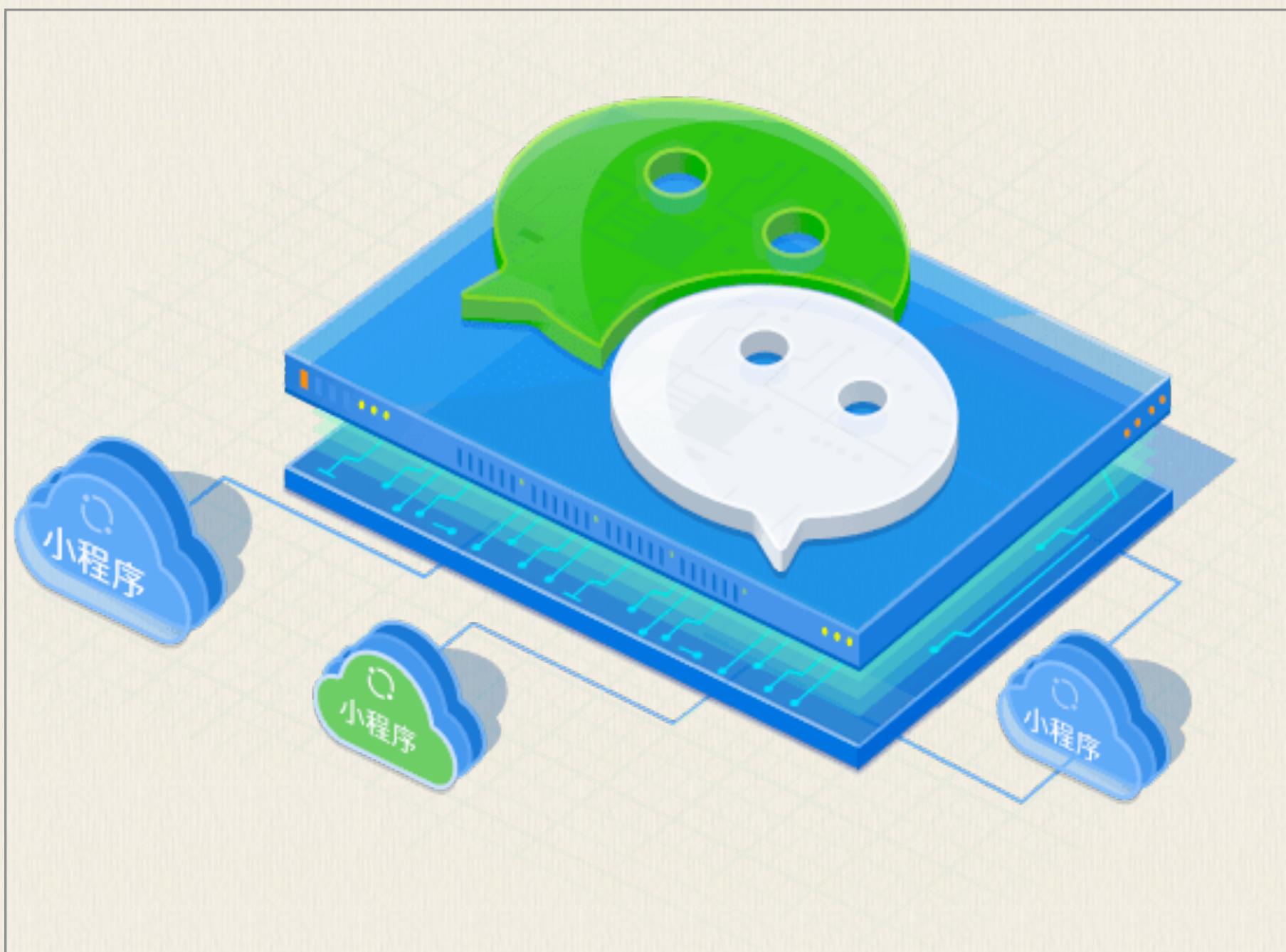


腾讯云微信小程序解决方案

五分钟搭建小程序完成服务器



简介

随着微信小程序的公测，无数个人和公司进入了小程序开发领域。

微信小程序只提供了客户端解决方案，并没有实现服务端，服务端需要用户自行搭建，而搭建服务器耗费大量时间，开发出一套弹性好、性能佳的更有难度。

腾讯云专门为微信小程序提供了服务器解决方案，实现一键部署、持续更新。

腾讯云的官方教程注重框架的技术和流程介绍，在简单快速使用上并没有花费心思，本文就专注于实战中如何快速上手使用，10分钟就可以从0开始实现Hello world。

本文也对使用解决方案中的各种坑进行了详细说明。

为什么要用腾讯云微信小程序解决方案

Q: 和阿里云相比，小程序服务器为什么要腾讯云服务器？

A: 阿里云提供的服务类别和帮助文档上，比腾讯云要好，但是对于特定业务而言，合适的就是最好的。

1. 价格：同是20G容量、1核、2G内存、1年下，阿里云1101元，腾讯云1105元，因此阿里没有价格优势。
2. 速度：微信和腾讯云同属一家公司，因此微信访问腾讯云专门的速度优化和CDN加速，是阿里云不行的。
3. 稳定：同属BAT，腾讯公司的技术至上氛围要远胜于阿里，加上腾讯这种纯软件化的公司，技术刚刚的，不要操心稳定问题。

Q: 开发微信小程序服务器有什么难度？

A: 1. HTTPS：微信小程序只允许HTTPS链接，市面很多SSL公共证书是要付费的，而自己制作的，在访问的时候会有安全警告，并且配置SSL服务器也是要时间的。

2. 会话：小程序是CS模式的，和传统的WEB开发完全不同，没有会话也就是没有session功能，开发时需要专门处理。

3. 鉴权：登录是必备的功能，而微信登录需要复杂的鉴权流程。

Q: 为什么使用微信小程序解决方案？

A: 具体的好处就不多说了，官方都有，这里我就通俗的来说下，那就是可以让你一分钟就可以部署出涵盖登录、会话等功能的一个完整的小程序服务器，只需要直接添加业务代码即可。并且这个服务器有强大的团队持续更新，即便是成本来说也是比招聘一个员工更划算。官方说明：https://www.qcloud.com/solution/la.html?utm_source=qcloudbanner&utm_medium=banner3&utm_campaign=qcloud

