

Глава 0

УСТАНОВКА И НАСТРОЙКА РАБОЧЕГО ОКРУЖЕНИЯ. ОСНОВЫ DART

В этой главе мы рассмотрим, как выполнить установку и настройку рабочего окружения Flutter в различных операционных системах, а также познакомим вас с базовым синтаксисом Dart. Для более глубокого погружения в этот замечательный язык программирования обратитесь к книге Станислава Чернышева «Основы Dart».

0.1. Установка и настройка рабочего окружения

Написание кода, представленного в книге, будет вестись в Visual Studio Code, который можно скачать по ссылке <https://code.visualstudio.com/download>.

0.1.1. ОС Windows

Существует два варианта установки Flutter в Windows: автоматическая и ручная. Оба они не слишком сложны, но во втором случае у вас есть возможность явно выбрать каталог, куда будет выполнена распаковка Flutter SDK. Поэтому с него мы и начнем, воздав дань тем, кто любит, когда власть полностью сосредоточена в их руках.

Перейдите по следующей ссылке: <https://docs.flutter.dev/get-started/install>.

Либо зайдите на сайт фреймворка, в раздел Get started. Перед вами откроется страница выбора операционной системы, в которой вы будете вести разработку (рис. 0.1).

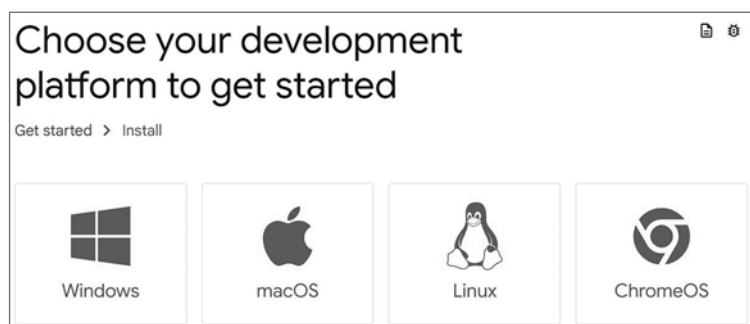


Рис. 0.1. Выбор ОС, в которой будет вестись разработка

На следующем шаге выберите предлагаемую по умолчанию конфигурацию запуска новых проектов на Flutter (рис. 0.2).

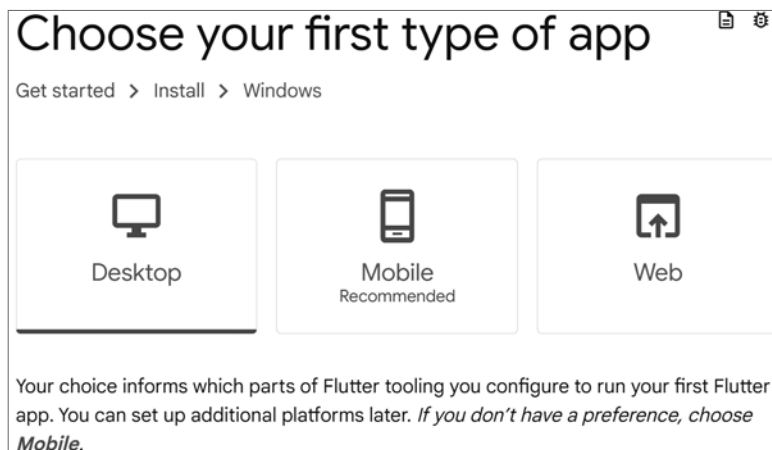


Рис. 0.2. Выбор конфигурации запуска новых проектов на Flutter

Далее найдите раздел **Install the Flutter SDK**, выберите **Download and install** и скачайте архив с актуальной версией фреймворка (рис. 0.3).

