**工作中用到了webview 下面把webview相关的知识进行一下总结**：

首先要在manifest.main文件中创建一个webview，然后再activity中定义这个webview然后进行一下相关操作。

1、添加权限：AndroidManifest.xml中必须使用许可"android.permission.INTERNET",否则会出Web page not available错误。

2、在要Activity中生成一个WebView组件：WebView webView = new WebView(this);

3、设置WebView基本信息：

如果访问的页面中有Javascript，则webview必须设置支持Javascript。

webview.getSettings().setJavaScriptEnabled(true);

触摸焦点起作用

requestFocus();

取消滚动条

this.setScrollBarStyle(SCROLLBARS\_OUTSIDE\_OVERLAY);

4如果希望点击链接由自己处理，而不是新开Android的系统browser中响应该链接。给WebView添加一个事件监听对象（WebViewClient)并重写其中的一些方法 shouldOverrideUrlLoading：对网页中超链接按钮的响应。 当按下某个连接时WebViewClient会调用这个方法，并传递参数：按下的url

onLoadResource

onPageStart

onPageFinish

onReceiveError

onReceivedHttpAuthRequest

5、如果访问的页面中有Javascript，则webview必须设置支持Javascript ，否则显示空白页面。

Java代码

       webview.getSettings().setJavaScriptEnabled(**true**);    
6、如果页面中链接，如果希望点击链接继续在当前browser中响应，而不是新开Android的系统browser中响应该链接，必须覆盖webview的WebViewClient对象：

Java代码

1. mWebView.setWebViewClient(**new** WebViewClient(){
2. **public** **boolean** shouldOverrideUrlLoading(WebView view, String url) {
3. view.loadUrl(url);
4. **return** **true**;
5. }
6. });

上述方法告诉系统由我这个WebViewClient处理这个Intent,我来加载URL。点击一个链接的Intent是向上冒泡的，shouldOverrideUrlLoading方法return true表示我加载后这个Intent就消费了，不再向上冒泡了。  
  
7、如果不做任何处理，在显示你的Brower UI时，点击系统“Back”键，整个Browser会作为一个整体“Back"到其他Activity中，而不是希望的在Browser的历史页面中 Back。如果希望实现在历史页面中Back，需要在当前Activity中处理并消费掉该Back事件：

Java代码

1. **public** **boolean** onKeyDown(**int** keyCode, KeyEvent event) {
2. **if** ((keyCode == KeyEvent.KEYCODE\_BACK) && mWebView.canGoBack()) {
3. mWebView.goBack();
4. **return** **true**;
5. }
6. **return** **super**.onKeyDown(keyCode, event);
7. }

对于Android 2.0开始又多出了一种新的方法，对于Activity 可以单独获取Back键的按下事件，直接重写                   onBackPressed 方法即可，代码如下

Java代码

@Override  
1、**public** **void** onBackPressed() {  
2、     // 这里处理逻辑代码，该方法仅适用于2.0或更高版本的sdk  
3、    **return** ;  
4、}

这里还有几个知识点：

1)为了让WebView从apk文件中加载 assets，Android SDK提供了一个schema，前缀为"file:///android\_asset/"。WebView遇到这样的schema，就去当前包中的 assets目录中找内容。如上面的"file:///android\_asset/demo.html"

2)addJavascriptInterface方法中要绑定的 Java对象及方法要运行另外的线程中，不能运行在构造他的线程中，这也是使用Handler的目的。

**Webview的两种显示网页的方法：**

（1）：webview.loadUrl(“www.baidu.com”);

**public** **class** WebviewTest **extends** Activity {

/\*\* Called when the activity is first created. \*/

**private** WebView mWebView;

**private** Button bt1;

**public** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*main*);

mWebView = (WebView)findViewById(R.id.*webview*);

WebSettings webSettings = mWebView.getSettings();

webSettings.setJavaScriptEnabled(**true**);

bt1 = (Button)findViewById(R.id.*Button01*);

bt1.setOnClickListener(**new** View.OnClickListener() {

**public** **void** onClick(View arg0) {

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

webview.loadUrl(“www.baidu.com”);

}

});

}

}

(2):自定义网页：Webview.data();

**public** **class** WebviewTest **extends** Activity {

/\*\* Called when the activity is first created. \*/

**private** WebView mWebView;

**private** Button bt1;

**public** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*main*);

mWebView = (WebView)findViewById(R.id.*webview*);

