1. 重写render()方法

namespace app\common\lib;  
use think\exception\Handle;  
  
class ApiHandleException extends Handle  
{  
 protected $httpCode = 500;  
 public function render(\Exception $e)  
 {  
 if(config('app\_debug')){  
 return parent::render($e);  
 }  
 if($e instanceof ApiException){  
 $this->httpCode = $e->httpCode;  
 }  
 return show($e->getCode(),$e->getMessage(),[],$this->httpCode);  
 }  
}

1. 配置文件

// 异常处理handle类 留空使用 \think\exception\Handle  
'exception\_handle' => '\app\common\lib\ApiHandleException',

1. 重写异常类

namespace app\common\lib;  
use think\Exception;  
class ApiException extends Exception  
{  
 protected $message;  
 protected $code;  
 public $httpCode;  
 public function \_\_construct($message = '',$code = 0,$httpCode = 0){  
 $this->message = $message;  
 $this->code = $code;  
 $this->httpCode = $httpCode;  
 }  
}

1. 使用

正常逻辑

return show(1,'OK',$users,200);

手动抛出异常

throw new ApiException('aaa',403);