
title: Contributing guide

revision: 31.01.2023

| Updated 31.01.2023 | Languages: EN, [FR](#), [CZ](#) |

Contributing guide

Compiling with SQLCipher encryption enabled

Add `cabal.project.local` to project root with the location of OpenSSL headers and libraries and flag setting encryption mode:

```
``` cp scripts/cabal.project.local.mac cabal.project.local
```

**or**

```
cp scripts/cabal.project.local.linux
cabal.project.local
```

```
```
```

OpenSSL on MacOS

MacOS comes with LibreSSL as default, OpenSSL must be installed to compile SimpleX from source.

OpenSSL can be installed with `brew install openssl@1.1`

You will have to add `/opt/homebrew/opt/openssl@1.1/bin` to your PATH in order to have things working properly

Project branches

In `simplex-chat` repo

- `stable` - stable release of the apps, can be used for updates to the previous stable release (GHC 9.6.2).
- `stable-android` - used to build stable Android core library with Nix (GHC 8.10.7).

- `stable-ios` - used to build stable iOS core library with Nix (GHC 8.10.7) – this branch should be the same as `stable-android` except Nix configuration files.
- `master` - branch for beta version releases (GHC 9.6.2).
- `master-android` - used to build beta Android core library with Nix (GHC 8.10.7).
- `master-ios` - used to build beta iOS core library with Nix (GHC 8.10.7) – this branch should be the same as `master-android` except Nix configuration files.

In `simplexmq` repo

- `master` - uses GHC 9.6.2 its commit should be used in master branch of `simplex-chat` repo.
- `master-ghc8107` - its commit should be used in `master-android` (and `master-ios`) branch of `simplex-chat` repo.

Development & release process

1. Make PRs to master branch only for both `simplex-chat` and `simplexmq` repos.
2. If `simplexmq` repo was changed, to build mobile core libraries you need to merge its master branch into `master-ghc8107` branch.
3. To build Android core library:
 - merge master branch to `master-android` branch.
 - update code to be compatible with GHC 8.10.7 (see below).
 - update `simplexmq` commit in `master-android` branch to the commit in `master-ghc8107` branch.
 - push to GitHub.
1. To build iOS core library, merge `master-android` branch to `master-ios` branch, and push to GitHub.
2. To build Desktop and CLI apps, make tag in master branch, APK files should be attached to the release.
3. After the public release to App Store and Play Store, merge:
 - `master` to `stable`
 - `master` to `master-android` (and compile/update code)
 - `master-android` to `master-ios`
 - `master-android` to `stable-android`
 - `master-ios` to `stable-ios`
1. Independently, master branch of `simplexmq` repo should be merged to stable branch on stable releases.

Differences between GHC 8.10.7 and GHC 9.6.2

1. The main difference is related to DuplicateRecordFields extension.

It is no longer possible in GHC 9.6.2 to specify type when using selectors, instead OverloadedRecordDot extension and syntax are used that need to be removed in GHC 8.10.7:

```
```haskell {-# LANGUAGE DuplicateRecordFields #-} -- use this in GHC
9.6.2 when needed {-# LANGUAGE OverloadedRecordDot #-}
```

```
-- GHC 9.6.2 syntax let x = record.field
```

```
-- GHC 8.10.7 syntax removed in GHC 9.6.2 let x = field (record :: Record)
```

It is still possible to specify type when using record update syntax, use this pragma to suppress compiler warning:

```
```haskell -- use this in GHC 9.6.2 when needed {-# OPTIONS_GHC -fno-
warn-ambiguous-fields #-}
```

```
let r' = (record :: Record) {field = value} ```
```

1. Most monad functions now have to be imported from Control.Monad, and not from specific monad modules (e.g. Control.Monad.Except).

```
haskell -- use this in GHC 9.6.2 when needed import Control.Monad
```

[This PR](#) has all the differences.