# 1.公众号申请

## 注册一个公众号

首先进入微信公众号平台，然后注册一个账号，类型选择订阅号，然后根据相应的提示完成信息的填写和验证。

# 2.后台服务器的搭建

## 安装NodeJS

首先执行以下命令

sudo su

sudo su

下载最新的稳定版 v6.10.3 到本地

wget https://nodejs.org/dist/v6.10.3/node-v6.10.3-linux-x64.tar.xz

wget https://nodejs.org/dist/v6.10.3/node-v6.10.3-linux-x64.tar.xz

下载完成后, 将其解压

tar xvJf node-v6.10.3-linux-x64.tar.xz

tar xvJf node-v6.10.3-linux-x64.tar.xz

将解压的 Node.js 目录移动到 /usr/local 目录下

mv node-v6.10.3-linux-x64 /usr/local/node-v6

mv node-v6.10.3-linux-x64 /usr/local/node-v6

配置 node 软链接到 /bin 目录

ln -s /usr/local/node-v6/bin/node /bin/node

## 配置 NPM

NPM 是 Node.js 的包管理和分发工具。它可以让 Node.js 开发者能够更加轻松的共享代码和共用代码片段

下载 node 的压缩包中已经包含了 npm , 我们只需要将其软链接到 bin 目录下即可。

ln -s /usr/local/node-v6/bin/npm /bin/npm

## 配置环境变量

将 /usr/local/node-v6/bin 目录添加到 $PATH 环境变量中可以方便地使用通过 npm 全局安装的第三方工具。

echo 'export PATH=/usr/local/node-v6/bin:$PATH' >> /etc/profile

生效环境变量

source /etc/profile

## 安装 Express

安装 Express 应用脚手架：

npm install express-generator -g

## 创建新的项目

执行

cd /data/

express yourApp

这时候会在 /data 生成应用目录 /yourApp，继续执行以下命令

cd yourApp

npm install

npm start

此时服务已经在 http://139.199.219.124:3000 上启动，可以在浏览器中浏览测试。

# 3.进入开发者模式

## 配置服务器信息，进入开发者模式

微信平台会去验证你填写的服务器的有效性，所以配置服务器前要先编写服务器有效性验证的代码。

在终端使用 Ctrl + C 终止刚才启动的服务，然后执行以下命令安装依赖的文件：

npm install xml2js sha1 querystring

编辑 app.js，参考下面的代码：

#### app.js

var express = require('express');

var app = express();

var http = require('http');

var sha1 = require('sha1');

var xml2js = require('xml2js');

var querystring = require('querystring');

process.env.PORT = '80';

function checkSignature(req) {

// 获取校验参数

var signature = req.query.signature;

var timestamp = req.query.timestamp;

var nonce = req.query.nonce;

// 此处为实验分配了一个 token，也可以修改为自己的 token

var token = 'iqoSQgM7';

// 按照字典排序

var params = [token, timestamp, nonce];

params.sort();

// 连接

var str = sha1(params.join(""));

// 返回签名是否一致

return str == signature;

}

// 接入验证

app.get('/', function(req, res) {

// 签名成功

if (checkSignature(req)) {

res.send(200, req.query.echostr);

} else {

res.send(200, 'fail');

}

});

module.exports = app;

## 启动服务

npm start

登录微信公众号平台，然后拉到页面的最下面，点击基本配置菜单。填写以下信息：

服务器地址（URL）：填写 139.199.219.124

这是您的服务器地址，服务器主要是用于实现机器人的逻辑。主机部分必须是公网 IP 或者可解析到公网 IP 的域名。

令牌（Token）：填写上面代码中使用到的 Token，如无修改，直接使用 iqoSQgM7

消息加解密密钥（EncodingAESKey）：随机字符串。可以让平台生成。

消息加解密方式：选择明文方式。

配置完毕后点击提交，提交成功后，在基本配置菜单页面点击启用。

此时，微信用户给公众号发送的信息，会由微信平台转发给您的服务器，然后您的服务器进行处理后返回给用户。

在页面点击提交，检查是否可以提交成功。

# 4.机器人的实现

## 实现机器人功能

首先在图灵机器人官网注册和申请一个机器人，具体步骤可以参考视频。

视频 - 申请图灵机器人

申请之后，我们来调用 API 实现机器人的功能。

编辑 app.js，参考下面的代码

#### app.js

var express = require('express');

