

AIUI 语义后处理历险记

目前讯飞 [AIUI 开放平台](#)的【智能硬件解决方案】和【移动端解决方案】已经上线有一段时间了。



作为 AIUI 平台的老司机我已经知道，只要用户购买了我开发的 AIUI 硬件产品或者安装了我开发的 APP 应用，我就可以使用上述两个方案很方便的理解用户说的话（指令），然后将语义理解结果返回到我的设备或者 APP 上。

就这么简单？当然不是！重点来了！

除了可以把结果返回到我的终端设备和 APP 应用以外，还可以返回到我的服务器上！

服务器拿到结果的话，可操作性就太强了！

我可以用这个结果来控制服务器的行为，也可以再把结果转发给某台 PC，甚至把结果进行一系列的个性化处理后在转发到手机上！等等各种姿势.....

作为程序界老司机的你是不是一下打开的脑洞？

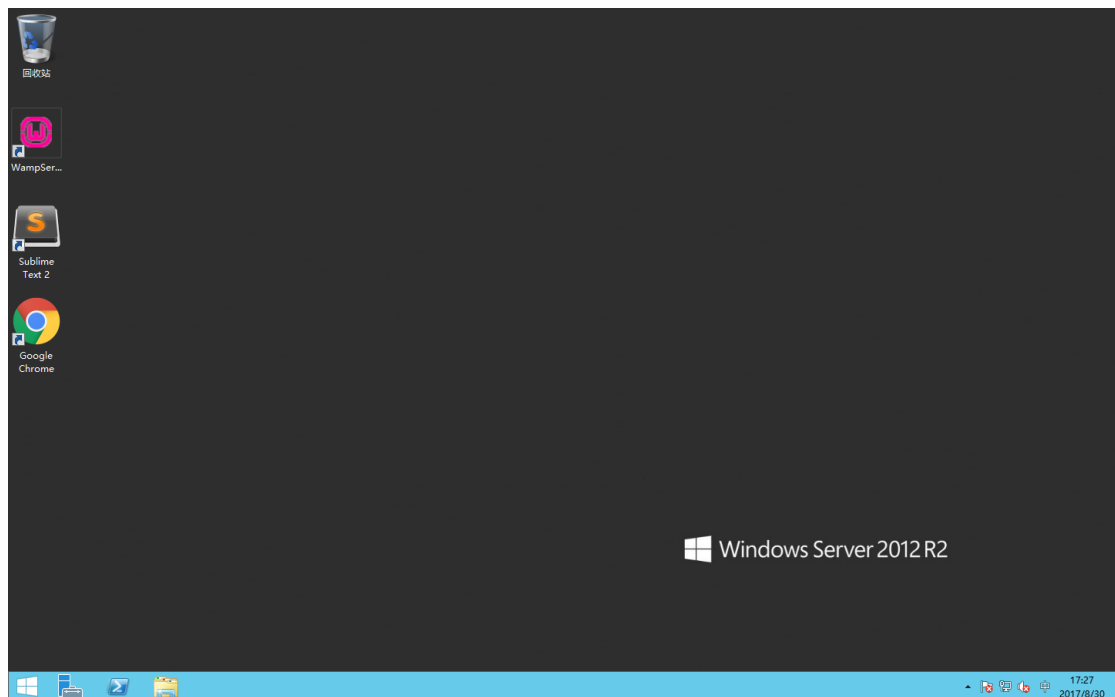
先关上脑洞，我们今天来看一下如何把这个后处理调通，让我们的服务器收到语义结果。

首先我们需要一台服务器，虽然是句废话，但是对于我来说还的确是个问题，因为我没有服务器，更没有外网 IP。怎么办？哈哈哈，现在这个世道，没钱也能办很多事情。度娘告诉我有许多家云服务器提供商，有的甚至还有免费体验活动，吆西，果断申请了一台云服务器。当然，身在土豪公司的土豪的你，请跳过上面的步骤。

远程连接进我的服务器，看着空荡荡的桌面，我知道我马上要大干一场了！

第一步装 Wamp（提供服务器和 PHP 环境）、Sublime（写代码用的）、Chrome（调试查

资料用的),



Wamp 装好后，直接打开 index.php，卷起袖子！代码撸起来！

原 index.php 的内容统统删掉，参考了 <http://aiui.xfyun.cn/info/docs> 处的后处理协议，发现服务器只需要实现两部分功能即可：

