

前言

- 获取数据完整地址的样子：
- 根目录/index.php?m=Home&c=User&a=login
 - 王博鑫注：我不是很清楚目前的地址，我目测是
../../index.php?m=Home&c=User&a=login
这个样子。真实情况估计还得你看下前端的代码。
上面那个地址里 c=后接是控制器名，a=后接方法名。方法名就是函数名，控制器名就是类名。
- 服务器端返回的结果如果是一个就是直接的字符串，如果返回2个或2个以上的就用 json。true、false、error 这样的返回标记，返回的不是 json，就是一个字符串
- @param [GET]:int state；这样标记的代表需要在地址里使用 get，然后用 post 正常请求其他参数
- 文件的上传目录是在 Public/这个底下，服务器会返回图片所在路径，比如：/Uploads/20141229/54a036b9dc8ff.png
也就是实际文件在 Public/Uploads/20141229/54a036b9dc8ff.png 这个路径上。
- 所有的接口都是通过 http 完成，默认都是用 post 传递参数（但是调试阶段为了方便调试 get 也可以用，但是如果上线的话会封 get，所以除非明确指出的参数外，其余参数一定要用 post）

调试：类名 **Index**

方法名：index（因为是默认的 index，所以直接访问根目录就会出来）

这个里面是一些调试用的接口，可以方便测试

用户部分：类名 User

```
/**
 * 用户登录函数
 * @param username,password
 * @return true,false
 */
public function login()
```

```
/**
 * 用户退出函数
 * @return true,false
 */
public function logout()//安全退出
```

```
/**
 * 注册函数
 * @param string $userName 用户名
 * @param string $userPassword 密码
 * @return true,false,error
 */
public function sign()
```

```
/**
 * 获得用户uid
 * @return int uid
 */
public function getUserId()
```

```
/**
 * 得到用户的信息
 * @param int uid
 * @return json 用户信息，为数据库中的一行
 * 如: {"uid":"1","name":"wbx@wbx.com","pwd":"wbx","realName":"","logoPic":"","phone":"","address":""}
 * @return false "" 失败
 * @return error "" uid非法
 */
public function getUserInfo()
```

```
/**
 * 修改用户信息（没有修改头像）
 * @param json 用户信息，为数据库中的一行；不能修改name
 * 如: {"uid":"1","pwd":"wbx","realName":"","logoPic":"","phone":"","address":""}
 * @return bool "" 是否成功
 * @return error "" uid非法
 */
public function editUserInfo()
```

```

/**
 * 设置用户信息（必须同时上传头像）
 * @param 表单 用户信息，为数据库中的一行；不能修改name
 * 如: {"uid":"1","pwd":"wbx","realName":"","logoPic":"","phone":"","address":""}
 * @return bool "" 是否成功
 * @return error "" uid非法
 */
public function setUserInfo()

```

```

/**
 * 设置用户头像（必须上传头像）
 * @param int uid
 * @param file 头像文件
 * @return bool "" 是否成功
 * @return error "" uid非法
 */
public function setUserInfoLogo()

```

```

/**
 * 关注toUid用户
 * @param int toUid 要关注的用户的uid
 * @return "error_over" 关注关系已存在
 * @return "error_sql" sql语句执行错误
 * @return "true" 成功
 */
public function follow()

```

```

/**
 * 取消关注toUid用户
 * @param int toUid 要取消关注的用户的uid
 * @return "error_no" 关注关系不存在
 * @return "error_sql" sql语句执行错误
 * @return "true" 成功
 */
public function unfollow()

```

```

/**
 * 是否关注了toUid用户
 * @param int toUid 要查询的用户uid
 * @return "true,false" 是, 否
 * @return "error_sql" sql语句执行错误
 */
public function isFollow()

```

```
/**
 * 查询toUid的用户的关注用户列表。
 * @param int toUid 要查询的用户的uid
 * @return json 关注的uid的列表, 如["2","3","4","5"], 代表关注了uid为2、3、4、5的用户。
 * @return "null" 该用户没有关注任何人
 * @return "error_sql" 查询出错
 */
public function userFollowList()
```

帖子部分

```
/**
 * 查询某个活动
 * @param int qid
 * @return error 错误
 * @return ReturnJson 返回数据
 */
public function queryOne()
```

```
/**
 * 查询所有未完成的活动;
 * @param int page 当前的页数
 * @return error "" page过大或没有此类活动
 * @return json 活动内容, 形如:
 * {
 *     "pageTotalNum":2,//总页数
 *     "content":[ //具体的内容
 *         {},//一个活动
 *         {}
 *     ]
 * }
 */
public function queryAll()
```

```
/**
 * 查询私戳用户列表
 * @param int page 当前的页数
 * @param int userPageNum 每页用户数量,默认为6
 * @return error "" page过大或没有此类用户
 * @return json 用户信息, 形如:
 * {
 *     "pageTotalNum":2,//总页数
 *     "content":[ //具体的内容
 *         {},//一个用户
 *         {}
 *     ]
 * }
 */
public function queryUser()
```

```
/**
 * 创建一个活动
 * @param 多个参数，名称参考returnJson,但是形式是post
 * @return true "" 成功
 *         其他任何东西 "" 失败
 *         error "" 非法操作
 */
public function create()
```

```
/**
 * 修改一个活动
 * @param 多个参数，名称参考returnJson,但是形式是post;
 * @param int mode 模式，为1修改活动（不改logo），为2是修改活动（改logo）
 * @return true "" 成功
 *         其他任何东西 "" 失败
 *         error "" 非法操作
 */
public function edit()
```

```
/**
 * 删除一个活动
 * @param int qid
 * @return bool "" 是否成功
 */
public function delete()
```

```
/**
 * 给当前用户的当前活动添加一个标签（作为组织者进行）
 * @param int qid
 * @param string tag 新增的标签
 * @return bool "" 是否成功
 */
public function addTag()
```

```
/**
 * 当前用户的当前活动修改标签（作为组织者进行）
 * @param int qid
 * @param ReturnJson_tag tag 全部的标签列表
 * @return bool "" 是否成功
 * @return error "" 出错
 */
public function editTag()
```

```

/**
 * 赞一下
 * 做了两件事：
 *     1.给question表的qid行的zan+1
 *     2.给user表的uid行的zanTable添加一个qid
 * @param int qid
 * @return bool "" 是否成功
 */
public function zan()

```

```

/**
 * 回复帖子id为qid的帖子
 * @param int qid
 * @param string content 回复的内容
 * @return "true,false"
 */
public function reply()

```

```

/**
 * 返回一个帖子的所有回复
 * @param int qid 要查询的帖子的qid
 * @return json 格式如下：
 * [{"qid":"1","uid":"1","qid":"1","createTime":"2015-08-10 17:26:24","content":"asdasdas","zan":"0"},{"c
 */
public function queryReply()

```

```

/**
 * 收藏帖子id为qid的帖子
 * @param int qid
 * @return "true,false"
 * @return "error_over" 关注关系已存在
 * @return "error_sql" sql语句执行错误
 */
public function collectq()

```

```

/**
 * 查询toUid的用户的关注问题列表。
 * @param int toUid 要查询的用户的uid
 * @return json 关注的qid的列表，如["2","3"]，代表关注了qid为2、3的帖子。
 * @return "null" 该用户没有关注任何帖子
 * @return "error_sql" 查询出错
 */
public function questionFollowList()

```