# **[YII2项目常用技能知识总结](https://www.cnblogs.com/sandea/p/5714830.html)**

**1、不通过日志获取AR执行的原生SQL语句和打印变量数据**

$query = User::find() ->select(['username'])->where(['id'=>[1,2,3,4])// get the AR raw sql in YII2

$commandQuery = clone $query;echo $commandQuery->createCommand()->getRawSql();$users = $query->all();

打印变量数据可以这样写：//引用命名空间use yii\helpers\VarDumper;//使用

VarDumper::dump($var);//使用2  第二个参数是数组的深度  第三个参数是是否显示代码高亮（默认不显示）

VarDumper::dump($var, 10 ,true);

**2、从数据库二维数组中返回一维数组并配合rules验证规则实现分类数据过滤。**

普通返回表记录的二维数组

Member::find()->select('userid')->asArray()->all();

Array

(

    [0] => Array

        (

            [userid] => 1

        )

    [1] => Array

        (

            [userid] => 2

        )

    [2] => Array

        (

            [userid] => 3

        )

)

返回字段的一维数组

Member::find()->select('userid')->asArray()->column();

或者：

\yii\helpers\ArrayHelper::getColumn(Member::find()->all(), 'userid')

Array

(

    [0] => 1

    [1] => 2

    [2] => 3

)

返回一维数组配合验证规则验证数据正确性，如分类catid正确分为只有1-4，但是在devTools打开修改catid为5，提交同样会到数据库，此时rules验证规则如下：

['catid', 'in', 'range' => category::find()->select('id')->asArray()->column()],

当然，这个也可以通过下面这样子写，一样的：

['catid', 'in', 'range' => \yii\helpers\ArrayHelper::getColumn(category::find()->all(), 'catid')],

这样就可以过滤不正确的分类数据了！

**3、友好时间表示方法**

之前一直使用自定义的友好时间函数。几天前发现万能的YII已经提供了友好时间访问，代码如下：

Yii::$app->formatter->asRelativeTime('1447565922'); //2小时前

**4、使用不同的响应类型或者自定义响应类型**

有效的格式:

FORMAT\_RAW

FORMAT\_HTML

FORMAT\_JSON

FORMAT\_JSONP

FORMAT\_XML

JSON响应

public function actionIndex()

{

    \Yii::$app->response->format = \yii\web\Response::FORMAT\_JSON;

    $items = ['some', 'array', 'of', 'data' => ['associative', 'array']];

    return $items;

}

返回：

{

    "0": "some",

    "1": "array",

    "2": "of",

    "data": ["associative", "array"]

}

自定义响应格式

让我们创建一个定制的响应格式。例子做点有趣和疯狂的事我返回PHP 数组。 首先，我们需要格式化程序本身。创建 components/PhpArrayFormatter.php：

<?phpnamespace app\components;use yii\helpers\VarDumper;use yii\web\ResponseFormatterInterface;class PhpArrayFormatter implements ResponseFormatterInterface

{

    public function format($response)

    {

        $response->getHeaders()->set('Content-Type', 'text/php; charset=UTF-8');

        if ($response->data !== null) {

            $response->content = "<?php\nreturn " . VarDumper::export($response->data) . ";\n";

        }

    }

}

组件配置：

return [

    // ...

    'components' => [

        // ...

        'response' => [

            'formatters' => [

                'php' => 'app\components\PhpArrayFormatter',

            ],

        ],

    ],

];

现在是准备使用。在 controllers/SiteController 创建一个新的方法 actionTest:

public function actionTest()

{

    Yii::$app->response->format = 'php';

    return [

        'hello' => 'world!',

    ];

}

返回如下：

<?phpreturn [

    'hello' => 'world!',

];

**5、AR入库前时间通过在模型重写behaviors方法实现优雅入库方式。**

如下：

public function behaviors()

{

    return [

        'timestamp' => [

            'class' => TimestampBehavior::className(),

            'attributes' => [

                ActiveRecord::EVENT\_BEFORE\_INSERT => 'creation\_time',

                ActiveRecord::EVENT\_BEFORE\_UPDATE => 'update\_time',

            ],

            'value' => function() { return date('U'); // unix timestamp },

        ],

    ];