- 1、接收讯飞服务器发来的 URL 验证消息，校验后回发 echo；
- 2、接收讯飞服务器发来的语义结果。

其中第一个是 get 请求，第二个是 post 请求，看到这里的老司机我的脑海里已经构思好所有的代码了！So easy！

Talk is cheap！Show you my code！

```
<?php
    if(getIsPostRequest()) {
        $input=file_get_contents("php://input");
        file_put_contents("log.txt", date('H:i:s')." ".$input."\r\n",FILE_APPEND);
    } else {
        //回传后处理 token，token 在官网上有
        echo sha1("99c51f20e3a8dxxx");
        foreach ($_GET as $key=>$value)
        {
            file_put_contents("log.txt", date('H:i:s')." ".$_GET: Key: $key; Value: $value."\r\n", FILE_APPEND);
        }
    }
    function getIsPostRequest()
    {
        Return isset($_SERVER['REQUEST_METHOD'])&& !strcasecmp($_SERVER['REQUEST_METHOD'],'POST');
    }
?>
```

代码写完，Wamp 服务器启起来！下面就可以进入 AIUI 开放平台来配置相应的后处理参数了。



点击上图中的【后处理】页签，配置后处理 URL，我的云服务器的 ip 地址是：116.196.105.xxx（作为老司机的我怎么会把完整的 ip 给你的呢？哈哈！），所以这里的填写的 URL 就是 <http://116.196.105.xxx/index.php>。另外后面的后处理 token 是需要配置的服务端程序里面的，具体参考上面的 php 代码。为了快速调通，后面勾选不加密。填完之后，不出意外的话就可以验证成功了，此时【后处理】旁边的√也自动打上了。



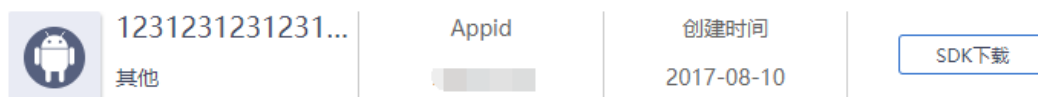
此时，好奇心驱使我偷偷瞄了一眼服务器保存的日志 log.txt：

```
16:18:03 _GET: Key: signature; Value: f5c32f598623e65f3a0cb98250a3d72013f8642f
16:18:03 _GET: Key: timestamp; Value: 1471420340
16:18:03 _GET: Key: rand; Value: cc
```

发现，的确收到了讯飞服务器发来的 url 验证消息，消息包含的参数有 signature、rand、timestamp，详细介绍参考[协议文档的链接](#)。

验证通过后，我就可以进行语义理解测试啦！

当然测试之前下载我的这个 Android 应用的 SDK：



然后在给它配置若干语义场景：



然后将下载的 SDK 中的 Android 的 demo 安装到我的测试机上 (demo 如何运行请参考 SDK 包中的相关文档)：



点击左图中的【立即体验语义理解】按钮，进入右图的界面。点击【语音语义】按钮后随便说点啥（比如：你好啊），然后就有语义结果返回了：



哈哈，手机上已经收到语义结果了，那我的服务器呢？此时，好奇心再次驱使我瞄了一眼服务端的日志 log.txt：

```
{ "MsgId": "cid6f1ced4a@ch00270d01dc320100171", "CreateTime": 1504088116, "AppId": "98bce98", "UserId": "d5725172476", "SessionParams": "eyJkc3JlIjoic2RrIiwiaWZHRzljoiMSIsImR0eXBBIjoiYXVkaW8iLCJtc2MubGF0IjoieMzMzMmNDI0IiwibWwXNjLmxuZyl6IjExNy4xMzg4NSIsInNjZW5lIjoibWFpbilSnNjaXR5IjoieY2giLCJzaWQiOiIjaWQ2ZjFjZWQ0YUBjaDAwMjcwZD AxZGMzMjAxMDAxBnYslN0bWkljoiYXVkaW8tMjMiLCJ2ZXJfdHlwZSI6Im1vYmIsZV9waG9uZSIsIndha2VfaWQiOiIxNTA0MDg4MTAzODg2NWwRMOTfkZWVkMDRkZTM5ZSJ9", "UserParams": "", "FromSub": "kc", "Msg": { "ContentType": "json", "Type": "text", "Content": "eyJpbmRIbnQiOnsiYW5zd2VyIjp7ImFuc3dlclR5cGUiOiIvcGVuUUEiLCJlbW90aW9uljoiZGVmYXVs dClSl nF1ZXN0aW9uljp7InF1ZXN0aW9uljoi5L2g5aW95ZWkliwicXVlc3Rpb25fd3MiOiIkva Dl pb0vVkkvLyAg5ZWKL1VFLy8ifSwidGV4dCI6luS9oOWIve+8jOWPiOingemdouS6huecn +W8gOW/g+OAgiSl nRvcGljljozMJE4NDE5OTUwNzgzNjMwMC4wLCJ0b3BpY0ElIjozMJE 4NDE5OTUwNzgzNjMwMC4wLCJ0eXBBIjoiVCJ9LCJtYW5faW50dil6IlIsIm5vX25sdV9yZXN 1bHQiOiJAslm9wZXJhdGlvbil6IkFOU1dFUilSl nJjowLCJzZXJ2aWNIIjoib3BlblFBliwic3Rh dH VzljowLCJ0ZXh0Ijoie5L2g5aW95ZWkliwidXVpZCI6ImF0bjAwMWQ5NTE2QGNoYmUwNzB kMDFkYz M0NmYyYTAxliwic2kljoiY2lkNmYxY2VkNGFAY2gwMDI3MGQwMWRjMzl wMT AwMTcfX0=" } }
```

```
{ "MsgId": "cid6f1ced4a@ch00270d01dc320100172", "CreateTime": 1504088116, "Appld": "598bce98", "UserId": "d5725172476", "SessionParams": "eyJkc3JlIjoic2RrIiwiaHRzljoiMSIsImR0eXBBIjoiYXVkaW8iLCJtc2MubGF0IjoieMzMzNDI0IiwibWwXNjLmxuZyl6IjExNy4xMzg4NSIsInNjZW5lIjoibWFpbilsInNjaXR5IjoieY2giLCJzaWQiOiIjaWQ2ZjFjZWQ0YUBjaDAwMjcwZD AxZGMzMmAxMDAxNyIsInN0bWkljoiYXVkaW8tMjMiLCJ2ZXJfdHlwZSI6Im1vYmlsZV9waG9uZSIsIndha2VfaWQiOiIxNTA0MDg4MTAzODg2NWRRmOTFkZWVkMDRkZTM5ZSJ9", "UserParams": "", "FromSub": "kc", "Msg": { "ContentType": "json", "Type": "text", "Content": "eyJpbmRIbnQiOnt9fQ==" } }
```

