Android Auto

Drive Your Car, Use Your Phone, and Don’t Hurt Anyone

Phil Shadlyn
@physphil
#DroidconBos

What is Android Auto?

- Safely use your device while driving
- Limited functionality
- Voice control

#DroidconBos
@physphil

What ISN'T Android Auto?

- Not a standalone version of Android
- No Google Play Store
- Can only use Auto-compatible apps

34 mins to LAX
Normal traffic on I-110 S
Los Angeles
Clear
79°
83°
63°
11:14
android auto
RADIO
MEDIA
PHONE
BLUELINK
SEEK TRACK
MAP VOICE
NAVI
INFO
SETUP
AUTO
73°F
SYNC
73°F

STOP
Hey what's up

A Brief History

- Announced at Google I/O 2014
- 2015 Hyundai Sonata was first to implement
- Adoption increased through 2015 and 2016
- Android Auto v2.0 released in November 2016

Limited Functionality

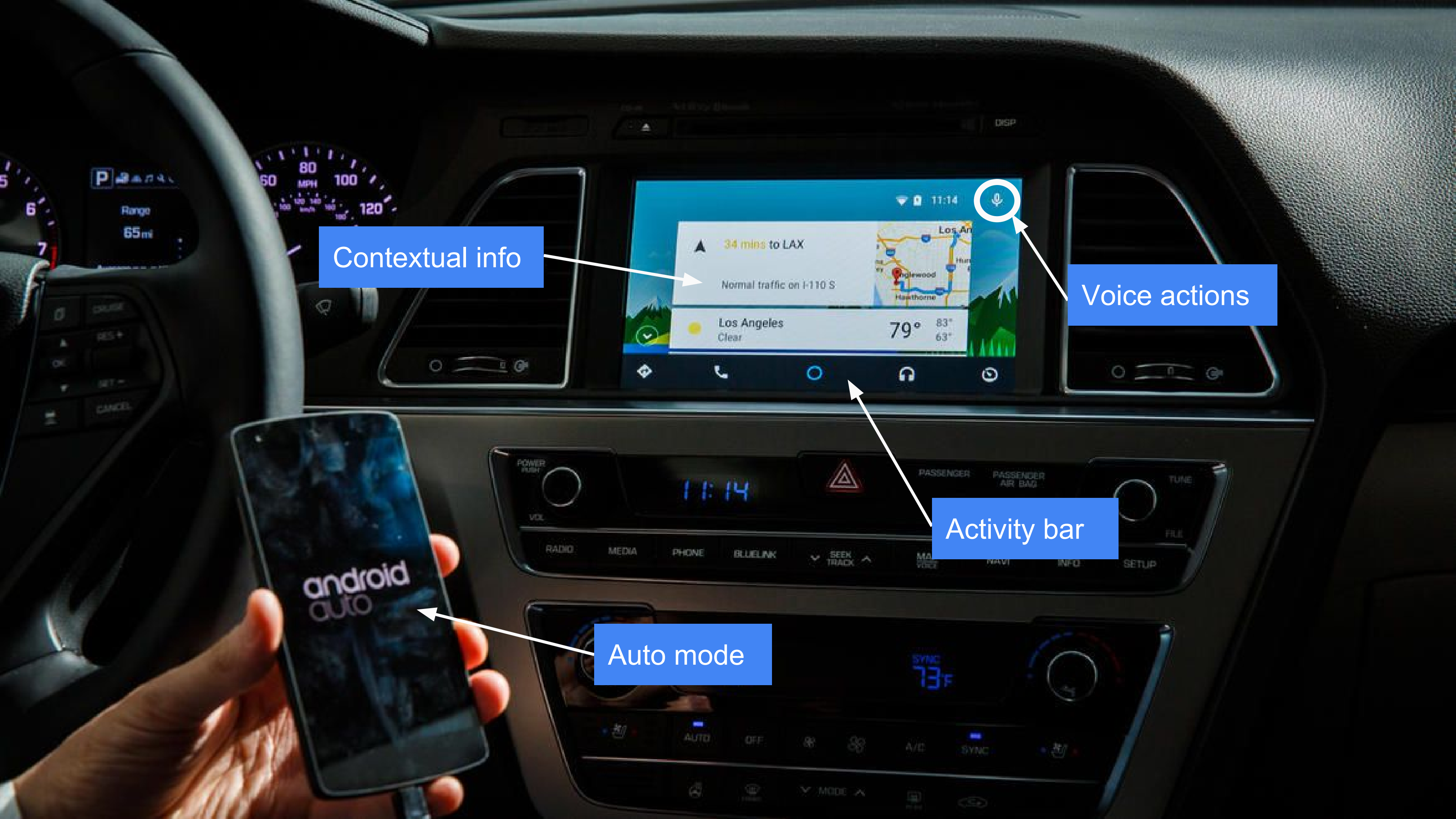
- Navigation
- Telephony
- Messaging
- Audio Control
- Web Search

