桂林电子科技大学2023-2024学年 第2学期

**Android应用开发 实验报告**

**课号：2322143         任课老师：刘斯韵**

|  |  |  |  |
| --- | --- | --- | --- |
| 项目名称 | **实验三、Activity和Service** | | |
| 院系 | **计算机与信息安全学院** | 专业 | **物联网工程** |
| 学号 | 2200100501 | 姓名 | **韩睿鑫** |
| 实验日期 | **2024年4月7日** | | |

一． 实验目的

掌握Activity的生命周期、创建和应用基本方法；

掌握Service的生命周期、创建和使用的基本方法；

掌握Activity调用Activity并传值的基本方法；

掌握Activity创建和绑定Service的基本方法；

掌握Activity和Service间通信的基本方法。

二． 实验内容

1. 创建ActivityA和ActivityB，并在配置文件中申明这些Activity；

2. 有ActivityA启动ActivityB，并传递信息，Toast显示传递信息后返回ActivityB；

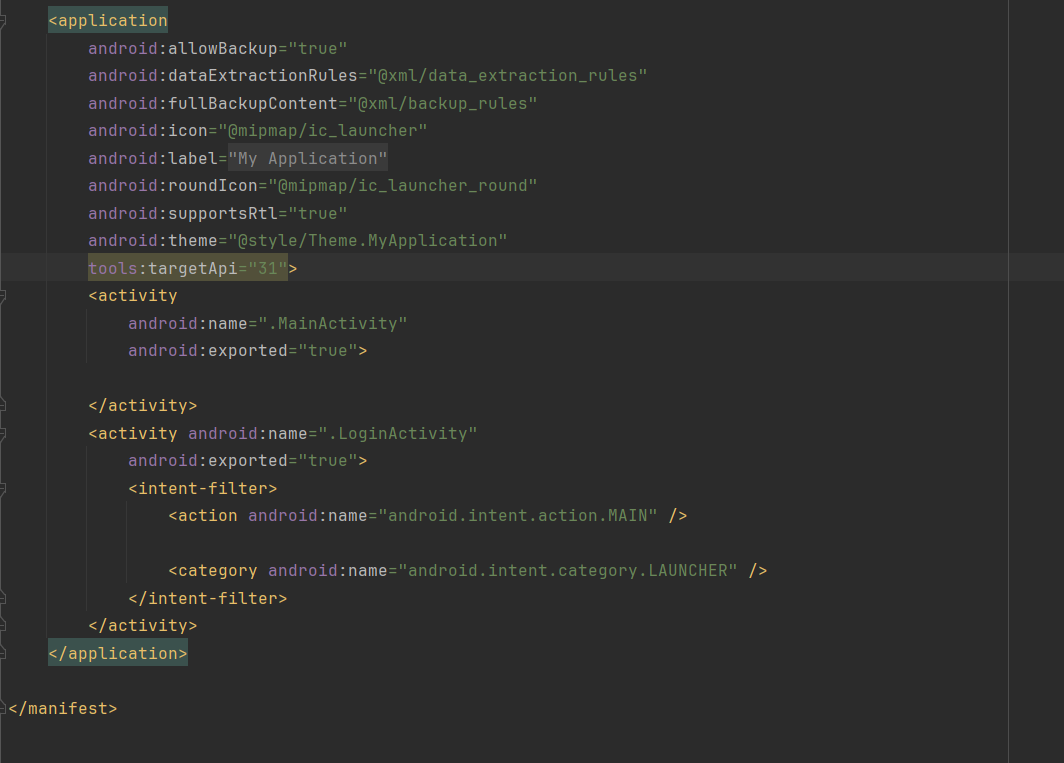
3. ActivityA绑定Service，并维护一个Binder；

4. Activity和Service间通过Binder传递消息。

**三、实验过程和代码**

过程步骤和截图，以及代码分析

 创建MainActivity和LoginActivity并配置申明



 LoginActivity启动MainActivity，并传递信息，Toast显示传递信息后返回MainActivity:传送给MainActivity的登录时间页面截图如下：



