## 日志记录接口使用指引v1.0.1

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 修改内容 | 修改人 | 日期 |
| 1.0.0 | 初版发布 | 互联网软件开发技术第三组 | 2019/4/20 |
| 1.0.1 | 贴上ClientInfo；添加自测图。 | 互联网软件开发技术第三组 | 2019/4/21 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[日志记录接口使用指引v1.0.1 1](#_Toc6819424)

[1目的 2](#_Toc6819425)

[2适用范围 2](#_Toc6819426)

[3 接口使用 2](#_Toc6819427)

[3.1 接口说明 2](#_Toc6819428)

[3.2使用方式 6](#_Toc6819429)

[3.3反馈 9](#_Toc6819430)

## 1目的

通过企业微信端和web端对数据库进行登录、添加、删除、修改操作时，需要记录是哪个用户操作的哪个数据表的哪些数据项的哪些字段，以及用户的操作时间、登录ip和登录方式等信息。因此实现了一个统一的日志模型**app\logmanage\model\Log，**所有的开发人员只要对数据有进行操作，就需要调用日志模型内的相应接口，按照指定的格式传递参数进行调用即可。

## 2适用范围

涉及数据库操作的所有开发人员。

## 3 接口使用

### 3.1 接口说明

统一接口：**app\logmanage\model\Log::recordLogApi($uid, $type, $table = '', $field = '')。**

/\*\*

\* 杨宇 董亚聪 苏恒杰

\* 功能：记录web端和企业微信端的登录/增加/修改/删除日志

\* @param

\* @uid：操作人的主键id，非学号

\* @type: 1登陆 2增加 3修改 4删除

\* @table ：操作的数据表名，如操作的数据表为'user\_info'，则 $table = 'user\_info'

\* @field ：操作的数据表的内容数组

\* @return int

\* 1登陆: 只需要传入$uid, $type

\* 2增加：需要传入$uid, $type, $table, $field(该字段传入你增加的所有数据的主键，如 $field = ['11'，'12'])

\* 3修改：假如同时操作了数据表中主键为22和23的两条数据的field1和field2字段, 则 $field = ['22'=>['field1'=> ['before value', 'after value'], 'field2'=> ['before value', 'after value']],'23'=>['field1'=> ['before value', 'after value'], 'field2'=> ['before value', 'after value']]]

\* 4删除：需要传入$uid, $type, $table, $field(该字段传入你删除的所有数据的主键，如 $field = ['11'，'12'])

\*/

public function recordLogApi($uid, $type, $table = '', $field = ''){

$client = new ClientInfo(); // ClientInfo是自己实现的类，这里没有粘贴出来，你只需要关注接口本身就行

$ip = $client->getIp();

$agent = [

'os' => $client->GetOS(),

'brower' => $client->getBrowser(),

];

$action = [

'table' => $table,

'id\_list' => $field,

];

if($type == 1) {

$data = ['user\_id' => $uid, 'operate\_type' => $type, 'operate\_time' => date('Y-m-d H:i:s', time()), 'user\_agent' => json\_encode($agent), 'ip' => $ip];

}else{

$data = ['user\_id' => $uid, 'operate\_type' => $type, 'operate\_time' => date('Y-m-d H:i:s', time()), 'operate\_action' => json\_encode($action), 'user\_agent' => json\_encode($agent), 'ip' => $ip];

}

$res = Db::name('log\_user')->insert($data);

return $res;

}

ClientInfo类：

class ClientInfo extends Model{

public function GetLang() {

$Lang = substr($\_SERVER['HTTP\_ACCEPT\_LANGUAGE'], 0, 4);

//使用substr()截取字符串，从 0 位开始，截取4个字符

if (preg\_match('/zh-c/i',$Lang)) {

//preg\_match()正则表达式匹配函数

$Lang = '简体中文';

}

elseif (preg\_match('/zh/i',$Lang)) {

$Lang = '繁體中文';

}

else {

$Lang = 'English';

}

return $Lang;

}

public function getBrowser() {

$user\_OSagent = $\_SERVER['HTTP\_USER\_AGENT'];

if (strpos($user\_OSagent, "Maxthon") && strpos($user\_OSagent, "MSIE")) {

$visitor\_browser = "Maxthon(Microsoft IE)";

} elseif (strpos($user\_OSagent, "Maxthon 2.0")) {

$visitor\_browser = "Maxthon 2.0";

} elseif (strpos($user\_OSagent, "Maxthon")) {

$visitor\_browser = "Maxthon";

} elseif (strpos($user\_OSagent, "Edge")) {

$visitor\_browser = "Edge";

