

## 后台交互（php）

### 1、测试连接到 url

```
$next = "error";//告诉机器人 nextStep 做什么
$answer = 'let it go';//返回一个信息给机器人，在用户和机器人对话后，机器人会告诉用户这个信息
//当一个 php 文件中只有这两句时，不管 action 是怎么设置和写法，只要 action 是正确的，能够正常运行的，并且正确连接到 URL，那么都会必然得到返回的信息。如果没有返回得到信息，那么有可能是 action 有误或者没更新成功，或者 url 有误
```

后台数据交互大致思路是：

- 1）、解析 post→判断答案→告诉机器人下一个 nextStep（不做存储）
- 2）、配置数据库信息连接数据库——解析 post（action\_name）——判断 switch（action\_name）→解析 post 答案→存储→判断答案→告诉机器人下一个 nextStep（做存储）

### 2、简单的数据交互步骤（不做存储）：

- 1）、解析 post 提交过来的答案中所有的回答
- 2）、判断哪些答案是正确答案
- 3）、告诉机器人下一个该回答的问题，依次判断所有的答案，如果答案不正确，那么反馈一个信息给机器人，也就是 \$next 指向前台 action 话题单元中的 nextStep，当所有答案都正确的时候，返回一个回答完所有问题的信号，\$answer 是机器人根据用户对话，再文本框中输出相应的信息。

示例：

```
$next = "error";
// step1 解析 post 提交过来的答案中所有回答
$name=from_post('name', 'null');
$age=from_post('age', 'null');
$high=from_post('high', 'null');

//step2 判断哪些是正确答案
$nameflag=false;
if($name!='null'){
    $nameflag=true;
}
$ageflag=false;
if($age!='null'){
    $ageflag=true;
}
$highflag=false;
if($high!='null'){
```

```

    $highflag=true;
}
//step3 告诉机器人下一个该回答的问题, 依次判断 name age high
if($nameflag==false) {
    $next='namecollect';
}else if($ageflag==false){
    $next='agecollect';
}else if($highflag==false){
    $next='highcollect';
}else{
    $next='finishall';
}
// Echo the post form
$answer = 'let it go';

```

比如在以上的示例中, 声明一个\$next 变量,指向的是前台 action 的 nextStep,如果答案不正确, 则\$next='namecollect';会告诉机器人下一步是执行:

process=nextStep=namecollect|name\_httppost=active|  
激活 name\_httppost 这个 action,重新询问收集答案

### 3、简单数据交互步骤（做数据存储）：

- 1）、配置数据库基本信息和连接数据库
- 2）、解析 action\_name, switch 条件判断, 如果找到匹配的 action\_name, 进入 action\_httppost 进行处理
- 3）、在 action\_httppost 中解析 post 提交过来的答案中所有的回答
- 4）、通过 chat\_key 从 mysql 获取历史记录
- 5）、判断有没有 chat\_key,如果没有 chat\_key,那么执行 insert sql 语句存储到数据库中, 如果已经存在 chat\_key, 那么执行 update sql 语句更新数据库中的信息
- 6）、判断回答是否正确, 告诉机器人下一个该回答的问题, 依次判断所有的答案, 如果答案不正确, 那么反馈一个信息给机器人, 也就是\$next 指向前台 action 话题单元中的 nextStep, 当所有答案都正确的时候, 返回一个回答完所有问题的信号, \$answer 是机器人根据用户对话, 再文本框中输出相应的信息。

注意: 当前端页面 actions 运行完一个模块后, 如连接下级出现错误, 就会显示 start answer

这是分模块化 action 处理, 也就是说, 假如一个模块里有 10 个问题, 那么 10 个问题都回答正确后跳转到下一个模块,可以一次性存储, 相对于在灵活性说, 一旦出现改动, 可能需要改动的地方比较多。

示例:

```

<?php
$next = "error";//声明变量并赋值
$answer = "start answer";
//将监听日志保存到文件中, 不是必要, 为了更加方便测试调试
$post_content = print_form($_POST);
file_put_contents("/home/ubuntu/test2.txt", $post_content, FILE_APPEND);

