

DANIEL GRAHAM PHD

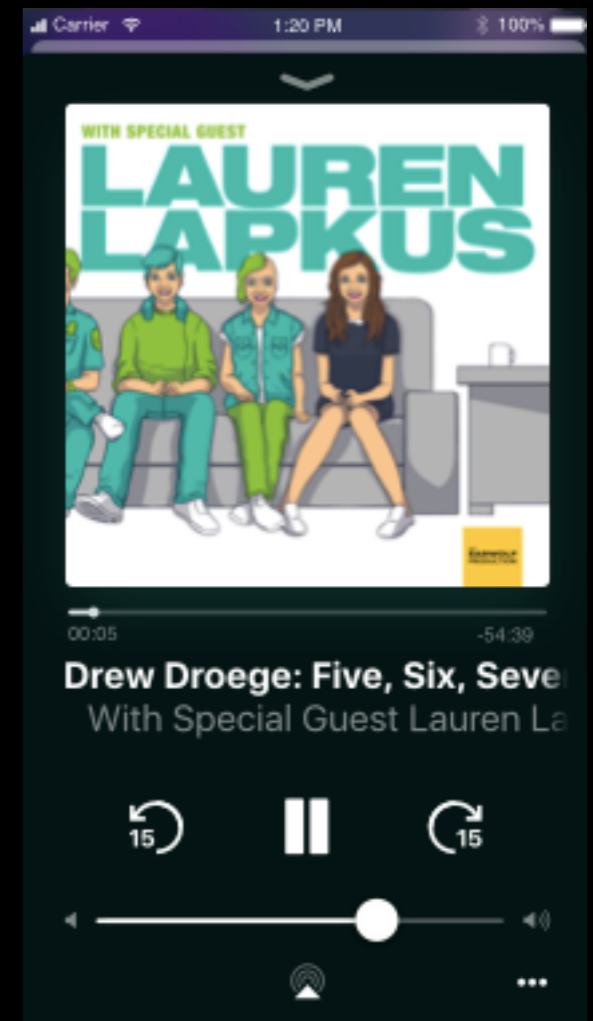
EXPO COMPONENTS

EXPO SDK REFERENCE

- Supports a collection of components to help you build your application.
- <https://docs.expo.io/versions/v32.0.0/sdk/>

