



Xamarin

# Developing Your First Android Wear App

**James Montemagno**  
Developer Evangelist, Xamarin  
[@JamesMontemagno](https://twitter.com/JamesMontemagno)

# Who is this guy?

---



## James Montemagno

Developer Evangelist, Xamarin

@JamesMontemagno

<http://motzcod.es>

[github.com/JamesMontemagno](https://github.com/JamesMontemagno)

---



smart  
devices





Image: <http://blog.apphappening.com/>



GLASS









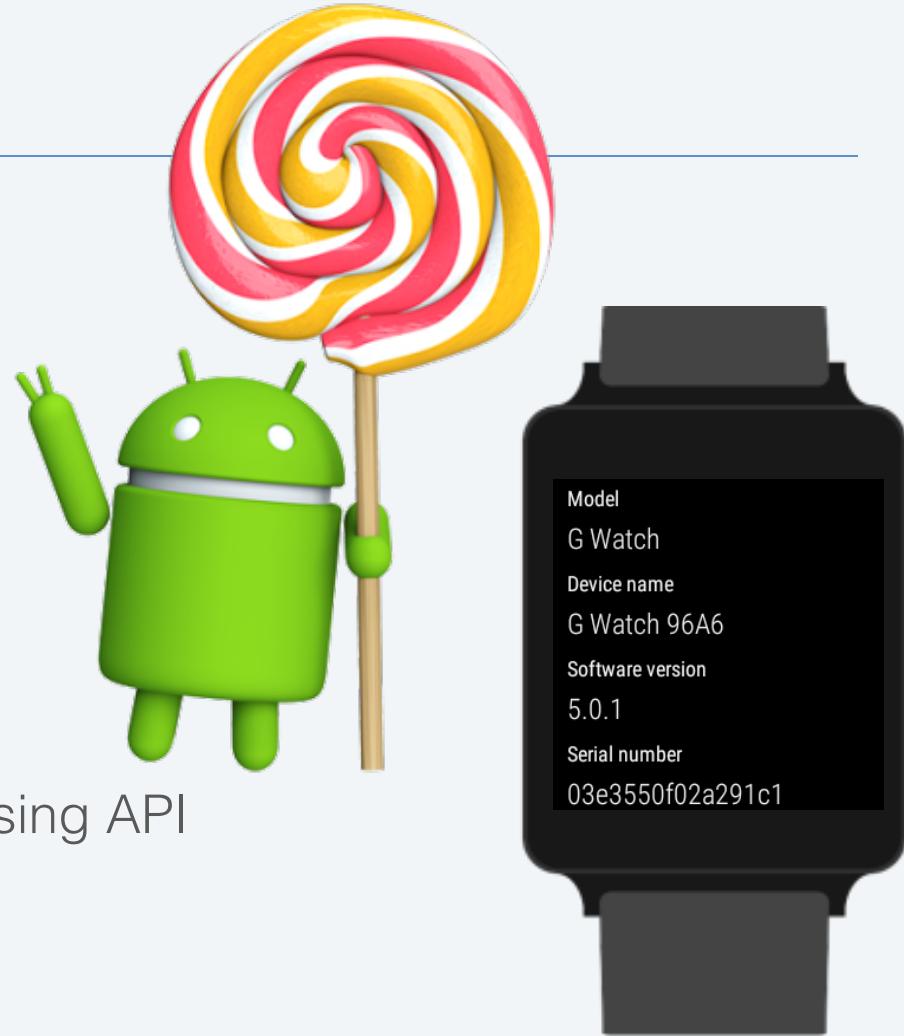
# What is Android Wear?



# Android Wear

---

- Android 5.0.1
  - No Access To:
    - `Android.Webkit`
    - `Android.Print`
    - `Android.App.Backup`
    - `Android.AppWidget`
    - `Android.Hardware.Usb`
  - Sensor Support
    - Call `HasSystemFeature()` before using API



