<http://www.2cto.com/kf/201308/233461.html>

注册与登录

Zhuce activity

/\* private EditText register\_username;

private EditText register\_passwd;private EditText register\_id;

private EditText reregister\_passwd;

private Button register\_submit;

@SuppressLint("NewApi")

@TargetApi(Build.VERSION\_CODES.GINGERBREAD)\*/

/\* StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();

StrictMode.setThreadPolicy(policy);

setContentView(R.layout.enroll);

register\_username=(EditText)findViewById(R.id.textView2);

register\_id=(EditText)findViewById(R.id.textView1);

register\_passwd=(EditText)findViewById(R.id.textView3);

reregister\_passwd=(EditText)findViewById(R.id.textView4);

register\_submit=(Button)findViewById(R.id.button1);

register\_username.setOnFocusChangeListener(new OnFocusChangeListener()

{

@Override

public void onFocusChange(View v, boolean hasFocus) {

// **TODO** Auto-generated method stub

if(!hasFocus){

if(register\_id.getText().toString().trim().length()!=5){

Toast.makeText(Zhuce.this, "ID必须为5个数字", Toast.LENGTH\_SHORT).show();

}

}

}

});

register\_passwd.setOnFocusChangeListener(new OnFocusChangeListener()

{

@Override

public void onFocusChange(View v, boolean hasFocus) {

// **TODO** Auto-generated method stub

if(!hasFocus){

if(register\_passwd.getText().toString().trim().length()<6){

Toast.makeText(Zhuce.this, "密码不能小于8个字符", Toast.LENGTH\_SHORT).show();

}

}

}

});

reregister\_passwd.setOnFocusChangeListener(new OnFocusChangeListener()

{

@Override

public void onFocusChange(View v, boolean hasFocus) {

// **TODO** Auto-generated method stub

if(!hasFocus){

if(!reregister\_passwd.getText().toString().trim().equals(register\_passwd.getText().toString().trim())){

Toast.makeText(Zhuce.this, "两次密码输入不一致", Toast.LENGTH\_SHORT).show();

}

}

}

});

register\_submit.setOnClickListener(new OnClickListener(){

@Override

public void onClick(View v) {

if(!checkEdit()){

return;

}

// **TODO** Auto-generated method stub

String httpUrl="http://192.168.1.100:8080/web-test/register.jsp";

HttpPost httpRequest=new HttpPost(httpUrl);

List<NameValuePair> params=new ArrayList<NameValuePair>();

params.add(new BasicNameValuePair("id",register\_id.getText().toString().trim()));

params.add(new BasicNameValuePair("username",register\_username.getText().toString().trim()));

params.add(new BasicNameValuePair("password",register\_passwd.getText().toString().trim()));

HttpEntity httpentity = null;

try {

httpentity = new UrlEncodedFormEntity(params,"utf8");

} catch (UnsupportedEncodingException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

httpRequest.setEntity(httpentity);

HttpClient httpclient=new DefaultHttpClient();

HttpResponse httpResponse = null;

try {

httpResponse = httpclient.execute(httpRequest);

} catch (ClientProtocolException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

if(httpResponse.getStatusLine().getStatusCode()==200)

{

String strResult = null;

try {

strResult = EntityUtils.toString(httpResponse.getEntity());

} catch (ParseException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

Toast.makeText(Zhuce.this, strResult, Toast.LENGTH\_SHORT).show();

}

else

{

Toast.makeText(Zhuce.this, "请求错误", Toast.LENGTH\_SHORT).show();

}

}

});

}

private boolean checkEdit(){

if(register\_id.getText().toString().trim().equals("")){

Toast.makeText(Zhuce.this, "职工号不能为空", Toast.LENGTH\_SHORT).show();

if(register\_username.getText().toString().trim().equals(""))

{

Toast.makeText(Zhuce.this, "用户名不能为空", Toast.LENGTH\_SHORT).show();

}

else if(register\_passwd.getText().toString().trim().equals(""))

{

Toast.makeText(Zhuce.this, "密码不能为空", Toast.LENGTH\_SHORT).show();