Q: 微信小程序解决方案支持什么语言？

A: php java nodejs



基础问题QA

很多人是第一次接触微信小程序，对很多基础问题不是很清楚，因此本文就解释一些基础性问题。

Q: 微信小程序和H5有什么差别?

A: 两者差别很大

- 通讯模式：小程序是CS模式的，和服务器的通讯是通过发送HTTP请求的方法来实现，没有session、cookie的概念，可以理解为传统C/S中客户端和服务器的通讯。H5本质上是HTML页面，因此是传统的WEB开发。
- 显示实现：虽然说小程序是JS，通过浏览器内核运行，但是小程序被微信深度优化，在界面显示之前已经预渲染好了，因此打开速度是极快的，而传统的H5是实时渲染，因此速度慢。
- 展示模式：导航栏：微信小程序可以定制化导航栏，比如颜色和文字等，而H5对导航栏没有控制力。 操作性：小程序对返回行为进行了定制，H5对返回上一步行为没有控制力。
- 能力：微信授予小程序调用微信客户端的能力，因此功能更为强大，而H5只能使用标准的功能。
- 开发：小程序优化了对页面的控制逻辑，开发者会发现操控页面超级简单，非常有利于提升开发速度。

- 分发：微信为小程序提供了专门的入口，未来必然为小程序做分发功能，就像小程序STORE。

-

Q：个人可以申请小程序开发吗？

A：目前小程序不支持个人开发，因此个人还不能申请。

A：很多人将小程序的开发权限和公众号的权限弄混，两者是完全不同的。

确认的方法很简单：

小程序的入口和公众号一样：<https://mp.weixin.qq.com/>

登录你的小程序账号，确认登录后的界面，这个才是小程序的界面，注意左上角的 LOGO，和公众平台是不同的，只要没这个就界面就是没权限



Q: 如何申请小程序的开发权限?

A: 1. 打开公众平台界面: mp.weixin.qq.com, 点击右上方的注册

2. 在注册页面选择“小程序”。

3. 注册完毕后, 打款认证即可, 目前不支持个人申请。

Q: 腾讯云解决方案的代码能单独使用吗?

A: 解决方案的代码分为服务端和客户端的SDK, 双方的通讯通过设置特殊的HTTP头来实现, 因此不可以单独使用, 必须配套, 任何一方都无法单独使用。

Q: 腾讯云小程序解决方案支持什么语言?

A: nodejs、php和java

Q: 小程序可以发布什么程序?

