

nodejs微信开 发

博客选编



前言

原文出处：[nodejs微信开发](#) 作者：[sinat_29843547](#)

本系列文章经作者授权在看云整理发布，未经作者允许，请勿转载！

nodejs微信开发

nodejs的npm开源包 wechat 和wechat-api是非常好用的微信开发框架，nodejs做后台服务器，利用nginx反向代理，使用起来非常方便！开源包有淘宝前端大神朴灵团队不断更新，朴神著有<<深入浅出nodejs>>一书，非常厉害！

验证路径token

微信api地址

<http://mp.weixin.qq.com/wiki/17/2d4265491f12608cd170a95559800f2d.html#.E7.AC.AC.E4.B8.80.E6.AD.A5.EF.BC.9A.E5.A1.AB.E5.86.99.E6.9C.8D.E5.8A.A1.E5.99.A8.E9.85.8D.E7.BD.AE>

微信api说明

开发者通过检验signature对请求进行校验（下面有校验方式）。若确认此次GET请求来自微信服务器，请原样返回echostr参数内容，则接入生效，成为开发者成功，否则接入失败。

加密/校验流程如下：

1. 将token、timestamp、nonce三个参数进行字典序排序
2. 将三个参数字符串拼接成一个字符串进行sha1加密
3. 开发者获得加密后的字符串可与signature对比，标识该请求来源于微信
4. 代码：

```
/**
 * Created by tangxuelong on 15-10-16.
 * validateToken
 * 此文件只用于TOKEN验证
 */
var http = require('http');//内置http module
var config = require('./http.config');//配置module
var api = require('wechat-api');//npm wx
var url = require("url");
var crypto = require("crypto");

//微信接口的哈希加密方法
function sha1(str) {
  var md5sum = crypto.createHash("sha1");
  md5sum.update(str);
  str = md5sum.digest("hex");
  return str;
}
//微信路径token验证
function validate_token(req,res){
  //获取请求的query排序以后加密
  var query = url.parse(req.url, true).query;
  var signature = query.signature;
  var echostr = query.echostr;
  var timestamp = query['timestamp'];
  var nonce = query.nonce;
  var oriArray = new Array();
  oriArray[0] = nonce;
  oriArray[1] = timestamp;
  oriArray[2] = "XXXXXX";
  oriArray.sort();
  var original = oriArray.join("");
  var scyptoString = sha1(original);
  if (signature == scyptoString) {
    res.end(echostr);
    console.log("Confirm and send echo back");
  } else {
    res.end("false");
    console.log("Failed!");
  }
}
//创建http服务器
http.createServer(function(req,res){
  validate_token(req,res);
}).listen(config.port, config.http_ip);
console.log('http server is running');
```

自动回复 wechat

安装express

```
npm install -g express
```

```
express -e node_wechat
```

```
cd node_wechat
```

```
npm install express
```

```
vi app.js
```

app.js 如下

在微信公众平台开发者中心配置好以后,上一步已经知道了验证url接口原理并实现了验证

微信在验证成功以后就会将用户发送的消息和事件都发送到这个URL

这里的1337端口是经过nginx路由到这里的, 直接访问域名就可以路由到1337这个端口

我们要接收请求就还是要有HTPP服务器, 创建了一个http服务器

用到expressquery解析http request

接收消息 并返回 需要返回的内容就完成了自动回复

代码如下：

```
var wechat = require('wechat');
var express = require('express');
var app = module.exports = express.createServer();
app.use(express.query()); // Or app.use(express.query());

app.use('/', wechat('caiqr', function (req, res, next) {
    // 微信输入信息都在req.weixin上
    var message = req.weixin;
    console.log(message);
    res.reply({type: "text", content: 'Hello world!'});
}));

app.listen(1337,function(){
    console.log('port 1337 is running');
});
```

