# 复习

## Bootstrap

是个css框架

自适应

.Col-xs-12:100%

.col-sm-6:50%

.col-md-4:33%

<div class=col-xs-12 col-sm-6 col-md-4>

<div class=col-xs-12 col-sm-6 col-md-4>

<div class=col-xs-12 col-sm-6 col-md-4>

## Vue

是个前端框架，代替jquery

用jquery可以添加标签，删除标签 .ajax()

v-for=”item in items” axios

vue是个生态，跟很多框架一块用，像springcloud

## 微信小程序

小应用：微信小程序

中应用：小程序+网站

大应用：小程序+网站+app

Wxml

Wxss

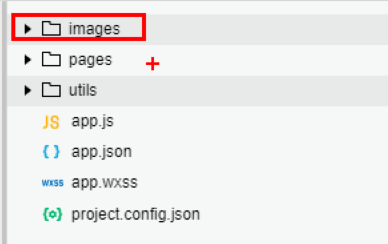
Js ws:for wx:for-item {{}}

# 拼多商城小程序端

## 首页

创建项目Pdweixin

拷贝图片，图片在\亿发,移动端课前资料\weixin\code\pdWeixinApp\images文件夹中。



修改index.wxml

<view class="container">

<image src='../../images/first.png'></image>

</view>

## 登录页

界面需求如图-50.

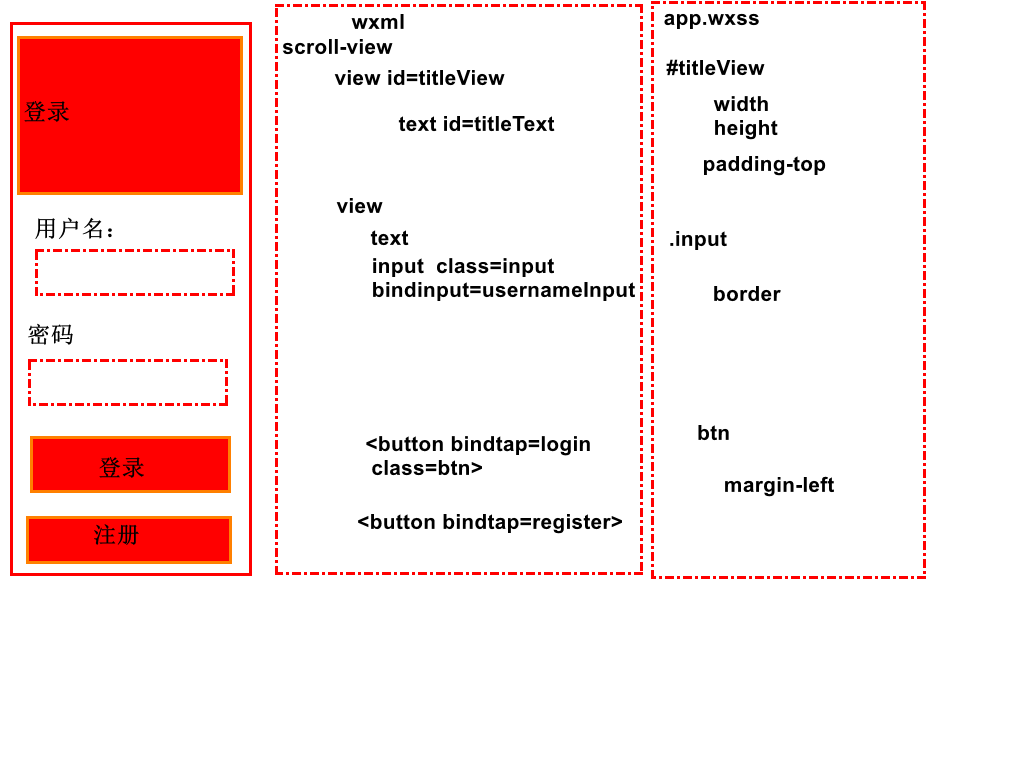


图- 50

Login.wxml

<scroll-view>

<view class="titleView">

<text class='titleText'>登录</text>

</view>

<view>

<text>用户名:</text>

<input class='input' bindinput='usernameInput'></input>

</view>

<view>

<text>密码:</text>

<input class='input' bindinput='passwordInput'></input>

</view>

<view>

<button class="btn" bindtap='login'>登录</button>

<button class="btn" bindtap='register'>注册</button>

</view>

</scroll-view>

App.wxss中定义的样式是全局样式，所有wxml文件都可以使用。

App.wxss

.titleView{

width: 100%;

height: 100rpx;

background-color:#FF670c;

padding-left: 48rpx;

padding-top: 26rpx;

}

.titleText{

font-size: 48rpx;

}

.input{

border: 1px black solid;

}

.btn{

margin-top: 24rpx;

width: 80%;

}

Index.js

onReady:function(){

setTimeout(function(){

wx.navigateTo({

url: '../login/login',

})

},3000);

}

## 注册页

Register.wxml

<!--pages/register/register.wxml-->

<scroll-view>

<view class="titleView">

<text class='titleText'>注册</text>

</view>

<view>

<text>用户名:</text>

<input class='input' bindinput='usernameInput' value='a001'></input>

</view>

<view>

<text>密码:</text>

<input class='input' bindinput='passwordInput' value='123456'></input>

</view>

<view>

<text>确认密码:</text>

<input class='input' bindinput='confirmPasswordInput' value='123456'></input>

</view>

<view>

<text>mail:</text>

<input class='input' bindinput='mailInput' value='a001@qq.com'></input>

</view>

<view>

<text>mobile:</text>

<input class='input' bindinput='mobileInput' value='13691482001'></input>

</view>

<view>

<button class="btn" bindtap='register'>注册</button>

</view>

</scroll-view>

Login.js中增加代码跳转到注册界面

register:function()

