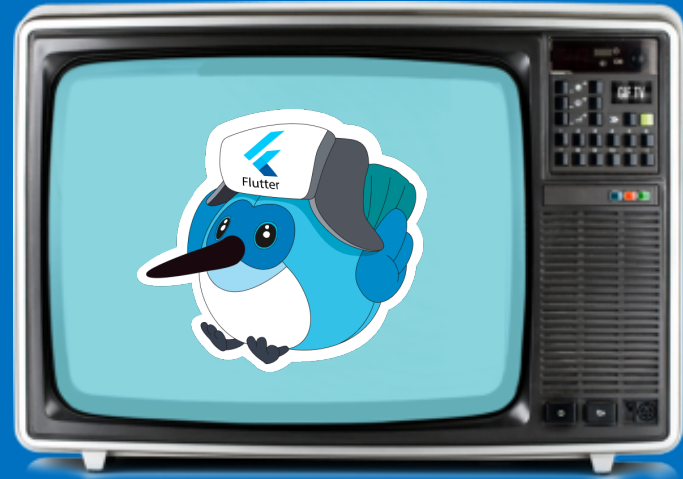


<epam>

# Flutter for TV



Aleksandr Denisov  
Advanced Software Engineer - EPAM  
Flutter and Dart GDE





Mobile



Web



Flutter



Desktop



Embedded



## Does Flutter support Android TV and Apple TV?

Asked 1 year, 3 months ago   Active 1 year, 3 months ago   Viewed 3k times



1



**No support for Apple TV or Android TV** at the moment from Flutter, and according to [this thread](#) there is no current plan for that (as of Jul 21, 2018). Actually theoretically, there is nothing stopping you from compiling the project and building Android TV app (can't say much about Apple TV), the binary will build to the device, but that's just the tip of the iceberg. Since **no official support** is there yet, presumably you'll be on your own if you try.



# TV doesn't support by Flutter

## Support making Apple TV apps #2058

 Closed Hixie opened this issue on Feb 20, 2016 · 17 comments

## Flutter does not support TV device #13633

 Closed coderJohnZhang opened this issue on Dec 16, 2017 · 12 comments

## Android TV: No support for getting the focused widget's coordinate #13749

 Closed coderJohnZhang opened this issue on Dec 22, 2017 · 16 comments


## How to use flutter for AndroidTV Development #26947

 Closed Scorpivolol opened this issue on Jan 23, 2019 · 1 comment

## Crash on Amazon FireTv #33646

 Closed Inkain opened this issue on May 31, 2019 · 3 comments

## Roku support #37159

 Open khat33b opened this issue on Jul 29, 2019 · 1 comment

## How to publish Flutter apps made for Android TV on PlayStore #48109

 Closed pharshdev opened this issue on Jan 3, 2020 · 8 comments

## call to unavailable function 'fork': not available on tvOS #39918

 Open lesnitsky opened this issue on Dec 24, 2019 · 15 comments

## Add Support for tvOS #47928

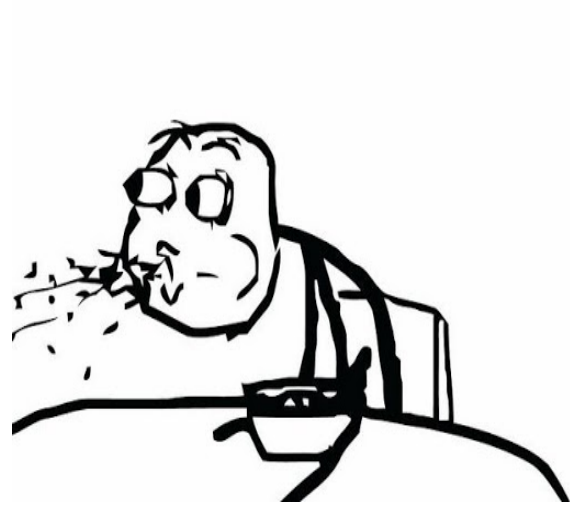
 Closed move  bot opened this issue on Dec 28, 2019 · 5 comments