设置自定义菜单

自定义菜单和自动回复不一样，自动回复是需要接收微信转发到服务器的请求和事件

所以不需要http服务器,只需要将请求发送给微信api接口就可以了

代码用wechat-api实现也比较方便

需要node 一下的js文件

```
var config = require('./config');
var API = require('wechat-api');

var api = new API(config.appid, config.appsecret);
/*null
{ accessToken: 'bMawBdyY5idQLeX_-ODTWZVhOd51yIkIU3oYPeelTg_8BmqrmUAEEvszsKGDLconZxH
Ox6Nwd32W50X7ThzBWzgOCQanOeg4Vwv3xMhN7KU',
  expireTime: 1445244891114 }
{ errcode: 0, errmsg: 'ok' }*/
api.getAccessToken(function (err, token) {
  console.log(err);
  console.log(token);
});

var menu = JSON.stringify(require('./fixture/wx_menu.json'));
api.createMenu(menu, function (err, result) {
  console.log(result);
});
```

config.js

```
/**
 * Created by tangxuelong on 15-10-19.
 */
module.exports = {
  appid : "wx01ca1966fe6f8a88",
  appsecret : "ecedd0e93feccac893b05b67f5fe8538"
}
```

菜单的json文件

```
{
  "button":[
    {
      "type":"click",
      "name":"今日歌曲",
      "key":"V1001_TODAY_MUSIC"
    },
    {
      "type":"click",
      "name":"歌手简介",
      "key":"V1001_TODAY_SINGER"
    },
    {
      "name":"菜单",
      "sub_button":[
        {
          "type":"view",
          "name":"搜索",
          "url":"http://www.soso.com/"
        },
        {
          "type":"view",
          "name":"视频",
          "url":"http://v.qq.com/"
        },
        {
          "type":"click",
          "name":"赞一下我们",
          "key":"V1001_GOOD"
        }
      ]
    }
  ]
}
```

调用微信JS SDK

首先要在后台获取的JSconfig传到页面

页面传递数据用ajax

后台nodejs代码：

```
var config = require('./config');
var API = require('wechat-api');
var express = require('express');
var bodyParser = require('body-parser');
var api = new API(config.appid,config.appsecret)
var app = module.exports = express.createServer();

//使用中间件body-parser获取post参数
app.use(bodyParser.urlencoded({extended: false}));
app.use(bodyParser.json());

app.use(function (req, res, next) {
  //使用wechat-api获取JSconfig
  var param = {
    debug: false,
    jsApiList: ['onMenuShareTimeline', 'onMenuShareAppMessage'],
    url: req.body.url
  };
  /*api.getTicket(function(err,result){
    console.log(err);
    console.log(result);
  });*/
  api.getJsConfig(param, function(err,result){
    res.send(result);
  });
})

app.listen(1338,function(){
  console.log('port 1338 is running!');
});
```

前台页面代码

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
  <title>微信测试页面</title>
  <script src="http://cdn.bootcss.com/jquery/2.1.4/jquery.min.js"></script>
  <script src="http://res.wx.qq.com/open/js/jweixin-1.0.0.js"></script>
```



```

<script src="http://res.wx.qq.com/open/js/jweixin-1.0.0.js"></script>
<script>
  $(function(){
    //获取jsconfig
    var param_get_js_config = {'cmd':'get_js_config','url':'http://xl.caiqr.com/test.html'};
    hander_data_from_wechat_api(param_get_js_config,hander_js_config);

  })
  //配置JSconfig
  function hander_js_config(data){
    /*wx.config({
      debug: true, // 开启调试模式,调用的所有api的返回值会在客户端alert出来,若要查看传入的参数
      ,可以在pc端打开,参数信息会通过log打出,仅在pc端时才会打印。
      appId: "", // 必填,公众号的唯一标识
      timestamp: "", // 必填,生成签名的时间戳
      nonceStr: "", // 必填,生成签名的随机串
      signature: 'sM4AOVdWfPE4DxkXGfEs8VJAAX0hDsNvapedYVgPudpde87_OEkUjdX4mFJAutu
8m3l0Zg2FbxavtkibLqYZHcw',// 必填,签名,见附录1
      jsApiList: [] // 必填,需要使用的JS接口列表,所有JS接口列表见附录2
    });*/
    wx.config(data);
    wx.ready(function(){