Requirements

- Android 5.0+ device
- Download Android Auto app
- Car with Android Auto head unit (optional)

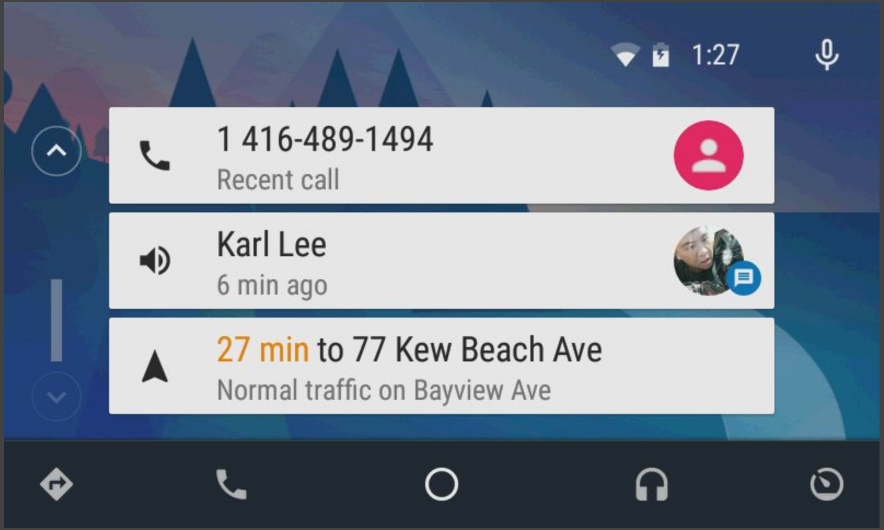
Basic Operation

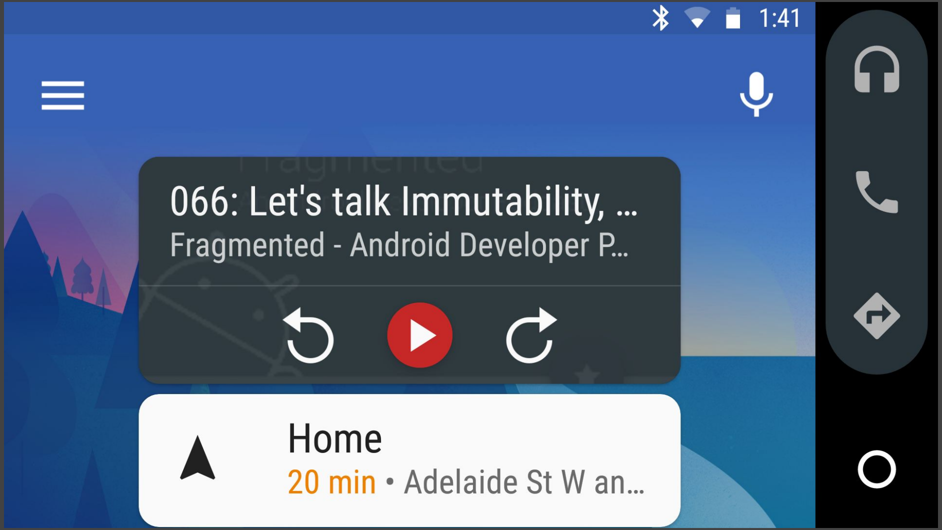
- Connect phone to car / open Android Auto app
- Phone goes into “Auto mode”
- Phone or Head Unit displays series of contextual cards
- Microphone button triggers voice actions
- Activity bar to start specific apps


Contextual info
Voice actions
Activity bar
Auto mode
11:14
34 mins to LAX
Normal traffic on I-110 S
Los Angeles
Clear
79°
83°
63°
android auto

Let’s take a tour...

