

公众号申请

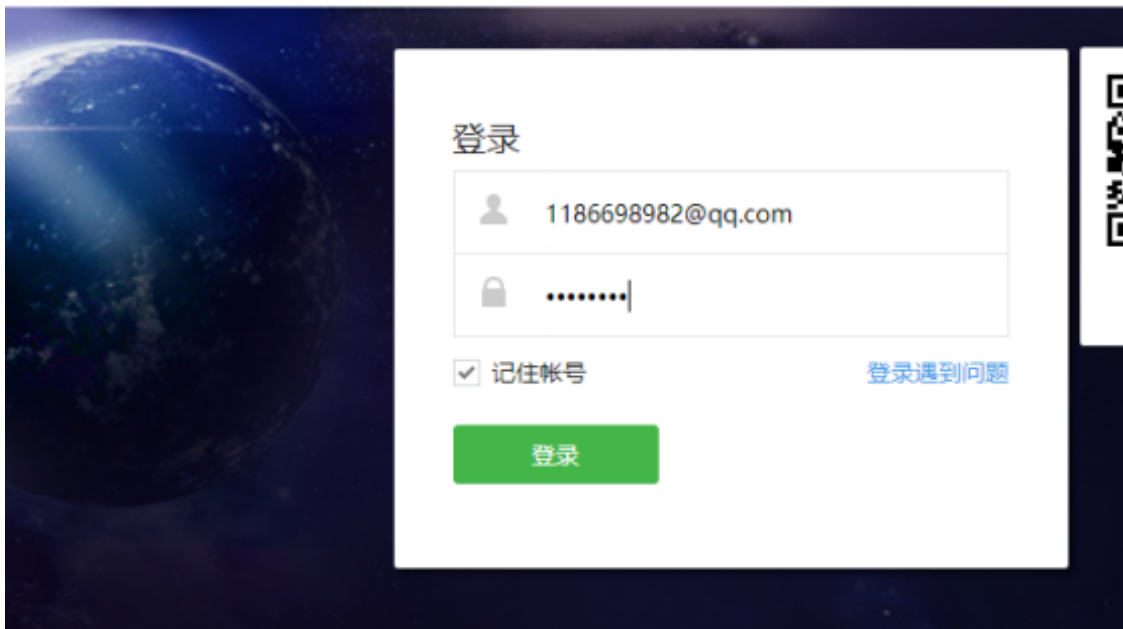
1. 注册小程序，填写法人代表

- 教程地址

<https://mp.weixin.qq.com/debug/wxadoc/introduction/#产品定位及功能介绍>

2. 新建完成小程序，获取Appid

第一次使用公众平台？[立即注册](#) | [腾讯客服](#)





3. 添加开发者

用户身份->成员管理



找模板

1. 网址 <http://www.getweapp.com/>

- 这个网址的接口全都不好用了，但是一些项目还是有原作者的github地址，这些项目可以运行

目录结构：

- imgs — 存放项目图片文件
- page — 存放项目页面文件
- utils — 存放格式化文件

开发环境：

微信web开发者工具 v0.11.122100

源码下载：

<https://github.com/getweapp/weapp-store>

感谢：

本项目原始版本由MMMsCheng提供：<https://github.com/MMMsCheng/wxapp>

2.让模板拼成的demo运行起来

- 把不同的项目中你需要的模块拼装到一起，运行起来，看效果

做成静态DEMO

1.先组合

- 某些不能显示的接口用静态数据替代，先让客户能看到一个大概的样式



2.修改和调整CSS 样式(下面列举两种常用样式参考)

- 商品两列显示

```
<!--xml中代码 -->
<view class="venues_box">
  <view class="venues_list">
    <block wx:for="{{venuesItems}}">
      <view class="venues_item" bindtap="goDetaile" data-index="{{index}}">
        <!-- <navigator url="../details/index?id={{item.id}}" -->
        <!-->
        <image
src="http://weixin.gowants.cn:8080/{{item.files}}"/>
```

```

        <!-- </navigator> -->
        <view class="venues_text">{{item.name}}</view>
    </view>
</block>
</view>
</view>

```

```

/* 两列商品 css样式*/
.venues_box {
    padding: 0 0rpx;
    width: 100%;
    text-align: center;
}

.venues_item {
    /* display: inline-block; */

    float: left;
    margin-left: 2%;
    width: 47%;

    margin-bottom: 20rpx;
}

.venues_item image{
    width: 336rpx;
    height: 189rpx;
}

.venues_text{
    text-align: center;
}

```

- 商品单列显示

```

<!--商品单列显示xml样式 ->
<view class="content">
    <view wx:for="{{data}}" class="pic" bindtap="goDetaile" data-
index="{{index}}" wx:key="">
        <image src="http://weixin.gowants.cn:8080{{item.files}}"
mode="aspectFill" data-id="{{index}}"></image>
        <view class="des ellipsis">{{item.intro}}</view>
    </view>

