

## 07. 连接数据库与模型初探

学习要点：

1. 连接数据库
2. 模型初探

本节课我们来了解一下数据库的连接方式，通过参数的配置连接 MySQL 数据库。

### 一. 连接数据库

1. ThinkPHP 采用内置抽象层将不同的数据库操作进行封装处理；
2. 数据抽象层基于 PDO 模式，无须针对不同的数据库编写相应的代码；
3. 使用数据库的第一步，就是连接你的数据库；
4. 在根目录的 `config` 下的 `database.php` 可以设置数据库连接信息；
5. 大部分系统已经给了默认值，你只需要修改和填写需要的值即可；
6. 本地测试，会优先采用 `.env` 的配置信息，我们和 `database` 配置对应上即可；

```
'type'           => Env::get('database.type', 'mysql'),
'hostname'       => Env::get('database.hostname', '127.0.0.1'),
'database'       => Env::get('database.database', 'grade'),
'username'       => Env::get('database.username', 'root'),
'password'       => Env::get('database.password', '123456'),
'hostport'       => Env::get('database.hostport', '3306'),
'charset'        => Env::get('database.charset', 'utf8'),
'prefix'         => Env::get('database.prefix', 'tp_');
```

7. 可以通过删除改变 `.env` 的配置，或删除 `.env` 来验证 `database` 的执行优先级；
8. 在 `database.php` 配置中，`default` 表示设置默认的数据库连接；
9. `connections` 配置数据库连接信息，可以是多个数据库，便于切换；
10. 默认的数据库连接名称为：'`mysql`'，再复制一组数据库链接信息：'`demo`' 切换；
11. 创建一个用于测试数据连接的控制器：`DataTest.php`；

```
namespace app\controller;
use think\facade\Db;

class DataTest
{
    public function index()
    {
        $user = Db::table('tp_user')->select();
        //$user = Db::connect('mysql')->table('tp_user')->select();
        return json($user);
    }
}
```

```
public function demo()  
{  
    $demo = Db::connect('demo')->table('tp_user')->select();  
    return json($demo);  
}
```

12. 暂时不必理解这里的代码，只要运行起来看到效果即可，后面章节会详解；

## 二. 模型初探

1. 在 app 目录下创建一个 model 目录，并创建 User.php 的模型类；

```
namespace app\model;  
use think\Model;  
  
class User extends Model  
{  
    protected $connection = 'demo';  
}
```

2. User 继承模型基类，即可实现数据调用，后面章节会详解讲解；
3. 而受保护的字段 \$connection，则是切换到 demo 数据库；
4. 控制器端的调用方式如下，具体原理，后面模型章节会详解；

```
public function getUser()  
{  
    $user = User::select();  
    return json($user);  
}
```