

FCM Upgrade

Why to upgrade?

This document is applicable for the Mizu VoIP servers and gateways (including the [MPUSH gateway](#)).

The [old/legacy FCM HTTP server protocol](#) have been deprecated by Google in 2023 and it will stop working from June 2024. If you are using FCM push notifications and your server/gateway have been installed before December 2023, then you must migrate until June 2024 to the new FCM HTTP v1 API.

The major changes are described [here](#). The new protocol requires [OAuth2](#) authentication and the [message format](#) was also changed.

Step-by-step instructions

Please follow these steps to upgrade:

- 1) Upgrade your server or gateway to the latest version as described [here](#)**
- 2) Modify the `fcmmessage.txt` file as described [here](#)**
- 3) Download the `service key file` and change the `fcm_key` setting as described [here](#)**
- 4) Set the `fcm_protocol` global setting to **1****
- 5) Apply the new settings as described [here](#) (reload or restart)**

Upgrade the binaries

The new FCM HTTP v1 API have been implemented in the latest Mizu gateway and server versions.

You must upgrade your gateway or server binaries to be able to use the new protocol, especially the `mserver.exe` and `oauth21.exe`

Step by step instructions:

1. Set the `fcm_protocol` global setting to 0 (to make sure that the service will keep using the old protocol until you are ready with all the changes as described [here](#))
2. Download and extract the new build from the URL provided to you by Mizutech support or contact support@mizu-voip.com if you haven't received it yet.
3. Backup the `mserver.exe` and the `fcmmessage.txt` from the app folder (MManage -> File menu -> Folders -> Server app directory)
4. Stop the `mserver` service (from OS Services control panel or from MManage -> Control menu -> Stop service)
You might also stop the `MServiceHost` service (this is a supervisor service and might restart the `mserver` service if it finds it stopped)
5. Copy the new files to the app folder (overwrite the old files)
6. Start the `mserver` service (from OS Services control panel or from MManage -> Control menu -> Start service)
Start also the `MServiceHost` service if you stopped it before.

At this stage the VoIP service will still keep using the old protocol. You can switch to the new protocol by following the above described instructions (upgrade the `fcmmessage.txt`, download the service key, change the `fcm_key`, set the `fcm_protocol` to 1)

Upgrade the `fcmmessage.txt`

Replace the [old `fcmmessage.txt`](#) with the [new `fcmmessage.txt`](#) in the app folder (MManage -> File menu -> Folders -> Server app directory).

If you made any change in this file then you should merge the changes!

Please note that the new json message format (defined in the fcmmessage.txt) is stricter with the old version. This is relevant only if you changed the default and wish to keep/merge your settings.

Notable changes:

- Everything must be encapsulated into a "message" tag.
- Only strings allowed for values (numbers must be quoted)
- The message can't include a top level to, "priority", "content_available" and "time_to_live" tags and "aps" section anymore.
- The "notification" can't include "sound" and "click_action" (only "title" and "body" is permitted)
- The android section have been changed
- More details [here](#) and [here](#).

Download service key file

The service key file is required for the Mizu server to authentication with the Google FCM cloud service.

1. In the Firebase console, open [Settings > Service Accounts](#)
2. Click Generate New Private Key, then confirm by clicking Generate Key and download the service-account.json file
3. Enable firebase cloud API if not already enabled:
<https://console.developers.google.com/apis/api/fcm.googleapis.com/overview>
4. Copy the service key to the Mizu server or gateway machine.
5. Set the fcm_key (or the fcm_keyX if you have [multiple apps](#)) global configuration to the file path.

For the old protocol the fcm_key configuration were used to store the FCM server API key.

For the new protocol the fcm_key configuration is used to point to the service key file path.

It is recommended to copy the service key to the Mizu server or gateway app folder as "fcmkey.json" (or as "fcmkey.packagename.json" if you have [multiple apps](#)) and in this case you don't even need to set the fcm_key configuration as the server will find it automatically after the fcm_app (package name).

