# 概要

Zend\_Db\_Adapter 是 zendframework 的数据库抽象层 API，基于 pdo，你可以使用 Zend\_Db\_Adapter 连接和处理多种数据库，包括 microsoft SQL Server，MySql，SQLite 等等。要针对不同的数据库实例化一个 Zend\_Db\_Adapter 对象，需要将 adapter 的名字和描述数据库连接的参数数组作为参数，静态调用 Zend\_Db::factory() 方法。

# 链接方法

## 设置配置文件

1.配置/application/config/application.ini文件

[mysql]

db.adapter=PDO\_MYSQL

db.params.host=192.168.0.71

db.params.username=root

db.params.password=

1. 修改bootstarp.php

//适配器

//配置文件的地址大写字母是一些框架里的常量

$url=constant("APPLICATION\_PATH").DIRECTORY\_SEPARATOR.'configs'.DIRECTORY\_SEPARATOR.'application.ini';

//去读数据库的配置文件具体的是去读 [mysql] 节点下的配置

$dbconfig=new Zend\_Config\_Ini($url,"mysql");

//设置一个数据库工厂读取

$db=Zend\_Db::factory($dbconfig->db);

//设置查询编码

$db->query('SET NAMES UTF8');

//设置默认适配器

Zend\_Db\_Table::setDefaultAdapter($db);

## 二、使用Zend\_Db\_Adapter

1.//引入文件

require\_once 'Zend/Db.php';

//设置参数

$params = array (

'host' => 'localhost',

'username' => 'root',

'password' => '',

'dbname' => 'test'

);

//获得API

$db = Zend\_Db::factory('PDO\_MYSQL', $params);

//设置默认适配器

Zend\_Db\_Table::setDefaultAdapter($db);

## 使用表模型

自建一个表模型model

class tableModel extends Zend\_Db\_Table

{

protected $\_name='p1\_role';

protected $\_primary='id';

}

# Zend\_db\_Adapter与zend\_db\_Table区别

（zend\_db\_Adapter是zend\_db\_Adapter\_Abstract类，zend没有zend\_db\_Adapter类）

Zend\_db\_Adapter可以使用事务。

Zend\_db\_Adapter没有find方法。

Zend\_db\_Table可以获取默认adapter。

Zend\_db生成对象就是通过zend\_db\_Adapter\_Abstract完成。

Zend\_db\_Table通过调用zend\_db\_Adapter\_Abstract完成操作。

# 获取Adapter

需设置默认适配器：Zend\_Db\_Table::setDefaultAdapter($db);。

--------------------------------、

1.初始化类（Zend\_db\_Table中有获取默认Adapter的方法）

$db = new Zend\_db\_Table;

2.获取默认的Adapter

$db->getAdapter();

或

$db::getDefaultAdapter();

# 使用zend\_db\_Adapter\_Abstract

## 一、添加

//$table 是表名，$addInfo填写的数据

$db->insert($table, $addInfo);

//获取最后插入的ID

$last\_insert\_id = $db->lastInsertId();

## 二、删除

//$table 是表名，$data 是数据

//注入，防sql注入

$where = $db->quoteInto('id = ?', $data);

//删除，删除数据并得到影响的行数

$db->delete($table, $where);

## 三、更新

$where = $db->quoteInto("id = ?", $data['id']);

// 更新表数据,返回更新的行数

$db->update($table, $set, $where);

## 查找

使用fetch系列函数

* fetchAll() 查询所有项（二维数组）
* fetchAssoc()
* fetchCol()
* fetchOne()
* fetchPairs()
* fetchRow() 查询一条（一维数组）

例：

--------------------------、

$result = $db->fetchAll(

"SELECT \* FROM round\_table WHERE noble\_title = :title",

array('title' => 'Sir')

);

--------------------------、

$role = 1;

$sql = "select \* from ${tableRole} where id=?";

$row = $db->fetchRow($sql, $role);

--------------------------、

## 执行自定义的sql

-----------------------------------------------------------------、

$sql = $db->quoteInto(

'SELECT \* FROM test WHERE id = ?',

'3'

);

$result = $db->query($sql);

// 使用PDOStatement 对象$result 将所有结果数据放到一个数组中

$rows = $result->fetchAll();

------------------------------------/

$stmt = $db->prepare('SELECT \* FROM example WHERE date > :placeholder');

$stmt->bindValue('placeholder', '2006-01-01');

$stmt->execute();

// 使用PDOStatement 对象$result 将所有结果数据放到一个数组中

$rows = $stmt->fetchAll();

# 使用zend\_db\_Table

# 样例：在适配器中设置相关

//适配器

$params = array (

'host' => '127.0.0.1',

'username' => 'root',

'password' => '',

'dbname' => 'pzhcdb'

);

$db = Zend\_Db::factory ( 'PDO\_MYSQL', $params );

// 为所有的Zend\_Db\_Table对象设定默认的adapter

require\_once 'Zend/Db/Table.php';

//设置一个数据库工厂读取

//设置查询编码

$db->query('SET NAMES UTF8');

//设置适配器

//这句可以没有，这是设置默认的adapter的

Zend\_Db\_Table::setDefaultAdapter($db);

$roleInfo = array(

'admin\_id' => $AdminId,

'role\_id' => $data['privilege'],

);

$table = "p1\_role\_admin";

$roleId = $db->insert($table, $roleInfo);