{

wx.navigateTo({

url: '../register/register',

})

}

## 联网注册

### 全局变量

App.js中的globalData用来定义全局变量。

globalData: {

userInfo: null,

serverIp:"http://192.168.1.106/"

}

Register.js

//调用getApp函数，放在page外面

var app=getApp();

page({

onReady: function () {

console.log(app.globalData.serverIp);

}

}

)

### 取值，联网

data: {

username:"",

password:"",

confirmPassword:"",

email:"",

phone:""

},

usernameInput: function (event) {

var username = event.detail.value;

this.setData({

username: username,

})

},

passwordInput: function (event) {

var password = event.detail.value;

this.setData({

password: password,

})

},

confirmPasswordInput: function (event) {

var confirmPassword = event.detail.value;

this.setData({

confirmPassword: confirmPassword,

})

},

mailInput: function (event) {

var mail = event.detail.value;

this.setData({

mail: mail,

})

},

mobileInput: function (event) {

var mobile = event.detail.value;

this.setData({

mobile: mobile,

})

},

register: function () {

var username = this.data.username;

var password = this.data.password;

var confirmPassword = this.data.confirmPassword;

var mail = this.data.mail;

var mobile = this.data.mobile;

var url = app.globalData.serverIp + "user/register.html?username=" + username + "&password=" + password + "&email=" + mail + "&phone=" + mobile;

wx.request({

url: url,

success: function (response) {

console.log(response);

var serverReturnData = response.data;

var status = serverReturnData.status;

if (status == 200) {

wx.showToast({

title: '成功',

});

} else {

wx.showToast({

title: '失败',

});

}

},

fail: function (e) {

console.log(e);

}

})

},

## 联网登录

var app=getApp();

data: {

username: "",

password: ""

},

usernameInput: function(event) {

var username = event.detail.value;

this.setData({

username: username,

})

},

passwordInput: function(event) {

var password = event.detail.value;

this.setData({

password: password,

})

},

login: function() {

var username = this.data.username;

var password = this.data.password;

var url = app.globalData.serverIp + "user/login.html?username=" + username + "&password=" + password;

wx.request({

url: url,

success: function(response) {

console.log(response);

var serverReturnData = response.data;

var status = serverReturnData.status;

if (status == 200) {

wx.showToast({

title: '成功',

});

} else {

wx.showToast({

title: '失败',

});

}

},

fail: function(e) {

console.log(e);

}

})

},

## 商城首页

创建目录storeIndex

创建page storeIndex

修改storeIndex.xml

<web-view src="http://192.168.1.106/mobileIndex.html"></web-view>

登录成功后跳转到storeIndex

修改login.js

if (status == 200) {

wx.showToast({

title: '成功',

});

wx.navigateTo({

url: '../storeIndex/storeIndex',

})

## tabBar

首面，消息，我的显示效果如图-51

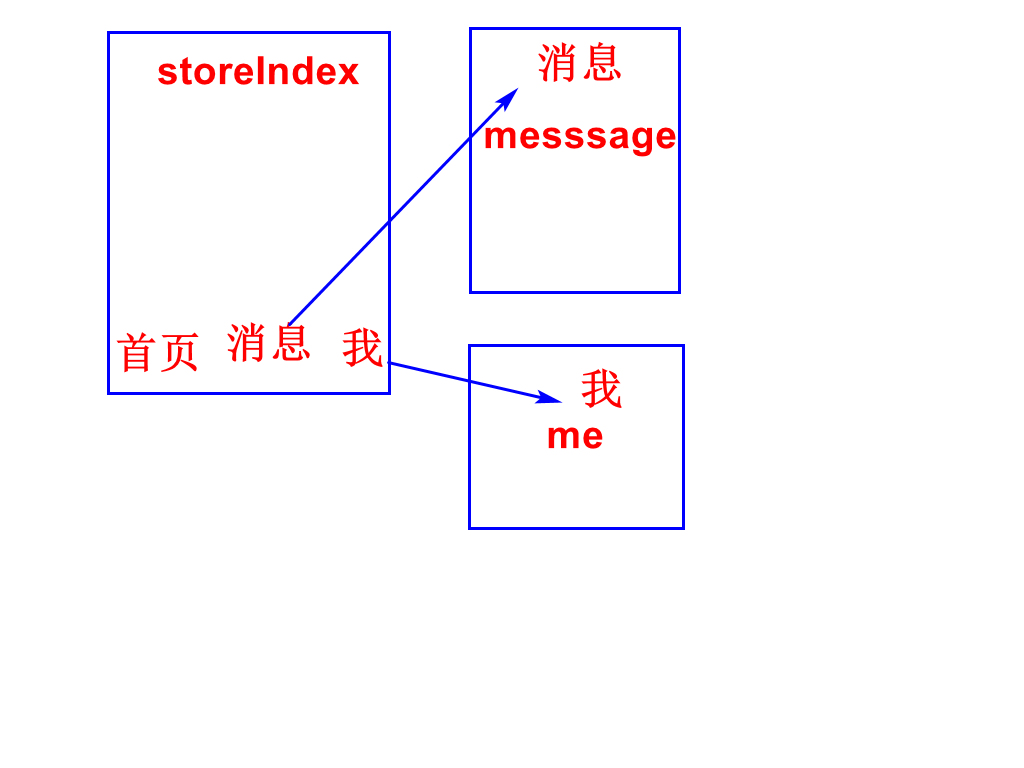


图- 51

### 创建消息

创建目录message

创建page message

### 创建我

创建目录me

创建page me

### app.json中添加tabbar

"tabBar": {

"color": "#000000",

"selectedColor": "#FF670c",

"list": [

{

"pagePath": "pages/storeIndex/storeIndex",

"text": "首页",

"iconPath": "images/ic\_tab\_index\_normal.png",

"selectedIconPath": "images/ic\_tab\_index\_selected.png"