```

```

file_put_contents("/home/ubuntu/test2.txt", "\n", FILE_APPEND);
//配置数据库基本信息 如果是需要存储到数据库，这些数据库的配置与连接设置是必须，
define('DB_HOST', 'localhost');
define('DB_USER', 'cp');
define('DB_PWD', '123456');
define('DB_NAME', 'cp_vt');
//连接 mysql 数据库
$connect = mysql_connect(DB_HOST, DB_USER, DB_PWD) or die('数据库连接失败，
错误信息: ' . mysql_error());
mysql_select_db(DB_NAME, $connect) or die('数据库连接错误，错误信息: ' .
mysql_error());
mysql_query("SET NAMES 'utf8'", $connect);

//从 post 提交获取参数 (action_http post) : action_name
$action_name = from_post('action_name', 'null');
//判断 action_http post，进行 action 逻辑处理，返回信息给机器人，在 action 中的
nextStep 根据反馈的信息，激活下个 action,实现 action 间的跳转
switch ($action_name) { //switch 条件判断
case 'main_http post':
$name = from_post('name', 'null'); //假如有 10 个问题就获取 10 个参数（参数名不
同）
// 通过 chat_key 从 mysql 获取历史记录。
$query = sprintf("SELECT * FROM `USER_INFO` WHERE `chat_key` = '%s'",
$chat_key);
$result = mysql_query($query) or die('sql 语句错误: ' . mysql_error());
$data = mysql_fetch_array($result);
//如果 chat_key 没有记录，则执行插入数据操作，否则执行更新操作
if ($data == false) {
$sql = "INSERT INTO `USER_INFO` (`chat_key`, `name`) VALUES
('$chat_key', '$name')";
if (!mysql_query($sql, $connect)) {
$answer = "insert fail" . mysql_error() . " " . $sql;
} else {
$answer = "insert success";
}
} else {
$name = from_post('name', 'null');
if (($name == 'null') && ($data['name'] != 'null')) {
$name = $data['name'];
}
$sql = "UPDATE `USER_INFO` SET `chat_key` = '$chat_key', `name` = '$name' WHERE
`chat_key` = '$chat_key'";
if (!mysql_query($sql, $connect)) {
$answer = "update fail " . mysql_error() . " " . $sql;
}
}
}

```

```

} else {
$answer = "update success";
}
}
//告诉机器人下一步 nextStep
if ($name == 'null') {
$next = 'namecollect';
$answer = $next;
} else {
$next = 'finish';
$answer = $next;
}
// $next = 'finish'; //假如在此处，表示是无论上面执行结果怎么样最终将执行下一个名为 finish 的 nextStep
// $answer = $n

```

```

ext; //假如在此处，表示是无论上面执行结果怎么样都会返回一个信息给机器人，体现在对话中
break; //执行完当前 action，break 程序中中断
case 'two_httppost': //当前台 process=nextStep=finish|two_httppost=active|
这时候激活 two_httppost，跳转到 two_httppost 继续收集信息，post 提交到 php 中会再次获取 action_name，switch 条件判断 two_httppost，找到 two_httppost 时，执行 two_httppost 中的逻辑代码，反馈信息给机器人
break;
case 'three_httppost':
break;
}
?>

```

#### 4、数据交互步骤（做数据存储）：

- 1）、配置数据库基本信息和连接数据库
- 2）、解析 action\_name，switch 条件判断，如果找到匹配的 action\_name，进入 action\_httppost 进行处理
- 3）、在 action\_httppost 中解析 post 提交过来的答案中所有的回答
- 4）、通过 chat\_key 从 mysql 获取历史记录
- 5）、判断有没有 chat\_key，如果没有 chat\_key，那么执行 insert sql 语句存储到数据库中，如果已经存在 chat\_key，那么执行 update sql 语句更新数据库中的信息
- 6）、判断回答是否正确，告诉机器人下一个该回答的问题，依次判断所有的答案，如果答案不正确，那么反馈一个信息给机器人，也就是 \$next 指向前台 action 话题单元中的 nextStep，当所有答案都正确的时候，返回一个回答完所有问题的信号，\$answer 是机器人根据用户对话，再文本框中输出相应的信息。