```

```
</view>
```

```
/*商品单列显示css样式*/
.content {
  overflow: hidden;
  margin-top: -25rpx;
}

.pic {
  /* display: inline-block; */
  text-align: center;
  /* width: 329rpx; */
  width: 100%;
  /* margin: 0 0 10px 0; */

  overflow: hidden;
}

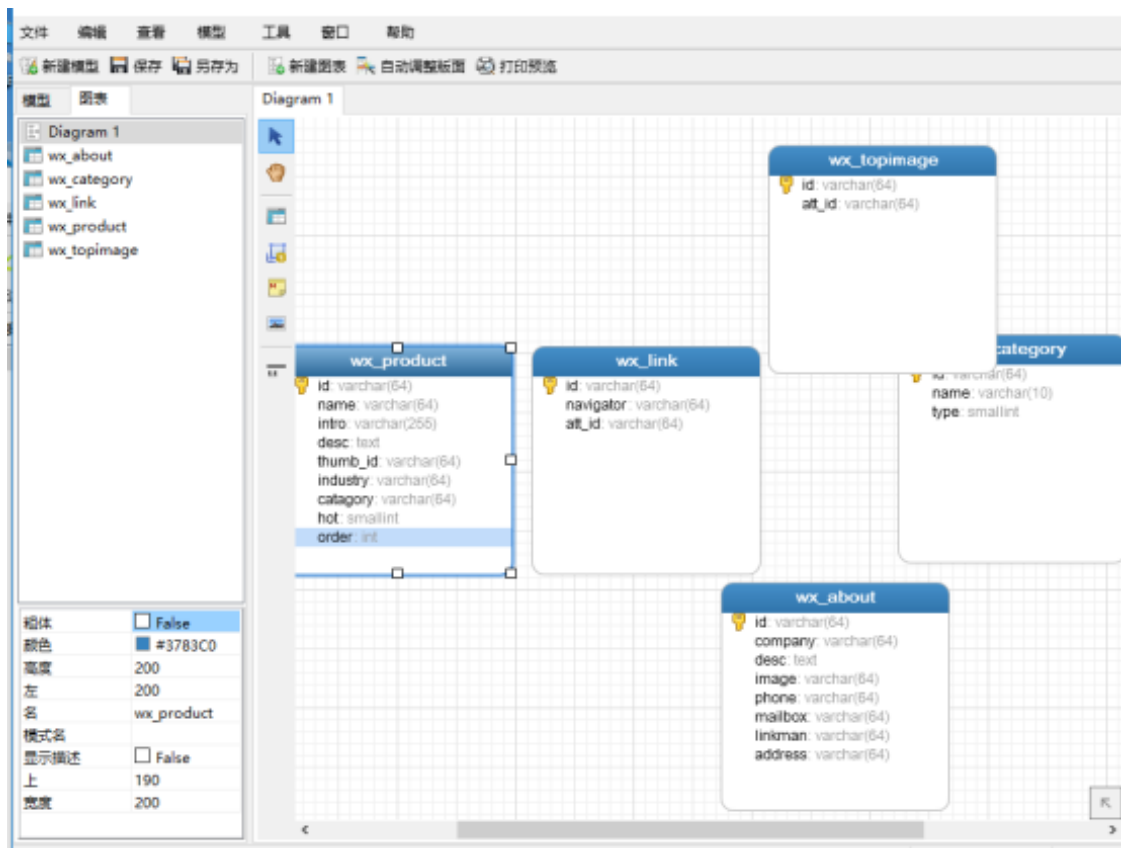
.pic image {
  /* width: 329rpx; */
  /* 这里改变图片的宽度，父元素修改会不居中 */
  width: 704rpx;
  height: 400rpx;
}
```

3.和客户定稿,给客户看

根据DEMO设计数据库

1.用navicat for mysql

- 用navicat自带的模型工具，建立数据库



2.jeesite框架每个表必须添加6个字段

wx_product @wx_boli (biolchim_com_cn) - 表

文件 编辑 窗口 帮助

新建 保存 另存为 添加栏位 插入栏位 删除栏位 主键 上移 下移

栏位	索引	外键	触发器	选项	注释	SQL 预览
名						
files3						varchar 64 0
industry						varchar 64 0
catagory						varchar 64 0
hot						smallint 2 0
order2						int 9 0
create_by						varchar 64 0
create_date						datetime 0 0
update_by						varchar 64 0
update_date						datetime 0 0
remarks						varchar 255 0
del_flag						char 1 0

默认: NULL

注释: 删除标记

字符集: utf8

排序规则: utf8_bin

键长度:

```
ALTER TABLE `jeesite`.`wx_about`  
ADD COLUMN `create_by` VARCHAR(64) NULL comment '创建人' AFTER  
`address`;  
  
ALTER TABLE `jeesite`.`wx_about`  
ADD COLUMN `create_date` datetime NULL comment '创建时间' AFTER  
`create_by`;  
  
ALTER TABLE `jeesite`.`wx_about`  
ADD COLUMN `update_by` VARCHAR(64) NULL comment '更新人' AFTER  
`create_date`;  
  
ALTER TABLE `jeesite`.`wx_about`  
ADD COLUMN `update_date` datetime NULL comment '更新时间' AFTER  
`update_by`;  
  
ALTER TABLE `jeesite`.`wx_about`  
ADD COLUMN `remarks` VARCHAR(255) NULL comment '备注信息' AFTER  
`update_date`;  
  
ALTER TABLE `jeesite`.`wx_about`  
ADD COLUMN `del_flag` CHAR(1) NULL comment '删除标记' AFTER `remarks`;
```

