构建环境

在你的对应Module中的 **build.gradle**文件中添加,然后**Rebuild Project**就可使用DataBinding了

android {

defaultConfig {  
 multiDexEnabled true  
}

dataBinding {

enabled = true

}

}

Layout 文件

<?xml version="1.0" encoding="utf-8"?>

<layout xmlns:android="http://schemas.android.com/apk/res/android">

<data>

<variable name="user" type="com.example.User"/>

</data>

<LinearLayout

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TextView android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/name"

android:text="@{user.firstName}"/>

<TextView android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@{user.lastName}"/>

</LinearLayout>

</layout>

使用**dataBinding**后根布局就需要改成**layout**，然后在

<variable name="user" type="com.example.User"/>

最后再在Activity的onCreate中

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

MainActivityBinding binding = DataBindingUtil.setContentView(this, R.layout.main\_activity);

User user = new User("Test", "User");

binding.setUser(user);

}

至此即实现了显示绑定

双向绑定

只需要在**@**后面加一个**=**即可

<TextView android:text="@={user.lastName}"/>

获取view

DataBinding.id即可获取view，且自动识别类型

MainActivityBinding binding = DataBindingUtil.setContentView(this, R.layout.main\_activity);

binding.user