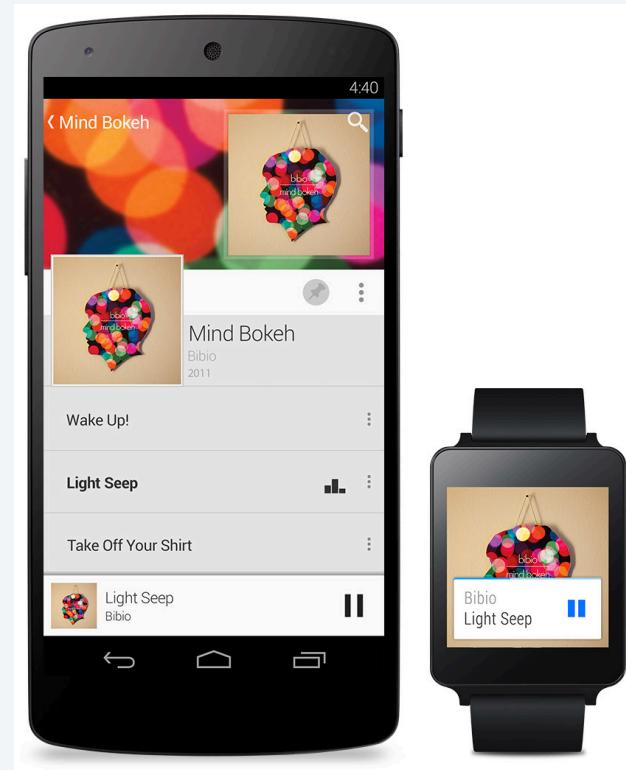
# Bluetooth 4.0 Low Energy

- Used for all communication



# Android Phone Required!

- Android 4.3+ (40% of devices in market)
- Android Wear App
- Bluetooth LE Required
- <http://g.co/WearCheck>



# Android Wear

Keeping you connected to what matters



**ASUS ZenWatch**  
Smart companion and  
wellness manager,  
powered by Android Wear.



**LG G Watch R**  
Classic watch design, only  
smarter.



**Sony  
SmartWatch 3**  
Useful information when  
you need it, powered by  
Android Wear.



**Moto 360**  
A modern timepiece  
powered by Android Wear.



**Samsung Gear Live**  
Bright and stylish, the  
Samsung Gear Live  
moves with you.



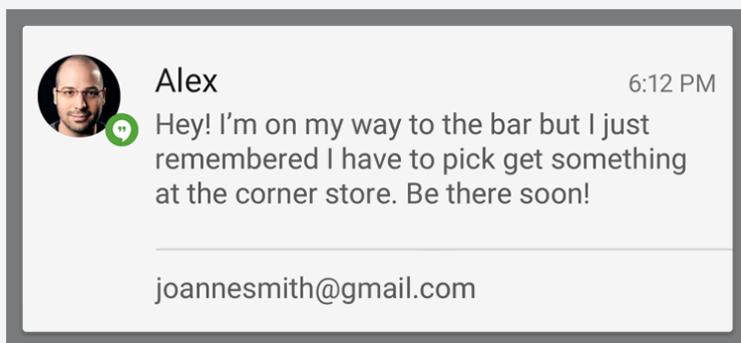
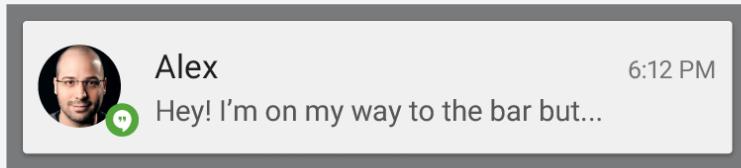
**LG G Watch**  
Always with you, always-  
on. Powered by Android  
Wear.

What can we do?

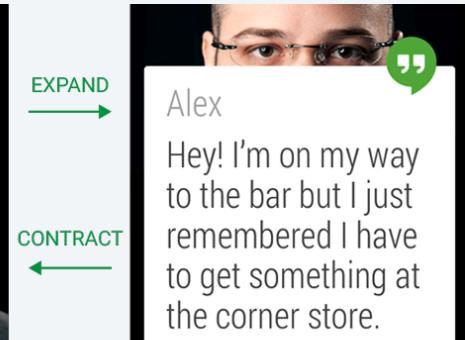
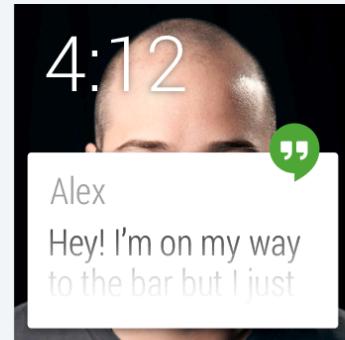
Notifications  
Companion Apps  
Watch Faces