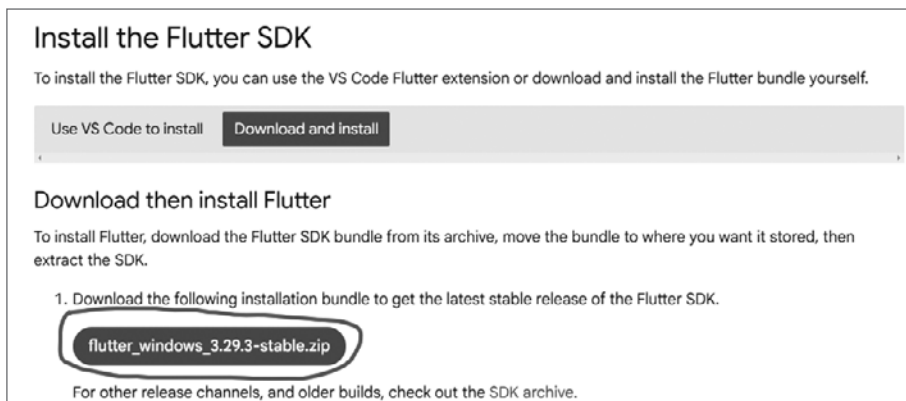


Рис. 0.3. Скачивание архива с актуальной версией Flutter

После того как архив с Flutter SDK загрузится, распакуйте его в удобную для вас папку. Обычно ею выступает корневой каталог диска (C, D, F и т. д.). Теперь необходимо прописать путь до распакованного Flutter SDK в переменных средах в переменной Path (рис. 0.4).

В зависимости от целевой платформы, под которую вы планируете разрабатывать приложения, предстоит скачать необходимый инструментарий. Для Windows это будет Visual Studio Community со следующим выбранным пакетом (рис. 0.5).

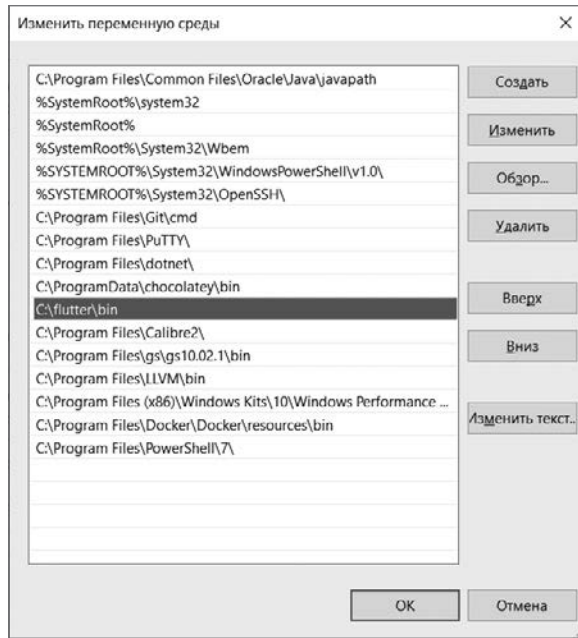


Рис. 0.4. Указываем путь до Flutter SDK в переменной Path

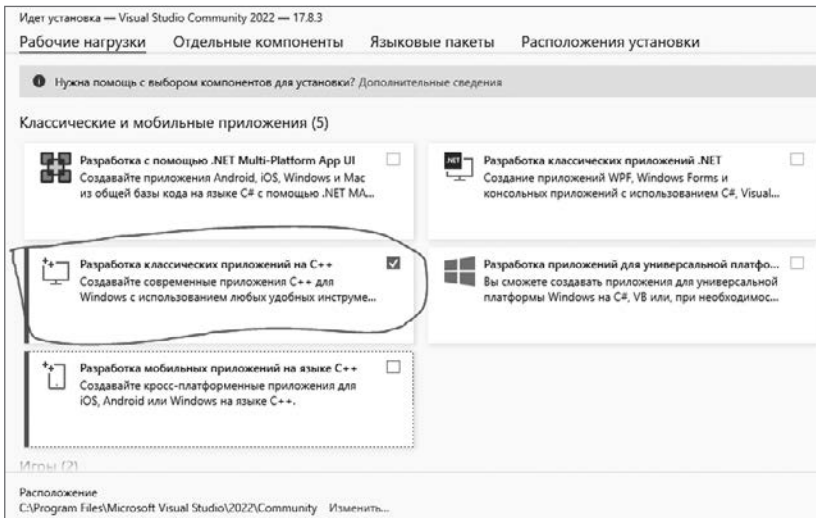


Рис. 0.5. Установка Visual Studio Community

Если планируете погружаться в мобильную разработку, то для Android необходимо скачать Android Studio и смириться с тем, что под iOS на Windows или Linux написать какое-либо приложение с наличием платформозависимого кода, а особенно

произвести сборку, не получится. Это связано с закрытостью экосистемы Apple. Поэтому для этих целей часто используется ноутбук с операционной системой macOS — его наличие позволяет одновременно вести разработку как под iOS, так и под Android. Но, так как не у всех есть возможность купить такой ноутбук, будем довольствоваться тем, что есть.