Home Screen



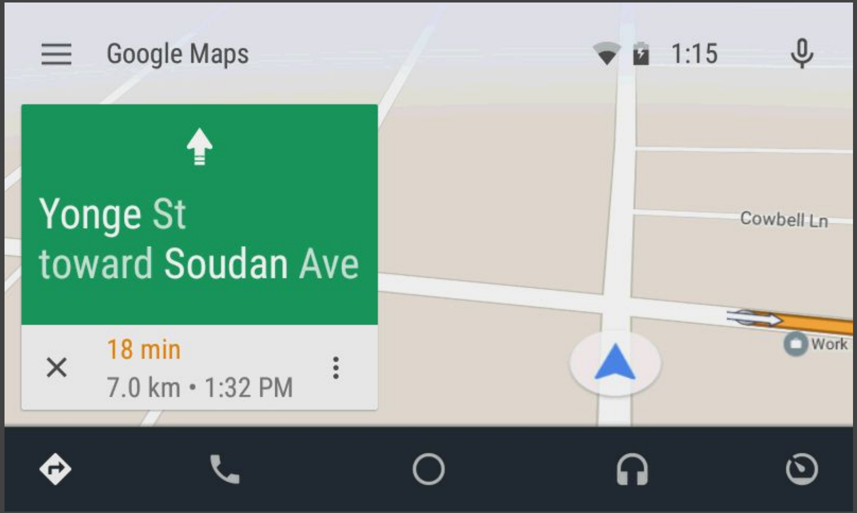

1:41
066: Let's talk Immutability, ...
Fragmented - Android Developer P...
Home
20 min • Adelaide St W an...