Global settings

Here are the new/related global settings (configurable from MManage -> Configurations form)

- **fcm_protocol**: specify the firebase cloud message sending protocol. -1: auto, 0: legacy (can work until June 2024), 1: HTTP v1 API
- **fcm_serverurl**: the FCM cloud service URL. Default is <https://fcm.googleapis.com/v1/projects/PROJECTID/messages/send> for the FCM HTTP v1 API (configurable also with the fcm_serverurl_v1 settings. The PROJECTID will be replaced at runtime) or <https://fcm.googleapis.com/fcm/send> for the legacy protocol (configurable also with fcm_serverurl_legacy setting)
- **fcm_removentification**: auto remove the "notification" section. -1: auto, 0: don't remove/always keep, 1: automatically remove for calls (keep only for chat and for custom messages), 2: always remove
- **fcm_maxfail**: maximum subsequent failures before to disable FCM. Useful to protect you on account banning on case of a wrong configuration. Disabled at first successful FCM request. Default is 10000.
- **fcm_app**: default FCM app package name
- **fcm_key**: default service key file path for the FCM HTTP v1 API (or the legacy FCM server API key if you are still using the old protocol)
- additional **fcm_appX/ fcm_keyX** pairs if you have multiple apps as described [here](#).

Note: until June 2024 it is also possible to mix apps with the old legacy protocol and the new HTTP v1 API. For this set the fcm_protocol setting to -1 and for apps with the legacy protocol keep the fcm_key as the FCM server API and for the apps with the HTTP v1 API set the key file path.

Multiple applications

In case if you have multiple client apps (with different package name), make sure to send your application package name with the X-MPUSH SIP header or pn-param Contact URI tag.

Note: Here on “application” we mean different SIP client software apps (not different user/instance of the same application as there is no special care required to handle multiple instances).

If you are using Firebase, add all your applications with the fcm_appX and fcm_keyX global config options where X a number from 1 to 1000 (several applications might have the same or different service key).

If the service key files are in the app folder and named like “fcmkey.packgename.json” then you don’t even need to set the fcm_keyX as the server will find them based on the fcm_appX (package name) setting.

The configuration should look like this with multiple apps:

- **fcm_app**: default FCM app package name
- **fcm_key**: default service key file path for the FCM HTTP v1 API (or legacy FCM server API key if you are still using the [old protocol](#))
- **fcm_app1**: additional FCM app package name
- **fcm_key1**: additional service key file path for the FCM HTTP v1 API (or legacy FCM server API key if you are still using the [old protocol](#))
- **fcm_app2**: additional package name
- **fcm_key2**: additional key
- ...
- **fcm_appX**: additional package name
- **fcm_keyX**: additional key

[Keeping the legacy FCM HTTP API](#)

The [old FCM HTTP server protocol](#) have been deprecated in June 2023 and it will stop working from June 2024.

You can still use with the **fcm_protocol** global config set to 0.

If the fcm_protocol is not set, then it automatically use the old protocol if you haven’t upgraded yet to the new version or before June 2024 if you haven’t configured the new protocol yet.

If you are using the legacy API, then the fcm_key must be the server key string and not the service key json file path.

The FCM server key must be loaded from [firebase console](#) -> Project settings (gear icon in the top left near the “Project Overview”) -> Cloud Messaging page -> Cloud Messaging API (Legacy) section -> Server key. (So it is NOT the “Web API Key” from the General page and NOT they “Key pair” and NOT the old “Legacy server key” ...this latter might not be displayed anymore on the website).

For more details see the [old VoIP push notification guide](#) in which we kept discussing about the old FCM API.