LoginActivity代码如下：

|  |
| --- |
| class LoginActivity : AppCompatActivity() {   private lateinit var usernameEditText: EditText  private lateinit var passwordEditText: EditText  private lateinit var loginButton: Button   private lateinit var sharedPreferences: SharedPreferences  private val sharedPrefFile = "login\_prefs"  private val keyUsername = "username"  private val keyPassword = "password"  private lateinit var dbHelper: DatabaseHelper  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  setContentView(R.layout.*activity\_login*)    // 初始化视图  usernameEditText = findViewById(R.id.*edit\_enter\_account*)  passwordEditText = findViewById(R.id.*edit\_enter\_password*)  loginButton = findViewById(R.id.*bt\_enter*)   // 初始化 Shared Preferences  sharedPreferences = getSharedPreferences(sharedPrefFile, Context.*MODE\_PRIVATE*)   // 检查 Shared Preferences 中是否已经存储了用户名和密码  val savedUsername = sharedPreferences.getString(keyUsername, "")  val savedPassword = sharedPreferences.getString(keyPassword, "")  if (savedUsername != null && savedPassword != null) {  // 如果已经存储了用户名和密码，则将其显示在相应的 EditText 中  usernameEditText.setText(savedUsername)  passwordEditText.setText(savedPassword)  }   // 点击登录按钮时保存用户名和密码到 Shared Preferences  loginButton.setOnClickListener **{** val username = usernameEditText.*text*.toString()  val password = passwordEditText.*text*.toString()  // 使用 SharedPreferences.Editor 保存数据到 Shared Preferences  val editor = sharedPreferences.edit()  editor.putString(keyUsername, username)  editor.putString(keyPassword, password)  editor.apply()  showToast("恭喜"+username+",您已成功保存并登录")  dbHelper = DatabaseHelper(this)   // 用户登录时将当前时间存储到数据库中  dbHelper.insertLoginTime()  showToast(username+"您的登录时间为"+dbHelper.getLastLoginTime())  val intent = Intent(this@LoginActivity,MainActivity::class.*java*)  val message = "你的登录时间为"+dbHelper.getLastLoginTime()  intent.putExtra("EXTRA\_MESSAGE", message)  startActivity(intent)   **}** }  fun showToast(message: String) {  Toast.makeText(this@LoginActivity, message, Toast.*LENGTH\_SHORT*).show()  } } |

MainActivity代码如下：

|  |
| --- |
| class MainActivity:AppCompatActivity() {  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  val message = *intent*.getStringExtra("EXTRA\_MESSAGE")  Toast.makeText(this@MainActivity, message, Toast.*LENGTH\_SHORT*).show()  }  } |

建立一个Service

代码如下：

|  |
| --- |
| class MainService : Service() {   private val binder = MainBinder()   inner class MainBinder : Binder() {  fun sendMessage(message: String) {  // 在此处处理从 Activity 接收到的消息  *println*("Received message in Service: $message")  }  }   override fun onBind(intent: Intent?): IBinder? {  return binder  } } |

Activity代码如下：

|  |
| --- |
| class MainActivity:AppCompatActivity() {  private var bound = false  private var myBinder: MainService.MainBinder? = null   private val connection = object : ServiceConnection {  override fun onServiceConnected(name: ComponentName?, service: IBinder?) {  myBinder = service as MainService.MainBinder  bound = true   // 在此处发送消息到 Service  myBinder?.sendMessage("Hello from ActivityA!")  }   override fun onServiceDisconnected(name: ComponentName?) {  bound = false  myBinder = null  }  }   override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  setContentView(R.layout.*activity\_main*)  val message = *intent*.getStringExtra("EXTRA\_MESSAGE")  Toast.makeText(this@MainActivity, message, Toast.*LENGTH\_SHORT*).show()  val intent = Intent(this, MainService::class.*java*)  bindService(intent, connection, Context.*BIND\_AUTO\_CREATE*)  }   override fun onDestroy() {  super.onDestroy()  if (bound) {  unbindService(connection)  bound = false  }  }  } |

在MainActivity中绑定了MainService，并通过Binder向Service发送消息，在Service中，的innner类中接收到Activity的消息

**四、遇到问题和解决办法**

 在学习的过程中我不知道Service到底是什么，具体的作用是什么。于是我进行上网查询学习到，Service(服务)是一个一种可以在后台执行长时间运行操作而没有用户界面的应用组件。服务可由其他应用组件启动（如Activity），服务一旦被启动将在后台一直运行，即使启动服务的组件（Activity）已销毁也不受影响。 此外，组件可以绑定到服务，以与之进行交互，甚至是执行进程间通信 。因此Service作为Android四大组件之一十分重要。

**五、实验总结**

在学习的过程中我总是会遇到很多问题，莫名其妙的报错，完全陌生的东西。因此我学会了询问身边的人和利用网络资源进行学习，我通过在b站、CSDN、掘金等平台上学习android使我的技术精进了不少，了解到了更多有关安卓的东西，在以后的学习道路上我要再接再厉，学会更多android必须的东西，不断完善我的APP!