A: 微信对小程序进行了严格控制, 很让人蛋疼, 我们根据微信设置的小程序的分类可以看出, 如果没有合适的分类放置, 预示着很难通过审查。



搭建服务器

Q: 我没有腾讯服务器，怎么搭建腾讯云微信解决方案服务器？

A: 这种情况搭建服务器最简单、快捷和便宜。

1. 使用前先要申请微信小程序的使用权限，获得微信小程序APPID，下一步的申请需要它 <https://mp.weixin.qq.com/wxopen/waregister?action=step1>
2. 去申请微信解决方案使用权限：<https://www.qcloud.com/act/apply/xiaochengxu>
3. 获得使用权限后，需要创建微信小程序服务器，此时需要输入APPID，设置完APPID后再修改就会比较麻烦。
4. 此时要选择第一次的购买时间，第一次购买为半价，因此建议直接选最长的时间。

Q: 我已经有了腾讯服务器，怎么搭建腾讯云微信解决方案服务器？

A: 腾讯云提供了专门的解决方案镜像。备份你的数据，然后使用镜像重装服务器，选择你需要的语言镜像。部署文档：<https://github.com/tencentyun/wafer/wiki>



微信小程序服务端NODEJS SDK镜像

提供商：腾讯科技（深圳）有限公司

最新版本：1.0.0 操作系统：CentOS ...

功能简介：腾讯云微信小程序服务 Node.js 示例。包含鉴权服务和信道服务使用 SDK 及聊天室示例。

优惠价：免费



微信小程序服务端JAVA SDK镜像

提供商：腾讯科技（深圳）有限公司

最新版本：1.0.0 操作系统：CentOS ...

功能简介：腾讯云微信小程序服务 Java 示例。包含鉴权服务和信道服务使用 SDK 及聊天室示例。

优惠价：免费



微信小程序服务端PHP SDK镜像

提供商：腾讯科技（深圳）有限公司

最新版本：1.0.0 操作系统：CentOS ...

优惠价：免费



在

Q: 微信小程序解决方案需要几个服务器?

A: 总共三个服务器:

业务管理服务器: 专门处理用户自定义业务和解决方案业务。

会话管理服务器: 也叫鉴权服务器, 专门用来处理微信登录鉴权信息。

数据库: 存储鉴权信息和业务相关数据。

目前官方镜像优化并不好, 会话管理服务器因为老版本使用了本地数据库, 因此在该服务器上安装了MySQL, 但是后来为了扩展便捷, 改用云数据库, 但是并没有卸载MySQL。

Q: 我的小程序APPPID、APPSECRET变了, 如何修改?

A: 小程序的这些信息存储在cAUTH数据库中, 数据库的配置文件位置: /opt/lampp/htdocs/mina_auth/system/db/db.ini

Q: 会话数据库在哪里?

A: 数据库的配置文件位置: /opt/lampp/htdocs/mina_auth/system/db/db.ini



hello world

Q: 本文的hello world使用哪个语言?

A: 解决方案使用nodejs、php和java语言，本教程使用php语言。各个语言的实现不同。

Hello World步骤:

前面已经准备完毕服务器，这里开始Hello world的实现。

一、项目准备:

解决方案是开源的，并提供了SDK调用，因此需要下载调用，总共需要4个项目。

1. 客户端小程序准备:

客户端SDK: <https://github.com/tencentyun/wafer-client-sdk>

说明: 必备，直接复制到小程序项目中，供小程序调用使用，通过此sdk连接服务器实现解决方案各个功能。

客户端DEMO: <https://github.com/tencentyun/wafer-client-demo>

说明: 腾讯云提供的微信小程序demo，非必备。

2.服务端：

服务端SDK： <https://github.com/tencentyun/wafer-php-server-sdk>

说明：hello world非必要，

实现了业务服务器的解决方案的业务逻辑，存放在业务服务器的/data/release/php-weapp-demo/vendor.

如果不打算修改解决方案的实现，则不需要修改。

服务端业务框架： <https://github.com/tencentyun/wafer-php-server-demo>

说明：必要，用来存放业务逻辑代码，我们的hello world也是在这里。

存放在业务服务器的/data/release/php-weapp-demo/vendor.