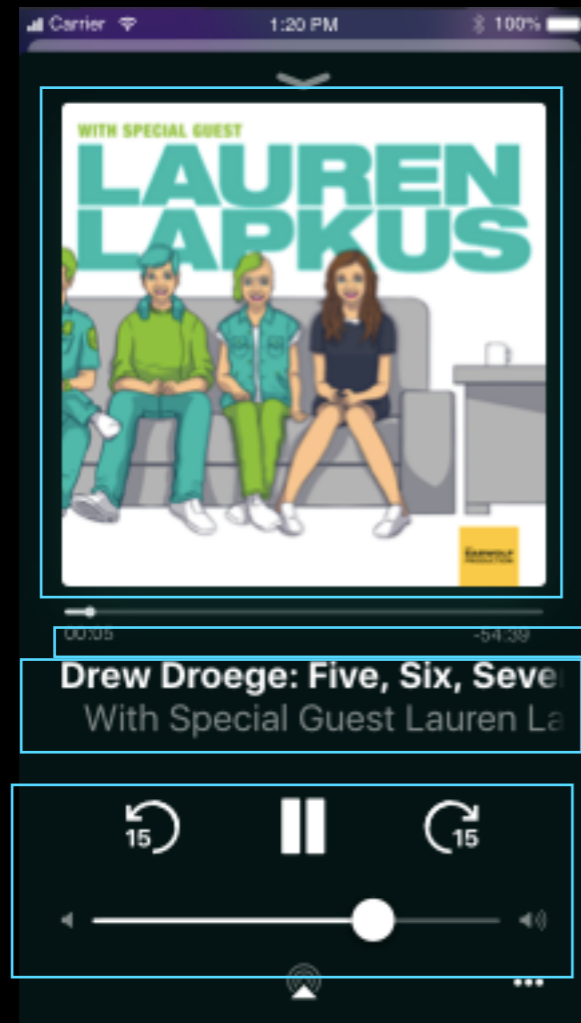
AUDIO COMPONENTS

- Let's build the the podcast play screen



- <https://docs.expo.io/versions/v32.0.0/sdk/audio/>

JSX COMPONENT BREAK DOWN



Image

Playback Progress

Title

Controls

LET'S START BY
BUILDING THE CONTROLS
SECTION

LET'S BEGIN BY JUST PLAYING A SOUND

WILL TAKE A POD ID

```
import React from 'react'
import {View, TouchableOpacity, Image, StyleSheet} from 'react-native'
import {Audio} from 'expo'

export default class PlayerScreen extends React.Component{
  constructor(props){
    super(props)
    var {navigation} = this.props
    this.state = {
      podID: navigation.getParam('pod', '0'),
      isPlaying: false
    }
  }
}
```

LOAD THE SOUND USING THE SOUND COMPONENT

```
componentDidMount(){  
  const soundObject = new Audio.Sound()  
  soundObject.loadAsync(require("../assets/tunes3.mp3")).then(  
    this.setState({  
      soundLoaded: true,  
      soundObject: soundObject  
    })  
  )  
}
```

TOGGLE SOUND

```
async toggleSound(){
  if(this.state.soundLoaded && !this.state.isPlaying){
    try {
      await this.state.soundObject.playAsync();
      this.setState({
        isPlaying: true
      })
    } catch (error) {
      console.log("error with sound")
    }
  }else{
    try {
      await this.state.soundObject.pauseAsync();
      this.setState({
        isPlaying: false
      })
    } catch (error) {
      console.log("error with sound")
    }
  }
}
```

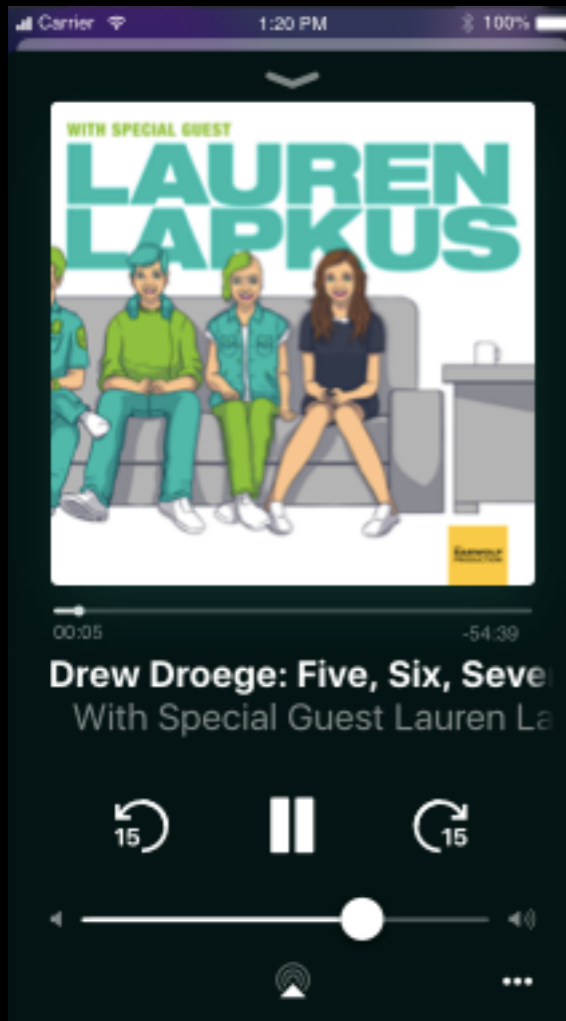
STYLES FOR THE CONTAINER

```
const styles = StyleSheet.create({
  container:{
    flex:1,
    backgroundColor: "#000D0C",
    flexDirection: 'column',
    justifyContent: 'center',
    alignItems: 'center'
  },
  ControlContainer:{
    flexDirection: 'row',
    justifyContent: 'center',
    alignItems: 'center'
  }
})
```


