Vue+php 开发个人博客系统

1. 配置数据库

$active\_group = 'default';

$query\_builder = TRUE;

$db['default'] = array(

'dsn' => 'mysql:dbname=数据库名称;host=数据库地址',

'hostname' => '数据库地址',

'username' => '数据库用户名',

'password' => '数据库密码',

'database' => '数据库名称',

'dbdriver' => 'mysqli',

'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

);

1. 配置路由

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

/\*

| -------------------------------------------------------------------------

| URI ROUTING

| -------------------------------------------------------------------------

| This file lets you re-map URI requests to specific controller functions.

|

| Typically there is a one-to-one relationship between a URL string

| and its corresponding controller class/method. The segments in a

| URL normally follow this pattern:

|

| example.com/class/method/id/

|

| In some instances, however, you may want to remap this relationship

| so that a different class/function is called than the one

| corresponding to the URL.

|

| Please see the user guide for complete details:

|

| https://codeigniter.com/user\_guide/general/routing.html

|

| -------------------------------------------------------------------------

| RESERVED ROUTES

| -------------------------------------------------------------------------

|

| There are three reserved routes:

|

| $route['default\_controller'] = 'welcome';

|

| This route indicates which controller class should be loaded if the

| URI contains no data. In the above example, the "welcome" class

| would be loaded.

|

| $route['404\_override'] = 'errors/page\_missing';

|

| This route will tell the Router which controller/method to use if those

| provided in the URL cannot be matched to a valid route.

|

| $route['translate\_uri\_dashes'] = FALSE;

|

| This is not exactly a route, but allows you to automatically route

| controller and method names that contain dashes. '-' isn't a valid

| class or method name character, so it requires translation.

| When you set this option to TRUE, it will replace ALL dashes in the

| controller and method URI segments.

|

| Examples: my-controller/index -> my\_controller/index

| my-controller/my-method -> my\_controller/my\_method

\*/

$route['default\_controller'] = 'welcome';

$route['404\_override'] = 'notfound';

$route['translate\_uri\_dashes'] = FALSE;

$route['test'] = 'common/Test/test';

//===========================================admin-后台管理-接口===================================

// 添加-已测试

// $route['a/create']['post'] = 'admin/Admin/create';

// 登录-已测试

$route['a/login']['post'] = 'admin/Admin/login';

// 获取七牛上传图片token-已测试

$route['a/qiniu/token']['get'] = 'admin/Qiniu/get\_qiniu\_token';

// 获取博客配置-已测试

$route['a/webConfig']['get'] = 'admin/WebConfig/get\_web\_config';

// 修改博客配置-已测试

$route['a/webConfig/modify']['post'] = 'admin/WebConfig/modify';

// 获取 关于我 页面-已测试

$route['a/webConfig/getAbout']['get'] = 'admin/WebConfig/get\_about\_me';

// 修改 关于我 页面-已测试

$route['a/webConfig/modifyAbout']['post'] = 'admin/WebConfig/modify\_about';

// 获取友链类型列表-已测试

$route['a/friends/typeList']['get'] = 'admin/Friends/get\_friends\_type';

// 获取友链列表-已测试

$route['a/friends/list']['get'] = 'admin/Friends/get\_friends\_list';

// 添加友链-已测试

$route['a/friends/add']['post'] = 'admin/Friends/add\_friend';

// 编辑友链-已测试

$route['a/friends/modify']['post'] = 'admin/Friends/modify\_friend';

// 删除友链-已测试

$route['a/friends/delete']['post'] = 'admin/Friends/del\_friend';

// 添加分类

$route['a/category/add']['post'] = 'admin/Categories/add\_category';

// 添加标签

$route['a/tag/add']['post'] = 'admin/Categories/add\_tag';

// 修改分类

$route['a/category/modify']['post'] = 'admin/Categories/modify\_category';

// 修改标签

$route['a/tag/modify']['post'] = 'admin/Categories/modify\_tag';

// 删除分类

$route['a/category/delete']['post'] = 'admin/Categories/del\_category';

// 删除标签

$route['a/tag/delete']['post'] = 'admin/Categories/del\_tag';

// 获取分类列表

$route['a/category/list']['get'] = 'admin/Categories/get\_category\_list';

// 获取标签列表

$route['a/tag/list']['get'] = 'admin/Categories/get\_tag\_list';

// 获取分类

$route['a/category']['get'] = 'admin/Categories/get\_category';

// 获取标签

$route['a/tag']['get'] = 'admin/Categories/get\_tag';

// 保存文章

$route['a/article/save']['post'] = 'admin/Article/save';