WebSettings webSettings = mWebView.getSettings();

webSettings.setJavaScriptEnabled(**true**);

bt1 = (Button)findViewById(R.id.*Button01*);

bt1.setOnClickListener(**new** View.OnClickListener() {

**public** **void** onClick(View arg0) {

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

String str = "asdas";

mWebView.loadData(

"<html><body>"+str+"</body></html>",

"text/html", "utf-8");

}

});

}

}

**Webview的常用方法实例：**

Manifest.xml中的代码：

加入权限：

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

布局文件中的代码：

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

<LinearLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:orientation=*"vertical"* android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"*>

<WebView android:id=*"@+id/WebView01"* android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"*></WebView>

</LinearLayout>

TestWebviewDemo中的代码：

**public** **class** TestWebviewDemo **extends** Activity {

/\*\* Called when the activity is first created. \*/

**private** **static** **final** String *TAG* = "TestWebviewDemo";

**private** WebView mWebView;

**private** Handler mHandler = **new** Handler();

**private** **int** mDensity;

**public** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*main*);

mWebView = (WebView) findViewById(R.id.*WebView01*);

mWebView.getSettings().setAllowFileAccess(**true**);// 设置允许访问文件数据

mWebView.getSettings().setBuiltInZoomControls(**true**);// 设置支持缩放

mWebView.getSettings().setSavePassword(**false**); // 设置是否保存密码

// 设置支持JavaScript脚本

mWebView.getSettings().setJavaScriptEnabled(**true**);

// 设置支持各种不同的设备

mWebView

.getSettings()

.setUserAgentString(

"Mozilla/5.0 (iPad; U; CPU OS 3\_2 like Mac OS X; en-us) AppleWebKit/531.21.10 (KHTML, like Gecko) Version/4.0.4 Mobile/7B334b Safari/531.21.10");

// 通过这个设置来执行加载webview网页时所要执行的一些方法

mWebView.setWebViewClient(**new** WebViewClient() {

// 新开页面时用自己定义的webview来显示，不用系统自带的浏览器来显示

**public** **boolean** shouldOverrideUrlLoading(WebView view, String url) {

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

// 当有新连接时使用当前的webview进行显示

view.loadUrl(url);

**return** **super**.shouldOverrideUrlLoading(view, url);

}

// 开始加载网页时要做的工作

**public** **void** onPageStarted(WebView view, String url, Bitmap favicon) {

**super**.onPageStarted(view, url, favicon);

}

//加载完成时要做的工作

**public** **void** onPageFinished(WebView view, String url) {

**super**.onPageFinished(view, url);

}

// 加载错误时要做的工作

**public** **void** onReceivedError(WebView view, **int** errorCode,

String description, String failingUrl) {

Log.*d*(*TAG*, "error=" + description);

Toast.*makeText*(TestWebviewDemo.**this**,

errorCode + "/" + description, Toast.*LENGTH\_LONG*)

.show();

}

});

// 处理网页中的一些对话框信息（提示对话框，带选择的对话框，带输入的对话框）

mWebView.setWebChromeClient(**new** WebChromeClient() {

// 对话框

**public** **boolean** onJsAlert(WebView view, String url, String message,

**final** JsResult result) {

// 构建一个Builder来显示网页中的alert对话框

Builder builder = **new** Builder(TestWebviewDemo.**this**);

builder.setTitle("提示对话框");

builder.setMessage(message);

builder.setPositiveButton(android.R.string.*ok*,

**new** AlertDialog.OnClickListener() {

@Override

**public** **void** onClick(DialogInterface dialog,

**int** which) {

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

result.confirm();

}

});

builder.setCancelable(**false**);

builder.create();

builder.show();

**return** **true**;

}

// 带按钮的对话框

**public** **boolean** onJsConfirm(WebView view, String url,

String message, **final** JsResult result) {

Builder builder = **new** Builder(TestWebviewDemo.**this**);

builder.setTitle("带选择的对话框");

builder.setMessage(message);

builder.setPositiveButton(android.R.string.*ok*,

**new** AlertDialog.OnClickListener() {

@Override

**public** **void** onClick(DialogInterface dialog,

**int** which) {

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

result.confirm();

}

});

builder.setNeutralButton(android.R.string.*cancel*,

**new** AlertDialog.OnClickListener() {

@Override

**public** **void** onClick(DialogInterface dialog,

**int** which) {

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

result.cancel();

