[https://youtu.be/FzrzDCcWh7Y](https://youtu.be/FzrzDCcWh7Y" \t "_blank)

**Работа с ToolBar**

По умолчанию базовая ToolBar отключена, это делается в файле themes

необходимо там поменять parent на

<style name="Base.Theme.Menu" parent="Theme.Material3.DayNight">

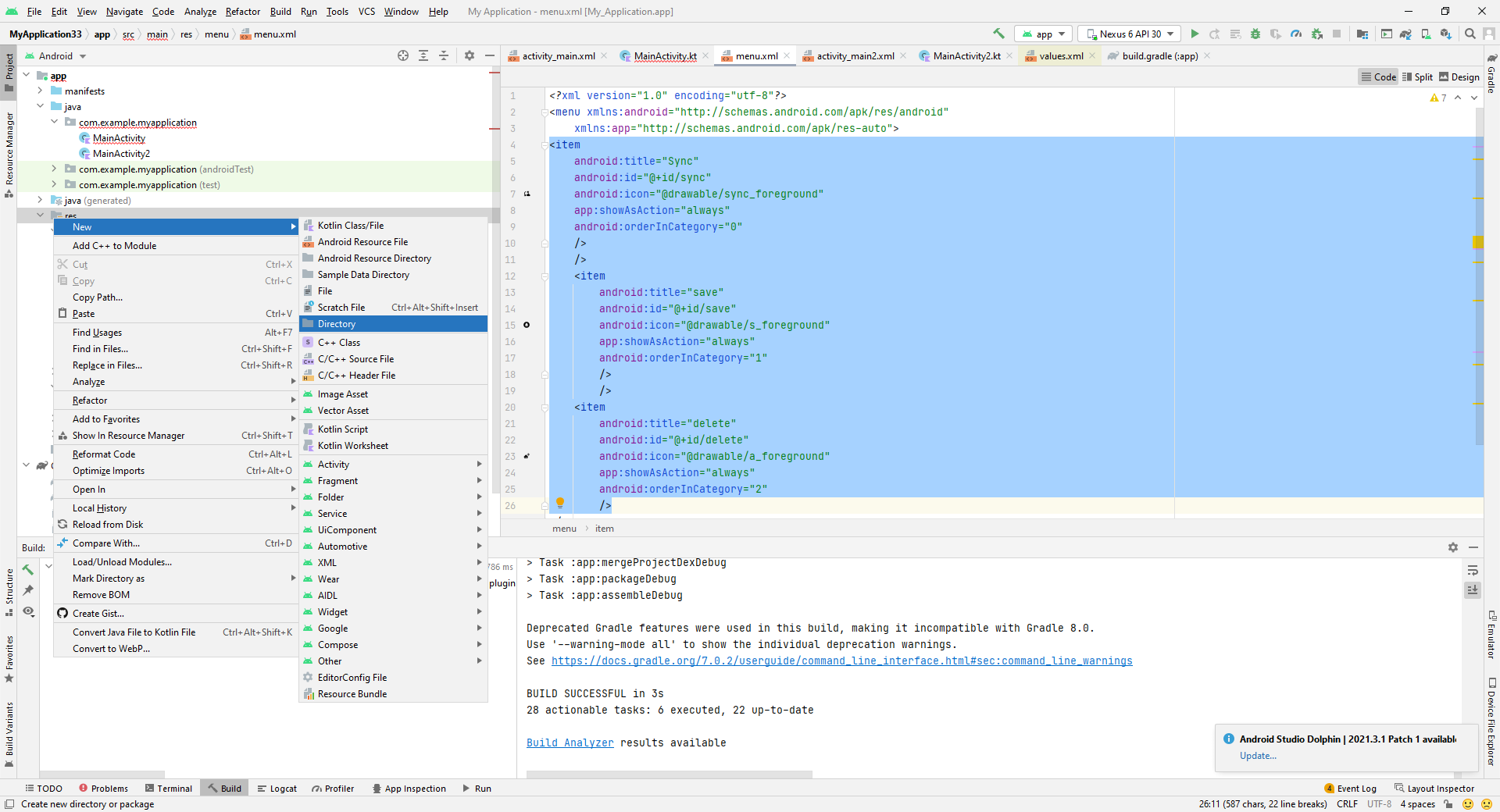
Определение кнопки возврата на ToolBar

override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
 *supportActionBar*?.setDisplayHomeAsUpEnabled(true)  
}

