# SQL注入

如果SQL中，存在浏览器端请求的数据（用户数据），用户通过特殊的数据形式，对SQL的行为作出影响，称之为SQL注入。（SQL injection）

"SELECT \* FROM `p34\_admin` WHERE admin\_name='$admin\_name' and admin\_pass=md5('$admin\_pass')"

正常数据：

SELECT \* FROM `p34\_admin` WHERE admin\_name='admin' and admin\_pass=md5('1234abcd')

SELECT \* FROM `p34\_admin` WHERE admin\_name='wrong' and admin\_pass=md5('abcd1234')

非正常数据：

‘ or 1#

SELECT \* FROM `p34\_admin` WHERE admin\_name='' or 1#' and admin\_pass=md5('afdadfa')

检索 条件永远为真

‘ or 1 or ‘

SELECT \* FROM `p34\_admin` WHERE admin\_name='' or 1 or '' and admin\_pass=md5('')

Tip:

不仅仅在登录时，可以被注入，任何用户数据参与执行，都可以被注入。

## 预防

业务逻辑上预防(尽可能使用白名单，将数据限制在合理的范围之内)

通过限定数据格式（比如用户名仅仅由字母数字和特定的字符组成），如果检测不满足要求，则直接不予参与SQL注入问题。

通过限定数据类型（比如，ID需要整型的数据，将用户传递请求数据强制转换成整型），进行避免。

Delete from `match` where m\_id=$\_GET[‘m\_id’]

(int) $\_GET[‘m\_id’] = 0 or 1#

特殊数据转义：

通过函数：

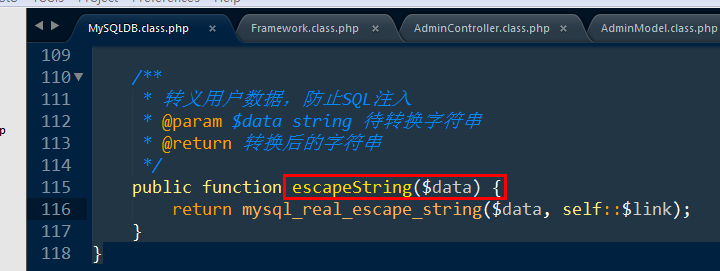
PHP字符串函数： addslashes()

MySQL提供数据转义函数：mysql\_real\_escape\_string(待字符串，连接)

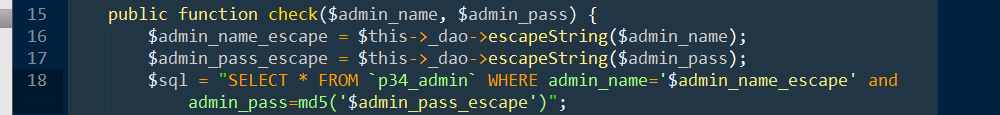


## 项目中添加防止SQL注入的代码

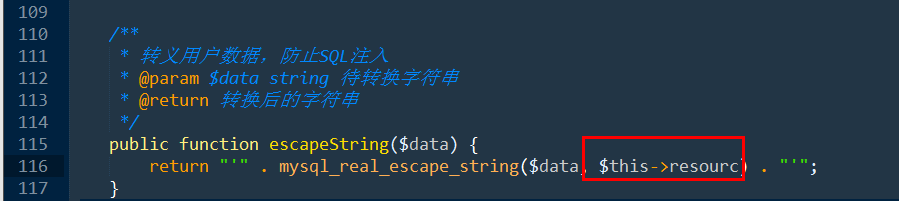
在MySQLDB中，添加一个可以完成转义的方法，在模型中，需要时调用！



项目中使用，AdminModel中，完成转换：



为了避免，整型数据可能不被SQL中增加引号，强制在转换后的数据使用引号包裹；

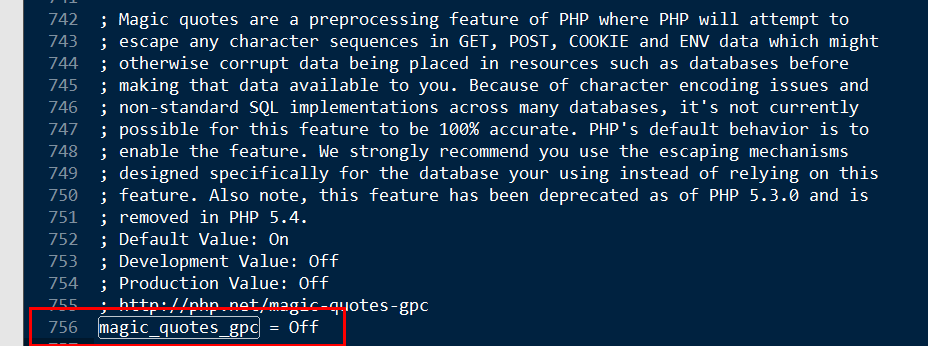


拼凑SQL的数据不需要增加引号:

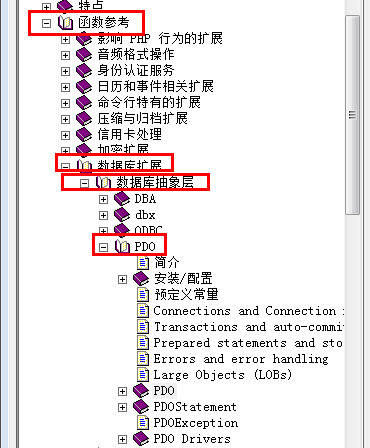


Tip:

PHP中魔术引号（magic quotes）,PHP中自动为请求数据（GET，POST）增加转义的一种防止SQL注入机制。

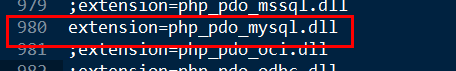


# PDO\_MySQL操作



PHP新版中推荐的基于面向对象操作数据库服务器的方式！

开启PDO\_mysql相关扩展：



Apache restart

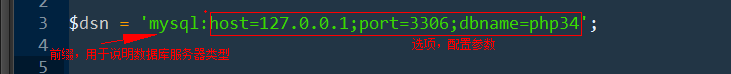
基本使用步骤

连接，认证，发送SQL，等待mysql服务器的执行结果，处理执行结果

## 实例化PDO对象

构造方法需要4个参数：

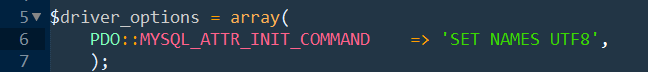
$dsn: Data Source Name数据源名称，需要的数据来源的描述



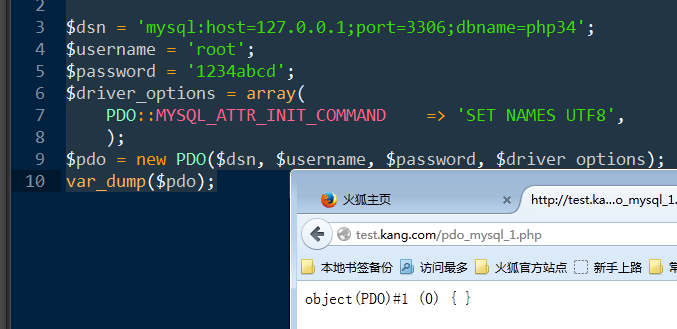
$username

$password

$driver\_options 驱动选项，对应当前数据库服务器一些特定的初始化工作。



测试：



## 发送SQL到服务器等待执行SQL的结果

Pdo对象方法query(SQL)完成执行SQL



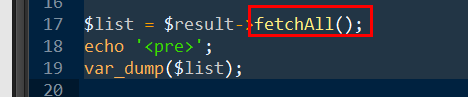
## 处理执行结果



逻辑上，返回的是结果集，被PDOStatememt类对象封装管理。

Tip：PDOStatement类，称之为PDO语句对象，SQL执行完（处理完）产生的结果对象。

利用该结果对象，获得我们需要的数据：



默认返回关联和索引的集合，可以传递参数，进行配置：



## PDO对象的常用方法

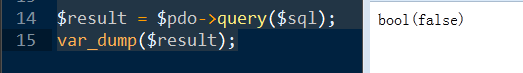
### errorInfo()

错误信息，包含错误信息的数组。

### errorCode()

错误代码

当SQL执行过程中，失败，会产生错误：





### beginTransaction();

开启事务

### rollback();

回滚事务

### commit();

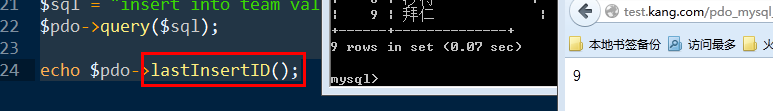
提交事务

### inTransaction();

判断是否处于事务中

### lastInsertID()

最后形成的auto\_increment字段的值。



### Exec()

又一个执行SQL的方法。

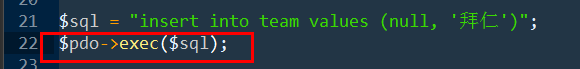
区别于query(),返回值类型不同。Exec()返回值类型为整型。(query()返回的是对象类型)。

该返回的整数，表示当前执行SQL所影响的记录数。

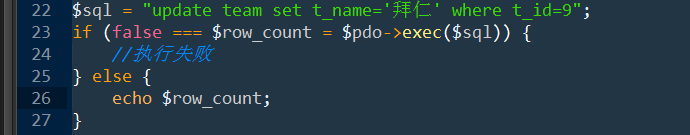
可见：

Query(), 执行查询类，show，select，desc

Exec(), 非查询类，insert，delete，update，DDL



注意，影响的记录数可能为0.但是不意味着执行失败！



# 项目中对PDO的使用

使用PDO，实现与mysqlDB这个DAO对象外部操作一致的另外一种DAO的实现，可以做到，模型切换DAO，是不需要更改模型内的任何代码（热插拔，无缝切换）

增加DAO的操作接口（I\_DAO），使所有DAO的定义（mysql扩展，PDO扩展）都实现相同的DAO操作接口（I\_DAO）