Далее нам необходимо установить и настроить Android SDK и Android Studio. Этот процесс вынесен в подраздел 0.1.4, так как он общий для всех операционных систем.

Чтобы проверить наличие всех компонентов для полноценной разработки приложений на Flutter в рамках используемой операционной системы, введите в терминале `flutter doctor` без дополнительных флагов и параметров:

```
C:\Users\MADTeacher>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[v] Flutter (Channel stable, 3.16.8, on Microsoft Windows [Version 10.0.19045.4046],
    locale ru-RU)
[v] Windows Version (Installed version of Windows is version 10 or higher)
[v] Android toolchain - develop for Android devices (Android SDK version 33.0.2)
[v] Chrome - develop for the web
[v] Visual Studio - develop Windows apps (Visual Studio Community 2022 17.8.3)
[v] Android Studio (version 2023.1)
[v] IntelliJ IDEA Community Edition (version 2023.1)
[v] VS Code (version 1.87.0)
[v] Connected device (3 available)
[v] Network resources

• No issues found!
```

Если такие компоненты, как Chrome или IntelliJ, помечены восклицательным знаком — ничего страшного, а вот отсутствие других (Android toolchain, Android Studio) — тревожный звоночек.

0.1.2. ОС macOS

В отличие от установки Flutter в Windows, для macOS нужна небольшая дополнительная подготовка. Если у вас компьютер на Apple Silicon, сначала необходимо настроить Rosetta 2. Для этого требуется открыть приложение Terminal и выполнить следующую команду:

```
sudo softwareupdate --install-rosetta --agree-to-license
```

После этого понадобится установить дополнительное программное обеспечение, а именно IDE Xcode и менеджер зависимостей для приложений под iOS, macOS и другие операционные системы от Apple — CocoaPods.

Xcode — IDE от Apple, которая нужна для разработки приложений для iOS, iPadOS, macOS, watchOS и других ОС. С ней также поставляются:

- система контроля версий git;
- SDK для операционных систем от Apple;
- языки программирования Swift и Objective-C;
- симуляторы iOS, iPadOS, watchOS и tvOS.

В установке Xcode нет ничего сложного, так как его можно скачать с помощью Apple App Store. Для этого воспользуйтесь поиском или перейдите в раздел **Разработка** в меню справа (рис. 0.6).