Notifications  
Companion Apps  
Watch Faces

# Notifications

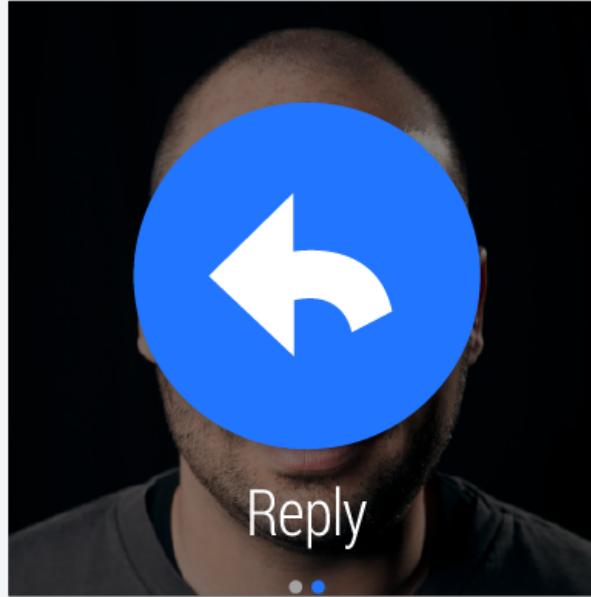
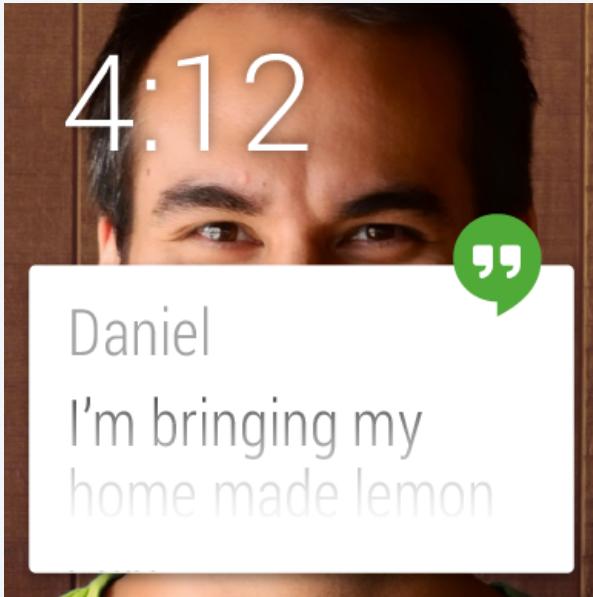


TEXT



# Notifications - Actions

---

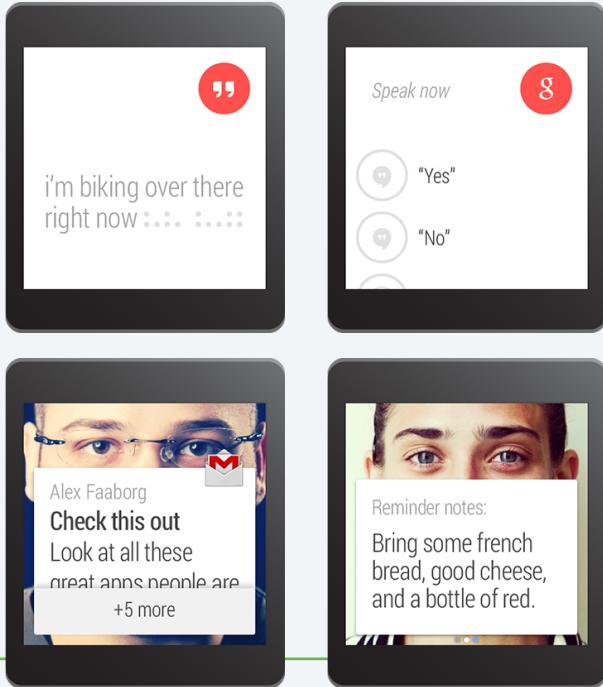


# Extending Existing Notifications

---

Lazy, but very effective!

- Uses Notification.WearableExtender
- Custom Backgrounds
- Actions specific for Wear
- Remote Input
  - Voice Input
  - Predefined actions
- Pages, Stacking, and More!