      // config信息验证后会执行ready方法,所有接口调用都必须在config接口获得结果之后,config是一个客户端的异步操作,所以如果需要在页面加载时就调用相关接口,则须把相关接口放在ready函数中调用来确保正确执行。对于用户触发时才调用的接口,则可以直接调用,不需要放在ready函数中。
      wx.checkJsApi({
        jsApiList: ['onMenuShareTimeline'], // 需要检测的JS接口列表,所有JS接口列表见附录2,
        success: function(res) {
          alert(JSON.stringify(res));
          // 以键值对的形式返回,可用的api值true,不可用为false
          // 如: {"checkResult":{"chooseImage":true},"errMsg":"checkJsApi:ok"}
        }
      });
      wx.onMenuShareTimeline({
        title: '分享的标题', // 分享标题
        link: 'www.baidu.com', // 分享链接
        imgUrl: '', // 分享图标
        success: function (res) {
          alert('share success');
          // 用户确认分享后执行的回调函数
        },
        cancel: function (res) {
          alert('share fail')
          // 用户取消分享后执行的回调函数
        }
      });
    });
    wx.error(function(res){
      alert(JSON.stringify(res));
      // config信息验证失败会执行error函数,如签名过期导致验证失败,具体错误信息可以打开config的debug模式查看,也可以在返回的res参数中查看,对于SPA可以在这里更新签名。

    });
  }

```

//传递参数请求wechat_api数据，请求成功后的回调函数fn处理得到的数据

```
function hander_data_from_wechat_api(param,fn) {
```

```
  console.info(param);
```

```
  //发送请求获取wx_js_config
```

```
  $.ajax({
```

```
    url: 'http://xl.caiqr.com/wechat',
```

```
    type: 'POST',
```

```
    data: param,
```

```
    datatype: 'json',
```

```
    success: function (data) {
```

```
      fn(data);
```

```
    }
```

```
  })
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

分享的时候可以设置分享标题拿到分享成功还是失败的状态就成功了

网页获取用户授权信息

思路是这样的，第一步在页面获取到授权以后的code

拿code去nodejs后台的httpserver请求

nodejs接到请求以后，拿到code去拿accessToken,然后通过accessToken获取用户信息

然后把用户信息扔给页面

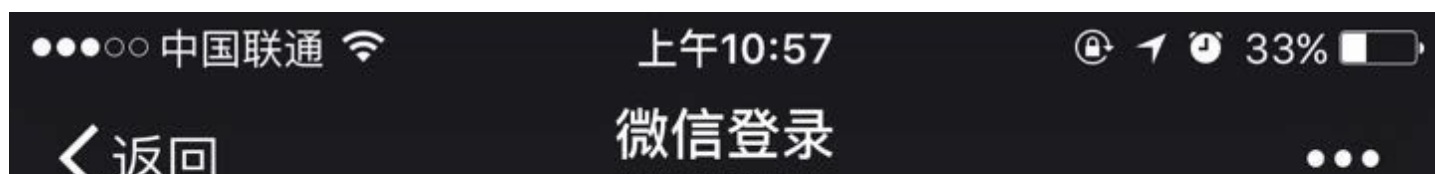
后台nodejs代码：

```
var request = require('request');
var nodeconfig = require('./nodeconfig');

//api.weixin

function wx(res, postData) {
  console.log(getUrlVars(postData));
  //微信获取用户资料第二步通过code换取网页授权access_token
  request.get({
    url : 'https://api.weixin.qq.com/sns/oauth2/access_token?appid=wx01ca1966fe6f8a88&secret=c893b05b38&' + postData + '&grant_type=authorization_code',
    form : {}
  }, function(error, response, body) {
    if (!error && response.statusCode == 200) {
      console.log(body);
      var jsondata = JSON.parse(body);
      //第四步拉取用户信息(需scope为 snsapi_userinfo)
      request.get({
        url : 'https://api.weixin.qq.com/sns/userinfo?access_token=' + jsondata.access_token + '&openid=' + jsondata.openid + '&lang=zh_CN',
        form : {}
      }, function(error, response, body) {
        if (!error && response.statusCode == 200) {
          var userdata = JSON.parse(body);
          console.log(userdata);
          nodeconfig.httpHeadSetting(res);
          res.end(JSON.stringify(userdata));
        }
      });
    }
  });
}

exports.wx = wx;
```





该网页由彩球体育开发，向其提供以下权限即可继续
操作

☒ 获得你的公开信息（昵称、头像等）

确认登录