[FCM push notification message](#)

The message sent by the Mizutech server or gateway can be configured by rewriting the following text files: fcmmessage.txt (This file can be found in the server app folder which is located by default at C:\Program Files (x86)\MPUSH\ or open the folder from MManage -> File menu -> Folders -> Server App directory)

The default message format is already optimized for VoIP (minimum delay, compatible with all platforms). Change it only if necessary for your specific use case.

The following keywords are replaced automatically at runtime:

- **PACKAGE**: your app package as received from X-MPUSH or pn-param in a previous REGISTER
- **TOKEN**: target user token as received from X-MPUSH or pn-prid in a previous REGISTER
- **TO**: called user name (callee)
- **TYPE**: 0=call,1=chat,2=cancel,3=custom message,4=refresh register binding
- **FROM**: caller/sender full name

- **FROMUSERNAME**: caller/sender username (caller id)
- **FROMDISPLAYNAME**: caller/sender displayname
- **CALLID**: SIP Call-ID of the incoming call or message
- **PRIORITY**: message priority. Default is "high"
- **NPRIORITY**: android notification_priority. Default is "PRIORITY_MAX" for calls and "PRIORITY_HIGH" for others
- **APRIORITY**: apn message priority number. Default is 10 (highest priority) for calls and 5 for others.
- **TTLNUM**: time to live (def to 0 for immediate delivery)
- **PUSHTYPE**: message type. Default value is "voip"
- **SOUND**: notification sound file to be played (you might instead play a sound from your app if required)
- **MSG**: chat message text/body
- **MESSAGE**: notification text (for call it looks like: Incoming VoIP call from FROM)
- **TITLE**: message title (for calls it looks like: Call from FROMUSERNAME)
- **BODY**: message body (for calls it looks like Incoming VoIP call from FROM)

For FCM the message is defined in the **fcmmessage.txt** file which can be found in the app folder and you can edit it after your needs.

The possibilities are described [here](#) and [here](#).

The default message looks like this:

```
{
  "message": {
    "token": "TOKEN",
    "notification": {
      "title": "TITLE",
      "body": "BODY"
    },
    "data": {
      "ntype": "TYPE",
      "nfromusername": "FROMUSERNAME",
      "nfrom": "FROM",
      "nmsg": "MSG",
      "ncallid": "CALLID",
      "ntouser": "TOUSERNAME"
    },
    "android": {
      "priority": "PRIORITY",
      "ttl": "TTLNUMs",
      "notification": {
        "notification_priority": "NPRIORITY",
        "sound": "SOUND",
        "visibility": "PUBLIC"
      }
    },
    "webpush": {
      "headers": {
        "ttl": "TTLNUM",
        "Urgency": "PRIORITY"
      },
      "fcm_options": {
        "link": ""
      }
    },
    "apns": {
      "headers": {
        "apns-priority": "APRIORITY",
        "apns-expiration": "TTLNUM",
        "apns-push-type": "PUSHTYPE",
        "apns-topic": "PACKAGE.voip"
      },
      "payload": {
        "aps": {
          "alert": {
            "title": "TITLE",
            "body": "BODY"
          },
          "content-available": 1,
          "sound": "SOUND"
        }
      }
    }
  }
}
```

}

The “webpush” or “apns” sections will be automatically removed if you are not using these (if you send FCM/Android messages only).

By default (depending on the fcm_removentification setting) the “notification” section is automatically removed for calls to avoid additional display on the OS tray icon (kept only for chat and for custom messages).

The bottom of the message from the --EOF—mark will be automatically removed.

The message must be in valid JSON format, which can be verified/validated with any tool like [this](#) or [this](#).

Send the “reloadfcm” command from the Server Console form if you made any changes to refresh its cached copy.

Apply the new settings

You must do one of the followings if you changed the FCM related settings:

- Restart the service (from OS Services control panel or from MManage -> Control menu -> Restart service).
Recommended if there is no any ongoing calls on the service.
- Send the “reload” command from the MManage Server Console form (or push the Apply Now button on the Configurations form).
Recommended if there are active calls to avoid forced disconnects.