package com.androiddevs.mvvmnewsapp.ui.fragments  
  
import android.os.Bundle  
import android.util.Log  
import android.view.View  
import androidx.core.widget.addTextChangedListener  
import androidx.fragment.app.Fragment  
import androidx.lifecycle.Observer  
import androidx.recyclerview.widget.LinearLayoutManager  
import com.androiddevs.mvvmnewsapp.R  
import com.androiddevs.mvvmnewsapp.adapters.NewsAdapter  
import com.androiddevs.mvvmnewsapp.ui.NewsActivity  
import com.androiddevs.mvvmnewsapp.ui.NewsViewModel  
import com.androiddevs.mvvmnewsapp.util.Constants.Companion.*SEARCH\_NEWS\_TIME\_DELAY*import com.androiddevs.mvvmnewsapp.util.Resource  
import kotlinx.android.synthetic.main.fragment\_search\_news.\*  
import kotlinx.coroutines.Job  
import kotlinx.coroutines.MainScope  
import kotlinx.coroutines.delay  
import kotlinx.coroutines.launch  
  
class SearchNewsFragment : Fragment(R.layout.*fragment\_search\_news*) {  
  
 lateinit var viewModel: NewsViewModel  
 lateinit var newsAdapter: NewsAdapter  
 val TAG = "SearchNewsFragment"  
  
 override fun onViewCreated(view: View, savedInstanceState: Bundle?) {  
 super.onViewCreated(view, savedInstanceState)  
 viewModel = (*activity* as NewsActivity).viewModel  
 setupRecyclerView()  
  
 var job: Job? = null  
 etSearch.*addTextChangedListener* **{** editable **->** job?.cancel()  
 job = *MainScope*().*launch* **{** delay(*SEARCH\_NEWS\_TIME\_DELAY*)  
 editable?.*let* **{** if(editable.toString().*isNotEmpty*()) {  
 viewModel.searchNews(editable.toString())  
 }  
 **}  
 }  
 }** viewModel.searchNews.observe(*viewLifecycleOwner*, *Observer* **{** response **->** when(response) {  
 is Resource.Success -> {  
 hideProgressBar()  
 response.data?.*let* **{** newsResponse **->** newsAdapter.differ.submitList(newsResponse.articles)  
 **}** }  
 is Resource.Error -> {  
 hideProgressBar()  
 response.message?.*let* **{** message **->** Log.e(TAG, "An error occured: $message")  
 **}** }  
 is Resource.Loading -> {  
 showProgressBar()  
 }  
 }  
 **}**)  
 }  
  
 private fun hideProgressBar() {  
 paginationProgressBar.*visibility* = View.*INVISIBLE* }  
  
 private fun showProgressBar() {  
 paginationProgressBar.*visibility* = View.*VISIBLE* }  
  
 private fun setupRecyclerView() {  
 newsAdapter = NewsAdapter()  
 rvSearchNews.*apply* **{** *adapter* = newsAdapter  
 *layoutManager* = LinearLayoutManager(*activity*)  
 **}** }  
}