Left: Head Unit

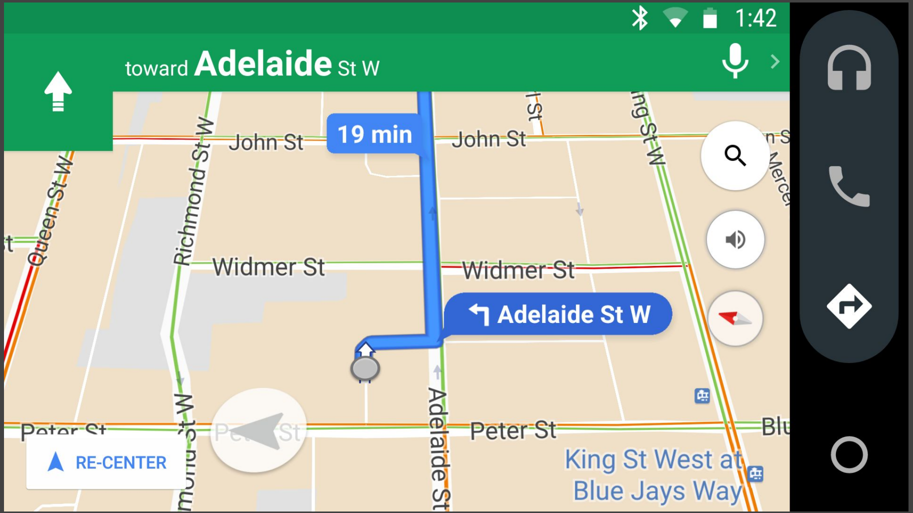
Right: Phone

#DroidconBos
@physphil

Navigation



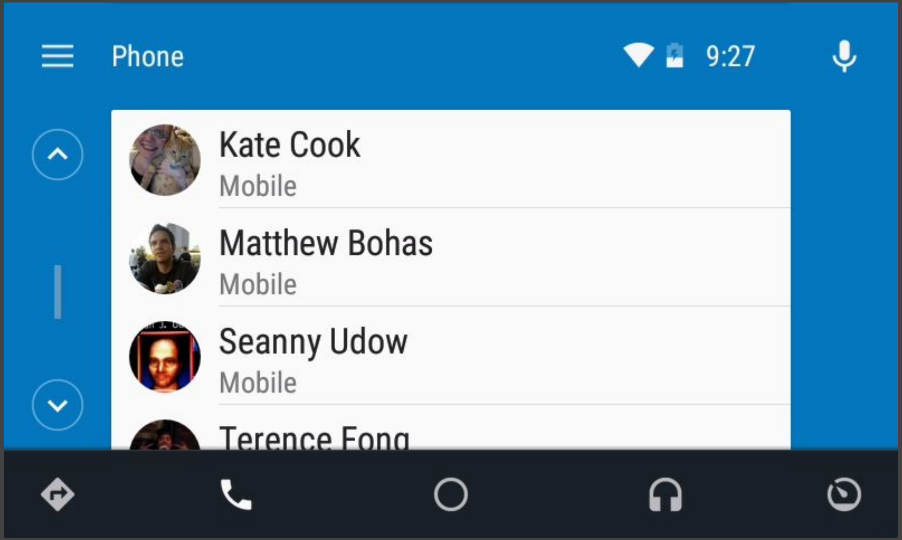
Left: Head Unit

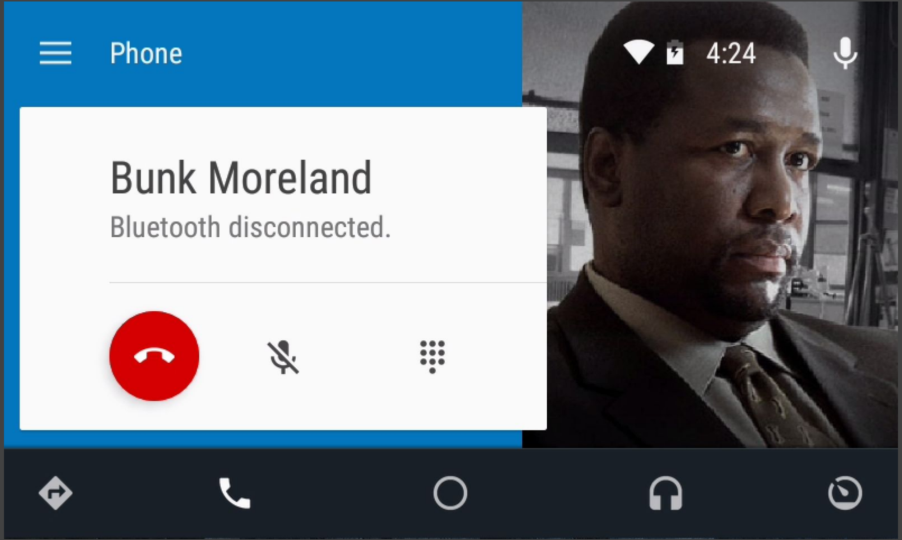


Right: Phone

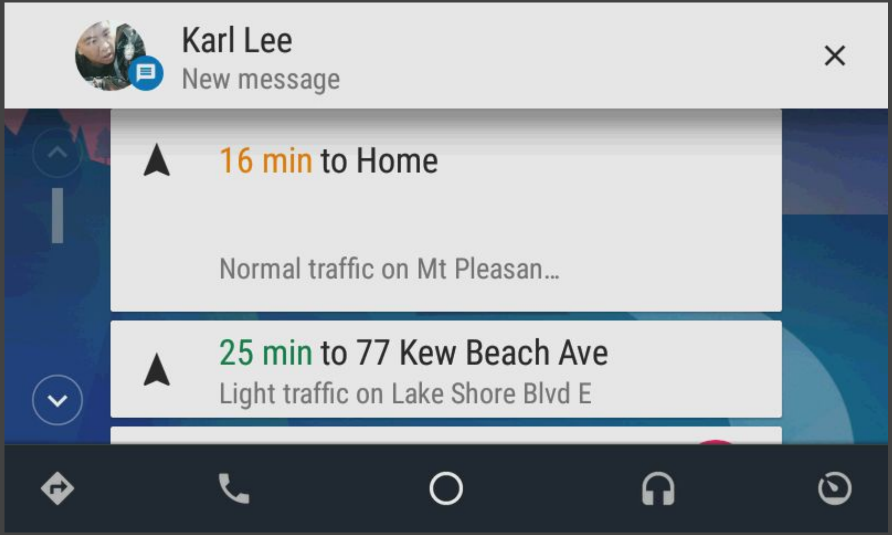
#DroidconBos
@physphil

Telephony





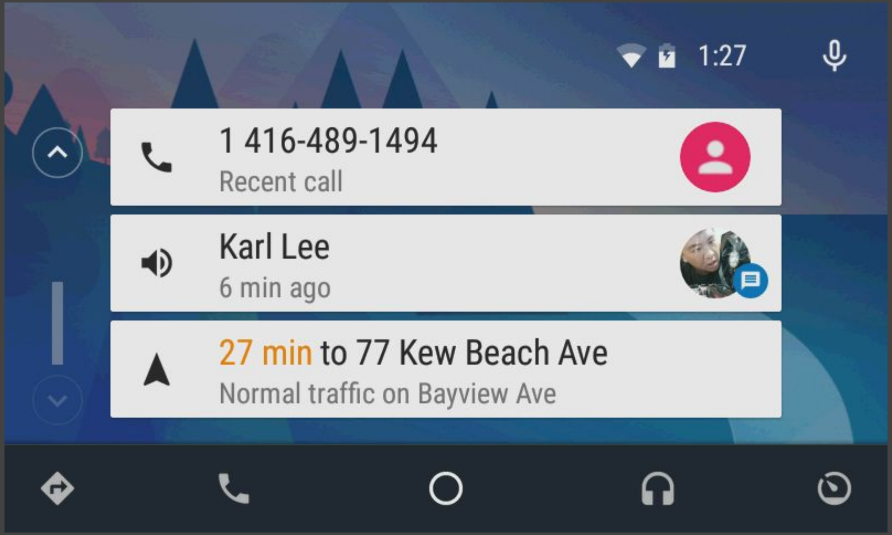
Messaging



Karl Lee
New message

16 min to Home
Normal traffic on Mt Pleasan...

25 min to 77 Kew Beach Ave
Light traffic on Lake Shore Blvd E



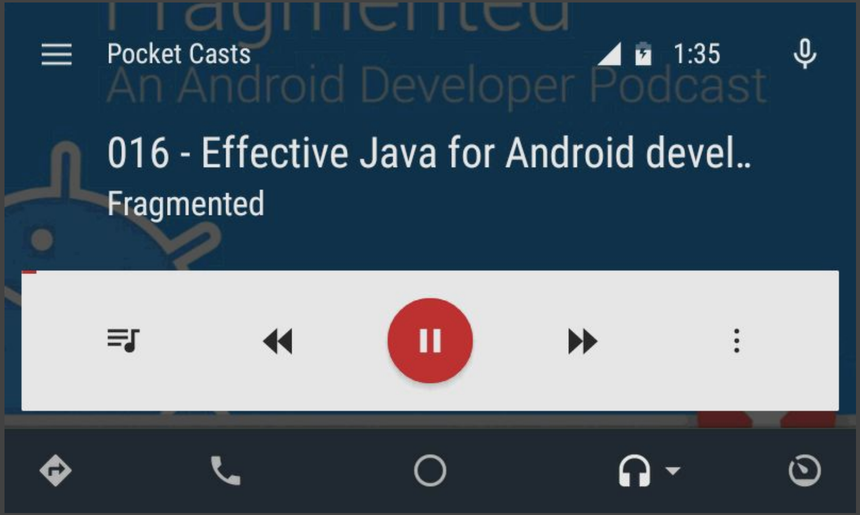