}

});

builder.setCancelable(**false**);

builder.create();

builder.show();

**return** **true**;

}

// 带输入框的对话框

**public** **boolean** onJsPrompt(WebView view, String url, String message,

String defaultValue, **final** JsPromptResult result) {

LayoutInflater inflater = LayoutInflater

.*from*(TestWebviewDemo.**this**);

**final** View v = inflater.inflate(R.layout.*prom\_dialog*, **null**);

// 设置 TextView对应网页中的提示信息

((TextView) v.findViewById(R.id.*TextView\_PROM*))

.setText(message);

// 设置EditText对应网页中的输入框

((EditText) v.findViewById(R.id.*EditText\_PROM*))

.setText(defaultValue);

Builder builder = **new** Builder(TestWebviewDemo.**this**);

builder.setTitle("带输入的对话框");

builder.setView(v);

builder.setPositiveButton(android.R.string.*ok*,

**new** AlertDialog.OnClickListener() {

@Override

**public** **void** onClick(DialogInterface dialog,

**int** which) {

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

String value = ((EditText) v

.findViewById(R.id.*EditText\_PROM*))

.getText().toString();

result.confirm(value);

}

});

builder.setNegativeButton(android.R.string.*cancel*,

**new** AlertDialog.OnClickListener() {

@Override

**public** **void** onClick(DialogInterface dialog,

**int** which) {

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

result.cancel();

}

});

builder

.setOnCancelListener(**new** DialogInterface.OnCancelListener() {

@Override

**public** **void** onCancel(DialogInterface dialog) {

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

result.cancel();

}

});

builder.create();

builder.show();

**return** **true**;

}

// 设置网页加载的进度条

**public** **void** onProgressChanged(WebView view, **int** newProgress) {

TestWebviewDemo.**this**.getWindow().setFeatureInt(

Window.*FEATURE\_PROGRESS*, newProgress \* 100);

**super**.onProgressChanged(view, newProgress);

}

// 设置应用程序的标题

**public** **void** onReceivedTitle(WebView view, String title) {

TestWebviewDemo.**this**.setTitle(title);

**super**.onReceivedTitle(view, title);

}

});

// 与网页进行交互的addJavascriptInterface()的方法

mWebView.addJavascriptInterface(**new** Object() {

**public** **void** clickOnAndroid(**final** String str) {

mHandler.post(**new** Runnable() {

@Override

**public** **void** run() {

// 逻辑代码

});

}

}, "demo");

**if** (mDensity == 240) { // 可以让不同的density的情况下，可以让页面进行适配

mWebView.getSettings().setDefaultZoom(ZoomDensity.*FAR*);

} **else** **if** (mDensity == 160) {

mWebView.getSettings().setDefaultZoom(ZoomDensity.*MEDIUM*);

} **else** {

mWebView.getSettings().setDefaultZoom(ZoomDensity.*CLOSE*);

}

String strUrl = "http://10.0.2.2:8080/WebTest3";

mWebView.loadUrl(strUrl);

}

//处理返回的事件，（后退进入前一个界面而不是全部退出）

@Override

**public** **boolean** onKeyDown(**int** keyCode, KeyEvent event) {

**if** (keyCode == KeyEvent.*KEYCODE\_BACK* && mWebView.canGoBack()) {

mWebView.goBack();// 返回前一个页面

**return** **true**;

}

**return** **super**.onKeyDown(keyCode, event);

}

**Webview中java与网页进行数据交互：**

先看我们的html文档：

<html>

<script language="javascript">

/\* This function is invoked by the activity \*/

function wave() {

alert("1");

document.getElementById("droid").src="android\_waving.png";

alert("2");

}

</script>

<body>

<!-- Calls into the javascript interface for the activity -->

//js调用Java方法

<a onClick="window.demo.clickOnAndroid()"><div style="width:80px;

margin:0px auto;

padding:10px;

text-align:center;

border:2px solid #202020;" >

<img id="droid" src="android\_normal.png"/><br>

Click me!