注意：关于筛选收集有效信息操作，如果数据库中需要存储的是筛选后的信息，可以在解析 `post` 后，进行逻辑处理，判断提交过来的答案是否是正确答案，然后存储/更新到数据库中，如果不是这正确答案，直接不执行存储并反馈下一个 `nextStep`，`break` 中断；或者先存储到数据中，再判断是否是正确答案，如果不是正确答案就再次询问，重新收集有效信息，这个需要数据库的数据类型能够匹配，如果不一致不能存储到数据库中

这是单问题化 `action` 处理，也就是说，假如一个模块里有 10 个问题，那么 10 个问题都分开写 `action`，回答正确后跳转到下一个问题 `action`，如果错误，返回错误的 `action` 继续换个询问方式收集信息，便于收集筛选有效信息，相对于在灵活性说，一旦出现改动，需要改动的地方比少。

示例：

```
<?php
$next = "error";
$answer = "start answer";
// save post
$post_content = print_form($_POST);
file_put_contents("/home/ubuntu/test2.txt", $post_content, FILE_APPEND);
file_put_contents("/home/ubuntu/test2.txt", "\n", FILE_APPEND);
// connect mysql
define('DB_HOST', 'localhost');
define('DB_USER', 'cp');
define('DB_PWD', '123456');
define('DB_NAME', 'cp_vt');
$connect = mysql_connect(DB_HOST, DB_USER, DB_PWD) or die('数据库连接失败，
错误信息： ' . mysql_error());
mysql_select_db(DB_NAME, $connect) or die('数据库连接错误，错误信息： ' .
mysql_error());
mysql_query("SET NAMES 'utf8'", $connect);

$action_name = from_post('action_name', 'null');
switch ($action_name) {
// //项目档案
//回答项目名称，只要回答就通过，不做筛选
case 'information_httpost':
$name = from_post('name', 'null'); //项目名称
$query = sprintf("SELECT * FROM `TEAM_INFORMATION` WHERE `chat_key` = '%s'",
$chat_key);
$result = mysql_query($query) or die('sql 语句错误： ' . mysql_error());
$data = mysql_fetch_array($result);
if ($data == false) {
$sql = "INSERT INTO `TEAM_INFORMATION` (`chat_key`, `name`)
VALUES('$chat_key', '$name')";
```

```

if (!mysql_query($sql, $connect)) {
$answer = "insert fail" . mysql_error() . " " . $sql;
} else {
$answer = "insert success";
}
} else {
$name = from_post('name', 'null'); //项目名称
if (($name == 'null') && ($data['name'] != 'null')) {
$name = $data['name'];
}
$sql = "UPDATE `TEAM_INFORMATION` SET
`chat_key`='$chat_key',`name`='$name' WHERE `chat_key`='$chat_key'";
if (!mysql_query($sql, $connect)) {
$answer = "update fail " . mysql_error() . " " . $sql;
} else {
$answer = "update success";
}
}
if (strlen($name) >= 1) {
$next = 'finishname';
$answer = $next;
} else {
$next = 'namecwu';
$answer = $next;
}
break;
//项目概括回答错误时再次询问 action 存在 chat_key
case 'namecwu_httppost':
$name = from_post('name', 'null'); //项目名称
$query = sprintf("SELECT * FROM `TEAM_INFORMATION` WHERE `chat_key` = '%s'",
$chat_key);
$result = mysql_query($query) or die('sql 语句错误: ' . mysql_error());
$data = mysql_fetch_array($result);
if ($data == false) {
$sql = "INSERT INTO `TEAM_INFORMATION` (`chat_key`,`name`)
VALUES('$chat_key','$name')";
if (!mysql_query($sql, $connect)) {
$answer = "insert fail" . mysql_error() . " " . $sql;
} else {
$answer = "insert success";
}
} else {
$name = from_post('name', 'null'); //项目名称
if (($name == 'null') && ($data['name'] != 'null')) {