# Multiplatform solution

---

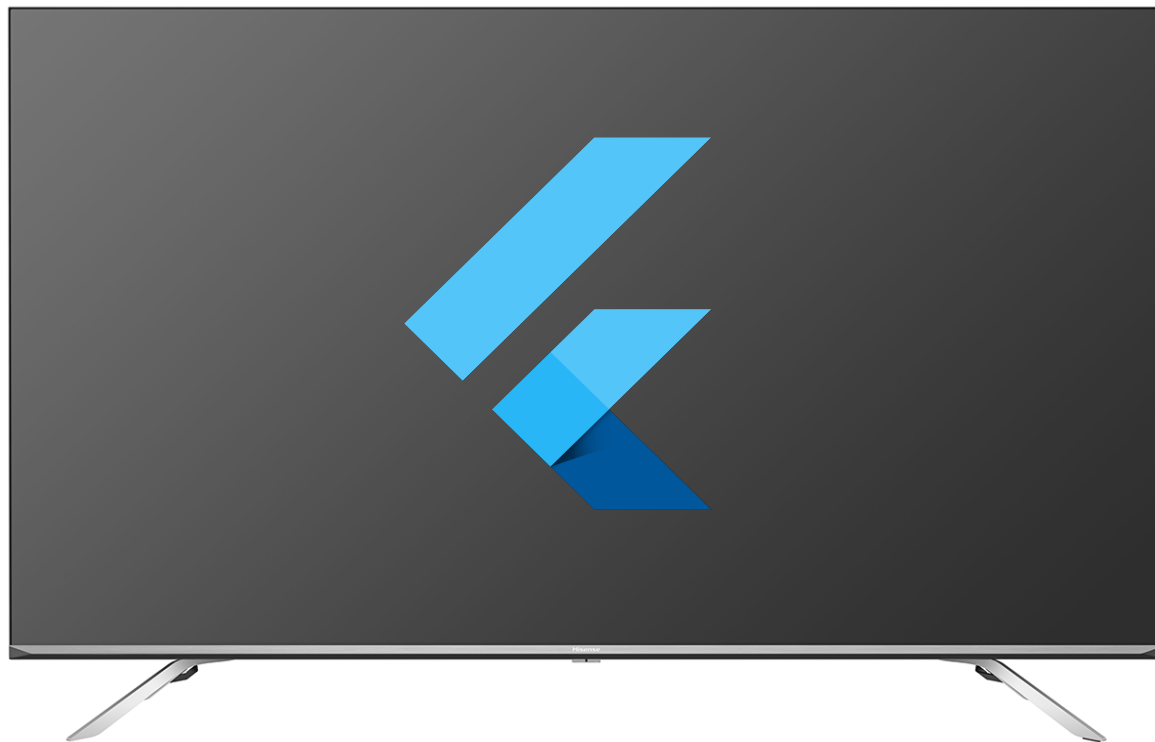
- Android
- iOS
- Web
- Embedded (Linux based box)
- Android TV
- Apple TV

6 platforms



# Flutter for TV platforms

---

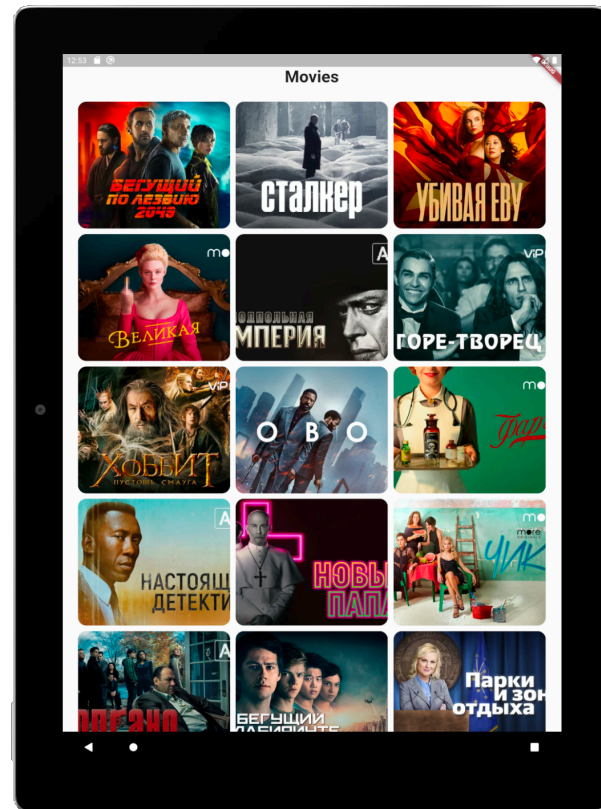
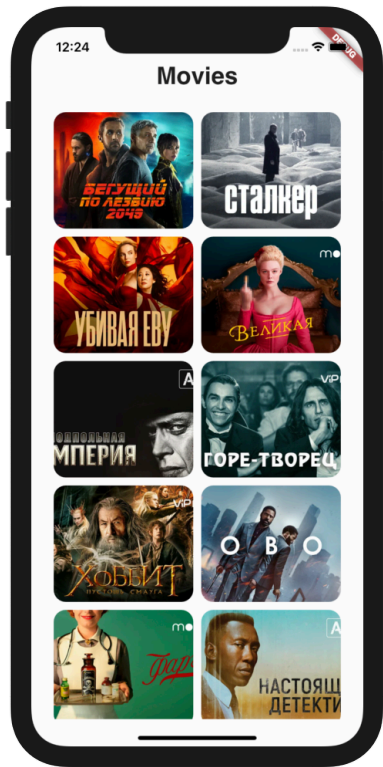


# How many TV Platforms are there?

---



# Demo application



LET'S TRY TO LAUNCH

The logo for Android TV, featuring the word "android" in a green, lowercase, sans-serif font, followed by "tv" in a dark grey, lowercase, sans-serif font. The entire logo is centered within a light grey rectangular background.