</div></a>

</body>

</html>

Manifest.Xml中：

加入权限：

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

再看我们的java 代码。

Java代码

public class WebViewDemo extends Activity {

private static final String LOG\_TAG = "WebViewDemo";

private WebView mWebView;

private Handler mHandler = new Handler();

@Override

public void onCreate(Bundle icicle) {

super.onCreate(icicle);

setContentView(R.layout.main);

mWebView = (WebView) findViewById(R.id.webview);

WebSettings webSettings = mWebView.getSettings();

webSettings.setSavePassword(false);

webSettings.setSaveFormData(false);

webSettings.setJavaScriptEnabled(true);

webSettings.setSupportZoom(false);

mWebView.setWebChromeClient(new MyWebChromeClient());

//自定义的Demo，供js网页调用

mWebView.addJavascriptInterface(new DemoJavaScriptInterface(), "demo");

mWebView.loadUrl("file:///android\_asset/demo.html");

}

final class DemoJavaScriptInterface {

DemoJavaScriptInterface() {

}

/\*\*

\* This is not called on the UI thread. Post a runnable to invoke

\* loadUrl on the UI thread.

\*/

public void clickOnAndroid() {

//用handler来更新UI

mHandler.post(new Runnable() {

public void run() {

//Java调用js方法

mWebView.loadUrl("javascript:wave()");

}

});

}

}

/\*\*

\* Provides a hook for calling "alert" from javascript. Useful for

\* debugging your javascript.

\*/

final class MyWebChromeClient extends WebChromeClient {

@Override

public boolean onJsAlert(WebView view, String url, String message, JsResult result) {

Log.d(LOG\_TAG, message);

result.confirm();

return true;

}

}

}

再看个例子：

Html中的代码：

<script type="text/JavaScript">

function s(){

alert("Good Morning!");

}

function d(){

confirm("Are you ok?")

}

function f(){

prompt("What’s your name?")

}

</script>

</head>

<body>

<body>

下面我们演示三种对话框

<br/><br/>

<input type="button" value="警告，提醒对话框" onclick="s()">

<br/><br/>

<input type="button" value="带选择的对话框" onclick="d()">

<br/><br/>

<input type="button" value="要求用户输入的对话框" onclick="f()">

</body>

Manifest.Xml中：

加入权限：

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

Java中的代码：

**public** **class** Test **extends** Activity {

/\*\* Called when the activity is first created. \*/

**private** EditText et = **null**;

**private** Button btn = **null**;

**private** WebView wv = **null**;

**private** WebSettings ws = **null**;

@Override

**public** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*main*);

et = (EditText) **this**.findViewById(R.id.*EditText01*);

btn = (Button) **this**.findViewById(R.id.*Button01*);

wv = (WebView) **this**.findViewById(R.id.*WebView*);

ws = wv.getSettings();

ws.setAllowFileAccess(**true**);//设置允许访问文件数据

ws.setJavaScriptEnabled(**true**);//设置支持javascript脚本

ws.setBuiltInZoomControls(**true**);//设置支持缩放

wv.requestFocus();

wv.setWebViewClient(**new** WebViewClient(){

**public** **boolean** shouldOverrideUrlLoading(WebView view,String url){

//当有新连接时，使用当前的 WebView

view.loadUrl(url);

**return** **true**;

}

});

wv.setWebChromeClient(**new** WebChromeClient(){

**public** **boolean** onJsAlert(WebView view,String url,String message,**final** JsResult result){

//构建一个Builder来显示网页中的alert对话框

Builder builder = **new** Builder(Test.**this**);

builder.setTitle("提示对话框");

builder.setMessage(message);

builder.setPositiveButton(android.R.string.*ok*, **new** AlertDialog.OnClickListener(){

@Override

**public** **void** onClick(DialogInterface dialog, **int** which) {

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

result.confirm();

}

});

builder.setCancelable(**false**);

builder.create();

builder.show();

**return** **true**;

}

**public** **boolean** onJsConfirm(WebView view,String url,String message,**final** JsResult result){

Builder builder = **new** Builder(Test.**this**);

builder.setTitle("带选择的对话框");

builder.setMessage(message);

builder.setPositiveButton(android.R.string.*ok*, **new** AlertDialog.OnClickListener(){

@Override

**public** **void** onClick(DialogInterface dialog, **int** which) {

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

result.confirm();

}

});

builder.setNeutralButton(android.R.string.*cancel*, **new** AlertDialog.OnClickListener(){

@Override

**public** **void** onClick(DialogInterface dialog, **int** which) {

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

result.cancel();

}

});

builder.setCancelable(**false**);

builder.create();

builder.show();

**return** **true**;

}

**public** **boolean** onJsPrompt(WebView view,String url,String message,String defaultValue,**final** JsPromptResult result){