}

else if(!register\_passwd.getText().toString().trim().equals(reregister\_passwd.getText().toString().trim())){

Toast.makeText(Zhuce.this, "两次密码输入不一致", Toast.LENGTH\_SHORT).show();

}

else{

return true;

}

return false;

}

return false;}

}

\*/

# 主入口

/\*@TargetApi(Build.VERSION\_CODES.GINGERBREAD)

@SuppressLint("NewApi")

private EditText login\_id;

private EditText login\_username;

private EditText login\_password;

private Button user\_register\_button;

@TargetApi(Build.VERSION\_CODES.GINGERBREAD)

@SuppressLint("NewApi")

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();

StrictMode.setThreadPolicy(policy);

setContentView(R.layout.activity\_main);

initWidget();

}

private void initWidget()

{ login\_id=(EditText)findViewById(R.id.textView1);

login\_password=(EditText)findViewById(R.id.textView3);

user\_login\_button=(Button)findViewById(R.id.button1);

user\_register\_button=(Button)findViewById(R.id.button2);

user\_login\_button.setOnClickListener(this);

user\_register\_button.setOnClickListener(this);

login\_id.setOnFocusChangeListener(new OnFocusChangeListener()

{

@Override

public void onFocusChange(View v, boolean hasFocus) {

// **TODO** Auto-generated method stub

if(!hasFocus){

String username=login\_username.getText().toString().trim();

if(username.length()<4){

Toast.makeText(MainActivity.this, "用户名不能小于4个字符", Toast.LENGTH\_SHORT);

}

}

}

});

login\_password.setOnFocusChangeListener(new OnFocusChangeListener()

{

@Override

public void onFocusChange(View v, boolean hasFocus) {

// **TODO** Auto-generated method stub

if(!hasFocus){

String password=login\_password.getText().toString().trim();

if(password.length()<4){

Toast.makeText(MainActivity.this, "密码不能小于4个字符", Toast.LENGTH\_SHORT);

}

}

}

});

}

@Override

public void onClick(View v) {

// **TODO** Auto-generated method stub

switch(v.getId())

{

case R.id.button1:

if(checkEdit())

{

login();

}

break;

case R.id.button2:

Intent intent2=new Intent(MainActivity.this,Zhuce.class);

startActivity(intent2);

break;

}

}

private boolean checkEdit(){

if(login\_username.getText().toString().trim().equals("")){

Toast.makeText(MainActivity.this, "用户名不能为空", Toast.LENGTH\_SHORT).show();

}else if(login\_password.getText().toString().trim().equals("")){

Toast.makeText(MainActivity.this, "密码不能为空", Toast.LENGTH\_SHORT).show();

}else{

return true;

}

return false;

}

private void login(){

String httpUrl="http://192.168.1.102:8080/web-test/login.jsp";

HttpPost httpRequest=new HttpPost(httpUrl);

List<NameValuePair> params=new ArrayList<NameValuePair>();

params.add(new BasicNameValuePair("username",login\_username.getText().toString().trim()));

params.add(new BasicNameValuePair("password",login\_password.getText().toString().trim()));

HttpEntity httpentity = null;

try {

httpentity = new UrlEncodedFormEntity(params,"utf8");

} catch (UnsupportedEncodingException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

httpRequest.setEntity(httpentity);

HttpClient httpclient=new DefaultHttpClient();

HttpResponse httpResponse = null;

try {

httpResponse = httpclient.execute(httpRequest);

} catch (ClientProtocolException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

if(httpResponse.getStatusLine().getStatusCode()==200)

{

String strResult = null;

try {

strResult = EntityUtils.toString(httpResponse.getEntity());

} catch (ParseException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

Toast.makeText(MainActivity.this, strResult, Toast.LENGTH\_SHORT).show();

Intent intent=new Intent(MainActivity.this,IndexActivity.class);

startActivity(intent);

}

else

{

Toast.makeText(MainActivity.this, "登录失败！", Toast.LENGTH\_SHORT).show();

}

}

}

\*/

# Muke

http://www.imooc.com/learn/107