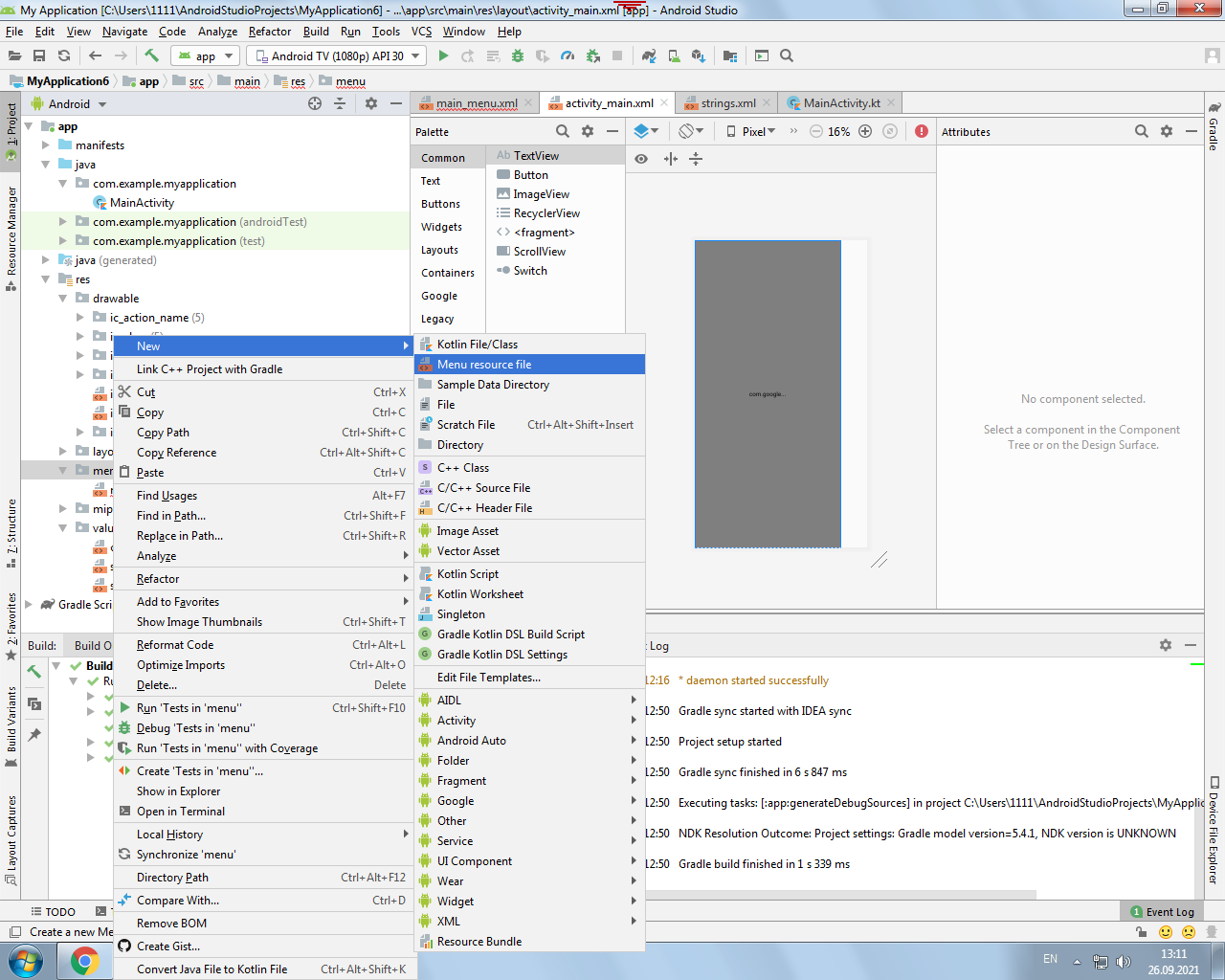
Определение действия при нажатии на кнопку

override fun onOptionsItemSelected(item: MenuItem): Boolean {  
  
 finish()  
  
