



开发手册

完全开发

PHP 3.2.3 WEB应用开发框架

ThinkPHP 中文WEB

清华大学出版社

卷二

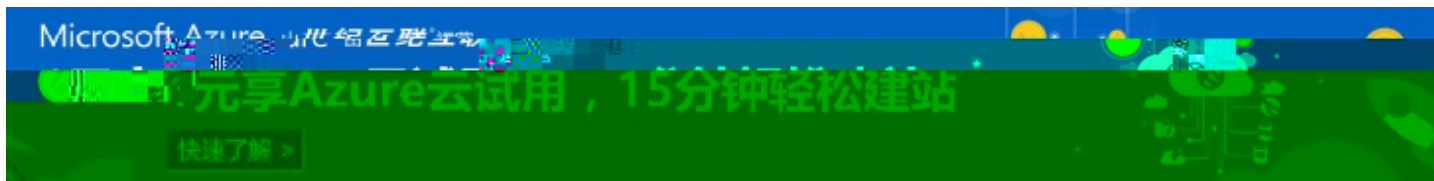
ThinkPHP中文



ThinkPHP3.2.3

3.2
3.2.3

< -- -- -- >



1.0

thinkphp@qq.com

ThinkPHP

[<http://www.thinkphp.cn/topic>]

ThinkPHP

ThinkPHP

<http://www.thinkphp.cn>

ThinkPHP

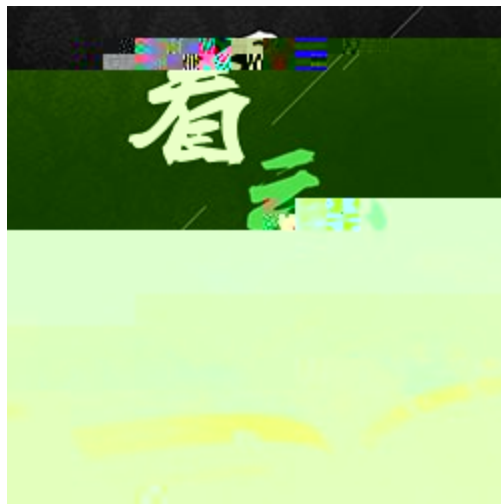
ThinkPHP

WEB

10

WEB

^_^



ThinkPHP

MVC

PHP

Apache2

WEB

3.2

3.2.3

- PDO
- insertAll
-
-
- Mongo
-
-
-
-
- CURD SQL
- using/index/fetchSql/strict/token
- setInc setDec
- |
- Action
- SQL
-
-
-
-
- memcache Session
- session

ThinkPHP

<http://thinkphp.cn/down/framework.html>

SAE

github

Git

Github <https://github.com/liu21st/thinkphp>

Oschina <http://git.oschina.net/liu21st/thinkphp.git>

Code <https://code.csdn.net/topthink2011/ThinkPHP>

Coding <https://coding.net/u/liu21st/p/thinkphp/git>

3.2 composer composer web

composer create-project topthink/thinkphp your-project-name

ThinkPHP

WEB

ThinkPHP

PHP

- PHP5.3 PHP5.3dev PHP6

- Windows/Unix
- Apache IIS nginx WEB
- Mysql MsSQL PgSQL Sqlite Oracle Ibase Mongo

PHP	ThinkPHP	WAMPServer	wampserver
Apache	PHP MySQL	PHP	MySQL Apache
ThinkPHP			

web

```
www WEB
  index.php
  README.md   README
  Application
  Public
  ThinkPHP
```

Application

Public

README.md

Application

ThinkPHP

```
ThinkPHP                               web
  Common
  Conf
  Lang
  Library
    Think      Think
    Behavior
    Org      Org
    Vendor
    ...
  Mode
  Tpl
  LICENSE.txt
  logo.png    LOGO
  README.txt   README
  ThinkPHP.php
```

Public

WEB

-
-
-
-

```
define('APP_PATH','./Application/');
require './ThinkPHP/ThinkPHP.php';
```

Application

Apps

APP_PATH

```
define('APP_PATH','./Apps/');
require './ThinkPHP/ThinkPHP.php';
```

APP_PATH

" /"

```
define('APP_PATH','./Application/');
require './Think/ThinkPHP.php';
```

THINK_PATH

```
define('APP_PATH','./Application/');
define('THINK_PATH',realpath('../Think').'/');
require THINK_PATH.'ThinkPHP.php';
```

APP_PATH THINK_PATH

" /"

THINK_PATH APP_PATH



欢迎使用 ThinkPHP !

Application

Common

Home

Runtime

Application

Common

Common

Conf

Home

Home

Conf

Common

Controller

Model

View

Runtime

Cache

Data

Logs

Temp

Windows

Application

Runtime

common-runtime.php

index.html

ThinkPHP

URL

index.html

default.html

```
define('DIR_SECURE_FILENAME', 'default.html');  
define('APP_PATH', './Application/');  
require './ThinkPHP/ThinkPHP.php';
```

```
define('BUILD_DIR_SECURE', false);
```

Application

```
Application  
  Common  
  Home  
  Admin  
  ...  
  Runtime
```

```
Module  
  Conf  
  Common  
  Controller  
  Model  
  Logic  
  Service  Service  
  ...  
  View
```

MVC

Conf Common

Application/Home/Controller

IndexController.class.php

Index

+ Controller

+ class.php

Home

Index

index

index

```

namespace Home\Controller;
use Think\Controller;
class IndexController extends Controller {
    public function index(){
        echo 'hello,world!';
    }
}

```

hello,world!

IndexController

```

namespace Home\Controller;

```

Home

Home\Controller/IndexController

Home/Controller/IndexController.class.php

PHP

```

use Think\Controller;

```

Think\Controller

```

namespace Home\Controller;
use Think\Controller;
class IndexController extends Controller

```

```
namespace Home\Controller;
class IndexController extends \Think\Controller
```

ThinkPHP

- .class.php ThinkPHP DbMysql.class.php
- Home\Controller\UserController
- Application/Home/Controller/UserController.class.php
- Unix ThinkPHP
- Windows
- UserController
- InfoModel.class.php
- UserController.class.php Info
- client_ip
- " _" getUsername
- _parseType
- " _" tableName _instance
- " _" _call _autoload
- HAS ONE MANY_TO_MANY
- HTML_CACHE_ON
- MY_LANG
- _CLASS_NOT_EXIST_
- ThinkPHP .html
-

ThinkPHP
BOM

UTF-8
BOM

UTF-8

ThinkPHP

-
-

ThinkPHP

PHP

PHP

ThinkPHP

PHP

```
//  
return array(  
    'DEFAULT_MODULE' => 'Index', //  
    'URL_MODEL'      => '2', //URL  
    'SESSION_AUTO_START' => true, //      session  
    //  
    //...  
);
```

```
//  
return array(  
    'default_module' => 'Index', //  
    'url_model'      => '2', //URL  
    'session_auto_start' => true, //      session  
    //  
    //...  
);
```

```
//
return array(
    'DEFAULT_MODULE' => 'Index', //
    'URL_MODEL'      => '2', //URL
    'SESSION_AUTO_START' => true, //      session
    'USER_CONFIG'    => array(
        'USER_AUTH' => true,
        'USER_TYPE' => 2,
    ),
    //
    //...
);
```

yaml/json/xml/ini

```
define('CONF_EXT','.ini');
```

.ini

ini

```
DEFAULT_MODULE=Index ;
URL_MODEL= 2 ;URL
SESSION_AUTO_START=on ;      session
```

xml

```
< config >
< default_module > Index < /default_module >
< url_model > 2 < /url_model >
< session_auto_start > 1 < /session_auto_start >
< /config >
```

yaml

```
default_module:Index #  
url_model:2 #URL  
session_auto_start:True # session
```

json

```
{  
  "default_module":"Index",  
  "url_model":2,  
  "session_auto_start":True  
}
```

yaml/json/xml/ini

```
define('CONF_EXT','.test'); //  
define('CONF_PARSE','parse_test'); //
```

var1 = val1 & var2 = val2

parse_test

```
function parse_test($str){  
  parse_str($str,$config);  
  return (array)$config;  
}
```

CONF_PARSE

PHP

ThinkPHP

->

->

->

->

->

->

->

ThinkPHP/Conf/convention.php

Application/Common/Conf/config.php

Application/Common/Conf/config_ .php

ThinkPHP/Conf/debug.php

Application/Common/Conf/debug.php

```
define('APP_STATUS','office');
```

Application/Common/Conf/office.php

```
define('APP_STATUS','home');
```

Application/Common/Conf/home.php

Application/ /Conf/config.php

Application/ /Conf/config_ .php

Application/ /Conf/ .php

C Config

C(' ')
URL

```
$model = C('URL_MODEL');
//
// $model = C('url_model');
```

" . "

url_model NULL

```
// my_config default_config
C('my_config',null,'default_config');
```

C

user.php db.php

db

Application/Common/Conf/user.php Application/Common/Conf/db.php

Home

Application/Home/Conf/user.php Application/Home/Conf/db.php

user.php

```
<?php
//
return array(
    'USER_TYPE' => 2, //
    'USER_AUTH_ID' => 10, // ID
    'USER_AUTH_TYPE' => 2, //
);
```

```
C('USER_AUTH_ID');
```

```
//
'LOAD_EXT_CONFIG' => array('USER' => 'user', 'DB' => 'db'),
```

```
C('USER.USER_AUTH_ID');
```

C

```
$config = array('WEB_SITE_TITLE' => 'ThinkPHP', 'WEB_SITE_DESCRIPTION' => ' PHP ');
C($config);
```

\$config

ThinkPHP / /

ThinkPHP

URL

PATHINFO

URL

http://serverName/index.php / / / / [/ ...]

ThinkPHP

> php.exe index.php(/ / / [/ ...]

	URL

MVC


```

www WEB
  index.php
  README.md  README
  Common
  Application
  Public
  ThinkPHP

```

Application

Common

Admin

```

// Admin
define('BIND_MODULE','Admin');
define('APP_PATH','./Application/');
require './ThinkPHP/ThinkPHP.php';

```

URL

http://serverName/index.php

Admin

Admin\Controller\IndexController

BUILD_CONTROLLER_LIST

```

// Admin
define('BIND_MODULE','Admin');
define('BUILD_CONTROLLER_LIST','Index,User,Menu');
define('APP_PATH','./Application/');
require './ThinkPHP/ThinkPHP.php';

```

```

Admin\Controller\IndexController
Admin\Controller\UserController
Admin\Controller\MenuController

```

Think\Controller

APP_USE_NAMESPACE

BUILD_MODEL_LIST

```
// Admin
define('BIND_MODULE','Admin');
define('BUILD_MODEL_LIST','User,Menu');
define('APP_PATH','./Application/');
require './ThinkPHP/ThinkPHP.php';
```

```
Admin\Model\UserModel
Admin\Model\MenuModel
```

Think\Model

APP_USE_NAMESPACE

Think\Build

```
// Admin Role
// Admin\Controller\RoleController
//
\Think\Build::buildController('Admin','Role');
// Admin Role
// Admin\Model\RoleModel
//
\Think\Build::buildModel('Admin','Role');
```

Think\Build

ThinkPHP

Common

Runtime Runtime

```
//
'MODULE_DENY_LIST' => array('Common','Runtime','Api'),
```

Api

URL

```
'MODULE_ALLOW_LIST' => array('Home','Admin','User'),
'DEFAULT_MODULE' => 'Home',
```

Home Admin User
URL

Home

```
//
'MULTI_MODULE' => false,
'DEFAULT_MODULE' => 'Home',
```

Home

index.php

admin.php

Admin

```
// Home
define('BIND_MODULE','Admin');
define('APP_PATH','./Application/');
require './ThinkPHP/ThinkPHP.php';
```

<http://serverName/index.php/Admin/Index/index>

<http://serverName/admin.php/Index/index>

```
define('BIND_MODULE', 'Home'); // Home
define('BIND_CONTROLLER', 'Index'); // Index
define('APP_PATH', './Application/');
require './ThinkPHP/ThinkPHP.php';
```

http://serverName/index.php/Home/Index/index

http://serverName/home.php/index

URL

URL

http://serverName/index.php/ / /

3.2 URL

URL

URL

ThinkPHP

URL

ThinkPHP

URL

Linux

URL

```
'URL_CASE_INSENSITIVE' => true,
```

URL_CASE_INSENSITIVE true URL

PATHINFO

PATHINFO

URL

SEO

URL

PATHINFO

URL

`http://localhost/index.php/home/user/login/var/value/`

PATHINFO

/

/

PATHINFO

URL

`http://localhost/index.php/home/user/login?var=value`

PATHINFO

URL

```
// PATHINFO  
'URL_PATHINFO_DEPR'=>'-'
```

URL

`http://localhost/index.php/home-user-login-var-value`

REWRITE

REWRITE

PATHINFO

URL

`index.php`

WEB

Apache

`.htaccess`

```
<IfModule mod_rewrite.c>  
RewriteEngine on  
RewriteCond %{REQUEST_FILENAME} !-d  
RewriteCond %{REQUEST_FILENAME} !-f  
RewriteRule ^(.*)$ index.php/$1 [QSA,PT,L]  
</IfModule>
```

URL

`http://localhost/home/user/login/var/value`

URL

URL

PATHINFO

URL

`http://localhost/?s=/home/user/login/var/value`

```
'VAR_PATHINFO' => 'path'
```

PATHINFO

```
// PATHINFO
'URL_PATHINFO_DEPR' => '-';
```

URL `http://localhost/?path=/home-user-login-var-value`

Web REWRITE URL

Apache .htaccess

```
<IfModule mod_rewrite.c>
RewriteEngine on
RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^(.*)$ index.php?s=/$1 [QSA,PT,L]
</IfModule>
```

REWRITE URL

`http://localhost/home/user/login/var/value`

MVC

ThinkPHP MVC Model-View-Controller - - multi-Layer

Model

Model

ThinkPHP Model Model Model

Model Logic Service

1. Model/UserModel
2. Logic/UserLogic
3. Service/UserService

Model

Home/Model/UserModel.class.php

```
namespace Home\Model;
use Think\Model;
class UserModel extends Model{
}
```

Home/Logic/UserLogic.class.php

```
namespace Home\Logic;
use Think\Model;
class UserLogic extends Model{
}
```

Home/Service/UserService.class.php

```
namespace Home\Service;
use Think\Model;
class UserService extends Model{
}
```

D

```
D('User') //      UserModel
D('User','Logic') //      UserLogic
D('User','Service') //      UserService
```

Model

```
'DEFAULT_M_LAYER' => 'Logic', //      Logic
```

```
D('User') //      UserLogic
D('User','Model') //      UserModel
D('User','Service') //      UserService
```

Model

View

UserEvent

```
A('User','Event');
```

Controller

```
'DEFAULT_C_LAYER' => 'Event', // Event
```

MVC

ThinkPHP

M

V

C

V

ThinkPHP

C

CBD

ThinkPHP

CBD

Core+

Behavior+

Driver

+

+

AOP

"

"

"

"

Core

ThinkPHP

ThinkPHP

```
ThinkPHP/Common/functions.php //
ThinkPHP/Conf/convention.php //
ThinkPHP/Conf/debug.php //
ThinkPHP/Mode/common.php //
ThinkPHP/Library/Think //
ThinkPHP/Library/Behavior //
ThinkPHP/Library/Think/App.class.php //
ThinkPHP/Library/Think/Cache.class.php //
ThinkPHP/Library/Think/Controller.class.php //
ThinkPHP/Library/Think/Db.class.php //
ThinkPHP/Library/Think/Dispatcher.class.php // URL
ThinkPHP/Library/Think/Exception.class.php //
ThinkPHP/Library/Think/Hook.class.php //
ThinkPHP/Library/Think/Log.class.php //
ThinkPHP/Library/Think/Model.class.php //
ThinkPHP/Library/Think/Route.class.php //
ThinkPHP/Library/Think/Storage.class.php //
ThinkPHP/Library/Think/Template.class.php //
ThinkPHP/Library/Think/Think.class.php //
ThinkPHP/Library/Think/View.class.php //
```

Behavior

Driver

3.2

```
ThinkPHP/Library/Think/Cache/Driver //
ThinkPHP/Library/Think/Db/Driver //
ThinkPHP/Library/Think/Log/Driver //
ThinkPHP/Library/Think/Session/Driver // Session
ThinkPHP/Library/Think/Storage/Driver //
ThinkPHP/Library/Think/Template/Driver //
ThinkPHP/Library/Think/Template/TagLib //
```

Behavior

Behavior ThinkPHP

Hello world



AOP " "

AOP

- app_init
- module_check 3.2.1
- path_info PATH_INFO
- app_begin
- action_name
- action_begin
- view_begin
- view_template
- view_parse
- template_filter
- view_filter
- view_end
- action_end
- app_end

```
// my_tag  
Think\Hook\Hook(my_tag')| g tyt;
```

```
; )
```

```
// my_tag
\Think\Hook::listen('my_tag','param');
```

ThinkPHP/Behavior/

BuildLite	Lite 3.2.1	app_init
ParseTemplate		view_parse
ShowPageTrace	Trace Trace	view_end
ShowRuntime		view_filter
TokenBuild		view_filter
ReadHtmlCache		app_init
WriteHtmlCache		view_filter

+ Behavior

```
namespace Home\Behavior;
class TestBehavior {
    // run
    public function run(& $params){
        if(C('TEST_PARAM')) {
            echo 'RUNTEST BEHAVIOR '.$params;
        }
    }
}
```

run ,

run

run

tags.php

```
return array(
    '1' => array('1', '2', ...),
    '2' => array('1', '2', ...),
);
```

app_init

CheckLangBehavior

```
return array(
    'app_init' => array('Home\Behavior\CheckLangBehavior'),
);
```

```
return array(
    'app_init' => array(
        'Home\Behavior\CheckLangBehavior',
        'Home\Behavior\CronRunBehavior'
    ),
);
```

tags.php

app_init

```
'app_begin' => array(
    'Behavior\ReadHtmlCacheBehavior', //
),
```

```
'app_begin' => array(
    'Home\Behavior\CheckModuleBehavior',
    'Home\Behavior\CheckLangBehavior',
),
```

app_begin

```
Library\Behavior\ReadHtmlCacheBehavior
Home\Behavior\CheckModuleBehavior
Home\Behavior\CheckLangBehavior
```

```
'app_begin' => array(
    'Home\Behavior\CheckModuleBehavior',
    'Home\Behavior\CheckLangBehavior',
    '_overlay' => true,
),
```

app_begin

```
Home\Behavior\CheckModuleBehavior
Home\Behavior\CheckLangBehavior
```

```
return array(
    'app_begin'=>array('Home\Behavior\TestBehavior'), // app_begin Test
    'app_end'=>array('Home\Behavior\TestBehavior'), // app_end Test
);
```

```
namespace Home\Behavior;
use Think\Behavior;
class AuthCheckBehavior extends Behavior {

    // run
    public function run(& $return){
        if(C('USER_AUTH_ON')) {
            // $return = true;
            // halt
        }
    }
}
```

AuthCheck

_initialize

```
B('Home\Behavior\AuthCheck');
```

3.2

3.2
PHP PHP

Org\Util\File

```
namespace Org\Util;
class File {
}
```

ThinkPHP/Library/Org/Util/File.class.php

```
$class = new \Org\Util\File();
```

ThinkPHP/Library/Org/Util/File.class.php

Org\Util\File

Org

ThinkPHP/Library

Library

Library

My

Test

```
namespace My;
class Test {
    public function sayHello(){
        echo 'hello';
    }
}
```

Test ThinkPHP/Library/My/Test.class.php

```
$Test = new \My\Test();  
$Test->sayHello();
```

```
namespace Home\Model;  
class UserModel extends \Think\Model {  
}
```

Application/Home/Model/UserModel.class.php

```
namespace Admin\Event;  
class UserEvent {  
}
```

Application/Admin/Event/UserEvent.class.php

3.2

PHP

```
//  
$class = new \stdClass();  
$xml = new \SimpleXMLElement($xmlStr);
```

3.2

\Org\Util\Auth

```
namespace Org\Util;  
class Auth {  
}
```

ThinkPHP/Library/Org/Util/Auth.class.php


```
new \Org\Util\Auth();
```

\Org\Util\Auth

ThinkPHP/Library/Org/Util/Auth.class.php

Library

```
Library
  Think      Think
  Org        Org
  ...
```

Library

Think Org

```
new \Think\Cache\Driver\File();
new \Org\Util\Auth();
new \Org\Io\File();
```

Library

Library

```
'AUTOLOAD_NAMESPACE' => array(
  'My' => THINK_PATH.'My',
  'One' => THINK_PATH.'One' . 'ne' . n           /Library/
```

```
new \Home\Model\UserModel();
new \Home\Event\UserEvent();
```

ThinkPHP/Library

Home

AUTOLOAD_NAMESPACE

Home

Home

```
Application/Home/Model/UserModel.class.php
Application/Home/Event/UserEvent.class.php
```

```
Think\Think::addMap('Think\Log',THINK_PATH.'Think\Log.php');
Think\Think::addMap('Org\Util\Array',THINK_PATH.'Org\Util\Array.php');
```

"

addMap

```
$map = array('Think\Log' => THINK_PATH.'Think\Log.php', 'Org\Util\Array' => THINK_PATH.'Org\Util\Array.php');
Think\Think::addMap($map);
```

alias.php

```
return array(
    'Think\Log' => THINK_PATH.'Think\Log.php',
    'Org\Util\Array' => THINK_PATH.'Org\Util\Array.php'
);
```

Test\MyClass


```
import("Org.User#Info");
```

import

import

RBA C/AccessDecisionManager.class.php

```
import("RBA C.AccessDecisionManager",dirname(__FILE__));
```

class.php

php

import

```
import("RBA C.AccessDecisionManager",dirname(__FILE__)," .php");
```

Unix

Linux

import

Vendor

.php

Vendor

Zend

Filter\Dir.php

Vendor

Dir

Vendor\Zend\Filter\Dir.php

vendor

```
Vendor('Zend.Filter.Dir');
```

Dir

Vendor

import

```
Vendor('Zend.Filter.Dir',dirname(__FILE__),' .class.php');
```

STORAGE_TYPE

Think\Storage

common

APP_MODE

```
// SAE SAE  
define('STORAGE_TYPE','sae');  
define('APP_MODE','sae');  
define('APP_PATH','./Application/');  
require './ThinkPHP/ThinkPHP.php';
```

SAE

SAE

sae

config_

sae

```
//  
Application/Common/Conf/config_sae.php  
//  
Application/Home/Conf/config_sae.php
```

config_sae

sae

sae

IO

Runtime

Application/Runtime

common~runtime.php

~runtime.php

18. ThinkPHP/Lang/zh-cn.php

19.

20. ThinkPHP/Conf/debug.php

21. Application/Common/Conf/debug.php

22. APP_STATUS

23. CHECK_APP_DIR RUNTIME_PATH

24. Think\App run

25. app_init

26.

27. Think\Dispatcher::dispatch URL

28. URL \$_SERVER['PATH_INFO']

29. APP_SUB_DOMAIN_DEPLOY

30. URL PATH_INFO

31.

32.

33.

34.

35. URL

36. URL_ROUTER_ON

37. PATH_INFO path_info

38. URL URL_DENY_SUFFIX URL_HTML_SUFFIX

39. URL

40. URL url_dispatch

41. app_begin

42. SESSION_OPTIONS Session

43.

44.

45. action_begin

46. ReadHtmlCache HTML_CACHE_ON

47. _initialize

48.

49.

50. Action

51. display

52. view_begin

53. Think\View fetch

54.

55. view_parse

- 56. ParseTemplate
- 57.
- 58. template_filter
- 59. ContentReplace
- 50. view_filter
- 51. WriteHtmlCache HTML_CACHE_ON
- 52. Think\View render
- 53. view_end
- 54.
- 55. action_end
- 56. app_end
- 57. ShowPageTrace SHOW_PAGE_TRACE AJAX
- 58.