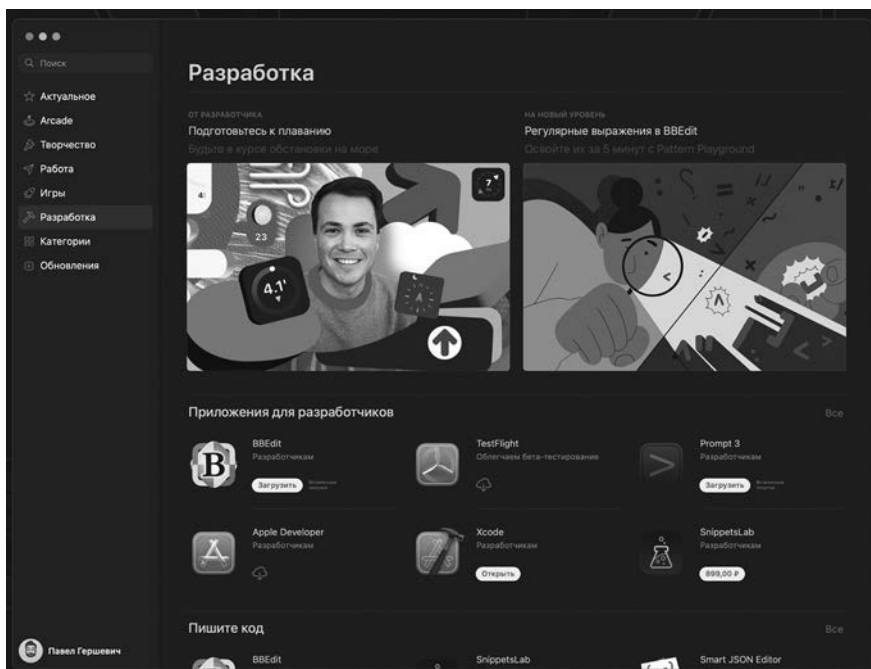


Рис. 0.6. Раздел Разработка в App Store на macOS

Для скачивания и установки Xcode (рис. 0.7) понадобится некоторое время, так как он занимает довольно много места на устройстве. После завершения установки перейдите в терминал и запустите команду для настройки командной строки:

```
sudo sh -c 'xcode-select -s /Applications/Xcode.app/Contents/Developer && xcodebuild -runFirstLaunch'
```

Следующим действием необходимо принять лицензию Xcode. Сделать это можно, не выходя из терминала:

```
sudo xcodebuild -license
```

Если ваш компьютер на Apple Silicon, то перед установкой CocoaPods нужно выполнить еще одну команду:

```
sudo gem install ffi
```

После чего выполните уже саму установку CocoaPods:

```
sudo gem install cocoapods
```

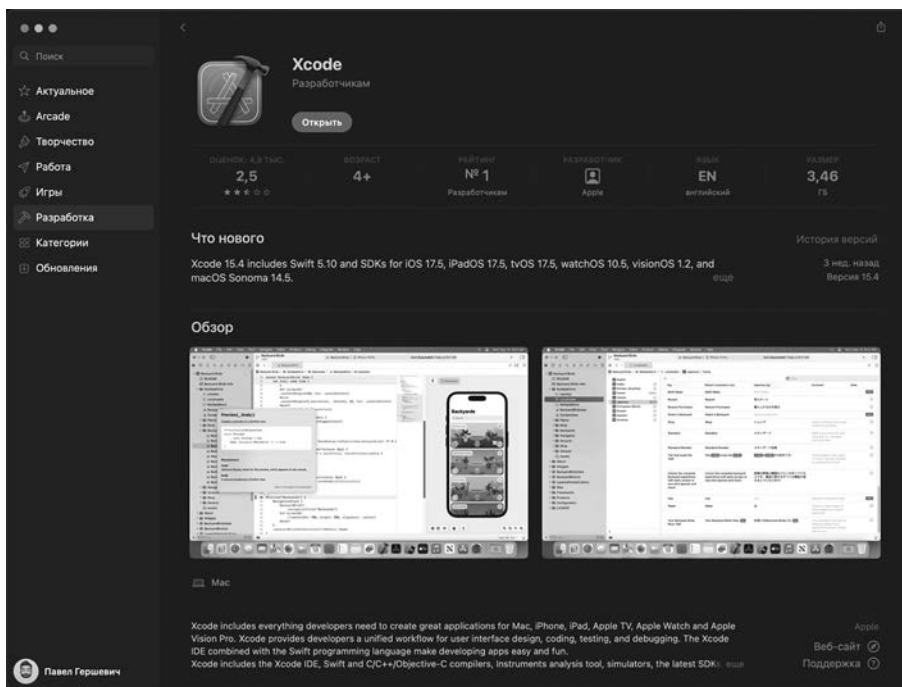


Рис. 0.7. Страница IDE Xcode в App Store на macOS

Теперь необходимо прописать путь до установленного нами CocoaPods в переменной `PATH`. macOS использует несколько различных терминалов в зависимости от версии. Мы рассмотрим тот, что присутствует в последних, — `zsh`. Для начала проверьте, существует ли файл `~/ .zshenv`, и если он отсутствует, то создайте. После чего откройте его любым текстовым редактором, будь то TextEdit или встроенный в терминал `nano`, и введите туда новую строку:

```
export PATH=$HOME/.gem/bin/:$PATH
```

где переменная `$HOME` — путь к нашему пользователю, переменная `$PATH` — та, к которой мы добавляем CocoaPods.

Сохраните файл и перезагрузите терминал. Для этого есть два способа: просто полностью выключить все текущие сессии терминала или выполнить команду:

```
source ~/.zshenv
```

CocoaPods можно установить и по-другому — с помощью пакетного менеджера Homebrew, который способен помочь и с установкой Flutter на компьютер. Для его установки введите в терминале следующие команды (их можно найти на сайте Homebrew <https://brew.sh/>):

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Как только на компьютере появляется Homebrew, для установки CocoaPods потребуется всего одна команда:

```
brew install cocoapods
```

Настала пора приступить к установке Flutter. Как и в случае с ОС Windows, перейдите на сайт фреймворка в раздел **Get started** или по следующей ссылке: <https://docs.flutter.dev/get-started/install>.

