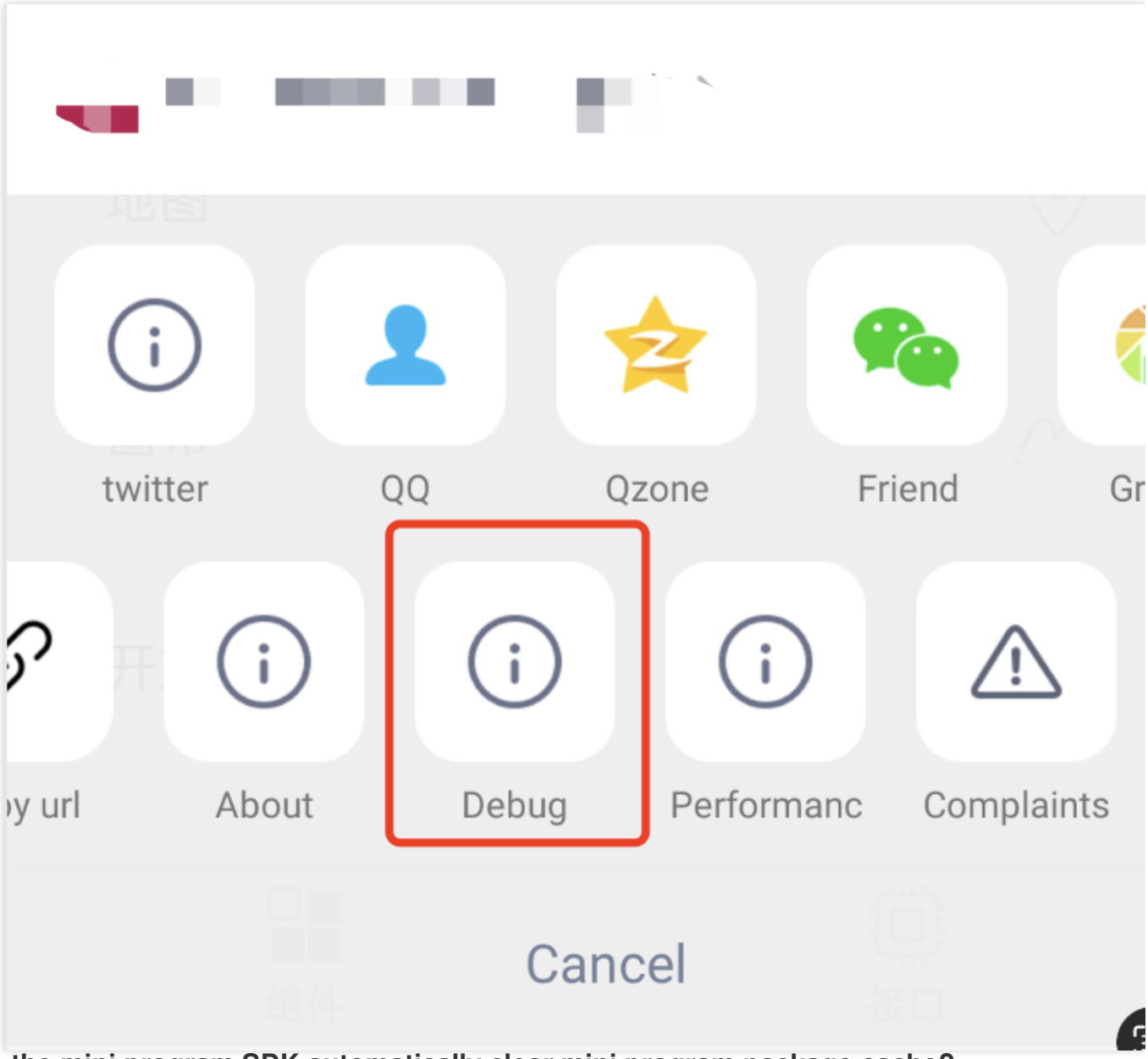
How to enable mini program debugging mode?

To facilitate debugging and viewing log outputs, the client needs to enable the mini program debug entry. Refer to Mini program host customization - "Custom Capsule" getMoreItems implementation.

```
/**
 * Returns the buttons for the capsule's more panel. The extended button IDs should
 * Note: This method is called in the mini program process.
 * @param builder
 * @return
 */
public ArrayList<MoreItem> getMoreItems(MoreItemList.Builder builder){
  builder.addDebug("Debug", icon) // Set the Debug button on the control panel
  return builder.build();
}
```

After setting this, the mini program control panel will display a debug button. Click it to enable debug mode, then restart the mini program to enter debug mode.

©2013-2022 Tencent Cloud. All rights reserved.



Does the mini program SDK automatically clear mini program package cache?

No, developers need to manually call the mini program SDK's provided deletion methods to clear the mini programs (Deleting mini programs). If not cleared proactively, the host app's cache will increase as more mini programs are used.

Why can't I use the mini program normally after enabling the R8 full mode?

Stricter obfuscation rules are applied in the R8 full mode, which may cause some annotations of the earlier version of the mini program SDK to be lost, and thus the mini program may not be started normally. Android Gradle Plugin forces the R8 full mode to be enabled during compilation on 8.0 and later versions. Please upgrade the mini program SDK mini_core to version 2.0.5 or later to be compatible with the R8 full mode.

©2013-2022 Tencent Cloud. All rights reserved.

Android Error Codes

Last updated：2024-07-29 16:28:55

Error Code	Description
0	Success
-11001	Shark Network Error
-11002	Error code returned by server
-11003	The response returned by the server is null.
-11004	The server returns Mini programs update type problem
-11005	Exception in data parsing returned by server
-11006	Mini programs do not exist or have been removed
-11007	Maximum monthly life limit reached
-11008	resource limit
-11011	Specific interface request frequency too high
-11012	Interface request frequency too high
-12001	Shark instance empty
-12002	Preview Mini programs requires login
-12003	Data parsing exception
-12004	Scan exception
-12005	Missing information for Mini programs
-12006	Code scanning error
-12007	Two-dimensional code of non-tmf Mini programs
-12008	Mini programs appId is empty
-12009	businessId is empty
-12010	Mini programs start abnormally

©2013-2022 Tencent Cloud. All rights reserved.

-12011	json parsing exception
-12012	Mini programs download failed
-12013	Mini programs failed to parse

©2013-2022 Tencent Cloud. All rights reserved.

Android Privacy Compliance

Last updated：2024-06-27 10:49:03

Information about the system permissions involved in the mini program SDK:

SDK name	Required system permissions	Permission usage description
mini_core (including gateway)	android.permission.INTERNET android.permission.READ_EXTERNAL_STORAGE android.permission.WRITE_EXTERNAL_STORAGE android.permission.READ_PHONE_STATE android.permission.ACCESS_WIFI_STATE android.permission.ACCESS_NETWORK_STATE android.permission.WAKE_LOCK android.permission.READ_MEDIA_AUDIO android.permission.CHANGE_NETWORK_STATE	Access the internet for mini program container and backend interactions. Access local storage for data persistence.
mini_extra_qrc ode	android.permission.CAMERA android.permission.WRITE_EXTERNAL_STORAGE android.permission.FLASHLIGHT	Access the camera for QR code scanning. Access local images for QR code recognition.
mini_extra_v8	None	-
mini_extra_dynamic_x5	None	-
mini_extra_static_x5	None	-
mini_extra_static_x5_new	None	-
mini_extra_public_x5	None	-
mini_extra_map	None	-
mini_extra_google_map	android.permission.ACCESS_FINE_LOCATION	Access precise location for location services.
mini_extra_huawei_map	android.permission.INTERNET android.permission.ACCESS_NETWORK_STATE android.permission.CHANGE_WIFI_STATE android.permission.ACCESS_COARSE_LOCATION android.permission.ACCESS_FINE_LOCATION	Required for Huawei map services integration.
mini_extra_trtc_live	None	-

©2013-2022 Tencent Cloud. All rights reserved.

mini_extra_nfc	android.permission.NFC android.permission.NFC_TRANSACTION_EVENT	Required for using the device's NFC capabilities.
mini_extra_lbs	android.permission.ACCESS_FINE_LOCATION	Access device location for location-related APIs.
mini_extra_bluetooth	android.permission.BLUETOOTH_ADMIN android.permission.BLUETOOTH_SCAN android.permission.BLUETOOTH_ADVERTISE android.permission.BLUETOOTH_CONNECT	Required for Bluetooth-related API capabilities.
mini_extra_contact	android.permission.WRITE_CONTACTS android.permission.READ_CONTACTS	Read and write contacts for mini program communication-related APIs.
mini_extra_soter	android.permission.USE_FINGERPRINT android.permission.USE_BIOMETRIC	Biometric permissions for mini program biometric-related APIs.
mini_extra_clipboard	None	-
mini_extra_calendar	android.permission.READ_CALENDAR android.permission.WRITE_CALENDAR	Calendar access for mini program schedule management APIs.
mini_extra_lamemp3	None	-
mini_extra_doc	None	-
mini_extra_wifi	android.permission.ACCESS_FINE_LOCATION android.permission.ACCESS_WIFI_STATE android.permission.CHANGE_WIFI_STATE android.permission.ACCESS_NETWORK_STATE android.permission.CHANGE_NETWORK_STATE	Network state and modification permissions for mini program Wi-Fi management APIs.
mini_extra_network	android.permission.INTERNET android.permission.ACCESS_NETWORK_STATE	Network and network state permissions for mini program network-related APIs.
mini_extra_media_support	None	-
tbscore	android.permission.INTERNET	-

©2013-2022 Tencent Cloud. All rights reserved.

dynamicx5	android.permission.INTERNET	-
com.tencent.tbs:tbssdk	Provided by third-parties, unknown.	-
tbs-doc-support	Provided by third-parties, unknown.	-

For more information, see Privacy Compliance.

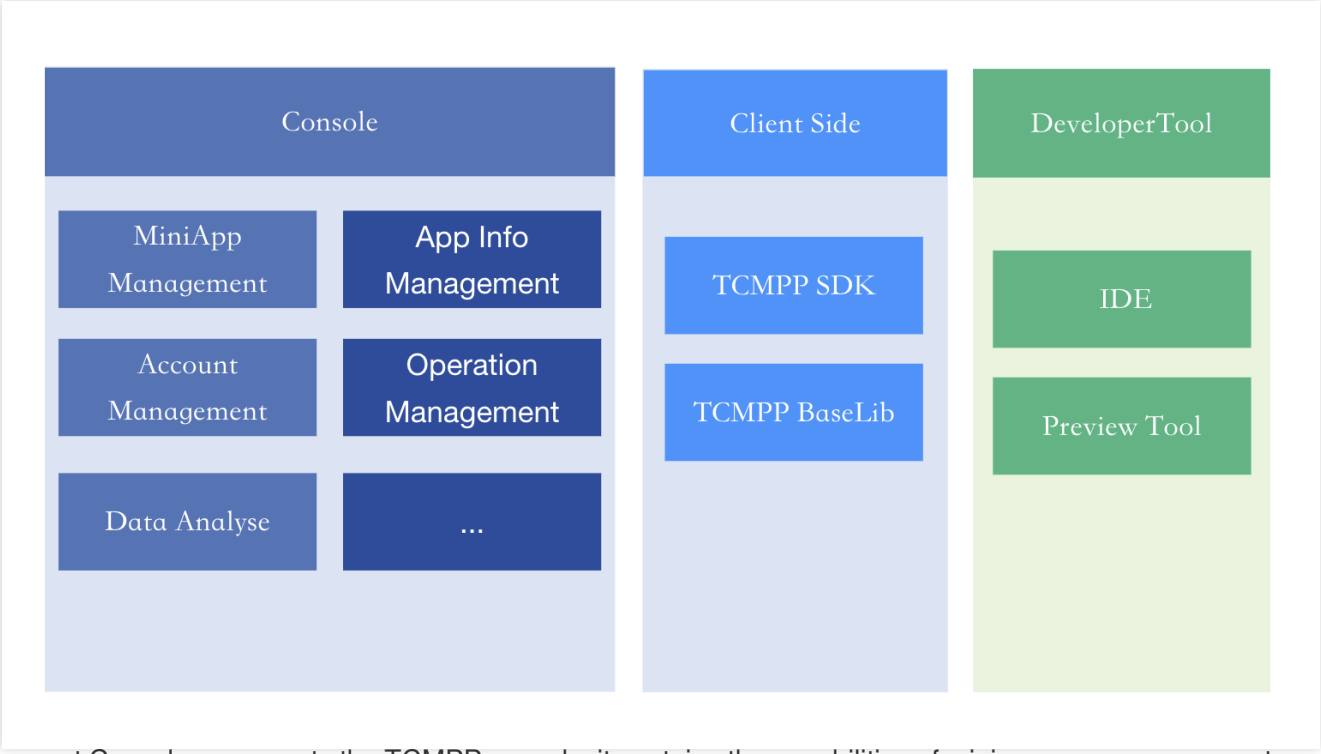
©2013-2022 Tencent Cloud. All rights reserved.

iOS
iOS SDK Description
SDK Introduction

Last updated：2024-07-03 18:00:00

TCMPP Introduction

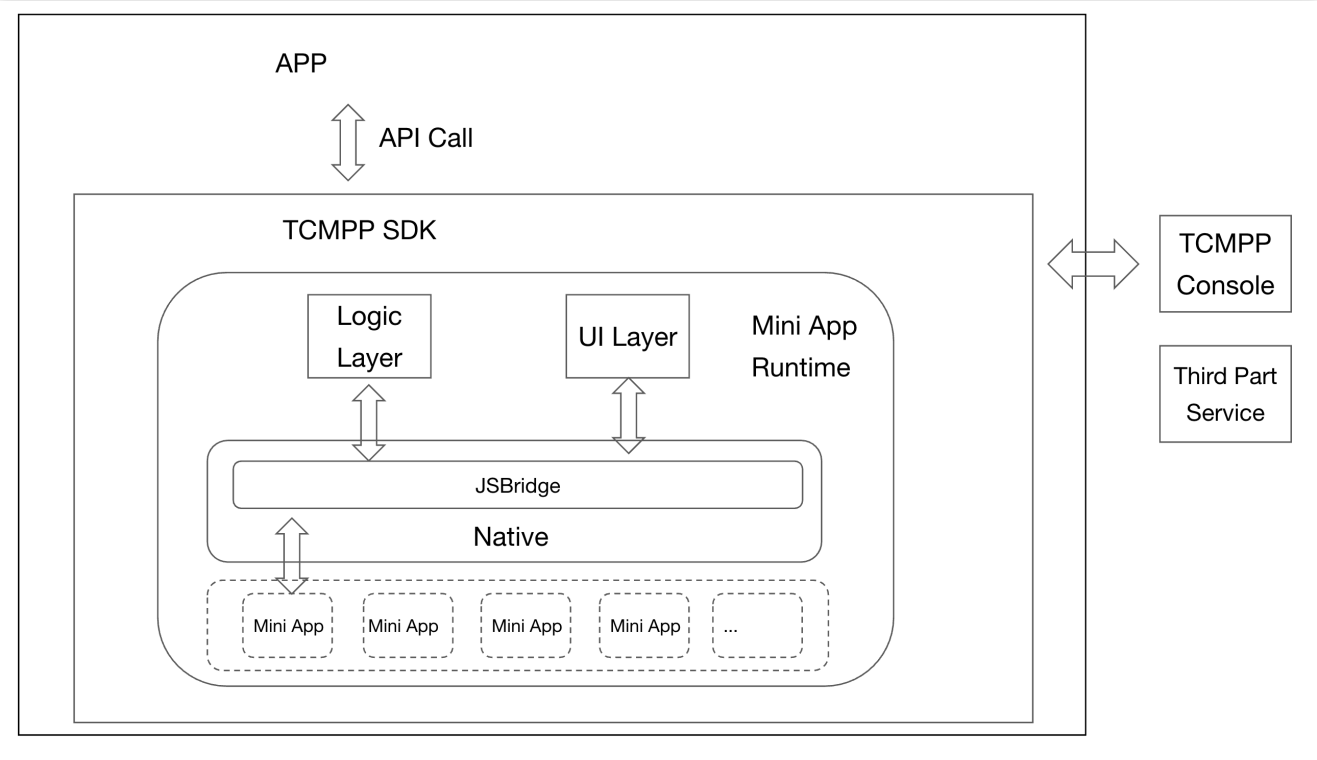
TCMPP consists of three parts: Management Console, Client SDK and mini program development tool (TCMPP IDE).



Management Console: represents the TCMPP console; it contains the capabilities of mini program management, client application management, mini program operation and maintenance management, and so on.
Client: represents the mobile client application that integrates TCMPP mini program SDK; TCMPP mini program SDK provides the runtime environment of mini program for the client application.
Developer tools: mini program developers use mini program IDE for mini program development, debugging and version submission; mini program developers use mini program preview assistant to preview and verify the mini program in the mobile client.

TCMPP Mini Program SDK Working Principle

©2013-2022 Tencent Cloud. All rights reserved.



The mini program SDK provides the runtime environment for the mini program to the client application; the mini program SDK communicates with the TCMPP backend to get the information of the mini program, and loads and runs the mini program in the runtime environment provided by the mini program SDK.

Mini program SDK is divided into core library (i.e. TCMPPSDK) and extension library (i.e. TCMPPExtXXX) for the consideration of reducing package size and system privilege and function control, most of the functions can be realised by integrating only the core library, and there is no need to apply for extra system privilege. The impact of integrating only the core library on the final app installation package is around 4M.

Please refer to iOS Extended component for the description of the Extension Library.

©2013-2022 Tencent Cloud. All rights reserved.

SDK Integration

Last updated：2024-09-25 20:37:44

Integration sample code

You can get the demo address from GitHub

Prerequisites

Environment requirements

iOS >= 9.0

Xcode >= 10.0

Dependent components

tars

MQQComponents

TMFShark

TMFProfile

SSZipArchive

PromiseObjC

MJRefresh

Masonry

SocketRocket

Brotli

CocoaAsyncSocket

Lame

TMFJSBridge

TMFUploader

Integration Methods

TCMPPSDK can be integrated in one of the following two ways:

CocoaPods Integration SDK

Manual Integration SDK

CocoaPods Integration SDK

©2013-2022 Tencent Cloud. All rights reserved.

1. Add TCMPP sources and mini program dependencies to the Podfile file in your project:

```
#TCMPP Pods repository
source 'https://e.coding.net/tcmpp-work/tcmpp/tcmpp-repo.git'

target 'YourTarget' do
     # ——— TCMPP ------------------------------------------------------------- #
     pod 'TCMPPSDK'
     pod 'TCMPPExtScanCode'
     pod 'TCMPPExtMedia'

end
```

where YourTarget is the name of the target where you need to introduce TCMPP to your project.

2. Terminal cd to the directory where the Podfile file is located and run pod install to install the components.

```
$ pod install
```

Note：

If the error Couldn't determine repo type for URL: `'https://e.coding.net/tcmpp-work/tcmpp/tcmpp-repo.git'` : is reported, you need to run the pod repo add specs before executing the pod install.

`https://e.coding.net/tcmpp-work/tcmpp/tcmpp-repo.git`

Manual SDK Integration

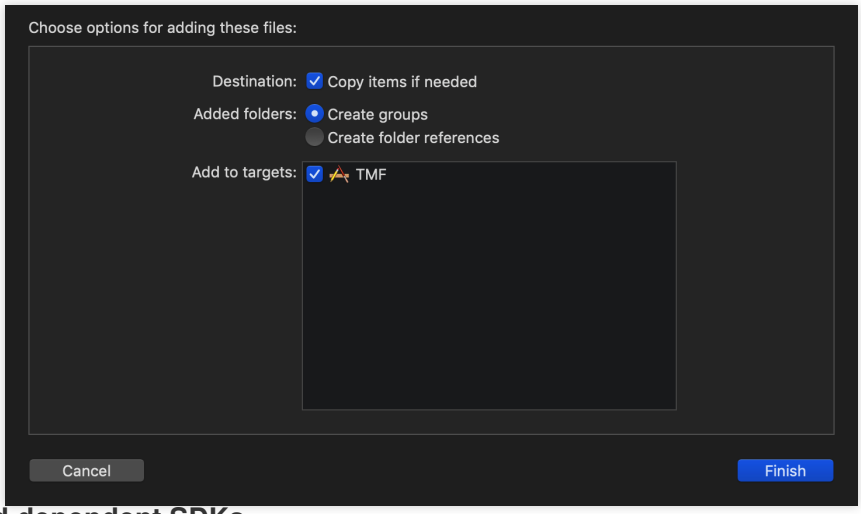
> **Integration Sample Code**
>
> You can download the manual integration demo by clicking

1. **Add SDK**

Add the directory of the TCMPPSDK component to the appropriate location in your project's Xcode Project and select the appropriate target.

You can drag the component directory directly from Finder to Xcode Project for quick addition.

©2013-2022 Tencent Cloud. All rights reserved.



2. **Add dependent SDKs**

Add all the components that TCMPPSDK depends on to the project. For a list of dependent components, see Component Dependencies in Pre-requisites.

3. **Add dependent libraries**

To add TCMPPSDK dependent libraries to the project, open the project settings page in Xcode, select the relevant target, click General, and add them under "Linked Frameworks and Libraries".

4. **The system library dependencies are as follows:**

Foundation.framework

CoreTelephony.framework

CFNetwork.framework

Security.framework

SystemConfiguration.framework

CoreService.framework

CoreFoundation.framework

libz.tdb

libc++.tbd

libc.tbd

libbz2.tbd

libsqlite3.0.tbd

5. **project settings**

After adding TCMPPSDK, you need to set up the related project settings. Open the Project Settings page in Xcode, select the relevant target and make the following settings:

Select Build Settings > Linking > Other Linker Flags and add: -ObjC.

Select Build Settings > Apple Clang - Custom Compiler Flags > Other C Flags, add:

```
-fshort-wchar
-D__FIXWCHART__
```

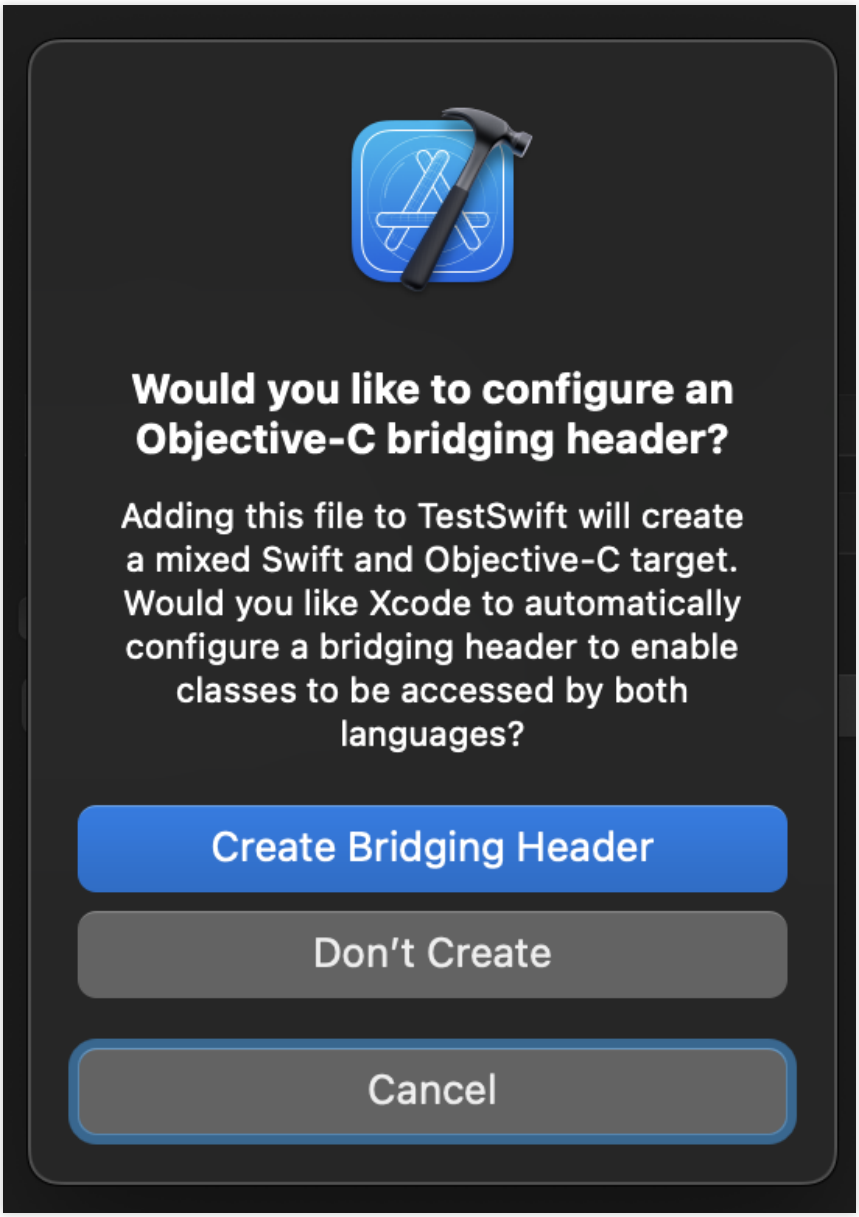
©2013-2022 Tencent Cloud. All rights reserved.

Swift Project Integration SDK

Integration Sample Code

You can download the Swift Project Integration Demo here.

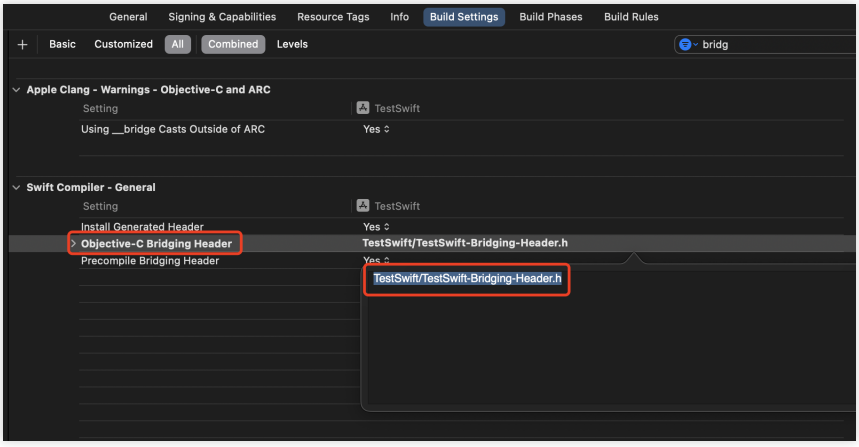
The first time you introduce an OC file, Xcode will prompt you to create a bridge file, click **Create**.



If you clicked cancel, you can manually create a new Header named "{target-name-Bridging-Header.h}", such as "TestSwift-Bridging-Header.h".

In the Build Settings option of the project's targets, find the Objective-c Bridging Header option and set it to the created bridging file.

©2013-2022 Tencent Cloud. All rights reserved.

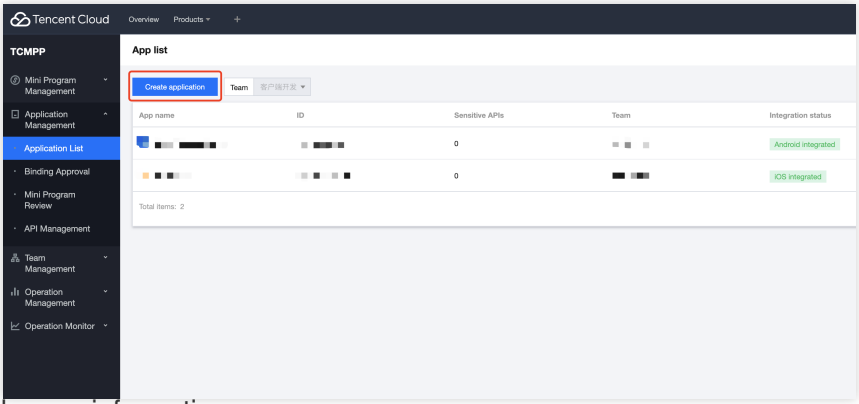


Add the code in this bridging file:

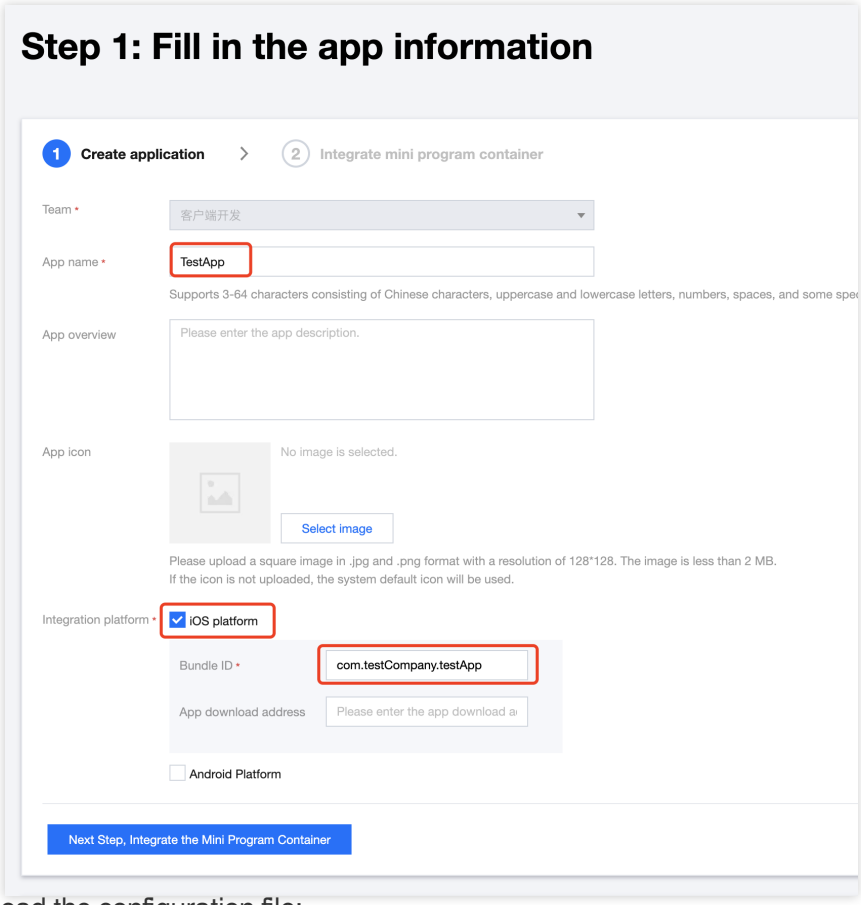
```
#import "TCMPPSDK.h"
```

Getting the configuration file

The initialisation of the mini program SDK relies on the mini program SDK configuration file obtained from the mini program console; before you start integrating the mini program SDK, you need to obtain the mini program SDK configuration file from the mini program console.
After logging into the console, please click Create App:



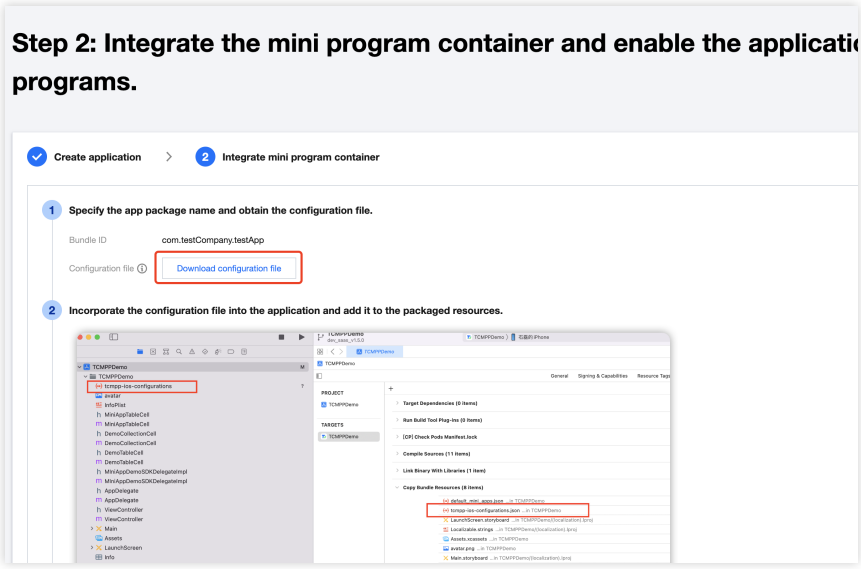
Fill in the app information:



Download the configuration file:

Note：

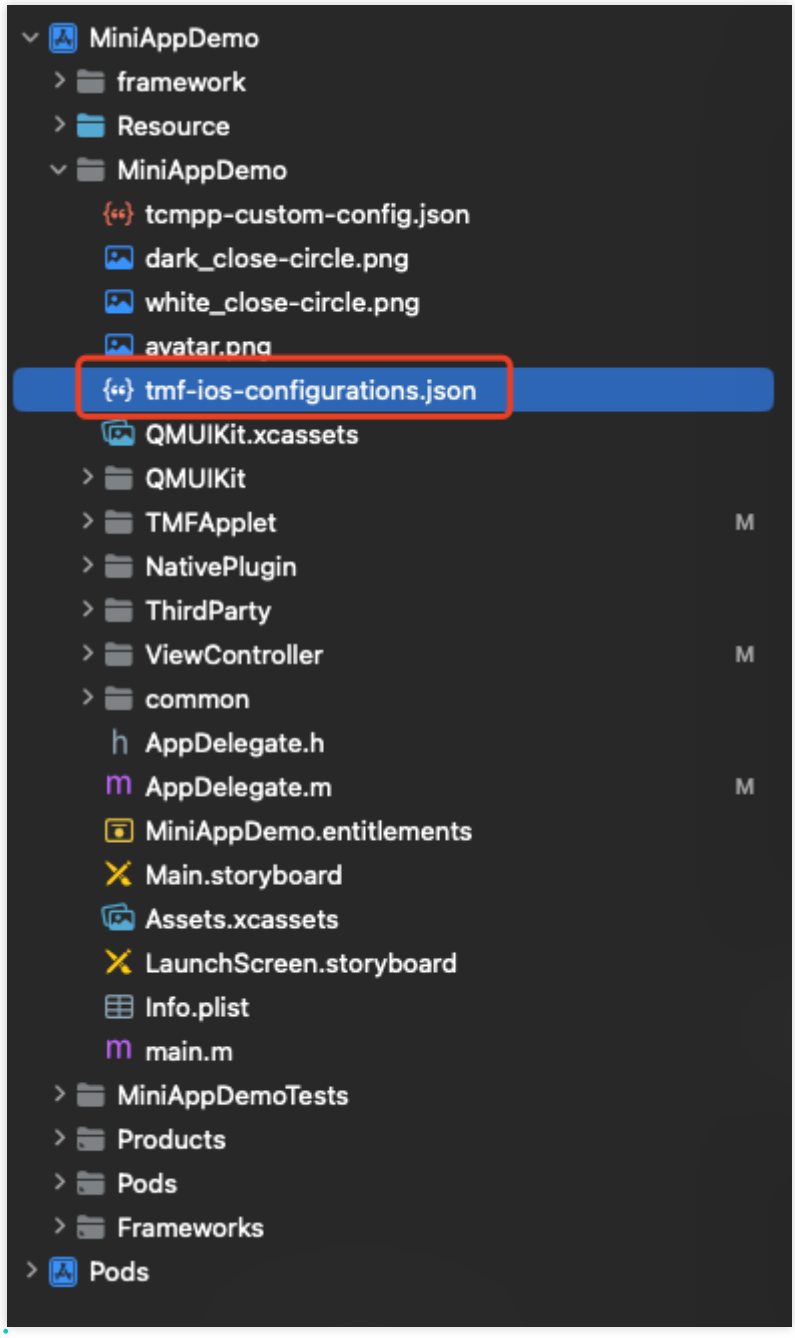
The name of the configuration file downloaded by default is: tcmpp-ios-configurations.json



Adding Configuration Files to the Project

After obtaining the configuration file, you need to add the configuration file to the application source project:

©2013-2022 Tencent Cloud. All rights reserved.