THE PLAYBUTTON.

```
render(){
  return(
    <View style={styles.container}>
      <View style={ styles.container} >
        <TouchableOpacity
          onPress={
            ()=>{this.toggleSound()}
          }
        >
          {!this.state.isPlaying ? (
            <Image
              source={require('../assets/playButton.png')} />
            ):(
              <Image
                source={require('../assets/pauseButton.png')} />
            )}
        </TouchableOpacity>
      </View>
    </View>
  )
}
```

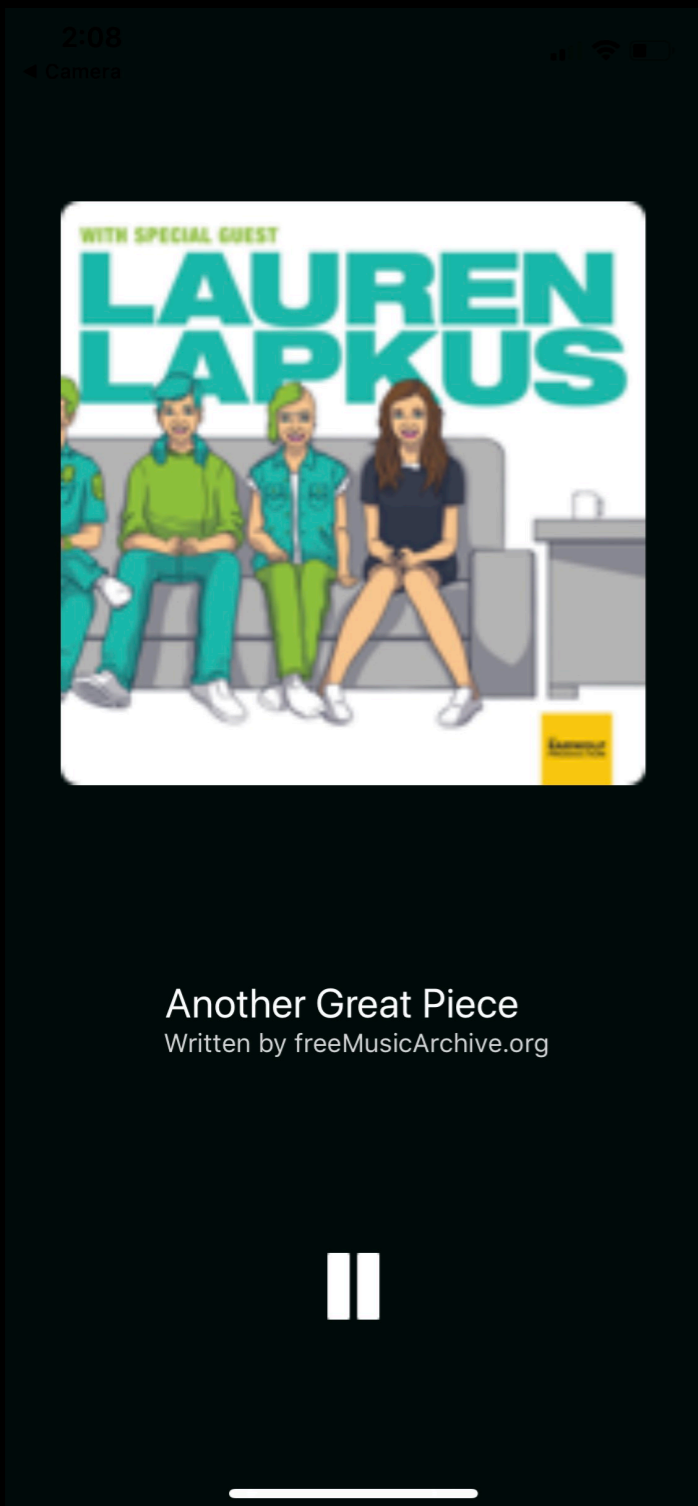
ADDING THE FEATURE PLAY IMAGE



```
<Image style={styles.featurePlayerImage} source={{uri: this.state.image}} />
```

```
featurePlayerImage: {  
  width: 315,  
  height: 315,  
},
```

