
Sedge

Sedge 数据访问层(DAL)说明文档

开源框架：作业提交平台的数据访问层(DAL)说明文档

1. 概述

本文档对数据访问层的代码、功能、接口进行说明，便于开发者在上层开发中正确调用。

此外，**对数据库的所有操作，必须通过 DAL 接口进行**，不得直接执行 SQL 语句。

2. 包含文件

要操作数据库，首先包含 DAL 层的“__DB.php”文件，使用 require_once()函数确保只包含一次。

例如 BLL 层的_login.php 中要进行数据库操作，在文件的开始处使用：

```
require_once('dal/__DB.php');
 //假设__DB.php 位于 dal 文件夹下
```

3. 创建实例

DAL 采用面向对象开发，对每张**数据表**的操作封装成一个**类**。在使用时，要对哪张数据表操作，就创建那个类的实例。

例如要进行用户登录的业务，那么就得查 user 表。user 表的类名是 User，因此：

```
$user = new User;
```

然后调用 User 类提供的接口进行操作。

User 类中提供了登录函数，例如要使用**邮箱登录**

```
if($user -> login_by_email($login, $pass))
{
        // 返回 true，登录成功
}
else
{
        // 返回 false，登录失败
}
```

怎么样，短短几行代码就完成了用户登录，完全不需要了解 SQL 底层过程。

4. 类的概述

在介绍每个类的接口前，先对类的结构进行简单描述，使读者对类有个大概印象。

（1） Base_Class

首先要说的是抽象类（abstract）Base_Class，这个类中的 protected 方法提供给子类调用，要注意的是 public 方法（public 关键字默认省略），可以被子类的实例直接调用。
也就是说，Base_Class 中的 public 方法的代码，虽然不会出现在子类中，但是每个子类都继承了这些方法，因此每个类都可以使用。
这些方法分别是：

a. 对数据库的增删改查

```
insert($para)
delete_by_id($id)
update_by_id($para, $id)
get_by_id($id)
```

这里的$para 是一个关联数组，本质是 key-value 对，key 是字段名，value 是值。例如要插入用户名和密码，使用 array 函数构造数组：

```
$para = array(
    “user_name” => “月之叶落”,
    “password” => “123456”,
);
```

这样，使用 insert($para)就可以向数据表相应字段插入数据。
（此例有两个不妥当的地方，一是 Sedge 的 user 表中这两个字段的名称应该是 user_login、user_pass，二是密码实际上是密文不是明文）
这里的$id 是指主键的 id，这些方法是根据主键进行删改查。

b. mode 切换

```
simple_mode()
complex_mode()
```

类的封装不能照顾到每一个 SQL 查询，既要提高查询效率，又要简化接口功能，因此使用具有**两种模式**的折衷方案。

在做 SQL 查询的时候，如 get_by_id($id) 方法是查找主键值等于$id 的记录，比如我们要查找用户 id 为 6 的记录，在 User 的实例中使用 get_by_id(6)即可。问题是，如果我们只想获得用户的昵称（user_nickname），却要返回整条记录，在效率上是很不划算的（因为除了 user_nickname，还有 10 个字段）。如果让用户传入想查找的字段，这样就给编码增加了负担，增加了维护成本（如果用户想查找其中的 9 个字段，就得键入 9 个单词，并且有可能出错）。

两种模式是这样工作的，simple_mode 返回部分字段，这些字段是最常被使

用的字段，可以满足一般需要。complex_mode 返回全部字段，用于特殊情况。这样就平衡了效率与复杂性，真牛 X 啊。

返回部分字段的设置存储在$what 中，比如 User 类的$what = 'ID, user_nickname';

有些表因为字段太少，所以没有 simple_mode。

c. 字段设置

有些读者可能要问了，我觉得两种模式还不够用，如果我偏要返回某几个字段怎么办？

嗯，已经给你想好了，以下两个方法：

set_select($what)

用于设置字段，设置完成后，在 simple_mode 下查询即可。

set_default()

用于将字段恢复到默认设置。

d. Order 设置

在 SQL 中 select 的时候，不是还有个 Order by 语句吗？如果我想让查询结果按照某个字段排序，用以下方法：

set_order($order)

设置 order by 的内容，比如想按照 date 字段逆序：

```
set_order('date desc');
```

以后每次查询，都会按照 date 逆序排序。

clear_order()

清除 order 设置，不排序。

好了，以上方法是每个类都有的。

（2） 子类

大部分子类都有以下几个属性

```
protected $table_name = 'ng_user';  // 数据表名
protected $id_name = 'ID';
// 主键的键名，若主键大于 1 个，此属性无意义
protected $mode = 0;          // 默认为 simple_mode
protected $default_what = 'ID, user_nickname';
protected $what = 'ID, user_nickname';

//字段设置，两个变量的值相等，$default_what 用于恢复默认
```