LayoutInflater inflater = LayoutInflater.*from*(Test.**this**);

**final** View v = inflater.inflate(R.layout.*prom\_dialog*, **null**);

//设置 TextView对应网页中的提示信息

((TextView)v.findViewById(R.id.*TextView\_PROM*)).setText(message);

//设置EditText对应网页中的输入框

((EditText)v.findViewById(R.id.*EditText\_PROM*)).setText(defaultValue);

Builder builder = **new** Builder(Test.**this**);

builder.setTitle("带输入的对话框 ");

builder.setView(v);

builder.setPositiveButton(android.R.string.*ok*, **new** AlertDialog.OnClickListener(){

@Override

**public** **void** onClick(DialogInterface dialog, **int** which) {

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

String value = ((EditText)v.findViewById(R.id.*EditText\_PROM*)).getText().toString();

result.confirm(value);

}

});

builder.setNegativeButton(android.R.string.*cancel*, **new** AlertDialog.OnClickListener(){

@Override

**public** **void** onClick(DialogInterface dialog, **int** which) {

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

result.cancel();

}

});

builder.setOnCancelListener(**new** DialogInterface.OnCancelListener(){

@Override

**public** **void** onCancel(DialogInterface dialog) {

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

result.cancel();

}

});

builder.create();

builder.show();

**return** **true**;

}

//设置网页加载的进度条

**public** **void** onProgressChanged(WebView view,**int** newProgress){

Test.**this**.getWindow().setFeatureInt(Window.*FEATURE\_PROGRESS*, newProgress\*100);

**super**.onProgressChanged(view, newProgress);

}

//设置应用程序的标题

**public** **void** onReceivedTitle(WebView view,String title){

Test.**this**.setTitle(title);

**super**.onReceivedTitle(view, title);

}

});

btn.setOnClickListener(**new** Button.OnClickListener(){

@Override

**public** **void** onClick(View v) {

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

String url =et.getText().toString();

// String url = "http://10.0.2.2:8080/WebTest3";

//判断输入的内容是不是网址

**if**(URLUtil.*isNetworkUrl*(url)){

Log.*d*("++++++++++++", "sadas");

wv.loadUrl(url);

Toast.*makeText*(Test.**this**, url, Toast.*LENGTH\_SHORT*).show();

}**else**{

et.setHint("输入的网址不合法，请重新输入");

// et.setText("输入的网址不合法，请重新输入");

}

}

});

}

@Override

**public** **boolean** onKeyDown(**int** keyCode, KeyEvent event) {

**if**(keyCode==KeyEvent.*KEYCODE\_BACK* && wv.canGoBack()){

wv.goBack();//返回前一个页面

**return** **true**;

}

**return** **super**.onKeyDown(keyCode, event);

}

}

需要注意的地方，很多数据类型js中不认识，最好是在android那边封装好，提供必要的方法接口。比如传到js中的list，在js中是没办法去得到里面的元素的。

Manifest.Xml中：

加入权限：

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

html的带码

<script language="javascript">

window.onload= function(){

var i=window.javatojs.getSize();

for(var n=0;n<i;n++){

var jsdata= window.javatojs.getObject(n);//拿到activity里面的属性javadata

var datalistdiv = document.getElementById("datalist"); //得到页面的div

pnode = document.createElement("p");//创建一个p标签，再建个textnode

tnode = document.createTextNode(jsdata);

pnode.appendChild(tnode);//p中加入数据

datalistdiv.appendChild(pnode);//div中键入新的p

}

}

</script>

<body>

<div id = "datalist">

this is a demo

</body>

Java的代码：

public class JavaToWebview extends Activity {

private WebView web;

public List list;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

initData();

setContentView(R.layout.webview);

web = (WebView)this.findViewById(R.id.webview);

web.getSettings().setJavaScriptEnabled(true);//开启javascript设置

web.addJavascriptInterface(this, "javatojs");//把RIAExample的一个实例添加到js的全局对象window中， //这样就可以使用window.javatojs来调用它的方法

web.loadUrl("file:///android\_asset/demo5.html");//加载网页

}

void initData(){

list=new ArrayList<String>();

for(int i=0;i<5;i++){

list.add("我是从数据库中读取的哈哈");

}

}

/\*\*

\* 该方法将在js脚本中，通过window.javatojs.....()进行调用

\* @return

\*/

public Object getObject(int index){

return list.get(index);

}

public int getSize(){

return list.size();

}

}

**再看一个查地图的例子：**

Manifest.Xml中：

加入权限：

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

布局文件中的代码：

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

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

android:orientation="vertical"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

>

<TextView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="Welcome to Mr Wei's Blog."