3.对着界面进行推演

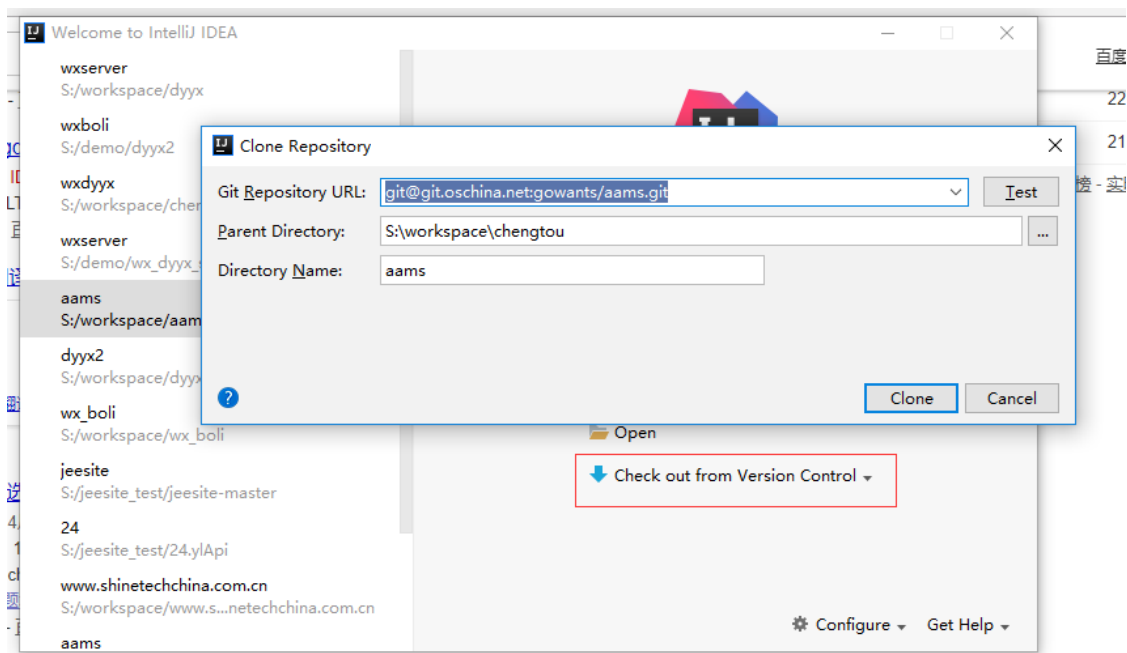
- 根据界面上的功能，修改或者添加数据库字段，字段长度，字段属性。富文本用text，是否属性用smallint 字段的备注一定要写好，后面框架会直接把字段名当成列明