	URL	ThinkPHP	URL
PATHINFO	URL		URL

ThinkPHP

-
-
- URL
-

	URL	PATH_INFO	URL	URL
--	-----	-----------	-----	-----

```
//  
'URL_ROUTER_ON' => true,
```

URL_ROUTE_RULES

```
'URL_ROUTE_RULES'=>array(  
    'news/:year/:month/:day' => array('News/archive', 'status=1'),  
    'news/:id'               => 'News/read',  
    'news/read/:id'         => '/news/:1',  
),
```

```
array('id' => '1234')
```

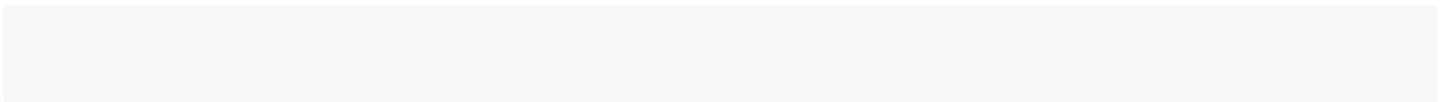
	<code>/^blog\/(\d+)\\$/</code>
	<code>blog/:id</code>

URL

6

1	<code>'[/]? 1= 1& 2= 2...'</code>
2	<code>array([' /], ' 1= 1& 2= 2...')</code>
3	<code>array([' /], array(' 1=> 1', ' 2=> 2...'), [])</code>
4	<code>301</code>
5	<code>array(';', '[])</code>
6	<code>function(\$name){ echo 'Hello, '.\$name;}</code>

{ ec { c



```
'blog/:id' => '/blog/read/id/:1'
```

```
'blog/:id' => 'blog/read'
```

URL 301 URL

avatar/123 /member/avatar/id/123_small

```
'avatar/:id' => '/member/avatar/id/:1_small'
```

:1 :2

URL

```
'blog/:id' => 'http://blog.thinkphp.cn/read/:1'
```

<http://thinkphp.cn> blog/123

http://blog.thinkphp.cn/read/123

301

```
'blog/:id' => array('http://blog.thinkphp.cn/read/:1',302);
```

URL

1= 1& 2= 2

```
array('1'=>' 1','2'=>' 2...')
```

```
'blog/:id' => 'blog/read?status=1&app_id=5',
'blog/:id' => array('blog/read?status=1&app_id=5'),
'blog/:id' => array('blog/read','status=1&app_id=5'),
'blog/:id' => array('blog/read',array('status'=>1,'app_id'=>5)),
```

status app_id URL

URL

```
'blog/:id' => array('blog/read','status=1&app_id=5',array('ext'=>'html')),
```

html

```
'blog/:id' => array('blog/read','status=1&app_id=5',array('method'=>'get')),
```

GET

```
'blog/:id' => array('blog/read','status=1&app_id=5',array('callback'=>'checkFun')),
```

checkFun

false

ThinkPHP

```
'my'      => 'Member/myinfo', //  
'blog/:id' => 'Blog/read', //  
'new/:year/:month/:day' => 'News/read', //  
'user/:blog_id' => 'Blog/read', //
```

" /"

URL_PATHINFO_DEPR

```
    " :"  
        $_GET['id']          :year  :month  :day          :id  
$_GET['month']  $_GET['day']  $_GET['year']  
GET
```

```
'blog/:id\d' => 'Blog/read',
```

```
'blog/:id\d|md5' => 'Blog/read',
```

```
    id      md5      read      $_GET['id']  
md5($_GET['id'])
```

```
'blog/:year\d/[[:month\d]]' => 'Blog/archive',
```

```
[[:month\d]]  [ ]
```

URL

```
http://serverName/index.php/Home/blog/2013  
http://serverName/index.php/Home/blog/2013/12
```


"/"

```
'#^blogV(\d+)$#' => 'Blog/read/id/:1'
```

```
'/^newV(\d{4})V(\d{2})$/' => 'News/achive?year=:1&month=:2',
```

```
:1 :2
```

```
'/^newV(\d{4})V(\d{2})$/' => 'News/achive?year=:1|format_year&month=:2',
```

```
year=:1|format_year
```

```
format_year
```

```
format_year
```

URL

```
'URL_ROUTER_ON' => true,
'URL_MAP_RULES'=>array(
  'new/top' => 'news/index?type=top'
)
```

URL_MAP_RULES

```
http://serverName/Home/new/top
```

```
http://serverName/Home/news/index/type/top
```

```
http://serverName/Home/new/top/var/test
```

new/top

news/index/type/top

URL

http://serverName/Home/new/top.html

[/ ?] 1= 1& 2= 2

```
'URL_ROUTE_RULES'=>array(
  'test' =>
    function(){
      echo 'just test';
    },
  'hello/:name' =>
    function($name){
      echo 'Hello, '.$name;
    }
)
```

```
'hello/:name' =>
function($name){
  echo 'Hello, '.$name;
}
```

URL

http://serverName/Home/hello/thinkphp

Hello,thinkphp


```
'blog/:year/:month' =>
function($year,$month){
    echo 'year= '.$year.'& month= '.$month;
}
```

```
 '/^newV(\d{4})V(\d{2})$/' =>
function($year,$month){
    echo 'year= '.$year.'& month= '.$month;
}
```

http://serverName/Home/new/2013/03

year= 2013& month= 03

```
'hello/:name' =>
function($name){
    echo 'Hello, '.$name.'<br/>';
    $_SERVER['PATH_INFO'] = 'blog/read/name/'.$name;
    return false;
}
```

\$_SERVER['PATH_INFO']

false

Blog

read

blog

read

```
public function read($name){
    echo 'read, '.$name.'!<br/>';
}
```

URL

http://serverName/Home/hello/thinkphp

Hello,thinkphp
read,thinkphp!

News

```
namespace Home\Controller;
use Think\Controller;
class NewsController extends Controller{
    public function read(){
        $New = M('New');
        if(isset($_GET['id'])) {
            // id
            $data = $New->find($_GET['id']);
        }elseif(isset($_GET['name'])){
            // name
            $data = $New->getByName($_GET['name']);
        }
        $this->data = $data;
        $this->display();
    }

    public function archive(){
        $New = M('New');
        $year = $_GET['year'];
        $month = $_GET['month'];
        $begin_time = strtotime($year . $month . "01");
        $end_time = strtotime("+ 1 month", $begin_time);
        $map['create_time'] = array(array('gt',$begin_time),array('lt',$end_time));
        $map['status'] = 1;
        $list = $New->where($map)->select();
        $this->list = $list;
        $this->display();
    }
}
```

```
'URL_ROUTER_ON' => true, //  
'URL_ROUTE_RULES' => array( //  
    'new/:id\d' => 'News/read',  
    'new/:name' => 'News/read',  
    'new/:year\d/:month\d' => 'News/archive',  
),
```

<http://serverName/index.php/Home/new/8>

<http://serverName/index.php/Home/News/read/id/8>

<http://serverName/index.php/Home/new/hello>

<http://serverName/index.php/Home/News/read/name/hello>

<http://serverName/index.php/Home/new/2012/03>

<http://serverName/index.php/Home/News/archive/year/2012/month/03>

<http://serverName/index.php/Home/new/2012/> URL
2012 id

1

```
'URL_ROUTE_RULES' => array( //  
    'new/:year\d/:month\d' => 'News/archive',  
    'new/:id\d' => 'News/read',  
    'new/:name' => 'News/read',  
),
```

<http://serverName/index.php/Home/new/2012/03>

2

```
'URL_ROUTE_RULES' => array( //
    'new/:id\d$'      => 'News/read',
    'new/:name$'     => 'News/read',
    'new/:year\d/:month\d$' => 'News/archive',
),
```

\$

URL

<http://serverName/index.php/Home/new/2012/03/01>

3

```
'URL_ROUTE_RULES' => array( //
    '/^new\\(\d+)$/'   => 'News/read?id=:1',
    '/^new\\(\w+)$/'   => 'News/read?name=:1',
    '/^new\\(\d{4})\\(\d{2})$/' => 'News/achive?year=:1&month=:2',
),
```

ThinkPHP

```
<?php
namespace Home\Controller;
use Think\Controller;
class IndexController extends Controller {
    public function hello(){
        echo 'hello,thinkphp!';
    }
}
```

Home/IndexController Home Index hello

Home/IndexController hello

http://serverName/index.php/Home/Index/hello

hello,thinkphp!

Controller

Home\Controller/IndexController

```
<?php
namespace Home\Controller;
use Think\Controller;
class IndexController extends Controller {
    public function hello(){
        echo 'hello';
    }

    public function test(){
        echo 'test';
    }
}
```

Home/Controller/IndexController.class.php

IndexController	hello	test	URL
-----------------	-------	------	-----

```
http://serverName/Home/Index/hello
http://serverName/Home/Index/test
```

```
hello
//
test
```

hello

test

```
<?php
namespace Home\Controller;
use Think\Controller;
class IndexController extends Controller {
    public function hello(){
        echo 'hello';
    }

    protected function test(){
        echo 'test';
    }
}
```

```
display  
get  
show  
fetch  
theme  
assign  
error  
success
```

```
'ACTION_SUFFIX' => 'Action', //
```

Action

```
<?php  
namespace Home\Controller;  
use Think\Controller;  
class IndexController extends Controller {  
    public function listAction(){  
        echo 'list';  
    }  
  
    public function helloAction(){  
        echo 'hello';  
    }  
  
    public function testAction(){  
        echo 'test';  
    }  
}
```

URL

ThinkPHP

Controller

```

Controller
  UserController.class.php
  BlogController.class.php
...
Event
  UserEvent.class.php
  BlogEvent.class.php
...

```

DEFAULT_C_LAYER

Controller

URL

http://serverName/Home/User/index ,

Controller

Controller

```

<?php
namespace Home\Event;
class UserEvent {
    public function login(){
        echo 'login event';
    }

    public function logout(){
        echo 'logout event';
    }
}

```

UserEvent

Home/Event/UserEvent.class.php

2

```
'CONTROLLER_LEVEL' => 2,
```



```

Controller
  User User
    UserTypeController.class.php
    UserAuthController.class.php
  ...
  Admin Admin
    UserController.class.php
    ConfigController.class.php
  ...

```

```

<?php
namespace Home\Controller\Admin;
use Think\Controller;
class IndexController extends Controller {
    public function hello(){
        echo 'hello';
    }

    public function test(){
        echo 'test';
    }
}

```

URL

```

http://serverName/Home/User/UserType
http://serverName/Home/Admin/User

```

URL

URL

```

//      Home      User
$User = new \Home\Controller\UserController();
//      Admin      Blog
$Blog = new \Admin\Controller\BlogController();

```

A

```
//          Home
$User = A('User');
$Blog = A('Admin/Blog');
```

A Controller

```
//          Home
//      Event
$User = A('User','Event');
$Blog = A('Admin/Blog','Event');
```

```
//      Home      User
$User = new \Home\Event\UserEvent();
//      Admin      Blog
$Blog = new \Admin\Event\BlogEvent();
```

```
<?php
namespace Home\Controller;
use Think\Controller;
class IndexController extends Controller{
    //
    public function _before_index(){
        echo 'before<br/>';
    }
    public function index(){
        echo 'index<br/>';
    }
    //
    public function _after_index(){
        echo 'after<br/>';
    }
}
```

http://serverName/index.php/Home/Index/index

```
before  
index  
after
```

1.

2.

exit

error

success

Action

Action

URL

Action

URL

URL_PARAMS_BIND

```
'URL_PARAMS_BIND' => true, //URL
```

archive

read

id

archive

Blog

year

read

month

```

namespace Home\Controller;
use Think\Controller;
class BlogController extends Controller{
    public function read($id){
        echo 'id=' . $id;
    }

    public function archive($year='2013',$month='01'){
        echo 'year=' . $year . '& month=' . $month;
    }
}

```

URL

```

http://serverName/index.php/Home/Blog/read/id/5
http://serverName/index.php/Home/Blog/archive/year/2013/month/11

```

URL	id	year	month	read	archive
-----	----	------	-------	------	---------

URL

```

id=5
year=2013&month=11

```

URL

```

http://serverName/index.php/Home/Blog/archive/month/11/year/2013

```

URL

URL

```

http://serverName/index.php?s=/Home/Blog/read/id/5
http://serverName/index.php?s=/Home/Blog/archive/year/2013/month/11
http://serverName/index.php?c=Blog&a=read&id=5
http://serverName/index.php?c=Blog&a=archive&year=2013&month=11

```

URL

```
http://serverName/index.php/Home/Blog/read/
```

```
:id
```

```
read
```

```
id
```

```
URL
```

```
id
```

```
read
```

```
id
```

```
public function read($id=0){
    echo 'id= '.$id;
}
```

```
http://serverName/index.php/Home/Blog/read/
```

```
id=0
```

```
http://serverName/index.php/Home/Blog/archive/
```

```
year= 2013& month=01
```

```
URL
```

```
URL_PARAMS_BIND_TYPE 1
```

```
'URL_PARAMS_BIND_TYPE' => 1, //
```

```
URL
```

```
http://serverName/index.php/Home/Blog/read/5
http://serverName/index.php/Home/Blog/archive/2013/11
```

```
id= 5
year= 2013& month= 11
```

```
http://serverName/index.php/Home/Blog/archive/11/2013
```

```
year=11&month=2013
```

URL

PATHINFO

URL

```
http://serverName/index.php?c=Blog&a=read&id=5
```

```
http://serverName/index.php?c=Blog&a=archive&year=2013&month=11
```

URL

```
http://serverName/index.php?s=/Home/Blog/read/5
```

```
http://serverName/index.php?s=/Home/Blog/archive/2013/11
```

```
'URL_PARAMS_BIND' => false
```

URL SEO ThinkPHP URL

URL_HTML_SUFFIX URL

```
'URL_HTML_SUFFIX' => 'html'
```

URL http://serverName/Home/Blog/read/id/1

http://serverName/Home/Blog/read/id/1.html

URL

URL

html


```
http://serverName/index.php/Home/Index/index
http://serverName/index.php/Home/index/index
http://serverName/index.php/home/index/index
```

Linux

URL

Linux

http://serverName/index.php/home/index/index

```
Application/home/Controller/indexController.class.php
```

IndexController

indexController

ThinkPHP

Linux

indexController

Index

URL

```
'URL_CASE_INSENSITIVE' => true
```

Linux

URL

```
http://serverName/index.php/Home/Index/index
//
http://serverName/index.php/home/index/index
```

URL

UserTypeController

URL

```
//
http://serverName/index.php/home/user_type/index
//          linux
http://serverName/index.php/home/usertype/index
```

U

URL

URL

```
'URL_CASE_INSENSITIVE' => false
```

URL

http://serverName/index.php/Home/UserType/add

URL

URL

ThinkPHP URL U URL URL

U

U('[' ']' '[' '])

[/ / # @]? 1= 1& 2= 2...

U('User/add') // User add URL
 U('Blog/read?id=1') // Blog read id 1 URL
 U('Admin/User/select') // Admin User select URL

U

U('Blog/cate',array('cate_id'=>1,'status'=>1))
 U('Blog/cate','cate_id=1&status=1')
 U('Blog/cate?cate_id=1&status=1')

Blog cate cate_id 1 status 1 URL

U('Blog/cate/cate_id/1/status/1');

U

URL


```
http://serverName/index.php/Home/news/1
```

U {U(' 1',' 2'...)}

U

```
U('Blog/read@ blog.thinkphp.cn','id=1');
```

@

SSL https://

U 4 true

APP_SUB_DOMAIN_DEPLOY APP_SUB_DOMAIN_RULES

U URL

```
U('Blog/read#comment?id=1');
```

URL

```
http://serverName/index.php/Home/Blog/read/id/1#comment
```

A J A X

ThinkPHP

A J A X

\Think\Controller

ajaxReturn

A J A X

JSON JSONP XML EVAL

ajaxReturn

```
$data = 'ok';
$this->ajaxReturn($data);
```

```
$data['status'] = 1;
$data['content'] = 'content';
$this->ajaxReturn($data);
```

JSON

DEFAULT_AJAX_RETURN

```
// XML
$data['status'] = 1;
$data['content'] = 'content';
$this->ajaxReturn($data,'xml');
```

data

JSON/JSONP

JSON

XML

XML

EVAL

data

JSON JSONP

JSON

JSONP

JSONP

jsonpReturn

```
'DEFAULT_JSONP_HANDLER' => 'myJsonpReturn', // JSONP
```

callback

ajax_return

\Think\Controller

success error

ajax

```

$User = M('User'); //      User
$result = $User->add($data);
if($result){
    //      $_SERVER['HTTP_REFERER']
    $this->success('      ', '/User/index');
} else {
    //
    $this->error('      ');
}

```

success error

```

//      3      /Article/index
$this->success('      ', '/Article/index',3);
//      5      /Article/error
$this->error('      ', '/Article/error',5);

```

success

`$_SERVER["HTTP_REFERER"]` error

`javascript:history.back(-1);`

success 1 error 3

success error

```

//
'TMPL_ACTION_ERROR' => THINK_PATH . 'Tpl/dispatch_jump.tpl',
//
'TMPL_ACTION_SUCCESS' => THINK_PATH . 'Tpl/dispatch_jump.tpl',

```

```

//
'TMPL_ACTION_ERROR' => 'Public:error',
//
'TMPL_ACTION_SUCCESS' => 'Public:success',

```

\$message	
\$error	
\$waitSecond	

\$jumpUrl	

success error

Ajax

Ajax

ajaxReturn

ajax

success error

```
$data['info'] = $message; //
$data['status'] = $status; // success 1 error 0
$data['url'] = $jumpUrl; //
```

Controller redirect

redirect

U

URL

```
// New Category
$this->redirect('New/category', array('cate_id' => 2), 5, '...');
```

5

New category

URL

URL

redirect

```
// URL
redirect('/New/category/cate_id/2', 5, '...')
```

Redirect

URL

redirect

redirect

URL

URL

Web

ThinkPHP

```

$id = $_GET['id']; //    get
$name = $_POST['name']; //    post
$value = $_SESSION['var']; //    session
$name = $_COOKIE['name']; //    cookie
$file = $_SERVER['PHP_SELF']; //    server

```

|

| ThinkPHP

```

I(' . / '[' ']' ']' ')

```

get	GET
post	POST
param	GET POST PUT
request	REQUEST
put	PUT
session	\$_SESSION
cookie	\$_COOKIE
server	\$_SERVER
globals	\$GLOBALS
path	PATHINFO URL
data	

GET

|

```

echo I('get.id'); //    $_GET['id']
echo I('get.name'); //    $_GET['name']

```

```
echo I('get.id',0); //      $_GET['id']      0
echo I('get.name',''); //      $_GET['name']
```

```
//      htmlspecialchars      $_GET['name']
echo I('get.name','','htmlspecialchars');
```

```
//      $_GET
I('get.');
```

post

```
I('post.name','','htmlspecialchars'); //      htmlspecialchars      $_POST['name']

I('session.user_id',0); //      $_SESSION['user_id']      0
I('cookie.');//      $_COOKIE
I('server.REQUEST_METHOD');//      $_SERVER['REQUEST_METHOD']
```

param

```
echo I('param.id');
```

	GET		\$_GET['id']		POST	PUT
\$_POST['id']	PUT	id				
param	I			param		

```
I('id');//      I('param.id')
I('name');//      I('param.name')
```

path	URL	PATHINFO	GET	POST
	URL	<code>http://serverName/index.php/New/2013/06/01</code>		

```
echo I('path.1');//      2013
echo I('path.2');//      06
echo I('path.3');//      01
```

data


```
int
boolean
float
validate_regexp
validate_url
validate_email
validate_ip
string
stripped
encoded
special_chars
unsafe_raw
email
url
number_int
number_float
magic_quotes
callback
```

```
//
l('get.name','','/^ [A-Za-z]+ $/');
l('get.id',0,'/^ \d+ $/');
```

DEFAULT_FILTER

```
//
l('get.name','');
l('get.id','',false);
```

false

|

```
l( . / ' )
```

```
l('get.id/d'); //  
l('post.name/s'); //  
l('post.ids/a'); //
```

s	
d	
b	
a	
f	

GET POST PUT DELETE

IS_GET	GET
IS_POST	POST
IS_PUT	PUT
IS_DELETE	DELETE
IS_AJAX	AJAX
REQUEST_METHOD	

```

class UserController extends Controller{
    public function update(){
        if (IS_POST){
            $User = M('User');
            $User->create();
            $User->save();
            $this->success('');
        }else{
            $this->error('');
        }
    }
}

```

ajax VAR_AJAX_SUBMIT JQUERY ajax

URL

_empty

CityController

_empty

```

<?php
namespace Home\Controller;
use Think\Controller;
class CityController extends Controller{
    public function _empty($name){
        //                      city
        $this->city($name);
    }

    //                      city                      protected
    protected function city($name){
        // $name
        echo '                      ' . $name;
    }
}

```

```
http://serverName/index.php/Home/City/beijing/  
http://serverName/index.php/Home/City/shanghai/  
http://serverName/index.php/Home/City/shenzhen/
```

City	beijing	shanghai	shenzhen
<code>_empty</code>	<code>_empty</code>		URL

```
:beijing  
:shanghai  
:shenzhen
```

Think\Controller

`__call`

(EmptyController)

URL

URL

```
http://serverName/index.php/Home/City/shanghai/
```

```
http://serverName/index.php/Home/shanghai/
```

index

EmptyController

```
<?php
namespace Home\Controller;
use Think\Controller;
class EmptyController extends Controller{
    public function index(){
        //
        $cityName = CONTROLLER_NAME;
        $this->city($cityName);
    }
    // city protected
    protected function city($name){
        // $name
        echo ' ' . $name;
    }
}
```

```
http://serverName/index.php/Home/beijing/
http://serverName/index.php/Home/shanghai/
http://serverName/index.php/Home/shenzhen/
```

```

        beijing  shanghai  shenzhen
Emptyz          n
```

```
namespace Addon\SystemInfo\Controller;
class InfoController extends \Think\Controller{
    public function index(){
        echo 'Addon SystemInfo';
    }
}
```

`http://serverName/Home/info/index/addon/SystemInfo`

Addon SystemInfo

Addon Plugin

```
'VAR_ADDON' => 'plugin'
```

URL

`http://serverName/Home/info/index/plugin/SystemInfo`

ThinkPHP3.2

ACTION_BIND_CLASS

```
'ACTION_BIND_CLASS' => True,
```

URL

`http://serverName/Home/Index/index`

Application/Home/Controller/IndexController.class.php

```
namespace Home\Controller;
use Think\Controller;
class IndexController extends Controller{
    public function index(){
        echo 'Index      index      ';
    }
}
```

Application/Home/Controller/Index/index.class.php

```
namespace Home\Controller\Index;
use Think\Controller;
class index extends Controller{
    public function run(){
        echo '    Index        index    ';
    }
}
```

Home\Controller\Index\index

run

run

Action

A

R

R

`_before_run``_after_run`

```
namespace Home\Controller\Index;
use Think\Controller;
class index extends Controller{
    public function _before_run(){
        echo 'before_'.ACTION_NAME;
    }

    public function run(){
        echo '    Index        index    ';
    }

    public function _after_run(){
        echo 'after_'.ACTION_NAME;
    }
}
```

Application/Home/Controller/_empty

`_empty`

URL `http://serverName/Home/Test/index`,`Application/Home/Controller/Test``Application/Home/Controller/_empty``Application/Home/Controller/_empty/index.class.php`

```

namespace Home\Controller\_empty;
use Think\Controller;
class index extends Controller{
    public function run(){
        echo '    ' . 'CONTROLLER_NAME.' . '    ' . 'ACTION_NAME.' . '    ';
    }
}

```

`http://serverName/Home/Test/index`

Test index

`Home\Controller\Index_empty`

Index

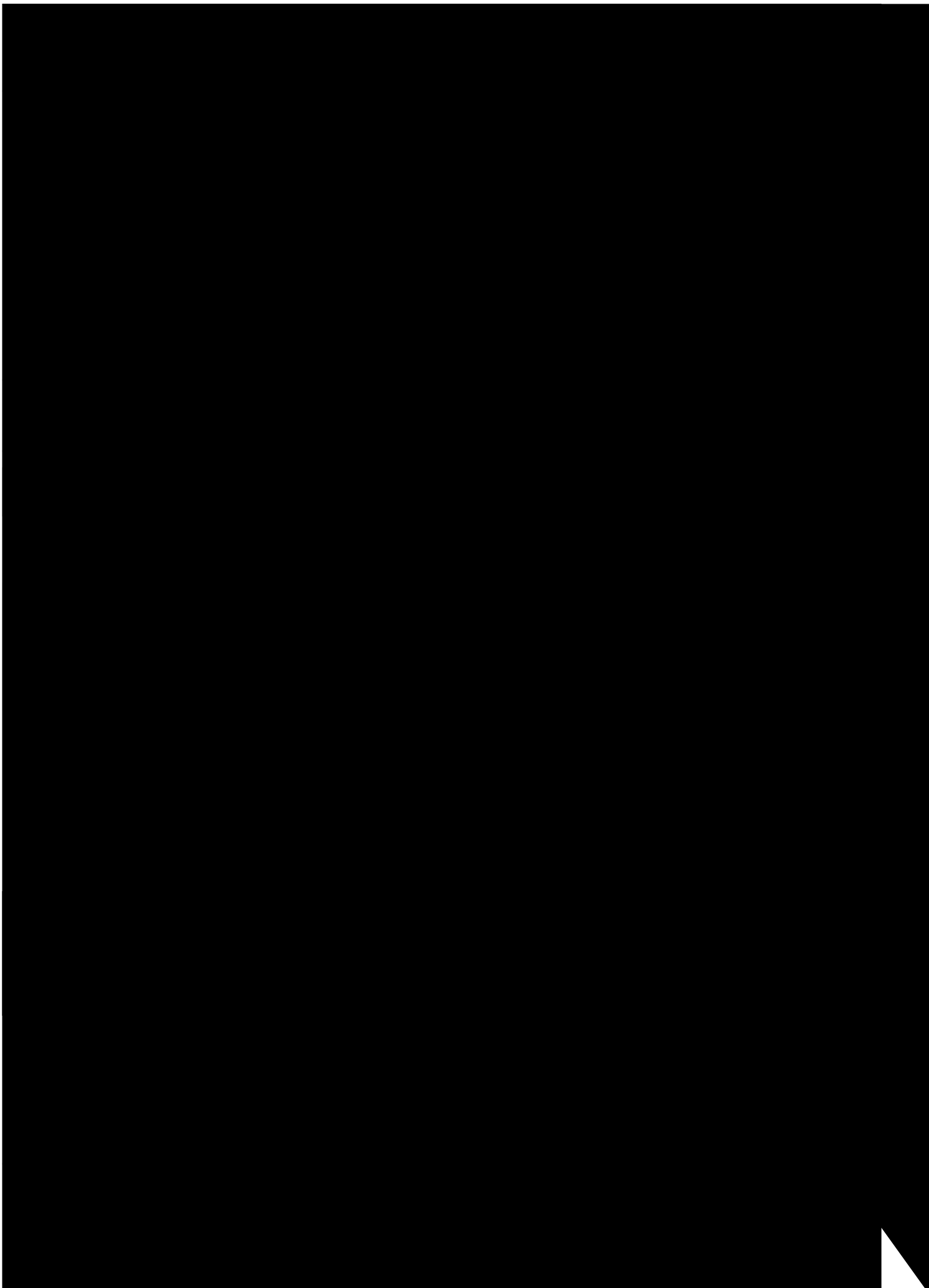
```

namespace Home\Controller\Index;
use Think\Controller;
class _empty extends Controller{
    public function run(){
        echo '    ' . 'Index' . '    ' . 'ACTION_NAME.' . '    ';
    }
}

```

`http://serverName/Home/Index/test`

Index test



tablePrefix	DB_PREFIX
tableName	
trueTableName	
dbName	

think_categories

CategoryModel

Category

think_category

think_categories

tableName

DB_PREFIX

think_

```
namespace Home\Model;
use Think\Model;
class CategoryModel extends Model {
    protected $tableName = 'categories';
}
```

think_

CategoryModel

top_category

```
namespace Home\Model;
use Think\Model;
class CategoryModel extends Model {
    protected $tablePrefix = 'top_';
}
```

category

tablePrefix

```
namespace Home\Model;
use Think\Model;
class CategoryModel extends Model {
    protected $tablePrefix = '';
}
```