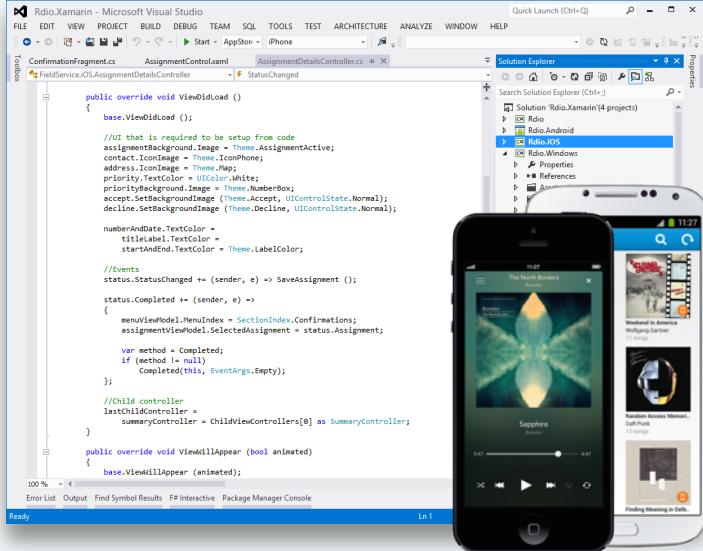


Demo

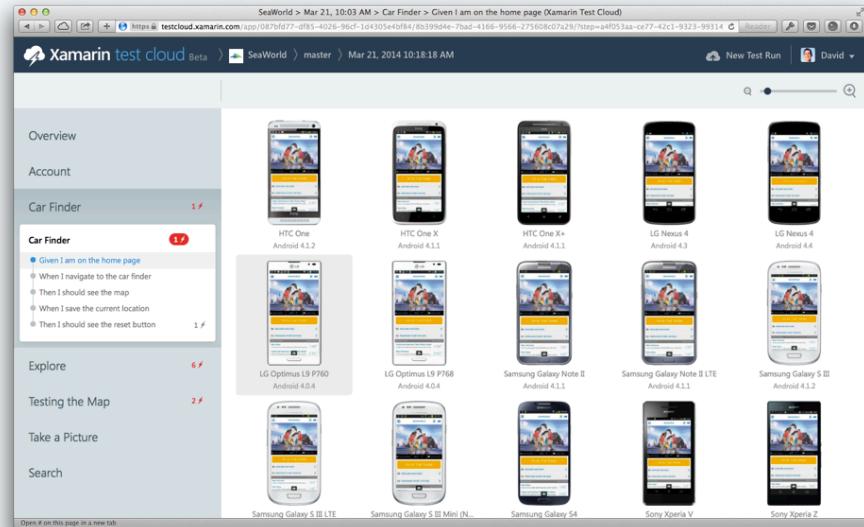
# What is Xamarin?



# Xamarin Platform



Create native iOS, Android, Mac and Windows apps in Visual Studio and C#



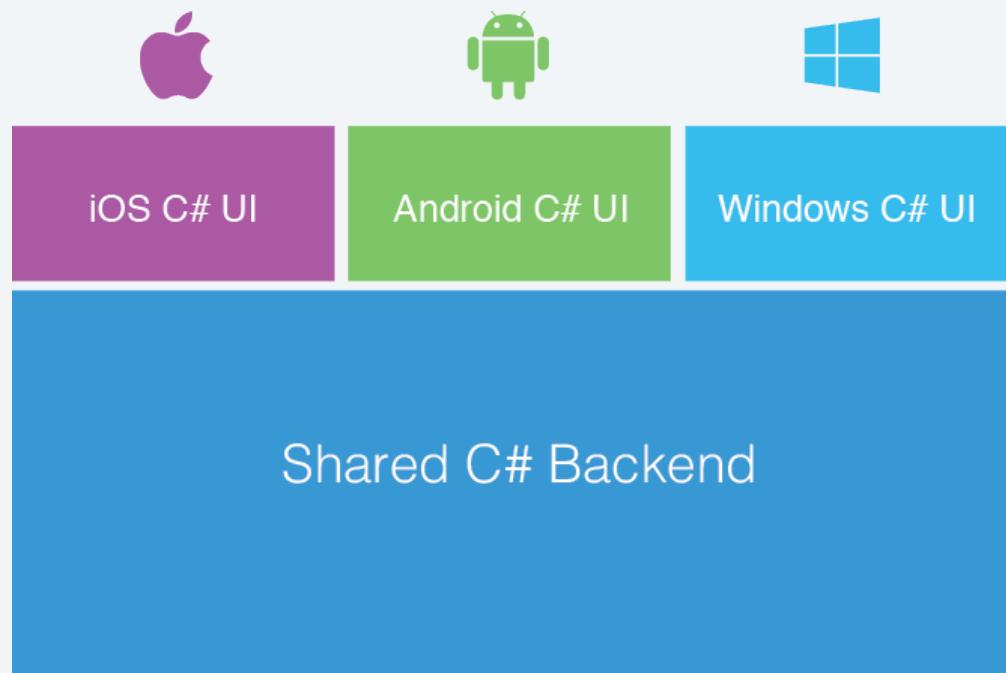
Automatically test your app on hundreds of mobile devices