后台开发

1.复制后台项目模板

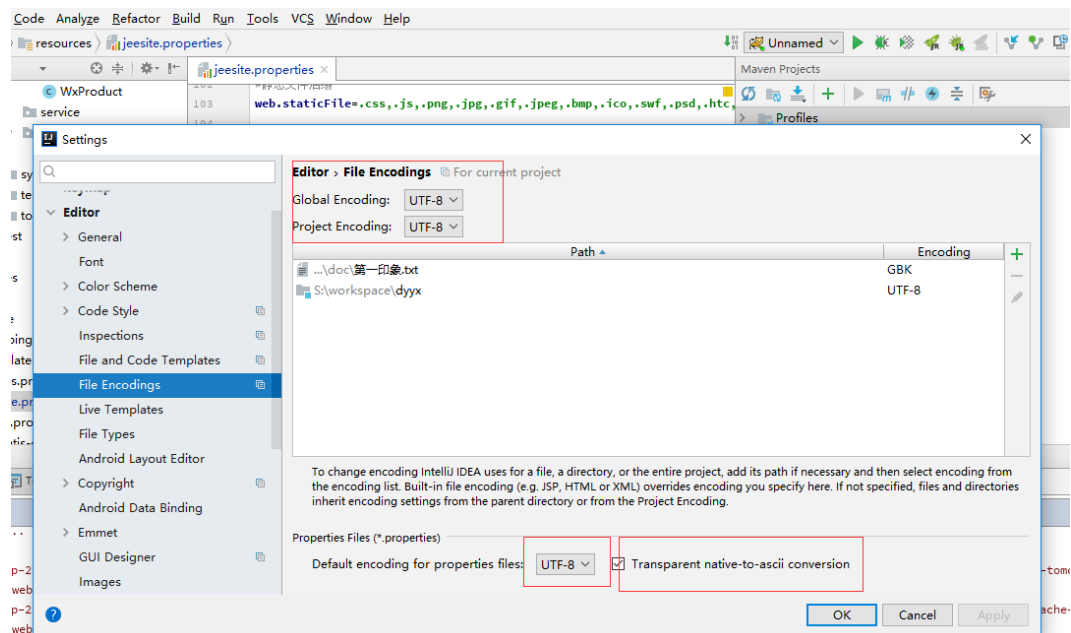
2.导入到IDEA开发工具

- 通过码云，复制项目的ssh地址clone 到本地
- IDEA file->Open
- 或者直接check out 从git检出

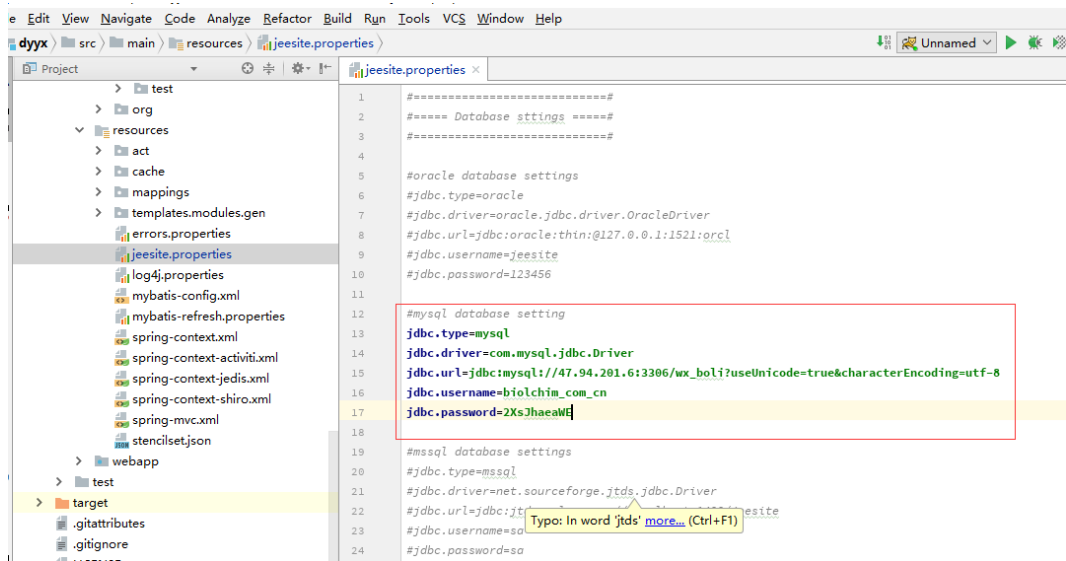


3.配置 tomcat 数据库链接 encoding默认utf8

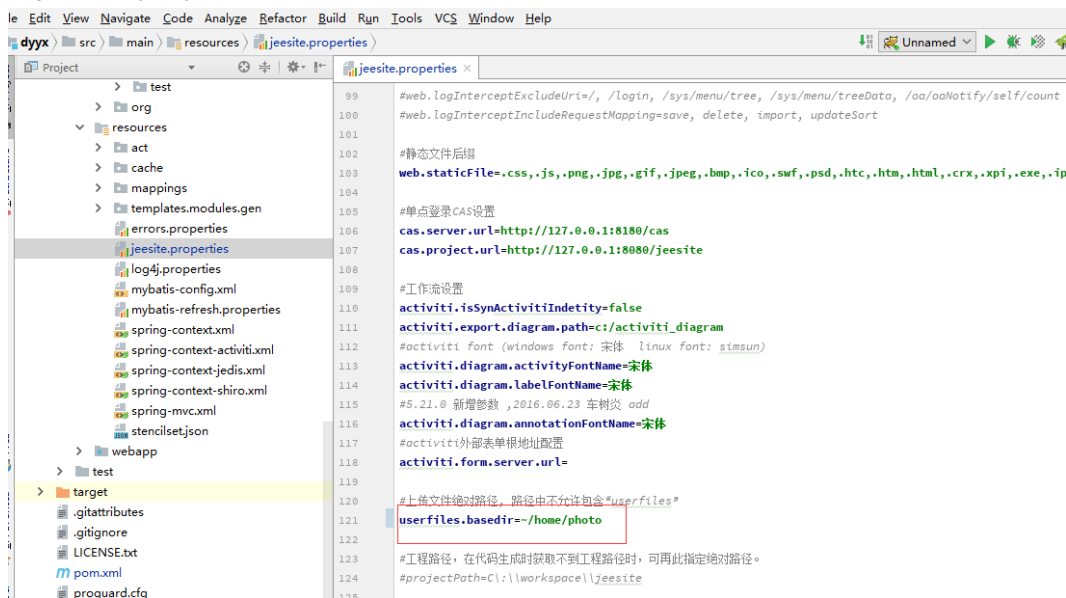
- 首先将idea编码改成utf8



- 在jeesite.properties里面配置数据库连接项

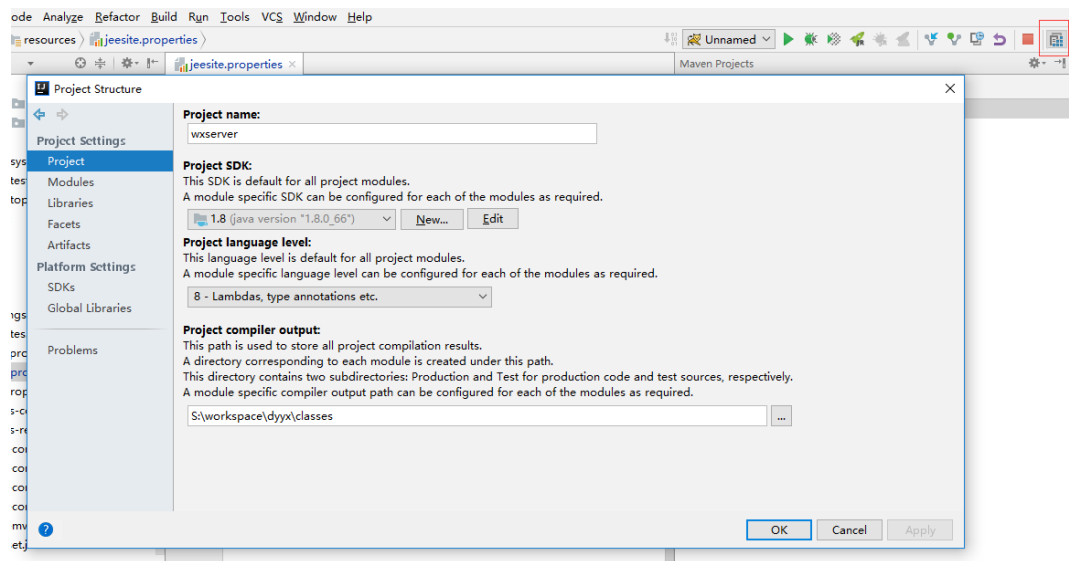


- 在jeesite.properties里面配置上传的文件存放的位置

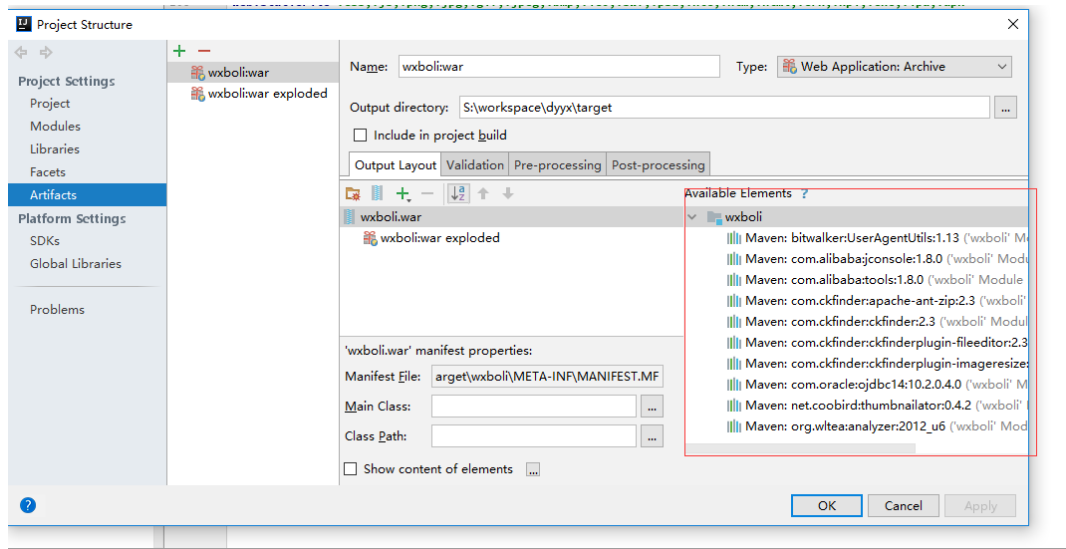


4.让空项目运行起来(参考文档)

- 点击红框位置
- 配置相关选项
- project修改project name ProjectSDK



- 注意这里的maven包要都加进去 -



5.浏览器访问,打开后台

6.设计后台模块

7.代码生成

- 添加表
- 这里的表就是在数据库中创建的表，选中将要生成代码的表

微信小程序后台

我的面板

在线办公

系统设置

代码生成

代码生成

业务表列表

业务表添加

业务表配置

生成方案配置

生成示例

表名: act_evt_log

下一步

返回

- 在添加表当中修改字段属性

字段列表

列名	说明	物理类型	Java类型	Java属性名称	主键	可否空	插入	编辑	列表	查询	查询匹配方式	显示表单类型	字典类型	排序
id	主键	varchar(54)	String	id	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	=	单行文本		10
name	产品名	varchar(54)	String	name	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Like	单行文本		20
intro	简介	varchar(255)	String	intro	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	=	单行文本		30
desc2	产品描述	text	String	desc2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	=	单行文本		40