# Android TV

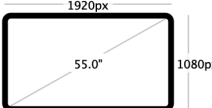
Virtual Device Configuration

## Select Hardware

Choose a device definition

| Category   | Name               | Play Store | Size  | Resolution | Density |
|------------|--------------------|------------|-------|------------|---------|
| TV         | Android TV (720p)  |            | 55.0" | 1280x...   | tvdpi   |
| Phone      | Android TV (1080p) |            | 55.0" | 1920x...   | xhdpi   |
| Wear OS    |                    |            |       |            |         |
| Tablet     |                    |            |       |            |         |
| Automotive |                    |            |       |            |         |

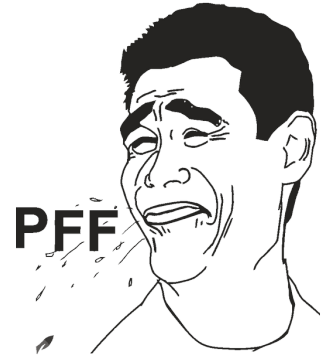
Android TV (1080p)

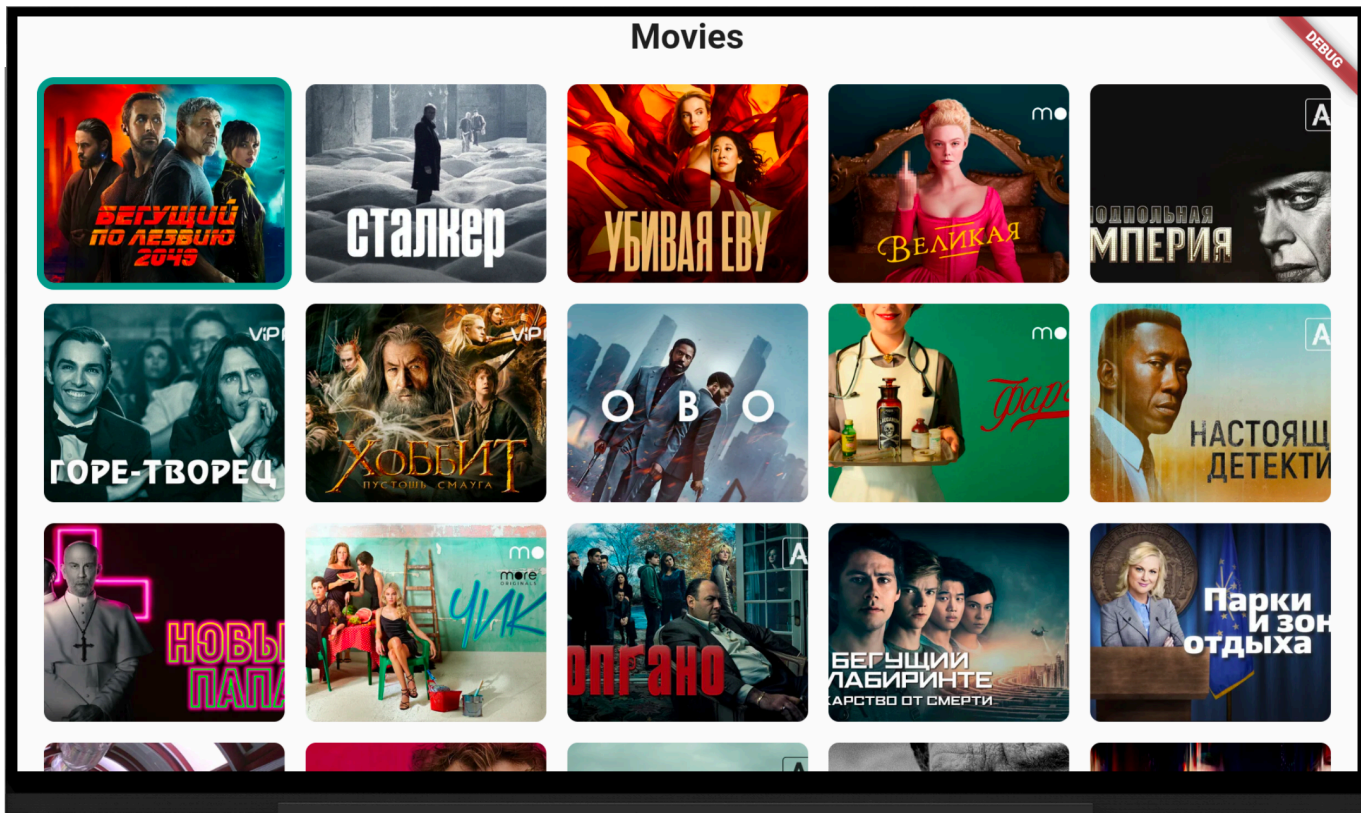


Size: xlarge  
Ratio: long  
Density: xhdpi

Clone Device...