// 发布文章

$route['a/article/publish']['post'] = 'admin/Article/publish';

// 编辑文章

$route['a/article/modify']['post'] = 'admin/Article/modify';

// 删除文章

$route['a/article/delete']['post'] = 'admin/Article/delete';

// 获取文章信息

$route['a/article/info']['get'] = 'admin/Article/get\_article';

// 获取文章列表

$route['a/article/list']['get'] = 'admin/Article/get\_article\_list';

// 获取系统日志

$route['a/sys/log']['get'] = 'admin/System/get\_sys\_log';

// 获取首页面板显示的统计信息

$route['a/statistics/home']['get'] = 'admin/Statistics/get\_home\_statistics';

// 获取文章评论列表

$route['a/comments/list']['get'] = 'admin/Comments/get\_comments';

// 添加评论

$route['a/comments/add']['post'] = 'admin/Comments/add';

// 删除评论

$route['a/comments/delete']['post'] = 'admin/Comments/delete';

// 获取所有评论

$route['a/comments/alllist']['get'] = 'admin/Comments/get\_all\_comments';

// 获取 我的简历 页面

$route['a/webConfig/getResume']['get'] = 'admin/WebConfig/get\_resume';

// 修改 我的简历 页面

$route['a/webConfig/modifyResume']['post'] = 'admin/WebConfig/modify\_resume';

//===========================================admin-后台管理-接口-结束===============================

//===========================================web-前端-接口===================================

// 获取 关于我 页面-已测试

$route['w/getAbout']['get'] = 'web/WebConfig/get\_about\_me';

// 获取文章归档列表

$route['w/article/archives']['get'] = 'web/Article/get\_article\_archives';

// 获取文章信息

$route['w/article']['get'] = 'web/Article/get\_article';

// 获取文章列表

$route['w/article/list']['get'] = 'web/Article/get\_article\_list';

// 获取分类列表

$route['w/category/list']['get'] = 'web/Categories/get\_category\_list';

// 获取标签列表

$route['w/tag/list']['get'] = 'web/Categories/get\_tag\_list';

// 获取博客信息

$route['w/blogInfo']['get'] = 'web/WebConfig/get\_blog\_info';

// 获取友链列表

$route['w/friends/list']['get'] = 'web/Friends/get\_friends\_list';

// 获取文章评论列表

$route['w/comments/list']['get'] = 'web/Comments/get\_comments';

// 添加评论

$route['w/comments/add']['post'] = 'web/Comments/add';