В нашем случае сайт сам подсветит выбранную macOS (рис. 0.8).

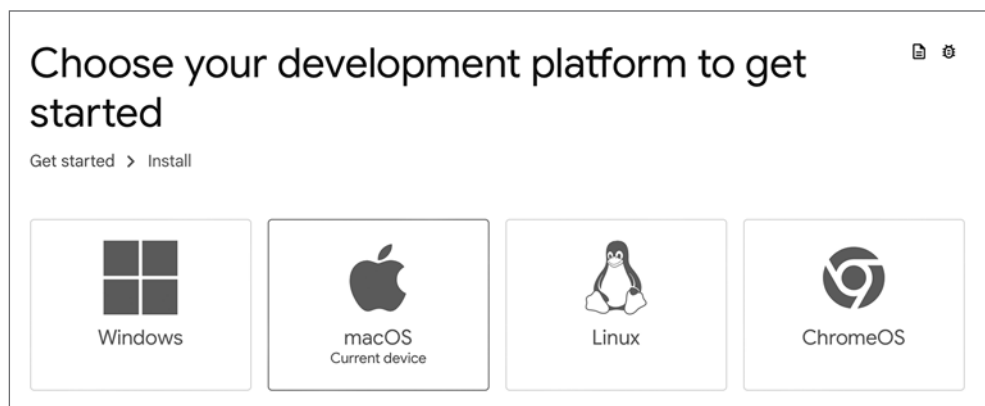


Рис. 0.8. Выбор ОС, в которой будет вестись разработка

На следующем шаге выберите предлагаемую по умолчанию конфигурацию запуска новых проектов на Flutter (рис. 0.9).

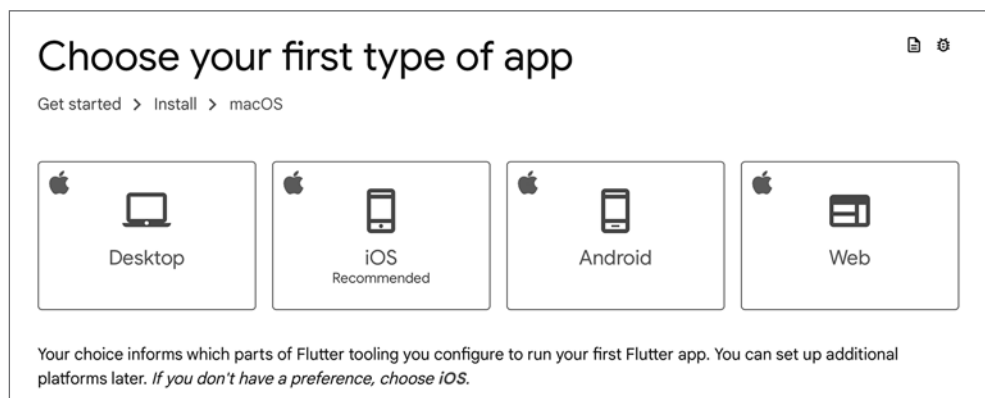


Рис. 0.9. Выбор конфигурации запуска новых проектов на Flutter

Далее найдите раздел **Install the Flutter SDK**, выберите **Download and install** и скачайте архив с актуальной версией фреймворка под ваш процессор (рис. 0.10).

Install the Flutter SDK

To install the Flutter SDK, you can use the VS Code Flutter extension or download and install the Flutter bundle yourself.

Use VS Code to install Download and install

Download then install Flutter

To install Flutter, download the Flutter SDK bundle from its archive, move the bundle to where you want it stored, then extract the SDK.

1. Download the following installation bundle to get the latest stable release of the Flutter SDK.

Intel Processor

flutter_macos_3.22.2-stable.zip

Apple Silicon

flutter_macos_arm64_3.22.2-stable.zip

For other release channels, and older builds, check out the SDK archive.

The Flutter SDK should download to the macOS default download directory: ~/Downloads/.