DB_PREFIX

top_categories

trueTableName

```
namespace Home\Model;
use Think\Model;
class CategoryModel extends Model {
    protected $trueTableName = 'top_categories';
}
```

trueTableName

top.top_categories

```
namespace Home\Model;
use Think\Model;
class CategoryModel extends Model {
    protected $trueTableName = 'top_categories';
    protected $dbName = 'top';
}
```

tableName

trueTableName

trueTableName

ThinkPHP

ThinkPHP

```
$User = new \Home\Model\UserModel();
$info = new \Admin\Model\InfoModel();
//
$New = new \Home\Model\NewModel('blog','think',$connection);
```

\Think\Model

Model(['', ''], [''], [''], ['']);

null

1

DSN

:// : @ IP: / #

```
new \Home\Model\NewModel('blog','think_', 'mysql://root:1234@localhost/demo');
```

2

```
$connection = array(
    'db_type' => 'mysql',
    'db_host' => '127.0.0.1',
    'db_user' => 'root',
    'db_pwd' => '12345',
    'db_port' => 3306,
    'db_name' => 'demo',
    'db_charset' => 'utf8',
);
new \Home\Model\NewModel('new','think_',$connection);
```

```

$connection = array(
    'db_type' => 'mysql',
    'db_host' => '192.168.1.2,192.168.1.3',
    'db_user' => 'root',
    'db_pwd' => '12345',
    'db_port' => 3306,
    'db_name' => 'demo',
    'db_charset' => 'utf8',
    'db_deploy_type' => 1,
    'db_rw_separate' => true,
    'db_debug' => true,
);
//
new \Home\Model\NewModel('new','think_',$connection);

```

db_debug

APP_DEBUG

3

```

//          1
'DB_CONFIG1' => array(
    'db_type' => 'mysql',
    'db_user' => 'root',
    'db_pwd' => '1234',
    'db_host' => 'localhost',
    'db_port' => '3306',
    'db_name' => 'thinkphp'
),
//          2
'DB_CONFIG2' => 'mysql://root:1234@localhost:3306/thinkphp',

```

```

new \Home\Model\NewModel('new','think_','DB_CONFIG1');
new \Home\Model\BlogModel('blog','think_','DB_CONFIG2');

```

```
'DB_TYPE' => ", //
'DB_HOST' => ", //
'DB_NAME' => ", //
'DB_USER' => ", //
'DB_PWD' => ", //
'DB_PORT' => ", //
'DB_PREFIX' => ", //
'DB_DSN' => ", //          DSN      PDO
'DB_CHARSET' => 'utf8', //          utf8
```

D

D

8

M

D

CURD

M

```
// M
$user = M('User');
// $user = new \Think\Model('User');
//
$user->select();
```

M

```
// M db_name ot_user
$user = M('db_name.User','ot_');
//
$user->select();
```

M

\Think\Model

```
$New = M('new','think_',$connection);
// $New = new \Think\Model('new','think_',$connection);
```

M

\Think\Model

```
$User = M('\Home\Model\CommonModel:User','think_','db_config');
// $User = new \Home\Model\CommonModel('User','think_','db_config');
```

M

M

SQL


```
//
$model = new Model();
//      M
$model = M();
//      SQL
$model->query('SELECT * FROM think_user WHERE status = 1');
```

table

	D	M		M
	D			M

Runtime/Data/_fields/

. + .php

```
demo.think_user.php // User
demo.top_article.php // Article
```

M D

DB_FIELDS_CACHE

```
//
'DB_FIELDS_CACHE' => false
```

```
$User = M('User');
$fields = $User->getDbFields();
```

Data/_fields

10

```
namespace Home\Model;
use Think\Model;
class UserModel extends Model {
    protected $fields = array('id', 'username', 'email', 'age');
    protected $pk    = 'id';
}
```

pk

id

id

```
namespace Home\Model;
use Think\Model;
class ScoreModel extends Model {
    protected $fields = array('user_id', 'lesson_id', 'score');
    protected $pk    = array('user_id', 'lesson_id');
}
```

```
namespace Home\Model;
use Think\Model;
class UserModel extends Model {
    protected $fields = array('id', 'username', 'email', 'age',
        '_type' => array('id' => 'bigint', 'username' => 'varchar', 'email' => 'varchar', 'age' => 'int')
    );
}
```

```
//
'DB_TYPE' => 'mysql', //
'DB_HOST' => '127.0.0.1', //
'DB_NAME' => 'thinkphp', //
'DB_USER' => 'root', //
'DB_PWD' => '123456', //
'DB_PORT' => 3306, //
'DB_PARAMS' => array(), //
'DB_PREFIX' => 'think_', //
'DB_CHARSET' => 'utf8', //
'DB_DEBUG' => TRUE, //
```

SQL

DB_TYPE

DB_TYPE	
mysql	mysql
pgsql	pgsql
sqlite	sqlite
sqlsrv	sqlserver
oracle	oracle
firebird	ibase
mongo	mongo

```
'DB_PARAMS' => array(PDO::ATTR_PERSISTENT => true),
```

DB_PARAMS

PDO

connection

```
//  
namespace Home\Model;  
use Think\Model;  
class UserModel extends Model{  
    protected $connection = array(  
        'db_type' => 'mysql',  
        'db_user' => 'root',  
        'db_pwd' => '1234',  
        'db_host' => 'localhost',  
        'db_port' => '3306',  
        'db_name' => 'thinkphp',  
        'db_charset' => 'utf8',  
        'db_params' => array(), //  
    );  
}
```

```
://      :    @      :      /      #
```

```
//  
namespace Home\Model;  
use Think\Model;  
class UserModel extends Model{  
    //  
    protected $connection = 'mysql://root:1234@localhost:3306/thinkphp#utf8';  
}
```

```
//          1
'DB_CONFIG1' => array(
    'db_type' => 'mysql',
    'db_user' => 'root',
    'db_pwd' => '1234',
    'db_host' => 'localhost',
    'db_port' => '3306',
    'db_name' => 'thinkphp',
    'db_charset' => 'utf8',
),
//          2
'DB_CONFIG2' => 'mysql://root:1234@localhost:3306/thinkphp#utf8';
```

```
//
namespace Home\Model;
use Think\Model;
class UserModel extends Model{
    //          1
    protected $connection = 'DB_CONFIG1';
}
```

```
//
namespace Home\Model;
use Think\Model;
class InfoModel extends Model{
    //          1
    protected $connection = 'DB_CONFIG2';
}
```

M

```
$User = M('User','other_', 'mysql://root:1234@localhost/demo#utf8');
```

```
User          demo          other_user
                DB_CONFIG2
```

```
$User = M('User','other_', 'DB_CONFIG2');
```

```
Model->db("      ","      ");
```

```
0
0
```

connection

Db

```
$this->db(1,"mysql://root:123456@localhost:3306/test")->query("      SQL");
```

1

```
$this->db(1)->query("      SQL");
```

```
$this->db(0);
```

```
//      1
'DB_CONFIG1' = array(
    'db_type' => 'mysql',
    'db_user' => 'root',
    'db_pwd' => '1234',
    'db_host' => 'localhost',
    'db_port' => '3306',
    'db_name' => 'thinkphp'
),
//      2
'DB_CONFIG2' => 'mysql://root:1234@localhost:3306/thinkphp';
```

db

```
$this->db(1,"DB_CONFIG1")->query(" SQL");
$this->db(2,"DB_CONFIG2")->query(" SQL");
```

table

```
$this->db(1)->table(" top_user")->find();
```

ThinkPHP

DB_DEPLOY_TYPE 1

```
//
'DB_DEPLOY_TYPE'=> 1, //
'DB_TYPE'      => 'mysql', //
'DB_HOST'      => '192.168.0.1,192.168.0.2',
'DB_NAME'      => 'thinkphp', //
'DB_USER'      => 'user1,user2',
'DB_PWD'       => 'pwd1,pwd2',
'DB_PORT'      => '3306',
'DB_PREFIX'    => 'think_',
```

DB_HOST

IP

```
'DB_PORT'=>'3306,3306'
```

```
'DB_PORT'=>'3306'
```

```
'DB_USER'=>'user1',
'DB_PWD'=>'pwd1',
```

```
'DB_USER'=>'user1,user1',  
'DB_PWD'=>'pwd1,pwd1',
```

```
'DB_RW_SEPARATE'=>true,
```

```
DB_MASTER_NUM
```

```
DB_SLAVE_NO
```

```
3.2.3
```

```
CURD
```

```
SQL
```

```
execute
```

```
query
```

```
ThinkPHP
```

```
CURD
```

```
User
```

```
1 10
```

```
$User->info >
```


validate		
auto		
filter		
scope*		
bind*		
token		
comment	SQL	
index		3.2.3
strict		3.2.3

this *

WHERE

where

ThinkPHP

ThinkPHP ORM

where

```
$User = M("User"); // User
>User->where('type=1 AND status=1')->select();
```

SQL

```
SELECT * FROM think_user WHERE type=1 AND status=1
```

```
$Model->where("id=%d and username='%s' and xx='%f'",array($id,$username,$xx))->select();
```

```
$Model->where("id=%d and username='%s' and xx='%f'", $id, $username, $xx)->select();
```

\$id

URL

vsprintf

where

ThinkPHP

```
$User = M("User"); // User
$map['name'] = 'thinkphp';
$map['status'] = 1;
//
$User->where($map)->select();
```

SQL

```
SELECT * FROM think_user WHERE `name`='thinkphp' AND status=1
```

SQL

```
$map[' 1'] = array(' 1', ' 1');
$map[' 2'] = array(' 2', ' 2');
$Model->where($map)->select(); //
```

EQ	=
NEQ	<>
GT	>
EGT	>=
LT	<

ELT	< =
LIKE	
[NOT] BETWEEN	
[NOT] IN	IN
EXP	SQL

where

```
$map['a'] = array('gt',1);
$where['b'] = 1;
$Model->where($map)->where($where)->where('status=1')->select();
```

TABLE

table

table

- 1.
- 2.

```
$Model->table('think_user')->where('status>1')->select();
```

table

```
$Model->table('db_name.think_user')->where('status>1')->select();
```

table

```
$Model-> table('__USER__')-> where('status>1')-> select();
```

think_user

table

```
$Model-> field('user.name,role.title')
-> table('think_user user,think_role role')
-> limit(10)-> select();
```

mysql

```
$Model-> field('user.name,role.title')
-> table(array('think_user'=>'user','think_role'=>'role'))
-> limit(10)-> select();
```

table

ALIAS

alias

join

```
$Model = M('User');
$Model-> alias('a')-> join('__DEPT__ b ON b.user_id= a.id')-> select();
```

SQL

```
SELECT * FROM think_user a INNER JOIN think_dept b ON b.user_id= a.id
```

DATA

data

create

```
$Model = D('User');
$Model->create();
//
$Model->add();
```

```
$Model = M('User');
$Model->name = '    ';
$Model->email = 'thinkphp@qq.com';
$Model->add();
```

data

```
$Model = M('User');
$data['name'] = '    ';
$data['email'] = 'thinkphp@qq.com';
$Model->data($data)->add();
```

create

data

data

```
$Model = M('User');
$obj = new \stdClass;
$obj->name = '    ';
$obj->email = 'thinkphp@qq.com';
$Model->data($obj)->add();
```

```
$Model = M('User');
$data = 'name=    &email=thinkphp@qq.com';
$Model->data($data)->add();
```

add

```
$Model = M('User');
$data['name'] = 'thinkphp';
$data['email'] = 'thinkphp@qq.com';
$Model->add($data);
```

data

data

```
$Model = M('User');
$data['id'] = 8;
$data['name'] = 'thinkphp';
$data['email'] = 'thinkphp@qq.com';
$Model->data($data)->save();
```

```
$Model = M('User');
$data['id'] = 8;
$data['name'] = 'thinkphp';
$data['email'] = 'thinkphp@qq.com';
$Model->save($data);
```

data

save

```
$Model = M('User');
$data['name'] = 'thinkphp';
$data['email'] = 'thinkphp@qq.com';
$Model->data($data)->where('id=8')->save();
```

data

```
$User = M('User');
$map['name'] = 'thinkphp';
$User->where($map)->find();
//
$data = $User->data();
```

FIELD

field

field

```
$Model->field('id,title,content')->select();
```

field

id,title,content

SQL

```
SELECT id,title,content FROM table
```

```
$Model->field('id,nickname as name')->select();
```

SQL

```
SELECT id,nickname as name FROM table
```

SQL

field

```
$Model->field('id,SUM(score)')->select();
```

SQL

```
SELECT id,SUM(score) FROM table
```

select

find

field

field

```
$Model->field(array('id','title','content'))->select();
```



```
$Model->field('content',true)->select();
```

content

```
$Model->field('user_id,content',true)->select();
//
$Model->field(array('user_id','content'),true)->select();
```

field

--

field

create

```
$Model->field('title,email,content')->create();
```

title , email content

field

add save

```
$Model->field('title,email,content')->save($data);
```

data

title,email,content

ORDER

order

```
$Model->where('status=1')->order('id desc')->limit(5)->select();
```

select

```
$Model->where('status=1')->order('id desc,status')->limit(5)->select();
```

desc

asc

asc

mysql

```
$Model->where('status=1')->order(array('order','id'=>'desc'))->limit(5)->select();
```

LIMIT

limit

ThinkPHP limit

10







PAGE

page

limit

page

limit

10

```
$Article = M('Article');
$Article->limit('0,10')->select(); //
$Article->limit('10,10')->select(); //
```

Page

limit

page

```
$Article = M('Article');
$Article->page('1,10')->select(); //
$Article->page('2,10')->select(); //
```

page

page

limit

page

2

```
$Article->page(1,10)->select();
//
$Article->page('1,10')->select();
```

page

limit

```
$Article->limit(25)->page(3)->select();
```

page

limit

```
$Article->page('3,25')->select();
```

GROUP

GROUP

- LEFT JOIN :
- RIGHT JOIN :
- FULL JOIN :

join

```
$Model = M('Artist');  
$Model  
->join('think_work ON think_artist.id = think_work.artist_id')  
->join('think_card ON think_artist.card_id = think_card.id')  
->select();
```

join

```
$Model  
->join('__WORK__ ON __ARTIST__.id = __WORK__.artist_id')  
->join('__CARD__ ON __ARTIST__.card_id = __CARD__.id')  
->select();
```

`__WORK__`

`__CARD__`

`think_work`

`think_card`

UNION

SELECT

```
$Model->field('name')
->table('think_user_0')
->union('SELECT name FROM think_user_1')
->union('SELECT name FROM think_user_2')
->select();
```

```
$Model->field('name')
->table('think_user_0')
->union(array('field'=>'name','table'=>'think_user_1'))
->union(array('field'=>'name','table'=>'think_user_2'))
->select();
```

```
$Model->field('name')
->table('think_user_0')
->union(array('SELECT name FROM think_user_1','SELECT name FROM think_user_2'))
->select();
```

UNION ALL

```
$Model->field('name')
->table('think_user_0')
->union('SELECT name FROM think_user_1',true)
->union('SELECT name FROM think_user_2',true)
->select();
```

```
$Model->field('name')
->table('think_user_0')
->union(array('SELECT name FROM think_user_1','SELECT name FROM think_user_2'),true)
->select();
```

union n

DISTINCT

DISTINCT

```
$Model->distinct(true)->field('name')->select();
```

SQL

```
SELECT DISTINCT name FROM think_user
```

distinct

LOCK

Lock

```
lock(true);
```

SQL

```
FOR UPDATE
```

```
FOR UPDATE NOWAIT
```

Oracle

CACHE

cache

cache

```
select
```

```
find
```

```
getField
```

cache

find

cache

```
$Model = M('User');
$Model->where('id=5')->cache(true)->find();
```

DATA_CACHE_TIME DATA_CACHE_TYPE

cache


```
$Model = M('User');
$Model->cache(true,60,'xcache')->find();
```

xcache

60

cache

```
$Model = M('User');
$Model->cache('key',60)->find();
```

S

```
$Model = M('User');
$result = $Model->cache('key',60)->find();
$data = S('key');
```

COMMENT

COMMENT

SQL

```
$this->comment('          ')
->field('username,score')
->limit(10)
->order('score desc')
->select();
```

SQL

```
SELECT username,score FROM think_score ORDER BY score desc LIMIT 10 /*
```

*/

RELATION

USING

fetchSql

fetchSql

SQL

CURD

```
$result = M('User')->fetchSql(true)->find(1);
```

result

```
SELECT * FROM think_user where id = 1
```

TOKEN

token

```
$model->token(false)->create();
```

TOKEN_ON

STRICT

strict 3.2.3

:

```
$model->strict(true)->add($data);
```

INDEX

index

```
$Model->index('user')->select();
```

user

user

CURD

`_scope``scope``_scope`

```

namespace Home\Model;
use Think\Model;
class NewsModel extends Model {
    protected $_scope = array(
        //          normal
        'normal' => array(
            'where' => array('status' => 1),
        ),
        //          latest
        'latest' => array(
            'order' => 'create_time DESC',
            'limit' => 10,
        ),
    );
}

```

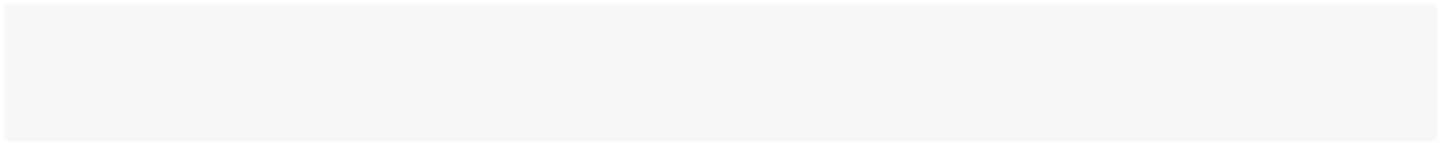
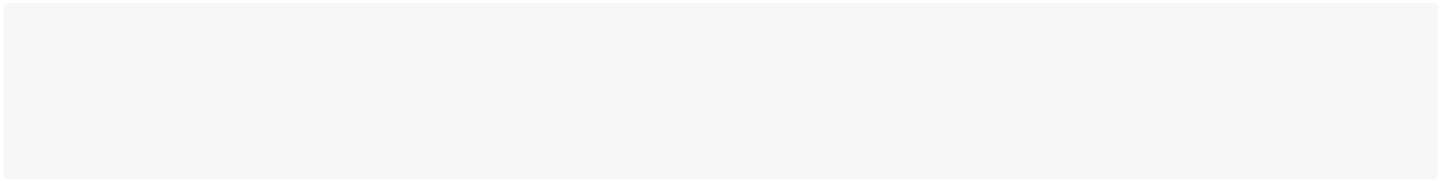
`_scope`

```

    'normal' => array(
        '1' => ' 1',
        '2' => ' 2',
        ...
    )

```

where	
field	
order	
table	



```
SELECT * FROM think_news WHERE status=1 ORDER BY create_time DESC LIMIT 10
```

```
$Model->scope('normal,new')->select();
```

new

normal

SQL

```
SELECT * FROM think_news WHERE status=1
```

default

```
protected $_scope = array(  
    //  
    'default' => array(  
        'where' => array('status' => 1),  
        'limit' => 10,  
    ),  
);
```

default

```
$Model->scope()->select();
```

```
$Model->scope('default')->select();
```

0

normal


```
$Model->limit(8)->scope('normal')->order('id desc')->select();
```

SQL

```
SELECT id,title FROM think_news WHERE status= 1 ORDER BY id desc LIMIT 10
```

scope

```
$Model->scope('normal',array('limit'=>5))->select();
```

```
$Model->normal(array('limit'=>5))->select();
```

normal(array('limit'=>5))

normal

array('limit'=>5)

__call

CURD

ThinkPHP

CURD

CURD

```
//      POST
$data['name'] = $_POST['name'];
$data['email'] = $_POST['email'];
//
//.....
```

ThinkPHP

```
//      User
$user = M('User');
//      POST
$user->create();
```

Create

```
$data['name'] = 'ThinkPHP';
$data['email'] = 'ThinkPHP@ gmail.com';
$user->create($data);
```

```
//      User      Member
$user = stdClass();
$user->name = 'ThinkPHP';
$user->email = 'ThinkPHP@ gmail.com';
$member = M("Member");
$member->create($user);
```

```
$data['name'] = 'ThinkPHP';
$data['email'] = 'ThinkPHP@ gmail.com';
$user->create($data);
//
echo $user->name;
echo $user->email;
//
$user->name = 'onethink'; //      name
$user->status = 1; //
```


create

```
$Member = M("User");
//
$Member->create($_POST,Model::MODEL_UPDATE);
```

Model::MODEL_INSERT 1 Model::MODEL_UPDATE 2

Model::MODEL_UPDATE

3

```
$Member = M("User");
//
$Member->create($_POST,3);
```

create

create

1	POST	
2		false
3		
4		
5		false
6		false
7		
8		
9		

create

create

```
$data['name'] = 'thinkphp';  
$data['email'] = 'thinkphp@gmail.com';  
$data['status'] = 1;  
$User = M('User');  
$data = $User->create($data);  
dump($data);
```

```
array (siz
```

field		
validate		
auto		
token		

create

field

```
$data['name'] = 'thinkphp';  
$data['email'] = 'thinkphp@gmail.com';  
$data['status'] = 1;  
$data['test'] = 'test';  
$User = M('User');  
$data = $User->field('name,email')->create($data);  
dump($data);
```

```
array (size=2)  
'name' => string 'thinkphp' (length=8)  
'email' => string 'thinkphp@gmail.com' (length=18)
```

name

ThinkPHP

add

```

$user = M("User"); // User
$data['name'] = 'ThinkPHP';
$data['email'] = 'ThinkPHP@gmail.com';
$user->add($data);

```

Mysql

:

```
add($data="", $options=array(), $replace=false)
```

```

add($data, $replace (true/false)
data

```

```

$user = M("User"); // User
$user->data($data)->add();

```

add

create

data

add

create

```

$user = M("User"); // User
// POST
if($user->create()){
    $result = $user->add(); //
    if($result){
        //
        $insertId = $result;
    }
}

```

create

add

table		
data		
field		

relation		
validate		
auto		
filter		
scope		
bind		
token		
comment	SQL	
fetchSql	SQL	SQL

SQL

SQL

```

$user = M("User"); //      User
$data['name'] = 'ThinkPHP';
$data['email'] = 'ThinkPHP@ gmail.com';
$sql = $user->fetchSql(true)->add($data);
echo $sql;
//
// INSERT INTO think_user (name,email) VALUES ('ThinkPHP','ThinkPHP@ gmail.com')

```

```

$data['name'] = 'thinkphp';
$data['email'] = 'thinkphp@ gmail.com';
$data['test'] = 'test';
$user = M('User');
$user->data($data)->add();

```

test

3.2.2

[test=> test]

add

field

```
$data['name'] = 'thinkphp';
$data['email'] = 'thinkphp@gmail.com';
$data['test'] = 'test';
$user = M('User');
$user->field('name')->data($data)->add();
```

name

email test

email

filter

```
$data['name'] = '<b>thinkphp</b>';
$data['email'] = 'thinkphp@gmail.com';
$user = M('User');
$user->data($data)->filter('strip_tags')->add();
```

name

thinkphp

filter

```
//
$dataList[] = array('name'=>'thinkphp','email'=>'thinkphp@gmail.com');
$dataList[] = array('name'=>'onethink','email'=>'onethink@gmail.com');
$user->addAll($dataList);
```

3.2.3

3.2.3

mysql

ThinkPHP

where		
table		

--	--	--

alias		
field		
order		
group	group	
having	having	
join	join	
union	union	
distinct	distinct	
lock		
cache		
relation		
result		
scope		
bind		
comment	SQL	
fetchSql	SQL	SQL

limit find

find

```
$User = M("User"); //      User
//      status      1name      think
$data = $User->where('status=1 AND name="thinkphp")->find();
dump($data);
```

find

where

where

find

false

NULL

```
array (size= 3)
  'name' => string 'thinkphp' (length=8)
  'email' => string 'thinkphp@gmail.com' (length=18)
  'status'=> int 1
```

find

order

data

```
$User = M("User"); //      User
//      status    1name    think
$user->where('status= 1 AND name=" thinkphp")->find();
dump($User->data());
```

select

```
$User = M("User"); //      User
//      status    1                                10
$list = $User->where('status= 1')->order('create_time')->limit(10)->select();
```

select

false

NULL

getField

```
$User = M("User"); //      User
//      ID    3
$nickname = $User->where('id= 3')->getField('nickname');
```

```
$User->getField('id',true); //      id
//      array(1,2,3,4,5)      value    id
```



```
$User = M("User"); //      User
//      ID
$list = $User->getField('id,nickname');
//      array('id'=>'nickname')      id      key  nickname      value
```

list

id

nickname

```
$list = $User->getField('id,nickname,email');
//      array('id'=>array('id'=>value,'nickname'=>value,'email'=>value))
key  id      value      array      select()      id      key
```

select

```

$user = M("User"); //      User
//
$user->name = 'ThinkPHP';
$user->email = 'ThinkPHP@ gmail.com';
$user->where('id=5')->save(); //

```

save

save

false

save

```

$user->save($data);

```

```

$user = M("User"); //      User
//
$data['id'] = 5;
$data['name'] = 'ThinkPHP';
$data['email'] = 'ThinkPHP@ gmail.com';
$user->save($data); //

```

id

where		
table		
alias		
field		
order		
lock		
relation		
scope		

bind			
comment	SQL		
fetchSql	SQL	SQL	

add

save

field

filter

```

$User = M("User"); //      User
//
$data['name'] = 'test';
$data['email'] = '<b>test@ gmail.com </b>';
$User-> where('id= 5')-> field('email')-> filter('strip_tags')-> save($data); //

```

field('email')

email

strip_tags

name

create

data

save

```

$User = M("User"); //      User
//
$data['name'] = 'ThinkPHP';
$data['email'] = 'ThinkPHP@ gmail.com';
$User-> where('id= 5')-> data($data)-> save(); //

```

create

```

$User = M("User"); //      User
//          POST
$User-> create();
$User-> save(); //

