准备域名

任务时间：20min ~ 40min

微信公众平台需要配置服务器地址 URL 访问，在实验开始之前，我们要准备域名。

域名注册

如果您还没有域名，可以[在腾讯云上选购](https://dnspod.qcloud.com/?fromSource=lab)，过程可以参考下面的视频。

* 视频 - 在腾讯云上购买域名

域名解析

域名购买完成后, 需要将域名解析到实验云主机上，实验云主机的 IP 为：

<您的 CVM IP 地址>

在腾讯云购买的域名，可以[到控制台添加解析记录](https://console.qcloud.com/domain)，过程可参考下面的视频：

* 视频 - 如何在腾讯云上解析域名

域名设置解析后需要过一段时间才会生效，通过 ping 命令检查域名是否生效 [[?](https://cloud.tencent.com/developer/labs/lab/10196)]，如：

ping www.yourmpdomain.com

如果 ping 命令返回的信息中含有你设置的解析的 IP 地址，说明解析成功。

注意替换下面命令中的 www.yourmpdomain.com 为您自己的注册的域名

申请微信个人订阅号

任务时间：5min ~ 10min

在开始搭建我们的订阅号服务器之前，需要先拿到订阅号相关信息。

注册开发者账号

如果你还不是微信订阅号开发者，请先在微信公众平台注册：

https://mp.weixin.qq.com

具体注册流程可参考如下视频：

* 视频 - 注册开发者账号

若您已注册，请点击下一步。

获取微信订阅号公众平台认证字段信息

我们需要获取3个字段：AppID Token EncodingAESKey。

登录微信公众平台，依次进入 开发 - 基本配置 可以拿到 AppID。

在基本配置 - 服务器配置 - 修改配置 表单中：

URL 填第一步申请的域名;

Token 用户根据提示填写，用于后面校验服务端合法性;

EncodingAESKey 点击随机生成按钮来生成。

当点击表单提交按钮时，微信会通过 Token 来校验 URL 的合法性，这个我们在后面步骤实现，此界面暂时保留不关闭。

AppID Token EncodingAESKey 这3个参数具体的获取步骤也可以参照下面的视频

* 视频 - 获取微信订阅号信息

搭建 HTTP 服务

任务时间：15min ~ 30min

下面的步骤，将带大家在服务器上使用 Node 和 Express 搭建一个 HTTP 服务器

安装 NodeJS 和 NPM

使用下面的命令安装 NodeJS 和 NPM

curl --silent --location https://rpm.nodesource.com/setup\_8.x | sudo bash -

yum install nodejs -y

安装完成后，使用下面的命令测试安装结果

node -v

编写 HTTP Server 源码

**创建工作目录**

使用下面的命令在服务器创建一个工作目录：

mkdir -p /data/release/weapp

进入此工作目录

cd /data/release/weapp

**创建 package.json**

在刚才创建的工作目录创建 package.json，添加我们服务器包的名称和版本号，可参考下面的示例。

**示例代码：/data/release/weapp/package.json**

{

"name": "weapp",

"version": "1.0.0"

}

完成后，使用 Ctrl + S 保存文件

**添加 Server 源码**

在工作目录创建 app.js，使用 Express.js 来监听 5050 端口[[?](https://cloud.tencent.com/developer/labs/lab/10196)]，可参考下面的示例代码(注：请将 app.js文件中的token/appid/encodingAESKey等配置项替换为您的订阅号对应的取值)。

**示例代码：/data/release/weapp/app.js**

// 引用 express 来支持 HTTP Server 的实现

const express = require('express');

// 引用微信公共平台自动回复消息接口服务中间件

var wechat = require('wechat');

// 创建一个 express 实例

const app = express();

// 配置微信公众平台参数，在教程第二步中获取

var config = {

token: 'your token', // 填第二步中获取的 `token`

appid: 'your appid', // 填第二步中获取的 `appid`

encodingAESKey: 'your encodingAESKey', // 填第二步中获取的 `encodingAESKey`

checkSignature: true // 可选，默认为true。由于微信公众平台接口调试工具在明文模式下不发送签名，所以如要使用该测试工具，请将其设置为false

};

app.use(express.query());

app.use('/', wechat(config, function (req, res, next) {

res.reply({

content: '你好，Hello World!',

type: 'text'