Stored the image as part
of the state



```
<View>  
  <Text style={styles.title}> {this.state.title}</Text>  
  <Text style={styles.subTitle}> {this.state.subTitle}</Text>  
</View>
```

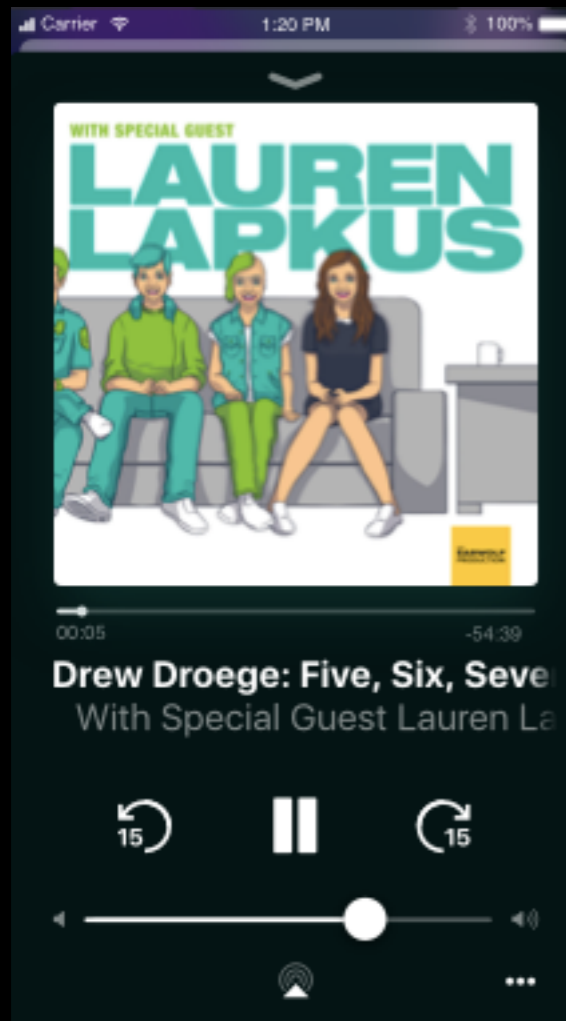
GROUPED SO THEY STAY TOGETHER

```
subTitle:{  
  color: "#D8D8D8"  
},  
title:{  
  fontSize: 22,  
  color: "#FFFFFF"  
},
```



SUBSCRIBING TO EVENTS

- In order to display the current progress of the song we need subscribe to updates

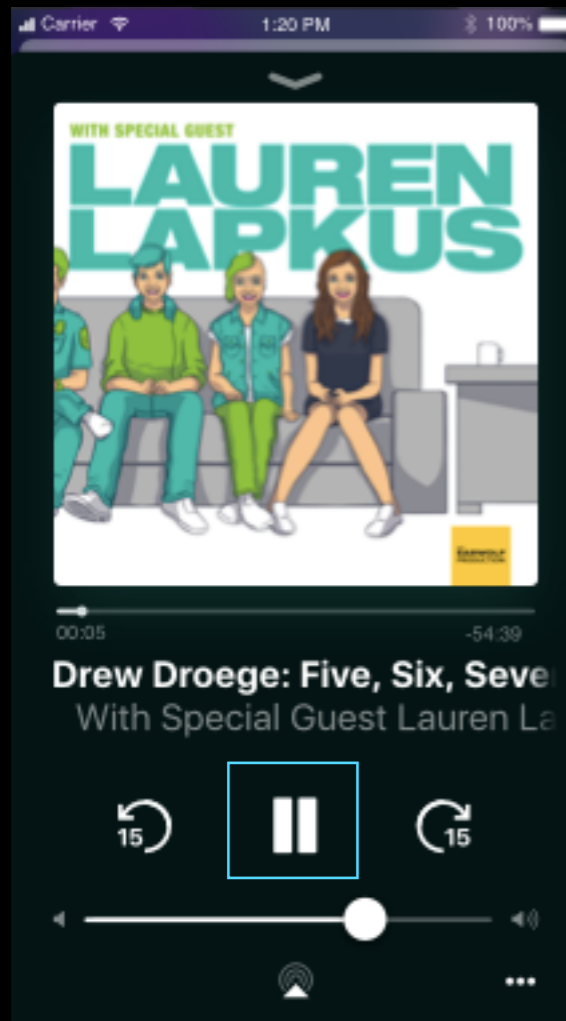


```
componentDidMount(){
  const soundObject = new Audio.Sound()
  soundObject.loadAsync(require("../assets/tunes3.mp3")).then( ()=>{
    this.setState({
      soundLoaded: true,
      soundObject: soundObject
    })
    soundObject.setOnPlaybackStatusUpdate( ()=>{
      this.onPlaybackStatusUpdate()
    })
  })
}
```

```
onPlaybackStatusUpdate(){
  let status = JSON.parse(this.state.soundObject._lastStatusUpdate)
  this.setState({
    soundObjectStatus: status
  })
}
```