Cancel Previous Next Finish







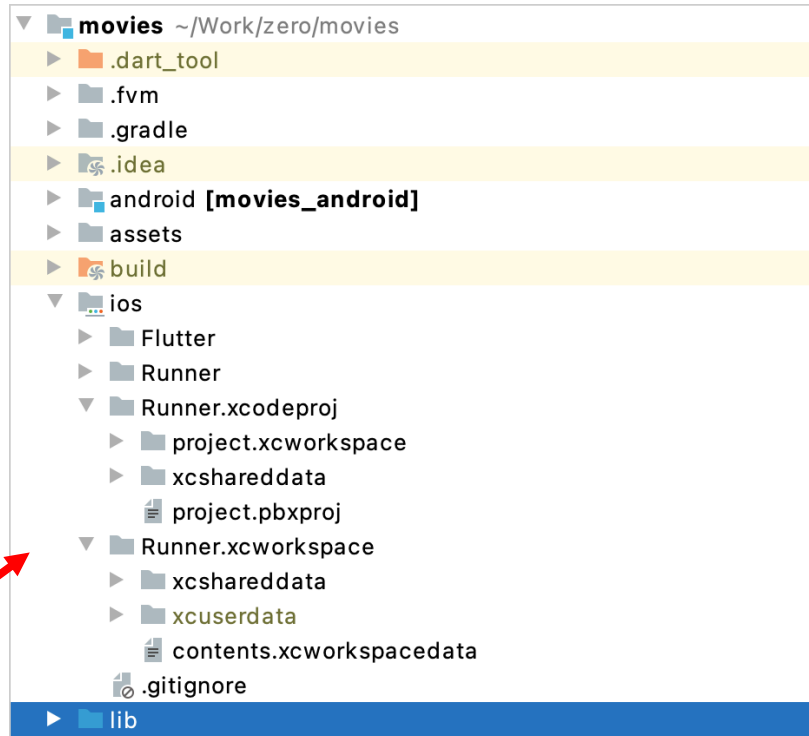
# Apple TV

```
Launching lib/main.dart on Apple TV in debug mode...

Running Xcode build...
└─Compiling, linking and signing...          11.1s
Xcode build done.                            48.8s
Unable to install
/Users/aleksandr_denisov/Work/zero/episode_1/build/ios/iphonesimulator/Runner.ap
p on A3627C63-CE2E-463A-911B-4D55B40C7A92. This is sometimes caused by a
malformed plist file:
ProcessException: Process exited abnormally:
An error was encountered processing the command
(domain=IXUserPresentableErrorDomain, code=6):
Unable To Install "episode_1"
This app is not made for this device.
This app was not built to support this device family; app is compatible with (
  1,
  2
) but this device supports (
  3
)
Underlying error (domain=MIInstallerErrorDomain, code=10):
  This app was not built to support this device family; app is compatible with (
    1,
    2
  ) but this device supports (
    3
  )
Command: /usr/bin/xcrun simctl install A3627C63-CE2E-463A-911B-4D55B40C7A92
/Users/aleksandr_denisov/Work/zero/episode_1/build/ios/iphonesimulator/Runner.
app
Error launching application on Apple TV. _
```



# Apple TV



# Inside the Flutter Engine

---

tvOS  $\neq$  iOS



# Contributing to Flutter

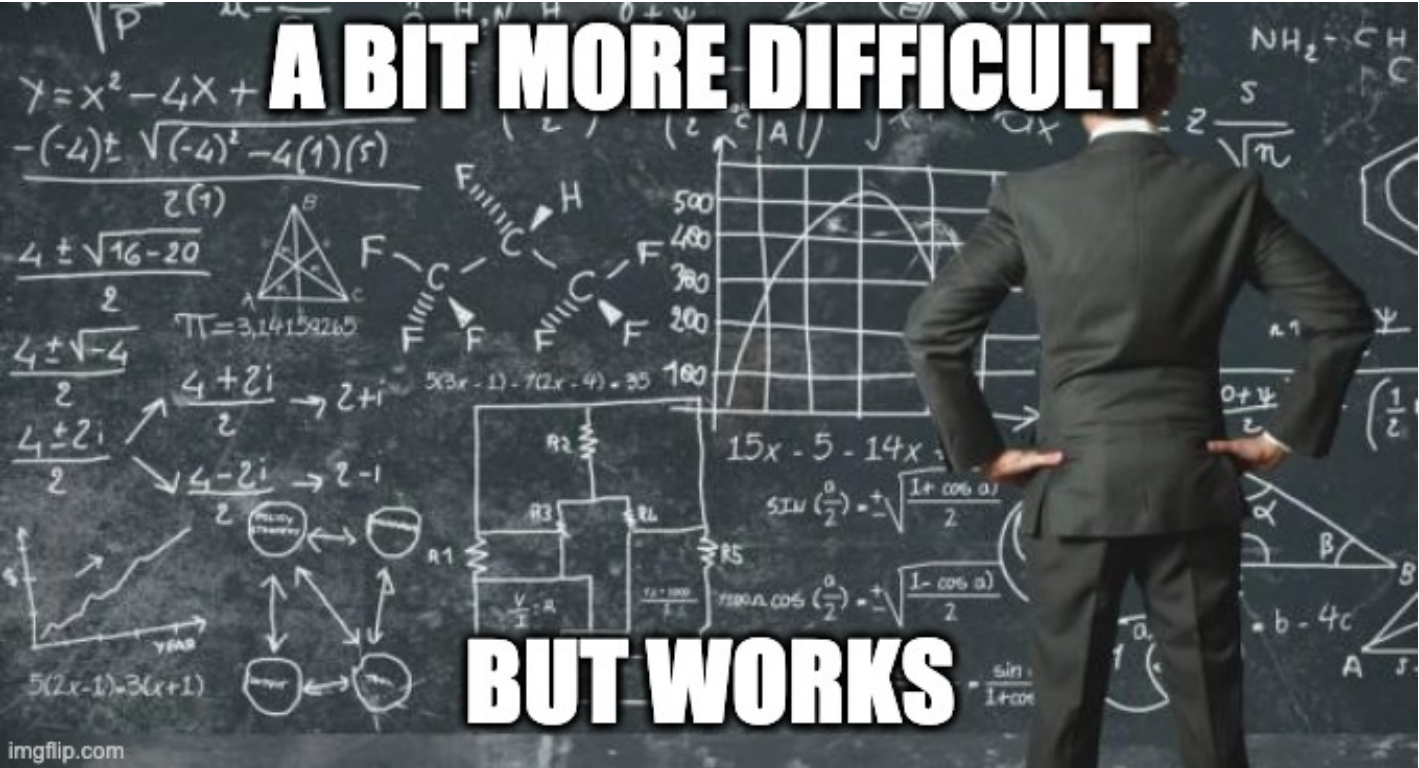
---

## Developing for Flutter

1. Our code of conduct
2. Values, which talks about what we care most about
3. Setting up your engine development environment
4. Setting up your framework development environment
5. Tree hygiene
6. Issue hygiene
7. Our style guide
8. Flutter design doc template