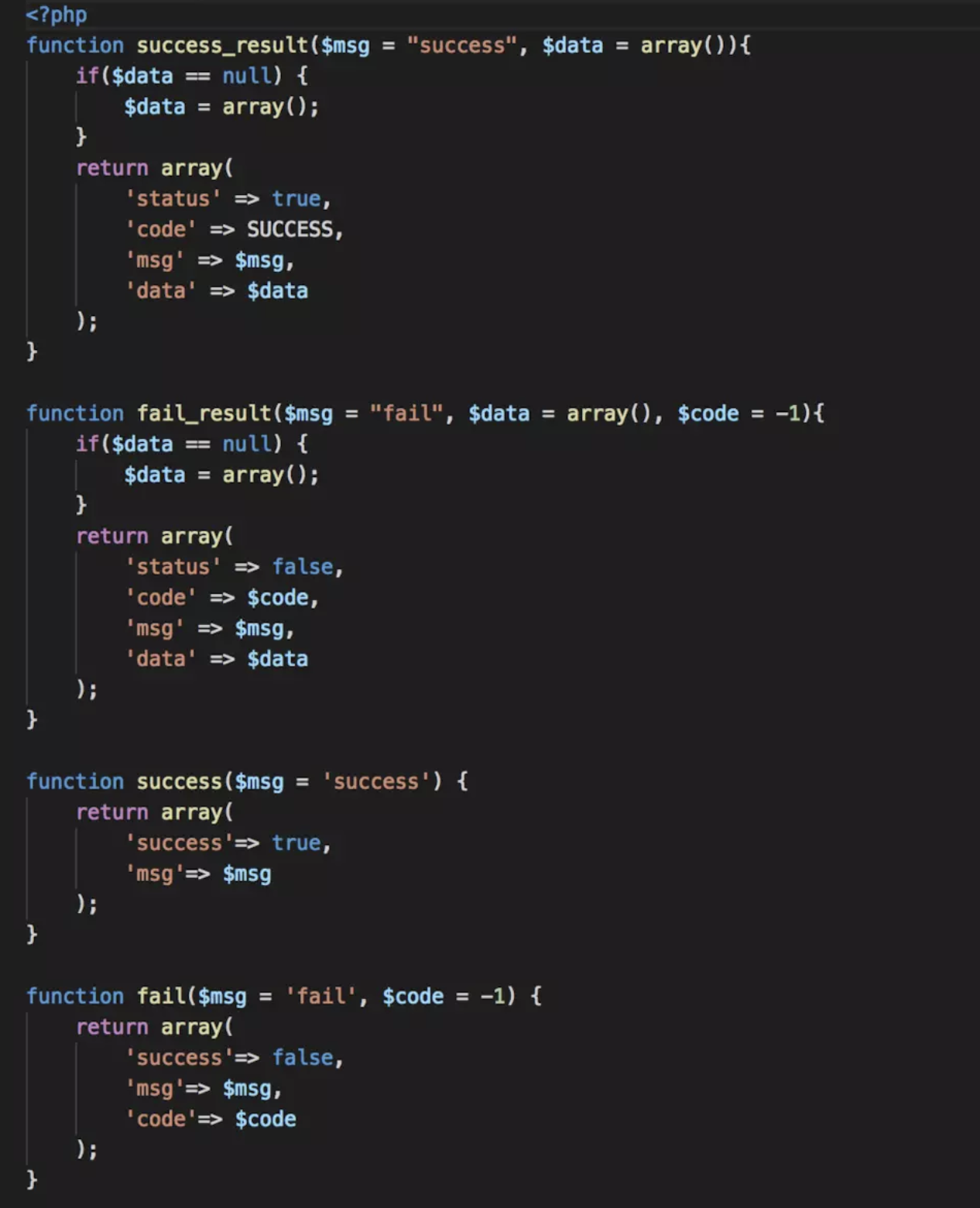
// 获取 我的简历 页面-已测试

$route['w/getResume']['get'] = 'web/WebConfig/get\_resume';

// 按文章标题和简介搜索

$route['w/article/search']['get'] = 'web/article/search';

1. 返回值helper



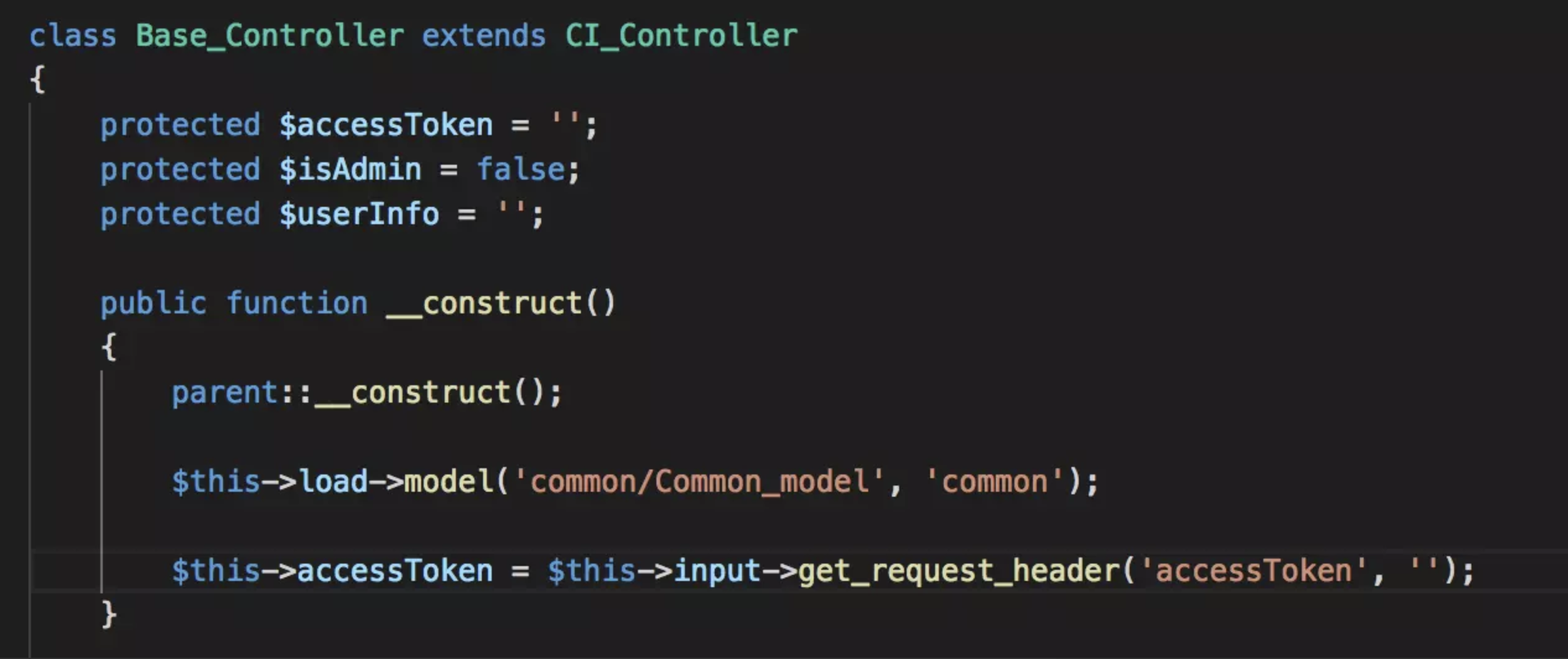
其中success\_result、fail\_result这两个方法是接口返回的json结构，success、fail这两个方法是model返回给controller的json结构，这样封装统一返回接口，比较好管理。

