

```

package com.internshala.echo.adapters
import android.content.Context
import android.os.Bundle
import android.support.v4.app.FragmentActivity
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 com.internshala.echo.R
import com.internshala.echo.Songs
import com.internshala.echo.fragments.SongPlayingFragment
/**
 * Created by Harsh Deep Singh on 2/27/2018.
 */

class FavoriteAdapter(_songDetails: ArrayList<Songs>, _context: Context) :
RecyclerView.Adapter<FavoriteAdapter.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({

            /*Let's discuss this peice of code*/
            /*Firstly we define an object of the SongPlayingFragment*/
            val songPlayingFragment = SongPlayingFragment()

            /*A bundle is used to transfer data from one point in your activity to another
            * Here we create an object of Bundle to send the song details to the fragment
            so that we can display the song details there and also play the song*/
            var args = Bundle()

            /*putString() function is used for adding a string to the bundle object
            * the string written in green is the name of the string which is placed in the
            bundle object with the value of that string written alongside
            * Note: Remember the name of the strings/entities you place inside the bundle
            object as you will retrieve them later using the same name. And these names are case-
            sensitive*/
            args.putString("songArtist", songObject?.artist)

```

```

        args.putString("songTitle", songObject?.songTitle)
        args.putString("path", songObject?.songData)
        args.putInt("SongID", songObject?.songID?.toInt() as Int)
        args.putInt("songPosition", position)

        /*Here the complete array list is sent*/
        args.putParcelableArrayList("songData", songDetails)

        /*Using this we pass the arguments to the song playing fragment*/
        songPlayingFragment.arguments = args

        /*Now after placing the song details inside the bundle, we inflate the song
        playing fragment*/
        (mContext as FragmentActivity).supportFragmentManager
            .beginTransaction()
            .replace(R.id.details_fragment, songPlayingFragment)
            .commit()
    })
}

/*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
    }
}

```