Note：

The bundleId of the iOS project must be consistent with the bundleId in the configuration file, otherwise the mini program SDK cannot pass the package name verification at runtime, resulting in SDK initialisation exceptions.

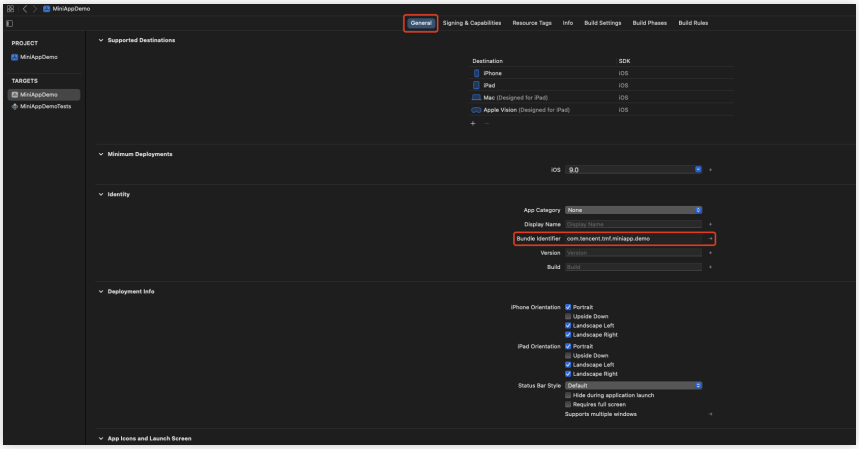
If it is inconsistent, you can't directly modify the bundleId field in the configuration file, you should correct it in the following two ways:

Modify the bundleId of the application in the project to be consistent with the bundleId in the configuration file.

The console modifies the bundleId of the application to be consistent with the bundleId of the application in the project and re-downloads the configuration file.

The packageName field in the configuration file needs to be consistent with the package name of the current project.

©2013-2022 Tencent Cloud. All rights reserved.



Adding Permission Settings

If you only integrate the core library of TCMPP, that is, TCMPPSDK, you need to add the permission application information in the info.plist file.

Among the permissions involved in the core SDK (FinApplet):

Permission	KEY	API of the permission
Write Album	NSPhotoLibraryAddUsageDescription	saveImageToPhotosAlbum、saveVideoToPhotosAlbum
Camera	NSCameraUsageDescription	CameraContext(Camera component)
Microphone	NSMicrophoneUsageDescription	CameraContext(Camera component)

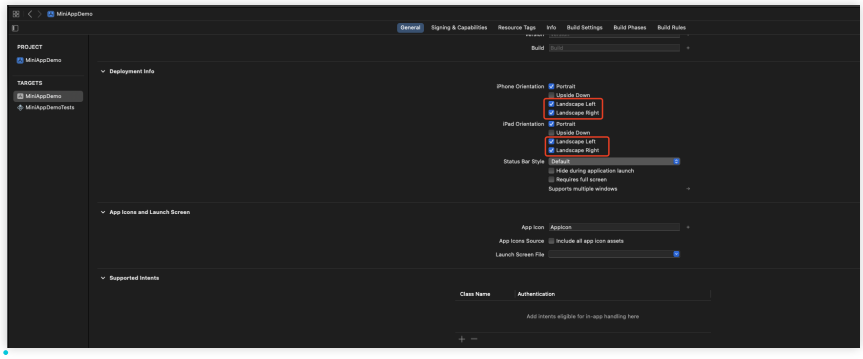
f you need to use the extension SDK, then you need to add the corresponding permission request information in the info.plist file of your project.

You can check the documentation for the iOS Extended component.

©2013-2022 Tencent Cloud. All rights reserved.

Setting the device to support landscape

The loading page and video components in TCMPP SDK support landscape effect, but only if the app project supports landscape.
Please tick the following options in Xcode settings:



Note：

If landscape is not ticked, then the functions in the SDK that involve landscape have no effect.

Introduce the header file

Introduce TMFMiniAppSDKManager.h and TMFAppletConfigManager.h into AppDelegate.

```
//TCMPP
#import <TCMPPSDK/TCMPPSDK.h>
```

Configuration Information Setting

Initialise the TMAServerConfig object according to the configuration file and use TMAServerConfig to initialise the TCMPP mini program engine.
The SDK can support direct engine initialisation, prepare the network link in advance, and update the base library information and configuration information to accelerate the post mini program loading, or it can support initialisation when needed.

Simple Code

```
// Configure the usage environment
NSString *filePath = [[NSBundle mainBundle] pathForResource:@"tcmpp-ios-configurat
if(filePath) {
  TMAServerConfig *config = [[TMAServerConfig alloc] initWithFile:filePath];
  // Initialise directly
  [[TMFMiniAppSDKManager sharedInstance] setConfiguration:config];
```

©2013-2022 Tencent Cloud. All rights reserved.

```
}
```

Other Initialisation Actions

Users can set up open interface implementation instances as needed. If you need to integrate the extension module, initialise the extension interface ready.

```
//Set up the mini program engine proxy class implementation
[TMFMiniAppSDKManager sharedInstance].miniAppSdkDelegate = [MIniAppDemoSDKDelegateI
```

MIniAppDemoSDKDelegateImpl must implement the TMFMiniAppSDKDelegate protocol, you can refer to the MIniAppDemoSDKDelegateImpl file in Customised SDK capabilities and Demo projects.

Open mini program

You can open a mini program by calling the API directly through AppId.

```
[[TMFMiniAppSDKManager sharedInstance] startUpMiniAppWithAppID:appId parentVC:self
   NSLog(@"open applet error:%@",error);
}];
```

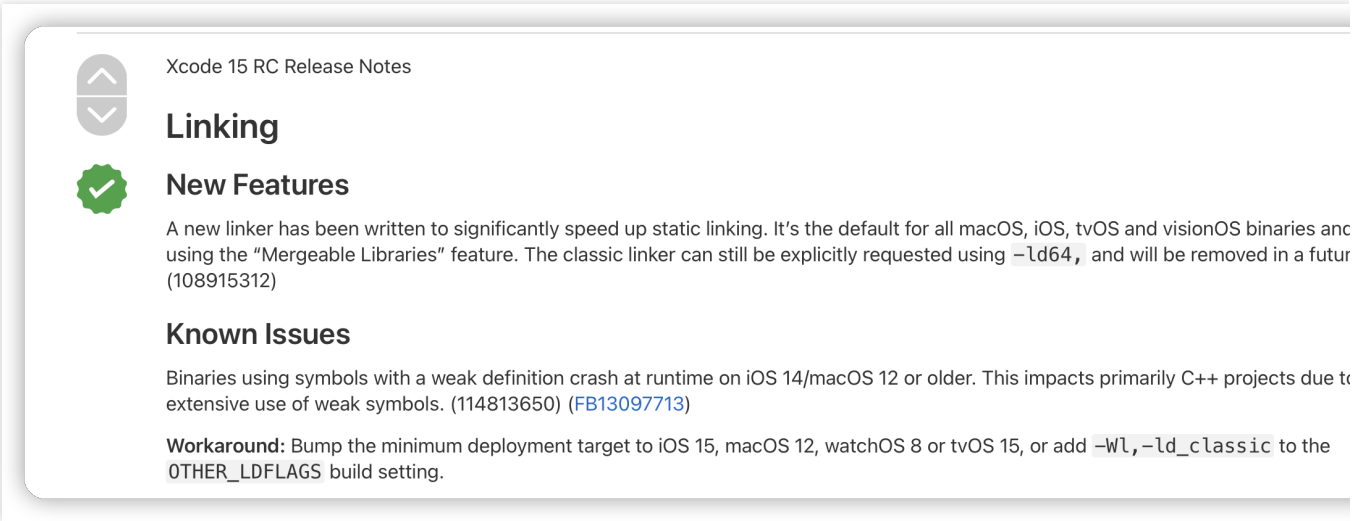
©2013-2022 Tencent Cloud. All rights reserved.

SDK integration FAQs

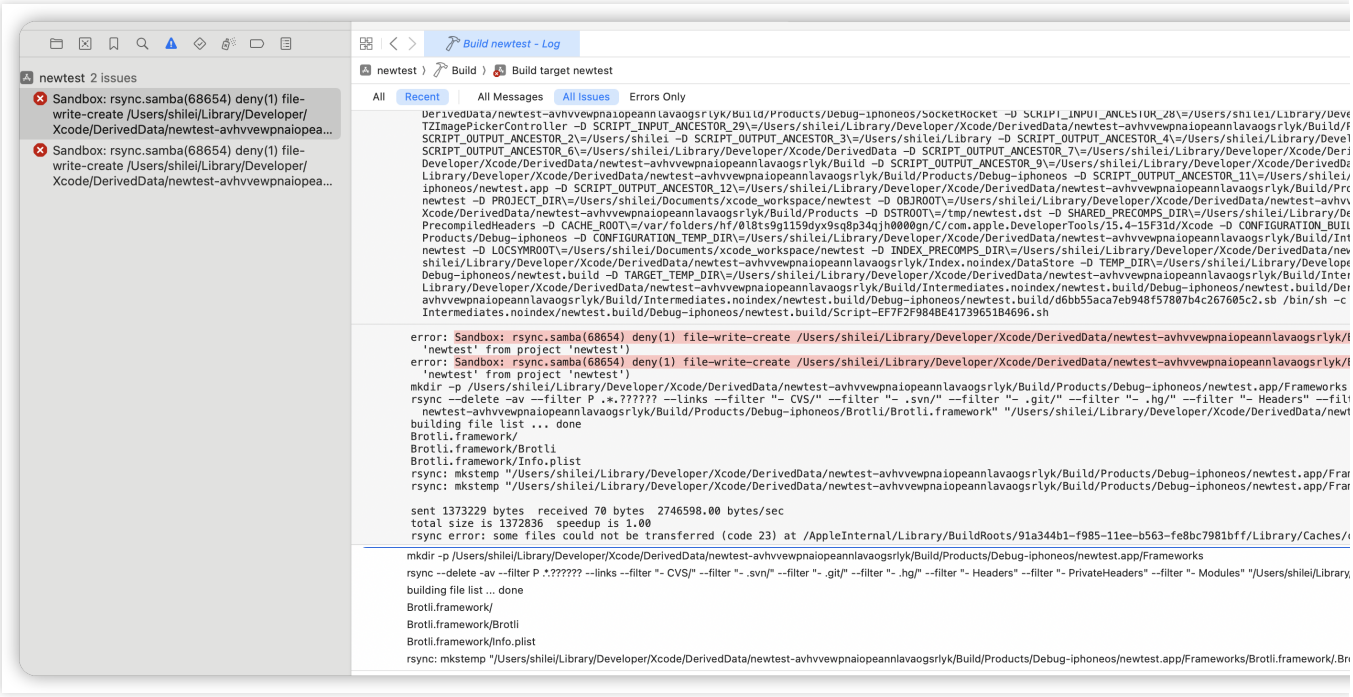
Last updated：2024-07-24 09:57:05

For XCode 15 and earlier versions, the mini program crashes on iOS 12 and later versions

Add `-Wl,-ld_classic` to the project's compilation options.

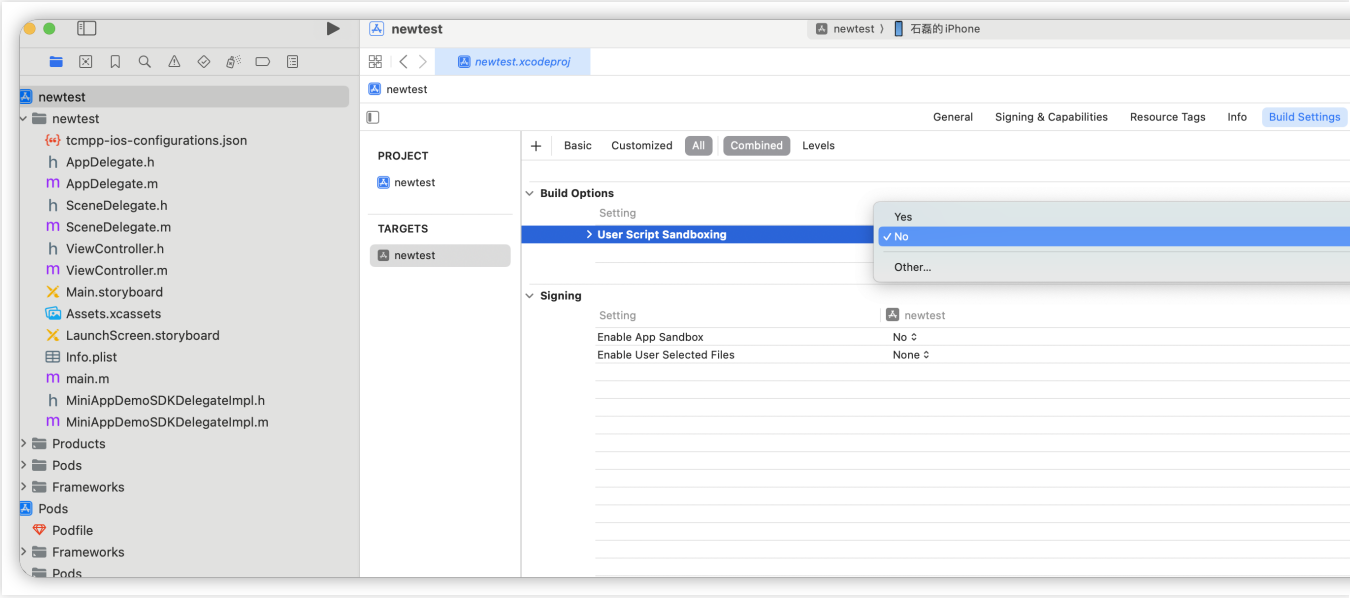


Xcode15 compiles a new project and introduces the pod library with the error Sandbox: rsync.sanba deny(1) file-write-create xxx



Solution: Build Settings search for sandbox, change User Script Sandboxing in Build Options to NO.

©2013-2022 Tencent Cloud. All rights reserved.



Apple M-Series Chip PC Emulator runs abnormally

The current version of the TCMPP SDK needs to be run through Rosetta for running on M-Series chip PC emulators.

Before xcode14:

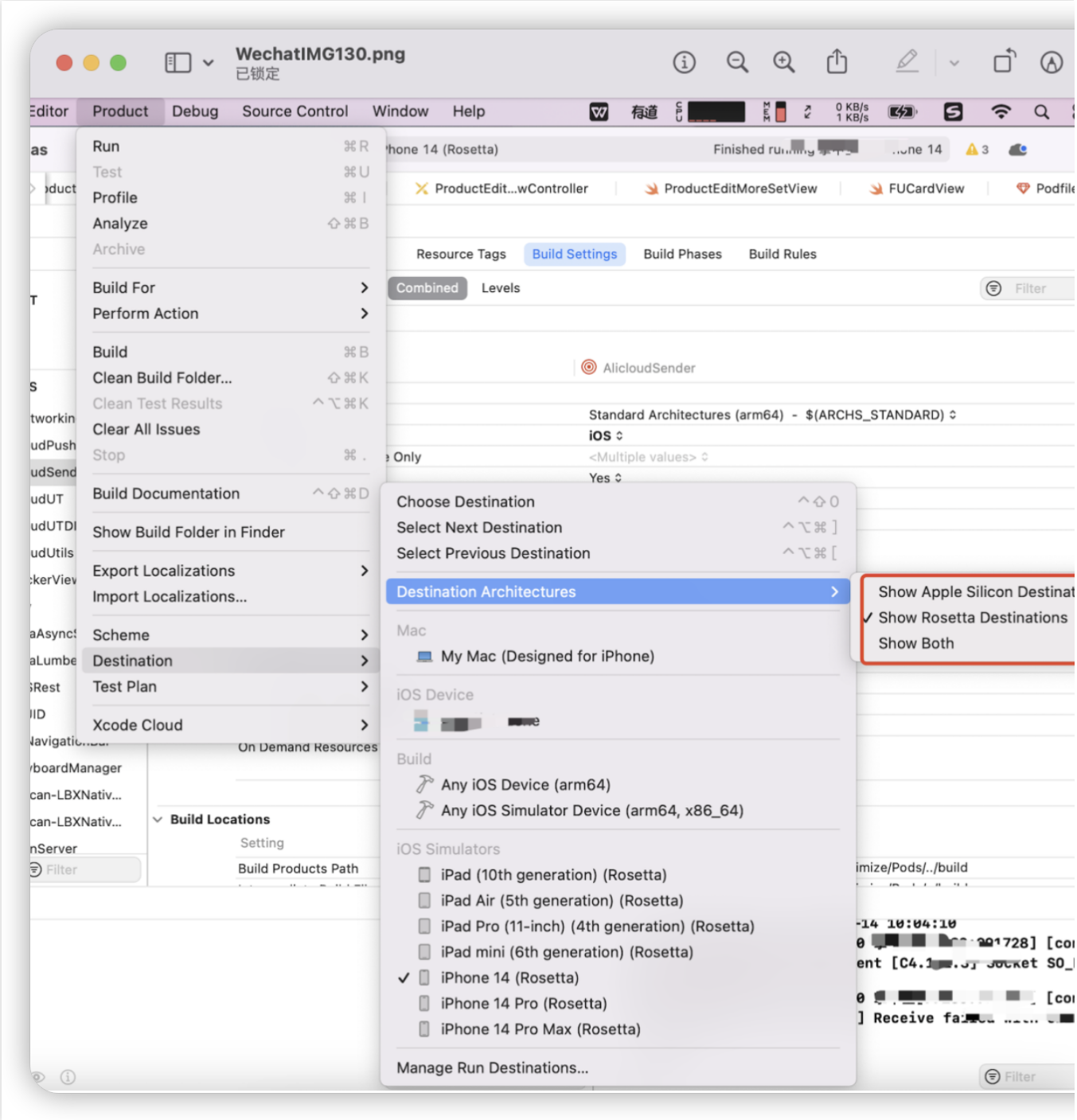
We can right click xcode->show profile->check open with Rosetta, so it is running on the emulator.

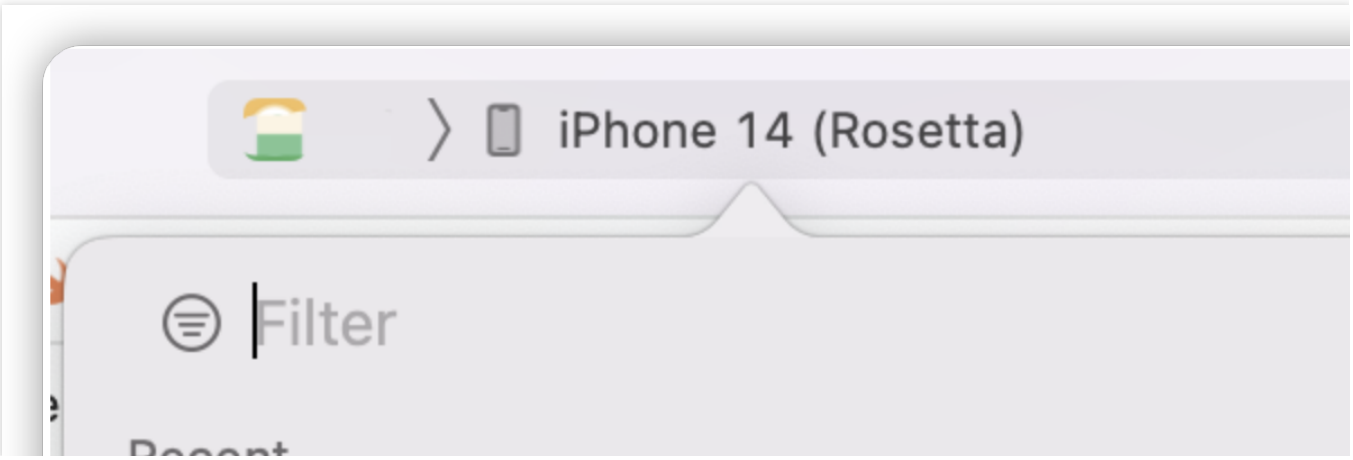
After xcode14:

xcode Open Project > Product > Destination > Destination Architectures and you can choose which mode of simulator to open it with.

We will choose the emulator ending with (Rosetta).

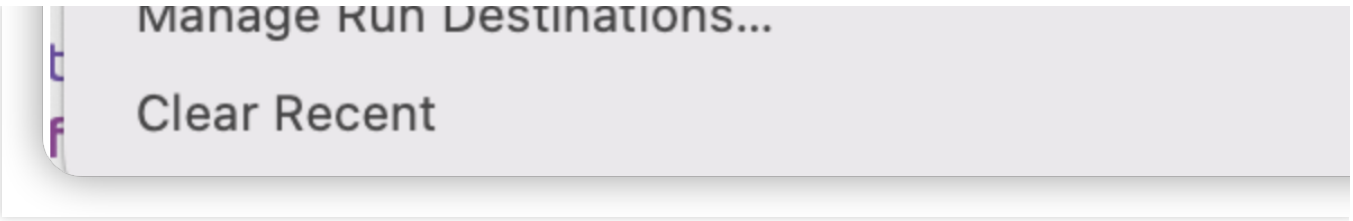
©2013-2022 Tencent Cloud. All rights reserved.





©2013-2022 Tencent Cloud. All rights reserved.

©2013-2022 Tencent Cloud. All rights reserved.



©2013-2022 Tencent Cloud. All rights reserved.

iOS API

Mini program management APIs

Searching mini programs

Last updated：2024-07-03 18:00:50

The mini program SDK provides an API for searching online mini programs by keywords and categories.

Note：

firstType and secondType specify the primary and secondary categories for the mini program search.

The completion parameter is used to retrieve the search results.

```
// Search mini programs
// Search mini program
// @param name Search by name keyword - search name keyword
// @param completion Search results - search results
- (void)searchAppletsWithName:(NSString *)name
                   completion:(void (^)(NSArray<TMFAppletSearchInfo *> * _Nullab

// Search mini programs
// Search mini program
// @param name Search by name keyword - search name keyword
// @param firstType Primary category name - first level classification name
// @param secondType Secondary category name - second level classification name
// @param completion Search results - search results
- (void)searchAppletsWithName:(NSString * _Nullable )name
                  firstType:(NSString * _Nullable)firstType
                  secondType:(NSString * _Nullable)secondType
                  completion:(void (^)(NSArray<TMFAppletSearchInfo *> * _Nullable,
```

Search by keyword

Sample code:

```
[[TMFMiniAppSDKManager sharedInstance] searchAppletsWithName:searchString completio
    if (error) {
      // Search failed, or the list is empty
    } else {
      //Search successful, list not empty
    }
}];
```

©2013-2022 Tencent Cloud. All rights reserved.

Search by single category

Sample code:

```
[[TMFMiniAppSDKManager sharedInstance] searchAppletsWithName:nil firstType:@"firstT
    if (error) {
      // Search failed, or the list is empty
    } else {
      //Search successful, list not empty
    }
}];
```

Bicategory search

Note：

The result of a bicategory search is the intersection of two categories.

Sample code:

```
[[TMFMiniAppSDKManager sharedInstance] searchAppletsWithName:nil firstType:@"firstT
    if (error) {
      // Search failed, or the list is empty
    } else {
      //Search successful, list not empty
    }
}];
```

©2013-2022 Tencent Cloud. All rights reserved.

Opening mini programs

Last updated：2024-10-18 11:19:13

Opening mini programs

When opening a mini program, the system will check whether there is a locally cached one. If not, it will automatically download the mini program from the remote server and then open it. If a cached version is available, the local mini program will be opened first while the system checks in the background for any new versions on the server.

Notes:

If a new version is available, it will be downloaded, and the next time you open the mini program, the updated version will be used.

```
// Open a mini app through the mini program ID

// @param appID - Mini program ID
// @param verType - The version type of the mini program to open
// @param scene - Scene value
// @param firstPage - The first page of the mini program to open
// @param paramsStr - The parameter used to open the mini program
// @param parentVC - The first view controller to call
// @param completion - Error callback
- (void)startUpMiniAppWithAppID:(NSString *)appID
                        verType:(TMAVersionType)verType
                          scene:(TMAEntryScene)scene
                      firstPage:(NSString * _Nullable)firstPage
                      paramsStr:(NSString * _Nullable)paramsStr
                       parentVC:(UIViewController *)parentVC
                     completion:(void (^)(NSError * _Nullable))completion;
```

Parameters supported by options:

Name	Required	Type	Description
appID	YES	NSString	ID of the mini program to open
verType	YES	TMAVersionType	Type of the mini program to open
scene	YES	TMAEntryScene	The scene value for opening the mini program
firstPage	NO	NSString	The first page of the mini program to open
paramsStr	NO	NSString	The parameter used to open the mini

©2013-2022 Tencent Cloud. All rights reserved.

			program
parentVC	YES	UIViewController	The first view controller to call
completion	YES	block	Error callback

Opening a mini program by mini program ID (appId)

To open the released version of a mini program :

```
[[TMFMiniAppSDKManager sharedInstance] startUpMiniAppWithAppID:appId parentVC:self
   NSLog(@"open applet error:%@",error);
}];
```

Notes:

The appId field is the ID of the mini program, which can be obtained from the mini program developer or via the mini program search API.

To open the Preview or development version of a mini program:

```
[[TMFMiniAppSDKManager sharedInstance] startUpMiniAppWithAppID:appId verType:verTyp
   NSLog(@"open applet error:%@",error);
}];
```

Notes:

The appVerType should match the version of the mini program. The value of the appVerType can be obtained from the TMFAppletSearchInfo object instance returned by the API (getRecentList).

The value of appVerType for the Preview of the mini program should be TMAVersionPreview.

The value of appVerType for the development version of the mini program should be TMAVersionDevelop.

Calling QRCode API to open a mini program

The mini program SDK provides a capability to open mini programs by calling the QRCode API. You need to integrate with the extension library TCMPPExtScanCode before calling the API.

```
// Open the mini program through QRCode
// @param parentVC - The first view controller to call
// @param completion - Error callback
- (void)startUpMiniAppWithQRCodeWithParentVC:(UIViewController *)parentVC
                                  completion:(void (^)(NSError * _Nullable))complet
```

Opening a mini program through QR code

The mini program SDK provides an API to open mini programs based on QR code scanning in the console. Before using the scanning capability provided by the mini program SDK, you need to integrate the scanning extension capability. For details about the integration, refer to the scanning capability documentation.

©2013-2022 Tencent Cloud. All rights reserved.

After integrating the scanning capability, you can **start the QR code scanning and open the mini program as follows:**

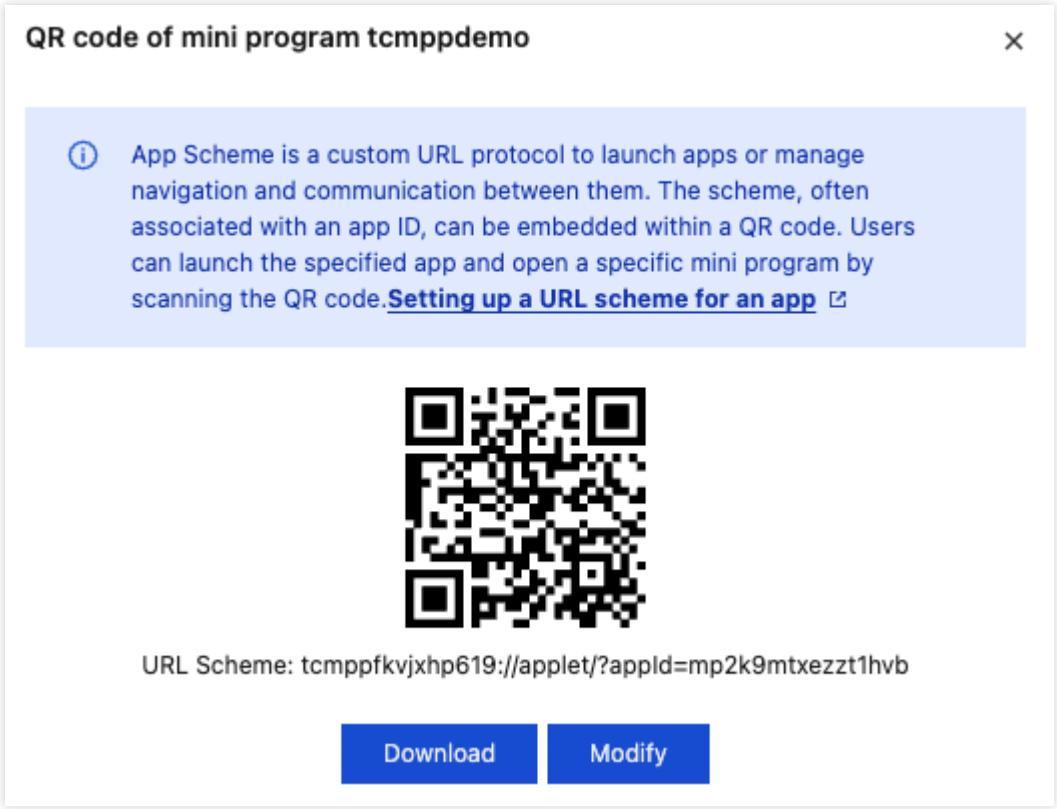
```
/// Open a mini program via the QR code
/// @param qrData - QR code content
/// @param parentVC - The first view controller to call
/// @param completion - Error callback
- (void)startUpMiniAppWithQrData:(NSString *)qrData
                       parentVC:(UIViewController *)parentVC
                      completion:(void (^)(NSError * _Nullable error))completion;
```

Opening the released version of a mini program through QR code

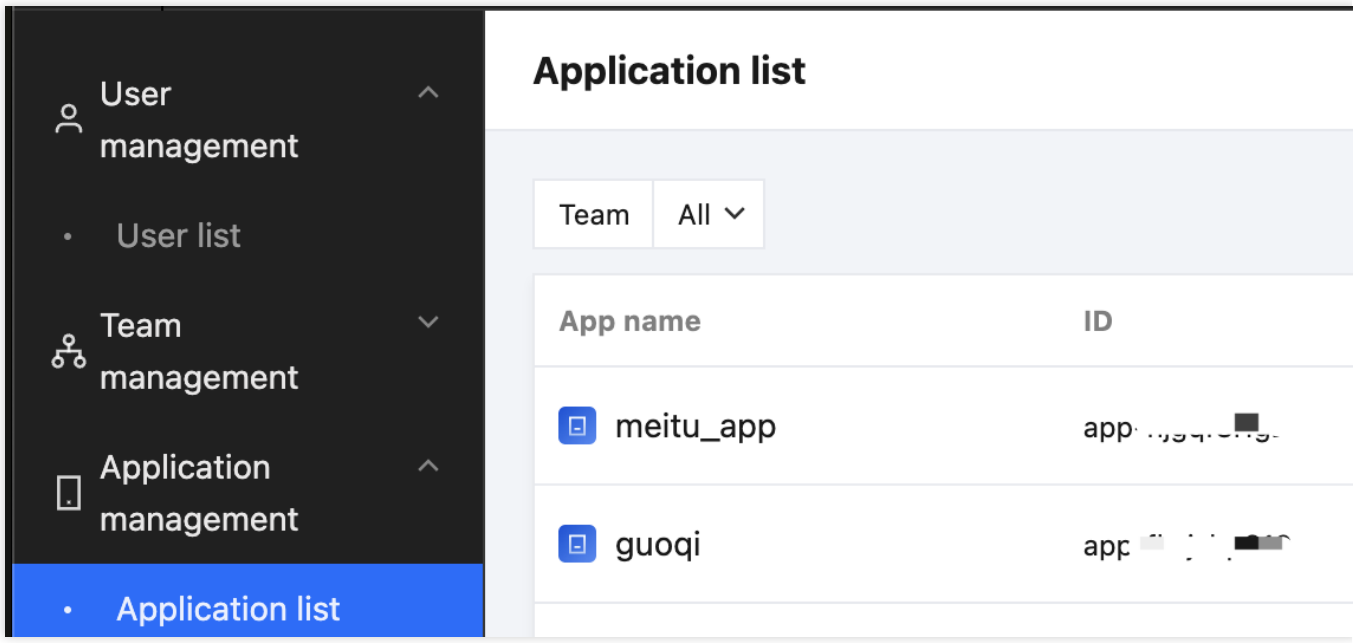
Starting from version 2.0.9, the QR code for the released version of a mini program can be generated and modified in the console. Once the app scheme is configured, you can open the mini program by scanning the QR code with the system camera.