# Xamarin's Unique Approach

---

Native With  
Code Sharing

---



```
var action = new NotificationCompat.Action.Builder(Resource.Drawable.ic_stat_rating_favorite,
    "Favorite", contentIntent)
    .Build();

var wearableExtender = new NotificationCompat.WearableExtender()
    .SetHintHideIcon(true)
    .SetBackground(BitmapFactory.DecodeResource(Resources, Resource.Drawable.ic_background))
    .AddAction(action);

var style = new NotificationCompat.BigTextStyle();
style.BigText(messageLong);

//Generate a notification with just short text and small icon
var builder = new NotificationCompat.Builder(this)
    .setContentIntent(contentIntent)
    .setSmallIcon(Resource.Drawable.ic_stat_hexagon_blue)
    .setContentTitle(title)
    .setContentText(message)
    ..setStyle(style)
    .setWhen(Java.Lang.JavaSystem.currentTimeMillis())
    .setAutoCancel(true)
    .addAction(Resource.Drawable.ic_stat_social_reply,
        "Reply", contentIntent)
    .extend(wearableExtender);
```

```
var action = new NotificationCompat.Action.Builder(Resource.Drawable.ic_stat_rating_favorite,
    "Favorite", contentIntent)
    .Build();

var wearableExtender = new NotificationCompat.WearableExtender()
    .SetHintHideIcon(true)
    .SetBackground(BitmapFactory.DecodeResource(Resources, Resource.Drawable.ic_background))
    .AddAction(action);

var style = new NotificationCompat.BigTextStyle();
style.BigText(messageLong);

//Generate a notification with just short text and small icon
var builder = new NotificationCompat.Builder(this)
    .setContentIntent(contentIntent)
    .setSmallIcon(Resource.Drawable.ic_stat_hexagon_blue)
    .setContentTitle(title)
    .setContentText(message)
    ..setStyle(style)
    .setWhen(Java.Lang.JavaSystem.currentTimeMillis())
    .setAutoCancel(true)
    .addAction(Resource.Drawable.ic_stat_social_reply,
        "Reply", contentIntent)
    .extend(wearableExtender);
```

```
var action = new NotificationCompat.Action.Builder(Resource.Drawable.ic_stat_rating_favorite,
    "Favorite", contentIntent)
    .Build();

var wearableExtender = new NotificationCompat.WearableExtender()
    .SetHintHideIcon(true)
    .SetBackground(BitmapFactory.DecodeResource(Resources, Resource.Drawable.ic_background))
    .AddAction(action);

var style = new NotificationCompat.BigTextStyle();
style.BigText(messageLong);

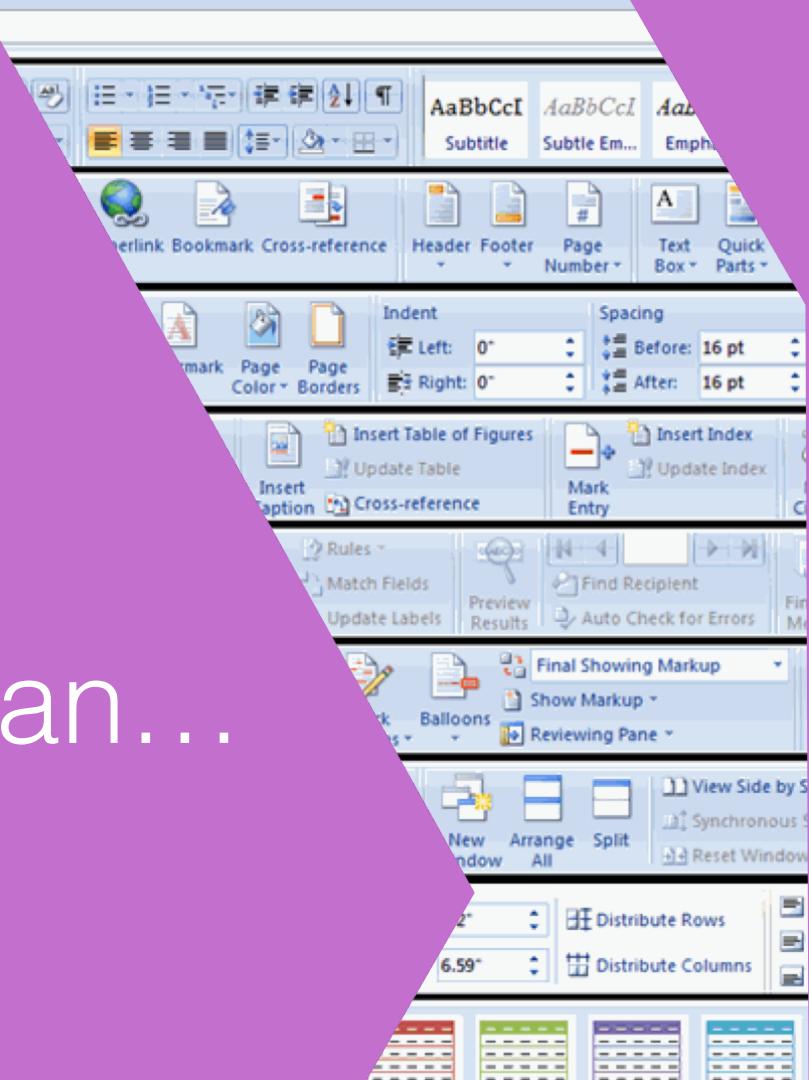
//Generate a notification with just short text and small icon
var builder = new NotificationCompat.Builder(this)
    .setContentIntent(contentIntent)
    .setSmallIcon(Resource.Drawable.ic_stat_hexagon_blue)
    .setContentTitle(title)
    .setContentText(message)
    ..setStyle(style)
    .setWhen(Java.Lang.JavaSystem.currentTimeMillis())
    .setAutoCancel(true)
    .addAction(Resource.Drawable.ic_stat_social_reply,
        "Reply", contentIntent)
    .extend(wearableExtender);
```