files	缩略图	varchar(54)	String	files	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	=	单行文本		50
files2	中图	varchar(54)	String	files2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	=	单行文本		60
files3	大图	varchar(54)	String	files3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	=	单行文本		70
industry	行业	varchar(54)	String	industry	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	=	单行文本		80
catagory	分类	varchar(54)	String	catagory	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	=	单行文本		90
hot	是否热门	smallint(2)	String	hot	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	=	单行文本		100
order2	排序	int(9)	String	order2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	=	单行文本		110

- 然后根据表生成代码

微信小程序后台

我的面板

在线办公

系统设置

代码生成

代码生成

生成方案列表

生成方案修改

业务表配置

生成方案配置

生成示例

方案名称: 产品管理

模板分类: 增删改查 (单表) 生成结构: {包名}/{模块名}/{分层(dao,entity,service,web)}/{子模块名}/{java类}

生成包路径: com.thinkgem.jeesite.modules 建议模块包: com.thinkgem.jeesite.modules

生成模块名: product 英文, 生成Java代码的包名 可理解为子系统名, 例如 sys 包含entity、controller、service等

生成子模块名: 可选, 分层下的文件夹, 例如

生成功能描述: 产品管理 将设置到类描述

生成功能名: 产品管理 用作功能提示, 如: 保存“某某”成功

生成功能作者: 李云涛 功能开发者

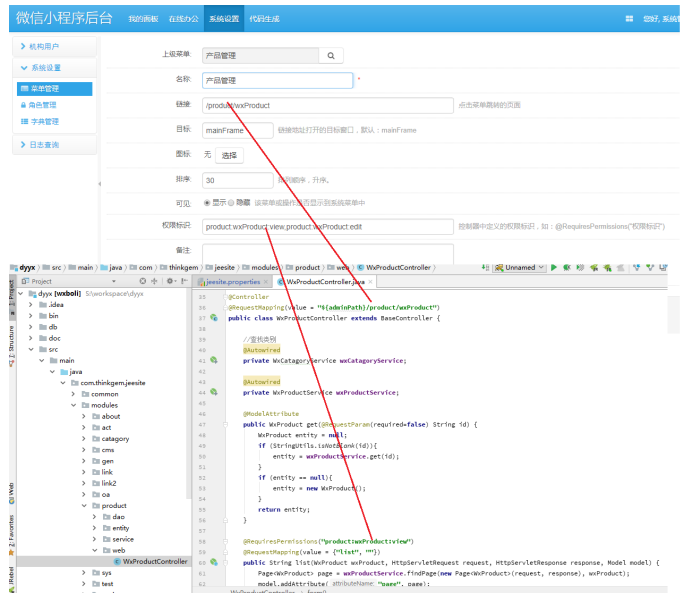
业务表名: wx_product : 产品表 生成的数据表, 一对多情况下请选择主表。

