

# **Tencent Cloud Mini Program Platfor**

## **Getting Started**

### **Product Documentation**



## 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



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

Getting Started

About Platform Prerequisites

# Getting Started

## About Platform Prerequisites

Last updated : 2024-03-04 22:39:12

## Understanding Platform Roles and Permissions

### Role Description

#### Super Administrator

**Role Overview:** The highest authority within the company, which is existent and unique.

**Scope of Permission:** Possesses all functional permissions, capable of adding personnel teams, assigning any role and joining any team, and authorized to approve the joining information of registered members.

#### Platform Administrator

**Role Overview:** Can only be set by the Super Admin (multiple platform administrators allowed), equivalent to the management team of a company.

**Scope of Permission:** Inheriting the Super Admin's permissions but only applicable to the roles of team administrators and regular users.

#### Team administrator

**Role Overview:** Can be added or set as an administrator (or multiple administrators) by the Super Admin or Platform Administrator (multiple allowed), equivalent to the team administrator of a company.

**Scope of Permission:** Can add regular members, set roles for regular members, apply to join other teams, and approve registration information for new members.

#### Team Operator

**Role Overview:** Primarily responsible for internal operations within the affiliated team, requiring approval from the Team Administrator.

**Scope of Permission:** Can add new applications, handle the review process for Mini Program launches, but cannot review registration information for new members.

#### Team Developer

**Role Overview:** Primarily responsible for internal operations within the affiliated team, requiring approval from the Team Administrator.

**Scope of Permission:** Can write Mini Program code, associate applications with Mini Programs, submit Mini Programs for review, but cannot approve registration information for new members.

#### Read-Only Role

**Role Overview:** Primarily used by auditing and inspection personnel, providing permission to view data only.

# Preparation Before Code Connection

## Runtime Environment

### Runtime Environment of the Mini Program

TCMPP Mini Programs can operate on a variety of platforms: the iOS/iPadOS WeChat client, the Android WeChat client, the Mac WeChat client, and the WeChat Developer Tools used for debugging, among others.

In different runtime environments, the script execution environment and the environment used for component rendering are different, resulting in varying performance characteristics:

On iOS, iPadOS, and MacOS, the JavaScript code for the logic layer of the Mini Program runs in JavaScriptCore, while the view layer is rendered by WKWebView. The environments include iOS 14, iPad OS 14, Mac OS 11.4, and others.

On Android, the JavaScript code for the logic layer of the Mini Program operates within [V8](#), while the view layer is rendered by WeChat's proprietary XWeb engine, which is based on the Mobile Chromium kernel.

On the development tools, the JavaScript code for the logic layer of the Mini Program runs within [NW.js](#), while the view layer is rendered by Chromium Webview. JavaScriptCore cannot enable JIT compilation (Just-In-Time Compiler), resulting in noticeably lower performance under equivalent conditions compared to other platforms.

### Platform differences

The differences among various runtime environments are as follows:

Inconsistencies in JavaScript syntax and API support: Developers can circumvent syntax issues by enabling the ES6 to ES5 conversion feature. Additionally, the Mini Program's basic library incorporates necessary Polyfills to compensate for API discrepancies.

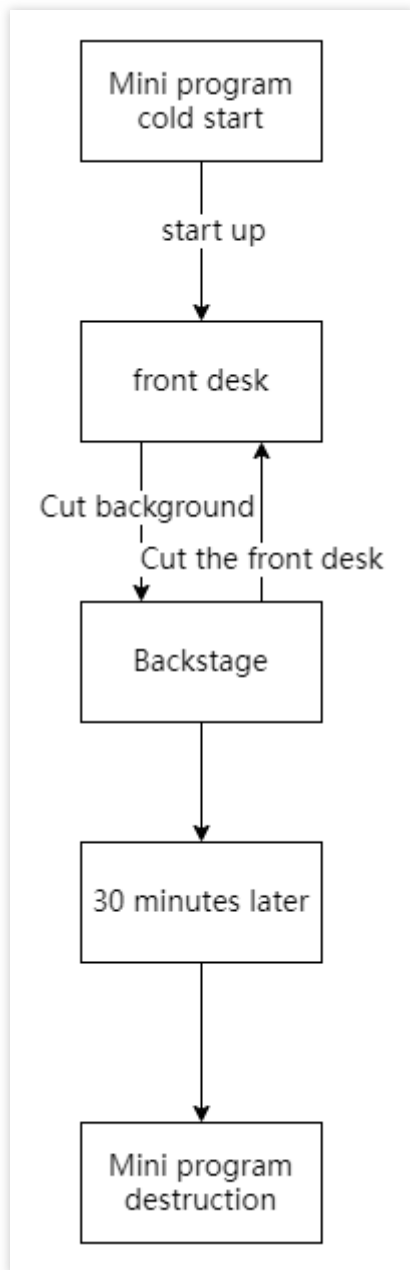
Inconsistent rendering of WXSS: Despite the ability to circumvent most issues by enabling certain features, it is still recommended that developers individually inspect the actual performance of the Mini Program on each end.