},

{

"pagePath": "pages/message/message",

"text": "消息",

"iconPath": "images/ic\_tab\_msg\_normal.png",

"selectedIconPath": "images/ic\_tab\_msg\_selected.png"

},

{

"pagePath": "pages/me/me",

"text": "我",

"iconPath": "images/ic\_tab\_me\_normal.png",

"selectedIconPath": "images/ic\_tab\_me\_selected.png"

}

]

}

### 修改login.js

跳到带有tabBar的界面必须用wx.switchTab

// wx.navigateTo({

// url: '../storeIndex/storeIndex',

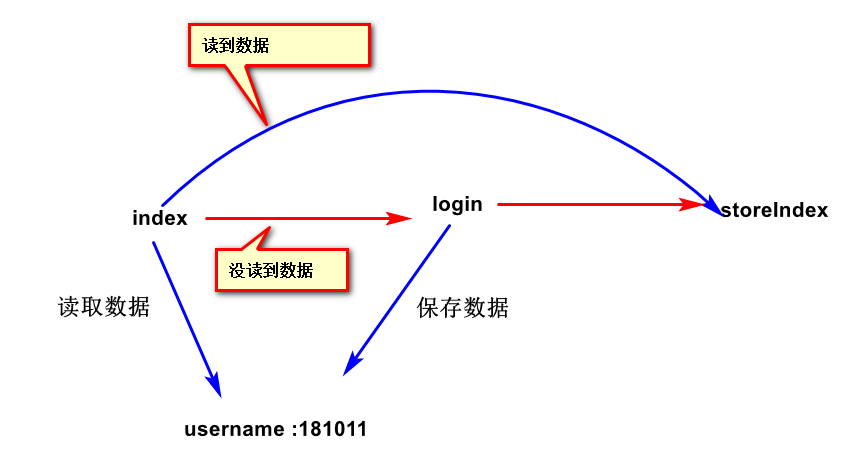
// })

wx.switchTab({

url: '../storeIndex/storeIndex',

})

## 保存数据



登录成功后添加

wx.setStorageSync("user", username)

在storage中查看保存的数据, 如图-52

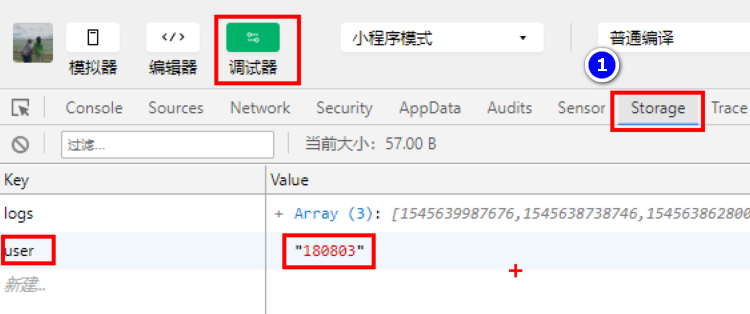


图- 52

Index.js中判断有没有用户名，如果有直接跳转到storeIndex.

onReady: function() {

var username = wx.getStorageSync("user");

console.log("username=" + username)

setTimeout(function() {

if (username) {

wx.switchTab({

url: '../storeIndex/storeIndex',

})

} else {

wx.navigateTo({

url: '../login/login',

})

}

}, 3000);

