

Please visit website: <http://cxyroad.com>

## 利用hook开源项目开发一个摸鱼提示小助手

=====

### 前言

==

前段时间发了一个接入ChatGPT机器人火了，自己就找了类似的项目，最近看到在GitHub上面还是有很多人开发类似机器人项目，其中发现了一个项目挺不错的，是一个可以利用\*\*http 请求驱动的机器人\*\* 的项目

原理就是使用web PC协议登录，然后接收任何消息之后 把消息发送到自定义的 网址，根据相关的逻辑进行回复或者处理。

### 部署

==

现在就开始本地部署测试一下，依旧是使用docker部署，先亮出项目地址：

[github.com/danni-cool/...](http://cxyroad.com/  
"https://github.com/danni-cool/wechatbot-webhook")

![截图\_20240422000945.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/8a809102b0f84f5da04d67da504b991b~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1344&h=1375&s=398907&e=jpg&b=ffff)

项目介绍只有一句话，我就是喜欢这句话：**\*\*磨平了自己开发的障碍\*\***，开源的机器人项目大多都是Golang, Python, node语言开发的，修改别人的开发完毕之后的成熟项目很复杂，又很难嵌入到自己的项目中，这个就非常好，可以用自己熟悉的编程语言开发机器人。

部署方式非常简单，第一步找到项目根目录下的docker-compose.yaml文件，修改三个地方

...

```
version: '3.8'
services:
  wxBotWebhook:
    image: dannicool/docker-wechatbot-webhook
    container_name: wxbot_app
    volumes:
      - ./wxBot_logs:/app/log
    ports:
      - "3001:3001"
    environment:
      - LOG_LEVEL=info # 调整容器输出级别（不影响日志文件输出等级）运行时提示的消息等级（默认info，debug级别会输出详细的日志）
      # - DISABLE_AUTO_LOGIN=true # 如果不希望登录一次后就记住当前账号，想每次都扫码登陆，填 true
      - ACCEPT_RECVD_MSG_MYSELF=true # 如果希望机器人可以自己接收消息，填 true
      - RECVD_MSG_API=https://***.***.com/****/**** # 如果想自己处理收到消息的逻辑，在下面填上你的API地址，默认为空
      - LOGIN_API_TOKEN= ***** # 登录地址Token访问地址
      : http://localhost:3001/login?token=[LOCAL_LOGIN_API_TOKEN]
    restart: unless-stopped
```

...

ACCEPT\\_RECVD\\_MSG\\_MYSELF 打开机器人接收消息的开关

RECVD\\_MSG\\_API 自己接收该项目转发出来的消息，HTTP 协议API的地址（非常重要）

LOGIN\\_API\\_TOKEN 对接API 的token（为了进行安全的交互）

修改完成过后，使用docker-compose up -d 启动项目，接着就可以访问API直接登录了，根据GitHub项目的介绍

![截图\_20240422001926.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/20548616dcc7499ba8ed20c8c03a72d4~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1599&h=709&s=147559&e=jpg&b=ffffff)

如果自己部署该项目的地址是公网可以访问的，访问地址是

: [www.myspace.com, ](http://cxyroad.com/"https://www.myspace.com/%EF%BC%8C") LOGIN\\_API\\_TOKEN填写的密码是123456，那么获取登录二维码的地址就是

: [www.myspace.com/login?token...](http://cxyroad.com/"https://www.myspace.com/login?token=123456"), 浏览器直接访问就可以获得一个二维码, 用扫码就可以直接登录。

我这里已经登录了, 再次访问就会出现下面的提示

![截图\_20240422003156.png](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/6e8fab8bf90344f38e0cf73f6308584c~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=733&h=291&s=20528&e=jpg&b=ffffff)

简单开发

====

项目介绍有一个接口就是检测是否掉线, 我觉得这个功能非常重要, 写了一个接口, 十分钟检测调用一次这个接口, 如果检测掉线就使用企业通知我, 同时发送登录链接, 我就可以第一时间直接扫码重新登录了, 核心代码我使用了 PHP, 下面贴上代码给大家参考一下:

```
...
#[Crontab(rule: "*\*/10 * * * *", memo: "checkHealthy")]
public function checkHealthy(): void
{
    $this->logger->debug('掉线检测' . date('Y-m-d H:i:s', time()));
    $res = RequestClientTool::instance()->get($this->checkUrl . $this->token)->plain();
    if ($res['status']) {
        if ($res['data'] !== 'healthy') {
            $this->sendWarning();
        }
    }
}

/**
 * 发送警告
 * @return void
 */
private function sendWarning(): void
{
    $data = [
        "msgtype" => "text",
        "text" => [
```

```

        "content" => "已掉线请重新扫码" . date('Y-m-d H:i:s', time()) .
        ",点击扫码重新登录：" . $this->loginUrl . $this->token,
    ]
    ];
    $res = RequestClientTool::instance()->post($this->botUrl)-
    >setJson($data)->plain();

}

...

```

有部分代码的封装，逻辑仅供参考，接下来我又做了一个小工具，每天早上发送定时任务消息，提示摸鱼

![图片](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/759c916428fa467a9471e1b194885379~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=517&h=1110&s=58186&e=jpg&b=212121)

具体PHP代码如下，摸鱼接口也同时分享出来给有兴趣的同学们玩玩

```

...
/**
 * @param string $day
 * @return string
 */
public function moyuData(string $day = '10'): string
{
    $url = 'https://moyu.awsl.icu/api/moyu_json?day=10';
    $res = RequestClientTool::instance()->get($url)->plain();
    return $res['data'];
}

#[Crontab(rule: "0 10 * * *", memo: "moyu")]
public function moyu(): void
{
    $moyuData = $this->moyuData();
    $moyuData = str_replace('"', "", $moyuData);
    $moyuData = str_replace('\n', PHP_EOL, $moyuData);
    $data = [
        'to' => 'beyonds',
        'isRoom' => true,
        'data' => [
            'type' => 'text',

```

```
        'content' => $moyuData
    ]
];
$res = RequestClientTool::instance()->post($this->wxUrl .
'/webhook/msg/v2?token=' . $this->token)->setJson($data)->plain();
var_dump($res['message']??'');
$this->logger->info('摸鱼消息已发送' . date('Y-m-d H:i:s', time()).'
'.$res['message']??'');
}
...

```

浏览器访问这个 API地址也可以直接得到这个摸鱼提醒的文案：

```
[moyu.awsl.icu/api/moyu\_js...](http://cxyroad.com/
"https://moyu.awsl.icu/api/moyu_json?day=10")

```

最后  
==

这个项目只简单做到这里，不知道大家有什么更好的想法

如果觉得内容不错，欢迎点个赞~

你的支持是我更新的动力。

原文链接: <https://juejin.cn/post/7362752442410270759>