生成选项: ☐ 是否替换现有文件

8.菜单中添加模块

模型管理	/act/model	100	显示	act.model.edit	修改 删除 添加下级菜单
产品管理		330	显示		修改 删除 添加下级菜单
产品管理	/product/wxProduct	30	显示	product.wxProduct.view.prod...	修改 删除 添加下级菜单
类别管理	/catagory/wxCatagory	60	显示	catagory.wxCatagory.view.ca...	修改 删除 添加下级菜单
首页大广告	/topimage/wxTopimage	90	显示	topimage.wxTopimage.view.to...	修改 删除 添加下级菜单
关于我们	/about/wxAbout	120	显示	about.wxAbout.view.about.wx...	修改 删除 添加下级菜单
快速入口	/link/wxLink	150	显示	link.wxLink.view.link.wxLin...	修改 删除 添加下级菜单
快速入口-热门商品	/link2/wxLink2	180	显示	link2.wxLink2.view.link2.wx...	修改 删除 添加下级菜单

- 这里配置的权限，就是权限标识，在下面角色管理里选中这个菜单的角色才能查看这个菜单



- 在角色管理里面配置某个角色是否有查看某个菜单的权限，这个权限标识就是上面填写的。比如这里让普通用户具有查看商品的权限

微信小程序后台	我的面板	在线办公	系统设置	代码生成	您好, 系统
机构用户	角色列表	角色添加			
系统设置					
菜单管理					
角色管理					
字典管理					
日志查询					
	角色名称	英文名称	归属机构	数据范围	操作
	公司管理员	hr	山东省总公司	所在公司及以下数据	分配 修改 删除
	本公司管理员	a	山东省总公司	所在公司数据	分配 修改 删除
	本部门管理员	c	山东省总公司	所在部门数据	分配 修改 删除
	部门管理员	b	山东省总公司	所在部门及以下数据	分配 修改 删除
	普通用户	d	公司领导	仅本人数据	分配 修改 删除
	系统管理员	dept	公司领导	所有数据	分配 修改 删除
	济南市管理员	e	济南市分公司	按明细设置	分配 修改 删除

微信小程序后台 我的面板 在线办公 系统设置 代码生成

机构用户 角色列表 角色修改

归属机构: 公司领导

角色名称: 普通用户

英文名称: d

角色类型: 任务分配

是否系统数据: 是

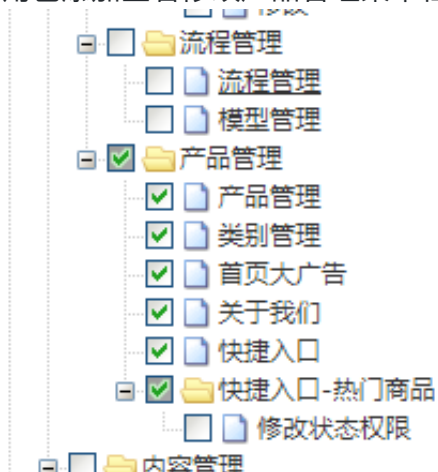
是否可用: 是

数据范围: 仅本人数据

角色授权:

- 功能菜单
 - 我的面板
 - 个人信息
 - 修改密码
 - 文件管理
 - 查看
 - 上传

- 给‘普通用户’角色添加查看修改产品管理菜单栏的权限



- 然后添加dyxx用户的时候给dyxx用户绑定‘普通用户’的权限,这样dyxx用户就可以访问产品管理菜单

9.配置一个admin角色
在系统设置里面用户添加

归属公司	归属部门	登录名	姓名	电话	手机
山东省总公司	公司领导	admin	管理员		
山东省总公司	公司领导	superadmin	系统管理员	8675	8675
山东省总公司	综合部	dyyx	第一印象		
山东省总公司	市场部	sd_scb	市场部		
山东省总公司	技术部	sd_jsb	技术部		
山东省总公司	研发部	sd_yfb	研发部		

开发前台数据对接

1.部署接口项目 具体询问项目经理

2.http 封装

- 在app.json里面添加封装好的用来连接数据库的接口类

```
http: {
  server: 'https://dyyx.raisesky.cn/',
  api: {
    selectSqlAction: 'DBCtrl.select.do',
  }
}
```

```

        executeSqlAction: 'DBCtrl.execute.do'
    },
    post: function (url, data, success, fail = function (e) {
        //console.error('fail:')
        //console.error(e)
    }, complete) {
        var self = this;
        wx.request({
            url: self.server + url,
            header: {
                "content-type": "application/x-www-form-urlencoded"
            },
            data: data,
            dataType: 'json',
            method: 'POST', // OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE,
CONNECT
            // header: {}, // 设置请求的 header 默认是application/json
            success: success,
            fail: fail,
            complete: complete || function (e) {
                //console.info(e)
            }
        });
    },
    postSelectSql: function (data, success, fail, complete) {
        this.post(this.api.selectSqlAction, data, success, fail, complete)
    },
    postExecuteSql: function (data, success, fail, complete) {
        this.post(this.api.executeSqlAction, data, success, fail, complete)
    }
},

```

- 微信小程序必须在后台设置request请求的合法域名，在设置中添加，开发时可以暂时按照下面步骤暂时不校验



- 微信小程序的域名必须是https的，所以要申请https域名。申请方法参考文章末尾
- 可以在开发时选择不校验安全域名、TLS版本以及HTTPS证书来暂时开发