Рис. 0.10. Скачивание архива с актуальной версией Flutter

Документация Flutter рекомендует создавать каталог и распаковывать архив с помощью терминала, но никто не запрещает сделать это вручную. Обычно каталогом для распаковки выступает папка **development** в корневом каталоге вашего пользователя. Для его создания и распаковки архива воспользуемся командами:

```
mkdir ~/development
unzip ~/Downloads/flutter_macos_arm64_3.22.2-stable.zip \ -d ~/development
```

На следующем шаге необходимо прописать в уже знакомом нам по предыдущим шагам файле `~/ .zshenv` путь до распакованного Flutter SDK в переменной `PATH`. Откройте его любым удобным способом и добавьте в конец строку:

```
export PATH=$HOME/development/flutter/bin:$PATH
```

После сохранения файла не забудьте про перезагрузку терминала, которую можно выполнить, введя следующую команду:

```
source ~/.zshenv
```

Второй способ установки Flutter — с помощью Homebrew. Если вы уже установили Homebrew для CocoaPods (см. ранее), то для установки Flutter SDK достаточно вызвать одну команду:

```
brew install --cask flutter
```

и дождаться завершения процесса.

Если хотите запускать Flutter-приложения на Android, то вам необходимы Android SDK и Android Studio, установка и настройка которого описаны в подразделе 0.1.4.

ВАЖНОЕ ЗАМЕЧАНИЕ!

Для работы с Android требуется JDK, официальные версии которого работают только на процессорах Intel. Поэтому, если у вас компьютер на Apple Silicon, то необходимо установить Azul Zulu OpenJDK 17 под эту архитектуру. Скачать его (рис. 0.11) можно на сайте Azul по ссылке <https://www.azul.com/downloads/?version=java-17-lts&os=macos&architecture=arm-64-bit&package=jdk#zulu>.

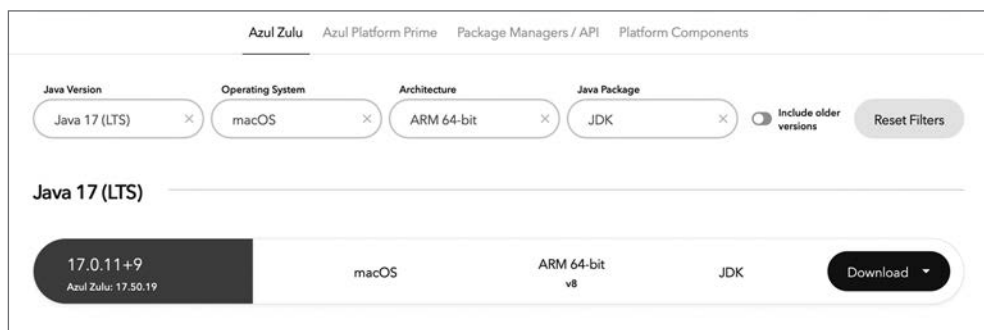


Рис. 0.11. Скачивание Azul Zulu OpenJDK 17 под ARM

Для разработки под iOS у нас практически все готово, осталось только докачать необходимые для работы симуляторов файлы. Это делается такой командой:

```
xcodebuild -downloadPlatform iOS
```

Если есть необходимость запуска на реальном iPhone или iPad, то сначала нужно купить подписку Apple Development Program, иначе ничего не получится. Правила у Apple довольно строгие, и это необходимо, чтобы никто не мог навредить своему устройству.

Чтобы проверить наличие всех компонентов для полноценной разработки приложений на Flutter в рамках используемой операционной системы, введите в терминал `flutter doctor` без дополнительных флагов и параметров:

```
fognature@MBP-Pavel-2 ~ % flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[v] Flutter (Channel stable, 3.22.2, on macOS 14.0 23A344 darwin-arm64, locale ru-RU)
[v] Android toolchain - develop for Android devices (Android SDK version 33.0.1)
[v] Xcode - develop for iOS and macOS (Xcode 15.4)
[v] Chrome - develop for the web
[v] Android Studio (version 2023.1)
[v] VS Code (version 1.89.0)
[v] Connected device (4 available)
[v] Network resources
```

- No issues found!

Если такие компоненты, как Chrome или IntelliJ, помечены восклицательным знаком — ничего страшного, а вот отсутствие других (Android toolchain, Android Studio, Xcode) — тревожный звоночек.