},

## 上线

开发管理中提交审核，如图-53

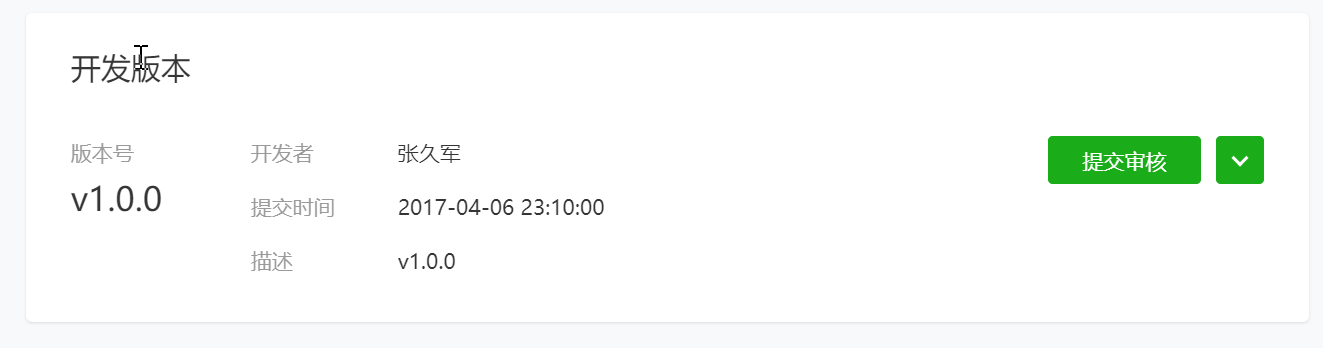


图- 53

确认提交审核中点“下一步”， 如图-54



图- 54

进入审核中，如图-55

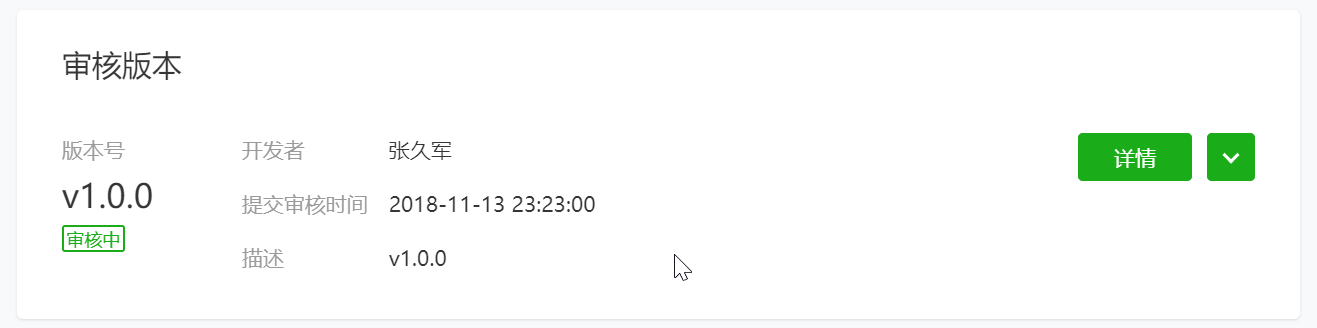


图- 55

上线成功后，成为线上版本如图-56

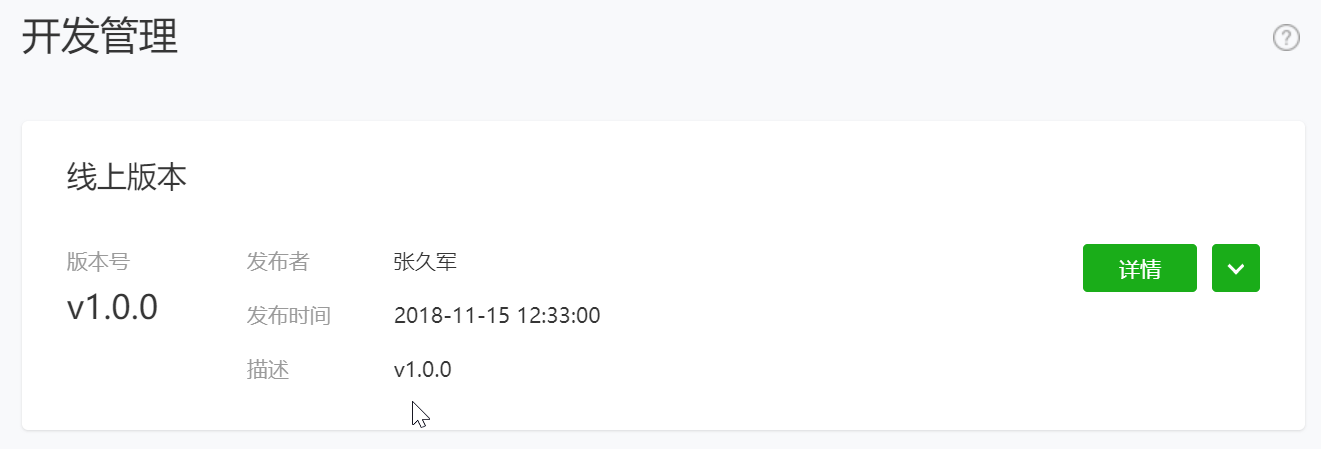


图- 56

## 官方demo

创建项目指向官方demo,运行demo,查看微信小程序的功能

# Android

Android是新的开发平台，用java语言，做出来的程序运行在android手机上。叫app

Java是一种开发平台，程序运行在jvm中

J2ee是一种开发平台，程序运行在tomcat

前端js,html,css,jquery,ajax,bootstrap,vue是一种开发平台，程序运行在浏览器中

微信小程序是一种开发平台，程序运行在微信

## c/s和b/s的区别

c/s client/server

微信，QQ,用户要安装client

优点：功能强大

缺点：客户端要安装

b/s browser/server

用户不用安装客户端，客户端是浏览器

优点：客户端不用安装

缺点：功能简单，界面加载慢，简单，使用html5 canvas把界面复杂。

做软件运行在电脑，推荐用b/s

做软件运行在手机上，叫app,推荐用c/s

电脑上和手机上的软件推荐使用c/s+b/s 叫混合开发模式，也叫混编，hybrid

## 运行AndroidStudio

参见“AndroidStudio开发环境搭建步骤详解.doc”

