@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 = **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);

}

}

Intellij IDEA 错误

Cannot reload AVD list: cvc-enumeration-valid: Value '280dpi' is not facet-valid with respect to enumeration '[ldpi, mdpi, tvdpi, hdpi, xhdpi, 400dpi, xxhdpi, 560dpi, xxxhdpi]'. It must be a value from the enumeration.  
Error parsing D:\sdkforas\android-sdk-windows\system-images\android-22\android-wear\armeabi-v7a\devices.xml  
cvc-enumeration-valid: Value '280dpi' is not facet-valid with respect to enumeration '[ldpi, mdpi, tvdpi, hdpi, xhdpi, 400dpi, xxhdpi, 560dpi, xxxhdpi]'. It must be a value from the enumeration.  
Error parsing D:\sdkforas\android-sdk-windows\system-images\android-22\android-wear\x86\devices.xml

**用/sdk/tools/lib/devices.xml去替换system-images\android-22\android-wear\x86\devices.xml和system-images\android-22\android-wear\armeabi-v7a\devices.xml中的devices.xml,我就是这样解决的**

**Java基础，获取屏幕尺寸**

Dimension screensize = Toolkit.*getDefaultToolkit*().getScreenSize();  
**int** width = (**int**) screensize.getWidth();  
**int** height = (**int**) screensize.getHeight();

所用intellij可视化工具（拖控件）

<http://jdev.tw/blog/3823/intellij-idea-gui-designer-swing-form>

JTable中添加jbutton

t.getColumn("Button").setCellRenderer(new ButtonRenderer());

t.getColumn("Button").setCellEditor(new ButtonEditor(new JTextField()));

# 我在2015/11/3用gson，把serviceGUI端的pojo转成json，到service那里转成了pojo对象，测试成功，但是抛了别的异常。

严重: Servlet.service() for servlet [SpringMVC] in context with path [/RESTSIQS] threw exception [Request processing failed; nested exception is org.springframework.dao.DataIntegrityViolationException: Duplicate entry '1' for key 'PRIMARY'; SQL [n/a]; constraint [null]; nested exception is org.hibernate.exception.ConstraintViolationException: Duplicate entry '1' for key 'PRIMARY'] with root cause

com.mysql.jdbc.exceptions.jdbc4.MySQLIntegrityConstraintViolationException: Duplicate entry '1' for key 'PRIMARY'

at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)

at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:57)

at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45)

at java.lang.reflect.Constructor.newInstance(Constructor.java:526)

at com.mysql.jdbc.Util.handleNewInstance(Util.java:406)

at com.mysql.jdbc.Util.getInstance(Util.java:381)

at com.mysql.jdbc.SQLError.createSQLException(SQLError.java:1015)

at com.mysql.jdbc.SQLError.createSQLException(SQLError.java:956)

at com.mysql.jdbc.MysqlIO.checkErrorPacket(MysqlIO.java:3515)

at com.mysql.jdbc.MysqlIO.checkErrorPacket(MysqlIO.java:3447)

at com.mysql.jdbc.MysqlIO.sendCommand(MysqlIO.java:1951)

at com.mysql.jdbc.MysqlIO.sqlQueryDirect(MysqlIO.java:2101)

at com.mysql.jdbc.ConnectionImpl.execSQL(ConnectionImpl.java:2554)

at com.mysql.jdbc.PreparedStatement.executeInternal(PreparedStatement.java:1761)

at com.mysql.jdbc.PreparedStatement.executeUpdate(PreparedStatement.java:2046)

at com.mysql.jdbc.PreparedStatement.executeUpdate(PreparedStatement.java:1964)

at com.mysql.jdbc.PreparedStatement.executeUpdate(PreparedStatement.java:1949)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:606)

at org.hibernate.engine.jdbc.internal.proxy.AbstractStatementProxyHandler.continueInvocation(AbstractStatementProxyHandler.java:122)

at org.hibernate.engine.jdbc.internal.proxy.AbstractProxyHandler.invoke(AbstractProxyHandler.java:81)

at com.sun.proxy.$Proxy19.executeUpdate(Unknown Source)

at org.hibernate.engine.jdbc.batch.internal.NonBatchingBatch.addToBatch(NonBatchingBatch.java:56)

at org.hibernate.persister.entity.AbstractEntityPersister.insert(AbstractEntityPersister.java:2859)

at org.hibernate.persister.entity.AbstractEntityPersister.insert(AbstractEntityPersister.java:3300)