}

**6、除配置组件记录不同级别日志外，也可以自定义在某个地方记录LOG日志**

use yii\log\Logger;

\Yii::getLogger()->log('User has been created', Logger::LEVEL\_INFO);

**7、 ActiveForm类不让生成label标签**

//方法一，通过ActiveForm类

$form->field($model, '字段名')->passwordInput(['maxlength' => true])->label(false) ?>

//方法二，通过 HTML类

Html::activeInput($type,$model,'字段名')

Yii2给必填项加星，样式如下：

div.required label:after {

    content: " \*";

    color: red;

}

**8、Yii2 获取接口传过来的 JSON 数据：**

接收get和post的数据很容易，那么接收json数据呢？！没关系，看这里：

Yii::$app->request->rawBody;

**9、座机和手机号码必须填写一个：**

public function rules()

{

    return [

        [['telephone', 'mobile'], function ($attribute, $param) {//至少要一个

            if (empty($this->telephone) && empty($this->mobile)) {

                $this->addError($attribute, 'telephone/mobile至少要填一个');

            }

        }, 'skipOnEmpty' => false],

    ];

}

**10、where多条件查询示例：**

//and复杂示例：

$time = time();

Member::find()->where(['and', ['userid' => 1, 'company' =>'测试公司'], ['>', 'addtime', $time]]);//SELECT \* FROM `member` WHERE ((`userid`=1) AND (`company`='测试公司')) AND (`addtime` > 1447587486)

//and和or组合示例：

$query = Member::find()->where(['and', ['>','userid',2], ['or', ['company' => '深圳市新民家具有限公司'], ['address' => '深圳']]]);//SELECT \* FROM `member` WHERE (`userid` > 2) AND ((`company`='深圳市新民家具有限公司') OR (`address`='深圳'))

**11、关于事务：**

优雅的写法

Yii::$app->db->transaction(function() {

    $order = new Order($customer);

    $order->save();

});

这相当于下列冗长的代码：

$transaction = Yii::$app->db->beginTransaction();try {

    $order = new Order($customer);

    $order->save();

    $transaction->commit();

} catch (\Exception $e) {

    $transaction->rollBack();

    throw $e;

}

**12、rest风格API获取客户端提交的get和post的数组**

// post

Yii::$app->request->bodyParams// get

Yii::$app->request->queryParams;

**13、一个控制器调用其他控制器action的方法：**

方法一：

是经典的重写actions方法

  public function actions()

    {

        return [

            'error' => [

                'class' => 'yii\web\ErrorAction',

            ],

            'captcha' => [

                'class' => 'yii\captcha\CaptchaAction',

                'fixedVerifyCode' => YII\_ENV\_TEST ? 'testme' : null,

            ],

        ];

    }

actions继承yii\base\Actions类，并重写父类的run方法。

方法二：

site控制器如下，访问MemberController控制器下面的index方法。

class SiteController extends Controller

{

    public function actionIndex(){

     Yii::$app->runAction('member/index', ['param'=>'123']);

    }

}

MemberController控制器如下：

class MemberController extends Controller

{

    public function actionIndex($param = '456'){

     echo "second Controller".$param;

    }

}

访问：<http://www.yii.dev/site/index.html>

输出：second Controller123

**14、点击下载，如下载安卓APK文件。**

public function actionDownload(){

    return \Yii::$app->response->setDownloadHeaders("http://xxx.com/apk/com.trade.activity.3.0.8.apk");

    //return \Yii::$app->response->sendFile("./com.trade.activity.3.0.8.apk");

}

**15、YII模块IP白名单设置，增加安全性**

$config['modules']['gii'] = [

     'class' => 'yii\gii\Module',

     'allowedIPs' => ['127.0.0.1', '::1','10.10.1.\*'],

];

$config['modules']['debug'] = [

    'class' => 'yii\debug\Module',

    'allowedIPs' => ['127.0.0.1', '::1', '192.168.0.\*', '192.168.33.1'],

];