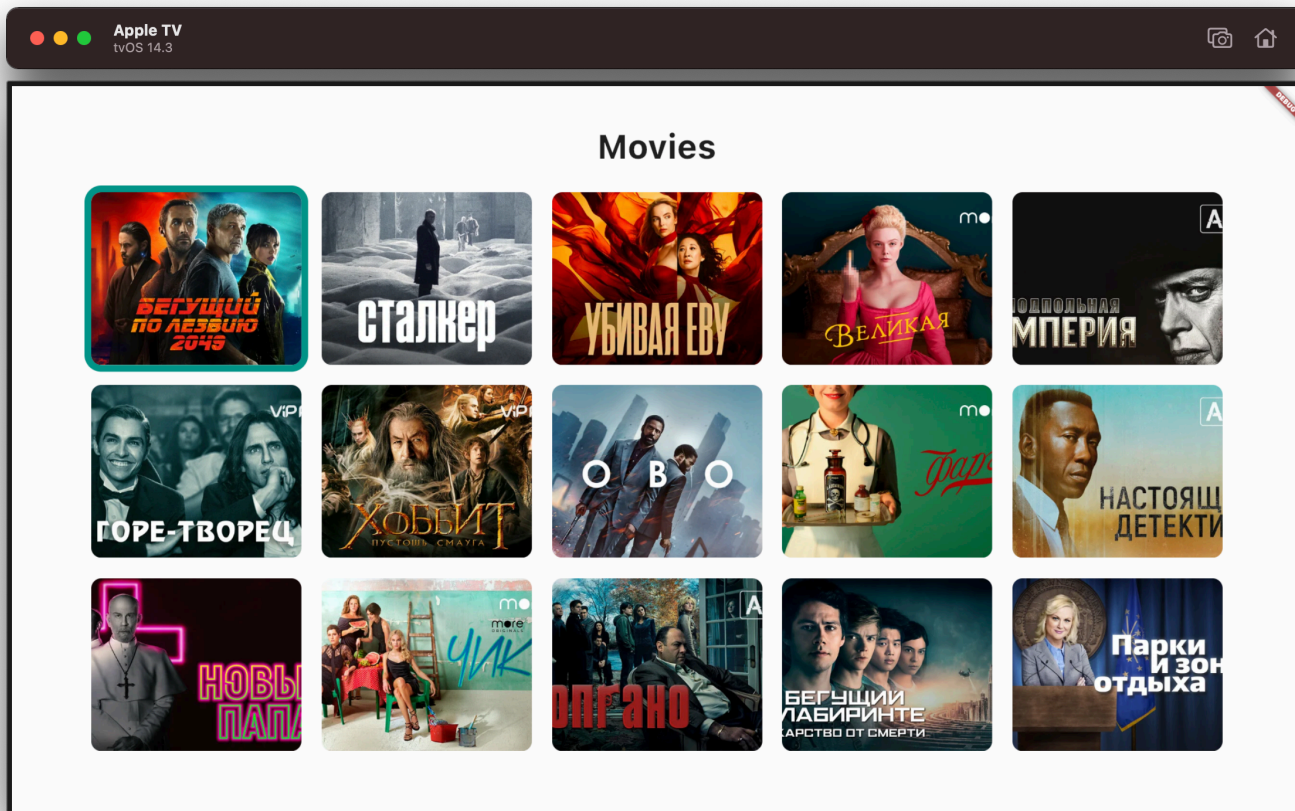
<https://github.com/flutter/flutter/blob/master/CONTRIBUTING.md>