Go to **Application management** - **Mini program approval** - **Submitted** in the console, click **Download QR code** in the **Operation** column, and the following pop-up appears:

©2013-2022 Tencent Cloud. All rights reserved.

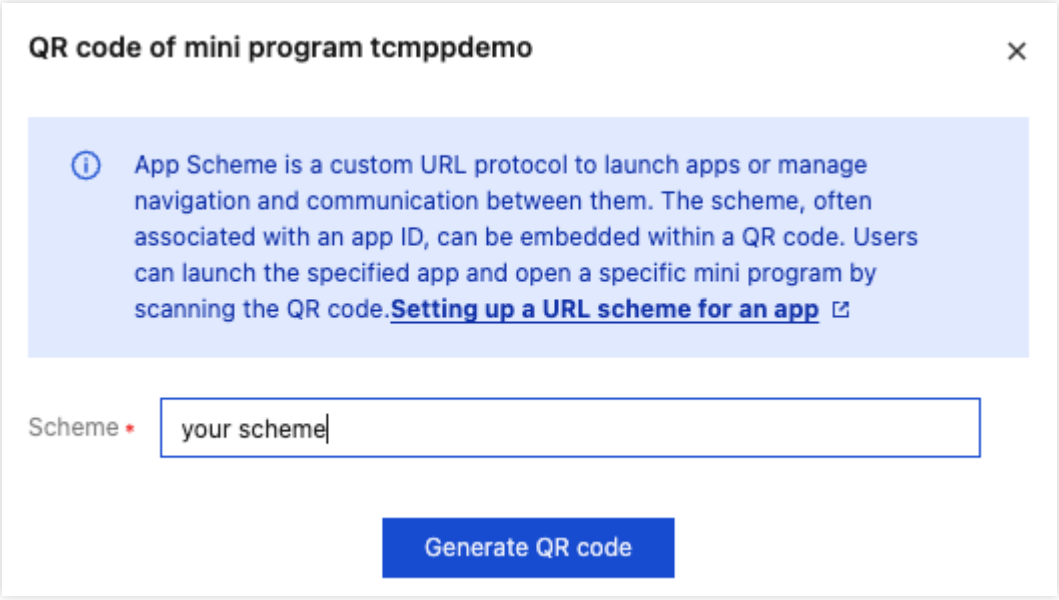


The default scheme is tcmpp plus last few characters of the appid, which can be found through **Application management** - **Application list**.



For example, if the appid is app-ylk2jebx9q, then the schema is tcmppylk2jebx9q. You can click **Modify** to modify the scheme.

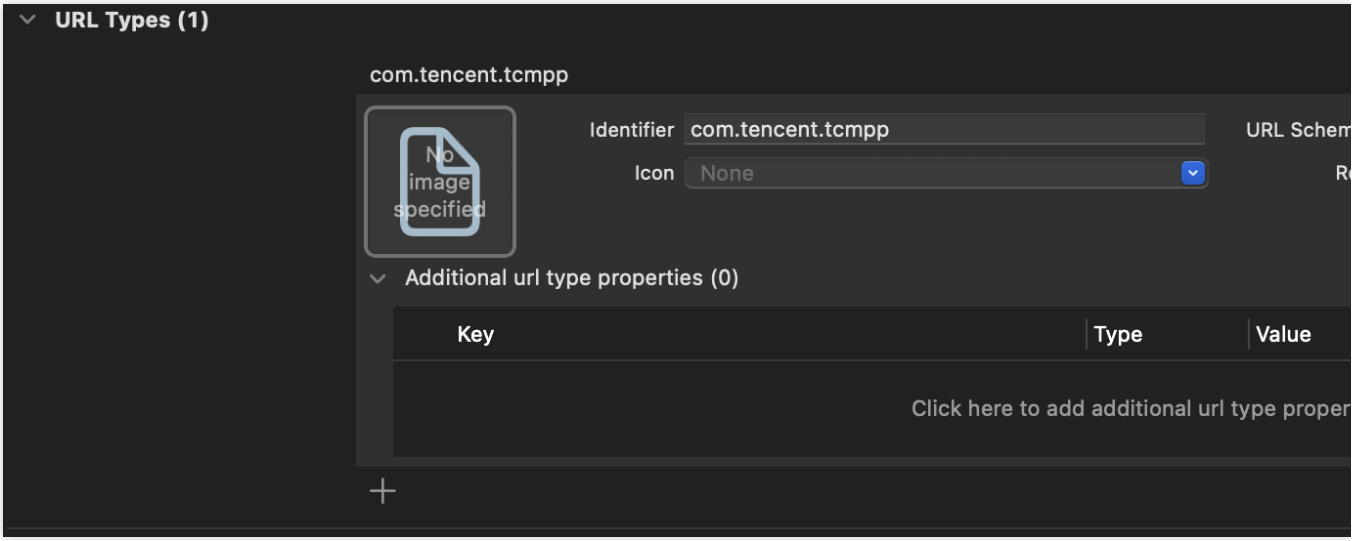
©2013-2022 Tencent Cloud. All rights reserved.



The scheme name is returned through the proxy API **getAppScheme** in the client. See the sample code below:

```
- (NSString *)getAppScheme{
    return @"tcmpp";
}
```

Configure the same scheme in the URL Types of the main project's info.plist file:



Implement the handleOpenUrl method in the openUrl method. See the sample code below:

```
- (BOOL)application:(UIApplication *)app openURL:(NSURL *)url options:(NSDictionary
```

©2013-2022 Tencent Cloud. All rights reserved.

```
    if ([[TMFMiniAppSDKManager sharedInstance] handleOpenUrl:url]){
        return YES;
    }
    return YES;
}
```

Once completing the above steps, you can scan the QR code in the console with the system camera to open the released version of the mini program.

©2013-2022 Tencent Cloud. All rights reserved.

Closing mini programs

Last updated：2024-07-03 18:01:09

Closing mini programs

Note：

Closing a mini program means only closing its page, switching the mini program to the background. It remains in memory, allowing for a quick relaunch with minimal delay.

Note：

1. A mini program remains active in memory for 10 minutes after closing.
2. Up to 3 mini programs can remain active simultaneously.

1. Closing a specific mini program

```
/// Close a specific mini program
/// @param appID The appID of the mini program to close
- (void)closeMiniAppWithAppID:(NSString *)appID;
```

2. Closing current mini program

```
/// Close the currently running mini program
- (void)closeCurrentApplet;
```

3. Closing all running mini programs

```
/// Close all running mini programs in memory
- (void)closeAllApplications;
```

Terminating mini programs

Note：

Terminating a mini program means clearing it from memory. The next time it is opened, it will need to be reloaded into memory, which takes more time.

1. Terminating the specified mini program

```
/// Terminate the specified mini program object.
```

©2013-2022 Tencent Cloud. All rights reserved.

```
/// @param appID The appID to terminate, all versions are deleted by default.
- (void)terminateMiniAppWithAppID:(NSString *)appID;
```

2. Terminating the current mini program

```
/// Terminate the currently running mini program object.
- (void)terminateCurrentApplet.
```

3. Terminate all mini programs.

```
/// Terminate all mini programs on the stack and alive
- (void)terminateAllApplications.
```

©2013-2022 Tencent Cloud. All rights reserved.

Deleting mini programs

Last updated：2024-07-03 18:01:44

Note：

As the mini program runs, it caches the mini program package and mini program information locally so that it can be opened more quickly next time. If you want to delete all the information of a mini program, you can call the following API to delete a mini program or delete all mini programs.

Delete all mini program caches

```
/// Remove all caches related to mini programs including resource packs, base libra
- (void)clearMiniAppCache;
```

Delete the specified mini program cache

```
/// Delete the local cache mini program.
/// @param appID appID to be deleted, defaults to all versions.
- (void)clearCacheWithAppID:(NSString *)appID;
```

Deletes the specified version and type of mini program

```
// Delete local cache mini programs
// Delete the local cache mini program.
// @param appID appID to delete, defaults to all versions - appID to delete, defaul
// @param verType the type of local cache applet to delete - the version type of th
- (void)clearCacheWithAppID:(NSString *)appID verType:(TMAVersionType)verType;
```

Sample code:

```
[[TMFMiniAppSDKManager sharedInstance] clearCacheWithAppID:appId verType:verType];
```

©2013-2022 Tencent Cloud. All rights reserved.

Getting recently accessed mini program list and information

Last updated：2024-07-03 18:02:27

Getting the information of all mini programs opened recently

```
// Get the information of all mini programs opened recently
///@return Mini program array<TMFMiniAppInfo>
- (NSArray *)loadAppletsFromCache;
```

Getting the information of a running mini program

```
/// Get the object of the currently running mini program
/// @return TMFAppletInfo Mini program information
- (TMFMiniAppInfo *)currentApplet;
```

©2013-2022 Tencent Cloud. All rights reserved.

Pre-downloading mini programs

Last updated：2024-07-03 18:02:36

To reduce the waiting time when opening a mini program and optimize user experience, the following API is provided to pre-download mini program packages.

```
///Pre-prepare mini program information
///@param appIds  Information of mini program to prepare
///@param isDownload Whether to download directly
///@param complete Callback for batch updating mini programs
- (void)preloadMiniApps:(NSArray *)appIds isDownload:(BOOL)isDownload complete:(voi
```

©2013-2022 Tencent Cloud. All rights reserved.

Mini program file management

Last updated：2024-07-03 18:02:53

Converting wxfile path to absolute path

In some scenarios, you may need to convert a mini program file path to an absolute path to access the file data. For example, when using the mini program's forwarding feature, the returned image path is a mini program file path. You can convert it to an absolute path to access the image data and then initiate third-party sharing. Similarly, in custom APIs, you can pass the mini program file path as a parameter, and the host app can convert it to an absolute path to access the file data.

Sample code:

```
TMAFileManager *fm = [[TMFMiniAppSDKManager sharedInstance] getFileManagerWithAppID
NSString *filePath = [fm translateWxfilePathToAbsolutePath:wxPath];
```

Converting absolute path to wxfile path

The SDK supports converting file paths created in the mini program's cache directory to mini program file paths for internal use.

Sample code:

```
TMAFileManager *fm = [[TMFMiniAppSDKManager sharedInstance] getFileManagerWithAppID
NSString *tmpPath = [fm createMediaTmpPathWithFileName:@"a.pdf" type:TMATmpPathType

NSString *wxPath = [fm translateAnyPathToWxfilePath:tmpPath];
```

Creating files in the mini program temporary directory

The SDK supports creating files in the mini program's cache directory natively.

Sample code:

```
TMAFileManager *fm = [[TMFMiniAppSDKManager sharedInstance] getFileManagerWithAppID
NSString *tmpPath = [fm createMediaTmpPathWithFileName:@"a.pdf" type:TMATmpPathType
```

©2013-2022 Tencent Cloud. All rights reserved.

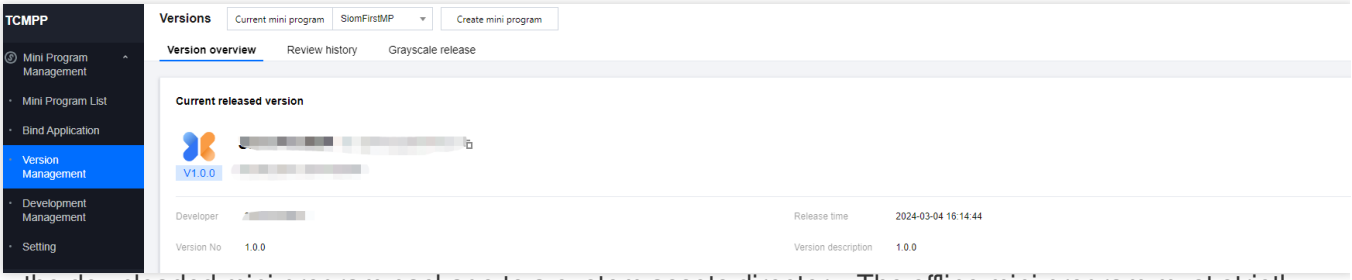
Preloading offline mini programs

Last updated：2024-07-03 18:03:03

Offline mini programs are embedded within the host app. You need to download the mini program package from the console, import it into the host app project, and package its together with the app. During app usage, users can open an embedded mini program without needing to download it from the backend, allowing it to run even without an internet connection.

Preloading process

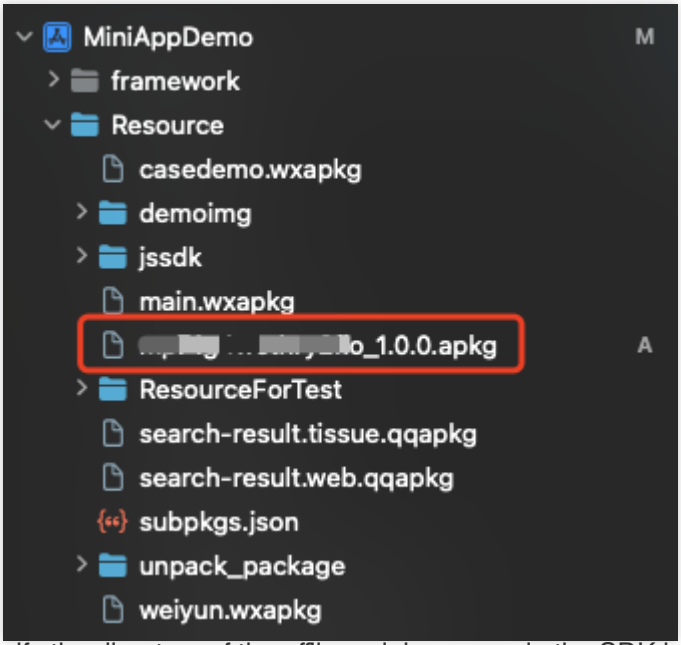
1. Download the required mini program from the console.



2. Copy the downloaded mini program package to a custom assets directory. The offline mini program must strictly follow the naming convention and cannot be arbitrarily modified.

Note:

Naming convention for mini programs: {miniAppId}_{miniAppVersion}.apkg



3. Specify the directory of the offline mini program in the SDK initialization configuration.

```
[[TMFMiniAppSDKManager sharedInstance] setOffLineApkgsPath:[[[NSBundle mainBundle]
```

©2013-2022 Tencent Cloud. All rights reserved.

Notes

Offline mini programs must go through the mini program binding and releasing process. Only mini programs in the online state can download offline packages;
Offline mini programs adhere to version management logic such as new version releases, version rollbacks, and removing. If the online version differs from the preloaded version, the client will fetch the online version;
If a mini program is removed, the corresponding offline mini program preloaded in the app will also become unusable.

©2013-2022 Tencent Cloud. All rights reserved.

Customize Mini Program Capabilities
Custom mini program APIs

Last updated：2024-09-25 20:38:44

The TCMPP SDK engine provides an extension mechanism that allows host apps to customise APIs for mini programs to call.

Implementation steps

1. Customise a class and import the TMAExternalJSPlugin.

```
#import <TCMPPSDK/TCMPPSDK.h>

@interface NativePluginTest : NSObject

@end
```

2. Declare TMA_REGISTER_EXTENAL_JSPLUGIN and add a custom API via TMAExternalJSAPI_IMP().

reference case

```
#import "NativePluginTest.h"
#import "TMAExternalJSPlugin.h"
#import "TMFMiniAppInfo.h" #import "TMFMiniAppInfo.h" #import "TMFMiniAppInfo.h".

Implementing NativePluginTest

TMA_register_extenal_jsplugin; //Custom sync api.

// custom sync api
TMAExternalJSAPI_IMP(testSync) {
    TMFMiniAppInfo *appInfo = context.tmfAppInfo; NSDictionary *data = context.tmfA
    NSDictionary *data = params[@"data"];

    NSLog(@"************ invokeNativePlugin testSync,appId:%@,data is %@",appInfo.a

    TMAExternalJSPluginResult *pluginResult = [TMAExternalJSPluginResult new]; [TMA
    pluginResult.result = @{}; return
    return pluginResult;
}

TMAExternalJSAPI_IMP(test) {
    TMFMiniAppInfo *appInfo = context.tmfAppInfo; TMFMiniAppInfo *appInfo = context
    NSDictionary *data = params[@"data"];
```

©2013-2022 Tencent Cloud. All rights reserved.

```
    NSLog(@"************ invokeNativePlugin test,appId:%@,data is %@",appInfo.appId

    // asynchronous processing, return the result to the mini program in an async c
    //{
    // TMAExternalJSPluginResult *pluginResult = [TMAExternalJSPluginResult new]; /
    // pluginResult.result = @{@"result": result.data}; // [context doCallback
    // [context doCallback:pluginResult].
    // }

    return nil;
}

@end
```

It can be used like this in a mini program.

```
// Asynchronous api calls
var opts = {
   api_name: 'test',
   success: function(res) {},
   failure: function(res) {},
   completion: function(res) {},
   data: { // Input
      Name : 'kka',
      age : 22
   }
 }
 wx.invokeNativePlugin(opts); // Synchronise api calls.

// Synchronise api calls
var opts = {
   api_name: 'testSync',
   sync:true
}
var rst = wx.invokeNativePlugin(opts); var rst = wx.
```

Advanced use

custom api supports configuration in the way of terminal app configuration file, which is invoked in the mini program by calling wx.api directly.

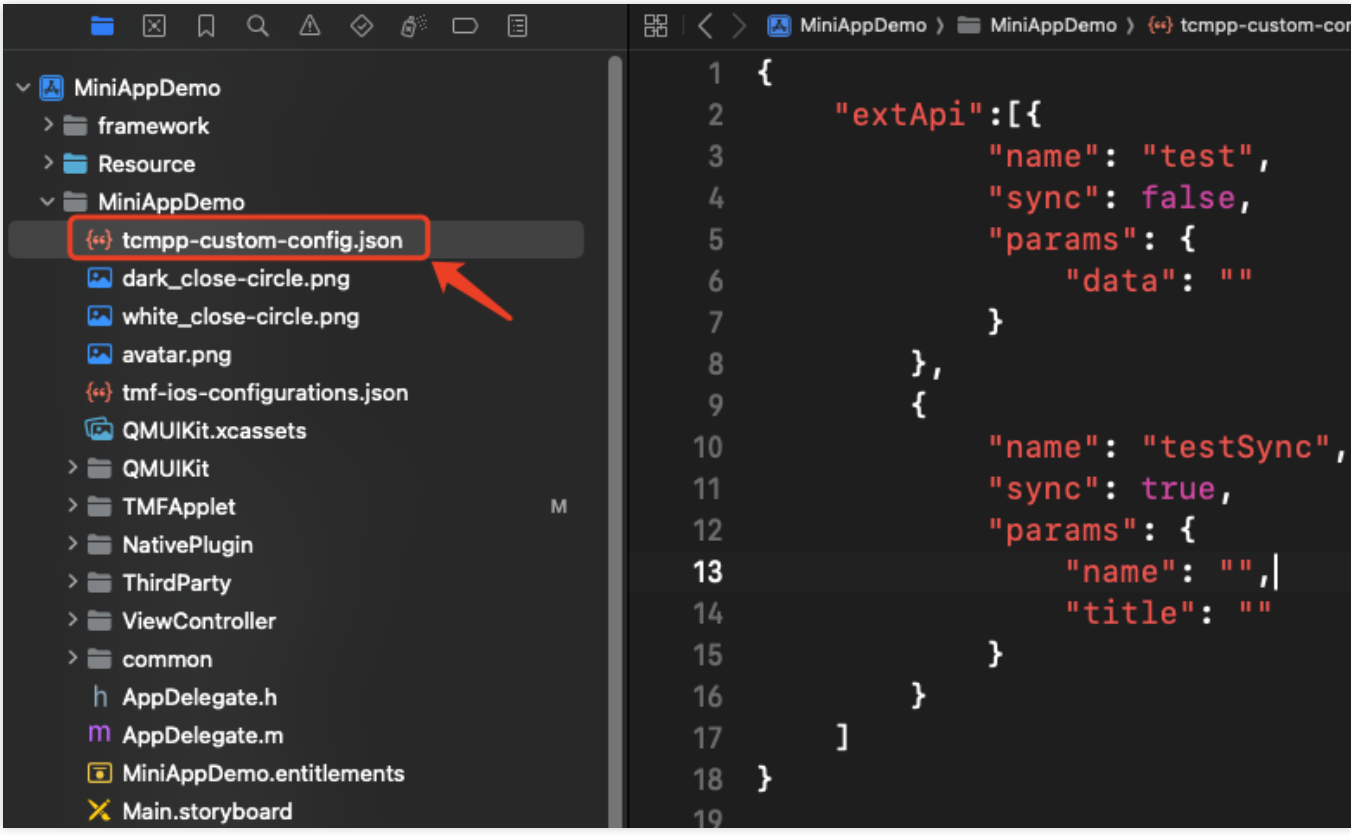
1. Unify the configuration file implemented in the app in tcmpp-custom-config.json with the following reference:

```
{
    "extApi":[{
            "name": "test",
```

©2013-2022 Tencent Cloud. All rights reserved.

```
            "sync": false,
            "params": {
                "data": ""
            }
        },
        {
            "name": "testSync",
            "sync": true,
            "params": {
                "name": "",
                "title": ""
            }
        }
    ]
}
```

2. Put tcmpp-custom-config.json into the iOS project:



3. Call it directly in the mini program with wx.test().

```
//Asynchronous API calls
var opts = {
```

©2013-2022 Tencent Cloud. All rights reserved.

```
    success: function(res) {},
    fail: function(res) {},
    complete: function(res) {},
    data: {
      name : 'kka',
      age : 22
    }
}
wx.test(opts);

//Synchronise api calls
var rst = testSync(opts);
```

©2013-2022 Tencent Cloud. All rights reserved.

Customize mini program view components

Last updated：2024-07-03 18:03:57

Using customized native components

TCMPP supports customized UI components. The customized component on the mini program side is <external-element>. To use customized UI components, follow these steps:

1. Create a new class that inherits from UIView after integrating the SDK, e.g., QMATestView, and import the TMAExternalJSPlugin.h file. Ensure QMATestView conforms to the TMAExternalElementView protocol.
2. Register the QMATestView class as maTestView using the TMARegisterExternalElement method.
3. Implement the createWithParams and operateWithParams methods from the TMAExternalElementView protocol.

```
#import "QMATestView.h"
#import "TMAExternalJSPlugin.h"

@interface QMATestView () <TMAExternalElementView>

@end

@implementation QMATestView {
    UILabel *_textLabel;
    UIButton *_clickButton;
    id<TMAExternalJSContextProtocol> _context;
}

TMARegisterExternalElement(maTestView);
+ (UIView *)createWithParams:(NSDictionary *)params context:(id<TMAExternalJSContex
    QMATestView *testView = [[QMATestView alloc] initWithFrame:CGRectZero];
    NSDictionary *testViewParams = QQ_Dict_DictValue(params, @"params");
    [testView setText:QQ_Dict_StringValue(testViewParams, @"text")];
    testView->_context = context;
    return testView;
}

//Handle events called from the mini program side
- (void)operateWithParams:(NSDictionary *)param context:(id<TMAExternalJSContextPro
    NSDictionary *data = QQ_Dict_DictValue(param, @"data");
    NSDictionary *params1 = QQ_Dict_DictValue(data, @"params1");
    NSInteger age = [QQ_Dict_NumberValue(params1, @"age") integerValue];
    NSString *name = QQ_Dict_StringValue(params1, @"name");
    qq_weakify(self);
    [MAUtils executeOnThread:[NSThread mainThread] block:^{
      qq_strongify(self);
      if (self) {
```

©2013-2022 Tencent Cloud. All rights reserved.

```
        self->_textLabel.text = [NSString stringWithFormat:@"name = %@ , age = %ld"
        // Return the result to the mini program side
        TMAExternalJSPluginResult *result = [TMAExternalJSPluginResult new];
        result.result = @{@"result":@"success"};
        [context doCallback:result];
      }
    }];
}
```

Sending events to the mini program side

To send events to the mini program side from a customized native component, first record the context in the `createWithParams:context:` method:

```
_context = context;
```

Send an event as follows:

```
- (void)onClickButton:(UIButton *)button {
    _textLabel.text = @"What do you want me to do";
   //Assemble data and send the event
    NSString *data = [MAUtils JSONStringify:@{@"externalElementId":_elementId,@"typ
    [_context doSubscribe:kTMAOnExternalElementEvent data:data];
}
```

Usage on the mini program side

1. Include the following in the mini program wxml:

```
<external-element
    id="comp1"
    type="maTestView"
    _insert2WebLayer
    style="width: 200px;height: 100px;"
    bindexternalelementevent="handleEvent"
></external-element>
```

Note：

The `type` must be agreed upon with the native side. If not on the same layer, do not set the _insert2WebLayer attribute.

`bindexternalelementevent` can listen to operations passed from the native side, with callback parameters including:

```
{
target,
currentTarget,
```

©2013-2022 Tencent Cloud. All rights reserved.

```
timeStamp,
touches,
detail, // Parameters passed from native side
}
```

2. Create an instance in the mini program js.

```
this.ctx = wx.createExternalElementContext('comp1');
```

Note：

The parameter for wx.createExternalElementContext is the element id in wxml.

3. Call instance methods in the mini program:

```
this.ctx.call({
params1: {
name: 'name1',
age: 11
},
params2: {
name: 'name2',
age: 22
},
success: (e) => {
console.log('====operate success=====', e)
},
fail: (e) => {
console.log('====operate fail=====', e)
},
complete: (e) => {
console.log('====operate complete=====', e)
}
})
```

Note：

The instance provides a call method with success, fail, and complete callbacks. The base library will call the invoke method to pass parameters to the native side.

©2013-2022 Tencent Cloud. All rights reserved.

Customised SDK capabilities Introduce

Last updated：2024-09-25 20:39:24

In order to enrich the usage scenarios, the mini program SDK opens a part of the interface to the outside world, which can be customised and implemented by the host application in order to provide the capabilities featured by the host application for the mini program to use.

Prerequisites

To implement the interface customisation of the mini program SDK, you need to create a class and follow the TMFMiniAppSDKDelegate protocol:

```
#import <TCMPPSDK/TCMPPSDK.h>

@interface MIniAppDemoSDKDelegateImpl : NSObject <TMFMiniAppSDKDelegate>

@end
```

Once the prerequisite steps are complete, you can follow the sub-documentation below to customize the mini program SDK interface.

Related Documents

Customize mini program UI
Customize mini program information API
Customize sharing capability
Customize user attributes
Logging and event reporting
Open APIs
Others

©2013-2022 Tencent Cloud. All rights reserved.

Customize mini program UI

Last updated：2024-07-03 18:14:11

Customized image previewThe mini program SDK allows developers to customize certain native UI elements of the mini program. Developers can configure these elements according to the following guidelines.

Setting the Loading Page UI

The TCMPP mini program engine allows the host app to redefine the loading page that appears when a mini program is loading, replacing the default SDK loading page. This is achieved by implementing the customLoadingViewWithAppInfo method in the TMFMiniAppSDKDelegate protocol. Sample code:

```
- (UIView *)customLoadingViewWithAppInfo:(TMFMiniAppInfo *)appInfo frame:(CGRect)fr
    UIView *view = [[UIView alloc] initWithFrame:frame];
    //todo Set specific view content here

    return view;
}
```

Setting mini program navigation bar resources

The TCMPP mini program engine allows the host app to redefine the navigation bar resources to match its own style. This is done by implementing the stringWithConfigKey method in the TMFMiniAppSDKDelegate protocol. The following keys are supported:

Key	Description
TMA_SK_MINIAPP_CloseButton	Close button icon
TMA_SK_MINIAPP_CloseButtonDark	Close button icon for dark theme
TMA_SK_MINIAPP_HomeButton	Home button icon
TMA_SK_MINIAPP_HomeButtonDark	Home button icon for dark theme
TMA_SK_MINIAPP_BackButton	Back button icon
TMA_SK_MINIAPP_BackButtonDark	Back button icon for ark theme
TMA_SK_MINIAPP_MoreButton	More button icon

©2013-2022 Tencent Cloud. All rights reserved.

TMA_SK_MINIAPP_MoreButtonDark	More Button icon for dark Theme
TMA_SK_MINIAPP_RecordButton	Record button icon
TMA_SK_MINIAPP_RecordButtonDark	Record button icon for dark Theme
TMA_SK_MINIAPP_MoreBackground	Capsule background image
TMA_SK_MINIAPP_MoreBackgroundDark	Capsule background image for dark theme

Sample code:

```
- (NSString *)stringWithConfigKey:(NSString *)key {
    //Set the close button in light mode
    if([key isEqualToString:TMA_SK_MINIAPP_CloseButton]) {
        return [[[NSBundle mainBundle] resourcePath] stringByAppendingPathComponent
    } else if([key isEqualToString:TMA_SK_MINIAPP_CloseButtonDark]) {
        return [[[NSBundle mainBundle] resourcePath] stringByAppendingPathComponent
    }


    return nil;
}
```

Setting the mini program more menu

The TCMPP mini program engine allows the host app to redefine the more button menu in the navigation bar, including:

1. Adding custom menu items in capsule

Custom menu items can be added to the capsule view by implementing the `customizedConfigForShare` method in the TMFMiniAppSDKDelegate protocol. The added options are categorized as follows:
MAUIDelegateShareViewTypeCustomizedShare: Share type, triggers the share logic within the mini program. The share operation is implemented by triggering shareMessageWithMod in TMFMiniAppSDKDelegate.
MAUIDelegateShareViewTypeCustomizedAction: Custom action type, allows custom callback events.

Sample code:

```
- (NSArray<TMASheetItemInfo *> *)customizedConfigForShare {
    NSMutableArray *arrays = [[NSMutableArray alloc] init];
    TMASheetItemInfo *item1 = [[TMASheetItemInfo alloc] initWithTitle:@"More sharin
    item1.icon = [UIImage imageNamed:@"icon_moreOperation_shareChat"];
    [arrays addObject:item1];
```

©2013-2022 Tencent Cloud. All rights reserved.

```
    TMASheetItemInfo *item2 = [[TMASheetItemInfo alloc] initWithTitle:@"click" type
        NSLog(@"click Click");
    }];

    item2.icon = [UIImage imageNamed:@"icon_moreOperation_collect"];
    [arrays addObject:item2];

    return arrays;
}
```

2. Customizing the pop-up capsule view

The capsule view can be customized by implementing the `showShareViewWithTitle` method in the TMFMiniAppSDKDelegate protocol.

```
/// Sharing panel
/// If this method is not implemented, `showActionSheetWithTitle:cancelButtonTitle:
/// @param title Title
/// @param cancelAction Cancel operation
/// @param otherButtonTitleAndActions Other buttons and response operations
/// @param dismissBlock Operations to be executed after the panel is collapsed (be
/// @param parentVC The VC that calls the panel
- (void)showShareViewWithTitle:(nullable NSString *)title
                  cancelAction:(nullable dispatch_block_t)cancelAction
   otherButtonTitleAndActions:(nullable NSArray *)otherButtonTitleAndActions
                  dismissBlock:(nullable dispatch_block_t)dismissBlock
                     parentVC:(UIViewController *)parentVC;
```

Setting mini program transition animations

The transition animation for mini program startup can be customized by implementing the getTMFSlideAnimationType method in the TMFMiniAppSDKDelegate protocol. Supported types include bottom-to-top, top-to-bottom, left-to-right, right-to-left, and default (bottom-to-bottom).

```
//Set the transition animation to bottom-to-top when the mini program launches
- (TMFSlideAnimationType)getTMFSlideAnimationType{
    return TMFSlideAnimationTypeBottomToTop;
}
```

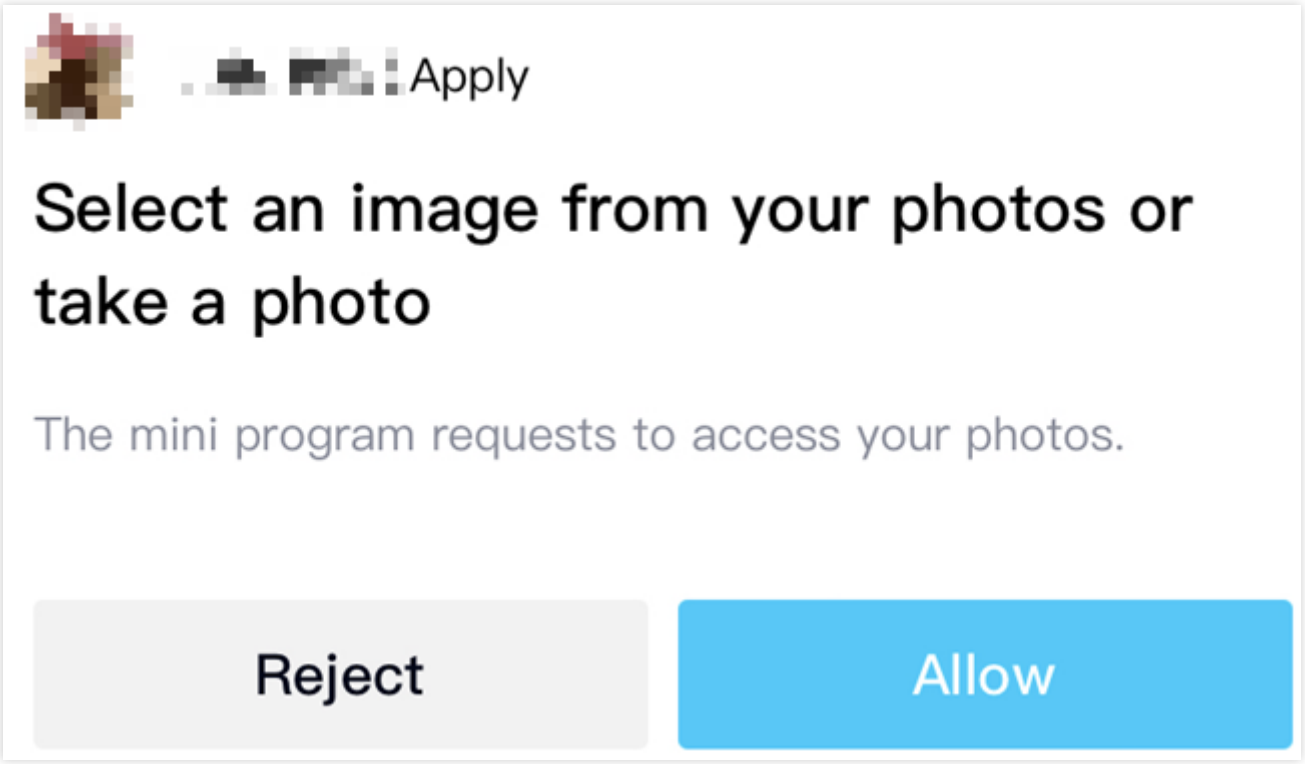
Setting mini program permission dialogs

©2013-2022 Tencent Cloud. All rights reserved.