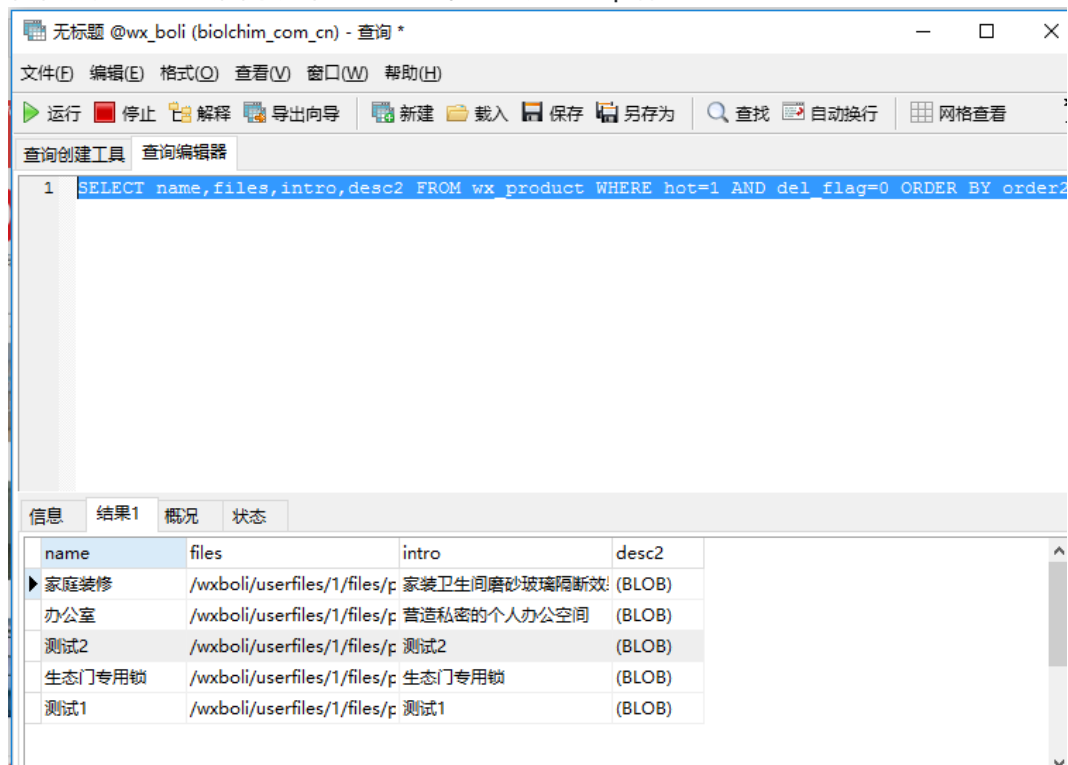


3.测试满足当前页面的要求 paw\postman

- 用postman向接口发送请求，查看返回数据类型。

4.编写sql

- 在需要查询的页面添加在navicat中运行好的sql语句



```
//将在navicat中验证成功的sql写到s
var app = getApp()
Page({
  sql: {
    selectHot:"SELECT name,files,intro,desc2 FROM wx_product WHERE hot=1
AND del_flag=0 ORDER BY order2",
    selectTopImage:"",
  },
});
```

- 在需要调用数据的页面.js添加如下代码

```
var app = getApp();

var self=this;
app.http.postSelectSql({
  sql: self.sql.selectHot,//将调用上面定义的sql语句
  params: []//这里填写的是sql参数
}, function (res) {
  var od = res.data;
  if (od && od.length > 0) {
    //将数据添加到data数据下调用时，在xml中用{{venuesItems}}调用
    self.setData({
```

```

        venuesItems:od
      })
    }
  }

```

5.双向绑定

- 通过第4步从后台端口向数据库查询到数据后，通过self.setData()把数据添加到了data:{} 然后就可以在.xml页面上通过{{}}获取数据

```

<view class="venues_box">
  <view class="venues_list">
    <block wx:for="{{venuesItems}}">
      <view class="venues_item" bindtap="goDetaile" data-index="{{index}}">
        <!-- <navigator url="../details/index?id={{item.id}}">
        -->
        <image
src="http://weixin.gowants.cn:8080{{item.files}}"/>
        <!-- </navigator> -->
        <view class="venues_text">{{item.name}}</view>
      </view>
    </block>
  </view>
</view>

```

7.富文本编辑（因为客户向后台保存的商品描述等数据可能是html格式的富文本，微信小程序不支持富文本，所以需要转换）

- 首先将插件文件夹复制到项目里面（可以到旧项目里面拷）

```

├── utils
├── wxParse
├── app.js
├── app.json
└── app.wxss

```

- 在.js中添加如下代码

```

var app = getApp();

```

```
var WxParse = require('../../../../wxParse/wxParse/wxParse.js');
```

```
/**
 * WxParse.wxParse(bindName , type, data, target,imagePadding)
 * 1.desc2绑定的数据名(必填)
 * 2.'html'可以为html或者md(必填)
 * 3.self.data.desc2为传入的具体数据(必填)
 * 4.this为Page对象(必填)
 * 5.imagePadding为当图片自适应是左右的单一padding(默认为0, 可选), 这里填了5
 */
WxParse.wxParse('desc2', 'html', self.data.desc2, this, 5);
```