1. 后台管理权限检查

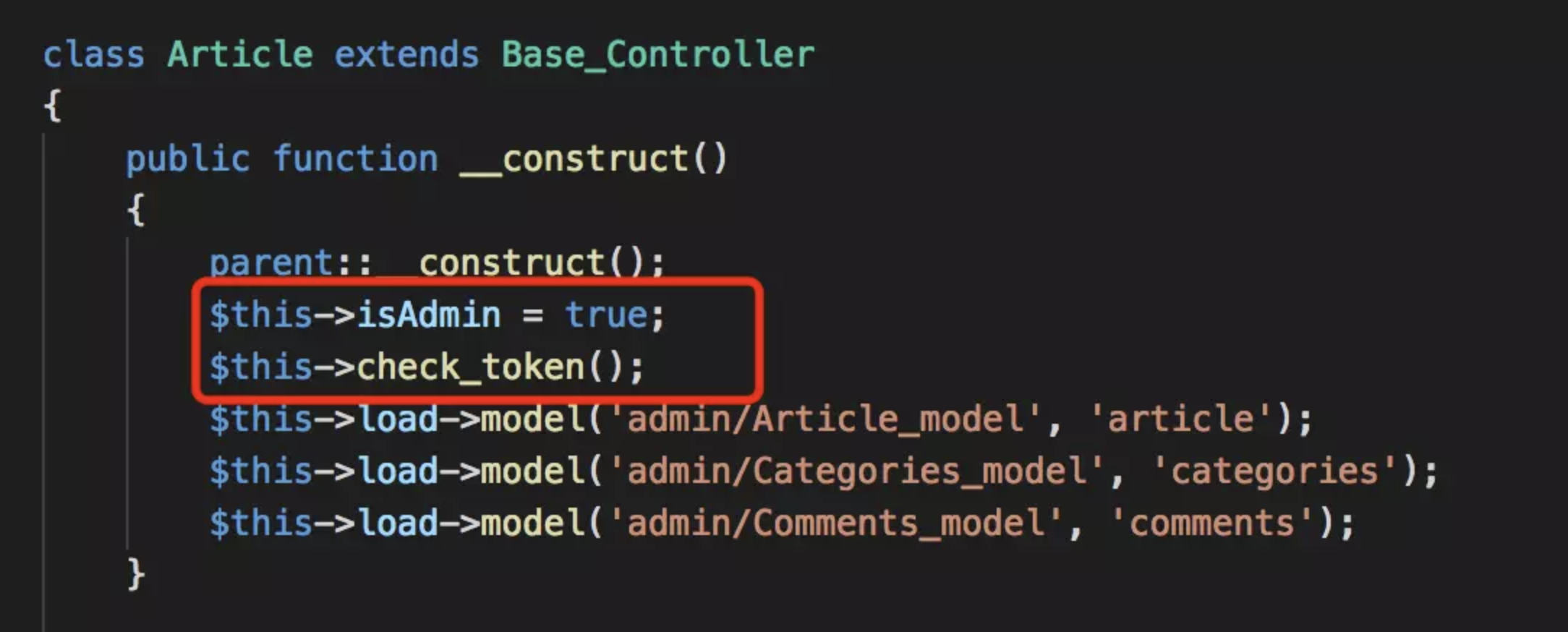
这里我使用了token作为权限认证，每次登陆都会重新生成一个新的token以及更新有效期，保存进数据库。 在Common\_model中实现token检查：



在Base\_Controller中加载Common\_model并获取客户端传来的accessToken：



其他所有controller都继承Base\_controller，在需要权限验证的controller构造方法里调用权限检查：



1. axios封装统一处理

