

# PDO

## 实例化PDO类。

- PDO类的构造方法。
  - `PDO::__construct(`  
    `string dsn,`  
    `string username,`  
    `string password`  
    `)`
- 什么是\$dsn?
  - 例子：
    - `$dsn = 'mysql:host=localhost;dbname=myframe;charset=utf8';`
  - 语法：
    - PDO驱动名称 + ':' + PDO驱动程序特有的连接语法。
    - 通过示例来理解。
      - PDO驱动名称: `mysql`;
      - PDO驱动程序特有的连接语法（即mysql特有的连接语法）: `host(主机名), dbname(数据库名称), charset(字符集)`。（注意是用';'分割）
- 实例化PDO类。
  - `$dsn = 'mysql:host=localhost;dbname=myframe;charset=utf8';`
  - `$pdo = new PDO($dsn,$username,$password);`

## 执行SQL语句的两个方法。

- `query();`
  - 作用。
    - 解析结果集。
    - 实现预处理。
    - 事务处理。
  - 返回值。
    - 成功返回PDOStatement类的对象。
- `exec();`
  - 作用。
    - 实现数据的增删改。
  - 返回值。
    - 成功返回受影响行数。

## 处理结果集的三个方法。

- fetch();
  - 从结果集中获取下一行记录。
  - 常用参数：PDO::FETCH\_ASSOC;(用于返回关联数组)
  - 用法：

```
$res = $pdo->query($sql)
while($data = $res->fetch()){
    ...
}
```

- 相对应的mysql函数：mysql\_fetch\_row();
- fetchColumn();
  - 用于获取结果集中的单独一列。
  - 语法：
    - PDOStatement::fetchColumn([int \$column\_number];
    - 参数：行中列的索引号，默认从0开始。（可选）
  - 返回值：
    - 成功：返回单独的一列。
    - 返回：false;
  - 用法：

```
$res = $pdo->query($sql)
while($data = $res->fetchColumn(1)){
    echo $data.'|';
}
```

• Allen|James|Rose|aaa|小李|Andy|Bruce|张三|

- fetchAll();
  - 获取结果集中所有的行。
  - 常用参数：PDO::FETCH\_ASSOC;(用于返回关联数组)
  - 相对应的mysql函数：mysql\_fetch\_array();

## 预处理机制。

- 为啥要用预处理机制？
  - 效率更高。
    - 传统方式：将发送的数据和SQL写在一起，每条SQL都需要经过分析、编译和优化的周期。
  - 编译的是模板而不是具体的SQL语句。也就是说进行了抽象。

- 预先编译一次SQL模板，在操作时，发送相关数据即可完成对应的操作。
  - 无需考虑数据中的特殊字符（如单引号。）
- 常用方法。
  - prepare();方法。
    - 作用：用于准备预处理的SQL语句。
    - 语法：
      - PDO::prepare(\$sql);
    - 返回值：
      - 成功：PDOStatement类对象。
      - 失败：false。
  - execute();方法。
    - 作用：用于执行一条预处理语句。
    - 语法：
      - bool PDOStatement::execute([array \$input\_params]);
    - 返回值：布尔值。
  - 预处理例子。

```
<?php
$dsn = 'mysql:host=localhost;dbname=myframe;charset=utf8';
$pdo = new PDO($dsn,'root','root');
<!-- SQL语句模板 -->
$sql = 'INSERT INTO student(`name`,`gender`) VALUES(?,?)';
<!-- 准备预处理的SQL语句 -->
$stmt = $pdo->prepare($sql);
<!-- 执行一条预处理语句 -->
$stmt->execute(['张三',1]);
$stmt->execute(['李四',1]);
$stmt->execute(['小红',0]);
```

为占位符绑定数据。

- bindParam();方法。
- bindValue();方法。
- 以后再写。

PDO错误处理。

- PDOException类。
  - 三种错误处理模式

模式	返回	代码
----	----	----

模式	返回	代码
SILENT模式 (默认)	只是简单地设置错误代码	PDO::ERRMODE_SILENT
WARNING模式	错误代码+WARNING信息	PDO::ERRMODE_WARNING
EXCEPTION模式	在错误发生时抛出相关异常 (较为实用)	PDO::ERRMODE_EXCEPTION

- 设置异常处理模式。

```
<!-- 设置为SILENT模式-->
$pdo->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_SILENT);
<!-- 设置为WARNING模式-->
$pdo->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_WARNING);
<!-- 设置为EXCEPTION模式-->
$pdo->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);
```

野火烧不尽，春风吹又生。