

# 统计系统开发文档

## 前言

该系统基于ci3.0版本开发，还要懂一点Linux。开发中遇到问题可以参考[官方文档](#)，关于服务器环境等问题，要参考《服务器环境配置说明》

## 数据库操作

- 数据库配置

查看 `app/config/database.php` 文件，其实里面定义的事一个二维数组，键名就是数据库连接的名称，值时数据的具体配置。

```
$db['sdk'] = array(
    'dsn'      => '',
    'hostname' => '服务器ip(:端口)', // 如果数据库端口非3306，那么这里要
    指定端口哦
    'username' => '用户名',
    'password' => '数据库密码',
    'database' => '数据库名称',
    'dbdriver' => 'mysqli',
    // 'dbdriver' => 'pdo',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt'  => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);
```

- 连接数据库

要连接数据库时很简单的，看代码吧，详细的可以看[文档](#)  
实例化数据库连接，第一个参数是数据库连接的名称，比如上面的代码定义的sdk，第二个参数传TRUE,表示要返回连接ID，返回连接id的话，就可以在一个模型里面连接不同的数据了。

可以参考 `model/base_model.php` 文件

```
$db_sdk = $this->load->database('sdk', TRUE);  
$db_default = $this->load->database('default', TRUE);
```

- 执行查询

也是看一段代码,[点我看文档](#)

```
$sql = "select * from u_login limit 1";  
$query = $db_sdk->query($sql);  
// 这里判断数据库查询是否成功  
if ($query) {  
    // 获取数据，的到的是一个二维数组哦  
    $data_array = $query->result_array();  
    // 这样就得到一个对象了  
    $data_obj = $query->result();  
    // 释放结果集，该方法释放掉查询结果所占的内存，并删除结果的资源标识。  
    // 通常来说， PHP 会在脚本执行结束后自动释放内存。  
    // 但是，如果你在某个脚本中执行大量的查询，你可能需要在每次查询之后释放  
    // 掉查询结果，以此来降低内存消耗  
    $query->free_result();  
}
```

## 前端模板

链接地址：<http://byrushan.com/projects/ma/1-5-2/jquery/index.html>,想要什么效果，在浏览器里面的开发者模式中查看吧。

## 控制器-前端请求

与前端的交互，使用ajax技术。

- 代码示例,实现一个控制器,具体代码可以看 `controllers/Home.php`

```
//一定要包含这个基类。
include 'MY_Controller.php';
//继承基类
class Home extends MY_Controller {
    // 写一个Hello方法
    // 浏览器只要打开 http://域名/index.php/Home/Hello
    public function Hello()
    {
        // 这个判断很重要,目前前端的查询请求,都是通过这个判断的。
        if (parent::isAjax()) { // 判断是否ajax提交
            // 接收get提交的参数。第二个参数可选,用于决定是否使用 XSS
            // 过滤器对数据进行过滤。
            $name = $this->input->get('name', TRUE);
            // 接收post提交的参数。
            $age = $this->input->post('age');
            echo 'hello' , $name;
        }
        echo 'hello man';
    }
}
```

- 加载模型, 注意 ⚠️, 模型首字母必须大写!

加载模型, 使用 `$this->load->model('模型名称')`。具体代码, 可以看 `controllers/Home.php`;

```
//加载 Summary_model 这个模型
$this->load->model('Summary_model');
// 调用模型里面的方法
$this->Summary_model->getData();
```

- 加载其它的东西

参考文档: [使用 CodeIgniter 类库,辅助函数](#)

```
// 加载辅助函数
$this->load->helper(array('form', 'language', 'url'));
// 加载类库
$this->load->library(array('ion_auth', 'form_validation', 'session'));
```

# 统计系统的数据是怎么来的

后台的数据的来源可以分为两类，一类直接从原始的游戏数据库（`sdk`）查询、统计得到；另一类是服务器每日定时执行统计脚本将统计好的数据存入 `mydb` 库后得到的，比如 `留存` 就是这么做的。

## 首先说 `mydb` 库的数据

- 定时任务，是用Linux的 `crontab` 做的。
- `controllers\AutoRunDay.php` ,就是每日执行统计的脚本。
- `controllers\AutoRunHour.php` ,就是每小时执行统计的脚本。

## 接着说 `sdk` 库的数据

- 先看 `controllers\Api.php` 文件

接收到客户端提交上来的数据，判断接口的优先级，优先级高的，就直接保存如数据库来。优先级低的，就先将数据存入队列，然后队列服务慢慢的把数据存到数据库里面。队列服务用的是 `httpsqs`。

- 再看 `controllers\RunQueue.php` 文件

这是服务器后台一直在执行的程序，`cli` 命令行下执行，可以说是一个死循环吧。是通过`supervisor`管理这个后台进程的。这个程序的作用就是将队列里面的数据，一条条取出来，存入到数据库里面。

- 如何增加一个新的接口
  - 首先，在 `controllers\Api.php` 文件中增加具体的方法，比如 `SaveLogin` ,优先级不高的，只要定义就可以了。

```
//Api.php
public function SaveLogin(){}
```

- 接着，在 `controllers\RunQueue.php` 中，写一个相同名称的方法,但是要写具体的保存操作了。

```
//RunQueue.php
public function SaveLogin(){
    // 将数据保存入库
    $this->db->insert('表名称', $this->data);
}
```

```
}
```

- 这样代码虽然写好了，但是还没有真正生效哦。提交到服务器后，需要重启队列服务才算完工。

```
# 进入到supervisor控制台,直接运行supervisorctl命令就好,然后执行restart 队列名称即可。像下面这样
$ supervisorctl
httpsqs                                RUNNING    pid 21561, uptime 10 days, 0:21:37
queue                                  RUNNING    pid 3326, uptime 1:38:30
queue-2                                RUNNING    pid 3331, uptime 1:26:33
supervisor> restart queue
supervisor> restart queue-2
```

## config 目录说明

- api.php: 配置接口名称对应的数据库表名称
- channel\_list.php : 渠道配置
- consume\_types.php : 消费类型配置
- event\_click\_config.php : 游戏注册流程配置
- menu\_config.php : 菜单配置