- 然后在.xml文件中双向绑定数据

```
<!--detail.wxml-->
<import src="../../wxParse/wxParse/wxParse.wxml" />
<view class="title">
<text>商品描述</text>
</view>

<view class="my-body">
<view class="my-desc">
<template is="wxParse" data="{{wxParseData.desc2.nodes}}"/>
</view>
</view>
```

发布服务器

1. 申请域名、服务器、配置DNS解析

- 申请域名, 比如: youdomain.com
- 申请阿里云服务器, 获得外网IP地址、root账号和密码
- 配置域名A记录到阿里云服务器外网IP

2. 申请ssl证书

- 申请免费或收费ssl证书用于配置web服务器https安全访问
- 免费证书申请地址, 有些可能有时间限制: <https://freessl.me>

3. 下载安装nginx，配置ssl和反向代理

- 安装：yum install nginx 或者 apt-get install nginx
- 配置ssl，上传申请的ssl证书到nginx.conf同级目录下，找到nginx.conf文件，一般在/etc/nginx目录下，添加一个server配置如下
-

```
server {  
    listen 443;  
    server_name yun.raisesky.cn; #这个地方配置你的域名  
    client_max_body_size 1000M;  
    ssl on;  
    ssl_certificate 1_root_bundle.crt; #配置你的证书文件  
    ssl_certificate_key 2_yun.raisesky.cn.key; #配置你的证书文件  
    ssl_session_timeout 5m;  
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;  
    ssl_ciphers  
    AESGCM:ALL:!DH:!EXPORT:!RC4:+HIGH:!MEDIUM:!LOW:!aNULL:!eNULL;  
    ssl_prefer_server_ciphers on;  
    location / {  
        proxy_pass http://127.0.0.1:8000/wxserver/;  
        #在这里设置一个代理，也就是开启https访问的服务器http路径  
    }  
}
```

- 配置反向代理

```
server {  
    listen 80;  
    server_name yun.raisesky.cn; #这个地方配置你的域名  
    location / {  
        proxy_pass http://127.0.0.1:8000/wxserver/; #在这里设置  
        #一个反向代理，和upstream的名字一样  
    }  
}
```

4. 下载安装java运行时、tomcat服务器

- java安装：

[link](#)

- tomcat安装:

[link](#)

5.上传war(通过后台接口服务)到tomcat

- 检查，日志级别修改，tomcat端口确认和修改，图片上传路径修改，jdbc配置修改
- 打包
- `scp wxserver.war root@ip地址:/tomcat目录下的/webapps目录/`

tomcat webapps目录一般是: `/usr/share/tomcat/webapps`

6.配置tomcat端口，配置自启，启动tomcat

找到tomcat配置文件，一般是`/etc/tomcat/server.xml` 搜索8080，全部修改为8000

7.防火墙开启tomcat端口

- 找到iptables配置文件，一般在`/etc/sysconfig/iptables` 复制一条已有端口，然后把端口修改为8000端口

8.访问测试

- 重新加载nginx配置: `nginx -s reload` 或重启nginx: `service nginx restart` 等
- 启动tomcat:

`service tomcat start`

- 重启防火墙:

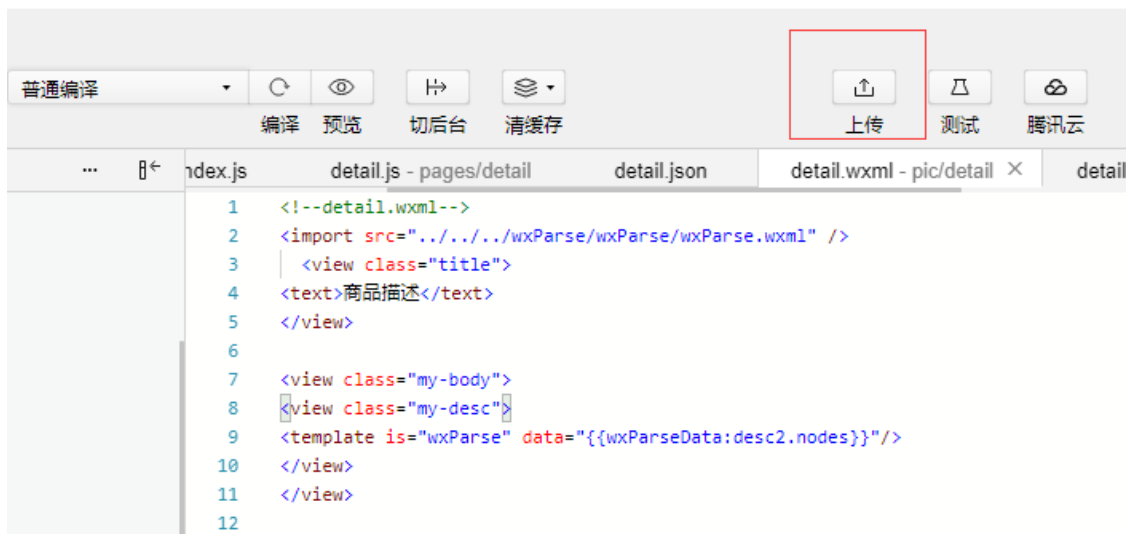
`service iptables restart` 9.打开浏览器输入域名后回车，页面中出现内容即表示启动成功

注意：如果service没有这个命令，则是redhat架构系统，可以对应换成如下模式的命令：
`systemctl status tomcat.service`
`systemctl start tomcat.service`
`systemctl restart nginx.service`

上线

登陆小程序微信公众号

1.上传小程序



2. 登录小程序微信公众号，在开发管理里面就会看到上传的小程序



3. 添加客服

