**MVP模式开发详解**

1.Mvp知识的详解

MVP的全称是Model-View-Presenter;MVP是从经典的开发模式mvc演变而来的;Controller/Presenter负责逻辑处理，Model提供数据，View负责显示。

1. mvp和mvc的区别

在mvp中view并不直接使用Model，他们之间的通信是通过Presenter(MVC中的Controller)来控制，所有的交互都是发生在Presenter内部，而MVC中的View会直接的从Model中读取数据而不是通过Controller。

1. 具体实例
2. 创建一个bean对象

**public class** IUser {  
 **private** String **ID**;  
 **private** String **userName**;  
 **private int age**;  
**public** IUser(){}  
 **public** IUser(String ID, String userName, **int** age) {  
 **this**.**ID** = ID;  
 **this**.**userName** = userName;  
 **this**.**age** = age;  
 }  
 **public** String getID() {  
 **return ID**;  
 }  
 **public void** setID(String ID) {  
 **this**.**ID** = ID;  
 }  
 **public** String getUserName() {  
 **return userName**;  
 }  
 **public void** setUserName(String userName) {  
 **this**.**userName** = userName;  
 }  
 **public int** getAge() {  
 **return age**;  
 }  
 **public void** setAge(**int** age) {  
 **this**.**age** = age;  
 }

}

1. 建立model（处理业务逻辑，这里指数据读写），先写接口，后写实现

**public interface** IUserModel {  
 **public** String getID() ;  
 **public void** setID(String ID);  
 **public void** setUserName(String userName);  
 **public void** setAge(**int** age);  
 **public boolean** saveUser(IUser user);  
 **public** IUser loadUser(String ID);  
}

(3)创建第二步的接口的实现类(类的内部实现具体的操作)

**public class** UserModel **implements** IUserModel {  
 **private** String **ID**;  
 **private** String **userName**;  
 **private int age**;  
 **private** MyDB **db**;  
 **public** UserModel(Context context){  
 **db** = **new** MyDB(context);  
 }  
 @Override  
 **public** String getID() {  
 **return this**.**ID**!=**null**?**this**.**ID**:**null**;  
 }  
 @Override  
 **public void** setID(String ID) {  
 **this**.**ID**=ID;  
 }  
 @Override  
 **public void** setUserName(String userName) {  
 **this**.**userName**=userName;  
 }  
 @Override  
 **public void** setAge(**int** age) {  
 **this**.**age**=age;  
 }  
 @Override  
 **public boolean** saveUser(IUser user) {  
 **return db**.addUser(user);  
 }  
 @Override  
 **public** IUser loadUser(String ID) {  
 **return db**.findUserById(ID);  
 }  
}

1. 建立view（更新ui中的view状态），这里列出需要操作当前view的方法，也是接口

**public interface** IUserView {  
 **public** String getID();  
 **public void** setID(String ID);  
 **public** String getUserName();  
 **public void** setUserName(String userName);  
 **public int** getAge();  
 **public void** setAge(**int** age);  
  
}

(5) 建立presenter（主导器，通过iView和iModel接口操作model和view），activity可以把所有逻辑给presenter处理，这样java逻辑就从手机的activity中分离出来。

**public class** UserPresenter {  
 **private** IUserView **iUserView**;  
 **private** IUserModel **iUserModel**;  
 **public** UserPresenter(IUserView view){  
 **this**.**iUserView**=view;  
 **iUserModel**=**new** UserModel(((Context)view));  
 }  
 **public boolean** saveUser(String ID,String userName,**int** age){**return iUserModel**.saveUser(**new** IUser(ID,userName,age));  
 }  
 **public void** loadUser( String id) {  
 IUser user = **iUserModel**.loadUser(id);  
 **iUserView**.setUserName(user.getUserName());  
 **iUserView**.setAge(user.getAge());  
 }  
}

1. activity中实现iview接口，在其中操作view，实例化一个presenter变量。

**public class** MainActivity **extends** AppCompatActivity **implements** IUserView,View.OnClickListener{  
**private** EditText **editId**,**editName**,**editAge**;  
 **private** UserPresenter **presenter**;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 **editId** = (EditText)findViewById(R.id.***userID***);  
 **editName**=(EditText)findViewById(R.id.***userName***);  
 **editAge** = (EditText)findViewById(R.id.***userAge***);  
 ((Button)findViewById(R.id.***save***)).setOnClickListener(**this**);  
 ((Button)findViewById(R.id.***load***)).setOnClickListener(**this**);  
 **presenter** = **new** UserPresenter(**this**);  
 }  
 @Override  
 **public** String getID() {  
 **return editId**.getText().toString().trim();  
 }  
 @Override  
 **public void** setID(String ID) {  
 }  
 @Override  
 **public** String getUserName() {  
 **return editName**.getText().toString().trim();  
 }  
 @Override  
 **public void** setUserName(String userName) {  
 **editName**.setText(userName);  
 }  
 @Override  
 **public int** getAge() {  
 **return** Integer.*parseInt*(**editAge**.getText().toString().trim());  
 }  
 @Override  
 **public void** setAge(**int** age) {  
 **editAge**.setText(age+**""**);  
 }  
 @Override  
 **public void** onClick(View v) {  
 **switch** (v.getId()){  
 **case** R.id.***save***:  
 Boolean tag = **presenter**.saveUser(getID(),getUserName(),getAge());  
 **if** (tag){  
 Toast.*makeText*(**this**,**"保存成功"**,Toast.***LENGTH\_SHORT***).show();  
 }**else** {  
 Toast.*makeText*(**this**,**"保存失败"**,Toast.***LENGTH\_SHORT***).show();  
 }  
 **break**;  
 **case** R.id.***load***:  
 **presenter**.loadUser(getID());  
 **break**;  
 **default**:  
 **break**;  
 }  
 }  
}