Custom permission dialogs can be created by implementing the createAuthorizeAlertViewWithFrame method in the TMFMiniAppSDKDelegate protocol. This method includes parameters for the mini program and the required permissions.

```
/**
 * @brief Create a customized authorization window
 *
 * @param frame Window size
 * @param scope Refer to WeChat authorization scope
 * @ param title Permission name
 * @ param desc Permission description information
 * @ param privacyApi Currently calling API
 * @ param appInfo Current mini program information
 * @ param allowBlock Allows callbacks
 * @ param denyBlock Rejects callbacks
 */

- (UIView *)createAuthorizeAlertViewWithFrame:(CGRect)frame
                          scope:(NSString *)scope
                          title:(NSString *)title
                          desc:(NSString *)desc
                          privacyApi:(NSString *)privacyApi
                         appInfo:(TMFMiniAppInfo *_Nullable)appInfo
                          allowBlock:(void (^)(void))allowBlock
                           denyBlock:(void (^)(void))denyBlock;
```

©2013-2022 Tencent Cloud. All rights reserved.



Setting internal mini program UI

The TCMPP engine also supports customizing internal mini program UI elements by implementing corresponding methods in the TMFMiniAppSDKDelegate protocol. Supported elements include:

Mini program API	TMFMiniAppSDKDelegate method
wx.showLoading	`- (void)showLoading:(TMALoadingInfo * _Nullable)infos;`
wx.hideLoading	`- (void)hideLoading;`
wx.showToast	`- (void)showToast:(TMAToastInfo *)infos;`
wx.hideToast	`- (void)hideToast;`
wx.showActionSheet (actionSheetType = 0)	`- (void)showActionSheetWithTitle:(nullable NSString *)title cancelButtonTitle:(nullable NSString *)cancelBut cancelAction:(nullable dispatch_block_t)can otherButtonTitleAndActions:(nullable NSArray *)otherButto`

©2013-2022 Tencent Cloud. All rights reserved.

	`dismissBlock:(nullable dispatch_block_t)dis` `presentingViewController:(UIViewController *)presenting`

wx.showActionSheet (actionSheetType = 1)	`- (void)showShareViewWithTitle:(nullable NSString *)title`
	`cancelAction:(nullable dispatch_block_t)cance`
	`otherButtonTitleAndActions:(nullable NSArray *)otherButtonT`
	`dismissBlock:(nullable dispatch_block_t)dismi`
	`parentVC:(UIViewController *)parentVC;`

wx.showModal	`- (void)showAlertWithTitle:(nullable NSString *)title`
	`withMessage:(nullable NSString *)message`
	`actionBlocks:(nullable NSArray<AlertActionInfo*> *)acti`
	`presentingViewController:(UIViewController*)presentingV`

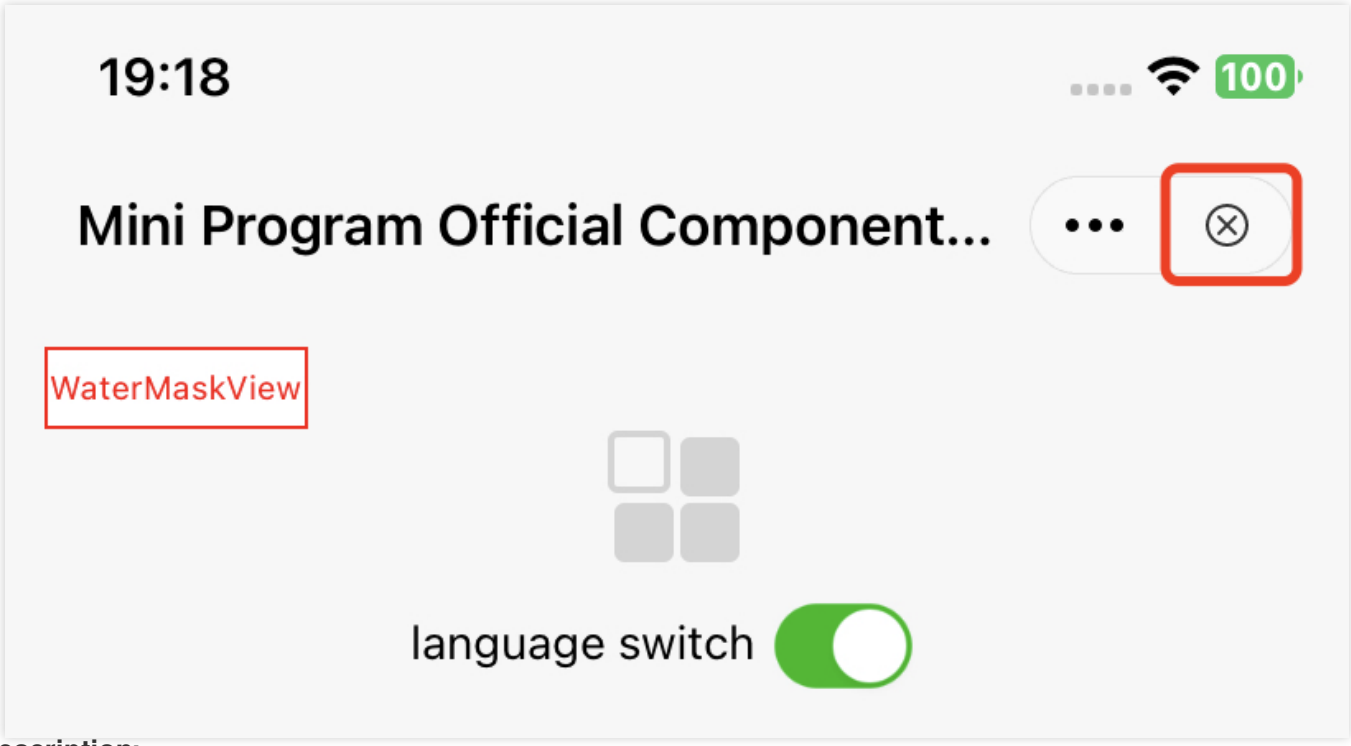
Customizing capsule button features

Customizing close button event listener

The close button event can be customized to allow the host app to receive callback events when the close button is clicked.

Close button diagram:

©2013-2022 Tencent Cloud. All rights reserved.



API description:

```
#pragma mark - exit retention - exit retention
- (BOOL)shouldDetainUser:(TMFMiniAppInfo *)app;
```

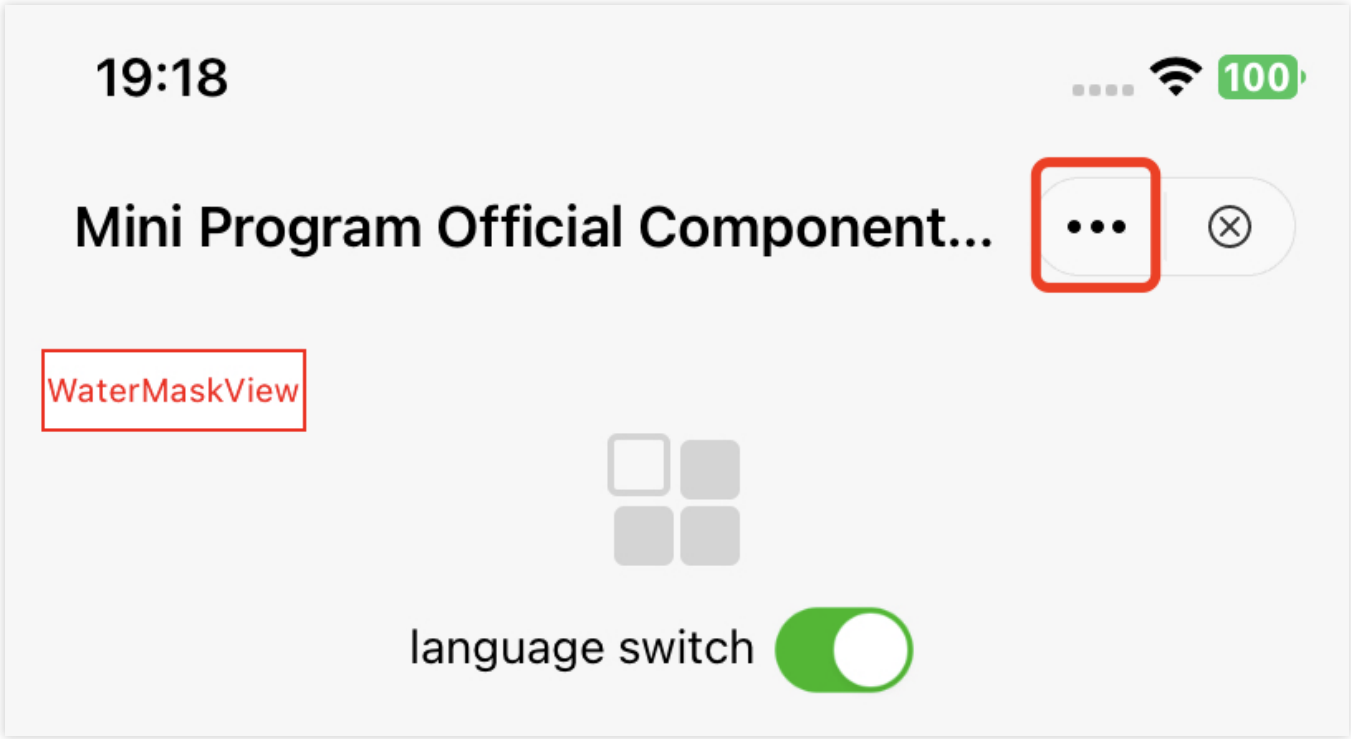
This API is triggered when the close button is clicked, if it returns YES, a pop-up window will appear to retain the user, if it returns NO, the mini program will be exited directly.

Customise more button event listeners

Customising the event listener of the More button allows the host application to listen to the callback event of the corresponding item when the More button is clicked.

Diagram of the More button:

©2013-2022 Tencent Cloud. All rights reserved.



API description:

```
// Click the capsule button to call up the panel
// If this method is not implemented, showActionSheetWithTitle:cancelButtonTitle:ca
// @param app Mini program information
// @param cancelButtonTitle cancel title
// @param cancelAction cancel the operation
// @param otherButtonTitleAndActions other buttons and response operations
// @param dismissBlock The operation that needs to be performed after the panel is
// @param parentVC - calls up the vc of the panel

- (void)showMoreButtonActionSheetWithApp:(TMFMiniAppInfo *)app
                             cancelButtonTitle:(nullable NSString *)cancelButtonTitle
                                  cancelAction:(nullable dispatch_block_t)cancelActio
                    otherButtonTitleAndActions:(nullable NSArray *)otherButtonTitleAn
                                  dismissBlock:(nullable dispatch_block_t)dismissBloc
                                  parentVC:(UIViewController *)parentVC;
```

Customize more button display list

When the user triggers more button click events, the following optional extended button list will pop up.

By overriding the customizedConfigForShare method, you can customize the sharing path and determine the display order.

API description:

```
// The host App can customize the sharing channel and determine the display order.
```

©2013-2022 Tencent Cloud. All rights reserved.

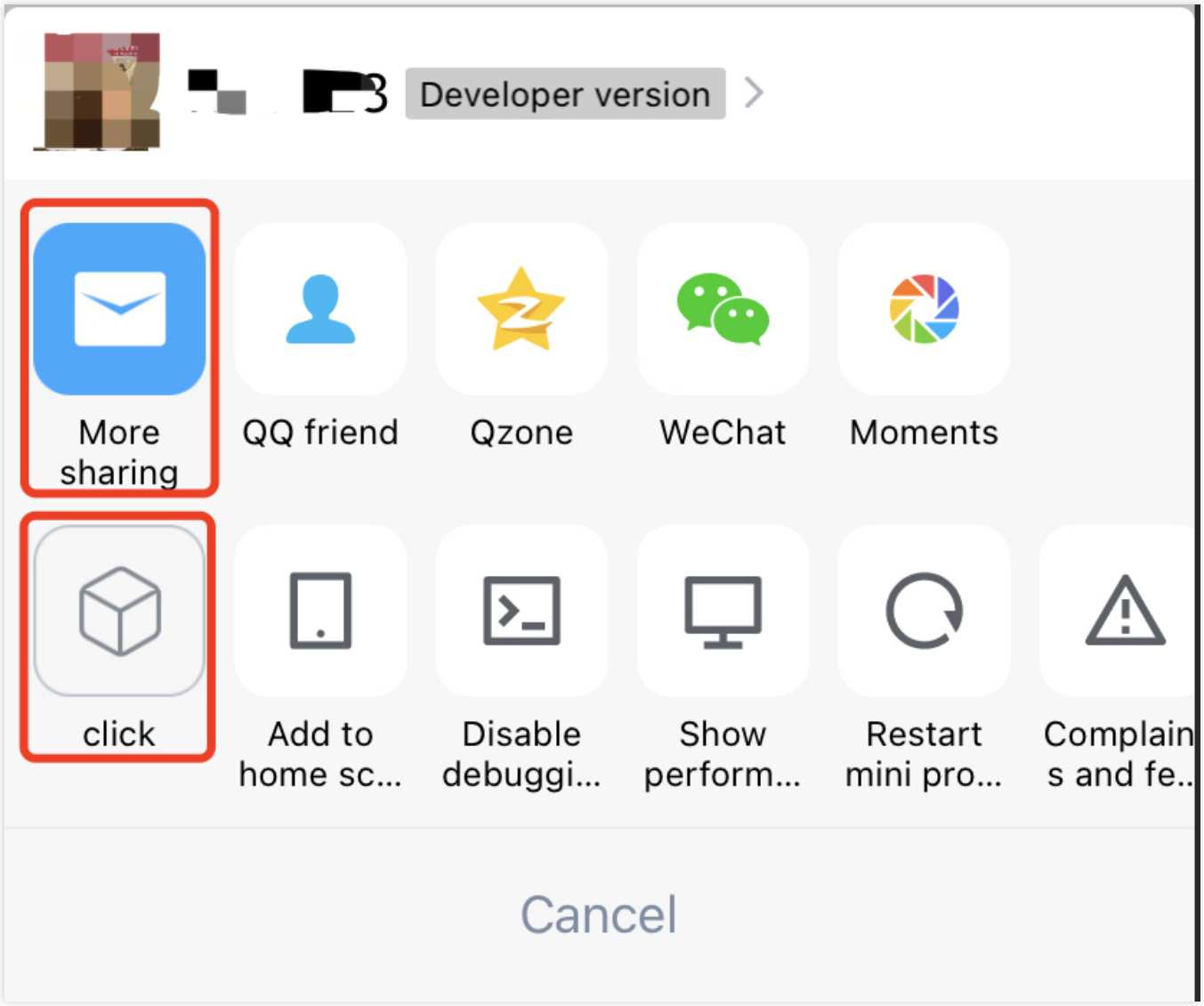
```
// 1. Default channels: QQ friends, QQ space, WeChat, Moments (see MAUIDelegateShar
// 2. Custom sharing channels: Host App customization (type is filled in MAUIDelega
// 3. Custom event processing: Host App customization (type is filled in MAUIDelega
// The display order of the above three channels supports mixed arrangement
///
// The host App can customize the sharing path and determine the display order. It
// 1. Default channels: QQ Friends, QQ Space, WeChat, Moments (for specific types,
// 2. Customized sharing channel: Host App customization (type fills in MAUIDelegat
// 3. Custom event processing: Host App customization (type fills in MAUIDelegateSh
// The display order of the above three channels
- (NSArray<TMASheetItemInfo *> *)customizedConfigForShare;
```

Sample code:

```
- (NSArray<TMASheetItemInfo *> *)customizedConfigForShare {
    NSMutableArray *arrays = [[NSMutableArray alloc] init];
    TMASheetItemInfo *item1 = [[TMASheetItemInfo alloc] initWithTitle:@"More sharin
    item1.icon = [UIImage imageNamed:@"icon_moreOperation_shareChat"];
    [arrays addObject:item1];

    TMASheetItemInfo *item2 = [[TMASheetItemInfo alloc] initWithTitle:@"click" type
        NSLog(@"click click");
    }];
    item2.icon = [UIImage imageNamed:@"icon_moreOperation_collect"];
    [arrays addObject:item2];
    return arrays;

}
```

The effect is as follows:

©2013-2022 Tencent Cloud. All rights reserved.



Picture selection

The mini program SDK provides a default image selection, which uses the Android system's built-in album selector by default, and is triggered by calling multimedia selection (wx.chooseImage) in the mini program.

Customized image selection can be achieved by overriding the selectMediaFromPickerWithModel method.

```
// Select media from the image picker - Select media from the image picker
// @param model configuration - configuration
// @param parentVC vc
// @param completionBlock After the selection is completed, data needs to be return
parentVC:(UIViewController *)parentVC
completionBlock:(void(^)(NSArray * _Nullable medias, NSError * _Nullable error))com

// Shooting media - Shooting media
```

©2013-2022 Tencent Cloud. All rights reserved.

```
// @param model configuration - configuration
// @param parentVC vc
// @param completionBlock After the selection is completed, data needs to be return
- (void)selectMediaFromCameraWithModel:(TMAMediaChooseConfigModel *)model
parentVC:(UIViewController *)parentVC
completionBlock:(void(^)(id _Nullable media, NSError * _Nullable error))completionB

// Get image data from PHAsset - Get image data from PHAsset
// @param phAsset media object - media object
// @param needCompress whether compression is required - whether compression is req
- (NSData *)imageDataFromPhAsset:(PHAsset *)phAsset needCompress:(BOOL)needCompress
```

Customized image preview

The Mini Program SDK provides a default image preview implementation, which is triggered when the Mini Program calls the multimedia selection (wx.previewImage).

Customized image preview can be implemented by overriding the - (void)navigationController:(UINavigationController *)navigationController presentImageWithCurrentUrl:(NSString *)currentAbsoluteUrl imageUrlArray:(NSArray *)absUrlsInPreviewArray method.

```
// Image Preview
// @param navigationController calls up the navigation bar for image preview
// @param currentAbsoluteUrl current page address
// @param absUrlsInPreviewArray pictures to be previewed
- (void)navigationController:(UINavigationController *)navigationController
presentImageWithCurrentUrl:(NSString *)currentAbsoluteUrl
imageUrlArray:(NSArray *)absUrlsInPreviewArray;
```

Customize document preview

iOS SDK provides a default document preview function. If you want to implement the document preview function yourself, please rewrite the following interface:

```
// Open document preview - Open document preview
// @param filePath document path - document path
// @param titleName title - title
// @param appInfo Mini program appinfo - Mini program appinfo
- (void)openDocument:(NSString *)filePath
title:(NSString *)titleName
appInfo:(TMFMiniAppInfo *_Nonnull)appInfo;
```

©2013-2022 Tencent Cloud. All rights reserved.

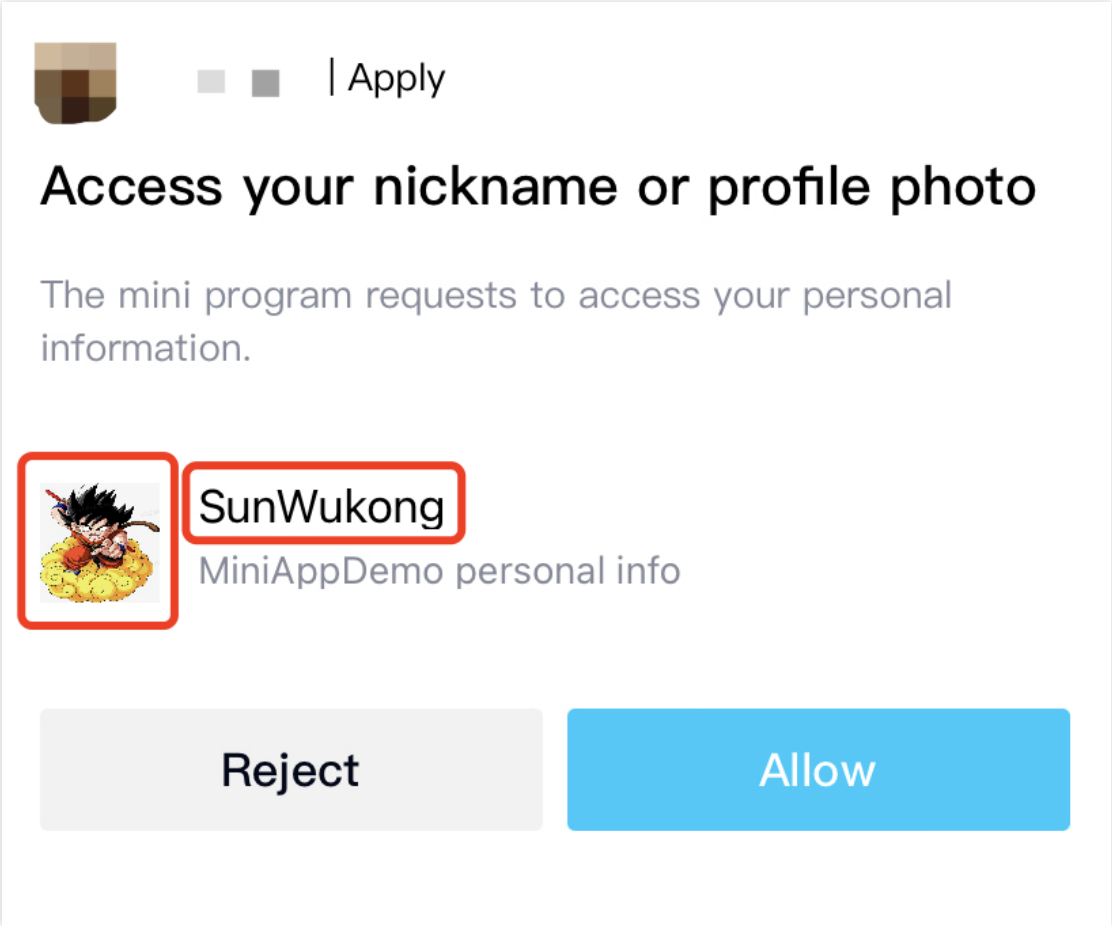
©2013-2022 Tencent Cloud. All rights reserved.

Customize mini program information API

Last updated：2024-07-03 18:10:49

Customizing user information in authorization pop-up box

When a mini program requests user information, an authorization pop-up will appear. The host app can customize certain parts of this pop-up, including the profile image and username, as shown in the image below:



You can achieve this customization by overriding the fetchAppUserInfoWithScope method.

```
/**
 * @brief Fetch the profile photo and username from the SDK host platform based on
 - Pull the user avatar and nickname of the SDK host platform according to the scop
 */
- (void)fetchAppUserInfoWithScope:(NSString *)scope block:(TMAAppFetchUserInfoBlock
```

Sample code:

©2013-2022 Tencent Cloud. All rights reserved.

```
- (void)fetchAppUserInfoWithScope:(NSString *)scope block:(TMAAppFetchUserInfoBlock
    if (block) {
        UIImage *defaultAvatar = [UIImage imageWithContentsOfFile:[[[NSBundle mainB
        UIImageView *avatarView = [[UIImageView alloc] initWithImage:defaultAvatar]
        TMAAppUserInfo *userInfo = [TMAAppUserInfo new];
        userInfo.avatarView = avatarView;
        userInfo.nickName = @"SunWukong";
        block(userInfo);
    }
}
```

Customizing the userAgent in mini program WebView

Override the customUserAgent method to customize the user agent. Example code:

```
- (NSString *)customUserAgent:(NSString *)defaultUserAgent;
```

Customizing code scaning capability

The mini program SDK provides a default scan implementation (refer to the extension library support - Scan), and also offers an API for users to customize the scan capability.

```
// Scan the code to call the client's scan code module scancode - Scan the code to
// @param scanPrams Scan code parameter dictionary - scan code parameter dictionary
// @param navigationController Calls vc from the page - calls vc from that page
// @param completionHandler Callback result - callback result
- (void)scanCode:(NSDictionary *)scanPrams
navigationController:(UINavigationController *)navigationController
completionHandler:(MACommonCallback)completionHandler;
```

Monitoring mini program lifecycle

Override the lifeCycleForApp method to monitor the mini program lifecycle.

```
// Transparently transmit the application life cycle processing, and the business w
// @param appInfo Mini program information - The mini program information
// @param type Status - The state
```

©2013-2022 Tencent Cloud. All rights reserved.

```
- (void)lifeCycleForApp:(TMFMiniAppInfo *)appInfo type:(TMAAppLifeCycleType)type;
```

Customizing base library update strategy and monitoring

Override the canUpdateJSBaseLib method to monitor the update information of the mini program's base library.

```
// Base library update control - Basic library update control
// @param libInfo Update information including version: version number, url: downlo
// @return Whether to agree to update - Whether agree to update
- (BOOL)canUpdateJSBaseLib:(NSDictionary *)libInfo;
```

©2013-2022 Tencent Cloud. All rights reserved.

Customize sharing capability

Last updated：2024-07-03 18:08:04

Default Sharing Channels Configuration

The default sharing channels can be customised by overriding the defaultSharingChannels method.

API Description:

```
// Default sharing channel after clicking capsule button - MAShareTargetQ, MAShareT
// The default sharing channel after clicking the pill button - MAShareTarget  (MASh
// (The configuration of the mini program must be a subset of the App)
- (NSArray<NSNumber *> *)defaultSharingChannels;
```

Sample code:

```
- (NSArray<NSNumber *> *)defaultSharingChannels{
    return @[@(MAShareTargetQQ),@(MAShareTargetWXFriends),@(MAShareTargetWXMoment)]
}
```

Adding customised share buttons

By overriding the customisedConfigForShare method, you can customize the sharing route and determine the display order.

API description

```
// The host App can customise the sharing channels and determine the display order,
// 1. Default channels: QQ friends, Qzone, WeChat, Circle of Friends (see MAUIDeleg
// 2, custom sharing channels: host App custom (type fill MAUIDelegateShareViewType
// 3, custom event processing: host App custom (type fill MAUIDelegateShareViewType
// The above three channels display order support mixed arrangement
///
// The host app can customise the share path and determine the display order. Curre
// 1. Default channels: QQ Friends, Qzone, WeChat, Moment (see MAUIDelegateShareVie
// 2. Custom sharing channels: host App custom (type fill in MAUIDelegateShareViewT
// 3. Custom event handling: host app custom (type fill in MAUIDelegateShareViewTyp
// The order in which the above three channels are displayed
- (NSArray<TMASheetItemInfo *> *)customisedConfigForShare;
```

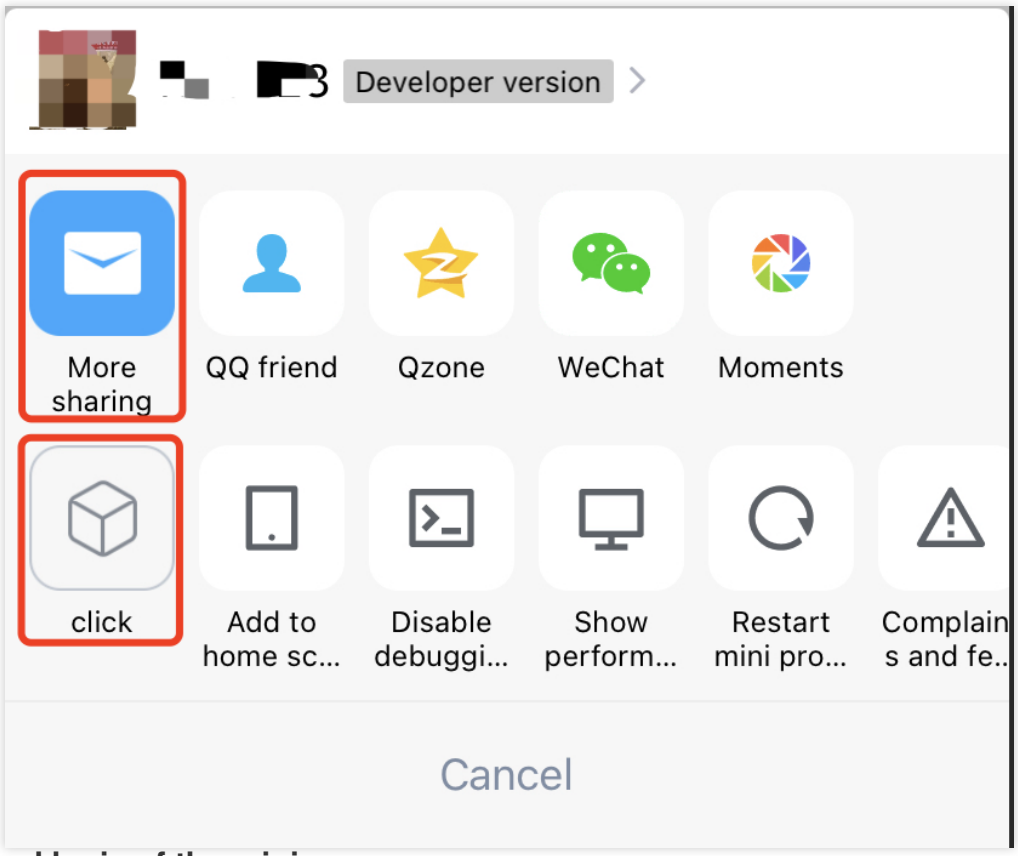
Sample code:

```
- (NSArray<TMASheetItemInfo *> *)customizedConfigForShare {
    NSMutableArray *arrays = [[NSMutableArray alloc] init];
    TMASheetItemInfo *item1 = [[TMASheetItemInfo alloc] initWithTitle:@"More sharin
```

©2013-2022 Tencent Cloud. All rights reserved.

```
    item1.icon = [UIImage imageNamed:@"icon_moreOperation_shareChat"];
    item1.shareTarget = 10001;
    [arrays addObject:item1];

    TMASheetItemInfo *item2 = [[TMASheetItemInfo alloc] initWithTitle:@"click" type
        NSLog(@"click Sample code:");
    }];
    item2.icon = [UIImage imageNamed:@"icon_moreOperation_collect"];
    [arrays addObject:item2];
    return arrays;

}
```

The effect is as follows:



Internal logic of the mini program

The mini program internally listens for user clicks on the share class button via onShareAppMessage and customises the forwarded content.

Share Logic Implementation

After the host APP receives the sharing data returned from the mini program, it triggers shareMessageWithModel to execute the final sharing action, and the developer can process the content to be shared according to the shareTarget and other data by themselves, and docking to the three-party sharing platform.

©2013-2022 Tencent Cloud. All rights reserved.

```
/**
 * @brief shareModel Shared Interface - Shared Interface
 * @param shareModel shareModel - shared model
 * @param appInfo mini program appinfo - Mini program appinfo
 * @param completionBlock Callback - Callbacks
 */
- (void)shareMessageWithModel:(TMAShareModel *_Nonnull)shareModel
                      appInfo:(TMFMiniAppInfo *_Nonnull)appInfo
              completionBlock:(nullable void(^)(NSError * _Nullable error))completi
```

©2013-2022 Tencent Cloud. All rights reserved.

Customize user attributes

Last updated：2024-07-03 18:13:58

The host application developer can set the user's nickname and avatar information through the method provided by the SDK。

Setting up user nicknames

```
/**
 * @brief SDK host platform's user nickname, returns "TCMPP" by default -- SDK host
 */
- (NSString *)getAppNickName;
```

Setting up user avatars

```
/**
* @ brief SDK host platform user avatar, the default return demo image, SDK host pl
 */
- (UIImage *)getAppAvatar;
```

Get current user account identifier

```
/**
 * @brief Get the current user account identifier of the SDK host platform, usually

 * Note: Returning nil will invalidate some caches within the SDK. If you are not l
 */
- (NSString *_Nonnull)getAppUID;
```

©2013-2022 Tencent Cloud. All rights reserved.

