whmcs 系统下的 CURD 操作

作者：liumapp
原文链接：https://ld246.com/article/1496906573985
来源网站：链滴

论如何通过whmcs自带的模块进行数据库交互

前提

首先，我们并不需要额外的去建立对数据库的连接，因为大多数情况下（即使是在插件下面），whmc都已经帮我们把数据库的连接建立起来了，所以只需要直接包含跟数据库进行交互的类即可。

```
use WHMCS\Database\Capsule;
```

Query

在whmcs系统下，因为它是直接将Laravel的数据库模块独立出来，所以我们也可以直接按照Laravel面的Query来写，Laravel的相关文档：Laravel’ s query documentation，下面直接上代码：

批量查询、再更具条件来更新用户表的用户名称:

```
<?php

use WHMCS\Database\Capsule;

// Print all client first names using a simple select.

/** @var stdClass $client */
foreach (Capsule::table('tblclients')->get() as $client) {
    echo $client->firstname . PHP_EOL;
}

// Rename all clients named "John Deo" to "John Doe" using an update statement.
try {
    $updatedUserCount = Capsule::table('tblclients')
        ->where('firstname', 'John')
        ->where('lastname', 'Deo')
        ->update(
            [
                'lastname' => 'Doe',
            ]
        );

    echo "Fixed {$updatedUserCount} misspelled last names.";
} catch (\Exception $e) {
    echo "I couldn't update client names. {$e->getMessage()}";
}
```

order和limit

```
Capsule::table('tblknowledgebase')->select('order','title' , 'article')->orderBy('order' , 'desc')->
kip(0)->take(6)->get()
```

Schema

原文链接：whmcs 系统下的 CURD 操作

使用schema方法，我们可以通过脚本来进行表格的创建，这个在写模块的时候经常会用到。

```
<?php

use WHMCS\Database\Capsule;

// Create a new table.
try {
    Capsule::schema()->create(
        'my_table',
        function ($table) {
            /** @var \Illuminate\Database\Schema\Blueprint $table */
            $table->increments('id');
            $table->string('name');
            $table->integer('serial_number');
            $table->boolean('is_required');
            $table->timestamps();
        }
    );
} catch (\Exception $e) {
    echo "Unable to create my_table: {$e->getMessage()}";
}
```

PDO

基本通过query就可以实现所有CURD操作，但是呢，实际应用过程中，还要考虑执行sql失败的情况这个时候就需要用到回滚操作了。

下面来一个插入数据，如果失败执行回滚的例子：

```
<?php

use WHMCS\Database\Capsule;

// Perform potentially risky queries in a transaction for easy rollback.
$pdo = Capsule::connection()->getPdo();
$pdo->beginTransaction();

try {
    $statement = $pdo->prepare(
        'insert into my_table (name, serial_number, is_required) values (:name, :serialNumber, :is
equired)'
    );

    $statement->execute(
        [
            ':name' => $_POST['name'],
            ':serialNumber' => $_POST['serialNumber'],
            ':isRequired' => (bool) $_POST['isRequired'],
        ]
    );

    $pdo->commit();
```

```
} catch (\Exception $e) {
    echo "Uh oh! {$e->getMessage()}";
    $pdo->rollBack();
}
```