

# Android Wear: Create a watch face

This codelab will introduce you to the key concepts to creating your own watch face for Android Wear. It will then walk you through customising an analog watch face. There is also a bonus section on using the Palette API to automatically choose a color for the watch face if you have the time. By the end of the codelab, you'll have a customized watch face that you can call your own.

## Steps

The codelab is split into the following steps:

1. Concepts and Setup
2. Change the background
3. Change the watch hands
4. Account for ambient mode
5. Automatic color selection using Palette API

Let's get started! Scroll down to see your first step.

## Concepts and Setup

**An introduction to Android Wear and watch faces.**

- [The development environment](#)
- [Get the starter project](#)
- [Import the project](#)
- [Overview of key components](#)
- [Additional information - not required for the code lab](#)
- [Summary](#)

## Concepts

To start off let's learn a little bit about Android Wear and its most prominent UI element - the watch face.



Android Wear is a wearable platform designed for small, powerful devices, worn on the body. It is designed to deliver useful information when you need it most, intelligent answers to spoken questions, and tools to help reach fitness goals.

Being such a personal device, style is a big part of it. Aside from offering a choice of manufacturers, customisable watch faces give users even more ways to express their personal style. This is what we are going to create today.

A watch face is essentially a native service that runs in the background on an Android Wear device. Within this service, there is an engine that renders each screen. You can think of this as an animator flipping through a book of moving cartoon drawings. Our code will fill each of these pages, making your watch face move.

So let's get started!

## Setup

### The development environment

This codelab uses [Android Studio](#) 1.1, an integrated development environment (IDE) for developing Android apps. In addition, it will require Android Wear SDK components to be downloaded. Please refer to [this guide](#) for installing both the Android Studio and the required SDK components.

### Get the starter project

To get you started as quickly as possible, we have prepared a starter project for you to build on. It contains some basic code and application settings necessary for building watch faces.

If you have git installed, clone this repository:

```
git clone https://github.com/googlesamples/android-codelab-watchface.git
```

If you do not have git, you can download the code in zip format [here](#).

Either way, get the sample code and point Android Studio to it.

## Import the project

- Start Android Studio
- Select "Open an existing Android Studio project"
- Open the project directory
- Double click on the `build.gradle` file in the `android-codelab-watchface` directory
- Click OK on the "Import Project from Gradle" screen without making changes

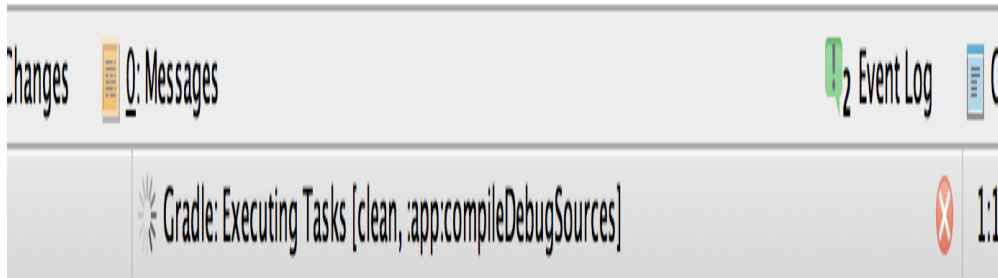


In the upper left hand corner of the project window, you should see something like this:



There are five folder icons. Each of them are known as a *module*. Please note that Android Studio might take several seconds to compile the project in the background for the first time. During this time

you will see a spinner in the bottom status bar:



The lower right corner

during execution

We recommend that you wait until this has finished before making code changes. This will allow Android Studio to pull in all the necessary components. In addition, if you get a prompt asking you to reload the project in order to make language changes effective, select “Yes”.

## Understand the starter project

All right, you’re set up and ready to start creating your own watch face. We’ll set off using the *1-base* module, which is the starting point for the watch face that we’ll be building upon. You will be adding code from each step to *1-base*.

Each of the following chapters can be used as reference points to check your work or for reference if you encounter any issues. The number in front of the module name corresponds with the codelab step.

### Overview of key components

#### **MyWatchFaceService.java**

Our watch face and it has a watch face engine called `MyWatchFaceService.Engine`. This file is located in the directory `1-base/java/com/android/example/watchface`. In Android Studio, this is located under `1-base/java/com.android.example.watchface`. Within the `Engine` class, we will mainly be working on three methods:

#### **onCreate**

We will initiate new classes such as the bitmap image object for our background, etc. This code is run once when the `Engine` is first started.

#### **onSurfaceChanged**

This is the first time when we have the dimension of the screen. Armed with this new information, we

can resize any screen element required for drawing. This code is also expected to only be run once at the start.