1 416-489-1494
Recent call

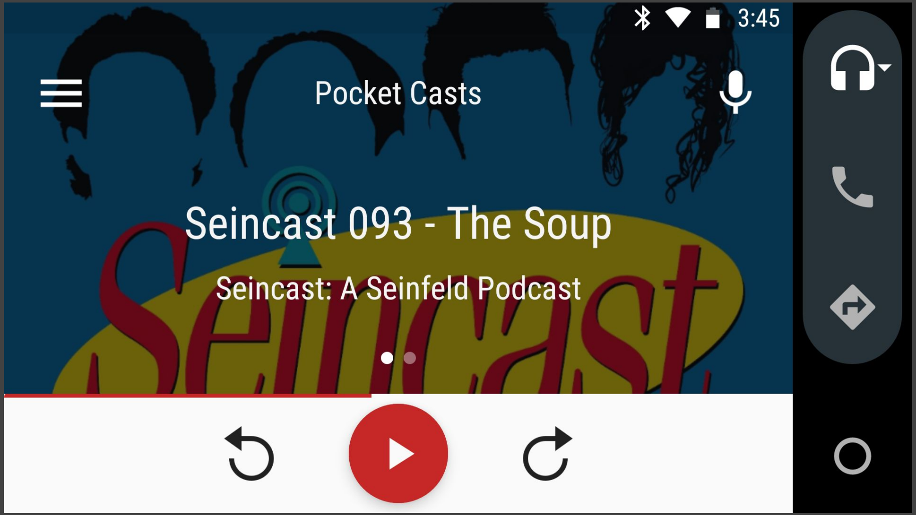
Karl Lee
6 min ago

27 min to 77 Kew Beach Ave
Normal traffic on Bayview Ave

#DroidconBos
@physphil

Audio Control


Fragmented
Pocket Casts
1:35
An Android Developer Podcast
016 - Effective Java for Android devel..
Fragmented



Left: Head Unit

Right: Phone

Developing Apps for Auto

Developing Apps for Auto

- Extend existing apps - think Chromecast, Android Wear
- Currently can only extend Audio or Messaging apps
- targetSdkVersion 21+
- No UI code!