### 启动夜神模拟器

### 查看夜神安装目录如图-57



图- 57

### 在dos下执行nox\_adb.exe

在图-57中点“打开文件所在的位置”进入资源管理器，输入cmd,打开dos,如图-58. 执行的命令是

nox\_adb.exe connect 127.0.0.1:62001

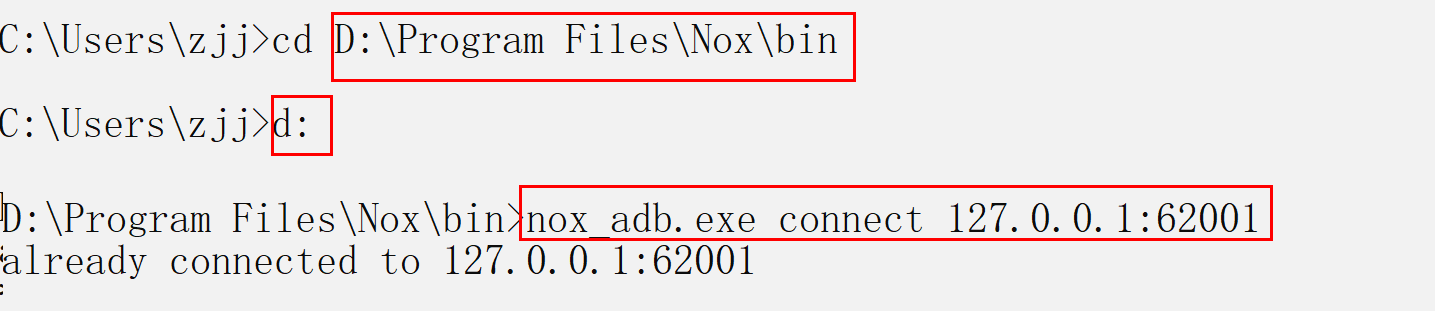


图- 58

### 运行程序

点下图中绿箭头，会出现”Samsung Nexus”, 如图-59.如果不出现，在dos窗口中再执行一次” nox\_adb.exe connect 127.0.0.1:62001”

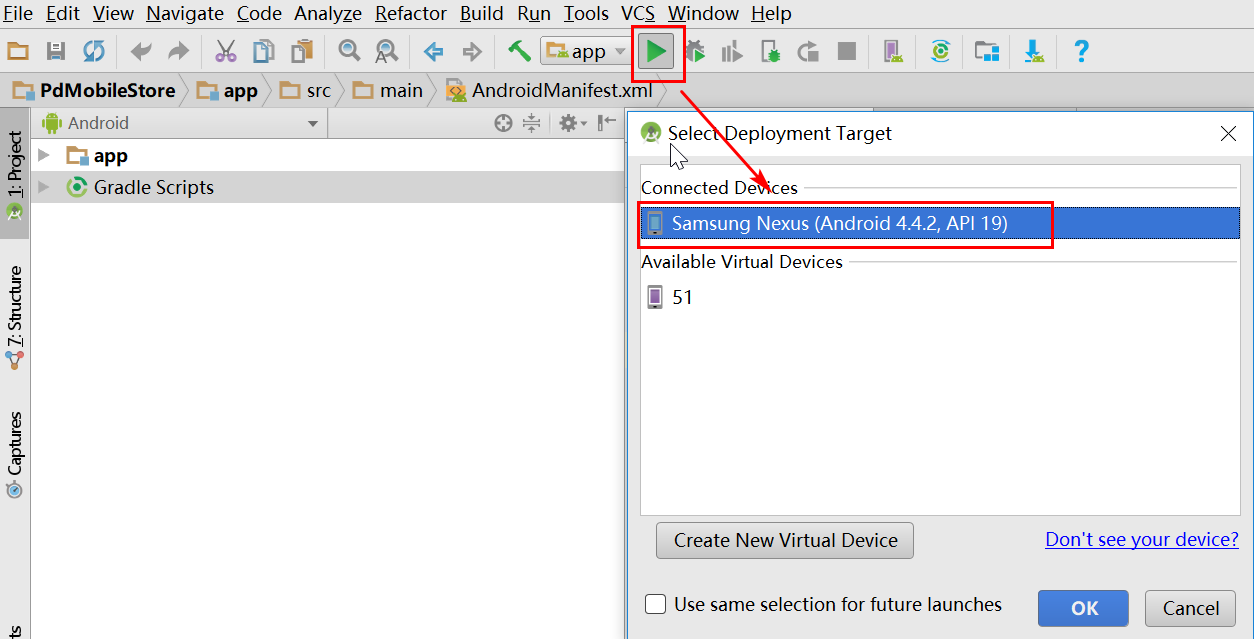


图- 59

点“Proceed without Instant Run”不用闪电安装功能，如图-60.

