gb3android

Public

Notifications · Fork 0 · Star 0

main · Go to file · Code

zorel Add screenshot to README ✓ f378fa5 · 15 hours ago

.github/workflows	Attach signed APK to ...	last week
app	Allow selecting gb2c t...	4 days ago
gradle	Initial commit	2 weeks ago
scripts	Initial commit	2 weeks ago
.gitignore	Initial commit	2 weeks ago
LICENSE	Initial commit	2 weeks ago
README.md	Add screenshot to RE...	15 hours ago
build.gradle.kts	Initial commit	2 weeks ago
gradle.properties	Initial commit	2 weeks ago
gradlew	Initial commit	2 weeks ago
gradlew.bat	Initial commit	2 weeks ago
screenshot.jpeg	Add screenshot to RE...	15 hours ago
settings.gradle.kts	Initial commit	2 weeks ago

About

No description, website, or topics provided.

- Readme
- MIT license
- Activity
- 0 stars
- 0 watching
- 0 forks

Report repository

Releases 7

gb3android v0.3.1 Latest
4 days ago

+ 6 releases

Contributors 1

zorel Aurélien DEHAY

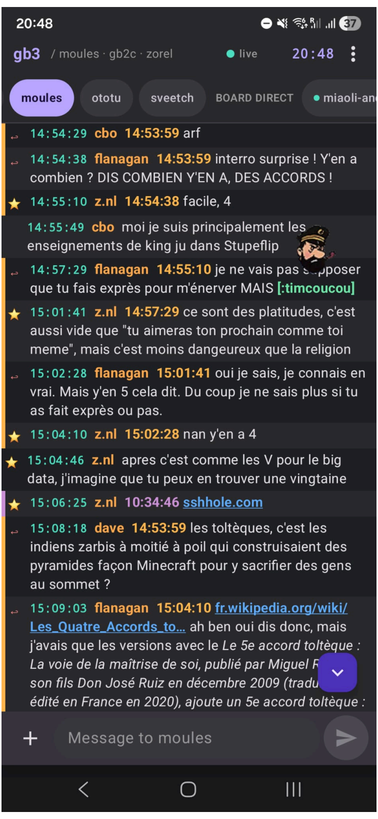
Languages

Kotlin 98.8% Shell 1.2%

README MIT license

gb3 for Android

A native Android client for the gb3 `gb2c` server. It talks directly to the same HTTP/SSE API as `gb3tui` and `gc2`, and can also act as its own tribune client: it downloads each selected direct board's backend and posts to its URL itself, without a gb2c server in the middle. Like roinroin, it can use one or more gb2c servers **and** direct boards at the same time. There is no WebView for the tribune itself; a small WebView is only used to complete the LinuxFR OAuth flow.



14:54:29 **cbo** **14:53:59** arf

14:54:38 **flanagan** **14:53:59** interro surprise ! Y'en a combien ? DIS COMBIEN Y'EN A, DES ACCORDS !

★ 14:55:10 **z.nl** **14:54:38** facile, 4

14:55:49 **cbo** moi je suis principalement les enseignements de king ju dans Stupeflip

14:57:29 **flanagan** **14:55:10** je ne vais pas s..pposer que tu fais exprès pour m'énerver MAIS [:timcoucou]

★ 15:01:41 **z.nl** **14:57:29** ce sont des platitudes, c'est aussi vide que "tu aimeras ton prochain comme toi meme", mais c'est moins dangeureux que la religion

15:02:28 **flanagan** **15:01:41** oui je sais, je connais en vrai. Mais y'en 5 cela dit. Du coup je ne sais plus si tu as fait exprès ou pas.

★ 15:04:10 **z.nl** **15:02:28** nan y'en a 4

★ 15:04:46 **z.nl** apres c'est comme les V pour le big data, j'imagine que tu peux en trouver une vingtaine

★ 15:06:25 **z.nl** **10:34:46** sshhole.com

15:08:18 **dave** **14:53:59** les toltèques, c'est les indiens zarbis à moitié à poil qui construisaient des pyramides façon Minecraft pour y sacrifier des gens au sommet ?