Developing Apps for Auto

Create `automotive_app_desc.xml` config file

```
<automotiveApp>

   <uses name="media"/>
   <uses name="notification"/>

</automotiveApp>
```

Add to `<application>` in manifest

```
<meta-data
   android:name="com.google.android.gms.car.application"
   android:resource="@xml/automotive_app_desc" />
```

Extending a Messaging App

https://developer.android.com/training/auto/messaging/index.html

Extending a Messaging App

- Extend existing notification objects with `CarExtender`
- Provide `Intents` to be triggered when messages are heard and replied to
- Above `Intents` trigger `BroadcastReceivers`, which can update app
- `RemoteInput` object captures reply spoken by driver

Handle “Message Heard” Action

```
// Create Intent to be triggered when user hears message
final Intent messageHeardIntent = new Intent();
messageHeardIntent.setAction("com.physphil.android.ACTION_MESSAGE_HEARD");
messageHeardIntent.addFlags(Intent.FLAG_INCLUDE_STOPPED_PACKAGES);
messageHeardIntent.putExtra("message_id", message.getId());


// Create PendingIntent to trigger above intent
final PendingIntent messageHeardPendingIntent =
PendingIntent.getBroadcast(getApplicationContext(),
        message.getId(),
        messageHeardIntent,
        PendingIntent.FLAG_UPDATE_CURRENT);
```

Handle “Message Reply” Action

```
// Create Intent to be triggered when user replies message
final Intent messageReplyIntent = new Intent();
messageReplyIntent.setAction("com.physphil.android.ACTION_MESSAGE_REPLY");
messageReplyIntent.addFlags(Intent.FLAG_INCLUDE_STOPPED_PACKAGES);
messageReplyIntent.putExtra("message_id", message.getId());


// Create PendingIntent to trigger above intent
final PendingIntent messageReplyPendingIntent =
PendingIntent.getBroadcast(getApplicationContext(),
        message.getId(),
        messageHeardIntent,
        PendingIntent.FLAG_UPDATE_CURRENT);


// Create RemoteInput to capture spoken reply
RemoteInput remoteInput = new RemoteInput.Builder("key_voice_reply").build();
```

Add Unread Messages to UnreadConversation

```
// Create UnreadConversation object for all unread messages to display
UnreadConversation.Builder unreadConversation =
       new UnreadConversation.Builder(conversation.getSenderName())
               .setReadPendingIntent(messageHeardPendingIntent)
               .setReplyAction(messageReplyPendingIntent, remoteInput);


// Add each unread message
for (Message message : conversation.getUnreadMessages()) {
   unreadConversation.addMessage(message.getText())
           .setLatestTimestamp(message.getTimestamp());
}
```

Great, now what?

Extend Existing Notification

#DroidconBos
@physphil

```
// Generate Notification
NotificationCompat.Builder builder = new NotificationCompat.Builder(this)
        .setSmallIcon(R.drawable.ic_stat_notification)
        .setContentIntent(pendingIntent)
        .setAutoCancel(true)
        .setLargeIcon(BitmapFactory.decodeResource(getResources(), R.drawable.icon))
        .setContentTitle(getString(R.string.new_message))
        .setContentText(message.getText());



NotificationManager notificationManager = getSystemService(NOTIFICATION_SERVICE);
notificationManager.notify(NOTIFICATION_ID, builder.build());
```

Extend Existing Notification

#DroidconBos
@physphil

```java
// Generate Notification
NotificationCompat.Builder builder = new NotificationCompat.Builder(this)
        .setSmallIcon(R.drawable.ic_stat_notification)
        .setContentIntent(pendingIntent)
        .setAutoCancel(true)
        .setLargeIcon(BitmapFactory.decodeResource(getResources(), R.drawable.icon))
        .setContentTitle(getString(R.string.new_message))
        .setContentText(message.getText())
        .extend(new NotificationCompat.CarExtender()
                .setUnreadConversation(unreadConversation.build()));

NotificationManagerCompat notificationManager = NotificationManagerCompat.from(this);
notificationManager.notify(NOTIFICATION_ID, builder.build());
```

