1. **RESTful API**
   1. 面向资源
   2. http动词：get,put,post,delete
   3. Api数据格式一般json：[‘status’ => 1,’msg’ => ‘success’,’data’ => []]
   4. HTTP状态码
      1. 200请求成功
      2. 201创建成功
      3. 202更新成功
      4. 400无效请求
      5. 401地址不存在
      6. 403禁止访问
      7. 404请求资源不存在
      8. 500服务器错误
2. **TP5 API**
   1. 创建控制器：php think make:controller blog;

| **标识** | **请求类型** | **生成路由规则** | **对应操作方法（默认）** |
| --- | --- | --- | --- |
| index | GET | blog | index |
| create | GET | blog/create | create |
| save | POST | blog | save |
| read | GET | blog/:id | read |
| edit | GET | blog/:id/edit | edit |
| update | PUT | blog/:id | update |
| delete | DELETE | blog/:id | delete |

* 1. 资源路由Route::resource(‘blog’,’index/blog’);
  2. 配置默认返回：’default\_return\_type’ => ‘json’;
  3. json方法封装：

function show($status,$msg,$data[],$httpCode = 200){

$data = [

‘status’ => $status,

‘msg’ => $msg,

‘data’ => $data

];

return json($data,$httpCode);

}

* 1. 异常重写Handel.php->render(/Exception $e)

Class ApiHandleException extends Handle{

Public function $httpCode = 500;

Public function render(\Exception $e){

If(config(‘app\_debug’) == true){

Return parent::render($e);

}

If($e instanceof ApiException){

$this->httpCode = $e->httpCode;

}

Return show(0,$e->getMessage(),[],$this->httpCode);

}

}

* 1. 修改配置：’exception\_handle’ => ‘app\common\lib\exception\ApiHandelException’,
  2. 重写异常类

Class ApiException extends Exception{

Public function \_\_construct($msg = ‘’,$httpCode = 0,$code =0){

$this->httpCode = $httpCode;

$this->msg = $msg;

$this->code = $code;

}

}

1. **API 数据安全**
   1. 加密方式MD5 AES RSA
   2. 如何做
      1. 基本参数放入header：sign,version,app\_type,did设备号,mdoel机型
      2. 每次http请求都携带sign
      3. Sign唯一性保证:客户端加密，服务端解密
      4. 请求参数、返回数据按安全性适当加密
      5. Access\_token
   3. 新建公共控制器Common
      1. 校验数据

Public function checkRequestAuth(){

//获取header头信息

$headers = request()->header();

//todo

//基础参数校验

If(empty($headers[‘sign’])){

throw new ApiException(‘sign不存在’，400);

}

If(in\_array($headers[‘app\_type’],config(‘app.apptypes’))){

throw new ApiException(app\_type不合法’，400);

}

//检查sign是否正常

checkSignPass($data){

//解密$str = $data[‘sign’]

//分解为数组parse\_str($str,$arr);

If(!is\_array($arr) || empty($arr[‘did’] || $arr[‘did’] != $data[‘did’])){

}

}

}

* 1. Sign加密：客户端完成
     1. $data = [‘did’=>’123456’,’version’=>1];
     2. setSign($data = []){}
     3. 按字段排序ksort($data);
     4. 拼接字符串$string = http\_build\_query($data); //did=123456&version=1
     5. aes加密字符串$string = (new Aes())->encrypt($string);
  2. Sign解密：服务器端完成
     1. aes解密字符串$string = (new Aes())->encrypt($string);

1. **版本控制**
   1. 添加版本
      1. 新建版本：app\api\controller\v1\\*
      2. 定义路由：Route::get(‘api/:version/index’,’api/:version.index/index’);
   2. 升级版本
      1. 类型：自主选择更新，强制更新
      2. 新建表ent\_version:id,app\_type,version(int8内部版本号),version\_code(var20外部版本号),is\_force(tin1是否强制更新)，apk\_url(var255 apk包最新地址)，upgrade\_point(var255 升级提示),status,create\_time,update\_time
      3. 定义路由：Route::get(‘api/:ver/init’,’api/:ver.index/init’);
      4. public function init(){

$version = model(‘Version’)->get([‘app\_type’ => 获取最新版本信息]);

If(empty($version)){

//

}

If($version->version > $this->headers[‘version’]){

$version->is\_update = $version->is\_force == 1 ? 2 : 1;

}else{

$version->is\_update = 0;

}

return show(config(‘code.success’),’OK’,$version,200);

}

* + 1. 新建表用于记录用户的基本信息env\_app\_active:id,version,app\_type,did,model,create\_time,update\_time

1. 首页接口开发
2. 列表页接口开发
3. 搜索接口开发
4. 排行接口开发
5. 详情页接口开发
6. 登录接口开发
7. 个人中心接口开发
8. **点赞接口开发**
   1. 新建表ent\_user\_news：id,user\_id,news\_id,create\_time
   2. 定义路由：Route::post(‘api/:version/upvote’,’api/:version.upvote/save’);
   3. public function save(){

$id = input(‘post.id’,0,intval’);

//validate

$data[‘user\_id’] = $this->user->id;

$data[‘news\_id’] = $id;

//查询是否点赞过

$userNews = model(‘UserNews’)->get($data);

If($userNews){

return show();

}

try{

$id = model(‘UserNews’)->add($data);

If($id){

//更新新闻表中点赞数量+1

model(‘News’)->where(‘id’,$id)->setInc(‘upvote\_count’);

}else{

}

}catch(\Exception $e){

return throw new ApiException();

}

}

* 1. 取消点赞：Route::delete(‘api/:version/upvote’,’api/:version.upvote/delete’);同时自减upvote\_count字段

1. **评论接口开发**
   1. 新建表env\_comment:id,user\_id,news\_id,content,to\_user\_id

,create\_time,parent\_id,status

* 1. 创建路由:Route::post(‘api/:version/comment’,’api/:version.comment/save’);
  2. Public function save(){

$data = input(‘post.’,’’,’过滤’);

//validate

$data[‘user\_id’] = $this->user->id;

try{

$id = model(‘Comment’)->add($data);

If($id){

}else{

}

}catch(){

Throw new ApiException();

}

}

* 1. 评论列表：//定义路由

function read(){

//原生select \* from ent\_comment as c join ent\_user as u on c.user\_id = u.id

//and c.news\_id = 7

//TP5关联查询

$newsId = input(‘param.id’,0,’intval’);

If(empty($newsId)){

Return new ApiException();

}

try{

$comments = model(‘Comment’)->where()->select();

}catch(\Exception $e){

}

$this->assign();

}

模型写入:

public function user(){

return $this->belongsTo('comment');

}