4. 类的接口

下面将详细介绍每个类的接口。（只介绍 public 方法，protected 方法仅在类的内部使用，并不是外部接口）

约定：
多个参数用分号“;”隔开。
空白表示“无”。
返回值中，**数组**代表 php 中的一个普通数组，意味着 SQL 返回了一条记录；**二维数组**也是普通数组，只不过每一个元素都是数组，用 **foreach** 函数遍历，意味着 SQL 返回了一条或多条记录。**多数组**代表 SQL 返回的数组指针，需要用 **mysql_fetch_array** 函数遍历，意味着返回了一条或多条记录。

（1）Base_Class 类

方法	参数	返回值	功能
get_by_id($id)	数值	数组或 false	查找主键等于 id 的记录
update_by_id($para, $id)	关联数组; 数值	true 或 false	更新主键等于 id 的记录，更新内容为$para 数组
delete_by_id($id)	数值	true 或 false	删除主键值等于 id 的记录
insert($para)	关联数组	true 或 false	插入一条记录
set_select($what)	字符串		设置 simple mode 下默认字段
set_default()			恢复 simple mode 字段默认值
simple_mode()			切换到 simple mode 模式
complex_mode()			切换到 complex mode 模式
set_order($order)	字符串		设置 order by 语句
clear_order()			清除 order by 语句

以下描述中省略 Base_Class 中类的方法（包括被重写的方法）

（2）User 类 （user 表）

方法	参数	返回值	功能
get_by_login($login)	字符串	数组或 false	查找 user_login 等于$login 的记录（根据邮箱查找）
get_by_stuid($stuid)	字符串	数组或 false	查找 user_stuid 等于$stuid 的记录（根据学号查找）
login_by_email($login, $pass)	字符串（邮箱）; 字符串（密码）	true 或 false	根据邮箱登录，验证密码是否正确
login_by_stuid($stuid, $pass)	字符串（学号）; 字符串（密码）	true 或 false	根据学号登录，验证密码是否正确

（3） iClass 类 （class 表）

class 是 php 中标识符，所以只能定义成 iClass 类

方法	参数	返回值	功能
get_by_name($name)	字符串	多数组或 false	查找 class_name 中含有 $ name 的记录（根据班名查找）
get_by_subhead($name)	字符串	多数组或 false	查找 class_subhead 中含有 $ name 的记录（根据教师姓名查找，subhead 字段一般存储教师姓名）
get_by_parent($id)	数值	多数组或 false	查找 class_parent 等于$ id 的记录（查找班级 id 的子班级）
get_by_status($id)	数值	多数组或 false	查找 class_status 等于$ id 的记录（查找班级状态等于 $id 班级）
get_by_level($level)	数值	多数组或 false	查找 class_level 等于 $ level 的记录（查找班级 level 等于$level 班级）

（4） User_Class 类 （user_class 表,学生-班级表，存储学生选班情况）

方法	参数	返回值	功能
get_by_user_id($id)	数值	多数组或 false	查找 user_id 等于$ id 的记录（查找用户的选班情况）
get_by_class_id($id)	数值	多数组或 false	查找 class_id 等于$ id 的记录（查找班级中的用户）
get_by_parent($id)	数值	多数组或 false	查找 class_parent 等于$ id 的记录（查找班级 id 的子班级）
get_by_status($id)	数值	多数组或 false	查找 class_status 等于$ id 的记录（查找班级状态等于 $id 班级）
get_by_level($level)	数值	多数组或 false	查找 class_level 等于 $ level 的记录（查找班级 level 等于$level 班级）

（5） Submit 类 （submit 表,存储学生提交作业情况）

方法	参数	返回值	功能
get_by_user_id($id)	数值	多数组或 false	查找 user_id 等于$ id 的记录（查找某用户提交的所有作业）
get_by_post_id($id)	数值	多数组或 false	查找 post_id 等于$ id 的记录（查找提交某作业的所有用户）
count_inc($user, $post)	数值；数值	多数组或 false	给用户$user 提交$post 作业的提交次数加一

（6） Log 类 （log 表,存储登录记录）

方法	参数	返回值	功能
get_last_by_user($user)	数值	数组或 false	查找用户$user 最后一次登录记录
get_by_user($user)	数值	多数组或 false	查找用户$user 的所有登录记录
login($user, $ip, $agent)	数值；数值；字符串	true 或 false	增加一条登录记录。用户 id 为$user，ip 为整型$ip，浏览器信息$agent