var app = express();

var http = require('http');

var sha1 = require('sha1');

var xml2js = require('xml2js');

var querystring = require('querystring');

process.env.PORT = '80';

function checkSignature(req) {

// 获取校验参数

var signature = req.query.signature;

var timestamp = req.query.timestamp;

var nonce = req.query.nonce;

// 此处为实验分配了一个 token，也可以修改为自己的 token

var token = 'iqoSQgM7';

// 按照字典排序

var params = [token, timestamp, nonce];

params.sort();

// 连接

var str = sha1(params.join(""));

// 返回签名是否一致

return str == signature;

}

// 接入验证

app.get('/', function(req, res) {

// 签名成功

if (checkSignature(req)) {

res.send(200, req.query.echostr);

} else {

res.send(200, 'fail');

}

});

function handler(req, res) {

let buf = '';

// 获取XML内容

req.setEncoding('utf8');

req.on('data', function(chunk) {

buf += chunk;

});

// 内容接收完毕

req.on('end', function() {

xml2js.parseString(buf, function(err, json) {

if (err) {

err.status = 400;

} else {

req.body = json;

}

});

let data = req.body.xml;

var msg = {

"toUserName" : data.FromUserName[0],

"fromUserName" : data.ToUserName[0],

"createTime" : data.CreateTime[0],

"msgType" : data.MsgType[0],

"content" : data.Content[0],

"msgId" : data.MsgId[0]

};

request(msg, req, res)

});

}

function request(data, req, res) {

var msg = {

"key":'6d76234cf7ee488d84aa1a54397ae866', // 可以填入自己申请的机器人的apiKey

"info": data.content,

"userid": ~~(Math.random() \* 99999)

};

var text = querystring.stringify(msg);

var options = {

hostname: 'www.tuling123.com',

path: '/openapi/api',

method: 'POST',

headers: {

'Content-Type': 'application/x-www-form-urlencoded; charset=UTF-8'

}

};

var requestObj = http.request(options, function (response) {

var result = '';

response.setEncoding('utf8');

response.on('data', function (chunk) {

result += chunk;

});

response.on('end',function() {

try{

var obj = JSON.parse(result);

}

catch(e){

data.content = e.message;

echo(data, res);

return;

}

data.content = obj.text;

echo(data, res);

})

});

requestObj.on('error', function (e) {

console.log('problem with request: ' + e.message);

data.content = e.message;

echo(data, res);

});

requestObj.write(text);

requestObj.end();

}

function echo(data, res) {

var time = Math.round(new Date().getTime() / 1000);

var output = "" +

"<xml>" +

"<ToUserName><![CDATA[" + data.toUserName + "]]></ToUserName>" +

"<FromUserName><![CDATA[" + data.fromUserName + "]]></FromUserName>" +

"<CreateTime>" + time + "</CreateTime>" +

"<MsgType><![CDATA[" + data.msgType + "]]></MsgType>" +

"<Content><![CDATA[" + data.content + "]]></Content>" +

"</xml>";

res.type('xml');

res.send(output);

}

// Start

app.post('/', function(req, res) {

handler(req, res);

});

module.exports = app;

## 重启

操作键盘按下 Ctrl + C 终止之前的服务，然后重新启动：

npm start

# 5.部署完成

## 访问服务

在微信公众号中关注自己的公众号，然后发送信息给公众号，测试是否有返回。在微信公众号平台可以对公众号信息进行管理。更复杂的逻辑可以参考微信公众号平台开发者文档和图灵机器人开发文档。

## 大功告成

恭喜您已经完成了搭建微信 AI 机器人的学习，您可以留用或者购买 Linux 版本的 CVM 继续学习。