# Update (hack) the Flutter Framework and Flutter Engine





# Let's check





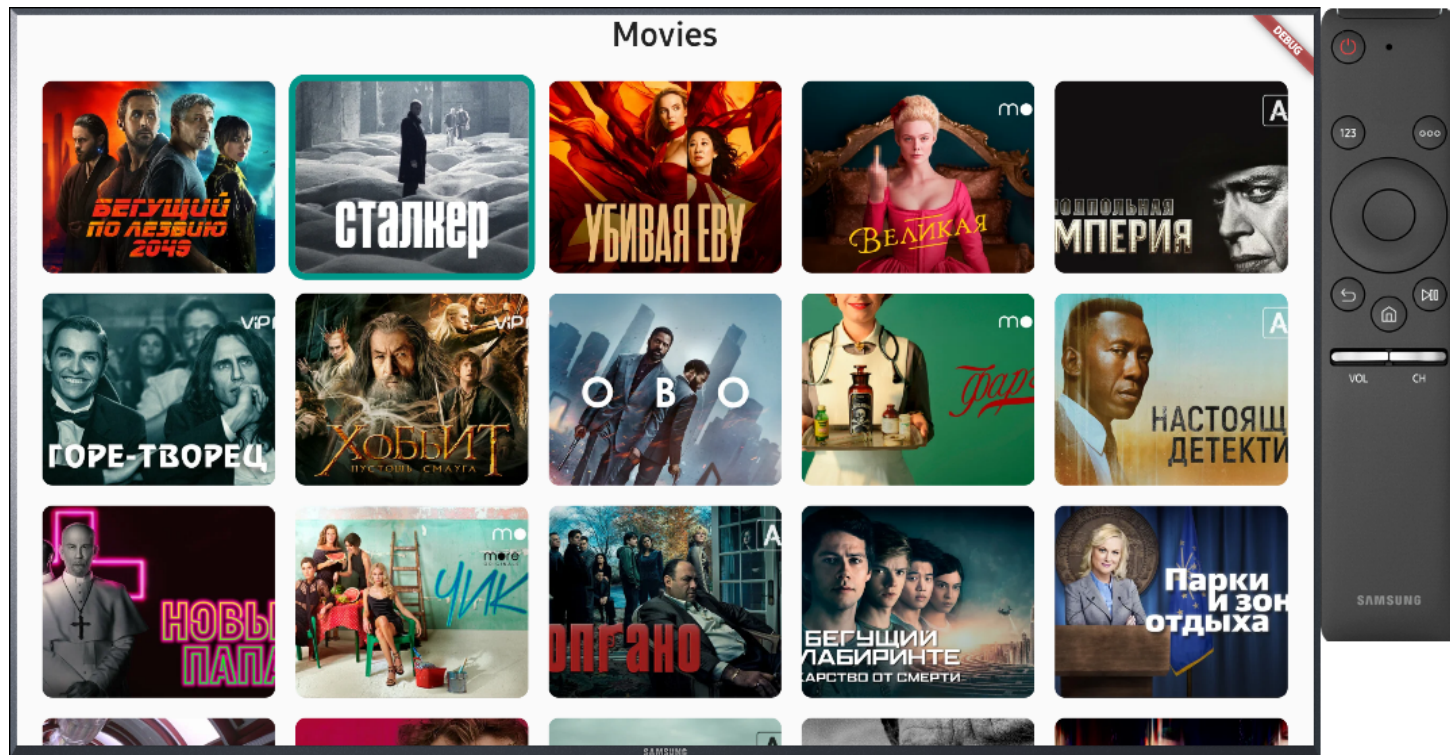
## Flutter for Tizen

```
flutter-tizen help  
flutter-tizen create  
flutter-tizen doctor  
flutter-tizen devices  
flutter-tizen build tpk --target-arch arm,x86  
flutter-tizen run -d tizen
```

<https://github.com/swift-kim/flutter-tizen>



# Tizen is ready





webOS

**Roku TV**

THAT'S NOT ENOUGH

# A lot of questions there

---

- How to detect the platform?
- How to organize the user interaction?
- Which plugins are supported and which are not?



## HOW TO DIFFERENTIATE PLATFORMS

# How to differentiate platforms

android   
androidtv

Platform.isAndroid == true

 iOS     tv

Platform.isIOS == true

```
class MyPlatform {  
    static const tvMode = String.fromEnvironment('TV_MODE');  
  
    static bool get isTv => tvMode == 'ON';  
  
    static bool get isIOS => !isTv && Platform.isIOS;  
  
    static bool get isAndroid => !isTv && Platform.isAndroid;  
  
    static bool get isTVOS => isTv && Platform.isIOS;  
  
    static bool get isAndroidTV => isTv && Platform.isAndroid;  
}
```

## HOW TO ORGANIZE USER INTERACTION

# Android TV remote control



RawKeyboard

# Android TV remote control

---

- [RawKeyDownEvent](#) and [RawKeyUpEvent](#), the classes used to describe specific raw key events.
- [RawKeyboardListener](#), a widget that listens for raw key events.
- [SystemChannels.keyEvent](#), the low-level channel used for receiving events from the system.

# Focus

---

## Focus class Null safety

A widget that manages a [FocusNode](#) to allow keyboard focus to be given to this widget and its descendants.



# Android TV remote control



```
RawKeyboard.instance.addListener((event) {  
  if (LogicalKeyboardKey.arrowLeft == event.logicalKey) {  
    _moveLeft();  
  } else if (LogicalKeyboardKey.arrowRight == event.logicalKey) {  
    _moveRight();  
  } else if (LogicalKeyboardKey.arrowUp == event.logicalKey) {  
    _moveUp();  
  } else if (LogicalKeyboardKey.arrowDown == event.logicalKey) {  
    _moveDown();  
  }  
}
```

# Android TV remote control

---

```
void _moveRight() {  
    FocusManager.instance.primaryFocus.focusInDirection(TraversalDirection.right);  
}
```



# Focus

---

## Focus class Null safety

A widget that manages a [FocusNode](#) to allow keyboard focus to be given to this widget and its descendants.



