自动加载机制psr-4

Loader::addNamespace(别名，路径) 注册命名空间定义

runtime/classmap.php 类库映射文件（自行创建）

如 classmap.php

return [

‘别名’ => ‘绝对路径’;

];

路由定义（application/config.php ->url\_route\_must = 是否强制开启路由）

直接定义application/route.php

‘路由地址’ =>’控制器地址’,

‘路由地址/:[param]’ => [‘控制器地址’, [过滤条件] , [参数过滤条件]],

‘路由分组-<param>-<param2>’ => [

‘路由分组’ => ‘路由地址’,

‘路由分组/:param$’ =>[控制器地址, [‘ext’ =>’请求文件后缀’, ‘method’ =>’请求方式’], [‘param’ =>’参数过滤正则表达’]]

],

请求（Request $request(聚合超全局请求变量)）

$request ->param() 或函数助手 input(‘key’,’key2’, ‘key3’) 获取请求参数(所有类型)

$request ->post() 获取post请求参数,参数为要获取的参数键

$request ->get() 获取get请求参数

$request ->cookie() 获取cookie信息

$request ->session() 获取session信息

$request ->file() 获取上传文件信息

input函数助手

input(‘post.key’) 获取post请求参数

input(‘file.filename’) 获取上传文件信息

（$request ->请求方式(键),否则为所有信息）

(except(), only(), has() , isAjax(), type())

响应(application/config.php ->default\_retrun\_type = 默认返回值类型)

retrun 响应类型(响应值，响应状态码)

view

json

jsonp

xml

redirect 页面跳转

success 使用thinkphp/tpl/dispatch\_jump.tpl作为模板跳转

error 错误指示跳转

(参考thinkphp/helper.php函数助手)

数据库（use think\Db;）application/database.php

原生查询sql

Db::query(‘查询sql语句’); 返回查询信息

可直接使用预处理,如：

Db::query(‘select \* from user where id=?’, 1); 占位

Db::query(‘select \* from user where name=:name’, [‘name’=>’value’]); 关联

Db::execute(‘执行sql语句’); 返回操作行数

切换数据库（本次操作有效）

Db::connect([

数据库连接信息，参考application/database.php

]);

$link = Db::connect([连接信息])；

保存连接对象

查询构造器

Db::table(‘表名’)

->insert([‘key’ =>’val’]) 插入数据

->update([‘key’=>’val’]) 更新数据

->where ->delete() 删除数据

查询数据

->where() 删除/修改/查询条件

->limit() 筛选

->order() 排序

->field() 选择查询字段

->select() 开始查询

->find() 查询单个记录

->column() 查询列

->chunk() 分块查询

->value() 查询值

->count() 聚合查询

事务 ( Db::transaction(function (){事务处理}) )

Db::startTrans() 开始事务

try{

Db::commit() 提交事务

}catch(\Exception $e){

Db::rollback() 回滚事务

}

查询表达式

where(‘字段’, ‘条件类型’, ‘值’)

where(‘id’, ‘in’, [1,2,3])

where(‘id’, ‘bwteen’, [1,10])

多个条件

where([

‘id’ =>[‘条件’， 字符串参数值]

‘id’ =>[‘between’, ‘1,10’]

])

快捷查询

where(‘id$status’, 1) 查询id和状态=1

where(‘id|status’, 1) 查询id或状态=1

闭包查询

$result = DB::name(‘tablename’) ->select (function ($query){

$query->where(‘id’,1)

});

分块查询

$result = Db::table(‘tablename’) ->chunk(10, function($list){将筛选到的所有数据÷10查询});

聚合查询

count()

max()

min()

avg()

sum();

数据库模型