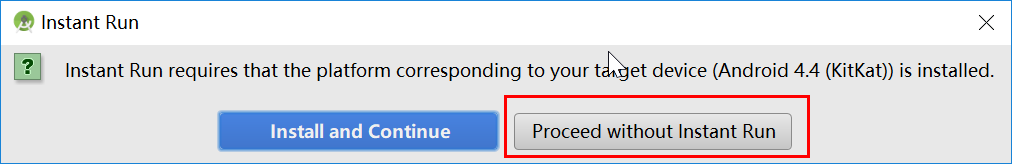


图- 60

**有的电脑报错**

**“could not find com.google.android.gms:play-service”**

**解决方法是androidStudio 🡪build🡪rebuild**

**Baidu上查找解决方案**

****

### 设置夜神模拟器屏幕大小，如图-61



图- 61

## Android入门案例

### 工具使用

单击project可以显示或隐藏项目如图-62

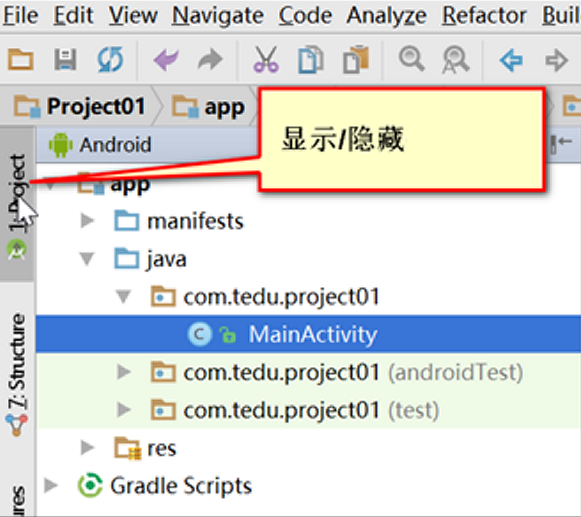


图- 62

点图-63中的绿箭头运行程序。

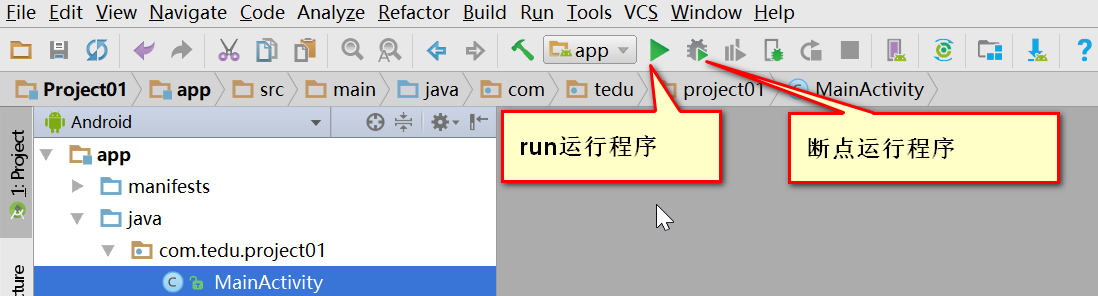


图- 63

单击图-64中的text能打开xml编辑窗口。

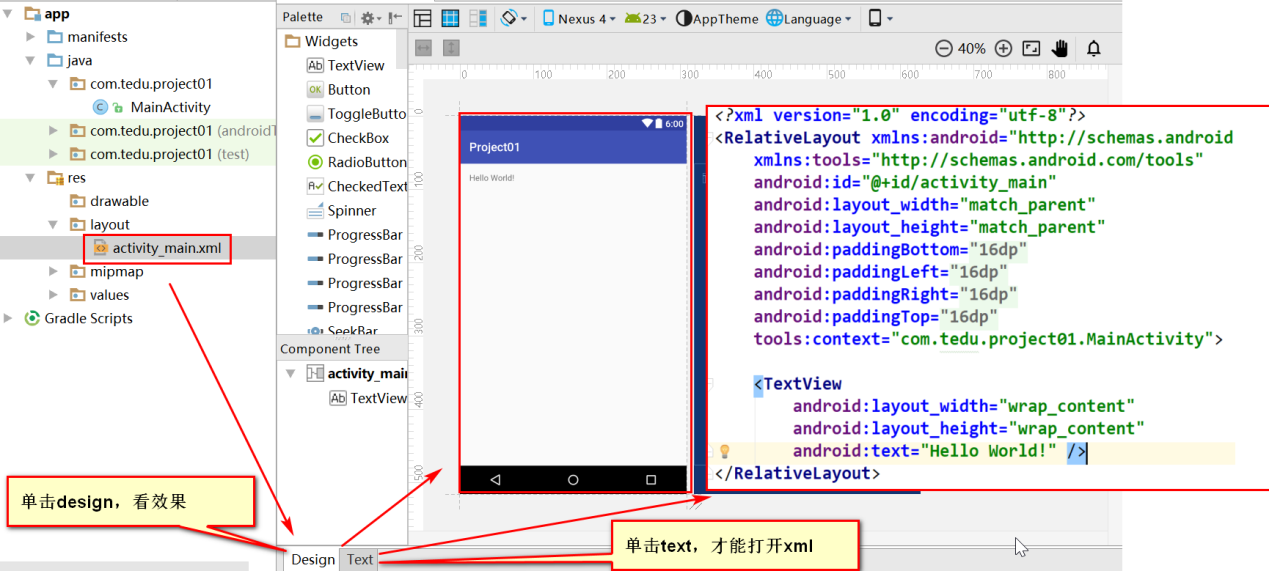


图- 64

如果没有导包出错，可以按alt+enter弹出提示，如图-65.