15:09:03 **flanagan** **15:04:10** fr.wikipedia.org/wiki/Les_Quatre_Accords_to... ah ben oui dis donc, mais j'avais que les versions avec le *Le 5e accord toltèque : La voie de la maîtrise de soi, publié par Miguel R son fils Don José Ruiz en décembre 2009 (tradu édité en France en 2020), ajoute un 5e accord toltèque :*

Features

The client covers the feature set of `gb3tui` and `gc2`, adapted to touch:

- **multiple gb2c servers and direct boards at the same time**, like roinroin: one SSE stream per enabled gb2c server, one poll loop per selected board, and source-qualified tribunes so the same board name can appear from several sources without mixing posts, badges or last-read markers;
- live posts from `/gb2c/poll` over server-sent events, with automatic reconnection and foreground-only streaming;
- **direct boards**: gb2c's predefined boards plus user-defined boards, polled every 30 seconds straight from their TSV backend and posted to directly. Malformed TSV rows, invalid ids, oversized lines and control characters are rejected or sanitized before they reach the UI;
- posting to any tribune through `/gb2c/post` or directly to the selected board, including LinuxFR (`dlfp`) OAuth login, token refresh and restoration of the message while authorizing;
- tribune switching with source-grouped chips (`gb2c · server name`, then `board direct`), horizontal swipes and unread/reply/bigorno badges. The header shows the active tribune and its source, e.g. `/ moules · gb2c · local` or `/ moules · board direct`;
- clock references (norloges): tap one to jump to the referenced post, tap a post clock to quote it in the composer, and see reverse references highlighted while a post is selected;
- conversation grouping with a stable, high-contrast color rail, plus the same conversation filter as `gc2`:
  - hide a login/user-agent and every direct or indirect reply,
  - either hide those posts or show them grayed with their message masked;
- server-side blocked identities from `/gb2c/config` merged with local settings;
- bigorno, "this is a reply to me" and "mine" indicators in a reserved leading slot, so clocks and author names stay aligned;
- message formatting: `<b>`, `<i>`, `<u>`, `<s>`, `<tt>`,

`<code>`, tappable spoilers, links, `[:totoz]`, `nick<` bigornos and normal bouchot whitespace;

- totoz search with image grid and preview, either through gb2c or straight from a configurable totoz server (`https://nsfw.totoz.eu` by default), emoji search, attachment upload (4 MiB server limit) and the ALT formatting shortcuts;
- plop-list message replacement;
- English and French UI, following the Android system language;
- light/dark Material 3 theme following the system, with the gc2 palette;
- the alarm clock that counts seconds during the last minute of each hour.

Requirements

- JDK 17 (a full JDK with `javac`, not only a Java runtime)
- Android SDK 35 (`compileSdk = 35`, Build-Tools 35.0.0)
- Android Studio Ladybug or newer, or the Gradle wrapper

The project includes a Gradle wrapper. Build from this directory:

```
./scripts/build.sh :app:assembleDebug
```

`build.sh` finds a JDK that actually contains `javac`, sets `JAVA_HOME` and `ANDROID_HOME`, and forwards the arguments to Gradle. You can also call Gradle directly after exporting a valid JDK:

```
export JAVA_HOME=/path/to/jdk-17
export ANDROID_HOME=/path/to/android-sdk
./gradlew :app:assembleDebug
```

If Gradle reports:

```
Toolchain installation '/usr/lib/jvm/...' does no
capabilities: [JAVA_COMPILER]
```

the default `java` is a JRE, not a JDK. Install `openjdk-17-jdk` (or point `JAVA_HOME` at a full JDK). The checked-in `gradle/gradle-daemon-jvm.properties` selects JDK 17 for the Gradle daemon.

If the SDK is not installed at all, bootstrap it with:

```
./scripts/install-android-sdk.sh
```

The script installs the command-line tools, accepts the Android SDK licences, installs `platform-tools`, `platforms;android-35` and `build-tools;35.0.0`, and writes an ignored `local.properties` with `sdk.dir`. It supports Linux and macOS.

`buildToolsVersion` only selects a Build-Tools revision; it does not download anything. AGP's `android.builder.sdkDownload=true` can install missing SDK components when an SDK root already exists and licences are accepted, but it cannot bootstrap a completely missing SDK. Use the script for that.

The debug APK is written to `app/build/outputs/apk/debug/app-debug.apk`.

For an installable release APK:

```
./gradlew :app:assembleRelease
```

The APK is written to `app/build/outputs/apk/release/app-release.apk`.