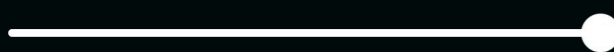
SUBSCRIBING TO EVENTS

- Also update the status checks for the play button. And updates in other places [see git repo for more details](#).



```
{ this.state.soundLoaded ? (  
  <View style={ styles.controlContainer} >  
    <TouchableOpacity  
      onPress={  
        ()=>{this.toggleSound()}  
      }  
    >  
      {this.state.soundObjectStatus === null ||  
!this.state.soundObjectStatus.isPlaying? (  
        <Image  
          source={require('../assets/playButton.png')} />  
      ):(  
        <Image  
          source={require('../assets/pauseButton.png')} />  
      )}  
    </TouchableOpacity>  
  </View>  
):( <View/>  
)  
}
```

Use expo slider for slider



Another Great Piece
Written by freeMusicArchive.org



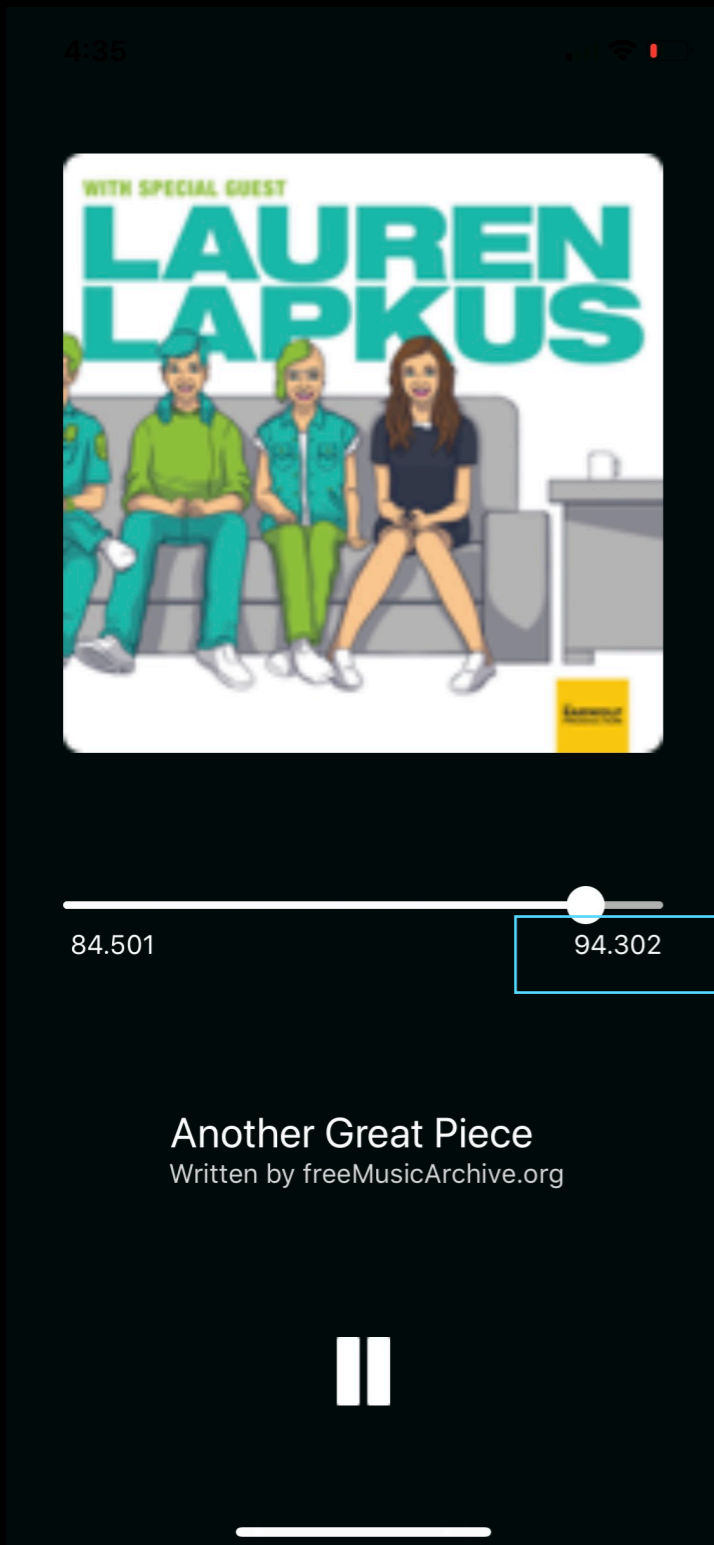
```
import Slider from "react-native-slider";
```