} elseif (strpos($user\_OSagent, "Trident")) {

$visitor\_browser = "IE";

} elseif (strpos($user\_OSagent, "MSIE")) {

$visitor\_browser = "IE";

} elseif (strpos($user\_OSagent, "MSIE")) {

$visitor\_browser = "MSIE";

} elseif (strpos($user\_OSagent, "NetCaptor")) {

$visitor\_browser = "NetCaptor";

} elseif (strpos($user\_OSagent, "Netscape")) {

$visitor\_browser = "Netscape";

} elseif (strpos($user\_OSagent, "Chrome")) {

$visitor\_browser = "Chrome";

} elseif (strpos($user\_OSagent, "Lynx")) {

$visitor\_browser = "Lynx";

} elseif (strpos($user\_OSagent, "Opera")) {

$visitor\_browser = "Opera";

} elseif (strpos($user\_OSagent, "MicroMessenger")) {

$visitor\_browser = "WeiXinBrowser";

} elseif (strpos($user\_OSagent, "Konqueror")) {

$visitor\_browser = "Konqueror";

} elseif (strpos($user\_OSagent, "Mozilla/5.0")) {

$visitor\_browser = "Mozilla";

} elseif (strpos($user\_OSagent, "Firefox")) {

$visitor\_browser = "Firefox";

} elseif (strpos($user\_OSagent, "U")) {

$visitor\_browser = "Firefox";

} elseif (strpos($user\_OSagent, "Safari/")) {

$visitor\_browser = "Safari";

} else {

$visitor\_browser = "Other Browser";

}

return $visitor\_browser;

}

public function GetOS() {

$OS = $\_SERVER['HTTP\_USER\_AGENT'];

if (preg\_match('/win/i',$OS)) {

$OS = 'Windows';

}

elseif (preg\_match('/mac/i',$OS)) {

$OS = 'MAC';

}

elseif (preg\_match('/linux/i',$OS)) {

$OS = 'Linux';

}

elseif (preg\_match('/unix/i',$OS)) {

$OS = 'Unix';

}

elseif (preg\_match('/bsd/i',$OS)) {

$OS = 'BSD';

}

else {

$OS = 'Other';

}

return $OS;

}

public function GetIP() {

if (!empty($\_SERVER['HTTP\_CLIENT\_IP'])) {

//如果变量是非空或非零的值，则 empty()返回 FALSE。

$IP = explode(',',$\_SERVER['HTTP\_CLIENT\_IP']);

}

elseif (!empty($\_SERVER['HTTP\_X\_FORWARDED\_FOR'])) {

$IP = explode(',',$\_SERVER['HTTP\_X\_FORWARDED\_FOR']);

}

elseif (!empty($\_SERVER['REMOTE\_ADDR'])) {

$IP = explode(',',$\_SERVER['REMOTE\_ADDR']);

}

else {

$IP[0] = 'None';

}

return $IP[0];

}

private function GetAddIsp() {

$IP = $this->GetIP();

$AddIsp = mb\_convert\_encoding(file\_get\_contents('http://open.baidu.com/ipsearch/s?tn=ipjson&wd='.$IP),'UTF-8','GBK');

//mb\_convert\_encoding() 转换字符编码。

if (preg\_match('/noresult/i',$AddIsp)) {

$AddIsp = 'None';

}

else {

$Sta = stripos($AddIsp,$IP) + strlen($IP) + strlen('来自');

$Len = stripos($AddIsp,'"}')-$Sta;

$AddIsp = substr($AddIsp,$Sta,$Len);

}

$AddIsp = explode(' ',$AddIsp);

return $AddIsp;

}

public function findCityByIp($ip){

$data = file\_get\_contents('http://ip.taobao.com/service/getIpInfo.php?ip='.$ip);

return json\_decode($data,$assoc=true);

}

public function GetAdd() {

$Add = $this->GetAddIsp();

return $Add[0];

}

public function GetIsp() {

$Isp = $this->GetAddIsp();

if ($Isp[0] != 'None' && isset($Isp[1])) {

$Isp = $Isp[1];

}

else {

$Isp = 'None';

}

return $Isp;

}

}

### 3.2使用方式

首先把日志模型引用到你需要调用的文件中：**use app\logmanage\model\Log as LogModel;**

* 登录记录

……

$model = new LogModel();

$uid = 110; // 操作人主键id，非学号

$type = 1;

$model->recordLogApi ($uid, $type); //需要判断调用是否成功

……

* 添加记录日志

……

$model = new LogModel();