**16、防止 SQL 和 Script 注入**

use yii\helpers\Html;use yii\helpers\HtmlPurifier;echo Html::encode($view\_hello\_str) //可以原样显示<script></script>代码  echo HtmlPurifier::process($view\_hello\_str)  //可以过滤掉<script></script>代码

**17、验证某个ID值是否存在**

//之前一直用$model->findOne($id);exists()方法，资源节约，有没有？！

public function validateAttribute($model, $attribute)

{

   $value = $model->$attribute;

   if (!Status::find()->where(['id' => $value])->exists()) {

       $model->addError($attribute, $this->message);

   }

}

**18、批量查询**

如查询并循环10000条数据。一次性拿1万条内存会有压力，通过批量查询，每次拿1000条，那么内存始终只有1000条的占有量。

foreach(Member::find()->batch(1000) as $value){

    //do something

    //print\_r(count($value));

}

**19、关于CSRF验证**

方法一：关闭Csrf，除非必要，否则不推荐

public function init(){

    $this->enableCsrfValidation = false;

}

方法二：普通提交，form表单中加入隐藏域

<input name="\_csrf" type="hidden" id="\_csrf" value="<?= Yii::$app->request->csrfToken ?>">

方法三：ajax异步提交，加入\_csrf字段

var csrfToken = $('meta[name="csrf-token"]').attr("content");

$.ajax({

  type: 'POST',

  url: url,

  data: {\_csrf:csrfToken},

  success: success,

  dataType: dataType

});

**20、YII命令行生成数据库文件**

自动列出可用的migrate文件

php yii migrate

 从vendor/callmez/wechat/migrations目录下生成数据表

php yii migrate --migrationPath=@callmez/wechat/migrations

从当前应用/migrations/db1下初始化数据到db1表

php yii migrate --migrationPath=@app/migrations/db1 --db=db1

21.关联查询

//客户表Model：CustomerModel //订单表Model：OrdersModel//国家表Model：CountrysModel//首先要建立表与表之间的关系 //在CustomerModel中添加与订单的关系

      Class CustomerModel extends \yii\db\ActiveRecord

{

    ...

    public function getOrders()

    {

        //客户和订单是一对多的关系所以用hasMany

        //此处OrdersModel在CustomerModel顶部别忘了加对应的命名空间

        //id对应的是OrdersModel的id字段，order\_id对应CustomerModel的order\_id字段

        return $this->hasMany(OrdersModel::className(), ['id'=>'order\_id']);

    }

    public function getCountry()

    {

        //客户和国家是一对一的关系所以用hasOne

        return $this->hasOne(CountrysModel::className(), ['id'=>'Country\_id']);

    }

    ....

}

      // 查询客户与他们的订单和国家

CustomerModel::find()->with('orders', 'country')->all();

// 查询客户与他们的订单和订单的发货地址

CustomerModel::find()->with('orders.address')->all();

// 查询客户与他们的国家和状态为1的订单

CustomerModel::find()->with([

    'orders' => function ($query) {

        $query->andWhere('status = 1');

        },

        'country',

])->all();

22、yii2中关闭debug后return $this->redirect($url)；不能跳转，服务器报500错误。

问题分析：

1.必须 return 才能让$this->redirect($url);立马跳转, 而不执行后续代码;

2.redirect() 中指定了响应的 http status code，默认是302；

3.当执行$this->redirect($url)时，不管是否在后面加return false 、return true都没有用，还是继续执行完代码。使用header("Location:$url");exit;可以解决此问题，但是，这不是yii2的逻辑，并不完美。

解决办法：

【本文由php\_sir的博客 http://blog.sina.com.cn/phpsir原创，未经授权禁止转载】

1.在正常情况下，使用 return $this->redirect($url);

2.在解决方案1不生效时，用$this->redirect($url);Yii::$app->response->send();

3.在解决方案2不生效时，用$this->redirect($url);Yii::$app->end();

总结：

用Yii::$app->end();、Yii::$app->response->send();不管在actionXXX还是init方法都能终止代码，而return只能在action终止代码，是因为在init()里仅仅是代码的执行，return只是代码返回。