Android refuses to install unsigned APKs. For local testing, the release build falls back to the locally generated Android debug key, so it is installable but must not be published. Provide a real release key through the environment for distribution:

```
export GB3_RELEASE_KEYSTORE=/path/to/release.jks
export GB3_RELEASE_KEYSTORE_PASSWORD=...
export GB3_RELEASE_KEY_ALIAS=...
export GB3_RELEASE_KEY_PASSWORD=...
./gradlew :app:assembleRelease
```

Run the unit tests with:

```
./scripts/build.sh :app:testDebugUnitTest
```

The test suite includes an SSE burst regression test (gb2c replays its whole history in one burst when a client connects; the client applies backpressure instead of silently dropping events) and parser/conversation tests.

Continuous integration

GitHub Actions runs the unit tests and builds both the debug and release APKs for pushes and pull requests to `main` (`.github/workflows/android-ci.yml`). Successful runs upload the APKs as the `gb3-android-debug` and `gb3-android-release-*` workflow artifacts.

To sign the release APK in CI with a real release key, add these repository secrets:

- `GB3_RELEASE_KEYSTORE_BASE64`: the keystore file as a single-line base64 string (`base64 -w0 release.jks` on Linux, `base64 -i release.jks` on macOS);
- `GB3_RELEASE_KEYSTORE_PASSWORD`;
- `GB3_RELEASE_KEY_ALIAS`;
- `GB3_RELEASE_KEY_PASSWORD`.

Without them, the release APK uses the debug-key fallback described above and its artifact name ends in `debug-key`; it must not be published.

Publishing a GitHub Release triggers `.github/workflows/release.yml`, which builds a signed release APK and attaches it to the release as `gb3android-<tag>.apk`. That workflow requires the four signing secrets and fails instead of attaching a debug-signed APK when they are missing.

Sources

The app can read several sources at once, like roinroin:

- **gb2c servers** - enter the base URL of one or more `gb2c` servers, for example `https://gb3.example.com` or `http://192.168.1.10:16666`. Each one streams over `/gb2c/poll` and is posted to through `/gb2c/post`. The first enabled server also handles emoji search, attachment uploads and LinuxFR OAuth.

- **Direct boards** - check any of the built-in or custom boards. The app downloads each board's TSV backend itself and posts to its post URL. They work alongside the gb2c servers, with their own connection status.

A gb2c server and a direct board can expose the same tribune name. They are kept as two separate entries, each with its own posts, badges and last-read marker. The tribune bar groups chips by source and the header names the active source. Settings can add, remove, rename or disable gb2c servers, check/uncheck direct boards, and choose which discovered tribunes to display per gb2c server (all are shown by default; **All** and **None** buttons are provided).

The nickname is sent as the post `info`; if it is left empty, a gc2-style random nickname is generated. On an Android emulator, `localhost` is the emulator itself: use `http://10.0.2.2:16666` to reach a gb2c process running on the host machine.

Direct boards

The app ships the same default boards as gb2c:

Board	Backend
`euromussels`	`https://faab.euromussels.eu/data/bac`
`sveetch`	`http://sveetch.net/tribune/remote/ts`
`moules`	`https://moules.org/board/backend/tsv`
`dlfp`	`https://linuxfr.org/board/index.tsv`
`ototu`	`https://www.ototu.me/goboard/backen`
`devnewton`	`https://gb3.devnewton.fr/gb0/tsv`
`gabuzomeu`	`https://gb3.plop.cc/gb0/tsv`
`reveildutroll`	`https://miaoli.im/%2B6VtcdSCI0Y/post`

Open **Settings -> Boards -> Add custom board** to add any other bouchot-style board, then leave it checked to show its tribune next to the gb2c sources. A custom board needs:

- a unique name without spaces or commas (it becomes the tribune name);
- the backend URL returning five-column TSV (`id`, `time`, `info`, `login`, `message`);
- the post URL accepting an `application/x-www-form-urlencoded` POST;
- the post field name (defaults to `message`);
- an optional info field. When it is empty, the nickname is sent in the `User-Agent` header, exactly like gb2c;
- an optional LinuxFR OAuth2 flag for boards that require a Bearer token.

Custom boards are stored as JSON in `SharedPreferences` and can be removed from the same section. Built-in boards are read-only.

Totoz server

The **Services** section in Settings (and the first-run setup screen) contains the totoz server URL. It defaults to `https://nsfw.totoz.eu`, the same search server gb2c uses internally.

- Search requests go directly to `<server>/search.xml?terms=...`.
- Preview images are loaded from `<server>/img/<name>`.
- Totoz search therefore works without any gb2c server.
- Both a base URL (`https://nsfw.totoz.eu`) and a full `https://nsfw.totoz.eu/search.xml` endpoint are accepted.
- Leaving the field empty falls back to gb2c's `/gb2c/totoz/search` endpoint when a gb2c URL is configured.