格式说明

参数	描述
MsgId	消息id, 字符串类型
CreateTime	消息创建时间, 整型
AppId	开发者应用Id, 字符串类型
UserId	AIUI唯一用户标注, 字符串类型
UserParams	开发者自定义参数, Base64格式字符串
FromSub	上游业务类型, 字符串 (lat为听写结果, kc为语义结果)
Msg	消息内容, json object参考Msg消息内容格式
SessionParams	本次会话交互参数, Base64格式字符串, 解码后为json格式

```

文本内容
{
  "Type": "text",
  "ContentType": "Json",
  "Content": "eyJzbiI6MiwiYm9uRydwUStmJnIjowLj1ZCi6McWid3Mi01t7ImJnIjowLjJjdjI6W3sic2Mi0jAsIncioiLv18ifV19XX0="
}

```

参数	描述
Type	text
ContentType	内容格式, Json plain xml
Content	Base64内容字符串

原来真正的语义结果就在 Msg->Content 字段里面：

eyJpbnRlbnQiOnsiYW5zd2Vyljlp7lmFuc3dlcmlR5cGUiOiJvcGVuUUEiLCJlbW90aW9uljoiZGVmYXVsdCIsInF1ZXN0aW9uljlp7lnF1ZXN0aW9uljoi5L2g5aW95ZWkwiicXVlc3Rpb25fd3MiOiLkvaDlpb0vVkkvLyAg5ZWKL1VFly8ifSwidGV4dCI6IuS9oOWIve+8jOWPiOingemdouS6huecn+W8gOW/g+OAgilsInRvcGlljjozMjE4NDE5OTUwNzgzNjMwMC4wLlCj0b3BpY0IEljozMjE4NDE5OTUwNzgzNjMwMC4wLlCj0eXBlljoiVCJ9LCJtYW5faW50diI6IuIsIm5vX25sdV9yZXN1bHQiOiJAsIm9wZXJhdGlvbil6IkFOU1dFUilSnJljlowlCJzZXJ2aWNlljoiib3BlbIFBliwic3RhhdHVzljowlCj0ZXh0Ijoi5L2g5aW95ZWkwiidXVpZC1mF0bjAwMwQ5NTE2QGNoYmUwNzBkMDFkYzMOmNyYTAXliwic2lkloiY2IkNmYxY2VKNzY2gWMDI3MGQwMWRjMzlwMTAwMTcifX0=

果然，拿去做了 base64 编码后得到了我要的 json 语义结果：



至此，AIUI 语义后处理功能算是调通啦！

哈哈，我和我的服务器都笑了.gif。

备注：

- 1、上面讲的移动端的解决方案，其实智能硬件解决方案里也有后处理功能，调试步骤完成一样；
- 2、这里为了简单易懂，部分步骤被简化或者省去了，比如服务消息没有加密、url 验证消息没有验证就直接回发 token、语义结果的提取和 base64 解码也没有在服务端代码里实现等等。更详细的实现和介绍请参考[协议文档](#)。
- 3、细心的老司机应该发现了，手机上发起一次语义请求，讯飞服务器发来的语义结果消息并非一条，只有其中一条包含了完整的语义结果。