Demo

Notifications  
Companion Apps  
Watch Faces

# Wearable App Design

# Just because you can...

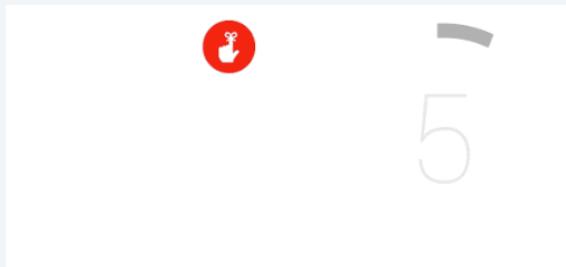


# Doesn't mean you should!

# Focus on not stopping the user

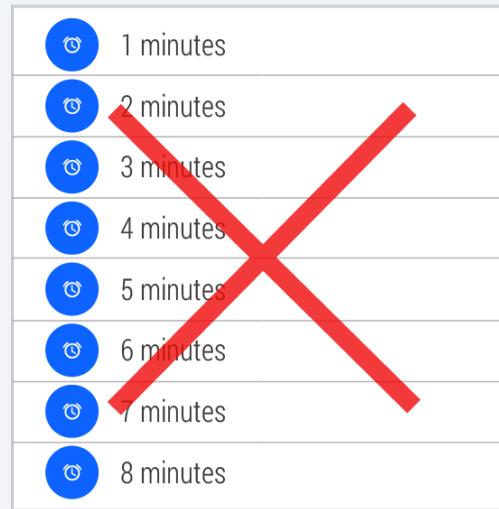
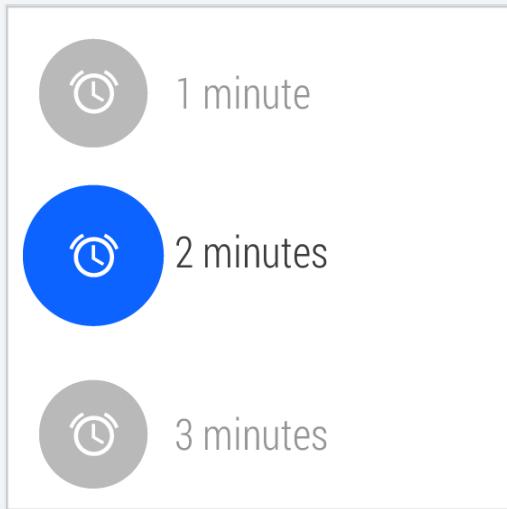
---

- 5 Second Rule



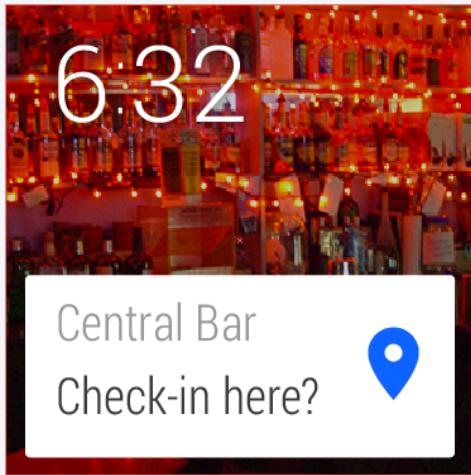
# Design for big gestures

---

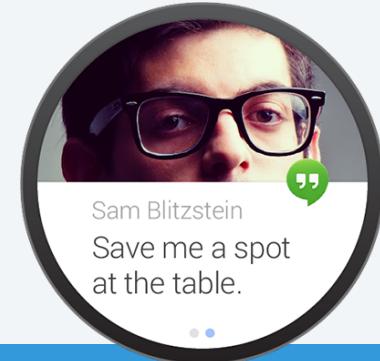
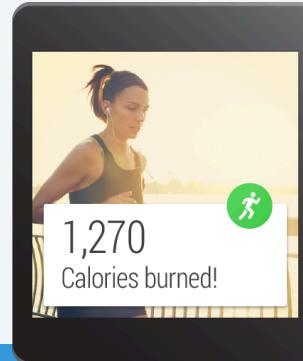


