@PathVariable 可以使类似/academy/{id}中的id匹配到函数

SpringMVC用的不是@Path，是@RequestMapping

Spring4有个@Restcontroller，我这里用的还是3的@Controller

事务

<bean id="transactionManager"

class="org.springframework.orm.hibernate4.HibernateTransactionManager">

<property name="sessionFactory" ref="sessionFactory" />

</bean>

<tx:annotation-driven transaction-manager="transactionManager" />

<bean id="StudentDAO" class="com.restsiqs.DAO.StudentDAO">

<property name="sessionFactory">

<ref bean="sessionFactory" />

</property>

</bean>

用来处理requestMapping的

<bean

class="org.springframework.web.servlet.mvc.annotation.DefaultAnnotationHandlerMapping" />

<bean

class="org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter">

<property name="messageConverters">

<list>

<bean

class="org.springframework.http.converter.StringHttpMessageConverter">

<property name="supportedMediaTypes">

<list>

<value>text/plain;charset=UTF-8</value>

</list>

</property>

</bean>

</list>

</property>

</bean>

自动扫描各种@Controller，@Service

<context:annotation-config />

<context:component-scan base-package="com.restsiqs.\*" />

欢迎界面：只要在oncreate函数那里加这个就行

**new** Handler().postDelayed(**new** Runnable() {  
  
 @Override  
 **public void** run() {  
 Intent intent = **new** Intent(MainActivity.**this**, SecondActivity\_.**class**);  
 startActivity(intent);  
 MainActivity.**this**.finish();  
 }  
}, 2500);

Android联网许可（添加在application外）

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

使用线程、httpclient获得rest服务器的接口内容

Runnable networkTask = new Runnable() {

@Override

public void run() {

HttpEntity entity = null;

try {

String account = etAccount.getText().toString();

String password = etPassword.getText().toString();

String restURL;

restURL = processURL+account;

HttpClient httpclient = new DefaultHttpClient();

HttpGet request = new HttpGet(restURL);

// Log.i("devouty", "test:" + request.getURI());

HttpResponse response = httpclient.execute(request);

// Log.i("devouty", "response:" + response.getStatusLine());

entity = response.getEntity();

String result = EntityUtils.toString(entity);

Log.i("devouty", "test:" + response.getStatusLine()+"----------"+restURL);

Log.i("devouty", "test:" + result);

// get account & password into database

DatabaseUtil databaseUtil = new DatabaseUtil(LoginActivity.this);

SQLiteDatabase writableDatabase = null;

writableDatabase = databaseUtil.getWritableDatabase();

ContentValues cv = new ContentValues();

cv.put("account", account);

cv.put("password", password);

writableDatabase.insert("account", null, cv);

Cursor cursor = writableDatabase.query("account", null, null, null, null, null, null);

// jump to course list

Intent intent = new Intent(LoginActivity.this, CourseListActivity\_.class);

intent.putExtra("account", account);

// intent.putExtra("entity",EntityUtils.toString(entity));

startActivity(intent);

} catch (ClientProtocolException e) {

e.printStackTrace();

Log.i("devouty", "ClientProtocolException");

} catch (IOException e) {

e.printStackTrace();

Log.i("devouty", e.getMessage());

}

}

@Click(R.id.btnLogin)

public void login() {

new Thread(networkTask).start();

}

sqliteOpenHelper的oncreate方法一般就执行一次，，错了的话要清除手机信息。

后退两次退出

private long exitTime = 0;

@Override

public boolean onKeyDown(int keyCode, KeyEvent event) {

if (keyCode == KeyEvent.KEYCODE\_BACK) {

exit();

return false;

}

return super.onKeyDown(keyCode, event);

}

public void exit() {

if ((System.currentTimeMillis() - exitTime) > 2000) {

Toast.makeText(getApplicationContext(), "再按一次退出程序",

Toast.LENGTH\_SHORT).show();

exitTime = System.currentTimeMillis();

} else {

finish();

System.exit(0);

}

}