Define BroadcastReceivers in AndroidManifest.xml

```xml
<receiver
   android:name=".MessageHeardReceiver"
   android:enabled="true"
   android:exported="true">
   <intent-filter>
       <action android:name="com.physphil.android.ACTION_MESSAGE_HEARD"/>
   </intent-filter>
</receiver>


<receiver
   android:name=".MessageReplyReceiver"
   android:enabled="true"
   android:exported="true">
   <intent-filter>
       <action android:name="com.physphil.android.ACTION_MESSAGE_REPLY" />
   </intent-filter>
</receiver>
```

Extract Reply and Send Message

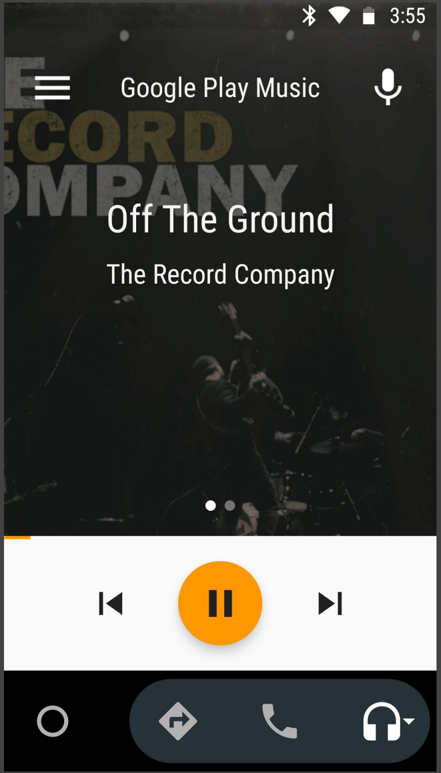
```java
public class MessageReplyReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {

        // Get message id
        final int id = intent.getIntExtra("message_id", -1);

        // Get voice reply and send reply through our messaging app
        final Bundle ri = RemoteInput.getResultsFromIntent(intent);
        if (ri != null) {
            final CharSequence reply = ri.getCharSequence("key_voice_reply");
            MessageService.getInstance().sendReply(id, reply);
        }
    }
}
```

Extending an Audio App

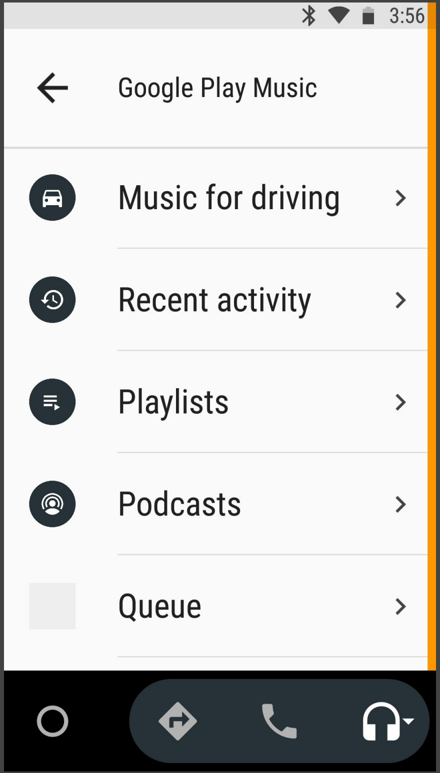
https://developer.android.com/training/auto/audio/index.html

Extending an Audio App

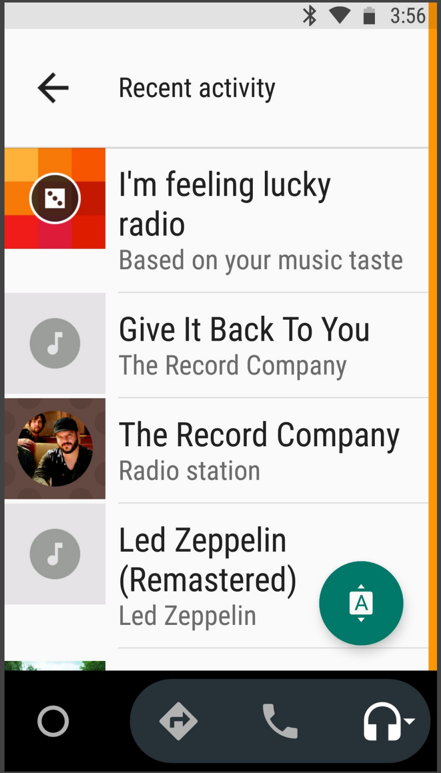
- Extend `MediaBrowserService` to provide content hierarchy
- `MediaSession.Callback` implements playback controls
- Register for voice actions with `IntentFilter` in manifest
- Examples and best practices found in Universal Music Player



Android Auto triggers `MediaSession` callbacks



`onGetRoot()` returns root node and session token



`onLoadChildren()` called to provide media hierarchy

Ok… so how do I do this?

