```
package com.internshala.echo.adapters
import android.content.Context
import android.support.v7.widget.RecyclerView
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.RelativeLayout
import android.widget.TextView
import android.widget.Toast
import com.internshala.echo.R
import com.internshala.echo.Songs
/**
 * Created by Harsh Deep Singh on 2/13/2018.
 */

/*This adapter class also serves the same function to act as a bridge between the single row
view and its data. The implementation is quite similar to the one we did
* for the navigation drawer adapter*/
class MainScreenAdapter(_songDetails: ArrayList<Songs>, _context: Context) :
RecyclerView.Adapter<MainScreenAdapter.MyViewHolder>() {

    /*Local variables used for storing the data sent from the fragment to be used in the
adapter
    * These variables are initially null*/
    var songDetails: ArrayList<Songs>? = null
    var mContext: Context? = null

    /*In the init block we assign the data received from the params to our local
variables*/
    init {
        this.songDetails = _songDetails
        this.mContext = _context
    }
    override fun onBindViewHolder(holder: MyViewHolder, position: Int) {
        val songObject = songDetails?.get(position)

        /*The holder object of our MyViewHolder class has two properties i.e
        * trackTitle for holding the name of the song and
        * trackArtist for holding the name of the artist*/
        holder.trackTitle?.text = songObject?.songTitle
        holder.trackArtist?.text = songObject?.artist

        /*Handling the click event i.e. the action which happens when we click on any
song*/
        holder.contentHolder?.setOnClickListener({
            Toast.makeText(mContext, " Hey " + songObject?.songTitle,
Toast.LENGTH_SHORT).show()
        })
    }

    /*This has the same implementation which we did for the navigation drawer adapter*/
    override fun onCreateViewHolder(parent: ViewGroup?, viewType: Int): MyViewHolder {
        val itemView = LayoutInflater.from(parent?.context)
                .inflate(R.layout.row_custom_mainscreen_adapter, parent, false)
        return MyViewHolder(itemView)
    }
    override fun getItemCount(): Int {

        /*If the array list for the songs is null i.e. there are no songs in your device
        * then we return 0 and no songs are displayed*/
        if (songDetails == null) {
            return 0
        }
```

```
        /*Else we return the total size of the song details which will be the total number
of song details*/
        else {
            return (songDetails as ArrayList<Songs>).size
        }
    }

    /*Every view holder class we create will serve the same purpose as it did when we
created it for the navigation drawer*/
    class MyViewHolder(view: View) : RecyclerView.ViewHolder(view) {

        /*Declaring the widgets and the layout used*/
        var trackTitle: TextView? = null
        var trackArtist: TextView? = null
        var contentHolder: RelativeLayout? = null

        /*Constructor initialisation for the variables*/
        init {
            trackTitle = view.findViewById(R.id.trackTitle) as TextView
            trackArtist = view.findViewById(R.id.trackArtist) as TextView
            contentHolder = view.findViewById(R.id.contentRow) as RelativeLayout
        }
    }
}
```