# Other tips:

---



- Think stream cards first
- Do one thing, really fast
- Design for corner of the eye
- Don't be a constant shoulder tapper (aka don't vibrate the watch)
- Be aware of Square and Round







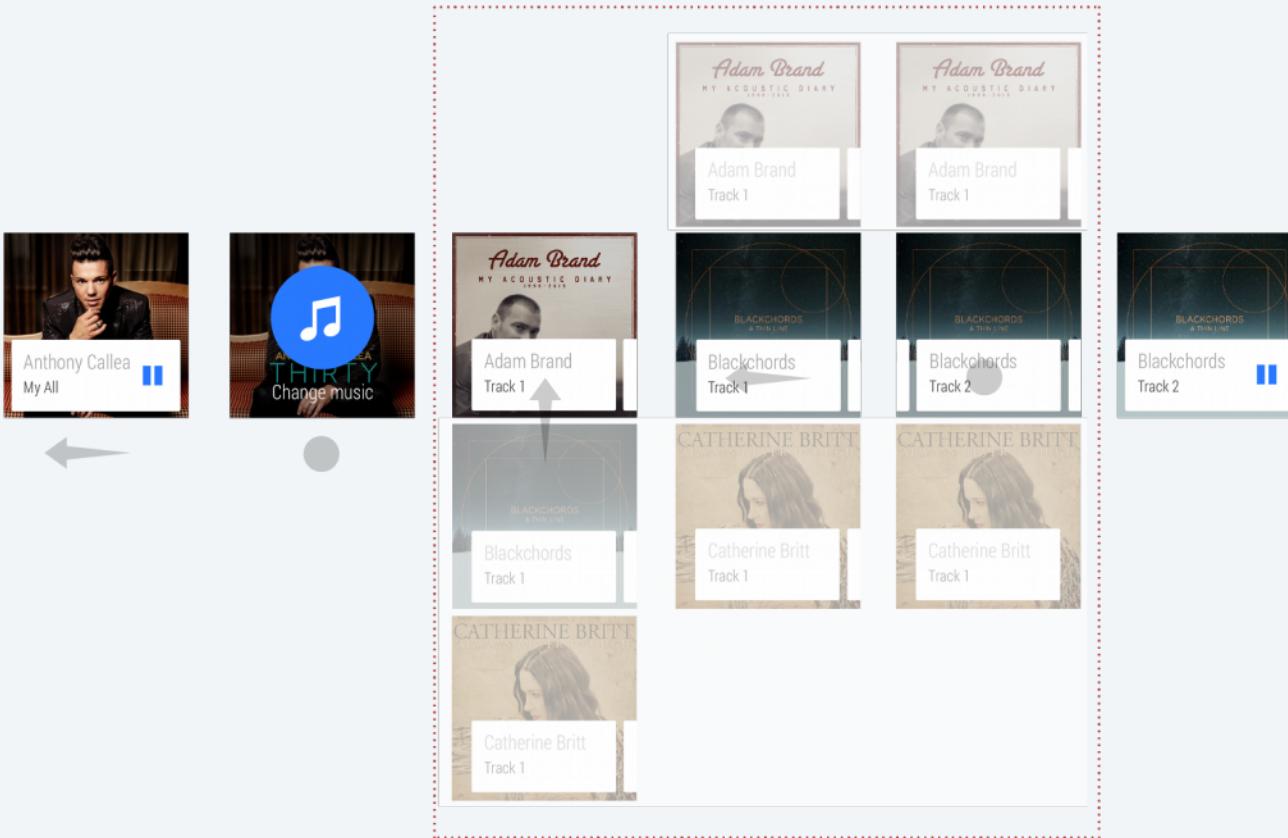
# Android Wear – Building Apps

---

- Companion App
  - Android app that runs on a wearable
- New Themes
- New UI Widgets
  - BoxInsetLayout
  - CircledImageView
  - WearableListView
  - More...