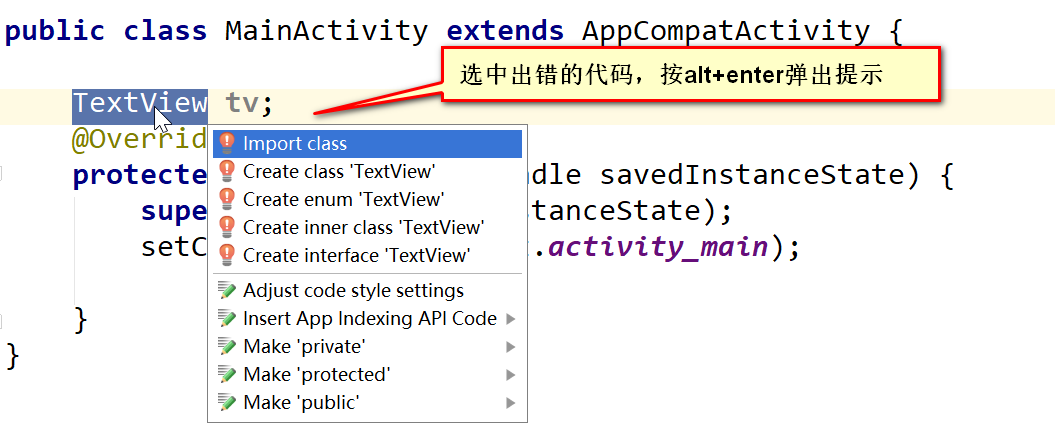


图- 65

### Android项目结构

创建项目project01，各个文件夹的作用如图-66所示。

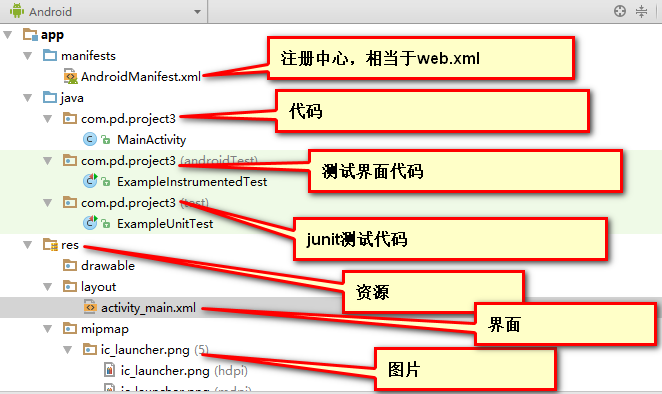


图- 66

### 程序是如何运行

R类对应res文件夹，如图-67所示。

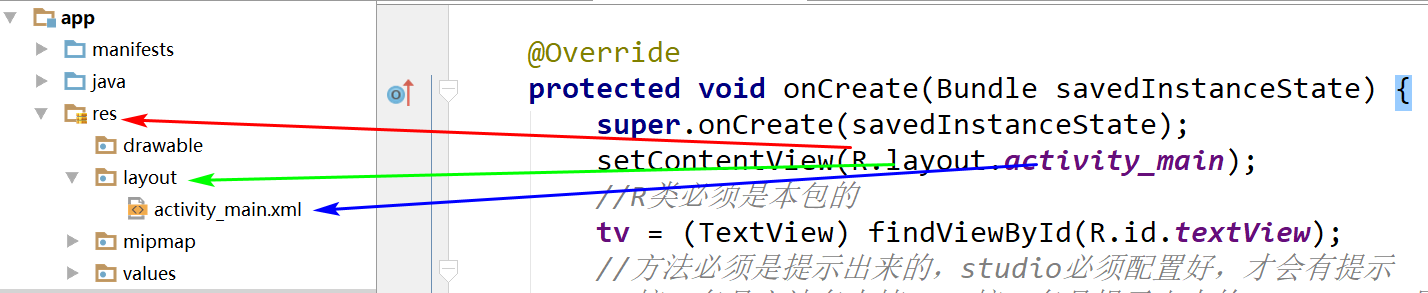


图- 67

界面显示过程如图-68所示。

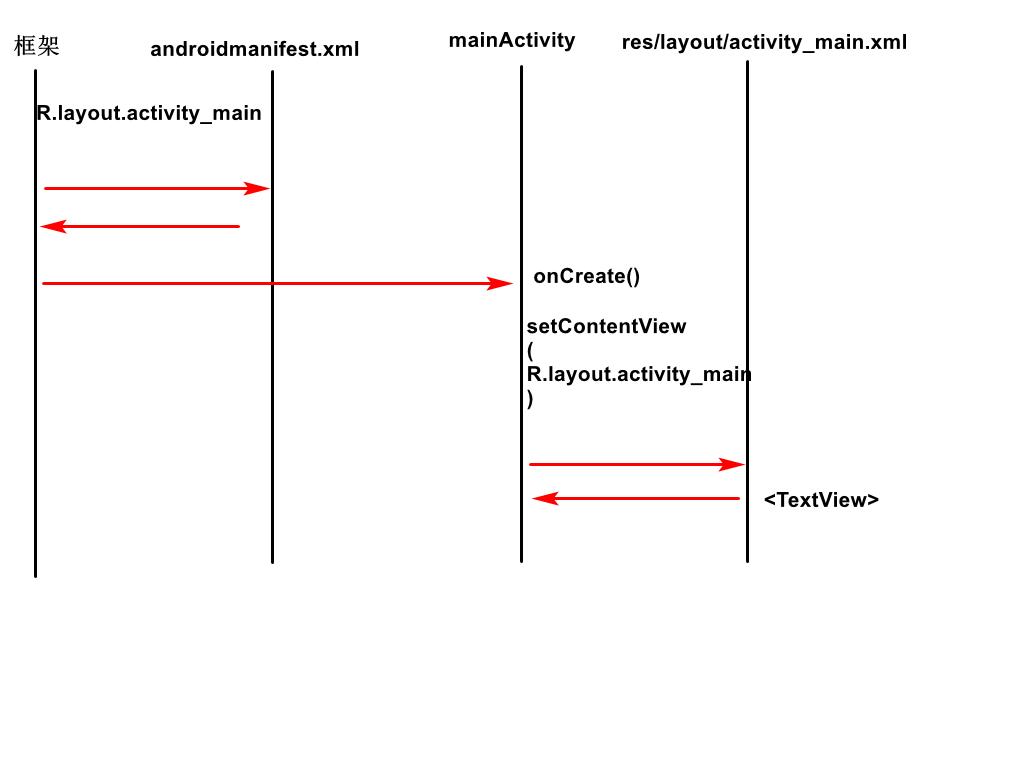


图- 68

### 修改界面

在android视图中修改activity\_main.xml

1. 修改文本，字号
2. 设置id

*<!--match\_parent是父容器的大小-->  
<!--学习match\_parent,wrap\_content的区别 ,design,text的作用-->*