```

setField

```

$User = M("User"); //      User
//          name
$User-> where('id= 5')-> setField('name','ThinkPHP');

```

setField

```
$User = M("User"); //      User
//      name  email
$data = array('name'=>'ThinkPHP','email'=>'ThinkPHP@gmail.com');
$user->where('id=5')->setField($data);
```

`setInc` `setDec`

```
$User = M("User"); //      User
$user->where('id=5')->setInc('score',3); //      3
$user->where('id=5')->setInc('score'); //      1
$user->where('id=5')->setDec('score',5); //      5
$user->where('id=5')->setDec('score'); //      1
```

3.2.3 `setInc` `setDec`

```
$Article = M("Article"); //      Article
$Article->where('id=5')->setInc('view',1); //      1
$Article->where('id=5')->setInc('view',1,60); //      1      60
```

ThinkPHP `delete`

```
$Form = M('Form');
$form->delete(5);
```

`5` `delete`

```
$User = M("User"); //      User
$user->where('id=5')->delete(); //      id  5
$user->delete('1,2,5'); //      1,2  5
$user->where('status=0')->delete(); //      0
```

`delete` `false` `SQL` `0`

`order` `limit`


```
$User = M("User"); //      User
//
$User->name = 'ThinkPHP';
$User->email = 'ThinkPHP@gmail.com';
//
$User->add();
```

create

```
$User = D("User");
$User->create(); //      User
//
$User->status = 1;
$User->create_time = time();
//
$User->add();
```

AR

ThinkPHP

8

```
$User = M("User"); //      User
//      id 8
$User->where('id=8')->find();
```

AR

```
$User->find(8);
```

ThinkPHP

```
$User = M("User"); //      User
$User->getByName("ThinkPHP");
```

```
echo $User->name;
echo $User->email;
```

```
//          1 3 8  
$userList = $User->select('1,3,8');
```

```
$User->find(1); //          1  
$User->name = 'TOPThink'; //  
$User->save(); //
```

```
$User->id = 1;  
$User->name = 'TOPThink'; //  
$User->save(); //
```

```
$User->find(2);  
$User->delete(); //
```

```
$User->delete(8); //          8  
$User->delete('5,6'); //          5 6
```

ThinkPHP

User	username	email
------	----------	-------


```
//      User
$User = D('User');
$data = $User->find(3);
$data = $User->parseFieldsMap($data);
```

	find	select	data	name	mail
username		email			

ThinkPHP

```
$User = M("User"); //      User
$User->where('type=1 AND status=1')->select();
```

SQL

```
SELECT * FROM think_user WHERE type=1 AND status=1
```

```
$User = M("User"); //      User
$condition['name'] = 'thinkphp';
$condition['status'] = 1;
//
$User->where($condition)->select();
```

SQL

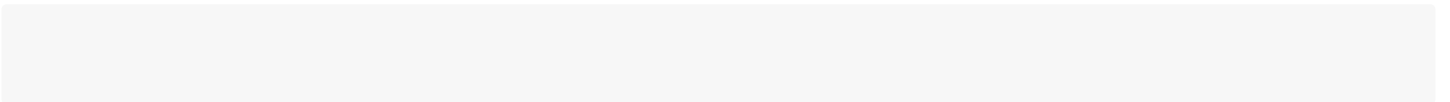
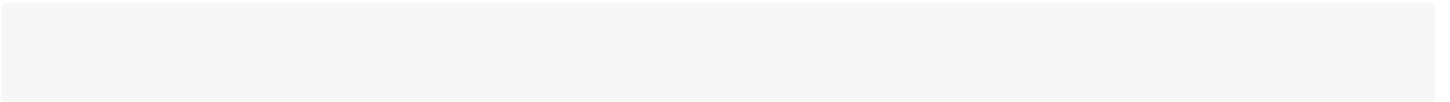
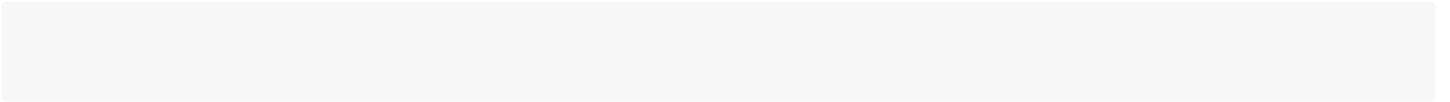
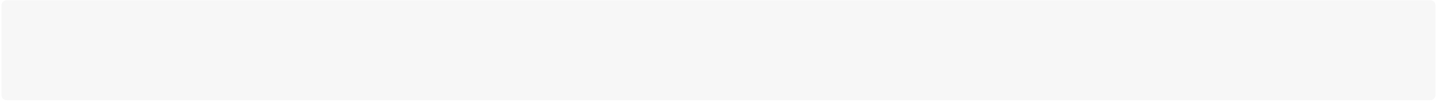
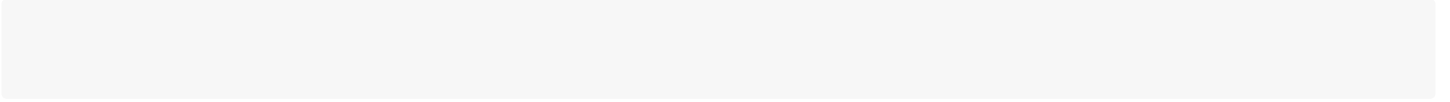
```
SELECT * FROM think_user WHERE `name`='thinkphp' AND status=1
```

AND

_logic

\$p

```
$map['id'] = 100;
```



`id <= 100`

[NOT] LIKE sql LIKE

```
$map['name'] = array('like','thinkphp%');
```

`name like 'thinkphp%'``DB_LIKE_FIELDS`

```
'DB_LIKE_FIELDS'=>'title|content'
```

```
$map['title'] = 'thinkphp';
```

`title like '%thinkphp%'`

```
$map['a'] = array('like',array('% thinkphp%', '% tp'),'OR');
$map['b'] = array('notlike',array('% thinkphp%', '% tp'),'AND');
```

```
(a like '%thinkphp%' OR a like '% tp') AND (b not like '%thinkphp%' AND b not like '% tp')
```

[NOT] BETWEEN sql [not] between

```
$map['id'] = array('between','1,8');
```

```
$map['id'] = array('between',array('1','8'));
```

`id BETWEEN 1 AND 8`

[NOT] IN sql [not] in

```
$map['id'] = array('not in','1,5,8');
```

```
$map['id'] = array('not in',array('1','5','8'));
```

```
id NOT IN (1,5, 8)
```

EXP

```
$map['id'] = array('in','1,3,8');
```

```
$map['id'] = array('exp',' IN (1,3,8)');
```

exp

SQL

```
$User = M("User"); //      User
//
$data['name'] = 'ThinkPHP';
$data['score'] = array('exp','score+1');//      1
>User->where('id=5')->save($data); //
```

OR & AND

```
$User = M("User"); //      User
$map['name|title'] = 'thinkphp';
//
>User->where($map)->select();
```

```
$User = M("User"); //      User
$map['name'] = 'thinkphp';
$map['title'] = 'thinkphp';
$map['_logic'] = 'OR';
//
$User->where($map)->select();
```

name= 'thinkphp' OR title = 'thinkphp'

```
$User = M("User"); //      User
$map['status& title'] = array('1','thinkphp','_multi'=>true);
//
$User->where($map)->select();
```

```
$User = M("User"); //      User
$map['status'] = 1;
$map['title'] = 'thinkphp';
//
$User->where($map)->select();
```

'_multi'=>true

status= 1 AND title = 'thinkphp'

```
$map['status& score& title'] = array('1',array('gt','0'),'thinkphp','_multi'=>true);
```

```
$map['status'] = 1;
$map['score'] = array('gt',0);
$map['title'] = 'thinkphp';
```

status= 1 AND score > 0 AND title = 'thinkphp'

" |" " & "


```
( `id` != 1 ) AND ( `name` = 'ok' ) AND ( status=1 AND score>10 )
```

URL

```
$map['id'] = array('gt','100');  
$map['_query'] = 'status=1&score=100&_logic=or';
```

```
`id`>100 AND ( `status` = '1' OR `score` = '100' )
```

```
$where['name'] = array('like', '%thinkphp%');  
$where['title'] = array('like', '%thinkphp%');  
$where['_logic'] = 'or';  
$map['_complex'] = $where;  
$map['id'] = array('gt',1);
```

```
( id > 1 ) AND ( ( name like '%thinkphp%' ) OR ( title like '%thinkphp%' ) )
```

_complex

```
$where['id'] = array('gt',1);  
$where['_string'] = ' ( name like "%thinkphp%" ) OR ( title like "%thinkphp%" ) ';
```

SQL

ThinkPHP

Count	
Max	
Min	
Avg	
Sum	

```
$User = M("User"); // User
```

```
$userCount = $User->count();
```

```
$userCount = $User->count("id");
```

```
$maxScore = $User->max('score');
```

0

```
$minScore = $User->where('score>0')->min('score');
```

```
$avgScore = $User->avg('score');
```

```
$sumScore = $User->sum('score');
```

SQL

ThinkPHP

ORM ActiveRecord

ThinkPHP
SQL

SQL

Db

QUERY

query

SQL

false

select

```
$Model = new \Think\Model() //          model
$Model->query("select * from think_user where status=1");
```

query

SQL

query

query

```
$Model = new \Think\Model() //          model
$Model->query("select * from __PREFIX__user where status=1");
// 3.2.2
$Model->query("select * from __USER__ where status=1");
```

EXECUTE

execute

sql

false

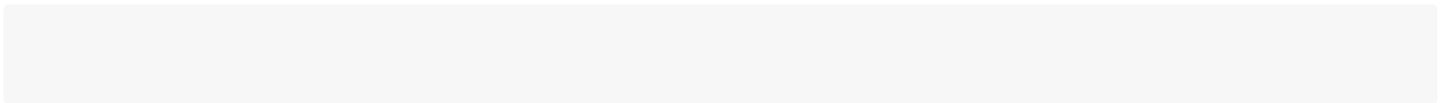
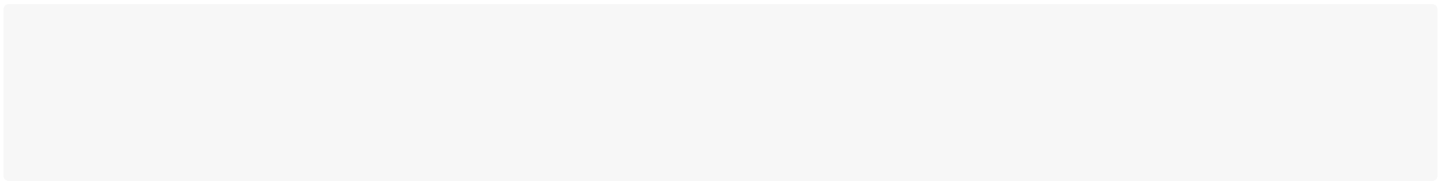
```
$Model = new \Think\Model() //          model
$Model->execute("update think_user set name='thinkPHP' where status=1");
```

execute

execute

SQL

e^c execute a N u
N(1) eTe u H p e



3.0

1 select select false SQL

```
//            SQL
$subQuery = $model->field('id,name')->table('tablename')->group('field')->where($where)->order('s
tatus')->select(false);
```

select false SQL

2 buildSql

```
$subQuery = $model->field('id,name')->table('tablename')->group('field')->where($where)->order('s
tatus')->buildSql();
```

buildSql SQL

SQL

```
//
$model->table($subQuery.' a')->where()->order()->select()
```

SQL ThinkPHP table where

ThinkPHP create

1. \$_validate
2. validate

in	
notin	3.1.2
length	3,12 3 12
between	array(1,31) 1,31
notbetween	3.1.2
expire	2012-1-15,2013-1-15 2012-1-15 2013-1-15
ip_allow	IP IP 201.12.2.5,201.12.2.6
ip_deny	IP ip 201.12.2.5,201.12.2.6
unique	unique

- self::MODEL_INSERT 1
- self::MODEL_UPDATE 2
- self::MODEL_BOTH 3

`$validate`

```

namespace Home\Model;
use Think\Model;
class UserModel extends Model{
    protected $_validate = array(
        array('verify','require',          '), //
        array('name','',                    ',0','unique',1), //           name
        array('value',array(1,2,3),        ',2','in'), //
        array('repassword','password',     ',0','confirm'), //
        array('password','checkPwd',       ',0','function'), //
    );
}

```

create

```

$User = D("User"); //           User
if (!$User->create()){
    //
    exit($User->getError());
}else{
    //
}

```

getError

D

create

POST

post

```

$User = D("User"); //           User
$data = getData(); //           getData
if (!$User->create($data)){
    // data
    exit($User->getError());
}else{
    //
}

```

create


```

$user = D("User"); //      User
if (!$user->create($_POST,1)){ //
    //
    exit($user->getError());
}else{
    //
}

```

create

create

4

```

namespace Home\Model;
use Think\Model;
class UserModel extends Model{
    protected $_validate = array(
        array('verify','require','      '), //
        array('name','checkName','      ',1,'function',4), //
        array('password','checkPwd','      ',1,'function',4), //
    );
}

```

```

$user = D("User"); //      User
if (!$user->create($_POST,4)){ //
    //
    exit($user->getError());
}else{
    //
}

```

```

$rules = array(
    array('verify','require',''), //
    array('name','', '0','unique',1), // name
    array('value',array(1,2,3),'', '2','in'), //
    array('repassword','password','', '0','confirm'), //
    array('password','checkPwd','', '0','function'), //
);

$user = M("User"); // User
if (!$user->validate($rules)->create()){
    //
    exit($user->getError());
}else{
    //
}

```

M

```

protected $_validate = array(
    array('verify','require','{%VERIFY_CODE_MUST}'),
    array('name','','{%ACCOUNT_EXISTS}',0,'unique',1),
);

```

VERIFY_CODE_MUST ACCOUNT_EXISTS

L

```

$rules = array(
    array('verify','require',L('VERIFY_CODE_MUST')),
    array('name','',L('ACCOUNT_EXISTS'),0,'unique',1),
);

```

patchValidate true false

```
protected $patchValidate = true;
```

getError()

```
array(" 1" => " 1", " 2" => " 2" ... )
```

json

```
$User = D("User"); //      User
if (!$User->create()){
    //
    $this->ajaxReturn($User->getError());
}else{
    //
}
```

ThinkPHP

create

ThinkPHP

create

add

save

1. `$_auto`
2. `auto`

```
array(
    array( 1, '[', ','],
    array( 2, '[', ','],
    .....
);
```

self::MODEL_INSERT	1
self::MODEL_UPDATE	2
self::MODEL_BOTH	3

function	
callback	
field	
string	
ignore	3.1.2

`_auto`

```

namespace Home\Model;
use Think\Model;
class UserModel extends Model{
    protected $_auto = array (
        array('status','1'), //          status          1
        array('password','md5',3,'function'), // password          md5
        array('name','getName',3,'callback'), // name              getName
        array('update_time','time',2,'function'), // update_time
    );
}

```

create

```

$user = D("User"); //          User
if (!$user->create()){ //
    //
    exit($user->getError());
}else{
    //
    $user->add();
}

```

create

```
$User = D("User"); //      User
$User->create(); //
$User->add(); //
```

```
$User = D("User"); //      User
$User->create(); //
$User->add(); //
```

create

post

```
$User = D("User"); //      User
$userData = getUserData(); //
$User->create($userData,2); //      userData
$User->add();
```

create

create

create

3.1.2

ignore

```
array('password','',2,'ignore')
```

password

```
$rules = array (
    array('status','1'), //      status      1
    array('password','md5',3,'function') , //      password      md5
    array('update_time','time',2,'function'), //      update_time
);
$User = M('User');
$User->auto($rules)->create();
$User->add();
```

create

```
$User = D("User"); //      User
$User->create(); //
$User->status = 2; //      status
$User->register_time = NOW_TIME; //      register_time
$User->add(); //
```

add

save

save

sa





Model

```
namespace Home\Model;
Class UserModel extends \Think\Model {
    Protected $autoCheckFields = false;
}
```

autoCheckFields

false

ThinkPHP

Model

```
namespace Home\Model;
Class UserModel {
}
```

CURD

Db

ThinkPHP

Model

\Think\Model

Model

Home

Model

Logic

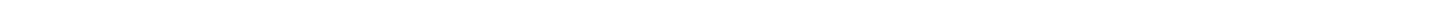
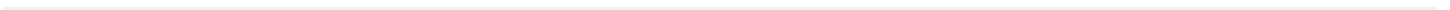
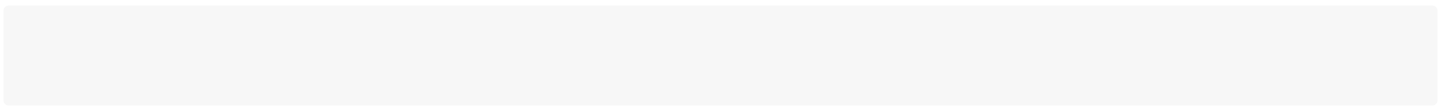
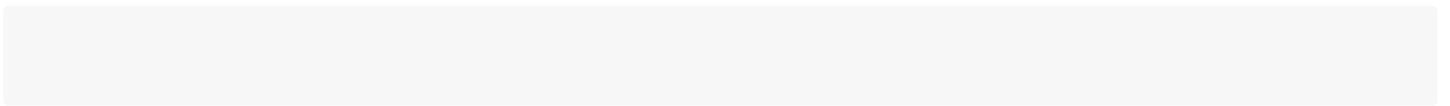
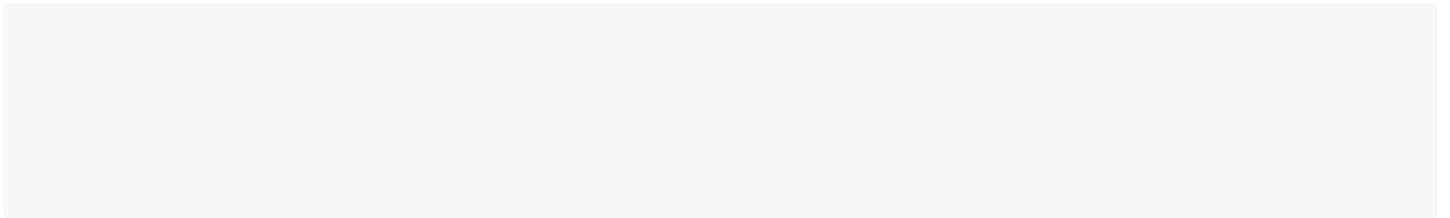
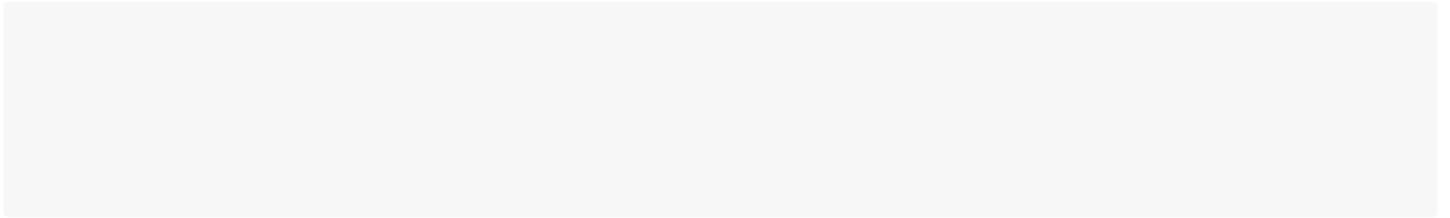
Service

- Home\Model\UserModel
- Home\Logic\UserLogic
- Home\Service\UserService

Model

Home\Model\UserModel.class.php

nameamm m



ThinkPHP
BELONGS_TO

HAS_ONE

Think\Model\ViewModel

viewFields

		BlogView		Blog	id	name	title	User
name	Category	title		BlogView				User
	Blog							

```
namespace Home\Model;
use Think\Model\ViewModel;
class BlogViewModel extends ViewModel {
    public $viewFields = array(
        'Blog' => array('id','name','title'),
        'Category' => array('title' => 'category_name', '_on' => 'Blog.category_id=Category.id'),
        'User' => array('name' => 'username', '_on' => 'Blog.user_id=User.id'),
    );
}
```

\$viewFields

```
'Blog' => array('id','name','title');
```

BlogView		Blog	id	name	title
			Blog		Blog

```
'_table' => "test_user"
//
// '_table' => "__USER__"
```

```
'_as' => 'myBlog'
```

BlogView Blog Category User

```
'Category' => array('title' => 'category_name');
```

```

                BlogView                Category                title
title

```

```
'title' => 'category_name'
```

```

Category    title    category_name
    _on

```

```
'_on' => 'Blog.category_id=Category.id'
```

```
User
```

```
Blog.categoryId=Category.id AND Blog.userId=User.id
```

```
SQL
```

```
SQL
```

```

Select
Blog.id as id,
Blog.name as name,
Blog.title as title,
Category.title as category_name,
User.name as username
from think_blog Blog JOIN think_category Category JOIN think_user User
where Blog.category_id=Category.id AND Blog.user_id=User.id

```

```
Blog
```

```
Blog
```

```
Blog
```

```
Think_
```

```
think_blog
```

```
ThinkPHP
```

```
JOIN
```

```

public $viewFields = array(
    'Blog' => array('id','name','title','_type' => 'LEFT'),
    'Category' => array('title' => 'category_name','_on' => 'Category.id=Blog.category_id','_type' => 'RIGHT'),
    'User' => array('name' => 'username','_on' => 'User.id=Blog.user_id'),
);

```

```
_type
```

```
Blog
```

```
'_type'=>'LEFT'
```

Category

SQL

```
Select
Blog.id as id,
Blog.name as name,
Blog.title as title,
Category.title as category_name,
User.name as username
from think_blog Blog LEFT JOIN think_category Category ON Blog.category_id=Category.id RIGHT JOIN think_user User ON Blog.user_id=User.id
```

```
'Category'=>array('title'=>'category_name','COUNT(Blog.id)'=>'count','_on'=>'Category.id=Blog.category_id'),
```

```
$Model = D("BlogView");
$Model->field('id,name,title,category_name,username')->where('id > 10')->order('id desc')->select();
```

group

```
$Model->field('id,name,title,category_name,username')->order('id desc')->group('id')->select();
```

```
$Model = D("Blog");
$Model->table('think_blog Blog,think_category Category,think_user User')
->field('Blog.id,Blog.name,Blog.title,Category.title as category_name,User.name as username')
->order('Blog.id desc')
->where('Blog.category_id=Category.id AND Blog.user_id=User.id')
->select();
```

JOIN

```
ONE_TO_ONE    HAS_ONE    BELONGS_TO
ONE_TO_MANY   HAS_MANY   BELONGS_TO
MANY_TO_MANY
```

-

-

-

-

-

-

-

HAS_ONE
BELONGS_TO

HAS_MANY
MANY_TO_MANY

ThinkPHP

CURD

HAS_ONE BELONGS_TO HAS_MANY MANY_TO_MANY

\$_link

Think\Model\RelationModel

```

namespace Home\Model;
use Think\Model\RelationModel;
class UserModel extends RelationModel{
    protected $_link = array(
        ' 1' => array(
            ' 1' => ' ',
            ' N' => ' ',
        ),
        ' 2' => array(
            ' 1' => ' ',
            ' N' => ' ',
        ),
        ' 3' => HAS_ONE, //
        ...
    );
}

```

HAS_ONE

HAS_ONE

UserModel

```

namespace Home\Model;
use Think\Model\RelationModel;
class UserModel extends RelationModel{
    protected $_link = array(
        'Profile' => self::HAS_ONE,
    );
}

```

```

namespace Home\Model;
use Think\Model\RelationModel;
class UserModel extends RelationModel{
    protected $_link = array(
        'Profile' => array(
            'mapping_type' => self::HAS_ONE,
            'class_name' => 'Profile',
            //
            .....
        ),
    );
}

```

HAS_ONE

mapping_type :

HAS_ONE

HAS_ONE

class_name

class_name

Profile

Profile

Profile

mapping_name

class_name

mapping_name

class_name

mapping_name

mapping_name

foreign_key

_id

UserModel

think_user

think

think_user

user_id

foreign_key

condition

condition

mapping_fields

mapping_fields

as_fields

ONE_TO_ONE

BELONGS_TO

Belongs_to

```
'Dept' => self::BELONGS_TO
```

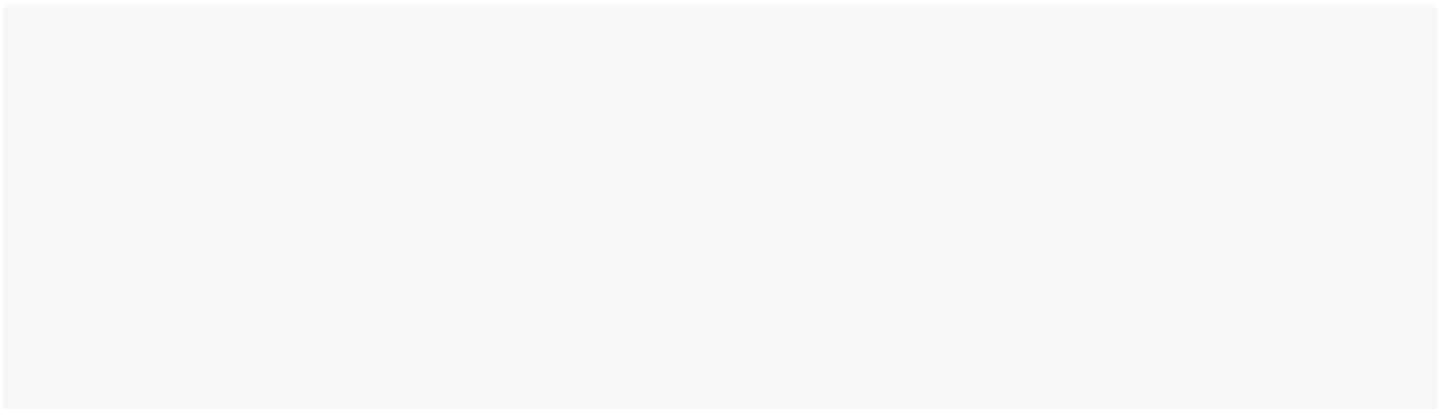
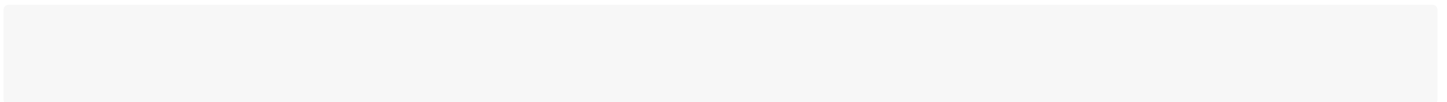
```
'Dept' => array(
    'mapping_type' => self::BELONGS_TO, // Ifs_name' n 'dept':B
    'class_name' => 'Dept',
    'foreign_key' => 'userId',
    'mapping_name' => 'dept',
    //
    './...',
),
```