0.1.3. ОС Ubuntu 20.04 LTS+ и Debian 11+

Как и в предыдущих случаях, перейдите на сайт фреймворка в раздел **Get started** или по следующей ссылке: <https://docs.flutter.dev/get-started/install> и выберите вариант установки Flutter в Linux. Так как мы рассмотрим установку Flutter с конфигурацией, позволяющей разрабатывать под Desktop и Android (для Web нужно будет только установить браузер Chrome), то нет разницы, что вы выберете на этом шаге (рис. 0.12).

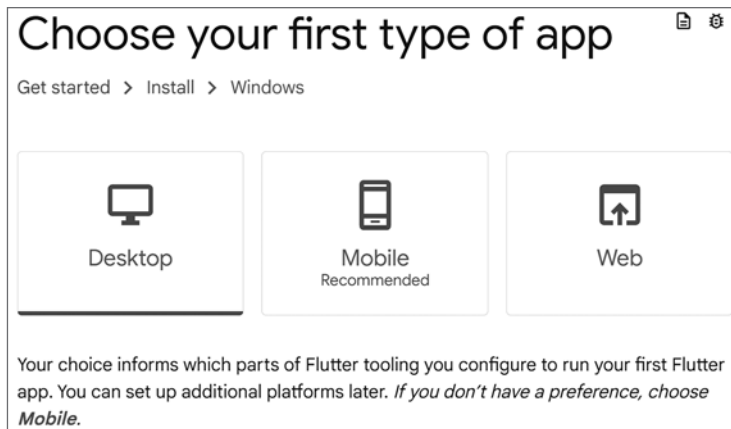


Рис. 0.12. Выбор конфигурации запуска новых проектов на Flutter

Далее найдите раздел **Install the Flutter SDK**, выберите **Download and install** и скачайте архив с актуальной версией фреймворка под ваш процессор (рис. 0.13).

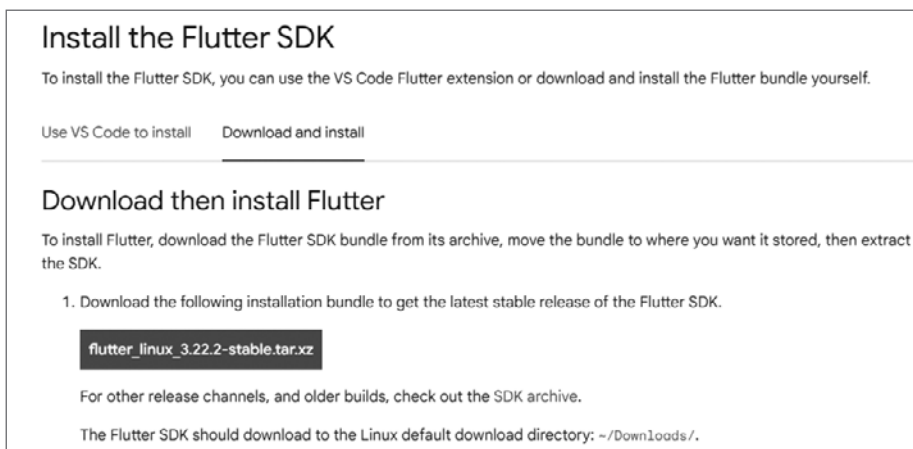


Рис. 0.13. Скачивание архива с актуальной версией Flutter

Первым делом после скачивания Flutter откройте терминал и убедитесь, что у вас установлены следующие инструменты:

```
which bash file mkdir rm which
/usr/bin/bash
/usr/bin/file
/usr/bin/mkdir
/usr/bin/rm
/usr/bin/which
```

Далее установим необходимые пакеты:

```
sudo apt-get update -y && sudo apt-get upgrade -y;
sudo apt-get install -y curl git unzip xz-utils zip libglu1-mesa
```

Для одновременной разработки приложений под Desktop и Android нам потребуется установить довольно большой набор пакетов:

```
sudo apt-get install \
  clang cmake git \
  ninja-build pkg-config \
  libgtk-3-dev liblzma-dev \
  libstdc++-12-dev \
  libc6:i386 libncurses5:i386 \
  libstdc++6:i386 lib32z1 \
  libbz2-1.0:i386
```

Если вы уже скачали Visual Studio Code, то переходите к следующему шагу, в противном случае сделайте это, перейдя по ссылке <https://code.visualstudio.com/download>.

