Debugging of Collaboration Mobile Apps

Introduction

Debugging of Android and iOS mobile applications can be performed by collecting logs. Logs are especially useful in order to identify crashes/ issues with the application.

In this Guide you can find description on how to collect logs from Android/ iOS devices in case you have any issues with mobile Collaboration.

Useful links:

- Android Collaboration User Guide
- iOS Collaboration User Guide
- Mobile Apps for Android and iOS - FAQ
- Admin level Documentation

Collecting logs from Android app

To gather logs from Android app, follow the next steps:

Step 1. Install adb on you PC
Android Debug Bridge (adb) is a command-line tool that allows you to communicate with your device. As it is included in the Android SDK Platform-Tools package, you need to install this package.

- Unpack the package

⚠️ Note: If you use MacOS, you can skip downloading and install the package via Terminal:

- First, install Homebrew (package management software) using the following command:

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

- Install Android SDK Platform-Tools with this command:

```
brew cask install android-platform-tools
```

**Step 2. Connect your device to PC and enable USB debugging**

To use adb tool with your Android device connected over USB, you need to enable *USB debugging* option in the device system settings, under *Developer options* menu.

On Android 4.2 and higher, *Developer options* menu is hidden by default. To make it visible:

- Go to *Settings -> About phone* and tap *Build number* seven times. You will see a pop-up message “You are now a developer”
- Return to *Settings -> Additional settings* and find *Developer options* menu
- Allow USB debugging by tapping *OK*

⚠️ Note: On some Android devices, *Developer options* menu might be located or named differently. Please consult the documentation of your Android device to reveal the menu.
Step 3. Launch collecting of logs

After you enable USB debugging and connect your device to PC, use the following commands (depending on your OS) to launch collecting of logs. The separate file with logs will be created.

- **On MacOS** (execute the command in Terminal):

  ```bash
  adb logcat -v time > android-logs.txt
  ```

  Note: On MacOS you can also perform log grabbing without creating a separate file, right in Terminal:

  ```bash
  adb logcat -v time
  ```

- **On Windows** (execute the command in cmd: press "Win + R", type cmd, press Enter):

  ```cmd
  C:\Users\user>D:\platform-tools\adb.exe logcat -v time > D:\android-logs.txt
  ```

  Note: When you connect the device, the system asks your permission to accept an RSA key that allows debugging through your computer. This helps to protect the device by ensuring that USB debugging cannot be executed without your permission.
Step 4. Reproduce the issue and collect logs

- Reproduce the issue
- Copy logs from logcat tabs
- To stop collecting logs, disable *USB debugging* option in your device’s settings

Collecting logs from iOS apps

Starting from iOS Collaboration Version 6.0.35105, it is possible to capture logs right off your mobile device.

To capture logs from iOS app:

- Go to *Settings -> Advanced* and enable *Debug* option:

  ![Advanced Settings](image)

  - Reproduce the issue
  - Return to *Settings -> Advanced* and tap *Send* (appears after enabling *Debug* option):
Email template with the attached logs is automatically created.

In the *Subject* field you need to describe the current behavior of the application and fill in information about your device model, iOS version and Collaboration mobile version.

Change/ add new recipients if needed and tap **Send** to forward the report:

**Check / remove push notification subscription on Android/ iOS app**

Important! After the report with logs is sent, you need to disable *Debug* option.
Check push notifications subscription

To check if push notifications (Web push / iOS / Android push) are enabled for an extension:

- Access via SSH as root and launch the following command:

  kcmd htable.dump mobile_devices_data | grep <EXT>

Where:

<EXT> is the extension of a user.

Remove push notifications subscription

To remove push notifications, use the script: push_remove

- Remove push notifications for one extension:

  push_remove <EXT>

- Remove push notifications for several extensions:

  push_remove <EXT> <EXT> <EXT>

- Display list of extensions that subscribed to push notifications:

  push_remove -list

- Display help:

  push_remove -h