/>

<WebView

android:id="@+id/webview"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

/>

<Button

android:id="@+id/button"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="Change the webview content"

/>

</LinearLayout>

在assets目录下新建一个demo.html文件,代码如下：

<html>

<script language="javascript"><!--

function fillContent(){

document.getElementById("content").innerHTML =

"This Content is showed by Android invoke Javascript function.";

}

// --></script>

<body>

<p><a onClick="window.demo.startMap()" href="">Start GoogleMap</a></p>

<p id="content"></p>

<p>A Demo ----Android and Javascript invoke each other.</p>

<p>Author:Frankiewei</p>

</body>

</html>

activity代码：

public class WebViewDemo extends Activity {

private WebView mWebView;

private Button mButton;

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

setupViews();

}

//初始化

private void setupViews() {

mWebView = (WebView) findViewById(R.id.webview);

WebSettings mWebSettings = mWebView.getSettings();

//加上这句话才能使用javascript方法

mWebSettings.setJavaScriptEnabled(true);

//增加接口方法,让html页面调用

mWebView.addJavascriptInterface(new Object() {

//这里我定义了一个打开地图应用的方法

public void startMap() {

Intent mIntent = new Intent();

ComponentName component = new ComponentName(

"com.google.android.apps.maps",

"com.google.android.maps.MapsActivity");

mIntent.setComponent(component);

startActivity(mIntent);

}

}, "demo");

//加载页面

mWebView.loadUrl("file:///android\_asset/demo.html");

mButton = (Button) findViewById(R.id.button);

//给button添加事件响应,执行JavaScript的fillContent()方法

mButton.setOnClickListener(new Button.OnClickListener() {

public void onClick(View v) {

mWebView.loadUrl("javascript:fillContent()");

}

});

}

}

**其他例子如下：（activity中获取js界面输入框的值）**

Jsp中的代码：

<body>

<form action="" method="post">

宝宝预产期：<br>

<select id="shijian" name="date">

<option value="2006">2006</option>

<option value="2007">2007</option>

<option value="2008">2008</option>

<option value="2009">2009</option>

<option value="2010">2010</option>

</select><br>

常用邮箱号：

<input id="email" type="text" name="emailID" />

<br>

宝宝昵称：

<input id="name" type="text" name="username" />

<br>

宝宝性别：<br>

<input id="men" type="radio" name="sex" value="men"/>男

<input id="women" type="radio" name="sex" value="women"/>女

<br>

<input type="submit" value="注册"

onclick="f()"/>

<input type="button" value="取消" />

</form>

</body>

<script type="text/JavaScript" language="javascript">

function f(){

var email = document.getElementById('email').value;

var name = document.getElementById('name').value;

var date = document.getElementById('shijian').value;

if(document.getElementById('men').checked && !document.getElementById('women').checked){

var sex = document.getElementById('men').value;

}else if(!document.getElementById('men').checked && document.getElementById('women').checked){

var sex = document.getElementById('women').value;

}

window.demo.clickOnAndroid(date+"|"+email+"|"+name+"|"+sex);

}

</script>

Manifest.xml中的代码：

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

<manifest xmlns:android=*"http://schemas.android.com/apk/res/android"*

package=*"com.ruixin.login"* android:versionCode=*"1"* android:versionName=*"1.0"*>

<application android:icon=*"@drawable/icon"* android:label=*"@string/app\_name"*>

<activity android:name=*".TestWebviewDemo"* android:label=*"@string/app\_name"*>

<intent-filter>

<action android:name=*"android.intent.action.MAIN"* />

<category android:name=*"android.intent.category.LAUNCHER"* />

</intent-filter>

<intent-filter>

<data android:mimeType=*"vnd.android.cursor.dir/vnd.ruixin.login"* />

</intent-filter>

<intent-filter>

<data android:mimeType=*"vnd.android.cursor.item/vnd.ruixin.login"* />

</intent-filter>

</activity>

<provider android:name=*"MyProvider"* android:authorities=*"com.ruixin.login"* />

</application>

<uses-sdk android:minSdkVersion=*"8"* />

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

<uses-permission android:name=*"android.permission.READ\_CONTACTS"*></uses-permission>

