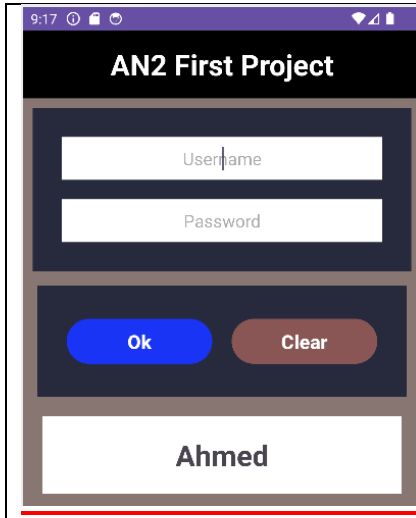


Android Studio

اهم الأدوات والخصائص والاوامر البرمجية



Palette

	Tools Name	اسماء الأدوات
1	Button	زر
2	TextView	مربع عرض نص
3	Textfield → Plain text	مربع ادخال نص
4	LinearLayout (Horizontal)	أطار تعرض الأدوات بداخله بشكل أفقي
5	LinearLayout (Vertical)	أطار تعرض الأدوات بداخله بشكل رأسي

Attributes

	Properties	كل أداة لها خصائص خاصة بها
1	android:text="Login"	الاسم الظاهر للمستخدم على الأداة
2	android:hint="Enter Your Name"	نص يظهر بداخل مربع الادخال للمساعدة ويختفي بالنقر بداخل الأداة PlainText (تلميح)
3	android:layout_margin="10dp"	الحدود الخارجية للأداة
4	android:padding="10dp"	الحدود الداخلية للأداة
5	android:id="@+id/login"	الاسم المستخدم في البرمجة id هذا ثابت (@+id/)
6	android:background="@color/black"	لون الخلفية لجميع العناصر ما عدا الزر
7	app:backgroundTint="#ff455a64"	لون الخلفية للزر فقط
8	android:textSize="30dp"	حجم الخط
9	android:textStyle="bold"	خط عريض
10	android:textColor="#fff8f8f8"	لون الخط
11	android:layout_width="match_parent"	عرض الأداة بعرض الاطار التي هي بداخله
12	android:layout_height="wrap_parent"	ارتفاع الأداة بارتفاع الاطار التي هي بداخله
13	android:orientation="vertical"	الأدوات بداخل الاطار رأسيه
14	android:orientation="horizontal"	الأدوات بداخل الاطار أفقية
15	android:gravity="center"	توسيط
16	app:backgroundTint="#ff455a64"	لون الخلفية للزر فقط

Connect between Activity & Layout page:

الربط بين ملف التصميم (.xml) والملف البرمجي (.kt)

```
setContentView(R.layout.activity_main)
```

Connect between the tools on the layout & activity:

الربط بين الأدوات في شاشة التصميم والملف البرمجي

```
val username = findViewById<EditText>(R.id.username)
val password = findViewById<EditText>(R.id.password)
val ok = findViewById<Button>(R.id.ok)
val clear = findViewById<Button>(R.id.clear)
val close = findViewById<Button>(R.id.close)
val show = findViewById<TextView>(R.id.show)
```

When Click on Button (ok)

عند الضغط على زر ok

```
ok.setOnClickListener { it: View!
    show.text = username.text.toString() + "\n" + password.text.toString()
}
```

```
ok.setOnClickListener { it: View!
    if (!username.text.isEmpty() && !password.text.isEmpty()){
        if(username.text.toString() == "aaa" && password.text.toString() == "123"){
            show.text = username.text.toString() + "\n" + password.text.toString()
        }else
            Toast.makeText(context: this, text: "Error", Toast.LENGTH_LONG).show()
    }else
        Toast.makeText(context: this, text: "Empty", Toast.LENGTH_LONG).show()
}
```

When Click on Button (clear)

عند الضغط على زر clear

```
clear.setOnClickListener { it: View!
    show.text = "Show"
    username.text.clear()
    password.text.clear()
}
```

When Click on Button (close)