Logging and event reporting

Last updated：2024-07-03 18:07:06

SDK runtime log printing

Implement the log output API during the development phase to facilitate troubleshooting.

Note：

It is recommended to disable log output when the app is released to ensure security and performance.

```
/// Print log
/// @param level Log level. For more information, see `PLTLogLevel`.
/// @param msg Log message
- (void)log:(MALogLevel)level msg:(NSString *)msg;
```

Reporting the event

The host app can implement the event reporting API to override the internal reporting logic of the TCMPP SDK.

This includes mini program operation events and data reported by the mini program's internal wx.reportEvent.

```
typedef NS_ENUM(NSInteger,TMAReportEventID){.
    // undefined, report user-defined events
    // undefined, report user-defined event
    TMAReportEventID_None = 0,
    // Open the mini program.
    // Open the mini program.
    TMAReportEventID_OPEN_MINIAPP = 1,
    // update the mini program
    // Update the mini program.
    TMAReportEventID_UPDATE_MINIAPP = 2,
    // Download the mini program.
    // Download the mini program.
    TMAReportEventID_DOWNLOAD_MINIAPP = 3,
    // Open the mini program page.
    // Open the mini program page.
    TMAReportEventID_MINIAPP_PAGE_VIEW = 4,
    // Exit the mini program.
    // close the mini program
    TMAReportEventID_EXIT_MINIAPP = 5,
    // mini program behaviour event, atcion:0 background; 1 foreground
    // mini program behaviour event, atcion: 0 onHide; 1 onShow
    TMAReportEventID_MINIAPP_ACTION = 6
};

// Upload data - report data
// @param event event, refer to TMAReportEventID - event ID, refer to TMAReportEven
```

©2013-2022 Tencent Cloud. All rights reserved.

```
// @param eventName eventName - the name of the event
// @param params params - Parameters
// @param appInfo appletInfo - information about the mini program
// @return Whether to intercept internal reporting - Whether to intercept internal
- (BOOL)reportEvent:(int)eventId
          eventName:(NSString *)eventName
             params:(NSDictionary *)params
            appinfo:(TMFMiniAppInfo *)appInfo;
```

Real-time log reporting for mini programs

The host app can implement an event realtime log reporting interface to override the reporting logic inside the TCMPP SDK.

Including the log data written by calling wx.getRealtimeLogManager inside the mini program.

```
// Report log data - Report log data
// @param appId appID
// @param jsVersion base library version - base library version
// @param page current page - Current page
// @param filterMsgs Filtered Content - Filtered Content
// @param logs log events - log events
// @param completionBlock - The result of the callback.
// @return Whether to block internal reporting - Whether to block internal reportin
- (BOOL)reportRealTimeLogWithAppId:(NSString *)appId
                         jsVersion:(NSString *)jsVersion
                              page:(NSString *)page
                         filterMsgs:(NSArray <NSString *>*)filterMsgs
                              logs:(NSArray <TMARealtimeLogItem *>*)logs
                   completionBlock:(void (^)(NSError * _Nullable error))completionB
```

Internal log reporting for mini programs

The host app can implement event real-time log reporting interface to override the reporting logic inside TCMPP SDK. Including the log data written by wx.getLogManager within the mini program, and the user can upload the printed log by using the open-type="feedback" of the button component.// Upload the logs of the mini program corresponding to the appID - Upload the logs of the mini program corresponding to the appID

```
// Upload the logs of the mini program corresponding to the appID - Upload the logs
// Implemented using TMFMiniAppSDKManager's `sandBoxPathWithAppID:` interface to ge
// @param appID appID of the mini program - appID of the applet/game.

- (void)uploadLogFileWithAppID:(NSString *)appID;
```

©2013-2022 Tencent Cloud. All rights reserved.

©2013-2022 Tencent Cloud. All rights reserved.

Open APIs

Last updated：2024-07-03 18:06:49

The following methods of the Weixin mini program can be implemented through the proxy class, with the app implementing the specific logic and returning it to the mini program.
Mini program methods:
wx.login ；
wx.getUserInfo ；
wx.getUserProfile ；
wx.checkSession ；
wx.requestPayment ；
wx.getPhoneNumber 。

Related APIs in proxy class

```
// Start payment - Request payment

// @param app Mini program/mini game instances - Mini program/mini game instance

// @param params Parameters - parameters

// @param completionHandler Callback results - Result callback

- (void)requestPayment:(TMFMiniAppInfo *)app params:(NSDictionary *)params completi


// login

// @param app Mini program/mini game instances - Mini program/mini game instance

// @param params Parameters - parameters

// @param completionHandler Callback results - Result callback

- (void)login:(TMFMiniAppInfo *)app params:(NSDictionary *)params completionHandler


// checkSession

// @param app Mini program/mini game instances - Mini program/mini game instance

// @param params Parameters - parameters
```

©2013-2022 Tencent Cloud. All rights reserved.

```
// @param completionHandler Callback results - Result callback

- (void)checkSession:(TMFMiniAppInfo *)app params:(NSDictionary *)params completion



// getUserProfile

// @param app Mini program/mini game instances - Mini program/mini game instance

// @param params Parameters - parameters

// @param completionHandler Callback results - Result callback

- (void)getUserProfile:(TMFMiniAppInfo *)app params:(NSDictionary *)params completi



// getPhoneNumber

// @param app Mini program/mini game instances - Mini program/mini game instance

// @param params Parameters - parameters

// @param completionHandler Callback results - Result callback

- (void)getPhoneNumber:(TMFMiniAppInfo *)app params:(NSDictionary *)params completi



// getUserInfo

// @param app Mini program/mini game instances - Mini program/mini game instance

// @param params Parameters - parameters

// @param completionHandler Callback results - Result callback

- (void)getUserInfo:(TMFMiniAppInfo *)app params:(NSDictionary *)params completionH
```

Instance code

```
// Start payment - Request payment
```

©2013-2022 Tencent Cloud. All rights reserved.

```
// @param app Mini program/mini game instances - Mini program/mini game instance
// @param params Parameters - parameters
// @param completionHandler Callback results - Result callback
- (void)requestPayment:(TMFMiniAppInfo *)app params:(NSDictionary *)params completi

// login
// @param app Mini program/mini game instances - Mini program/mini game instance
// @param params Parameters - parameters
// @param completionHandler Callback results - Result callback
- (void)login:(TMFMiniAppInfo *)app params:(NSDictionary *)params completionHandler

// checkSession
// @param app Mini program/mini game instances - Mini program/mini game instance
// @param params Parameters - parameters
// @param completionHandler Callback results - Result callback
- (void)checkSession:(TMFMiniAppInfo *)app params:(NSDictionary *)params completion


// getUserProfile
// @param app Mini program/mini game instances - Mini program/mini game instance
// @param params Parameters - parameters
// @param completionHandler Callback results - Result callback
- (void)getUserProfile:(TMFMiniAppInfo *)app params:(NSDictionary *)params completi



// getPhoneNumber
```

©2013-2022 Tencent Cloud. All rights reserved.

```
// @param app Mini program/mini game instances - Mini program/mini game instance

// @param params Parameters - parameters

// @param completionHandler Callback results - Result callback

- (void)getPhoneNumber:(TMFMiniAppInfo *)app params:(NSDictionary *)params completi



// getUserInfo

// @param app Mini program/mini game instances - Mini program/mini game instance

// @param params Parameters - parameters

// @param completionHandler Callback results - Result callback

- (void)getUserInfo:(TMFMiniAppInfo *)app params:(NSDictionary *)params completionH
```

©2013-2022 Tencent Cloud. All rights reserved.

Others

Last updated：2024-08-13 17:11:32

Host app APIs

```
/**
 * @brief SDK host app name
 * This API is mainly used to indicate the app name.
 */
- (NSString *)appName;

/**
 * @brief SDK host app version
 * @return A lowercase string, such as 1.0.0, is returned.
 */
- (NSString *)getAppVersion;

/**
 * @brief The network status of the SDK host app
 */

- (TMANetWorkStatus)getAppNetworkStatus;

/**
 * @brief Model of the SDK host app
 */
- (NSString *)getAppIPhoneModel;

/**
 * @brief Device information of the SDK host app
 * @return format: {@"brand":@"iPhone",@"model":@"iPhone 11&lt;iPhone12,1>",@"syste
 */
- (NSDictionary *)getAppDeviceInfo;

/**
 * @brief Obtain basic information of the host app
 * @return Format: {@"SDKVersion":@"2.32.2",@"model":@"iPhone 11&lt;iPhone12,1>",@"
 */
- (NSDictionary *)getAppBaseInfo;

/**
 * @brief Obtain current language set by the host app
 * @return Format: "zh-Hans"
 */
- (NSString *)getCurrentLocalLanguage;
```

©2013-2022 Tencent Cloud. All rights reserved.

```
/**
 * @brief The current theme set by the host app. If it is not implemented via this
 */
- (NSString *)getAppTheme;

/**
 * @brief Clipboard frequency control
 */
- (NSNumber *)getClipboardInterval;

// The maximum number of mini programs kept alive. Default value: 3



- (NSInteger)maxMiniAppKeepAliveCount;


// Set the desktop shortcut for the host app scheme

- (NSString *)getAppScheme;
```

Screen capture, screen recording and watermarking

```
- (void)applet:(TMFMiniAppInfo *)appletInfo screenCaptureStatusChanged:(BOOL)isCapt

- (void)appletDidTakeScreenshot:(TMFMiniAppInfo *)appletInfo atPagePath:(NSString *

- (nullable UIView *)appletCustomizeWatermarkView:(TMFMiniAppInfo *)appletInfo;
```

Process special URLs in web-view component

Special URLs in web-view component pages should be passed to the host app for processing.

```
// Triggered when redirecting to non-http/https in the web-view component. It is us
// @param app {TMFMiniAppInfo} Information of the mini-program to which the web-vie
// @param url {NSURL} The URL to be accessed
// @return Whether to intercept
- (BOOL)webViewCustomUrlLoading:(TMFMiniAppInfo *)app url:(NSURL *)url {
    NSLog(@"webViewCustomUrlLoading:%@,appid:%@",[url absoluteString],app.appId);
    if ([[UIApplication sharedApplication] canOpenURL:url]) {
        if (@available(iOS 10.0, *)) {
            [[UIApplication sharedApplication] openURL:url options:@{} completionHa
                if (success) {
                    NSLog(@"webViewCustomUrlLoading:%@,appid:%@, open sucess",[url
```

©2013-2022 Tencent Cloud. All rights reserved.

```
                }
            }];
        } else  {
            [[UIApplication sharedApplication] openURL:url];
        }
        return YES;
    } else  {
        NSLog(@"webViewCustomUrlLoading:%@,appid:%@,cann't open!!!",[url absoluteSt
    }
    return NO;
}
```

©2013-2022 Tencent Cloud. All rights reserved.

API Introduction

Last updated：2024-07-03 18:15:06

TMAVersionType

Mini program version

```
    //Development version
    // Development version
    TMAVersionDevelop = 0,

    //Preview version
    // Preview version
    TMAVersionPreview = 1,

    //Reviewed version
    // Audit version
    TMAVersionAudit = 2,

    //Online version (released version)
    // Online version (release version)
    TMAVersionOnline = 3,

    //Local preloaded version, will not be updated (Not available currently)
    // Local preset version will not be updated (not used yet)
    TMAVersionLocal = 10,
```

TMAEntryScene

The scene where the mini program is opened.

```
    // Initialization value
    // initialization value
    TMAEntrySceneNone                                = 0,

    // Main entry
    // main entrance
    TMAEntrySceneAIOEntry                            = 1001,

    // Search
    // search
    TMAEntrySceneSearch                              = 1005,

    // Scan QR code
    //Scan QR code
```

©2013-2022 Tencent Cloud. All rights reserved.

```
    TMAEntrySceneScanQRCode                              = 1011,
```

MAShareTarget

```
    //Share to QQ
    // Share to QQ
    MAShareTargetQQ = 0,

    //Share to Qzone
    // Share to Qzone
    MAShareTargetQzone,

     | 2 | Share to QQ, quickly share to the current chat window |
    // Share to QQ, and share to the current chat window quickly
    MAShareTargetQQFastForward,

    //WeChat friends
    // WeChat friends
    MAShareTargetWXFriends,

    // WeChat Moments
    // WeChat Moments
    MAShareTargetWXMoment,

    //Quick sharing of recent contacts, triggered by clicking on the sharing panel
    // Quick sharing of recent contacts, triggered by clicking on the sharing
    MAShareTargetChatFastShare,

    // Share contacts quickly
    //Quickly share friends list
    MAShareTargetFastShareTable,

    //openId sharing, click and share the open data domain of the source mini game
    // openId sharing, click and share the open data domain of the source mini game
     MAShareTargetOpenIdShare,
    MAShareTargetGuild,
```

TMAAppLifeCycleType

```

     TMAAppLifeCycleTypeNone,

     TMAAppLifeCycleTypeonColdOpen,

     TMAAppLifeCycleTypeonHotOpen,
```

©2013-2022 Tencent Cloud. All rights reserved.

```
    TMAAppLifeCycleTypeOnShow,

    TMAAppLifeCycleTypeOnHide,

    TMAAppLifeCycleTypeOnClose,

    TMAAppLifeCycleTypeTerminate,
```

©2013-2022 Tencent Cloud. All rights reserved.

iOS Extended component

Last updated：2024-07-03 18:15:27

In development and usage, many system-authorized APIs require preset authorization in the project's info.plist file. However, your app might not need these features. Therefore, the TCMPP SDK splits out an extension SDK, eliminating unnecessary authorizations and reducing the core module size.
The TCMPP SDK engine provides both core and extension modules, allowing users to integrate based on their needs.

Connect to and use extended SDKs

The mini program engine classifies SDKs into the core SDK and extended SDKs, and the latter is the supplement to the former. Therefore, extended SDKs need to depend on the core SDK. To guarantee the SDK security and stability, you should place APIs requiring permissions in extended SDKs as much as possible.

TCMPPExtMedia

`TCMPPSDKExtMedia` provides the default implementations of three APIs: `chooseMedia`, `chooseVideo`, and `chooseImage`. If the host app already has these capabilities, we recommend you implement them in open APIs. To use the media selection feature provided by TMF, you need to use this plugin.
Usage

```
pod 'TCMPPSDK'
pod 'TCMPPSDKExtMedia'
```

Add the following in info.plist file: Privacy - Photo Library Usage Description
After adding the Media extended SDK, the supported mini program API list is as follows:

API Name	Description
wx.chooseMedia	select a photo or video
wx.chooseVideo	Select a video
wx.chooseImage	Select a photo

Permissions involved:

Permission	Description
Album access permissions	Requires permissions to access the album

©2013-2022 Tencent Cloud. All rights reserved.

TCMPPExtScanCode

`TCMPPSDKExtScanCode` provides the processing logic for `wx.scanCode`. If the host app has code scanning and recognition capabilities, we recommend you use the following code to connect to the existing code scanning module: TMFMiniAppSDKDelegate.scanCode:(NSDictionary *)scanPrams navigationController:(UINavigationController *)navigationController completionHandler:(MACommonCallback)completionHandler;
To use the code scanning feature provided by TCMPP, you need to use this plugin.
Usage:

```
pod 'TCMPPSDK'
pod 'TCMPPSDKExtScanCode'
```

Add the following in info.plist file: Privacy - Camera Usage Description
After adding the Scan Code extended SDK, the supported mini program API list is as follows:

API Name	Description
wx.scanCode	Invokes the client's code scanning UI to scan a code

Permissions involved:

Permission	Description
Camera permission	Requires camera access for scanning codes
Gallery access rights	You need to request access to the gallery

TCMPPExtQMap

TCMPPSDKExtQMap provides Tencent Map SDK capabilities for users in Chinese mainland.
Usage：

```
pod "TCMPPSDKExtQMap"
```

Apply for an appkey on Tencent Maps Open Platform and set it in the code:

```
[TMFAppletQMapComponent setQMapApiKey:@"XXXXXXXXXXX"];
```

TCMPPExtBaiduMap

©2013-2022 Tencent Cloud. All rights reserved.

TCMPPSDKExtBaiduMap provides the capabilities of the Baidu Maps SDK for users in mainland China.

Usage：

```
pod "TCMPPExtBaiduMap"
```

Apply for an appkey on the Baidu Map open platform and set it in the code:

```
[TMFAppletBaiduMapComponent setBaiduMapApiKey:@"XXXXXXXXXXX"];
```

TCMPPExtAMap

TCMPPSDKExtBaiduMap provides users in mainland China with relevant capabilities of the Gode Map SDK.

Usage：

```
pod "TCMPPExtAMap"
```

Apply for appkey in the Gaode map open platform and set it in the code:

```
[TMFAppletAMapComponent setAMapApiKey:@"XXXXXXXXXXX"];
```

After adding the Tencent, Baidu, and AMap extended SDKs, the supported mini program API list is as follows:

API Name	Description
Map	Supports map-related APIs, including map display, location selection, and POI queries

Permissions involved:

Permission	Description
Locationa	Requires location access for map display and positioning

TCMPPExtLive

If you need to use live components (live-player and live-pusher) for live streaming scenarios, you need to add the following SDKs to support the live component functionalities.

```
pod "TCMPPSDKExtLive"
pod 'TXLiteAVSDK_Professional', :podspec => 'https://liteav.sdk.qcloud.com/pod/lite
```

In addition to adding the above dependencies, you also need to implement the following methods in TMFMiniAppSDKDelegate to provide the LicenseUrl and LicenseKey required by the live component for initialization.

©2013-2022 Tencent Cloud. All rights reserved.

If you do not configure the correct LicenseUrl and LicenseKey, the live component functionality will be unavailable.

Note：

For details about obtaining LicenseUrl and LicenseKey, see Adding and Renewing License.

```
- (NSString *)setLiveLicenceURL {
    return @"https://xxx.license";
}

- (NSString *)setLiveLicenceKey {
    return @"xxx";
}
```

Add the following in info.plist file:

Privacy - Camera Usage Description

Privacy - Microphone Usage Description

After adding the Live extension SDK, the supported mini program API list is as follows:

API Name	Description
wx.createLivePusherContext	Creates a live streaming pusher context.
LivePusherContext	Supports APIs related to LivePusherContext
wx.createLivePlayerContext	Creates a live streaming player context
LivePlayerContext	Supports APIs related to LivePlayerContext
Components	-
live-pusher	Tag for pushing
live-player	Tag for playing

Permissions involved:

Permission	Definition
Camera permission	-
Recording permission	-

TCMPPExtAuthentication

TCMPPExtAuthentication provides capabilities related to biometric authentication.

©2013-2022 Tencent Cloud. All rights reserved.

Usage：

```
pod "TCMPPExtAuthentication"
```

Usage

Add the following in info.plist file: Privacy - Face ID Usage Description

After adding the biometric authentication extended SDK, the supported mini program API list is as follows:

API Name	Description
wx.startSoterAuthentication	-
wx.checkIsSupportSoterAuthentication	-
wx.checkIsSoterEnrolledInDevice	-

Permissions involved:

Permission	Definition
Biometric authentication access	Requires biometric authentication permission

TCMPPExtBLE

TCMPPExtBLE provides capabilities related to low-energy Bluetooth and beacons.

Usage：

```
pod "TCMPPExtBLE"
```

Add the following in info.plist file:

Privacy - Bluetooth Always Usage Description

Privacy - Bluetooth Peripheral Usage Description

After adding the LBS extended SDK, the supported mini program API list is as follows:

API Name	Description
Bluetooth - general	General Bluetooth APIs
Bluetooth - Low energy peripherals	APIs related to peripheral devices
Bluetooth - Low energy central	APIs related to central devices
Bluetooth - Beacon	APIs related to Bluetooth beacons

©2013-2022 Tencent Cloud. All rights reserved.

Permissions involved:

Permission	Description
Bluetooth	Required for Bluetooth operations
Location	Required for Bluetooth device search

TCMPPExtCalendar

TCMPPExtCalendar provides calendar-related capabilities.

Usage：

```
pod "TCMPPExtCalendar"
```

Add the following in `info.plist` file:

Privacy - Calendars Usage Description

Privacy - Reminders Usage Description

API list:

API name	Description
addPhoneRepeatCalendar	Adds a repeated event to the system calendar
addPhoneCalendar	Adds an event to the system calendar

TCMPPExtClipBoard

TCMPPExtClipBoard provides clipboard-related capabilities.

Usage：

```
pod "TCMPPExtClipBoard"
```

API List：

API name	Description
setClipboardData	Sets the content of the system clipboard
getClipboardData	Gets the content of the system clipboard

©2013-2022 Tencent Cloud. All rights reserved.

TCMPPExtContact

TCMPPExtContact provides contact-related capabilities.

Usage：

```
pod "TCMPPExtContact"
```

Add the following in `info.plist` file: Privacy - Contacts Usage Description

API LIst：

API name	Description
chooseContact	Select a contact
addPhoneContact	Add phone contacts

TCMPPExtLBS

TCMPPExtLBS provides capabilities related to location services, maps, compass, accelerometer, device orientation, and gyroscope.

Usage：

```
pod "TCMPPExtLBS"
```

Add the following in info.plist file:

Privacy - Location Always and When In Use Usage Description

Privacy - Location Always Usage Description

Privacy - Location Usage Description

Privacy - Location When In Use Usage Description

Privacy - Motion Usage Description

API List：

API name	Description
Location	A series of APIs related to location
System map	A series of APIs related to system maps
Compass, accelerometer, device orientation, and gyroscope	A series of APIs related to these sensors

TCMPPExtMDNS

©2013-2022 Tencent Cloud. All rights reserved.

TCMPPExtMDNS provides capabilities related to local network communication.

Usage：

```
pod "TCMPPExtMDNS"
```

Add the following in `info.plist` file:

Privacy - Local Network Usage Description

Privacy - Bonjour services

API List：

API name	Description
MDNS	A series of APIs related to MDNS

TCMPPExtNetwork

TCMPPExtNetwork provides capabilities related to TCP/UDP communication.

Usage：

```
pod "TCMPPExtNetwork"
```

Add the following in info.plist file:

App Transport Security Settings

Allow Arbitrary Loads - YES

Privacy - Bonjour services

API List：

API name	Description
TCP	A series of APIs related to TCP
UDP	A series of APIs related to UDP

TCMPPExtMp3Encoder

TCMPPExtMp3Encoder provides the capability to save recordings as MP3 format when using RecorderManager.

Usage：

```
pod "TCMPPExtMp3Encoder"
```

Usage：

©2013-2022 Tencent Cloud. All rights reserved.

The recording to MP3 format depends on the Lame library, which is open-source under the GNU Library or Lesser General Public License version 2.0 (LGPLv2) and GNU General Public License version 2.0 (GPLv2). For details, see LAME. You can integrate as needed.

©2013-2022 Tencent Cloud. All rights reserved.

iOS FAQs

Last updated：2024-10-18 10:48:24

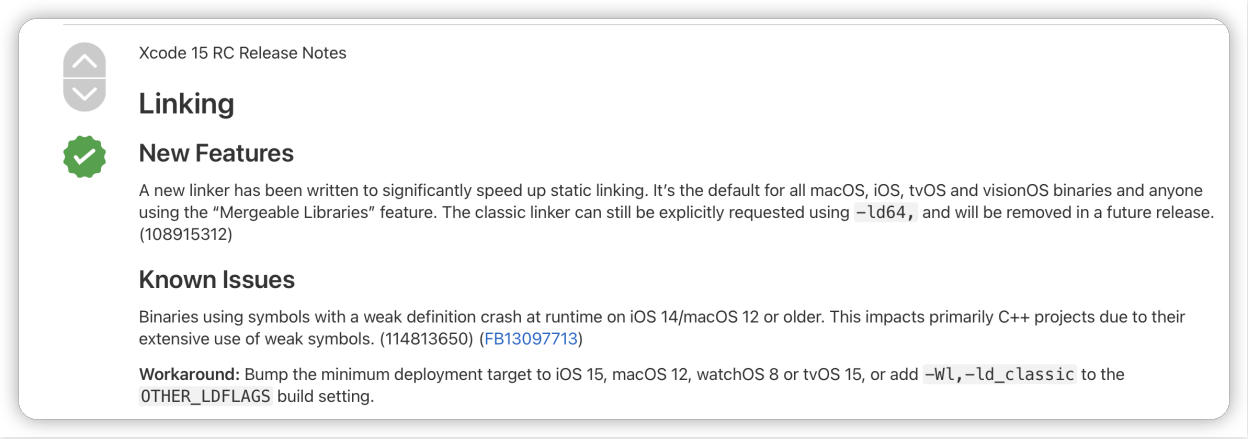
Integration FAQs

What is the minimum configuration required to integrate the SDK?

The TCMPP mini program SDK supports iOS 9 later versions. The versions earlier than iOS 9 are not supported.

Crashes on iOS 12 and earlier versions after compiling with XCode 15

Add the -Wl,-ld_classic configuration in the project compilation options.



What is the impact on the installation package size after integrating the SDK?

The TCMPP SDK uses a core library and extension libraries, allowing users to integrate the SDK as needed. Integrating only the core library impacts the overall installation package size by approximately 4 MB.

FAQs during usage

Troubleshooting mini program startup failures

If the mini program fails to start, it could be due to the following reasons:
Reason 1: Incorrect configuration file path. If the configuration file is in a sub-directory, you need to append the directory path, e.g., server/tcmpp-ios-configurations.json;
Reason 2: Modifying the mini program configuration file content is not allowed; otherwise, the mini program will not run properly;

©2013-2022 Tencent Cloud. All rights reserved.

Reason 3: The bundleId in the configuration file must match the application's bundleId. Otherwise, the app will fail to run because the bundleId in the configuration file is verified during initialization.

Is it supported to completely hide the navigation bar on a page of a mini program? If it is supported, how to implement it?

A: Support.

The navigationStyle of mini programs like WeChat, Alipay, Baidu, etc. has two values of default/custom, while TCMPP mini program SDK adds a new hide on top of that, that is, it has three values of default/custom/hide.

So on the page where you need to hide the navigation bar, just set navigationStyle to hide.

How do I set up grey scale release conditions?

The host application developer can set the user's attribute information (region or account number) through the method provided by the SDK, which is convenient for use in scenarios such as grey-scale push.

Set User ID

```
// Report user ID, if parameter is null it is considered unbound
// Report user ID, if parameter is null it is considered unbound.
// @param customisedUserID User customised user ID - customised user ID
- (void)updateCustomizedUserID:(nullable NSString *)customizedUserID;
```

Setting location information

©2013-2022 Tencent Cloud. All rights reserved.

```
// Report location information
// Report location information
// @param country Country - country
// @param province - province
// @param city - city
- (void)updateAreaInfoWithCountry:(nullable NSString *)country Province:(nullable N
```

```
[[TMFMiniAppSDKManager sharedInstance] updateCustomizedUserID:@"abc123"];
[[TMFMiniAppSDKManager shared instance] updateAreaInfoWithCountry:@"China" Province
```

Does it support uploading to a tri-party service when data is reported?

Support APP to report mini program data to any service, see Logging and event reporting.

Does mini program support data isolation for different login users?

Yes.

TCMPP SDK supports segregated storage of mini program data according to different logged-in users for data protection and business logic confusion, which requires the developer to implement the return of AppUID in the agent, and the SDK will store the data according to the AppUID.

```
/**
 * @brief Get the current user account identifier of the SDK host platform, usually

 * Note: Returning nil will invalidate some caches within the SDK. If not logged in
 */

- (NSString *_Nonnull)getAppUID;
```

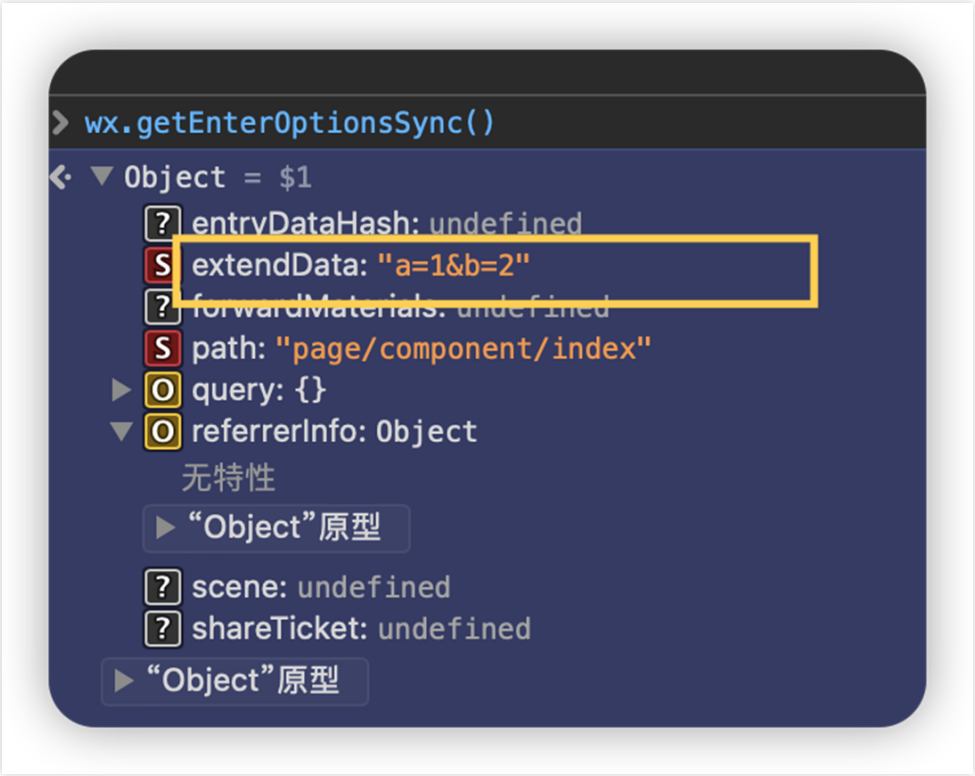
How do I pass parameters when Mini programs open?

Mini programs can be opened by specifying the paramsStr parameter:

```
[[TMFMiniAppSDKManager sharedInstance] startUpMiniAppWithAppID:@"mpbz0uqwzddj5zbt"
```

In Mini programs, you can get the parameters passed in when opening in the extendData field of wx.getEnterOptionsSync:

©2013-2022 Tencent Cloud. All rights reserved.



Debugging Related

How to debug a mini program?

After integrating the mini program SDK and running the mini program in Xcode (apps packaged using the develop profile can also be viewed using Safari on your computer), you can open Safari on your computer, and then in the toolbar Develop, select the running emulator or real device, you can select the list of pages opened by the mini program, and by selecting the currently opened page, you can review page elements, view network calls, and some logs.

After ios16.4, because of system limitations, you need to turn on inspectable to support safari development mode, you can implement the relevant interfaces in the proxy class and return YES.

```
// Whether to enable checkable after ios16.4, you can develop and debug via safari
// Whether or not inspectable is enabled after ios16.4, you can develop and debug v
- (BOOL)inspectableEnabled;
```

How to open vConsole?

The development version and preview version of mini programs support to open the debug switch in the capsule menu to view the VConsole information, the official version of the mini program can not open the VConsole, you need to open the VConsole through the mini program internal call wx.setEnableDebug.

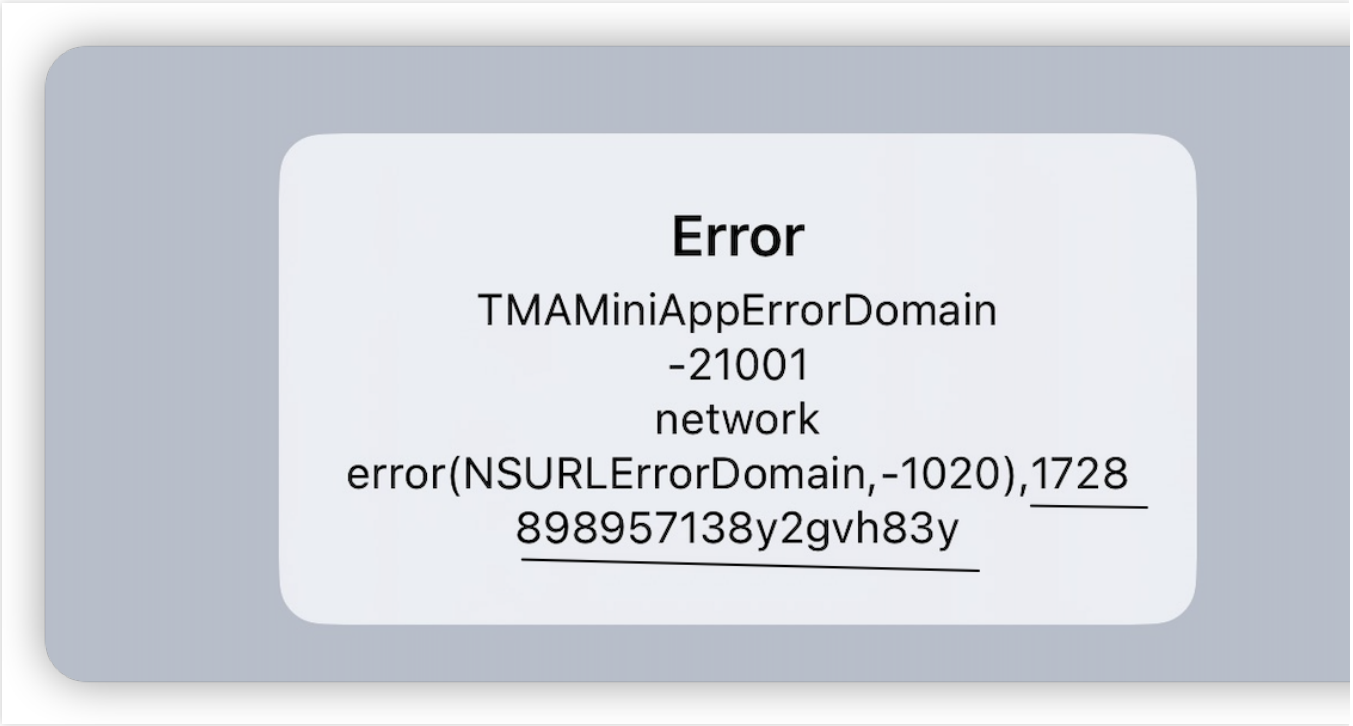
TCMPP SDK also supports to open the vConsole mode of all mini programs automatically in the app settings, you need to implement the related interface in the proxy class.

