|  |
| --- |
| **학번 : 201911204, 이름 : 이주형** |
| **2023년 10주차 실습 체크 리스트** |
| |  |  | | --- | --- | | **실습 1** | **완성** | |
| **실습 1 : [ HTML Parsing ]** |
| **activity\_main.xml** |
| <?xml version="1.0" encoding="utf-8"?> <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"  tools:context=".MainActivity">   <androidx.swiperefreshlayout.widget.SwipeRefreshLayout  android:id="@+id/swipe"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent">   <androidx.recyclerview.widget.RecyclerView  android:id="@+id/recyclerView"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_marginStart="8dp"  android:layout\_marginTop="8dp"  android:layout\_marginEnd="8dp"  android:layout\_marginBottom="8dp" />   </androidx.swiperefreshlayout.widget.SwipeRefreshLayout>  </LinearLayout> |
| **row.xml** |
| <?xml version="1.0" encoding="utf-8"?> <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:orientation="vertical"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content">    <TextView  android:id="@+id/newstitle"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_margin="5dp"  android:padding="5dp"  android:textSize="15sp"  android:textStyle="bold"  android:text="TextView" /> </LinearLayout> |
| **MainActivity.kt** |
| package com.frost.lab01html\_parsing  import android.content.Intent import android.net.Uri import androidx.appcompat.app.AppCompatActivity import android.os.Bundle import android.widget.Toast import androidx.recyclerview.widget.DividerItemDecoration import androidx.recyclerview.widget.LinearLayoutManager import com.frost.lab01html\_parsing.databinding.ActivityMainBinding import kotlinx.coroutines.CoroutineScope import kotlinx.coroutines.Dispatchers import kotlinx.coroutines.launch import kotlinx.coroutines.withContext import org.jsoup.Jsoup  class MainActivity : AppCompatActivity() {  lateinit var binding: ActivityMainBinding  lateinit var adapter: MyDataAdapter  val url = "https://www.melon.com/chart/index.htm"  val scope = *CoroutineScope*(Dispatchers.IO)  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  binding = ActivityMainBinding.inflate(*layoutInflater*)  setContentView(binding.*root*)  initLayout()  }   fun getnews(){  scope.*launch* **{**//IO 스코프라서 Main 스레드 내용 갱신 불가능 => withContext로 잠시 변경  adapter.items.clear()  //호출할 때마다 배열 비워서 다시 생성   //jsoup으로 network의 정보 갖고 오기  //document 내용에서 파싱하기  val doc = Jsoup.connect(url).get()  //Log.i("check",doc.toString())  val songs50 = doc.select("div>div.clfix>div.my\_fold>div>div>form>div.service\_list\_song.type02.d\_song\_list>table>tbody>tr.lst50")  // Log.i("check", songs.text())  for(song in songs50){  adapter.items.add(MyData(song.select("td>div.wrap>div.wrap\_song\_info>div.ellipsis.rank01>span>a").text(), song.select("td>div.wrap>div.wrap\_song\_info>div.ellipsis.rank02>span>a").text()))  }  val songs100 = doc.select("div>div.clfix>div.my\_fold>div>div>form>div.service\_list\_song.type02.d\_song\_list>table>tbody>tr.lst100")  // Log.i("check", songs.text())  for(song in songs100){  adapter.items.add(MyData(song.select("td>div.wrap>div.wrap\_song\_info>div.ellipsis.rank01>span>a").text(), song.select("td>div.wrap>div.wrap\_song\_info>div.ellipsis.rank02>span>a").text()))  }  withContext(Dispatchers.Main)**{**//잠시 Main 스레드로 변경  adapter.notifyDataSetChanged()  //UI 갱신  binding.swipe.*isRefreshing*=false  //갱신이 종료되었으니까 화면에서 안보이게 설정  **}  }** }  private fun initLayout() {  binding.swipe.setOnRefreshListener **{**//새로고침 콜백 함수  binding.swipe.*isRefreshing*=true  //true 설정시 progressBar가 화면에 보여짐  getnews()  **}** binding.recyclerView.*layoutManager* =  LinearLayoutManager(this, LinearLayoutManager.*VERTICAL*, false)  binding.recyclerView.addItemDecoration(DividerItemDecoration(this, LinearLayoutManager.*VERTICAL*))  //리사이클러뷰의 row마다 구분자로 사용 가능한 decoration 추가  adapter = MyDataAdapter(ArrayList<MyData>())  adapter.itemClickListener = object:MyDataAdapter.OnItemClickListener{  override fun OnItemClick(data:MyData, position: Int) {  Toast.makeText(this@MainActivity, data.name,Toast.*LENGTH\_SHORT*).show()  }  }  binding.recyclerView.*adapter*=adapter  getnews()  } } |
| **MyDataAdapter.kt** |
| package com.frost.lab01html\_parsing  import android.view.LayoutInflater import android.view.View import android.view.ViewGroup import android.widget.AdapterView.OnItemClickListener import androidx.recyclerview.widget.RecyclerView import com.frost.lab01html\_parsing.databinding.RowBinding  class MyDataAdapter (val items:ArrayList<MyData>):RecyclerView.Adapter<MyDataAdapter.ViewHolder>() {  interface OnItemClickListener{  fun OnItemClick(data:MyData, position: Int)  }  var itemClickListener:OnItemClickListener?=null   inner class ViewHolder(val binding: RowBinding):RecyclerView.ViewHolder(binding.*root*){  init {  binding.newstitle.setOnClickListener**{** itemClickListener?.OnItemClick(items[*adapterPosition*], *adapterPosition*)  **}** }  }  override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {  val view = RowBinding.inflate(LayoutInflater.from(parent.*context*),  parent, false)  return ViewHolder(view)  }   override fun getItemCount(): Int {  return items.size  }   override fun onBindViewHolder(holder: ViewHolder, position: Int) {  holder.binding.newstitle.*text* = items[position].title  } } |
| **MyData.kt** |
| package com.frost.lab01html\_parsing  data class MyData(var title:String, var name:String) {} |
| **실행결과** |
|  |