<!--id属性值中必须有@+id/  
在R类中生成内部类id  
在R.id类中生成常量textView  
R.id.textView=1000000;  
在activity中就可以通过R.id.textView  
找到控件  
-->

<**TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="#FF0000"  
 android:id="@+id/textView"  
 android:text="\n我的\n第一个\nandroid程序\n"** />

### 事件处理

修改MainActivity.java

**public class** MainActivity **extends** AppCompatActivity {  
 TextView **tv**;  
  
 *//创建时候执行 像js中window.onload=fun* @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 *//R类是系统自动生成  
 //加载布局文件res/layout/activity\_main.xml  
//layout是R类的内部类  
 //class R  
 // {class layout{  
 // int activity\_main  
 // }  
 // }* setContentView(R.layout.***activity\_main***);  
 *//找到textView  
 //findViewById返回的View  
 //TextView是view的子类  
 //用alt+enter有提示* **tv** = (TextView) findViewById  
 (R.id.***textView***);  
 *//设置事件监听  
 //js 中btn.onClick=fun  
 //发生单击事件 操作系统先知道，  
 // 把事件告诉android框架  
 //框架会调接口* MyListener myListener = **new** MyListener();  
 **tv**.setOnClickListener(myListener);  
  
 }  
  
 *//内部类* **class** MyListener **implements** View.OnClickListener {  
  
 @Override  
 **public void** onClick(View v) {  
*//在屏幕上显示* Toast.*makeText*(MainActivity.**this**,  
 **"android"**,  
 Toast.***LENGTH\_LONG***)  
 .show();  
 }  
 }  
  
}

程序运行效果如图-69所示。

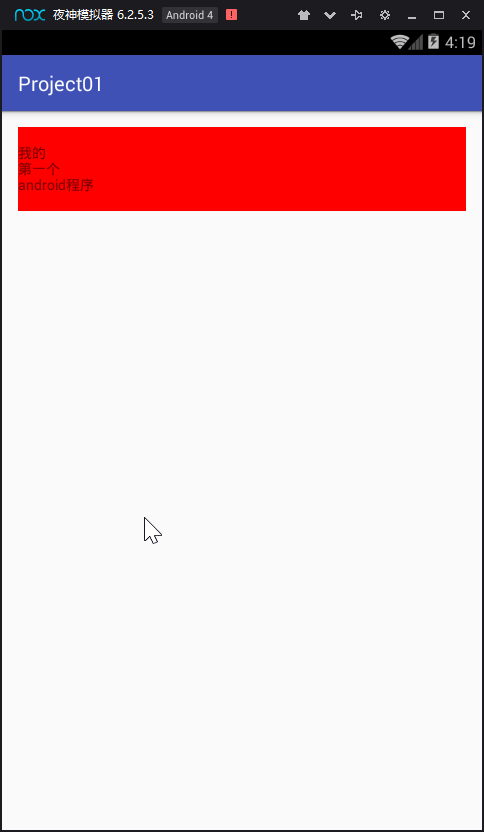


图- 69

### 从网站上下载apk

1. 到\app\build\outputs\apk文件夹下找到apk，拷贝到\pd-web\src\main\webapp\文件夹中。

在androidStudio中将视图切换成project,找到apk文件，如图-70所示。

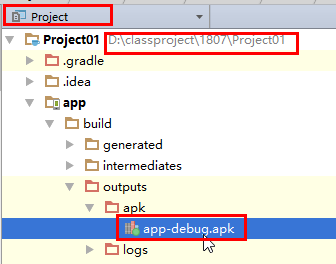


图- 70

1. 制作apk.html

<body>

<h1>

<a href=*"app-debug.apk"*>apk</a>

</h1>

</body>

1. 删除模拟器中的程序，如图-71所示。



图- 71

1. 启动模拟器中的浏览器，如图-72所示。



图- 72

1. 下载apk，如图-73所示。

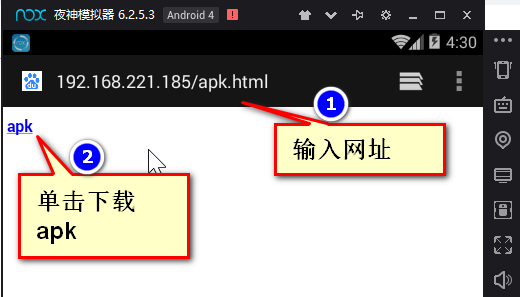


图- 73

显示安装中，如图-74所示。



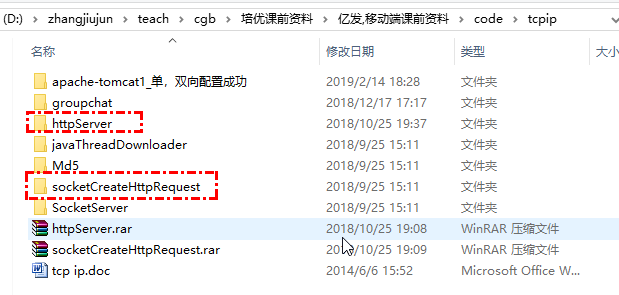
图- 74

打开apk，如图-75所示。

  
图- 75

# 问题

## 手写tomcat和浏览器



## 支付

## Android书

**第一行代码 Android 第2版**

**第一行代码 Android 第2版**