BELONGS_TO

class_name	
mapping_name	
foreign_key	
mapping_fields	
condition	
parent_key	parent_id
as_fields	

HAS_MANY

HAS_MA

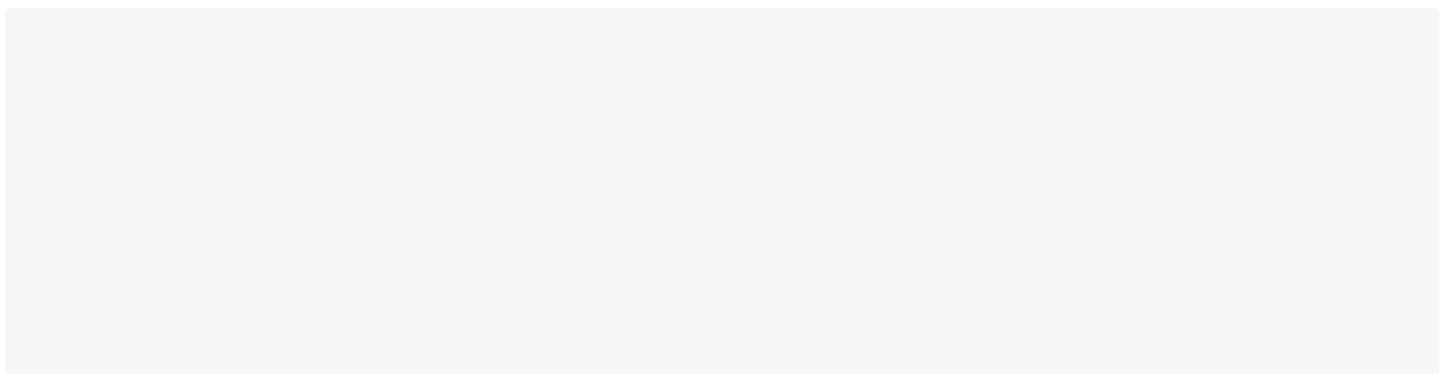
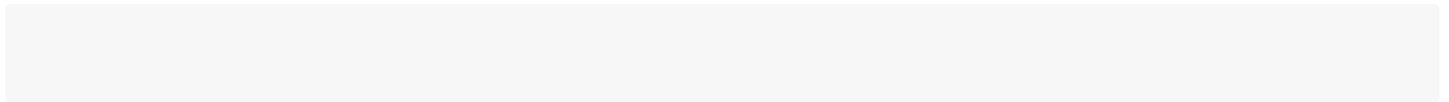


class_name	
mapping_name	
foreign_key	
parent_key	parent_id
condition	condition
mapping_fields	mapping_fields
mapping_limit	
mapping_order	

_id UserModel think_user think
 think_user user_id
 foreign_key

MANY_TO_MANY

MANY_TO_MANY



mapping_name	
foreign_key	_id
relation_foreign_key	_id
mapping_limit	
mapping_order	
relation_table	

```

think_user  think_group  group
            think_group_user  user  think_user_group
                    Model
            id  user_id  group_id
            getRelationTableName  User  Group
                    user_group  relation_table

```

3.2.2 relation_table

```
'relation_table'=>'__USER_GROUP__'
```

relation

relation

```
$User = D("User");
$user = $User->relation(true)->find(1);
```

\$user

```
array(
  'id' => 1,
  'account' => 'ThinkPHP',
  'password' => '123456',
  'Profile' => array(
    'email' => 'liu21st@gmail.com',
    'nickname' => ' ',
  ),
)
```



```

$user = D("User");
$user = $user->find(1);
//
$profile = $user->relationGet("Profile");

```

User

```

$user = D("User");
$data = array();
$data["account"] = "ThinkPHP";
$data["password"] = "123456";
$data["Profile"] = array(
    'email' => 'liu21st@gmail.com',
    'nickname' => ' ',
);
$result = $user->relation(true)->add($data);

```

Profile

```

$result = $user->relation("Profile")->add($data);

```

MANY_TO_MANY

```

$user = D("User");
$data["account"] = "ThinkPHP";
$data["password"] = "123456";
$data["Profile"] = array(
    'email' => 'liu21st@gmail.com',
    'nickname' => ' ',
);
$result = $user->relation(true)->where(array('id' => 3))->save($data);

```

Relation(true)

User

```
$result = $User->relation("Profile")->save($data);
```

Profile

HAS_ONE

HAS_MANY

MANY_TO_MANY

```
//      ID  3
$result = $User->relation(true)->delete("3");
//
$result = $User->relation("Profile")->delete("3");
```

Think\Model\AdvModel

```
namespace Home\Model;
use Think\Model\AdvModel;
class UserModel extends AdvModel{
}
```

UserModel

Think\Model\AdvModel

Create

create

Model

`$_filter`

```
protected $_filter = array(
    ' ' => array(' ' , ' '),
)
```

255

```
protected $_filter = array(
    'content' => array('contentWriteFilter', 'contentReadFilter'),
)
```

contentWriteFilter

contentReadFilter

serializeField

```
protected $serializeField = array(
    'info' => array('name', 'email', 'address'),
);
```

Info

name email address

User

name email address

info

```
$User = D("User"); //      User
//
$User->name = 'ThinkPHP';
$User->email = 'ThinkPHP@ gmail.com';
$User->address = '          ';
//          name email address          info
$User->add();

$User->find(8);
//          info          name email address
//
echo $User->name;
echo $User->email;
echo $User->address;
```

ThinkPHP

Text Blob

blobFields

Blog

content

```
Protected $blobFields = array('content');
```

readonlyField

```
protected $readonlyField = array('name', 'email');
```

name email

save

```
$User = D("User"); // User
$User->find(8);
//
$User->name = 'TOPThink';
$User->email = 'Topthink@gmail.com';
$User->address = ' ';
//
$User->save();
```

name email

address

name

email

ThinkPHP

"

Pessimistic Locking

"

"

Optimistic Locking

"

Pessimistic Locking

for update

ThinkPHP

```
$User->lock(true)->save($data);//
```

Optimistic Locking

Version

" version"

ThinkPHP

optimLock

optimLock

lock_version

User

User

lock_version

partition

```
protected $partition = array(  
    'field' => 'name',//  
    'type' => 'md5',//          id year mod md5  
    'expr' => 'name',//  
    'num' => 'name',//  
);
```

CURD

getTableName

getPartitionTableName

k

- 3.
4. SQL
- 5.
- 6.
7. MongoCode

Mongo

Mongo

_id

pk

```
namespace Home\Model;
use Think\Model\MongoModel;
class UserModel extends MongoModel {
    protected $pk = 'id';
}
```

_idType

self::TYPE_OBJECT 1	Mongoid
self::TYPE_INT 2	_autoInc
self::TYPE_STRING 3	hash

```
namespace Home\Model;
use Think\Model\MongoModel;
class UserModel extends MongoModel {
    protected $_idType = self::TYPE_INT;
    protected $_autoinc = true;
}
```

MongoModel

Mongo

Mongo

autoCheckFields true

MongoModel

group union join having lock distinct

```
$Model = new Think\Model\MongoModel("User");
$Model->field("name,email,age")->order("status desc")->limit("10,8")->select();
```

Mongo

- 1.
2. MongoDb
3. count MongoCode

MongoModel

```
_string MongoCode
_query
_complex MongoDB
```

MongoModel MongoCode MongoCode

```
$map[''] = array('','');
```

MongoDb MongoModel

Mongo

		Mongo
neq ne		\$ne
lt		\$lt
lte elt		\$lte
gt		\$gt
gte egt		\$gte

		Mongo
like	MongoRegex	
mod		\$mod
in	in	\$in
nin not in not	in	\$nin
all		\$all
between		
not between		
exists		\$exists
size		\$size
type		\$type
regex	MongoRegex	MongoRegex
exp	MongoCode	

like

mysql

Mysql	Mongo
array('like','%thinkphp%');	array('like','thinkphp');
array('like','thinkphp%');	array('like','^ thinkphp');
array('like','%thinkphp');	array('like','thinkphp\$');

LIKE sql LIKE

```
$map['name'] = array('like','^ thinkphp');
```

```
name like 'thinkphp%
```

Mongo

		Mongo
inc		\$inc
set		\$set
unset		\$unset
push		\$push
pushall		\$pushall

		Mongo
addtaset		\$addtaset
pop		\$pop
pull		\$pull
pullall		\$pullall

```
$data['id'] = 5;  
$data['score'] = array('inc',2);  
$Model->save($data);
```

MongoModel

mongoCode MongoCode

getMongoNextId ([]) ID

Clear

ThinkPHP

```
/[ /] / +
```

View

```
.html
```

User add

```
./Application/Home/View/User/add.html
```

View

```
'DEFAULT_V_LAYER' => 'Template', //
```

```
./Application/Home/Template/User/add.html
```

```
.html
```

TMPL_TEMPLATE_SUFFIX

```
'TMPL_TEMPLATE_SUFFIX' => '.tpl'
```

User add

```
./Application/Home/View/User/add.tpl
```

TMPL_FILE_DEPR

```
'TMPL_FILE_DEPR' => '_'
```

```
define('TMPL_PATH','./Template/');
```

```
./Application/Home/View/User/add.html
```

```
./Template/Home/User/add.html
```

```
VIEW_PATH
```

```
'VIEW_PATH'=>'./Theme/'
```

```
Theme
```

```
View
```

```
./Application/Home/View/User/add.html
```

```
./Theme/User/add.html
```

```
TMPL_PATH
```

```
VIEW_PATH
```

```
VIEW_PATH
```

```
DEFAULT_THEME
```

```
//
'DEFAULT_THEME' => 'default'
```

```
View/User/add.html //
View/default/User/add.html //
```

```
//
$this->theme('blue')->display('add');
```

assign

```
$this->assign('name',$value);
//
$this->name = $value;
```

assign

display show

{\$name}

```
$array['name'] = 'thinkphp';
$array['email'] = 'liu21st@gmail.com';
$array['phone'] = '12335678';
$this->assign($array);
```

name email phone

PHP

```
<?php echo $name.'['.$email.'.'.$phone.'];?>
```

{\$name} [{\$email} {\$phone}]

display

```
display(['', ''], [''], [''], [''])
```


[@] [:] []	theme

```
//
$this->display();
```

```

        / / / .html
    / / / / .html

    TMPL_FILE_DEPR 'TMPL_FILE_DEPR'=>'_'
    / / _ .html
    / / / _ .html
```

display

View

```
//
$this->display('edit');
```

edit

```
$this->display('Member:read');
```

Member

read

```
$this->theme('blue')->display('User:edit');
```

blue

User

edit

```
'TMPL_LOAD_DEFAULTTHEME'=>true
```



html

edit

edit

Public

Public

menu

```
$this->display('Public:menu');
```

display

```
$this->display('read', 'utf-8', 'text/xml');
```

XML

display

```
$this->display('../Template/Public/menu.html');
```

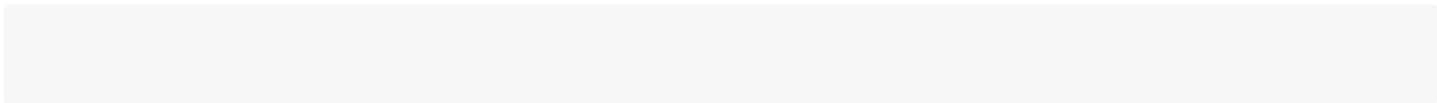
```
>/l | | | | mTemplate/Public
```

```
$this->display('../Template/Public/menu.tpl');
```

```
../Template/Public/menu.tpl
```

ty

ue



```

T('Public/menu');
//          /View/Public/menu.html
T('blue/Public/menu');
//          /View/blue/Public/menu.html
T('Public/menu','Tpl');
//          /Tpl/Public/menu.html
T('Public/menu');
//  Tmpl_FILE_DEPR      _      /Tpl/Public_menu.html
T('Public/menu');
//  Tmpl_TEMPLATE_SUFFIX .tpl      /Tpl/Public/menu.tpl
T('Admin@ Public/menu');
//  Admin/View/Public/menu.html
T('Extend://Admin@ Public/menu');
//  Extend/Admin/View/Public/menu.html (Extend      AUTOLOAD_NAMESPACE

```

```
display      T
```

```

//  T
$this->display(T('Admin@ Public/menu'));

```

```
T
```

```
fetch
```

```
fetch      display
```

```
fetch('      ')      display      fetch
```

```
$content = $this->fetch('Member:edit');
```

```
fetch
```

```
show
```

```
show
```

```
show('      '['      ']'      ')      $this->show($content);
```

```
$this->show($content, 'utf-8', 'text/xml');
```

show

PHP

XML

PHP

PHP

'TMPL_ENGINE_TYPE' => 'PHP'

TMPL_ENGINE_TYPE

ThinkPHP

XML
ThinkTemplate

ThinkTemplate
XML

ThinkPHP

- XML

```
$name = 'ThinkPHP';  
$this->assign('name',$name);  
$this->display();
```

Hello,{\$name}

Hello,<?php echo(\$name);?>

Hello,ThinkPHP

{ \$

Hello,{ \$name}

name

Hello,{ \$name}

{

}

TMPL_L_DELIM

TMPL_R_DELIM

```
'TMPL_L_DELIM' => '<{',  
'TMPL_R_DELIM' => '>}',
```

Hello,<{\$name}>

```
$data['name'] = 'ThinkPHP';  
$data['email'] = 'thinkphp@qq.com';  
$this->assign('data',$data);
```

```
Name {$data.name}
Email {$data.email}
```

```
Name {$data['name']}
Email {$data['email']}
```

```
data          name  email
```

```
Name {$data:name}
Email {$data:email}
```

```
Name {$data->name}
Email {$data->email}
```

```
{Think
```

```
{Think.server.script_name} // $_SERVER['SCRIPT_NAME']
{Think.session.user_id} // $_SESSION['user_id']
{Think.get.pageNumber} // $_GET['pageNumber']
{Think.cookie.name} // $_COOKIE['name']
```

```
$_SERVER
```

```
$_ENV
```

```
$_POST
```

```
$_GET
```

```
$_REQUEST
```

```
$_SESSION
```

```
$_COOKIE
```

```
{Think.const.MODULE_NAME}
```

```
{Think.MODULE_NAME}
```

```
{Think.config.db_charset}  
{Think.config.url_model}
```

```
{Think.lang.page_error}  
{Think.lang.var_error}
```

```
{data.name|md5}
```

```
<?php echo (md5($data['name'])); ?>
```

```
{create_time|date="y-m-d",###}
```

date

y-m-d

create_time

###

```
<?php echo (date("y-m-d",$create_time)); ?>
```



```
{ $data.name|substr=0,3}
```

```
<?php echo (substr($data['name'],0,3)); ?>
```

```
{ $data.name|substr=###,0,3}
```

" |"

```
{ $name|md5|strtoupper|substr=0,3}
```

```
<?php echo (substr(strtoupper(md5($name)),0,3)); ?>
```

```
{:substr(strtoupper(md5($name)),0,3)}
```

PHP

```
{ $user.nickname|default=" " }
```

```
{ $Think.get.name|default=" " }
```

```
{Think.getName|getName|default=" " }
```

```
" + " " " " " *" " /" " %" "
```

+	{a+b}
-	{a-b}
*	{a*b}
/	{a/b}
%	{a%b}
++	{a++} {++a}
--	{a--} {--a}
	{a+b*10+c}

```
{user.score+10} //
{user['score']+10} //
{user['score']*user['level']} //
{user['score']|myFun*10} //
{user['score']+myFun(user['level'])} //
```

Java Struts JSP
XML

Cx

taglib

```
< taglib name="html" />
```

html

```
< taglib name="html,article" />
```

article

read

```
< article:read name="hello" id="data" >  
{ $data.id } : { $data.title }  
< /article:read >
```

```
< article:read > ... < /article:read >
```

```
< article:read name="hello" />
```

XML

ThinkPHP

Cx

Cx

```
< eq name="status" value="1" >  
< /eq >
```

Cx

```
< cx:eq name="status" value="1" >  
< /cx:eq >
```

Cx

```
'TAGLIB_BUILD_IN' => 'cx,article'
```

```
<read name="hello" id="data" >  
{$data.id};{$data.title}  
</read>
```

```
'TAGLIB_PRE_LOAD' => 'article,html'
```

```
<taglib name="html,article" />
```

```
<article:read name="hello" id="data" >  
{$data.id};{$data.title}  
</article:read>
```

block

extend

block

```
<block></block>
```

```
<block name="title"><title>          </title></block>
```

block

name

block

```
<block name="title"><title>{$web_title}</title></block>
```

```
<block name="include"><include file="Public:header" /></block>
```

base.html

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<block name="title"><title>          </title></block>
</head>
<body>
<block name="menu">          </block>
<block name="left">          </block>
<block name="main">          </block>
<block name="right">          </block>
<block name="footer">          </block>
</body>
</html>
```

```

<extend name= "base" />
<block name= "title" >< title>{$title}</title></block>
<block name= "menu" >
<a href= "/" > </a>
<a href= "/info/" > </a>
<a href= "/bbs/" > </a>
</block>
<block name= "left" ></block>
<block name= "content" >
<volist name= "list" id= "vo" >
<a href= "/new/{$vo.id}" >{$vo.title}</a><br/>
{$vo.content}
</volist>
</block>
<block name= "right" >

<volist name= "news" id= "new" >
<a href= "/new/{$new.id}" >{$new.title}</a><br/>
</volist>
</block>
<block name= "footer" >
@ ThinkPHP2012
</block>

```

extend

extend

include

```

<extend name= "Public:base" />

```

```

<extend name= "../Template/Public/base.html" />

```

```

<block name= "title" >< title>{$title}</title></block>
<a href= "/" > </a>
<a href= "/info/" > </a>
<a href= "/bbs/" > </a>

```

XML

```
{ }
```

```
{ $name } { $vo.name }
```

```
{ $vo['name']|strtoupper }
```

```
TMPL_L_DELIM //
TMPL_R_DELIM //
```

```
'TMPL_L_DELIM' => '<{'
'TMPL_R_DELIM' => '}>'
```

```
{ $name } { $vo.name }
```

```
< { $name } >
```

```
< { $vo.name } >
```

XML

XML

XML

XML

XML

```
TAGLIB_BEGIN //
TAGLIB_END //
```

```
'TAGLIB_BEGIN'=>'[',  
'TAGLIB_END'=>']',
```

```
<eq name="name" value="value" >  
  
<else/>  
  
</eq>
```

```
[eq name="name" value="value"]  
  
[else/]  
  
[/eq]
```

XML

```
{ $status? ' ' ' ' }  
{ $info['status']? $info['msg']: $info['error'] }
```

include

```
<include file=' 1, 2,...' />
```


@ / /

```
<include file="Public/header" /> //      header  
<include file="Public/
```

include		file
import	js css load	file,href,type,value,basepath
volist		name,id,offset,length,key,mod
foreach		name,item ,key
for	For	name,from ,to,before,step
switch		name
case	switch	value,break
default	switch	
compare	eq neq lt gt egt elt heq nheq	name,value,type
range	in notin between notbetween	name,value,type
present		name
notpresent		name
empty		name
notempty		name
defined		name
notdefined		name
define		name,value
assign		name,value
if		condition
elseif	if	condition
else		
php	php	

Volist

volist

select

select

volist

```
$User = M('User');
$list = $User->limit(10)->select();
$this->assign('list',$list);
```

```
<volist name="list" id="vo" >
{$vo.id};{$vo.name}<br/>
</volist>
```

Volist

name

id

name

```
<volist name="list" id="data" >
{$data.id};{$data.name}<br/>
</volist>
```

5 15

```
<volist name="list" id="vo" offset="5" length='10'>
{$vo.name}
</volist>
```

```
<volist name="list" id="vo" mod="2" >
<eq name="mod" value="1">{$vo.name}</eq>
</volist>
```

Mod

```
<volist name="list" id="vo" mod="5" >
{$vo.name}
<eq name="mod" value="4"><br/></eq>
</volist>
```

```
<volist name="list" id="vo" empty="" >
    {$vo.id}|{$vo.name}
</volist>
```

empty

html

```
$this->assign('empty','<span class="empty"> </span>');
$this->assign('list',$list);
```

```
<volist name="list" id="vo" empty="$empty" >
    {$vo.id}|{$vo.name}
</volist>
```

```
<volist name="list" id="vo" key="k" >
    {$k}.{$vo.name}
</volist>
```

key

i

```
<volist name="list" id="vo" >
    {$i}.{$vo.name}
</volist>
```

key

key

```
<volist name="list" id="vo" >
    {$key}.{$vo.name}
</volist>
```

```
<volist name=":fun('arg')" id="vo">
    {$vo.name}
</volist>
```

Foreach

foreach volist

```
<foreach name="list" item="vo">
  {$vo.id};{$vo.name}
</foreach>
```

name item

```
<foreach name="list" item="vo" >
  {$key}{$vo}
</foreach>
```

```
<foreach name="list" item="vo" key="k" >
  {$k}{$vo}
</foreach>
```

For

```
<for start="            " end="            " comparison="" step="            " name="            " >
</for>
```

comparison

lt name i 1

```
<for start="1" end="100">
  {$i}
</for>
```

```
for ($i=1;$i<100;$i+=1){
  echo $i;
}
```

Switch

```
< switch name= "      " >
< case value= " 1" break= "0 1" >          1 </case>
< case value= " 2" >          2 </case>
< default />
</switch>
```

```
< switch name= "User.level" >
  < case value= "1" > value1 </case>
  < case value= "2" > value2 </case>
  < default /> default
</switch>
```

name

```
< switch name= "Think.get.userId|abs" >
  < case value= "1" > admin </case>
  < default /> default
</switch>
```

case value " |"

```
< switch name= "Think.get.type" >
  < case value= "gif|png|jpg" >          </case>
  < default />
</switch>
```

\$_GET["type"] gif png jpg

Case break break break break

```
< switch name= "Think.get.userId|abs" >
  < case value= "1" break= "0" > admin </case>
  < case value= "2" > admin </case>
  < default /> default
</switch>
```

case value

```
< switch name= "User.userId" >
  < case value= "$adminId" > admin < /case >
  < case value= "$memberId" > member < /case >
  < default /> default
< /switch >
```

if

```
<      name= "      " value= "      " >
< /      >
```

eq	equal
neq	notequal
gt	
egt	
lt	
elt	
heq	
nheq	

else

name value

```
< eq name= "name" value= "value" > value < /eq >
```

```
< equal name= "name" value= "value" > value < /equal >
```

```
else
```

```
<eq name="name" value="value">
```

```
<else/>
```

```
</eq>
```

```
name          5
```

```
<gt name="name" value="5">value</gt>
```

```
name          5
```

```
<egt name="name" value="5">value</egt>
```

```
vo
```

```
5
```

```
<eq name="vo.name" value="5">
```

```
{vo.name}
```

```
</eq>
```

```
vo          5
```

```
<eq name="vo:name" value="5">
```

```
{vo.name}
```

```
</eq>
```

```
$vo['name']  5
```

```
<eq name="vo['name']" value="5">
```

```
{vo.name}
```

```
</eq>
```

```
vo
```

```
5
```

```
<eq name="vo:name|strlen" value="5">{vo.name}</eq>
```

```
<eq name="Think.get.name" value="value"> <else/> </eq>
```


\$a

```
<eq name="vo:name" value="$a">{$vo.name}</eq>
```

compare

compare

name 5

```
<compare name="name" value="5" type="eq">value</compare>
```

```
<eq name="name" value="5" >value</eq>
```

type

in notin between notbetween

IN NOTIN

id 1

```
$id = 1;
$this->assign('id',$id);
```

in

```
<in name="id" value="1,2,3">
id
</in>
```

id

id

```
<in name="id" value="1,2,3">
  id
<else/>
  id
</in>
```

name

```
<in name="Think.get.id" value="1,2,3">
  $_GET['id']
</in>
```

value

```
<in name="id" value="$range">
  id
</in>
```

\$range

value

```
<in name="id" value="$Think.post.ids">
  id
</in>
```

BETWEEN NOTBETWEEN

between

```
<between name="id" value="1,10">
  1
</between>
```

notbetween

```
<notbetween name="id" value="1,10">
  2
</notbetween>
```

else

```
<between name="id" value="1,10">
  1
<else/>
  2
</between>
```

between

value

```
<between name="id" value="1,3,10">
  1
</between>
```

1~3

1~10

```
<between name="id" value="A,Z">
  1
</between>
```

name

```
<between name="Think.post.id" value="1,5">
  1
</between>
```

value

```
<between name="id" value="$range">
  1
</between>
```

```
<between name="id" value="$Think.get.range">
  1
</between>
```

RANGE

range

```
<range name="id" value="1,2,3-4 |,3-4 |,3range" |,3-4 d" |,3~ |
```

type

in/not in/between/not between

IN

BETWEEN

IF

```
< if condition="($name eq 1) OR ($name gt 100) "> value1
< elseif condition="$name eq 2"/> value2
< else /> value3
</if>
```

condition

eq

" >" " <"

```
< if condition="$id < 5 ">value1
  < else /> value2
</if>
```

```
< if condition="$id lt 5 ">value1
< else /> value2
</if>
```

condition

php

```
< if condition="strtoupper($user['name']) neq 'THINKPHP'">ThinkPHP
< else /> other Framework
</if>
```

condition

user