at org.hibernate.action.internal.EntityInsertAction.execute(EntityInsertAction.java:88)

at org.hibernate.engine.spi.ActionQueue.execute(ActionQueue.java:362)

at org.hibernate.engine.spi.ActionQueue.executeActions(ActionQueue.java:354)

at org.hibernate.engine.spi.ActionQueue.executeActions(ActionQueue.java:275)

at org.hibernate.event.internal.AbstractFlushingEventListener.performExecutions(AbstractFlushingEventListener.java:326)

at org.hibernate.event.internal.DefaultFlushEventListener.onFlush(DefaultFlushEventListener.java:52)

at org.hibernate.internal.SessionImpl.flush(SessionImpl.java:1214)

at org.hibernate.internal.SessionImpl.managedFlush(SessionImpl.java:403)

at org.hibernate.engine.transaction.internal.jdbc.JdbcTransaction.beforeTransactionCommit(JdbcTransaction.java:101)

at org.hibernate.engine.transaction.spi.AbstractTransactionImpl.commit(AbstractTransactionImpl.java:175)

at org.springframework.orm.hibernate4.HibernateTransactionManager.doCommit(HibernateTransactionManager.java:480)

at org.springframework.transaction.support.AbstractPlatformTransactionManager.processCommit(AbstractPlatformTransactionManager.java:754)

at org.springframework.transaction.support.AbstractPlatformTransactionManager.commit(AbstractPlatformTransactionManager.java:723)

at org.springframework.transaction.interceptor.TransactionAspectSupport.commitTransactionAfterReturning(TransactionAspectSupport.java:393)

at org.springframework.transaction.interceptor.TransactionInterceptor.invoke(TransactionInterceptor.java:120)

at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:172)

at org.springframework.aop.framework.Cglib2AopProxy$DynamicAdvisedInterceptor.intercept(Cglib2AopProxy.java:622)

at com.restsiqs.Services.CourseService$$EnhancerByCGLIB$$c7a80417.save(<generated>)

at com.restsiqs.Resources.CourseResource.update(CourseResource.java:91)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:606)

at org.springframework.web.bind.annotation.support.HandlerMethodInvoker.invokeHandlerMethod(HandlerMethodInvoker.java:176)

at org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter.invokeHandlerMethod(AnnotationMethodHandlerAdapter.java:436)

at org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter.handle(AnnotationMethodHandlerAdapter.java:424)

at org.springframework.web.servlet.DispatcherServlet.doDispatch(DispatcherServlet.java:923)

at org.springframework.web.servlet.DispatcherServlet.doService(DispatcherServlet.java:852)

at org.springframework.web.servlet.FrameworkServlet.processRequest(FrameworkServlet.java:882)

at org.springframework.web.servlet.FrameworkServlet.doPut(FrameworkServlet.java:800)

at javax.servlet.http.HttpServlet.service(HttpServlet.java:647)

at javax.servlet.http.HttpServlet.service(HttpServlet.java:725)

at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:291)

at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)

at org.springframework.web.filter.HiddenHttpMethodFilter.doFilterInternal(HiddenHttpMethodFilter.java:77)

at org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:76)

at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:239)

at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)

at org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:52)

at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:239)

at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)

at org.springframework.web.filter.CharacterEncodingFilter.doFilterInternal(CharacterEncodingFilter.java:88)

at org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:76)

at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:239)

at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)

at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:219)

at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:106)

at org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:506)

at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:142)

at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:79)

at org.apache.catalina.valves.AbstractAccessLogValve.invoke(AbstractAccessLogValve.java:610)

at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:88)

at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:537)

at org.apache.coyote.http11.AbstractHttp11Processor.process(AbstractHttp11Processor.java:1081)

at org.apache.coyote.AbstractProtocol$AbstractConnectionHandler.process(AbstractProtocol.java:658)

at org.apache.coyote.http11.Http11NioProtocol$Http11ConnectionHandler.process(Http11NioProtocol.java:222)

at org.apache.tomcat.util.net.NioEndpoint$Sock

etProcessor.doRun(NioEndpoint.java:1566)

at org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.run(NioEndpoint.java:1523)

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1145)

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615)

at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61)

at java.lang.Thread.run(Thread.java:744)

# when I use:

**try** {  
 **courseTable**.setModel(**model**);  
} **catch** (ArrayIndexOutOfBoundsException e) {  
}

there is exception:

Exception in thread "AWT-EventQueue-0" java.lang.ArrayIndexOutOfBoundsException: 13 >= 12

at java.util.Vector.elementAt(Vector.java:474)

at javax.swing.table.DefaultTableColumnModel.getColumn(DefaultTableColumnModel.java:294)

at javax.swing.plaf.basic.BasicTableUI.paintCells(BasicTableUI.java:2012)

at javax.swing.plaf.basic.BasicTableUI.paint(BasicTableUI.java:1812)

at javax.swing.plaf.ComponentUI.update(ComponentUI.java:161)

at javax.swing.JComponent.paintComponent(JComponent.java:778)

at javax.swing.JComponent.paint(JComponent.java:1054)

at javax.swing.JComponent.paintToOffscreen(JComponent.java:5222)

at javax.swing.RepaintManager$PaintManager.paintDoubleBuffered(RepaintManager.java:1572)

at javax.swing.RepaintManager$PaintManager.paint(RepaintManager.java:1495)

at javax.swing.RepaintManager.paint(RepaintManager.java:1265)

at javax.swing.JComponent.\_paintImmediately(JComponent.java:5170)

at javax.swing.JComponent.paintImmediately(JComponent.java:4981)

at javax.swing.RepaintManager$4.run(RepaintManager.java:824)

at javax.swing.RepaintManager$4.run(RepaintManager.java:807)

at java.security.AccessController.doPrivileged(Native Method)

at java.security.ProtectionDomain$1.doIntersectionPrivilege(ProtectionDomain.java:75)

at javax.swing.RepaintManager.paintDirtyRegions(RepaintManager.java:807)

at javax.swing.RepaintManager.paintDirtyRegions(RepaintManager.java:782)

at javax.swing.RepaintManager.prePaintDirtyRegions(RepaintManager.java:731)

at javax.swing.RepaintManager.access$1300(RepaintManager.java:64)

at javax.swing.RepaintManager$ProcessingRunnable.run(RepaintManager.java:1720)

at java.awt.event.InvocationEvent.dispatch(InvocationEvent.java:311)

at java.awt.EventQueue.dispatchEventImpl(EventQueue.java:749)

at java.awt.EventQueue.access$500(EventQueue.java:97)

at java.awt.EventQueue$3.run(EventQueue.java:702)

at java.awt.EventQueue$3.run(EventQueue.java:696)

at java.security.AccessController.doPrivileged(Native Method)

at java.security.ProtectionDomain$1.doIntersectionPrivilege(ProtectionDomain.java:75)

at java.awt.EventQueue.dispatchEvent(EventQueue.java:719)

at java.awt.EventDispatchThread.pumpOneEventForFilters(EventDispatchThread.java:201)

at java.awt.EventDispatchThread.pumpEventsForFilter(EventDispatchThread.java:116)

at java.awt.EventDispatchThread.pumpEventsForHierarchy(EventDispatchThread.java:105)

at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:101)

at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:93)

at java.awt.EventDispatchThread.run(EventDispatchThread.java:82)

# wtf is



# JTextField限制为数字

//number input limitation

KeyListener NumLimitListener = new KeyListener() {

public void keyTyped(KeyEvent e) {

if (e.getKeyChar() < '0' ||

e.getKeyChar() > '9') {

e.consume();

return;

}

}

public void keyPressed(KeyEvent e) {

}

public void keyReleased(KeyEvent e) {

}

};

*//number limit***tfTest1**.addKeyListener(**NumLimitListener**);  
**tfTest2**.addKeyListener(**NumLimitListener**);  
**tfTest3**.addKeyListener(**NumLimitListener**);  
**tfE1**.addKeyListener(**NumLimitListener**);  
**tfE2**.addKeyListener(**NumLimitListener**);  
**tfE3**.addKeyListener(**NumLimitListener**);  
**tfE4**.addKeyListener(**NumLimitListener**);  
**tfE5**.addKeyListener(**NumLimitListener**);  
**tfFinalTest**.addKeyListener(**NumLimitListener**);  
**tfDailyMark**.addKeyListener(**NumLimitListener**);  
**tfCredit**.addKeyListener(**NumLimitListener**);  
**tfSum**.addKeyListener(**NumLimitListener**);

# Jtable 能双击的重点在于一个好model

**model** = **new** DefaultTableModel(**data**, names) {  
 **public boolean** isCellEditable(**int** row, **int** column) {  
 **return false**;  
 }  
};

# Jtable的显示设置

*// courseTable.getTableHeader().setFont(new Font("宋体", Font.BOLD, 18));  
// courseTable.setFont(new Font("宋体", 0, 20));  
// courseTable.setRowHeight(23);*