©2013-2022 Tencent Cloud. All rights reserved.

```
// Whether debugging is enabled
// Whether debugging is enabled
- (BOOL)vConsoleEnabled;
```

How to troubleshoot when a mini program opens abnormally?

When opening a mini program, you can get the error code and error description in the mini program callback interface. The error description contains the traceId of the current request. It is strongly recommended that you save the error information to facilitate troubleshooting.



©2013-2022 Tencent Cloud. All rights reserved.

iOS Error Codes

Last updated：2024-07-03 18:15:44

Error codes for opening mini program

Returns error code description

```
// Initialization information is incorrect or not initialized
TMAMiniAppErrorConfigInitError = -21000,

// The networkerror
TMAMiniAppErrorNetworkError = -21001,

// The server returns an error code
TMAMiniAppErrorResponseCodeError= -21002,

// The server returns an empty response
TMAMiniAppErrorResponseDataNull = -21003,

// The mini program version does not exist
TMAMiniAppErrorVersionNotExist= -21005,

// The mini program has been removed from the shelves
TMAMiniAppErrorForbidden= -21006,

//The frequency of specific interface calls is too high
TMAMiniAppCmdIDRequestFrequent = -21011,

//Interface calls are too frequent
TMAMiniAppRequestFrequent = -21012,

// Unsupported link
TMAMiniAppErrorErrorLink = -21101,

// TCMPPExtScanCode extension library not integrated
// Unsupported link
TMAMiniAppErrorNoScanCodeSDK= -21102,

// delegate is not implemented
TMAMiniAppErrorNotImplemented = -21103,

// Mini program base library download failed
TMAMiniAppErrorJSSDKDownloadFail= -22001,
```

©2013-2022 Tencent Cloud. All rights reserved.

```
// Mini program base library decompression failed
TMAMiniAppErrorJSSDKUnzipFail = -22002,

// Mini program base library does not need to be upgraded
TMAMiniAppErrorJSSDKNeedUpgrade = -22003,

// Mini program basic library verification failed
TMAMiniAppErrorJSSDKVerifyFail= -22004,

TMAMiniAppErrorResponseInvalid= -22010,

// UnPack mini program Apkg package failed
TMAMiniAppErrorUnpackApkgFail = -22011,

TMAMiniAppErrorConsistencyInvalid = -22012,

TMAMiniAppErrorHTTPStatusInvalid= -22013,

// The download storage path of the mini program Apkg package is wrong
TMAMiniAppErrorCreatePathFail = -22014,

// The download storage path of the mini program Apkg package is wrong
TMAMiniAppErrorValidateFileFail = -22015,

// The device system does not support mini programs
TMAMiniAppErrorSystemNotSupport = -22017,

// AppInfo error when starting the mini program
TMAMiniAppErrorAppInfoError = -22018,

// The mini program is already being displayed
TMAMiniAppErrorMiniAppIsShowing = -22019,

// The mini program view is an invalid view
TMAMiniAppErrorParentVCInvalid= -22020,

// Application error when starting the mini program
TMAMiniAppErrorApplicationError = -22021,

// The agent for starting the mini program is not implemented

TMAMiniAppErrorDelegateError= -22022,
TMAMiniAppErrorNilApplication = -22023,
TMAMiniAppErrorUserNotLoggedIn= -22024,
TMAMiniAppErrorFlutterResourceInvalid = -22025,
```

©2013-2022 Tencent Cloud. All rights reserved.

```
// An exception occurs when the mini program reads the resource file (for example,
TMAMiniAppErrorReadPackageFail= -22027,

// Download file md5 verification failed
TMAMiniAppErrorFileMd5VerifyFail = -22029,

// Failed to copy domain name configuration file to target location
TMAMiniAppErrorFCopyDomainsFileFail = -22030,

// MAWebView white screen error
TMAMiniAppErrorWebViewNoCompositingView = -23001,
TMAMiniAppErrorWebViewJSError = -23002,
TMAMiniAppErrorWebViewDidTerminate= -23003,
TMAMiniAppErrorWebViewScriptAgainError= -23004,
TMAMiniAppErrorWebViewPageNotFound= -23005,
TMAMiniAppErrorWebViewApplicationNotFound = -23006,

TMAMiniAppErrorHotlaunchParamInvalid= -23007,
TMAMiniAppErrorWebViewRenderBlank = -23008,

// No available cloud resources to use
TMAMiniAppErrorResourceLimited = -25001,
TMAMiniAppErrorResourceOverflow = -25002
```

©2013-2022 Tencent Cloud. All rights reserved.

iOS Privacy Compliance

Last updated：2024-07-03 18:16:25

System permissions information involved in the mini program SDK:

SDK Name	System permissions involved	Permission usage description
TCMPPSDK	NSPhotoLibraryAddUsageDescription NSCameraUsageDescription NSMicrophoneUsageDescription	Used to realize the functions of taking pictures, recording audio, saving
TCMPPExtMedia	NSPhotoLibraryUsageDescription	Used to select photos and videos
TCMPPExtScanCode	NSCameraUsageDescription	Used to scan QR codes
TCMPPExtQMap	NSLocationAlwaysAndWhenInUseUsageDescription NSLocationUsageDescription NSLocationAlwaysUsageDescription NSLocationWhenInUseUsageDescription	Used to locate and use Tencent map services
TCMPPExtBaiduMap	NSLocationAlwaysAndWhenInUseUsageDescription NSLocationUsageDescription NSLocationAlwaysUsageDescription NSLocationWhenInUseUsageDescription	Used for positioning and using Baidu map service
TCMPPExtAMap	NSLocationAlwaysAndWhenInUseUsageDescription NSLocationUsageDescription NSLocationAlwaysUsageDescription NSLocationWhenInUseUsageDescription	Used for positioning and using Amap service
TCMPPExtLive	NSCameraUsageDescription NSMicrophoneUsageDescription	Used for implementing real-time audio and video recording function
TCMPPExtAuthentication	NSFaceIDUsageDescription	Used for using device biometric authentication function
TCMPPExtBLE	NSBluetoothAlwaysUsageDescription NSBluetoothPeripheralUsageDescription	Used to implement Bluetooth related

©2013-2022 Tencent Cloud. All rights reserved.

		functions, such as Bluetooth device connection, etc.
TCMPPExtCalendar	NSCalendarsUsageDescription NSRemindersUsageDescription	Used to implement date selection and adding reminders
TCMPPExtClipBoard	-	Used to access the pasteboard
TCMPPExtContact	NSContactsUsageDescription	Used to obtain and add contacts
TCMPPExtLBS	NSLocationAlwaysAndWhenInUseUsageDescription NSLocationUsageDescription NSLocationAlwaysUsageDescription NSLocationWhenInUseUsageDescription NSMotionUsageDescription	Used to implement positioning, system maps and motion-related functions
TCMPPExtMDNS	NSBonjourServices	Used to provide LAN communication capabilities
TCMPPExtNetwork	NSAppTransportSecurity NSAllowsArbitraryLoads NSBonjourServices	Used to provide TCP/UDP communication capabilities
TCMPPExtMp3Encoder	-	Used for audio MP3 format encoding

For more information, see Privacy Compliance.

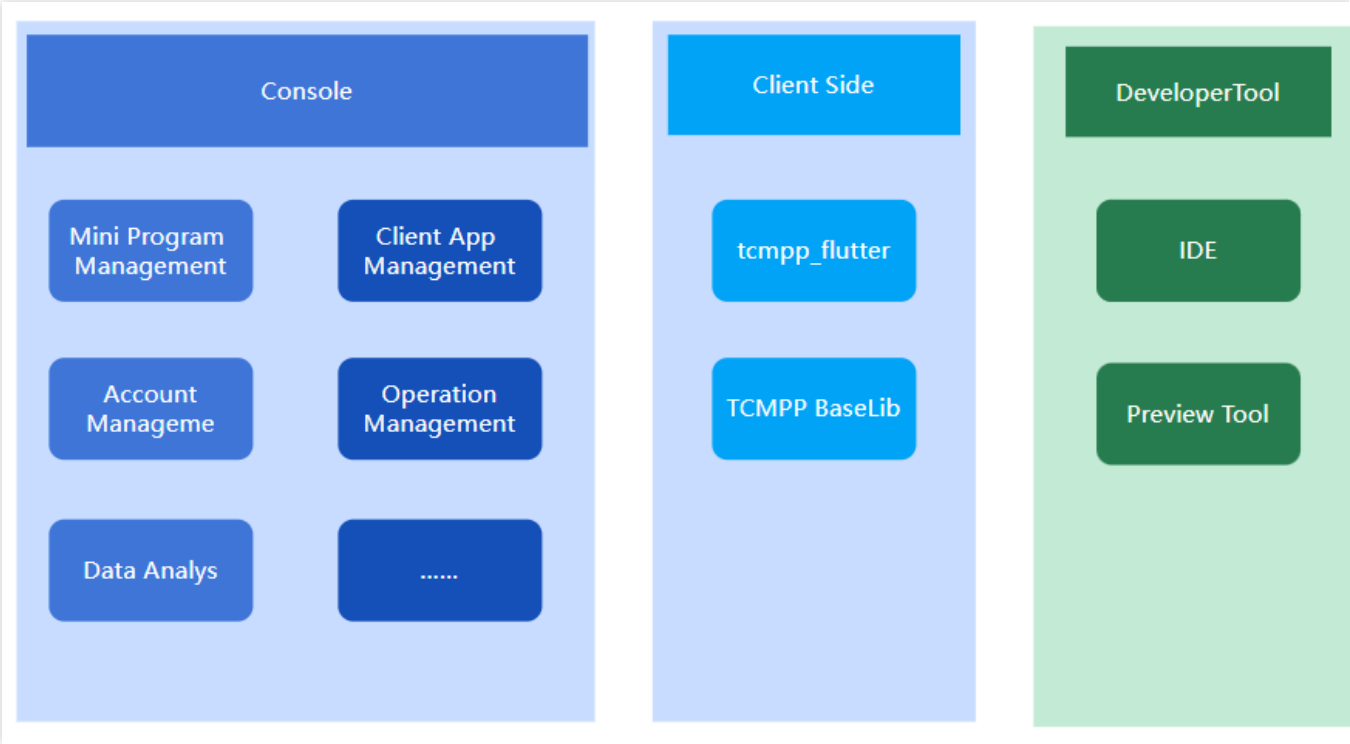
©2013-2022 Tencent Cloud. All rights reserved.

Flutter
Flutter SDK Description
SDK Overview

Last updated：2024-08-06 16:30:10

TCMPP

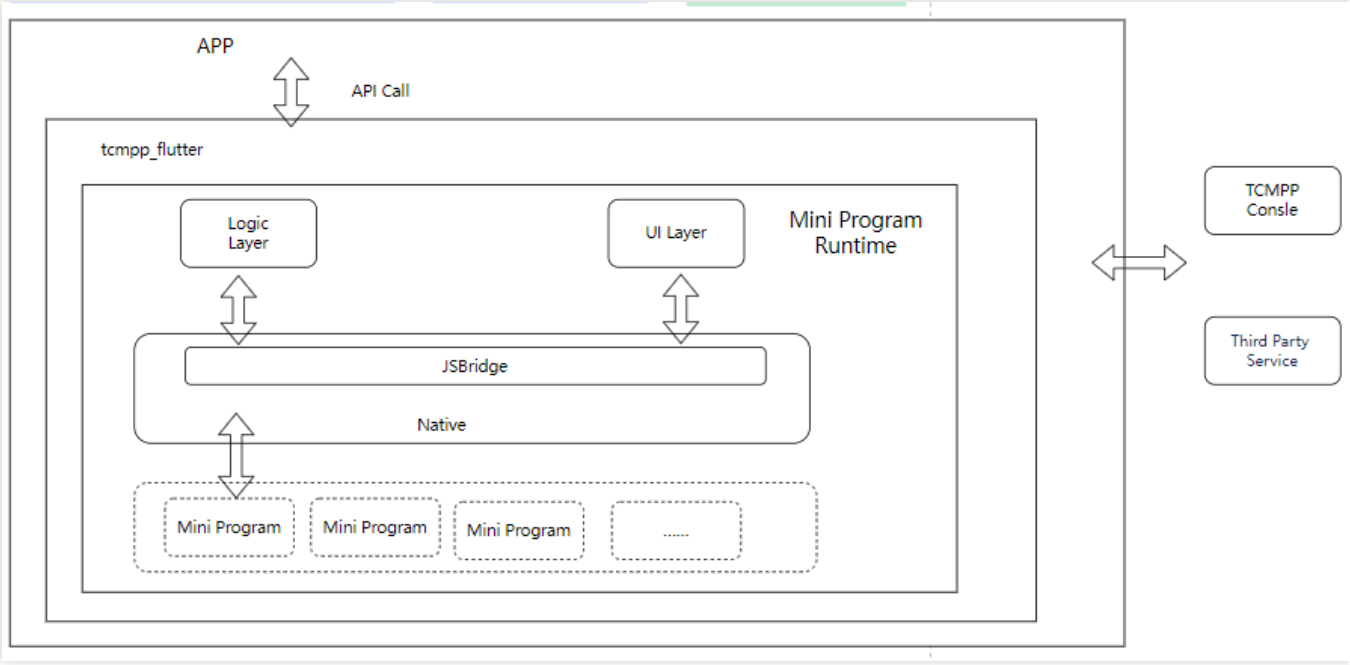
TCMPP consists of three parts: The console, tcmpp_flutter, and DevTools.



Console: The TCMPP console provides capabilities such as mini program management, client app management, and mini program operations management.
Client: The mobile client app that integrates tcmpp_flutter. tcmpp_flutter provides a runtime environment for mini programs in client apps.
DevTools: Use TCMPP IDE for development, debugging, and version submission, and the mini program preview assistant to preview on mobile.

How tcmpp_flutter works

©2013-2022 Tencent Cloud. All rights reserved.



tcmpp_flutter provides a runtime environment for mini programs in client apps. It communicates with the TCMPP backend to retrieve, load, and run mini-programs within its runtime environment.

To reduce package size and minimize system permissions, tcmpp_flutter is split into a core library (tcmpp_flutter) and extension libraries (tcmpp_flutter_xxx). Most features are available with just the core library, which adds about 4 MB to the final app package and requires no extra system permissions.

For details about the extension library, see Flutter - Extensions.

©2013-2022 Tencent Cloud. All rights reserved.

SDK Quick Integration

Last updated：2024-09-18 17:18:55

Sample code for integration.
Download stable versions of tcmpp_flutter.

Prerequisites

Environment requirements

sdk: '>=3.0.0 <4.0.0'

flutter: '>=3.3.0'

Plugin dependencies

plugin_platform_interface

flutter_plugin_android_lifecycle

Integration method

Adding TCMPP core plugins dependency

1. Depend on it: Open the pubspec.yaml file located inside the app folder, and add tcmpp_flutter: ${version} under dependencies.

```
23
24  # Dependencies specify other packages that your package needs in order to work.
25  # To automatically upgrade your package dependencies to the latest versions
26  # consider running `flutter pub upgrade --major-versions`. Alternatively,
27  # dependencies can be manually updated by changing the version numbers below to
28  # the latest version available on pub.dev. To see which dependencies have newer
29  # versions available, run `flutter pub outdated`.
30  dependencies:
31    flutter:
32      sdk: flutter
33
34    tcmpp_flutter: ^1.0.0
35
36
```

2. Install it:

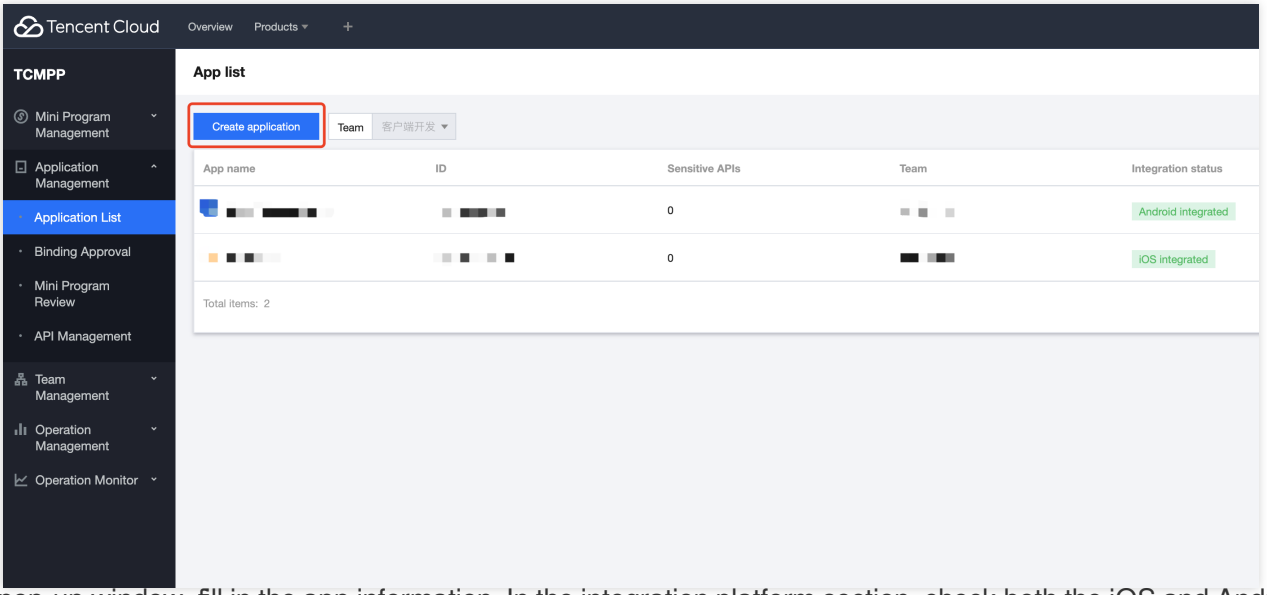
From the terminal: Run flutter pub get.

©2013-2022 Tencent Cloud. All rights reserved.

From VS Code: Click Get Packages located in right side of the action ribbon at the top of pubspec.yaml indicated by the Download icon.
From Android Studio/IntelliJ: Click Pub get in the action ribbon at the top of pubspec.yaml.
For details, see Packages & plugins > Using packages.

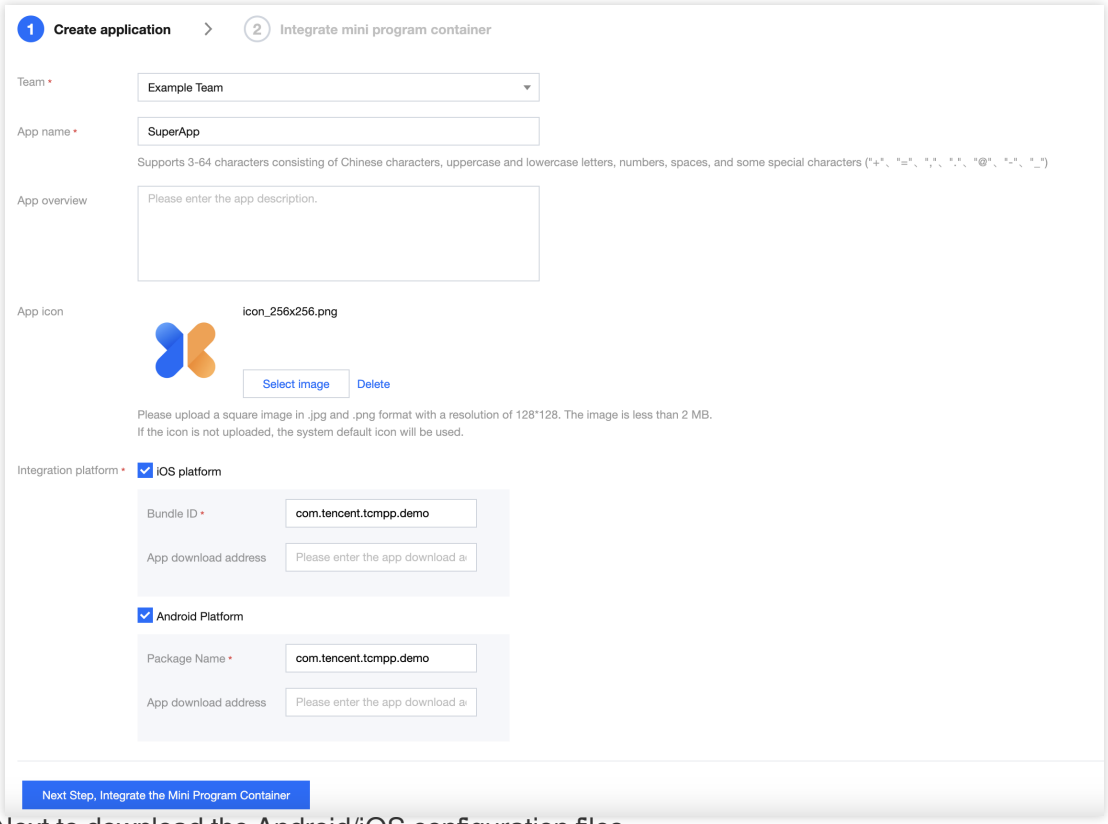
Obtaining the configuration file

To initialize the mini program SDK, first obtain the SDK configuration file from the mini program console.
Log in to the console, and click Create application.

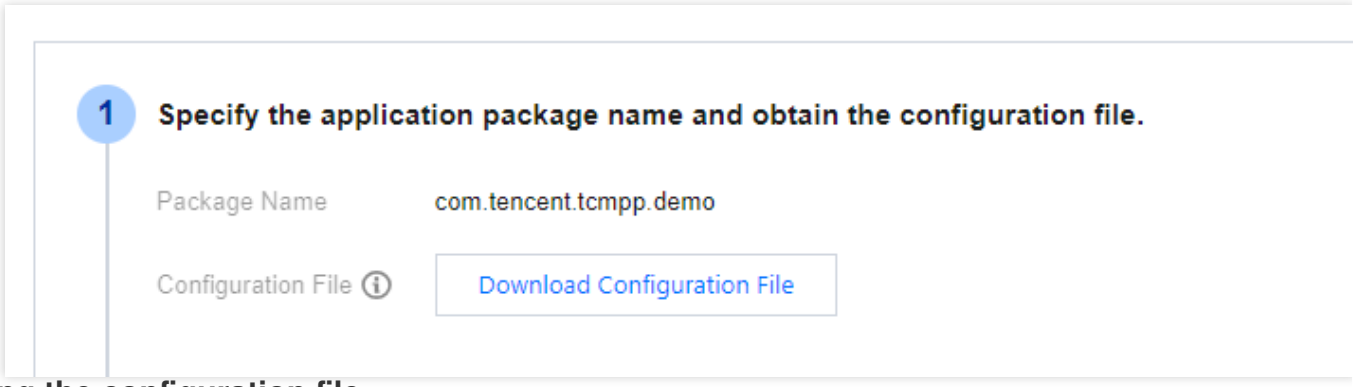


In the pop-up window, fill in the app information. In the integration platform section, check both the iOS and Android platforms. Enter the bundle ID or application ID in the package name field.

©2013-2022 Tencent Cloud. All rights reserved.



Click Next to download the Android/iOS configuration files.



Adding the configuration file

1. Place the TCMPP configuration file in the Flutter project.
2. Add a file named tcmpp-plugin-settings.json in the root directory of the project.
3. Edit the pubspec.yaml file to include the above files as a Flutter resource.

In the **tcmpp-plugin-settings.json** file, you can configure the SDK settings as needed.

Android:

Field	Type	Description
configAssetsName	string	Path to TCMPP configuration file in Android
debug	bool	Enable SDK debugging and logging

©2013-2022 Tencent Cloud. All rights reserved.

enableX5Core	bool	Use TBS service (X5 core) as the mini program runtime
x5LicenseKey	string	TBS license if the runtime is enabled
X5DocLicenseKey	string	TBS documentation license if the runtime is enabled
x5Core32Url	string	The download address of the 32-bit TBS runtime
x5Core64Url	string	The download address of the 64-bit TBS runtime

iOS:

Field	Type	Description
configAssetsName	string	Path to TCMPP configuration file in iOS
debug	bool	Enable SDK debugging
logEnable	bool	Enable SDK logging
inspectableEnabled	bool	Enable inspectable feature for iOS 16.4 and later, allowing mini program debugging via Safari

General:

Field	Type	Description
appName	string	Host app name, mainly for copyright notices in the mini program
appVersion	string	Host app version, mainly for copyright notices in the mini program

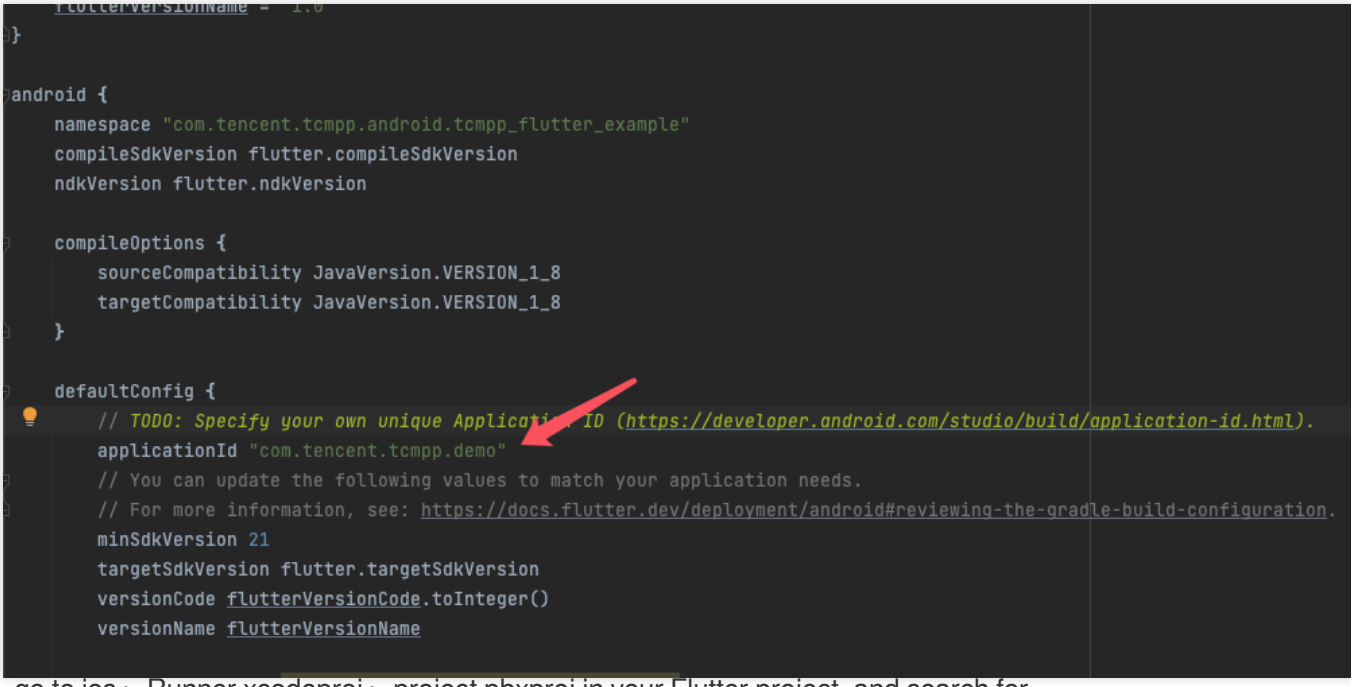
Sample

```
{
  "android": {
    "configAssetsName": "tcmpp-android-configurations.json",
    "debug": true,
    "enableX5Core": true,
    "x5LicenseKey": "",
    "x5DocLicenseKey": "",
    "x5Core32Url": "",
    "x5Core64Url": ""
  },
  "ios": {
    "configAssetsName": "tcmpp-ios-configurations.json",
    "debug": true,
    "logEnable": true,
```

©2013-2022 Tencent Cloud. All rights reserved.

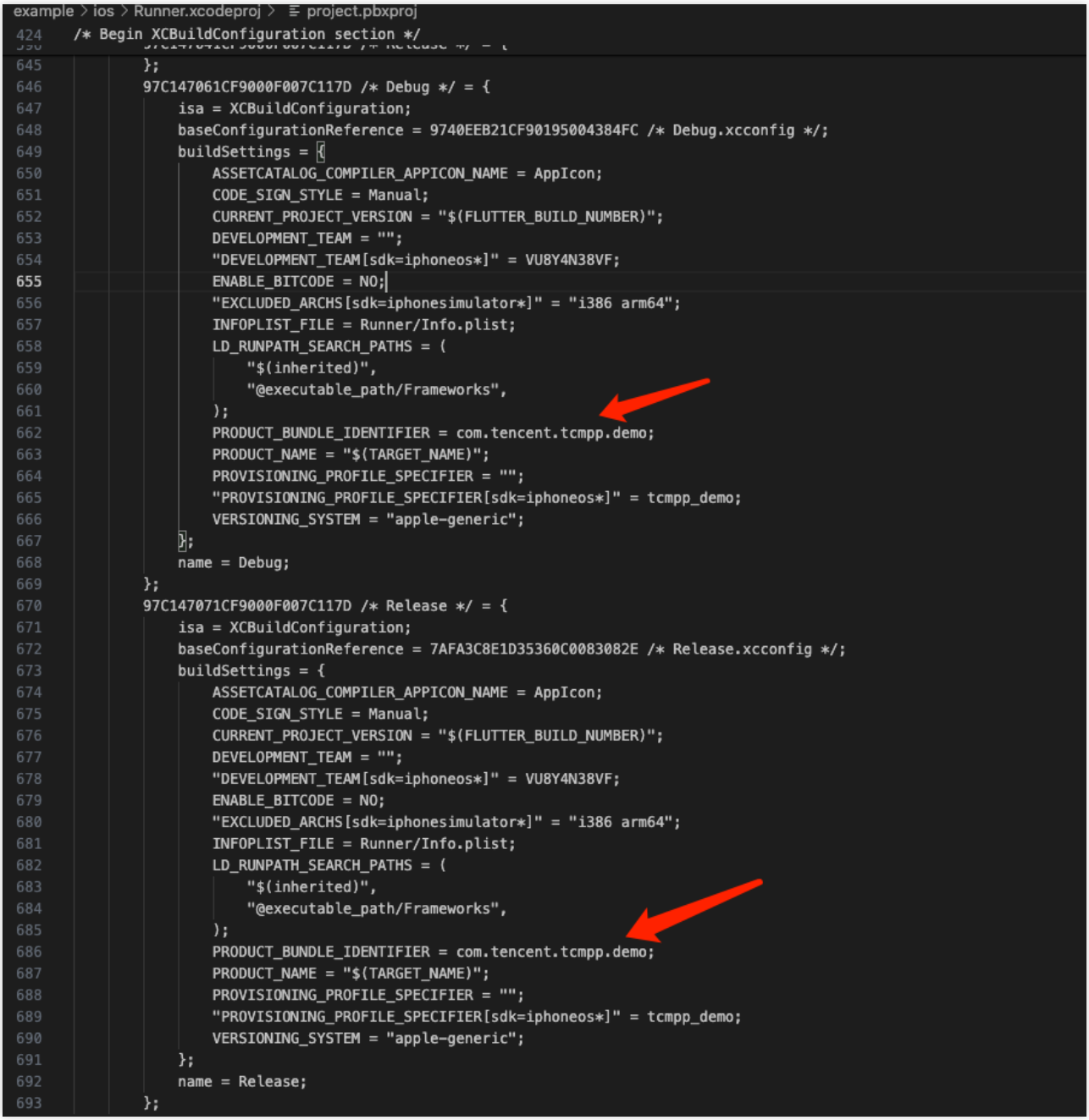
```
    "inspectableEnabled": true
  },
  "common":{
    "appName": "tcmpp-flutter",
    "appVersion": "1.0.0"
  }
}
```

In Android, go to android > app > build.gradle in your Flutter project, and the applicationId can be found in android > defaultConfig.



In iOS, go to ios > Runner.xcodeproj > project.pbxproj in your Flutter project, and search for PRODUCT_BUNDLE_IDENTIFIER.

©2013-2022 Tencent Cloud. All rights reserved.

