# PHP HTTP SDK 2.0

## 运行环境

```
PHP ≥ 5.2
插件版本 ≥ 2.0
```

## 目录结构

```
- CoolQ.class.php  酷Q API Class
- CoolQ.config.php 接口配置
- CoolQ.demo.php 存放处理函数，可在这里编写业务逻辑
- CoolQ.event.php 事件管理器，负责解析事件数据并回调给处理函数
- index.php 入口文件
```

## 配置方法

1. 将`[目录结构]`中提到的所有文件放到插件内设置的接口目录下
2. 打开`CoolQ.config.php`，根据 插件设置 进行修改配置
   配置参数说明：

| 参数 | 类型 | 默认值 | 说明 |
| --- | --- | --- | --- |
| `CoolQ_Config_url` | string | `127.0.0.1` | 酷Q所在服务器的IP地址，，暂只支持IPV4地址，可使用域名，如果使用IPV6地址，请在IP两头加上`[]`，如`[240c::6666]`。该值不为`NULL`时，视为启用动态交互。 |
| `CoolQ_Config_port` | int | `9999` | 动态交互的监听端口。 |
| `CoolQ_Config_key` | string | - | 校验数据所需的key值，该值需与插件设置保持一致。 |
| `CoolQ_Config_effectTime` | int | `30` | 数据有效时间，单位：秒。 |
| `CoolQ_Config_Receive_format` | string | `JSON` | 事件数据的格式，可选值：`KV`[即Key=Value的URL格式]。该值需与插件保持一致，当获取数据的方式改为`obtain`[即从插件主动获取]时，该值强制为`JSON`。 |
| `CoolQ_Config_Receive_from` | string | `push` | 事件数据的获取方式，可选值：`push`/插件推送、`obtain`/从插件获取(需启用动态交互功能)。 |
| `CoolQ_Config_Receive_limit` | int | `5` | 单次获取的事件数量，仅获取方式为`obtain`时有效。 |
| `CoolQ_Config_openDebug` | bool | `false` | 调试模式开关。 |

3. 测试`动态交互`配置是否正确 (可选)

- 在`CoolQ.demo.php`的非函数体内调用任意API，
  如调用获取运行信息API：`print_r($CQ->geRunningState);`。
  如果日志有信息输出，或PHP有输出相关数据，则表示配置正确。

# 执行流程

1. 运行`index.php`入口文件。
2. 运行`CoolQ.class.php`，声明API类：`CoolQ`。
3. 运行`CoolQ.config.php`，根据配置生成类变量。
4. 运行`CoolQ.demo.php`，声明与事件相关的处理函数。
5. 运行`CoolQ.event.php`，解析事件数据，并回调给处理函数。
6. 待处理函数执行完成后，回收相关资源。

# 编写业务逻辑

- 已在`CoolQ.demo.php`中针对目前的所有事件做了函数化处理方法，可在每个函数体内编写对应的业务逻辑。
- 如果没有特殊需求，请在`CoolQ.demo.php`中编写业务逻辑，不建议改动已经封装好的其余文件。
- 当处理函数运行完后，API类变量`$CQ`会被释放，如果仍需要调用API，请在`index.php`内注释掉`unset($CQ);`

# 注意事项

0. **除此之外都是注意事项**
1. 从插件获取数据时(`CoolQ_Config_Receive_from`设置为`obtain`)，需保证插件版本在`2.1.4.0`以上。
2. 如果不启用动态交互，则无法使用获取信息类API(如`取群列表[getGroupList]`)。
3. 如果启用校验数据(即`CoolQ_Config_key`不为`NULL`)，则需要将酷Q和PHP所在主机的 **时间以及时区** 设置一致，否则可能提示`数据过期`。PHP需要更改`php.ini`，或在`index.php`中执行`date_default_timezone_set('Asia/Shanghai');`
4. 如果需要自行获取插件推送的事件数据，不论数据格式`[CoolQ_Config_Receive_format]`设置`KV`还是`JSON`，都需要从POST数据中获取：`KV`：`$_POST`；`JSON`：`file_get_contents('php://input')`
5. 使用`obtain[从插件获取]`获得到的事件数据与`push[插件推送]`形式一致。

# 从旧版SDK升级

## 快速升级

- 用新的`CoolQ.class.php`覆盖旧文件，可以快速升级，
  如果之前因为要更好的适应业务需求而对旧版`CoolQ.class.php`进行修改，请自行修改新版文件。

## 全面升级

- 使用`[目录结构]`中提到的所有文件。
- 如果之前编写的业务逻辑代码均在`demo.php`，且保持下列结构，
  那么直接将`case x:`与`break;`之间的业务逻辑代码复制到`CoolQ.demo.php`中的对应处理函数内。
  如果涉及到使用 包含整个事件数据 的变量`$array`，需要替换成函数参数中的`$info`。

```
switch($array['type']) {
  case 1:
    //收到私聊信息
    /** 业务逻辑代码 **/
    break;

  case 2:
    //收到群信息
    /** 业务逻辑代码 **/
    break;
}
```