</manifest>

布局文件中的代码：

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

<LinearLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:orientation=*"vertical"* android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"*>

<WebView android:id=*"@+id/WebView01"* android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"*></WebView>

</LinearLayout>

TestWebviewDemo中的代码：

**public** **class** TestWebviewDemo **extends** Activity {

/\*\* Called when the activity is first created. \*/

**private** **static** **final** String *TAG* = "TestWebviewDemo";

**private** WebView mWebView;

**private** Handler mHandler = **new** Handler();

**private** **int** mDensity;

**public** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*main*);

mWebView = (WebView) findViewById(R.id.*WebView01*);

mWebView.getSettings().setAllowFileAccess(**true**);// 设置允许访问文件数据

mWebView.getSettings().setBuiltInZoomControls(**true**);// 设置支持缩放

mWebView.getSettings().setSavePassword(**false**); // 设置是否保存密码

// 设置支持JavaScript脚本

mWebView.getSettings().setJavaScriptEnabled(**true**);

// 设置支持各种不同的设备

mWebView

.getSettings()

.setUserAgentString(

"Mozilla/5.0 (iPad; U; CPU OS 3\_2 like Mac OS X; en-us) AppleWebKit/531.21.10 (KHTML, like Gecko) Version/4.0.4 Mobile/7B334b Safari/531.21.10");

// 通过这个设置来执行加载webview网页时所要执行的一些方法

mWebView.setWebViewClient(**new** WebViewClient() {

// 新开页面时用自己定义的webview来显示，不用系统自带的浏览器来显示

**public** **boolean** shouldOverrideUrlLoading(WebView view, String url) {

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

// 当有新连接时使用当前的webview进行显示

view.loadUrl(url);

**return** **super**.shouldOverrideUrlLoading(view, url);

}

// 开始加载网页时要做的工作

**public** **void** onPageStarted(WebView view, String url, Bitmap favicon) {

**super**.onPageStarted(view, url, favicon);

}

//加载完成时要做的工作

**public** **void** onPageFinished(WebView view, String url) {

**super**.onPageFinished(view, url);

}

// 加载错误时要做的工作

**public** **void** onReceivedError(WebView view, **int** errorCode,

String description, String failingUrl) {

Log.*d*(*TAG*, "error=" + description);

Toast.*makeText*(TestWebviewDemo.**this**,

errorCode + "/" + description, Toast.*LENGTH\_LONG*)

.show();

}

});

// 处理网页中的一些对话框信息（提示对话框，带选择的对话框，带输入的对话框）

mWebView.setWebChromeClient(**new** WebChromeClient() {

// 对话框

**public** **boolean** onJsAlert(WebView view, String url, String message,

**final** JsResult result) {

// 构建一个Builder来显示网页中的alert对话框

Builder builder = **new** Builder(TestWebviewDemo.**this**);

builder.setTitle("提示对话框");

builder.setMessage(message);

builder.setPositiveButton(android.R.string.*ok*,

**new** AlertDialog.OnClickListener() {

@Override

**public** **void** onClick(DialogInterface dialog,

**int** which) {

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

result.confirm();

}

});

builder.setCancelable(**false**);

builder.create();

builder.show();

**return** **true**;

}

// 带按钮的对话框

**public** **boolean** onJsConfirm(WebView view, String url,

String message, **final** JsResult result) {

Builder builder = **new** Builder(TestWebviewDemo.**this**);

builder.setTitle("带选择的对话框");

builder.setMessage(message);

builder.setPositiveButton(android.R.string.*ok*,

**new** AlertDialog.OnClickListener() {

@Override

**public** **void** onClick(DialogInterface dialog,

**int** which) {

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

result.confirm();

}

});

builder.setNeutralButton(android.R.string.*cancel*,

**new** AlertDialog.OnClickListener() {

@Override

**public** **void** onClick(DialogInterface dialog,

**int** which) {

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

result.cancel();

}

});

builder.setCancelable(**false**);

builder.create();

builder.show();

**return** **true**;

}

// 带输入框的对话框

**public** **boolean** onJsPrompt(WebView view, String url, String message,

String defaultValue, **final** JsPromptResult result) {

LayoutInflater inflater = LayoutInflater

.*from*(TestWebviewDemo.**this**);

**final** View v = inflater.inflate(R.layout.*prom\_dialog*, **null**);

// 设置 TextView对应网页中的提示信息

((TextView) v.findViewById(R.id.*TextView\_PROM*))