});

}));

// 监听端口，等待连接

const port = 5050;

app.listen(port);

// 输出服务器启动日志

console.log(`Server listening at http://127.0.0.1:${port}`);

本实验会以 5050 端口的打开作为实验步骤完成的依据，为了后面的实验步骤顺利进行，请不要使用其它端口号

运行 HTTP 服务

**安装 PM2**

在开始之前，我们先来安装 [[PM2](https://cloud.tencent.com/developer/labs/lab/10196)]

npm install pm2 --global

PM2 安装时间可能稍长，请耐心等候 [[?](https://cloud.tencent.com/developer/labs/lab/10196)]

**安装 Express**

我们的服务器源码里使用到了 Express 模块，下面的命令使用 NPM 来安装 Express

cd /data/release/weapp

npm install express --save

**安装 Wechat**

我们的服务器源码里使用到了 Wechat 模块，下面的命令使用 NPM 来安装 Wechat

cd /data/release/weapp

npm install wechat --save

**启动服务**

安装完成后，使用 PM2 来启动 HTTP 服务

cd /data/release/weapp

pm2 start app.js

现在，您的 HTTP 服务已经在 http://<您的 CVM IP 地址>:5050 运行

要查看服务输出的日志，可以使用下面的命令：

pm2 logs

如果要重启服务，可以使用下面的命令：

pm2 restart app

我们使用 PM2 来进行 Node 进程的运行、监控和管理

NPM 仓库在国内访问速度可能不太理想，如果实在太慢可以尝试使用 CNPM 的 Registry 进行安装：npm install pm2 -g --registry=https://r.cnpmjs.org/

搭建 nginx 对外服务

任务时间：15min ~ 30min

NodeJs只是侦听的机器上的 5050 端口，我们使用 nginx 侦听 80 端口提供对外域名服务

安装 Nginx

在 CentOS 上，可直接使用 yum 来安装 Nginx

yum install nginx -y

安装完成后，使用 nginx 命令启动 Nginx：

nginx

此时，访问 http://<您的域名> 可以看到 Nginx 的测试页面 [[?](https://cloud.tencent.com/developer/labs/lab/10196)]

如果无法访问，请重试用 nginx -s reload 命令重启 Nginx

配置 HTTP 反向代理

外网用户访问服务器的 Web 服务由 Nginx 提供，Nginx 需要配置反向代理才能使得 Web 服务转发到本地的 Node 服务。

Nginx 配置目录在 /etc/nginx/conf.d，我们在该目录创建 wechat.conf

**示例代码：/etc/nginx/conf.d/wechat.conf**

server {

listen 80;

server\_name www.example.com; # 改为第一步申请的域名

location / {

proxy\_pass http://127.0.0.1:5050;

}

}

按 Ctrl + S 保存配置文件，让 Nginx 重新加载配置使其生效：

nginx -s reload

在浏览器通过 http 的方式访问你解析的域名来测试 HTTP 是否成功启动

使用Server端回复微信消息

任务时间：1min ~ 5min

提交服务端配置

我们将第二步微信公众平台中保留的表单提交，同时将 基本配置 - 服务器配置 启用

关注、发送与消息回复

首先通过二维码关注微信订阅号

在聊天界面向微信公众号发送一条消息

最终我们会回到一条 你好，Hello World! 的回复

大功搞成

恭喜！您已经完成了搭建微信订阅号后台服务的实验内容！您可以留用或者[购买 Linux 版本的 CVM](https://buy.qcloud.com/?fromSource=lab) 继续学习。

**推荐活动**

[学生主机 10 元起](https://cloud.tencent.com/act/campus)

**推荐实验**

[基于 CentOS 搭建 WordPress 个人博客](https://cloud.tencent.com/developer/labs/lab/10001)

[基于 Ubuntu 搭建微信小程序服务](https://cloud.tencent.com/developer/labs/lab/10125)

[基于 CentOS 搭建 Discuz 论坛](https://cloud.tencent.com/developer/labs/lab/10030)

[搭建微信订阅号后台服务](https://cloud.tencent.com/developer/labs/lab/10196)

[基于 ubuntu+jekyll 搭建个人博客](https://cloud.tencent.com/developer/labs/lab/10421)

**相关产品**