 return true  
}

Создание папки для меню

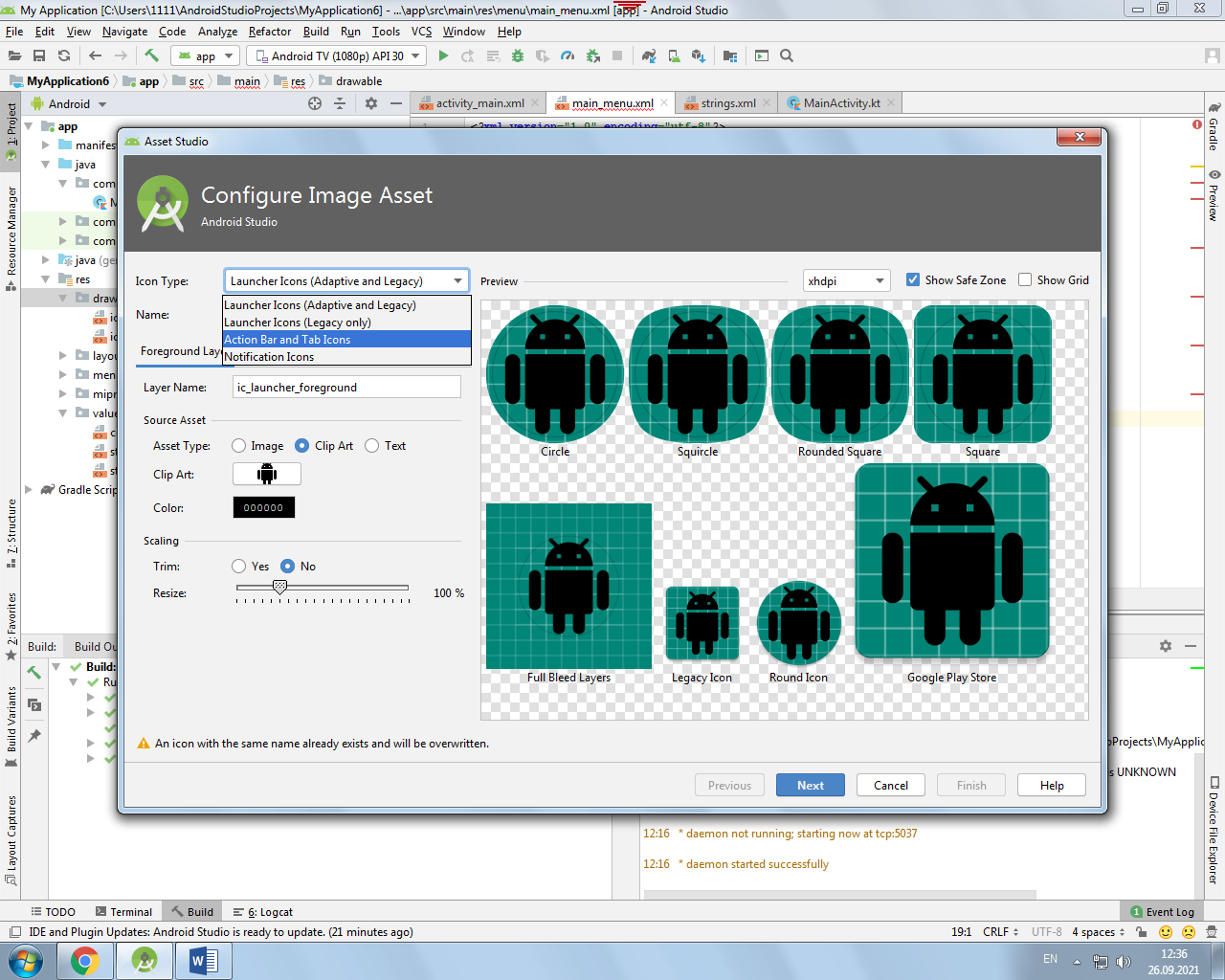


Создание файла меню в этой папке



Формирование ресурсов для меню

- 3 Иконок (имена можно давать произвольно, но не забудьте потом их при описании на свои поменять)



Формирования списка меню в файле меню

<item  
 android:title="Sync"  
 android:id="@+id/sync"  
 android:icon="@drawable/sync\_foreground"  
 app:showAsAction="always"  
 android:orderInCategory="0"  
 />  
 />  
 <item  
 android:title="save"  
 android:id="@+id/save"  
 android:icon="@drawable/s\_foreground"  
 app:showAsAction="always"  
 android:orderInCategory="1"  
 />  
 />  
 <item  
 android:title="delete"  
 android:id="@+id/delete"  
 android:icon="@drawable/a\_foreground"  
 app:showAsAction="always"  
 android:orderInCategory="2"  
 />

Перенести на форму компонент ToolBar , привязать его к форме и дать ему id=tb

Подключение меню к ToolBar через свойство menu

проверяем подключен ли файле сборки gradle (Module)

buildFeatures**{** viewBinding=true  
**}**

и прописываем в файле MainActivity в событии оnCreate

перед событием оnCreate

lateinit var binding: ActivityMainBinding

в событии оnCreate

binding = ActivityMainBinding.inflate(*layoutInflater*)  
setContentView(binding.*root*)  
  
binding.tb.setOnMenuItemClickListener()**{** when (**it**.*itemId*) {  
 android.R.id.*home* -> finish()  
 R.id.*sync* -> {  
 Toast.makeText(this, "Sync", Toast.*LENGTH\_SHORT*).show()  
 }  
 R.id.*save* -> {  
  
 Toast.makeText(this, "Save", Toast.*LENGTH\_SHORT*).show()  
 }  
  
 R.id.*delete* -> {  
 Toast.makeText(this, "Delete", Toast.*LENGTH\_SHORT*).show()  
 }  
 }  
 true  
  
**}**