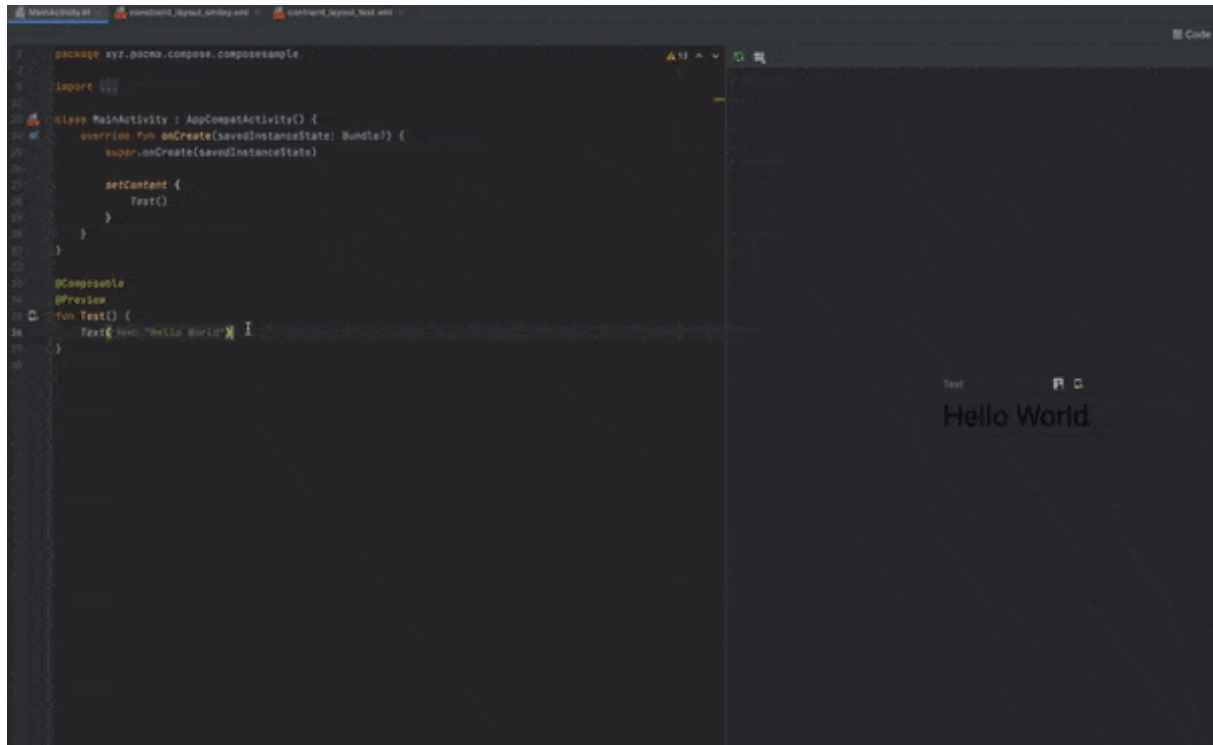

recompose

recompose is a tool for converting Android layouts in XML to Kotlin code using Jetpack Compose. It can be used on the command line, as IntelliJ / Android Studio plugin or as a library in custom code.



Running

IntelliJ / Android Studio plugin

The plugin does not get published yet. If you want to try them then you'll have to build them from code. Use the Gradle task `runIde` to run a test IDE with the plugin installed or create an installable plugin with the `buildPlugin` task. The plugin zip will be placed in `recompose-idea/build/distributions/`. To learn how to install it, read the Install plugin from disk section in the IntelliJ docs.

Command-line interface (CLI)

Like the plugin, the command-line interface does not get published yet. You can run it directly from Gradle via `./gradlew recompose-cli:run --args="list file paths.."`.

Alternatively you can run the `assembleDist` which will place a zip/tar containing a binary into `recompose-cli/build/distributions/`.

```
1 Usage: recompose [OPTIONS] INPUT...
2
3 Options:
4   -o, --output PATH  Output directory for Kotlin code
5   -h, --help          Show this message and exit
6
7 Arguments:
8   INPUT  Layout XML files to convert to Kotlin
```

Building

Either import the project into IntelliJ IDEA or use Gradle on the command line (via the provided `gradlew` wrapper).

Modules

- **recompose-ast**: Contains the data classes for the Abstract Syntax Tree (AST) representing a parsed XML layout.
- **recompose-cli**: A command-line interface (CLI) for running recompose in a shell.
- **recompose-composer**: Responsible for taking an AST and transforming it into the equivalent `Composable` Kotlin code.
- **recompose-idea**: An IntelliJ IDEA / Android Studio plugin that allows copying XML layouts and pasting as `Composable` Kotlin code.
- **recompose-test**: Contains test data and helpers for unit tests.

Important gradle tasks

- **clean**: Deletes the build directories.
- **test**: Runs all unit tests in all modules.
- **runIde**: Runs IntelliJ IDEA with the plugin installed.
- **buildPlugin**: Builds the IntelliJ / Android Studio plugin and makes it available in `recompose-idea/build/distributions`.

How does this work?

The Parser (`recompose-parser`) takes the input XML and transforms it into an Abstract Syntax Tree (`recompose-ast`). The Composer (`recompose-composer`) takes the AST and translates it into Kotlin code calling Composables.

The IntelliJ / Android Studio plugin (`recompose-idea`) uses that to perform the translation when pasting copied XML code. And the CLI (`recompose-cli`) uses it to translate files.

Can I contribute?

Yes, absolutely. There are a ton of [Views](#) and attributes to support. The list of issues labeled with good first issue are a good place to start. An issue from the list labeled with help wanted may be a good follow-up. Just comment on any issue that interests you.

License

```
1 Copyright 2020 Sebastian Kaspari
2
3 Licensed under the Apache License, Version 2.0 (the "License");
4 you may not use this file except in compliance with the License.
5 You may obtain a copy of the License at
6
7 http://www.apache.org/licenses/LICENSE-2.0
8
9 Unless required by applicable law or agreed to in writing, software
10 distributed under the License is distributed on an "AS IS" BASIS,
11 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied
12
13 See the License for the specific language governing permissions and
14 limitations under the License.
```