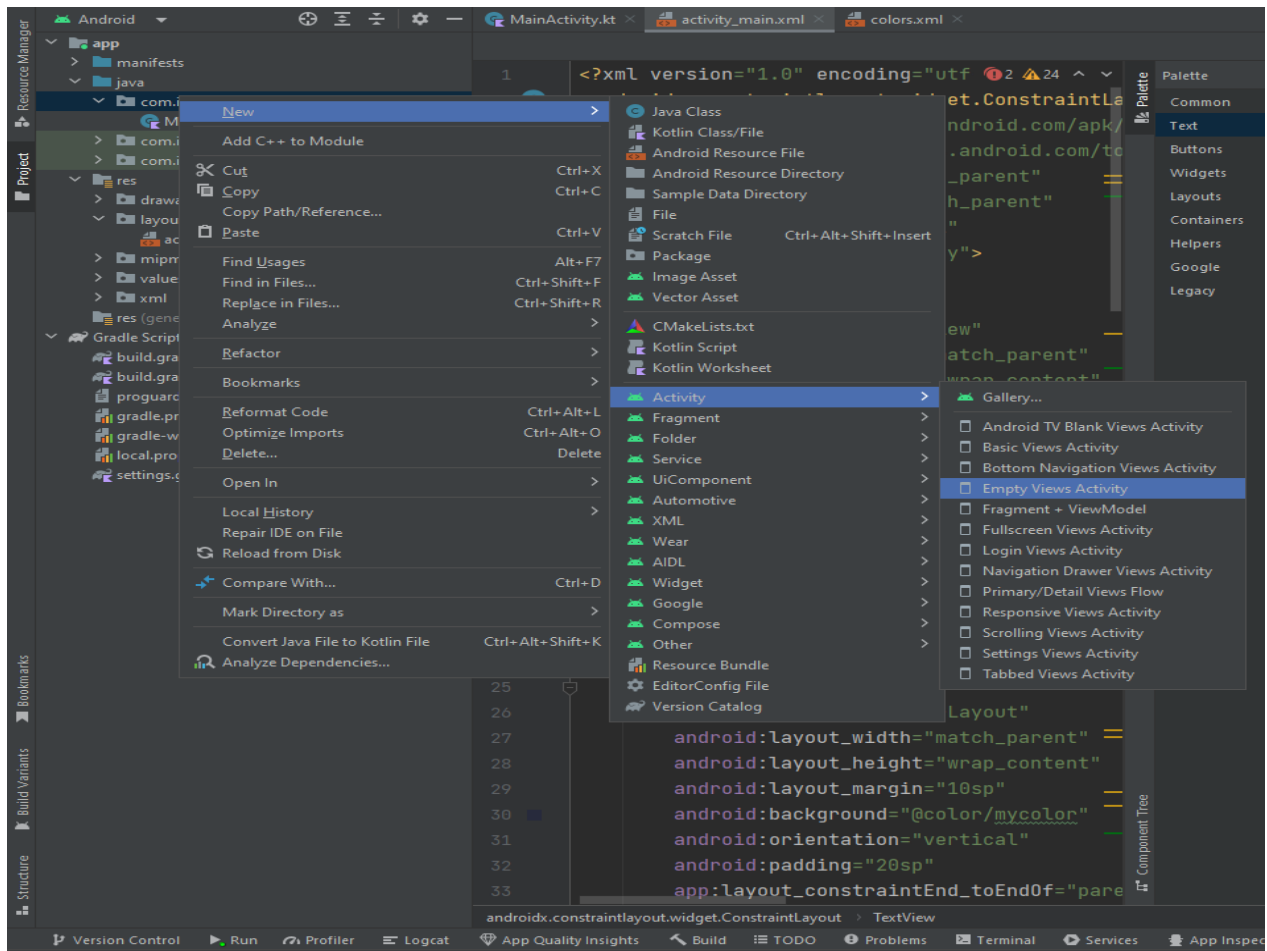
عند الضغط على زر close

```
close.setOnClickListener { it: View!
    System.exit( status: 0)
}
```

```
close.setOnClickListener { it: View!
    exitProcess( status: -1)
}
```

Add a new Activity (Page)