# FocusableWidget

```
class FocusableWidget extends StatefulWidget {
```

```
...
```

```
@override
```

```
Widget build(BuildContext context) {
```

```
  var focusChild = widget.child;
```

```
  return Focus(
```

```
    autofocus: true,
```

```
    key: _key,
```

```
    onFocusChange: (value) =>
```

```
      setState(() {
```

```
        _isFocused = value;
```

```
      }));
```

```
    onKey: _onKeyHandler,
```

```
    child: focusChild,
```

```
  );
```

```
  bool _onKeyHandler(FocusNode node, RawKeyEvent event) {
```

```
    if (event is RawKeyDownEvent) {
```

```
      return true;
```

```
    }
```

```
    if (event.hasSubmitIntent) {
```

```
      _onSubmit();
```

```
      return true;
```

```
    }
```

```
    return false;
```

```
  }
```

# Focus

---

```
extension SubmitAction on RawKeyEvent {  
  
    bool get hasSubmitIntent =>  
        this is RawKeyUpEvent && (logicalKey == LogicalKeyboardKey.select);  
  
}
```

# The same behavior as on Mobile

```
void _extractSubmit(Element element) {  
  final widget = element.widget;
```

```
  if (widget is GestureDetector) {  
    _onSubmit = widget.onTap;  
    return;  
  }
```

```
  if (widget is InkWell) {  
    _onSubmit = widget.onTap;  
    return;  
  }
```

```
  element.visitChildElements(_extractSubmit);  
}
```

```
@override  
void initState() {  
  super.initState();  
  SchedulerBinding.instance.addPostFrameCallback((_) {  
    _key.currentContext.visitChildElements(_extractSubmit);  
  });  
}
```

# Apple TV remote control

---



- Remote doesn't have arrows, touchpad only
- iOS target doesn't handle touchpad taps as RawKeyboard events

# Update Engine

---

**GOTO:**



```
engine/src/flutter/shell/platform/darwin/ios  
/framework/Source/FlutterViewController.mm
```

- Create system channel for tap events (flutter/keyevent)
- Create platform channel for touchpad events
- Add handlers for remote control taps
- Update existing handler for touchpad events

# Update Engine

```
self.keyEventChannel =  
  [FlutterBasicMessageChannel  
   messageChannelWithName:@"flutter/keyevent"  
   binaryMessenger:[self binaryMessenger]  
   codec:[FlutterJSONMessageCodec sharedInstance]];
```

```
self.gamepadTouchEventChannel =  
  [FlutterBasicMessageChannel  
   messageChannelWithName:@"flutter/gamepadtouchevent"  
   binaryMessenger:[self binaryMessenger]  
   codec:[FlutterJSONMessageCodec sharedInstance]];
```

# Update Engine - key events handler

```
- (void)handleTap: (UITapGestureRecognizer *)recognizer
withType:(NSString *)keyMapType keyType:(int)key {
    if (recognizer.state == UIGestureRecognizerStateBegan) {
        [self sendTap:key withType:@"macos" ofType:@"keydown"];
    } else if (recognizer.state == UIGestureRecognizerStateEnded) {
        [self sendTap:key withType:@"macos" ofType:@"keyup"];
    }
}
```



# Update Engine - touch pad handler

```
- (void)touchesMoved:(NSSet*)touches withEvent:(UIEvent*)event {
#ifdef TARGET_OS_TV
    for (UITouch* touch in touches) {
        CGPoint location = [touch locationInView:self.view];
        [self sendGamepadTouchesWithType:@"move"
                                         x:location.x y:location.y];
    }
#else
    [self dispatchTouches:touches pointerDataChangeOverride:nullptr];
#endif
}
```

# Apple TV remote control


```
static const channel = BasicMessageChannel<dynamic>('flutter/gamepadtouchevent',  
                                                    JSONMessageCodec());
```

```
channel.setMessageHandler(_onTouch);
```

```
Future<void> _onTouch(dynamic arguments) {  
  final x = arguments['x'];  
  final y = arguments['y'];  
  final type = arguments['type'];
```

```
  if (type == 'started') {  
    ...  
  } else if (type == 'move') {  
    ...  
  } else if (type == 'ended') {
```

```
    if ((moveX.abs() >= 250) || (moveY.abs() >= 250)) {  
      if (moveX.abs() >= moveY.abs()) {  
        if (moveX >= 0) {  
          _moveLeft();  
        } else {  
          _moveRight();  
        }  
      } else {  
        if (moveY >= 0) {  
          _moveUp();  
        } else {  
          _moveDown();  
        }  
      }  
    }  
  }  
}
```



Хоббит. Пустошь Смауга

2013, фэнтези, приключения, США, 2 ч 41 мин

Продолжение путешествия хоббита Бильбо Бэггинса, волшебника Гэндальфа и 13 отважных гномов. Их компания должна добраться до Одинокой горы. А там они встретятся с величайшей опасностью из всех — созданием куда более ужасающим, чем все их прошлые противники, чудовищем, которое проверит на прочность не только их мужество, но крепость их дружбы и правильность выбранного пути, — драконом Смаугом.