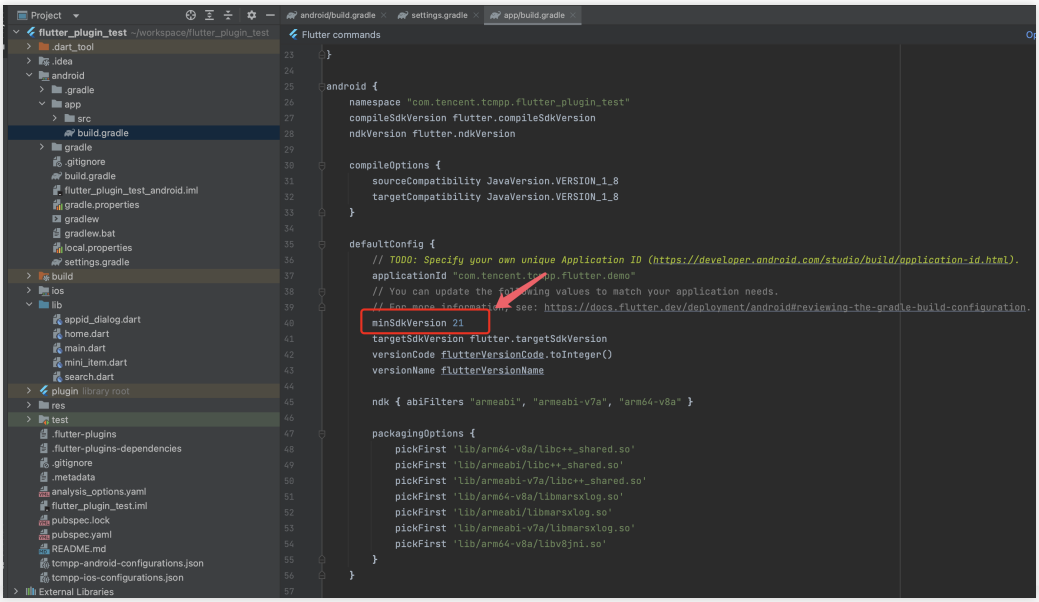


Platform-specific

For Android

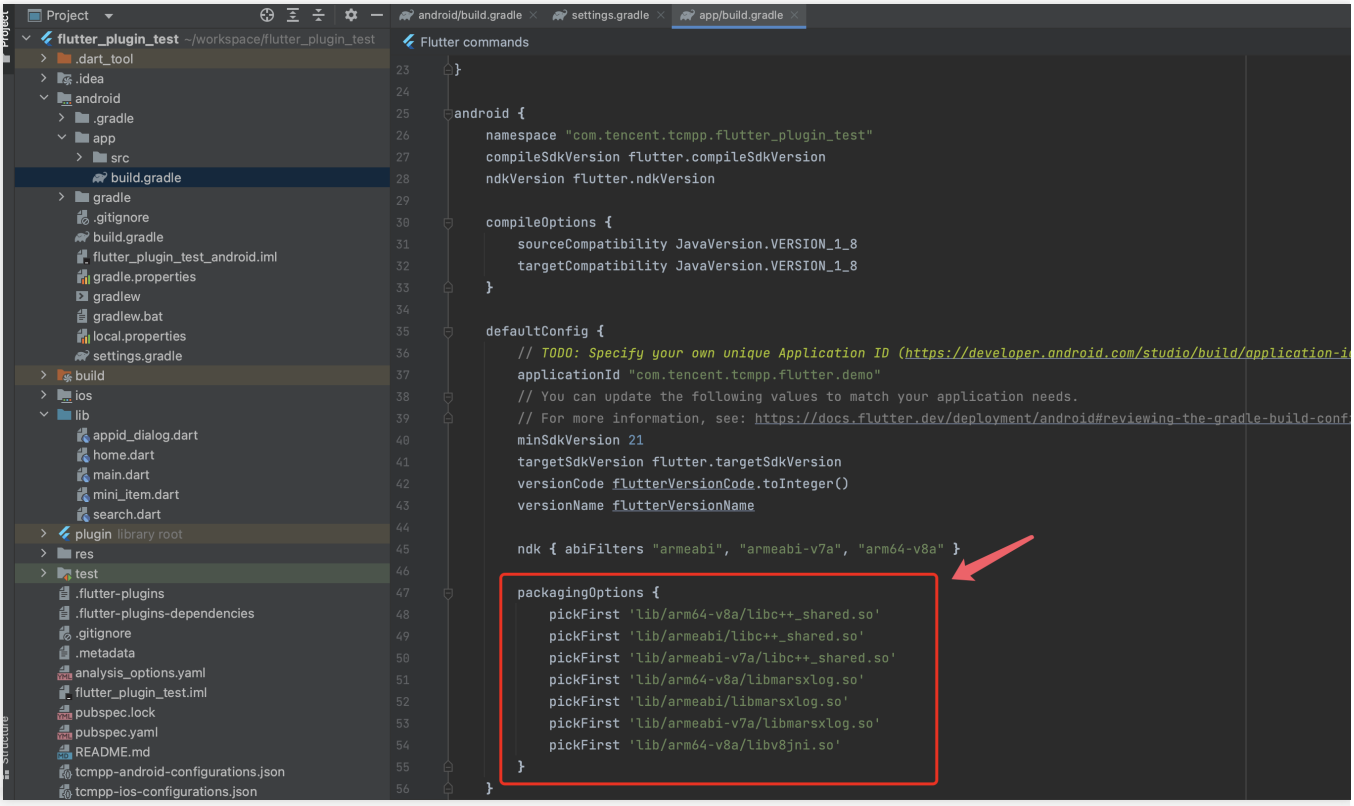
1. Go to android > app, and edit the build.gradle file.
2. In the android > defaultConfig section, set minSdkVersion to at least 21.

©2013-2022 Tencent Cloud. All rights reserved.



```
minSdkVersion 21
```

3. In the android > defaultConfig section, add packagingOptions.



```
packagingOptions {
    pickFirst 'lib/arm64-v8a/libc++_shared.so'
    pickFirst 'lib/armeabi/libc++_shared.so'
    pickFirst 'lib/armeabi-v7a/libc++_shared.so'
    pickFirst 'lib/arm64-v8a/libmarsxlog.so'
    pickFirst 'lib/armeabi/libmarsxlog.so'
```

©2013-2022 Tencent Cloud. All rights reserved.

```
    pickFirst 'lib/armeabi-v7a/libmarsxlog.so'
    pickFirst 'lib/arm64-v8a/libv8jni.so'
}
```

For iOS

Adding the source code

Add the source code in the Podfile located in the ios directory.

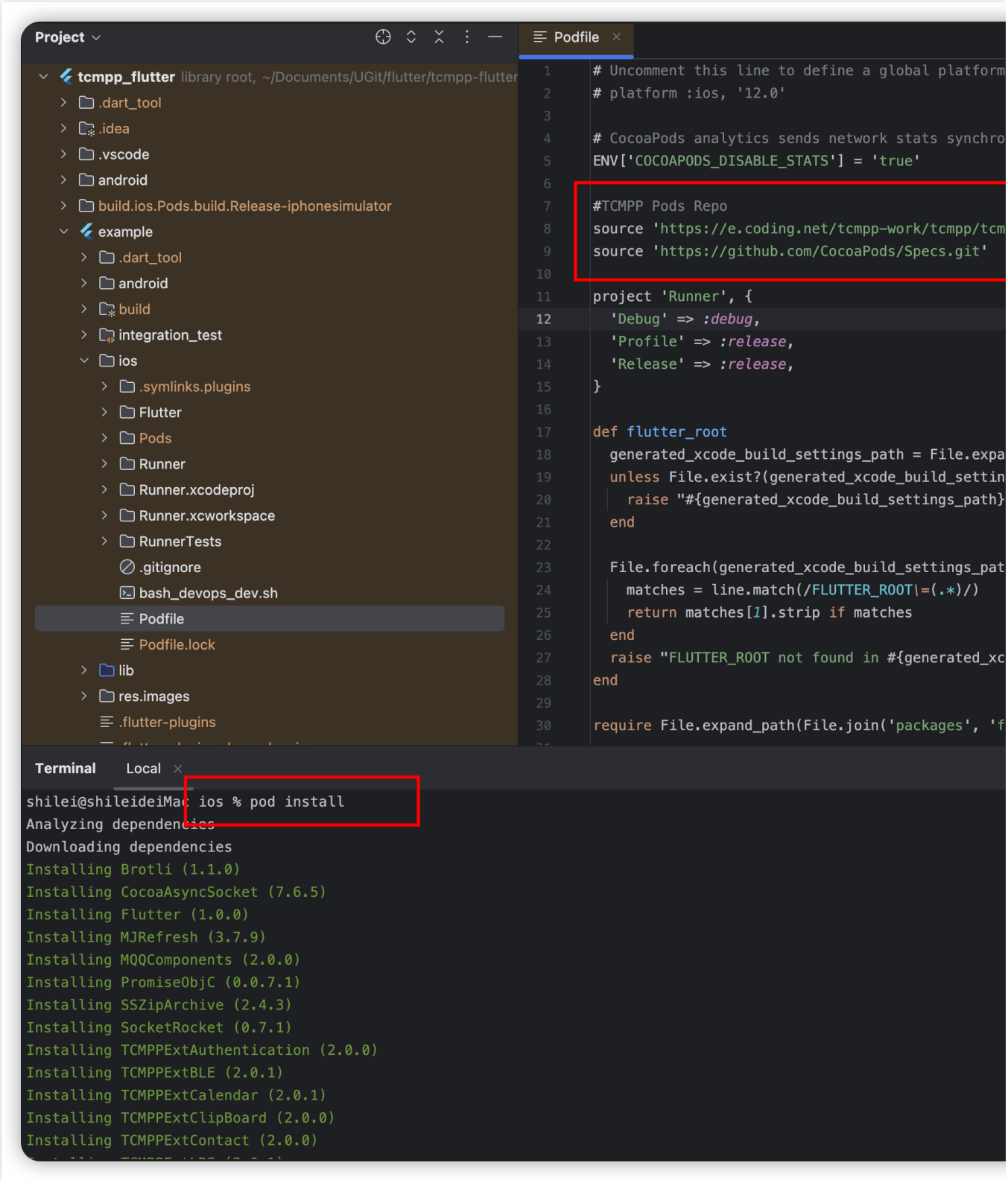
```
source 'https://e.coding.net/tcmpp-work/tcmpp/tcmpp-repo.git'
source 'https://github.com/CocoaPods/Specs.git'
```

Running pod install

Run pod install by navigating to the iOS directory and executing the command.

```
pod install
```

©2013-2022 Tencent Cloud. All rights reserved.



Opening the mini program

Run the command to import the TCMPP API: import 'package:tcmpp_flutter/tcmpp_flutter.dart'.

©2013-2022 Tencent Cloud. All rights reserved.

example > lib > main.dart > ...

```dart
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:tcmpp_flutter/tcmpp_flutter.dart';
import 'package:tcmpp_flutter_example/appid_dialog.dart';
import 'package:tcmpp_flutter_example/mini_item.dart';
import 'package:tcmpp_flutter_example/search.dart';

Run | Debug | Profile
void main() {
  runApp(const MyApp());
}

class MyApp extends StatefulWidget {
  const MyApp({super.key});

  @override
  State<MyApp> createState() => _MyAppState();
}

class _MyAppState extends State<MyApp> {
  @override
  Widget build(BuildContext context) {
    SystemChrome.setPreferredOrientations([
      DeviceOrientation.portraitUp,
    ]);
    return const MaterialApp(home: HomePage());
  }
}

class HomePage extends StatefulWidget {
  const HomePage({super.key});

  @override
  State<StatefulWidget> createState() => _HomePageState();
}
```

Create a TcmppFlutter object and call the TCMPP API with the object to open the mini program.

©2013-2022 Tencent Cloud. All rights reserved.

```dart
class _HomePageState extends State<HomePage> with WidgetsBindingObserver {
  final _tcmppFlutterPlugin = TcmppFlutter();
  final List<AppInfo> _appInfoList = [];

  @override
  void initState() {
    super.initState();
    WidgetsBinding.instance.addObserver(this);
    if (WidgetsBinding.instance.lifecycleState == AppLifecycleState.resumed) {
      getRecentMini();
    }
    _tcmppFlutterPlugin.registerOpenApiHandler(MyOpenApiHandler());
    _tcmppFlutterPlugin.registerPlatformEventHandler(MyPlatformHandler());
    _tcmppFlutterPlugin.registerMiniAppApi("testState", myApiHandler);
    _tcmppFlutterPlugin.registerMiniAppApi("myRequestPayment", myApiHandler1);
    _tcmppFlutterPlugin.setAccount(AccountInfo(
        uid: "12345",
        avatarUrl: "https://picsum.photos/250?image=9",
        accountName: "SimpleAccount"));
    _tcmppFlutterPlugin.setLocale("en", region: "us");
    _tcmppFlutterPlugin.setTheme(MiniTheme.dark);
  }

  @override
  void didChangeAppLifecycleState(AppLifecycleState state) {
    if (state == AppLifecycleState.resumed) {
      getRecentMini();
    }
  }
```

©2013-2022 Tencent Cloud. All rights reserved.

Common SDK Integration Issues

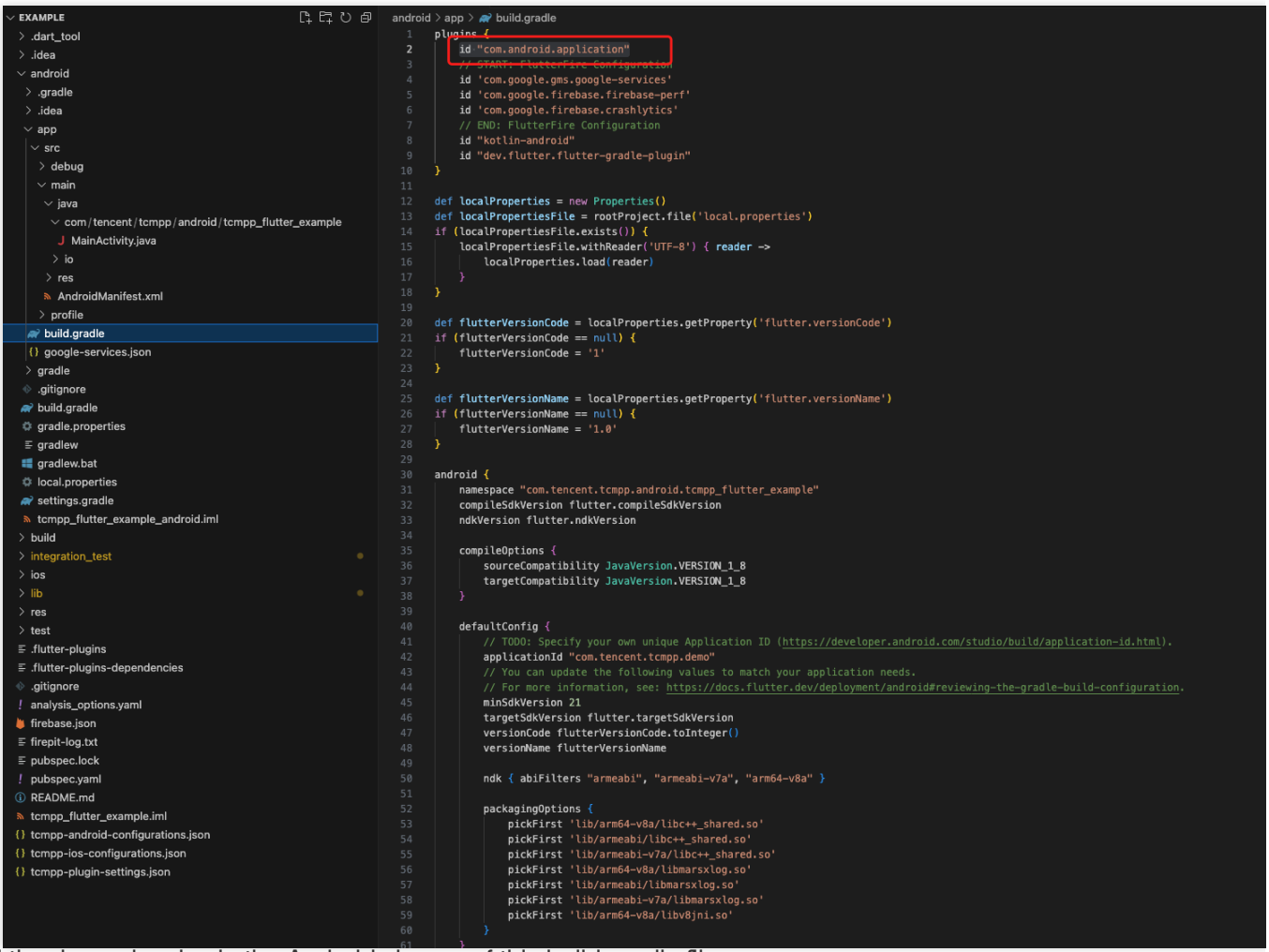
Last updated：2024-08-06 16:21:48

Crash issues with Firebase Flutter plugin

In Android, the Firebase Flutter plugin does not support multi-process handling by default, which may cause the TCMPP mini program subprocess to crash. To use TCMPP Flutter plugins with Firebase Flutter plugin, you need to initialize Firebase in the TCMPP mini program subprocess.

Adding the TCMPP dependency

1. Go to the root directory of the Flutter project. In the Flutter project, search `id "com.android.application"`
The build.gradle file containing this string is located in the root directory of your Flutter Android project.



2. Add the dependencies in the Android closure of this build.gradle file:

```
dependencies {
            compileOnly 'com.tencent.tcmpp.android:mini_core:+'
```

©2013-2022 Tencent Cloud. All rights reserved.

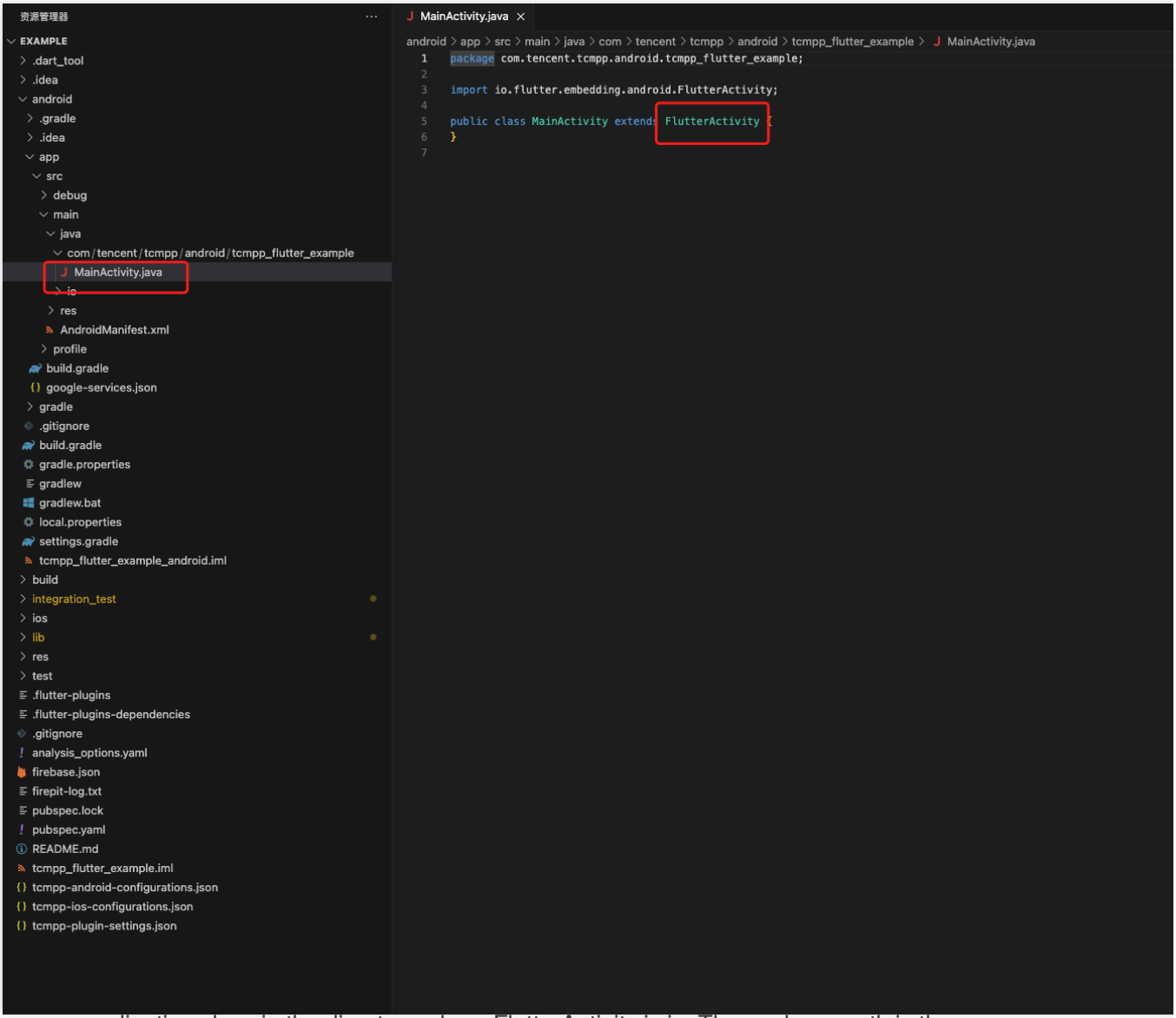
```
            compileOnly 'com.google.firebase:firebase-common:+'
        }
```

Creating an application class

Note：

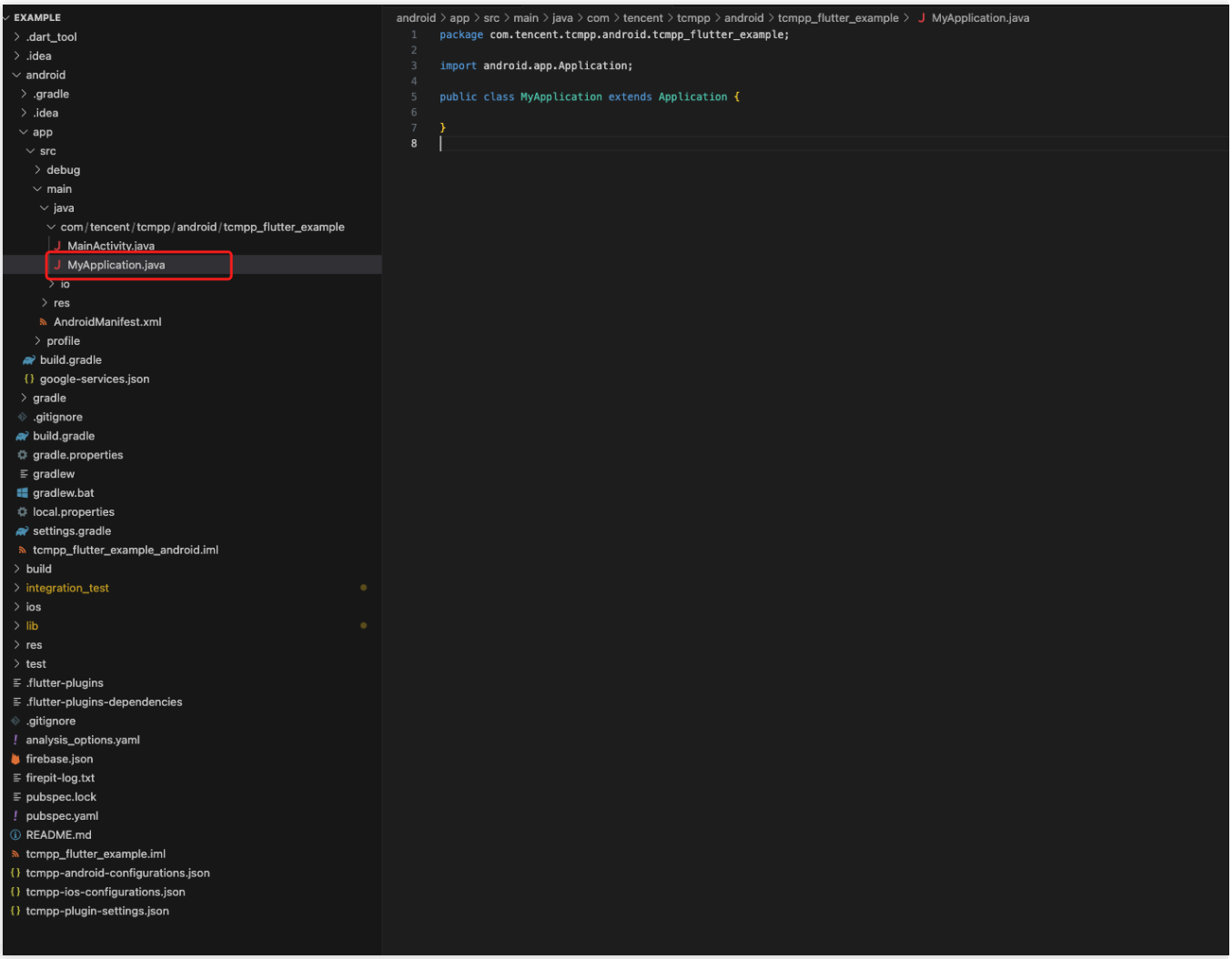
Skip this step if the Flutter project already has a custom Android application class.

In the src directory, locate FlutterActivity:



Create a new application class in the directory where FlutterActivity is in. The package path is the same as FlutterActivity:

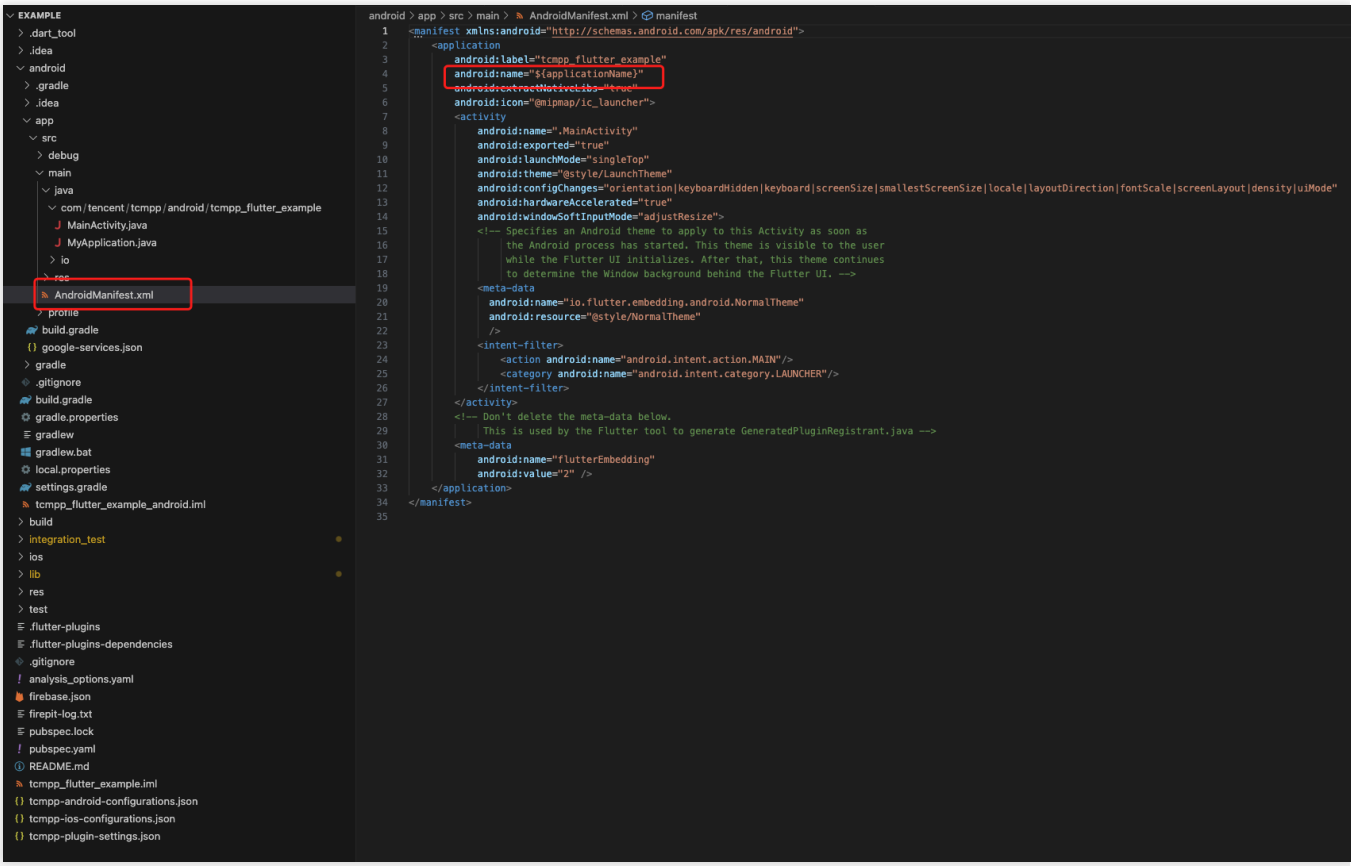
©2013-2022 Tencent Cloud. All rights reserved.




```
import android.app.Application;
public class MyApplication extends Application {}
```

Open and edit the AndroidManifest.xml under src > main. In the <application> tag, change the android:name to match your application class:

©2013-2022 Tencent Cloud. All rights reserved.




```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"><application
```

Initializing Firebase in the subprocess

1. Override the onCreate function in your Android application class. Skip this step if it is already done.

```
public void onCreate() {
        super.onCreate();
    }
```

2. Add the Firebase initialization code for the subprocess.

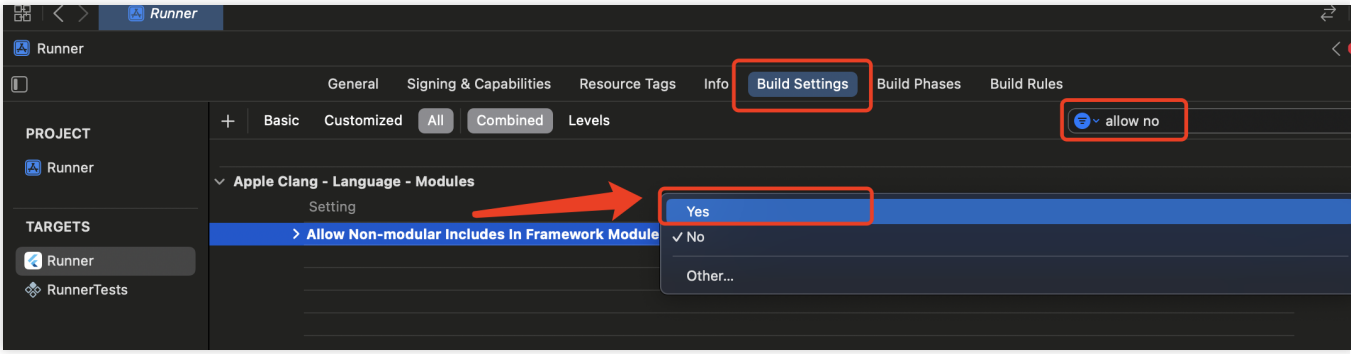
```
import android.app.Application;
    import com.google.firebase.FirebaseApp;
    import com.tencent.tmf.mini.api.TmfMiniSDK;

    public class MyApplication extends Application {

        @Override
        public void onCreate() {
            super.onCreate();

            ...
```

©2013-2022 Tencent Cloud. All rights reserved.

```
            if (TmfMiniSDK.isMiniProcess(this)) {
                FirebaseApp.initializeApp(this);
            }

            ...
        }
    }
```

The header file cannot be found in compilation in iOS

Open the project with Xcode, select the right target and go to Build Settings. Enter "Allow Non-modular Includes In Framework Modules" in the search box and set it to "Yes":



Apple M-series chip-based computer emulator issues

The current version of TCMPP SDK need to run via Rosetta on the M-series chip-based computer emulator.

