Video app документация.

P.S. Я так же повставлял комментарии ко всему прямо в коде, так что если тут недостаточно информации, можно посмотреть в код.

Чтобы сделать приложение, я использовал библиотеки:

**camera:** ^0.4.2 #Для того, чтобы снимать на камеру.  
**rxdart:** ^0.18.0 #Для удобных rx контроллеров.  
**path\_provider:** ^0.5.0+1 #Для удобной работы с файловой системой  
**video\_player:** ^0.10.0+2 #Для воспроизведения снятых видео.  
**chewie:** ^0.9.6 # UI Обертка video\_player, сам по себе video\_player просто воспроизводит видео.

Из нативных только одну:

**implementation** **'com.googlecode.mp4parser:isoparser:1.1.21' // Для обработки mp4 (в моем случае для обрезки и склеивания).**

**Описание проекта:**

**blocs пакет:**

providers.dart - Корневой виджет, который хранит в себе компоненты логики. Чeрез него экране получают свои компоненты логики.

*camera\_bloc.dart* - компонент логики, который хранит в себе контроллеры (стримы) значений, необходимые объекты и методы для camera\_screen.dart.

**static List<CameraDescription> cameras; -** объекты, представляющие найденные операционной системой камеры.

**Stopwatch \_stopwatch = Stopwatch();** - секундомер для отсчета времени записи.  
**Timer \_timer; -** таймер, который каждую секунду добавляет в стрим секунд (\_timerController) текущее время.

**BehaviorSubject<bool> \_recordingStateController = BehaviorSubject<bool>(); -** стрим состояния записи. Его используют вью и данный компонент логики (компонент логики включает/выключает секундомер, вью немного изменяют UI, в зависимости о того, записывает или нет) **get recording => \_recordingStateController.stream; -** просто стрим из контроллера выше **Sink<bool> get recordingSink => \_recordingStateController.sink;** -просто объект, через который мы добавляем данные в стрим.

**BehaviorSubject<int> \_timerController = BehaviorSubject<int>();** - контроллер, по стриму которого "течет" текущее время записи. **get timerSecondsElapsed => \_timerController.stream;** - стрим контроллера

Методы startTimer(), stopTimer(), dispose() - старт отсчета времени, остановка отсчета времени, избавиться от ресурсов компонента.

*video\_list\_bloc.dart* - компонент логики, который хранит в себе контроллеры (стримы) значений, необходимые объекты и методы для video\_list\_screen.dart. Хранит стрим листа записанных и созданных видео. Далее я не буду так подробно описывать, как в прошлый раз, по факту все блоки - просто контейнеры стримов, методов для удобства и объектов. Обычно название метода объясняет, что делает метод.

*video\_player\_bloc.dart* - компонент логики, который хранит в себе контроллеры (стримы) значений, необходимые объекты и методы для video\_player\_screen.dart. Хранит в себе стрим выбранных юзером моментов для обрезки видео. Так же содержит метод для сохранения видео вне пространства приложения и метод для добавления моментов, на которых юзер хочет обрезать видео.

**Screens пакет:**

*camera\_screen.dart* - экран для записи видео с камеры.

*screen\_values.dart* - просто контейнер констант, которые нужны для экранов.

*video\_list\_screen.dart* - экран с листом видео. Имеет два режима - режим сохранения и дефолтный. В режиме сохранения при нажатии на видео, оно сохраняется на девайс юзера, а в обычном режиме при нажатии видео открывается экран воспроизведения видео.

*video\_player\_screen.dart* - экран воспроизведения видео. Воспроизводит видео, а так же позволяет юзеру вырезать необходимые куски из видео.

**Корневой пакет:**

*main.dart* - файл с методом запуска корневого виджета, который запускает первый экран. Так же тут создаются темы для оформления приложения.

*native\_api.dart* - файл с классом, через который вызываются нативные методы.

*util.dart* - файл с классом со статик методами для удобства сохранения, удаления записанных видео.

Нативные android классы:

*MainActivity.java* - лаунчер флаттер приложения. Тут же при создании создает обработчика вызовов нативных методов с флаттера.

*NativeApi.java* - класс с реализацией нативных методов, а так же с названиями аргументов, которые нужны для получения их из вызова с флаттера.