

微擎开发介绍

微擎工作流程介绍

模块开发指南

第三方接口说明

页面设计规范

模板机制

全局变量和预定义常量介绍

数据库操作函数

其他辅助函数

相关下载

第三方接口(轻量级扩展)

微擎还提供另外一种扩展系统功能的方式, 使用第三方接口模块, 你可以快速和轻量级的来扩展你的微擎系统. 甚至可以使用远程接口的形式兼容已有的微信管理功能.

第三方接口以一个系统模块的形式提供. 第三方接口模块提供两种形式的扩展方式: 本地文件和远程地址.

本地文件扩展

本地文件扩展适合于简单的 **Web** 抓取处理的场景. 这种场景不需要复杂的业务处理, 但是需要请求 **Web** 获取内容, 并将内容处理为合适的格式返回给公众平台. 例如天气预报, 快递查询等.

本地文件扩展应当将每个扩展功能实现为单独的**PHP**文件, 并将文件放置于微擎目录下的 `/source/modules/userapi/api` 位置. 此**PHP**文件的执行环境位于 **WeModuleProcessor** 对象的 `respond` 函数中(请参阅微擎工作流程-模块定义), 并被直接 `include` 并获得返回值后返回给公众平台.

数据传递

因为本地文件扩展的执行位置位于 **WeModuleProcessor** 对象的 `respond` 函数中, 因此可以直接使用 `$this->message` 成员变量来获取公众平台发送过来的消息内容. 其他的附加数据请参阅 微擎工作流程-模块定义.

***注意:** 你同样可以在文件扩展中使用 `$this->beginContext()` 和 `$this->endContext()` 来处理上下文锁定, 与模块中的处理不同之处在于在文件扩展中使用上下文锁定将会将会话锁定至当前的文件扩展中*

返回结果时, 请在**PHP**文件中直接使用 `return` 语句. 返回的结构与 **WeModuleProcessor** 对象的 `respond` 函数返回值相同.

下边有一个示例来实现查询快递的功能.

```

$matches = array();
$ret = preg_match('/^(?P<express>申通|圆通|中通|汇通|韵达) (?P<sn>[a-z\d]{1,})$/i', $this->message['content'], $matches);
if(!$ret) {
    return $this->respText('请输入合适的格式，快递公司+空格+单号，例如：申通 2309381801');
}
$express = $matches['express'];
$sn = $matches['sn'];
$mappings = array(
    '申通' => 'shentong',
    '圆通' => 'yuantong',
    '中通' => 'zhongtong',
    '汇通' => 'huitongkuaidi',
    '韵达' => 'yunda'
);
$code = $mappings[$express];
$url = "http://www.kuaidi100.com/query?type={$code}&postid={$sn}";

$dat = ihttp_get($url);
if(!empty($dat) && !empty($dat['content'])) {
    $traces = json_decode($dat['content'], true);
    if(is_array($traces)) {
        $traces = $traces['data'];
        if(is_array($traces)) {
            $traces = array_reverse($traces);
            $reply = '';
            foreach($traces as $trace) {
                $reply .= "{$trace['time']} - {$trace['context']}\n";
            }
            if(!empty($reply)) {
                $reply = "已经为您查到相关快递记录：\n" . $reply;
            }
            return $this->respText($reply);
        }
    }
}
return $this->respText('没有查找到相关的数据，请检查您的输入格式，正确格式为：快递公司+空格+单号，例如：申通 2309381801');

//以上的内容被保存为 express.php 后放置于 /source/modules/userapi/api 中，然后新增一条规则记录。使用正则表达式匹配方式匹配格式为：/^(?:申通|圆通|中通|汇通|韵达) (?:[a-z\d]{1,})$/i 然后既可以利用此公众号查询快递了。

```

远程地址扩展

远程地址扩展通过调用远程的URL来扩展微擎功能. 使用这样的方式你可以很简便的整合第三方微信平台(如乐享, 小猪, 甚至是另一个微擎), 其原理是模拟公众平台向第三方的接口地址发送请求(完全兼容官方的请求格式), 然后将返回的结果数据返回给公众平台. 采用这样的方式可以不动原有接口(特殊情况仅需要改动很少量代码), 即可将原有的微信管理工具接入微擎. 使用这样的方式你可以整合多个微信管理平台, 博采众家之长, 更好的为你的公众微信号服务.

处理方式

调用远程地址扩展你需要在添加自定义接口回复规则时选择远程地址. 系统会要求你填写远程地址的 URL 以及 Token.

这里的 URL 和 Token 请按照你原平台提供给微信公众平台的URL和Token填写即可. (而你在微信公众平台填写的 URL 和 Token 应该是微擎系统提供的)

- 公众平台发送粉丝用户的数据到达至微擎
- 微擎判断消息内容符合定义的规则(远程地址扩展方式)
- 微擎按照填写的 Token 重新签名请求数据
- 微擎将请求数据(重新签名)发送至远程地址
- 微擎将远程地址返回的内容返回给微信公众平台

上下文处理

如果需要在远程地址中处理上下文锁定, 需要在远程地址返回的数据中附加特定标记. 在上述处理流程的最后一步 "微擎将远程地址返回的内容返回给微信公众平台" 中, 微擎系统如果检测到返回的数据中包含 {begin-context} 内容时, 将会锁定会话上下文至当前的远程地址扩展中. 如果 检测到返回的数据包含 {end-context} 内容, 将会解除上下文锁定. 注意: {begin-context} 标记和 {end-context} 标记将会被微擎系统过滤掉. 因此最终的粉丝用户是看不到这样的附加数据的, 对用户的使用习惯没有任何破坏.

在远程地址扩展中处理上下文会话锁定, 可能需要你稍微改动一下你在其他平台中设定的处理规则或处理代码.

微擎新浪微博 (<http://weibo.com/we7team>)

微擎腾讯微博 (<http://t.qq.com/we7team>)

微擎微信公众号

关于微擎 服务协议 联系

我们