



## Рекомендации по началу разработки на Flutter под Aurora OS

Дисциплина «Мобильная разработка на Flutter»

### Особенности разработки под Aurora OS

Как и другие платформы, Aurora OS требует плагины для использования прямо из Flutter её функций, а также немного платформозависимого кода.

Последний можно добавить с помощью консольной команды Flutter с поддержкой Aurora OS, когда он будет установлен. Далее в сгенерированных файлах потребуется лишь добавлять зависимости в .spec файле или разрешения в .desktop файле. При необходимости можно заменить иконки, название приложения и организации.

Ограничения в виде зависимостей и разрешений накладывают подключаемые плагины. Так как Flutter на Aurora OS только развивающаяся технология, все имеющиеся Aurora OS реализации популярных плагинов необходимо искать не в pub.dev, а в репозиториях Открытой мобильной платформы:

- <https://gitlab.com/omprussia/flutter/packages>
- <https://gitlab.com/omprussia/flutter/flutter-community-plugins>

Инструкция по подключению каждого плагина имеется в его README.md файле. Также реализацию искомого плагина можно спросить в чате сообщества:

- [https://t.me/aurora\\_devs/23477](https://t.me/aurora_devs/23477)



- или проявить инициативу в разработке собственного решения.

В данный момент ещё не все основные плагины реализованы командой ОМП, но со временем все стандартные потребности по платформенному функционалу будут закрыты.

## Начало разработки

Для того, чтобы начать разработку на Aurora OS, необходимо установить Aurora SDK, Aurora PSDK и форк Flutter с поддержкой AuroraOS. Это можно сделать либо с официального сайта и гитлаба согласно инструкциям:

- <https://omprussia.gitlab.io/flutter/flutter/install/linux/>
- <https://gitlab.com/omprussia/flutter/flutter>
- либо воспользоваться консольным приложением aurora-cli: <https://keygenqt.github.io/aurora-cli/>

Рекомендуется воспользоваться вторым вариантом установки, так как aurora-cli выполняет большую часть работы по установке всего необходимого, активно поддерживается разработчиком, а также облегчает саму разработку приложения. Отдельный совет выбрать вариант установки Aurora SDK с Oracle VM VirtualBox, так как этот вариант более стабильный, нежели установка через Docker.

После установки всех трёх компонент, необходимо создать платформенную часть командой 'flutter-aurora create . -platform=aurora' и добавить необходимые плагины, согласно их инструкциям. Теперь приложение готово к запуску.



## Работа с эмулятором и настройка подписания

В дальнейшем гайде будет использоваться `aurora-cli` как основной способ работы с приложениями.

Эмулятор Aurora OS устанавливается вместе с Aurora SDK. Его архитектура - `x86_64`. Запустить его можно с помощью `VirtualBox`, либо с помощью `aurora-cli`. Для того, чтобы Flutter мог работать с эмулятором, его, как и любой другой Aurora девайс, необходимо добавить с помощью команды `'aurora-cli flutter custom-devices'`. Подробнее про эту команду можно прочитать тут:

- <https://keygenqt.github.io/aurora-cli/cli/flutter/custom-device/>

После этого Flutter сможет воспринимать его как устройство для запуска приложения.

Кроме того, для установки `rpm` пакета с приложением, его необходимо подписать ключами. Больше информации, а также общедоступные ключи можно найти здесь:

- [https://developer.auroraos.ru/doc/sdk/app\\_development/packaging/package\\_signing](https://developer.auroraos.ru/doc/sdk/app_development/packaging/package_signing)

Пакет подписывается с помощью `PSDK`, но `aurora-cli` позволяет вызывать его функции сразу после сборки, если заранее указать ключи в файле `.aurora-cli/configuration3.yaml`. Подробнее здесь:

- <https://keygenqt.github.io/aurora-cli/config/#keys>



## Запуск приложения

Когда эмулятор будет распознан и ключи будут настроены, для сборки, подписания и установки приложения для эмулятора достаточно воспользоваться командой `'aurora-cli flutter project build -i'` и выбрать `x86_64` архитектуру. Flutter соберёт release сборку приложения и установит на эмулятор. Достаточно будет запустить приложение по иконке. Подробности по запуску приложения можно узнать здесь:

- <https://keygenqt.github.io/aurora-cli/cli/flutter/project/build/>

Для того, чтобы запустить отладку Flutter необходимо воспользоваться командой `'aurora-cli flutter project build -i -r dart -d'`, а после успешного запуска приложения подключиться к Dart VM по ссылке, что будет выведена в консоль или через IDE с помощью сгенерированного файла `launch.json`. При подключении в Dart VM убедитесь, что путь к `main.dart` указан верно.

Также `aurora-cli` позволяет проводить отладку нативного кода с помощью команды `aurora-cli flutter project build -i -r gdb -d`.

## Решение проблем

Если в процессе сборки будут какие-либо ошибки, воспользуйтесь флагом `-v` для большей информации. Так же для получения большей информации можете воспользоваться `'flutter-aurora doctor'`. Проверьте ещё раз все этапы установки,



настройки и инструкции подключения плагинов. Если проблему не удалось решить, обратитесь за помощью к чату сообщества:

- [https://t.me/aurora\\_devs/23477](https://t.me/aurora_devs/23477)

Также можно воспользоваться информацией из других гайдов:

- [https://m.vk.com/video-155491659\\_456239054?from=video](https://m.vk.com/video-155491659_456239054?from=video)
- [https://youtu.be/mK7MDtxJ94M?si=\\_0Gr83fJlsUtfTLN](https://youtu.be/mK7MDtxJ94M?si=_0Gr83fJlsUtfTLN)

Однако данные гайды могли потерять свою актуальность за время.