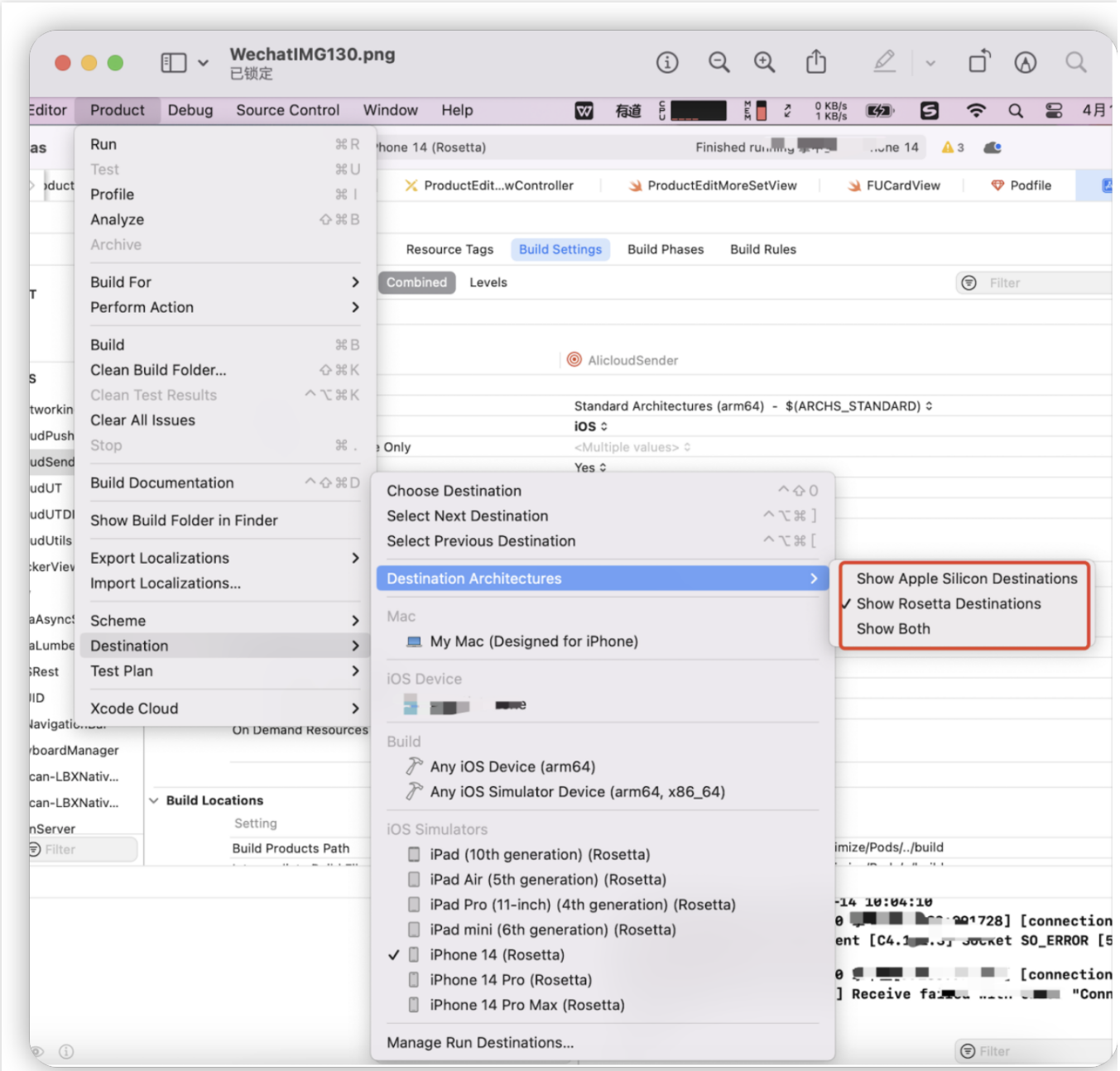
For versions earlier than Xcode14: Right-click Xcode -> Get Info, and check Open using Rosetta:

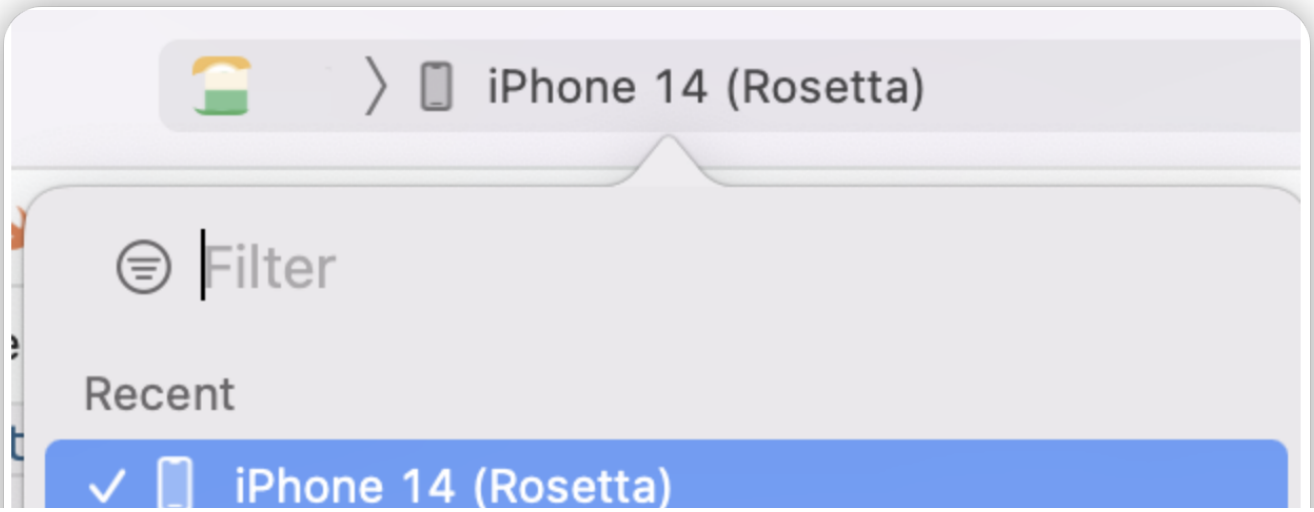
©2013-2022 Tencent Cloud. All rights reserved.

For versions later than Xcode14:

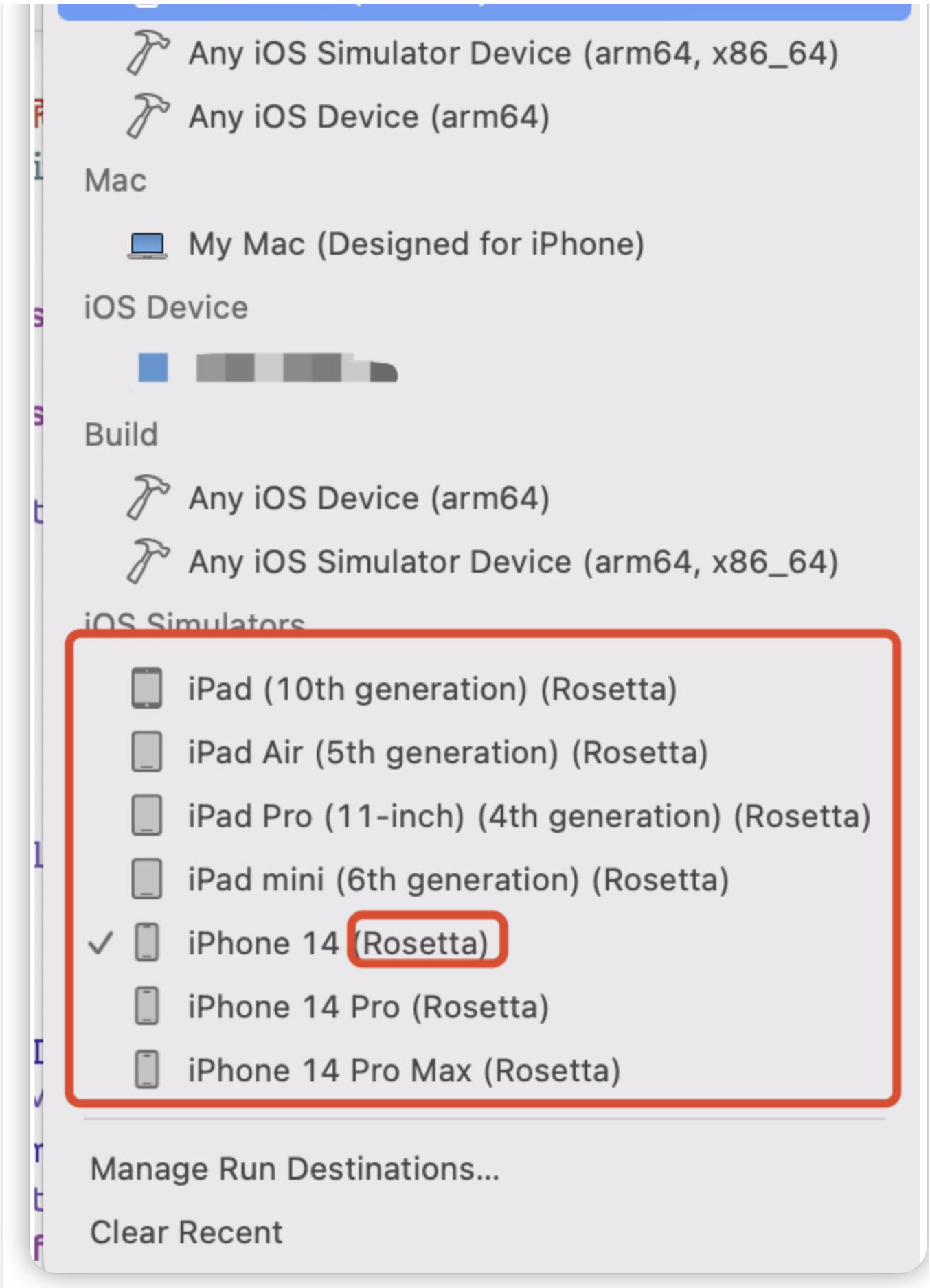
Open the project with Xcode -> Product -> Destination -> Destination Architecturesk. Choose Show Rosetta Destinations:

©2013-2022 Tencent Cloud. All rights reserved.





©2013-2022 Tencent Cloud. All rights reserved.



©2013-2022 Tencent Cloud. All rights reserved.

Flutter API
Mini Program Management API
Opening the Mini Program

Last updated：2024-08-06 16:00:50

Opening the mini program

When opening the mini program, the system will check whether there is a locally cached one. If not, it will automatically download the mini program from the remote server and then open it. If a cached version is available, the local mini program will be opened first while the system checks in the background for any new versions on the server.

Note：

If a new version is available, it will be downloaded, and the next time you open the mini program, the updated version will be used.

Start up the mini program with the given appId

appId: The ID of the mini program

Options: Startup options for the mini program

```
Future<void> startMiniAppWithId(String appId, MiniStartOptions? options)
```

Sample code:

```
 _tcmppFlutterPlugin.startMiniAppWithId(appId!, null);
```

Startup options

```
class MiniStartOptions {
  /// The path to enter the mini program
  String? entryPath;

  /// Always update the mini program at startup
  bool? isForceUpdate;

  /// String parameters passed to the mini program at startup
  String? params;
}
```

Start up the mini program via the given link

Link: The URI to start up the mini program

options: The startup options for the mini program

©2013-2022 Tencent Cloud. All rights reserved.

```
Future<void> startMiniAppWithLink(String link, MiniStartOptions? options)
```

Sample code:

```
_tcmppFlutterPlugin.startMiniAppWithLink(appId!, null);
```

©2013-2022 Tencent Cloud. All rights reserved.

Closing the Mini Program

Last updated：2024-08-06 15:59:43

Note：

Closing the mini program will clear it in memory. When the mini program is reopened, it needs to be loaded into memory again, which can take some time.

Stop the mini program with the given ID.

appId: The ID of the mini program:

```
Future<void> stopMiniApp(String appId)
```

Sample code:

```
 _tcmppFlutterPlugin.stopMiniApp(appId!)；
```

Stop all running mini programs:

```
Future<void> stopAllMiniApp()
```

Sample code:

```
 _tcmppFlutterPlugin.stopAllMiniApp()；
```

©2013-2022 Tencent Cloud. All rights reserved.

Deleting the Mini Program

Last updated：2024-08-06 15:56:25

Note：

The mini program package and information is cached locally for faster opening of the mini program next time. If you want to delete all information of the mini program, you can call the following API to delete a mini program or all mini programs.

Deletes all local data of the mini program with a given ID:

appId: The ID of the mini program

appVerType: The type of the mini program

version: The version of the mini program to delete

```
Future<void> deleteMiniAppCache(String appId,
   {int appVerType = 0, String? version})
```

appVerType

```
/// Constants for the mini program package types
class AppVerType {
  static const int online = 0;
  static const int develop = 1;
  static const int preview = 2;
  static const int experience = 3;
}
```

Sample code:

```
 _tcmppFlutterPlugin.deleteMiniAppCache(appId!,0,'1.0.0')；
```

©2013-2022 Tencent Cloud. All rights reserved.

Searching for the Mini Program

Last updated：2024-08-06 15:55:59

Search for an online mini program with the given keywords：
keyWord: The keyword of the mini program to search.
pageIndex: The page returned if the value of pageSize is not zero, that is, the pagination feature is enabled (for Android only).
pageSize: The maximum size of the page returned by the server. When set to 0, the pagination feature is disabled (for Android only).Obtaining the List of Recently Accessed Mini Programs.

```
Future<List<AppInfo>?> searchMiniApp(String keyword,
   {int pageIndex = 0, int pageSize = 0})
```

Sample code:

```
Future<void> doSearch() async {
    List<AppInfo>? infoList;
    FocusScope.of(context).unfocus();
    infoList = await _tcmppFlutterPlugin
        .searchMiniApp(searchController.text)
        .catchError((err) {
      ScaffoldMessenger.of(context).showSnackBar(SnackBar(content: Text(err.toStrin
    });
    setState(() {
      _appInfoList.clear();
      if (infoList != null && infoList.isNotEmpty) {
        _appInfoList.addAll(infoList);
      } else {
        _emptyText = 'No mini app found';
      }
    });
  }
```

©2013-2022 Tencent Cloud. All rights reserved.

Get a list of recent visits to the mini program

Last updated：2024-08-06 15:52:10

The tcmpp_flutter plugin provides the following API for accessing the list of recently used mini programs.

```
Future<List<AppInfo>?> getRecentList()
```

Sample code:

```
_tcmppFlutterPlugin.getRecentList();
```

©2013-2022 Tencent Cloud. All rights reserved.

Pre-downloading a Mini Program

Last updated：2024-08-06 15:51:49

Since the mini program checks for updates and synchronizes versions upon opening, there may be a waiting period. To minimize the startup time and enhance the user experience, the following API is provided to pre-download the mini program packages.

appIdList: A list of mini program IDs to be loaded.

isDownload: If set to "true", the mini program package will be downloaded after preloading. The default value is "false".

```
Future<void> preDownloadMiniApp(List<String> appIdList,
   {bool isDownload = false})
```

Sample code:

```
List<String> appIdList = ['app_id_1', 'app_id_2', 'app_id_3'];
_tcmppFlutterPlugin.preDownloadMiniApp(appIdList, isDownload: true);
```

©2013-2022 Tencent Cloud. All rights reserved.

Mini Program Capabilities Customization
Customizing Mini Program APIs

Last updated：2024-09-10 18:52:52

If a mini program needs capabilities from the host app that the SDK doesn't support, developers can register custom APIs to enable these capabilities. This allows the mini program to call custom APIs provided by the host application. Define a custom mini program API and associate it with an API handler. apiName: The name of the custom API. The mini program can use this API by calling wx.invokeNativePlugin().
apiHandler: The function that handles the API calls.

```
void registerMiniAppApi(String apiName, TcmppMiniAppApiHandler apiHandler)
```

Sample code:

```
final _tcmppFlutterPlugin = TcmppFlutter();

...

@override
void initState() {
   super.initState();

   ...

   /// The custom APIs must be registered prior to mini program startup
   _tcmppFlutterPlugin.registerMiniAppApi("myApiName",myApiHandler);
}
```

The mini program uses wx.invokeNativePlugin with api_name to call the Flutter-registered myApiName.

Sample code:

```
example code:
   /// Mini Program Call
 var opts = {
 api_name: 'myApiName',
 success: function(res) {
   log(res);
 },
 fail: function(res) {
   log(res);
 },
 complete: function(res) {
   log(res);
```

©2013-2022 Tencent Cloud. All rights reserved.

```
  },
  data: {
      name : 'kka',
      age : 22
   }
  }
  wx.invokeNativePlugin(opts);
```

Implement the myApiHandler function on the Flutter side:

```
/// client API
Future<Map<String, dynamic>?> myApiHandler(MiniApiCall call) async {
    print("API : ${call.apiName}");
    print("AppInfo: ${call.appInfo}");
    print("WebView ID: ${call.webViewId}");
    print("params: ${call.params}");


    return {"result": "success","method":"myApiHandler"};
}
```

©2013-2022 Tencent Cloud. All rights reserved.

Plugin Customization
Customizing Plugin Capabilities

Last updated：2024-08-06 15:44:43

You can associate the data interaction between the mini program and the host app by implementing the OpenApiHandler and TcmppPlatformEventHandler abstract classes.

Sample code:

```
_tcmppFlutterPlugin.registerOpenApiHandler(MyOpenApiHandler());
_tcmppFlutterPlugin.registerPlatformEventHandler(MyPlatformHandler());
```

Sample code for API class implementation:

```
class MyPlatformHandler extends TcmppPlatformEventHandler {


  @override
  Future<List<CustomMenu>> getCustomMenus() async {
    CustomMenu menu1 = CustomMenu(
        '100', 'res/images/mini_app_wechat_friend.png', 'Share To', true,
        shareKey: 'twitter');
    CustomMenu menu2 = CustomMenu(
        '101',
        'https://staticintl.cloudcachetci.com/cms/backend-cms/8WGP653_%E5%BC%80%E5%
        'Custom',
        false);


    return [
      menu1,
      menu2,
    ];
  }


  @override
  Future<void> customMenuClick(String menuId, ShareData? shareMenu) async {
    print("click menuId:$menuId shareMenu:$shareMenu");
    throw UnimplementedError();
  }


  @override
  Future<bool> reportEvent(int eventId, String eventName, AppInfo appInfo,
```

©2013-2022 Tencent Cloud. All rights reserved.

```
      Map<Object?, Object?> params) async {
    print("reportEvent:$eventName  appinfo:$appInfo params:$params");
    // TODO: implement reportEvent
    return true;
  }


  @override
  Future<void> onMiniProgramStateChange(
      String appId, MiniProgramState state) async {
    print("app state change: appid=$appId, state=$state");
  }
}
```

©2013-2022 Tencent Cloud. All rights reserved.

Customizing Mini Program Information API

Last updated：2024-08-06 15:36:18

Monitoring mini program lifecycle

The Flutter app host can receive notifications by overriding the onMiniProgramStateChange method when the mini program starts, enters the foreground, background, or closes.
Sample code:

```
@override
Future<void> onMiniProgramStateChange(
    String appId, MiniProgramState state) async {
  print("app state change: appid=$appId, state=$state");
}
```

©2013-2022 Tencent Cloud. All rights reserved.

Customizing the Sharing Capability

Last updated：2024-08-06 15:35:33

Developers can customize the buttons on the operation panel of the mini program, including the sharing button and other buttons. The operation panel can be brought up by clicking the More button in the top right corner of the mini program:

```
@override
Future<List<CustomMenu>> getCustomMenus() async {
/// The menu contains menuId, image (local paths and network images are supported),
  CustomMenu menu1 = CustomMenu(
      '100', 'res/images/mini_app_wechat_friend.png', 'Share To', true,
      shareKey: 'twitter');
  CustomMenu menu2 = CustomMenu(
      '101', 'https://staticintl.cloudcachetci.com/cms/backend-cms/8WGP653_%E5%BC%8


  return [
    menu1,
    menu2,
  ];
}
```

Menu click callback:

```
@override
Future<void> customMenuClick(String menuId, ShareData? shareMenu) async {
  /// Callback for menu button click
  print("click menuId:$menuId shareMenu:$shareMenu");
  throw UnimplementedError();
}
```

The menu has built-in buttons for four sharing channels: QQ, Qzone, WeChat, and WeChat Circle of Friends. If you want to display these sharing buttons, you can configure them in the following ways.

```
_tcmppFlutterPlugin.defaultShareTargets([ShareTarget.qq, ShareTarget.wxMoment]);
```

©2013-2022 Tencent Cloud. All rights reserved.

Customizing User Attributes

Last updated：2024-08-06 15:32:03

Set the current login account for data isolation and user information display.

The account must be set before mini program startup.

info: The information of the current login account. It is left empty if no user is logged in.

```
void setAccount(AccountInfo? info)
```

Sample code:

```
_tcmppFlutterPlugin.setAccount(AccountInfo(
        uid: "12345",
        avatarUrl: "https://picsum.photos/250?image=9",
        accountName: "SimpleAccount"));
```

©2013-2022 Tencent Cloud. All rights reserved.

Event Reporting

Last updated：2024-08-06 15:24:26

The event reporting API can be implemented in the host app to override the reporting logic in tcmpp_flutter.

```
Future<bool> reportEvent(int eventId, String eventName, AppInfo appInfo,
      Map<Object?, Object?> params) async {
    return false;
  }
```

Sample code:

```
@override
  Future<bool> reportEvent(int eventId, String eventName, AppInfo appInfo,
      Map<Object?, Object?> params) async {
    print("reportEvent:$eventName  appinfo:$appInfo params:$params");
    // TODO: implement reportEvent
    return true;
  }
```

©2013-2022 Tencent Cloud. All rights reserved.

Open APIs

Last updated：2024-08-06 15:23:50

The mini program SDK provides some open APIs for implementing capabilities such as login, obtaining user information, and payment, which are provided by host apps. See the following table for the supported open APIs:

Mini program	MiniOpenApiProxy	Description
wx.login	login	Login API
wx.getUserInfo	getUserInfo	Obtain basic user information
wx.getUserProfile	getUserProfile	Obtain user attribute information
wx.getPhoneNumber	getPhoneNumber	Obtain phone number
wx.requestPayment	requestPayment	Initiate a payment
wx.checkSession	checkSession	Check if the login has expired

You can associate data interactions between the mini program and the host application by implementing the OpenApiHandler abstract class

```
abstract class OpenApiHandler {
  // The API is called when the mini program calls wx.requestPayment to request a t

  Future<Map<String, dynamic>> requestPayment(
      AppInfo appInfo, Map<Object?, Object?> params);

  /// The API is called when the mini program calls wx.getUserProfile to request th
  ///
  Future<Map<String, dynamic>> getUserProfile(
      AppInfo appInfo, Map<Object?, Object?> params);

  /// The API is called when the mini program calls wx.login to request the login c
  ///
  Future<Map<String, dynamic>> login(
      AppInfo appInfo, Map<Object?, Object?> params);

  /// The API is called when the mini program calls wx.checkSession to request the
  /// Check if the login has expired
  ///
  Future<Map<String, dynamic>> checkSession(
      AppInfo appInfo, Map<Object?, Object?> params);
```

©2013-2022 Tencent Cloud. All rights reserved.

```
  /// The API is called when the mini program calls wx.getUserProfile
  /// Compatible with earlier mini program APIs
  ///
  Future<Map<String, dynamic>> getUserInfo(
      AppInfo appInfo, Map<Object?, Object?> params);

  /// The API is called when the mini program calls wx.getPhoneNumber to request th
  ///
  Future<Map<String, dynamic>> getPhoneNumber(
      AppInfo appInfo, Map<Object?, Object?> params);
}
```

Sample code for obtaining user information instance:

```
@override
  Future<Map<String, dynamic>> getUserProfile(
      AppInfo appInfo, Map<Object?, Object?> params) async {
    print("getUserProfile:$appInfo params:$params");
    Map<String, dynamic> result = {
      "userInfo": {
        "nickName": "xcode",
        "avatarUrl":
            "https://staticintl.cloudcachetci.com/cms/backend-cms/8WGP653_%E5%BC%80
        "gender": 1,
        "country": "China",
        "province": "ChongQing",
        "city": "ChongQing",
      }
    };
    return Future.value(result);
  }
```

©2013-2022 Tencent Cloud. All rights reserved.

Others

Last updated：2024-08-06 15:21:02

Setting the topic for the mini program container

The topic must be set before mini program startup. You can choose light mode, dark mode, or using default settings.

```
Future<void> setTheme(MiniTheme theme)
```

Sample code:

```
_tcmppFlutterPlugin.setTheme(MiniTheme.dark);
```

Localization setting for the mini program

It must be set before the mini program startup.

language: The language used in the mini program. The ISO 639 alpha-2 or alpha-3 language code should be used.

The ISO 3166 alpha-2 country code should be used.

variant: An arbitrary value used to represent a variant of the local language.

```
Future<void> setLocale(String language,
      {String? region, String? variant})
```

Sample code:

```
_tcmppFlutterPlugin.setLocale("en", region: "us");
```

©2013-2022 Tencent Cloud. All rights reserved.

API Description

Last updated：2024-08-06 15:17:13

MiniStartOptions

```
class MiniStartOptions {
  /// Entry path to the mini program.
  String? entryPath;

  /// Always update the mini program on startup.
  bool? isForceUpdate;

  /// String parameters to pass to the mini program on
  String? params;
}
```

ScanResult

```
class ScanResult {
  /// The result string for a qrcode or barcode contains
  String? result;

  /// code type
  String? scanType;

  /// Character encoding of the result string
  String? charset;
}
```

AppInfo

```
class AppInfo {
  /// Mini program id.
  String appId;

  /// The mini program package type (distribution, development, etc.). See [AppVerT
```

©2013-2022 Tencent Cloud. All rights reserved.

```
  int appVerType;

  /// The mini program version.
  String version;

  /// Name of the mini program.
  String? name;

  /// The mini program icon link.
  String? iconUrl;

  /// Description of the mini program.
  String? appIntro;

  /// Mini program developer.
  String? appDeveloper;

  /// Mini program release time.
  int time;
}
```

AppVerType

```
/// Constants for mini program package types, see [AppInfo].
class AppVerType {
  static const int online = 0;
  static const int develop = 1;
  static const int preview = 2;
  static const int experience = 3;
}
```

AccountInfo

```
class AccountInfo {
  /// Unique ID of the current account
  String? uid;

  /// Link to the avatar
  String? avatarUrl;
```

©2013-2022 Tencent Cloud. All rights reserved.

```
  /// Current account name
  String? accountName;
}
```

OpenApiHandler

```
abstract class OpenApiHandler {
  // Called when the mini program calls wx.requestPayment to request a third-party
  Future<Map<String, dynamic>> requestPayment(
      AppInfo appInfo, Map<Object?, Object?> params);

  //// Called when the mini program calls wx.getUserProfile to request user informa
  ///
  Future<Map<String, dynamic>> getUserProfile(
      AppInfo appInfo, Map<Object?, Object?> params);

   /// Called when the mini program calls wx.login, requesting the host application
  ///
  Future<Map<String, dynamic>> login(
      AppInfo appInfo, Map<Object?, Object?> params);

  /// Called when the mini program calls wx.checkSession, requesting the host appli
  /// Checks if the login has expired
  ///
  Future<Map<String, dynamic>> checkSession(
      AppInfo appInfo, Map<Object?, Object?> params);

  /// Called when the mini program calls wx.getUserInfo, which has been outdated by
  /// Compatible with earlier mini program api.
  ///
  Future<Map<String, dynamic>> getUserInfo(
      AppInfo appInfo, Map<Object?, Object?> params);

  /// Called when the mini program calls wx.getPhoneNumber to get the current user'
  ///
  Future<Map<String, dynamic>> getPhoneNumber(
      AppInfo appInfo, Map<Object?, Object?> params);
}
```

TcmppPlatformEventHandler

©2013-2022 Tencent Cloud. All rights reserved.

```
abstract class TcmppPlatformEventHandler {


  Future<String> getAppName() async {
    return "";
  }

  Future<String> getAppVersion() async {
    return "";
  }

  Future<List<CustomMenu>> getCustomMenus() async {
    return [];
  }


  Future<void> customMenuClick(String menuId, ShareData? shareMenu) async {}

  Future<void> onMiniProgramStateChange(
      String appId, MiniProgramState state) async {}


  Future<bool> reportEvent(int eventId, String eventName, AppInfo appInfo,
      Map<Object?, Object?> params) async {
    return false;
  }
}
```

©2013-2022 Tencent Cloud. All rights reserved.

Extensions

Last updated：2024-08-06 15:20:32

Many APIs that require system authorization for development and use need to be pre-authorized in the info.plistfile of iOS and the AndroidManifest.xmlfile of Android. However, your app may not need this feature, so tcmpp_flutter splits out the extension SDK, which eliminates unnecessary authorization and reduces the size of the core module. tcmpp_flutter provides the core module and the expansion module for users to access as needed.
Some TCMPP mini program APIs may require additional privacy and permissions. To use these APIs, additional Flutter plugin dependencies are required.

APIs	Plugin names	Required permissions
LBS-related APIs (location and POI search)	tcmpp_flutter_lbs	Access location
MDNS APIs	tcmpp_flutter_mdns	Access local network
TCP/UDP APIs	tcmpp_flutter_network	Access network
Media APIs (images & videos)	tcmpp_flutter_media	Access image library
Wireless API, bluetooth API, calendar API, contacts API, clipboard API, biometric authentication API	tcmpp_flutter_device	Wireless API, bluetooth API, calendar API, contacts API, clipboard API, biometric authentication API
Map APIs	tcmpp_flutter_googlemap (for general) tcmpp_flutter_petalmap (for Huawei device)	Access location

These plugins must be used together with tcmpp_flutter plugins. Map APIs are required only in Android, and they are included in iOS by default.

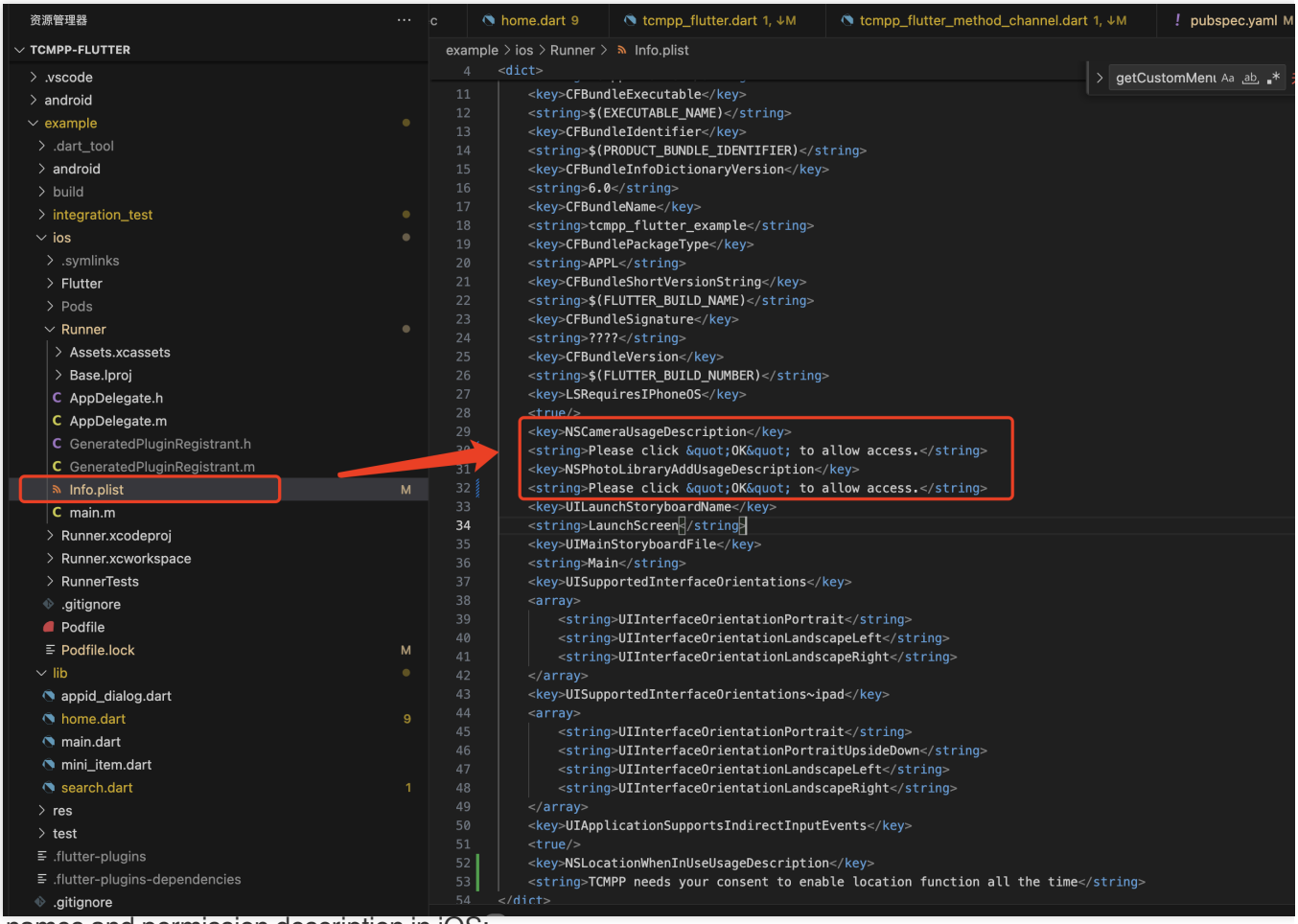
©2013-2022 Tencent Cloud. All rights reserved.

Flutter Privacy Compliance

Last updated：2024-08-06 15:18:18

Note：

Permission descriptions should be provided in the plist file in iOS.



Plugin names and permission description in iOS:

Plugin name	System permissions
tcmpp_flutter_lbs	NSLocationAlwaysUsageDescription NSLocationWhenInUseUsageDescription NSLocationAlwaysAndWhenInUseUsageDescription NSLocationUsageDescription NSMotionUsageDescription
tcmpp_flutter_mdns	NSBonjourServices
tcmpp_flutter_network	NSAppTransportSecurity
tcmpp_flutter_media/tcmpp_flutter	NSPhotoLibraryUsageDescription

©2013-2022 Tencent Cloud. All rights reserved.

	NSCameraUsageDescription
tcmpp_flutter_device	NSCalendarsUsageDescription NSRemindersUsageDescription NSContactsUsageDescription NSFaceIDUsageDescription NSBluetoothAlwaysUsageDescription NSBluetoothPeripheralUsageDescription

Note：

Permission descriptions should be provided in the android/app/src/main/AndroidManifest.xml file in Android.

Plugin names and permission description in Android:

Plugin name	System permissions
tcmpp_flutter	android.permission.CAMERA android.permission.WRITE_EXTERNAL_STORAGE android.permission.FLASHLIGHT
tcmpp_flutter_lbs	android.permission.ACCESS_FINE_LOCATION
tcmpp_flutter_network	android.permission.INTERNET android.permission.ACCESS_NETWORK_STATE
tcmpp_flutter_media	none
tcmpp_flutter_device	android.permission.BLUETOOTH_ADMIN android.permission.BLUETOOTH_SCAN android.permission.BLUETOOTH_ADVERTISE android.permission.BLUETOOTH_CONNECT android.permission.WRITE_CONTACTS android.permission.READ_CONTACTS android.permission.USE_FINGERPRINT android.permission.USE_BIOMETRIC android.permission.READ_CALENDAR android.permission.WRITE_CALENDAR android.permission.ACCESS_FINE_LOCATION android.permission.ACCESS_WIFI_STATE android.permission.CHANGE_WIFI_STATE android.permission.ACCESS_NETWORK_STATE android.permission.CHANGE_NETWORK_STATE
tcmpp_flutter_googlemap	android.permission.ACCESS_FINE_LOCATION
tcmpp_flutter_petalmap	android.permission.INTERNET android.permission.ACCESS_NETWORK_STATE

©2013-2022 Tencent Cloud. All rights reserved.

	android.permission.CHANGE_WIFI_STATE android.permission.ACCESS_COARSE_LOCATION android.permission.ACCESS_FINE_LOCATION

©2013-2022 Tencent Cloud. All rights reserved.