（7） Post 类 （post 表,存储作业）

方法	参数	返回值	功能
get_by_user($user)	数值	多数组或 false	查找用户$user 发布的所有作业
get_by_class($class)	数值	多数组或 false	查找班级$class 的所有作业
get_by_parent($id)	数值	多数组或 false	查找作业$id 的子作业
get_list($user)	数值	二维数组或 false	查找用户$user 的所在班级的所有作业（也就是查找与用户有关的作业）
new_post($post_user, $post_class, $post_deadline, $post_type, $post_title, $post_subhead, $post_content)	数值；数值；日期(整型)；字符串；字符串；字符串；字符串	true 或 false	发布一个新作业。参数分别是发布者用户号、班级、截止日期、文件类型、标题、副标题、正文内容。

（8） Comment 类 （comment 表,存储评论、消息记录）

评论是发表在作业上的，**消息**是系统发送给用户，或用户发送给用户的。下表中一律用**评论**代替。

方法	参数	返回值	功能
get_by_user($user)	数值	多数组或false	查找用户$user 发表的所有评论
get_by_post($post)	数值	多数组或false	查找发表在作业$post 的所有评论
get_by_parent($id)	数值	多数组或false	查找评论$id 的子评论
good($comment_id, $user_id)	数值；数值	true 或 false	赞评论。参数分别是评论号、点赞的用户号。将点赞记录写入 Rec 表。
good_cancel($comment_id, $user_id)	数值；数值	true 或 false	取消赞。参数同上。删除 Rec 中的点赞记录。
bad($comment_id, $user_id)	数值；数值	true 或 false	弱评论。参数同上。将点弱记录写入 Rec 表。
bad_cancel($comment_id, $user_id)	数值；数值	true 或 false	取消弱。参数同上。删除 Rec 中的点弱记录。
new_comment($post, $user, $content)	数值；数值；字符串	true 或 false	发表评论。参数分别是作业号、用户号、评论内容。给作业发布者发送通知。（写 Noti 表）
reply($comment_id, $user, $content)	数值；数值；字符串		回复评论。参数同上。给作业发布者、被回复者发送通知。（写 Noti 表）

oh！发送消息的方法还没写，以后再补充吧。

（9） Rec 类 （rec 表,存储点赞、点弱记录，防止用户无限次点赞/弱）

方法	参数	返回值	功能
new_rec($user, $comment, $type)	数值；数值；数值	true 或 false	增加一条记录。参数分别是用户号、评论号、类型。类型：1 为赞，2 为弱。

（10） Noti 类 （noti 表,存储通知记录，比如有人评论你、有人点赞。oh！有人点赞发送通知还没写，以后再说吧）

方法	参数	返回值	功能
get_by_user($user)	数值	多数组或 false	查找用户$user 的所有通知
new_noti($comment, $user, $type)	数值；数值；数值	true 或 false	增加新通知。参数分别是评论号、用户号、类型。类型：0 为评论，1 为赞，2 为弱，3 为消息
read_noti($noti_id)	数值	true 或 false	将这条通知设为已读

（11） User_Meta 类 （usermeta 表,存储用户辅助信息）

meta 代表元，也就是元信息，是 key-value 对，适合进行字段扩展，用于存储辅助信息。不一定常用。比如用户头像、个性签名、绑定手机可以放在此表中。

方法	参数	返回值	功能
get_by_meta_id($id)	数值	数组或 false	查找 meta_id 等于$id 的记录
get_by_user_id($id)	数值	多数组或 false	查找某用户的所有 meta 信息
get_by_meta_key($user_id, $meta_key)	数值；字符串	数组或 false	查找某用户的 key 为$meta_key 的 meta 信息
update_by_meta_key($para, $user_id, $meta_key)	数值；字符串	true 或 false	更新某用户的 key 为$meta_key 的 meta 信息
delete_by_meta_key($user_id, $meta_key)	数值；字符串	true 或 false	删除某用户的 key 为$meta_key 的 meta 信息
new_meta($user_id, $meta_key, $meta_value)	数值；字符串；字符串；	true 或 false	新增一条 meta。

剩下没介绍的类是：Comment_Meta、Post_Meta、Meta 和 Option，分别是**评论辅助、作业辅助、全局辅助、全局设置**。都是 key-value 对的形式，方法也都类似，一看源代码就懂，所以不再赘述。

5. 版本与声明

5.1 版本

本文档名称：《Sedge 数据访问层(DAL)说明文档》

版本：2014.2.4

5.2 声明

本文档以**最新版**为准，最新版发布在 github.com/ng1091/Sedge 及 Sedge 开发 QQ 群，群号 185288022。