$uid = 110; // 操作人主键id，非学号

$type = 2;

$table = 'user\_info';

$field = ['1801221','1801220','1801222']; // 增加的主键列表，不是学号

$model->recordLogApi ($uid, $type, $table, $field); //需要判断调用是否成功

……

* 修改记录日志

……

$model = new LogModel();

$uid = 110; // 操作人主键id，非学号

$type = 3;

$table = 'user\_info';

$field = [

'22'=>[

'field1'=> ['before value', 'after value'],

'field2'=> ['before value', 'after value']

],

'23'=>[

'field1'=> ['before value', 'after value'],

'field2'=> ['before value', 'after value']

]

];

$model->recordLogApi ($uid, $type, $table, $field); //需要判断调用是否成功

……

* 删除记录日志

……

$model = new LogModel();

$uid = 110; // 操作人主键id，非学号

$type = 4;

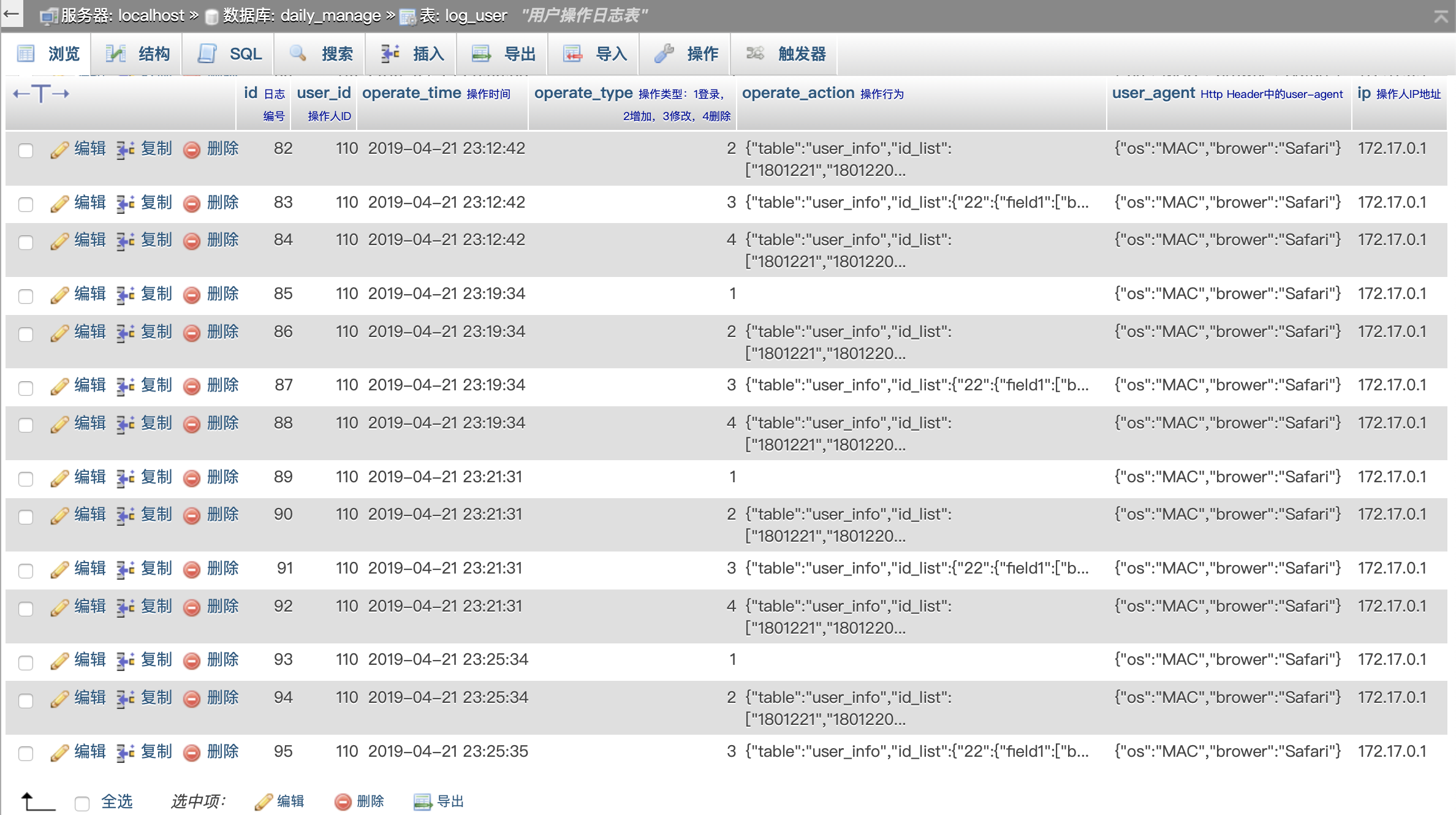
$table = 'user\_info';