```

```

$name = $data['name'];
}
$sql = "UPDATE `TEAM_INFORMATION` SET
`chat_key`='$chat_key',`name`='$name' WHERE `chat_key`='$chat_key'";
if (!mysql_query($sql, $connect)) {
$answer = "update fail " . mysql_error() . " " . $sql;
} else {
$answer = "update success";
}
}
if (strlen($name) >= 1) {
$next = 'finishname';
$answer = $next;
} else {
$next = 'namecwu';
$answer = $next;
}
break;
//请用一句话概括您的项目,10个字符或者以上
case 'founder_httpost':
$founder = from_post('founder', 'null'); //项目名称
$query = sprintf("SELECT * FROM `TEAM_INFORMATION` WHERE `chat_key` = '%s'",
$chat_key);
$result = mysql_query($query) or die('sql 语句错误: ' . mysql_error());
$data = mysql_fetch_array($result);
if ($data == false) {
$sql = "INSERT INTO `TEAM_INFORMATION`(`chat_key`,`founder`)
VALUES('$chat_key','$founder')";
if (!mysql_query($sql, $connect)) {
$answer = "insert fail" . mysql_error() . " " . $sql;
} else {
$answer = "insert success";
}
} else {
$founder = from_post('founder', 'null'); //项目名称
if (($founder == 'null') && ($data['founder'] != 'null')) {
$founder = $data['founder'];
}
$sql = "UPDATE `TEAM_INFORMATION` SET
`chat_key`='$chat_key',`founder`='$founder' WHERE
`chat_key`='$chat_key'";
if (!mysql_query($sql, $connect)) {
$answer = "update fail " . mysql_error() . " " . $sql;
} else {

```

```

$answer = "update success";
}
}
if (strlen($founder) >= 30) {
$next = 'finishfounder';
$answer = $next;
} else {
$next = 'foundercwu';
$answer = $next;
}
break;
//项目概括回答错误时再次询问 action 存在 chat_key
case 'foundercwu_httppost':
$founder = from_post('founder', 'null'); //项目名称
$query = sprintf("SELECT * FROM `TEAM_INFORMATION` WHERE `chat_key` = '%s'",
$chat_key);
$result = mysql_query($query) or die('sql 语句错误: ' . mysql_error());
$data = mysql_fetch_array($result);
if ($data == false) {
$sql = "INSERT INTO `TEAM_INFORMATION` (`chat_key`, `founder`)
VALUES('$chat_key', '$founder')";
if (!mysql_query($sql, $connect)) {
$answer = "insert fail " . mysql_error() . " " . $sql;
} else {
$answer = "insert success";
}
} else {
$founder = from_post('founder', 'null'); //项目名称
if (($founder == 'null') && ($data['founder'] != 'null')) {
$founder = $data['founder'];
}
$sql = "UPDATE `TEAM_INFORMATION` SET
`chat_key`='$chat_key', `founder`='$founder' WHERE
`chat_key`='$chat_key'";
if (!mysql_query($sql, $connect)) {
$answer = "update fail " . mysql_error() . " " . $sql;
} else {
$answer = "update success";
}
}
}
if (strlen($founder) >= 30) {
$next = 'finishfounder';
$answer = $next;
} else {

```



```

$next = 'foundercwu';
$answer = $next;
}
break;
}
?>

```

##### 5、数据交互简单逻辑处理：

1)、一般的逻辑处理为大部分使用 if 语句      if...else 语句      if...else if ...else 语句      switch 语句，

2)、一般数据类型转换，需要进行调用函数，返回一个和数据库同类型的值。比如日期转换，首先需要判断 post 提交过来的格式去空、是不是合法的日期格式，（Y/m/d 是自己设定前台 action 输入的模式），然后再进行 if 判断，如果为 true，则表示这是一个合法的日期格式，然后就可以进行调用 date 函数转换日期,就可以存储到数据库中（数据类型为 date 类型）

```

//检查一个字符串是否是合法的日期格式
function checkDateTime($str, $format="Y/m/d"){
    $unixTime = strtotime($str);
    $checkDate = date($format, $unixTime);
    if($checkDate == $str)
        return true;
    else
        return false;
}

$conversion_date= from_post('conversion_date', 'null'); //获取日期字符串
$conversion_date=trim($conversion_date);//去掉前后空格

if (checkDateTime($conversion_date )) { //如果是合法的日期格式，true 进入日期转换
    //日期字符串转换成日期类型
    $conversion_date = date('Y/m/d', strtotime($conversion_date));
}

```