npm install react-native-slider

```
{ this.state.soundLoaded && this.state.soundObjectStatus !== null ?(  
  <Slider  
    style={styles.slider}  
    value = {this.state.soundObjectStatus.positionMillis/1000}  
    maximumValue = {this.state.soundObjectStatus.durationMillis/1000}  
    minimumTrackTintColor = "#FFF"  
    thumbTintColor = "#FFF"  
    minimumValue = {0}  
  />):  
  (  
    <Slider  
      style={styles.slider}  
      value = {0}  
      maximumValue = {10}  
      minimumTrackTintColor = "#FFF"  
      thumbTintColor = "#FFF"  
      minimumValue = {0}  
    />  
  )  
}
```



ADD A TEXT VIEW



```
{ this.state.soundLoaded && this.state.soundObjectStatus !== null ?(  
  <View>  
    <Slider  
      style={styles.slider}  
      value = {this.state.soundObjectStatus.positionMillis/1000}  
      maximumValue = {this.state.soundObjectStatus.durationMillis/1000}  
      minimumTrackTintColor = "#FFF"  
      thumbTintColor = "#FFF"  
      minimumValue = {0}  
    />  
    <View style={styles.sliderContainer}  
style={{paddingTop: 10, flexDirection: 'row', justifyContent: 'space-between'}}>  
      <Text style={{color: "#fff"}}>  
        {this.state.soundObjectStatus.positionMillis/1000}</Text>  
      <Text style={{color: "#fff"}}>  
        {this.state.soundObjectStatus.durationMillis/1000}</Text>  
    </View>  
  </View>  
) :  
(  
  <Slider  
    style={styles.slider}  
    value = {0}  
    maximumValue = {10}  
    minimumTrackTintColor = "#FFF"  
    thumbTintColor = "#FFF"  
    minimumValue = {0}  
  />  
)  
}
```

4:47



94.302

94.302

Another Great Piece

Written by freeMusicArchive.org



```
<TouchableOpacity>  
  <Image  
    source={require('../assets/Backward15.png')} />  
</TouchableOpacity>
```

```
async rewind(time){  
  try {  
    await this.soundObject.setPositionAsync(  
this.state.soundObjectStatus.positionMillis - time)  
  }catch(err){  
    console.log(err)  
  }  
}
```

```
<TouchableOpacity onPress={() =>{this.rewind(15000)}} >  
  <Image source={require('../assets/Backward15.png')} />  
</TouchableOpacity>
```


THE DEMO

MAP COMPONENT

<https://github.com/react-native-community/react-native-maps>

<https://snack.expo.io/@professorxii/map-snack>

LOCATION

<https://docs.expo.io/versions/latest/sdk/location/>

<https://snack.expo.io/@professorxii/expo-map-and-location-example>

PERMISSIONS