.setText(message);

// 设置EditText对应网页中的输入框

((EditText) v.findViewById(R.id.*EditText\_PROM*))

.setText(defaultValue);

Builder builder = **new** Builder(TestWebviewDemo.**this**);

builder.setTitle("带输入的对话框");

builder.setView(v);

builder.setPositiveButton(android.R.string.*ok*,

**new** AlertDialog.OnClickListener() {

@Override

**public** **void** onClick(DialogInterface dialog,

**int** which) {

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

String value = ((EditText) v

.findViewById(R.id.*EditText\_PROM*))

.getText().toString();

result.confirm(value);

}

});

builder.setNegativeButton(android.R.string.*cancel*,

**new** AlertDialog.OnClickListener() {

@Override

**public** **void** onClick(DialogInterface dialog,

**int** which) {

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

result.cancel();

}

});

builder

.setOnCancelListener(**new** DialogInterface.OnCancelListener() {

@Override

**public** **void** onCancel(DialogInterface dialog) {

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

result.cancel();

}

});

builder.create();

builder.show();

**return** **true**;

}

// 设置网页加载的进度条

**public** **void** onProgressChanged(WebView view, **int** newProgress) {

TestWebviewDemo.**this**.getWindow().setFeatureInt(

Window.*FEATURE\_PROGRESS*, newProgress \* 100);

**super**.onProgressChanged(view, newProgress);

}

// 设置应用程序的标题

**public** **void** onReceivedTitle(WebView view, String title) {

TestWebviewDemo.**this**.setTitle(title);

**super**.onReceivedTitle(view, title);

}

});

// 与网页进行交互的addJavascriptInterface()的方法

mWebView.addJavascriptInterface(**new** Object() {

**public** **void** clickOnAndroid(**final** String str) {

mHandler.post(**new** Runnable() {

@Override

**public** **void** run() {

StringTokenizer stringTokenizer = **new** StringTokenizer(

str, "|");

*date* = stringTokenizer.nextToken();

*email* = stringTokenizer.nextToken();

*username* = stringTokenizer.nextToken();

*sex* = stringTokenizer.nextToken();

//将数据出入数据库

dBlite1.add(*email*,*username*,*date*,*sex*);

// 用contentResolver访问存入到contentprovider中的数据

contentResolver = TestWebviewDemo.**this**

.getContentResolver();

Log.*d*("++++", RuiXin.*CONTENT\_URI*.toString());

Cursor cursor = contentResolver.query(

RuiXin.*CONTENT\_URI*, **new** String[] {

RuiXin.*EMAIL*, RuiXin.*USERNAME*,

RuiXin.*DATE*,RuiXin.*SEX* }, **null**, **null**, **null**);

Log.*d*("@@@----", cursor.toString());

// startManagingCursor(cursor);

**while** (cursor.moveToNext()) {

Toast.*makeText*(

TestWebviewDemo.**this**,

cursor.getString(cursor.getColumnIndex(RuiXin.*EMAIL*))

+ " "

+ cursor.getString(cursor

.getColumnIndex(RuiXin.*USERNAME*))

+ " "

+ cursor.getString(cursor

.getColumnIndex(RuiXin.*DATE*))

+ " "

+ cursor.getString(cursor

.getColumnIndex(RuiXin.*SEX*)),

Toast.*LENGTH\_SHORT*).show();

}

startManagingCursor(cursor); //关闭游标

}

});

}

}, "demo");

**if** (mDensity == 240) { // 可以让不同的density的情况下，可以让页面进行适配

mWebView.getSettings().setDefaultZoom(ZoomDensity.*FAR*);

} **else** **if** (mDensity == 160) {

mWebView.getSettings().setDefaultZoom(ZoomDensity.*MEDIUM*);

} **else** {

mWebView.getSettings().setDefaultZoom(ZoomDensity.*CLOSE*);

}

String strUrl = "http://10.0.2.2:8080/WebTest3";

mWebView.loadUrl(strUrl);

}

//处理返回的事件，（后退进入前一个界面而不是全部退出）

@Override

**public** **boolean** onKeyDown(**int** keyCode, KeyEvent event) {

**if** (keyCode == KeyEvent.*KEYCODE\_BACK* && mWebView.canGoBack()) {

mWebView.goBack();// 返回前一个页面

**return** **true**;

}

**return** **super**.onKeyDown(keyCode, event);

}