The developer tools are intended solely for debugging purposes, with the final performance being determined by the client.

## Operational mechanism

### Mini Program Lifecycle

From initiation to eventual termination, the Mini Program undergoes various states, each exhibiting distinct characteristics.



### Mini Program Initiation

From a user's perspective, the initiation of a Mini Program can be categorized into two scenarios: Cold Start and Warm Start.

**Cold Start:** This refers to the scenario where a user opens the Mini Program for the first time, or reopens it after it has been terminated. In this case, the Mini Program needs to be reloaded and initiated, hence termed as a Cold Start.

**Warm start:** This refers to the scenario where a user reopens a Mini Program within a certain time frame after having previously opened it. In this case, the Mini Program has not been terminated, but merely transitions from backend running to frontend running, a process referred to as a warm start.

From the perspective of the Mini Program's lifecycle, a cold start is generally referred to as **Initiation**, while a warm start is termed as **Transition from Backend to Frontend**.

## Frontend and Backend

Upon initiation, the Mini Program's interface is displayed to the user, at which point the Mini Program runs in the **Frontend**.

When a user **exits** the Mini Program, it doesn't truly close, but rather transitions to **Backend** running. At this point, the Mini Program can still run for a brief period, although the usage of some APIs will be restricted. The methods of transitioning to backend running include, but are not limited to, the following:

Clicking the capsule button in the top right corner to exit the Mini Program.

Swiping rightward from the left side of the screen to exit the Mini Program on iOS system.

Pressing the "Back" button to exit the Mini Program on Android system.

While the Mini Program is running in the frontend, WeChat can be transitioned to the background directly through gestures or by pressing the Home button.

Directly locking the screen while the Mini Program is running in the backend.

When the user re-enters WeChat and opens the Mini Program, the Mini Program will once again transition to **frontend** running.

## Terminate Mini Programs

If the Mini Program has not been used by the user for a long time, or if system resources are strained, the Mini Program will be **terminated**, that is, it will cease to run entirely. This includes the following scenarios:

After the Mini Program has been transitioned to the backend and **suspended**, if it does not re-enter the frontend for an extended period of time (currently 30 minutes), the Mini Program will be terminated.

When the Mini Program consumes excessive system resources, it may be terminated by the system or actively reclaimed by the WeChat client.

On the iOS system, when the WeChat client continuously receives system memory warnings within a certain time interval, it will proactively terminate the Mini Programs according to a specific strategy, and alert the user to **insufficient running memory. Reopen the Mini Programs** after that.

The specific strategy will continue to be adjusted and optimized.

## Debug

Developers can debug the Mini Program using the following tools:

**vConsole:** vConsole allows you to view logs output by the console API and additional debugging information on your mobile device. Using it can facilitate easier identification and resolution of issues.

**Source Map:** Source Map can restore JS error stacks, enabling you to trace where the code has gone wrong, thereby facilitating faster code debugging.

**Logs:** By utilizing the built-in log function of the Mini Program, vulnerabilities can be swiftly identified and issues pinpointed. Logs record crucial operational processes and error log information. Developers can locate and resolve issues by analyzing these log details.

The aforementioned tools are essential aids when developing Mini Programs. They enable developers to debug Mini Program code more swiftly, thereby enhancing development efficiency.

## Issue Summary

### Usage of Webview within Mini Programs

When a Mini Program page directly loads an H5 page using webview, if there are errors in the page or if the page resources have HTTP requests as follows:

```
11757-11757 E/[TMF_MINI]InnerWebView: evaluateJavascript miniAppWebViewStr callback
11757-11757 E/[TMF_MINI]InnerWebView.js: Mixed Content: The page at 'https://wx.vzan.com/plugin-ins/?v=638877544580943418#/FixupIndex/144679432?shareid=9' was loaded over HTTPS,
but requested an insecure image 'http://static1.weixin.qq.com/livecontent/ent_tv/images/vzanqrqr.jpg'. This content should also be served over HTTPS.
```

During the loading process, the following error pages may appear:

### Solution

Replace HTTP resources with HTTPS resources.

### QQ Share Error





Caused by: java.lang.ClassNotFoundException: Didn't find class  
"org.apache.http.conn.scheme.SchemeRegistry"

If the above error occurs during QQ sharing, please add the following configuration:



```
<application
<uses-library android:name="org.apache.http.legacy" android:required="false"/>
</application>
```