```
< if condition="$user.name neq 'ThinkPHP'">ThinkPHP
< else /> other Framework
</if>
```

user

```
< if condition="$user:name neq 'ThinkPHP'">ThinkPHP
< else /> other Framework
</if>
```

if condition

php

Switch

switch

if

switch

IF

php

PHP

Present

present

```
<present name="name" >
name
</present>
```

```
<notpresent name="name" >
name
</notpresent>
```

```
<present name="name" >
name
<else />
name
</present>
```

present name

```
<present name="Think.get.name" >
$_GET['name']
</present>
```

Empty

empty

```
<empty name="name">  
name  
</empty>
```

```
<notempty name="name">  
name  
</notempty>
```

```
<empty name="name">  
name  
<else />  
name  
</empty>
```

name

```
<empty name="Think.get.name">  
$_GET['name']  
</empty>
```

Defined

DEFINED

```
<defined name="NAME">  
NAME  
</defined>
```

name

```
<notdefined name="NAME">  
NAME  
</notdefined>
```

```
<defined name="NAME">
NAME
<else />
NAME
</defined>
```

Assign

ASSIGN

```
<assign name="var" value="123" />
```

var 123

name

```
<assign name="Think.get.id" value="123" />
```

\$_GET['id'] 123

value

```
<assign name="var" value="$val" />
```

var

```
<assign name="var" value="$Think.get.name" />
```

\$var = \$_GET['name'];

Define

DEFINE

```
<define name="MY_DEFINE_NAME" value="3" />
```

MY_DEFINE_NAME

value

```
<define name="MY_DEFINE_NAME" value="$name" />
```

```
<define name="MY_DEFINE_NAME" value="$Think.get.name" />
```

```
volist itn < | i < t | o < v < Y < a < v
```



```
<import file="Js.Util.Array" />
```

```
<import file="Js.Util.Array,Js.Util.Date" />
```

CSS type

```
<import type='css' file="Css.common" />
```

import Public basepath

```
<import file="Js.Util.Array" basepath="./Common" />
```

load JS CSS

```
<load href="/Public/Js/Common.js" />
<load href="/Public/Css/common.css" />
```

href

```
<load href="__PUBLIC__/Js/Common.js" />
```

Load type

js css load

```
<js href="/Public/Js/Common.js" />
<css href="/Public/Css/common.css" />
```

PHP

Php

PHP

php

```
<php>echo 'Hello,world!';</php>
```

PHP

php

PHP

php

```
<?php echo 'Hello,world!'; ?>
```

php

php

XML

```
<php><eq name='name'value='value'>value</eq></php>
```

Php

eq

```
<php>if( {$user} != 'ThinkPHP' ) echo 'ThinkPHP' ;</php>
```

Php

{\$user}

```
<php>if( $user.name != 'ThinkPHP' ) echo 'ThinkPHP' ;</php>
```

Php

\$user.name

PHP

PHP

TMPL_DENY_PHP

true

PHP

PHP

literal

```
<literal>
  <if condition="$name eq 1 "> value1
  <elseif condition="$name eq 2"/>value2
  <else /> value3
</if>
</literal>
```

if literal

if

php

{ \$user }

XML

literal

```
<php>echo '{$Think.config.CUSTOM.'.$key.'}'</php>
```

php

{ \$Think

literal

```
<php><literal>echo '{$Think.config.CUSTOM.'.$key.'}'</literal></php>
```

Literal

JS

JS

literal

```
{/* */} {/ }
```

```
{/ }
```

```
{
```

```
{/* */}
```

Html

ThinkPHP

LAYOUT_ON

LAYOUT_NAME layout

```
'LAYOUT_ON' => true,
'LAYOUT_NAME' => 'layout',
```

LAYOUT_ON

```
namespace Home\Controller;
use Think\Controller;
class UserController extends Controller{
    public function add() {
        $this->display('add');
    }
}
```

LAYOUT_ON

Application/Home/View/User/add.html

Application/Home/View/layout.html

{__CONTENT__}

layout.html

```
<include file="Public:header" />
{__CONTENT__}
<include file="Public:footer" />
```

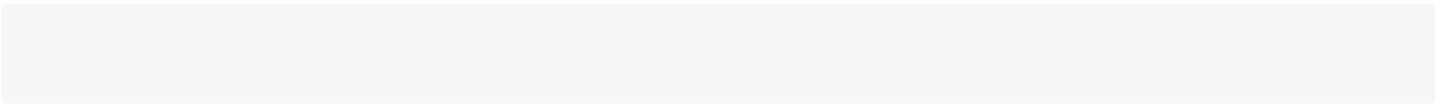
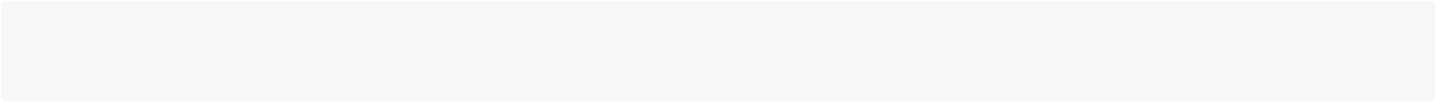
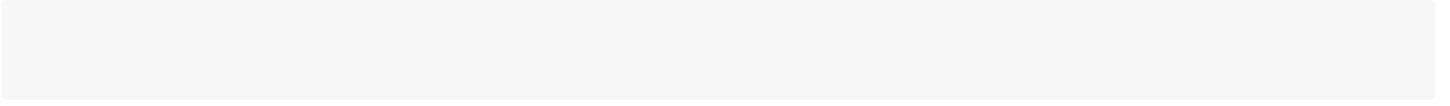
layout

User/add.html

layout

{CONTENT}

```
'TMPL_LAYOUT_ITEM' => '{__REPLACE__}'
```



```
<include file="Public:header" />
<div id="main" class="main" >
  {__CONTENT__}
</div>
<include file="Public:footer" />
```

header footer

layout

header

```
<layout name="menu" />
```

menu

layout

layout

LAYOUT_ON

```
namespace Home\Controller;
use Think\Controller;
Class UserController extends Controller{
    Public function add() {
        layout(true);
        $this->display('add');
    }
}
```

layout

```
namespace Home\Controller;
use Think\Controller;
Class UserController extends Controller{
    Public function add() {
        layout('Layout/newlayout');
        $this->display('add');
    }
}
```

layout

```

namespace Home\Controller;
use Think\Controller;
class UserController extends Controller{
    public function add() {
        layout(false); //
        $this->display('add');
    }
}

```

```

__ROOT__
__APP__          URL
__MODULE__      URL
__CONTROLLER__  __URL__          URL
__ACTION__      URL
__SELF__        URL
__PUBLIC__      /Public/

```

TMPL_PARSE_STRING

```

'TMPL_PARSE_STRING' => array(
    '__PUBLIC__' => '/Common', // /Public
    '__JS__'     => '/Public/JS/', // JS
    '__UPLOAD__' => '/Uploads', //
)

```

__PUBLIC__

__PUBLIC__

```
'TMPL_PARSE_STRING' => array(  
    '--PUBLIC--' => '__PUBLIC__', //    `__PUBLIC__`  
)
```

`__PUBLIC__`

--PUBLIC--

ThinkPHP

ThinkPHP

```
<?php
//
define('APP_DEBUG', true);
//
define('APP_PATH', './Application/');
//
require './ThinkPHP/ThinkPHP.php';
```

```
<?php
//
define('APP_DEBUG', false);
//
define('APP_PATH', './Application/');
//
require './ThinkPHP/ThinkPHP.php';
```

-
-
- SQL SQL
-
- Windows Linux
- Trace

Trace

debug.php

debug.php

IO

```
'SHOW_ERROR_MSG' => true, //
```

PHP

ThinkPHP



无法加载模块: test

错误位置

FILE: E:\www\test\ThinkPHP\Library\Think\Dispatcher.class.php LINE: 171

TRACE

#0: E:\www\test\ThinkPHP\Library\Think\Dispatcher.class.php



无法加载模块: test

ThinkPHP 3.2.0 (Fast & Simple OOP framework } -- [WE CAN DO IT]

```
'SHOW_ERROR_MSG' => true, //
```

```
E('');
```

```
0
```

```
E('',25);
```

```
throw
```

```
throw new \Think\Exception('');
```

```
Tpl/think_exception.tpl
```

```
TMPL_EXCEPTION_FILE
```

```
'TMPL_EXCEPTION_FILE' => APP_PATH.'/Public/exception.tpl'
```

```

$file]
[line]
$message]
$trace]      Trace

```

```
PHP
```

```
PHP
```

```
'SHOW_ERROR_MSG' => false,
'ERROR_MESSAGE' => ''
```

```
""
```

```
ERROR_PAGE
```

```
ERROR_PAGE
```

```
URL
```

```
'ERROR_PAGE' => '/Public/error.html'
```

```
'ERROR_PAGE' => 'http://www.myDomain.com/Public/error.html'
```

ERROR_PAGE

SQL

Think\Log

LOG_RECORD

```
'LOG_RECORD' => true, //  
'LOG_LEVEL' => 'EMERG,ALERT,CRIT,ERR', //      EMERG ALERT CRIT ERR
```

ThinkPHP

- EMERG
 - ALERT
 - CRIT
 - ERR
 - WARN
 - NOTICE
 - INFO
 - DEBUG
 - SQL SQL
-

LOG_TYPE

```
'LOG_TYPE' => 'File', //
```

File

Library/Think/Log/Driver/File.class.php

Log

3

Log::record()	
Log::save()	
Log::write()	

Log::save

Log::record

record

```
Think\Log::record('');
```

ERR

```
Think\Log::record('','WARN');
```

record

```
'LOG_LEVEL' => 'EMERG,ALERT,CRIT,ERR', // EMERG ALERT CRIT ERR
```

record

```
Think\Log::record('','WARN',true);
```

record

write

```
Think\Log::write('
```

Trace

ThinkPHP

Trace

ThinkPHP

SQL

Trace

Trace

Trace

Trace

```
//      Trace  
'SHOW_PAGE_TRACE' => true,
```

ThinkPHP LOGO

LOGO

Trace

Trace 6

SQL

Trace

SQL	SQL

Trace

```
'TRACE_PAGE_TABS' => array(
    'base' => ' ',
    'file' => ' ',
    'think' => ' ',
    'error' => ' ',
    'sql' => 'SQL',
    'debug' => ' '
)
```

```
'TRACE_PAGE_TABS' => array(
    'base' => ' ',
    'file' => ' ',
    'think' => ' ',
    'error' => ' ',
    'sql' => 'SQL',
    'debug' => ' ',
    'user' => ' '
)
```

```
'TRACE_PAGE_TABS' => array(
    'base' => ' ',
    'file' => ' ',
    'think' => ' ',
    'error|debug|sql' => ' ',
    'user' => ' '
)
```

trace

```
trace($user, ' ', 'user');
```

TRACE_PAGE_TABS

Trace

trace

PAGE_TRACE_SAVE

```
'PAGE_TRACE_SAVE' => true
```

trace

Trace

_trace.log

12-06-21_trace.log


```
'PAGE_TRACE_SAVE' => array('base','file','sql');
```

base file sql

Trace

Trace

trace

trace

AJAX

Trace

```
trace(' ',' ',' ','')
```

```
$info = ' ';
trace($info, '');
```

Trace

```
trace($info, '','user');
```

user

ERR

debug

trace

'TRACE_EXCEPTION' => true

```
trace($info, '','ERR');
```

trace

1. AJAX
2. SHOW_PAGE_TRACE false Trace
3. trace true

trace

```
'ERR' //      :
'WARN' //     :
'NOTIC' //    :
'INFO' //     :
'DEBUG' //    :
'SQL' // SQL  SQL
```



Trace

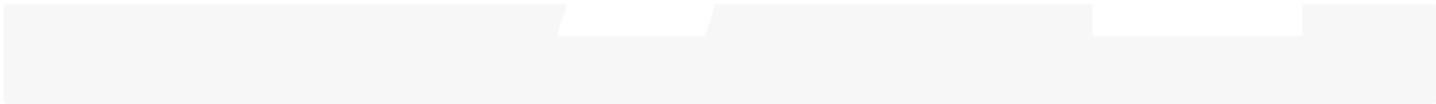
ThinkPHP

trace

```
$blog = D("Blog");
$vo = $blog->create();
trace($vo,'create vo');
$vo = $blog->find();
trace($vo,'find vo');
```

php var_dump print r

ThinkPHP




```
$Blog = D("Blog");
$blog = $Blog->find(3);
dump($blog);
```

```
array(12) {
  ["id"] => string(1) "3"
  ["name"] => string(0) ""
  ["user_id"] => string(1) "0"
  ["cate_id"] => string(1) "0"
  ["title"] => string(4) "test"
  ["content"] => string(4) "test"
  ["create_time"] => string(1) "0"
  ["update_time"] => string(1) "0"
  ["status"] => string(1) "0"
  ["read_count"] => string(1) "0"
  ["comment_count"] => string(1) "0"
  ["tags"] => string(0) ""
}
```

G

```
G('begin');
// ...
G('end');
// ...
//
echo G('begin','end').'s';
```

```
G('begin','end')      begin      end                begin
                    end                end                end
```

0.0056s

4

```
G('begin','end',6).'s';
```

0.005587s

i i i b i r

```
echo G('begin','end','m').'kb';
```

m

625kb

end

end

E

```
//
E($msg);
```

SQL

sql

SQL

getLastsql

```
$User = M("User"); //      User
>User->find(1);
>User->getLastSql();
// 3.2
>User->_sql();
```

```
SELECT * FROM think_user WHERE id = 1
```

SQL

getLastSql

SQL

```

$user = M("User"); //      User
$info = M("Info"); //     Info
$user->find(1);
$info->find(2);
echo M()->getLastSql();
echo $user->getLastSql();
echo $info->getLastSql();

```

```

SELECT * FROM think_info WHERE id = 2
SELECT * FROM think_user WHERE id = 1
SELECT * FROM think_info WHERE id = 2

```

getLastSql

sql

SQL

Trace

Mongo

SQL

SQL

getLastSql

SQL

```

$user = M("User"); //      User
$result = $user->find(1);
if(false === $result){
    echo $user->getDbError();
}

```

CURD

false

getDbError

ThinkPHP

APC Db Memcache Shmop Sqlite Redis

Eaccelerator Xcache

ThinkPHP

S

```
//
S(array('type' => 'xcache', 'expire' => 60));
```

expire	
prefix	
type	

Apachenote Apc Db Eaccelerator File Memcache Redis Shmop Sqlite Wincache Xca

S type DATA_CACHE_TYPE
 prefix DATA_CACHE_PREFIX expire
 DATA_CACHE_TIME

Memcache

```
S(array(  
    'type' => 'memcache',  
    'host' => '192.168.1.10',  
    'port' => '11211',  
    'prefix' => 'think',  
    'expire' => 60  
));
```

prefix

```
//  
S('name',$value);
```

```
//          300  
S('name',$value,300);
```

```
//          300  
S('name',$value,array('type' => 'file','expire' => 300));
```

```
//  
$value = S('name');
```

false

```
//  
S('name',null);
```

name


```
F('data',$Data,TEMP_PATH);
```

```
$Data = F('data');
```

```
F('data',NULL);
```

F

```
$Model->cache(true,60,'xcache')->select();
```

```

                xcache                60
            cache                                Cache
        key                                S

```

```
$value = S('cache_name');
```

```
select                find  getField
```

```
//                60
$Model->where($map)->cache('key',60)->find();
```

```
HTML_CACHE_ON
```

```
HTML_CACHE_RULES
```

```

'HTML_CACHE_ON' => true, //
'HTML_CACHE_TIME' => 60, //
'HTML_FILE_SUFFIX' => '.shtml', //
'HTML_CACHE_RULES' => array( //
    //      1
    '      ' => array('      ', '      ', '      '),
    //      2
    '      ' => '      ',
)

```

1

2

```
HTML_CACHE_TIME
```

HTML_PATH

GET

read

```
'read' => array('{id}',60)
```

{id}

\$_GET['id']

60

User

```
'user:' => array('User/{:action}_{id}','600')
```

{:action}

Blog

read

```
'blog:read' => array('{id}',0)
```

```
'*' => array('{$_SERVER.REQUEST_URI|md5}')
```

URL

1

_GET

_REQUEST

_SERVER

_SESSION

_COOKIE

```
{$_x × × |function}
```



```
'read' => array('Think{id},{name}','60', 'md5')
```

```
md5('Think'.$_GET['id']. ' '.$_GET['name']);
```

XSS

-
-
- |
- | stripslashes htmlentities
htmlspecialchars strip_tags

|

|

htmlspecialchars

|

DEFAULT_FILTER

```
'DEFAULT_FILTER' => 'strip_tags',
```

DEFAULT_FILTER

|

strip_tags

```
'DEFAULT_FILTER' => 'strip_tags,stripslashes',
```

|

```
l('post.id','0','intval'); // intval $_POST['id']
l('get.title','','strip_tags'); // strip_tags $_GET['title']
```

```
$_GET $_POST $_REQUEST
$_GET l('get.')
```

|

filter

```
$this->data($data)->filter('strip_tags')->add();
```

Think\Model

create

create

create

insertFields

updateFields

insertFields

updateFields

create

insertFields updateFields

```
namespace Home\Model;
class UserModel extends \Think\Model{
    protected $insertFields = array('account','password','nickname','email');
    protected $updateFields = array('nickname','email');
}
```

add

'account','password','nickname','email'

'nickname','email'

create

insertFields updateFields

```
D('User')->create();
```

field

insertFields updateFields

create

field

```
M('User')->field('account,password,nickname,email')->create();
```

```
M('User')->field('nickname,email')->create();
```

field

ThinkPHP

tags.php

```
return array(  
    //  
    'view_filter' => array('Behavior\TokenBuild'),  
    // 3.2.1  
    'view_filter' => array('Behavior\TokenBuildBehavior'),  
);
```

view_filter

```
'TOKEN_ON' => true ;
```


`{_TOKEN_}`

```
C('TOKEN_ON',false);
$this->display();
```

`autoCheckToken`

create
false

```
$User = M("User"); // User
//
if (!$User->autoCheckToken($_POST)){
//
}
```

SQL

WEB

SQL

```
$User = M("User"); // User
$user->find($_GET["id"]);
```

id

ThinkPHP

escape_string (real_escape_string,mysql_escape_string)

SQL

-
-
-
-

PDO

where

```
$Model->where("id=%d and username='%s' and xx='%f'",array($id,$username,$xx))->select();  
//  
$Model->where("id=%d and username='%s' and xx='%f'",$id,$username,$xx)->select();
```

query execute

```
$model->query('select * from user where id=%d and status=%d',$id,$status);  
//  
$model->query('select * from user where id=%d and status=%d',array($id,$status));
```

execute query

URL

index.html

default.html

```
define('DIR_SECURE_FILES',array('r' => 'r', 'w' => 'w', 'a' => 'a'));
```

```
// dirs
\Think\Build::buildDirSecure($dirs);
```

.htaccess

Apache

View

.htaccess

```
<Files *.html>
Order Allow,Deny
Deny from all
</Files>
```

html

*.html

WEB

Think\Upload

Think\Upload

XSS

XSS

Cookie

- JavaScript
- Html htmlentities htmlspecialchars
- XSS remove_xss
- URL XSS

-
-
-
-
-
-

URL

ThinkPHP

ThinkPHP/Library

Think
Org

Org/Util/

Image.class.php

```
namespace Org\Util;  
class Image {  
}
```

Image

```
$image = new \Org\Util\Image;
```

ThinkPHP/Library

Com

Com\Sina\App

Com/Sina/App.class.php

```
namespace Com\Sina;  
class App {  
}
```

Com\Sina\Rank

Com/Sina/Rank.class.php

```
namespace Com\Sina;  
class Rank {  
}
```

Library

```
'AUTOLOAD_NAMESPACE' => array(
    'Lib' => APP_PATH.'Lib',
)
```

Lib

Lib

Lib\Sina

Lib\Sina\App

Lib/Sina/App.class.php

```
namespace Lib\Sina;
class App {
}
```

Lib\Sina\Rank

Lib/Sina/Rank.class.php

```
namespace Lib\Sina;
class Rank {
}
```

import

Counter

Com/Sina/Util/Counter.class.php

```
class Counter {
}
```

```
import('Com.Sina.Util.Couter');
$object = new \Counter();
```

Home\Model\UserModel

Application\Home\Model

```
namespace Home\Model;
use Think\Model;
class UserModel extends Model{
}
```

```
namespace Common\Util;
class Pay {
}
```

```
namespace Admin\Api;
use Think\Model;
class UserApi extends Model{
}
```

ThinkPHP

3.2

Think\Log

Think\Log\Driver

Think\Db

Think\Db\Driver

Home\Driver\Cache\Sae.class.php

Cache

Think\Cache\Driver

APC Db Memcache

Shmop Sqlite Redis Eaccelerator Xcache

Think\Cache

	__construct(\$options='')
	get(\$name)

ThinkPHH

	<code>set(\$name,\$value,\$expire= null)</code>
	<code>rm(\$name)</code>
	<code>clear()</code>


```
namespace Think\Cache\Driver;
use Think\Cache;
/**
 * Test
 */
class Test extends Cache {
    /**
     *
     * @ access public
     * @ param string $name
     * @ return mixed
     */
    public function get($name) {
        //      name
    }
    /**
     *
     * @ access public
     * @ param string $name
     * @ param mixed $value
     * @ param integer $expire
     * @ return boolean
     */
    public function set($name, $value, $expire = null) {
        //
    }
    /**
     *
     * @ access public
     * @ param string $name
     * @ return boolean
     */
    public function rm($name) {
        //      name
    }

    /**
     *
     * @ access public
     * @ return boolean
     */
    public function clear() {
        //
    }
}
```

0

set

```
if($this->options['length']>0) {  
    //  
    $this->queue($name);  
}
```

CACHE x l q]hu-h

	delete(\$options= array())
	select(\$options= array())

selectSql

```
'SELECT%DISTINCT% %FIELD% FROM %TABLE%%JOIN%%WHERE%%GROUP%%HAVING%%ORDER
%%LIMIT% %UNION%'
```

parseTable		%TABLE%
parseWhere		%WHERE%
parseLimit	Limit	%LIMIT%
parseJoin	JOIN	%JOIN%
parseOrder		%ORDER%
parseGroup	group	%GROUP%
parseHaving	having	%HAVING%
parseDistinct	distinct	%DISTINCT%
parseUnion	union	%UNION%
parseField		%FIELD%

Db

parseKey	
parseValue	
parseSet	set
parseLock	

```
'DB_TYPE'=>'odbc', //
```

Think\Log\Driver ,

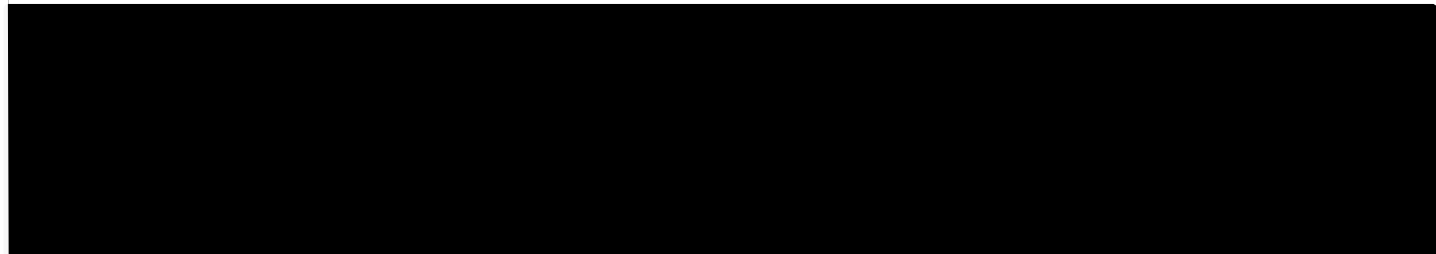
	__construct(\$config= array())
	write(\$log,\$destination= '')

log

Session

session

Think\Session\Driver ,



	__construct(\$config='')
	read(\$filename,\$type='')
	put(\$filename,\$content,\$type='')
	append(\$filename,\$content,\$type='')
	load(\$filename,\$vars=null,\$type='')
	has(\$filename,\$type='')
	unlink(\$filename,\$type='')
	get(\$filename,\$name,\$type='')

type

runtime

html

F F

```
define('STORAGE_TYPE', 'MyStorage');
```

Smarty

TemplateLite SmartTemplate EaseTemplate

Think\Template\Driver

fetch(\$templateFile,\$var)

Smarty

```
namespace Think\Template\Driver;  
class Smarty {
```

```
/**
```

```
*
```

```
* @ access public
```

```
* @ param string $templateFile
```

```
* @ param array $var
```

```
* @ return void
```

Think\Template\TagLib

Think\Template\TagLib

```
namespace Think\Template\Taglib;
use Think\Template\TagLib;
Class Test extends TagLib{
}
```

```
protected $tags = array(
    //
    'input' => array('attr'=>'type,name,id,value','close'=>0), // input
    'textarea' => array('attr'=>'name,id')
);
```

tags

tags

attr	
close	0 1
level	
alias	

" _"

input

```

// input
public function _input($tag,$content) {
    $name = $tag['name'];
    $id = $tag['id'];
    $type = $tag['type'];
    $value = $this->autoBuildVar($tag['value']);
    $str = "<input type='".$type.'" id='".$id.'" name='".$name.'" value='".$value.'" />";
    return $str;
}
// textarea
public function _textarea($tag,$content) {
    $name = $tag['name'];
    $id = $tag['id'];
    $str = '<textarea id="'.$id.'" name="'.$name.'">'.$content.'</textarea>';
    return $str;
}

```

Test

taglib

```
< taglib name='Test' />
```

name

Test

Test

```

< test:input type='radio' id='test' name='mail' value='value' />
< test:textarea id="content" name="content"> $value</test:textarea>

```

XML

Input

value

value

value

```
$this->assign('value','my test value');
```

```

< input type='radio' id='test' name='mail' value='my test value' />
< textarea id="content" name="content"> my test vale</textarea>

```


Behavior

Hello world

AOP " "

AOP