В нашем случае был скачан пакет `code_1.90.1-1718141439_amd64.deb`. Для его установки необходимо перейти в терминале в каталог с загрузками, после чего ввести команду:

```
x@x-pc:~/Загрузки$ sudo apt install ./code_1.90.1-1718141439_amd64.deb
```

Для распаковки Flutter в каталог `/usr/bin/` введите в терминале (находясь в каталоге с загрузками) следующую команду:

```
x@x-pc:~/Загрузки$ sudo tar -xf ./flutter_linux_3.22.2-stable.tar.xz -C /usr/bin/
```

Чтобы добавить Flutter в переменную среды `PATH`, первым делом проверьте, какая оболочка консоли используется в вашей операционной системе:

```
x@x-pc:~/Загрузки$ echo $0
bash
```

Если `bash`, то для указания путей до Flutter используйте команду:

```
echo 'export PATH="/usr/bin/flutter/bin:$PATH"' >> ~/.bash_profile
```

Для `zsh`:

```
echo 'export PATH="/usr/bin/flutter/bin:$PATH"' >> ~/.zshenv
```

После чего закройте терминал. При наличии другой оболочки обратитесь к официальной документации Flutter: <https://docs.flutter.dev/get-started/install/linux/android?tab=download#add-flutter-to-your-path>.

Теперь перейдем к небольшим пляскам с бубном, а именно к установке Android Studio, скачать который можно по следующей ссылке: <https://developer.android.com/studio>.

По завершении скачивания распакуйте архив и откройте в терминале каталог с загрузками, переместив Android Studio в `/usr/local/`:

```
x@x-pc:~/Загрузки$ sudo mv ./android-studio /usr/local/
```

Сам процесс установки и последующего запуска IDE состоит из нескольких команд:

```
x@x-pc:~/Загрузки$ cd /usr/local/android-studio/bin/  
x@x-pc:/usr/local/android-studio/bin$ ./studio.sh
```

После завершения установки Android Studio ее необходимо перезапустить, а также инсталлировать дополнительные компоненты. Для этого на стартовом экране IDE нажмите на **More Actions** и выберите запуск **SDK Manager** (рис. 0.14).

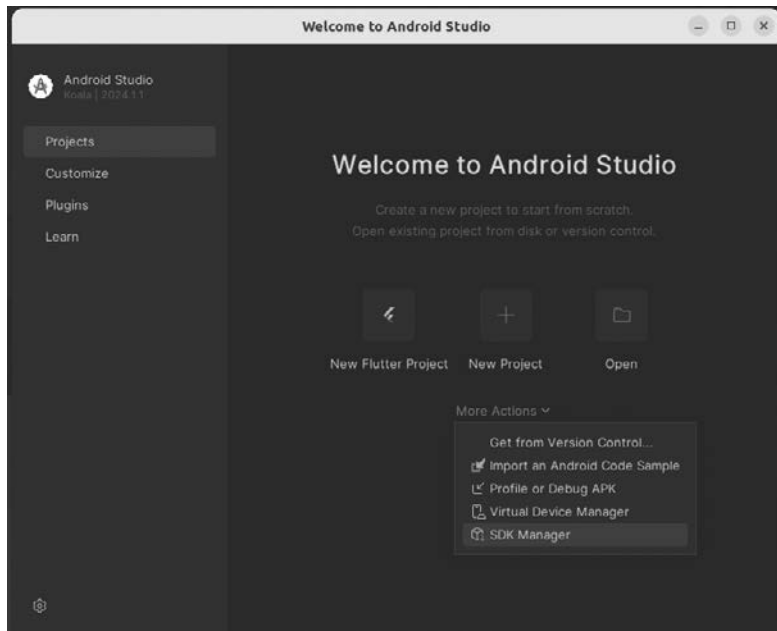


Рис. 0.14. Запуск SDK Manager

Далее перейдите на вкладку **SDK Tools** и убедитесь, что установлены все компоненты, выделенные галочками на рис. 0.15.

В противном случае поставьте галочки напротив:

- Android SDK Command-line Tools;
- Android SDK Build-Tools;
- Android SDK Platform-Tools;
- Android Emulator.