

# DB类

## 实例化自身的方法。

- 运用了单例模式。
  - 对数据的操作一定是很频繁的，为了减少一个类实例化的次数，减少资源消耗，采用了单例模式的做法。
  - 代码。

```
<?php
protected static $instance;
public static $config = [];
public static function getInstance(){
    if(!self::$instance){
        self::$instance = new static(static::$config);
    }
    return self::$instance;
}
```

## 初始化基本配置。

- 使用静态方法初始化基本配置。
- 原因：
  - 在实例化DB类的时候就需要传入基本配置，也就是说设置基本配置要在DB类实例化之前完成，所以必须使用静态方法。另外，基本配置从配置文件中获取。
  - `self::$instance = new static(static::$config);`

```
public static function getInstance(array $config){
    static::$config = $config;
}
```

## 初始化PDO类的方法。

- 和正常实例化PDO类的步骤相同，但是要使用try-catch结构。
  - 设置dsn;
  - `new PDO(dsn,username,pwd);`
  - 设置抛出异常的模式。
    - `$this->pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);`

```

public function initPDO()
{
    // 取出数据库连接信息。
    $type = $this->config['type'];
    $host = $this->config['host'];
    $dbname = $this->config['dbname'];
    $charset = $this->config['charset'];
    $dsn = "$type:host=$host;dbname=$dbname;charset=$charset";
    try {
        $this->pdo = new PDO($dsn, $this->config['username'], $this->config['password']);
    } catch (PDOException $e) {
        $error = $this->errorMsg($e);
        throw new Exception($error);
    }
    $this->pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
}

```

## 封装查询类操作。

- 查询类是指SELECT有结果集的操作。
- 查询一条数据(fetch();方法)。

```

public function fetchRow($sql, array $bind=[])
{
    try {
        $stmt = $this->pdo->prepare($sql);
        $stmt->execute($bind);
        return $stmt->fetch(PDO::FETCH_ASSOC);
        // while($data = $stmt->fetch(PDO::FETCH_ASSOC)){
        //     var_dump($data);
        //     echo '<hr/>';
        // }
    } catch (PDOException $e) {
        $error = $this->errorMsg($e, $sql);
        throw new Exception($error);
    }
}

```

- 查询所有数据(fetchAll();方法)。

```

public function fetchAll($sql, array $bind=[])
{
    try {
        $stmt = $this->pdo->prepare($sql);
        $stmt->execute($bind);
        return $stmt->fetchAll(PDO::FETCH_ASSOC);
    } catch (PDOException $e) {
        $error = $this->errorMsg($e, $sql);
        throw new Exception($error);
    }
}

```

## 封装执行类操作。

- 执行类是指INSERT,UPDATE,DELETE没有结果集的操作。
- 代码。

```

public function execute($sql, array $bind = [])
{
    try {
        $stmt = $this->pdo->prepare($sql);
        $stmt->execute($bind);
        return $stmt->rowCount();//返回受影响行。
    } catch (PDOException $e) {
        $error = $this->errorMsg($e, $sql);
        throw new Exception($error);
    }
}

```

## \_\_DIR\_\_ 和 \_\_FILE\_\_。

- \_\_DIR\_\_
  - 用于获取脚本文件所在的**目录**。
- \_\_FILE\_\_
  - 取得当前文件的**绝对地址**。