```
app_init  
path_info PATH_INFO  
app_begin  
action_name  
action_begin  
view_begin  
view_parse  
template_filter  
view_filter  
view_end  
action_end  
app_end
```

Common\Conf\tags.php

```
<?php  
return array(  
    'action_begin' => array('Home\Behaviors\TestBehavior', 'Home\Behaviors\Test1Behavior'),  
);
```

Home

test test1

Home

Behaviors

Home\Behaviors\TestBehavior

Home/Behaviors/TestBehavior.class.php

_initialize

```
\Think\Hook::add('action_begin','Home\Behaviors\TestBehavior');
```

```
, test
```

```
<?php
namespace Home\Behaviors;
class TestBehavior extends \Think\Behavior{
    //
    public function run(& $param){

    }
}
```

```
\Think\Behavior      run(& $param)
```

```
\Think\Hook::listen('tag', 'Home\Behaviors\TestBehavior');
//
// tag('tag', 'Home\Behaviors\TestBehavior');
```

```
Hook::listen
listen
run(& $param) $param
```

http://document.thinkphp.cn/manual_3_2.html#taglib

- ThinkPHP\Library\Think\Template\TagLib

```
< taglib name="Home\TagLib\MyTag" />
```

-

```
<?php
namespace Think\Template\TagLib;
use Think\Template\TagLib;
defined('THINK_PATH') or exit();
```

namespace

3.0

Cx

Widget

Widget

Widget

Widget

CateWidget

```
namespace Home\Widget;
use Think\Controller;
class CateWidget extends Controller {
    public function menu(){
        echo 'menuWidget';
    }
}
```

W

Widget

```
{:W('Cate/Menu')}
```

menuWidget

Widget

```
namespace Home\Widget;
use Think\Controller;
class CateWidget extends Controller {
    public function menu($id,$name){
        echo $id.':'.$name;
    }
}
```

```
{:W('Cate/Menu',array(5,'thinkphp'))}
```

menu

5:thinkphp

Widget

```
namespace Home\Widget;
use Think\Controller;
class CateWidget extends Controller {
    public function menu(){
        $menu = M('Cate')->getField('id,title');
        $this->assign('menu',$menu);
        $this->display('Cate:menu');
    }
}
```

CateWigetD i

ThinkPHP\Mode

lite.php

```
return array(  
    //  
    'config' => array(  
        THINK_PATH. nfi/onft ugnf f //
```

```
define('MODE_NAME','lite');  
define('APP_PATH','./Application/');  
require './ThinkPHP/ThinkPHP.php';
```

URL [index.php,](#)

[Apache]

1. httpd.conf [mod_rewrite.so](#)
2. AllowOverride None [None](#) [All](#)
3. [.htaccess](#)

```
<IfModule mod_rewrite.c>
RewriteEngine on
RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^(.*)$ index.php/$1 [QSA,PT,L]
</IfModule>
```

[IIS]

[ISAPI_Rewrite](#) [httpd.ini](#)

```
RewriteRule (.*)$ /index\php?s=$1 [I]
```

[IIS](#) [web.Config](#) [rewrite](#)

```
<rewrite>
<rules>
<rule name="OrgPage" stopProcessing="true">
<match url="^(.*)$" />
<conditions logicalGrouping="MatchAll">
<add input="{HTTP_HOST}" pattern="^(.*)$" />
<add input="{REQUEST_FILENAME}" matchType="IsFile" negate="true" />
<add input="{REQUEST_FILENAME}" matchType="IsDirectory" negate="true" />
</conditions>
<action type="Rewrite" url="index.php/{R:1}" />
</rule>
</rules>
</rewrite>
```

[Nginx]

[Nginx](#) [PATHINFO](#) [Nginx.conf](#)


```
location /{ // .....
    if (!-e $request_filename) {
        rewrite ^(.*)$ /index.php?s=$1 last;
        break;
    }
}
```

PATHINFO WEB

ThinkPHP

URL

ThinkPHP

Nginx

youdomain

```
location /youdomain/{
    if (!-e $request_filename){
        rewrite ^/youdomain/(.*)$ /youdomain/index.php?s=$1 last;
    }
}
```

URL

http://serverName/index.php/ / / /[/ ...]

http://serverName/ / / /[/ ...]

URL

URL

3.2

URL

```
http://serverName/Home/New/index // Home
http://serverName/Admin/Config/index // Admin
http://serverName/User/Member/index // User
```

http://serverName/New/index

Home

URL

```
//
'MODULE_ALLOW_LIST' => array('Home','Admin','User');
'DEFAULT_MODULE' => 'Home', //
```

<http://serverName/New/index>

Home

MODULE_ALLOW_LIST

MODULE_DENY_LIST

MODULE_ALLOW_LIST

Common

Runtime

Runtime

```
//
'MODULE_DENY_LIST' => array('Common','Runtime','User'),
```

<http://serverName/User/Member/index>

MODULE_DENY_LIST

```
'URL_MODULE_MAP' => array('test'=>'admin'),
```

Admin

test

<http://serverName/Admin>

<http://serverName/test>

Admin

MODULE_ALLOW_LIST

```
'MODULE_ALLOW_LIST' => array('Home','Test','User'),
'DEFAULT_MODULE' => 'Home',
'URL_MODULE_MAP' => array('test'=>'admin'),
```

ThinkPHP

IP

IP URL

URL

IP

APP_SUB_DOMAIN_DEPLOY

APP_SUB_DOMAIN_RULES

```
'APP_SUB_DOMAIN_DEPLOY' => 1, // IP
'APP_SUB_DOMAIN_RULES' => array(
    /*
    * 1: ' IP=> ' [/ ];
    * 2: ' IP=> array(' [/ ]','var1=a&var2=b&var3=*');
    */
)
```

2 1

2

IP

DNS

Apache

IP

```
'APP_SUB_DOMAIN_DEPLOY' => 1, //
'APP_SUB_DOMAIN_RULES' => array(
    'admin.domain1.com' => 'Admin', // admin.domain1.com Admin
    'test.domain2.com' => 'Test', // test.domain2.com Test
),
```

<http://www.domain.com/Admin/Index/index><http://www.domain.com/Test/Index/index><http://admin.domain1.com/Index/index><http://test.domain2.com/Index/index>

'APP_S

http://test.admin.domain.com/index

```
'APP_SUB_DOMAIN_DEPLOY' => 1, //
'APP_SUB_DOMAIN_RULES' => array(
  'admin' => 'Admin', // admin      Admin
  '*'     => array('Test','var1=1&var2=*'), //      Test
  '*.user' => array('User','status=1&name=*'), //      User
),
```

http://hello.domain.com/Index/index

Test \$_GET['var1'] = 1 \$_GET['var2'] = 'hello'

http://think.user.domain.com/Index/index

User \$_GET['status'] = 1 \$_GET['name'] = 'think'

" *"

IP

IP

```
'APP_SUB_DOMAIN_DEPLOY' => 1, //
'APP_SUB_DOMAIN_RULES' => array(
  '22.56.78.9' => 'Admin', // 22.56.78.9  Admin
),
```

URL

admin.php

Admin

admin.php

```
//      Admin
define('BIND_MODULE','Admin');
//
define('APP_PATH','./Application/');
require './ThinkPHP/ThinkPHP.php';
```

3.2.0

```
$_GET['m'] = 'Admin';
```

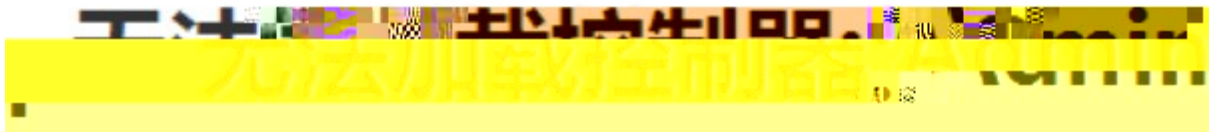
URL

Admin Index test

```
http://serverName/admin.php/Index/test/var/name
```

```
http://serverName/admin.php/Admin/Index/test/var/name
```

:(



```
//      Admin
define('BIND_MODULE','Admin');
//      Index
define('BIND_CONTROLLER','Index');
//
define('APP_PATH','./Application/');
require './ThinkPHP/ThinkPHP.php';
```

URL

http://serverName/admin.php/test/var/name

3.2.0

```
$_GET['m'] = 'Admin';
$_GET['c'] = 'Index';
```

```
//      Admin
define('BIND_MODULE','Admin');
//      Index
define('BIND_CONTROLLER','Index');
//      test
define('BIND_ACTION','test');
//
define('APP_PATH','./Application/');
require './ThinkPHP/ThinkPHP.php';
```

URL

http://serverName/admin.php/var/name

3.2.0

3.2

Lite

Lite

SAE

Lite

Lite

Lite

```
define('BUILD_LITE_FILE',true);
```

Runtime

lite.php

Lite

```
'RUNTIME_LITE_FILE' => APP_PATH.'lite.php'
```

Lite

APP_PATH.'lite.php'

Lite

lite.php

```
return array(
    THINK_PATH.'Common/functions.php',
    COMMON_PATH.'Common/function.php',
    CORE_PATH . 'Think'.EXT,
    CORE_PATH . 'Hook'.EXT,
    CORE_PATH . 'App'.EXT,
    CORE_PATH . 'Dispatcher'.EXT,
    CORE_PATH . 'Model'.EXT,
    CORE_PATH . 'Log'.EXT,
    CORE_PATH . 'Log/Driver/File'.EXT,
    CORE_PATH . 'Route'.EXT,
    CORE_PATH . 'Controller'.EXT,
    CORE_PATH . 'View'.EXT,
    CORE_PATH . 'Storage'.EXT,
    CORE_PATH . 'Storage/Driver/File'.EXT,
    CORE_PATH . 'Exception'.EXT,
    BEHAVIOR_PATH . 'ParseTemplateBehavior'.EXT,
    BEHAVIOR_PATH . 'ContentReplaceBehavior'.EXT,
);
```

lite.php

Lite

lite

Lite

SAE

Lite

Lite

Lite

```
require './ThinkPHP/ThinkPHP.php';
//
require './Runtime/lite.php';
```

Lite

lite.php

lite.php
index.php

Lite



Lite

SESSION

Session

session

Session

session

session

session

```
session(array('name' => 'session_id', 'expire' => 3600));
```

session

id	session_id
name	session_name
path	session_save_path
prefix	session
expire	session.gc_maxlifetime
domain	session.cookie_domain
use_cookies	session.use_cookies
use_trans_sid	session.use_trans_sid
type	session

Session

ThinkApp

SESSION_OPTIONS

SESSION_OPTIONS

session

session

session

SESSION_AUTO_START false

```
'SESSION_AUTO_START' => false
```

session_start

session(['start'])

session

session

Session

```
session('name','value'); // session
```

3.2.3 session

```
session('user.user_id',10); // session
```

session

Session

```
$value = session('name');
// session 3.2.2
$value = session();
```

3.2.3

```
$value = session('user.user_id');
```

session

session

```
session('name',null); // name
```

3.2.3

```
session('user.user_id',null); // session
```

session

```
session(null); // session
```

session

session

```
//      name  session
session('?name');
```

3.2.3

```
session('?user.user_id');
```

session

session

session

```
session([    ]);
```

start	session
pause	session
destroy	session
regenerate	session id

```
session('[pause]'); //      session
session('[start]'); //      session
session('[destroy]'); //      session
session('[regenerate]'); //      session id
```

```

      session          prefix          SESSION_PREFIX
session
session              session
session
session              session          $_SESSION['name']
$_SESSION['    ']['name']
```

session handler

```

session          type          SESSION_TYPE
Session          Library/Think/Session/Driver
```

Cookie

cookie

Cookie

Cookie

Cookie

```
cookie('name','value'); // cookie
cookie('name','value',3600); // cookie
```

cookie

cookie

3600

cookie think_

```
cookie('name','value',array('expire'=>3600,'prefix'=>'think_'))
```

query

```
cookie('name','value','expire=3600&prefix=think_')
```

```
prefix,expire,path,domain httponly **3.2.2 **
```

```
COOKIE_PREFIX COOKIE_EXPIRE COOKIE_PATH
```

```
COOKIE_DOMAIN COOKIE_HTTPONLY
```

cookie

JSON

```
cookie('name',array('value1','value2'));
```

Cookie

cookie

cookie

```
$value = cookie('name');
```

cookie

app_begin

```
'LANG_SWITCH_ON' => true, //
```

```
'LANG_AUTO_DETECT' => true, //
'LANG_LIST'       => 'zh-cn', //
'VAR_LANGUAGE'    => 'l', //
```

Lang

zh-cn.php

en-us.php

zh-tw

ThinkPHP

ThinkPHP

```
throw_exception ' ' ;
```

ADD_USER_ERROR

```
throw_exception L('ADD_USER_ERROR') ;
```

L

Lang

	/Lang/ .php
	/Lang/ .php
	/Lang/ / .php

Home

zh-cn

```

ThinkPHP/Lang/zh-cn.php
Application/Common/Lang/zh-cn.php
Application/Home/Lang/zh-cn.php Home
Application/Home/Lang/zh-cn/user.php Home      User

```

() English (United States) en-us

ThinkPHP

```

return array(
    'lan_define'=> ThinkPHP,
);

```

```

L('define2',' ');
$value = L('define2');

```

```

array('title','require',' ',1),

```

```

' lang_var'=> ' '

```

```

array('title','require','{%lang_var}',1),

```

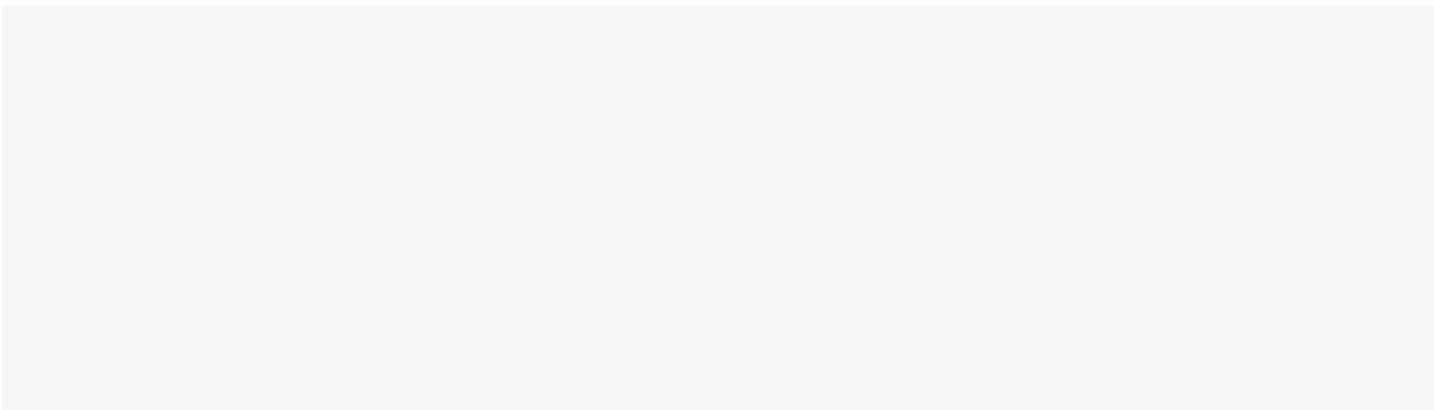
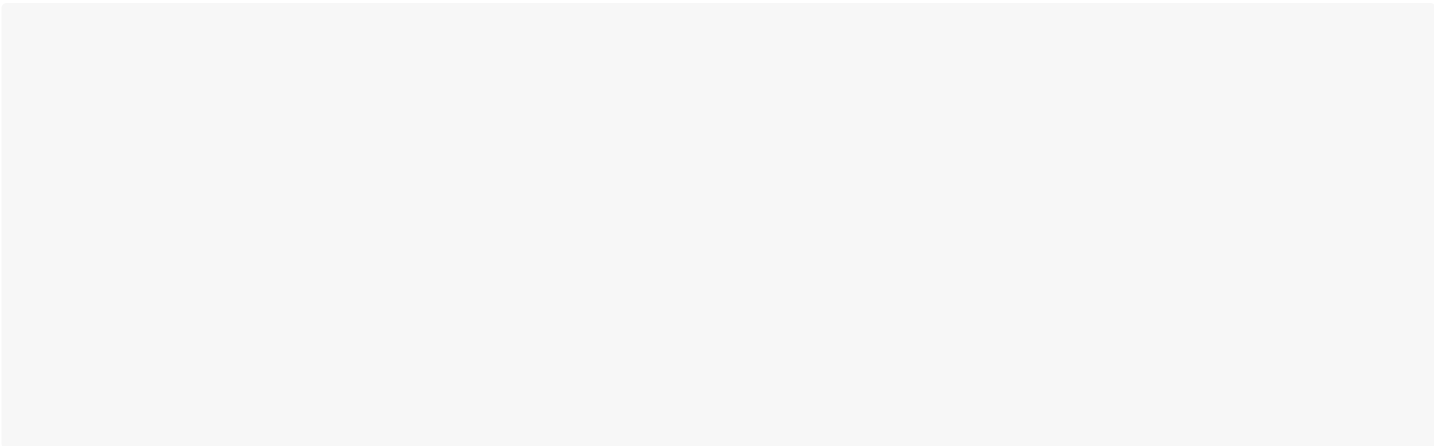
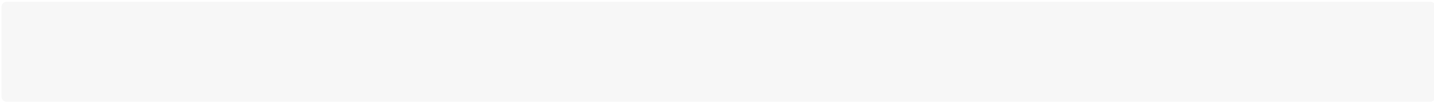
```

{$Think.lang.lang_var}

```

lang_var

'FILE_FORMAT' =>



```

$count    = $User->where($map)->count();//
$page    = new \Think\Page($count,25);//
//
foreach($map as $key=>$val) {
    $page->parameter[$key] = urlencode($val);
}
$show    = $page->show();//

```

Page

setConfig

```
$page->setConfig('header','');
```

setConfig

header	" %TOTAL_ROW% "
prev	" << "
next	" >> "
first	" 1..."
last	" ...%TOTAL_PAGE% "
theme	"%FIRST% %UP_PAGE% %LINK_PAGE% %DOWN_PAGE% %END% "

%FIRST%	
%UP_PAGE%	
%LINK_PAGE%	
%DOWN_PAGE%	
%END%	

```

prev          next          end          num          class          first
                                current

```

ThinkPHP

```
<form action="__URL__/upload" enctype="multipart/form-data" method="post" >
<input type="text" name="name" />
<input type="file" name="photo" />
<input type="submit" value="    " >
</form >
```

enctype="multipart/form-data"

```
<input type='file' name='photo'>
```

```
<input type='file' name='photo1'>
<input type='file' name='photo2'>
<input type='file' name='photo3'>
```

```
<input type='file' name='photo[]'>
<input type='file' name='photo[]'>
<input type='file' name='photo[]'>
```

ThinkPHP

Think\Upload

upload

upload

```

public function upload(){
    $upload = new \Think\Upload();//
    $upload->maxSize = 3145728;//
    $upload->exts = array('jpg', 'gif', 'png', 'jpeg');//
    $upload->rootPath = './Uploads/';//
    $upload->savePath = ''; //
    //
    $info = $upload->upload();
    if(!$info) {//
        $this->error($upload->getError());
    }else{//
        $this->success(' ');
    }
}
}

```

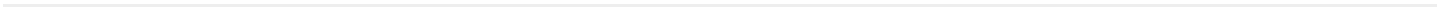
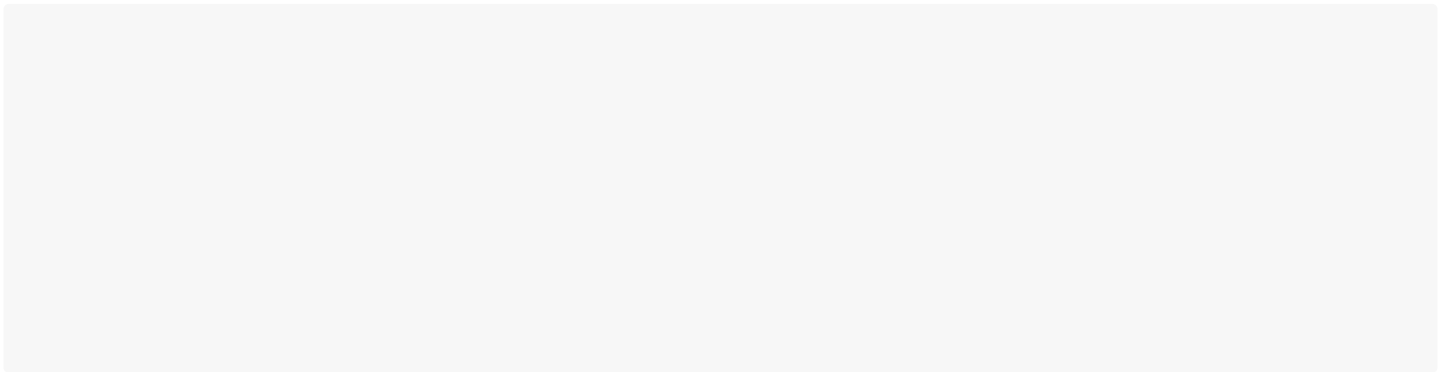
info

Upload

maxSize	0
rootPath	
savePath	
saveName	
saveExt	
replace	false
exts	
mimes	
autoSub	true
subName	
hash	hash true
callback	

```
$config = array(  
    'maxSize' => 3145728,  
    'rootP
```

|



```

$upload = new \Think\Upload();//
$upload->maxSize = 3145728;//
$upload->exts = array('jpg', 'gif', 'png', 'jpeg');//
$upload->rootPath = './Uploads/';//
$upload->savePath = '';//
//
$info = $upload->upload();
if(!$info) {//
    $this->error($upload->getError());
}else{//
    foreach($info as $file){
        echo $file['savepath'].$file['savename'];
    }
}

```

key		
savepath		
name		
savename		
size		
type	MIME	
ext		
md5	md5	hash
sha1	sha1	hash

```

$model = M('Photo');
//
$info = $upload->upload();
//
$data['photo'] = $info['photo']['savename'];
$data['create_time'] = NOW_TIME;
$model->add($data);

```

upload

Upload

uploadOne

```

public function upload(){
    $upload = new \Think\Upload();//
    $upload->maxSize = 3145728;//
    $upload->exts = array('jpg', 'gif', 'png', 'jpeg');//
    $upload->rootPath = './Uploads/';//
    //
    $info = $upload->uploadOne($_FILES['photo1']);
    if(!$info) {//
        $this->error($upload->getError());
    }else{//
        echo $info['savepath'].$info['savename'];
    }
}

```

uploadOne

upload

saveName

saveName

```

//
$upload->saveName = 'time';
// GUID
$upload->saveName = 'com_create_guid';

```

```

//
$upload->saveName = 'myfun';

```

uniqid

saveName

```
// date Y-m-d
$upload->saveName = array('date','Y-m-d');
//
$upload->saveName = array('myFun',array('__FILE__','val1','val2'));
```

FILE

```
myFun('','val1','val2')
```

```
$upload->saveName = time().'mt_rand();
```


saveName

subName

get_user_id

get_user_id

Local

```
'FILE_UPLOAD_TYPE' => 'Ftp',  
'UPLOAD_TYPE_CONFIG' => array(  
    'host' => '192.168.1.200', //  
    'port' => 21, //  
    'timeout' => 90, //  
    'username' => 'ftp_user', //  
    'password' => 'ftp_pwd', // )
```

Ftp

FTP

```
$config = array(  
    'maxSize' = 3145728,  
    'rootPath' = './Uploads/',  
    'savePath' = '',  
    'saveName' = array('uniqid',''),  
    'exts' = array('jpg', 'gif', 'png', 'jpeg'),  
    'autoSub' = true,  
    'subName' = array('date','Ymd'),  
);  
$ftpConfig = array(  
    'host' => '192.168.1.200', //  
    'port' => 21, //  
    'timeout' => 90, //  
    'username' => 'ftp_user', //  
    'password' => 'ftp_pwd', // );  
  
$upload = new \Think\Upload($config,'Ftp',$ftpConfig);//
```

Local

\$config'

```
$Verify = new \Think\Verify();
$Verify->entry();
```



session

```
array('verify_code'=>'          ','verify_time'=>'          ')
```

entry

1

```
//      1
$Verify = new \Think\Verify();
$Verify->entry(1);
```

2

```
//      2
$Verify = new \Think\Verify();
$Verify->entry(2);
```

expire	
useImgBg	false
fontSize	25
useCurve	true
useNoise	true

imageW	0
imageH	0
length	
fontttf	
useZh	
bg	rgb array(243, 251, 254)
seKey	
codeSet	
zhSet	

```
$config = array(
    'fontSize' => 30, //
    'length'   => 3,  //
    'useNoise' => false, //
);
$Verify = new \Think\Verify($config);
$Verify->entry();
```

```
$Verify = new \Think\Verify();
$Verify->fontSize = 30;
$Verify->length = 3;
$Verify->useNoise = false;
$Verify->entry();
```



```
$Verify = new \Think\Verify();
// ThinkPHP/Library/Think/Verify/ttfs/5.ttf
$Verify->fonttff = '5.ttf';
$Verify->entry();
```

```
$Verify = new \Think\Verify();
// ThinkPHP/Library/Think/Verify/bgs
$Verify->useImgBg = true;
$Verify->entry();
```



```
$Verify = new \Think\Verify();
// ThinkPHP/Library/Think/Verify/ttfs/5.ttf
$Verify->useZh = true;
$Verify->entry();
```



ThinkPHP/Library/Think/Verify/zhttfs/

3.2.1

codeSet

```
$Verify = new \Think\Verify();
//
$Verify->codeSet = '0123456789';
$Verify->entry();
```

zhSet

```
$Verify = new \Think\Verify();
$Verify->useZh = true;
//
$Verify->zhSet = ' ';
$Verify->entry();
```

Think\Verify `check`