# GridViewPager

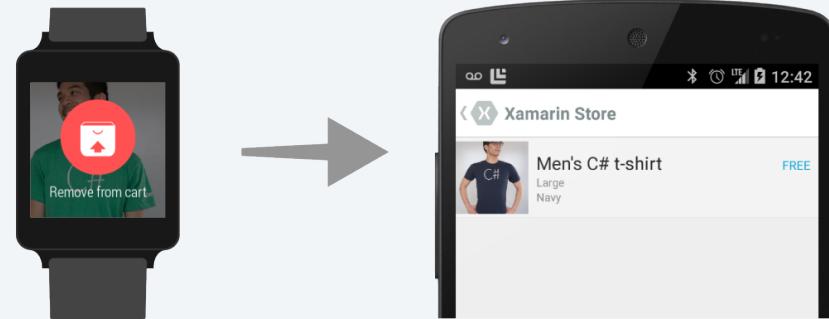


Demo

# Android Wear – Communication

---

- Data API
  - Sync datastore between phone and wearable
- Message API
  - Send small payloads and simple communication



# Packaging

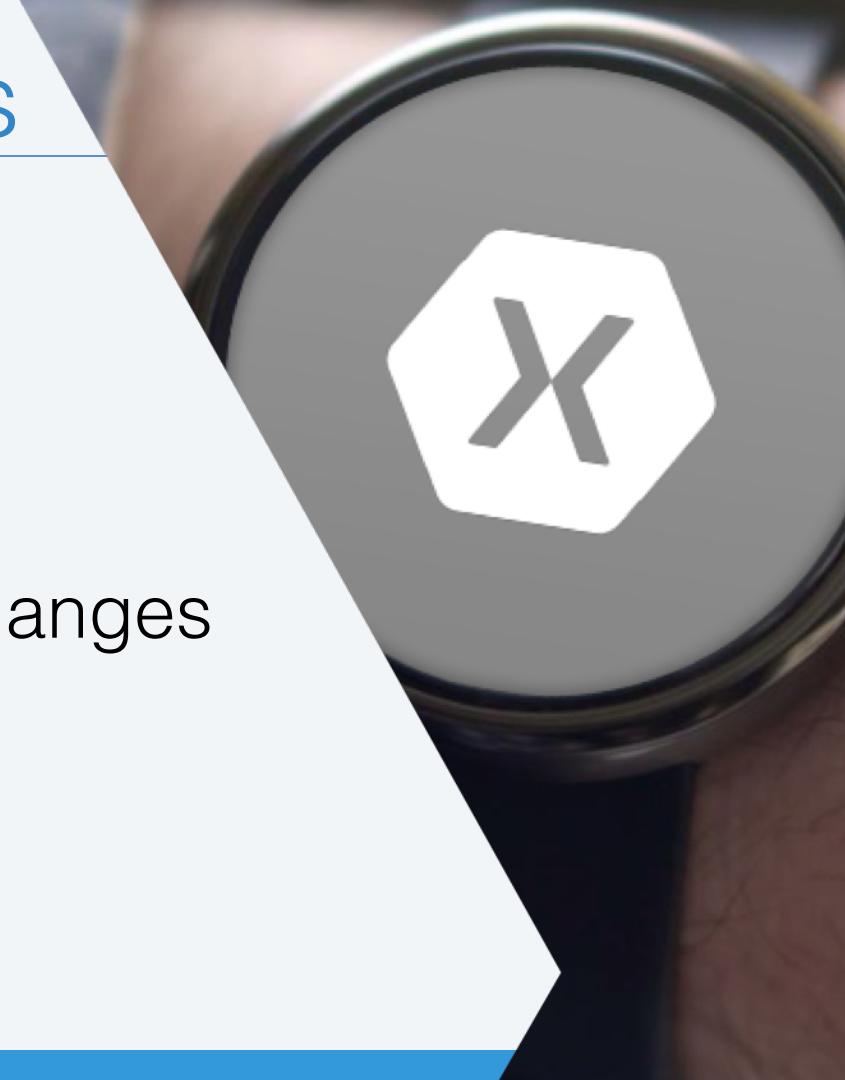
---

- Wear apps do not exist on their own
- Embedded inside ‘Companion’ apps
- Signed with same keystore
- Wear app’s .APK file embedded as Asset
- Resource XML points to .APK

Notifications  
Companion Apps  
**Watch Faces**

# Custom Watch Faces

- Wallpaper Service
- OnDraw (Canvas)
- Still an embedded app
- Ambient Mode / Visibility changes



# Resources

---

- Tips for Your First Wear App
  - <http://blog.xamarin.com/tips-for-your-first-android-wear-app/>
- Google Documentation
  - <http://developer.android.com/training/building-wearables.html>
- Samples
  - <http://developer.xamarin.com/samples/android/Android%20Wear/>

# Create native iOS, Android, Mac and Windows apps in C#.

Join our community of 867,548 developers.

[Download now](#)



Get Started Today: [xamarin.com/download](http://xamarin.com/download)



Find Me:

@JamesMontemagno

@XamarinHQ

<http://motzcod.es>

<http://blog.xamarin.com>

---

# Q & A