Implement MediaSession.Callback

```
public void onPlay()
public void onPlayFromMediaId(String mediaId, Bundle extras)
public void onPlayFromSearch(String query, Bundle extras)
public void onPlayFromUri(Uri uri, Bundle extras)
public void onPause()
public void onStop()
public void onSkipToNext()
public void onSkipToPrevious()
public void onSkipToQueueItem(long id)
public void onRewind()
public void onFastForward()
public void onSeekTo(long pos)
```

Register for Voice Actions

Attach to launcher activity

```
<!-- Use this intent filter to get voice searches, like "Play The Beatles" -->
<intent-filter>
   <action android:name="android.media.action.MEDIA_PLAY_FROM_SEARCH" />
   <category android:name="android.intent.category.DEFAULT" />
</intent-filter>
```

And what about this MediaBrowserService?

Extend MediaBrowserService

Define Service in AndroidManifest.xml

```
<service
   android:name=".MusicService"
   android:exported="true">
   <intent-filter>
       <action android:name="android.media.browse.MediaBrowserService" />
   </intent-filter>
</service>
```

Create MediaSession in onCreate()

```
// Start a new MediaSession in onCreate()
mSession = new MediaSessionCompat(this, "session_tag");
mSession.setCallback(mMediaSessionCallback);
mSession.setFlags(MediaSessionCompat.FLAG_HANDLES_MEDIA_BUTTONS |
       MediaSessionCompat.FLAG_HANDLES_TRANSPORT_CONTROLS);
```

Implement onGetRoot()

```
@Override
public BrowserRoot onGetRoot(@NonNull String clientPackageName, int clientUid,
     Bundle rootHints) {

   // Verify the app attempting to access media contents
   if (!mPackageValidator.isCallerAllowed(this, clientPackageName, clientUid)) {

       // If the request comes from an untrusted package, return empty root.
       return new MediaBrowserServiceCompat.BrowserRoot(MEDIA_ID_EMPTY_ROOT, null);
   }

   return new BrowserRoot(MEDIA_ID_ROOT, null);
}
```

Implement onLoadChildren

```
@Override
public void onLoadChildren(@NonNull final String parentMediaId,
                           @NonNull final Result<List<MediaItem>> result) {

    if (parentMediaId.equals(MEDIA_ID_EMPTY_ROOT)) {
        result.sendResult(new ArrayList<MediaItem>());
    }
    else if (mMusicProvider.isInitialized()) {
        // if music library is ready, return immediately
        result.sendResult(mMusicProvider.getChildren(parentMediaId, getResources()));
    }
    else {
        // otherwise, only return results when the music library is retrieved
        result.detach();
        mMusicProvider.retrieveMediaAsync(new MusicProvider.Callback() {
            @Override
            public void onMusicCatalogReady(boolean success) {
                result.sendResult(mMusicProvider.getChildren(parentMediaId, getResources()));
            }
        });
    }
}
```

Testing and Distribution

#DroidconBos
@physphil

Testing

- Enable Developer Options in Auto app to test on phone
- Use Desktop Head Unit to simulate in-car dashboard
- Use alpha/beta Play Store channels to test in actual car

Auto App Quality

- Review Auto App Quality checklist before publication
    - Must support voice actions
    - Only pertinent notifications
    - Only short-form messaging (ie no email clients)
    - Touch actions must be complete within 6 steps
    - No visual ads
    - No animations, images or scrolling text
    - No games

Google Play Distribution

- All apps subject to driver safety review, even updates
- On failure, notified via email with a list of deficiencies
- App will not be published until successful review

Want to Learn More?

- Android Auto Developer Site
- Android Auto DevBytes
- Android Auto Udacity Course
- Distribute to Android Auto
- Universal Music Player

Questions?

Phil Shadlyn
@physphil
#DroidconBos