```
//          $code
function check_verify($code, $id = ""){
    $verify = new \Think\Verify();
    return $verify->check($code, $id);
}
```

Think\Image

Gd

Imagick

Gif

```
$image = new \Think\Image();
```

GD

Imagick

```
$image = new \Think\Image(\Think\Image::IMAGE_IMAGICK);
//
$image = new \Think\Image('Imagick');
```



open

```
$image = new \Think\Image();
$image->open('./1.jpg');
```

```
$image = new \Think\Image(\Think\Image::IMAGE_GD, './1.jpg'); // GD
//
$image = new \Think\Image(\Think\Image::IMAGE_IMAGICK, './1.jpg'); // imagick
```

```
$image = new \Think\Image();
$image->open('./1.jpg');
$width = $image->width(); //
$height = $image->height(); //
$type = $image->type(); //
$mime = $image->mime(); //      mime
$size = $image->size(); //      0      1
```

crop save

```
$image = new \Think\Image();  
$image->open('./1.jpg');  
//          400x400          corp.jpg  
$image->crop(400, 400)->save('./crop.jpg');
```



100 30

```
$image = new \Think\Image();  
$image->open('./1.jpg');  
//          400x400          corp.jpg  
$image->crop(400, 400,100,30)->save('./crop.jpg');
```



thumb

```
$image = new \Think\Image();  
$image->open('./1.jpg');  
//                                150*150                thumb.jpg  
$image->thumb(150, 150)->save('./thumb.jpg');
```



150*150

150

Think\Image

```
IMAGE_THUMB_SCALE    = 1 ; //  
IMAGE_THUMB_FILLED   = 2 ; //  
IMAGE_THUMB_CENTER   = 3 ; //  
IMAGE_THUMB_NORTHWEST = 4 ; //  
IMAGE_THUMB_SOUTHEAST = 5 ; //  
IMAGE_THUMB_FIXED    = 6 ; //
```



```
$image = new \Think\Image();  
$image->open('./1.jpg');  
//          150*150          thumb.jpg  
$image->thumb(150, 150,\Think\Image::IMAGE_THUMB_CENTER)->save('./thumb.jpg');
```



```
$image = new \Think\Image();  
$image->open('./1.jpg');  
//          150*150          thumb.jpg  
$image->thumb(150, 150,\Think\Image::IMAGE_THUMB_NORTHWEST)->save('./thumb.jpg');
```



```
$image = new \Think\Image();  
$image->open('./1.jpg');  
//          150*150          thumb.jpg  
$image->thumb(150, 150,\Think\Image::IMAGE_THUMB_FILLED)->save('./thumb.jpg');
```



```
$image = new \Think\Image();
$image->open('./1.jpg');
//          150*150          thumb.jpg
$image->thumb(150, 150,\Think\Image::IMAGE_THUMB_FIXED)->save('./thumb.jpg');
```





```
$image = new \Think\Image();
$image->open('./1.jpg');
//          440x440          corp.jpg
$image->crop(440, 440)->save('./crop.jpg');
//          ./logo.png          water.gif
$image->water('./logo.png')->save("water.gif");
//          water_o.gif
$image->open('./1.jpg')->water('./logo.png')->save("water_o.gif");
```

water

Think\Image

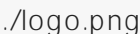

```
IMAGE_WATER_NORTHWEST = 1 ; //
IMAGE_WATER_NORTH     = 2 ; //
IMAGE_WATER_NORTHEAST = 3 ; //
IMAGE_WATER_WEST      = 4 ; //
IMAGE_WATER_CENTER    = 5 ; //
IMAGE_WATER_EAST      = 6 ; //
IMAGE_WATER_SOUTHWEST = 7 ; //
IMAGE_WATER_SOUTH     = 8 ; //
IMAGE_WATER_SOUTHEAST = 9 ; //
```

```
$image = new \Think\Image();  
//    
$image->open('./1.jpg')->water('./logo.png',\Think\Image::IMAGE_WATER_NORTHWEST)->save("water.jpg");
```



0~100

80

```
$image = new \Think\Image();  
//    
$image->open('./1.jpg')->water('./logo.png',\Think\Image::IMAGE_WATER_NORTHWEST,50)->save("water.jpg");
```



1.ttf

```
$image = new \Think\Image();  
//          ThinkPHP          new.jpg  
$image->open('./1.jpg')->text('ThinkPHP','./1.ttf',20,'#000000',\Think\Image::IMAGE_WATER_SOUTHE  
AST)->save("new.jpg");
```



RESTFuI

REST

REST(Representational State Transfer)

REST

1 resource

2 resource identifier

3 generic connector interface

4

5 stateless

REST

REST

HTTP URI

XML

HTML

REST

		REST
U etacr t	U etac r wU etacr n wU etacr	





```
namespace Home\Controller;
use Think\Controller\RestController;
class BlogController extends RestController{
}
```

REST

RestController		rest
allowMethod	defaultMethod	allowType
	defaultType	allowOutputType

allowMethod	REST	array('get','post','put','delete')
defaultMethod	REST	get
allowType	REST	array('html','xml','json','rss')
defaultType	REST	html
allowOutputType	REST	array('xml' => 'application/xml', 'json' => 'application/json', 'html' => 'text/html',)

REST

RESTFul

```

namespace Home\Controller;
use Think\Controller\RestController;
Class InfoController extends RestController {
    Public function rest() {
        switch ($this->_method){
            case 'get': // get
                if ($this->_type == 'html'){
                }elseif($this->_type == 'xml'){
                }
                break;
            case 'put': // put
                break;
            case 'post': // post
                break;
        }
    }
}

```

Rest

\$this->_type

\$this->_method

REST

response

REST

```
$this->response($data,'json');
```

Response

data

xml/json/html

Restful

	Restful	read_get_pdf
-	defaultMethod	read_pdf
-	defaultType	read_post

restful

InfoController

```

namespace Home\Controller;
use Think\Controller\RestController;
Class InfoController extends RestController {
    protected $allowMethod = array('get','post','put'); // REST
    protected $allowType = array('html','xml','json'); // REST

    Public function read_get_html(){
        // id 1 Info html
    }

    Public function read_get_xml(){
        // id 1 Info XML
    }
    Public function read_xml(){
        // id 1 Info XML
    }
    Public function read_json(){
        // id 1 Info json
    }
}

```

URL

```
http://www.domain.com/Info/read/id/1.xml
```

	Info	read		InfoController
read_get_xml	read_xml		URL	

```
http://www.domain.com/Info/read/id/1.json
```

read_json

URL

```
http://www.domain.com/Info/read/id/1.rss
```

rss

read_html

REST

3.2

REST

```
'blog/:id' => array('blog/read','status=1',array('ext'=>'xml','method'=>'get')),
```

blog/5

blog/read/id/5

xml

get


```
'blog/:id' => array('blog/update','',array('ext'=>'xml','method'=>'put')),
```

REST

```
array('blog/:id','blog/read','status=1',array('ext'=>'xml','method'=>'get')),
array('blog/:id','blog/update','',array('ext'=>'xml','method'=>'put')),
```

REST

rest

rest

RPC

RPC Remote Procedure Call Protocol

RPC

TCP UDP

OSI

RPC

RPC

RPC /

ThinkPHP

RPC

PHPRPC

HPRose

JsonRPC

Yar

PHPRPC

PHPRPC

3.0

phprpc

http://www.phprpc.org/zh_CN/

ThinkPHP

PHPRpc

Think\Controller\RpcController

```

namespace Home\Controller;
use Think\Controller\RpcController;
class ServerController extends RpcController{
    public function test1(){
        return 'test1';
    }
    public function test2(){
        return 'test2';
    }
    private function test3(){
        return 'test3';
    }
    protected function test4(){
        return 'test3';
    }
}

```

ServerController

PHPRpc

http://serverName/index.php/Home/Server

rpc

test1

test2

PHPRpc

URL

allowMethodList

```

namespace Home\Controller;
use Think\Controller\RpcController;
class ServerController extends RpcController{
    protected $allowMethodList = array('test1','test2');
    public function test1(){
        return 'test1';
    }
    public function test2(){
        return 'test2';
    }
    public function test3(){
        return 'test3';
    }
}

```

test1

test2

rpc

PHPRpc Server

ThinkPHP

```

namespace Home\Controller;
use Think\Controller;
class IndexController extends Controller {
    public function index(){
        Vendor('phpRPC.phprpc_client');
        $client = new \PHPRPC_Client('http://serverName/index.php/Home/Server');
        //
        // $client = new \PHPRPC_Client();
        // $client->useService('http://serverName/index.php/Home/Server');
        $result = $client->test1();
    }
}

```

test1

PHP

Hprose

Hprose (High Performance Remote Object Service Engine)

MIT

nodeJS, C++, .NET, Java,

Delphi, Objective-C, ActionScript, JavaScript, ASP, PHP, Python, Ruby, Perl

Hprose

PHPRPC

PHPRPC

hprose

PHPRPC

PHPRPC

<http://www.hprose.com/>

ThinkPHP

Hprose

PHPRPC

Think\Controller\HproseController

```

namespace Home\Controller;
use Think\Controller\HproseController;
class ServerController extends HproseController{
    public function test1(){
        return 'test1';
    }
    public function test2(){
        return 'test2';
    }
}

```

hprose

debug crossDomain P3P get

```

namespace Home\Controller;
use Think\Controller\HproseController;
class ServerController extends HproseController{
    protected $crossDomain = true;
    protected $P3P = true;
    protected $get = true;
    protected $debug = true;

    public function test1(){
        return 'test1';
    }
    public function test2(){
        return 'test2';
    }
}

```

ThinkPHP Hprose

```

namespace Home\Controller;
use Think\Controller;
class IndexController extends Controller {
    public function index(){
        vendor('Hprose.HproseHttpClient');
        $client = new \HproseHttpClient('http://serverName/index.php/Home/Server');
        //
        //$client = new \HproseHttpClient();
        //$client->useService('http://serverName/index.php/Home/Server');
        $result = $client->test1();
    }
}

```

JsonRPC

json-rpc json xml-rpc webservice
 hessian java-rpc

ThinkPHP3.2 JsonRPC

```

namespace Home\Controller;
use Think\Controller\JsonRpcController;
class ServerController extends JsonRpcController {
    public function index(){
        return 'Hello, JsonRPC!';
    }
    //
    public function test($name=''){
        return "Hello, {$name}!";
    }
}

```

public

```

namespace Home\Controller;
use Think\Controller;
class IndexController extends Controller {
    public function index(){
        vendor('jsonRPC.jsonRPCClient');
        $client = new \jsonRPCClient("http://serverName/index.php/Home/Server");
        $result = $client->index();
        var_dump($result); //      Hello, JsonRPC!
        $result = $client->test('ThinkPHP');
        var_dump($result); //      Hello, ThinkPHP!
    }
}

```

Yar

Yar (yet another RPC framework) PHP RPC , RPC (xml-rpc, soap) ,
(msgpack, json, php),

Yar

Yar

<http://pecl.php.net/package/yar>

Yar

```
namespace Home\Controller;
use Think\Controller\YarController;
class ServerController extends YarController {
    public function index(){
        return 'Hello, Yar RPC!';
    }
    public function hello($name=''){
        return 'Hello, {$name}!';
    }
}
```

Yar

GET

```
namespace Home\Controller;
use Think\Controller;
class IndexController extends Controller {
    public function index(){
        $client = new \Yar_client('http://serverName/index.php/Home/Server');
        $result = $client->index();
        var_dump($result); //      Hello, Yar!
    }
}
```

SAE

SAE

Sina App Engine

SAE

Web

SAE

SAE

IO

MemcacheX

Storage

IO

IO

IO

SAE

SAE

SAE

SAE

ThinkPHP3.2

SAE

ThinkPHP

SAE

SAE

ThinkPHP

SAE

SAE

SAE ThinkPHP SAE

SAE SAE

		SAE
S	File	Memcache
F	File	KVDB
	Html	KVDB
SQL	File Xcache APC	KVDB

SAE Log ThinkPHP SAE
 sae_debug Think\Log\Driver\Sae

Runtime/Logs

SAE SAE

<http://sae.sina.com.cn/?m=applog>

" debug"

Think\Upload

SAE Storage Storage

SAE Storage domain <http://sae.sina.com.cn/?m=storage>

m=storage

domain domain

```
$upload->rootPath = './Public/';
$upload->savePath = 'Uploads/';
```

Public domain

domain Uploads SAE Storage

UploadFile SAE

"/Public/upload/1.jpg" /Public Public

"/Public/upload/1.jpg" SAE Public/upload storage
 /Public/ storage SAE
 SAE Conf/config_sae.php

```
<?php
return array(
  'TMPL_PARSE_STRING' => array(
    '/Public/upload' => $st->getUrl('public','upload'),
  )
);
```

SAE /Public/upload storage SAE

index.php

SAE .htaccess SAE AppConfig
 config.yaml

```
handle:
- rewrite: if(!is_dir() && !is_file() && path~"^(.*)$") goto "index.php/$1"
```

http://serverName/index.php/Blog/read/id/1

http://serverName/Blog/read/id/1

ThinkPHP SAE SAE SAE

1 SAE

ThinkPHP ThinkPHP SAE KVDB
 ThinkPHP F SAE Memcache S
 SAE SAE ThinkPHP
 SAE

2 SAE

SAE

Conf/config.php

SAE

SAE

Conf/config_sae.php

SAE

IP

get_client_ip

IP

```
$ip = get_client_ip();
```

IP

Org\Net\IpLocation

IP

```
$ip = new \Org\Net\IpLocation('UTFWry.dat'); // IP
$area = $ip->getlocation('203.34.5.66'); // IP
```

IP

UTF8

ip

IP

IpLocation

```
URL_COMMON          URL  0
URL_PATHINFO PATHINFO URL  1
URL_REWRITE REWRITE URL  2
URL_COMPAT          URL  3
EXT                 .class.php
THINK_VERSION
```

```
THINK_PATH
APP_PATH
LIB_PATH          THINK_PATH.'Library/'
CORE_PATH        LIB_PATH.'Think/'
MODE_PATH        THINK_PATH.'Mode/'
BEHAVIOR_PATH    LIB_PATH.'Behavior/'
COMMON_PATH      APP_PATH.'Common/'
VENDOR_PATH      LIB_PATH.'Vendor/'
RUNTIME_PATH     APP_PATH.'Runtime/'
HTML_PATH        APP_PATH.'Html/'
CONF_PATH        COMMON_PATH.'Conf/'
LANG_PATH        COMMON_PATH.'Lang/'
LOG_PATH         RUNTIME_PATH.'Logs/'
CACHE_PATH       RUNTIME_PATH.'Cache/'
TEMP_PATH        RUNTIME_PATH.'Temp/'
DATA_PATH        RUNTIME_PATH.'Data/'
ADDON_PATH       APP_PATH.'Addon'  3.2.3
```

IS_CGI	CGI		
IS_WIN	Windows		
IS_CLI			
__ROOT__			
__APP__			
__MODULE__		URL	
__CONTROLLER__			URL
__ACTION__		URL	
__SELF__	URL		
__INFO__	PATH_INFO		
__EXT__	URL		
MODULE_NAME			

```

'APP_USE_NAMESPACE' => true, // 3.2.1
'APP_SUB_DOMAIN_DEPLOY' => false, //
'APP_SUB_DOMAIN_RULES' => array(), //
'APP_DOMAIN_SUFFIX' => "", // com.cn net.cn
'ACTION_SUFFIX' => "", //
'MULTI_MODULE' => true, // false DEFAULT_MODULE
'MODULE_DENY_LIST' => array('Common','Runtime'), //
'MODULE_ALLOW_LIST' => array(), //
'CONTROLLER_LEVEL' => 1,
'APP_AUTOLOAD_LAYER' => 'Controller,Model', //
3.2.1
'APP_AUTOLOAD_PATH' => "", // 3.2.1

```

```

'DEFAULT_M_LAYER' => 'Model', //
'DEFAULT_C_LAYER' => 'Controller', //
'DEFAULT_V_LAYER' => 'View', //
'DEFAULT_LANG' => 'zh-cn', //
'DEFAULT_THEME' => "", //
'DEFAULT_MODULE' => 'Home', //
'DEFAULT_CONTROLLER' => 'Index', //
'DEFAULT_ACTION' => 'index', //
'DEFAULT_CHARSET' => 'utf-8', //
'DEFAULT_TIMEZONE' => 'PRC', //
'DEFAULT_AJAX_RETURN' => 'JSON', // AJAX , JSON XML ...
'DEFAULT_JSONP_HANDLER' => 'jsonpReturn', // JSONP
'DEFAULT_FILTER' => 'htmlspecialchars', // I ...

```

Cookie

```

'COOKIE_EXPIRE' => 0, // Cookie
'COOKIE_DOMAIN' => "", // Cookie
'COOKIE_PATH' => '/', // Cookie
'COOKIE_PREFIX' => "", // Cookie
'COOKIE_HTTPONLY' => "", // Cookie httponly 3.2.2

```

```

'DB_TYPE'          => "", //
'DB_HOST'          => "", //
'DB_NAME'          => "", //
'DB_USER'          => "", //
'DB_PWD'           => "", //
'DB_PORT'          => "", //
'DB_PREFIX'        => "", //
'DB_FIELDTYPE_CHECK' => false, // 3.2.3
'DB_FIELDS_CACHE'  => true, //
'DB_CHARSET'       => 'utf8', // utf8
'DB_DEPLOY_TYPE'   => 0, // :0 ( )1 ( )
'DB_RW_SEPARATE'   => false, //
'DB_MASTER_NUM'    => 1, //
'DB_SLAVE_NO'      => "", //
'DB_SQL_BUILD_CACHE' => false, // SQL 3.2.3
'DB_SQL_BUILD_QUEUE' => 'file', // SQL file xcache apc 3.2.3
'DB_SQL_BUILD_LENGTH' => 20, // SQL 3.2.3
'DB_SQL_LOG'       => false, // SQL 3.2.3
'DB_BIND_PARAM'    => false, //
'DB_DEBUG'         => false, // 3.2.3
'DB_LITE'          => false, // Lite 3.2.3

```

```

'DATA_CACHE_TIME'   => 0, // 0
'DATA_CACHE_COMPRESS' => false, //
'DATA_CACHE_CHECK'  => false, //
'DATA_CACHE_PREFIX' => "", //
'DATA_CACHE_TYPE'   => 'File', // , :File|Db|Apc|Memcache|Shmop|Sqlite|Xcache|A
pachenote|Eaccelerator
'DATA_CACHE_PATH'   => TEMP_PATH, // ( File )
'DATA_CACHE_SUBDIR' => false, // ( )
'DATA_PATH_LEVEL'   => 1, //

```

SESSION

```
'SESSION_AUTO_START' => true, // Session
'SESSION_OPTIONS'    => array(), // session type name id path expire domain
'SESSION_TYPE'       => "", // session handler session handler
'SESSION_PREFIX'     => "", // session
```

```
'TMPL_CONTENT_TYPE' => 'text/html', //
'TMPL_ACTION_ERROR' => THINK_PATH.'Tpl/dispatch_jump.tpl', //
'TMPL_ACTION_SUCCESS' => THINK_PATH.'Tpl/dispatch_jump.tpl', //
'TMPL_EXCEPTION_FILE' => THINK_PATH.'Tpl/think_exception.tpl', //
'TMPL_DETECT_THEME' => false, //
'TMPL_TEMPLATE_SUFFIX' => '.html', //
'TMPL_FILE_DEPR' => '/', // CONTROLLER_NAME ACTION_NAME
'TMPL_ENGINE_TYPE' => 'Think', // Think
'TMPL_CACHFILE_SUFFIX' => '.php', //
'TMPL_DENY_FUNC_LIST' => 'echo,exit', //
'TMPL_DENY_PHP' => false, // PHP
'TMPL_L_DELIM' => '{', //
'TMPL_R_DELIM' => '}', //
'TMPL_VAR_IDENTIFY' => 'array', // , 'obj'
'TMPL_STRIP_SPACE' => true, // html
'TMPL_CACHE_ON' => true, // , false
'TMPL_CACHE_PREFIX' => "", //
'TMPL_CACHE_TIME' => 0, // 0 ( : )
'TMPL_LAYOUT_ITEM' => '{__CONTENT__}', //
'LAYOUT_ON' => false, //
'LAYOUT_NAME' => 'layout', // layout
```

URL

```
'URL_CASE_INSENSITIVE' => true, // false URL true
'URL_MODEL' => 1, // URL , 0 1 2 3,
// 0 ( ); 1 (PATHINFO ); 2 (REWRITE ); 3 ( ) PATHINFO
'URL_PATHINFO_DEPR' => '/', // PATHINFO
'URL_PATHINFO_FETCH' => 'ORIG_PATH_INFO,REDIRECT_PATH_INFO,REDIRECT_URL', //
PATH_INFO SERVER
'URL_REQUEST_URI' => 'REQUEST_URI', // REQUEST_URI
'URL_HTML_SUFFIX' => 'html', // URL
'URL_DENY_SUFFIX' => 'ico|png|gif|jpg', // URL
'URL_PARAMS_BIND' => true, // URL Action
'URL_PARAMS_BIND_TYPE' => 0, // URL 0 1
'URL_404_REDIRECT' => "", // 404
'URL_ROUTER_ON' => false, // URL
'URL_ROUTE_RULES' => array(), //
'URL_MAP_RULES' => array(), // URL
```

```
'VAR_MODULE'          => 'm', //
'VAR_CONTROLLER'      => 'c', //
'VAR_ACTION'          => 'a', //
'VAR_AJAX_SUBMIT'     => 'ajax', //    AJAX
'VAR_JSONP_HANDLER'   => 'callback',
'VAR_PATHINFO'        => 's', //    PATHINFO    ?s=/module/action/id/1
    URL_PATHINFO_DEPR
'VAR_TEMPLATE'        => 't', //
'VAR_ADDON'           => 'addon', //    3.2.2
```

```
'HTTP_CACHE_CONTROL' => 'private', //
'CHECK_APP_DIR'       => true, //    w
```

- index.php index_old.php
- Application App
- Application ThinkPHP index.php
- index.php



欢迎使用 ThinkPHP !

/ /

Application/Common	App	Common	Conf	Lang
Application/Home	Common/common.php		function.php	

App/Common/common.php	=>	Application/Home/Common/function.php
App/Common/extend.php	=>	Application/Home/Common/extend.php
App/Conf/Config.php	=>	Application/Home/Conf/config.php
App/Lang/zh-cn/common.php	=>	Application/Home/Lang/zh-cn.php

Lib	Application/Home
-----	------------------

App/Lib/Action	=>	Application/Home/Action
App/Lib/Model	=>	Application/Home/Model

Tpl	Application/Home	View
-----	------------------	------

App/Tpl	=>	Application/Home/View
---------	----	-----------------------

Application
 Home
 Conf
 Common
 Action
 Model
 View

Home

App/Common/common.php => Application/Common/Common/function.php
App/Common/Home/function.php => Application/Home/Common/function.php

Conf

App/Conf/Config.php => Application/Common/Conf/config.php
App/Conf/Home/config.php => Application/Home/Conf/config.php

App/Lang/zh-cn/common.php => Application/Common/Lang/zh-cn.php
App/Lang/zh-cn/Home/lang.php => Application/Home/Lang/zh-cn.php

App/Lib/Action/Home => Application/Home/Action
App/Lib/Action/Admin => Application/Admin/Action

App/Lib/Model => Application/Common/Model
App/Lib/Model/Home => Application/Home/Model

App/Tpl/Home => Application/Home/View

```
Application
  Common
    Common
    Conf
  Home      Home
    Action
    Common
    Conf
    Lang
    Model
    View
```

Home

```
App/Common/common.php => Application/Common/Common/function.php
```

Conf Lang Application/Common

```
App/Conf/Config.php      => Application/Common/Conf/config.php
App/Lang/zh-cn/common.php => Application/Common/Lang/zh-cn.php
```

App/Modules

```
App/Modules/Home => Application/Home
```

Home Tpl View

```

Application
  Common
    Common
    Conf
  Home      Home
    Action
    Common
    Conf
    Lang
    Model
    View

```

Home

Application/Common/config.php

```

'DEFAULT_C_LAYER'    => 'Action', //
'MODULE_ALLOW_LIST' => array('Home','Admin',...) //
'DEFAULT_MODULE'    => 'Home' //

```

```

'MULTI_MODULE'      => false, //
'DEFAULT_MODULE'    => 'Home' //

```

DB_HOST	localhost	ip
PHP5.3	TP	

```
'DB_HOST'=>'localhost'
```

```
'DB_HOST'=>'127.0.0.1'
```

```
APP_GROUP_LIST  
APP_GROUP_MODE  
APP_AUTOLOAD_PATH  
APP_TAGS_ON  
APP_GROUP_PATH  
DEFAULT_APP  
DEFAULT_GROUP  
VAR_GROUP  
LOG_DEST  
LOG_EXTRA
```

```
DEFAULT_MODULE => DEFAULT_CONTROLLER
```

```
'Think\Page' => CORE_PATH.'Page'.EXT,  
'Think\Auth' => CORE_PATH.'Auth'.EXT,
```

```
'app_begin' => array('Behavior\Cron','Behavior\BrowserCheck'),
```

3.2

Application/Common/config.php

URL_ROUTER_ON

URL_ROUTE_RULES

URL

Application/Home/Action


```

class UserAction extends Think\Action{
}
class UserModel extends Think\Model{
}

```

MODEL_INSERT	self::MODEL_INSERT
MODEL_UPDATE	self::MODEL_UPDATE
MODEL_BOTH	self::MODEL_BOTH
MUST_VALIDATE	self::MUST_VALIDATE
EXISTS_VALIDATE	self::EXISTS_VALIDATE
VALUE_VALIDATE	self::VALUE_VALIDATE

halt

_404

ThrowException

E

common.php

Common/function.php

Think\Controller

Think\Action

_get('id')	I('get.id')
_post('id')	I('post.id')
_put('id')	I('put.id')
_param('id')	I('id')
_request('id')	I('request.id')
_cookie('id')	I('cookie.id')
_server('id')	I('server.id')
_globals('id')	I('globals.id')

ThinkPHP_old

ThinkPHP3.2

ThinkPHP

QQ

Misn

luofei deeka yangweijie vus520