**WHICH PLUGINS ARE SUPPORTED?**

# Plugins and Android TV

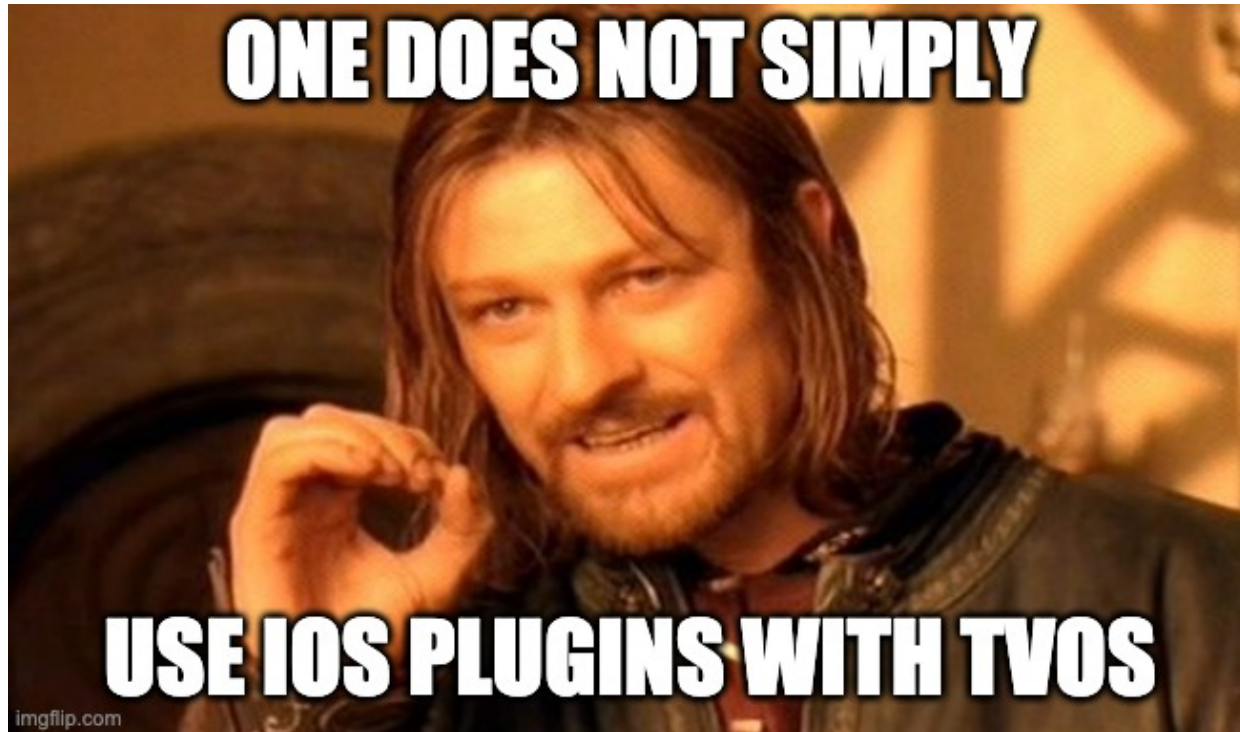
---

androidtv

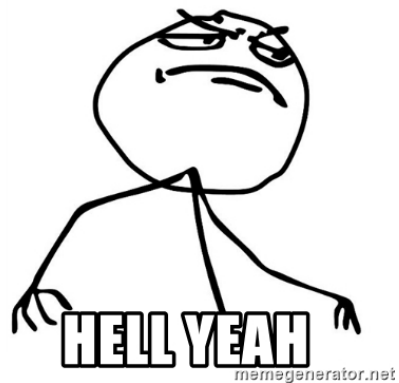


# Plugins and Tizen

| Package name                             | Original package                               | Pub        | Endorsed |
|--|--|------------|----------|
| <a href="#">battery_tizen</a>            | <a href="#">battery</a> (1st-party)            | pub v2.0.0 | No       |
| <a href="#">connectivity_tizen</a>       | <a href="#">connectivity</a> (1st-party)       | pub v2.0.0 | No       |
| <a href="#">device_info_tizen</a>        | <a href="#">device_info</a> (1st-party)        | pub v2.0.0 | No       |
| <a href="#">image_picker_tizen</a>       | <a href="#">image_picker</a> (1st-party)       | pub v2.0.0 | No       |
| <a href="#">integration_test_tizen</a>   | <a href="#">integration_test</a> (1st-party)   | pub v2.0.0 | No       |
| <a href="#">package_info_tizen</a>       | <a href="#">package_info</a> (1st-party)       | pub v2.0.0 | No       |
| <a href="#">path_provider_tizen</a>      | <a href="#">path_provider</a> (1st-party)      | pub v2.0.0 | No       |
| <a href="#">sensors_tizen</a>            | <a href="#">sensors</a> (1st-party)            | pub v2.0.0 | No       |
| <a href="#">share_tizen</a>              | <a href="#">share</a> (1st-party)              | pub v2.0.0 | No       |
| <a href="#">shared_preferences_tizen</a> | <a href="#">shared_preferences</a> (1st-party) | pub v2.0.0 | No       |
| <a href="#">url_launcher_tizen</a>       | <a href="#">url_launcher</a> (1st-party)       | pub v2.0.0 | No       |
| <a href="#">video_player_tizen</a>       | <a href="#">video_player</a> (1st-party)       | pub v1.0.0 | No       |
| <a href="#">wifi_info_flutter_tizen</a>  | <a href="#">wifi_info_flutter</a> (1st-party)  | pub v2.0.0 | No       |



# Final slide





## THE FINAL SLIDE



<https://github.com/denisovav/zero/movies>



<https://twitter.com/shuregdenisov>



<https://t.me/flutterdevpodcast>



<https://t.me/MobilePeopleTalks>

THANK YOU!  
QUESTIONS?