### **onDraw**

The core of what we will be doing. It renders every frame on the watch face canvas. Since it runs on every frame, we will try to keep this as fast as possible - no image resizing or object creation here!

### **Pro-tip**

To achieve a high frame rate which results in smooth animations no intense computations should happen in here. We can load images and resize them before we start drawing in `onCreate` and `onSurfaceChanged`.

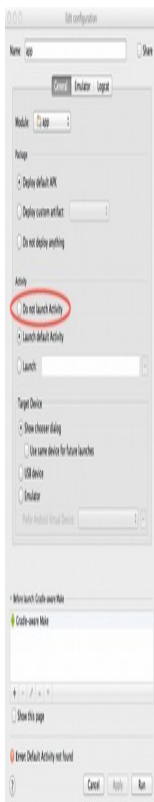
### **res/drawable-nodpi/**

This is the directory where we will be placing some additional image files.

## **Run the starter project for the first time**

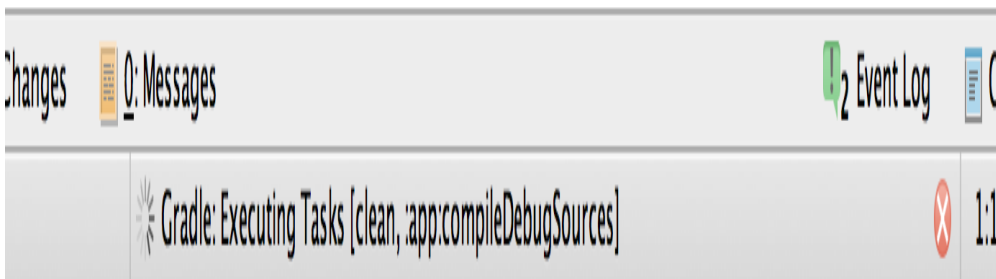
Let's run it on a watch / emulator. If you need help setting up an Android Wear emulator, please refer to the ["Set Up an Android Wear Emulator or Device"](#) section of the "Creating and Running a Wearable App" article.

- Connect your Android Wear device or start an emulator.
- In the toolbar select the *1-base* configuration from the drop down menu and click the green triangular *run button* next to it.
- If the following dialog box comes up, select "Do not launch Activity" and then click Run.  
Activity is a program that users launch, since watch faces run in the background automatically, there is no activity to run.



- Select your Android device or emulator and click OK. This will install the watch face on the Android Wear device or emulator.
- Wait a short time while the watch face is being built.

You will see a spinner in the status bar at the bottom of Android Studio while this is happening.



The lower right corner

during execution

If the focus is not already at the "4: Run" tab at the bottom left hand side of Android Studio, select that tab and you can see how the installation progresses. At the end of the installation process, you should see something similar to this:

```
Waiting for device.
Target device: sony-smartwatch_3-XXXXXXXXXXXXXX
Uploading file
    local path: ~/AndroidWearCodeLab/app/build/outputs/apk/app-debug.apk
    remote path: /data/local/tmp/com.android.example.watchface
Installing com.android.example.watchface
DEVICE SHELL COMMAND: pm install -r "/data/local/tmp/com.android.example.watchface"
pkg: /data/local/tmp/com.android.example.watchface
```

After the installation is complete

- long tap on the watch face of your device or emulator
- scroll across until you see a watch face labelled "Analog Codelab Watchface" and
- select it by tapping on it.

If you do not see this, try deploying the watch face again by pressing the green triangle button in the toolbar.

Here's what the watch face should look like. Don't worry if your emulator has a cloud with a strikethrough in place of the aeroplane icon. We will not need connection to a phone / internet for this code lab. Also note that the power button to the right might not appear - this is okay!



The watch face after a first successful deployment

### **Additional information - not required for the code lab**

Since Android Studio 1.1, we have added watch face templates which make it really simple to add watch faces to an existing application. To add watch faces to an existing project:

- Open your existing project

If it does not have an Android Wear module, add one by

- selecting "File > New Module"
- Then select an Android Wear module.

- Fill in the module name
- click "next"
- Select "Add No Activity"
- Click "Finish"

You now have an Android Wear module in your project. Now you can add a base template for a watch face by following these steps:

- Go to the Android Wear module,
- right click on the folder named "java".
- Select New > Wear > Watch Face

Go through the wizard to add either an analog or digital watch face. The resulting project will be very similar to *1-base* in this code lab.

## Summary

In this step you've learned about:

- Android Wear and the concepts behind watch faces
- The basics of our starting point - *1-base*
- How to connect to Android Wear device, deploy the watch face and run it

## Next up

**Let's start making this watch face our own by changing the background.**