إضافة شاشة جوال جديدة



Navigation from MainActivity to NextActivity

الانتقال من شاشة الى أخرى

```
next.setOnClickListener{ it: View!
    val intent = Intent( packageContext: this@MainActivity, NextActivity::class.java)
    startActivity(intent)
}
```

Navigation from MainActivity to NextActivity with pass data

الجميل البرمجية للانتقال من شاشة الى أخرى مع نقل بيانات (تمرير بيانات pass data)

```
next.setOnClickListener{ it: View!
    val intent = Intent( packageContext: this@MainActivity, NextActivity::class.java)
    intent.putExtra( name: "username", username.text.toString())
    intent.putExtra( name: "password", password.text.toString())
    startActivity(intent)
}
```

استقبال القيمة المرسلة وعرضها مباشرة في TextView

```
val back = findViewById<ImageButton>(R.id.back)
val show = findViewById<TextView>(R.id.show)

show.text = intent.getStringExtra( name: "username") + "\n" + intent.getStringExtra( name: "password")
```

استقبال القيمة المرسلة في متغير ثم التأكد هل استقبل قيمة أولا

```
var user = intent.getStringExtra( name: "username")
var pass = intent.getStringExtra( name: "password")

// check if null
if(!user.isNullOrEmpty() && !pass.isNullOrEmpty()){
    show.text = user + "\n" + pass
}else{
    Toast.makeText( context: this, text: "is empty", Toast.LENGTH_LONG).show()
}
```

اتصال برقم تلفون – Call

```
call.setOnClickListener {
    val i = Intent(Intent.ACTION_CALL,
        Uri.parse("tel:" + phone_web.text.toString().trim()))
    startActivity(i)
}
```

الاذونات Permissions

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.CALL_PHONE" />
```

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.CALL_PHONE" />
```

Permission exists without corresponding hardware <uses-feature android:name="android.hardware.telephony" required="false"> tag

Add uses-feature tag Alt+Shift+Enter More actions... Alt+Enter

```
<application android:allowBackup="true"
```

```
<uses-feature
    android:name="android.hardware.telephony"
    android:required="false" />
```

Go to App inf

فتح موقع مثل قوقل – Web

```
web.setOnClickListener {  
  
    val i = Intent(Intent.ACTION_VIEW,  
        Uri.parse(phone_web.text.toString().trim()))  
    startActivity(i)  
}
```

فتح موقع على الخريطة – Map

```
map.setOnClickListener {  
    var w = latitude.text.toString().trim()  
    var t = longitude.text.toString()  
    val i = Intent(Intent.ACTION_VIEW)  
    i.data = Uri.parse("geo:$w,$t")  
    val chooser = Intent.createChooser(i, "open map")  
    startActivity(chooser)  
}
```

ارسال ايميل - Send Email

```
class EmailActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_email)  
  
        val mail_from = findViewById<EditText>(R.id.mail_from)  
        val mail_to = findViewById<EditText>(R.id.mail_to)  
        val subject = findViewById<EditText>(R.id.subject)  
        val msg = findViewById<EditText>(R.id.msg)  
  
        val sendEmail = findViewById<Button>(R.id.sendEmail)  
  
        sendEmail.setOnClickListener { it: View!  
            val i = Intent(Intent.ACTION_SEND)  
            i.data = Uri.parse(uriString: "mailto:")  
            var to = arrayOf(mail_from.text.toString(), mail_to.text.toString())  
            i.putExtra(Intent.EXTRA_EMAIL, to)  
            i.putExtra(Intent.EXTRA_SUBJECT, subject.text.toString())  
            i.putExtra(Intent.EXTRA_TEXT, msg.text.toString())  
            i.type = "message/rfc822"  
            val chooser = Intent.createChooser(i, title: "Send an Email")  
            startActivity(chooser)  
        }  
    }  
}
```

The screenshot displays the user interface of the 'Email Activity'. It features a dark header with the title 'Email Activiy'. Below the header, there are four input fields: 'mail from', 'mail to', 'Subject', and a large text area labeled 'msg'. At the bottom of the form, there is a green button labeled 'send Email'.