二、添加服务端业务逻辑代码：

微信小程序业务服务端解决方案采用的是CodeIgniter3.1.0框架。我们的业务逻辑和解决方案的逻辑都是基于此框架的。

没有接触过CI框架的也不用害怕，特别简单，跟着步骤操作写完hellow world就学会使用了。

hello world完整逻辑：小程序POST提交用户信息表单到业务逻辑服务器，业务服务器存储到云数据库中。

打开下载的“服务端业务框架”代码目录。

1. 配置业务框架的数据库链接信息：

wafer-php-server-demo/application/config/database.php

配置云数据库的地址、用户和密码。

配置项详细说明：

http://codeigniter.org.cn/user_guide/database/configuration.html

database.php

```
$db['default'] = array(
    'dsn' => "",
    'hostname' => '10.66.xxx.xxx',
    'username' => 'root',
    'password' => 'xxxxxxxx',
    'database' => 'xxxxx',
    'dbdriver' => 'mysqli',//mysql、mysqli、postgres、odbc
    'dbprefix' => "",
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => "",
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => "",
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);
```

2. 创建一个用户数据库表

```
CREATE TABLE `user` (
  `nickname` varchar(20) NOT NULL,
  `uid` int(11) NOT NULL AUTO_INCREMENT,
  `gender` int(11) DEFAULT NULL,
  `language` varchar(255) NOT NULL,
  `city` varchar(255) DEFAULT NULL,
  `province` varchar(255) DEFAULT NULL,
  `country` varchar(255) DEFAULT NULL,
  `avatarUrl` varchar(255) NOT NULL,
  `openId` varchar(255) NOT NULL,
  PRIMARY KEY (`uid`)
)
```


3. 创建数据库操作文件

wafer-php-server-demo/application/models/user_modle.php

相关详细说明:

http://codeigniter.org.cn/user_guide/general/models.html

```
<?php
//引入 腾讯云微信云端服务器解决方案的云端SDK
use \QCloud_WeApp_SDK\Auth\LoginService as LoginService;

class User_model extends CI_Model {

    public function __construct()
    {
        $this->load->database();
    }
    //插入用户到数据库表格，并获取当前微信用户的uid
    public function regUser()
    {
        //从鉴权服务器获取完整的用户信息，包括openId
        $ui = LoginService::check();
        if($ui['code']==-1){
            return $ui['code'];
        }
        $uinfo = $ui['data']['userInfo'];
        //根据openid查询是否已经在用户表注册
        $query = $this->db->get_where('user',array('openId'=>$uinfo['openId']));
        $row = $query->row();
        if(!isset($row)){
            log_message('info', 'new user insert' . "\n");
            $row = $this->db->insert('user',array('nickname'=>$uinfo['nickName'],
            'gender'=>$uinfo['gender'],
            'language'=>$uinfo['language'],
            'city'=>$uinfo['city'],
            'province'=>$uinfo['province'],
            'country'=>$uinfo['country'],
            'avatarUrl'=>$uinfo['avatarUrl'],
            'openId'=>$uinfo['openId']));
        }
        return $row->uid;
    }
}
```

4. 创建一个控制器 如同java的servlet

wafer-php-server-demo/application/controllers/user.php

详细说明同上一步骤。

```
<?php
use \QCloud_WeApp_SDK\Helper\Util as Util;
use \QCloud_WeApp_SDK\Auth\Constants as Constants;
use \QCloud_WeApp_SDK\Auth>LoginService as LoginService;

class User extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
        //加载上面定义的User_model类
        $this->load->model('user_model');
    }

    public function reg(){
        $data = $_POST;
        $this->user_model->regUser();
    }
}
```

将上述步骤的代码上传到 业务服务器的 /data/release/php-weapp-demo 对应的目录。

5.配置微信小程序客户端。

a. 将下载的客户端SDK COPY到小程序根目录 wxyun

b. 打开app.js

c.

```

//加载 微信云客户端SDK
var qcloud = require('./wxyun/index');
var config = require('./config');

App({
  onLaunch: function () {
    qcloud.setLoginUrl(config.service.loginUrl);

    qcloud.request({
      // 要请求的地址
      url:"https://xxxxx/user/reg",
      // 请求之前是否登陆，如果该项指定为 true，会在请求之前进行登录
      login: true,
      success(result) {
        console.debug('请求成功完成');
        console.log('request success', result);
        wx.setStorage({
          key:"uid",
          data:result.data
        })
      },

      fail(error) {
        console.debug('请求失败', error);
        console.log('request fail', error);
      },

      complete() {
        console.log('request complete');
      }
    });

    console.log('App Launch');
  }
});

```




解决方案开发常见问题 解答

Q: 提示请配合客户端SDK登录后再进行请求



A: 解决方案的客户端在提交链接请求前，将用户信息写入HTTP头的X-WX-Encrypted-Data，在服务器取出来后判定是否登录等，若出现该问题，请确认：

1. 是否使用客户端sdk的qcloud.request发送的请求
2. 是否服务端的SDK是最新版