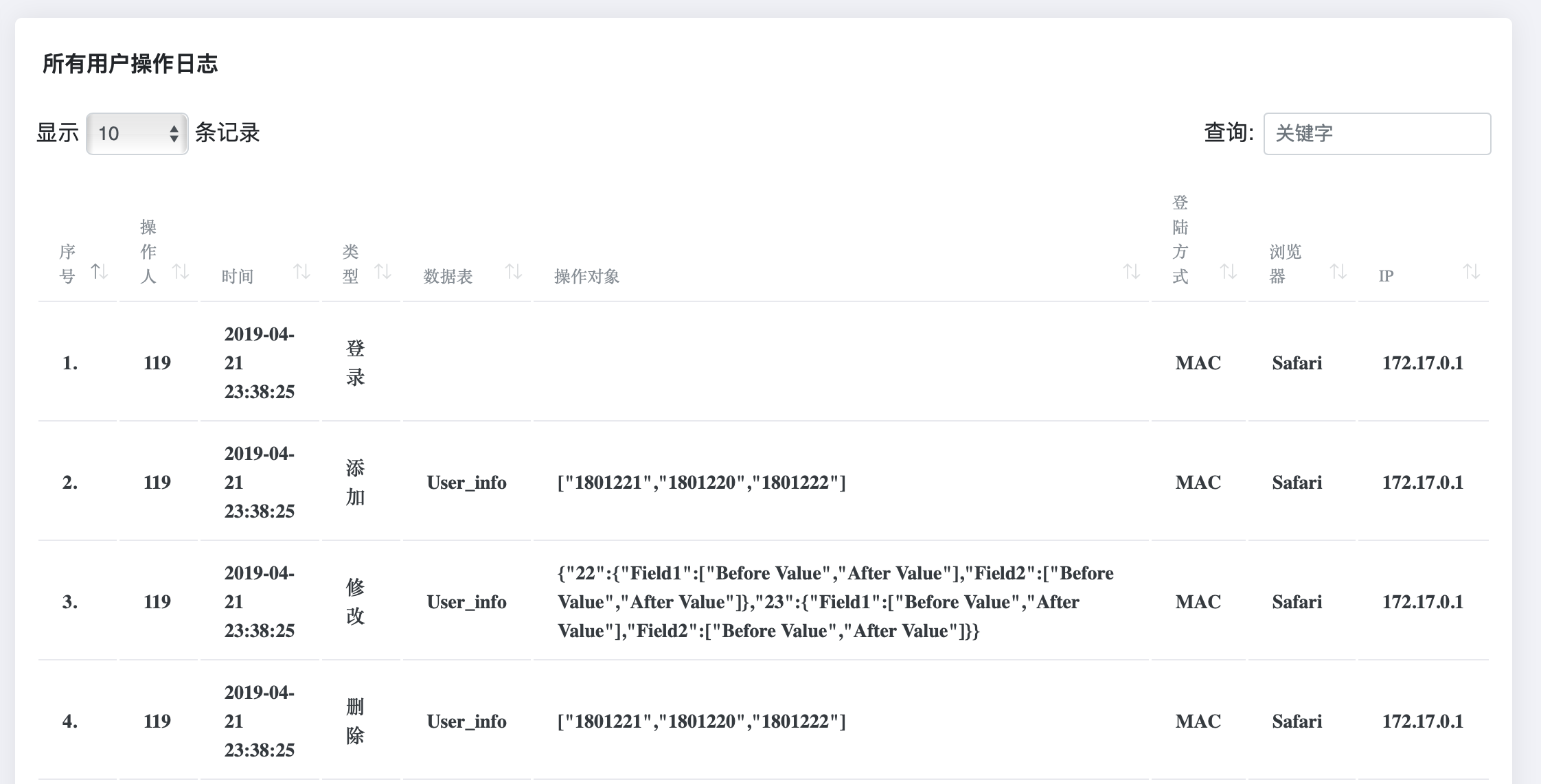
$field = ['1801221','1801220','1801222']; // 删除的主键列表, 不是学号

$model->recordLogApi ($uid, $type, $table, $field); //需要判断调用是否成功

……

下图是测试的图片，接口已经能够正常调用。







### 3.3反馈

如果有问题请及时反馈到第三组。