The XML response is parsed defensively: external entities and DOCTYPE declarations are disabled, responses are bounded to 2 MiB, and blank names are ignored. Image URLs are always rebuilt from the configured server, like gb2c does, instead of trusting an image URL supplied by the server.

Direct boards are read defensively: rows with fewer than five fields or with an invalid/non-positive id are skipped, duplicate ids keep the latest row, extra columns are dropped, control characters are removed, messages are capped at 64 KiB and backend responses are capped at 8 MiB.

Touch gestures

- tap a tribune chip, or swipe the post list left/right, to change tribune;
- tap a post to select it and highlight every clock reference pointing to it;
- tap its clock to quote it, or tap its author to add `author<` to the message;
- tap a clock reference inside a message to jump to the referenced post;
- long-press a post to reply, call the author, copy the message or show its metadata;
- tap a spoiler to reveal or hide it, and tap a totoz to show its image at its original size just above the totoz characters (animated GIFs and WebP play; tap anywhere to dismiss);

- use the `+` button in the composer, or the top-right menu, for attachments, totoz, emoji, formatting shortcuts and settings.

Settings are stored locally in `SharedPreferences`:

- gb2c server list (name, URL, enabled) and the displayed tribunes per server;
- selected direct boards and custom board definitions;
- local blocked identities and hide/gray mode;
- plop list;
- last posted message, active source-qualified tribune and per-tribune last-read markers;
- LinuxFR access/refresh tokens.

API endpoints used

Both source kinds can be active at the same time.

For each enabled gb2c server:

Endpoint	Usage
`/gb2c/list`	tribune list
`/gb2c/config`	server-side blocked identities
`/gb2c/poll?filter=...`	SSE live posts
`/gb2c/post`	post a message
`/gb2c/totoz/search`	totoz search
`/gb2c/emoji/search`	emoji search
`/gb2c/attachment/`	multipart attachment upload
`/gb2c/linuxfr/authorize`	OAuth authorization page
`/gb2c/linuxfr/refresh`	refresh a LinuxFR token

For each selected direct board:

Endpoint	Usage
`<board backendUrl>`	five-column TSV history
`<board postUrl>`	form-encoded post
`<totoz server>/search.xml?terms=...`	direct totoz search
`<totoz server>/img/<name>`	totoz preview image

Each configured board's `backendUrl` and `postUrl` are contacted directly. Totoz search and previews use the configured totoz server. The first enabled gb2c server is also used for emoji search, attachments and LinuxFR OAuth.

The LinuxFR redirect configured on the gb2c server must be reachable from the device. gb2c currently builds that redirect with `https://`, so a gb2c instance using plain HTTP on a LAN may not be able to complete OAuth; the rest of the app works normally. Direct `dlfp` posting works with an existing LinuxFR token, but obtaining or refreshing that token still needs a gb2c server with OAuth credentials configured.

Project layout

```
app/src/main/java/org/zorel/gb3android/
  MainActivity.kt              single-activity C
  data/                        models, preferenc
  domain/MessageParser.kt      native port of th
  domain/Conversation.kt       norloge index, re
  ui/Gb3ViewModel.kt           state, stream/boa
  ui/Gb3App.kt                 navigation, main
  ui/components/               post list, rows,
  ui/screens/                  setup, settings,
  ui/theme/                    Material 3 color
app/src/test/                  parser, conversat
```

Notes and limitations

- The stream runs while the app is in the foreground. There are no OS push notifications; unread/reply/bigorno indicators are in-app badges.
- Direct boards are refreshed every 30 seconds, like gb2c; there is no live push from the board itself.
- The client is read-only for server configuration. Blocked identities configured with `GB2C_BLOCKED_IDENTITIES` are displayed in Settings but can only be changed on the server. Local blocked identities are editable in the app.
- Attachments are inserted as their public URL in the composer, like gc2. They need at least one enabled gb2c server, even when using direct boards.
- Totoz images are loaded from the configured totoz server (`https://nsfw.totoz.eu` by default) and previewed at their original size.
- LinuxFR is only needed to post to the `dlfp` tribune. OAuth uses a dedicated WebView so the token stays in the app and passwords are only entered on linuxfr.org. Connecting or refreshing a LinuxFR account requires a gb2c server with OAuth credentials; an existing token can be reused by an OAuth2 custom board.

Terms Privacy Security Status Community Docs Contact Manage cookies
Do not share my personal information
© 2026 GitHub, Inc.